AdGAP: Advanced Global Average Pooling

Arna Ghosh
McGill University
Montreal, QC H3A 0G4. Canada.
Email: arna.ghosh@mail.mcgill.ca

Biswarup Bhattacharya *
University of Southern California
Los Angeles, CA 90089. USA.
Email: bbhattac@usc.edu

Somnath Basu Roy Chowdhury *
Indian Institute of Technology
Kharagpur, WB 721302. India.
Email: brcsomnath@ee.iitkgp.ernet.in

Abstract

Global average pooling (GAP) has been used previously to generate class activation maps. The motivation behind AdGAP comes from the fact that the convolutional filters possess position information of the essential features and hence, combination of the feature maps could help us locate the class instances in an image. Our novel architecture generates promising results and unlike previous methods, the architecture is not sensitive to the size of the input image, thus promising wider application.

Introduction

The current advances in computer vision and image recognition have seen a wide use of convolutional neural networks (CNNs). Works by (Zhou et al. 2016) have shown that the convolutional units of various layers of CNNs are capable of detecting the position of object in the image without any supervision on the location of object. They show that although earlier convolutional layers are capable of capturing only the low-level features in the image, higher layers are able to capture task-specific features. However, the information about the location of these features in the image is lost when fully connected layers are used for classification. Some fully convolutional networks have been proposed recently such as GoogleNet (Szegedy et al. 2015), and Network in Network (Lin, Chen, and Yan 2013) that aim to reduce the number of network parameters while maintaining performance by avoiding the fully-connected layers.

In the work of (Lin, Chen, and Yan 2013), the *Global Average Pooling* layers act as structural regularizer and prevent overfitting. However, (Zhou et al. 2016) shows that the average pooling layers can be used to retain the localization ability of the final layers of the network. This concept is also used in (Bolanos and Radeva 2016) for localization and recognition of food items. However, the use of Global Average Pooling layers in these works requires differential connection during training and deployment.

Looking into the biology of our nervous system, we know that this is not the case for connections in the brain. Therefore, we looked into methods to bridge this gap and came up with an architecture where we do not need to alter the connections of a network. The major motivation of our work comes from the fact that different high level features correspond to different classes and the classification of an image to a particular class depends on certain features being excited more heavily than others.

Therefore, we present a novel architecture that can be used to localize the positions of instances of all classes in an image and then classify each of those hotspots. Unlike previous methods, the architecture is not sensitive to the size of the input image and hence has a wider scope of application. The experiments are done on the MNIST dataset and further discussions are presented from the point of view of text extraction from natural images as a proof of concept, but can be easily extended to other datasets and domains.

Architecture

We begin with a simple CNN architecture, although the method proposed is easily scalable to any CNN. The network is trained on the MNIST dataset to classify images into 10 classes (digits in this case). Following the training of the network to obtain a reasonably good classification accuracy, we try to figure out the *importance* of each filter in the last layer for classifying the digits.

Since the final layer filters have the information of the spatial location of specific features as well, a linear combination of those filters could provide a heatmap of features in the image that are required to be classified into one of the 10 classes. We use the error in classification in absence of each filter as a metric of the *importance* of that filter, unlike training another network as in (Zhou et al. 2016) and (Bolanos and Radeva 2016). We record the error of classification in presence of all the filters as the baseline. The weights for the generation of heatmap using a linear combination of these filter responses is the difference of error from the baseline. Therefore, if $\mathbf{W}$ represents the weight vector for heatmap generation, the overall classification error (error with all filters present) is e, and $\mathbf{E}$ is the error vector, where E_i represents the error in classification when filter i is missing (or response from filter i is blocked from propagating into further layers), then

$$W_i = e - E_i \tag{1}$$

Suppose the filter response of the final layer filters is represented as $\mathbf{F}$ where $\mathbf{F}_i$ represents the filter response of the

*Equal contribution

Layer	Type	Maps and Neurons	Filter Size
0	Input	1M × 28 × 28N	-
1	Conv	6M × 28 × 28N	5×5
2	MaxPool	6M × 14 × 14N	2×2
3	Conv	16M × 14 × 14N	5×5
4	MaxPool	16M × 7 × 7N	2×2
5	FullyConn	120N	1×1
6	FullyConn	84N	1×1
7	FullyConn	10N	1×1

Table 1: Network architecture used for digit classification

i^{th} filter. Then the heatmap corresponding to the position of the class instances in image, represented by **H**, is given by a weighted average of the filter responses.

$$\mathbf{H} = \sum_i W_i \times \mathbf{F}_i \tag{2}$$

Table 1 depicts the layers of the network used.

Experiments

Setup

We use the MNIST handwritten digits dataset for experimentation. We train the network on the MNIST images. The training set has 60000 images and the test set has 10000 images. The network is then used to identify the position of class instances (here digits) on natural images. The network is trained using Adam Optimizer with a learning rate of 0.001 and learning rate decay of 0.00001. The batch size is kept at 300 images and the network is trained for 10 iterations (200 batches per epoch). The learning curve is shown in Figure 1. The results show that the network is able to segment the text from the images.

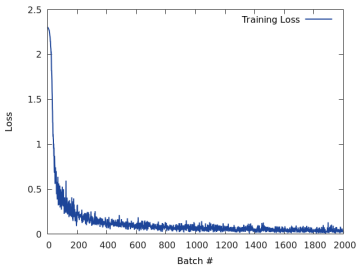
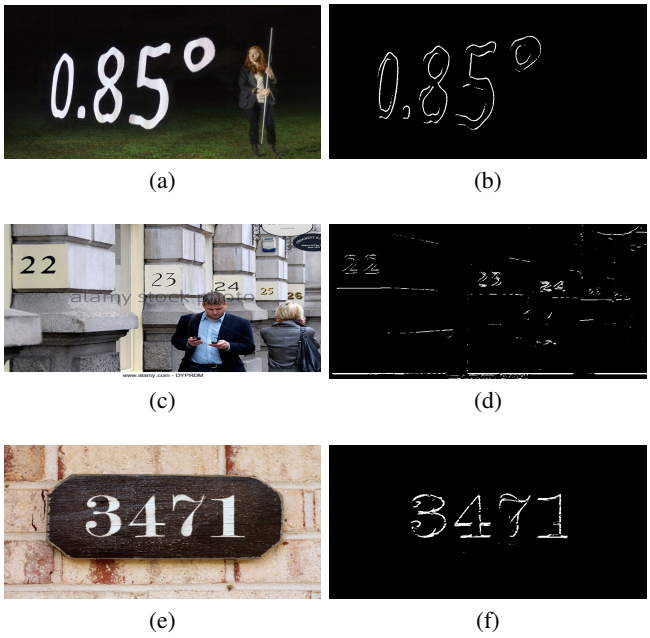


Figure 1: Training curve showing the training loss at each batch of forward pass (Code[1])

Results & Illustrations

The results are shown in Figure 2. The network is able to extract text and numbers from the images, due to their similarity in features. We use a high threshold value on the heatmap generated from the linear combination of filters to show the extraction property of the network. Although the network is trained on numbers, the network is capable of extracting text as well. This is because the features required to classify numbers and characters are very similar.

[1]https://github.com/brcsomnath/Advanced-GAP



(a) (b) (c) (d) (e) (f)

Figure 2: Results

Conclusion

The network architecture proposed can clearly be used for localization and recognition of trained class instances in an image. The results are presented on different size images to illustrate the versatility of the network. Further experiments would entail deployment on more complex datasets, using deeper networks, and possibly checking for class-specific or category-specific heatmaps to locate instances of objects belonging to similar "categories", i.e. objects that respond to similar features.

Acknowledgment

We would like to thank Dr. Debdoot Sheet, Assistant Professor, Department of Electrical Engineering at IIT Kharagpur for motivating discussions regarding the work.

References

Bolanos, M., and Radeva, P. 2016. Simultaneous food localization and recognition. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, 3140–3145. IEEE.

Lin, M.; Chen, Q.; and Yan, S. 2013. Network in network. *arXiv preprint arXiv:1312.4400*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on*, 2921–2929. IEEE.