GAP, Localization and CNN Explanations

Giacomo Boracchi

giacomo.boracchi@polimi.it

Localization

The Localization Task

The input image contains a single relevant object to be classified in a fixed set of categories

The task is to:

1) assign the object class to the image

hawk

The Localization Task

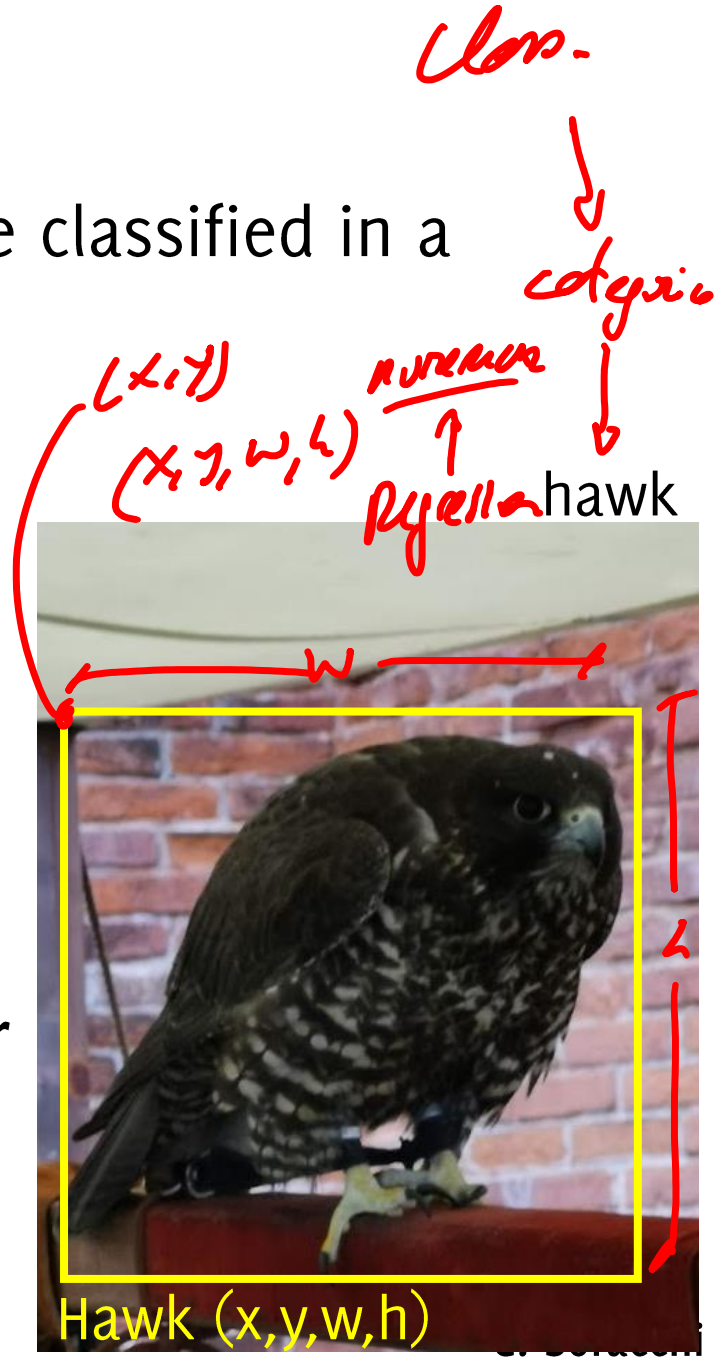
The input image contains a single relevant object to be classified in a fixed set of categories

The task is to:

1) assign the object class to the image
2) locate the object in the image by its bounding box

A training set of annotated images with **label and a bounding box** around each object is required

Extended localization problems involve regression over more complicated geometries (e.g. human skeleton)



Localization, the problem

Assign to an input image $I \in \mathbb{R}^{R \times C \times 3}$:

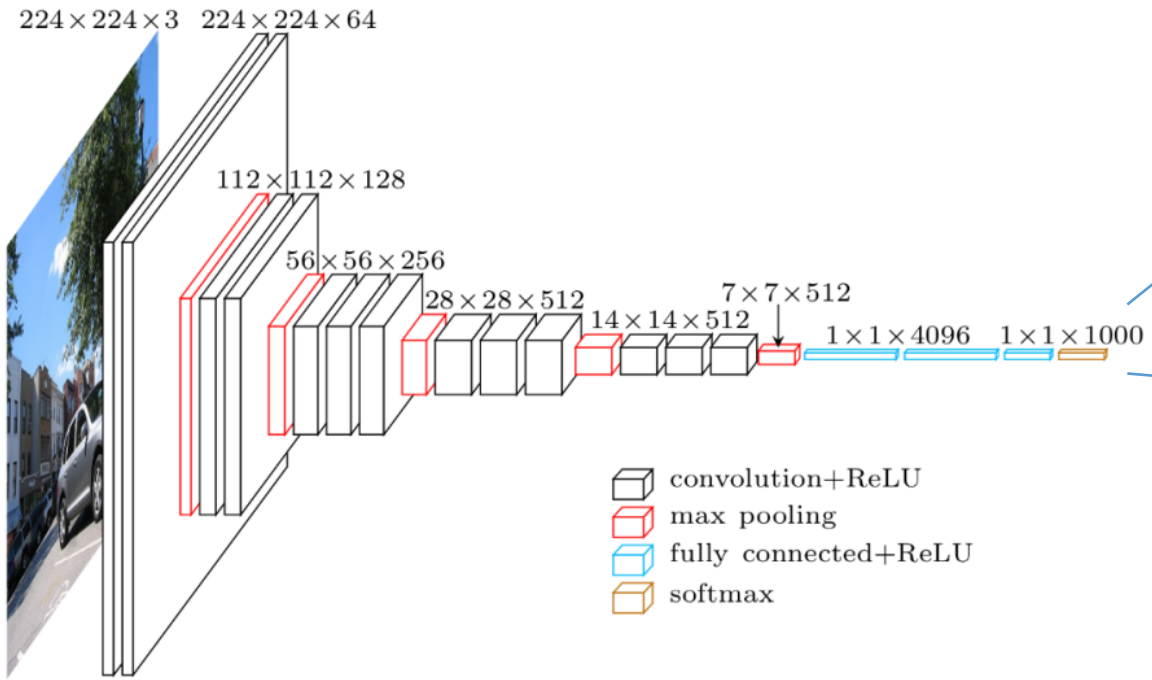
- a label l from a fixed set of categories
$$\Lambda = \{\text{"wheel"}, \text{"cars"}, \ldots, \text{"castle"}, \text{"baboon"}\}$$
- the coordinates (x, y, h, w) of the bounding box enclosing that object

$$I \rightarrow (x, y, h, w, l)$$

The Simplest Solution

Train a network to predict both the class label and the bounding box

"Two heads"



Target class

Soft-max loss $\mathcal{S}$

Categorical cross entropy loss

Bounding Box Coordinates

(x, y, w, h)

Regression loss $\mathcal{R}$, e.g. the $\ell^2, \ell^1, ..$ bounding box.

$$\|[\hat{x}, \hat{y}, \hat{w}, \hat{h}] - [x, y, w, h]\|_2$$

ESTIMATED one

TRUE B.B location

The Simplest Solution

The training loss has to be a single scalar since **we compute gradient of a scalar function** with respect to network parameters.

Minimize a multitask loss to merge two losses:

$$\mathcal{L}(x) = \alpha\, \mathcal{S}(x) + (1 - \alpha)\mathcal{R}(x)$$

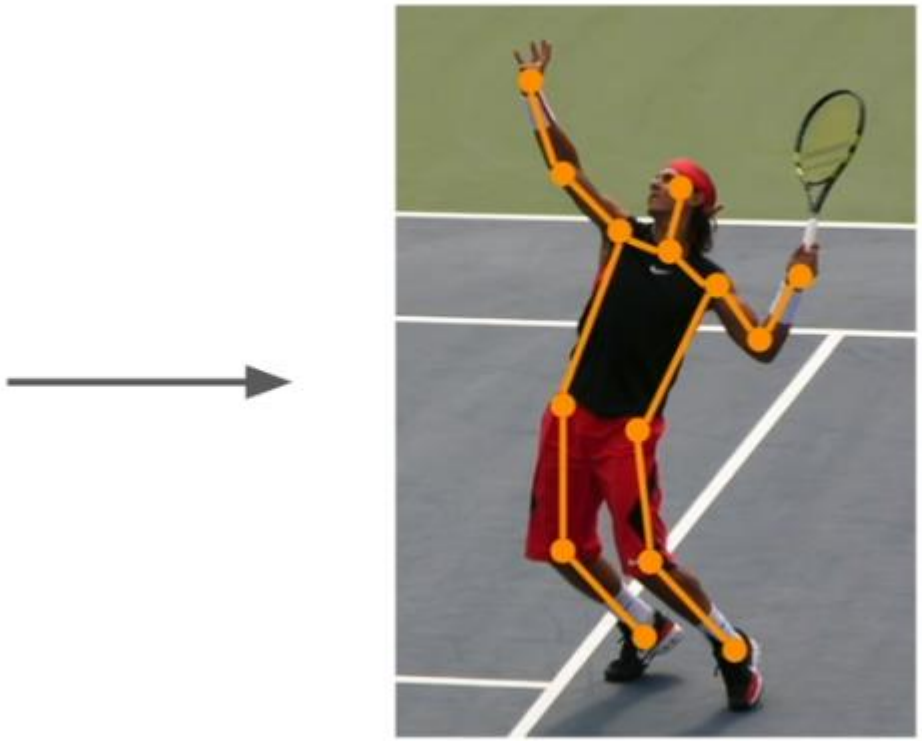
and α is an hyper parameter of the network.

Watch out that $\boldsymbol{\alpha}$ **directly influences the loss definition**, tuning might be difficult. Better to do cross-validation looking at some other loss (loss value for different values of α might be meaningless).

It is also possible to **adopt a pre-trained model** and then train the two FC separately… however it is always better to perform at least some fine tuning to train the two jointly.

Extension to Human Pose Estimation

Pose estimation is formulated as a **CNN-regression problem towards body joints**. It is possible to **address localization only here**



Represent pose as a set of 14 joint positions:

- Left / right foot
- Left / right knee
- Left / right hip
- Left / right shoulder
- Left / right elbow
- Left / right hand
- Neck
- Head top

6x2

2

14 locations to estimate
28 output neurons

Alexander Toshev and Christian Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks", CVPR 2014

Extension to Human Pose Estimation

Pose estimation is formulated as a **CNN-regression problem towards body joints.**

- **The network receives as input the whole image, capturing the full-context of each body joints.**
- The approach is **very simple to design and train.** Training problems can be **alleviated by transfer learning** of existing classification networks

Pose is defined as a vector of k joints location for the human body, possibly normalized w.r.t. the bounding box enclosing the human.

Train a CNN to predict **a $2k$ vector as output** by using an Alexnet-like architecture.

Alexander Toshev and Christian Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks", CVPR 2014

Training Human Pose Estimation Networks

Adopt a ℓ^2 regression loss of the estimated pose parameters over the annotations.

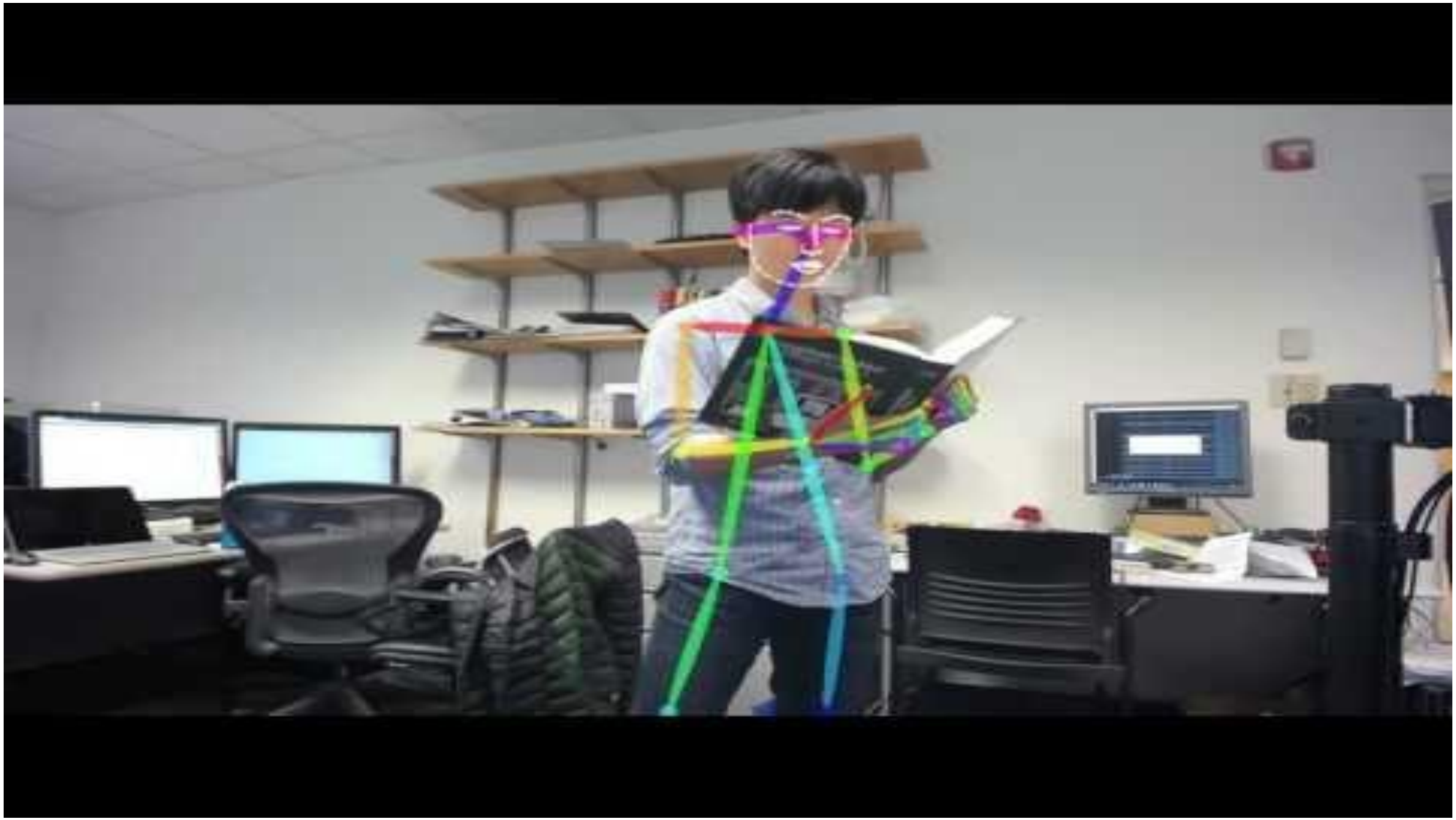
- This can be also defined when a few joints are not visible.

Reduce overfitting by augmentation (translation and flips).

Multiple networks have been trained to improve localization by refining joint position in a crop around the initial detection.

Alexander Toshev and Christian Szegedy, “DeepPose: Human Pose Estimation via Deep Neural Networks”, CVPR 2014

Open Pose



Cao, Z et al.. OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields. CVPR 2017

Pose Estimation

Cao, Z., Simon, T., Wei, S. E., & Sheikh, Y. (2017). Realtime multi-person 2d pose estimation using part affinity fields. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 7291-7299).

Weakly-Supervised Localization

… Global Averaging Pooling Revisited

… visualizing what matters most for CNN predicitons

G. Boracchi

Weakly supervised localization

Perform localization over an image without images with annotated bounding box

- Training set provided as for classification with image-label pairs $\{(I, \ell)\}$ where no localization information is provided

2016

Learning Deep Features for Discriminative Localization

Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, Antonio Torralba
Computer Science and Artificial Intelligence Laboratory, MIT
{bzhou,khosla,agata,oliva,torralba}@csail.mit.edu

G. Boracchi

The GAP revisited

The advantages of GAP layer **extend beyond** simply acting as a structural **regularizer** that prevents overfitting

In fact, **CNNs can retain a remarkable localization ability** until the final layer. By a simple tweak it is possible to easily identify the discriminative image regions leading to a prediction.

*A **CNN trained on object categorization** is successfully able to **localize the discriminative regions for action classification** as the objects that the humans are interacting with rather than the humans themselves*

Zhou, Bolei, et al. "Learning deep features for discriminative localization.“ CVPR 2016.

Class Activation Mapping

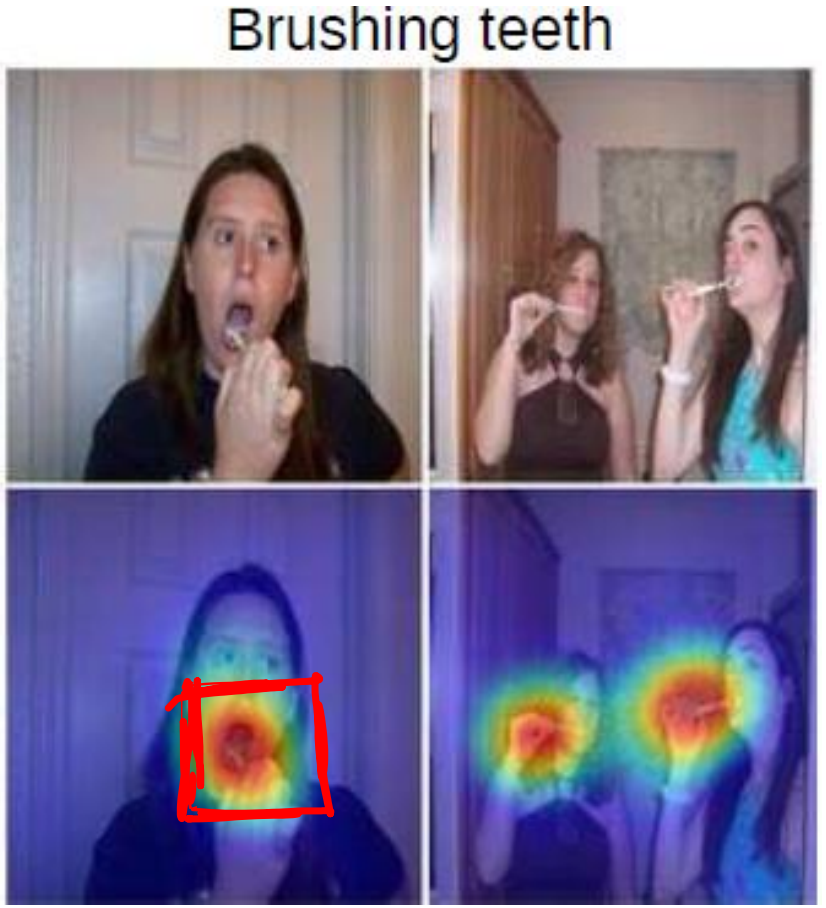
Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

Class Activation Mapping (CAM)

Identifying exactly **which regions** of an image **are being used for discrimination**.

CAM are very easy to compute. It just requires:

- FC layer after the GAP
- a minor tweak

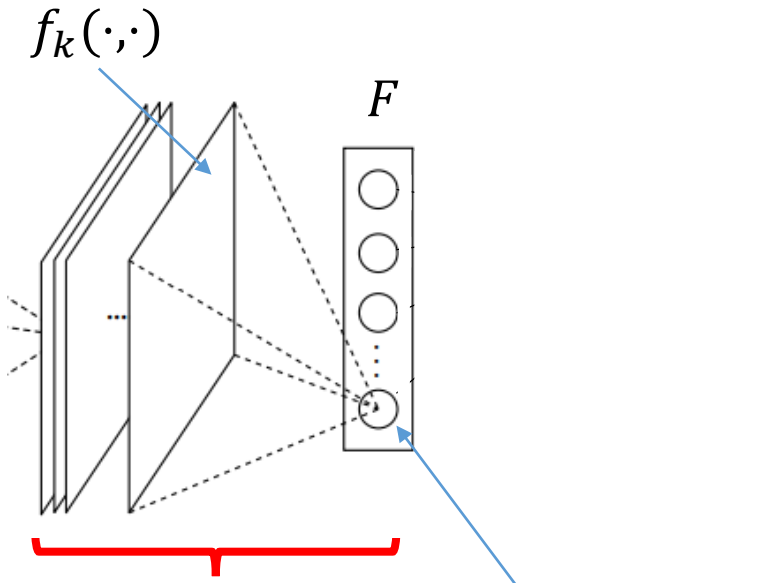


Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

The Global Averaging Pooling (GAP) Layer

A very simple architecture made only of convolutions and activation functions leads to a final layer having:

- n feature maps $f_k(\cdot,\cdot)$ having resolution "similar" to the input image
- a vector after GAP made of n averages F_k



Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

The Global Averaging Pooling (GAP) Layer

Add (and train) a **single FC layer** after the GAP.

The FC computes S_c for each class c as the weighted sum of $\{F_k\}$, where weights are defined during training

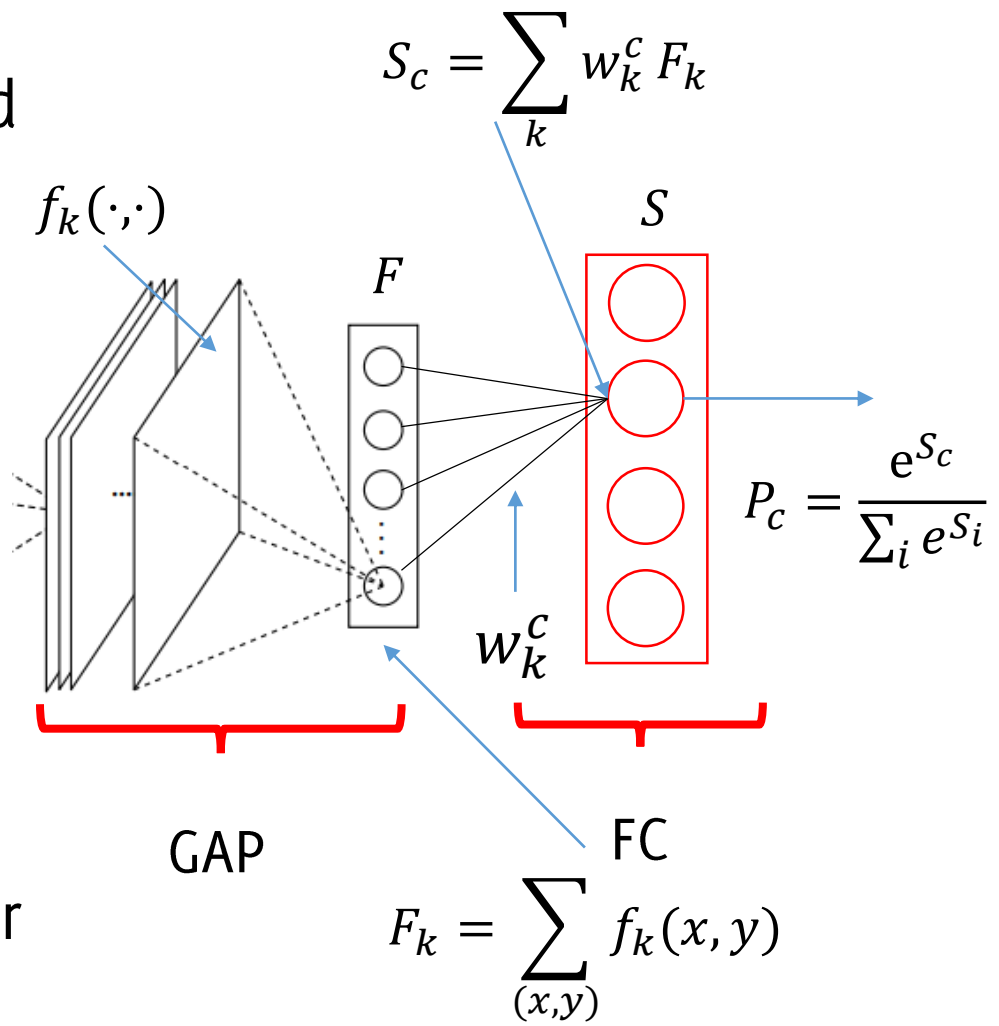
Then, the class probability P_c via soft-max (class c)

Remark: when computing

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

w_k^c encodes the importance of F_k for the class c,

$\{w_k^c\}_{k,c}$ are all the parameters of the last FC layer



Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

The Global Averaging Pooling (GAP) Layer

However

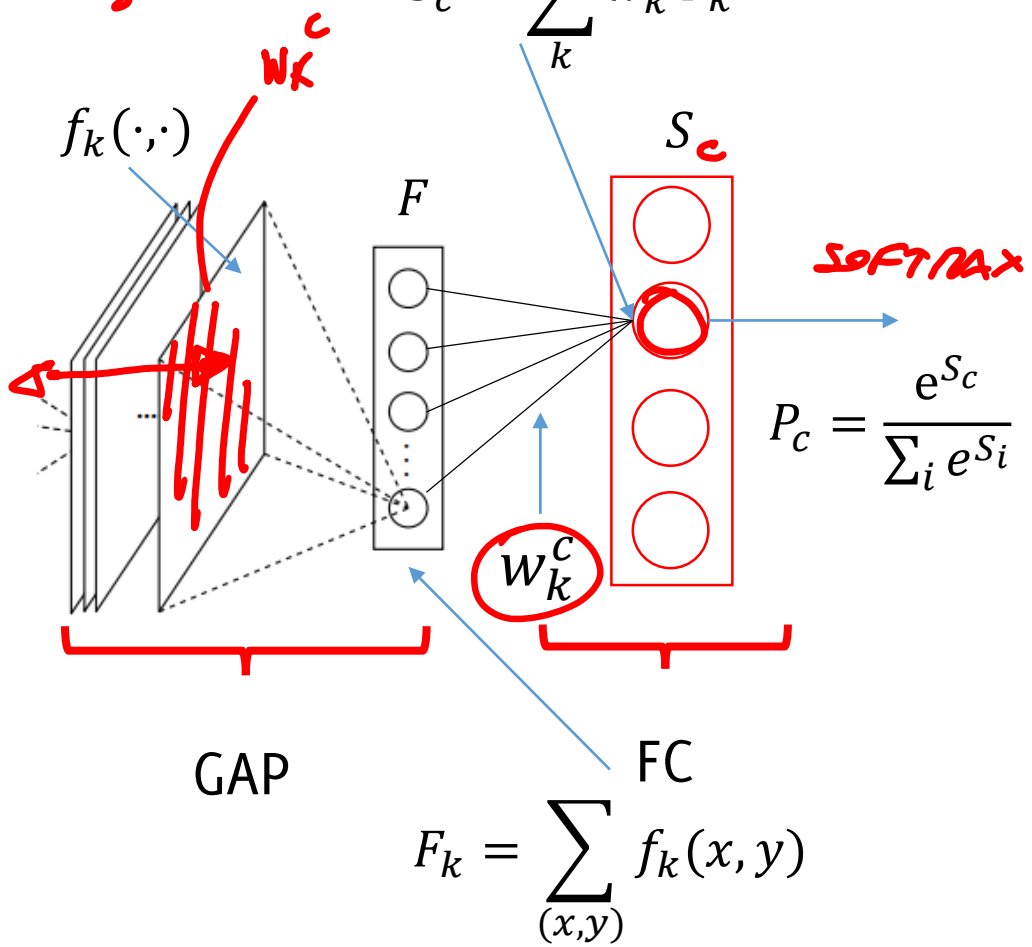
$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \sum_k w_k^c \, f_k(x,y)$$

And CAM is defined as

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

where $M_c(x,y)$ directly indicates the importance of the activations at (x,y) for predicting the class c

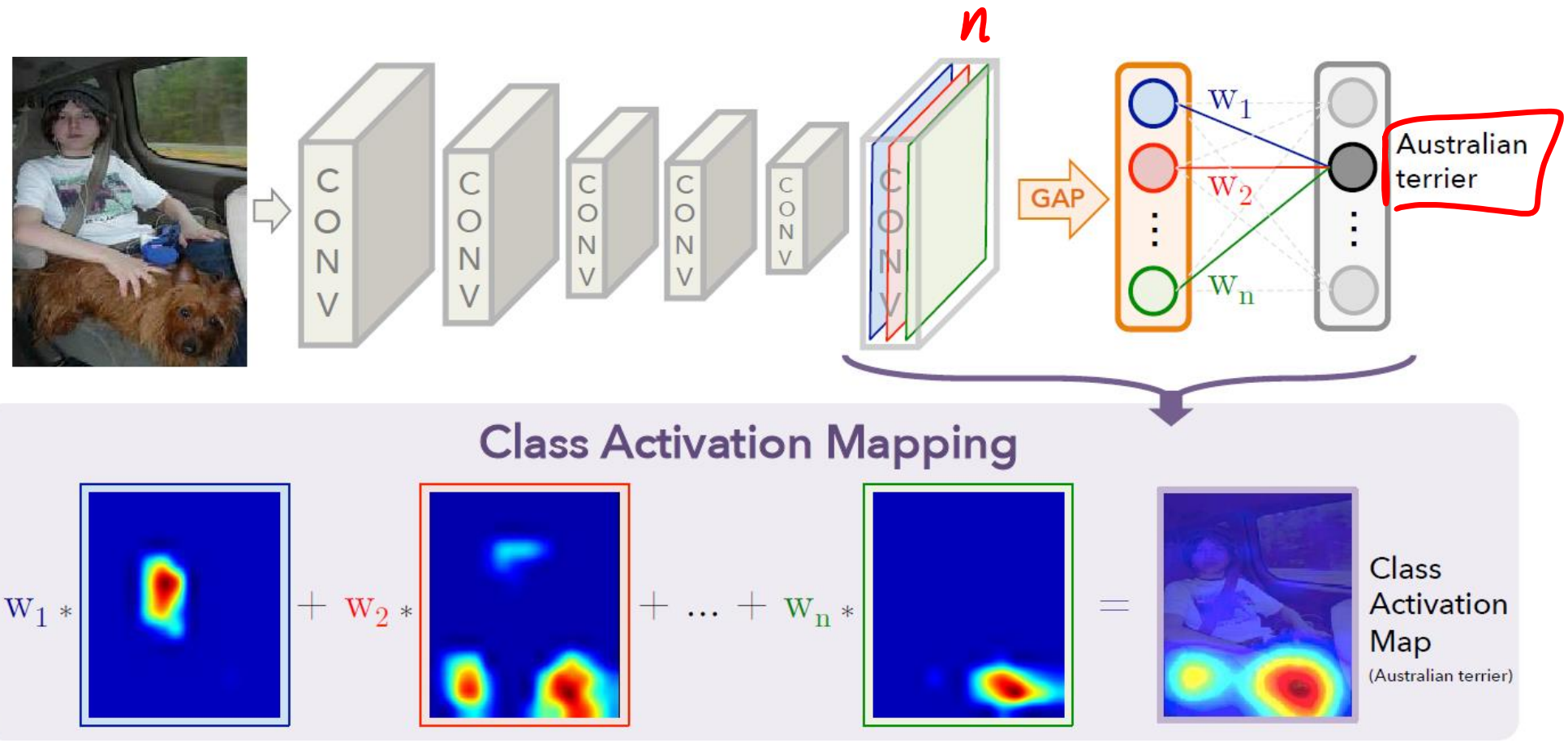
Rmk: unlike GAP, thanks to the softmax, **the depth of the last convolutional activations can differ from the number of classes**



Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

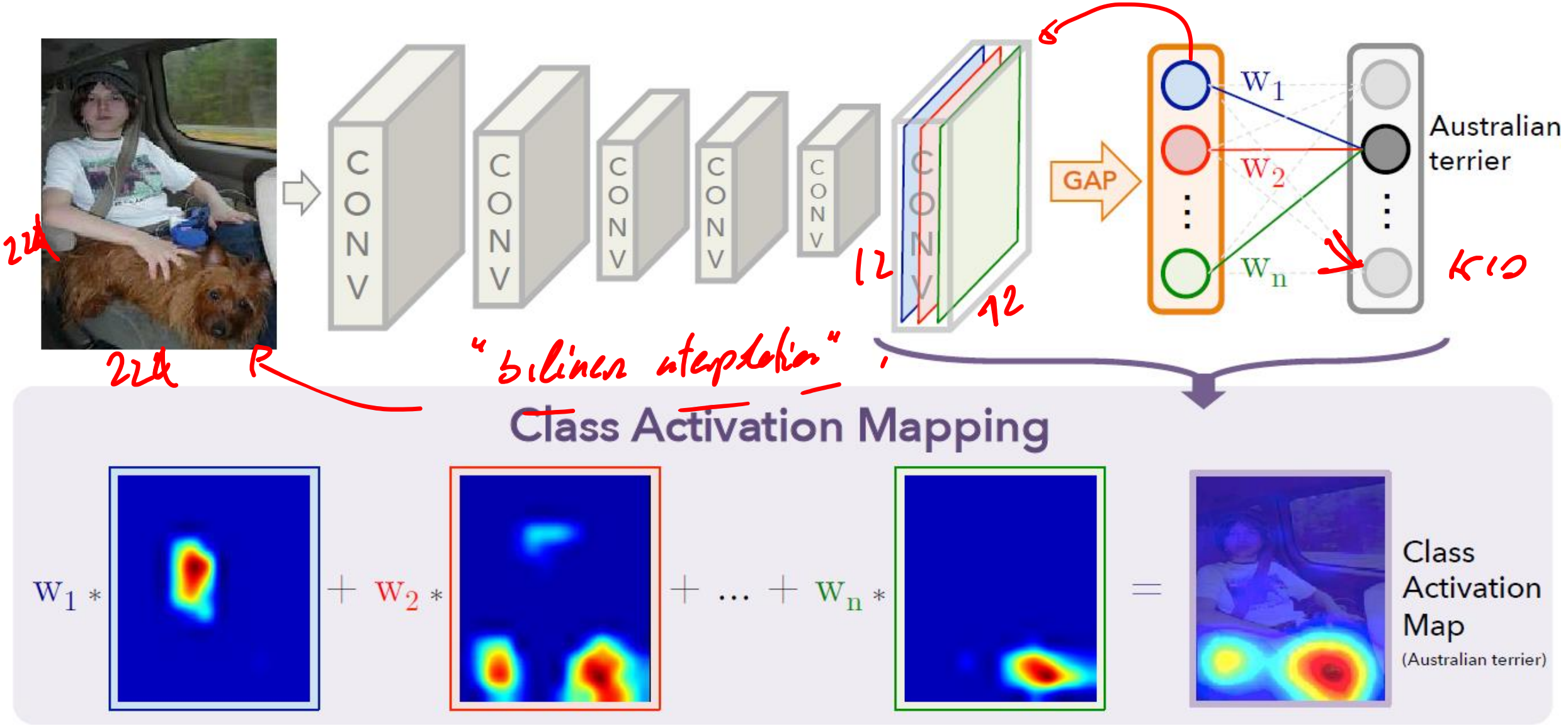
Class Activation Mapping

Now, the weights represents the importance of each feature map to yield the final prediction. Upsampling might be necessary to match the input image



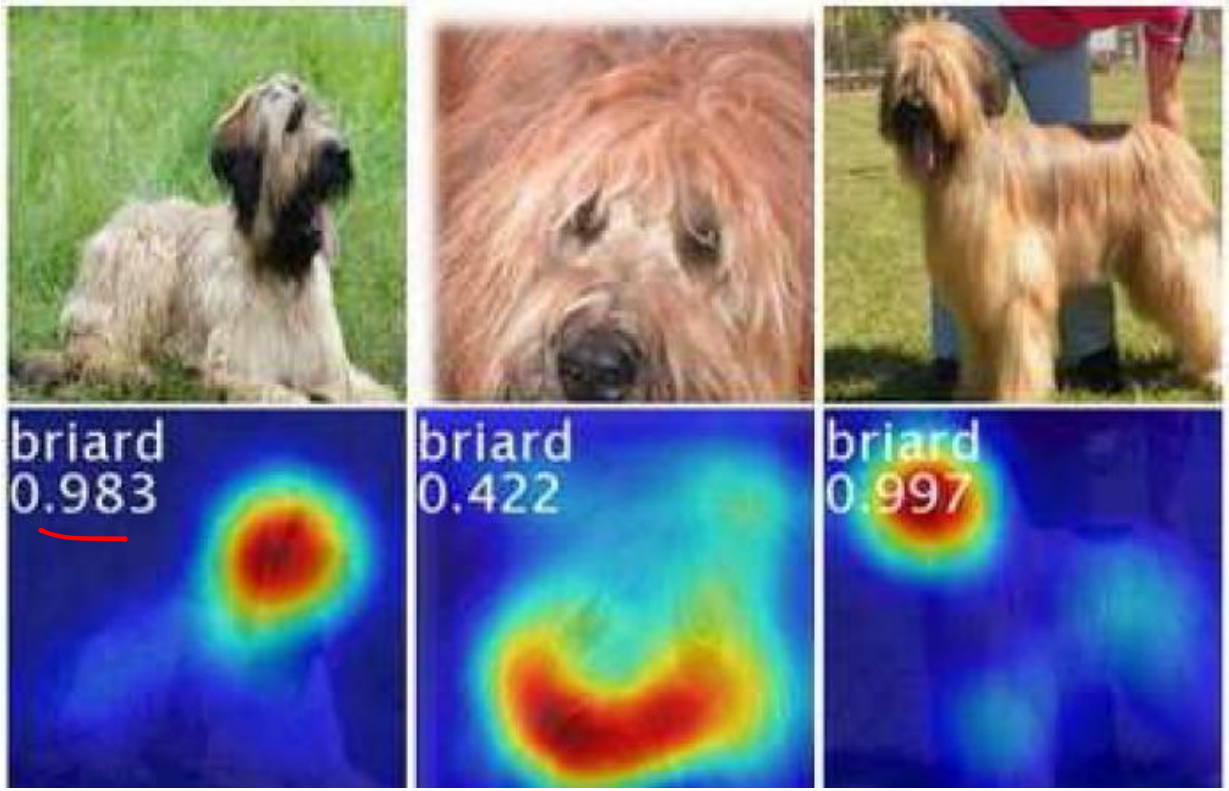
Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

Class Activation Mapping



Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

Class Activation Mapping



Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

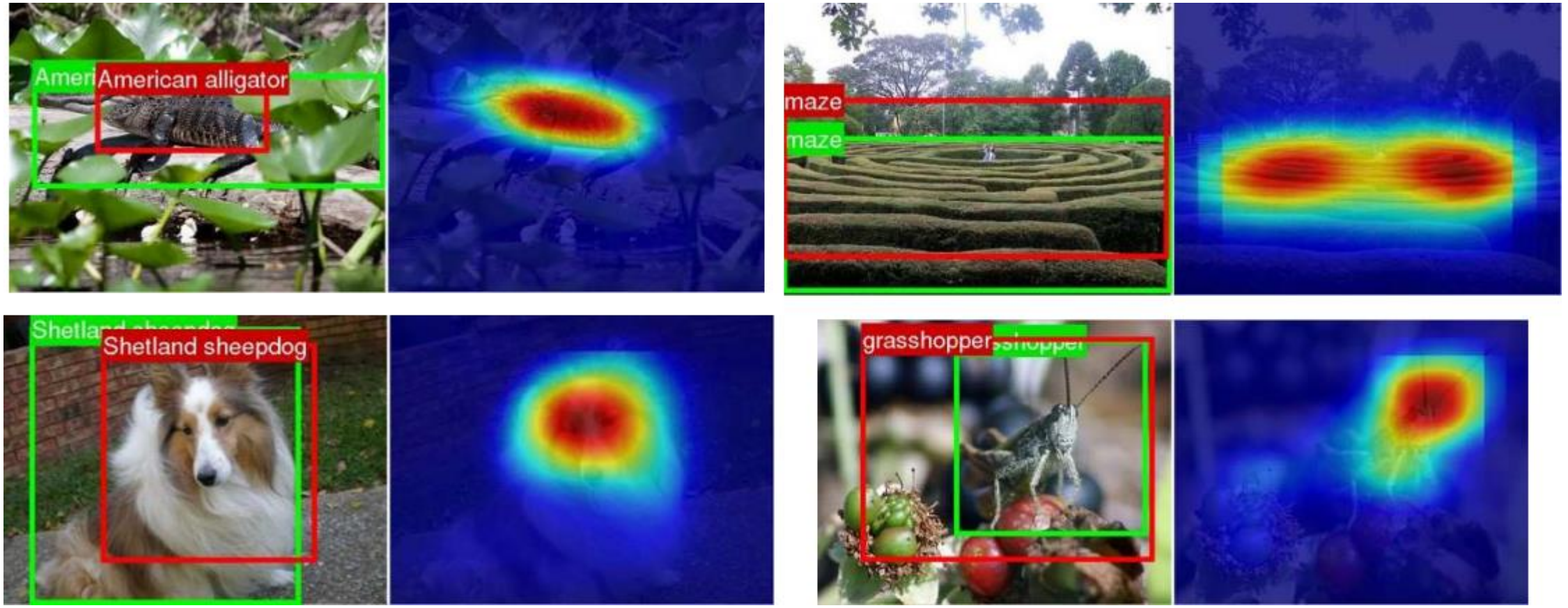
Remarks

- CAM can be **included in any pre-trained network**, as long as all the FC layers at the end are removed
- The **FC used for CAM is simple**, few neurons and no hidden layers
- **Classification performance might drop** (in VGG removing FC means loosing 90% of parameters)
- **CAM resolution** (localization accuracy) can improve by «**anticipating**» GAP to larger convolutional feature maps (but this reduces the semantic information within these layers)
- **GAP**: encourages the **identification of the whole object**, as all the parts of the values in the activation map concurs to the classification
- **GMP (Global Max Pooling):** it is enough to have a high maximum, thus **promotes specific discriminative features**

Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016.

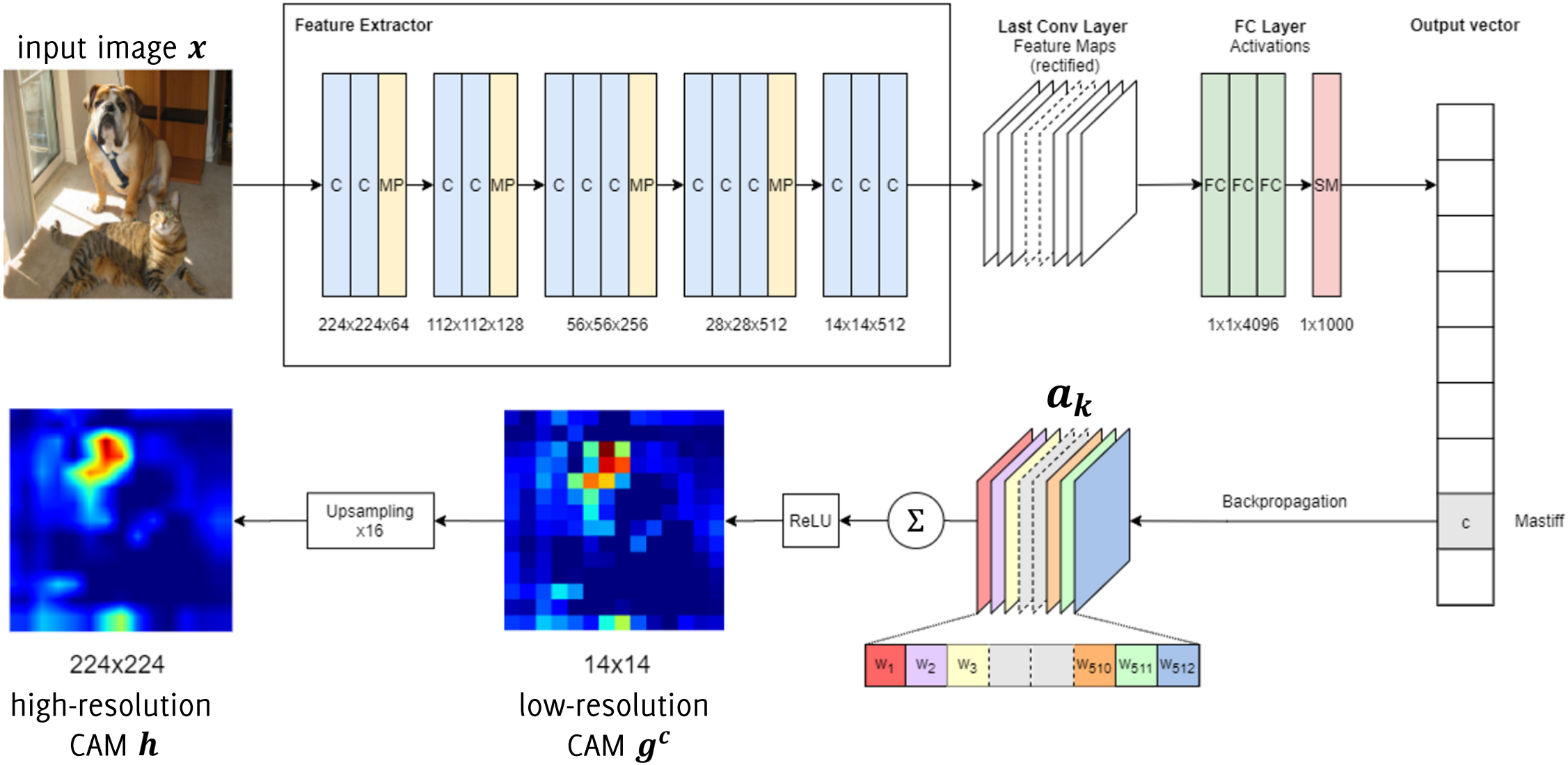
Weakly Supervised Localization

Use thresholding CAM values: > 20% max(CAM), then take the largest componet of the thresholded map (green GT, red estimated location)



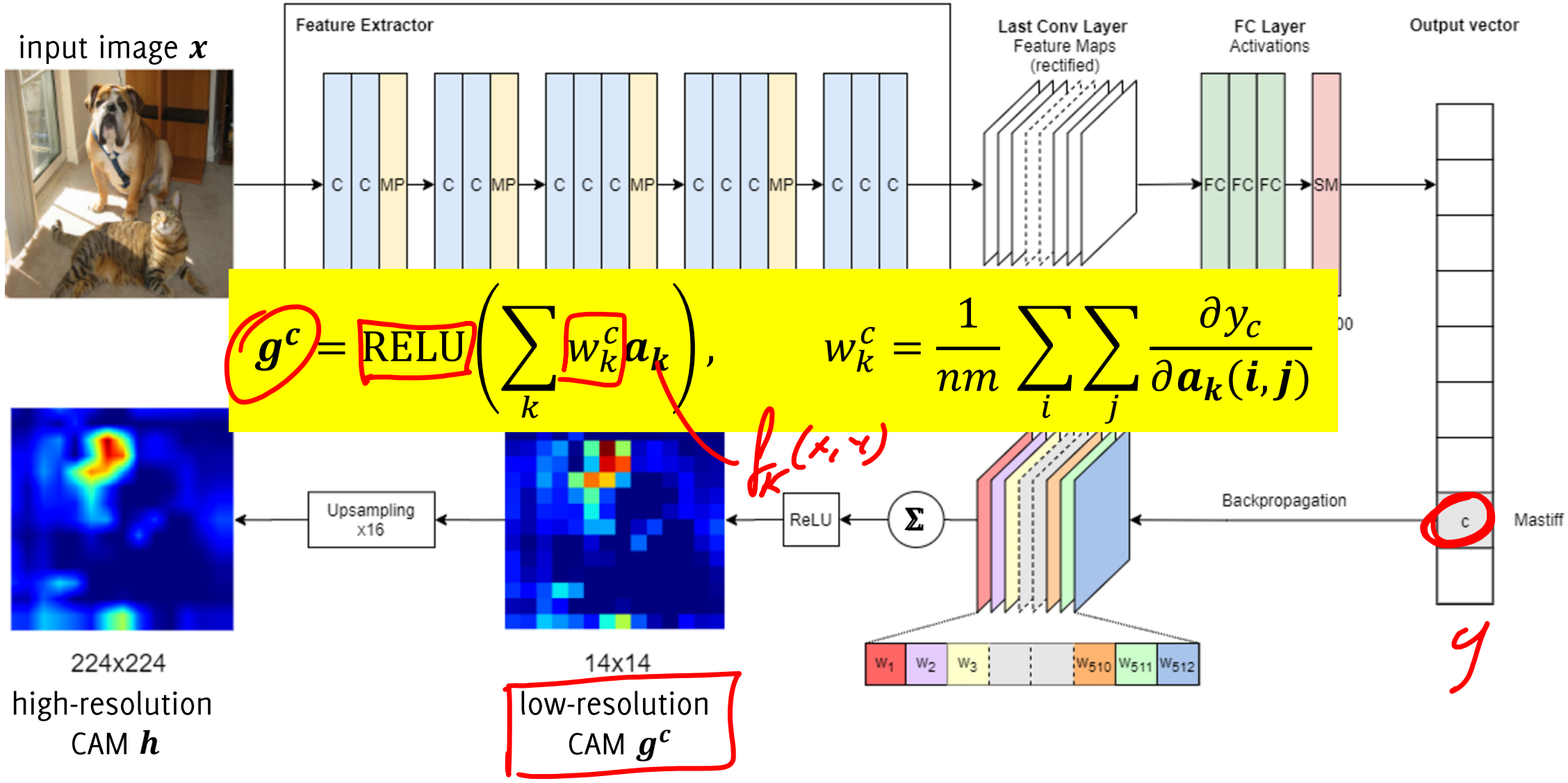
Zhou, Bolei, et al. "Learning deep features for discriminative localization." CVPR 2016

Grad-CAM and CAM-based techniques



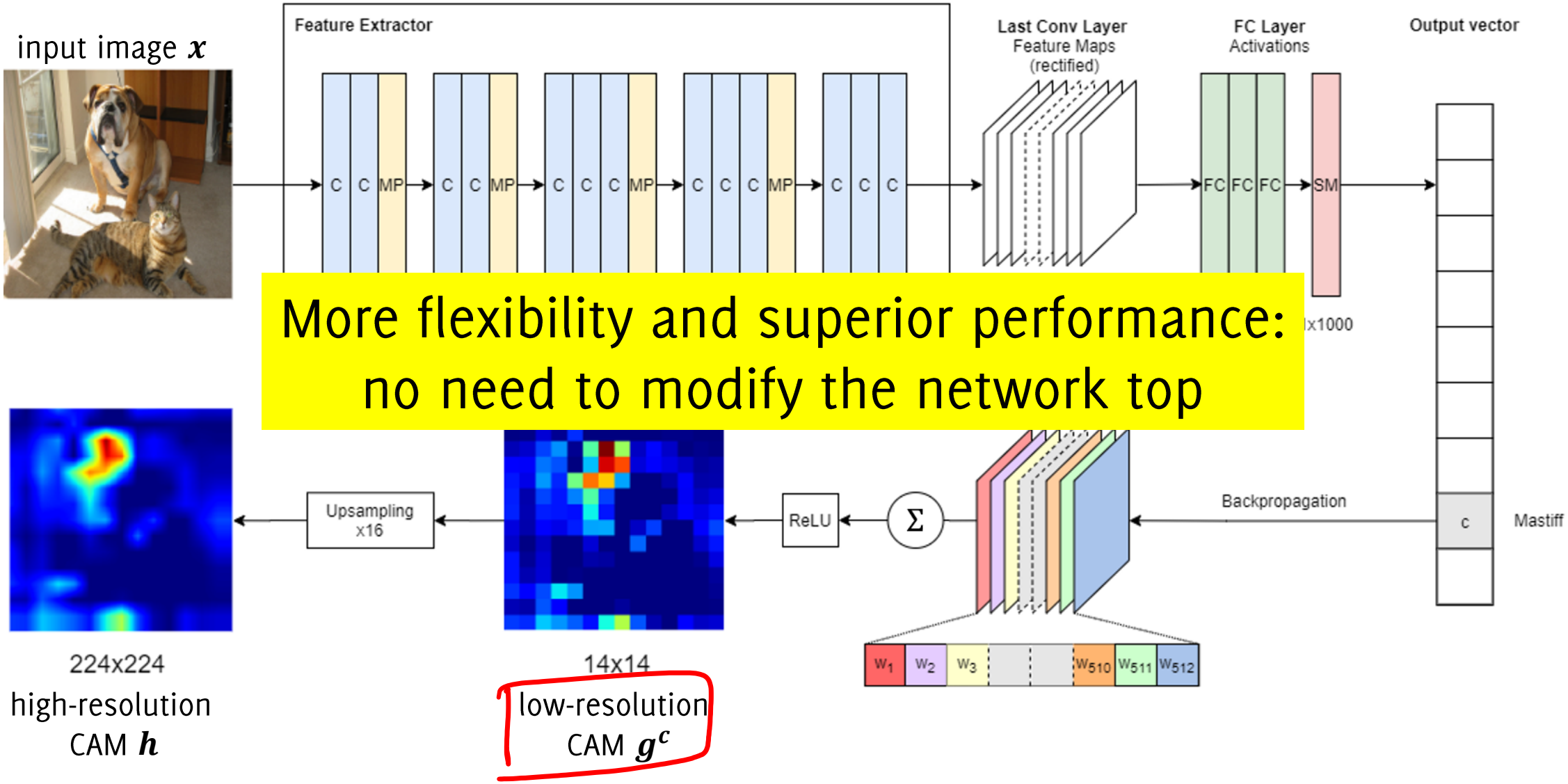
Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV 2017*

Grad-CAM and CAM-based techniques



Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. *ICCV2017*

Grad-CAM and CAM-based techniques



More flexibility and superior performance:
no need to modify the network top

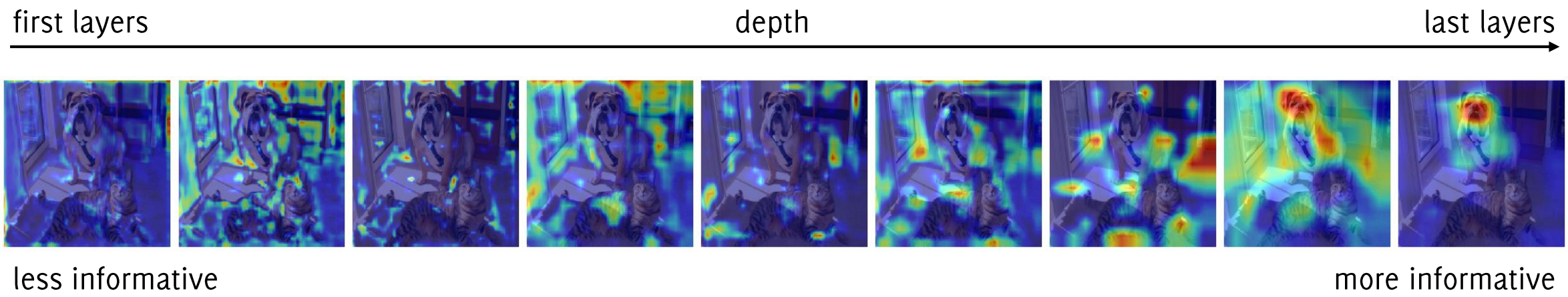
Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. Grad-cam: Visual explanations from deep networks via gradient-based localization. *ICCV2017*

Heatmaps Desiderata

Should be **class discriminative**

Should **capture fine-grained details** (high-resolution)

- This is critical in many applications (e.g. medical/industrial imaging)

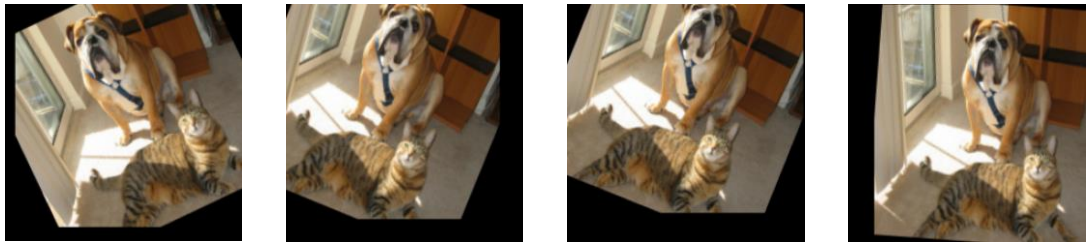


Augmented Grad-CAM

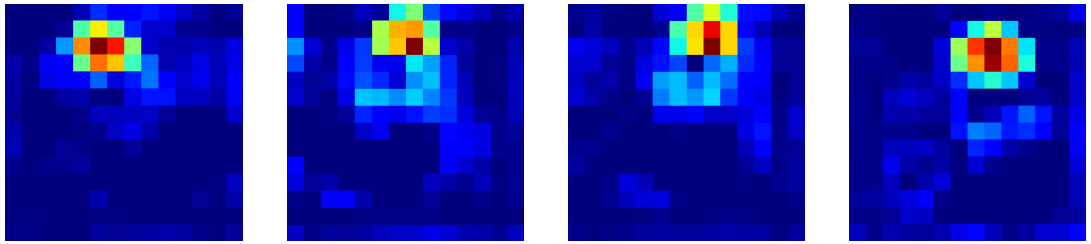
We consider the augmentation operator $\mathcal{A}_l: \mathbb{R}^{N\times M} \rightarrow \mathbb{R}^{N\times M}$, including random rotations and translations of the input image $\boldsymbol{x}$

Augmented Grad-CAM: increase heat-maps resolution through image augmentation

All the responses that the CNN generates to the **multiple augmented versions of the same input image** are very informative for reconstructing the high-resolution heat-map $\boldsymbol{h}$



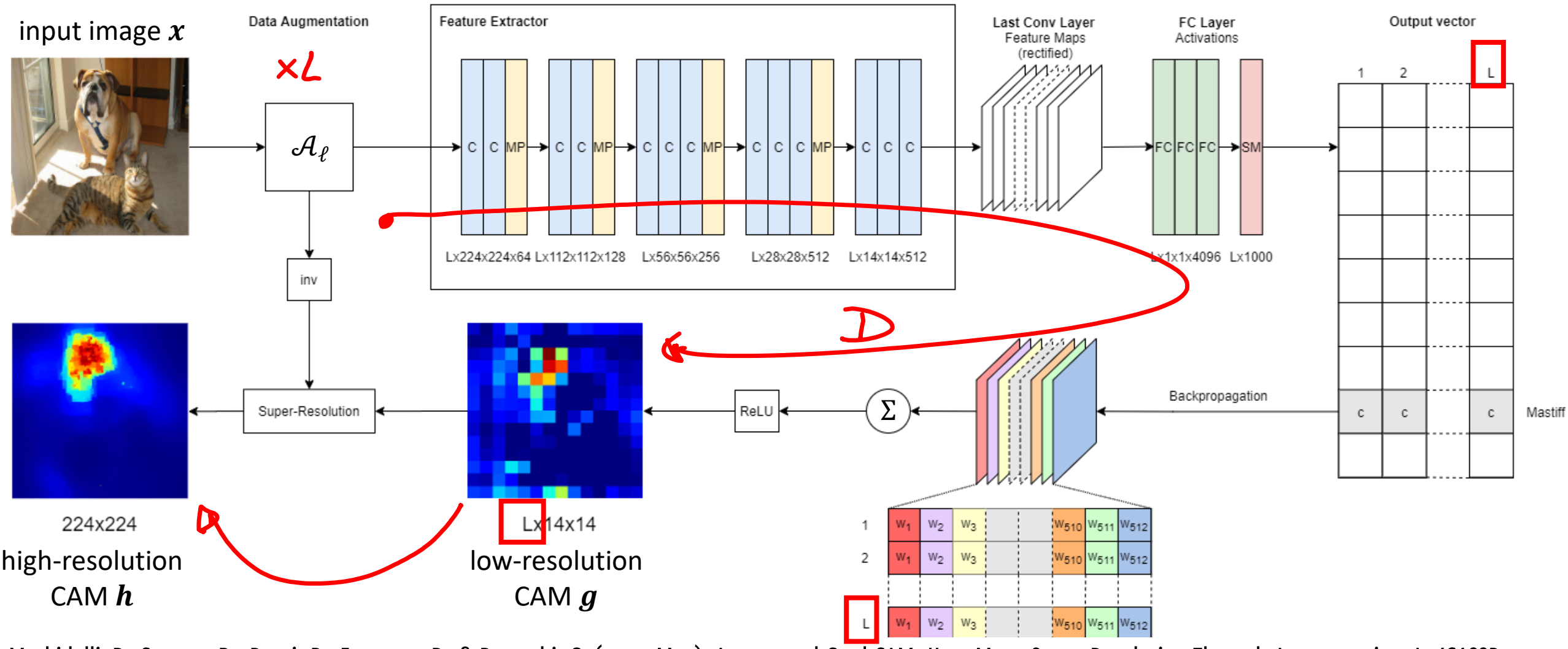
$x_1 = \mathcal{A}_1(x)$ $x_2 = \mathcal{A}_2(x)$ $x_3 = \mathcal{A}_3(x)$ $x_4 = \mathcal{A}_4(x)$



g_1 g_2 g_3 g_4

Morbidelli, P., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G. (2020, May). Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation. In *ICASSP 2020*

Augmented Grad-CAM

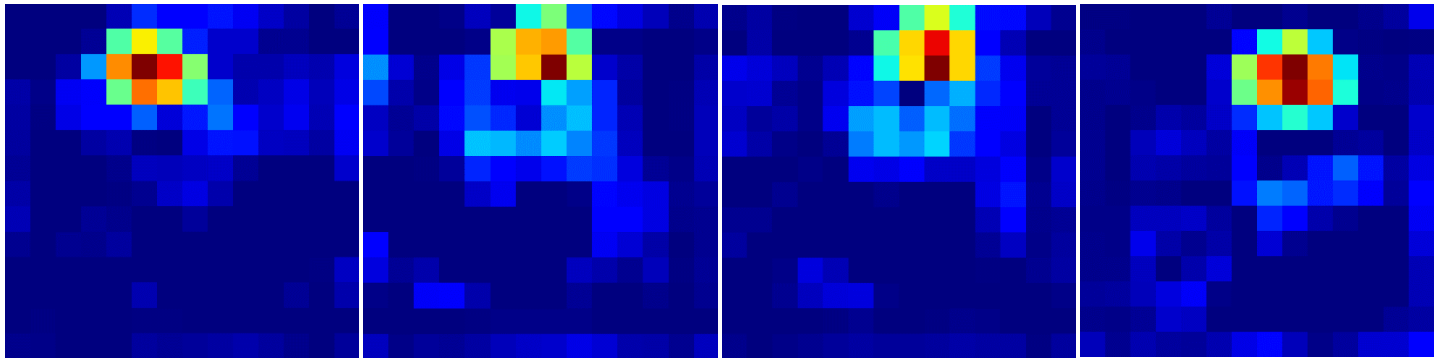


Morbidelli, P., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G. (2020, May). Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation. In *ICASSP 2020*

The Super-Resolution Approach

We perform heat-map Super-Resolution (**SR**) by taking advantage of the information shared in multiple low-resolution heat-maps computed from the **same input under different – but known – transformations**

CNNs are in general invariant to roto-translations, in terms of predictions, but each $\boldsymbol{g}_\ell$ actually contains different information



g_1 g_2 g_3 g_4

General approach, our SR framework can be combined with any visualization tool (not only Grad-CAM)

Morbidelli, P., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G "Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation". ICASSP 2020

The Super-Resolution Formulation

We model heat-maps computed by Grad-CAM as the result of an **unknown downsampling operator** $\mathcal{D} : \mathbb{R}^{N \times M} \rightarrow \mathbb{R}^{n \times m}$

The high-resolution heat-map $\boldsymbol{h}$ is recovered by solving an inverse problem

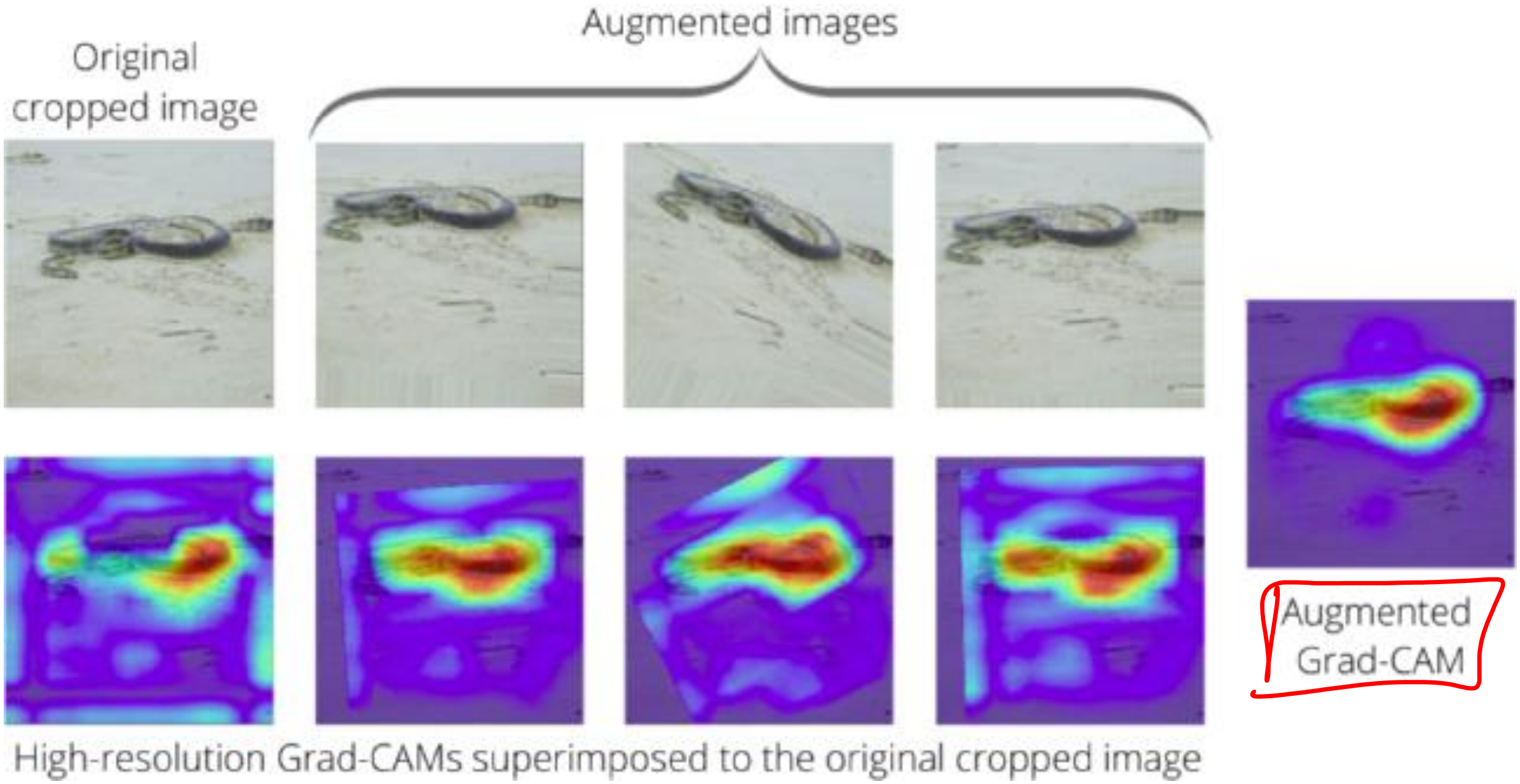
$$\underset{h}{\operatorname{argmin}} \frac{1}{2} \sum_{l=1}^{L} \| \mathcal{D}\mathcal{A}_\ell h - g_\ell \|_2^2 + \lambda TV_{\ell_1}(h) + \frac{\mu}{2} \|h\|_2^2 \quad (1)$$

TV_{ℓ_1}: Anistropic Total Variation regularization is used to preserve the edges in the target heat-map (high-resolution)

$$TV_{\ell_1}(\boldsymbol{h}) = \Sigma_{i,j} \|\partial_x \boldsymbol{h}(i,j)\| + \|\partial_y \boldsymbol{h}(i,j)\| \quad (2)$$

This is solved through Subgradient Descent since the function is convex and non-smooth

Morbidelli, P., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G "Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation". ICASSP 2020



Morbidelli, P., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G "Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation". ICASSP 2020

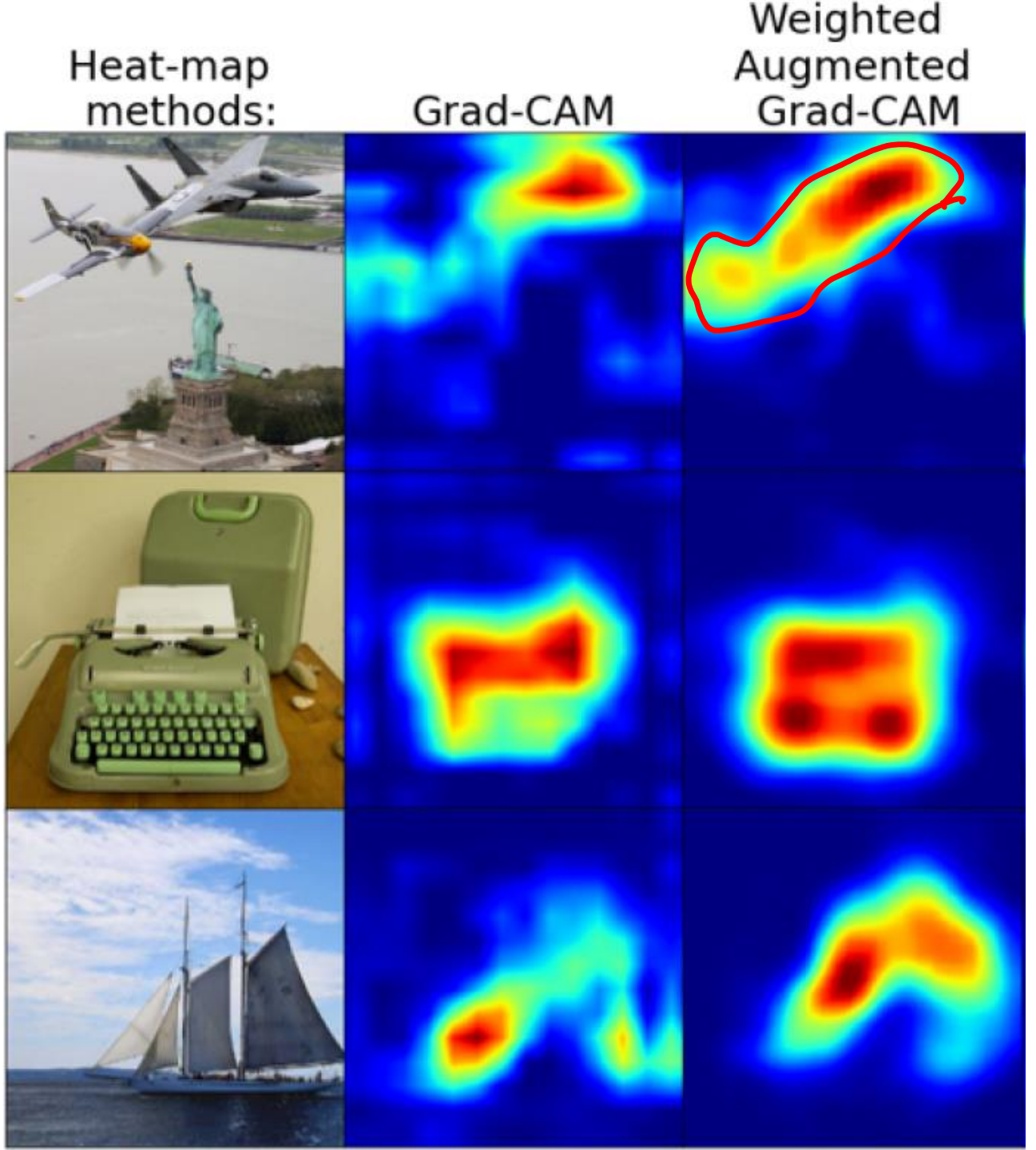
Augmented Grad-CAM

"mash"

(a) Grad-CAM. (b) Augmented Grad-CAM.

Morbidelli, P., Carrera, D., Rossi, B., Fragneto, P., & Boracchi, G. (2020, May). Augmented Grad-CAM: Heat-Maps Super Resolution Through Augmentation. In *ICASSP 2020*

Augmented Grad-CAM



Other Gradient-based Saliency Maps

Grad-CAM++ : Same formulation of CAM as Grad-CAM, but weights are computed by higher-order derivatives of the class score with respect to the feature maps. Increases the localization accuracy of the heat-maps in presence of multiple occurrence of the same object in the image.

Sharpen Focus: highlights only the pixels where the gradients are positive.

$$w_k^c = \frac{1}{nm} \sum_i \sum_j \text{RELU}\left(\frac{\partial y_c}{\partial \boldsymbol{a_k}(\boldsymbol{i}, \boldsymbol{j})}\right)$$

Smooth Grad-CAM++: it averages multiple heat-maps corresponding to noisy versions of the same input image.

A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," WACV, 2018.

D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam, "Smoothgrad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models"

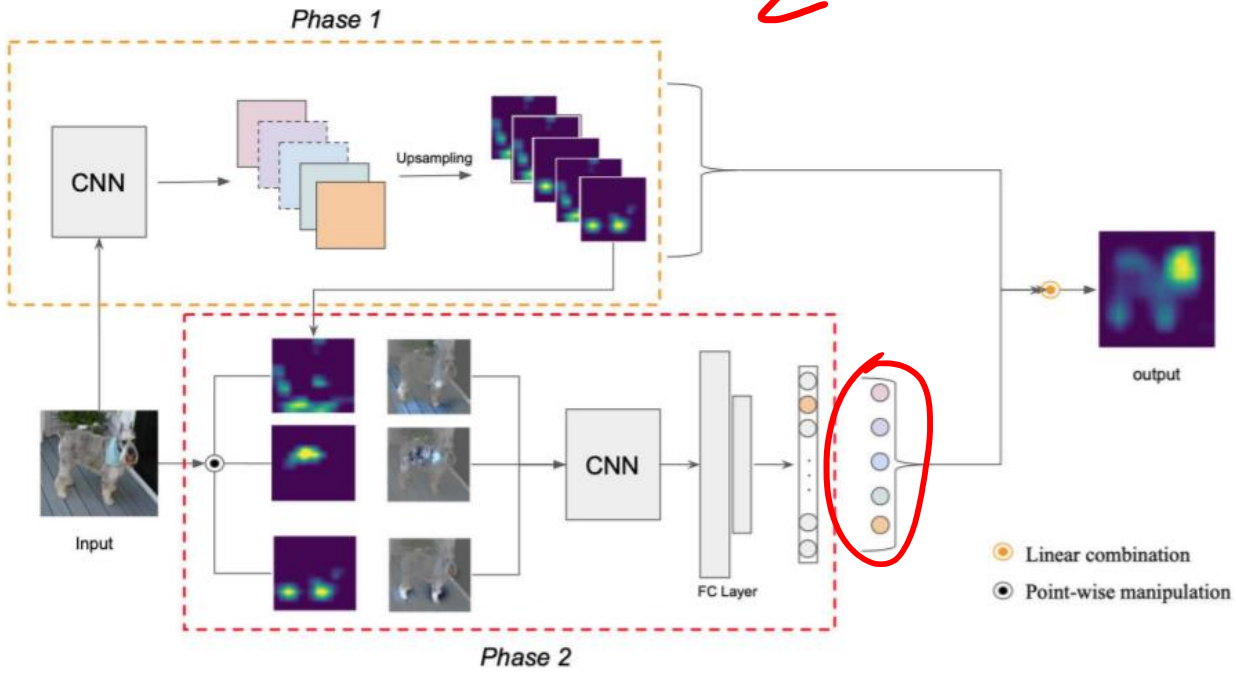
Other Perturbation-based Saliency Maps

Idea: Perturb the input image and assess how the class score changes.

Score-CAM: Each feature map (upsampled and normalized in [0,1]) operates as a mask on the original image, which is then forward-passed to obtain the score on the target class.

$$w_k^c = CNN(\boldsymbol{h_k^c} \circ \boldsymbol{x}) - CNN(\boldsymbol{x})$$

$\sum w_k^c f_k$



H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-cam: Score-weighted visual explanations for convolutional neural networks," 2020

Limitations of Saliency Maps

Figure 1: Based on saliency maps it is unclear why this image is labelled as a *cat* rather than a *laundry basket*. Grad-CAM [27] explanations are essentially the same for both classes.

L. Giulivi M.J. Carman G. Boracchi "Perception Visualization: Seeing Through the Eyes of a DNN" BMVC 2021

Perception Visualization

Perception Visualization: provides explanations by exploiting a neural network to invert latent representations

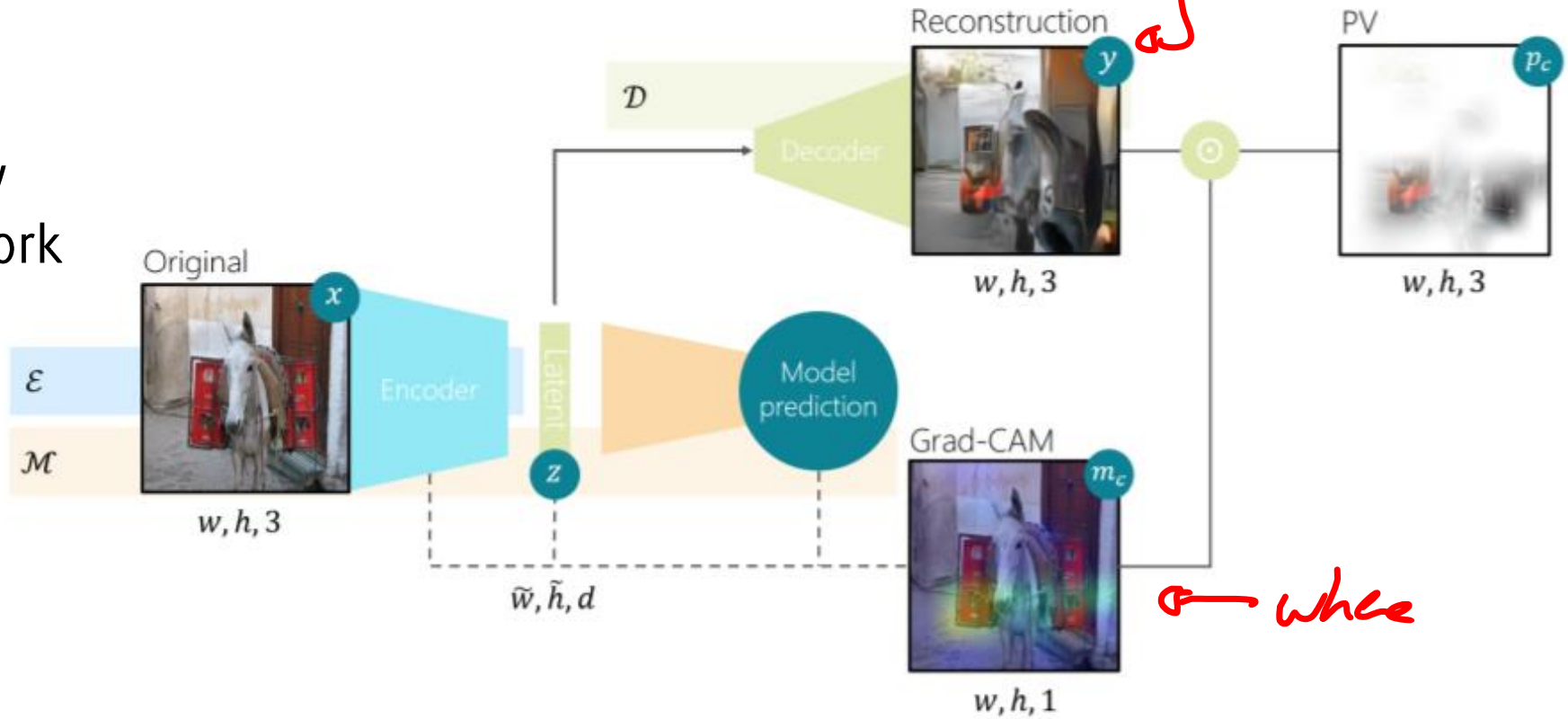


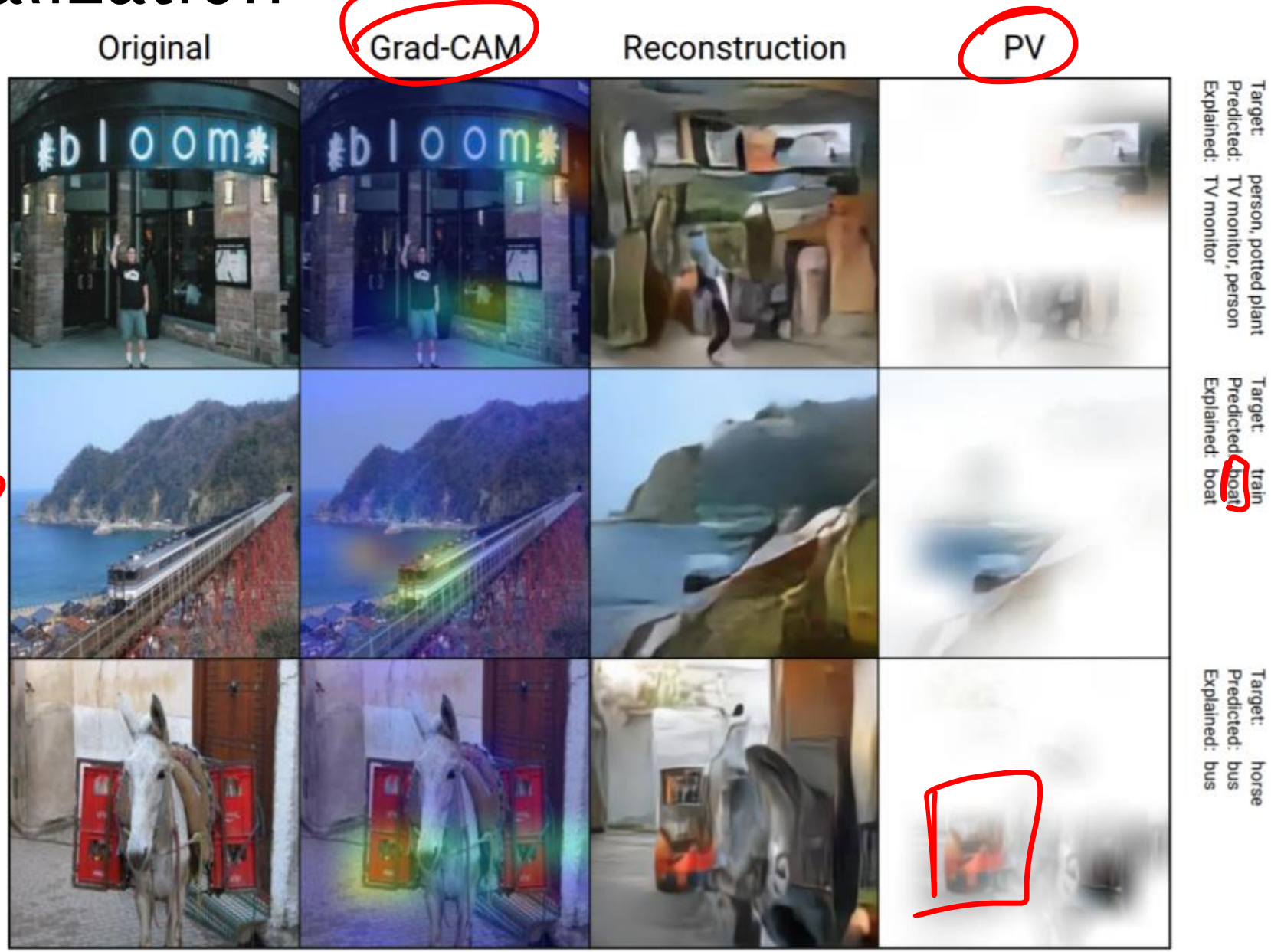
Figure 3: An overview of our method and interactions between the models involved. Encoder $\mathcal{E}$ is a truncation of the model $\mathcal{M}$ which we want to explain, decoder $\mathcal{D}$ is trained to reconstruct the encoder's latent representations. From these, we compute Grad-CAM saliency maps and reconstructions, which are then combined to obtain PV.

L. Giulivi M.J. Carman G. Boracchi "Perception Visualization: Seeing Through the Eyes of a DNN" BMVC 2021

Perception Visualization

Give better insight on the model's functioning than what was previously achievable using only saliency maps.

A study on circa 100 subjects shows that PV is able to **help respondents better determine the predicted class** in cases where the **model had made an error**



L. Giulivi M.J. Carman G. Boracchi "Perception Visualization: Seeing Through the Eyes of a DNN" BMVC 2021