International Journal of Advanced Smart Convergence Vol.5 No3 01-07 (2016)
http://dx.doi.org/10.7236/IJASC.2016.5.3.1

IJASC 16-1-3

Content-Aware Convolutional Neural Network for Object Recognition Task

Alvin Poernomo[1], Dae-Ki Kang[2]

[1]Department of Ubiquitous IT, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan 47011, Rep. of Korea
email: p[underscore]90alvin@yahoo.co.id
[2]Department of Computer & Information Engineering, Dongseo University, 47 Jurye-ro, Sasang-gu, Busan 47011, Republic of Korea
email: dkkang@dongseo.ac.kr

Abstract

In existing Convolutional Neural Network (CNNs) for object recognition task, there are only few efforts known to reduce the noises from the images. Both convolution and pooling layers perform the features extraction without considering the noises of the input image, treating all pixels equally important. In computer vision field, there has been a study to weight a pixel importance. Seam carving resizes an image by sacrificing the least important pixels, leaving only the most important ones. We propose a new way to combine seam carving approach with current existing CNN model for object recognition task. We attempt to remove the noises or the "unimportant" pixels in the image before doing convolution and pooling, in order to get better feature representatives. Our model shows promising result with CIFAR-10 dataset.

Keywords: *deep learning, convolutional neural network, seam carving, image resizing*

1. Introduction

In recent years, convolutional neural network (CNN) has shown remarkable results in image, video, and speech recognition. These can be achieved through the availability of large public image datasets on the Internet and also the recent advancement of GPU computing. In particular, image classification task has gained massive interests including the acknowledged ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [1,2]. Most researchers aim to come up with better recognition system by designing more complex and deeper network and trying to apply them to different large datasets.

The most important part of CNN is the convolution layers and usually is followed by subsampling/ pooling layers. The convolution layers are used to extract feature maps from the input by convolving local input regions with different filters. Subsampling/ pooling layers are basically a dimension reduction technique applied to the local regions of the feature maps, resulting a summarized response of the feature maps. Pooling segments each of the feature maps into several identical regions with a fixed stride and a pooling kernel size, then resizes each region independently. It is very common to insert pooling layer before doing next convolution in CNN architecture, so that the network can learn higher level features from the previous summarized input. By stacking multiple layers of convolution and subsampling layers, we get a hierarchy of features starting from low level features at early layers to high level features at latter layers. In object recognition task, we consider the object itself as the pixels with important values and the rest (the background) are less important.

Manuscript Received: July 04, 2016 / Revised: July 15, 2016 / Accepted: July 30, 2016
Corresponding Author: dkkang@dongseo.ac.kr
Tel: +82-51-320-1724, Fax: +82-51-327-8955
Department of Computer & Information Engineering, Dongseo University
47 Jurye-Ro, Sasang-Gu, Busan 47011, South Korea

The problem with current CNN architecture is that the network considers all the pixels are equally important. Furthermore, in pooling layers, the input images are segmented equally into several local regions with identical size and then subsampling is performed independently over all these regions without considering which regions contain more important pixels than the others. This will cause the network loses the more important pixels while in the same time maintaining the less important ones, because they are treated equally. For example, if the object is centered at the top right of the image, after doing several pooling operations, the result is the same images with only different aspect ratios. The object will still be centered on the top right of the image, because the subsampling operations are done independently over the equal-size of local regions.

In this paper, we consider how to maintain the important pixels inside the network architecture. By eliminating the less important pixels first instead of equally subsampled all the pixels, the network can prevent huge loss of important pixels and in the same time, eliminate the less important pixels of the input image. We propose to combine seam carving algorithm with CNN architecture. Seam carving is a computer vision approach to resize an image while maintaining the important object/pixels inside. This can be done by removing the least noticeable pixels in the image [5]. Assume that a pixel is similar to the surrounding pixels, then removing it may be unnoticed. This approach is considered a breakthrough in computer vision, in particularly image processing system. Compared to regular scaling or cropping method, this approach can maintain the objects inside an image better without losing too many important pixels. Applying this approach to CNN architecture seems logical since the network should generate better feature representatives from a less-noise image.

2. Theory

2.1 Convolutional Neural Network

Convolutional neural network is an extension of feed-forward artificial neural network, purposely designed to use minimal amounts of preprocessing. It was inspired by biological visual cortex which consists of a hierarchy of simple, complex, and hyper-complex cells [3]. Each neuron in those cells are arranged to respond only to local overlapping sub-regions across the visual field which are called receptive fields. This causes the information are processed in such a way that creates a feature hierarchy from low level features, like responses of a specific edge-like pattern to higher level features. This concept was later adapted in Neocognitron [4], which is known as a predecessor to CNN today. It was first applied to handwritten digit recognition and developed further to zip code and face recognition.

CNN takes this concept into further advancement that makes it extremely useful for image and speech recognition. CNN exploits spatially local correlation by enforcing a local connectivity pattern between neurons of adjacent layers, which is derived from the same concept in Neocognitron. Each filter, which is the same size as the receptive field, is replicated across the entire visual field and these replicated units share the same weight, allowing the features to be detected regardless of their positions in the visual field and also making the network more efficient as the numbers of parameters being learnt are greatly reduced. Spatial pooling which is a form of non-linear down-sampling, also plays an important part in CNN as it also greatly reduce the computation for upper layers and reduces the sensitivity of the output to input translations and slight distortions. Typical pooling operator is max pooling, which returns only the greatest response in each local regions.

A typical CNN architecture consists of multiple stages of convolution and pooling layers and ends with fully connected layers and softmax for the classifier. Each convolution filter generates one feature map from the input image and each pooling function summarizes each feature map into smaller dimension. A non-linear activation, such as RELU or hyperbolic tangent, usually follows after convolution/ pooling layer. One notable example of CNN model among the many that have been proposed for computer vision and image processing problems is LeNet CNN [6]. In fact, we use modified version of this model in our experiment.

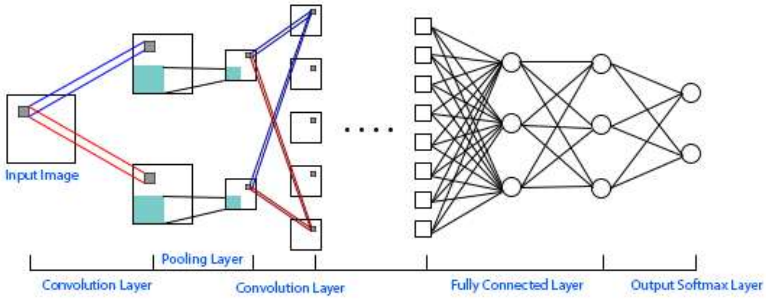

Figure 1. CNN model with several convolution and pooling layers performed alternately, combined with fully connected layers and softmax layer as the classifier.

2.2 Seam carving

Seam carving is an algorithm for content-aware image resizing [5], introduced by Shai Avidan and Ariel Shamir. Unlike other regular image resizing technique, seam carving is able to resize image while maintaining its important features intact. It is accomplished by establishing a number of seams, which are paths of least important pixels, in an image and removes them gradually until desired output size is obtained. Seams can be generated in 2 ways, depends on which aspect ratio we want to change. A vertical seam is a path of pixels connected from top to bottom with one pixel in each row of an image. A horizontal seam is a path of pixels connected from left to right with one pixel in each column. For example, consider we want to reduce an image's width by one pixel, then we generate a vertical seam with one pixel in each row. Removing that seam will make the image one pixel narrower. All the pixels of the image are then shifted left to compensate for the missing path. This processes is repeatable if there are more than one seams.

To generate a seam, first we need to compute the energy function that will map each pixel into energy value. There are several way to compute the energy function. One way is to use gradient of the pixel and sum the energy value for all channels.

$$e_1(I) = \left|\frac{\partial}{\partial x} I\right| + \left|\frac{\partial}{\partial y} I\right| \tag{1}$$

Given an energy function e, we define the cost of a seam/ path as:

$$E(s) = E(I_s) = \sum\nolimits_{i=1}^{n} e(I(s_i)) \tag{2}$$

From equation (), we want to find the optimal seam s* that minimizes this seam cost :

$$s^* = \min_s E(s) = \min_s \sum\nolimits_{i=1}^{n} e(I(s_i)) \tag{3}$$

To generate this optimal seam/ path with minimum energy value, one preferable way is using dynamic programming approach. Assume that we want to generate a vertical seam from an image. First step is to map all the pixel value of the image into energy value using equation (1). Second step is to compute the cumulative minimum energy M for all possible connected seams for each entry (i, j), where i indicates row and j indicates column :

$$M(i,j) = e(i,j) + \min(M(i-1,j-1), M(i-1,j), M(i-1,j+1)) \tag{4}$$

At the end of this process, the minimum value M of the last row indicates the end of the minimal connected vertical seam. Thus, we choose this value and backtrack/ traverse back to the first row, choosing connected pixels with minimum energy M to find the path of the optimal seam.

2.3 Context-aware CNN with Seam Carving Layer

In this paper, we propose to combine seam carving algorithm and CNN to generate better object feature representation while maintaining only the important pixels of the image. Seam carving layers are inserted within the network in order to preserve only the important pixels from the corresponding feature maps. The fact that CNN learns higher feature representatives from lower ones befits with our approach since we are applying seam carving layers consecutively and aiming to retarget the feature maps in each convolution layers. By preserving the important pixels and at the same time, removing the unimportant ones successively at every layer causes the network to be more compact and resistant to translation invariance. Consider that we have several input images where the positions of object of each image are randomly distributed. After doing convolution with seam carving at every layer, the general position of all images will be similar to each other.

By combining seam carving approach and CNN architecture, we achieve our model as shown in Figure 2. Our model utilizes all components in regular CNN with seam carving layers as addition to improve the model effectiveness for object recognition task. However, choosing the amount of pixels to be removed is not a trivial matter. Removing too many pixels can lead to distorted object, causing network performance degradation. One normal approach is using validation set to obtain the best parameter values for the network. On the other hand, a well choice of seam carving value can lead to better feature representation and more efficient network as most of all noises are removed. Our model can achieve a comparable result with CIFAR-10 dataset.

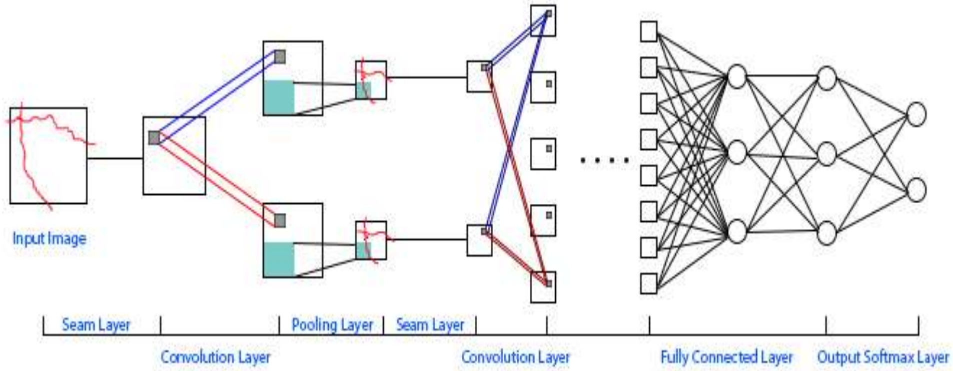

Figure 2. Modified version of LeNet model used for out experiments. We insert a seam carving layer before each convolution layer. The red seams in input image indicate the "unimportant" pixels in the image that are removed before

2.4 Seam Carving Layer Backpropagation

Calculating the error backpropagation at seam carving layer is similar to max-pooling layer. In max pooling, we take the a maximum value over several values in a region and omit it as the output value. In backpropagation phase, the error retrieved from next layer is propagated back to its original position. For example, we have value {3,4,5,6} in region A. After doing max pooling, we take value 6 as our output for region A. This value will later receive an error value from the next layer in backpropagation process. Afterwards, this error value will be propagated back to its original position in region A while the other values receive nothing. Based on the same concept, the error propagation in seam carving layer is implemented in the same manner. After removing the desired pixels from the input feature map, we maintain the original positions of each remaining pixels. By doing so, we can easily propagate each of the received errors back to their original positions we maintained before. Figure 3 shows the similarity between max-pooling backpropagation process and our approach.

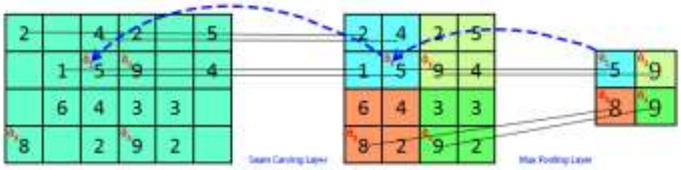

Figure 3. Backpropagation process of seam carving layer and max pooling layer. The similarity between them is they have no learned parameters and only passing the error value (delta) to the original positions of seam carving or max pooling.

3. Experiment

In our work , we use CIFAR-10 benchmark dataset. The dataset consists of 50,000 training and 10,000 test image respectively with total of 10 image categories. The 10 categories belong to the following objects: airplane, automobile, bird, cat, deer, dog, frog, horse, ship, and truck. Each image is 32 x 32 pixels with 3 color channels. We used MatConvnet [8], a Matlab toolbox designed for CNN implementation. We use default LeNet configuration in MatConvnet with slightly modification by inserting a seam carving layer before convolution layer as shown in Figure. All input images are preprocessed using contrast normalization as suggested in [7] and ZCA whitening. The original CNN layers are listed below :

1. Convolution layer : 5x5 pixels filter size, stride 1, padding 2
2. Max pooling layer : 3x3 pixels filter size, stride 2, padding 1
3. RELU layer
4. Convolution layer : 5x5 pixels filter size, stride 1, padding 2
5. RELU layer
6. Average pooling layer : 3x3 pixels filter size, stride 2, padding 1
7. Convolution layer : 5x5 pixels filter size, stride 1, padding 2
8. RELU layer
9. Average pooling layer : 3x3 pixels filter size, stride 2, padding 1
10. Convolution layer : 3x3 pixels filter size, stride 1, padding 0
11. RELU layer
12. Fully connected layer
13. Softmaxloss layer

To prevent over fitting, we use dropout 0.5 in the fully connected layer. The other hyper parameters we use are weight decay 0.0001, learning rate 0.05 for the first 30 epochs and 0.005 for the next 10 epochs with total of 40 epochs. We use mini batch gradient descent with 100 instances each batch.

4.Result and Discussion

4.1 Experiment result

We explore 2 model configurations and compare them to the original LeNet configuration.

Table 1. Model description

Model	Description
A	Original LeNet configuration
B	Seam carving layer before first convolution layer (1 pixel)
C	Seam carving layer before first and second convolution layer (2 pixels and 1 pixel respectively)

For our first model (B), we use 1 pixel-seam carving layer at first convolution layer. It means that if the input image is 32x32 pixels, then seam carving layer removes 1 pixel of its width and height dimension, resulting

31x31 pixels of its output. For our second model (C), we use 2 pixel-seam carving layer at the first convolution layer and 1 pixel-seam carving layer at the second convolution layer. We compared both of the accuracy performances to the original LeNet configuration (A). All of the models use the same hyper parameters as listed in the Experiment Section. The error rate of each model is given in Table 2.

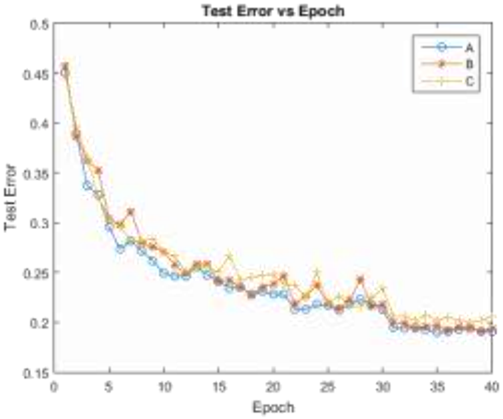

Figure 4. Error rate vs. epoch on Cifar-10 dataset

Table 2. Error rate on test set

Model	Top 1 error	Top 5 error
A	0.191	0.009
B	0.191	0.009
C	0.199	0.012

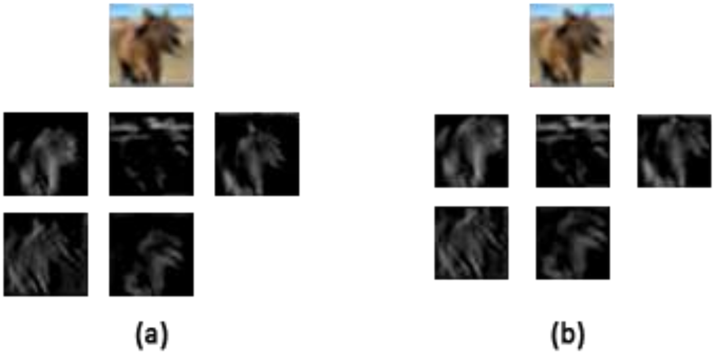

Figure 5. (a) Features extracted from original image without seam carving. (b) Features extracted from image which has been resized by seam carving layer.

4.2 Discussion

From Figure 4 and Table 2 , we can see that our approach shows promising result. Although the accuracy of our approach is not very remarkable, but we prove that some pixels in the image are not important or relevant to object classification. One possible reason why we can’t get better accuracy performance on cifar-10 dataset is because the input image size is very small and most of the objects are already centered in the middle. Hence, there are not many unimportant pixels to be removed. Our model B shows the same accuracy performance with the original LeNet model, with only one pixel removed from the input dimension (height and width). Model C shows lower accuracy performance with 2 times seam carving processes, 2 pixels dimension at the first convolution layer and 1 pixel dimension at the second convolution layer respectively. The limitation of this model is we need to find the optimal number of pixel dimension that should be removed. Too many pixel reductions can cause distorted objects and network performance degradation. Therefore, further study about this problem is needed.

5. Conclusion

In this work, we proposed a way to recognize object from image using Convolutional Neural Network while maintaining only the important pixels using Seam Carving in order to get better feature representatives. The initial result with cifar-10 dataset showed that our work was worth exploring. Although we can only achieve accuracy performance as high as regular LeNet model, it is still promising since we are able to get same result by reducing the initial image width and height dimension by one pixel. We believe that our approach can be useful for more complex object recognition task in larger images. We will try different datasets in our future work.

6. Acknowledgement

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (no. NRF-2015R1D1A1A01061328).

References

[1] Krizhevsky, Alex, Ilya Sutskever, and Geoffrey E. Hinton, "Imagenet classification with deep convolutional neural networks," *Advances in neural information processing systems*, 2012.

[2] Deng, Jia, et al, "Imagenet: A large-scale hierarchical image database," *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009.

[3] Hubel, David H., and Torsten N. Wiesel, "Receptive fields and functional architecture of monkey striate cortex," *The Journal of physiology* 195.1 (1968): 215-243.

[4] Fukushima, Kunihiko, "Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position," *Biological cybernetics* 36.4 (1980): 193-202.

[5] Avidan, Shai, and Ariel Shamir, "Seam carving for content-aware image resizing," *ACM Transactions on graphics (TOG)*. Vol. 26. No. 3. ACM, 2007.

[6] LeCun, Yann, et al. "Gradient-based learning applied to document recognition," *Proceedings of the IEEE* 86.11 (1998): 2278-2324.

[7] Coates, Adam, Honglak Lee, and Andrew Y. Ng, "An analysis of single-layer networks in unsupervised feature learning," *Ann Arbor* 1001.48109 (2010): 2.

[8] Vedaldi, Andrea, and Karel Lenc, "Matconvnet: Convolutional neural networks for matlab," *Proceedings of the 23rd ACM international conference on Multimedia*. ACM, 2015.