AdGAP: Advanced Global Average Pooling

Arna Ghosh, Biswarup Bhattacharya* & Somnath Basu Roy Chowdhury*
McGill University (Canada), University of Southern California (USA), Indian Institute of Technology Kharagpur (India)

Abstract

The current advances in computer vision and image recognition have seen a wide use of convolutional neural networks (CNNs). 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. Our algorithm hopes to provide a method to understand what features in the data guide the prediction of the convolutional neural network. Our implementation of the algorithm illustrates a good performance with MNIST datasets and localizes the positions of digits in some natural images.

Introduction

It has been demonstrated that although earlier convolutional layers are capable of capturing only the low-level features in the image, later layers are able to capture task-specific features. Prior work in [2] and [3] propose to use fully convolutional networks to exploit the positional information to these features to guide classification. It has been shown in [4] that the linear combination of the final convolutional layer outputs of CNNs are capable of detecting the position of object in the image without any supervision on the location of object. However, the use of another network to estimate the coefficients for linear combination demands a differential connection within the network. We propose to use the error induced by removing each filter map as an indicator to the coefficients.

Our goal: To develop a framework for generating class activation maps to locate the position of class instances in images using the information captured by the final layer filters.

Key Idea: Using the error induced by removal of each final layer filter as the coefficient of linear combination for generation of activation maps. This way, we aim to avoid differential connection for training and testing and hope to establish a more biologically-plausible algorithm.

Architecture

We present the architecture with a small CNN although the method is scalable to complex architectures as well:

- **Training the network**: The network is initially trained on the dataset with class labels to obtain a reasonably good classification accuracy.
- **Filter Map Importance:** 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. 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 [4] and [1]. If **W** represents the weight vector for heatmap generation, the overall classification error (error with all filters present) is e, and **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).

$$W_i = e - E_i$$

- **Generating Heatmap for images**: If the filter response of the final layer filters is represented as **F** where F_i represents the filter response of the i^{th} filter, 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.

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

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

Experiments

We use the **MNIST** handwritten digits dataset for our experiments. The training set has 60000 images and the test set has 10000 images. The network is trained using **Adam Optimizer** with a learning rate of 0.001 and learning rate decay of 0.00001. Once the network converges to an acceptable error value, the AdGAP procedure is applied to obtain filter map importances. Following this, the network is used to identify the position of class instances (here digits) on natural images.

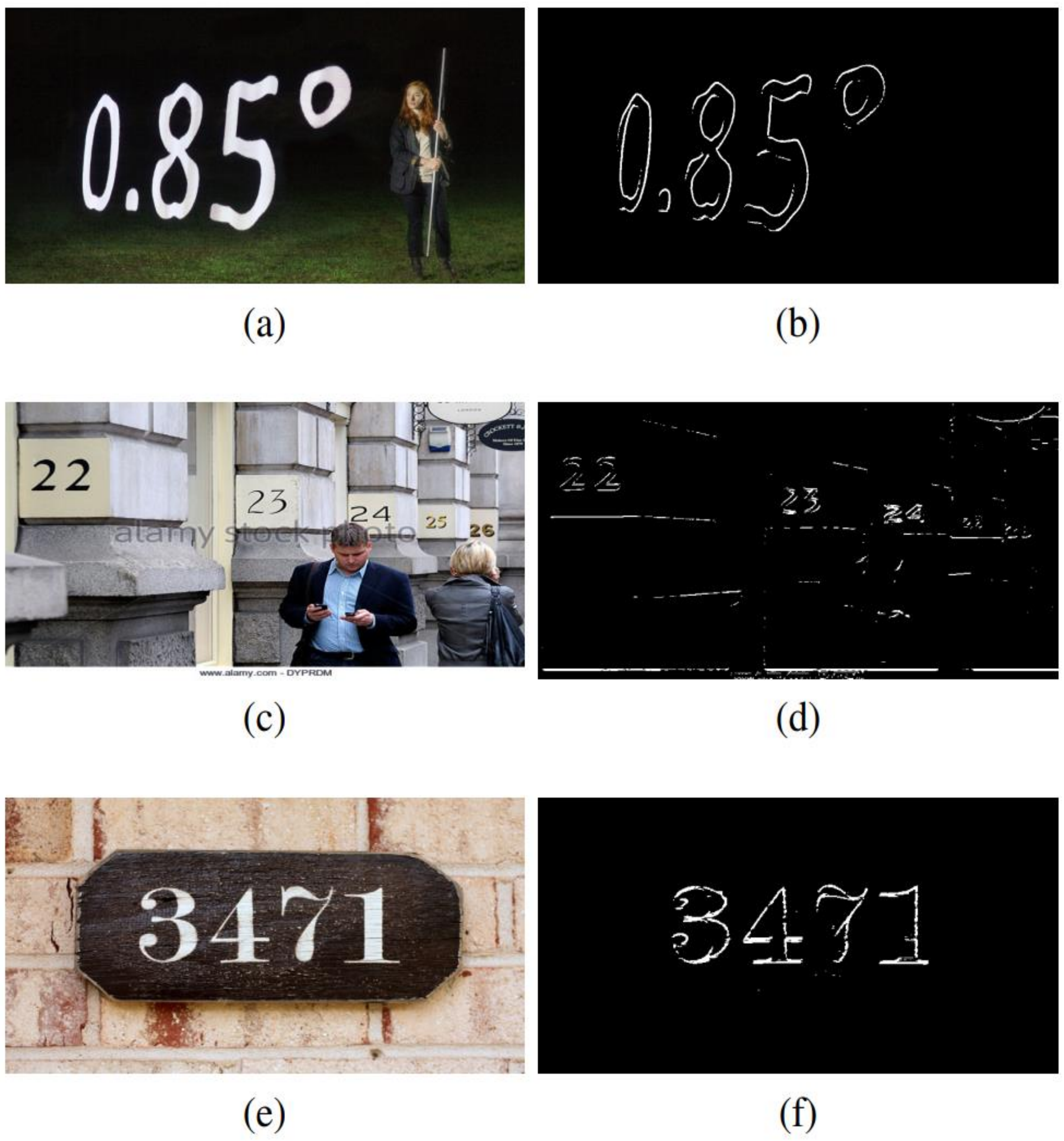
(a) (b)
(c) (d)
(e) (f)

Figure 1: Heatmap generated (right) for corresponding input images (left)

Further experiments would include training a network on the ILSVRC dataset and observe the performance on the object localization task.

References

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

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

[3] 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.

[4] 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

* Equal contribution

Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18), New Orleans, LA, USA