Low Resolution Face Recognition Using Mixture of Experts

Fatemeh Behjati Ardakani, Fatemeh Khademian, Abbas Nowzari Dalini, and Reza Ebrahimpour

Abstract—Human activity is a major concern in a wide variety of applications, such as video surveillance, human computer interface and face image database management. Detecting and recognizing faces is a crucial step in these applications. Furthermore, major advancements and initiatives in security applications in the past years have propelled face recognition technology into the spotlight. The performance of existing face recognition systems declines significantly if the resolution of the face image falls below a certain level. This is especially critical in surveillance imagery where often, due to many reasons, only low-resolution video of faces is available. If these low-resolution images are passed to a face recognition system, the performance is usually unacceptable. Hence, resolution plays a key role in face recognition systems. In this paper we introduce a new low resolution face recognition system based on mixture of expert neural networks. In order to produce the low resolution input images we down-sampled the 48×48 ORL images to 12×12 ones using the nearest neighbor interpolation method and after that applying the bicubic interpolation method yields enhanced images which is given to the Principal Component Analysis feature extractor system. Comparison with some of the most related methods indicates that the proposed novel model yields excellent recognition rate in low resolution face recognition that is the recognition rate of 100% for the training set and 96.5% for the test set.

Keywords—Low resolution face recognition, Multilayered neural network, Mixture of experts neural network, Principal component analysis, Bicubic interpolation, Nearest neighbor interpolation.

I. Introduction

AS multimedia applications become ubiquitous, increasing processor speeds face recognition algorithms are being extended and rewritten to take advantage of video and other sensor modalities that produce a continuous stream of frames instead of a single image [9]. Video based sensors can provide important visual information in a number of applications. For example, at an airport gate entrance, video cameras are being used instead of still image digital cameras. Cellphones are equipped now with cameras capable of capturing a sequence of frames instead of a single image. Camcorders are everywhere, and the need to parse video digital libraries to extract specific content (such as faces) is soon to become a daily activity of search engines. The applications of face recognition using multiple still images are not limited to entertainment, education, or surveillance. In video surveillance, the faces of interest are often of small size because of the great distance between the camera and the objects which leads to work with low resolution images. Image resolution is a potential factor affecting face recognition performance. In the low-resolution face images, many detailed facial features are lost and faces are indiscernible to human. We also notice that in many automatic face recognition systems, the size of face images are reduced, and also achieve satisfied performance. But how will the image resolution affect recognition accuracy is still open to discussion.

Fatemeh Behjati Ardakani, Fatemeh Khademian and Abbas Nowzari-Dalini are with the Center of Excellence in Biomathematics, School of Mathematics, Statistics and Computer Science, University of Tehran, Tehran, Iran, e-mail addresses: (f.behjati@ut.ac.ir, khademian@ut.ac.ir, nowzari@ut.ac.ir) and Reza Ebrahimpour is with the Department of Electrical Engineering, Shahid Rajaee University, Tehran, Iran, email address: (ebrahimpour@ipm.ir).

Several algorithms have been proposed to render a super-resolution face image from the low-resolution one such as super-resolution algorithms [3], [4], [7], and [14] that use some interpolation techniques to enhance the image quality. Actually, these algorithms preprocess the low resolution image and pass it to the next phase which is face recognition. In the recognition phase some classification algorithms are required in order to distinguish and identify the faces from each other. Algorithms such as k-nearest neighbor [5], artificial neural networks [13], local visual primitives [13] and coupled locality preserving mapping [12] have yet been deployed for the purpose of face recognition.

In this paper, we proposed a method that accomplishes the face recognition by using combined neural classifiers especially mixture of expert neural networks [1], [6] and also the Principal Component Analysis technique as a feature extraction tool [8]. The process in the whole is as follows: i) the first step is enhancing the given image quality by using the bicubic interpolation method, ii) the second one is extracting eigenvalue and eigenvector from the enhanced image using PCA, iii) and finally, feed these eigenvectors to the trained network and get the final result.

In order to train the mixture of expert network, we produced an artificial dataset from the ORL [8] dataset by reducing the size of its images using k-nearest interpolation algorithm.

The rest of the paper is organized as follows. In Section 2 our proposed model is introduced. It is followed by the experimental results in Section 3. Finally, Section 4 draws conclusion and summarizes the paper.

II. Basic Concepts

In this section we provide some basic information that are essential to understand our proposed method, including interpolation techniques, feature extraction and mixture of expert neural networks.

A. Interpolation

Interpolation works by using known data to estimate values at unknown points. Common interpolation algorithms can be grouped into two categories: adaptive and non-adaptive [10].

Adaptive methods change depending on what they are interpolating (sharp edges vs. smooth texture), whereas non-adaptive methods treat all pixels equally. Since we have used the non-adaptive interpolation methods in our algorithm it is worth to introduce some techniques in this category.

1) Nearest neighbor interpolation: Nearest neighbor is the most basic and requires the least processing time of all the interpolation algorithms because it only considers one pixel; the closest one to the interpolated point. This has the effect of simply making each pixel bigger. Since this method does not conserve the image quality we apply it in order to produce low resolution images. In other words, to produce the input images we down-sampled the 48×48 ORL images to 12×12 ones using this interpolation method.

2) Bilinear Interpolation: Bilinear interpolation considers the closest 2×2 neighborhood of known pixel values surrounding the unknown pixel. It then takes a weighted average of these 4 pixels to arrive at its final interpolated value. The bilinear interpolation results are much smoother looking images than nearest neighbor interpolation.

3) Bicubic interpolation: Bicubic goes one step beyond bilinear by considering the closest 4×4 neighborhood of known pixels, for a total of 16 pixels. Since these pixels are at various distances from the unknown pixel, closer pixels are given a higher weighting in the calculation. Bicubic produces noticeably sharper images than the previous two methods, and is perhaps the ideal combination of processing time and output quality. For this reason it is a standard in many image editing programs (including Adobe Photoshop), printer drivers and in-camera interpolation. In this work we use this interpolation method to enhance the given image quality and up-sample the 12×12 input images to 24×24 ones.

B. Principal Component Analysis

Principal Component Analysis (PCA) is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences [11]. Since patterns in data can be hard to find in data of high dimension, where the luxury of graphical representation is not available, PCA is a powerful tool for analyzing data. The other main advantage of PCA is that once you have found these patterns in the data, and you compress the data, i.e. by reducing the number of dimensions, without much loss of information. This technique includes the following 5 main steps

- Step 1. Subtract the mean of the input data: For PCA to work properly, we have to subtract the mean from each of the data dimensions. The mean subtracted is the average across each dimension.
- Step 2. Calculate the covariance matrix: Recall that covariance is always measured between 2 dimensions. If we have a data set with more than 2 dimensions, there is more than one covariance measurement that can be calculated.
- Step 3. Calculate the eigenvectors and eigenvalues of the covariance matrix: Since the covariance matrix is square, we can calculate the eigenvectors and eigenvalues for this matrix. These are rather important, as they tell us useful information about our data.
- Step 4. Choosing components and forming a feature vector: In general, once eigenvectors are found from the covariance matrix, the next step is to sort the eigenvalues in decreasing order. This gives you the components in order of significance.
- Step 5. Deriving the new data set: This is the final step in PCA, and is also the easiest. Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the left of the transpose of the original data set.

C. Mixture of Experts

Expert combination is a classic strategy that has been widely used in various problem solving tasks. A team of individuals with diverse and complementary skills tackle a task jointly such that a performance better than any single individual can make is achieved via integrating the strengths of individuals [6].

The mixture of experts (ME) architecture is composed of N local experts and there is a gating network whose outputs define the expert weights conditioned on the input (Figure 1). In our proposed method, each expert i is a multi layer perceptron (MLP) neural network with one hidden layer that computes an output O_i as a function of the input stimuli vector x, and a set of weights of hidden and output layers, and a sigmoid activation function. We assume that each expert specializes in a different area of the input space. The gating network assigns a weight g_i to each of the expert's output, O_i. The gating network determines the $g = \{g_1, g_2, \cdots, g_N\}$ as a function of the input vector x and a set of parameters such as weights of its hidden and output layers and a sigmoid activation function. Each element g_i of g can be interpreted as estimates of the prior probability that expert i can generate the desired output y. The gating network is composed of two layers: the first layer is an MLP neural network, and the second layer is a softmax nonlinear operator. Thus the gating network computes $\tau = \{\tau_1, \tau_2, \cdots, \tau_N\}$, which is the output vector of the MLP layer of the gating network, then applies the softmax function to get:

$$g_i = \frac{exp(\tau_i)}{\sum_{j=1}^{N} exp(\tau_j)} \quad i = 1, 2, \cdots, N,$$

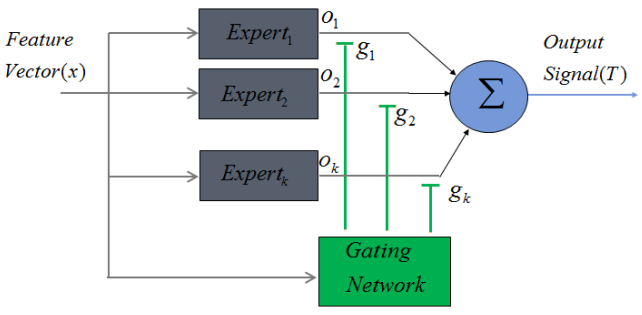

Fig. 1. The structure of mixture of experts neural network.

where N is the number of expert networks. So the g_is are nonnegative and sum to 1. The final mixed output of the entire network is:

$$T = \sum_i O_i g_i \quad i = 1, 2, \cdots, N.$$

The weights of MLPs are learned using the error backpropagation (BP) algorithm. For each expert i and the gating network, the weights are updated according to the following equations:

$$\Delta w_i = \eta_e h_i (y - O_i)(O_i(1 - O_i))\nu_i^T,$$

$$\Delta \omega_i = \eta_e h_i w_i^T (y - O_i)(O_i(1 - O_i))(\nu_i(1 - \nu_i))x,$$

$$\Delta \xi = \eta_g (h - g)(\tau(1 - \tau))\vartheta^T,$$

$$\Delta \zeta = \eta_g \xi^T (h - g)(\tau(1 - \tau))\vartheta(1 - \vartheta)x,$$

where η_e and η_g are learning rates for the expert and the gating networks, respectively. ω and w are the weight matrices of input to hidden, and hidden to output layer, respectively, for experts and ζ and ξ are the weight matrices of input to hidden and hidden to output layer, respectively, for the gating network. ν_i^T and ϑ^T are the transpose of ν_i and ϑ, the output matrices of the hidden layer of expert and gating networks, respectively. In the above formulas $h = \{h_1, h_2, \cdots, h_N\}$ is a vector such that each h_i is an estimate of the posterior probability that expert i can generate the desired output y, and is computed as follows:

$$h_i = \frac{g_i \ exp(\frac{-1}{2}(y - O_i)^T(y - O_i))}{\sum_j g_j \ exp(\frac{-1}{2}(y - O_j)^T(y - O_j))}.$$

As mentioned, in Figure 1 the structure of mixture of experts method is illustrated. The following section explains our algorithm based on the preceding concepts.

III. Proposed method

To accomplish this work we used the ORL dataset which contains a set of grey level 48×48 face images. In this dataset there are ten different images of each of 40 distinct subjects. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). The steps of our method are as follows.

A. Set up the proper dataset

Since we are about to work on the low resolution images, we must down-sample the ORL dataset images to get lower quality ones. For this, we used the k-nearest neighbor interpolation algorithm and decreased the dimension of the original images by 4 which results 12×12 low resolution images. In other words, we suppose that sizes of the actual input images are 12×12. It should be noted that, in the real applications the original images are low and this step is obviously omitted.

B. Enhancing the quality of input images

As it is stated in previous sections, in most of the face recognition systems there are two distinct phases. The first phase works on the given low resolution image and enhances its quality and the second phase works on the output of the enhanced image coming from the first phase and deploys some classification algorithms to match the input against the dataset and get the identification task done. We used the bicubic interpolation algorithm for the above mentioned first phase and improved the image size, reaching the 24×24 dimension. Figure 2 depicts the two samples of ORL dataset that have been modified in favor of producing artificial input images.

In the preprocessing phase the quality of input images is improved and their size is 24×24, which means that topology of the network needs 576 nodes in the input layer. This would make the network too complicated since there are too many free parameters and the convergence of the network might not be possible. Hence, we are going to use the feature extraction technique that returns only the informative features of the image. Among the related work in this field we inferred that PCA feature extraction is much more convenient method for low resolution face recognition problem [2], [12], and [13]. Practical experiments show that the first 50-th components are sufficient for this task. We should mention that these first 50-th components are sorted in descending order, so in this way we obtain the most informative PCA components.

C. Face recognition

Once the data got ready, it is time to give them to the neural network for the recognition purpose. As it is mentioned earlier, the designed neural network consists of several MLP neural networks that play the experts role and they are combined through the mixture of experts approach. The training set includes the eigenvectors of the 5 images of each individual which is 200 images over the total of 400 (5×40), and the other 200 images are left for the testing set.

Since the input data are in the 50 dimensional space the topology of the designed network would have 50 nodes in the input layer and also because the number of subjects is 40 then the number of the nodes used in the output layer must be 40 (each node represents one subject). Therefore, the topologies

Fig. 2. left: the original ORL dataset samples, middle: down sampled by nearest neighbor interpolation, right: enhanced images by bicubic interpolation.

of the designed neural networks in this project differ only in the number of the hidden layer nodes.

We have examined different network configurations by changing the complexity of the experts or the number of experts used in designing the combined neural network and also assigning different values to the used parameters. These experimental results are provided in the following section. A schematic representation of our proposed method is depicted in Figure 3.

IV. Experimental results

This section presents the experimental results of the proposed method. In all the experiment the gating learning rate was set to 0.5 and the number of its hidden nodes was 60 nodes, the expert learning rate was 0.8, and the network trained by 300 epochs. We ran our algorithm on different number of hidden nodes of experts and the results are provided in Table I.

As it is shown in the Table I, the recognition rate for the systems containing 10 nodes in their hidden layer is relatively low. By increasing the number of hidden nodes from 10 to 20 the recognition rate improves significantly, and in the 2-experts system with 30 hidden nodes the recognition rate of 93.5% shows that the number of experts was insufficient that could not divide the input space properly. In the 4-experts system with the same number of hidden nodes comparing to the system with 3-experts there are too many free parameters that makes the network too complex to get a better result than 3-experts. So the network with 3-experts and 30 numbers of hidden nodes divides the input space in the best way and establishes a balance of the number of experts and hidden nodes. Finally increasing the number of hidden nodes to 40 incurs a loss of recognition rate comparing to 30 nodes which is because of the high complexity of the networks.

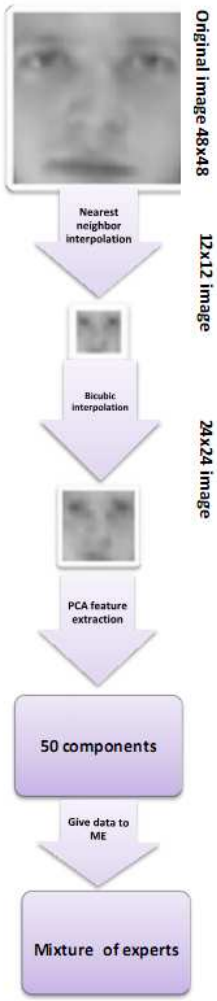

Fig. 3. The overview of our approach.

TABLE I
The recognition rates based on different number of hidden nodes and experts.

	2 experts	3 experts	4 experts
10 nodes	78 %	75 %	70.5 %
20 nodes	88%	90.5%	92.5%
30 nodes	93.5%	96.5%	93.5%
40 nodes	90.5%	96%	92.5%

We compared our method with some other related works, and the obtained results are as follows.

1) Kernel correlation filter (KCF) [2] method uses a set of MACE [2] filters to extract features from the generic training set. For every subject in the generic training set, this method builds a MACE filter. Ending up with 222 different filters, which is the total number of individuals in the generic training set. Thus the dimensionality of this feature space is 222. Each filter takes as input all $12,776$ generic training face images available. For the "authentic" class to whom the MACE filter belongs, the parameter of MACE filter values of all images belonging to this authentic class are set to 1. For all other images belonging to the remaining 221 "impostor" classes, KCF sets the corresponding parameter of MACE filter values to 0. This ensures that the filter exhibits no correlation between different subjects. The best reported recognition rate of this method is 92.31%.

2) The Hallucinating Facial Images and Features (HFIF) [13] proposed a method for simultaneous image and feature hallucination based on neighbor embedding. In HFIF method they make use of local visual primitives (LVPs) [13] in feature representation and propose local constrained neighbor selection in image/feature reconstruction. This method achieved 96% recognition rate.

3) Coupled Locality Preserving mappings (CLP) [12] is based on coupled mappings (CMs), projects the face images with different resolutions into a unified feature space which favors the task of classification. These CMs are learned through optimizing the objective function to minimize the difference between the correspondences. The best achieved recognition rate in this method is 90.1%.

Face recognition is one of the most common tasks that involves the neural network system of human brain at any time. Hence, we inspired our proposed method based on this fact and also in order to improve its efficiency we decided to combine various neural networks by the help of mixture of experts method. Figure 4 illustrates that our proposed method (ME) has better recognition rate comparing to the above mentioned

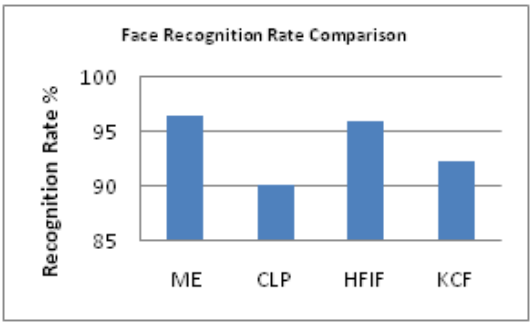

Fig. 4. Comparing the recognition rate of the known face recognition systems with our proposed method (ME)

methods. As we can see, the best performance is obtained with 3 experts and 30 number of hidden nodes. In this case the ability of generalization of patterns has been increased and it is much more stable comparing to the other cases.

V. Conclusion

The basic idea of this work is that in the networks learning process, the expert networks compete for each input pattern, while the gate network rewards the winner of each competition with stronger error feedback signals. Thus, over time, the gate partitions the input space in response to the experts performance. Consequently this approach leads us to achieve 100% recognition rate for training set and 96.5% recognition rate for the testing set that by comparing with the other face recognition systems it demonstrates that although this methods computational effort is low, it gets improved results.

References

[1] A. Iranzad, S. Masoudnia, F. Cheraghchi, A. Nowzari-Dalini, R. Ebrahimpour, in *Proceedings of International Conference on Soft Computing and Pattern Recognition*, IEEE Press, Paris, France, pp. 309–313, 2010.

[2] R. Abiantun, M. Savvides, and B. V. K. Vijaya Kumar, How low can you go? low resolution face recognition study using kernel correlation feature analysis on the FRGCv2 dataset, in *Special Session on Research at the Biometric Consortium Conference*, IEEE Press, New York, NY, USA, pp. 1–6, 2006

[3] S. Baker and T. Kanade, Hallucinating faces, in *Proceedings of 14th IEEE Conference on Automatic Face and Gesture Recognition*, IEEE Press, Los Alamitos, CA, USA, pp. 83–88, 2000.

[4] H. Chang, D. Yeung, and Y. Xiong, Super-resolution through neighbor embedding, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, Los Alamitos, CA, USA, pp. 275–282, 2004.

[5] C. Conde, A. Ruiz, and E. Cabello, PCA vs low resolution images in face verification, in *Proceedings of 12th International Conference on Image Analysis and Processing*, IEEE Press, Los Alamitos, CA, USA, pp. 63–67, 2003.

[6] R. Ebrahimpour, E. Kabir, and M.R Yousefi, Teacher-directed learning in view-independent face recognition with mixture of experts using overlapping eigenspaces, *Computer Vision and Image Understanding*, **111**, 2008, 195–206.

[7] W. Freeman, E. Pasztor, and O. Carmichael, Learning low-level vision, *International Journal of Computer Vision*, **40**, 2000, 25–47.

[8] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice Hall, Ontario, Canada, 1999.

[9] P.H. Hennings-Yeomans, S. Baker, and B.V.K. Vijaya Kumar, Recognition of low-resolution faces using multiple still images and multiple cameras, in *Proceeding of 2th IEEE Conference on Biometrics: Theory, Applications and Systems*, IEEE Press, New York, NY, USA, pp. 1–6, 2008.

[10] A.N. Htwe, Image Interpolation framework using non-adaptive approach and NL means, *International Journal of Network and Mobile Technologies*, **1**, 2010, 28–32.

[11] I.T. Jolliffe, *Principal Component Analysis*, Springer-Verlag, New York, NY, USA, 2002.

[12] B. Li, H. Chang, S. Shan, and X. Chen, Low-resolution face recognition via coupled locality preserving mappings, *IEEE Signal Processing Letters*, **17**, 2010, 20–23.

[13] B. Li, H. Chang, S. Shan, X. Chen, and W. Gao, Hallucinating facial images and features, in *Proceedings of 19th International Conference on Pattern Recognition*, IEEE Press, New York, NY, USA, pp. 1–4, 2008.

[14] C. Liu, H. Shum, and C. Zhang, A two-step approach to hallucinating faces: Global parametric model and local nonparametric model, in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, IEEE Press, Los Alamitos, CA, USA, pp. 192–198, 2001.