Proceedings of the Fifth IIEEJ International Workshop
on Image Electronics and Visual Computing 2017
Da Nang, Vietnam, February 28 - March 3, 2017

HANDWRITTEN DIGITS CLASSIFICATION BY USING CONVOLUTIONAL NEURAL NETWORKS PRE-TRAINED FOR LARGE-SCALE OBJECT IMAGE DATASET

Yoshihiro Shima†
†School of Science and Engineering, Meisei University

Yumi Nakashima††
††School of Information Science, Meisei University

ABSTRACT

Neural networks are powerful technology for classification of character patterns and object images. A new method for handwritten digits recognition is proposed by combining Convolutional Neural Networks(CNN) and Support Vector Machines(SVM). Pre-trained CNN, Alex-Net can be used as pattern feature extractor. Alex-Net is pre-trained for large-scale object image dataset. An SVM is used as trainable classifier. The training 60k samples on MNIST database are trained by the SVM. The feature vectors of character patterns are passed to the SVM from Alex-Net. Experimental results of test error rate 1.03% on the test 10k MNIST database shows that proposed method is effective in handwritten digits recognition.

1. INTRODUCTION

A convolutional neural Network (CNN) is a powerful machine learning technique in the field of deep learning. A neural network has been a machine learning technique in the field of character recognition[1]. A huge amount of labeled data are needed for training[2]-[6]. An easy way to leverage the power of CNNs without investing time and effort into training, is to use a pre-trained CNN as a feature extractor. For generating complex decision surfaces, Support Vector Machine (SVM) is an extremely economical way of representing complex surfaces in high-dimensional spaces, including polynomials and other type of surfaces[3][8][9].
The focus of this paper is on combination of CNN and SVM for handwritten digits recognition. As a pre-trained CNN, Alex-Net[4] trained for large-scale object image dataset, is used for extractor of character pattern feature. An SVM is used for trainable classifier. The performance of proposed handwritten digits recognition is experimentally evaluated by using MNIST database.

2. CHARACTER PATTERN RECOGNITION BY COMBINING PRE-TRAINED CNN AND SVM

The usual method of recognizing individual patterns consists essentially of two modules shown in Fig. 1. The first module is the feature extractor. The second module is the trainable classifier[1][3]. The feature extractor transforms the inputted raw image to the feature vectors. The classifier is trained by using a large amount of feature vectors and class category from raw images dataset. Evaluating the classification accuracy, the feature vector is passed to the classifier and the class category of inputted raw image is outputted.

Fig.2 shows a proposed structure of handwritten digits recognition. The first module is pre-trained CNN used as feature extractor. The second module is SVM used as trainable classifier. The CNN is pre-trained for large-scale object image dataset. The pre-trained CNN of object image dataset is used for handwritten digits recognition. That is the difference of the reference [3].

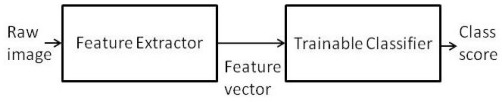


Fig. 1 A structure of pattern recognition.

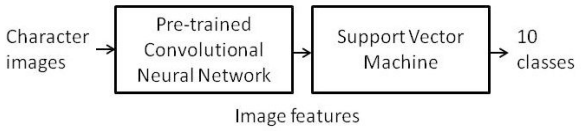


Fig. 2 A proposed structure of handwritten digits recognition using image category classification.

2.1. Pre-trained CNN as Feature Extractor

As a pre-trained CNN, Alex-Net is downloaded and used for image feature extractor[10]. Alex-Net has been trained on the ImageNet dataset, which has 1000 object categories and 1.2 million training images[5][6]. Fig. 3 shows the layer architecture of Alex-Net. TABLE I shows the detail specification of layers. The first layer defines the dimensions of input image as 227x227x3. The intermediate layers are a series of 5 convolution layers and 3 fully connected layers, interspersed with rectified linear units (ReLU) and max-pooling layers. The final layer is the classification layer and has 1000 classes. Fig. 4 shows the sample images in large-scale object image dataset, ImageNet. A small number of character string images such as street signboards are included in ImageNet. A pre-trained CNN for object image dataset is used as a feature extractor. Fig. 5 shows first convolution layer weights[7].

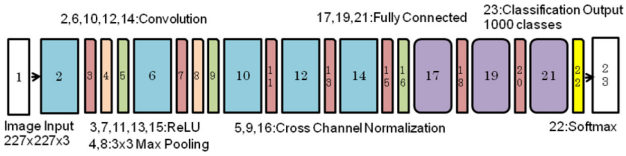


Fig. 3 Layer architecture of CNN (Alex-Net).

TABLE I Layers specification (Alex-Net).

No.	Name	Type	Explanation
1	input	Image Input	227x227x3 images with 'zerocenter' normalization
2	conv1	Convolution	96 11x11x3 convolutions with stride [4 4] and padding [0 0]
3	relu1	ReLU	ReLU
4	norm1	Cross Channel Normalization	cross channel normalization with 5 channels per element
5	pool1	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0]
6	conv2	Convolution	256 5x5x48 convolutions with stride [1 1] and padding [2 2]
7	relu2	ReLU	ReLU
8	norm2	Cross Channel Normalization	cross channel normalization with 5 channels per element
9	pool2	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0]
10	conv3	Convolution	384 3x3x256 convolutions with stride [1 1] and padding [1 1]
11	relu3	ReLU	ReLU
12	conv4	Convolution	384 3x3x192 convolutions with stride [1 1] and padding [1 1]
13	relu4	ReLU	ReLU
14	conv5	Convolution	256 3x3x192 convolutions with stride [1 1] and padding [1 1]
15	relu5	ReLU	ReLU
16	pool5	Max Pooling	3x3 max pooling with stride [2 2] and padding [0 0]
17	fc6	Fully Connected	4096 fully connected layer
18	relu6	ReLU	ReLU
19	fc7	Fully Connected	4096 fully connected layer
20	relu7	ReLU	ReLU
21	fc8	Fully Connected	1000 fully connected layer
22	prob	Softmax	softmax
23	classificationLayer	Classification Output	cross-entropy with other classes

Fig. 4 Example of sample images in large-scale object image dataset (ImageNet[6]) . Small number of signboard images contain some character patterns.

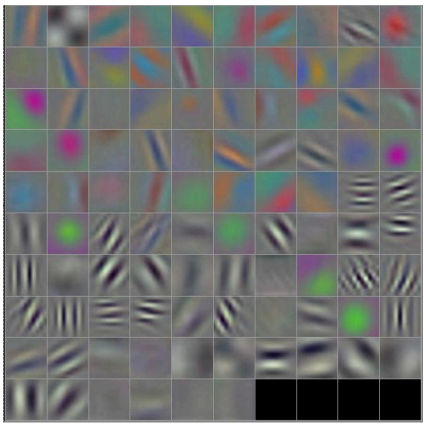


Fig. 5 First convolution layer weights (96 11x11x3 convolutions in Alex-Net). Those layer weights were pre-trained for ImageNet. The first layer of the network has learned filters for capturing blob and edge features.

2.2. SVM as Category Classifier

A multiclass SVM classifies data by finding the best hyperplane that separates all data points of one class from those of the other classes. SVM can represent complex surfaces including polynomials and radial basis function. The best hyperplane is the one with the largest margin between the two classes. Margin is the maximal width of the slab parallel to the hyperplane that has no interior data points. The support vectors are the data points that are closest to the separating hyperplane[8][9]. The multiclass SVM is trained using CNN features. A Stochastic Gradient Descent (SGD) solver is used for speed-up of the training when working with high-dimensional CNN feature vectors, which each have a length of 4096. Test image features are extracted by CNN and passed to the SVM classifier to measure the classification accuracy of the trained SVM classifier.

3. EXPERIMINTAL RESULTS

MATLAB[10] is used for the experiment of handwritten digits classification. As benchmark, the MNIST database is used for test and training samples.

3.1. Sample Images

The MNIST database is widely known handwritten digit recognition benchmark[1][2]. The database is modified version of NIST Special Database SD-3 and SD-1. The images are size normalized and centered by center of mass in 28x28 fields. The images contain grey levels as result of the anti-aliasing technique. MNIST training set contains 60,000 characters. MNIST test set contains 10,000 characters. Fig. 6 shows examples of the test set picked in order from the beginning. Fig. 7 shows average patterns created from MNIST database to observe the deviation of character pattern.

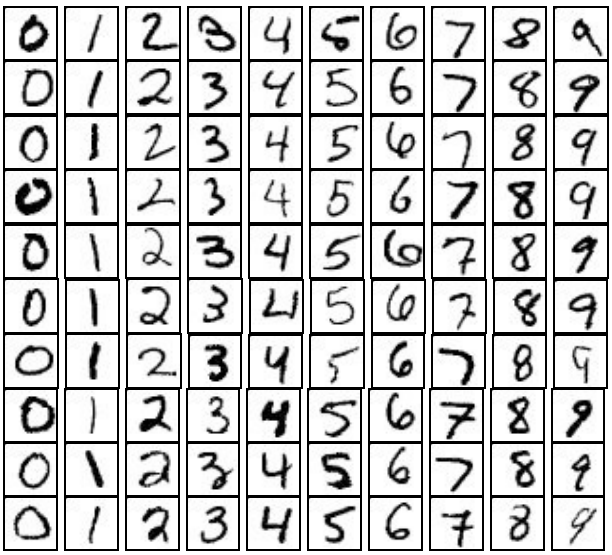
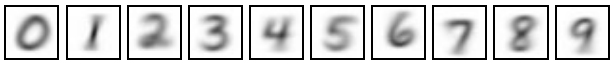


Fig. 6 Examples of test ten digits from MNIST database.



(a) training 60k samples

(b) test 10k samples

Fig.7 Average ten patterns of training and test samples created from MNIST database without normalization.

3.2. Error Rate

As test samples, 10,000 full test images are used from test 10k MNIST database. As training samples for SVM classifier, 60,000 training images are used without distortion[3][11] from training 60k MNIST database. The influence of the training set size is measured by picking training images randomly from training 60k MNIST database. The picking ratio is varied as 0.1, 0.3, 0.5, 0.7, 0.9, and 1.0. The picking population contains 5,410x10 class images selected from training 60k MNIST database. The reseason of less full samples is that the number of training images is balanced per category. Test and training error rate of proposed method is shown Fig. 8 and Fig. 9, respectively. The test error rate is 1.03% as the best score. The average of 5-times test is 1.336%. Fig. 10 shows the number of error samples in each class.

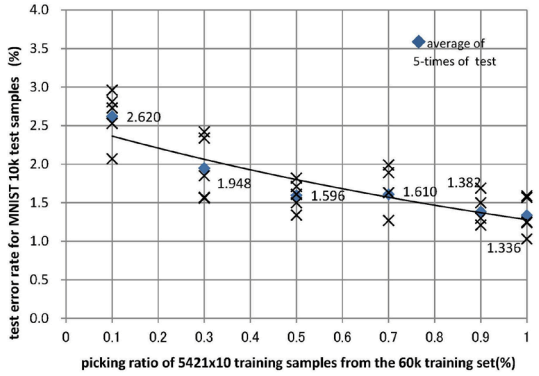


Fig. 8 Test error rate with various sizes of training sets. The best test error rate is 1.03% at the average 1.336%.

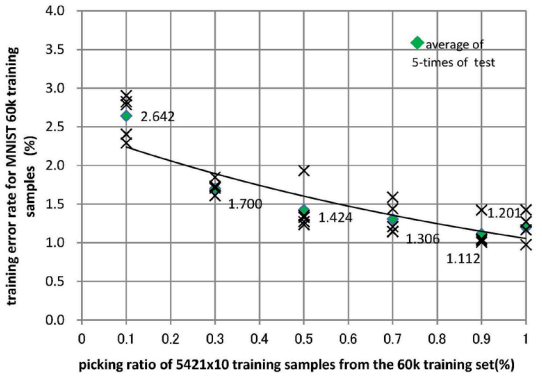


Fig. 9 Training error rate with various sizes of training sets. The number of evaluating images is a fixed constant, 60k.

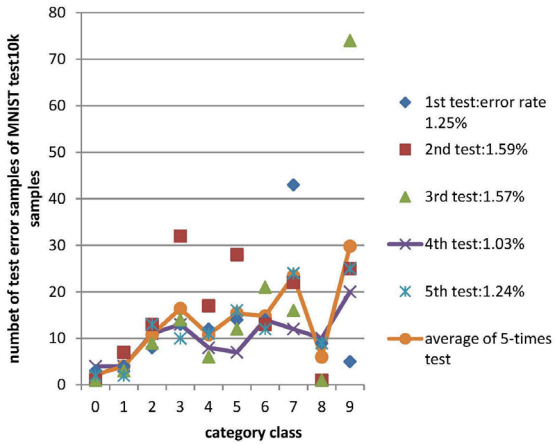


Fig. 10 Example of the number of error samples in each class for test 10k MNIST database. Data series of the graph is 5-times test result at the picking ratio 1.0. Pattern 7 and 9 are more misclassified, compared with pattern 0, 1 and 8.

3.3. Misclassified Samples and Confusion Matrix

All 103 misclassified test patterns are shown in Fig. 11. Some of those patterns are genuinely ambiguous. But several patters are perfectly identifiable by human eyes.

Fig. 11 The 103 test patterns misclassified by proposed method for test 10k MNIST database. Below each image is displayed the ground truth (left) and the misanswer (right).

TABLE II Example of Confusion matrix for test 10k MNIST database. The test error rate is 1.03%, that is the best score of 5-times test. The average accuracy is 98.97%.

Predicted /True	class 0	1	2	3	4	5	6	7	8	9	Total error	Sample No. in true class	Recall
class 0	976 99.59%	0	1 0.10%	1 0.10%	0	0	1 0.10%	0	0	1 0.10%	4 0.41%	980	99.59 %
class 1	0	1131 99.65	2 0.18	0	0	1 0.09	0	1 0.09	0	0	4 0.35	1135	99.65
class 2	0	1 0.10	1021 98.93	0	0	0	0	7 0.68	3 0.29	0	11 1.07	1032	98.93
class 3	1 0.10	0	1 0.10	997 98.71	0	7 0.69	0	2 0.20	2 0.20	0	13 1.29	1010	98.71
class 4	0	1 0.10	1 0.10	0	974 99.19	0	0	0	0	6 0.61	8 0.81	982	99.19
class 5	0	0	0	4 0.45	0	885 99.22	1 0.11	1 0.11	0	1 0.11	7 0.78	892	99.22
class 6	3 0.31	4 0.42	0	0	0	4 0.42	944 98.54	0	2 0.21	1 0.10	14 1.46	958	98.54
class 7	0	4 0.39	6 0.58	0	1 0.10	1 0.10	0	1016 98.83	0	0	12 1.17	1028	98.83
class 8	0	0	1 0.10	2 0.21	1 0.1	1 0.1	2 0.21	1 0.10	964 98.97	2 0.21	10 1.03	974	98.97
class 9	1 0.10%	0	2 0.20%	1 0.10%	7 0.70%	3 0.30%	0	6 0.60%	0	989 98.02	20 1.98%	1009	98.02
Total error	5	10	14	8	9	17	4	18	7	11	103 1.03%		
Sample No. in predetected class	981	1141	1035	1005	983	902	948	1034	971	1000		1000	
Precisiom	99.49%	99.12	98.65	99.20	99.08	98.12	99.58	98.26	99.30	98.90			98.97%

The example of confusion matrix for test 10k MNIST database is shown in TABLE II. Confusion matrix quantifies the probability that a given digit is clustered properly. The confusion matrix is made in case of the test error rate 1.03% achieved. The diagonal elements of the confusion matrix represent the number of samples for which the predicted class is equal to the true class, while off-diagonal elements are those that are mislabeled by the classifier. The correct rate of pattern 0, 1, and 8 is higher than pattern 7 and 9.

3.4. Processing Time

Elapsed time of training and classification procedure of MATLAB[10] is shown in TABLE III. Specification of the experimental system is shown in TABLE IV.

TABLE III Elapsed time measured.

	training	classification
time	1086 sec / 60k samples	0.287 sec/sample

TABLE IV Specification of the experimental system.

CPU	Intel Core i5-6400® (3.3GHz/4 core)
main memory	8GB PC3L-12800 (1600MHz)
graphic board	NVIDIA® GeForce® GT 730(4GB)

3.5. Discussion about Experimental Results

According to the report of the well-known CNN, LeNet-5[1], the error rate 0.95% was achieved without pattern distortion for test 10k MNIST database, while the test error rate 1.03% is achieved by the proposed method for the same test sets. These two error rates are close by value. One critical difference between LeNet-5 and the proposed method is that neural network layers of LeNet-5 are exclusively trained by handwritten digits pattern set, MNIST[2]. On the other hand, Alex-Net[4] was pre-trained by ImageNet[6], in which many variety of object images are included.

4. CONCLUSION

Our experiments have shown that using pre-trained convolutional neural networks as feature extractor for handwritten digits recognition is very promising approach. As feature extractor of handwritten digits patters, we use Alex-Net pre-trained for large-scale object image dataset, ImageNet. As trainable classifier, Support Vector Machine is used by combining with Alex-Net. Without distortion, test error rate 1.03% is achieved for test 10k MNIST database with training 60k MNIST database. Our error rate 1.03% is almost close by value to 0.95% of well-known LeNet-5. The utilization of ImageNet images for training CNN instead of MNIST images is very useful to recognize digits.

5. ACKNOWLEDGEMENTS

The authors would like to thank Prof. Michio Yasuda for valuable comments and helpful discussions.

6. REFERENCES

[1] Y. Lecun, L. Bottou, Y. Bengio, P. Haffner, “Gradient-based learning applied to document recognition,” Proceedings of the IEEE (Volume:86 ,Issue:11), pp. 2278-2324, Nov 1998.
[2] Yann LeCun, Corinna Cortes, Christopher J.C. Burges, THE MNIST DATABASE of handwritten digits, [Online]. Available: http://yann.lecun.com/exdb/mnist/, Sep. 2016.
[3] Fabien Lauer, Ching Y. Suen, Gerard Bloch, "A trainable feature extractor for handwritten digit recognition", Pattern Recognition, 40(6), pp. 1816-1824, Elsevier, 2007.
[4] Alex Krizhevsky, IIya Sutskever, Geoffrey E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Neural Information Processing Systems, 2012.
[5]Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li and Li Fei-Fei, “ImageNet: A Large-Scale Hierarchical Image Database,” IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009 , June 2009.
[6] IMAGENET: Large Scale Visual Recognition Challenge 2012 (ILSVRC2012), [Online]. Available: http://image-net.org/challenges/LSVRC/2012/browse-synsets, Sep. 2016.
[7] Matthew D. Zeiler, Rob Fergus, "Visualizing and understanding convolutional networks," ECCV 2014, Part 1, LNCS 8689, pp. 818-833, 2014.
[8] Chris J. C. Burges and Bernhard Schölkopf, “Improving the accuracy and speed of support vector machines,” in Advances in Neural Information Processing Systems 9, M. Joprdan M. Mozer and T. Petsche, Eds. 1997, The MIT Press.
[9] Dennis Decoste and Bernhard Schölkopf, “Training Invariant Support Vector Machines,” Machine Learning, 46 , pp.161-190, 2002, Kluwer Academic Publishers.
[10] MatConvNet, [Online]. Available: http://www.vlfeat.org/matconvnet/pretrained/, accessed Sep. 2016.
[11] Patrice Y. Simard, Dave Steinkraus, John C. Platt, “Best practices for convolutional neural networks applied to visual document analysis,” Seventh International Conference on Document Analysis and Recognition, Aug. 2003.