DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-231

Using Deep Convolutional Neural Networks for Image Retrieval

Chien-Hao Kuo[1], Yang-Ho Chou[2], and Pao-Chi Chang[1];
[1]Department of Communication Engineering, National Central University, Jhongli, Taoyuan, Taiwan
[2]Convergence Services Laboratory., Chunghwa Telecommunication Laboratories, Yangmei, Taoyuan, Taiwan

Abstract

In content-based image retrieval, the most challenging problem is the "semantic gap" between low-level visual features captured by machines and high-level semantic concepts perceived by human. This paper focuses on the high-level image features learning by the convolutional neural networks (CNN) in image retrieval. As a deep learning framework, CNN can extract meaningful image features in different layers, and transfer the image content into (abstract) semantic concepts. These high-level features descriptors can be better image representations than the hand-crafted feature descriptors, and further improve the image retrieval performance. The experimental results showed that layer-wise learning invariant feature hierarchies in CNN is good at feature representations. Using CNN for feature extractions on CIFAR-10 and CIFAR-100 dataset, it achieved 0.707 and 0.244 of mean average precision (MAP), respectively.

1. Introduction

With the rapid development of Internet and mobile devices, people can easily obtain audio and video information everywhere. Because of the exponential growth of multimedia information, how to efficiently retrieve and manage multimedia information from huge databases becomes an important issue. Up to now, many general purpose image retrieval systems have been developed.

The traditional image retrieval technique is based on text. In text-based image retrieval (TBIR), users can retrieve images based on keywords or textual descriptions which are annotated by human. But this technique has two major disadvantages: (1) a considerable level of human labor is required for manual annotation and (2) the annotation inaccuracy due to the subjectivity of human perception. In order to overcome these drawbacks, content-based image retrieval (CBIR) is introduced. In CBIR, user can retrieve images based on global or local features which are extracted from images such as color, texture, SIFT and HOG. Content-based image retrieval aims to search for images through analyzing their visual contents. Thus, learning effective feature representations and similarity measures are crucial to the retrieval performance of a CBIR system. Good feature representations basically depend on the feature descriptors, while most existing hand-crafted feature descriptors are considered low-level and far from what human normally perceived from the world.

One of the most challenging problems in CBIR is the "semantic gap" between low-level visual features captured by machines and high-level semantic concepts perceived by human [1]. Inspired by the recent successes of deep convolutional neural networks (CNN) in image classification tasks [2], it is possible for deep CNN to learn the hierarchies of feature representations effectively and fill the semantic gap.

Deep convolutional neural networks have been successfully applied to many recognition tasks including digit recognition (MNIST dataset [3]), face recognition [4, 15] or detection [16], and object recognition (NORB dataset [5]), and have drawn a lot of interest from the computer vision. Instead of applying to recognition tasks, this paper applies the high-level features learned by CNN to image retrieval tasks. CNN has exhibits its great potential when the networks are going wider (many maps per layer) and deeper (many layers) [6]. But it also consumes much more training time on CPUs. However, graphics processing units (GPUs) have been created for accelerating computations by parallel computing [4, 7] which can save lots of training time compared with using CPUs. As the GPU technologies creating amazing breakthroughs in accelerating computation, it has much more potential for using DNN.

In the following sections, we will review and describe the feature learning framework based on deep learning. In Section 4, we briefly introduce the common similarity measures in image retrieval tasks. Then, we show the experimental results on both image classification and retrieval tasks in Section 5.

2. Related works

In image classification tasks, Krizhevsky [8] trained a multi-layer generative model that learned to extract meaningful features which resemble those found in the human visual cortex. By pre-training a layer of features on a large set of unlabeled tiny images and training with restricted Boltzmann machine (RBM), objects classification on CIFAR-10 dataset was significantly improved. Based on RBM, Krizhevsky [9] further trained a two-layer convolutional deep belief network (DBN) which was composed of several RBMs, and focused on dealing with the boundary pixels of images by using global-connected units instead of convolutional units. Coates [10] applied several unsupervised feature learning algorithms using only single-layer networks including sparse auto-encoders, sparse RBMs on CIFAR-10 dataset. Chan [11] proposed a simple deep network for image classification. In this architecture, PCA was employed to learn multistage filter banks followed by simple binary hashing and block histograms for indexing and pooling.

In image retrieval tasks, Xia [12] developed a supervised hashing method for image retrieval, which simultaneously learned a good representation of images by deep convolutional neuron network. It firstly factorized the pairwise semantic similarity matrix into approximate hash codes for the training images and then trained a CNN with the approximate hash codes as well as the image tags. Lin [13] proposed an effective CNN framework to generate binary hash codes for fast image retrieval. Their method outperformed several state-of the-art hashing algorithms.

3. Feature Learning framework

The framework used in this paper could be divided into three stages: 1) pre-processing stages; 2) training stages; 3) testing stages, as shown in Fig. 1. The idea of features learning had been explored for learning features from labeled data by supervised training in neural networks. Here, we mainly focused on learning meaningful feature representations from training images by using the convolutional neural networks to learn representations of their input at the hidden layers. These hierarchies of learned features

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-231

could be regarded as discriminative feature representations of input image and applied to many recognition tasks.

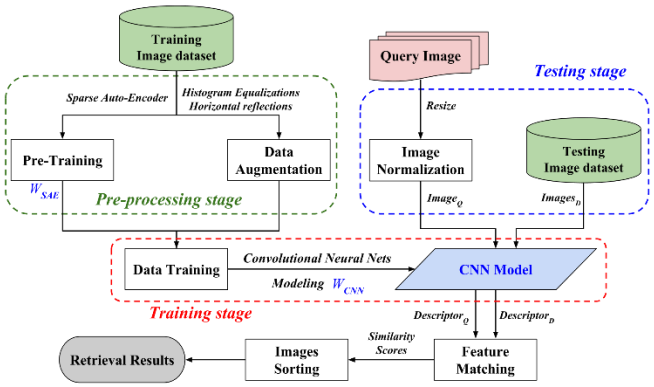

Figure 1. Block diagram of proposed image retrieval framework.

Pre-processing Stage

In this stage, two steps are performed before training the CNN model. The first step is using sparse auto-encoder (SAE) for pre-training the convolution kernels that will be used for extracting features in the CNN architecture. And the second step is artificially enlarging the training dataset via data augmentation scheme in order to prevent over-fitting. More details of pre-processing in two steps to learn better features are described as follows.

Pre-training the convolution kernels by SAE

In order to learn good feature representations, the model should have proper initial weights of the convolution kernels. We use an unsupervised learning algorithm for pre-training. In this work, the patch-based sparse auto-encoder was adopted [10]. The patches with size 5 by 5 were sampled randomly from unlabeled training images and used as the unlabeled input of an auto-encoder. And we also normalized the sampled patches by applying ZCA whitening.

The auto-encoder is trying to learn an approximation to the identity function, which means that its output $\hat{x}$ is similar to input x, as shown in Fig. 2.

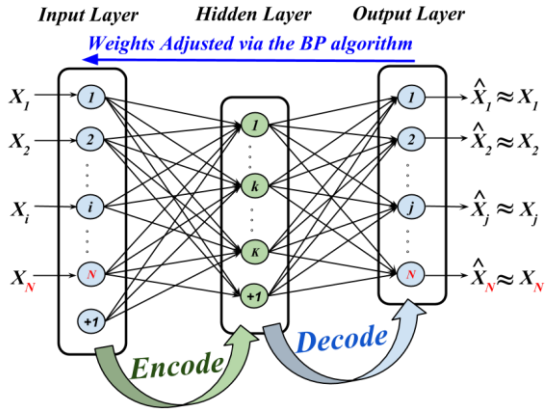

Figure 2. An auto-encoder aims to transform inputs into outputs with the minimum possible error and learns a compressed, distributed representation (hidden layer) for the input data.

Finally, back-propagation (BP) algorithm was applied to minimize squared reconstruction error with an additional sparsity penalty term restraining these hidden units to maintain a low average activation.

The pre-training weights $W_{SAE} \in \mathbb{R}^{64\times25}$ and biases $b_{SAE} \in \mathbb{R}^{64\times1}$ were taken as the initial weights of 64 convolution kernels with size 5 by 5 in CNN training model, as shown in Fig. 3. The proper initial weights can usually improve CNN training model which extracts the key features from the images.

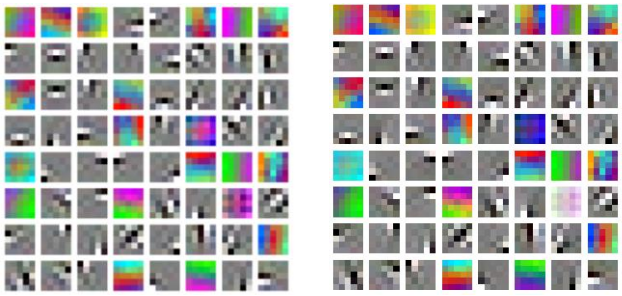

*Figure 3. In sparse auto-encoder, 64 features (with and without whitening) learned over 5*5 patches and these features will be taken as the convolution kernels in training stage.*

Data augmentation

The most common way to prevent over-fitting on an image dataset is data augmentation that artificially enlarge the training dataset using label-preserving transformations. Training images were augmented by two schemes. The first augmentation scheme was taking horizontal reflections of each image in order to produce different viewpoints of the training images, while the second augmentation scheme uses histogram equalization to increase the global contrast of the training images for image enhancement.

Notice that data augmentations should not change the class but the pixel values of the image. Here, we generated more data with translation and brightness invariant while not adding too much extra dimensions. Thus, the color images were converted to gray level before using horizontal reflections and histogram equalization schemes, as shown in Fig. 4 (c), (d). It created the fourth and fifth dimensions of training images with original three dimensions of RGB channels.

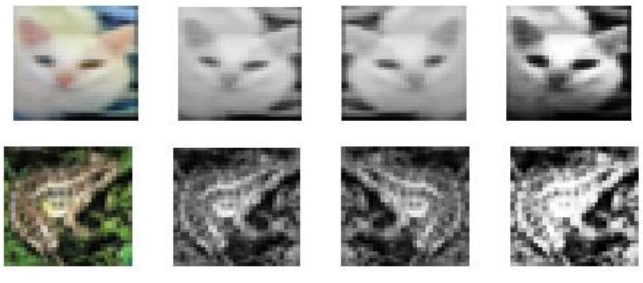

Figure 4. Data augmentation created additional feature information with the same label in different forms:
(a) RGB (b) Gray level (c) Horizontal reflections (d) Histogram equalization

Training Stage

As shown in Fig. 5, the 7-layer architecture of CNN was built. It took 64 convolution kernels with size 5 by 5 as convolution filters. It was composed of three pairs of convolutional and sub-sampling

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-231

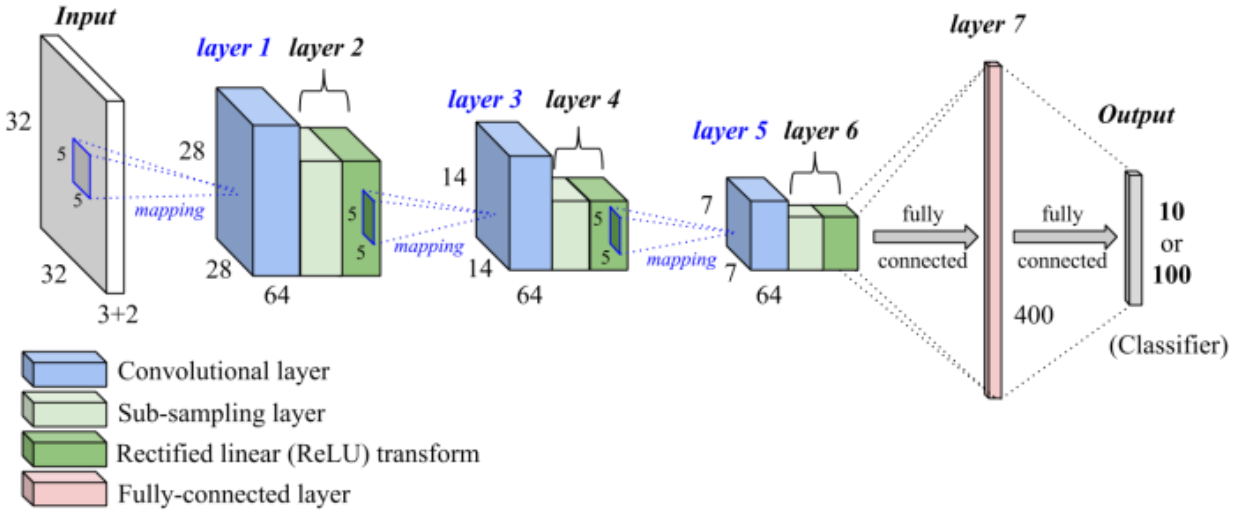

Figure 5. The 7-layer architecture of CNN model. We took the color images with size 32 by 32 as inputs and with additional two- dimensional image information created by data augmentation. Features were mapping layer-wisely and finally connected to a fully-connected layer as the representation of every input image

layers and was finally attached to a fully-connected layer that contains 400-dimensional neurons to represent the features of each image.

In Convolutional layers, the pre-training weights W_{SAE} was taken as the initial weights of convolution kernels, and features were extracted by applying a set of learned filters to the images in order to obtain a set of feature maps. In sub-sampling layers, it sub-sampled the feature maps of the previous layer in order to reduce variance.

The rectified linear transformation was adopted after each sub-sampling layer by transforming the neuron x into (1) to prevent vanishing gradient problem, as shown in Fig.6.

$$f(x) = \max(0, x) \tag{1}$$

In [14], neurons with this nonlinearity were regarded as Rectified Linear Units (ReLU), which made CNN training faster than their equivalents with saturating neurons.

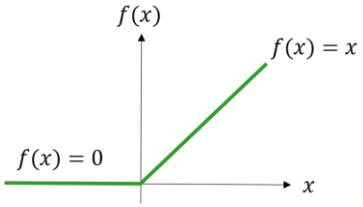

Figure 6. Rectified Linear Units (ReLU) through a non-linear transformations. [14]

After several convolutional layers alternating with sub-sampling layers and Rectified linear transformation, neurons were fully-connected to a 400-dimensional layer as feature vector. Each image can be represented by a 400-dimensional feature vector and we can just compared the difference between these feature vectors in recognition task.

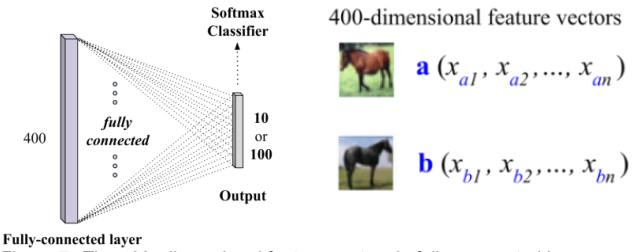

Figure 7. The 400- dimensional feature vectors in fully-connected layer can serve as feature descriptors of each image.

4. Similarity distance measures

As shown in Fig. 7, the extracted features vectors learned from training images can evaluate the feature descriptors. In classification tasks, the features emerging in the fully-connected layers of the CNN model can be the image representations, which are often attached to a softmax classifier for classifying images. While in retrieval tasks, the feature vectors learned to classify images were taken as the feature descriptors, and used for measuring the similarity between the images.

This paper used three different distance metrics to measure the similarity between images in image retrieval tasks. If the distance is small, there will be high degree of similarity, while a large distance leads to low degree of similarity. We compared the similarity between query image and the image dataset and sorted the final retrieval results based on the similarity distance metrics. Here are three distance metrics for measuring how much two images are alike:

Euclidean distance

The Euclidean distance between two points is the length of the path connecting them and it also known as L^2norm. For any two n-dimensional feature vectors, their Euclidean distance can be shown as (2):

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-231

$$d_{Euclidean}(a,b) = \sqrt{\sum_{k=1}^{n}(x_{ak} - x_{bk})^2} \quad (2)$$

Where x_{ak}, x_{bk} are two feature vectors of the images.

Manhattan distance

Manhattan distance is a metric in which the distance between two points is the sum of the absolute differences of their Cartesian coordinates and it also known as L^1 norm. For two n-dimensional feature vectors, their Manhattan distance is shown as (3):

$$d_{Manhattan}(a,b) = \sum_{k=1}^{n}|x_{ak} - x_{bk}| \quad (3)$$

Cosine distance

Cosine similarity is very efficient to evaluate sparse vectors and the Cosine distance is determined by Cosine of the angle between the two objects. Two vectors with the same orientation have a cosine similarity of 1, and two vectors diametrically opposed have a Cosine similarity of -1. Cosine similarity is defined as (4):

$$Similarity(a,b) = \frac{\sum_{k=1}^{n} x_{ak}x_{bk}}{\sqrt{\sum_{k=1}^{n} x_{ak}{}^2}\sqrt{\sum_{k=1}^{n} x_{bk}{}^2}} \quad (4)$$

Thus, Cosine distance between two vectors which is correlated with their Cosine similarity is shown as (5):

$$d_{Cosine}(a,b) = 1 - \frac{cos^{-1}\big(Similarity(a,b)\big)}{\pi} \quad (5)$$

5. Experimental Results

Our CNN model was evaluated on the image datasets for both classification and retrieval tasks by accuracy and mean average precision (MAP), respectively. By applying three different distance metrics, it exhibited much difference between their performances of sorting similar images in retrieval tasks.

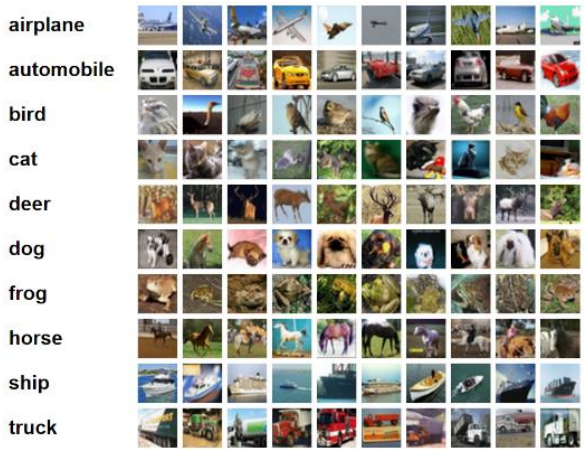

Figure 8. Example of images in the CIFAR 10 dataset. Each column shows samples belonging to the same category.

Dataset

The CIFAR-10 and CIFAR-100 dataset [8] was used for evaluation of performance in this experiment. The CIFAR-10 image dataset consists of 60,000 color images with size 32*32 which is a labeled subset of 80 million tiny images. It has 10 classes of different objects shown in Fig. 8. The training set has 5000 samples per class, and the testing set has 1000 samples per class.

In order to evaluate our CNN model for image retrieval tasks, which usually contain more image categories, we further used the CIFAR-100 dataset which is similar with CIFAR-10 but with more classes. It has 100 classes with training set 500 samples per class, and testing set 100 samples per class.

Image classification task

Table 2 showed the image classification accuracy on CIFAR-10 dataset. The method of {CNN_RGB} used the original training images as input of CNN without weights pre-training. It classified the CIFAR-10 dataset with 73.52% accuracy in 25 epochs. With two schemes of pre-training and data augmentation (DA), our method {Pre-train+CNN_RGB+DA} obtained the best accuracy of 79.29% in 25 epochs. It showed that the learned features were far more useful than the method of raw pixels.

Table 3 showed the classification results of our CNN model on CIFAR-100 dataset. For this task with more image classes, our method can still learn meaningful features from the training images. And our method {Pre-train+CNN_RGB+DA} also got better results than the method {CNN_RGB} with 2.4% increment in accuracy after 25 epochs.

In the classification task, it is important for the CNN being the good feature learning model which learns the meaningful feature representations. Therefore, we also measured the top-3 accuracy in 10 and 100 image classes. It showed that almost 97% of CIFAR-10 images were correctly classified in top-3 predicted classes, while 65% of CIFAR-100 images were predicted correctly in top-3 out of 100 classes. The experimental results indicated that our CNN model can learn useful feature descriptors on both datasets. Therefore, the features learned from images are considered to be discriminative descriptors for retrieval tasks.

Fig. 9 further showed the confusion matrix for CIFAR-10 by applying the method {Pre-train+CNN_RGB+DA}. The confusion matrix showed the classification accuracy in each class and that the classes of cat and dog were often misclassified to each other, while the transportation like automobile, ship and truck with over 85% accuracy.

Table 2. Image classification accuracy on CIFAR-10 dataset

Methods	Accuracy (%)		
	Top-1	Top-2	Top-3
Raw pixels (reported in [8])	37.3%		
RBM with BP [8]	64.8%		
Convolutional DBN [9]	78.9%		
Sparse auto-encoder [10]	73.4%		
PCANet [11]	78.7%		
Our CNN model in 25 epochs			
CNN_RGB	73.5%	87.4%	93.1%
CNN_RGB+DA	75.3%	88.4%	93.7%
Pre-train+CNN_RGB	77.9%	90.4%	95.0%
Pre-train+CNN_RGB+DA	79.3%	91.4%	96.9%

©2016 Society for Imaging Science and Technology
DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-231

Table 3. Image classification accuracy on CIFAR-100 dataset

Methods	Accuracy (%)		
	Top-1	Top-2	Top-3
Our CNN model in 25 epochs			
CNN_RGB	41.6%	54.9%	62.2%
CNN_RGB+DA	43.7%	56.3%	64.0%
Pre-train+CNN_RGB	43.2%	55.5%	62.6%
Pre-train+CNN_RGB+DA	44.0%	57.3%	65.2%

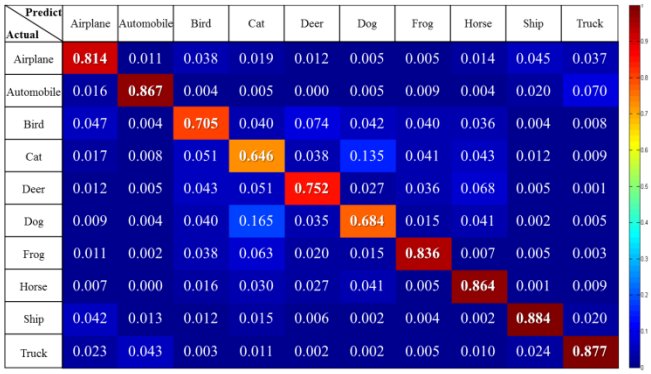

Figure 9. The Confusion matrix shows classification accuracy for the CIFAR-10 dataset. Actual class on vertical axis; predict class on horizontal axis.

Image retrieval task

In this task, we evaluated the retrieval results by using mean average precision (MAP). Each testing image had its representative feature vector in layer 7 of our CNN model which applies the method {Pre-train+CNN_RGB+DA}. The similarity between images was calculated by measuring the similarity distance between two feature vectors.

Figs. 10 and 11 showed the retrieval results based on sorting the cosine distance between query image and images from the dataset by top-30 minimum distance. For the query image of "dog", it's not surprisingly that some cats were appeared in this retrieval results. As for the query image of "automobile", some "truck" images showed in the ranked retrieval results.

Figure 10. The retrieval results of querying an image of "Dog" on CIFAR-10.

Figure 11. The retrieval results of querying an image of "Automobile" on CIFAR-10.

Table 4 and Table 5 showed the retrieval performance for CIFAR-10 and CIFAR-100 datasets in mean average precision. The best performance in two datasets was both applying Cosine distance as similarity distance metric. Their MAP performance can reach 0.707 in 10 classes and 0.244 in 100 classes, while Xia [12] using CNN as feature learning model and learning approximate hash codes with 48-bit reached 0.532 of MAP with CIFAR-10.

Table 4. Image retrieval MAP on CIFAR-10 dataset

Distance metrics	MAP	Matching time (s/query)
Euclidean Distance	0.671	0.085
Manhattan Distance	0.694	0.083
Cosine Distance	0.707	0.262

Table 5. Image retrieval MAP on CIFAR-100 dataset

Distance metrics	MAP	Matching time (s/query)
Euclidean Distance	0.189	0.115
Manhattan Distance	0.217	0.081
Cosine Distance	0.244	0.229

6. Conclusion

This paper proposed an image retrieval method using deep convolutional neural networks. It focused on the high-level image feature extraction by using deep learning to train the weights of neural network. The experimental results showed that layer-wise learning of feature hierarchies in CNN could extract the hierarchical image representation, which would fill the semantic gap. In order to learn the high-level semantic concepts of images well, we further pre-trained the initial weights of CNN and provided different a view of feature information by adopting the data augmentation scheme in our CNN model.

By using Cosine distance as the distance metric for similarity measure, we got the best retrieval results with MAP 0.707 and 0.244 on the CIFAR-10 and CIFAR-100 datasets, respectively.

DOI: 10.2352/ISSN.2470-1173.2016.2.VIPC-231

References

[1] J. Wan, D. Wang, C.H. Hoi, P.C Wu, "Deep Learning for Content-Based Image Retrieval: A Comprehensive Study," ACM International Conference on Multimedia, pp.157-166, 2014.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, pp. 1097-1105, 2012.

[3] Y. LeCun, F.J. Huang, and L. Bottou, "Learning Methods for Generic Object Recognition with Invariance to Pose and Lighting," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2004.

[4] D. Strigl, K. Kofler, and S. Podlipnig, "Performance and Scalability of GPU-Based Convolutional Neural Networks," Euromicro Conference on Parallel, Distributed, and Network-Based Processing, pp. 317-324, 2010.

[5] D. Cireşan, U. Meier, and J. Schmidhuber, "Multi-column Deep Neural Networks for Image Classification," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2012.

[6] D. Cireşan, et al., "Flexible, High Performance Convolutional Neural Networks for Image Classification," in International Joint Conference on Artificial Intelligence, pp.1237-1242, 2011.

[7] R. Uetz and S. Behnke, "Large-scale Object Recognition with CUDA accelerated Hierarchical Neural Networks," in IEEE International Conference on Intelligent Computing and Intelligent Systems, 2009.

[8] A. Krizhevsky, G. E. Hinton, "Learning Multiple Layers of Features from Tiny Images," Master's thesis, Department of Computer Science, University of Toronto, 2009.

[9] A. Krizhevsky, G. E. Hinton. "Convolutional Deep Belief Networks on CIFAR-10", Unpublished manuscript, 2010.

[10] A. Coates, A.Y. Ng, H. Lee, "An Analysis of Single-Layer Networks in Unsupervised Feature Learning," International Conference on Artificial Intelligence and Statistics, pp. 215-223, 2011.

[11] T.H. Chan, et al, "PCANet: A Simple Deep Learning Baseline for Image Classification?" IEEE Transactions on Image Processing, vol.24, no.12, pp.5017-5032, 2015.

[12] R. Xia, et al. "Supervised Hashing for Image Retrieval via Image Representation Learning", in Proceedings of the AAAI Conference on Artificial Intelligence, 2014.

[13] K. Lin, et al. "Deep Learning of Binary Hash Codes for Fast Image Retrieval," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. 2015.

[14] V. Nair and G. E. Hinton, "Rectified Linear Units Improve Restricted Boltzmann Machines," in Proceedings of International Conference on Machine Learning, 2010.

[15] H. Khalajzadeh, M. Mansouri, and M. Teshnehlab. "Face Recognition Using Convolutional Neural Network and Simple Logistic Classifier," Soft Computing in Industrial Applications. Springer International Publishing, 2014. 197-207.

[16] Farfade, S. S., Saberian, M., & Li, L. J., "Multi-view Face Detection Using Deep Convolutional Neural Networks,"arXiv preprint arXiv:1502.02766.

Author Biography

Chien-Hao Kuo *received his BS degree in communication engineering from National Central University (NCU), Taiwan, in 2009. He is currently pursuing his PhD degree at the Video-Audio Processing Laboratory in the Department of Communication Engineering at NCU, Taiwan. His research interests include video/image processing, object tracking and recognition.*

Yang-Ho Chou *received his BS degree and MS degree in communication engineering from National Central University, Taiwan, in 2013 and 2015, respectively. His research interest is image retrieval. He is currently researching into multimedia and deep learning techniques at Chunghwa Telecom Laboratories.*

Pao-Chi Chang *received his PhD degree in electrical engineering from Stanford University, California, in 1986. From 1986 to 1993, he was a research staff member at IBM T. J. Watson Research Center, New York. In 1993, he joined the faculty of NCU, Taiwan, where he is presently a professor in the Department of Communication Engineering. His main research interests include speech/audio coding, video/image compression, image and music retrievals, and deep learning techniques.*