INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
ISSN 1841-9836, 11(3):381-393, June 2016.

Increasing Face Recognition Rate

B. Lagerwall, S. Viriri

Brett Lagerwall
Serestina Viriri*
School of Mathematics, Statistics and Computer Science
University of KwaZulu-Natal, Westville Campus
Durban, South Africa
*Corresponding author: viriris@ukzn.ac.za

Abstract: This paper describes and discusses a set of algorithms which can improve face recognition rates. These algorithms include adaptive K-Nearest Neighbour, adaptive weighted average, reverse weighted average and exponential weighted average. Essentially, the algorithms are extensions to the basic classification algorithm used in most face recognition research. Whereas the basic classification algorithm selects the subject with the shortest associated distance, the algorithms presented in this paper manipulate and extract information from the set of distances between a test image and the training image set in order to obtain more accurate classifications. The base system to which the algorithms are applied uses the eigenfaces technique for recognition with an adapted Viola and Jones algorithm for face extraction. Most of the algorithms proposed show a consistent improvement over the baseline test.
Keywords: Face Recognition, Eigenfaces, Classification Algorithms, Weighting Algorithms.

1 Introduction

The process of obtaining the identity of a person from an image can be successfully performed by only looking at his/her face. This typically involves a face detection stage and a face recognition stage (taking detected faces and classifying them using an already existing database of faces).

The full potential of face recognition applications has not been realized, since most suffer from an inability to handle light and pose variations [1]. Yet, face recognition remains an important topic in computer vision because of the large number of real-world scenarios it can be applied to. Senior and Bolle [1] state that the three main application domains for face recognition are access control, identification systems and surveillance. It is evident that face recognition systems falling into these three domains have in fact become a part of our every day life. Example usage cases of the three domains are: access control systems in security environments (e.g. banking), identification systems in websites such as Facebook and surveillance systems which compare faces against a threat list in sporting events.

Moreover, humans are becoming increasingly reliant on face recognition systems to achieve day to day tasks. Hence, researchers continue to focus on methods of improving the current recognition rates.

The rest of the paper is structured as follows: *Section 2* explores the state-of-the-art of face recognition, *Section 3* gives a brief overview of preprocessing, *Section 4* describes an overall face recognition system, *Section 5* explores the novel classification techniques, *Section 6* presents experimental results and discussion; and *Section 7* draws the conclusions and future work.

2 Related Work

Lin [2] shows that face recognition is generally broken down into two modules. These are:

- A feature extractor which transforms the pixels of a face into a useful vector representation.
- A pattern recognition module which searches the database to find the best match for the inputted face.

However, it must be noted that the generic face recognition framework shown by Lin [2] is only concerned with classifying a face. A large number of algorithms will pre-process the face images before any classification occurs.

One of the earliest techniques used in face recognition was performed by Turk and Pentland [3]. They used principal component analysis (PCA) to encode facial images in what they call an information theory approach. Basically, all of the faces in a face database are manipulated to form a set of eigenvectors (eigenfaces in face recognition literature [4]). Each of the original faces can be reconstructed via linear combinations of the eigenvectors.

For classification, Turk and Pentland [3] reconstruct the input image by applying weights to each of the eigenvectors. The vector of weights for the input image is compared against the respective weight vectors for each of the images in the database. The subject in the input image is classified to be the subject in the database with the closest match.

Moon and Phillips [4] attempt to improve on the eigenfaces technique by introducing various ideas and optimizations. Most notably, they try using different nearest-neighbour distance classifiers. However, recognition rates for frontal, upright facial images (used in this study) do not seem to dramatically increase from the baseline PCA algorithm developed by Turk and Pentland [3].

Another face recognition system using an information theory approach was created by Liu and Weschler [5]. Their system uses independent component analysis (ICA) instead of PCA, since ICA provides a more powerful data representation. Improvements range between 0 and 4 percent depending on the number of features used [5].

Face recognition algorithms which do not use the information theory (or encoding of information) approach are less common. However, Bronstein et al. [6] attempted to use a 3D model of a face in order to overcome weaknesses suffered by 2D systems (head orientation and facial expressions). Other common methods of performing face recognition involve using physiological biometrics.

Zhang et. al. [15] compared and contrasted the sparse representation [18] and collaborative representation techniques ([16], [20]) for face recognition. The sparse representation based classification first codes a testing sample as a sparse linear combination of all the training samples, and then classifies the testing sample by evaluating which class leads to the minimum representation error. While the collaborative representation technique is based on the regularized least square. The experimental results show that both collaborative representation and sparse representation achieved accuracy rate of 93.7% on AR database. Furthermore, collaborative representation has significantly less complexity than sparse representation based classification.

Hao et. al. ([17], [21]) proposed a novel method, called heteroscedastic sparse representation based classification which addresses the complexity problem of sparse representation based classification. In the presence of noises, the sparse representation based classification model exists heteroscedasticity, which makes residual estimation inefficient. Therefore, heteroscedastic correction must be carried out for homoscedasticity by weighting various residuals with heteroscedastic estimation. The experimental results show that heteroscedastic sparse representation based classification has significantly less complexity than sparse representation based classification, while it is more robust.

Robust face recognition via sparse representation [18] considered the problem of automatically recognizing human faces from frontal views with varying expression and illumination, as well as occlusion and disguise. They cast the recognition problem as one of classifying among multiple linear regression models, and argue that new theory from sparse signal representation offers the key to addressing this problem. This new framework provides new insights into two crucial issues in face recognition: feature extraction and robustness to occlusion.

Furthermore, Wagner et. al. [19] proposed a conceptually simple face recognition system that achieves a high degree of robustness and stability to illumination variation, image misalignment, and partial occlusion. The system uses tools from sparse representation [18] to align a test face image to a set of frontal training images. They demonstrated how to capture a set of training images with enough illumination variation that they span test images taken under uncontrolled illumination. Their system can efficiently and effectively recognize faces under a variety of realistic conditions, using only frontal images under the proposed illuminations as training.

Many of the face recognition methods reviewed used different nearest-neighbour distance classifiers to calculate the distance between a test image and a training image. However, once the distances were calculated, the test image with the smallest distance was always selected to be the best match. This paper takes the classification process one step further by analyzing the results which can be achieved after post-processing the set of calculated distances.

3 Preprocessing

Before performing face recognition, a region of interest (ROI) within the image must be extracted. A modified version of the Viola and Jones [7] face detection algorithm was chosen for this purpose. The key points and modifications are mentioned below.

3.1 Classifier

A drawback to building a classifier is the sheer amount of training data required (5000 facial images and 10000 non-facial images in Viola and Jones [7]). Thus, an already available classifier is used: the haarcascade_frontalface_default classifier provided with OpenCV [8].

3.2 Passing the Classifier

The classifier being used consists of a number of stages and each stage consists of a number of features. At a particular stage, a sub-window can only pass that stage if the sum of the values generated by testing various features (specified in the haarcascade_frontalface_default classifier) is above a certain threshold. For the sub-window to be considered as a face, it must pass all 25 stages.

3.3 Merging Detections

Viola and Jones [7] said that they merged all overlapping detections. However, if two faces are close together this could accidentally be counted as a single detection. Instead, a merging algorithm partially based on work done by Rowley et al. [?] was constructed. For each pair of rectangles i and j, the rectangles can be said to be detecting the same face if both (1) and (2) hold.

$$euclideanDistance(c_i, c_j) \leq t \times width(i) \tag{1}$$

$$euclideanDistance(c_i, c_j) \leq t \times width(j) \tag{2}$$

where c_i is the centre of rectangle i, c_j is the centre of rectangle j and t is a threshold chosen to be 0.2 [9].

(a) Ghost face 1 (b) Ghost face 5 (c) Ghost face 50

Figure 1: Showing example ghost faces.

4 Face Recognition

The eigenfaces technique is used for face recognition. The key advantages of this algorithm are its quick classification of the probe/test image and its ease of implementation.

4.1 Training

Recognition must be performed against a base set of known images. This set is typically called the training set. On each image in the training set, illumination normalization is performed using:

$$image_i = \frac{image_i}{max} \tag{3}$$

where $image_i$ is the ith pixel of the image and max is the largest value of any of the pixels in the image. Then, the basic steps of the eigenfaces algorithm from Turk and Pentland [3] are performed on the training set, involving:

- Mapping the pixel intensities of an image onto a face vector. All of the face vectors from the training set are then placed into a matrix.
- Performing various matrix operations, including eigenvalue decomposition, to obtain eigenfaces or ghost faces (see Figure 1).
- Calculating a weight matrix, where each weight represents the amount to which an eigenface counts towards making up an image.

4.2 Classification

When a testing image is inputted, the algorithm must be able to classify the face in the image as one of the subjects in the database. Again, the steps described in Turk and Pentland [3] are performed on the testing image, eventually resulting in a matrix of weights specifying the contribution of each of the eigenfaces in making up the test image.

These weights are then compared to the weights of the training images/database images. A distance classifier (Euclidean distance in this research) is used to determine the distance between the test image and each of the training images. Typically, the person in the test image is classified to be the person in the training set who has the smallest distance between their training image and the testing image.

5 Novel Classification Algorithm – Introduction

In traditional face recognition algorithms, a subject in a test image is classified as the subject in an image in the database against which it has the shortest distance. More specifically, if all of the distances calculated between the n training images in the database and the test image are sorted and thus ranked from smallest to biggest, the subject in a test image is classified as the subject in a training image that corresponds to the first distance (or shortest distance) in the sorted list.

If a database contains only one image of each subject, then taking the top match is the only logical classification. However, if a database contains multiple images per subject, then a number of other classification possibilities occur. Described below are some of the classification algorithms proposed which can improve the recognition rates.

Many of the algorithms described rely on the same mathematical foundations. Thus, let us define the following:

n	number of images
m	number of subjects
k	a percentage of the images in the sorted list
$k\prime$	a percentage of the images of a subject
S_x	the set of training images of subject x
$S_{xk\prime}$	the set containing the top $k\prime$ percent of training images of subject x
$\lvert S_{xk}\rvert$	the number of images of subject x in the top k percent
I_j	the distance between the j_{th} training image and the testing image
w_j	the weight applied to the j_{th} training image

Note that where applicable, the algorithms will use a k percent of images, rather than k images. This is because using a percentage allows for better scaling when subjects in the same database have a different number of images.

Novel Classification Algorithm – Algorithm Description

5.1 Top Match

The test image is classified as the top (or first) image in the sorted list. This algorithm is used as the baseline for comparisons.

5.2 Top-k Average

The top k percent of images from the sorted list are chosen. Within the chosen images, the average distance from the images of each subject to the test image is calculated. The subject with the smallest average is selected. More specifically, the subject with the smallest value defined by (4) is chosen.

$$SubjectAvgK_x = \frac{\sum_{j=1}^{nk} I_j}{|S_{xk}|} \quad I_j \in G \tag{4}$$

where G is defined by:

$$G = \{I_p | p \in S_x\} \tag{5}$$

5.3 Adaptive K-nearest Neighbour

The K-nearest Neighbour algorithm is commonly used to solve pattern recognition problems. However, its performance suffers when subjects have differing numbers of training images. The Adaptive K-nearest Neighbour attempts to rectify this.

The top k percent of images from the sorted list are chosen. The total number of images of each subject within the top k percent of images is obtained. This total within k for each subject is then divided by the total number of images of the subject in the database – thus forming a ratio of the subject which falls within the k images. The subject with the highest ratio defined by (6) is chosen.

$$SubjectRatio_x = \frac{|S_{xk}|}{|S_x|} \tag{6}$$

5.4 $K\prime$ Subject Average

The top $k\prime$ percent of images of each subject are chosen. The average of the distances pertaining to the top $k\prime$ percent of images of a subject is calculated. The subject with the lowest average is selected. It must be re-emphasized that in this algorithm, the $k\prime$ percent is used for each subject, rather than the whole sorted list. The subject with the lowest average defined by (7) is chosen.

$$SubjectAvgK\prime_x = \frac{\sum_{j=1}^{n} I_j}{|S_x|k\prime} \quad I_j \in H \tag{7}$$

where H is defined by:

$$H = \{I_p | p \in S_{xk\prime}\} \tag{8}$$

5.5 Average

The average of all the distances for each subject is calculated. The subject which has the lowest average distance is then selected. Note that this algorithm is equivalent to $K\prime$ Subject Average with $k\prime = 100$. The subject with the smallest value defined by (9) is chosen.

$$SubjectAvg_x = \frac{\sum_{j=1}^{n} I_j}{|S_x|} \quad I_j \in G \tag{9}$$

5.6 Weighted Average

The average distance for each subject is calculated. However, the images belonging to a subject are given a weighting. The image with the largest distance is given a weighting of 1; the image with the second largest distance is given a weighting of 2, etc. Finally, if there are $|S_x|$ images of a particular subject, then the image with the smallest distance is given a weighting of $|S_x|$. The weighted sum can then be calculated using (10):

$$WeightedSum_x = \sum_{j=1}^{n} I_j w_j \quad I_j \in G \tag{10}$$

Table 1: ALGORITHM 1: WEIGHTED AVERAGE

```
sort(images)
for i = 0 to n do
   Add the weight of image i to the appropriate list in an array of lists arr, where each
   subject has its own list
end for
for i = 0 to m do
   w ⇐ arr[i].size()
   for j = 0 to arr[i].size() do
      wSum ⇐ wSum + arr[i].get(j) * w
      w ⇐ w − 1
   end for
   total[i] ⇐ wSum/arr[i].size()
end for
lowestName ⇐ 0
wAvg ⇐ total[0]
for i = 1 to m do
   if total[i] < wAvg then
      lowestName ⇐ i
      wAvg ⇐ total[i]
   end if
end for
return lowestName
```

When calculating the average, instead of dividing by the total number of images belonging to the subject, it is divided by the sum of the weights pertaining to the subject. The total weight can be calculated using (11):

$$TotWeight_x = \frac{|S_x|(|S_x| + 1)}{2} \tag{11}$$

Finally, the weighted average for a particular subject can be calculated using (12):

$$WeightedAverage_x = \frac{WeightedSum_x}{TotWeight_x} \tag{12}$$

The subject with the lowest weighted average is then selected.

See **Algorithm 1** for detailed pseudocode of this algorithm. Other weighting algorithms will follow similar pseudocode, but have a different weighting mechanism.

5.7 Reverse Weighted Average

This algorithm is the same as the Weighted Average algorithm except that the larger weights are applied to the images with larger distances. This algorithm attempts to find out if the less closely matched images are in fact better at distinguishing subjects.

5.8 Exponential Weighted Average

An average of the distances for each subject is calculated. However, each image belonging to the subject is given a weighting. The image of a subject which has the smallest distance is given an arbitrary weighting of 10.0. Subsequent images of the same subject are given weights exponentially less by dividing the previous weight by the $|S_x|^{th}$ root of 10.0. So, if image j has a weight of w_j, then the weight of image $j+1$ can be calculated with (13):

$$w_{j+1} = \frac{w_j}{\sqrt[|S_x|]{10.0}} \tag{13}$$

To find the exponential weighted average, the weighted sum must be divided by the sum of the weights (total weight). The weighted sum and total weight are computed using (10) and (14) respectively.

$$ExpTotWeight_x = \sum_{j=1}^{|S_x|} 10^{\frac{j}{|S_x|}} \tag{14}$$

Finally, the exponential weighted average for a particular subject is calculated using (15):

$$ExpWeightedAverage_x = \frac{WeightedSum_x}{ExpTotWeight_x} \tag{15}$$

The subject with the lowest exponential weighted average is then selected.

6 Results and Discussion

In order to test the validity of results, three publicly available test databases were obtained, namely Caltech Faces, the Georgia Face Database and the ORL Database of Faces. The Feret database ([10], [11]) is perhaps the most commonly used face database to benchmark the eigenfaces technique. However, the Feret database does not contain multiple images of each subject and hence is not useful for testing the novel classification algorithms proposed here.

6.1 Testing Procedure

A standard test procedure was used on each database. This involved the following:

1. Running the Viola and Jones [7] algorithm on each image in the database to extract faces.
2. Manually discarding all subjects from the database which do not have a minimum of 10 separate face detections from the Viola and Jones algorithm. This results in a reduction of the total number of images.
3. Setting aside a number of random images of each subject for testing. In this research, 5 per subject were used.
4. Performing training on the remaining images.
5. Performing the test procedure using the test images and the trained database.
6. Running 100 repetitions of steps 3-5 and taking the average as the final result.
7. Repeating steps 3-6 for each different classification algorithm.

Table 2: Caltech Faces Recognition Rates

Classification Algorithm	Rate (%)
Top Match (Baseline, (*Literature*))	87.71
Top-k Average (k=5)	73.52
Adaptive K-nearest Neighbour (k=5)	77.39
$K\prime$ Subject Average (k=40)	90.14
$K\prime$ Subject Average (k=60)	92.04
$K\prime$ Subject Average (k=80)	91.05
Average	92.63
Weighted Average	94.54
Reverse Weighted Average	72.97
Exponential Weighted Average	93.75

Table 3: Georgia Face Database Recognition Rates

Classification Algorithm	Rate (%)
Top Match (Baseline, (*Literature*))	58.26
Top-k Average (k=5)	51.46
Adaptive K-nearest Neighbour (k=5)	62.13
$K\prime$ Subject Average (k=40)	65.68
$K\prime$ Subject Average (k=60)	69.13
$K\prime$ Subject Average (k=80)	70.32
Average	71.67
Weighted Average	69.87
Reverse Weighted Average	64.67
Exponential Weighted Average	69.01

6.2 Caltech Faces

The Caltech face database, used by Kevenaar et al. [12], was obtained for testing. It contains 450 images of 27 subjects. The database is considered to be relatively easy since the faces are always upright, are uniform in presentation and never suffer from occlusion. Since the database does not provide a consistent number of images per subject, it was deemed necessary to ignore step 2 from the generic testing procedure. Thus, after performing face detection, the database was reduced to 435 images of 25 subjects with between 4 and 26 images per subject.

Table 2 shows that the Weighted Average algorithm improves on the baseline results by close to 7% (from 87.71% to 94.54%). This worked out to an extra 1441 correct recognitions over the 22400 image test. The results also show that the images with a smaller distance to the test image contribute more towards identifying the subject – since both Weighted Average and Exponential Weighted Average perform better than Average. The opposite is not true, with the Reverse Weighted Average (giving higher weightings to less closely matched images) does not perform well. Finally, it was envisaged that Top-k Average and Adapted K-nearest Neighbour would be used with small k values, but they did not improve on the baseline. Using higher k values would only result in the algorithms degenerating towards the Average classification algorithm.

6.3 Georgia Face Database

To determine whether the novel classification algorithms could raise recognition rates in sub-optimal conditions, a more difficult database was required. Thus, the Georgia Face Database,

Table 4: ORL Database of Faces Recognition Rates

Classification Algorithm	Rate (%)
Top Match (Baseline, (*Literature*))	83.82
Top-k Average (k=5)	77.46
Adaptive K-nearest Neighbour (k=5)	82.44
$K\prime$ Subject Average (k=40)	87.08
$K\prime$ Subject Average (k=60)	88.62
$K\prime$ Subject Average (k=80)	88.79
Average	88.24
Weighted Average	89.27
Reverse Weighted Average	81.27
Exponential Weighted Average	89.04

used by Chen et al. [13], was obtained. The database consists of 50 subjects and a total of 750 images, but this was reduced to 30 subjects and 355 images after steps 1 and 2 (pre-processing) in the testing procedure. The database is considered difficult because many of the faces are tilted, have different facial expressions and were taken under different lighting conditions. Further to that, the pictures were taken in separate sessions – thus, some subjects exhibited physical changes (e.g. grew a moustache).

Although the overall results from testing are worse than the Caltech Faces, Table 3 shows that the novel classification algorithms are still effective. In fact the best algorithm (Average) raises the recognition rate from 58.26% to 71.67% – an improvement of over 13%. It is interesting to note that the Average algorithm outperforms both the Weighted Average and Exponential Weighted Average. This is probably because the distance between the better images and the test image is relatively large (since the database is extremely difficult) and thus the distribution of distances for a particular subject is more even. With an even distribution of distance, weighting becomes superfluous.

6.4 ORL Database of Faces

The final test for the novel classification algorithms was to see how well they scaled. More specifically, would they still show improvements over the baseline if the number of subjects was increased? To test this, a third database was required. The ORL Database of Faces, created by Samaria and Harter [14], was obtained. The database consists of 40 subjects with 10 images per subject and all of these images remained intact after steps 1 and 2 of the testing procedure.

Table 4 shows that using $K\prime$ Subject Average, Average, Weighted Average or Exponential Weighted Average as a classification algorithm will give better results than the baseline algorithm. In fact, Weighted Average and Exponential Weighted Average both show improvements of over 5% – even with the large number of subjects being used.

6.5 Algorithm Complexity

Tables 5, 6 and 7 shows the timing increase per classification which occurs when using a novel classification algorithm versus using the baseline algorithm (top match). The increase in classification time is minimal for all of the algorithms running on any of the databases and thus confirms that the algorithms are feasible for use in real-time applications such as streaming video at 25 fps. In fact, the roughly 2ms worst case increase would only become a factor if classifying thousands of faces (e.g. at a sports stadium). These results were generated using an Intel Core

Table 5: Caltech Faces Timing Increase In ms

Classification Algorithm	Time
Top-k Average (k=5)	0.58
Adaptive K-nearest Neighbour (k=5)	0.98
$K\prime$ Subject Average (k=40)	1.39
$K\prime$ Subject Average (k=60)	1.98
$K\prime$ Subject Average (k=80)	1.56
Average	1.99
Weighted Average	1.32
Reverse Weighted Average	1.85
Exponential Weighted Average	1.33

Table 6: Georgia Face Database Timing Increase In ms

Classification Algorithm	Time
Top-k Average (k=5)	1.07
Adaptive K-nearest Neighbour (k=5)	0.74
$K\prime$ Subject Average (k=40)	0.98
$K\prime$ Subject Average (k=60)	0.69
$K\prime$ Subject Average (k=80)	1.32
Average	1.48
Weighted Average	1.52
Reverse Weighted Average	1.23
Exponential Weighted Average	1.26

Table 7: ORL Database of Faces Timing Increase In ms

Classification Algorithm	Time
Top-k Average (k=5)	0.27
Adaptive K-nearest Neighbour (k=5)	0.79
$K\prime$ Subject Average (k=40)	0.86
$K\prime$ Subject Average (k=60)	0.84
$K\prime$ Subject Average (k=80)	0.95
Average	0.87
Weighted Average	0.88
Reverse Weighted Average	0.94
Exponential Weighted Average	0.86

2 Duo E8400 (3.0GHz) with 2GiB of RAM running Ubuntu Linux 10.10 64-bit.

7 Conclusions and Future Works

Methods for increasing face recognition rates have been investigated with the primary focal area being the classification algorithm. A number of the new classification algorithms show improvements of the literature results termed as the baseline results. It can be concluded that the best performing classification algorithm is Weighted Average with improvements over the baseline of 6.83%, 11.61% and 5.45% in the three test databases. Importantly, the algorithms are computationally efficient making them feasible for use in real-time applications.

So far, the novel classification algorithms have only been applied to the eigenfaces technique. Future research will involve applying the classification algorithms to other face recognition techniques to see if similar improvements are obtained.

Bibliography

[1] A.W. Senior, R.M. Bolle (2002); *Face Recognition and its Applications: Biometric Solutions for Authentication in an E-World*, Kluwer Academic Publishers, 2002.

[2] Lin, Shang-hung (2000), An Introduction to Face Recognition Technology, *Informing Science The International Journal of an Emerging Transdiscipline*, 3(1): 1-7.

[3] M.A. Turk, A.P. Pentland (1991); IEEE Computer Society Conference on Computer Vision and Pattern Recognition, *Proceedings of Computer Vision and Pattern Recognition*, 586-591.

[4] H. Moon, P.J. Phillips (2001); Computational and performance aspects of PCA-based face-recognition algorithms, *Perception*, 30(3): 303-321.

[5] C. Liu, H. Wechsler (1999), Comparative assessment of Independent Component Analysis, *International Conference on Audio and Video Based Biometric Person Authentication, AVBPA'99, Washington D.C. USA*, 22-24.

[6] A.M. Bronstein et al (2003); 3D Face Recognition Without Facial Surface Reconstruction, *Technion – Computer Science Department.*

[7] Viola et al (2004); Robust Real-Time Face Detection, *Int. J. Comput. Vision*, 57(2): 137-154.

[8] G. Bradski (2000); The OpenCV Library, *Dr. Dobb's Journal of Software Tools.*

[9] H.A. Rowley et al (1995); Human Face Detection in Visual Scenes, *Advances in Neural Information Processing Systems 8*, 875-881.

[10] P.J. Phillips et al (1998); The FERET database and evaluation procedure for face recognition algorithms, *Image and Vision Computing Journal*, 16(5), 295-306.

[11] P.J. Phillips et al (2000); The FERET Evaluation Methodology for Face Recognition Algorithms, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22: 1090-1104.

[12] T.A.M. Kevenaar et al (2005), Face Recognition with Renewable and Privacy Preserving Binary Templates, *Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies*, 21-26.

[13] C. Ling et al (2005), Face recognition based on multi-class mapping of Fisher scores, *Pattern Recognition*, 38(6): 799-811.

[14] F.S. Samaria and A.C. Harter (1995); Parameterisation of a Stochastic Model for Human Face Identification, W*orkshop on Applications of Computer Vision.*

[15] L. Zhang et al (2011), Sparse Representation or Collaborative Representation: Which helps face recognition?, *Proceedings of the 2011 IEEE International Conference In Computer Vision (ICCV)*, 471-478.

[16] R. Khaji et al (2013), Collaborative Representation for Face Recognition based on Bilateral Filtering, *IJCSI International Journal of Computer Science*, 397-401.

[17] H. Zheng, J. Xie, Z. Jin (2012); Heteroscedastic Sparse Representation Based Classification for Face Recognition, *Neural Processing Letters*, 233-244.

[18] J. Wright et al (2009); Robust Face Recognition via Sparse Representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 233-244.

[19] W. Andrew et al (2012); Toward a practical face recognition system: Robust alignment and illumination by sparse representation, *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 372-386.

[20] Z. Pengfei et al (2012), Multi-scale Patch based Collaborative Representation for Face Recognition with Margin Distribution Optimization, *Computer Vision-ECCV 2012*, Springer Berlin Heidelberg, 822-835.

[21] B. Mikhail, P. Niyogi (2003); Laplacian Eigenmaps for Dimensionality Reduction and Data Representation, *Neural Computation*, 1373-1396.