Weakly Supervised Object Localization via Sensitivity Analysis

Mohammad K. Ebrahimpour
EECS Department
University of California, Merced
mebrahimpour@ucmerced.edu

David C. Noelle
EECS Department
University of California, Merced
dnoelle@ucmerced.edu

Abstract

Deep Convolutional Neural Networks (CNNs) have been repeatedly shown to perform well on image classification tasks, successfully recognizing a broad array of objects when given sufficient training data. Methods for object localization, however, are still in need of substantial improvement. In this paper, we discuss a fast and simple approach to the localization of recognized objects in images. Our method is predicated on the idea that a deep CNN capable of recognizing an object must implicitly contain knowledge about object location in its connection weights. We provide an easy method to interpret classifier weights in the context of individual classified images. This method involves the calculation of the derivative of output class label activation with regard to each input pixel, performing a sensitivity analysis that identifies the pixels that, in a local sense, have the greatest influence on object recognition. Our experimental results, using real-world data sets for which ground-truth localization information is known, reveal competitive accuracy from our extremely fast technique.

1. Introduction

Deep Convolutional Neural Networks (CNNs) have been shown to be effective at image classification, accurately performing object recognition even with thousands of object classes when trained on a sufficiently rich data set of labeled images [4]. One advantage of CNNs is their ability to learn complete functional mappings from image pixels to object categories, without any need for the extraction of hand-engineered image features [7]. To facilitate learning through stochastic gradient descent, CNNs are (at least approximately) differentiable with regard to connection weight parameters.

In this paper, we focus on object localization, identifying the position in the image of a recognized object. As is common in the localization literature, position information is output in the form of a bounding box. Previously developed techniques for accomplishing this task generally involve searching the image for the object, considering many candidate bounding boxes with different sizes and locations, sometimes guided by an auxilliary algorithm for heuristically identifying regions of interest [7, 3]. For each candidate location, the sub-image captured by the bounding box is classified for object category, with the final output bounding box either being the specific candidate region classified as the target object with the highlest level of certainty or some heuristic combination of neighboring or overlapping candidate regions with high classification certainty. These approaches tend to be time consuming, often requiring deep CNN classification calculations of many candidate regions at multiple scales. Efforts to speed these methods mostly focus on reducing the number of regions considered, typically by using some adjunct heuristic region proposal algorithm [5]. Still, the number of considered regions is often reported to be roughly 2,000 per image. While these approaches can be fairly accurate, their slowness limits their usefulness, particularly for online applications.

Table 1. Comparative Performance on ImageNet (478 Classes)

Method	Average CorLoc
Constant Center Box Baseline	12.34%
Top Objectiveness Box	37.42%
Co-Localization	53.20%
Sensitivity Maps	**82.76%**

In this paper, we discuss an approach to object localization that is both very fast and robust in the face of limited ground-truth bounding box training data. This approach is rooted in the assertion that any deep CNN for image classification must contain, implicit in its connection weights, knowledge about the location of recognized objects. Thus, this approach aims to leverage location knowledge that is already latent in trained image classification networks, without requiring a separate learning process for localization.

2. Method

Calculating derivatives of a function of network output with regard to network parameters, such as connection

weights, is a standard part of CNN training. It is common for learning in a deep CNN to involve stochastic gradient decent, which involves such derivatives. In image classification networks, the objective function is designed to have optima where training images are correctly classified. If we now see $G(x_i; w^*)$ as the output of such an image classification network, its gradient with regard to x_i would provide information about the sensitivity of the assigned category to individual pixels. Pixels with the largest absolute values of this derivative will, around the input x_i, produce the largest changes in the classification decision of the CNN if those pixels are changed. This can be seen as one measure of how important specific pixels are for classifying the object in the image. The calculation of this gradient can be performed as efficiently as a single "backward pass" through the classification network, producing a *sensitivity map* of the image. This is well illustrated by considering the case of a simple layered backpropagation network [6] in which the "net input" of unit i, η_i, is a weighted sum of the activations of units in the previous layer, and the activation of unit i is $g(\eta_i)$, where $g(\cdot)$ is the unit activation function. In this case, we can define a sensitivity value for each unit, s_i, as the derivative of the network output with regard to η_i. Using the chain rule of calculus, it is easy to show that the sensitivity of an output unit is $g'(\eta_i)$, and, for units in earlier layers ...

$$s_i \;=\; g'(\eta_i) \sum_k w_{ki}\, s_k \qquad (1)$$

... where k iterates over all units in the immediately downstream layer from unit i and w_{ki} is the connection weight from unit i to unit k. This calculation may be performed, layer by layer, from outputs to inputs, until s_i values for each input pixel are available. This demonstrates how efficiently a sensitivity map can be calculated for a given classified image. In the evaluation of our approach in Section 3, we report results using the gradient computation tools provided by TensorFlow. [1].

Object localization algorithms typically output the four coordinates of a bounding box to communicate the location of the target object. Such a bounding box is not intrinsic to a sensitivity map, however. Heuristic techniques could be used to identify a rectangular region that captures the majority of the high sensitivity pixels, while avoiding low sensitivity pixels, but we have taken a different approach. We have opted to learn a linear mapping from sensitivity maps to bounding box coordinates, using training images with ground truth location information.

3. Results

We evaluated our proposed method for object localization on the ImageNet 2012 data set [2]. The ImageNet data set is one of the largest publicly available data sets. It also contains many images annotated with ground truth object location bounding boxes. We conducted a large scale evaluation of our approach by using all images in ImageNet that are annotated with ground truth localization information. This subset contains 300,916 images involving 478 object classes. We divided this data set into a training set, a test set, and a validation set by sampling without replacement (i.e., the intersection between each pair of the three sets was empty). There were 225,687 images (75%) in the training set, and there were 45,137 images in each of the other two sets. We compared the performance of our approach with two methods discussed in Tang et al. [8] for which ImageNet results are explicitly reported: Top Objectiveness Box & Co-Localization. Also, we noted that many images in this data set presented the target object in the middle of the image, providing a bias that could be leveraged by learned localization systems. Thus, as a baseline of performance, we calculated the localization performance for a system that blindly offered the same bounding box in the middle of the image, with average size, for every input. Our results are shown in Table 1, where performance is reported in terms of the CorLoc statistic: the percentage of test images for which the area of intersection between the predicted bounding box and the ground-truth box is at least half the size of the area of the union of the two boxes. Note the relatively strong performance of our highly efficient method. Also note that the observed baseline performance was comfortingly low.

Some example localization predictions are illustrated in Figure 1. As might be expected, performance varies with class. Our algorithm appears to do well on some objects, such as balls and dogs. One might suspect that failures arise in the linear mapping from sensitivity maps to bounding box coordinates, but a perusal of the sensitivity maps, themselves, suggests that the pixel sensitivity values vary in utility across different object categories.

The proposed approach is quite general. Indeed, we are currently working on applying sensitivity analysis to deep networks trained on other tasks.

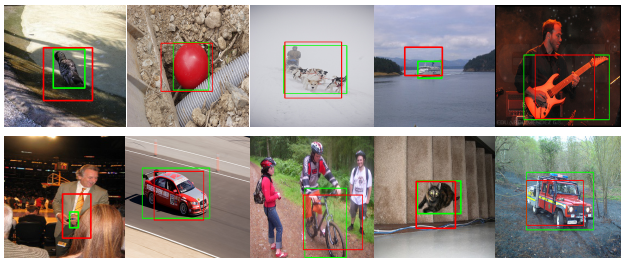

Figure 1. Example of some of some of the bounding box predictions of the proposed method on ten different categories. The green boxes are the ground truth locations, and the red ones are the predicted bounding boxes.

References

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org. 2

[2] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 248–255. IEEE, 2009. 2

[3] R. Girshick. Fast R-CNN. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1440–1448, 2015. 1

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, pages 1097–1105, 2012. 1

[5] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems*, pages 91–99, 2015. 1

[6] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323:533–536, 1986. 2

[7] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. OverFeat: Integrated recognition, localization and detection using convolutional networks. *arXiv:1312.6229*, 2013. 1

[8] K. Tang, A. Joulin, L.-J. Li, and L. Fei-Fei. Co-localization in real-world images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1464–1471, 2014. 2