

A New Convolution Network Based on Laplacian Eigenmap

Pengxin kang^{1,a}, Xiaohai He^{1,b}, Lingbo Qing^{1,c}, Qizhi Teng^{1,d}, Jie Su^{1,e}

¹College of Electronics and Information Engineering, Sichuan University, Chengdu 610065, China

^a2015222050147@stu.scu.edu.cn, ^bnic5602@scu.edu.cn, ^cqing_lb@scu.edu.cn, ^dqzteng@scu.edu.cn, ^e812787318@qq.com

Keywords: Convolution Network, Laplacian Eigenmap, texture classification, face recognition

Abstract. The best is to read these instructions and follow the outline of this text. Recently, convolutional deep neural network (ConvNet) has been widely used in the field of image classification. In this work, we propose a new feedback free convolution network for image classification. The proposed network could hierarchically and effectively extract the features from an image through a manually designed convolution network without relying on back-propagation. The network is designed in a cascaded fashion, where the Laplacian Eigenmap filter is used as convolution kernel to extract features in each of the cascaded stage. The final output of the network is achieved by a simply binary hashing and histogram encoding, and could be served as distinguishing features for many classification tasks. Experiments on different database, e.g. FERET datasets for face recognition, CURET for texture classification and MNIST for hand-written digits recognition, showed that the proposed method outperforms many other popular machine learning algorithms.

1 Introduction

Classification with images' visual information is faced with enormous challenging, which can be owing to within-class variability, generated by various lightings, non-rigid deformations, misalignment, occlusion and corruptions. Traditional machine learning algorithms that manually extract features usually cannot obtain good result in the field of visual classification, which was because that shallow features cannot reveal the nature of the target. Representative examples are local binary (LBP) [1] and Gabor [2] features for texture classification. Even these low-level features, such as color, edges and texture can achieve promising performance in some specific data and tasks. A new data set of different domain may result in bad performance because many hand-made features cannot be easily transferred to new condition [3]. Recently, the deep learning methods based on convolutional neural network have achieved great success in pattern recognition tasks. The convolutional deep neural network (ConvNet) [4] could automatically learn higher-level features by training a multilayer feed forward network through Back Propagation algorithm [5]. Large numbers of experiments have showed that this high-level semantic features, (eg, the last fully connected layer

of a ConvNet) outperforms many other traditional machine learning algorithms in pattern recognition tasks, [6]. However, it is time-consuming and expensive to train such a convolutional network due to the training of back-propagation, and the high-performance requirements of computers. In 2015, Tsung-Han Chan, etc, proposed a new convolutional learning network for visual classification—PCANet [7]. The main advantage of PCANet is that it could extract features without the phase of back-propagation, and hence conserves the training time. However, PCA only maps the input data to a lower dimensional sub-space and without the information of intra-class and inter-class taken into consideration. As a result, PCA can only identify gross variability and cannot distinguish ‘among-groups’ and ‘within-groups’ variability. Based on these, we propose a new convolution network based on Laplacian Eigenmap.

The remainder of the paper is written as follows. Section 2 describes the structure of our proposed method. In Section 3, we evaluate the performance of our proposed method in various datasets, including face, hand-written digits, texture discrimination dataset. Finally, Section 4 concludes.

2 Methodology

The section headings are in boldface capital and lowercase letters. Second level headings are typed as part of the succeeding paragraph (like the subsection heading of this paragraph). Different from PCA, Laplacian Eigenmaps build a graph from neighborhood information of the data set, and create a matrix of edge weights according to the distance of each point pairs. Specifically, larger weights are assigned to the dissimilar sample pairs, and small weights are given to the smaller sample pairs. In this way, Laplacian Eigenmaps could utilize both the information of intra-class and inter-class, and therefore created a more distinguishable sub-space. The proposed network can conserve the training time and show better performance than the traditional machine learning algorithms and PCANet.

Structures of the Laplacian Eigenmap Network (LENet). The same structure as PCANet is adopted in our proposed network and is presented in Figure 1. We suppose that the size of input images $\{I_i\}_{i=1}^N$ is $m \times n$ and the convolution kernel size (2D LE filter’s size) is $s_1 \times s_2$ in all layers. In the following, we describe the components of LEnet more particularly.

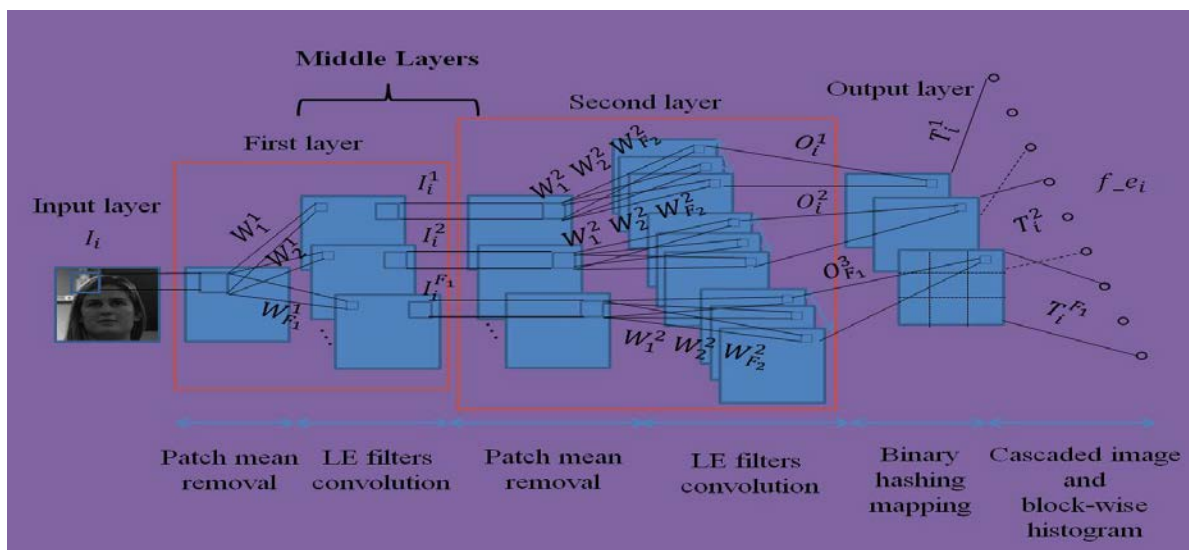


Fig. 1. The block diagram of the proposed (two-layer) LEnet

The Input layer. In our network, we collect $(m-s_1+1) \times (n-s_2+1)$ patches through a $s_1 \times s_2$ sliding window for each input image as PCANet does. We also remove the patch mean from all the vectorizational patches and generate into a matrix. And we repeat the above process for the input images and obtain a input matrix :

$$X = [\bar{X}_1, \bar{X}_2, \dots, \bar{X}_N] \in R^{s_1 s_2 \times Nmn}, \quad (1)$$

Where \bar{X}_1 denotes a patch matrix that remove the mean from each patch. Then we can get a covariance matrix:

$$C^1 = XX^T = \sum_{i=1}^N \bar{X}_i \bar{X}_i^T. \quad (2)$$

The Middle Layers. Now we focus on the convolution kernels in first convolution layer, which is generated by laplacian eigenmap, as is described in the Figure 1. Laplacian eigenmap usually need to construct an initial map by KNN algorithm [8], here we choose the covariance matrix as the initial map to avoid the high time consumption of KNN. The covariance matrix normally demonstrates the correlation between different dimensions for each input sample. As a result, the high correlation dimensions distributes closer in the feature space by transforming the covariance matrix to a low dimensional subspace, which can make our network extracting information more effectively. So we use the column vectors of C as the input of laplacian eigenmap and the objective function of Laplacian Eigenmap is:

$$\frac{1}{2} \sum_{i,i'}^N W_{i,i'} L_{i,i'}^1 \|Tc_i^1 - Tc_{i'}^1\|^2. \quad (3)$$

We need to define a matrix D and D is a diagonal matrix of which diagonal elements are the sum of all the elements in each row of matrix $W_{i,i'}$.

$$D^1 = \begin{cases} \sum_{i'=1}^N W_{i,i'} L_{i,i'}^1 & (i = i') \\ 0 & (i \neq i') \end{cases}, \quad (4)$$

and

$$W_{i,i'}^1 = \begin{cases} 1 & (c_i^1 \in N_k(c_{i'}) \text{ or } c_{i'}^1 \in N_k(c_i)) \\ 0 & (\text{others}) \end{cases}. \quad (5)$$

By taking linear transformation on Eq.(5), the foregoing optimization problem can be expressed as matrix vector :

$$\min_{T \in R^{k_1 k_2 \times L_1}} \text{tr}(TC^1 L(C^1)^T T^T), \quad s.t. TC^1 L(C^1)^T T^T = I_{L_1}, \quad (6)$$

$$L^1 = D^1 - W_{i,i'}^1. \quad (7)$$

Where L^1 is the laplacian matrix and $L_{i,i'}^1$ is the unit matrix of size $F_1 \times F_1$. We can consider the following matrix equation as obtain optimum solution:

$$L^1 \alpha = \lambda D^1 \alpha. \quad (8)$$

We compute the F_1 eigenvectors of $C^1(C^1)^T$ and sort those eigenvectors in in ascending order corresponding to eigenvalues. So the LENet filter can be denoted as:

$$W_f^1 = \text{mat}_{s_1 s_2} (q_f(L^1 \alpha = \lambda D^1 \alpha)) \in R^{s_1 \times s_2}, \quad f = 1, 2, 3 \dots F_1. \quad (9)$$

where $\text{mat}_{s_1 s_2}$ is a function that transforms $v \in R^{s_1 \times s_2}$ to $W \in R^{s_1 \times s_2}$, $q_f(L\alpha = \lambda D\alpha)$ represents the f th eigenvector of $L^1 \alpha = \lambda D^1 \alpha$ in ascending order. The smallest K ($K \leq F_1$) eigenvectors are employed to extract the features of all the mean-removing patches. After obtaining the convolution kernels in the first convolution layer, the output of the first convolution layer is considered to be:

$$I_i^f = I_f * W_f^1, \quad i = 1, 2 \dots N, \quad (10)$$

$*$ is a symbol of 2D convolution. Then, we pad the edge of I_f with zeros to make I_i^f with a same size as I_i . Next, we take I_i^f as the second convolution layer's input and repeat the first layer's process and build $\bar{Y}_i^f = [\bar{y}_{i,f,1} \dots \bar{y}_{i,f,m}] \in R^{k_1 k_2 \times mn}$ and $\bar{y}_{i,f,j}$ denotes the j th decentralized patch in I_i^f . Then we can use $Y_i^f = [\bar{Y}_1^f, \bar{Y}_2^f, \dots, \bar{Y}_N^f] \in R^{k_1 k_2 \times Nmn}$ as the matrix including all decentralized patches of the f th convolution kernel output and connect Y^f for all convolution kernel outputs as:

$$Y = [Y^1, Y^2, \dots, Y^{F_1}] \in R^{s_1 s_2 \times F_1 Nmn}. \quad (11)$$

According to the above process, we still establish $L^2 \alpha = \lambda D^2 \alpha$ and the LE filters of the second layer are denoted as:

$$W_f^2 = \text{mat}_{s_1, s_2}(q_1(L^2 \alpha = \lambda D^2 \alpha)) \in R^{s_1 \times s_2}, f = 1, 2, \dots, F_2. \quad (12)$$

Then the filters outputs can be expressed as O_i^f and the sum of outputs in the second layer is $F_1 F_2$. Only to repeat the above process can we build a deeper network.

$$O_i^f = \{I_i^f * W_f^2\}_{f=1}^{F_2}. \quad (13)$$

The Output Layer. Like PCANet, each input image also has $F_1 F_2$ real-valued outputs $\{I_i^f * W_f^2\}_{f=1}^{F_2}$ from the middle layers. After binarized and given weights, those outputs $\{I_i^f * W_f^2\}_{f=1}^{F_2}$ can generate a simple inter valued “image” T_i^f and its pixel belongs to $[0, 2^{F_2} - 1]$:

$$T_i^f = \sum_{t=1}^{F_2} 2^{t-1} H(I_i^f * W_t^2), f = 1, \dots, F_1. \quad (14)$$

Finally, we can obtain the feature vector of the input Image f_{-e_1} through dividing the T_i^f into B blocks and block-wise histograms encoding.

$$f_{-e_1} = [\text{Bhist}(T_i^1), \dots, \text{Bhist}(T_i^{F_1})]^T \in R^{(2^{F_2}) F_2 B}. \quad (15)$$

Our network’s parameters mainly include the convolution kernel size s_1, s_2 ; the number of convolution kernel in each layer F_1, F_2 ; the number of middle layers, and the block size for histograms in the output layer and BlkOverLapRatio which controls the size of our step length of taking the blocks. In our experiments, we empirically form two layers LENet.

3 Experiment

In addition, all images are changed to gray scale and LIBSVM [9] is used for classification. In this section, we demonstrate the effectiveness of our methods by presenting experiments on 4 data sets, including MNIST dataset, CURET dataset and FERET datasets. In MNIST’s experiments, the number of each layer’s filters are set to $F_1 = F_2 = 10$ and the filter size of the LENet and PCANet are set to $s_1 = s_2 = 7$, the block size as 7, the BlkOverLapRatio are set as 0.5. The LIBSVM classifier is used to classify the targets. For other dataset, the same parameters settings are also employed in other feature extracting methods except that the number of each layer’s filters are set to $F_1 = F_2 = 10$. In addition, all images are changed to gray scale and LIBSVM [9] is used for classification.

MNIST Dataset. MNIST Dataset [4] including 70000 images, each with the size of 28×28 . In this experiment, we compare LENet with the some traditional feature extraction methods (e.g., local binary pattern feature (LBP), Gabor feature, local ternary patterns feature (LTP) [10]) and PCANet. We choose randomly 50000 image for training and 12000 images for testing s. Table 1 summarizes the results.

Observing from the Table 1, LENet and PCANet achieve the better results than other methods. However, because of considering correction information from the input samples, i.e. utilizing both the information of intra-class and inter-class, LENet($F_1 = F_2 = 10$) obtain better performance than PCANet.

Table 1 Comparison of recognition accuracy (%) between different methods on MNIST

Methods	MNIST Dataset
LBP	97.25
Gabor	97.72
LTP	98.25
PCANet($F_1 = 10, F_2 = 0$)	90.06
PCANet($F_1 = F_2 = 10$)	99.29
LENet($F_1 = 10, F_2 = 0$)	90.04
LENet($F_1 = F_2 = 10$)	99.33

FERET Dataset. PIE face dataset [11] contains 41368 images of 68 people with different pose, expression, and illumination conditions. We select the whole dataset for training, the images are resized to 60×60 and fed to LENet and PCANet. FERET dataset [12] contains 14051 eight-bit grayscale images of human heads with views ranging from frontal to left and right profiles. The images are resized to 60×60 , 10051 image are randomly selected as the training set and the rest 4000 images are as the testing set. The results are shown in Table 2.

Table 2 Comparison of recognition accuracy (%) between different methods on FERET.

Methods	FERET Dataset
LBP	73.21
Gabor	72.05
LTP	72.13
PCANet($F_1 = 7, F_2 = 0$)	75.58
PCANet($F_1 = F_2 = 7$)	76.30
LENet($F_1 = 7, F_2 = 0$)	76.08
LENet($F_1 = F_2 = 7$)	76.35

As is shown in Table 2, we also found that LENet outperforms other methods, which indicates learning form Laplacian Eigenmap is more effective for feature extracting.

CURET Dataset for Texture Classification. CURET Dataset [13] contains 61 classes of image textures. Images of the same class are captured from the same material and under different pose and illumination conditions. We resize the textures images to 60×60 . Then, 9942 and 4000 images are randomly selected for training and testing. The results are shown in Table 3.

Table 3 Comparison of recognition accuracy (%) between different methods on CURET

Methods	CURET Dataset
LBP	82.70
Gabor	85.55
LTP	81.38
PCANet($F_1 = 7, F_2 = 0$),)	90.00
PCANet($F_1 = F_2 = 7$)	91.28
LENet($F_1 = 7, F_2 = 0$),)	90.00
LENet($F_1 = F_2 = 7$)	91.70

Observing from Table 3, we found that LENet and PCANet obtain better performance for texture classification than conventional methods of extracting features on CURET dataset. In addition, the performance of LENet is a litter better than PCANet.

4 Conclusions

In this paper, we propose a feed forward, and high-efficiency convolution network based on Laplacian Eigenmap without relying on back-propagation. The network is designed in a cascaded fashion, where the Laplacian Eigenmap filter is used as convolution kernel to extract features in each of the cascaded stage. The final output of the network is achieved by binary hashing mapping and computing of the blocks histograms. We design he Laplacian Eigenmap network based on samples' covariance matrix to utilize both the information of intra-class and inter-class, which makes samples of the same class distribute compactly and samples of different class distribute separately in the low dimension feature space. Experimental results on different datasets have proven that the proposed LENet outperforms many popular feature extracting methods, including low level features extracting algorithms such as LBP, LTP, and Gabor, and high level features extracting methods, PCANet.

Acknowledgements

This work is supported by the Technology Program of public wellbeing of Chengdu, China (2015-HM01-00293-SF) and the construction items of postgraduate course of Sichuan University of Chengdu, China (2016KCJS113).

References

- [1] T. Ahonen, A. Hadid, and M. Pietikainen, "Face description with local binary patterns: Application to face recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 28, no. 12, pp. 2037–2041, Dec. 2006
- [2] C. Liu and H. Wechsler, "Gabor feature based classification using the enhanced fisher linear discriminant model for face recognition," IEEE TIP, vol. 11, no. 4, pp. 467–476, 2002.
- [3] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: a review and new perspectives," IEEE TPAMI, vol. 35, no. 8, pp.1798–1828, 2013.

- [4] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.
- [5] Andrew, Alex M. "Backpropagation." *Kybernetes* 30.9/10(2001):85-104.
- [6] J. Bruna and S. Mallat, "Invariant scattering convolution networks," *IEEE TPAMI*, vol. 35, no. 8, pp. 1872–1886, 2013.
- [7] Chan, T. H., et al. "PCANet: A Simple Deep Learning Baseline for Image Classification?." *IEEE Transactions on Image Processing* 24.12(2014):5017-5032.
- [8] Zhang, Min Ling, and Z. H. Zhou. "M L-KNN : A lazy learning approach to multi-label learning." *Pattern Recognition* 40.7(2007):2038-2048.
- [9] Chang, Chih Chung, and C. J. Lin. "LIBSVM: A library for support vector machines." *Acm Transactions on Intelligent Systems & Technology* 2.3, article 27(2007):27.
- [10] X. Tan and B. Triggs, "Enhanced local texture feature sets for face recognition under difficult lighting conditions," *IEEE TIP*, vol. 19, no. 6, pp. 1635–1650, 2010.
- [11] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression Database," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 12, pp. 1615-1618, Dec. 2003.
- [12] P. J. Phillips, H. Wechsler, J. Huang, and P. J. Rauss, "The FERET database and evaluation procedure for face-recognition algorithms," *Image Vision Comput.*, vol. 16, no. 5, pp. 295–306, 1998.
- [13] M. Varma and A. Zisserman, "A statistical approach to material classification using image patch exemplars," *IEEE TPAMI*, vol. 31, no. 11, pp. 2032–2047, 2009.