Face Recognition Using Non-Linear Image Reconstruction

S. Duffner C. Garcia
Orange Labs
4, Rue du Clos Courtel
35512 Cesson-Sévigné, FRANCE

Abstract

We present a face recognition technique based on a special type of convolutional neural network that is trained to extract characteristic features from face images and reconstruct the corresponding reference face images which are chosen beforehand for each individual to recognize. The reconstruction is realized by a so-called "bottle-neck" neural network that learns to project face images into a low-dimensional vector space and to reconstruct the respective reference images from the projected vectors. In contrast to methods based on the Principal Component Analysis (PCA), the Linear Discriminant Analysis (LDA) etc., the projection is non-linear and depends on the choice of the reference images. Moreover, local and global processing are closely interconnected and the respective parameters are conjointly learnt. Having trained the neural network, new face images can then be classified by comparing the respective projected vectors. We experimentally show that the choice of the reference images influences the final recognition performance and that this method outperforms linear projection methods in terms of precision and robustness.

1. Introduction

Face recognition has been of increasing interest during the last decades due to a vast number of possible applications like biometrics, video-surveillance, advanced human-computer interaction or image and video indexation.

Many different approaches have been proposed in the literature [5, 17] which can roughly be divided into two groups. The first group consists of so-called local approaches which make use of special feature extractors in order to detect certain local characteristics. Subsequently, a global model combines these features and their arrangement in a certain way in order to classify the given face image. Brunelli and Poggio [3] for example use geometric models like the distances between pairs of feature points to classify face images. A probabilistic approach based on 2-dimensional Hidden Markov Models modeling local variations of shape and texture has been proposed by Perronnin *et al*. [12]. A method called "Elastic Bunch Graph Matching" has been proposed by Wiskott *et al*. [16]. Here the shape of each face is modeled by a graph where each node contains the possible appearances of a facial feature. Finally, methods based on Active Appearance Models introduced by Cootes *et al*. [6] have also been used for face recognition [7].

The second group is represented by global approaches which all realize a form of statistical projection of the high-dimensional image vectors into a lower-dimensional space where the final classification is performed. The most well-known of these methods are the "Eigenfaces" approach [14] using PCA and the "Fisherfaces" approach [1] using LDA. Many variants (*e.g*. [15, 4]) based on these works have followed.

The drawback of most of the *global* approaches is their sensitivity to illumination changes. This problem is mainly due to the *linear* processing whereas, under varying lighting conditions, the appearance of a face image undergoes a *non-linear* transformation. On the other hand, the drawback of *local* methods is that they often require an empirical choice of parameters, *e.g*. number of scales and orientations of the gabor filters or the positions where to apply the filters, which makes their implementation cumbersome.

We propose an approach that alleviates these problems by using a special type of convolutional neural network that learns to reconstruct from any face image of a given face database a reference face image that "best" represents the respective person and that is chosen beforehand.

The "bottle-neck" architecture of the neural network actually learns a non-linear projection of the face images into a sub-space of lower dimension and then reconstructs the respective reference images from this compressed representation. By using a convolutional neural network, an *empirical* choice of filter parameters is not necessary. Instead, the neural network learns these filters conjointly with the projection and reconstruction parameters while minimizing the overall reconstruction error. After training, face images can be classified by calculating the distances between projected vectors in the intermediate layer of the network or between the reconstructed images at the output of the network.

The remainder of this paper is organized as follows: The architecture of the convolutional neural network is de-

scribed in section 2. Section 3 outlines some alternatives to automatically select the reference images and then explains the training procedure of the neural network. In section 4 we describe how to apply the trained neural network to recognize faces and in section 5 we show some experimental results with two public face databases. Finally, we present our conclusions in section 6.

2 Neural network architecture

The proposed neural architecture is a specific type of neural network consisting of six layers where the first layer is the input layer, the three following layers are convolutional and sub-sampling layers and the last two layers are standard feed-forward neuron layers. The system is trained to transform an input face image into a reference image predefined for each face image. Fig.1 gives an overview of the architecture.

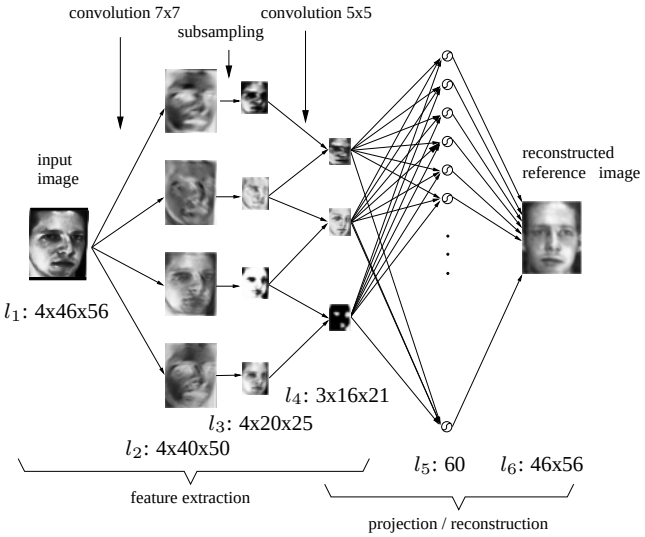

Figure 1: Architecture of the neural network

The retina l_1 receives a cropped face image of 46x56 pixels, containing gray values normalized between -1 and $+1$.

The second layer l_2 consists of four so-called feature maps which are all connected to the input map as follows: each unit of a feature map receives its input from a set of neighboring units of the input map (retina) as shown in Fig.2. This set of neighboring units is often referred to as a *local receptive field*, a concept which is inspired by Hubel and Wiesel's discovery of locally-sensitive, orientation-selective neurons in the cat visual system [9]. Such local connections have been used many times in neural models of visual learning [8, 10, 11]. They allow extracting elementary visual features such as oriented edges, end-

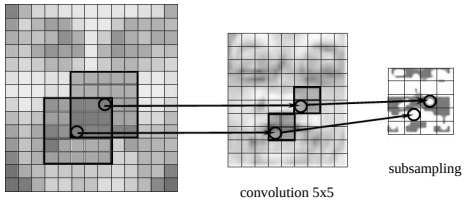

Figure 2: Example of a 5x5 convolution map followed by a 2x2 sub-sampling map

points or corners which are then combined by subsequent layers in order to detect higher-order features. Clearly, the position of particular visual features can vary considerably in the input image because of distortions or shifts. Additionally, an elementary feature detector can be useful in several parts of the image. For this reason, each unit of a feature map shares its weights with all other units of the same feature map so that each map has a fixed feature detector. Thus, each feature map y_{2i} of layer l_2 is obtained by convolving the input map y_1 with a trainable kernel w_{2i}:

$$y_{2i}(x,y) = \sum_{(u,v)\in K} w_{2i}(u,v)\, y_1(x+u, y+v) \; + b_{2i} \; ,$$

where $K = \{(u,v) \,|\, 0 \le u < s_x \text{ and } 0 \le v < s_y\}$ and $b_{2i} \in \mathbb{R}$ is a trainable *bias* which compensates for lighting variations in the input. In our system, the four feature maps of the second layer perform each a different 7x7 convolution ($s_x = s_y = 7$). Note that the size of the obtained convolutional maps in l_2 is smaller to avoid border effects in the convolution.

The third layer l_3 sub-samples its input feature maps into maps of reduced dimension by locally averaging neighboring units. Further, the average is multiplied by a trainable weight w_{3j} and a trainable bias b_{3j} is added before applying a sigmoid activation function $\Phi(x) = \arctan(x)$:

$$y_{3j}(x,y) = \Phi\Big(w_{3j} \sum_{(u,v)\in\{0,1\}^2} y_{2j}(2x+u, 2y+v) + b_{3j}\Big).$$

The goal of this layer is to make the system less sensitive to small shifts, distortions and variations in scale and rotation of the input at the cost of some precision.

Layer l_4 is another convolutional layer and consists of three feature maps each connected to two preceding maps as illustrated in figure 1. Basically, it operates in the same way as the second layer but performs a 5x5 instead of a 7x7 convolution. By combining the results of the low-level feature detectors, like edges or corners, it extracts higher-level features corresponding to more characteristic forms or patterns of a face image. Unlike the second layer, a sigmoid activation function is used here.

While the previous layers act principally as local feature extraction layers, layers l_5 and l_6 transform the local information into a more global model. Layer l_5 is composed of a reduced number of neurons fully connected to layer l_4. This is the so-called "bottle-neck" of the network where a compact representation of the input face images is learnt.

The architecture of this part of the network is inspired by *auto-associative* neural networks which are trained to reproduce an input pattern at their outputs while using a hidden layer containing much fewer neurons (bottle-neck) than the input and output layers. It was shown in [2] that there exists a close connection between auto-associative neural networks and PCA when the neurons' activation functions are linear. Here, a so-called hetero-association is performed in the last three layers, because the desired output in layer l_6 is different from the output of layer l_4. Moreover, the activation functions of the neurons in l_5 are non-linear. Thus, the operation is essentially different from that of a PCA but, nethertheless, it performs a dimensionality reduction as l_5 contains much fewer neurons than l_4 and l_6.

In our proposed architecture, Layer l_5 contains 60 neurons with sigmoid activation function and the output layer l_6 is composed of an array of neurons of size 46x56 representing a gray-scale image normalized between -1 and $+1$. The neurons are fully connected to the preceding neurons and use a linear activation function.

3 Training Procedure

The neural network is trained using a face database with a fixed number N of individuals (closed world). For each individual several images with varying pose, illumination and facial expressions are necessary. The training procedure consists of 3 succesive steps: division of the face database into training and test set, selection of the reference images and the actual training of the neural network.

The first step one image per individual chosen randomly and used for later testing (*leave-one-out* validation). The rest of the images constitute the training set. Steps two and three are detailed in the following:

3.1 Choosing the reference images

Let us denote im_{ij} the j-th example of individual $i = 1..N$ in the face database ($j = 1..M_i$). For each im_{ij} a reference image r_i among the face images in the training set has to be chosen. The neural network is then trained to respond for any input image of a given individual with the respective reference image for that individual. In this way, it will learn to extract features invariant to the intra-class variations present in the training images, *e.g.* pose, illumination or facial expressions.

We experimented with two different strategies for choosing the reference images. They are both based on a Euclidean distance measure between the image vectors im^+_{ij} which are the one-dimensional vectors obtained by concatenating the rows of the respective images im_{ij}. The strategies are the following:

1. *Choose most representative image:* the face image of the individual i that is closest to the mean image $\overline{im^+_i}$ of i is chosen:
$$r_i = \underset{im^+_{ij}}{\operatorname{argmin}} \|im^+_{ij} - \overline{im^+_i}\| \;\; \forall\, i \in 1..N, j \in 1..M_i \;. \quad (1)$$

2. *Choose most distant image:* the face image of the individual i that has the greatest distance to the face images of all the other individuals is chosen.
$$r_i = \underset{im^+_{ij}}{\operatorname{argmax}} \|im^+_{ij} - im^+_{kl}\|$$
$$\forall\, i,k \in 1..N,\; j \in 1..M_i,\; l \in 1..M_k,\; k \neq i \;. \quad (2)$$

We will call these strategies MEAN and DIST in the following.

3.2 Training the neural network

In order to construct the training set for the neural network the face images are normalized in the following way. First, each image is cropped in such a way that the face is centered and that the eyes and the mouth are roughly at predefined positions while keeping the aspect ratio. Then, each image is histogram-equalized and resized to the dimensions of the retina l_1 (46x56). Training is performed using the back-propagation algorithm which has been slightly adapted to account for the shared weights in layers l_2 to l_4. For a given example im_{ij} the objective function is the following:

$$E_i = \frac{1}{WH} \sum_{x=1}^{W} \sum_{y=1}^{H} (d_i(x,y) - y_6(x,y))^2 \quad , \quad (3)$$

which is the mean squared error (MSE) between the computed outputs $y_6(x,y)$ and the desired outputs $d_i(x,y)$, where d_i represents the respective reference image normalized between -1 and $+1$. Before the actual training, the weights are initialized at random. Then, they are updated after each presentation of a training example (online training). Training is stopped after 8000 iterations.

Note that by training the neural network, *i.e.* by minimizing the objective function, all parameters are learnt conjointly: the convolution filters, the projection and the reconstruction parameters. In other words, the proposed architecture optimizes the filters and, at the same time, the projection parameters in order to reconstruct best the respective reference images. This is a clear advantage compared to most other projection methods where separate preprocessing and projection steps necessitate a "manual" integration and parameter determination.

4 Recognizing Faces

Once the neural network is trained with a certain number of individuals, it can be applied to previously unseen face images of the same individuals in order to recognize them. To this end, a given face image is cropped and normalized in the same way as the training images (cf. section 3.2) and presented to the neural network. The neural network then reconstructs the reference image corresponding to the respective individual. Finally, a simple nearest neighbor classification based on the Euclidean distance between the neural network's output and all the reference images identifies the individual shown on the input face image.

More formally,

$$I = \underset{i}{\operatorname{argmin}} \|y_6 - d_i\| \quad \forall\, i \in 1..N \;, \tag{4}$$

where I is the resulting identity, y_6 is the output of the neural network and d_i is the reference image of individual i, normalized between -1 and $+1$.

In our experiments, however, we slightly modified this classification algorithm for efficiency reasons. Instead of classifying the outputs of the final layer l_6 we used the outputs of the neuron layer l_5 which represent the projected vectors. We then compare the projected vectors with the ones produced by the reference images. Thus, the classification formula becomes:

$$I = \underset{i}{\operatorname{argmin}} \|y_5 - v_i\| \quad \forall\, i \in 1..N \;, \tag{5}$$

where v_i represents the the output of layer l_5 when presenting r_i to the neural network.

The two classification formulas led to equivalent results but the second one is more efficient in terms of computation time. Thus, all the results presented in this paper were obtained using Eq. 5.

5 Experimental Results

We conducted experiments on two public face databases: the Olivetti Research Ltd. (ORL) face database [13] and the Yale database [1].

The ORL database contains 40 individuals with 10 images per individual showing slight pose variations, facial expressions and rather limited illumination changes. The Yale database contains only 15 individuals with 11 images each. They show virtually no pose variations but much more illumination variations (*e.g.* left/right/center light) and facial expressions (smile, sad expression, open/closed mouth).

In order to evaluate the different approaches we performed a "leave-one-out" validation, i.e. the neural network was initialized and retrained 30 times with a random separation into training and test set. Then, the mean of the respective recognition rates was calculated.

Fig. 3 shows the Receiver Operator Characteristic (ROC) curves of the proposed approach with both reference image selection strategies, MEAN and DIST. The ROC curves illustrate the recognition rates *vs*. the false accept rate while varying a distance threshold above which a face image is rejected. In general, the recognition rate of the ORL database is higher than that of the Yale database. The recognition rates without rejection are shown in table 1. Further, the MEAN approach performs better than DIST (cf. Sect. 3.1) for both test sets. Thus, the following experiments only show the results of the MEAN approach. Fig. 4 illustrates some face images (top row), the reconstructions (middle row) and the respective reference images (bottom row).

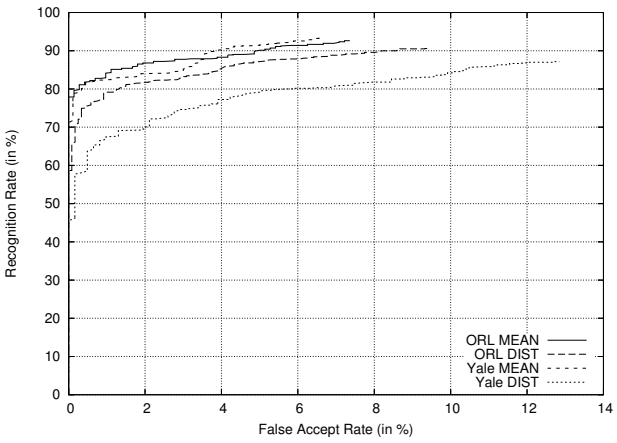

Figure 3: ROC curves for the ORL and Yale databases

(a) ORL (b) Yale

Figure 4: Examples of image reconstruction. *Top row:* input images, *middle row:* reconstructed images, *bottom row:* reference images

We also compared the proposed approach with the Eigenfaces and the Fisherfaces approaches. As for the preceding experiment, a leave-one-out validation with the same training and test sets was performed. Note that concerning the classification procedure the proposed approach is more efficient in terms of computation time and memory usage because it only requires the reference images in order to classify new face image whereas the other two approaches

	ORL	Yale
Eigenfaces	89.7%	77.9%
Fisherfaces	87.7%	85.2%
proposed approach: DIST	90.6%	87.1%
proposed approach: MEAN	**92.6%**	**93.3%**

Table 1: Average recognition rates

need the whole dataset. Fig. 5 and 6 show the ROC curves of the Eigenfaces and Fisherfaces methods together with the proposed approach for the ORL and Yale database respectively. For both of the databases the proposed method clearly outperforms the other methods. Table 1 summarizes the recognition rates of the preceding experiments.

We further evaluated the robustness of our approach with respect to noise and partial occlusions. In the first experiment we added Gaussian noise with increasing standard deviation σ to the images of the test set. Fig. 7 shows the respective recognition rates with varying σ. Note that a σ of 0.5 represents a considerable amount of noise as the gray values are between -1 and $+1$ in an image of size 46x56 (see illustration at the bottom of Fig. 7). The graphs show that the proposed method is very robust to Gaussian noise. For $\sigma < 0.5$ the recognition rate decreases by only 12% for the ORL database and by only 6% for the Yale database, and it remains above 80% for $\sigma < 0.6$. The Eigenfaces approach, on the other hand, shows even slightly better performance the recognition rate staying almost constant over the whole interval. This can be explained by the pure global processing of the PCA. As it is extracting rather lower frequency features it is less sensitive to high-frequency noise.

The last experiment demonstrates the robustness of the approach with respect to partial occlusion. To this end, the bottom part of the images is masked by a black band of varying height. Fig. 8 shows the respective results as well as some example images at the bottom to illustrate the type of occlusion (0%, 10%, 20%, 30% and 40%). Here, our approach clearly outperforms the Eigenfaces method. For both databases the recognition rate stays above 80% when the occluded proportion is less than 20% of the image whereas the performance of the Eigenfaces method drops considerably.

6 Conclusions

We presented a face recognition method based on a specific type of neural network that receives a face image at its input and reconstructs a reference face image pre-defined for each individual. The reconstruction is realized by a "bottleneck" architecture that projects the face images into a lower-dimensional vector space before reconstruction. Compared to most other statistical projection methods, the processing is *non-linear* and all parameters are learnt conjointly by the neural network, including a hierarchical set of convolution filters at the input. The performance of the method was evaluated with two public face databases: the ORL and the Yale database. The proposed approach achieved a recognition rate of 92.6% for the ORL and 93.3% for the Yale database and outperforms the well-known Eigenfaces and Fisherfaces methods. We further demonstrated that our method is very robust to Gaussian noise and partial occlusions.

References

[1] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegmann. Eigenfaces vs fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on PAMI*, 17(7):711–720, 1997.

[2] H. Bourlard and Y. Kamp. Auto-association by multilayer perceptrons and singular value decomposition. *Biological Cybernetics*, 59(4):291–294, 1988.

[3] Roberto Brunelli and Tomaso Poggio. Face recognition: Features versus templates. *IEEE Transactions on PAMI*, 15(10):1042–1052, 1993.

[4] H. Çevikalp, M. Neamtu, M. Wilkes, and A. Barkana. Discriminative common vectors for face recognition. *IEEE Transactions on PAMI*, 27(1):4–13, 2005.

[5] R. Chellappa, C.L. Wilson, and S. Sirohey. Human and machine recognition of faces: a survey. *Proceedings of the IEEE*, 83(5):705–741, 1995.

[6] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. *IEEE Transactions on PAMI*, 23(6):681–685, 2001.

[7] G.J. Edwards, C.J. Taylor, and T.F. Cootes. Interpreting face images using active appearance models. In *Autom. Face and Gesture Rec.*, pages 300–305, 1998.

[8] K. Fukushima. Cognitron: A self-organizing multilayered neural network. *Biological Cybernetics*, 20:121–136, 1975.

[9] D. Hubel and T. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *Journal of Physiology*, 160:106–154, 1962.

[10] Yann LeCun. Generalization and network design strategies. In *Connectionism in Perspective*, Zurich, 1989.

[11] Michael C. Mozer. *The perception of multiple objects: a connectionist approach*. MIT Press, Cambridge, MA, USA, 1991.

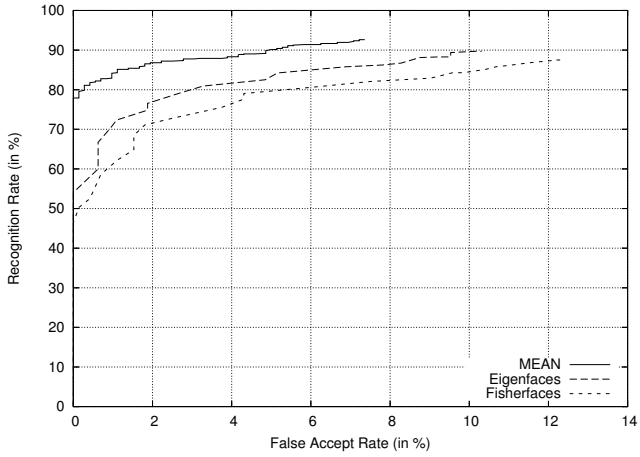

Figure 5: Comparison with the Eigenfaces approach: ORL database

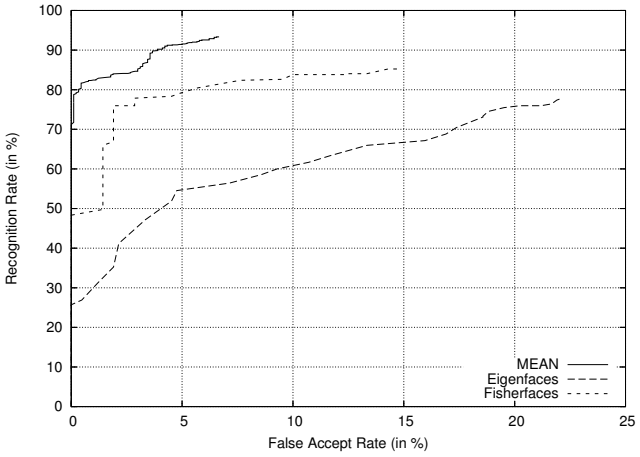

Figure 6: Comparison with the Eigenfaces approach: Yale database

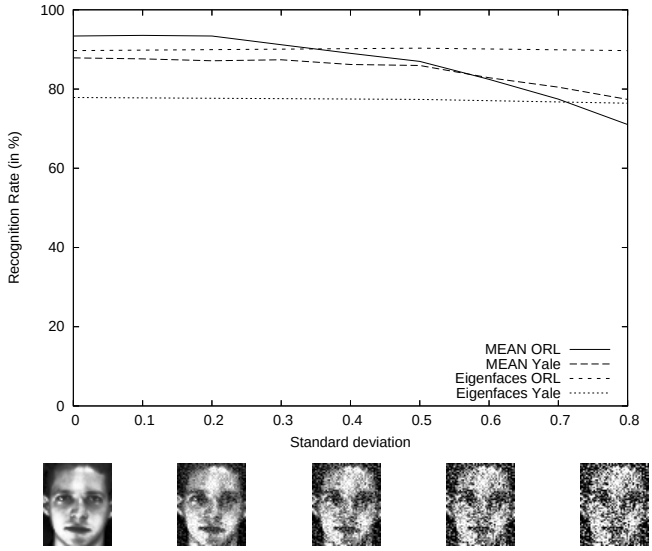

Figure 7: Sensitivity analysis: Gaussian noise

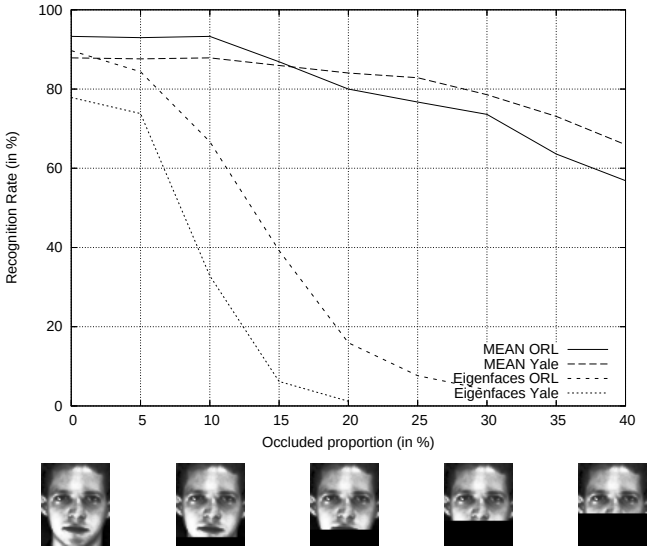

Figure 8: Sensitivity analysis: partial occlusion

[12] F. Perronnin, J.L. Dugelay, and K. Rose. A probabilistic model of face mapping with local transformations and its application to person recognition. *IEEE Transactions on PAMI*, 27(7):1157–1171, 2005.

[13] F. Samaria and A. Harter. Parametrisation of a stochastic model for human face identification. In *2nd IEEE Workshop on Applications of Computer Vision*, pages 138–142, 1994.

[14] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *Computer Vision and Pattern Recognition*, pages 586–591, 1991.

[15] M. Visani, C. Garcia, and J.M. Jolion. Normalized radial basis function networks and bilinear discriminant analysis for face recognition. In *AVSS*, Como, Italy, September 2005.

[16] L. Wiskott, J.-M. Fellous, N. Krüger, and C. von der Malsburg. Face recognition by elastic bunch graph matching. *IEEE Transactions on PAMI*, 19(7):775–779, 1997.

[17] W. Zhao, R. Chellappa, P.J. Phillips, and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys (CSUR)*, 35(4):399–458, 2003.