

# Learning Deep Features for Discriminative Localization

• [[1512.04150.pdf]]

## Basic Idea:

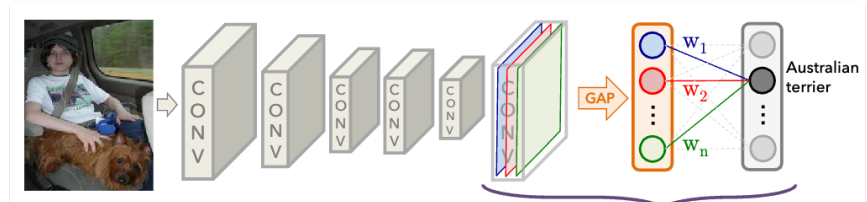
- In image classification, can we visualize/highlight regions in the image which led to the prediction of a particular class?

## Class Activation Mapping (CAM)

- **Definition:** P2 A class activation map for a particular category indicates the discriminative image regions used by the CNN to identify that category

## Architecture

P3



- Convolutional Backbone → Global Average Pooling (GAP) → Single FC layer for classification
- P2 Given this simple connectivity structure, we can identify the importance of the image regions by projecting back the weights of the output layer on to the convolutional feature maps, a technique we call class activation mapping.

## Computation of CAM

### Notations:

- $f_k(x, y)$ : Activation of unit  $k$  in the last convolutional layer at spatial location  $(x, y)$ .
- $F^k$ : The output for GAP for unit  $k$
- $S_c$ : Input to the softmax for class  $c$
- $w_k^c$ : Weight corresponding to class  $c$  for unit  $k$
- $P_c$ : Output of softmax for class  $c$  or probability for class  $c$

### Note that:

$$F^k = \sum_{x,y} f_k(x, y)$$

$$S_c = \sum_k w_k^c F_k$$

$$P_c = \frac{\exp(S_c)}{\sum_c \exp(S_c)}$$

- Intuitively,  $w_k^c$  represents the importance of  $F_k$  for class  $c$ .
- Here the bias term is ignored: input bias of the softmax is set to 0 P3 as it has little to no impact on the classification performance.
- Now, substituting the value of  $F_k$  in class score,  $S_c$ , we get



$$\begin{aligned} S_c &= \sum_k w_k^c \sum_{x,y} f_k(x, y) \\ &= \sum_{x,y} \sum_k w_k^c f_k(x, y) \end{aligned}$$

- The class activation map for class  $c$ ,  $M_c$  is defined as follows:





$$M_c(x, y) = \sum_k w_k^c f_k(x, y)$$

- P3 hence  $M_c(x, y)$  directly indicates the importance of the activation at spatial grid  $(x, y)$  leading to the classification of an image to class  $c$
- P3 The class activation map is simply a weighted linear sum of the presence of these visual patterns at different spatial locations.
- P3 By simply upsampling the class activation map to the size of the input image, we can identify the image regions most relevant to the particular category P4 We found that the

localization ability of the networks improved when the last convolutional layer before GAP had a higher spatial resolution, which we term the mapping resolution. In order to do this, we removed several convolutional layers from some of the networks. Can ideas of atrous convolutions help here?

-  **P2** The same technique can be applied to regression and other losses. 

- *Global Average Pooling (GAP) vs Global Max Pooling (GMP):*





-  **P3** We believe that GAP loss encourages the network to identify the extent of the object as compared to GMP which encourages it to identify just one discriminative part. 
-  **P3** This is because, when doing the average of a map, the value can be maximized by finding all discriminative parts of an object as all low activations reduce the output of  **P4** the particular map. On the other hand, for GMP, low scores for all image regions except the most discriminative one do not impact the score as you just perform a max.
- Empirically, they prove that GMP achieves similar classification performance to GAP. However, GAP outperforms GMP for localization.

- *Food for thought*


- The weighted summation of feature maps at each spatial location is similar to channel-wise attention in CNNs. The difference between attention and CAM is that the weights in attention are positive and lie between  $[0, 1]$ , whereas in CAM the weights can be any real number. Therefore, will taking  $\text{softmax}$  of  $w_k^c$  for each  $c$  (i.e., taking  $\text{softmax}$  across  $k$ ), help in improving the CAMs?

- **Results**



- *Weakly-supervised Object Localization:*

-  **P4** Note that it is important for the networks to perform well on classification in order to achieve a high performance on localization as it involves identifying both the object category and the bounding box location accurately. However, note that having high-classification performance is not just the only thing required for better localization performance. For example, GoogLeNet and GoogLeNet-GAP have 31.9% and 35% top-1 error rates for classification respectively. However, GoogLeNet-GAP achieves 56.40% top-1 error rate in comparison to GoogLeNet which achieves 60.09%. This can be attributed to  **P5** We believe that the low mapping resolution of GoogLeNet ( $7 \times 7$ ) prevents it from obtaining accurate localizations.
-  **P4** To generate a bounding box from the CAMs, we use a simple thresholding technique to segment the heatmap. We first segment the regions of which the value is above 20%  **P5** of the max value of the CAM. Then we take the bounding box that covers the largest connected component in the segmentation map.

- *Deep Features for Generic Localization:*




- The objective is to show that the features learned by GAP CNNs are generic enough to be transferred to other tasks.
-  **P5** To obtain the weights similar to the original softmax layer, we simply train a linear SVM [5] on the output of the GAP layer. The weights of the linear SVM could be used to compute CAM.

- *Pattern Discovery:*

-  **P6** In this section, we explore whether our technique can identify common elements or patterns in images beyond  **P7** objects, such as text or high-level concepts. Given a set of images containing a common concept, we want to identify which regions our network recognizes as being important and if this corresponds to the input pattern.
- Three tasks for pattern discovery:
  - *Scene categorization:* Instead of just recognizing objects, can we identify whether the image is of a dining room or bedroom? This requires learning about the presence of multiple objects.
  - *Weakly supervised Concept Learning:* Learn to identify concepts in images with specific concepts like mirror in lake, view out of window etc. Again no information about particular object is given.
  - *Weakly supervised text detector:* Detect text in the images without being explicitly trained, i.e., train the network on just whether the image contains text or not. Note this again involves learning in general what a text is because each image can contain different characters, be written in different fonts etc.

- *Visualizing Class-Specific Units:*

- The objective is to determine which convolutional units are most discriminative for a given class. These are called *class-specific units* of a CNN.

- Take each unit in the final convolutional layer and compute its receptive field. Since, the receptive field of this layer would be very high, we need to segment the portions which contribute highly to this activation. For this,  **P8** We follow a similar procedure as [33] for estimating the receptive field and segmenting the top activation images of each unit in the final convolutional layer. Now we can use  $w_k^c$  to rank the units according to their relative importance.  **P8** Then we simply use the softmax weights to rank the units for a given class.
-  **P8** Thus we could infer that the CNN actually learns a bag of words, where each word is a discriminative class-specific unit. A combination of these class-specific units guides the CNN in classifying each image. 