HIGH-RESOLUTION CLASS ACTIVATION MAPPING

Thanos Tagaris[1]*, Maria Sdraka and Andreas Stafylopatis*

School of Electrical and Computer Engineering
National Technical University of Athens, Greece
[1]thanos@islab.ntua.gr

ABSTRACT

Insufficient reasoning for their predictions has for long been a major drawback of neural networks and has proved to be a major obstacle for their adoption by several fields of application. This paper presents a framework for discriminative localization, which helps shed some light into the decision-making of Convolutional Neural Networks (CNN). Our framework generates robust, refined and high-quality Class Activation Maps, without impacting the CNN's performance.

Index Terms— Discriminative localization, Class Activation Map, Deep Learning, Convolutional Neural Networks

1. INTRODUCTION

Since the introduction of the first effective Convolutional Neural Network (CNN) for image classification (i.e. AlexNet) [1], their popularity has increased dramatically. Having dwarfed the performance of their predecessors on virtually every major competition [2, 3], they have opened the doors for a plethora of new fields of research in Computer Vision [4, 5] and have been almost universally adopted for any image-related application [6, 7, 8, 9].

Part of their success can be attributed to the fact that, contrary to their predecessors, they don't require any form of feature engineering beforehand; instead they attempt to extract the most important features for the task at hand. These features are tailored for classification and have proven to be almost mandatory for any network hoping of achieving high performance. One major drawback, however, is that networks trained on the features lose all their interpretability. There is no way to delve into the inner workings of a CNN and identify what it looks for in an image.

In this sense, Neural Networks are viewed as *black boxes* [10]; they can achieve a high performance but won't provide any reasoning for their prediction. Some basic questions can't be answered: For example, given a multiclass classification problem, there is no means of knowing why it performs better on class A than class B or what did it "see" to classify an image to class A. Due to this, CNNs have had a relatively slow adoption rate in some sensitive fields, such as medical imaging [11].

There have been some efforts to shed some light on the decision-making of CNNs. Zhou et al. [12] proposed a methodology for discriminative localization with CNNs trained for classification, simply with the addition of a Global Average Pooling (GAP) layer before the Fully Connected layer. This grants ability to generate the so-called *Class Activation Maps* (CAMs) from a network, which indicate what part of an image activates the network for a specific class. This idea will be presented in more detail in Section 2. While this methodology can help answer the questions posed above and provide insight in the CNN's decision-making process, it produces very crude maps which can be only used for providing a general idea of what the CNN is "looking at".

Another downside of this methodology is that in most architectures, including the one used by the authors, the addition of the GAP layer hurts the performance of the network for classification. For commercialized applications this can be an harsh trade-off. By gaining the ability of providing a reasoning for their decisions, the network's performance deteriorates significantly. Zhou et al. report a 37.1% top-5 error rate on ILSVRC 2014, a score which would be considered unsatisfactory even during the pre-deep learning era [13]. As a reference the current state-of-the-art lies at 2.251% [3].

While some work has been done in improving the CAM procedure in these regards, none have managed to completely alleviate the aforementioned problems, especially the one regarding the network's performance. The two most prominent studies [14, 15] involve recursively employing a discriminative localization model to produce more fine-grained CAMs. None of which, however, manage to achieve object-level detail or maintain a low computational cost (multiple training phases / inferences are required to produce a CAM).

The contributions of the present study are aimed at improving the quality and robustness of the produced CAMs, as well as the overall performance of the model. This will help both make discriminative localization more popular in neural networks, as well as deep learning models applicable in *sensitive* tasks.

Code available at: https://github.com/djib2011/high-res-mapping

978-1-5386-6249-6/19/$31.00 ©2019 IEEE

2. CLASS ACTIVATION MAPS

A Class Activation Map (CAM) is the region of the input image that the CNN uses to generate its prediction for that given class. In order for a CNN to be capable of producing CAMs, it needs to fulfill certain requirements; mainly it needs to conclude with a single Fully Connected (FC) layer, which feeds from a Global Average Pooling (GAP) layer.

For a given network, $f_k(x, y)$ is the activation of unit k of the last convolutional layer in the network, with a spatial location of (x, y). The next layer is a GAP which performs the following operation: $F^k = \sum_{x,y} f_k(x, y)$. The weighted average of this for all units is then passed to the softmax $S_c = \sum_k w_k^c F^k$, where w_k^c is the weight of unit k for class c and S_c is the class score (i.e. input of the sofmax for class c). By combining these two, the CAM for class c for each spatial location can be produced: $M_c(x, y) = \sum_k w_k^c f_k(x, y)$

In order to overlay the CAM on the original image, it needs to be resized to the same dimensions. The most common way is bilinear interpolation.

This constitutes one of the weaknesses of the procedure. The convolution layer prior to the GAP has much smaller dimensions than the original image, which constrains the whole pipeline to produce low-resolution CAMs, which then need to be resized to fit the original.

To solve this, we propose the addition of a secondary architecture, whose goal is to effectively upscale the low-resolution CAMs.

3. PROPOSED LOCALIZATION FRAMEWORK

The proposed framework for discriminative localization can be split into 3 parts: the initial pre-training of the classification/localization model, the training of the *expansion network* (i.e. a model aiming to produce high-resolution CAMs) and the postprocessing pipeline, which combines the two into a *refined CAM*.

3.1. Localization model

As mentioned previously, the addition of GAP layer normally drastically deteriorates the performance of the network. To overcome this, a DenseNet architecture [16] was employed.

This network is divided into *dense blocks*, which comprise multiple convolution layers each. The output of each of these layers is concatenated at the end of the block. Due to the large size of these filters, the architecture incorporates a GAP layer at the end to control the dimensions. This provides the model the ability to inherently produce a CAM for each of its predictions.

These models are at a near state-of-the-art level, achieving a top-5 error rate of around 3.6%; humans score around 5% at the same task. The model's architecture is represented in blue and black in Figure 1.

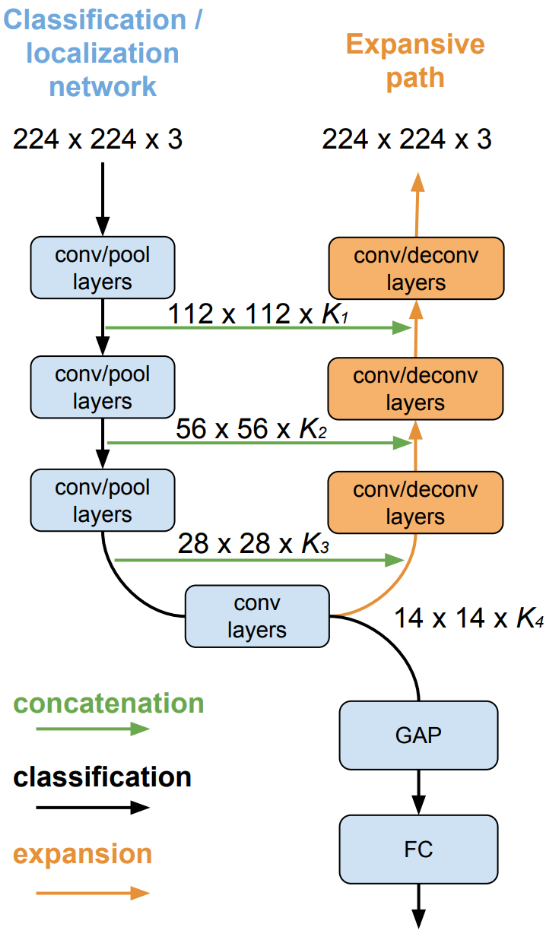

Fig. 1. The expansive path (orange), relative to the original classification model (blue). The network latches onto the final convolution layer of the classification model and upscales its feature maps to the original dimension. The green lines represent skip connections.

3.2. Expansion network

The second part of the framework, which will be referred to as the *expansion network*, involves a model trained for producing high-resolution CAMs. This was trained separately from the localization model for two reasons: to better control the training of both models and so that it can be trained completely unsupervised. The latter helps this part of the framework be more universal; the expansion network can be used on any pre-trained CNN capable of producing CAMs.

The proposed architecture draws inspiration from the UNet [5], a network used for image segmentation. The first half of the UNet is a typical CNN which contracts the image into a lower resolution while extracting useful information. The second half expands that low resolution image, again, into its original dimensions.

Given a CNN trained for image classification capable of generating CAMs, the goal of the expansion network is to enhance this in order to produce a better quality of CAMs. By adding a second part to the model, which mirrors the first and establishes skip connections forming a UNet-like architecture, the desired result can be accomplished. The proposed architecture is depicted in orange, in Figure 1.

The expansion network is trained completely unsupervised, by passing the input image as the label. During its training the original CNN's weights should be kept frozen. Because of this, the model never loses its capability of making predictions and generating CAMs (these will be denoted as "low-res CAMs"). The expansive path produces CAMs with the dimensions of the original image and a high level of detail (these will be denoted as "high-res CAMs"). Because this path isn't trained with the image labels, it does not retain its localization ability. In order to generate the final localized object, information from both CAMs needs to be combined during postprocessing.

Due to the selection of a DenseNet type architecure for classification, the expansive path is similar to the upscaling part of the FC-DenseNet [17].

3.3. Postprocessing pipeline

The main goal of the postprocessing pipeline is to extract information about the location of the object from the low-res CAM, while the exact size, shape and form of the object will be extracted from the high-res CAM.

Wherever operations between two different image forms are performed, the CAMs are normalized in $[0, 1]$ and the low-res CAM in particular is resized (bilinear interpolation).

The low-res CAM contributes in two ways: by providing a list of focal points and a Region Of Interest (ROI). The focal points are found by identifying the local maxima in the low-res CAM. The ROI is extracted through a threshold segmentation technique, aimed at keeping the 15% highest intensity pixels. These are used in the subsequent step to provide information for the object's location to the high-res CAM.

Due to the last up-scaling layer, the high-res CAM exhibits some "artifacts" which need to be removed. To accomplish this a blurring technique is performed through a 5×5 convolutional filter. A Sobel filter is then applied, which computes the gradient of the image and works as an edge detector [18].

The combination of the two types of CAMs will be accomplished through a threshold-based region growing segmentation technique, called *Connected Threshold* [19].

Initially, it requires one or more seeds from where to start the procedure from; the focal points will serve as the seeds. Additionally, the lower and upper bounds for the intensity that will stop the region's expansion are set at the first and third quartiles of the high-res CAM's pixels that lie in the ROI.

The resulting segmentation can be further refined by two methods. The first is by filling small "holes" that are below a specified threshold, while the other combines the segmentation with the low-res CAM to achieve a similar effect. The resulting map will be referred to as the *refined CAM*.

4. EXPERIMENTAL EVALUATION

4.1. Dataset construction and preprocessing

In order to test the effectiveness and robustness of the proposed methodology, a dataset needed to be selected that would meet the following parameters: First, the dataset needed to be large in size, so that a classifier could be effectively trained. Additionally, the images needed to have a high resolution. In cases where the images have a low resolution (e.g. MNIST, cifar), the low-res CAM would be sufficient. Finally, to make the localization task even harder and showcase the capabilities of the proposed methodology, the classes needed to be similar to one another.

For these reasons a subset of the ILSVRC-2012 dataset was selected, that contained only *animals* classes. This proved to be an ideal dataset for the purpose of this study, as it was large in size (i.e. 510,530 images), the image resolution was sufficiently high and the image classes are very homogeneous (e.g. 'tabby cat', 'tiger cat', 'Persian cat', 'Siamese cat', 'Egyptian cat'). The images were then resized to 224×224 and normalized.

4.2. Classifier pre-training

The first part of the proposed framework (Sec. 3.1) involved pretraining a CNN capable of producing CAMs. The network was trained with an adadelta optimizer [20] on a cross-entropy loss for 50 epochs. The validation accuracy (top-1) peaked 20 epochs at 47.9%. This mark is slightly lower than the ILSVRC state-of-the-art, which can be attributed to the high homogeneity in the classes, i.e. it is harder to distinguish between them.

The following data augmentation techniques were used during this step: left-right flip, rotation, $90 - 110\%$ scaling, $-20 - +20\%$ translation, $-5 - +5\%$ shear, all with a probability of 50%. All of these were selected empirically.

4.3. Expansion network training

Given the pretrained CNN classifier, the expansion network is trained to produce high-resolution CAMs. This is essentially built on top of the classifier, whose weights are kept frozen during the whole training phase.

An additional 50 epochs were required to train the expansion network, using the same augmentation scheme as in Sec. 4.2. Again, an adadelta optimizer was used, on a mean squared error reconstruction loss.

4.4. Postprocessing

The postprocessing pipeline described in Section 3.3 will be presented here step-by-step. A sample image is (Fig. 2) is fed to the model; the image's low-res CAM for the predicted class can be seen in the same figure. This map indicates where the image activates the neural network in order to make its prediction. This is a very coarse and low detail image that is good only in cases where the target class represents a single object that occupies a large part of the image.

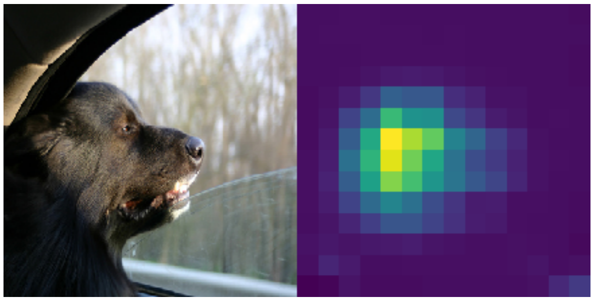

Fig. 2. On the left: a sample test set image. On the right: its low-resolution CAM.

To apply the region growing segmentation, the high-res map must first undergo a smoothing and edge detection procedure, as described in Sec. 3.3. Meanwhile the focal points and ROI are extracted from the low-res map. The processed high-res map, the focal points and the boundary selection procedure is illustrated in Figure 3.

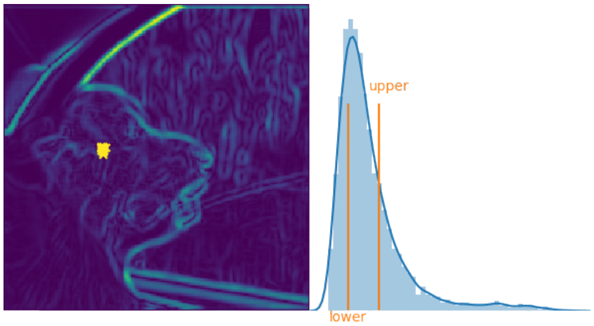

Fig. 3. On the left: the edges detected from the processed high-res CAM along with the focal points (in yellow). On the right: the boundary selection procedure.

The segmented high-res CAM, after the edges were added and its small holes were filled, is merged with the low-res map to produce the refined CAM, which is illustrated in Figure 4. The masked original image is also presented in the same figure.

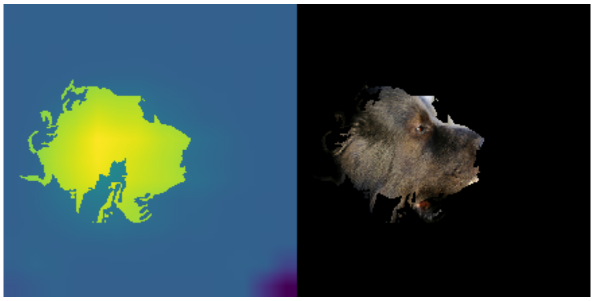

Fig. 4. On the left: Refined CAM. Occurs by merging the output of the region-growing pipeline with the low-res CAM. On the right: Original image after masking it, as dictated by the refined CAM.

5. DISCUSSION

As a result of combining the low and high resolution maps for producing the final CAM, it becomes more robust in case any of the two fails.

Most times when the model is unsure about a prediction, the low-res CAM is activated in a random spot depending on the specific model. The rest of the map might still be slightly activated in the region where the object is. Due to the consideration of local maxima for focal points and the region-growing technique applied, there is a high chance that the high-res CAM will still identify the correct object, despite the complete collapse of the low-res CAM.

The high-res CAM can fail at the region-growing phase if the features extracted during the expansion path are not so strong. A failure can either mean that the area turned out much smaller or that it turned out much larger than expected. In both cases, due to the eventual blending of the two maps, the output CAM will regress to its low-res state.

6. CONCLUSION

This paper proposes a novel framework for robust and high-quality Class Activation Mapping (CAM), while still achieving near state-of-the-art levels of performance. CAMs can be used both for object localization and for providing reasoning during classification.

The refined CAMs are produced by combining the original, low-res maps with the ones produced by a network called the *expansion network*. The latter is trained to generate high-resolution maps, which can provide more fine-grained detail than their lower-resolution counterparts. By combining both, the refined CAMs are much more robust and can work even in cases where either one of their components fail.

Future work will be directed towards jointly training the classification and expansion networks, further refining the postprocessing procedure and investigating the reasons for failure of each of the two types of CAMs.

7. REFERENCES

[1] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[2] Barret Zoph, Vijay Vasudevan, Jonathon Shlens, and Quoc V Le, "Learning transferable architectures for scalable image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 8697–8710.

[3] Jie Hu, Li Shen, and Gang Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7132–7141.

[4] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio, "Generative adversarial nets," in *Advances in neural information processing systems*, 2014, pp. 2672–2680.

[5] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*. Springer, 2015, pp. 234–241.

[6] Saikat Basu, Sangram Ganguly, Supratik Mukhopadhyay, Robert DiBiano, Manohar Karki, and Ramakrishna Nemani, "Deepsat: a learning framework for satellite imagery," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2015, p. 37.

[7] Brody Huval, Tao Wang, Sameep Tandon, Jeff Kiske, Will Song, Joel Pazhayampallil, Mykhaylo Andriluka, Pranav Rajpurkar, Toki Migimatsu, Royce Cheng-Yue, et al., "An empirical evaluation of deep learning on highway driving," *arXiv preprint arXiv:1504.01716*, 2015.

[8] Dimitrios Kollias, Athanasios Tagaris, and Andreas Stafylopatis, "On line emotion detection using retrainable deep neural networks," in *Computational Intelligence (SSCI), 2016 IEEE Symposium Series on*. IEEE, 2016, pp. 1–8.

[9] Athanasios Tagaris, Dimitrios Kollias, Andreas Stafylopatis, Georgios Tagaris, and Stefanos Kollias, "Machine learning for neurodegenerative disorder diagnosis-survey of practices and launch of benchmark dataset," *International Journal on Artificial Intelligence Tools*, vol. 27, no. 03, pp. 1850011, 2018.

[10] José Manuel Benítez, Juan Luis Castro, and Ignacio Requena, "Are artificial neural networks black boxes?," *IEEE Transactions on neural networks*, vol. 8, no. 5, pp. 1156–1164, 1997.

[11] Vimla L Patel, Edward H Shortliffe, Mario Stefanelli, Peter Szolovits, Michael R Berthold, Riccardo Bellazzi, and Ameen Abu-Hanna, "The coming of age of artificial intelligence in medicine," *Artificial intelligence in medicine*, vol. 46, no. 1, pp. 5–17, 2009.

[12] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba, "Learning deep features for discriminative localization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2921–2929.

[13] Jorge Sánchez and Florent Perronnin, "High-dimensional signature compression for large-scale image classification," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 1665–1672.

[14] Xinyang Feng, Jie Yang, Andrew F Laine, and Elsa D Angelini, "Discriminative localization in cnns for weakly-supervised segmentation of pulmonary nodules," in *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer, 2017, pp. 568–576.

[15] Jianlong Fu, Heliang Zheng, and Tao Mei, "Look closer to see better: Recurrent attention convolutional neural network for fine-grained image recognition," in *CVPR*, 2017, vol. 2, p. 3.

[16] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 4700–4708.

[17] Simon Jégou, Michal Drozdzal, David Vazquez, Adriana Romero, and Yoshua Bengio, "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," in *Computer Vision and Pattern Recognition Workshops (CVPRW), 2017 IEEE Conference on*. IEEE, 2017, pp. 1175–1183.

[18] Nick Kanopoulos, Nagesh Vasanthavada, and Robert L Baker, "Design of an image edge detection filter using the sobel operator," *IEEE Journal of solid-state circuits*, vol. 23, no. 2, pp. 358–367, 1988.

[19] Yian-Leng Chang and Xiaobo Li, "Adaptive image region-growing," *IEEE transactions on image processing*, vol. 3, no. 6, pp. 868–872, 1994.

[20] Matthew D Zeiler, "Adadelta: an adaptive learning rate method," *arXiv preprint arXiv:1212.5701*, 2012.