

## Classification of Handwriting Number Based on PCANet Network with Data Augmentation

Tianmei Guo<sup>1,a</sup>, Jiwen Dong<sup>2,b</sup>, Lei Wang<sup>3,c</sup>

<sup>1</sup>Department of Computer Science and Technology Shandong Provincial Key Laboratory of Network based Intelligent Computing University of Jinan, Jinan, China

<sup>2</sup>Department of Computer Science and Technology Shandong Provincial Key Laboratory of Network based Intelligent Computing University of Jinan, Jinan, China

<sup>3</sup>Department of Computer Science and Technology Shandong Provincial Key Laboratory of Network based Intelligent Computing University of Jinan, Jinan, China

<sup>a</sup>email: mengkeer\_ad@163.com, <sup>b</sup>email:ise\_dongjw@ujn.edu.cn, <sup>c</sup>email:ise\_wangl@ujn.edu.cn

**Keywords:** handwriting number recognition, Deep learning, PCANet Network, data augmentation

**Abstract.** In this paper we proposed an algorithm for classification of handwriting number which use depth of the structure of the simple convolution neural network-- PCANet Network with data augmentation. PCANet Network with data augmentation consists of four parts, one is data augmentation, two is the basic PCA filters, part three is binary quantization, four is the block-wise histograms. This simple convolution neural network improve some defects of classic convolution neural network such as training time too long, need special tuning parameters and technique problems. This allows for training networks with many features, making them insensitive to variability between class and class. Lots of experimental results show that the PCANet Network with data augmentation algorithms have higher recognition rate than the classification effect with that of SVM algorithm, BP neural network and PCANet.

### Introduction

Handwriting number recognition is an important research direction in the field of optical character recognition. It is widely used in the postal code, financial statements, bank notes and many other fields. According to the characteristics of extracting characters, existing handwriting number recognition algorithms can be generally divided into two categories[1]: one kind is based on geometric structure characteristics. Geometry features usually includes round, endpoint, intersection, outline and bump, with the method of template matching to realize automatic identification of handwriting numbers[1,2].Geometric structure characteristics need to deal with lots of details because of the arbitrariness of handwritten numbers change. In general, these methods are poor robustness[1,2] .The other kind is based on the statistical characteristics of handwritten numeral recognition algorithm. Statistical features usually include point density, feature region and so on. This kind of algorithm require adequate sample, characterized by estimating the characteristics of the spatial distribution of samples of different category training corresponding classifier, and the classifier to classify the unknown model[3, 4].

During the rise of deep learning, feature extraction and classifier has been integrated to a learning framework which overcomes the traditional method of feature selection difficulties. The idea of deep learning is to discover multiple levels of representation, with the hope that high-level features represent more abstract semantics of the data. One key ingredient of deep learning in image classification is the use of convolutional architectures.

Convolutional neural network is first introduced by LeCun in [5] and improved in [6]. They developed a multi-layer artificial neural network called LeNet-5 which can classify handwriting number. Like other neural network, LeNet-5 has multiple layers and can be trained with the

backpropagation algorithm[7]. It can obtain effective representations of the original image, which makes it possible to extract features directly from raw pixels without preprocessing. However, due to the lack of large training data and computing power at that time, LeNet-5 can not perform well on complex problem. A convolutional deep neural network (ConvNet)[8][9]consists of multiple trainable stage stacked on top of each other, followed by a supervised classifier. Each stage generally comprises of “three layers”--a convolutional filter bank layer, a nonlinear processing layer, and a feature pooling layer. To learn a filter bank in each stage of ConvNet, a variety of techniques has been proposed, such as restricted Boltzmann machines (RBM)[10] and regularized auto-encoders or their variations. In general, such a network is typically learned by stochastic gradient descent (SGD) method. However, learning a network useful for classification critically depends on expertise of parameter tuning and some special tricks.

In this paper we proposed an algorithm which use depth of the structure of the simple convolution neural network-- PCANet Network with data augmentation to classify. PCANet Network with data augmentation consists of four parts, one is data augmentation, two is the basic PCA filters, part three is binary quantization, four is the block-wise histograms. This simple convolution neural network improve some defects of classic convolution neural network such as training time too long, need special tuning parameters and technique problems. This allows for training networks with many features, making them insensitive to variability between class and class.

## PCANET network with Data Augmentation

### Data Augmentation

The easiest and most common method to reduce overfitting on image data is using label-preserving transformations[11,12].Data augmentation allows for training networks with many weights, making them insensitive to variability between class and class. Data augmentation consists of generating image translations and horizontal reflections.

### Basic PCA filters

Suppose that we use  $N \{I_i\}_{i=1}^N$  training images of size  $m \times n$ , and assume that the patch size is  $p_1 \times p_2$ .

Firstly, around each pixel, we take a  $p_1 \times p_2$  patch, and collect all patches of the  $i$ th image. It is expressed as below:

$$X_{i,1}, X_{i,2}, \dots, X_{i,m} \in R^{p_1 p_2} \quad (1)$$

In the above formula,  $X_{i,j}$  denotes the  $j$ th vectorized patch in  $I_i$ .

Secondly, we subtract patch mean from each patch and obtain each patch mean vector. It is expressed as below:

$$\bar{X}_i = [\bar{X}_{i,1}, \bar{X}_{i,2}, \dots, \bar{X}_{i,m}] \quad (2)$$

In the above formula,  $\bar{X}_{i,j}$  is a mean-removed patch.

Thirdly, for all of the input images, we perform the above operations and get the matrix:

$$X = [X_1, X_2, \dots, X_N] \in R^{p_1 p_2 \times Nm} \quad (3)$$

The process of the first PCA is to minimize the reconstruction error within a family of orthonormal filters,

$$\min_{V \in p_1 p_2 \times L_1} \|X - VV^T X\|_F^2, s.t. VV^T = I_{L_1} \quad (4)$$

In the above formula,  $L_1$  is the number of the first stage of PCA filters,  $I_{L_1}$  is identity matrix of size  $L_1 \times L_1$ .The solution is known as the  $L_1$  principal eigenvectors of  $XX^T$ . The first stage of PCA filters are therefore expressed as:

$$W_I^1 = \text{mat}_{p_1 p_2} \left( q_I \left( XX^T \right) \right) \in R^{p_1 p_2}, I = 1, 2, \dots, L_1 \quad (5)$$

Where  $\text{mat}_{p_1 p_2}(v)$  is a function that maps  $v \in R^{p_1 p_2}$  to a matrix  $W \in R^{p_1 p_2}$ , and  $q_I(XX^T)$  denotes the  $l$ th principal eigenvector of  $XX^T$ . The leading principal eigenvector capture the main variation of all the mean-removed training patches.

The process of the second PCA is the same process as the first stage. We suppose the  $l$ th filter output of the first stage is:

$$I_i^l = I_i * W_I^1, i = 1, 2, \dots, N \quad (6)$$

Where  $*$  denotes 2D convolution. Like the first stage, we can collect all the overlapping patches of  $I_i^l$ , subtract patch mean from each patch, and get:

$$\bar{Y}_i^l = \left[ \bar{y}_{i,1,1}, \bar{y}_{i,1,2}, \dots, \bar{y}_{i,1,m} \right] \in R^{p_1 p_2 \times mn} \quad (7)$$

Where  $\bar{y}_{i,1,1}$  is the  $l$ th mean-removed patch in  $I_i^l$ . We define the  $l$ th filter output of the second PCA and all the filter output as:

$$Y = \left[ Y_1, Y_2, \dots, Y_N \right] \in R^{p_1 p_2 \times L_1 N mn} \quad (8)$$

The second stage of PCA filters are therefore expressed as:

$$W_I^2 = \text{mat}_{p_1 p_2} \left( q_I \left( YY^T \right) \right) \in R^{p_1 p_2}, I = 1, 2, \dots, L_2 \quad (9)$$

For each input image  $I_i^l$ , the output is  $L_1 L_2$  through two stage of PCA. The output is:

$$O_i^l = \left\{ I_i^l * W_I^2 \right\}_{l=1}^{L_2} \quad (10)$$

### Binary Quantization

For the input image  $I_i^l$ , it has  $L_2$  real-valued outputs  $\left\{ I_i^l * W_I^2 \right\}_{l=1}^{L_2}$  from the convolutional filter bank layer. Then we binarize the image and get

$$\left\{ H \left( I_i^l * W_I^2 \right) \right\}_{l=1}^{L_2} \quad (11)$$

where  $H(\bullet)$  is a Heaviside step function whose value is one for positive entries and zero otherwise. Around each pixel, we view the vector of  $L_2$  binary bits as a decimal number. Adjust the output matrix  $L_2$  as a single integer image:

$$T_i^l = \sum_{l=1}^{L_2} 2^{l-1} H \left( I_i^l * W_I^2 \right) \quad (12)$$

whose every pixel is an integer in the range  $\left[ 0, 2^{L_2} - 1 \right]$ . The order of the output matrix has nothing to do with weights.

### Block-wise Histograms

For each input image  $T_i^l$ ,  $i = 1, 2, \dots, L_1$ , we divide it into  $B$  block, and then calculate the histogram of each block, and all the grade  $B$  block histogram couplet a vector. Through such encoding process, the characteristics of the input images is defined as:

$$f_i = \left[ \text{Bhist} \left( T_i^l \right), \dots, \text{Bhist} \left( T_i^{L_1} \right) \right]^T \in R^{(2^{L_2})_{L_1 B}} \quad (13)$$

PCANet with data augmentation model parameters including the size of filter  $k_1, k_2$ , the number of each stage filter  $L_1, L_2$  the block size of histogram and block overlap ratio.

### Experimental result and analysis

We use MNIST and MNIST variations to investigate the influence of different block overlap ratios and filter size for PCANet with data arugmentation, and then compare with PCANet, BP and SVM.

### Recognition rate and Classification of time

In our experiment, the number of filters is fixed to  $L_1 = L_2 = 8$ , and the filter size is  $p_1 = p_2 = 7$ , and block size of histogram is  $7 \times 7$ , and no overlap between block and block. We use linear SVM classification. The recognition rate is shown in Table1. The classification of time is shown in Table2.

Table1: Recognition rate

algorithm	accuracy
PCANet with data augmentation	98.8%
PCANet	98.2%
BP	91.8%
SVM	97.23%

Table2: Classification of time

algorithm	Classification of time/sec
PCANet with data augmentation	261.16
PCANet	267.41

### The influence of different block overlap ratios and different filter size

In our experiment, the number of filters is fixed to  $L_1 = L_2 = 8$ , and the filter size is  $p_1 = p_2 = 7$ , and block size of histogram is  $7 \times 7$ . We only vary block overlap ratio from 0 to 0.7. The recognition rate is shown in Figure1. We only vary filter size from 3 to 17. The recognition rate is shown in Figure1.

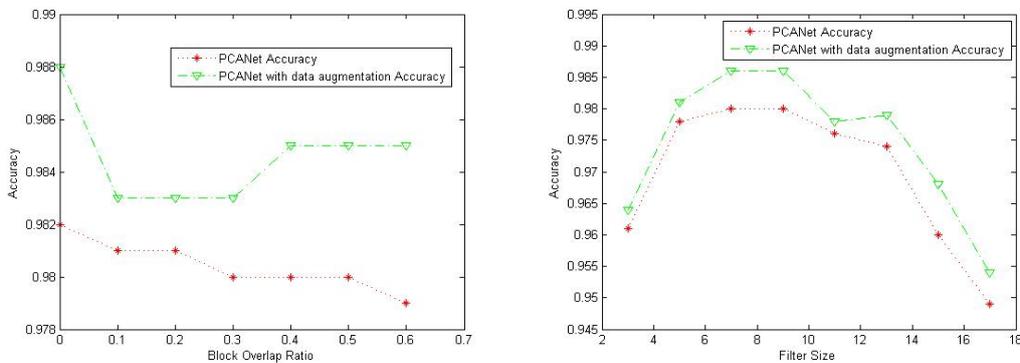


Figure1. Accuracy of PCANet and PCANet with data augmentation on MNIST basic test set for varying block overlap ratio and varying filter size

### Recognition rate on MNIST variations

In our experiment, the number of filters is fixed to  $L_1 = L_2 = 8$ , and the filter size is  $p_1 = p_2 = 7$ , and block size of histogram is  $7 \times 7$ , and no overlap between block and block. We test recognition rate on MNIST variations. MNIST variations are shown in Table3. The recognition rate is shown in Table4.

Table3: MNIST variations

Data Sets	Description	Num.of classes	Train-Valid-Test
MNIST	Standard MNIST	10	60000-0-10000
basic	Smaller subset of MNIST	10	10000-2000-50000
rot	MNIST with rotation	10	10000-2000-50000
bg-rand	MNIST with noise background	10	10000-2000-50000

bg-img	MNIST with image background	10	10000-2000-50000
bg-img-rot	MNIST with rotation and image background	10	10000-2000-50000
rect	Discriminate between tall and wide rectangles	2	1000-200-50000
rect-img	Dataset rect with image background	2	10000-2000-50000
convex	Discriminate between convex and concave shape	2	6000-2000-50000

Table4: Recognition rate on MNIST variations

Data Sets	PCANet with data augmentation	PCANet
basic	98.8%	98.2%
rot	92.83%	92.63%
bg-rand	94.11%	93.81%
bg-img	89.35%	89.05%
bg-img-rot	64.72%	64.52%
rect	99.86%	99.76%
rect-img	86.22%	85.92%
convex	95.94%	95.64%

### Filters visualization

Furthermore, we draw some the learned PCANet filters in Figure3.

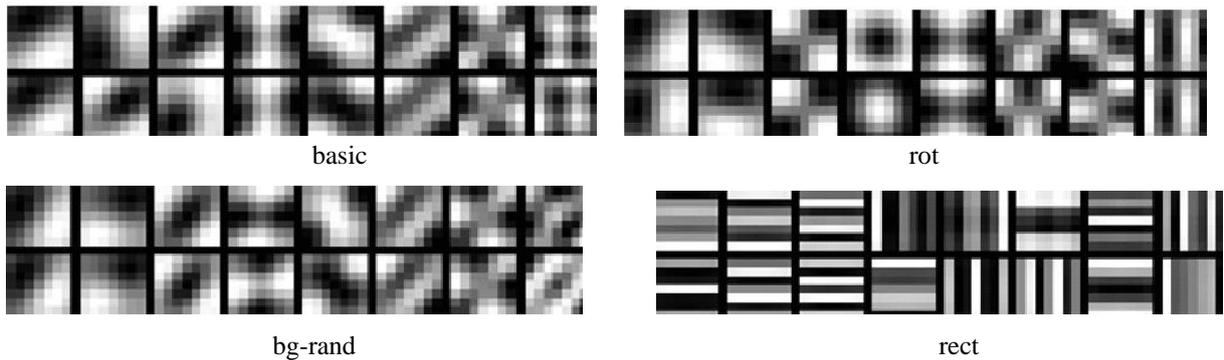


Figure3.The PCANet filters learned on basic, rot, bg-rand, rect .For each dataset, the top row shows the first filters, the bottom row shows the second filters.

As can be seen from the Table1 and Table2, PCANet with data augmentation is better than traditional SVM and BP in training time and recognition rate. The traditional BP neural network structure is simple, but the error rate is high, and easy to fall into local minimum. PCANet has a higher recognition rate than BP and SVM. Our recognition rate increased by 0.6% compared with PCANet.

We can see from Fig1, when block overlap ratio is between 0.4 and 0.6, recognition rate of PCANet with data augmentation is almost unchanged, but recognition rate of PCANet is on the decline, and the recognition rate is low.

We can see from Fig2, filter size is not the bigger the better, when the filter size is between 6 and 10, the recognition rate is high. The recognition rate is on the decline when the filter size is between 10 and 17. But overall, recognition rate of PCANet is superior to PCANet.

As MNIST has many training data, we report recognition on MNIST variations in Table4.We draw the learned PCANet filters on basic, rot, bg-rand, rect. For rect, we can see both horizontal and vertical stripes, for these patterns attempt to capture the edges of the rectangles.

## Summary

The PCANet with data augmentation is a very simple learning network, it can extract useful information efficiently for classification of digits. It solves several classic CNN algorithm's problems, such as long training time and need special skills to adjust parameters. We make improvement on the network's input and add data augmentation. This operation increase the diversity of training sample and make the network learn more weights. We use PCA to extract photos features convolution kernel initialization, because PCA can reduce the computational cost of random convolution kernel initialization. For sample insufficient situation, this method also has higher robustness.

## Acknowledgements

This work is supported in part by National Natural Science Foundation of China (Grant No. 61302063)

## References

- [1] Jun-sheng Chen. Composite structure characteristics of free handwritten numeral recognition algorithm [J]. Computer engineering and application, 2013, 49 (5) : 179-184.
- [2] Hui, guo-long chu, yellow GeFeng, etc. Handwritten numeral recognition method based on image recognition technology [J]. Computer technology and development, 2011, 21 (12) : 48-51.
- [3] Stone will fang. Support vector machine (SVM) and its application in handwritten numeral recognition [D]. Chongqing university, 2013.
- [4] Stone will fang, xiao-bing hu, Liu Ruijie, etc. Based on the heuristic GA - SVM handwritten digital character recognition research [J]. Computer technology and development, 2012, 10 (10) 5-9.
- [5] G. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," Neural Computation, vol. 18, no. 7, pp. 1527–1554, 2006.
- [6] Y. Bengio, A. Courville, and P. Vincent, "Representation learning:a review and new perspectives," IEEE TPAMI, vol. 35, no. 8, pp.1798–1828, 2013.
- [7] R. Hecht-Nielsen, "Theory of the backpropagation neural network," inInternational Joint Conference on Neural Networks, 1989, pp. 593–605.
- [8] A. Krizhevsky, I. Sutskever, and G. Hinton, "Imagenet classification with deep convolutional neural network," in NIPS, 2012.
- [9] K. Kavukcuoglu, P. Sermanet, Y. Boureau, K. Gregor, M. Mathieu, and Y. Le Cun, "Learning convolutional feature hierarchies for visual recognition," in NIPS, 2010.
- [10] H. Lee, R. Grosse, R. Rananth, and A. Ng, "Convolutional deepbelief networks for scalable unsupervised learnig of hierarchical representation," in ICML, 2009.
- [11] Simard P Y, Steinkraus D, Platt J C. Best Practice for Convolutional Neural Networks Applied to Visual Document Analysis[C]// In International Conference on Document Analysis & Recognition. 2003:958--962.
- [12] Schmidhuber J. Multi-column deep neural networks for image classification[J].Eprint Arxiv: 1202, 2012, 157(10):3642-3649.