Image Augmentation for Object Image Classification Based On Combination of Pre-Trained CNN and SVM

Yoshihiro Shima1, *

1School of Science and Engineering, Meisei University, Hino, Tokyo, 191-8506, Japan
*shima@ee.meisei-u.ac.jp

Abstract. Neural networks are a powerful means of classifying object images. The proposed image category classification method for object images combines convolutional neural networks (CNNs) and support vector machines (SVMs). A pre-trained CNN, called Alex-Net, is used as a pattern-feature extractor. Alex-Net is pre-trained for the large-scale object-image dataset ImageNet. Instead of training, Alex-Net, pre-trained for ImageNet is used. An SVM is used as trainable classifier. The feature vectors are passed to the SVM from Alex-Net. The STL-10 dataset are used as object images. The number of classes is ten. Training and test samples are clearly split. STL-10 object images are trained by the SVM with data augmentation. We use the pattern transformation method with the cosine function. We also apply some augmentation method such as rotation, skewing and elastic distortion. By using the cosine function, the original patterns were left-justified, right-justified, top-justified, or bottom-justified. Patterns were also center-justified and enlarged. Test error rate is decreased by 0.435 percentage points from 16.055% by augmentation with cosine transformation. Error rates are increased by other augmentation method such as rotation, skewing and elastic distortion, compared without augmentation. Number of augmented data is 30 times that of the original STL-10 5K training samples. Experimental test error rate for the test 8k STL-10 object images was 15.620%, which shows that image augmentation is effective for image category classification.

1. Introduction
A convolutional neural network (CNN) is a powerful machine-learning technique used in the field of deep learning. A simple neural network has been used as a machine-learning technique in the field of character recognition and image classification [1]. However, a huge amount of labeled data are needed for training such a neural network [2]-[6]. An easy way to leverage the power of CNNs, without investing time and effort in training, is to use a pre-trained CNN as a feature extractor. For generating complex decision surfaces, a support vector machine (SVM) is an extremely economical way of representing complex surfaces in high-dimensional spaces, including polynomials and other types of surface [7][8].

The first focus of this paper is a novel method that we propose that combines a CNN and SVM for classifying object images. The second is on image augmentation for expanding data sets. Pattern distortion is useful for improving accuracy. As a pre-trained CNN, Alex-Net [4], trained for large-scale object image datasets, is used for the extractor of character pattern features. An SVM is used for the trainable classifier. The performance of the proposed object image classification is experimentally evaluated by using the STL-10 database without and with distortion. In this paper, a CNN and a SVM are combined for image classification. As a pre-trained CNN, Alex-Net [4] pre-trained for the large-
scale object-image dataset ImageNet is used as extractor of features. To avoid the consuming time for training, we use a pre-trained CNN. An SVM is used for trainable classifier. The performance of classification of object images is experimentally evaluated by using the STL-10 object-image dataset [9]. Instead of training, Alex-Net [10][11], pre-trained for ImageNet [4] is used.

In this paper, we make the following contributions. First, we show in experiments that data augmentation method such as cosine translation are useful for improving the accuracy of image category classification based on a pre-trained CNN and SVM. Second, we show in experiments that error rates are increased by other augmentation method such as rotation, skewing and elastic distortion.

2. Related work

2.1. Classification by deep learning
As showcased by the significantly improved state-of-the-art accuracy demonstrated from ILSVRC-2010 to ILSVRC-2014, massive progress has been made in large-scale object recognition [5]. At ImageNet LSVRC-2010, large, deep CNN was trained to classify 1.2-million high-resolution images into 1000 different classes [4]; on the test data, it achieved top-1 and top-5 error rates of respectively 37.5% and 17.0%. In contrast, at ILSVRC-2014, GoogleNet achieved top-5 error rate of 6.7% for image classification. It exploited an improved CNN architecture combining the multi-scale idea and increasing the depth and width of the network [5]. After that, the STL-10 dataset was used as a benchmark for developing unsupervised learning [9].

2.2. Augmentation by pattern distortion
To improve the learning of a machine, new training samples can be created by using prior knowledge on transformation-invariance properties. If the number of training samples is small and if the distribution to be learned has transformation-invariance properties, expanding the training set, that is, generating additional data using pattern distortion, may improve the classification performance. Yann LeCun et al. report that the results of classification on the MNIST database were improved by applying affine transformations [1]. Patrice Y. Simard et al. propose elastic distortion for expanding data sets [12]. Fabien Lauer et al. compare the recognition rate between elastic distortions and affine transformations [3].

3. Image category classification for object images
A scheme for classifying images in categories from object images is proposed as follows. It is based on a combination of a pre-trained CNN and a SVM. The CNN was pre-trained for object images and used as a feature extractor.

3.1. Target images of category classification
The STL-10 dataset are used as object images. The number of classes is ten as shown in figure 1. Training and test samples are clearly split. The STL-10 dataset is a benchmark for unsupervised learning of object images [9]. Images were acquired from labeled examples on ImageNet. The dataset contains 10 classes: “airplane,” “bird,” “car,” “cat,” “deer,” “dog,” “horse,” “monkey,” “ship,” and “truck.” Images are 96×96 pixels in color. Number of training images is 500 per class. Number of test images is 800 per class. The standardized testing protocol is recommended for unsupervised feature learning of the unlabeled dataset [9]. The standardized testing protocol was not used, since the pre-trained Alex-Net was used and trained by the labeled object images of ImageNet.

3.2. Recognition architecture combining pre-trained CNN and SVM
As shown in figure 2, the usual method of recognizing individual patterns essentially consists of two modules. The first module is a feature extractor; the second module is a trainable classifier [1][3]. The feature extractor transforms the inputted raw image into feature vectors. The classifier is trained by using a large amount of feature vectors and class categories from a dataset of raw images. To evaluate
classification accuracy, a feature vector is passed to the classifier, and the class category of the inputted raw image is outputted.

We use the combination of CNN and SVM. Combined structure for image category classification of object images is shown in figure 3. The first module is a pre-trained CNN used as a feature extractor. The second module is a SVM used as a trainable classifier. The CNN is pre-trained for a large-scale object-image dataset. Using features extracted by using a CNN for digit recognition was previously reported [3]. In our study, however, a CNN pre-trained on an object-image dataset is used for object image classification [10][11].

**Figure 1.** Examples of ten test object images from STL-10 dataset, which contains 10 classes: “airplane(s0),” “bird(s1),” “car(s2),” “cat(s3),” “deer(s4),” “dog(s5),” “horse(s6),” “monkey(s7),” “ship(s8),” and “truck(s9).”

**Figure 2.** Usual structure for pattern recognition.

**Figure 3.** Combined structure for image category classification.

**Figure 4.** Layer architecture of CNN (Alex-Net).

**Figure 5.** First convolution layer weights (96 $11 \times 11 \times 3$ convolutions in Alex-Net).
3.2.1. Pre-trained CNN as feature extractor. As a pre-trained CNN, Alex-Net was downloaded and used for an image-feature extractor [10][11]. Alex-Net has been trained on the ImageNet dataset, which has 1000 object categories and 1.2-million training images [4][5]. The layer architecture of Alex-Net is shown in figure 4.

The first layer defines the dimensions of an input image as 227×227×3. The layer weights were pre-trained for ImageNet. The first layer of the network learns filters for capturing blob and edge features. The intermediate layers are a series of five convolution layers and three fully connected layers, interspersed with rectified linear units (ReLU) and max-pooling layers. The final layer is the classification layer, which has 1000 classes. A pre-trained CNN for an object-image dataset is used as a feature extractor. First-convolution-layer weights are shown in figure 5 [10][11]. Instead of training object images such as those in the STL-10 database, a CNN pre-trained for object images, i.e., Alex-Net, was used as a feature extractor. The 17th layer of Alex-Net, named “fc7,” is connected to the SVM classifier. Feature vectors obtained at the 17th layer are passed to the SVM classifier [10][11].

3.2.2. 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. The SVM can represent complex surfaces including polynomials and a radial basis function. The best hyperplane is the one with the largest margin between the two classes. In other words, the 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 [7][8].

The multiclass SVM is trained using CNN features. A stochastic gradient descent (SGD) solver is used to speed up the training when high-dimensional CNN feature vectors, which each have a length of 4096 layers, are used [10]. To measure the classification accuracy of the trained SVM classifier, test image features are extracted by the CNN and passed to the SVM classifier.

3.3. Augmentation by pattern distortion

As data augmentation methods, affine transformation and elastic distortion are well-known in character recognition [3][12][13]. They are used to generate new samples from original samples and to expand training sets. Simple distortions such as translations, rotations, scaling, and skewing can be generated by applying affine displacement fields to patterns. Elastic distortion is an image transformation for imitating the variations in handwriting styles [12].

We use the pattern transformation method with the cosine function [13]. We also apply some augmentation method such as rotation, skewing and elastic distortion [14]. Figure 6 shows examples of patterns augmented by the cosine function. The number of training images was increased by 31 times as large as the original 5k training samples. By using the cosine function, the original patterns were left-justified, right-justified, top-justified, or bottom-justified. Patterns were also center-justified and enlarged. Figure 7 shows examples of patterns augmented by rotation, skewing and elastic distortion.

![Figure 6](image-url)

**Figure 6.** Training-data augmentation of STL-10 object images. Original images are translated and justified to the left, right, top, or bottom by cosine function. Images are also center-justified and enlarged.
4. Experimental results concerning category classification

MATLAB [10][11] was used for experimentally classifying object images. As a benchmark, the STL-10 dataset were used for test and training samples. Error rates for the dataset were compared with and without augmentation.

4.1. Error rate with and without augmentation

As test samples, 8,000 full-test images from test STL-10 dataset were used. First, as training samples for the SVM classifier, 5,000 training images from the training 5k STL-10 dataset were used without augmentation [9]. The influence of the training-set size was measured by picking training images randomly from the training 5k STL-10 dataset. Picking ratio was varied as 0.1, 0.2, 0.5, 0.8, and 1.0. Number of training samples was changed to 500, 2500, 4000, and 5000, respectively. Test error rate of the proposed method is shown in figure 8. As the best score, test error rate is 14.95%. Average error rate of five tests is 16.055%. Table 1 shows the comparison of test error rates for training dataset without augmentation and with cosine transformation for expanding the training set. Test error rate is decreased by 0.435 percentage points from 16.055% to 15.620% by augmentation. Number of augmented data is 30 times that of the original STL-10 5K training samples. Test error rates are compared by augmentation method, which is shown in table 2. Error rates are increased by other augmentation method such as rotation, skewing and elastic distortion.

Figure 7. Other training-data augmented STL-10 object images. Original images are rotated, skewed to x- and y-direction, and elastic distorted by random numbers.

Figure 8. Classification test error rate for STL-10 object images with various sizes of training sets. Best test error rate is 14.95% at average of 16.055%.
Table 1. Comparison of test error rates for training dataset without augmentation and with cosine transformation for expanding the training set.

| training dataset | STL-10 without augmentation; 5000(=10×500) images | STL-10 augmented by cosine function; 155000(=31×10×500) images |
|------------------|----------------------------------|-------------------------------------------------|
| test dataset     | STL-10 test dataset 8000 (=10×800) | STL-10 test dataset 8000 (=10×800)               |
| average error rate for five tests | 16.055% | 15.620% |
| minimum error rate         | 14.95%   | 15.50%   |
| maximum error rate         | 16.85%   | 15.78%   |
| five test error rates     | 14.95, 15.81, 16.19, 16.48, 16.85% | 15.50, 15.53, 15.60, 15.70, 15.78% |

Table 2. Test error rate comparison of augmentation method.

| augmentation method | average error rate for five tests | number of training images |
|---------------------|----------------------------------|--------------------------|
| rotation            | 18.645%                          | 105000(=21×10×500)       |
| skewing             | 16.625%                          | 205000(=41×10×500)       |
| elastic distortion(1) | 17.005%                        | 155000(=31×10×500)       |
| elastic distortion(2) | 16.655%                        | 20000(=4×10×500)         |
| cosine function     | 15.620%                          | 155000(=31×10×500)       |

Table 3. Confusion matrix for STL-10 database’s 8k test set with cosine transformation.

| Predicted class | True class | s0 | s1 | s2 | s3 | s4 | s5 | s6 | s7 | s8 | s9 | Total error | Error rate |
|-----------------|------------|----|----|----|----|----|----|----|----|----|----|-------------|------------|
| airplane : s0   | 739        | 8  | 3  | 2  | 1  | 2  | 1  | 2  | 23 | 19 |    | 61          | 7.63%      |
| bird : s1       | 9          | 721| 2  | 26 | 12 | 8  | 2  | 18 | 1  | 1  |    | 79          | 9.86%      |
| car : s2        | 9          | 3  | 724| 6  | 4  | 1  | 0  | 1  | 11 | 41 |    | 76          | 9.50%      |
| cat : s3        | 4          | 35 | 0  | 625| 54 | 33 | 16 | 31 | 1  | 1  |    | 175         | 21.88%     |
| deer : s4       | 3          | 20 | 0  | 37 | 707| 1  | 20 | 1  | 1 | 1 |    | 93          | 11.63%     |
| dog : s5        | 4          | 31 | 1  | 83 | 38 | 517| 66 | 56 | 3  | 1  |    | 283         | 35.38%     |
| horse : s6      | 2          | 26 | 5  | 38 | 53 | 24 | 624| 18 | 2  | 8  |    | 176         | 22.00%     |
| monkey : s7     | 1          | 41 | 1  | 60 | 41 | 15 | 3  | 637| 1  | 0  |    | 163         | 20.38%     |
| ship : s8       | 22         | 5  | 4  | 1  | 0  | 1  | 0  | 750| 17 | 5  |    | 50          | 6.25%      |
| truck : s9      | 13         | 3  | 26 | 3  | 1  | 2  | 5  | 1  | 30 | 716|    | 84          | 10.50%     |
| Total error     | 67         | 172| 42 | 256| 204| 87 | 113| 137| 73 | 89 |    | 1240        | 15.50%     |

Table 4. Measured elapsed time for image classification.

|                        | Training | Classification |
|------------------------|----------|----------------|
| STL-10 without augmentation | 120 s/5k samples | 0.285 s/sample |
| STL-10 with augmentation   | 3480 s/155k samples | 0.285 s/sample |

Table 5. Specification of experimental system.

|                |                        |
|----------------|------------------------|
| CPU            | Intel Core i5-6400® (3.3 GHz/4 cores) |
| Main memory    | 16 GB PC3L-12800 (1600 MHz) |
| Graphic board  | NVIDIA® GeForce® GT 730 (4 GB) |
4.2. Misclassified samples and confusion matrix
Examples of misclassified test images are shown in figure 9. Number of test STL-10 dataset images is 8k. Ground truth (left) and incorrect answer (right) are displayed below each image. Some object images are genuinely ambiguous.

An example of a confusion matrix for the STL-10 database’s 8k test set is shown in Table 3. It was made for the case where the error rate of 15.50% was achieved. The correct rate of images “airplane (s0),” “bird(s1),” “car(s2)” and “ship(s8)” was higher than that of images “cat(s3),” “dog(s5),” “horse(s6)” and “monkey(s7).”

4.3. Processing time
Elapsed time for the training and classification procedure of MATLAB [10][11] is listed in table 4. Clearly, training time increases with increasing number of training samples. For each test sample, classification time for the mixture dataset is similar to that for the discrete dataset. Specification of the experimental system is listed in table 5.

4.4. Discussion about classification accuracy
The state of the art of supervised learning on STL-10 dataset is the accuracy rate of 70.1 %, which was obtained by K. Swersky, et al. with 1k training set [15]. The state of the art of unsupervised learning is the accuracy of 73.15 %, which was obtained by S. Yang, et al. [16]. Testing protocol of our work is different from those of state-of-the-art learning methods. So it is not appropriate to compare with our model. Our supervised model archived 84.38 % test-set accuracy, exceeding the previous state-of-the-art. The pre-trained Alex-Net is used with 1000k training samples on ImageNet, by combining the SVM with 5k training set.

5. Conclusion
It was experimentally shown that using a pre-trained CNN as feature extractor for object images is a very promising approach. As a feature extractor, Alex-Net (pre-trained for the large-scale object-image dataset ImageNet) was used. As a trainable classifier, a support vector machine was used in combination with Alex-Net. Without augmentation, average test error rate was 16.055% for the test 8k STL-10 database and the training 5k STL-10 database. With augmentation of the training STL-10 datasets by cosine function, average test error rate is improved to 15.620%.
References

[1] Lecun Y, Bottou L, Bengio Y and Haffner P, “Gradient-based Learning Applied to Document Recognition,” Proceedings of the IEEE (Volume: 86, Issue: 11), pp. 2278-2324, Nov. 1998.

[2] LeCun Y, Cortes C and Burges C J C, The MNIST DATABASE of Handwritten Digits, [Online]. Available: http://yann.lecun.com/exdb/mnist/, Feb. 2017.

[3] Lauer F, Suen C Y and Bloch G, “A Trainable Feature Extractor for Handwritten Digit Recognition,” Pattern Recognition, 40(6), pp. 1816-1824, Elsevier, 2007.

[4] Krizhevsky A, Sutskever I and Hinton G E, “ImageNet Classification with Deep Convolutional Neural Networks,” Neural Information Processing Systems, 2012.

[5] Russakovsky O, Deng J, Su H, Krause J, Satheesh S, Ma S, Huang Z, Karpathy A, Khosla A, Bernstein M, Berg A C and Fei-Fei L, “ImageNet Large Scale Visual Recognition Challenge,” arXiv pre-print, arXiv: 1409.0575 v3[cs.CV], Jan. 2015.

[6] Zeiler M D and Fergus R, “Visualizing and Understanding Convolutional Networks,” ECCV 2014, Part I, LNCS 8689, pp. 818-833, 2014.

[7] Burges C J C and Schölkopf B, “Improving be Accuracy and Speed of Support Vector Machines,” in Advances in Neural Information Processing Systems 9, Joprdan M, Mozer M and Petsche T, Eds. 1997, The MIT Press.

[8] Decoste D and Schölkopf B, “Training Invariant Support Vector Machines,” Machine Learning, 46, pp. 161-190, 2002, Kluwer Academic Publishers.

[9] Coates A, Lee H and Ng A Y, “An Analysis of Single Layer Networks in Unsupervised Feature Learning,” AISTATS, 2011, [Online]. Available: https://cs.stanford.edu/~acoates/stl10/, Feb. 2017.

[10] MatConvNet, [Online]. Available: http://www.vlfeat.org/ matconvnet/ pretrained/, accessed Feb. 2017.

[11] MATLAB, Computer Vision System Toolbox, R2016a, MathWorks, [Online]. Available: http://www.mathworks.com/help/vision/examples/image-category-classification-using-deep-learning.html, 2016.

[12] Simard P Y, Steinkraus D and Platt J C, “Best Practices for Convolutional Neural Networks Applied to Visual Document Analysis,” Seventh International Conference on Document Analysis and Recognition, Aug. 2003.

[13] Guo J, Sun N, Nemoto Y, Kimura M, Echigo H and Sato R, “Recognition of Handwritten Characters Using Pattern Transformation Method with Cosine Function,” Trans. IEICE Japan, J76-D2(4), pp. 835-842, 1993 (in Japanese).

[14] Shima Y, Nakashima Y and Yasuda M, “Pattern Augmentation for Handwritten Digit Classification based on Combination of Pre-trained CNN and SVM,” 6th International Conference on Informatics, Electronics & Vision (ICIEV2017), Sept. 2017.

[15] Swersky K, Snoek J and Adams R P, “Multi-Task Bayesian Optimization,” Proceedings of the 26th International Conference on Neural Information Processing Systems (NIPS’13), Volume 2, pp. 2004-2012, Dec. 2013.

[16] Yang S, Luo P, Loy C C, Shum K W and Tang X, “Deep Representation Learning with Target Coding,” Proceedings of Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI’15), pp. 3848-3854, Jan. 2015.