IEIE Transactions on Smart Processing and Computing, vol. 3, no. 6, December 2014
http://dx.doi.org/10.5573/IEIESPC.2014.3.6.366

Wild Image Object Detection using a Pretrained Convolutional Neural Network

Sejin Park and Young Shik Moon

Department of Computer Science and Engineering, Hanyang University / Ansan South Korea
sjpark@visionlab.or.kr, ysmoon@hanyang.ac.kr

***** Corresponding Author: Sejin Park

Received February 20, 2014; Revised April 20, 2014; Accepted August 28, 2014; Published December 31, 2014

** Short paper*

** Invited Paper: None.*

** Review Paper: This paper reviews the recent progress possibly including previous works in a particular research topic, and has been accepted by the editorial board through the regular reviewing process.*

** Extended from a Conference: Preliminary results of this paper were presented at the IEIE Conference Fall 2014. This present paper has been accepted by the editorial board through the regular reviewing process that confirms the original contribution.*

***Abstract*:** This paper reports a machine learning approach for image object detection. Object detection and localization in a wild image, such as a STL-10 image dataset, is very difficult to implement using the traditional computer vision method. A convolutional neural network is a good approach for such wild image object detection. This paper presents an object detection application using a convolutional neural network with pretrained feature vector. This is a very simple and well organized hierarchical object abstraction model.

***Keywords*:** Image object detection, Machine learning, Deep learning, Convolutional neural network

1. Introduction

This paper reports a machine learning approach for image object detection. Object detection and localization in a wild image, such as a STL-10 image dataset, is very difficult to implement using traditional computer vision method. (SIFT, LBP, Haar, Adaboost [1-3])

The convolutional neural network is good approach for such wild image object detection. This network is a very simple and well organized hierarchical object abstraction model.

In the computer vision area, wild image object detection is a challenging subject because there are few common features and no general methodology. Fig. 1 gives an example of a wild image dataset from STL-10. The visual objects can be divided into 10 classes (car, airplane, cat, dog…), which are from natural scenes. They have distorted shapes, poses, colors, illumination, and occlusion. Furthermore, they are weakly correlated with each other in the same image class.

The most important item of natural scene object detection is how to represent the image data. Deep learning is one of the manifold learning approaches from raw image data with a very high dimensional transfer structure. Among the various deep learning methods, the most practically successive one is the Convolutional Neural Network (CNN).

CNN is basically composed of a convolutional layer, pooling layer and classification layer. At the convolutional layer, the transfer input image into convolved and overlapped feature space. At the pooling layer, it aggregates the results of the convolutional layer into a pooled adjacent feature. The activation of a convolutional layer followed by a pooling layer is connected to a fully connected layer that is composed of softmax regression or feedforward neural network. Between the pair of convolutional layers and pooling layers, there can be an additive pair of convolutional and pooling layers repeatedly.

Fig. 1. An example of a wild image dataset (STL-10).

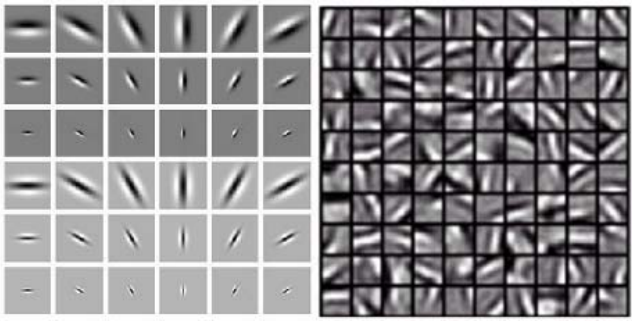

Fig. 2. Gabor filter (left) and Learned feature by self-taught learning (right).

The machine learning approach like CNN has many good points compared to the traditional approach. First, it is very easy to build huge database for a large training data set. A neural net or SVM is useful for such purposes. Second, it is easy to learn very complicated feature space from an image dataset directly. Machine learning attempts to use statistical reasoning to find approximate solutions for the difficulties of modelling natural scene features.

- This paper proposes the approach of a wild image object detection method using a deep neural network. In particular, a convolutional neural network is very efficient for solving a given problem. The contributions of this paper are as follows.
- This paper reports the feasibility of a machine learning approach to detect a natural scene image object;
- The approach uses GPU-based implementation for a very time consuming calculation of a convolutional neural network;

This approach improves the localization performance of pretrained CNN for partial objects in a large image;

The remainder of this paper is organized as follows. The next section describes the related works. The proposed scheme section presents a convolutional neural network model for natural scene image object detection. The performance evaluation section includes performance evaluations and discussions of the effects of the proposed scheme. Finally, the last section reports the conclusions.

2. Related Work

The traditional visual object detection method based on the feature based model is to search for discontinuities in image brightness. For example, the Canny edge detector extracts the boundaries in a hysteresis manner. The local boundaries detected are manipulated as a template, and a pattern recognition approach is performed to detect an image object. On the other hand, an edge based model approach cannot capture the very complicated variations of natural scene objects [4].

Another common approach is the local feature based model. SIFT [1], LBP [2] and Haar detector [3] captures the local feature based on a cortex like feature. Those methods are used widely in visual object detection, such as stereo camera correspondence mapping, face detection and content-based image retrieval. On the other hand, it difficult to build a huge model of local feature base from a massive training data set.

Recently, on the ImageNet challenge, which is famous wild image object recognition competition, the CNN based method exhibited state-of-the-art performance [5]. The method trained large, deep convolutional neural networks to classify the 1.2 million high-resolution wild images in the ImageNet contest into the 1000 different classes. On the test data, they achieved top-1 and top-5 error rates. This is proof of the essential use of machine learning methods on wild image recognition problem.

3. Pretrained Convolutional neural networks

This approach to represent a raw image is to use self-taught learning, which is one of unsupervised machine learning methods [6]. On self-taught learning, it is not assumed that unlabeled data follows the same class labels or generative distribution as labeled data. Therefore, a large number of unlabeled images can improve a given image classification. Normally, it outperforms the self-taught learning feature than a classification with labeled data only.

3.1 Learning Discriminative Feature

One of the good features in a raw image can be achieved by directional edge filters. The Canny edge detector, Histogram of Gradient and Gabor filter can be used to the edge features from a raw image. This method is well formulated and easy to exploit to computer vision applications but it cannot guarantee that those features are the best solution to image recognition.

Self-taught learning can learn such an discriminative feature from the image data directly. The result of self-taught learning normally resembles the edge filters.

The power of self-taught learning is that there is no need to be concerned about the data types of domain knowledge to obtain the discriminative features for new data groups. This means that a Gabor filter can be used only in 2 dimensional image space, but self-taught learning can be used not only in a 2 dimensional image space but

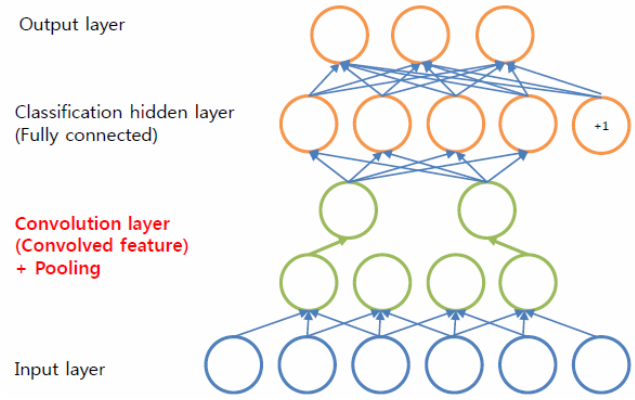

Fig. 3. Structure of the Convolutional neural network.

also more high dimensional data space like 3 channel color image space.

Self-taught learning can be performed using a sparse auto encoder. This is a feedforward neural network with the same input and output. The energy function of the sparse auto encoder is (1).

$$\text{minimize}_{W,b} \ \sum_i \left\| x_u^{(i)} - h_{W,b}(x) \right\|_2^2 + \beta \| W \|_2 \quad (1)$$

To complete the sparse auto encoder, one more constraint is needed, the energy function of the sparsity condition. This is composed of a squared sum of the activation function. In addition, it is added into the energy function as a sparsity term. This means that a sparse auto encoder is trained to minimize the cardinality of the activation values in a hidden unit.

On the other hand, it is affected by the sparsity parameter, which controls the sparsity of the activation of the hidden unit. This should be decided carefully by some experiment. In the present experiment, the sparsity parameter was chosen as 0.03.

The result of the sparse auto encoder can be called a pretrained feature. The pretrained feature had 8 x 8 dimensions, which corresponds to same image size of a local patch from a 64 x 64 full image.

3.2 Pretrained feature for CNN

CNN has a very special structure in a deep layered network composed by several different functional neurons. They are the convolutional layer, pooling layer and fully connected layer. CNN can be utilized with a pretrained feature set as an initial weight vector.

The CNN structure is described in Fig. 3.

The CNN is basically composed of a convolutional layer, pooling layer and classification layer. At the convolutional layer, the input image is transferred into convolved and overlapped feature space. At the pooling layer, the result of the convolutional layer is aggregated into the pooled adjacent feature. Activation of the convolutional layer followed by the pooling layer is fed to the fully connected layer, which is composed of softmax regression or feedforward neural network. Between the pair of convolutional and pooling layers, there can be an additive pair of convolutional and pooling layers, repeatedly.

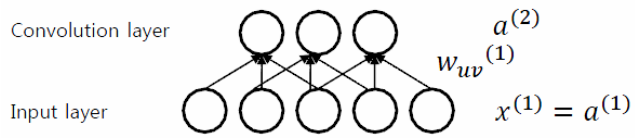

Fig. 4. Convolutional layer.

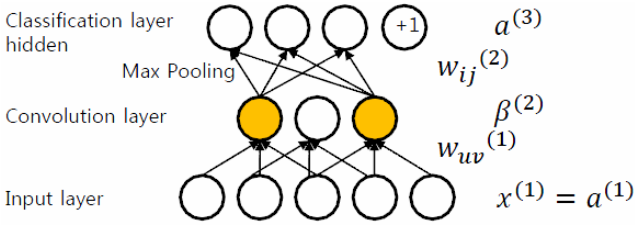

Fig. 5. Pooling layer.

Regarding the input image, there are convolution layer, which has a weight vector of the pretrained feature of an 8 x 8 patch. In the convolution feature, pretrained feature is shared in the weight vector of the convolutional layer. This is described in Fig. 4.

If a N x N input image is connected to the convolution layer and a pretrained feature (convolution kernel) size of m x m, the size of the convolved layer is (N - m + 1) x (N - m + 1). The activation of the convolution layer is calculated by (2).

$$\begin{aligned} &for\ i,j = 0,1\ \ldots\ N-m \\ &z_{ij}^{(2)} = \sum_{u=0}^{m-1}\sum_{v=0}^{m-1} w_{uv}^{(1)} x_{(i+u)(j+v)}^{(1)} \end{aligned} \quad (2)$$

where w means the convolution weight vector, x means the input image.

Pooling layer aggregates result of the convolution layer by averaging the image area of a k x k size. In that case, there are (N / k) x (N / k) of pooling layer neurons. This is described in Fig. 5.

The pooling layer is calculated using Eq. (3).

$$\begin{aligned} \beta_j^{(2)} &= MaxPool(a^{(2)}) \\ z_i^{(3)} &= w_{ij}^{(2)} \beta_j^{(2)} + b_i \\ a_i^{(3)} &= f(z_i^{(3)}) \end{aligned} \quad (3)$$

The pooling layer is followed by a fully connected layer or another convolutional layer. A fully connected layer is a simple softmax regression.

Fig. 6 describes the complete structure of a convolutional neural network.

3.3 Training of Pretrained CNN

The back propagation algorithm can also be adopted to

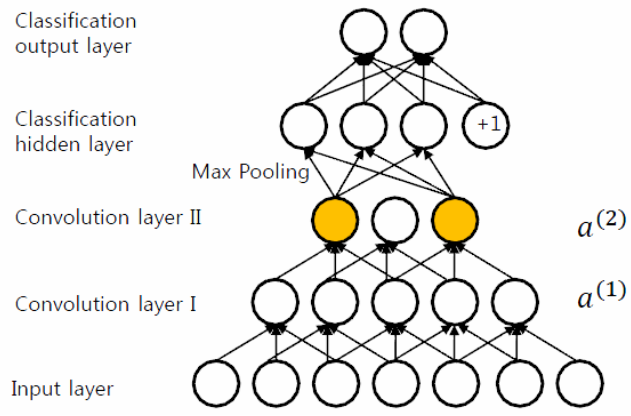

Fig. 6. Structure of the convolutional neural network.

a convolutional neural network. The only difference from a general feedforward neural network is gradient propagation at the pooling layer to the convolution layer. Propagated gradient can be calculated using Eq. (4).

$$\frac{\partial E}{\partial a_{ij}^{(1)}} = \sum_{u=0}^{m-1}\sum_{v=0}^{m-1} \frac{\partial E}{\partial z_{(i-u)(j-v)}^{(2)}} \cdot \frac{\partial z_{(i-u)(j-v)}^{(2)}}{\partial a_{ij}^{(1)}} = \sum_{u=0}^{m-1}\sum_{v=0}^{m-1} \frac{\partial E}{\partial z_{(i-u)(j-v)}^{(2)}} \cdot w_{uv}^{(1)} \quad (4)$$

where m means the element size of the training samples. z means the sigmoid activation of each neuron, and a means the weight vector and input convolved value. w is the weight vector at each layer.

The optimization algorithm, which is used in CNN training is a quasi-newton method. This is one of the 2nd order optimization algorithms, and a variation of the Newton method. The quasi Newton method calculates the approximated Hessian alternatively because the Hessian needs to calculate every 2nd order derivative for a k x k dimensional vector and it is too expensive to compute at every iteration.

3.4 Negative Image Training

To detect a partial object in a large image, one more class is needed for a negative image. This means none of the trained class categories were made from negative images from another natural scene image by downloading from the web. This contains very different images from the training images, e.g., wall, sea, sky, bricks on the wall, roads, and forests. Negative images are used for the extra output of a fully connected layer of CNN. The negative images are also gathered from the training image of STL-10. 10000 negative images were used for training.

3.5 Predict Natural Scene Image

The prediction step was performed by a feedforward calculation of CNN. First, the target images are given to the CNN input layer. This is propagated through the convolutional and pooling layer sequentially. Fig. 7. gives

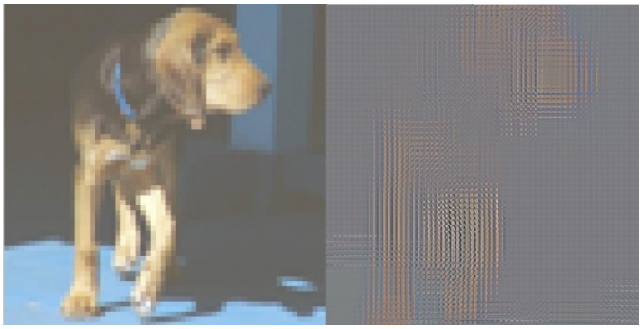

Fig. 7. Transferred image in a convolution layer.

Table1. Matrix Calculation Speed Comparison by C++, Matlab and cuBlas (msec).

	[1,000 x 1,000] x [1,000 x 1,000]	[1,000 x 10,000] x [10,000 x 1,000]
C++	820	7245
Matlab	82	787
cuBlas	40	385

an example of a transferred image. The transferred feature image is very sparse and clustered around the edges of an image. This means that the CNN transfers a pixel based raw image into an edge based sparse image. Therefore, a sparse edge image can make a recognition to robust and reliable. (This is why an edge image is used instead of a raw image in traditional image recognition.)

The important thing in a CNN feature transformation is that it is performed generatively from the data itself without any prior or given arbitrary condition.

At the final pooling layer, all features are aggregated into pooled small dimensional features. Finally, the features are predicted by a softmax regression into the probability of one of the output layers.

3.6 GPU Acceleration

GPU was used to perform the matrix vector calculation of the CNN layer. In particular, cuBlas can be used easily for the matrix dot product. This study tested the speed increase among general C++ calculation, Matlab and GPU. [1,000 x 1,000] x [1,000 x 1,000] matrix and [1,000 x 10,000] x [10,000 x 1000] matrix product are under test. The results are listed in Table 1.

In the CNN calculation, GPU was up to 10 times faster than pure the C++ implementation.

4. Performance Evaluation

CUDA and C++ were used to implement the CNN based object detection application.

MNIST handwritten digits datasets were used for the evaluation. In addition, the STL-10 image dataset was used for the experiments with 2000 training images and 3000 test images. This has 10 classes of categories for classification.

Table 2. Accuracy Comparison of STL-10 and MNIST image set.

Method	STL-10	MNIST
Pretrained CNN	85.2 %	98.5 %
CNN	83.4 %	97.4 %
Feedforward NN	80.1 %	95.6 %

Fig. 8. Example of STL-10 image recognition.

Table 3. Accuracy comparison of the localization performance in a large test image.

Method	128 x 128	256 x 256
Pretrained CNN	87.1 %	86.5 %
CNN	80.7 %	79.3 %
Feedforward NN	75.6 %	74.2 %

Firstly, pretrained CNN was compared with the general CNN, which does not use pretrained weight vector. In addition, the general feedforward neural network was compared. All comparisons were performed in the recall precision measurement. The results are listed in Table 2.

In the result, the pretrained CNN shows the best performance comparable to the general CNN and feedforward neural network. Fig. 8 gives an example of the STL-10 image recognition .

The same procedure was tested on a large image from the web. The images were a 128 x 128 large image with 4 classes of partial images (car, airplane, cat, and dog).

The accuracy measurement method was to calculate the ratio of minimum overlapped region described in (6).

$$e = \frac{1}{n} \cdot \sum_{k} \min_i \min_m \max\left[d(c_i, C_k), f(b_i, B_{km})\right] \quad (6)$$

where is 1 if the result is a true positive, otherwise 0. is 1 if the ground truth region and recognized region are overlapped over 50 %, otherwise 0.

The result shows that the pretrained CNN outperforms the general CNN and feedforward NN in a large image localization.

In Fig. 8, examples of image localization are described.

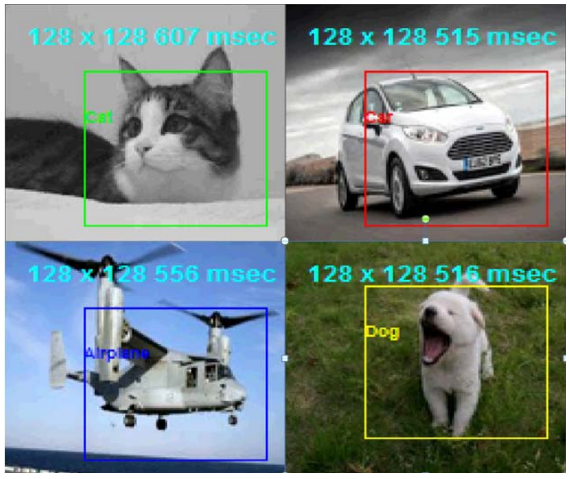

Fig. 9. Example of image localization result.

5. Conclusion

This paper proposed an approach for wild image object detection and localization using a pretrained convolutional neural network. This is composed of a general convolutional neural network and sparse autoencoder. The pretrained model of the CNN outperformed the general CNN in a natural scene image recognition.

The GPU acceleration increases the inspection speed in pretrained CNN compared to the general C^{++} implementation or Matlab vectorization.

Large image localization can be applied using the pretrained CNN. The testing accuracy of pretrained CNN was found to outperform the general CNN and feedforward neural network.

Acknowledgement

This study was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2012002464).

References

[1] David Lowe, “Object recognition from local scale-invariant features,” *in The Proceedings of the Seventh IEEE International Conference*, 1999. Article (CrossRef Link)

[2] Zhenhua Guo, “A Completed Modeling of Local Binary Pattern Operator for Texture Classification,” *in IEEE Transactions on Image Processing*, 2010. Article (CrossRef Link)

[3] Paul Viola, Michael Jones, “Robust real-time face detection,” *in International Journal of Computer Vision*, 2004. Article (CrossRef Link)

[4] David Martin, “Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues,” *in IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2004. Article (CrossRef Link)

[5] Alex Krizhevsky, Ilya Sutskever, Geoffrey Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *in Neural Information Processing Systems*, 2012. Article (CrossRef Link)
[6] Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, Andrew Ng, "Self-taught Learning : Transfer Learning from Unlabeled Data," *in The Proceedings of the 24th international conference on Machine learning*, 2007. Article (CrossRef Link)
[7] Steve Lawrence, "Face recognition: a convolutional neural-network approach," *in IEEE Transactions on Neural Networks*, 1997. Article (CrossRef Link)

Young Shik Moon received his B.S. and M.S. degrees in electronics engineering from Seoul National University and Korea Advanced Institute of Science and Technology (KAIST), Korea, in 1980 and 1982, respectively, and Ph.D. degree in electrical and computer engineering from the University of California at Irvine, CA, in 1990. From 1982 to 1985, he was a researcher at the Electronics and Telecommunications Research Institute, Daejon, Korea. In 1992 he joined the department of Computer Science and Engineering at Hanyang University, Korea, as an Assistant Professor, and is currently a Professor. His research interests include computer vision, image processing, pattern recognition, and computational photography. Dr. Moon served as General Chair of the 2014 IEEE International Symposium on Consumer Electronics, and he is now the President of the Institute of Electronics and Information Engineers, Korea.

Sejin Park received his B.S. and M.S. degree in Computer Science and Engineering from Ajou University, Korea, in 2007. Currently, he is a Ph.D student at the department of Computer Science and Engineering at Hanyang University. His research interests include Computer Vison, Computational Photography, Machine Learning and Deep Learning.