SRDANet: An Efficient Deep Learning Algorithm for Face Analysis

Lei Tian(✉), Chunxiao Fan, Yue Ming, and Jiakun Shi

Beijing Key Laboratory of Work Safety Intelligent Monitoring, School of Electronic Engineering, Beijing University of Posts and Telecommunications, Beijing 100876, People's Republic of China
tianlei189@bupt.edu.cn, {fcxg100,sjk2012sjk}@163.com, myname35875235@126.com

Abstract. In this work, we take advantage of the superiority of Spectral Graph Theory in classification application and propose a novel deep learning framework for face analysis which is called **Spectral Regression Discriminant Analysis Network (SRDANet)**. Our SRDANet model shares the same basic architecture of Convolutional Neural Network (CNN), which comprises three basic components: convolutional filter layer, nonlinear processing layer and feature pooling layer. While it is different from traditional deep learning network that in our convolutional layer, we extract the leading eigenvectors from patches in facial image which are used as filter kernels instead of randomly initializing kernels and update them by stochastic gradient descent (SGD). And the output of **all** cascaded convolutional filter layers is used as the input of nonlinear processing layer. In the following nonlinear processing layer, we use hashing method for nonlinear processing. In feature pooling layer, the block-based histograms are employed to pooling output features instead of max-pooling technique. At last, the output of feature pooling layer is considered as one final feature output of our model. Different from the previous single-task research for face analysis, our proposed approach demonstrates an excellent performance in face recognition and expression recognition with 2D/3D facial images simultaneously. Extensive experiments conducted on many different face analysis databases demonstrate the efficiency of our proposed SRDANet model. Databases such as Extended Yale B, PIE, ORL are used for 2D face recognition, FRGC v2 is used for 3D face recognition and BU-3DFE is used for 3D expression recognition.

Keywords: SRDA Network · Deep learning · Spectral Regression Discriminant Analysis · Face recognition · Expression recognition

1 Introduction

During the last decade, the single face analysis task has been extensively studied, such as single face recognition and single expression recognition. Numerous efforts have been made to design the hand-crafted features for 2D face analysis [1–5] and 3D face analysis [6–9]. However, most of the hand-crafted features

H. Liu et al. (Eds.): ICIRA 2015, Part I, LNAI 9244, pp. 499–510, 2015.
DOI: 10.1007/978-3-319-22879-2_46

are not competent for multiple face analysis tasks simultaneously, since their successes mainly depend on the successes of separate domains knowledge of face/expression recognition. Besides, the traditional single face analysis methods are not robustness to extreme intra-class variability. So, with the development of synergetic analysis for multiple visual tasks in the context of big data, the newly proposed algorithms need to provide an excellent performance for face recognition and expression recognition simultaneously.

In order to solve the problems mentioned above, learning higher-level features from the data is considered as a plausible way [10–12]. Researchers hope to discover a multi-level representation through deep neural networks, mainly because higher-level features can represent more abstract semantics of the data. There are two examples of such method: Convolutional Neural Network (CNN) [13–17] and Auto-Encoders (AEs) [18,19]. However, their successes depend not only on parameter tuning but also on depth of their architectures. In another word, when their architectures are not deep enough, the performances won't be as good as the hand-crafted features.

To overcome the problems mentioned above, we translate the idea of the Spectral Regression Discriminant Analysis (SRDA) into deep learning framework. The Spectral Regression Discriminant Analysis (SRDA) [20] is derived from Linear Discriminant Analysis (LDA) but solve the problems of LDA such as **singularity issue** and **small sample issue**. In previous work, SRDA, a supervised version of PCA, is used to extract the leading eigenvectors of input data. In this paper, SRDA is used to extract the principle component of the patches in face image which are used as filter kernels of the deep network. Our proposed SRDANet model has fewer deep network layers than traditional deep learning framework, without tuning parameter. **In the convolutional layer**, we extract the leading eigenvectors from patches in face image by SRDA which are used as filter kernels instead of randomly initializing kernels and update them by Stochastic Gradient Descent (SGD). **In the nonlinear processing layer**, there is no nonlinear processing layer until the end of **all** convolutional filter layer, and the binary hashing method is applied to nonlinearly process feature maps, instead of the *Sigmoid* or *ReLU* function [21], which is different from traditional deep learning network. It means that feature maps are extracted from the all cascaded convolutional layers, and they are used as the input of nonlinear processing layer. **In the feature pooling layer**, the block-based histograms method is applied to pooling feature. Since the output of nonlinear processing layer is a hashed decimal-valued image, conventional pooling methods such as Max-Pooling and Average-Pooling are not fit for our model. **In the output layer**, we vectorize the output of feature pooling layer as the final feature of our model. Figure 1 illustrates how our SRDANet model extracts multiple features from face images.

Organization of this Paper. In Section 2, we give a detailed illustration of SRDANet algorithm. In Section 3, we evaluate the computational complexity of our SRDANet. The extensive experimental results are presented in Section 4. Finally, we conclude our work in Section 5.

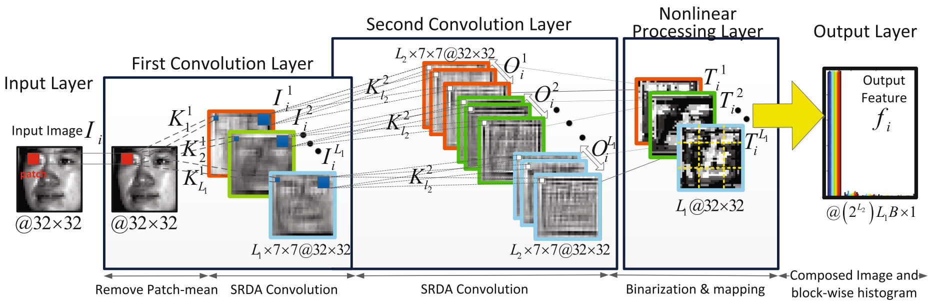

Fig. 1. The detailed layer architecture diagram of the two-stage SRDANet

2 SRDA Network Deep Learning Algorithm

SRDA Network takes both advantage of Spectral Graph Theory and the architecture of Convolutional Neural Network (CNN). Suppose that we are given a set of N samples $\{\boldsymbol{I_i}\}_{i=1}^{N}$ which belongs to c classes, and its size is $d_1 \times d_2$ $(d_1 \times d_2 = d)$. The patch size is $j_1 \times j_2$ at all stages. The number of filters in ith layer is L_i. In order to illustrate precisely, we set the number of stages to be 2.

2.1 The Convolutional Filter Layer

In order to take advantage of convolution, we reserve the basic convolutional processing of CNN. As for learning filter kernels, we just replace SGD with SRDA. For each input image, we take a $j_1 \times j_2$ patch around each pixel and collect all overlapping patches in the ith image I_i. Then, we vectorize all patches and subtract patch mean from each patch, then we obtain $\bar{\boldsymbol{P}}_i = \left[\bar{\boldsymbol{p}}_{i,1}, \bar{\boldsymbol{p}}_{i,2}, \cdots, \bar{\boldsymbol{p}}_{i,d}\right]$ for the ith image. We construct the same matrix for all input images and we have

$$\boldsymbol{P} = \left[\bar{\boldsymbol{P}}_1, \bar{\boldsymbol{P}}_2, \cdots, \bar{\boldsymbol{P}}_N\right] \in \mathbb{R}^{j_1 j_2 \times Nd} \tag{1}$$

Then, we learn the transformation vectors $\boldsymbol{V^i}$ from data matrix $\boldsymbol{P}$. The SRDA is aimed to search for the project axis on which the data points of different classes are separate from each other while the data points of same classes are close to each other. Its objective function is that:

$$V^* = \arg\max \frac{V^T S_b V}{V^T S_t V} \tag{2}$$

where we call S_b the between-class scatter matrix and S_t the total scatter matrix. In order to clearly denote, $SR_{l_1}(\boldsymbol{a})$ means data $\boldsymbol{a}$ through SRDA algorithm and get the $l_1 th$ leading eigenvectors. Then we map each eigenvector to matrix of size $j_1 \times j_2$.

$$\boldsymbol{K}_{l_1}^1 = vec2mat_{j_1,j_2}\left(SR_{l_1}\left(\boldsymbol{P}\right)\right), l_1 = 1, 2, \cdots, c-1 \tag{3}$$

In order to extract meaningful and robust feature from input image, we convolute SRDA filter kernels with input image $\boldsymbol{I_i}$ and obtain the $l_1 th$ filter output of the first stage:

$$\boldsymbol{I}_i^{l_1} = \boldsymbol{I}_i * \boldsymbol{K}_{l_1}^1, i = 1, 2 \cdots, N \tag{4}$$

where * denotes 2D convolution. In order to make $\boldsymbol{I}_i^{l_1}$ to have the same size of $\boldsymbol{I}_i$, we pad the boundary of $\boldsymbol{I}_i$ with zero before convolution operation.

Similar to LDA, the output number of non-zero eigenvectors of SRDA is $(c-1)$, where c is the number of classes. So we get $(c-1)$ output $\boldsymbol{I}_i^{l_1}$ in the first stage. Because the output of the first convolutional layer $\boldsymbol{I}_i^{l_1}$ is used as the input of second convolutional layer, it is infeasible for SRDANet algorithm to run in a limited memory when c is large. So further dimensionality reduction is essential. We apply PCA algorithm to reduce dimension from $(c-1)$ to L_1. Therefore we replace $\boldsymbol{K}_{l_1}^1$ in Eq. (3) with the following equation when c is too large to load into memory:

$$\boldsymbol{K}_{l_1}^1 = vec2mat_{j_1,j_2}\left(PCA_{l_1}\left(SR\left(\boldsymbol{P}\right)\right)\right), l_1 = 1, 2, \cdots, L_1 \tag{5}$$

where $PCA_{l_1}\left(\boldsymbol{X}\right)$ denotes the $l_1 th$ principal eigenvector of $\boldsymbol{X}\boldsymbol{X}^{\boldsymbol{T}}$.

Different from the architecture of conventional CNN, we only insert nonlinear processing layer when all convolutional layers have been processed. Since the operation that we insert nonlinear processing layer (such as *Sigmoid* or *ReLU* function) after each convolutional layer shows no chance of improving the performance of our SRDANet model through extensive experiments. So we simplified our network's architecture by considering the efficiency of training. The output of the first convolutional layer $\boldsymbol{I}_i^{l_1}$ is used as the input of second convolutional layer instead of nonlinear processing layer. Almost repeating the same process as the first convolutional layer, we collect all patches of $\boldsymbol{I}_i^{l_1}$ and obtain $\boldsymbol{Q} = \left[\boldsymbol{\bar{Q}_1}, \boldsymbol{\bar{Q}_1}, \cdots, \boldsymbol{\bar{Q}_N}\right]$. Then we compute the leading eigenvectors from data matrix $\boldsymbol{Q}$ by SRDA and perform dimensionality reduction by PCA. Similar to the first stage, we obtain the filter kernels of the second layer $\boldsymbol{K}_{l_2}^2$. For each input $\boldsymbol{I}_i^{l_1}$ of the second convolutional layer, we obtain L_2 outputs $\boldsymbol{O}_i^{l_1}$:

$$\boldsymbol{O}_i^{l_1} = \left\{\boldsymbol{I}_i^{l_1} * \boldsymbol{K}_{l_2}^2\right\}_{l_2=1}^{L_2}, l_2 = 1, 2, \cdots, L_2 \tag{6}$$

where $l_1 = 1, \cdots, L_1$ and $i = 1, 2, \cdots, N$.

2.2 The Nonlinear Processing Layer

We choose $\boldsymbol{O}_i^{l_1}$ as the input of next nonlinear processing layer. It is obvious that the method of learning filter kernels in our model is different from conventional CNN, so the traditional nonlinear processing functions aren't suitable for our SRDANet model. For each $\boldsymbol{I}_i^{l_1}$, there are L_2 outputs $\boldsymbol{O}_i^{l_1}$ in the second convolutional layer. The Heaviside step function[1] is employed to binarize these outputs,

[1] $H\left(x\right) = \begin{cases} 1, x > 0 \\ 0, x \leq 0 \end{cases}$

and we obtain binary images

$$\boldsymbol{H}_i^{l_1} = \left\{ \max \left(\boldsymbol{I}_i^{l_1} * \boldsymbol{W}_{l_2}^2, 0 \right) \right\}_{l_2=1}^{L_2}. \tag{7}$$

We consider the L_2 binary bits as a decimal number and convert the L_2 outputs in $\boldsymbol{O}_i^{l_1}$ into a decimal image:

$$T_i^{l_1} = \sum_{l_2=1}^{L_2} 2^{l_2-1} H_i^{l_1}, \tag{8}$$

whose pixel is a decimal number in the range $\left[0, 2^{L_2} - 1\right]$.

2.3 The Feature Pooling Layer

As far as we know, the max-pooling and average-pooling methods are not suitable to process the output of our nonlinear processing layer. We partition the image $\boldsymbol{T}_i^{l_1}$ into B blocks and compute the histogram of the decimal values in each block.

We further concatenate all the B histograms into one vector as $Bhist\left(\boldsymbol{T}_i^{l_1}\right)$. Finally, the feature of input image I_i is defined as a set of block-wise histograms vector:

$$\boldsymbol{f_i} = \left[Bhist\left(\boldsymbol{T}_i^1\right), \cdots, Bhist\left(\boldsymbol{T}_i^{L_1}\right) \right]^T \in \mathbb{R}^{\left(2^{L_2}\right) L_1 B} \tag{9}$$

3 The Computational Complexity of the SRDANet Algorithm

In this section, we provide evidence to show how light the computational complexity of our SRDANet algorithm is. We take the two-stage SRDANet as an example for analysis. Forming the patch-mean-removed matrix $\boldsymbol{P}$ costs $(j_1 j_2 + j_1 j_2 d_1 d_2)$ *flops*[2] in each stage of SRDANet; And the computational costs of learning the SRDA filter kernels from whole training sets is $[\frac{1}{2} N d_1 d_2 (j_1 j_2)^2 + O\left(N d_1 d_2 j_1 j_2\right)]$; The convolution of SRDA filters take $L_i d_1 d_2 j_1 j_2$ *flops* for stage i; The conversion of L_2 binary bits to a decimal number costs $2 L_2 d_1 d_2$ *flops* in the output stage; The operation of concatenated block-wise histogram is of complexity $O\left(d_1 d_2 B L_2 \log 2\right)$.

Assuming $d \gg \max\left(j_1, j_2, L_1, L_2, B\right)$, the overall complexity of SRDANet is simplify as

$$O\left(d j_1 j_2 \left(L_1 + L_2\right) + d (j_1 j_2)^2\right) \tag{10}$$

[2] *flops*, An acronym for **FL**oating-point **O**perations **P**er **S**econd.

4 Experimental Results

In this section, we evaluate the performance of our proposed SRDANet in different face analysis tasks, such as *face recognition* and *expression recognition* in 2D or 3D facial images. Figure 2 shows the cropped face images from ORL, PIE, FRGC v2 and BU-3DFE database.

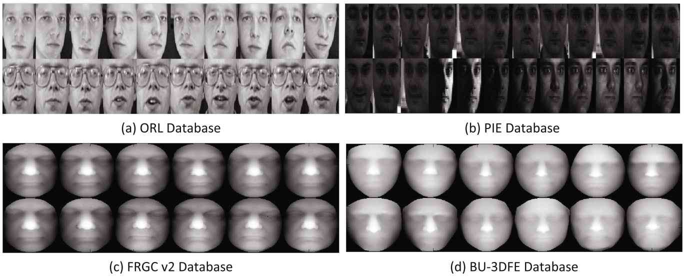

Fig. 2. Sample face images from face analysis database. (a) The ORL Database. (b) The PIE Database. (c) The FRGC v2 Database. (d) The BU-3DFE database.

4.1 Impact of Number of Filter Kernels

Before comparing SRDANet with existing LDANet [22] and other subspace methods, we first investigate the impact of the number of filter kernels of one-stage SRDANet and two-stage SRDANet. We use Yale database and down-sample the image to 32×32 pixels. We random choose 5 images per individual as training set and average the results over 10 random splits. For one-stage SRDANet, we vary the number of filter kernels in the first stage L_1 from 4 to 14. For two-stage SRDANet, We set the number of filter kernels in the second stage $L_2 = 8$ and vary L_1 from 4 to 14. The size of filter kernel is $j_1 = j_2 = 7$ and their non-overlapping block size is 7×7. The results are shown in Figure 3.

One can see that the accuracy of one-stage SRDANet and two-stage SRDANet increases for larger L_1. What's more, both one-stage SRDANet and two-stage SRDANet achieve best performance when L_1 approximately in the range $[8, 10]$.

4.2 2D Face Recognition on ORL Datasets

The ORL face database is used in this test. It consists of a total of 400 face images of 40 individuals (10 samples per individual). We random choose $n\,(= 2, 3, 4, 5)$ images per individual as training set. The rest of the database to form the testing set. For each given n, we average the results over 50 random splits. We compare the performance of SRDANet with LDANet [22], Discriminant Face Descriptor (DFD) [23], Sparse Representation Classifier (SRC) [24], Spectral Regression Discriminant Analysis (SRDA) [20], Spatially Smooth LDA (S-LDA) [25] and Completed LBP (C-LBP) [26]. We also use PCA as baseline algorithms.

The parameters of SRDANet are set to $L_1 = L_2 = 8$, $j_1 = j_2 = 7$, and the non-overlapping block size is 7×7. We use SVM classifier for SRDANet, LDANet, SRDA, C-LBP and use NN classifier with cosine distance for DFD and S-LDA. Different classifier is to secure the best performances of respective features.

The results are given in Table 1. One can see that our proposed algorithm again achieves the best accuracy. A prominent message drawn from the above experiments is that our proposed SRDANet can be very effective to extract the abstract representation of face images, and it is quite competitive to the current state-of-the-art face recognition algorithms.

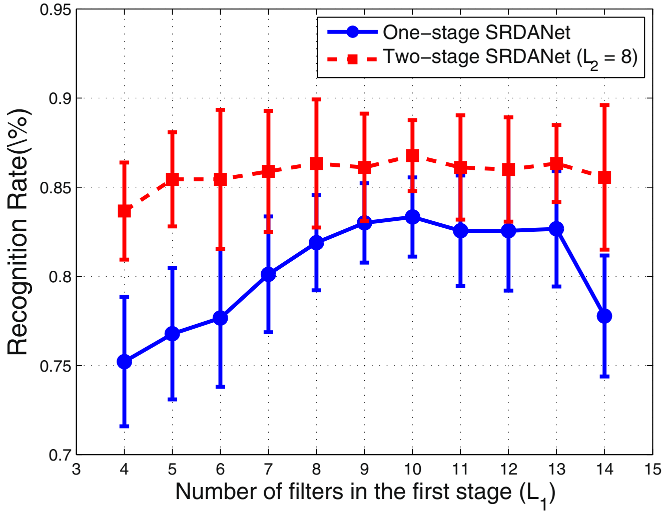

Fig. 3. Recognition accuracy of SRDANet on Yale for varying number of kernels in the first layer L_1.

4.3 2D Face Recognition on PIE Dataset

We investigate the performance of proposed SRDANet on PIE datasets for 2D face recognition. In PIE databases, original images were cropped into 32×32 pixels. The five near frontal poses is used in this test. There are around 170 images for each individual. We random choose n(=5,10,20,30) images per individual as training set. The rest to form the testing set. For each given n, we average the results over 10 random splits. The parameters of SRDANet are same as the experiments of Section 4.2. Support Vector Machine (SVM) classifier is applied.

The results are given in Table 2. One can see that SRDANet algorithms achieves the best performance for all test sets. It is also observed that the standard deviation of SRDANet less than the standard deviation of LDANet. Compared to LDA, the SRDA provides more meaningful and stable eigen-solutions. Therefore, the SRDANet can extract more discriminative and stable features, so the SRDANet outperforms the LDANet and other subspace learning algorithms for different training samples.

Table 1. Comparison of face recognition rates of various methods with SVM classifier on ORL datasets (mean±std-dev(%))

Method	2 Train	3 Train	4 Train	5 Train
PCA	70.39 (±2.9)	77.17 (±2.4)	82.47 (±2.2)	85.09 (±2.4)
S-LDA [25]	81.71 (±2.5)	88.67 (±2.5)	92.58 (±1.7)	94.98 (±1.6)
SRDA [20]	80.93 (±3.0)	88.61 (±2.3)	92.08 (±1.9)	94.34 (±1.5)
SRC+PCA [24]	78.33 (±2.5)	86.46 (±2.2)	90.84 (±1.7)	93.54 (±1.6)
C-LBP [26]	83.06 (±2.6)	89.21 (±1.7)	94.21 (±1.6)	95.25 (±1.4)
DFD ($S = 5$) [23]	75.78 (±2.1)	82.57 (±2.5)	88.71 (±1.8)	92.35 (±1.3)
LDANet [22]	83.04 (±5.2)	89.91 (±5.5)	95.20 (±2.1)	97.05 (±1.9)
SRDANet	**84.94 (±2.6)**	**92.47 (±2.5)**	**96.42 (±1.5)**	**97.68 (±1.2)**

Table 2. Comparison of face recognition rates of various methods with SVM classifier on PIE datasets (mean±std-dev(%))

Method	5 Train	10 Train	20 Train	30 Train
PCA	58.04 (±1.3)	74.60 (±1.2)	85.51 (±0.5)	89.35 (±0.3)
S-LDA [25]	74.85 (±1.1)	86.52 (±0.6)	92.60 (±0.3)	94.55 (±0.3)
SRDA [20]	75.56 (±0.7)	87.18 (±0.7)	92.61 (±0.5)	94.34 (±0.2)
SRC+PCA [24]	68.96 (±0.7)	77.24 (±0.7)	85.52*	91.98*
C-LBP [26]	70.79 (±1.1)	86.85 (±0.3)	94.65 (±0.3)	96.98 (±0.2)
LDANet [22]	76.95 (±4.4)	89.39 (±2.9)	96.89 (±0.7)	97.99 (±0.4)
SRDANet	**77.99 (±2.6)**	**91.42 (±1.0)**	**96.96 (±0.4)**	**98.29 (±0.3)**

* Note that the results are from the 1st spilt.

4.4 3D Face Recognition on FRGC v2 Dataset

We choose FRGC v2 database for evaluating the performance of our SRDANet in 3D face recognition. Face images in FRGC v2 database were cropped into 128×128 pixels. We choose the subjects whose the number of samples more than 20, and leaves us a total of 1397 face images of 60 individuals. We random choose $n\,(= 2, 4, 6)$ images per individual as training set. The rest of the database to form the testing set. For each given n, we average the results over 10 random splits. In our SRDANet, the number of filters, the filter size and the non-overlapping block size are set to $L_1 = L_2 = 8$, $j_1 = j_2 = 7$ and 7×7, respectively.

The results are given in Figure 4. One can see that the our proposed SRDANet achieves the best performance for all test sets in 3D face recognition. Since the **small samples problem** of LDA also bring into LDANet model, and SRDA solve this problem by Spectral Regression Theory. Therefore, LDANet just achieves 70.12% accuracy, but our SRDANet achieves **92.99%** accuracy when the number of training samples is 2.

We can draw a conclusion that our proposed SRDANet can be very effective to extract the discriminative feature not only from 2D face images, but also from 3D face images.

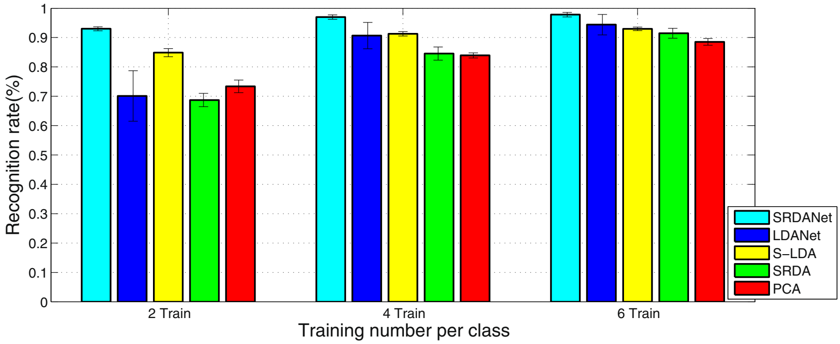

Fig. 4. Comparison of depth image face recognition rates(%) of various methods with SVM classifier on FRGC v2 datasets.

4.5 3D Expression Recognition on BU-3DFE Dataset

In this section, We evaluate the performance of SRDANet on BU-3DFE expression database for 3D expression recognition. The BU-3DFE database contains 100 subjects with 2500 facial expression, each individual has six different expressions (happiness, disgust, fear, angry, surprise and sadness) includes four levels of intensity and one neutral expression. Therefore, there are 25 (24 prototypic expressions and 1 neutral expression) expressions for each individual. We random choose $n_1\,(=50, 100, 150, 200)$ images from each prototypic expression and $n_2\,(=20, 25, 40, 50)$ images for the neutral expression as the training set. For each $(n_1 + n_2)$, we average the results over 10 random splits.

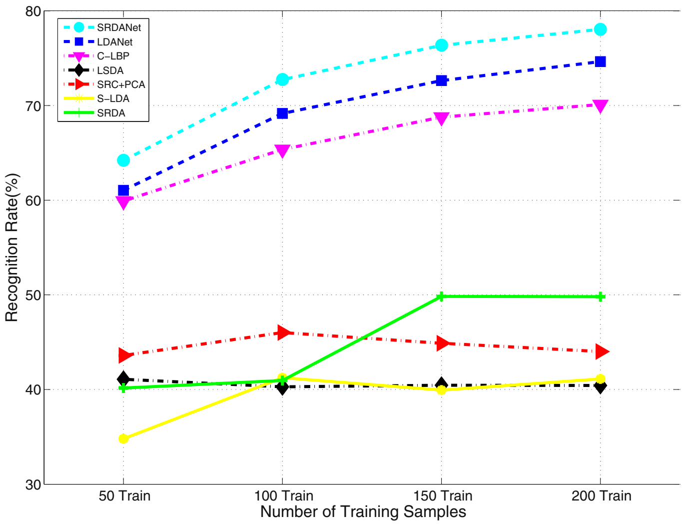

Fig. 5. Comparison of expression recognition rates(%) of various methods on BU-3DFE datasbase.

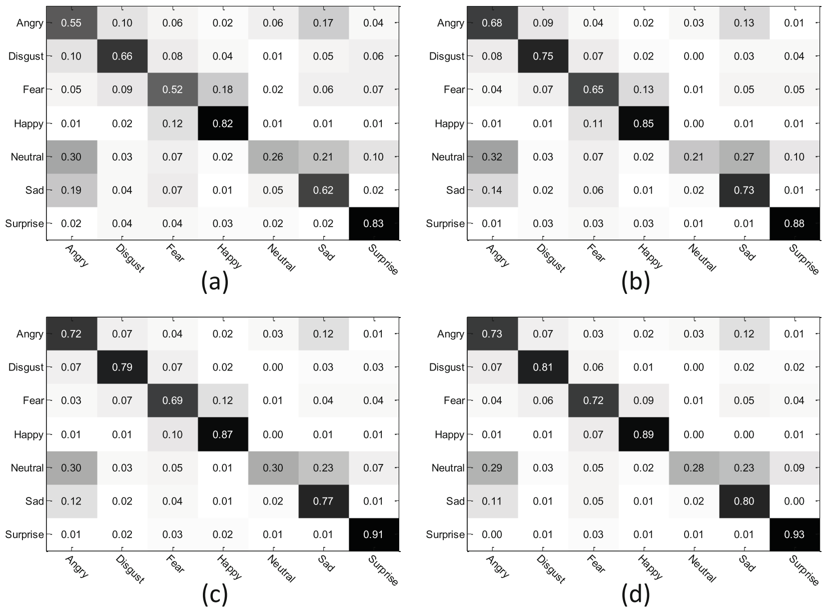

Fig. 6. Confusion matrices of different training number on the BU-3DFE dataset by our proposed SRDANet algorithm.

Figure 5 compares the our proposed SRDANet with the other existing algorithm, and Figure 6 presents the confusion matrices of different training number on the BU-3DFE dataset by our proposed SRDANet algorithm. We can get a similar conclusion as the section 4.4, the SRDANet outperforms LDANet and other algorithms, especially in the small training sample case. A prominent message drawn from the above experiment in Section 4.4 and 4.5 is that the SRDANet filter kernels learned from database itself can effectively extract the abstract representation of 3D face images. And our SRDANet demonstrates an excellent performance in face recognition and expression recognition with 3D facial images simultaneously. We can expect that the performance of our SRDANet could be further improved if the filter kernels of SRDANet are learned from a wide 3D face dataset that contains extensive inter-class and intra-class variations.

5 Conclusion

In this paper, we proposed a novel deep learning framework, called SRDA Network. Our model can be considered as a simplified version of Convolutional Neural Network. Our SRDANet model not only takes advantage of the architecture of conventional CNN, but also learns the filter kernels by Spectral Regression Discriminant Analysis in the convolutional layer. Different from other CNNs, there is no nonlinear processing layer until images are processed through all convolutional filter layer, which is a main characteristic of our SRDANet model. Then we perform nonlinear computation on the output of convolutional layer by hashing method and pooling the decimal-valued image using block-wise histogram. At last, the block-wise histogram is considered as the final features of input image. Extensive experiments demonstrate that our model can achieve excellent performance when faced with various face analysis tasks. So SRDANet model can effectively improve the performance and robustness of face analysis.

Acknowledgments. The work presented in this paper was supported by the National Natural Science Foundation of China (Grants No. NSFC-61402046, NSFC-61170176), Fund for Beijing University of Posts and Telecommunications (No.2013XZ10, 2013XD-04), Fund for the Doctoral Program of Higher Education of China (Grants No.20120005110002).

References

1. Tzimiropoulos, G., Zafeiriou, S., Pantic, M.: Subspace learning from image gradient orientations. IEEE Transactions on Pattern Analysis and Machine Intelligence **34**(12), 2454–2466 (2012)
2. Kang, C., Liao, S., Xiang, S., Pan, C.: Local sparse discriminant analysis for robust face recognition. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 846–853, June 2013
3. Ren, C.-X., Dai, D.-Q., Yan, H.: Coupled kernel embedding for low-resolution face image recognition. IEEE Transactions on Image Processing **21**(8), 3770–3783 (2012)
4. Ramirez Rivera, A., Castillo, R., Chae, O.: Local directional number pattern for face analysis: Face and expression recognition. IEEE Transactions on Image Processing **22**(5), 1740–1752 (2013)
5. Juefei-Xu, F., Savvides, M.: Subspace-based discrete transform encoded local binary patterns representations for robust periocular matching on nist's face recognition grand challenge. IEEE Transactions on Image Processing **23**(8), 3490–3505 (2014)
6. Ming, Y.: Robust regional bounding spherical descriptor for 3d face recognition and emotion analysis. Image and Vision Computing **35**, 14–22 (2015)
7. Chu, B., Romdhani, S., Chen, L.: 3d-aided face recognition robust to expression and pose variations. In: 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1907–1914, June 2014
8. Ming, Y.: Rigid-area orthogonal spectral regression for efficient 3d face recognition. Neurocomputing **129**, 445–457 (2014)
9. Ming, Y., Ruan, Q.: Robust sparse bounding sphere for 3d face recognition. Image and Vision Computing **30**(8), 524–534 (2012). Special Section: Opinion Papers
10. Liong, V.E., Lu, J., Wang, G.: Face recognition using deep pca. In: 2013 9th International Conference on Information, Communications and Signal Processing (ICICS), pp. 1–5, December 2013
11. Lu, C., Zhao, D., Tang, X.: Face recognition using face patch networks. In: 2013 IEEE International Conference on Computer Vision (ICCV), pp. 3288–3295, December 2013

12. Bengio, Y., Courville, A., Vincent, P.: Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence **35**(8), 1798–1828 (2013)
13. Kavukcuoglu, K., Sermanet, P., Boureau, Y.-L., Gregor, K., Mathieu, M., Cun, Y.L.: Learning convolutional feature hierarchies for visual recognition. In: Advances in Neural Information Processing Systems, pp. 1090–1098 (2010)
14. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: 2009 IEEE 12th International Conference on Computer Vision, pp. 2146–2153, September 2009
15. Huang, G.B., Lee, H., Learned-Miller, E.: Learning hierarchical representations for face verification with convolutional deep belief networks. In: 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2518–2525, June 2012
16. Burges, C.J.C., Platt, J.C., Jana, S.: Distortion discriminant analysis for audio fingerprinting. IEEE Transactions on Speech and Audio Processing **11**(3), 165–174 (2003)
17. Kang, L., Kumar, J., Ye, P., Li, Y., Doermann, D.: Convolutional neural networks for document image classification. In: 2014 22nd International Conference on Pattern Recognition (ICPR), pp. 3168–3172, August 2014
18. Ngiam, J., Coates, A., Lahiri, A., Prochnow, B., Le, Q.V., Ng, A.Y.: On optimization methods for deep learning. In: Proceedings of the 28th International Conference on Machine Learning (ICML 2011), pp. 265–272 (2011)
19. Rifai, S., Mesnil, G., Vincent, P., Muller, X., Bengio, Y., Dauphin, Y., Glorot, X.: Higher order contractive auto-encoder. In: Gunopulos, D., Hofmann, T., Malerba, D., Vazirgiannis, M. (eds.) ECML PKDD 2011, Part II. LNCS, vol. 6912, pp. 645–660. Springer, Heidelberg (2011)
20. Cai, D., He, X., Han, J.: Srda: An efficient algorithm for large-scale discriminant analysis. IEEE Trans. on Knowl. and Data Eng. **20**(1), 1–12 (2008)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems, pp. 1097–1105 (2012)
22. Chan, T.-H., Jia, K., Gao, S., Lu, J., Zeng, Z., Ma, Y.: Pcanet: A simple deep learning baseline for image classification? arXiv preprint arXiv:1404.3606 (2014)
23. Lei, Z., Pietikainen, M., Li, S.Z.: Learning discriminant face descriptor. IEEE Transactions on Pattern Analysis and Machine Intelligence **36**(2), 289–302 (2014)
24. Wright, J., Yang, A.Y., Ganesh, A., Sastry, S.S., Ma, Y.: Robust face recognition via sparse representation. IEEE Transactions on Pattern Analysis and Machine Intelligence **31**(2), 210–227 (2009)
25. Cai, D., He, X., Hu, Y., Han, J., Huang, T.: Learning a spatially smooth subspace for face recognition. In: Proc. IEEE Conf. Computer Vision and Pattern Recognition Machine Learning (CVPR 2007) (2007)
26. Guo, Z., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. IEEE Transactions on Image Processing **19**(6), 1657–1663 (2010)