

Low-Resolution Face Recognition via Sparse Representation of Patches

Liansheng ZHUANG, Mengliao WANG, Wen YU, Nenghai YU, Yangchun QIAN
 MOE-MS Key Laboratory of Multimedia Computing and Communication
 University of Science and Technology of China, Hefei, 230027, P.R.China
 Email: {lszhuang@ustc.edu.cn}

Abstract

Images resolution plays an important role during face recognition. Low-resolution face images will reduce drastically the performance of face recognition algorithms. In this paper, we propose a novel approach for low-resolution face recognition. Our method first extracts patches with different size from the face images. Each patch is represented by its LBP feature. Then, we find the sparse representation of these patches based on corresponding LBP features of high-resolution face image patches. At last, we use AdaBoost to select the most discriminative patches, each of which is treated as a weak classifier, and make the ensemble of these patches weak classifiers for final decision. Experiments on Extended Yale B face database showed our method achieved high performance for low-resolution face recognition.

1. Introduction

Automatic face recognition has been a hot topic for many years, and the result on FRVT 2006 and ICE 2006 Large-Scale [13] reveals the fact that current face recognition algorithms almost perform perfectly on high-resolution frontal images. However, in real-world applications, it is hard to acquire high resolution images, especially in many surveillance scenarios. And it is also shown in [13] that most face recognition algorithms degrade when the faces are of low resolution. Thus we need to consider the problem of accurately classifying face images under low-resolution condition.

This problem could be treated as matching the low-resolution test face images to high-resolution gallery face images. There are two standard approaches to solve this problem. The first method is to downsample the gallery images and matches them in low-resolution level images. Another method is to recover low-resolution images to high-resolution ones, and then use the recovered images for face recognition. We call the method of recovering low-resolution images as super-resolution.

Super-resolution has become a hot topic in recent years. In 2000, W. Freeman et al. [4] proposed an example-based

super-resolution method, which first claimed to recover image based on the patches of image. Then Yang et al. [9] proposed a super-resolution method from the perspective of compressed sensing. In 2000 Simon Baker [1] specialize the super-resolution method to face images, and he call it Face Hallucination. Yang et al. [12] proposed a face hallucination method which recovers face images via sparse coding. Besides, P.H. Hennings-Yeomans et al. [3] presented a novel face recognition method which simultaneously fit the super-resolution constraints and feature constraints, and it performs well on low quality face images. However, all these algorithms require long time to recover a single face image. Thus it is unpractical to classify low-resolution face images after these super-resolution algorithms on the large test image set.

Except producing high-resolution image from single low-resolution, M. Elad et al [8] proposed a method which could produce a high-resolution image from a sequence of low-resolution images. Also a novel approach of face recognition from low-resolution video sequence is presented [6], but it also requires a video sequence for enrollment. Thus this method does not suit the most face recognition applications.

Apart from the standard low-resolution face recognition method, Yang et al [2] proposed a novel approach from the sparse representation perspective. Although this method does not aim at classifying low quality face images, it outperforms greatly than representative Nearest Neighbor and Nearest Subspace method on dealing with low-resolution images. However, the method of [2] only found the sparse representation of the whole image and ignored the important local texture information of the facial image. Besides, due to the limit of the least-square solution, it requires dimensionality reduction of the image data. As a result, we lost a number of details for classifying.

In this paper, motivated by some patch-based super-resolution methods, we propose a novel approach for low-resolution face recognition which classifies the face images from local perspective. We firstly extract patches with different size from the face image, and each patch is represented by its LBP feature [14]. Then, we find the

sparse representation of these patches based on corresponding LBP features of high-resolution face images patches. To reduce the number of patches, we use AdaBoost to select most discriminative patches and make the ensemble of these weak classifiers for final decision. Our approach makes full use of local information which is omitted in [2], and gets better performance.

To verify the effectivity of our method, we conduct the experiment on the Extended Yale B database. For the 32×21 face images which are down-sampled and blurred from 192×168 face images, we only need 10 best patches to receive the recognition rate of 74.60%, while the algorithm of [2] could only receive the recognition rate of 65.30%.

The remainder of this paper is organized as follows: Section 2 details our algorithm. In Section 3, we discuss the preparation of training data and the result of our experiment on Extended Yale B database. The insufficiency of our algorithm and future work are discussed.

2. Face Recognition via Sparse Representation of Patches

The task of low-resolution face recognition requires us to classify a given low-resolution face image, according to the given gallery which is made up of N high-resolution images for K persons.

In [2] the image data is not trained but only simply recognize the face image from the global perspective, so the local information of the image is ignored. In this paper we consider finding the sparse representation of each local patch of the facial image, and then use these patches to calculate the residual for recognition. When dealing with the local patches, we do not need to do dimensionality reduction as in [2], because the dimension of patches is low enough. As a result, we could keep more image details.

However, we know that only several patches are efficient at recognizing facial images, such as eyes, nose and mouth. Thus we need to choose several crucial regions for recognition. Here we use AdaBoost for training and selecting patches. Now we start to describe the details of our algorithm.

For each face images, we choose patches with different height and width with small shift in each direction started from the upper-left corner. Finally we could find K patches from the whole image: $\{S_1, S_2, \dots, S_K\}$, and what we need to do is to prepare the positive and negative data of each patch for AdaBoost training.

For patch S_i , we exact LBP feature of corresponding region from the high-resolution training images to form a high resolution dictionary A_H :

$$\overline{A_H} = [\overline{v_{1,1}}, \overline{v_{1,2}}, \dots, \overline{v_{1,n_1}}, \overline{v_{2,1}}, \dots, \overline{v_{k,n_k}}] \in R^{D_i \times N} \quad (1)$$

where $\overline{v_{i,j}}$ ($j=1, \dots, n_i$) means LBP feature of S_i from the j th image of i th person, which is a vector. D_i is the dimension of the LBP feature vector of patch S_i (equal to 59 in our experiments), and n_i is the number of images associated with subject i . Notice that here we do not need to reduce dimension as [2] in most cases, because the dimension of data exacted from the patch is always smaller than 1,000. When dealing with large patch which dimension exceeds the number of gallery images, we could slightly reduce its dimension.

When the high resolution dictionary is formed, we could exact the LBP feature of corresponding region of patch S_i from each low-resolution image to form the vector $y_j \in R^{D_i}$. Then we need to find its sparse representation of the dictionary matrix A_H . The problem of finding such a sparse representation could be formulated as follows:

$$\min \|x\|_0 \quad \text{s.t. } y_j = A_H x, \quad (2)$$

Although the optimization problem (2) is proved to be NP-hard, recent result in [3] indicates that we could take minimizing the l^1 -norm for instead. Considering the error of noise, we take the formulation (3) instead of (2):

$$\min \|x\|_1 \quad \text{s.t. } \|y_j - A_H x\|_2 \leq \varepsilon, \quad (3)$$

Theoretically, x should be a sparse signal and its non-zero coefficients should concentrate on the class it belongs. Once x is found, the residuals of each class could be calculated as follows:

$$r_i(y_j) = \|y_j - A_H \delta_i(x)\|_2, \quad (4)$$

And the residual of y_j to the right class is the minimum value. Then for each patch, positive data and negative data could be arranged as below:

$$\begin{aligned} [\text{pos_data}] &= \{r_i(y_j) | i = \text{class}(y_j)\} \\ [\text{neg_data}] &= \{r_i(y_j) | i \neq \text{class}(y_j)\} \end{aligned}, \quad (5)$$

Where $\text{class}(y_j)$ means the person which y_j belongs to.

When the training data is well prepared, we could use AdaBoost algorithm [5] for training. After T iterations, we could find T top patches with minimum error together with their weight obtained from AdaBoost procedure.

In the recognition procedure, for a probe image y , we use the best T patch to classify. For each patch, we find its residual $r_{t,i}(y)$, $t=1, 2, \dots, T$ with formulation (3) and (4). Then with the weight of each patch, the final residual to each class is as below:

$$r_i(y) = \sum_{t=1}^T \beta_t r_{t,i}(y), t = 1, 2, \dots, T, \quad (7)$$

Where β_t is the weight of the t th patch. Finally we set identity of the probe image to the person with minimum residual.

Algorithm 1 (Face Recognition via Sparse Representation of Patches).

1: **Input:** a high-resolution training set \mathfrak{T}_H , a low-resolution training set \mathfrak{T}_L , a gallery image set G , a test image y , and an error tolerance ε

2: **for** each patch S_i , start from the upper-left corner with different size:

- Extract the LBP feature of patch S_i to form dictionary A_H
- **for** each image in low-resolution training set
 - Exact the LBP feature of patch S_i into vector l_i
 - Solve the optimization problem:

$$\min \|x\|_1, s.t. \|l_i - A_H x\|_2 \leq \varepsilon$$

- figure out the residuals:

$$r_i(l_i) = \|l_i - A_H \delta_i(x)\|_2$$

- **end**
- make the residuals into positive and negative data

3: **end**

4: Use AdaBoost algorithm for training, in order to select T best patches and their weight β .

5: **for** $t = 1, 2, \dots, T$

- Extract the LBP feature of patch S_i from low-resolution test image y into vector y_t
- Solve the optimization problem:

$$\min \|x\|_1, s.t. \|y_t - A_H x\|_2 \leq \varepsilon$$

- figure out the residuals:

$$r_{i,t}(y_t) = \|y_t - A_H \delta_i(x)\|_2$$

- **end**

6: figure out the final residual:

$$r_i(y) = \sum_{t=1}^T \beta_t r_{t,i}(y), t = 1, 2, \dots, T$$

7: **Output:** identity(y) = $\arg \min_i r_i(y)$

3. Experiments

3.1 Data Preparation

We conduct experiment on the Extended Yale B database. There are 2414 images of 38 persons under different laboratory-controlled lighting conditions, and the cropped and normalized facial image are in size of 192×168. We randomly choose 20 facial images from each person, and together 760 images for training. Then we randomly select half of the other 1654 images to be the gallery set.

Then the low-resolution images in the test image set are processed and acquired from the other half of 1654 high-resolution images. The general steps we take to produce low-resolution images always contain motion blur, aliasing and adding noise. The low-resolution images are blurred and down-sampled from origin high quality images, and then we add the Gaussian noise on them. The simulation of producing a low-resolution image is shown in Figure 1.

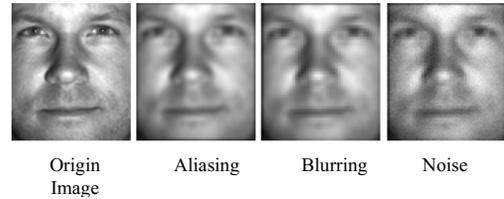


Figure 1. Stages of producing a low-resolution image

To make the low-resolution images the same size as high-resolution images, we use bicubic interpolation for recovering. Some images of the high-resolution gallery image set and low-resolution test image set are shown in Fig.2.



Figure 2. High-resolution image in gallery database and low-resolution images in test database

First we should prepare the data for training. As discussed before, we will start from the upper-left corner of the image with size of 6×4. These patch sizes capture small details well. The height and width will be increased separately by the multiple of 1.7 and 1.5, until the size of patch reaches quarter of the origin image. For the size of 192×168, we find 8428 patches at all.

For the selected patches, as talked before, we use AdaBoost to choose T best patches. Here we test our algorithm when $T = 10, 40, 70, 200$ to see the result. The top 3 patches we choose are shown in Fig.3.

3.2 Results

We conduct two experiments on the Extended Yale B database to validate our algorithm and compare our algorithm to [2].

In the first experiment, we compare our algorithms to [2] on different down-sample factor on horizontal and vertical direction, from 6 to 12. We test our algorithm with 10 patches and 40 patches. The result is shown in Table 1.

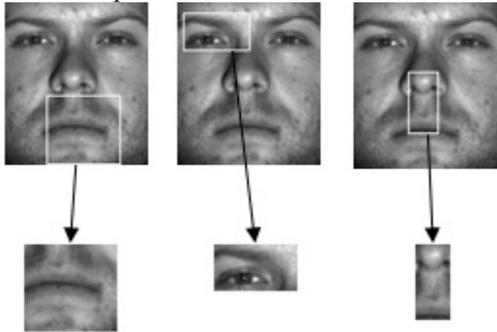


Fig 3. Top 3 patches we choose by AdaBoost

Table 1. Result compared to Yang et al [2] on different down-sample factor

Image Size	32×28	24×21	16×14
A.Y.Yang[2]	65.30%	55.39%	45.48%
BoostSparse with 40 Patches	77.70%	52.42%	38.04%
BoostSparse with 70 Patches	78.31%	53.28%	38.91%

From the result in Table 1, we find that when the quality of images degrades, the performance of our algorithm degrades faster than [2]. This is because that more local information lost when quality is worse. So unsurprisingly, we find that [2] which recognize from global perspective perform better on image of smaller size. However, when dealing with the low-resolution images which are down-sampled by 6 horizontally and vertically from high-resolution images, we can see that our BoostSparse algorithm exceeds [2] greatly. This means that for images which are not too bad in quality, our algorithm outperforms [2].

Then in the second experiment, we test our algorithm on different number of patches to find the number of

patches we need to reach fine performance. The result is shown in Table 2.

It is clear to see that we only need 40 patches to achieve good result. When more patches are involved, the enhancement of performance is limited. We can see from Table 2 that the recognition rate reaches maximum when involving 200 patches around. If we use more patches, the performance will start to degrade.

4. Discussion

The result from the experiment validates our idea which focuses on finding sparse representation of the local patch on the image. But there are still some open problems for

Table 2. Result on different patch number

Patch Number	Recognition Rate (%)
10	74.60
40	77.70
70	78.31
200	79.55
300	78.81
400	78.19

us to concentrate on. Whether prevailing texture feature could help our algorithm perform better is a valuable question. Besides, tighter connection to compressed sensing and latest research results on this field could help us to improve our algorithm on finding the sparse representation of patches more efficiently.

Acknowledgements

This work was supported by the Science Research Fund of MOE-Microsoft Key Laboratory of Multimedia Computing and Communication (Grant No.07122809), and the Science Research Fund of University of Science and Technology of China for Young Scholars.

Reference

- [1] S. Baker and T. Kanade. "Hallucinating faces". In *IEEE International Conference on Automatic Face and Gesture Recognition*, March 2000.
- [2] A. Y. Yang, J. Wright, Y. Ma, and S. S. Sastry, "Feature selection in face recognition: A sparse representation perspective," *submitted to PAMI*, August 2007.
- [3] D. L. Donoho. "For most large underdetermined systems of linear equations, the minimal ℓ_1 -norm solution is also the sparsest solution", *Comm. on Pure and Applied Math*, Vol. 59, No. 6, 2006. 3.

- [4] W. T. Freeman, E. C. Pasztor, and O. T. Carmichael. "Learning Low-Level Vision". *IJCV*, 40(1):25-47, 2000.
- [5] Paul Viola, Michael J. Jones, "Robust Real-Time Face Detection", *ICCV*, 2001.
- [6] O. Arandjelović and R. Cipolla. "A manifold approach to face recognition from low quality video across illumination and pose using implicit super-resolution". In Proc. *IEEE 11th Conference on Computer Vision (ICCV2007)*, Oct. 2007.
- [7] J. Wright, A. Ganesh, A.Y. Yang, and Y. Ma, "Robust Face Recognition via Sparse Representation", *Submitted to IEEE Trans. PAMI*, August 2007
- [8] M. Elad and A. Feuer. "Super-resolution reconstruction of image sequences". *PAMI*, 21(9):817-834, 1999
- [9] J. Yang, J. Wright, T. Huang, Y. Ma, "Image Super-Resolution as Sparse Representation of Raw Image
- [10] C. Liu, H. Y. Shum, and W. T. Freeman. "Face hallucination: theory and practice". *IJCV*, Vol. 75, No. 1, pp. 115-134, October, 2007.
- [11] P. H. Hennings-Yeomans, S. Baker, and B. Vijaya Kumar. "Simultaneous super-resolution and feature extraction for recognition of lowresolution faces". In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1-8, June 2008.
- [12] J. Yang, H. Tang, Y. Ma, T. Huang, "Face Hallucination via Sparse Coding", *ICIP*, 2008
- [13] P. J. Phillips, W. T. Scruggs, A. J. O'Toole, P. J. Flynn, K. W. Bowyer, C. L. Schott, M. Sharpe, "FRVT 2006 and ICE 2006 Large-Scale Results", *NISTIR*, 2007.
- [14] Timo Ahonen, Abdenour Hadid, Matti Pietikäinen, "Face Recognition with Local Binary Patterns", *ECCV 2004*, pp. 469-481, May 2004.