Image Classification Using Convolutional Neural Networks With Multi-stage Feature

Junho Yim, Jeongwoo Ju, Heechul Jung, and Junmo Kim

Department of Electrical Engineering
KAIST 291 Daehak-ro, Yuseong-gu, Daejeon, Republic of Korea
{creationi,veryju,heechul,junmo.kim}@kaist.ac.kr
https://sites.google.com/site/siitkaist

Abstract. Convolutional neural networks (CNN) have been widely used in automatic image classification systems. In most cases, features from the top layer of the CNN are utilized for classification; however, those features may not contain enough useful information to predict an image correctly. In some cases, features from the lower layer carry more discriminative power than those from the top. Therefore, applying features from a specific layer only to classification seems to be a process that does not utilize learned CNN's potential discriminant power to its full extent. This inherent property leads to the need for fusion of features from multiple layers. To address this problem, we propose a method of combining features from multiple layers in given CNN models. Moreover, already learned CNN models with training images are reused to extract features from multiple layers. The proposed fusion method is evaluated according to image classification benchmark data sets, CIFAR-10, NORB, and SVHN. In all cases, we show that the proposed method improves the reported performances of the existing models by 0.38%, 3.22% and 0.13%, respectively.

1 Introduction

Image classification is an important topic in artificial vision systems, and has drawn a significant amount of interest over the last decades. This field aims to classify an input image based on visual content. Currently, most researchers have relied on hand-crafted features, HoG [1] or SIFT [2] to describe an image in a discriminative way. After that, learnable classifiers, such as SVM, random forest and decision tree are applied to extracted features to make a final decision. However, when a lot of images are given, it is too difficult problem to find features from those. This is the one of reasons that deep neural network model is coming. A few years ago, Hinton et al. [3] revealed the fascinating performance of deep belief nets, which use an effective deep learning algorithm, contrastive divergence (CD), in which each layer is trained layer by layer. Owing to deep learning, it becomes feasible to represent the hierarchical nature of features using many layers and corresponding weights. However, when the input dimension is too large to use, the deep belief network takes a long time to train. At that time, CNN [4],

J.-H. Kim et al. (eds.), *Robot Intelligence Technology and Applications 3,*
Advances in Intelligent Systems and Computing 345, DOI: 10.1007/978-3-319-16841-8_52

sharing weights by convolution method, solved this problem and improved the classification performance for various datasets. It should be noted that all those studies mentioned above have only used features from the top layer to train the following fully-connected layers. In contrast to this approach, Pierre et al. [5] bridged between the lower layer's output and the classifier to take the global shape and local details into account. This use of multi-stage features improved the accuracy over systems that use single stage features on a number of tasks, such as in pedestrian detection and certain sorts of classification. Motivated by many advantages of the multi-layers features, we propose an alternative multi-stage strategy that can be applied to a standard one track CNN whose weight parameter is fixed after the training has been finished without the multi-stage strategy in mind. The experiment results show that our approach can further improve performance of a standard one track CNN. Note that the proposed approach is different from the one in [5] in that the work in [5] trains the multi-stage architecture from the beginning, whereas the proposed method can be applied to a standard CNN whose training has been already finished. This paper includes our approach's motivation in Section 2 to easily help to understand why this model provides good result. The following, Section 3, describes our proposed model and explains how it works. We report experiment result on a various image classification data sets in Section 4, and conclude our research in Section 5.

2 Motivation

Since Matthew et al. [6] invented a probe to look inside a feature map, if one carefully observes the visualized features at each layer, one can obtain intuition as to why multi-stage features could enable further improvement of image classification. When comparing the visualization of features and the corresponding image patches, the latter has the greater variation since CNN mainly focuses on a discriminant structure. For other discoveries in [6], lower layer features are usually simpler than those of higher layers. The meaning of this discovery is that simple images are well activated at lower layers and complex images have high activation value at higher layers. Also, the lower layer features are focused on a smaller area in an image and the higher layer features are focused on a larger area in an image. For these reasons, the deep neural network model that uses the last layer features only finds it hard to classify the dataset which contains both simple and complex objects. This forces us to bind features from multi-stages in an effective way.

3 Model Discription

We propose a novel architecture using a two track deep neural network model. The first track is the deep convolutional neural network model, in which we want to enhance the ability; the second track is the assistance model which can raise the ability of the first track model by using multi-layer features. Any deep

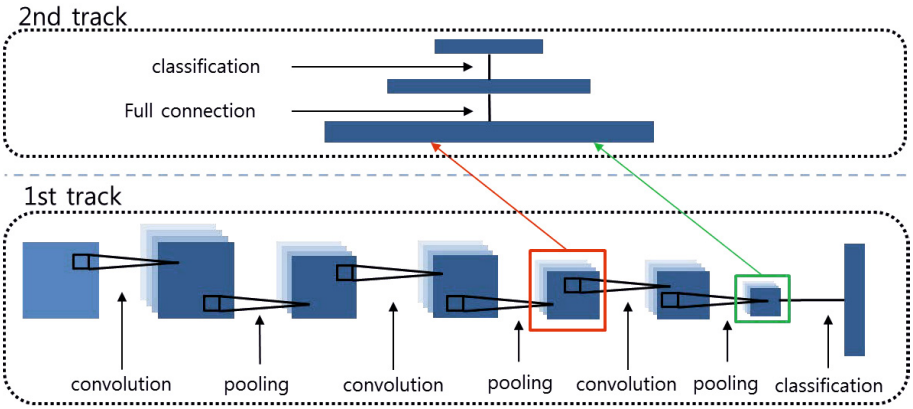

Fig. 1. Architecture of our proposed two track deep neural networks (DNN) model, composed of the first track, a learned CNN model, and the second track, assists the first model to enhance the ability. The particular layers features of the first track go to the input node of the second track model. For the purpose of mixing each layers feature information coming from the first track, the second track model is composed by fully connected layer.

convolutional neural network model that is composed of at least two convolutional layers with a pooling layer can be suitable for the first track model. As mentioned above, by using a pooling layer for the CNN, lower layer features and higher layer features are focused on different ranges. This is the reason that the CNN model with the pooling layer is suitable for the first track. For the second track, we use the restricted Boltzmann machine, RBM for the fully connected layer. Utilizing unsupervised training for the RBM, this model produces good initial weights for the following back propagation system which is supervised learning using label information. We append one more fully connected layer on the top layer with a classifier function as a softmax classifier. The second track operates after the learning of the first track has completed. As illustrated in Fig 3, the visible node of the second track comes from the particular layers feature of the first track. The number of nodes of the second layer in the second track is affected by the number of input nodes.

4 Experiments

In our experiments, we took the first track model for the simple convolutional neural network feature extractor described in Fig 3; this model is composed of three convolutional pooling layers, a fully connected layer, and the softmax classification layer. For the inputs of the second track model, the second and third pooling layers outputs of the first track model were used. The reason that we did not take the first layers output is that this output had much lower level features like edge levels and dimensions that were too large to use for the input.

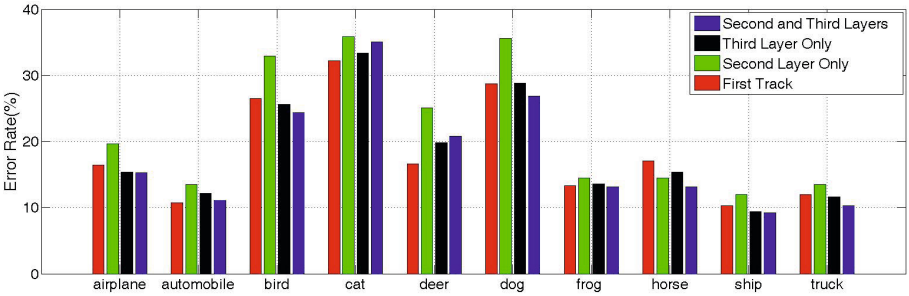

Fig. 2. CIFAR-10 classification error for each class. For the second track input, three different features are used; second layer only, third layer only, and the both layers. In addition, the first track result is attached for a comparison. For horse case, using second layer features show lower error rate than the first track model. Using both layers features improves performance in most cases.

For the unsupervised learning of fully connected layers of the second track, we used a learning rate of 0.001 with 50 epochs. Once the fully connected layers were trained we used them for the initial weights of supervised learning using label information. We trained 100 epochs for the supervised training. The whole set of experiments had the same setup with above.

4.1 CIFAR-10

CIFAR-10 is a dataset of natural RGB images of 32×32 pixels [7]. It contains 10 classes with 50,000 training images and 10,000 test images. All of these images have different backgrounds with different light sources. Objects in the image are not restricted to the one at center, and these objects have different sizes that range in orders of magnitude. For the first track model, we used the convolutional neural network model described in [8] (layers-18pct.cfg); this model is composed of three convolutional layers with 5×5 filters and 32 feature maps per layer, with a fully connected layer at the top of the layer. This model is shown in Fig 1. Before the training of the first track, we subtracted the mean values of the training set from each image. We trained for 120-10-10 epochs with an initial learning rate of 0.001 and a weight decay factor of 10. After the learning of the first track model, we extracted the second and third pooling layer features for each image and used them as the input for the second track model[1], described above. As shown in Table 1, using the second track model with the first track model is better than using only the first track model. Due to the fact that we used the features from the first model and not from the other model, it can be suggested that we enhanced the first model. To demonstrate this insight, we performed an additional experiment. For the training of the second track model, we used three different features, which are from the second pooling layer, the

[1] The number of nodes of second layer : 2000

Fig. 3. Ten classes of CIFAR-10 dataset images. Images have a different background with various light sources. Object in the image is not located on a center with various sizes.

third pooling layer, and both layers. We compared the results by class. Using the third layer feature was found to be better than using the second layer feature for most classes. However, what we have to focus on is that the result of using second layer feature only is pretty well and gets a better performance than the result of first track model for some case, horse class. This means that useful features are existing in the second layer feature. Furthermore, for most cases, by using both layers features, we were able to obtain a better performance than was possible when using the first track only.

Table 1. CIFAR-10 classification error (%) using our model and the first track model

Task	Proposed Method	First Track	Improvement
CIFAR-10	18.0 ± 0.11	18.38	0.38

4.2 NORB

We evaluated our two track model on the small NORB dataset (normalized-uniform), which is intended for 3D object recognition systems [9]. This dataset contains images of 50 toys belonging to 5 generic categories with 6 sets of lighting conditions, 9 elevations, and 18 azimuths. Each image consists of the binocular pair of 96 × 96 gray images with a normalized object size and a uniform background. We trained and tested the system on 24,300 images. Before using the images, the images were down-sampled to 48 × 48 and subtracted by the per-pixel mean. We used the same setup as that used in the CIFAR-10 experiment for the first track model. However, the first and third convolutional layers had 64 feature maps; the second convolutional layer had 32 feature maps. We trained

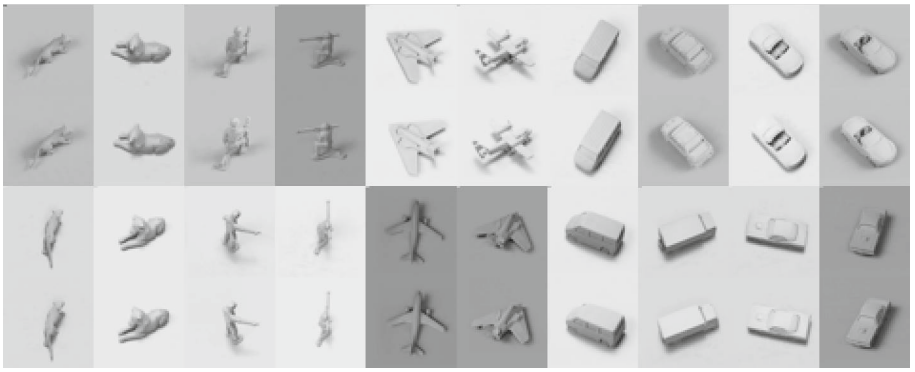

Fig. 4. Five classes of small NORB dataset images. Each two columns represent one class. Images have a same background with various light sources and object is on the center of image.

this model for 150-10-10 epochs with an initial learning rate of 0.001 and a weight decay factor of 10. For second track learning, we used the second and third pooling layer features of the first track. Because the input had large dimensions, we used the 3,500 unit fully connected layer for the second track. In this experiment, we saw a surprising improvement. As can be seen in Table 2, the proposed method reduces the error rate by 3.22%

Table 2. Small NORB classification error (%) using our model and the first track model.

Task	Proposed Method	First Track	Improvement
small-NORB	7.69 ± 0.13	10.91	3.22

4.3 SVHN

The Street View House Numbers dataset (SVHN) contains 10 digits [10], similar to the MNIST dataset [11]. Each image represents one digit. The challenging point of the SVHN dataset is that each image may contain multiple digits with different colors and various light sources. The training set contains 73,257 images; the testing set consists of 26,032 images. All images are cropped to 32×32 size and subtracted by per-pixel means. Our experiment for the SVHN dataset was set up in the same way as the CIFAR-10 experiment for the first track model. We trained the CNN model which contains three convolutional layer with 64, 64, and 128 feature maps with 5×5 filter size in layers 1, 2, and 3, respectively. This model was trained for 500-30-30 epochs with an initial learning rate of 0.001 and weight decay factor of 10. When the first track model was finished,

Fig. 5. Ten classes of SVHN dataset images. Each image represents the multiple digit in the real-world house number images. Images are cropped to 32 × 32 color images.

the second and third pooling layer features were input into the input node of the second track model, which contained 3,000 units with fully connected layers. Table 3 shows the classification performance of the proposed training model. Our proposed two track model enhances the performance of the first track model by 0.13%.

Table 3. SVHN classification error (%) using our model and the first track model.

Task	Proposed Method	First Track	Improvement
SVHN	5.95 ± 0.04	6.08	0.13

5 Conclusion

We propose a two track deep neural network model that is composed of an already learned CNN model and a fully connected layer model. Our model improves the learned CNN model's performance by using intermediate layer features. Via experiments in which we used our model on various datasets, we were able to demonstrate the needs of not only the top layers features but also those of the other layers features. The improved performance that resulted from the use of our model is due to the characteristic in which each layer's features focus on a different range of images. In the future, we will deal with a fine-grained dataset that requires a system to consider both global and local shape features.

Acknowledgement. This research was supported by the MOTIE (The Ministry of Trade, Industry and Energy), Korea, under the Technology Innovation Program supervised by KEIT (Korea Evaluation Institute of Industrial Technology), 10045252, Development of robot task intelligence technology.

References

1. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)
2. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision 60(2), 91–110 (2004)
3. Hinton, G.E., Osindero, S., Teh, Y.W.: A fast learning algorithm for deep belief nets. Neural Computation 18(7), 1527–1554 (2006)
4. Jarrett, K., Kavukcuoglu, K., Ranzato, M., LeCun, Y.: What is the best multi-stage architecture for object recognition? In: 2009 IEEE 12th International Conference on Computer Vision, pp. 2146–2153. IEEE (2009)
5. Sermanet, P., Kavukcuoglu, K., Chintala, S., LeCun, Y.: Pedestrian detection with unsupervised multi-stage feature learning. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3626–3633. IEEE (2013)
6. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional neural networks. arXiv preprint arXiv:1311.2901 (2013)
7. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images. Computer Science Department, University of Toronto, Tech. Rep. (2009)
8. Krizhevsky, A., Sutskever, I., Hinton, G.: Imagenet classification with deep convolutional neural networks. Advances in Neural Information Processing Systems 25, 1106–1114 (2012)
9. LeCun, Y., Huang, F.J., Bottou, L.: Learning methods for generic object recognition with invariance to pose and lighting. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2004, vol. 2, pp. II–97. IEEE (2004)
10. Netzer, Y., Wang, T., Coates, A., Bissacco, A., Wu, B., Ng, A.Y.: Reading digits in natural images with unsupervised feature learning. In: NIPS Workshop on Deep Learning and Unsupervised Feature Learning, vol. 2011 (2011)
11. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE 86(11), 2278–2324 (1998)