ARTICLE IN PRESS

Pattern Recognition ■ (■■■■) ■■■–■■■

Contents lists available at ScienceDirect

Pattern Recognition

journal homepage: www.elsevier.com/locate/pr

Face image classification by pooling raw features

Fumin Shen [a,*], Chunhua Shen [b], Xiang Zhou [a], Yang Yang [a], Heng Tao Shen [a,c]

[a] School of Computer Science and Engineering, University of Electronic Science and Technology of China, No. 2006, Xiyuan Ave, West Hi-Tech Zone, Chengdu 611731, PR China
[b] School of Computer Science, The University of Adelaide, SA5005, Australia
[c] School of Information Technology and Electrical Engineering, The University of Queensland, QLD 4072, Australia

ARTICLE INFO

Article history:
Received 23 December 2014
Received in revised form
16 August 2015
Accepted 11 January 2016

Keywords:
Face recognition
Image classification
Feature pooling

ABSTRACT

We propose a very simple, efficient yet surprisingly effective feature extraction method for face recognition (about 20 lines of Matlab codes), which is mainly inspired by spatial pyramid pooling in generic image classification. We show that, coupled with a linear classifier, features formed by simply pooling local patches over a multi-level pyramid can achieve state-of-the-art performance on face recognition. The simplicity of our feature extraction procedure is demonstrated by the fact that no learning is involved (except PCA whitening). It is shown that *multi-level spatial pooling* and *dense extraction of multi-scale patches* play critical roles in face image classification. The extracted facial features can capture strong structural information of individual faces with no label information being used. We also find that pre-processing on local image patches such as contrast normalization can have an important impact on the classification accuracy. In particular, on the challenging face recognition datasets of FERET and LFW-a, our method improves previous best results by large gaps. Promising results are also achieved on the general image classification database Caltech-101.

1. Introduction

Face recognition has been a long-standing research topic in computer vision. It remains a challenging problem despite the extensive research effort spent on this problem. These challenges are typically caused by various intra-class variations (e.g., face expressions, poses, ages, and image contaminations), or lack of sufficient training samples [1]. To achieve encouraging recognition accuracies, it is believed that the key is to find effective and robust representations of facial images. Conventional face image features in the literature include eigenfaces [2], Fisher-faces [3], and Laplacian faces [4]. These features work well on certain tasks. However, they often cannot cope with the aforementioned problems well such as image occlusions and large pose variations.

Recently, sparse representation based face classification has shown encouraging success [5,6]. Different from previous methods, which make decisions based on standard classifiers (e.g., k-nearest-neighbour or support vector machines), the classification rule of these methods is based on the minimum representation error of the probe image over a set of training samples or a dictionary learned from training images. As a typical example, the sparse representation based classifier (SRC) [5] linearly represents a probe image by all the training images under the L_1 norm based sparsity constraint. The success of SRC has induced a few sparse representation based approaches [6–9], which achieve state-of-the-art performance on face recognition with image corruptions [5,10], face disguises [6,9,11] and small-size training data [8,12].

Compared to these methods mainly working on holistic features, researchers found that face recognition based on local features or patches is often more robust to mis-alignment and occlusions. For example, histograms of local binary patterns (LBP) [13], Gabor features [14–16] and their fusions [17] have proven to improve the robustness of face recognition systems. Another simple way to extract local features is the modular approach, which first divides the entire facial image into several blocks and then aggregates the classification outputs that were made independently on each of these local blocks [5]. One of the widely used aggregation strategies might be voting methods [5,18,19]. Recently, Zhu et al. [20] proposed a linear representation based method, which fuses the results of multi-scale patches using boosting.

Local features or image patches have also been widely used in generic image classification. The bag-of-features (BoF) model, which represents an image as a collection of orderless local features, is the standard approach in image classification [21–23]. The spatial pyramid matching (SPM) method in [22,23] has made a remarkable success in image classification by explicitly incorporating spatial

* Corresponding author. Tel.: +86 15208131819; fax: +86 28 61831561.
E-mail addresses: fumin.shen@gmail.com (F. Shen), chunhua.shen@adelaide.edu.au (C. Shen), Johnfly2000@gmail.com (X. Zhou), dlyyang@gmail.com (Y. Yang), shenhengtao@hotmail.com (H.T. Shen).

http://dx.doi.org/10.1016/j.patcog.2016.01.010
0031-3203/

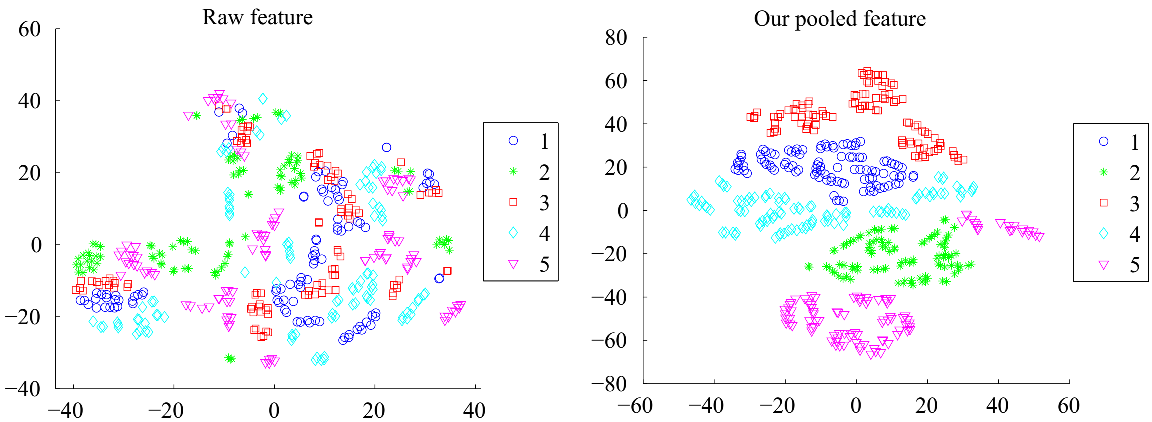

Fig. 1. Visualization of raw pixel features (left) and our pooled features (right) of face images by the unsupervised embedding method t-SNE [24] into 2D. Here the first 5 classes of the CMU-PIE dataset [25] are used, with each class having 100 face images. Data points belonging to the same class are marked with the same colour and marker. Note that no label information is used here. We can clearly see that using our feature representations, faces in the same class tend to be grouped into a cluster and different classes are much better separated. Best viewed in colour. (For interpretation of the references to colour in this figure caption, the reader is referred to the web version of this paper.)

geometry information into classification. SPM divides an image to increasingly fine spatial grids at different scales with $2^l \times 2^l$ cells at the lth level. Codes by encoding the local features (e.g., SIFT features) within each spatial cell over a pre-trained dictionary are pooled by average pooling or max-pooling. Despite the success of the spatial pooling model in image classification, it has rarely been used in face recognition problems.

Mainly inspired by the spatial pyramid pooling method, here we propose a simple feature extraction method for face recognition. Instead of pooling the encoded features over a pre-trained dictionary, *we compute the facial features by performing pooling directly on the local patches, which are densely extracted from the face image*. In other words, our pipeline removes the dictionary learning and feature encoding from the standard unsupervised feature learning based generic image classification pipeline. As a result, our feature extraction method is very simple, efficient and can be implemented in less than 20 lines of Matlab code. A surprising result is that, however, coupled with a linear classifier, our method outperforms almost all the most recently reported methods by a large margin on several benchmark face recognition datasets. The extracted facial features are shown to have strong structural information of individuals without using label information. Fig. 1 illustrates an example.

In summary, the contributions of this work mainly include:

1. We propose a new facial feature extraction method based on spatial pooling of local patches and apply it to face identification problems. For the first time, we advocate directly applying pooling on local patches.
2. The proposed simple yet very effective method can capture strong structural information of individual faces with no label information being used. The present method outperforms several recently reported face recognition methods on four tested public benchmark databases. Accuracy improvements of over 10% and 20% are obtained by our method over the previous best results on the FERET and LFW-a datasets, respectively. The proposed method also works well on the small training size, and in particular the single-sample-per-person (SSPP) face recognition problems.
3. Compared to other local feature (e.g., LBP, Gabor feature) or local patch (e.g., BoF model) based methods, the proposed pooling method is much simpler, and achieves close or even better results.

Moreover, through a thorough evaluation, we show that dense extraction of multi-scale patches and spatial pooling over a deep pyramid play critical roles in achieving high-performance face recognition. We also show that the pre-processing operations such as contrast normalization and polarity splitting can have important impacts on the classification performance as well.

2. Our method

In this section we present the details of the proposed method. Fig. 2 summarizes our feature extraction method.

2.1. Local patch extraction and pre-processing

Without loss of generality, suppose that a facial image is of $d \times d$ pixels. As the first step, we extract overlapped local patches of size $r \times r$ pixels with a step of s pixels. Set $l = \lfloor \frac{d-r}{s} + 1 \rfloor$, then each image is divided into $l \times l$ patches. Let each local patch be a row vector $\mathbf{x}$. We then perform normalization on $\mathbf{x}$ as: $\hat{x}_i = (x_i - m)/v$, where x_i is the ith element of x, and m and v are the mean and standard deviation of elements of $\mathbf{x}$. This operation contributes to local brightness and contrast normalization as in [26].

With each patch extracted from all training images contrast normalized, an optional step is to conduct dimensionality reduction on these patches by principal component analysis (PCA) [2]. One of the benefits by doing this is de-noising. The PCA step on the local patches will also prevent the dimensionality of the final pooled features from becoming too large. Later, in the experiments we show that transforming the patches to a low dimensional (e.g., 10D, to capture 90% of the energy) eigenspace is typically sufficient.

It has been shown in object recognition that the simple absolute value rectification [27] or polarity splitting [28] on the extracted features before pooling often improves the classification performance. Here we apply the polarity splitting in our algorithm directly on the normalized patches: we split the positive and negative components of each patch $\mathbf{x}$ into separate parts: $\max\{\mathbf{x}, 0\}$ and $\max\{-\mathbf{x}, 0\}$. These two parts are then concatenated to form a vector which is twice as long as the original vector.

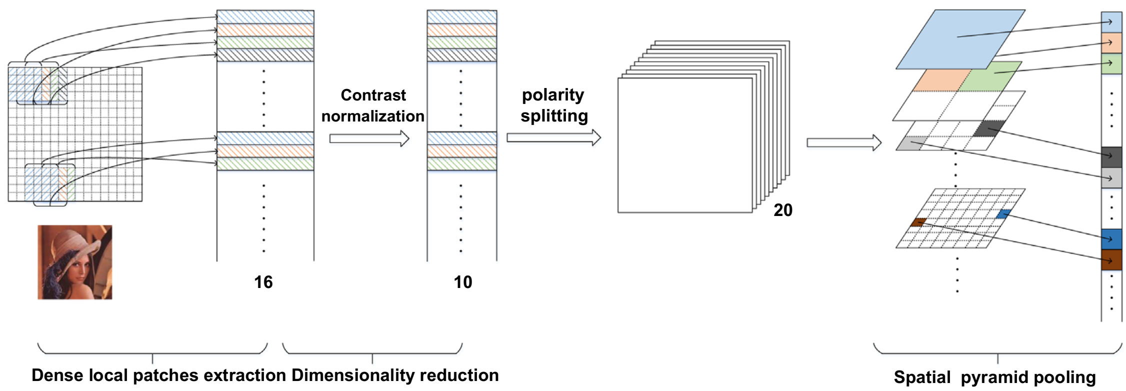

Fig. 2. Flowchart demonstrating the proposed feature extracting method. The patch size of 4×4 is used for illustration.

2.2. *Spatial pyramid pooling*

Then we come to the most important part of our algorithm: spatially pooling the pre-processed patches. Different from all the feature coding methods [23,28], which conduct spatial pooling on the encoded features over a pre-trained dictionary, *we propose to perform pooling directly on the local patches*. Suppose that we set the pooling pyramid of L levels to $\{c_1, c_2, \ldots, c_L\}$, such that the grid at level l have c_l cells at each dimension. Then we have in total $\sum_{l=1}^{L} c_l^2$ pooling cells for a 2D image.

The dimension of the processed patches can be very low, which makes it possible to apply a much larger number of pooling levels. As an example, after applying PCA, the dimension of each local patch is about 10. While for unsupervised feature learning or bag-of-features based methods [23,28], the encoded feature dimension for each extracted local feature is the same as the dictionary size, which is usually a few thousand—two orders of magnitude larger. We show in the next section that, *for the proposed method, a large number of pooling pyramid levels are critical to achieve a high accuracy in face recognition.*

Those popular pooling methods, e.g., average pooling and max-pooling, can be used in our algorithm. They can easily be performed by the average- or max-operation along each normalized pixel of the patches (again, not on the encoded features but directly on the pixels), in the pooling cell:

$$f_i = \sum_j x_i^{(j)} / m, \quad \text{(average pooling)}$$

$$f_i = \max_j x_i^{(j)} \quad \text{(max pooling).}$$

Here $x_i^{(j)}$ is the ith element of the jth local patch in the current pooling cell. m is the number of patches in the pooling cell and $\mathbf{f} = [f_1, \ldots, f_i, \ldots]$ is the pooled feature for that cell. Note that those overlapped pixels in neighbouring patches generally have different pixel values after applying contrast normalization. A comparison of these two pooling methods is given in Section 3.

Images (patches) of multiple scales have been widely used in computer vision to incorporate information at different scales. For our method, we have also found that extracting multi-scale patches almost always improves the classification accuracy. Therefore, we have used image patches of different sizes (e.g., $4 \times 4, 6 \times 6$ or 8×8 pixels) and concatenated all the pooled features on multi-scale patches to form the final feature vector.

Thus far, we have discussed how to extract the final features. In practice, we find that standardizing the extracted features before training the linear classifier almost always leads to superior results (see Table 2). To perform feature standardization, we subtract the mean and divide by the standard deviation of each feature in the training set, following [28].

2.3. *Linear multi-class classification*

In this work, we feed the extracted features to a linear classifier to recognize the probe face image. A simple ridge regression based multi-class classifier was used in [29]. We use the same linear classifier for its computational efficiency. The ridge regression classifier has a closed-form solution, which makes the training even faster than the specialized linear support vector machine (SVM) solver Liblinear[30]. Despite its simplicity, the classification performance of this ridge regression approach is on par with linear SVM [29]. Another benefit of this classifier is that, compared to the one-versus-all SVM, it only needs to compute the classification matrix $\mathbf{W}$ once.

We summarize our method in Algorithm 1. It is very simple and the entire algorithm can be implemented within 20 lines of Matlab codes (see Appendix A).

Algorithm 1. Face recognition with spatial pooling of local patches.

Input: Training images and test samples

Define the image patch sizes and pooling pyramids;

for *each patch size* **do**

- Densely extract local patches for both training and testing data;
- Conduct contrast normalization;
- Apply dimensionality reduction on the patches by using PCA;
- Apply polarity splitting on the obtained features;
- Perform spatial pooling over the pyramid grids;

Concatenate the obtained features with all patch sizes;

Standardize the obtained feature data;

Train a multi-class linear classifier;

Output: The class label of a test face using the trained classifier

3. Model selection

In this section, we test the impact of the components of the proposed algorithms. The key parameters are the pooling pyramid levels, and the multiple different patch sizes. The evaluation is conducted on the LFW-a dataset [31], and all the images are down-sampled to 64×64 pixels. The dataset's description can be found

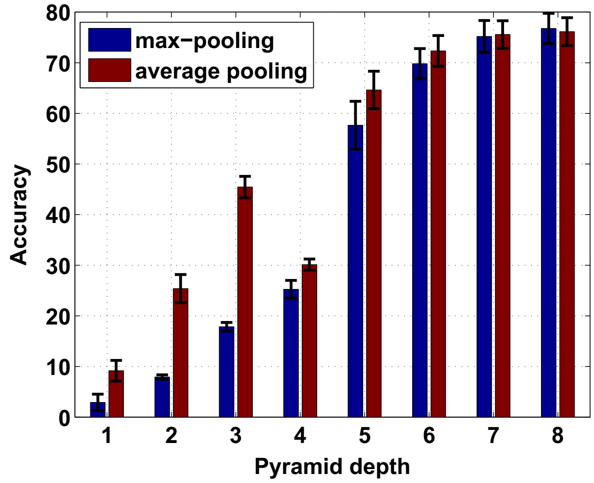

Fig. 3. Classification accuracy (%) with the varying pooling pyramid levels. Here the 8-level pyramid is set to {1, 2, 4, 6, 8, 10, 12, 15}.

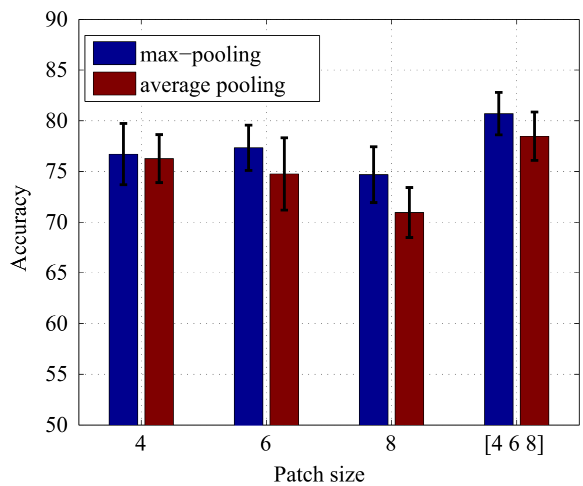

Fig. 4. Accuracy (%) with the varying patch size. 4, 6, 8 means patch sizes of 4×4, 6×6, 8×8 pixels. Here we use a 8-level pyramid for pooling.

in Section 4. We set the number of training and testing samples per subject to 5 and 2, respectively. All the results reported in this section are based on 5 independent random training/testing data splits. We empirically reduce the dimension of image patches to 10 by PCA, which preserves about 90% of the eigenvalue energy.

3.1. Multi-level spatial pooling

First, we examine the importance of pooling for face recognition. For this purpose, we test different pooling pyramids: from the 1-level pyramid {1} to a maximum 8-level pyramid $\{1, 2, 4, 6, 8, 10, 12, 15\}$. With this 8-level pyramid, pooling is performed on regular grids of 1×1, 2×2, …, 15×15 and the obtained pooled features are concatenated altogether.

We also test the two popular pooling strategy: max-pooling and average pooling. As the default setup, local patches of size 4×4 pixels are densely extracted with stride step of 1 pixel on the input raw images. Thus we in total extract 3721 local patches for each image at this setting.

The recognition accuracies for different pyramid settings are shown in Fig. 3. It is clear that, for both max-pooling and average pooling, the face recognition performance is significantly and almost consistently improved as the number of pooling levels increases. Pooling effectively encodes multiple levels of spatial information, which appears extremely useful for classification. From this experiment, it seems that a 8-level pyramid is sufficient. *This is in contrast to the SPM for generic image classification, for which one usually does not observe significant performance improvement beyond 3 levels of pooling as shown in* [22].

Fig. 3 also shows an interesting result that average pooling achieves much better results than max-pooling when no more than 6 pooling levels are applied. This might be due to the fact that with fewer pooling levels, max-pooling has discarded too much information while for average pooling, all pixels in the pooling cell contribute to the final result.

With the pyramid levels being larger than 6, max pooling achieves close or slightly higher accuracy than average pooling. This is very different to the results reported in generic image classification, where max pooling used in the BoF models usually leads to superior performance than average pooling with a pyramid of only 3 levels [32, 23].

3.2. Dense extraction of local patches

We then test the impact of patch sizes on the recognition accuracy. We use different patch sizes of $\{4 \times 4, 6 \times 6, 8 \times 8\}$ pixels, and combinations of them. The comparative results are reported in Fig. 4. Here we can clearly see that smaller patch sizes of 4×4 and 6×6 pixels lead to better results than the patch size of 8×8. Similarly, smaller receptive fields (patch sizes) are critical for achieving high performance in the generic image classification with unsupervised feature learning [26]. It is also clear that the combination of the features extracted with different patch sizes achieves significant performance improvements. Specifically, the combination use of these three patch sizes obtains at least 3.4% and 2.2% accuracy gains for max-pooling and average pooling, respectively.

All the previous evaluations are based on dense extractions of local patches with a stride step size of 1 pixel. Table 1 shows that if we use a larger step size, the performance deteriorates. Clearly, smaller steps are favoured. Again, the same observation is made in [26] for generic image classification.

Through the evaluation of pooling pyramid depth and patch extraction, we can conclude that *multiple-level pooling and dense extraction of multi-scale image patches are critical for achieving high classification accuracies in face recognition*. From the perspective of dimensionality, higher dimensional features (due to the deeper pooling pyramid and multi-scale image patches) lead to better performance. This is consistent with the recent results reported in [33]. The intuition is that the high-dimensional feature vectors encode rich information for the sequential linear classification. The high-dimensional vectors may contain noises but the linear classifier is able to select the relevant features and suppress the useless ones.

3.3. Feature pre-processing

In this section, we evaluate the effects of pre-processing steps before pooling features in our Algorithm 1: contrast normalization and polarity splitting. The evaluation is conducted on both of the LFW-a and FERET datasets with the image size of 64×64 pixels. See the datasets' description in Section 4. Table 2 reports the recognition accuracies of our method without each of these operations and with all of them.

Please cite this article as: F. Shen, et al., Face image classification by pooling raw features, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.01.010

Table 1
Impact of the step size on the recognition rate (%) with different patch sizes. A 5-level pyramid of {1, 2, 4, 6, 8} is used here since the number of patches is only 16-by-16 at most when using a step size of 4 pixels.

Step size	1 pixel	2 pixels	4 pixels
4×4 patch	64.0 ± 4.2	50.3 ± 2.4	27.4 ± 1.9
6×6 patch	66.5 ± 3.6	56.3 ± 3.1	35.3 ± 2.9
8×8 patch	62.6 ± 2.4	60.6 ± 3.0	41.1 ± 3.0

Table 2
Accuracy (%) of our method without the operation of contrast normalization, polarity splitting or feature standardization and with all of them. –: this operation is not applied; √: applied. Note that the first two operations are performed before pooling while standardization is conducted on the pooled features before classification. The combination of patch sizes $\{4 \times 4, 6 \times 6, 8 \times 8\}$ is used.

Operation	FERET				LFW-a			
Contrast Norm.	–	√	√	√	–	√	√	√
Polarity splitting	√	–	√	√	√	–	√	√
Standardization	√	√	–	√	√	√	–	√
Accuracy	88.9 ± 1.2	92.6 ± 0.9	92.0 ± 0.9	93.4 ± 0.6	67.3 ± 0.7	78.2 ± 1.8	73.0 ± 2.8	80.7 ± 2.1

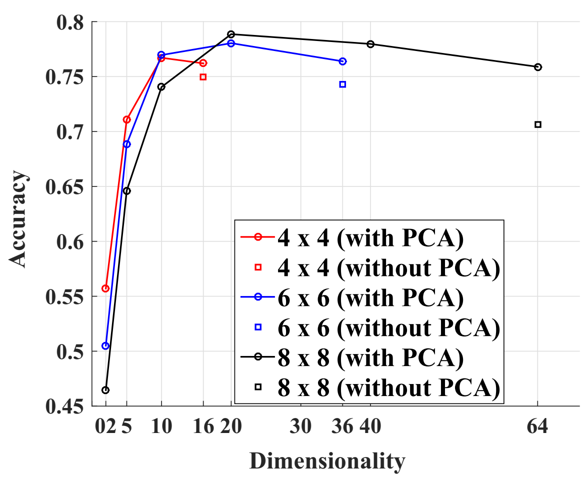

Fig. 5. PCA dimension versus accuracy using 4×4, 6×6 and 8×8 patches. The results of raw patches without applying PCA are marked as rectangles.

We can clearly see that there is a significant benefit of contrast normalization on both datasets. Specifically, improvements of 4.5% and 13.4% accuracies are achieved by the simple normalization step on the FERET and challenging LFW-a, respectively. Polarity splitting of features also slightly improves the recognition performance, which increases the accuracy by 0.8% on FERET and 2.5% on LFW-a. We also find that standardizing the extracted features before training a linear classifier almost always helps the final classification. In this experiment, feature standardization improves the accuracy by 1.4% and 7.7% on the two datasets, respectively.

Last, we will test the impact of the reduced dimensionality on the recognition accuracy. Three different patch sizes (4×4, 6×6 and 8×8) are tested. The results of PCA dimension versus accuracy are shown in Fig. 5. The importance of applying PCA on raw patches is clearly shown. The original patches perform even worse than the 10-dim PCA features.

All these feature processing steps are extremely simple. However, as we can see, they play critical roles in practical face recognition tasks. The experimental results suggest that these operations should always be applied in our algorithm.

4. Experimental results

In this section, we focus on the face identification problems. Our method is tested on four public facial image datasets, including AR [34], CMU-PIE [25], FERET [35] and the challenging uncontrolled faces, LFW [36]. The compared face recognizers are the sparse representation based classifier (SRC) [5]), robust sparse coding (RSC) [9]), superposed SRC (SSRC) ([8]), local patch based Volterra [19] and multi-scale patch based collaborative representation based classification (MSPCRC) [20]. These methods are reported to achieve the state-of-the-art results on the face identification problems in most recent years. In Section 4.3, we also compare our method to the popular generic image classification method locality-constrained linear coding (LLC [37]) with multi-level spatial pyramid pooling.

As the authors recommend, we set the parameter λ to 0.001, 0.001, 0.005 and 0.001 for SRC, RSC, SSRC and MSPCRC, respectively. For Volterra [19], we use the linear kernel since it always achieves better results than the quadratic one. For RSC, Volterra and MSPCRC, we use the implementation code provided by the original authors. In addition, we also compare with several recently published results.

According to the evaluation in the last section, for our two methods, we use max-pooling with a 8-level pooling pyramid {1, 2, 4, 6, 8, 10, 12, 15} and the combination of patch sizes $\{4 \times 4, 6 \times 6, 8 \times 8\}$ pixels as the default setting.[1] All of the patches are transformed to a 10D by applying PCA.

4.1. AR

The AR dataset [34] consists of over 4000 facial images from 126 subjects (70 men and 56 women). For each subject 26 facial images were taken in two separate sessions. The images exhibit a number of variations including various facial expressions (neutral, smile, anger, and scream), illuminations (left light on, right light on and all side lights on), and occlusion by sunglasses and scarves. Of the 126 subjects available 100 have been selected in our experiments (50 males and 50 females).

Since AR has been used very often to evaluate face recognition algorithms, we compare our method to several published state-of-

[1] For images of size 32×32 (or smaller), we only use one small patch size $\{4 \times 4\}$ due to the numerical problem with large-size patches.

Table 3
Comparison with recent state-of-the-art results (%) on AR with four different settings. For the first six methods, we quote the best reported results from the corresponding published papers, with the same experiment setting.

Situation	Clean	Sunglasses	Scarves	All
SRC [5]	92.9	87.0	57.5	–
RSC [9]	96.0	99.0	97.0	–
DRDL [38]	95.0	–	–	–
L_2 [39]	–	78.5	79.5	95.9
SSRC [8]	–	90.9	90.9	98.6
FDDL [6]	92.0	–	–	–
Volterra	90.9	96.1	92.1	87.5
Ours	**99.3**	**100.0**	**98.5**	**98.7**

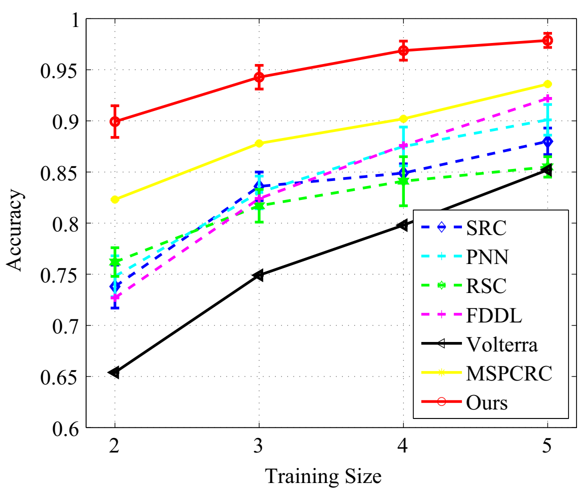

Fig. 6. Recognition accuracy on AR with images resized 32 × 32 pixels. Results are based on 20 independent runs. The results for PNN (patch based nearest neighbour) [18], Volterra and MSPCRC are quoted from [20] with identical experiment settings. Again, ours outperforms the previous best results by a large margin.

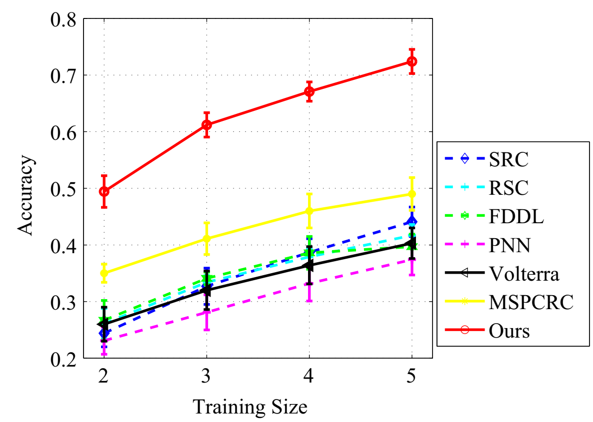

Fig. 7. Recognition accuracy on LFW with downsampled 32 × 32 images. Results are based on 20 independent runs. The results for PNN, Volterra and MSPCRC are quoted from [20] with identical settings.

the-art results on this dataset with four different situations. For the 'All' situation, all the 13 samples in the first session are used for training and the other 13 images in the second session for test. For the 'clean' situation, the 7 samples in each session with only illumination and expression changes are used. For the 'sunglasses' or 'scarf' situation, 8 clean samples from two sessions are for training and 2 images with sunglasses or scarf disguises are for test. All the images are resized to 64 × 64 pixels. The results are reported in Table 3.

It is clear that our method achieves better results than all other algorithms in all of the four situations. Specifically, our method outperforms the best of all other methods by 3.3%, 1.0%, 1.5%, and 0.1% in the 'clean', 'sunglasses', 'scarf' and 'All' situation, respectively. In particular, our algorithm obtains a perfect recognition with the sunglasses disguise. To our knowledge, ours is the best reported result in this scenario.

Following the settings in [20], we further test the small-training-size problem on this dataset. Among the samples with only illumination and expression changes in each class, we randomly select 2–5 images of the first session for training and 3 of the second session for test. All the images are resized to 32 × 32 pixles. We independently run the experiments for 20 times and report the average results in Fig. 6. We can see that our method outperforms all other methods by large margins on this small-training-size problem. The local patch based PNN and Voltera methods achieve inferior results to the multi-scale based MSPCRC. The holistic representation based methods such as SRC [5], RSC [9] and FDDL [6] do not perform as well as the local patches based MSPCRC, either. The superior performance of our method and MSPCRC conforms the importance of extracting features based on multi-scale local patches.

4.2. CMU-PIE

The CMU-PIE dataset [25] is widely used for face recognition. We use this dataset to compare several popular feature extraction methods (including Eigenfaces [2], Fisherfaces [3], Laplacianfaces [4] and its orthogonal variation OLAP [40]) and more recent methods such as SRC, MSPCRC, Volterra and the kernel ridge regression KRR [41].

In our experiments, we use the data and settings provided by the authors of [40]. The data includes 170 samples each subject from five near frontal poses (C05, C07, C09, C27, C29). From 2 to 30 samples are used for training and all the rest are for testing. Table 4 compares the mean results of our method and several published results based on 10 independent runs.

Table 4 clearly demonstrates the advantage of the proposed method. In particular, *our method achieves* 72.2% *and* 99.7% *accuracy with* 2 *and* 30 *training samples, which are higher than those of all other methods by about* 15.5% *and* 1.4%, *respectively*. Volterra performs the second best on this dataset. The recent state-of-the-art algorithms SRC, RSC and MSPCRC do not show better performance than OLAP on this relatively simple dataset.

Please cite this article as: F. Shen, et al., Face image classification by pooling raw features, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.01.010

Table 4
Comparison of average accuracy (%) on CMU-PIE with varying training size. For the first six methods, we quote the best reported results from the cited papers. SRC, MSPCRC and our methods are conducted on images of 64×64 pixels.

Training size	2	5	10	20	30
Eigenfaces [40]	–	30.1	44.3	61.9	72.1
Fisherfaces [40]	–	68.5	77.6	84.6	92.2
Laplacianfaces [40]	–	69.2	78.9	85.9	92.8
OLAP [40]	–	78.6	88.6	93.5	95.2
KRR [41]	–	73.6	86.9	94.0	96.0
Volterra [19]	57.0	80.8	90.9	95.4	98.3
SRC	46.0	72.3	85.7	93.1	95.1
RSC	43.7	72.3	85.4	92.5	94.5
MSPCRC	51.5	74.1	87.4	93.2	94.9
Ours	**72.2**	**94.4**	**98.5**	**99.5**	**99.7**

Table 5
Recognition accuracies (%) on FERET with standard protocols.

Method	Fb (expression)	Fc (illumination)	DupI (aging)	DupII (aging)
HGPP [15]	97.5	**99.5**	79.5	77.8
Zou's method [16]	99.5	**99.5**	85.0	79.5
Tan's method [17]	98.0	98.0	**90.0**	85.0
LLC	99.4	96.9	82.3	77.8
Ours	**99.7**	98.5	**89.1**	**85.5**

Table 6
Average accuracy (%) on FERET using 5 training samples each class with images of 32×32 and 64×64 pixels. Results are based on 10 independent runs.

Image size	MSPCRC	Volterra	SRC	RSC	SSRC	Ours
32×32 pixels	41.2 ± 1.5	46.2 ± 1.6	75.3 ± 1.7	73.3 ± 1.8	84.1 ± 1.5	$\mathbf{87.0 \pm 1.2}$
64×64 pixels	42.3 ± 2.8	50.4 ± 2.3	73.8 ± 2.0	73.4 ± 2.6	82.1 ± 1.6	$\mathbf{93.4 \pm 0.6}$

Table 7
Comparison of our method with LLC on LFW-a with 5 training samples each class. Results are reported based on 10 independent runs.

Method	32×32 pixels	64×64 pixels
LLC	57.7 ± 1.8	72.5 ± 2.1
Ours	$\mathbf{72.4 \pm 2.1}$	$\mathbf{80.7 \pm 2.1}$

4.3. *Face recognition from single sample per person on feret*

Local feature (or patch) based face recognition methods tends to show superior results to the holistic feature (or patch) based ones, especially on the single sample per person (SSPP) problem. Since only one training sample for each person is registered in the training data, many holistic methods (e.g., SRC) tend to fail in this case. Due to this, in this experiment we test our method against several representing local feature based methods, which were reported to achieve good results on SSPP problems. These methods include Histogram of Gabor Phase Patterns (HGPP [15]), Zou's method [16] and Tan's method [17].

The public FERET dataset is used. The FERET dataset [35] is an widely used standard face recognition benchmark set provided by DARPA. Since the images of FERET dataset are acquired in multiple sessions during several years, they have complex intraclass variability. Taking the subset Fa with one image for each of 1196 subjects as a gallery, Fb (expression variations) and Fc (illumination variations), and DupI and DupII (age variations) are taken as the probe subsets. Each image is cropped based on the locations of eyes and then normalized to 150×130 pixels. From Table 5, we can see that our approach achieves very competitive results with the local feature based methods, even if ours is much simpler.

We also compare our method with the popular BoF approach locality-constrained linear coding (LLC [37]), which can be regarded as an efficient version of the ScSPM method [23]. The dictionary is generated by K-means on 6×6 patches with 1024 basis. We set the regularization parameter $\lambda = 0.001$ and the pyramid as [1,2,4]. For fair comparison, contrast normalization and polarity splitting are performed for LLC as in our method, which are shown to improve the results.

Compared to the BoF model LLC, our method performs much better by excluding the dictionary training and feature encoding procedures. This is mainly due to the fact that our method can involve much more levels of pooling than LLC. The encoding procedure (with hundreds or thousands of dictionary basis) makes the dimension of the learned feature too large to conduct high-level pooling. Spatial pooling are shown to be more critical than feature encoding, at least on the face recognition problems.

Please cite this article as: F. Shen, et al., Face image classification by pooling raw features, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.01.010

Table 8
Average classification rates (%) on Caltech-101, using 30 training samples for each class.

Method	ScSPM [23]	LLC [37]	Ours	Ours
Feature	SIFT	SIFT	Raw pixels	SIFT
Accuracy	73.2	73.4	**75.2**	**75.4**

To further evaluate our methods on the pose problem in face recognition, we further conduct an experiment using a subset of FERET which includes 200 subjects. Each individual contains 7 samples, exhibiting facial expressions, illumination changes and up to 25 degrees of pose variations. It is composed of the images whose names are marked with 'ba', 'bj', 'bk', 'be', 'bf', 'bd' and 'bg'. The images are cropped and resized to 80×80 pixels [42]. We randomly choose 5 samples from each class for training and the remaining 2 samples for testing. The mean results of 10 independent runs with images down-sampled to 32×32 and 64×64 pixels are reported in Table 6.

We can clearly see that our method achieves the highest recognition rate. In particular, with image size of 64×64 pixels, *our method obtains an accuracy of* 93.4%, *which is much higher than the second best* (82.1% *of SSRC* [8]) *by* 11.3%. RSC achieves similar results with SRC. The patch based MSPCRC and Volterra do not perform well, which may be because they are incapable to cope with the pose variations in this dataset.

4.4. LFW-a

Following the settings in [20], here we use LFW-a, an aligned version of LFW using commercial face alignment software [31]. A subset of LFW-a including 158 subjects with each subject more than 10 samples are used. The images are cropped to 121×121 pixels. From 2 to 5 samples are randomly selected for training and another 2 samples for testing. For a fair comparison, all the images are finally resized to 32×32 pixels as in [20]. Fig. 7 lists the average results of 20 independent runs.

Consistent with the previous results, *our method outperforms all other algorithms by even larger gaps on this very challenging dataset*. The representation based methods SRC, RSC and FDDL do not show significant differences and perform sightly better than the patch based nearest neighbour classifier PNN. We obtain higher accuracies than the second best method MSPCRC by $14.4\% \sim 23.4\%$. Actually if we use a larger image size 64×64, the accuracy of our method will increase to 80.7%, while MSPCRC is not able to improve the accuracy (47.7%) in this case. In our experiments, we find that for our method, a large image size almost always achieves better results than using a smaller one.

Table 7 compares our method with the BoF method LLC. It can be seen that the proposed pooling method outperforms the BoF model LLC by 14.7% and 7.2% with 32×32 and 64×64 images, respectively.

4.5. Comparison to BoF on general image classification dataset Caltech-101

In this section, we evaluate our feature extraction method on the general image classification tasks. The Caltech-101 database is used. The Caltech-101 dataset contains 101 classes, with 31–800 images in each class. We followed the common experiment setup: 30 images are randomly selected for training and the rest for testing. Each image is resized to 97×97 for our method. The popular BoF model ScSPM [23] and LLC [37]are taken into comparison. We report the results based on 10 independent training/test data splits. From Table 8, we can clearly see that the proposed simple feature extraction method (with both raw pixels and SIFT features) achieves better classification performance than the BoF model ScSPM and LLC. This results show that our method can also applied in the general image classification problems. Compared to the popular image feature learning and encoding methods, one major advantage of our method is that it has no need for dictionary training or feature encoding.

5. Conclusion

In this work, we have proposed an extremely simple yet effective face recognition method based on spatial pooling of local patches. The proposed feature extraction method has two major components: local patch extraction and spatial pooling, omitting the two important parts in the BoF model: dictionary learning and feature encoding. We show that *dense extraction* and *multi-level pooling* are critical for local patch based face recognition. In addition, the pre-processing step of contrast normalization is also shown to have an important impact on the classification accuracy.

The extracted facial features by the proposed method demonstrate strong characteristics of clusters without using the label information. Despite its simplicity, the proposed method surprisingly outperforms many recent state-of-the-art methods (both holistic and local patch based) by large margins on four public benchmark datasets. The proposed method also works better than the BoF model LLC (with the additional feature encoding step) on the problems of both face recognition (FERET and LFW-a) and general image classification (Caltech-101). The MATLAB codes for the proposed method can be found in Appendix A.

Due to its simplicity and efficacy, the proposed feature extraction method can potentially be applied in many other vision tasks, such as object classification [43] and image retrieval [44,45]. How to adjust the feature extraction model in these tasks deserves further studies.

Conflict of interest

None declared.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China under Project 61502081 and Project 61572108.

Appendix A. MATLAB codes (20 lines) of our feature extraction method

```
function xp = fea_pooling(x, rfSize, eigvector, Pyramid)
% x: an n-by-n image
% rfSize: the patch size
% Pyramid: e.g., Pyramid = [1 1;2 2; 4 4; 6 6; 8 8; 10 10; 12 12; 15 15];
% eigvector: the pre-computed PCA projection matrix on training data
% xp: pooled feature
patches = im2col(x, [rfSize rfSize])';
prows = size(x,1)-rfSize+1; pcols = size(x,2)-rfSize+1;
% contrast normalization
patches = bsxfun(@rdivide, bsxfun(@minus, patches, mean(patches,2)),...
                  sqrt(var(patches,[],2)+10));
% applying PCA
patches = patches*eigvector;
% fliping
patches = [ max(patches, 0), -min(patches, 0) ];
% pooling
patches = reshape(patches, prows, pcols, size(eigvector,2)*2);
xp = [];
for lev = 1 : size(Pyramid,1);
    nRow = Pyramid(lev,1); nCol = Pyramid(lev,2);
    r_bin = round(prows/nRow); c_bin = round(pcols/nCol);
    for i_row = 1 : nRow
        for i_col = 1 : nCol
            r_bound = i_row*r_bin; c_bound = i_col*c_bin;
            if i_row == nRow, r_bound = prows;end
            if i_col == nCol, c_bound = pcols;end
            tem = max(max(patches(((i_row-1)*r_bin+1):r_bound,...
                                 ((i_col-1)*c_bin+1):c_bound,:),[],1),[],2);
            xp = [xp, tem(:)'];
        end
     end
end
```

References

[1] S.Z. Li, A.K. Jain, Handbook of Face Recognition, Springer, London, 2011.

[2] M. Turk, A. Pentland, Eigenfaces for recognition, J. Cogn. Neurosci. 3 (1) (1991) 71–86.

[3] P.N. Belhumeur, J.a.P. Hespanha, D.J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, IEEE Trans. Pattern Anal. Mach. Intell. 19 (1997) 711–720.

[4] X. He, S. Yan, Y. Hu, P. Niyogi, H.-J. Zhang, Face recognition using Laplacianfaces, IEEE Trans. Pattern Anal. Mach. Intell. 27 (3) (2005) 328–340.

[5] J. Wright, A.Y. Yang, A. Ganesh, S.S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Trans. Pattern Anal. Mach. Intell. 31 (2009) 210–227.

[6] M. Yang, L. Zhang, X. Feng, D. Zhang, Fisher discrimination dictionary learning for sparse representation, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 543–550.

[7] W. Deng, J. Hu, J. Guo, Extended SRC: undersampled face recognition via intra-class variant dictionary, IEEE Trans. Pattern Anal. Mach. Intell. 34 (9) (2012) 1864–1870.

[8] W. Deng, J. Hu, J. Guo, In defense of sparsity based face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 399–406.

Please cite this article as: F. Shen, et al., Face image classification by pooling raw features, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.01.010

[9] M. Yang, L. Zhang, J. Yang, D. Zhang, Robust sparse coding for face recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 625–632.
[10] F. Shen, C. Shen, R. Hill, A. van den Hengel, Z. Tang, Fast approximate l_∞ minimization: speeding up robust regression, Comput. Stat. Data Anal. 77 (0) (2014) 25–37.
[11] F. Shen, C. Shen, A. van den Hengel, Z. Tang, Approximate least trimmed sum of squares fitting and applications in image analysis, IEEE Trans. Image Process. 22 (5) (2013) 1836–1847.
[12] F. Shen, Z. Tang, J. Xu, Locality constrained representation based classification with spatial pyramid patches, Neurocomputing 101 (0) (2013) 104–115.
[13] T. Ahonen, A. Hadid, M. Pietikainen, Face description with local binary patterns: application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 28 (12) (2006) 2037–2041.
[14] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. von der Malsburg, R. Wurtz, W. Konen, Distortion invariant object recognition in the dynamic link architecture, IEEE Trans. Comput. 42 (3) (1993) 300–311.
[15] B. Zhang, S. Shan, X. Chen, W. Gao, Histogram of Gabor phase patterns (HGPP): a novel object representation approach for face recognition, IEEE Trans. Image Process. 16 (1) (2007) 57–68.
[16] J. Zou, Q. Ji, G. Nagy, A comparative study of local matching approach for face recognition, IEEE Trans. Image Process. 16 (10) (2007) 2617–2628.
[17] X. Tan, B. Triggs, Fusing Gabor and LBP feature sets for kernel-based face recognition, in: International Workshop on Analysis and Modeling of Faces and Gestures, 2007, pp. 235–249.
[18] R. Kumar, A. Banerjee, B. Vemuri, H. Pfister, Maximizing all margins: pushing face recognition with kernel plurality, in: Proceedings of the IEEE International Conference on Computer Vision, 2011, pp. 2375–2382.
[19] R. Kumar, A. Banerjee, B.C. Vemuri, H. Pfister, Trainable convolution filters and their application to face recognition, IEEE Trans. Pattern Anal. Mach. Intell. 34 (7) (2012) 1423–1436.
[20] P. Zhu, L. Zhang, Q. Hu, S. Shiu, Multi-scale patch based collaborative representation for face recognition with margin distribution optimization, in: Proceedings of the European Conference on Computer Vision, 2012, pp. 822–835.
[21] K. Grauman, T. Darrell, The pyramid match kernel: discriminative classification with sets of image features, in: Proceedings of the IEEE Conference on Computer Vision, vol. 2, 2005, pp. 1458–1465.
[22] S. Lazebnik, C. Schmid, J. Ponce, Beyond bags of features: spatial pyramid matching for recognizing natural scene categories, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2006, pp. 2169–2178.
[23] J. Yang, K. Yu, Y. Gong, T. Huang, Linear spatial pyramid matching using sparse coding for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 1794–1801.
[24] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (2008) 2579–2605.
[25] T. Sim, S. Baker, M. Bsat, The CMU pose, illumination, and expression database, IEEE Trans. Pattern Anal. Mach. Intell. 25 (2003) 1615–1618.
[26] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of the International Conference on Artificial Intelligence and Statistics, 2011, pp. 215–223.
[27] K. Jarrett, K. Kavukcuoglu, M. Ranzato, Y. LeCun, What is the best multi-stage architecture for object recognition? in: Proceedings of the IEEE Conference on Computer Vision, 2009, pp. 2146–2153.
[28] A. Coates, A. Ng, The importance of encoding versus training with sparse coding and vector quantization, in: Proceedings of the International Conference on Machine Learning, 2011, pp. 921–928.
[29] Y. Gong, S. Lazebnik, Comparing data-dependent and data-independent embeddings for classification and ranking of internet images, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 2633–2640.
[30] R. Fan, K. Chang, C. Hsieh, X. Wang, C. Lin, Liblinear: a library for large linear classification, J. Mach. Learn. Res. (2008) 1871–1874.
[31] L. Wolf, T. Hassner, Y. Taigman, Similarity scores based on background samples, in: Proceedings of the Asian Conference on Computer Vision, 2010, pp. 88–97.
[32] Y. Boureau, F. Bach, Y. LeCun, J. Ponce, Learning mid-level features for recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 2559–2566.
[33] D. Chen, X. Cao, F. Wen, J. Sun, Blessing of dimensionality: high-dimensional feature and its efficient compression for face verification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 3025–3032.
[34] A. Martinez, R. Benavente, The AR face database, CVC, Technical Report.
[35] P. Phillips, H. Wechsler, J. Huang, P. Rauss, The FERET database and evaluation procedure for face-recognition algorithms, Image Vis. Comput. 16 (5) (1998) 295–306.
[36] G.B. Huang, M. Ramesh, T. Berg, E. Learned-Miller, Labeled faces in the wild: a database for studying face recognition in unconstrained environments, Technical Report 07-49, University of Massachusetts, Amherst, October 2007.
[37] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, Y. Gong, Locality-constrained linear coding for image classification, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3360 –3367.
[38] L. Ma, C. Wang, B. Xiao, W. Zhou, Sparse representation for face recognition based on discriminative low-rank dictionary learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012, pp. 2586–2593.
[39] Q. Shi, A. Eriksson, A. vanden Hengel, C. Shen, Is face recognition really a compressive sensing problem? in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2011, pp. 553–560.
[40] D. Cai, X. He, J. Han, H.-J. Zhang, Orthogonal Laplacianfaces for face recognition, IEEE Trans. Image Process. 15 (11) (2006) 3608–3614.
[41] S. An, W. Liu, S. Venkatesh, Face recognition using kernel ridge regression, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2007, pp. 1–7.
[42] J. Yang, J. Yang, A. Frangi, Combined fisherfaces framework, Image Vis. Comput. 21 (12) (2003) 1037–1044.
[43] F. Shen, C. Shen, W. Liu, H.T. Shen, Supervised discrete hashing, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, Boston, 2015, pp. 37–45.
[44] F. Shen, C. Shen, Q. Shi, A. van den Hengel, Z. Tang, H.T. Shen, Hashing on nonlinear manifolds, IEEE Trans. Image Process. 24 (6) (2015) 1839–1851.
[45] F. Shen, W. Liu, S. Zhang, Y. Yang, H. Tao Shen, Learning binary codes for maximum inner product search, in: The IEEE International Conference on Computer Vision (ICCV), 2015, pp. 4148–4156.

Fumin Shen received his Bachelor degree at 2007 and Ph.D. degree at 2014 from Shandong University and Nanjing University of Science and Technology, China, respectively. Now he is a lecturer of University of Electronic Science and Technology of China. His major research interests include computer vision and machine learning, including face recognition, image analysis and hashing methods.

Chunhua Shen received the Ph.D. degree from the University of Adelaide, Adelaide, Australia, in 2006. He is currently a professor with the School of Computer Science, University of Adelaide. He was with the Computer Vision Program, National ICT Australia, Canberra Research Laboratory, Canberra, Australia. His current research interests include intersection of computer vision and statistical machine learning. Shen received the Australian Research Council Future Fellowship from 2012 to 2016.

Xiang Zhou is currently a undergraduate student in University of Electronic Science and Technology of China. His major research interests include computer vision and machine learning.

Yang Yang is currently with University of Electronic Science and Technology of China. He was a research fellow in National University of Singapore during 2012–2014. He was conferred his Ph.D. Degree (2012) from The University of Queensland, Australia. During the Ph.D. study, Yang Yang was supervised by Prof. Heng Tao Shen and Prof. Xiaofang Zhou. He obtained Master Degree (2009) and Bachelor Degree (2006) from Peking University and Jilin University, respectively.

Heng Tao Shen is a professor with School of Information Technology and Electrical Engineering, The University of Queensland, Australia and School of Computer Science and Engineering, University of Electronic Science and Technology of China. He obtained his B.Sc. (with 1st class Honours) and Ph.D. from Department of Computer Science, National University of Singapore in 2000 and 2004, respectively. He then joined the University of Queensland as a lecturer and became a professor in 2011. His research interests include multimedia/mobile/web search and big data management. He is the winner of Chris Wallace Award for outstanding Research Contribution in 2010 from CORE Australasia. He is an associate editor of IEEE TKDE, and will serve as a PC Co-Chair for ACM Multimedia 2015.

Please cite this article as: F. Shen, et al., Face image classification by pooling raw features, Pattern Recognition (2016), http://dx.doi.org/10.1016/j.patcog.2016.01.010