High-Resolution Class Activation Mapping

Thanos Tagaris*, Maria Sdraka and Andreas Stafylopatis
*thanos@islab.ntua.gr

National Technical University of Athens

Code Available at github.com/djib2011/high-res-mapping

Abstract

- Neural Networks can't provide sufficient reasoning for their decisions. They operate like *black boxes*. This is especially true in Deep Neural Networks.
- What a Convolutional Neural Network (CNN) could also output *why* it made a given prediction.
- We present a framework for discriminative localization that helps shed some light into the decision-making of CNNs.
- Compared to related work, our approach generates robust, refined and high-quality Class Activation Maps, without impairing the CNN's performance.

Motivation

- Even though Deep Neural Networks have achieved state-of-the-art performance in several image and text related tasks, they have experienced a **slow adoption rate** by some industries.
- More specifically, in sensitive domains where mistakes matter and there is a notion of **responsibility** and **accountability**.
- Some examples are the fields of medical diagnoses and autonomous driving. Even though research indicates that Deep Learning algorithms surpass human performance, these techniques haven't been adopted in these fields. The issue of **trust** is the main reason.
- If there was a way for a CNN to provide a reasoning for its decisions, then it would be easier for humans to *trust* it.

Class Activation Mapping

- A Class Activation Map (CAM) is the region of the input image that the CNN uses to predict a given class.
- A CNN can produce CAMs if it concludes with a Global Average Pooling (GAP) and a Fully Connected (FC) layer
- Unfortunately, the GAP layer can deteriorate the network's performance. This happens because it is used to collapse (i.e. summarize) the information of a whole feature map into a single scalar.
- Here we face a **tradeoff** between the network's performance and CAM resolution:
  - If the feature maps before the GAP have a **high-resolution** then the information lost due to the "collapse" will be large, which **reduces** the network's performance significantly.
  - If the maps have a **low-resolution**, the network's performance isn't affected but the CAMs won't be very detailed.

Proposed Localization Framework

The proposed framework consists of 3 parts: the *Localization Network*, the *Expansion Network* and the *Postprocessing Pipeline*. Compared to similar frameworks, it is capable of:

1. Producing high-resolution, refined and robust Class Activation Maps.
2. Maintaining the state-of-the-art performance.

Localization Network

- This is a regular CNN with the ability of producing CAMs.

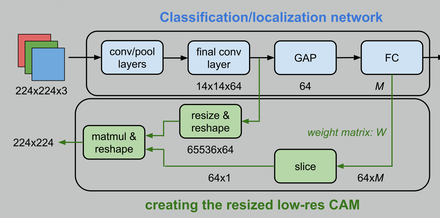

Figure: The top of the image represents the classification path of the Network. The bottom depicts the process of generating a low-res CAM.

- In order to maintain state-of-the-art performance, the network needs to have **low-resolution** feature maps before its GAP layer.
- The architecture elected is **DenseNet**, a network capable of achieving a top-5 error rate of 3.6% on the ImageNet dataset.

Expansion Network

- The "Expansion Network" is built to *mirror* the Localization Network
- Its goal is to take the low-resolution feature maps and expand them to the dimensions of the original image, used to produce the **High-Resolution CAMs**.
- Its inputs are the feature maps of the *Localization Network* and its target is the original image (i.e. like an autoencoder).
- The high-res CAMs **don't retain the localization capability** of the low-res CAMs, because they aren't trained on the class labels.
- The following Figure depicts the **combined** architecture that includes both the Localization and Expansion Networks.

Combined Model Architecture

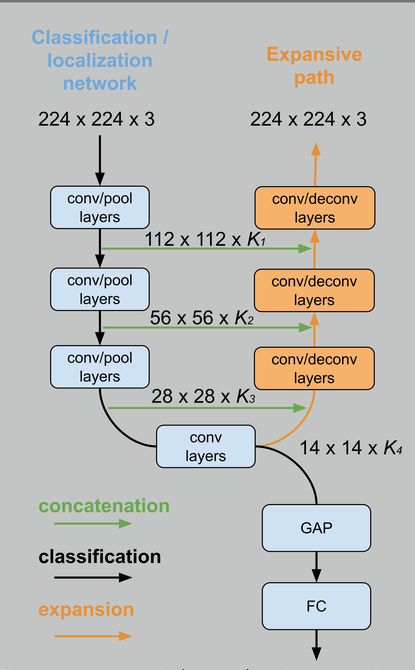

Figure: The expansive path (orange), relative to the original classification model (blue). The network latches onto the final convolution layer of the classification model and upscales its feature maps to the original dimension. The green lines represent skip connections.

Postprocessing Pipeline

The goal of this pipeline is to combine the low and high-resolution maps into a single **Refined CAM**. To accomplish this the following postprocessing steps were drafted:

1. The *Low-Resolution CAM* contributes to the *Refined CAM* in two ways: Its focal points are identified and a Region of Interest (ROI) is extracted.
2. The high-resolution CAM is first blurred then passed through a Sobel filter for edge detection.
3. A threshold-based Region Growing segmentation technique is applied on the previous high-res map, using the data (i.e. starting seeds, threshold values) from the low-res one.
4. The resulting segmented map is combined with the upscaled low-res one, after first having having filled its small "holes", to produce the *Refined CAM*.

Experimental Procedure

- The framework was evaluated on the **animals** classes from the ImageNet dataset. This offers the unique characteristic of having a large number of classes (i.e. 398) that are very similar to one another (e.g. 'tabby cat', 'tiger cat', 'Persian cat', 'Siamese cat', 'Egyptian cat').
- The first step is to pre-train the *Localization Network* for classification.
- Afterwards, the combined architecture (i.e. localization and expansion) is trained in unsupervised fashion, while keeping the localization network's weights frozen.
- To provide a *Refined CAM*, the two CAMs generated from the network (i.e. low and high-res) are passed through the *Postprocessing pipeline*.

Example Pipeline

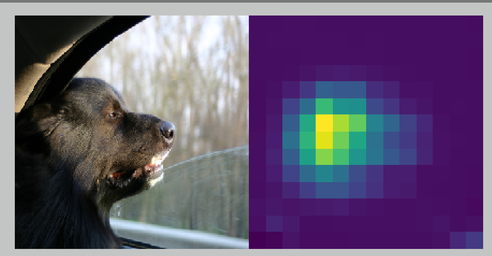

Figure: Left: an example image. Right: the low-res CAM.

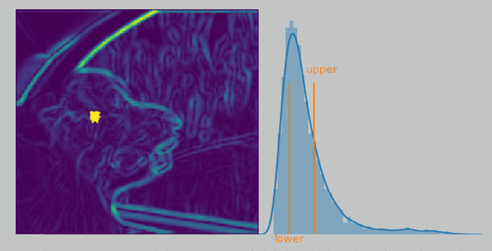

Figure: The processed high-res CAM with the focal points to the left; the boundary selection process to the right.

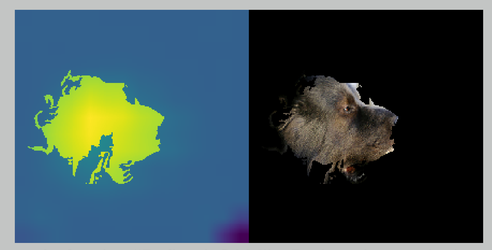

Figure: Left: the "Refined CAM". Right: the masked image.