Face Recognition Using Local PCA Filters

Yida Wang, Shasha Li, jiani Hu, and Weihong Deng

Beijing University of Posts and Telecommunications, Beijing, China
{wangyida1,lishasha,jnhu,whdeng}@bupt.edu.cn
http://www.bupt.edu.cn/

Abstract. We propose an efficient feature extraction architecture based on PCANet. Our method performs far better than many traditional artificial feature extraction methods with the help of standalone filter learning and multiscale local feature combination. Such structure cascaded by both linear layers with convolution filters and non-linear layers in binarization process shows better adaptability in different databases. With the help of parallel computing, training time is much shorter than PCANet and also more fixed compared to convolutional neural network. Experiment in LFW and FERET shows that such a data oriented structure shows good performance both on stability and accuracy in various environments.

Keywords: Feature extraction, PCANet, Filter learning, Standalone training, Multiscale, Binarization

1 Introduction

Face recognition shows more and more values in machine learning research with typical databases and is also widely used in real life applications. As alignment of face based on key points being more accurate, feature description plays a determining factor on face recognition. Different classical hand-crafted features aiming at ad hoc recognition goals behave well. Some unsupervised features like LBP and Gabor capture discriminant feature while getting rid of ill effects caused by different lightings, occlusion, corruption and solving problems related to rotation and displacemen because unlike holistic methed such as Eigenface and Fisherface. They are local feature based descriptors which can eliminate some intra person interference. But such methods are still powerless towards some difficulties like flexible deformation impacts. Within-class variance components could make a huge misleading to the classifier result. Shallow layer based methods like Gabor, LBP, SIFT and Shallow Neural Networks share some common ground which all extracting feature with just one hidden layer. By increasing the number of layers, DNN is advanced in liberating multifarious works of structure matching process regarding to distinct recognition condition with the help of back propagation algorithm. Cascaded layers in CNN [2] make parameters more suitable for sophisticated condition compared to shallow networks. PCANet [1] which consists of concatenated layers stacked one by one behave well in field of face recognition. Its filter learning process is driven by local image patches for just once and brings advantages in speeding up in contrast with convolution kernels learning in CNN by using eigenvalue decomposition on the output of previous layer. The convolution output of the bottom layer directly passed to the next convolution layer. Such simplified structure reduces the computation and abates the uncertainty of time for convergence. PCANet also have greater ability in task immigration with fixed structure along with higher accuracy in comparison with LBP and Gabor. Novel improved descriptors derived from this architecture has been proposed recently, such as in [3]. We further modifies this architecture to achieve higher recognition accuracy and adaptability in different missions. Our improved approach which is also based on such structure shows better performance than the original one.

Our structure fully exploits the information by constructing filter banks directly from data which are complementary with each other. It could satisfy the expectation without optimization through iterations. There are two main modifications about filters learning based on PCANet. Filter banks solved by eigenvalue decomposition is driven by the previous layer's output separately. Outputs from different convolutional kernels differ much in texture because of their orthogonalization relations, we train every few filters from exact one part of the output of former layer. We could fully exploit the discriminative information in previous layers and enhance the robustness for feature extraction. We select a group of continuous size for filter training and feature extraction. Training result shows that smaller filters isn't just an approximation to any sub region in bigger filters. So training PCA filters in such manner is guaranteed on a more adequate feature representation, discriminant statistic pattern extracted from different filters might lie in various position of the same structure of network. Some experiments prefer to mix different features together by adding up similarity scores calculated by normalized feature rather than concatenate all features together for purpose of saving memory. Separately trained structure with form of cascaded network and mutiscale feature combination extracts richer information which is robust and discriminative. Experiment on Feret and LFW data base shows that there are about 2% recognition accuracy improvement in difficult classification environments while the feature dimension keeps unchanged. There are even better improvement in accuracy which achieves up to 2.6% in further experiments with key point alignment and affine transformation applied on the image.

2 Method

2.1 Filter Learning

the 1st Stage Training Like the 1st stage training in PCANet, given N training samples $\{I_i\}_{i=1}^{N}$ with the same size of $m \times n$. The patch size doesn't change in a certain cascading queue, so multiscale filter banks is trained separately.

Overlapping patches in a particular size of images are collected for training a filter bank using SVD(Singular Value Decomposition). Patches are combined of adjacent pixels. As for the pixels on the edge of a image, we use zero padding to make it still useful. Patch mean is subtracted from each patch before SVD. Then the whole resource for filter training is represented as $X = [\bar{X}_1, \bar{X}_2, ..., \bar{X}_N] \in R^{k_1 k_2 * Nmn}$ where the image size of all N samples is $m \times n$ and the patch size is $k_1 \times k_2$. A single patch feature matrix $\bar{X}_1$ extracted from an image is formed by a set of vectors as $[\bar{x}_{i,1}, \bar{x}_{i,2}, ..., \bar{x}_{i,mn}]$ where $\bar{x}_{i,j}$ denotes the jth vectorized patches in the ith image.

The meaning of PCA is projecting the original data to another orthogonal space which uses as less basis as possible maximizing the variance of data, so basis in such orthogonal space is selected follows constraint: $\max_{V \in R^{k_1 k_2 \times S_1}} ||V^T X||_F^2$ while $V^T V = I_{S_1}$ it is solved by eigenvalue decomposition of XX^T, so the convolution kernel could be expressed as

$$W_l^1 = mat_{k_1,k_2}(q_l(XX^T)) \in R^{k_1 \times k_2}, l = 1, 2, ..., S_1 \tag{1}$$

where S_1 means the number of the set of principle eigenvectors in the 1st layer, $mat_{k_1,k_2}(v)$ is a reshaping function aims to transform v to the size of $k_1 \times k_2$ and function $q_l(XX^T)$ denotes the lth principal eigenvector of XX^T or the lth left singular vector of X.

Such goal is equivalent to minimize the reconstruction error with a set of eigenvectors as shown below where I_{S_1} is identity matrix with the size of $L_1 \times L_2$:

$$\min_{V \in R^{k_1 k_2 \times S_1}} ||X - VV^T X||_F^2, s.t : V^T V = I_{S_1} \tag{2}$$

Concatenated Filter Learning We optimize the concatenated structure to extract efficient feature by taking full advantage of textures in images and rearranging them properly. For the aim of enriching discriminative features and exploiting benefits from the detachment of the back propagation process, we train the cascaded layers only using the output of the particular convolution kernel. Feature extraction is also formed as a tree-like structure. No input-output pairs uses the same filter banks between two layers. As shown in Figure 2, this feature extraction process is a 'tree' like concatenated structure rather than a 'chain' like one of traditional PCANet. The number of filters in higher stages will be no smaller than previous stage, number of filters will only keep as the same when the remaining dimension of SVD is fixed as always 1 except for the $1st$ stage.

Assuming that layer t contains S_t filter sets where each set contains l_t filters which could be represented as $l_t = \prod\limits_{L=1}^{t} S_L / S_t$ and such tree-like concatenate structure has $l_{total} = \prod\limits_{L=1}^{n} S_L$ outputs for one sample in all where n is the number of layers used. In such a feature enriched network, benefit comes with the cost for the increasing of convolution kernels, which is $F = \sum_{t=1}^{n} S_t l_t$ in total; Filters in stage t is trained separately one by one from a single group of outputs of previous layer where the lth filter output of the $(t-1)$th stage is $I_i^l = I_i * W_l^{t-1}$ while $i = 1, 2, ..., N$. One set of filters in such standalone training process only uses reconstructed samples from one filter in previous stage as source data, so different filters in previous layer would produce distinguishing learning resource where $*$ denotes 2D convolution and N is always equal to the number of input images due to the standalone training process. Overlapping patches are collected as the same manner as the 1st layer. Patch means are removed from each patch as $\bar{Y}_i = [\bar{y}_{i,l,1}, \bar{y}_{i,l,2}, ..., \bar{y}_{i,l,mn}]$ where $\bar{y}_{i,l,j}$ is the jth mean-subtracted patch in I_i^l. A single filter bank is then obtained from eigenvalue decomposition of $Y^l = [\bar{Y}_1^l, \bar{Y}_2^l, ..., \bar{Y}_N^l] \in R^{k_1 k_2 \times Nmn}$. To make images in different layers having the same size, zeros padding is applied before 2D convolution. Filter is solved as: $W_{l_t}^l = mat_{k_1,k_2}(ql(YY^T)) \in R^{k_1 \times k_2}$ while $l_t = 1, 2, ..., S_l$. As filters is learned standalone, which means that the training data in previous stage is much fewer than PCANet. This means that data would be much fewer the original one for a single branch of filter learning, so less time will be cost on convolution with filter of previous stage. With the help of parallel computing, separate branches of current stage will be computed at the same time. Much training time will be saved. Feature extracted by multiscale filters is beneficial to recognition. Here we just choose continuous odd numbers for filter scales k_1 and k_2. Convolution kernels are squares matrix for most of our experiment which means that $k_1 = k_2$.

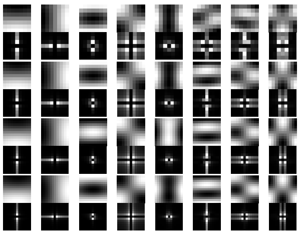

Fig. 1: Multi scale Filters(odd-numbered rows) learned in the 1st layer and modulus of their FFT represented by 10 based log function(even-numbered rows).

2.2 Feature Coding

Number of output images in the last layer equals to the amount of convolution filters, we represent discriminant features by regrouping and combining sets of outputs. All pixels in output images are converted to binaries with unit step function $S(.)$ whose output is 1 for positive input and 0 otherwise. A single threshold makes it possible for convolution results combining with each other properly and forming a more robust feature. Decimal number representing a single pixel is formed by concatenated binaries in the same position which are converted from convolutional output corresponding to a particular convolution kernel in the penultimate layer. Such reconstructed integer-valued (in range of $[0, 1, ..., 2^{S_n - 1}]$) image could be expressed as $\mathscr{O}^l = \sum_{s=1}^{S_n} 2^{s-1} S(\mathscr{I}^l * W_s^n)$ where s is the id of sets, $\mathscr{I}^l$ and $\mathscr{O}^l$ are a pair of input and output image in the last layer corresponding to the lth filter in the penultimate layer.

Every output image for a single scale of filter bank are partitioned into B blocks for precisely statistics in histogram for images in normal view. Block histograms each with the same length of 2^{S_n} in an image are concatenated in a vector afterwards. In the next step, all vectors deriving from the same input image at the beginning are concatenated together in the same sequence to form the feature f based on the filter scale K_i which are all set as odd numbers.

$$f_k = [[Hist(\mathscr{I}_1^1), ..., Hist(\mathscr{I}_B^1)], ..., [Hist(\mathscr{I}_1^{l_{n-1}}), ..., Hist(\mathscr{I}_B^{l_{n-1}})]] \quad (3)$$

All features extracted from filters in several proper scales are concatenated to represent the feature of a sample $\mathscr{F} = [f_{K_1}, f_{K_2}, ..., f_{K_n}$. Based on the fact that the likelihood of concatenated features equals to the form represented as cosine distance computed by normalized feature vectors: $\sum_{k=K_1}^{K_n} < Norm(f_k^i), Norm(f_k^j) >$ equals to $< \mathscr{F}^i, \mathscr{F}^j >$ where $<,>$ denotes the matrix inner product, $\mathscr{F} = [Norm(f_{K_1}), Norm(f_{K_2}), ..., Norm(f_{K_n})]$, we carried out our experiment on such manner instead of concatenating all discriminant sub-feature together for the sake of saving memory in some databases. Such process could also be carried out after feature projection using classifiers as a mean of similarity fusion.

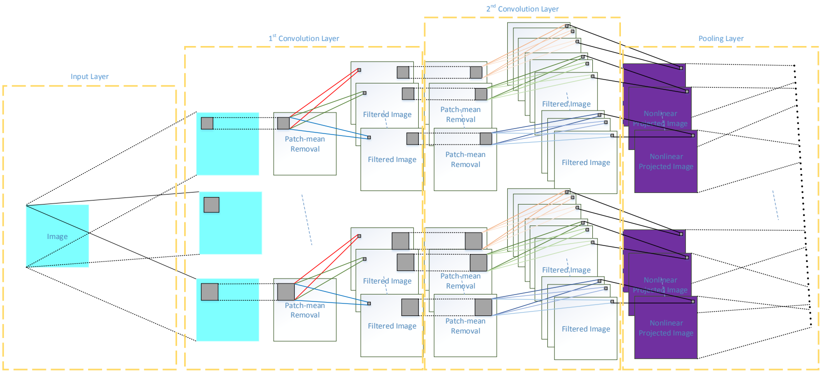

Fig. 2: Architecture of Enriched Feature Network

3 Experiment

3.1 Experiment on Feret Database

FERET tests employed frontal images gathered between 1993 and 1996. All tests are based on a single gallery containing 1196 images for training. test data with name of fb, fc and dup1, dup2 represents that expression, illumination and aging effect is the main changes from gallery data separately. Experiments of original PCANet [1] shows that Feret database has weaker environmental disturbance compared to LFW database so accuracy on Feret is better and filters trained on

FERET database show greater advantage to randomly initialized filters. LDA used in LFW database is not applied because samples of each individual aren't enough for intraclass variance rebuilding. Inner products between features in four separate test sets and pseudo inverse matrix of gallery feature are regarded as the similarity for classification. Testes are carried on structure with two convolution layers which each contains 8 filters which suit for the 8 threads of CPU for parallel computing. Output images in the last layer are partitioned into 10*10 blocks for histogram statistic. Multiscale similarities calculated from scale 7 and 9 are summarized together for experiment aiming at improving performance on accuracy and stability. Performance of such a two stage network and its variances are also compared with other local pattern feature extracting algorithms such as DFD [4]using the same classifier and others such as DT-LBP [5] copied from the original paper.

From experiments of parameter adjustment, we found that 2 stage is enough in such structure, the number of filters in the first stage should be the multiples of 2 for parallel computing.

Table 1: Accuracy on FERET, Multiscale means that we are using joint similarity for classification

Structure	fb	fc	dup-1	dup-2	Time for Training
DT-LBP [5]	99	100	84	80	*
DFD [4]	99.25	94.33	79.36	67.95	*
G-LQP [6]	99.90	100	93.20	91.00	*
sPOEM+POD [7]	99.70	100	94.90	94.00	*
GOM [8]	99.90	100	95.70	93.10	*
PCANet [1]	99.50	100	94.18	93.59	240s
Whitened PCA	99.58	100	94.88	94.02	253s
Standalone+PCA	99.58	100	94.74	94.44	73s
Standalone+Multiscale+PCA	99.75	100	95.57	95.30	123s

3.2 Experiment on LFW Database

LFW [9] data set contains 13233 images of faces collected from the web detected by the Viola-Jones face detector. 1680 of all 5749 individuals have two or more distinct photos in the data set which make it possible training LDA classifier used for projecting the high dimensional feature on low-dim spaces. Most of the experiments were carried on this database because all photos are not captured in restricted condition. All extracted features are projected with supervised matrix learning after dimension reduction using PCA. We mainly tested our method on LFW-a database and another database processed with affine transformation by using three manually annotated key points of the two apple of eyes and the center of corners of mouth for precisely histogram statistic.

Training uses photos not included in 10 test folds. LDA matrix is learned from labeled data in the rest 9 folds. Similarity between two feature of samples is defined as cosine distance between two low-dim features got by projection with LDA matrix after firstly projected by a PCA to form a low rank matrix.

Our experiments are mainly carried on a 2 stages network which each contains 8 sets of filters separately for convenience of parallel computing. All images are cropped to 150*80 and the histogram block is set to 25*20, resulting in a dimension of 81920 in most experiments of Table 1 for convenience of comparison. The dimension of PCA matrix for rank-reducing for LDA is 700 and the dimension of LDA matrix is selected between 40 and 300 in trial. We select 7, 9, 11, 13 as side length of filters in multiscale filter experiment. The similarity score is the summation of ones computed in each scale. The accuracy is defined as the average of verification rates and true negative rate $Accuracy = (TP + TN)/2$

Table 2: Accuracy on LFW Database; Standalone represent a standalone training precess; Multiscale represent concatenating multiscale features

Structure	LFW(aligned)	LFW(affine transformed)	Time for Training
MRF-MLBP [10]	79.08	*	*
SFRD [11]	84.81	*	*
I-LQP [12]	86.20	*	*
OCLBP [13]	86.66	*	*
Fisher vector faces [14]	87.47	*	*
Eigen-PEP [15]	88.97	*	*
PCANet [1]	88.95	89.58	639s
Standalone+PCA	89.05	90.15	185s
Standalone+Whitened PCA	89.08	90.18	192s
Standalone+Multiscale+PCA	91	92.21	696s

LFW database are aligned with a commercial face alignment software and is applied with affine transformation based on 3 key points annotated by hand. Results show that the enriched feature did improves the performance. Experiment results is the average performance of each method and * denotes that there are no comparable result which could be found. As shown in the table, experiment on our structure and PCANet both using Matlab for training on a 4 core Intel i7 4770 CPU, with the help of parallel computing, the training time of standalone training is just one-third of PCANet with the same filter numbers. Though the using of multiscale feature mixtrue, its only takes a bit more time than PCANet and get a more perfect prediction result. The standalone training saves much time for training, we also observed that using filters learned from FERET database only reduce around 2 percent of accuracy.

4 Conclusion

Experiments show that such an unsupervised feature extraction architecture may do well in different classification issues. The concatenated network achieves better performance with the help of reconstruction in convolution kernels and joint similarity. Visualization of trained filters and their 2D FFT indicates that filters learned from image patches extract low frequency textures which are complementary with each others. This solver has certain ability to do the same job carried by back propagation process in traditional convolutional neural network.

Acknowledgements The authors would like to thank the anonymous reviewers for their thoughtful and constructive remarks that are helpful to improve the quality of this paper. This work was partially sponsored by supported by the Fundamental Research Funds for the Central Universities under Grant No. 2014ZD03-01, NSFC (National Natural Science Foundation of China) under Grant No. 61375031, No. 61471048, and No.61273217. This work was also supported by the Beijing Higher Education Young Elite Teacher Program, and the Program for New Century Excellent Talents in University.

References

1. Chan, T., 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)
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet Classification with Deep Convolutional Neural Networks. Advances in Neural Information Processing Systems 25. Curran Associates, Inc. pp. 1097–1105 (2012)
3. Zeng, R., Wu, J., Shao, Z., Senhadji, L., Shu, H.: Multilinear Principal Component Analysis Network for Tensor Object Classification. Eprint Arxiv (2014)
4. Lei, Z., Pietikainen, M., Li, S.Z.: Learning Discriminant Face Descriptor. IEEE Transactions on PAMI. vol. 36(2), pp, 289–302 (2014)
5. Maturana, D., Mery, D., Soto, A.: Face recognition with decision tree-based local binary patterns. CV ACCV 2010. pp. 618–629. Springer Berlin Heidelberg (2011)
6. Hussain, S.U., Napoleon, T., Jurie, F.: Face Recognition using Local Quantized Patterns. In: BMVC, Guildford, United Kingdom (2012)
7. Vu, N.S.: Exploring Patterns of Gradient Orientations and Magnitudes for Face Recognition. Information Forensics and Security, IEEE Transactions. vol. 8(2), pp. 295–304 (2013)
8. Chai, Z., Sun, Z., Mendez-Vazquez, H., He, R., Tan, T.: Gabor Ordinal Measures for Face Recognition. Information Forensics and Security, IEEE Transactions on. vol. 9(1), pp. 14–26 (2014)
9. Huang, G.B., Ramesh, M., Berg, T., Learned-Miller, E.: Labeled Faces in the Wild: A Database for Studying Face Recognition in Unconstrained Environments. Technical Report. pp. 07–49, University of Massachusetts, Amherst (2007)
10. Arashloo, S.R., Kittler, J.: Efficient processing of mrfs for unconstrained-pose face recognition. In: BTAS, 2013 IEEE Sixth International Conference. pp. 1–8 (2013)
11. Cui, Z., Li, W., Xu, D., Shan, S., Chen, X.: Fusing Robust Face Region Descriptors via Multiple Metric Learning for Face Recognition in the Wild. In: CVPR, 2013 IEEE Conference. pp. 3554–3561 (2013)
12. Hussain, S., Triggs, B.: Visual Recognition Using Local Quantized Patterns. Computer Vision ECCV 2012. pp. 716–729. Springer Berlin Heidelberg (2012)
13. Barkan, O., Weill, J., Wolf, L., Aronowitz, H.: Fast High Dimensional Vector Multiplication Face Recognition. In: Computer Vision (ICCV), 2013 IEEE International Conference. pp. 1960–1967 (2013)
14. Simonyan, K., Parkhi, O.M., Vedaldi, A., Zisserman, A.: Fisher Vector Faces in the Wild. In: BMVC. (2013)
15. Li, H., Hua, G., Shen, X., Lin, Z., Brandt, J.: Eigen-PEP for Video Face Recognition. Computer Vision ACCV 2014. pp. 17–33. Springer International Publishing (2015)