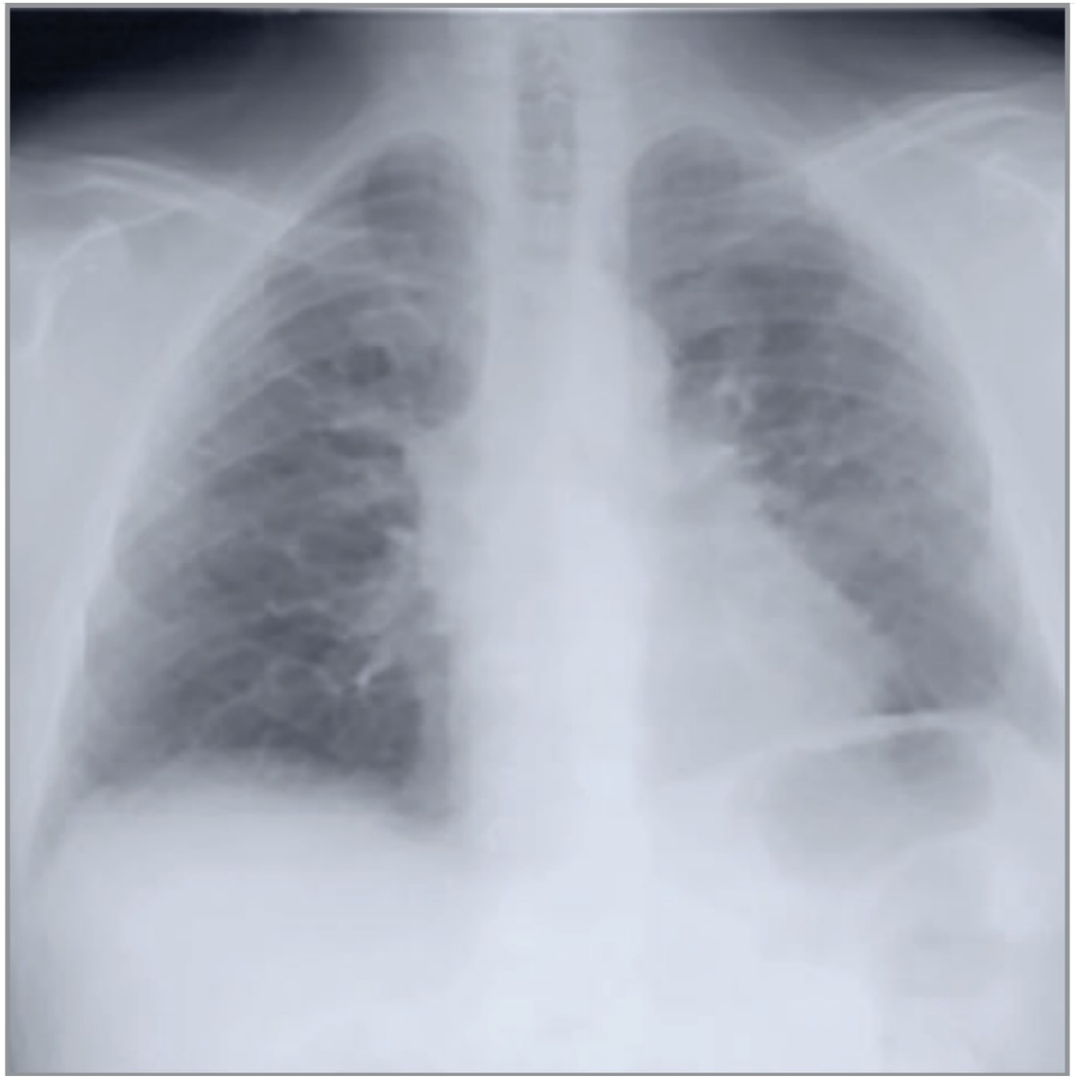
Saliency Maps

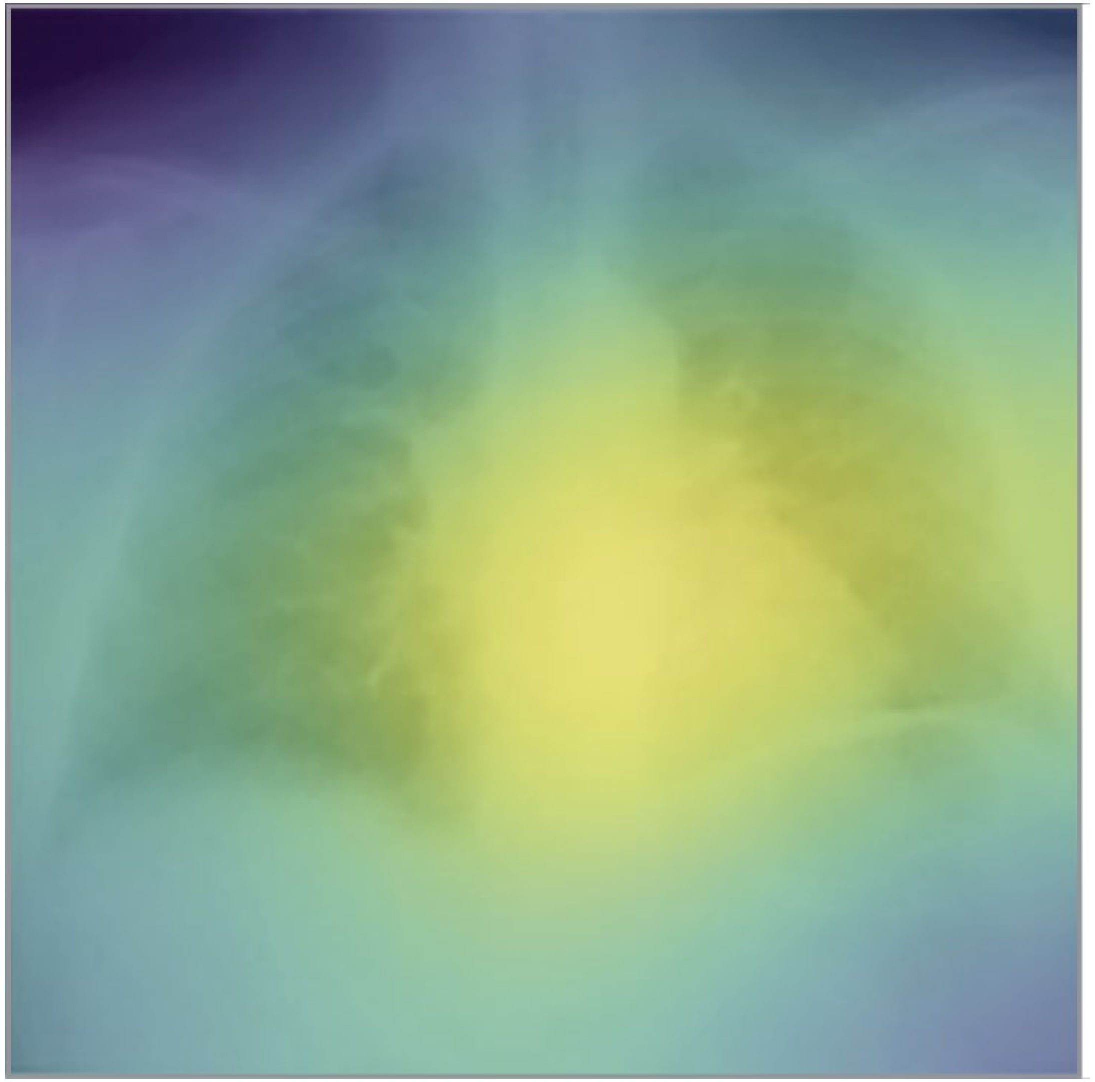
Why saliency maps could extend your life



an X-ray image of the chest from a 60-year-old man

Lu, M. T., Ivanov, A., Mayrhofer, T., Hosny, A., Aerts, H. J., & Hoffmann, U. (2019). Deep learning to assess long-term mortality from chest radiographs. *JAMA network open*, *2*(7), e197416-e197416.

Why saliency maps could extend your life



Saliency map highlights an enlarged heart with prominent pulmonary vasculature indicating pulmonary edema

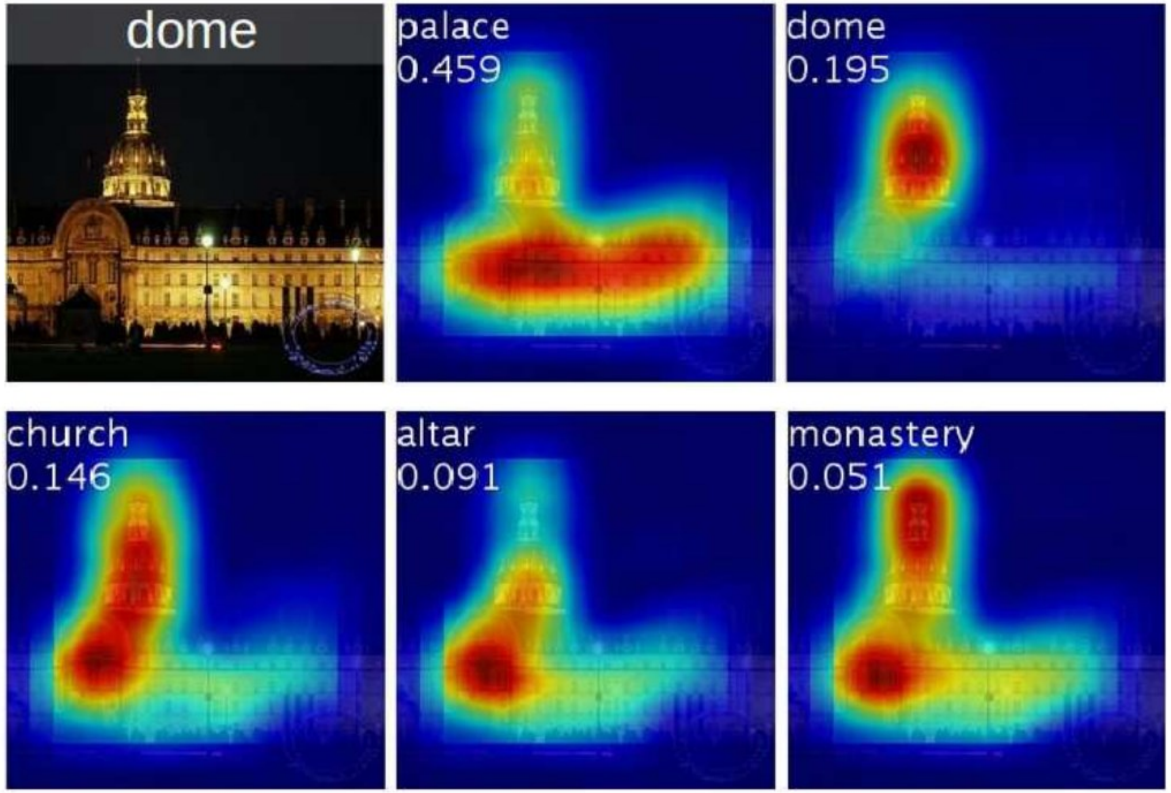
Lu, M. T., Ivanov, A., Mayrhofer, T., Hosny, A., Aerts, H. J., & Hoffmann, U. (2019). Deep learning to assess long-term mortality from chest radiographs. *JAMA network open*, *2*(7), e197416-e197416.

Structure of the presentation

- Examples of different saliency maps
- Metrics to evaluate the accuracy saliency maps
- Different methods to generate CAMs
- Recommendation which methods should be used
- Where I used saliency maps so far

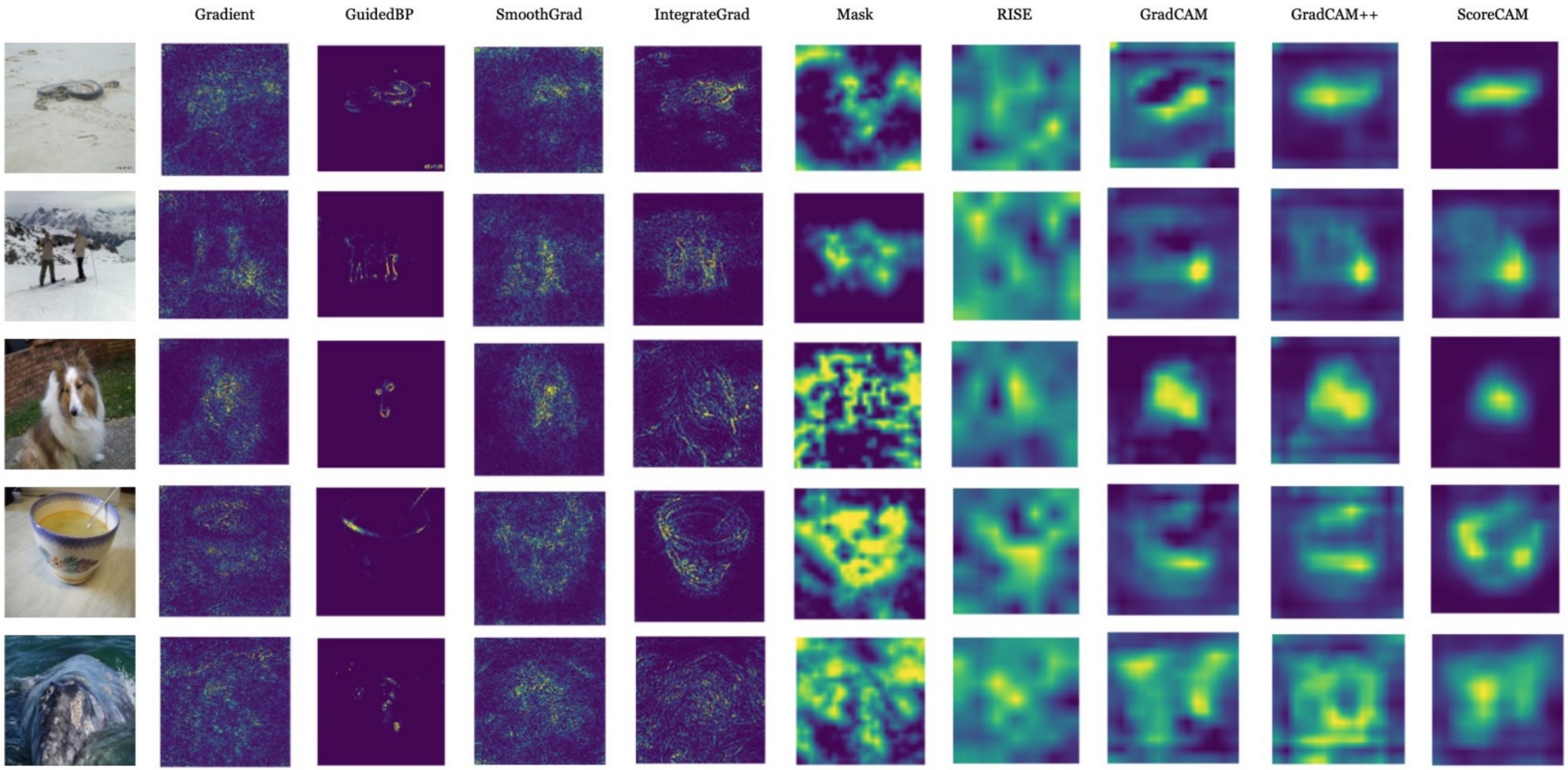
Class activation maps (CAMs)

- Is a simple technique to get the discriminative image regions used by a CNN to identify a specific class in the image.



Source: Learning Deep Features for Discriminative Localization

Different kinds of saliency maps

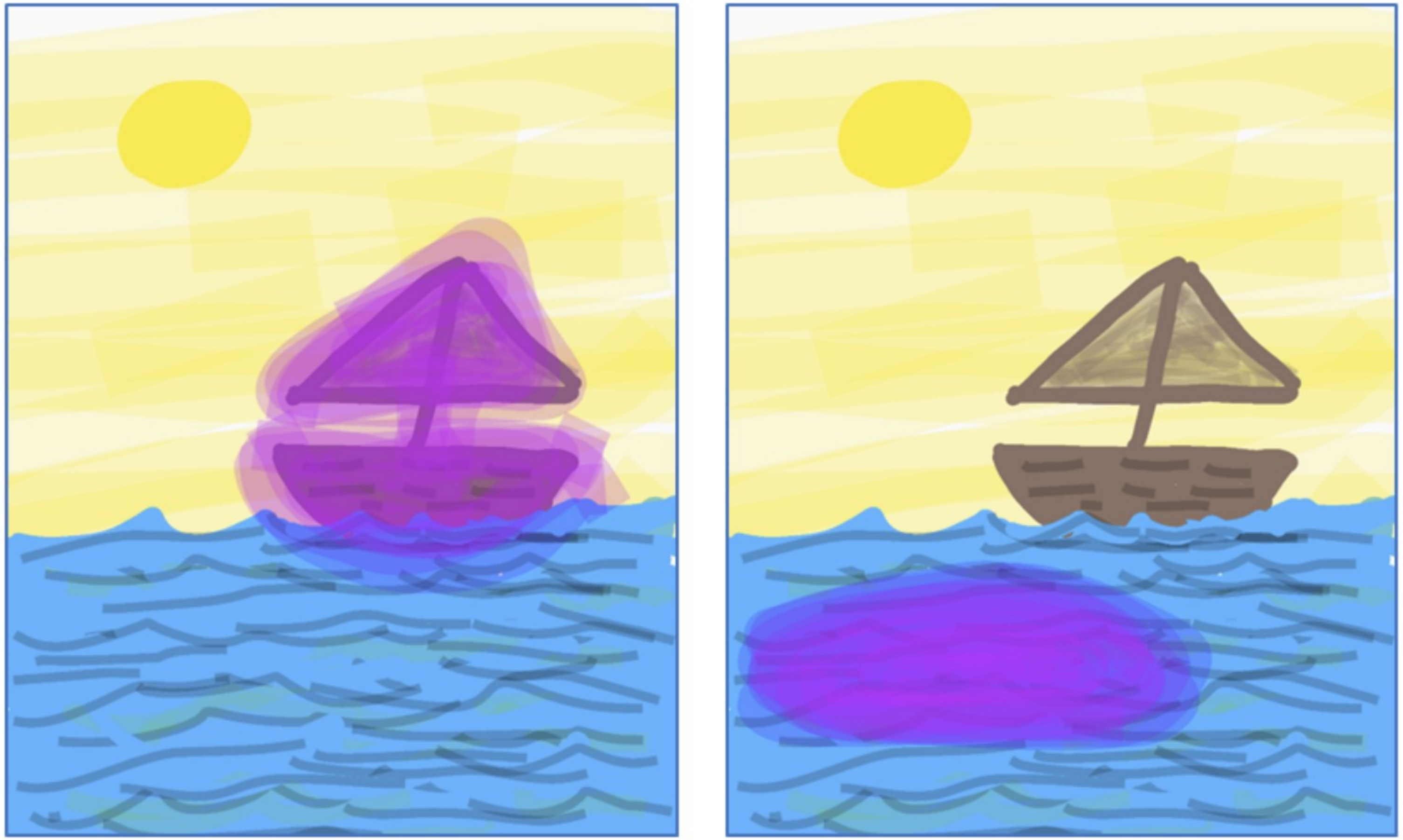


Weakly object detection with saliency maps

- Use trained convolutional neural networks (mostly trained for image classification) to detect objects

1. Select some images from a dataset for object detection
2. Feed these images into a trained network
3. Create the saliency maps
4. Extract the region of interest (ROI) where the object could be located according to a certain threshold

Model Explanation is Not Weakly Supervised Segmentation

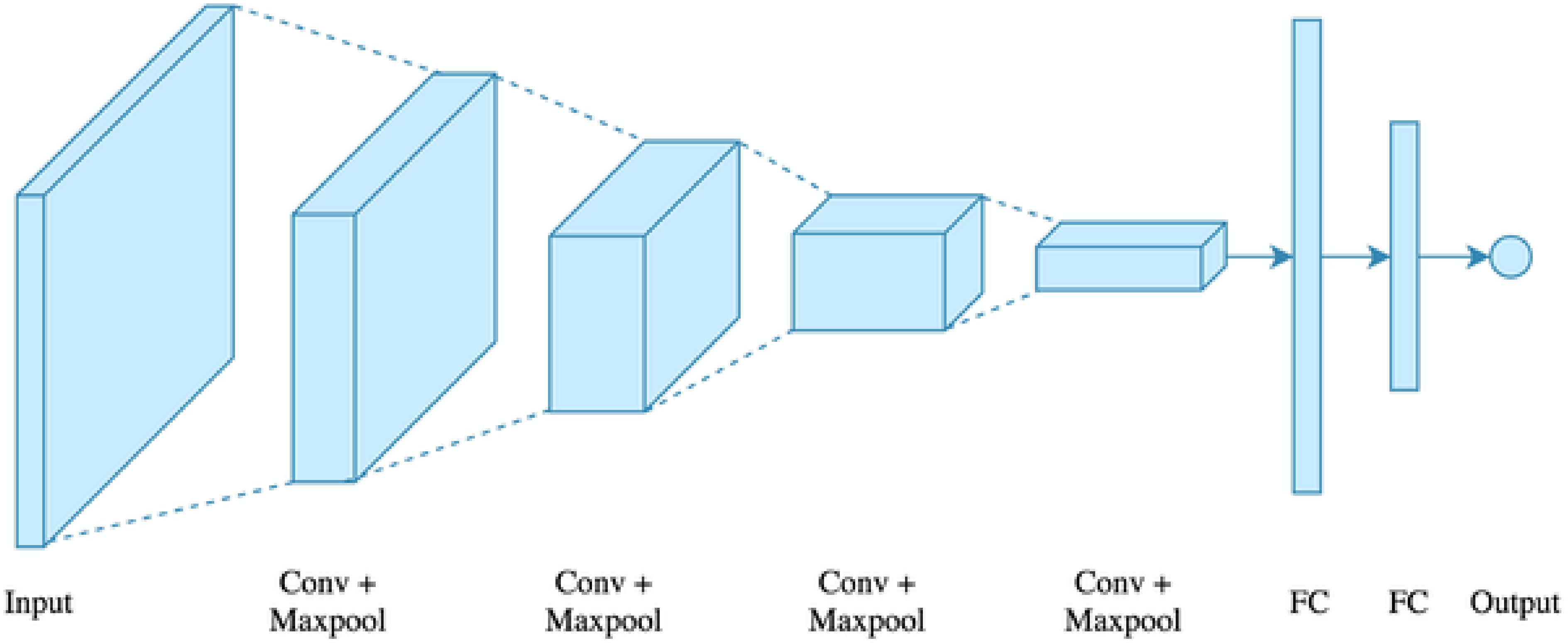


https://glassboxmedicine.com/2022/07/02/model-explanation-is-not-weakly-supervised-segmentation/

Metrics to compare saliency maps

- Weakly object detection
- **Insertion metric:** measures how fast the model output increases when adding the salient image pixels to a baseline image (higher score is better)
- **Deletion metric:** measures the decrease of the output when removing salient image pixels. (lower score is better)
- Use humans to evaluate the generated heatmaps (but the humans have no information about the network weights!)

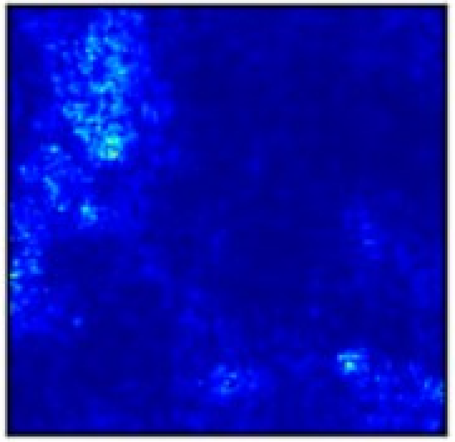
CNN architectures

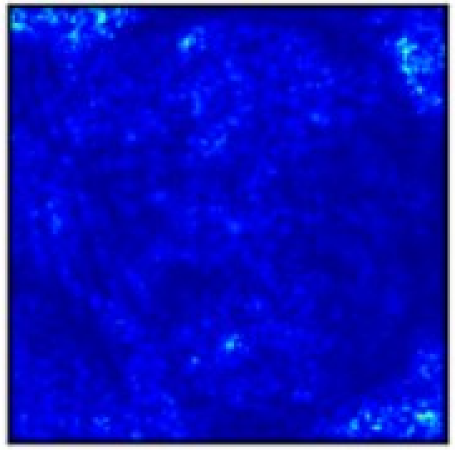


Source: https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

Pure Gradients

- Calculate the derivative of the predicted score for a certain class with respect to the given image.
- For an RGB image we get three derivatives for each position (take the max value)
- High gradients indicate high sensitivity (a small change in this area leads to a huge change in the output)
- Often only the positive gradients are used (apply ReLU)



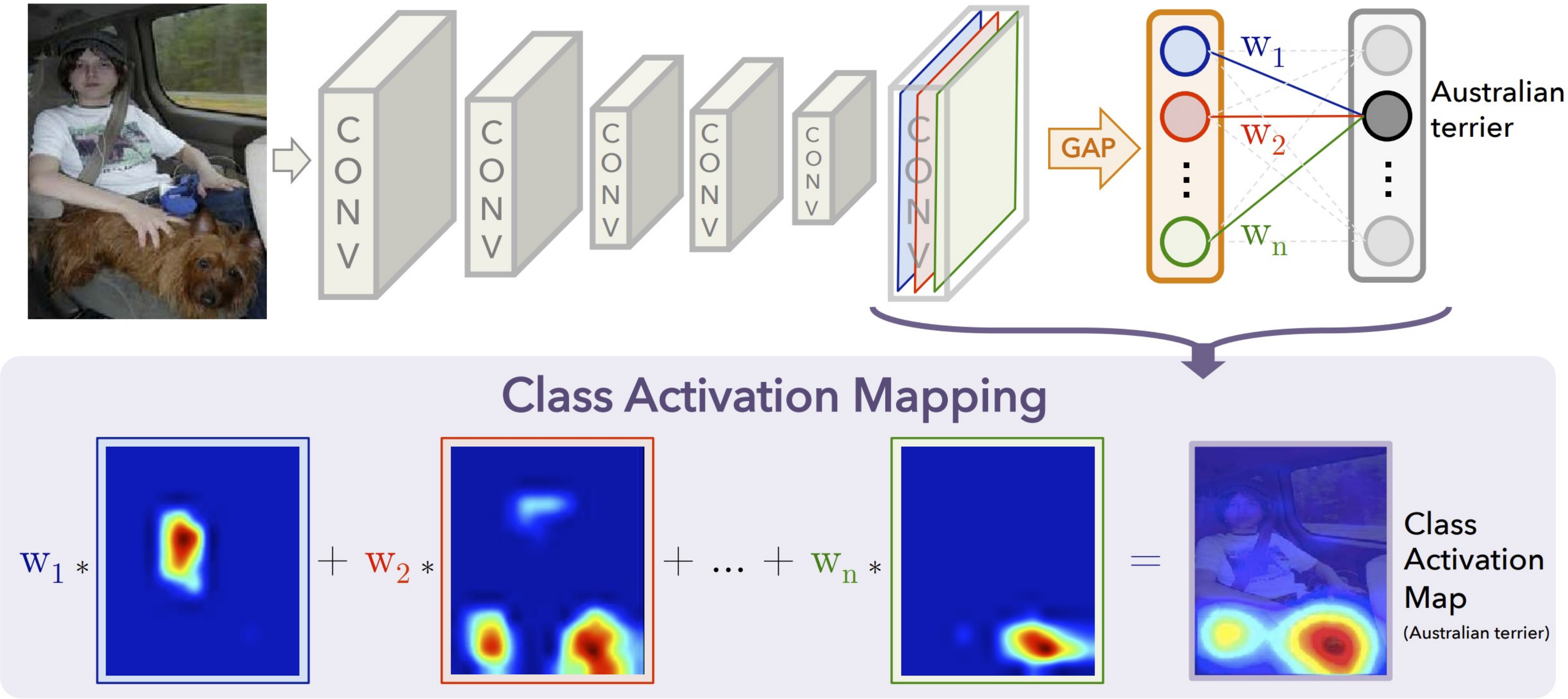


Pros and Cons of pure gradients

- Pros:
  - The CAM has the same resolution as the input no upscaling has to be done
  - Post hoc method (no retraining has to be carried out)

- Cons:
  - Gradients can be noisy and can vanish due to the saturation problem

CAM



Source: Learning Deep Features for Discriminative Localization

CAM

1. Delete all hidden layers from the network add set weights of the Conv-layers (trainable=False)
2. Add a new hidden layer right after the last Conv-layer which is a global average pooling layer which has no trainable weights.
3. Add a hidden layer with the same number of neurons as feature maps in the last Conv-layer
4. Retrain the new network
5. Feed the image into the network and calculate the weighted average of the resulting outputs of the Conv-layer according to the trained weights
6. Apply the ReLU function to get positive values

Formula for CAM

- The CAM for the class c can be written as:

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

Where $f_k(x, y)$ is the activation of unit k in the last Conv-Layer

Problems with CAM

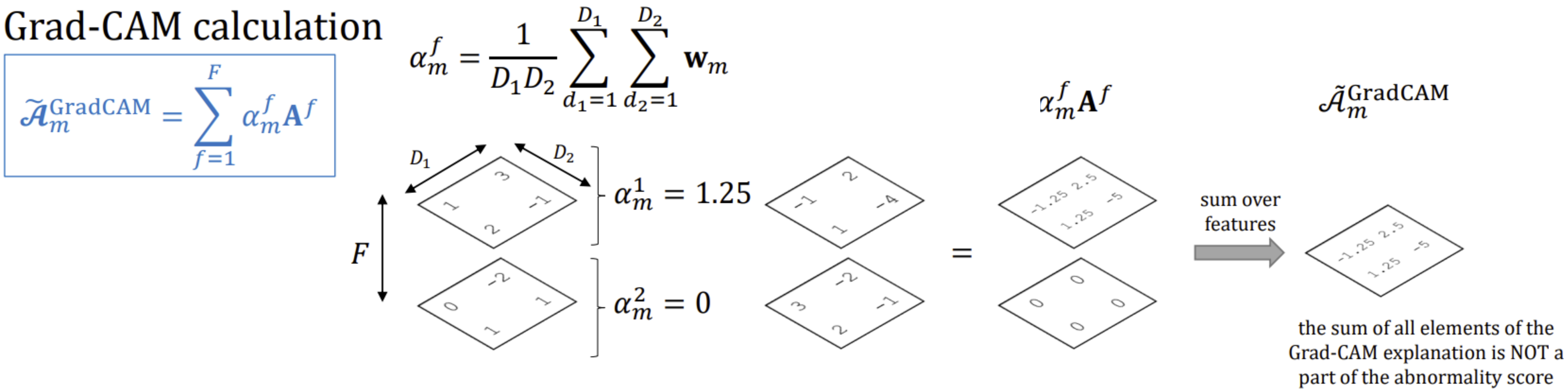
- New network must be trained to get a direct mapping from the features to the actual classes
- The GAP of the feature maps leads to huge loss of information.
- Also, Max Pooling was tested but did not work as well as the GAP-Pooling
- The resolution of the CAM has the size of the last hidden layer

GRAD-CAM (the better CAM)

- Make the class activation maps more accurate and more general (no retraining is needed)
- Calculate the weights using the partial derivatives of the output of the specific class c with respect to the activation map

$$w_k^c = \frac{1}{D_1 \cdot D_2} \sum_{d_1=1}^{D_1} \sum_{d_1=1}^{D_1} \frac{\partial y^c}{\partial f_k(d_1, d_2)}$$

GRAD-CAM



Grad-CAM calculation

$$\boxed{\tilde{\mathcal{A}}_m^{\text{GradCAM}} = \sum_{f=1}^{F} \alpha_m^f \mathbf{A}^f}$$

$$\alpha_m^f = \frac{1}{D_1 D_2} \sum_{d_1=1}^{D_1} \sum_{d_2=1}^{D_2} \mathbf{w}_m$$

the sum of all elements of the Grad-CAM explanation is NOT a part of the abnormality score

Source: Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks

Pros and Cons of Grad-Cam

- Pros:
  - No retraining is needed
- Cons:
  - As a side effect of the gradient averaging step, Grad-CAM sometimes highlights locations which the model did not actually use

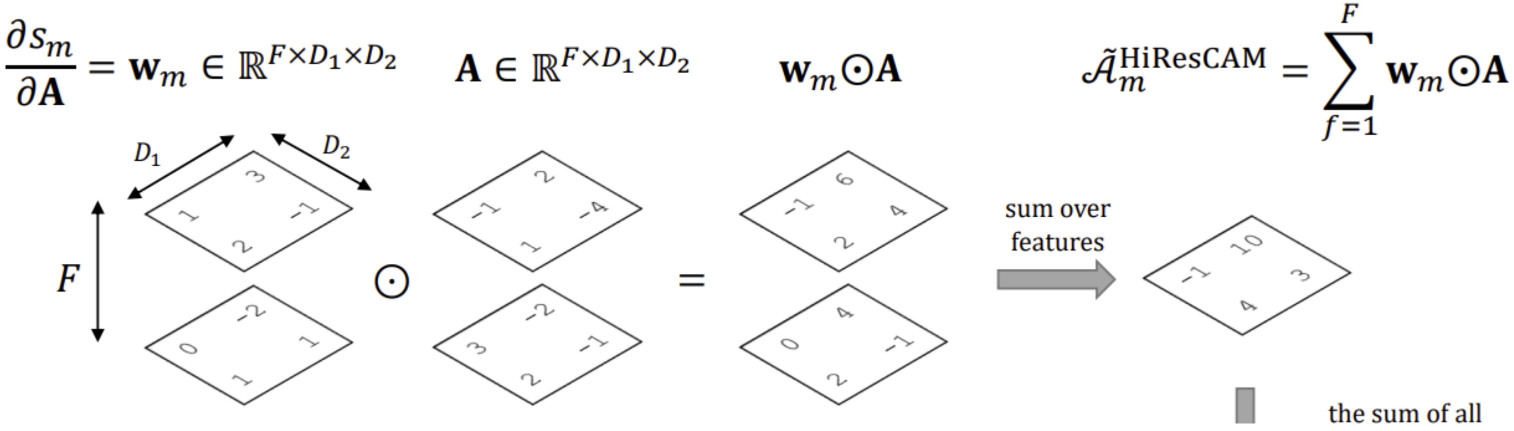
HiRes-CAM (the better Grad-Cam)

- Paper “Use HiRes-CAM instead of Grad-CAM for faithful explanations of convolutional neural networks”
- Grad-CAM without the GAP
- Multiplication of the features with the gradients before the summation

HiResCAM

HiResCAM calculation

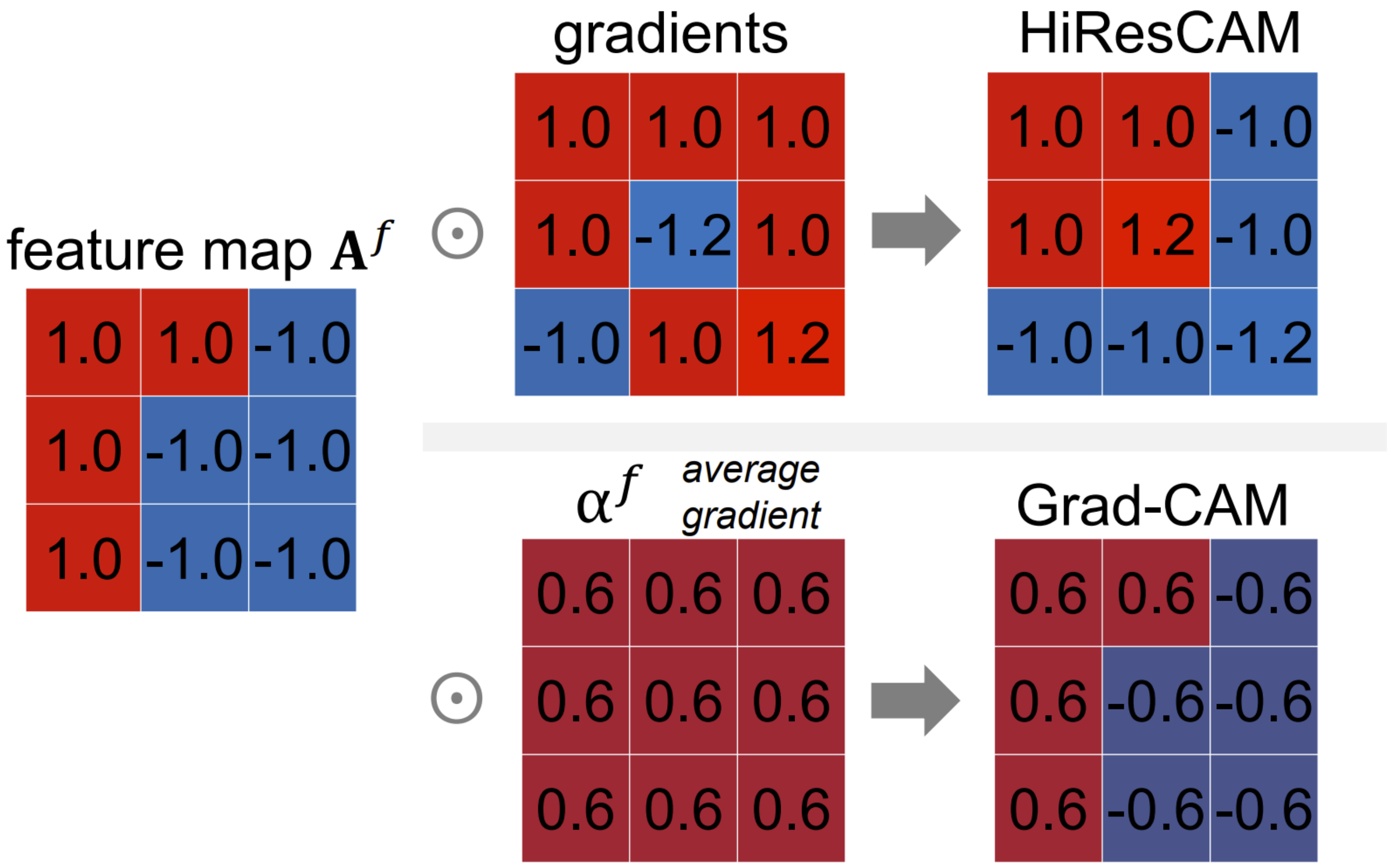
$$\tilde{\mathcal{A}}_m^{\text{HiResCAM}} = \sum_{f=1}^{F} \frac{\partial s_m}{\partial \mathbf{A}^f} \odot \mathbf{A}^f$$



$\frac{\partial s_m}{\partial \mathbf{A}} = \mathbf{w}_m \in \mathbb{R}^{F \times D_1 \times D_2}$
$\mathbf{A} \in \mathbb{R}^{F \times D_1 \times D_2}$
$\mathbf{w}_m \odot \mathbf{A}$
$\tilde{\mathcal{A}}_m^{\text{HiResCAM}} = \sum_{f=1}^{F} \mathbf{w}_m \odot \mathbf{A}$
D_1
D_2
F
3 1 −1 2
−2 0 1 1
⊙
2 −1 −4 1
−2 3 −1 2
=
6 −1 4 2
4 0 −1 2
sum over features
10 −1 3 4
the sum of all

Source: Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks

HiResCam vs Grad-CAM



feature map $\mathbf{A}^f$

1.0	1.0	-1.0
1.0	-1.0	-1.0
1.0	-1.0	-1.0

$\odot$ gradients

1.0	1.0	1.0
1.0	-1.2	1.0
-1.0	1.0	1.2

→ HiResCAM

1.0	1.0	-1.0
1.0	1.2	-1.0
-1.0	-1.0	-1.2

$\odot$ α^f *average gradient*

0.6	0.6	0.6
0.6	0.6	0.6
0.6	0.6	0.6

→ Grad-CAM

0.6	0.6	-0.6
0.6	-0.6	-0.6
0.6	-0.6	-0.6

Source: Use HiResCAM instead of Grad-CAM for faithful explanations of convolutional neural networks

Layer-CAM

- Almost the same as HiRes-CAM
- The HiRes-CAM and Layer-Cam papers do not mention the existence of each other
- The only difference is that a ReLU is applied before multiplying the gradient with the feature map:

1. $w_{ij}^{kc} = relu(g_{ij}^{kc})$ 2. $\hat{A}_{ij}^{k} = w_{ij}^{kc} \cdot A_{ij}^{k}$ 3. $M^c = \mathrm{ReLU}\left(\sum_k \hat{A}^k\right)$

Grad-Cam++

- More general Grad-CAM
- ReLU function is used to only consider positive gradients
- Similar to Grad-CAM With Positive Gradients

$$w_k^c = \sum_i \sum_j \alpha_{ij}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ij}^k})$$

$$Y^c = \sum_k [\sum_i \sum_j \{\sum_a \sum_b \alpha_{ab}^{kc} . relu(\frac{\partial Y^c}{\partial A_{ab}^k})\} A_{ij}^k]$$

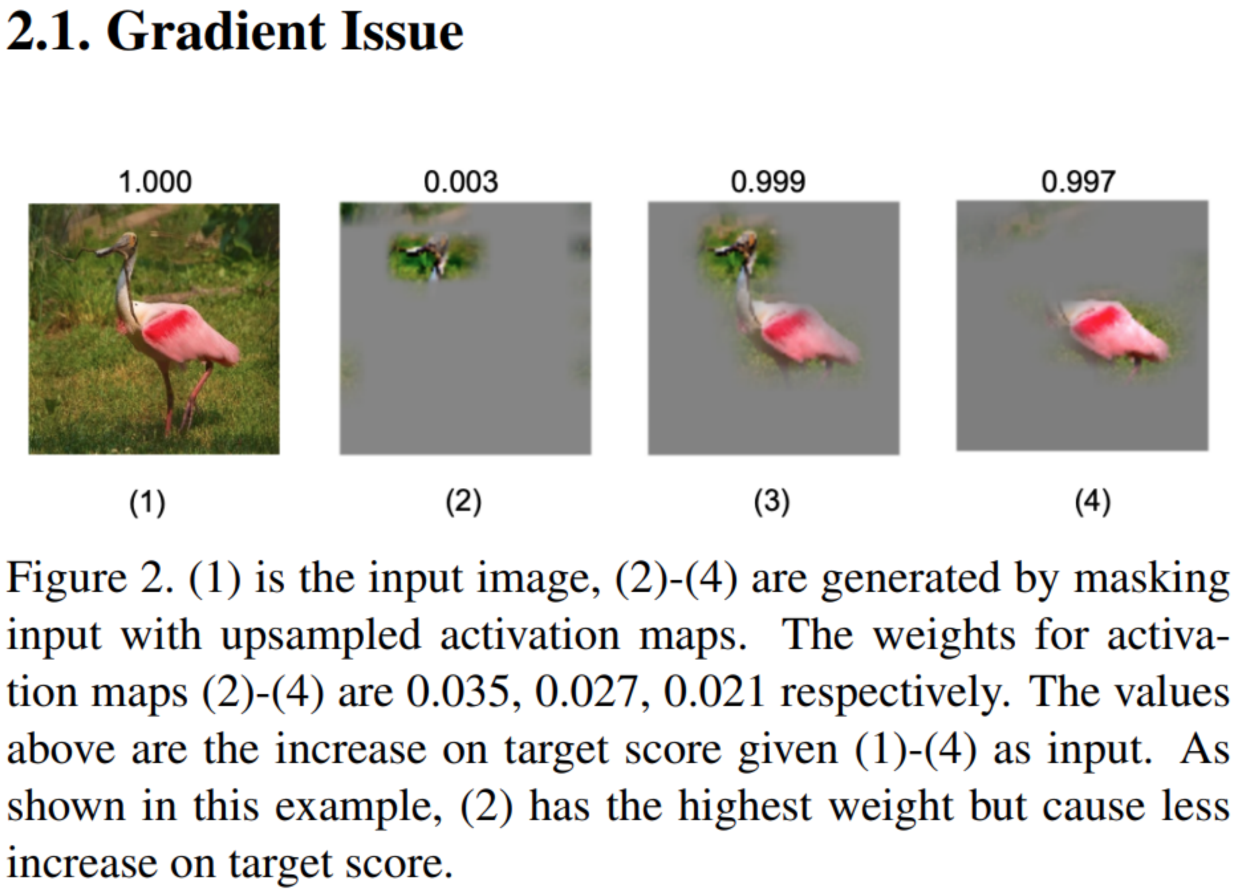
$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 Y^c}{(\partial A_{ij}^k)^2} + \sum_a \sum_b A_{ab}^k \{\frac{\partial^3 Y^c}{(\partial A_{ij}^k)^3}\}}$$

Guided methods

- Used to get sharper saliency maps
- Propagating the error back for every positive input, only propagate back positive error signals.
- The gradient is guided not only by the input, but also by the error signal
- Multiply the up-sampled CAM with the calculated gradients

→ This was a widely used method. But it was shown that this procedure could also mask irrelevant areas on the image. More on this topic comes later.

Do we always have to use gradients?



2.1. Gradient Issue

1.000
0.003
0.999
0.997
(1)
(2)
(3)
(4)

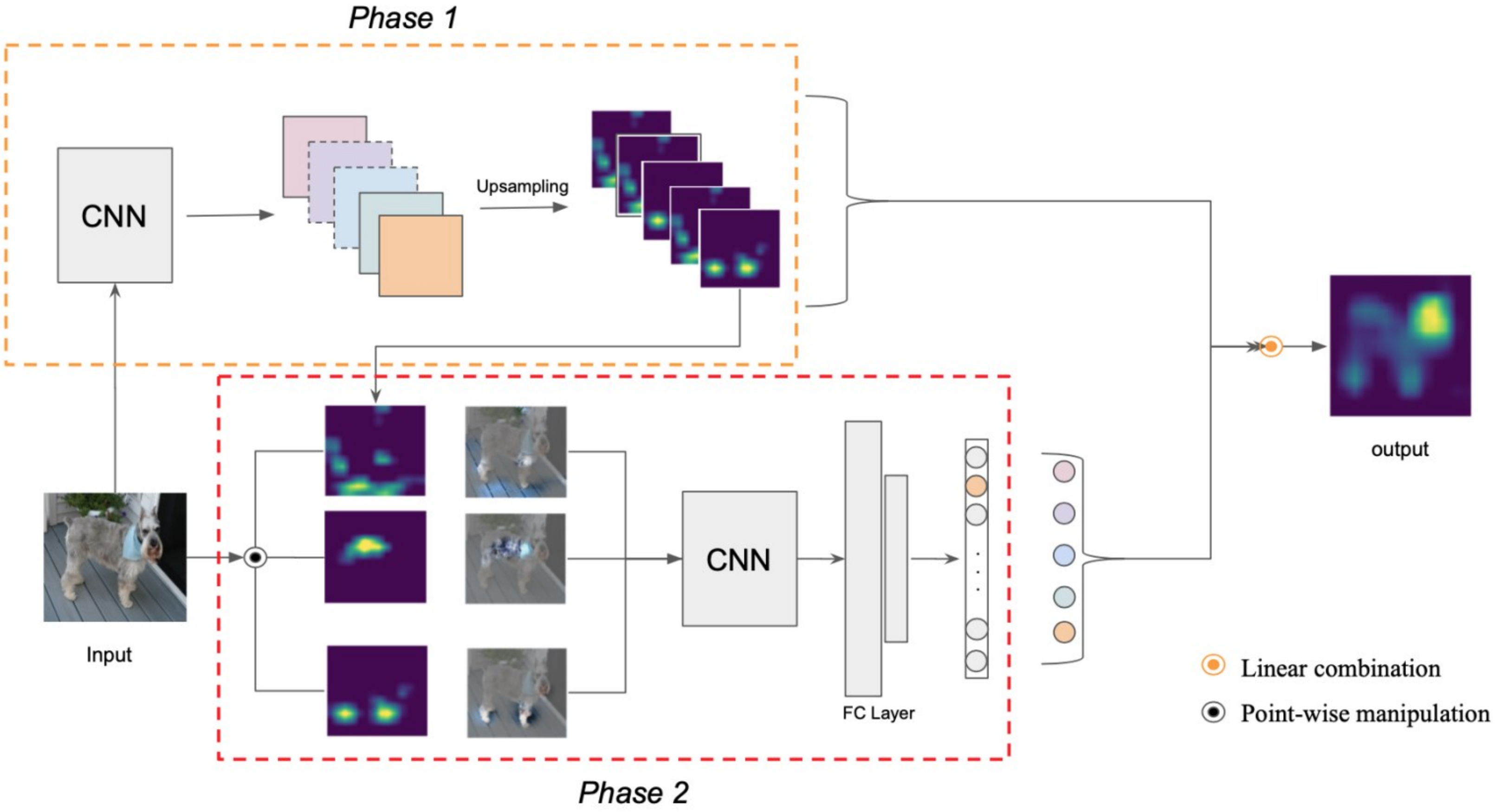
Figure 2. (1) is the input image, (2)-(4) are generated by masking input with upsampled activation maps. The weights for activation maps (2)-(4) are 0.035, 0.027, 0.021 respectively. The values above are the increase on target score given (1)-(4) as input. As shown in this example, (2) has the highest weight but cause less increase on target score.

Source: Score-Weighted Visual Explanations for Convolutional Neural Networks

Score-CAM (Gradient free visual explanation)

- Gradients can be noisy and can vanish due to the saturation problem
- As in all other CAM-methods a weighting for the activation maps of a certain Conv-Layer has to be calculated

How Score-CAM works



Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Networks

Channel-wise Increase of Confidence CIC

- Calculate the channel-wise increase of confidence CIC:

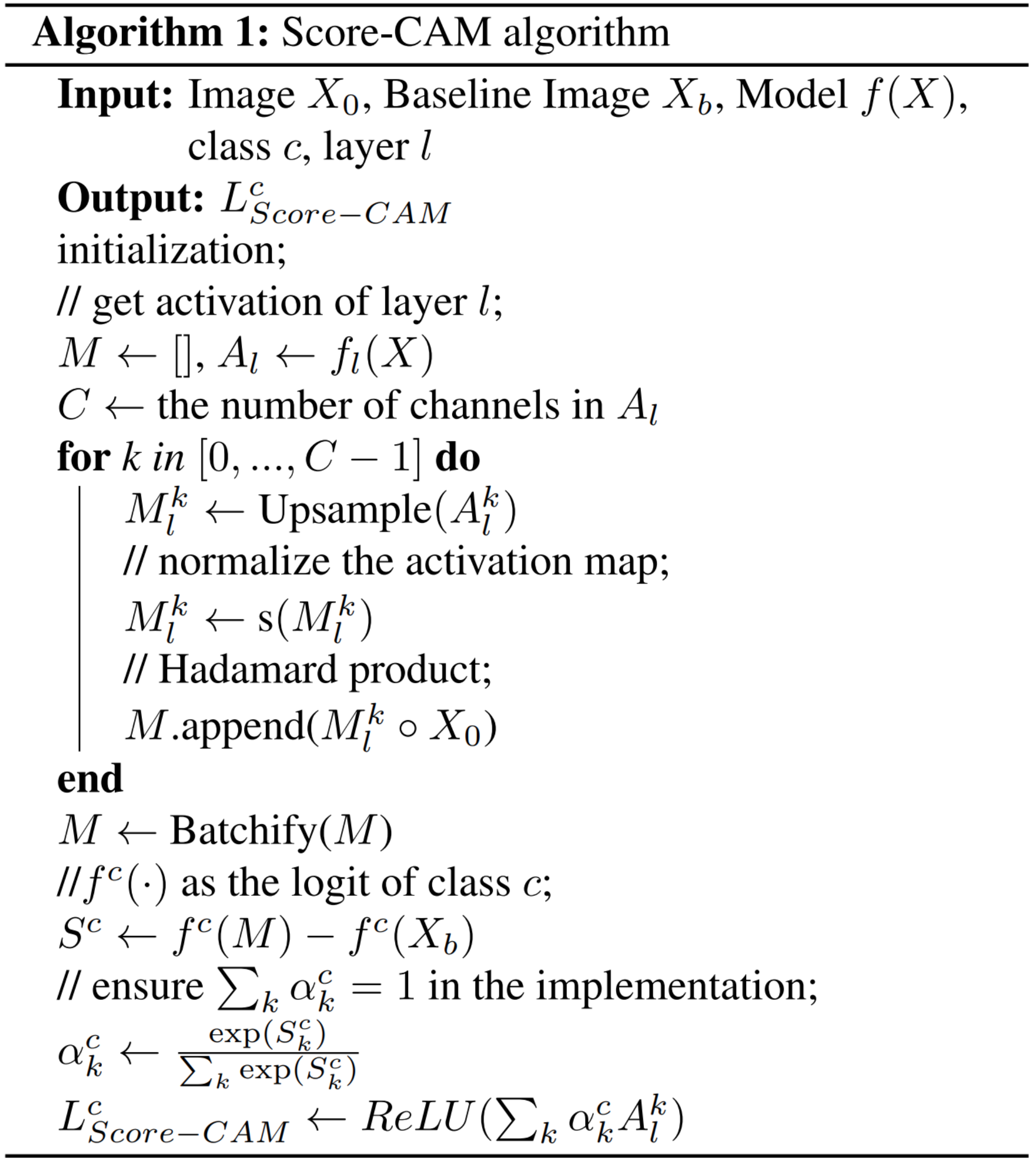
$$C(A_l^k) = f(X \circ H_l^k) - f(X_b)$$

where

$$H_l^k = s(Up(A_l^k))$$

- ***Up()*** is a function which up-samples the activation map to the input size
- ***s()*** is a normalization function mapping all values to the range [0,1]
- The CIC-score is the weighting for the kth activation map of the Conv-Layer l which is denoted as A_l^k

Score-CAM algorithm



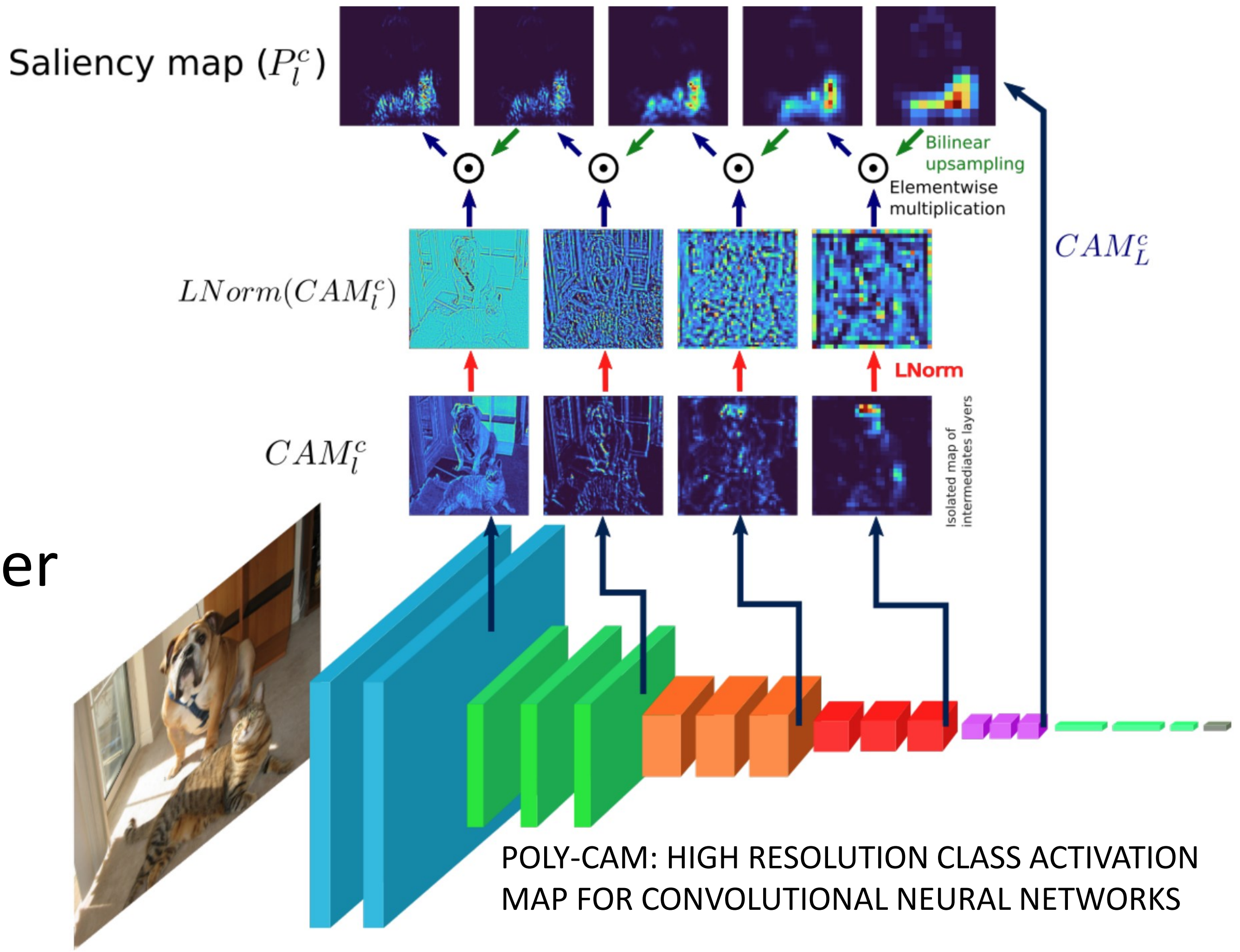
Algorithm 1: Score-CAM algorithm

```
Input: Image $X_0$, Baseline Image $X_b$, Model $f(X)$,
       class $c$, layer $l$
Output: $L^c_{Score-CAM}$
initialization;
// get activation of layer $l$;
$M \leftarrow []$, $A_l \leftarrow f_l(X)$
$C \leftarrow$ the number of channels in $A_l$
for $k$ in $[0, ..., C-1]$ do
    $M^k_l \leftarrow$ Upsample$(A^k_l)$
    // normalize the activation map;
    $M^k_l \leftarrow$ s$(M^k_l)$
    // Hadamard product;
    $M$.append$(M^k_l \circ X_0)$
end
$M \leftarrow$ Batchify$(M)$
//$f^c(\cdot)$ as the logit of class $c$;
$S^c \leftarrow f^c(M) - f^c(X_b)$
// ensure $\sum_k \alpha^c_k = 1$ in the implementation;
$\alpha^c_k \leftarrow \frac{\exp(S^c_k)}{\sum_k \exp(S^c_k)}$
$L^c_{Score-CAM} \leftarrow ReLU(\sum_k \alpha^c_k A^k_l)$
```

Score-CAM: Score-Weighted Visual Explanations for Convolutional Neural Network

Poly-CAM

- Use the information of multiple layers
- Use Bilinear up-sampling to backpropagate the information to the input layer
- Use the CIC $w_{l,k}$ as weighting



POLY-CAM: HIGH RESOLUTION CLASS ACTIVATION MAP FOR CONVOLUTIONAL NEURAL NETWORKS

$$P_l^c = \begin{cases} ReLU(\sum_k w_{l,k}(c)\boldsymbol{A}_l^k) \\ LNorm\left(ReLU\left(\sum_k w_{l,k}(c)\boldsymbol{A}_l^k\right), \frac{s_{l+1}}{s_l}\right) \odot \uparrow_{bi}\left(\boldsymbol{P}_{l+1}^c, \frac{s_{l+1}}{s_l}\right) & \text{for } 1 \leq l \leq L-1 \end{cases}$$

Summary and recommendation about the different types of saliency maps

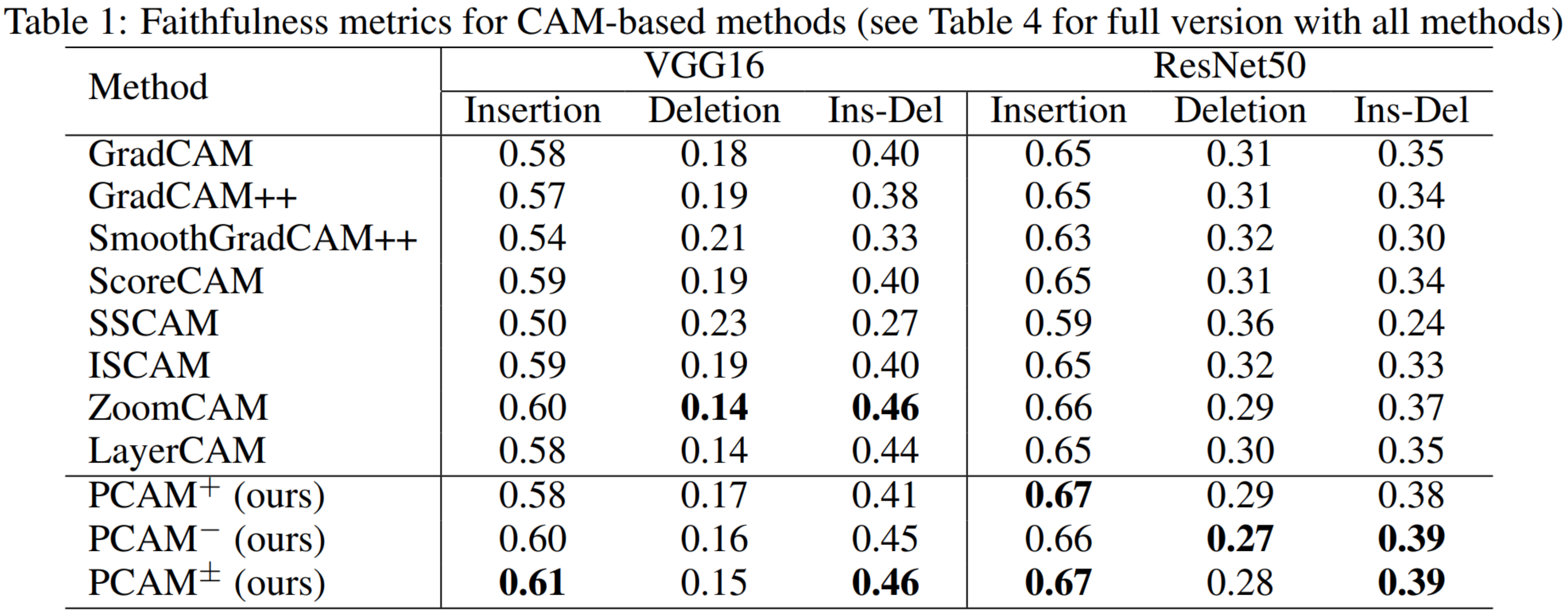


Table 1: Faithfulness metrics for CAM-based methods (see Table 4 for full version with all methods)

Method	VGG16			ResNet50		
	Insertion	Deletion	Ins-Del	Insertion	Deletion	Ins-Del
GradCAM	0.58	0.18	0.40	0.65	0.31	0.35
GradCAM++	0.57	0.19	0.38	0.65	0.31	0.34
SmoothGradCAM++	0.54	0.21	0.33	0.63	0.32	0.30
ScoreCAM	0.59	0.19	0.40	0.65	0.31	0.34
SSCAM	0.50	0.23	0.27	0.59	0.36	0.24
ISCAM	0.59	0.19	0.40	0.65	0.32	0.33
ZoomCAM	0.60	**0.14**	**0.46**	0.66	0.29	0.37
LayerCAM	0.58	0.14	0.44	0.65	0.30	0.35
PCAM$^+$ (ours)	0.58	0.17	0.41	**0.67**	0.29	0.38
PCAM$^-$ (ours)	0.60	0.16	0.45	0.66	**0.27**	**0.39**
PCAM$^\pm$ (ours)	**0.61**	0.15	**0.46**	**0.67**	0.28	**0.39**

POLY-CAM: HIGH RESOLUTION CLASS ACTIVATION MAP FOR CONVOLUTIONAL NEURAL NETWORKS

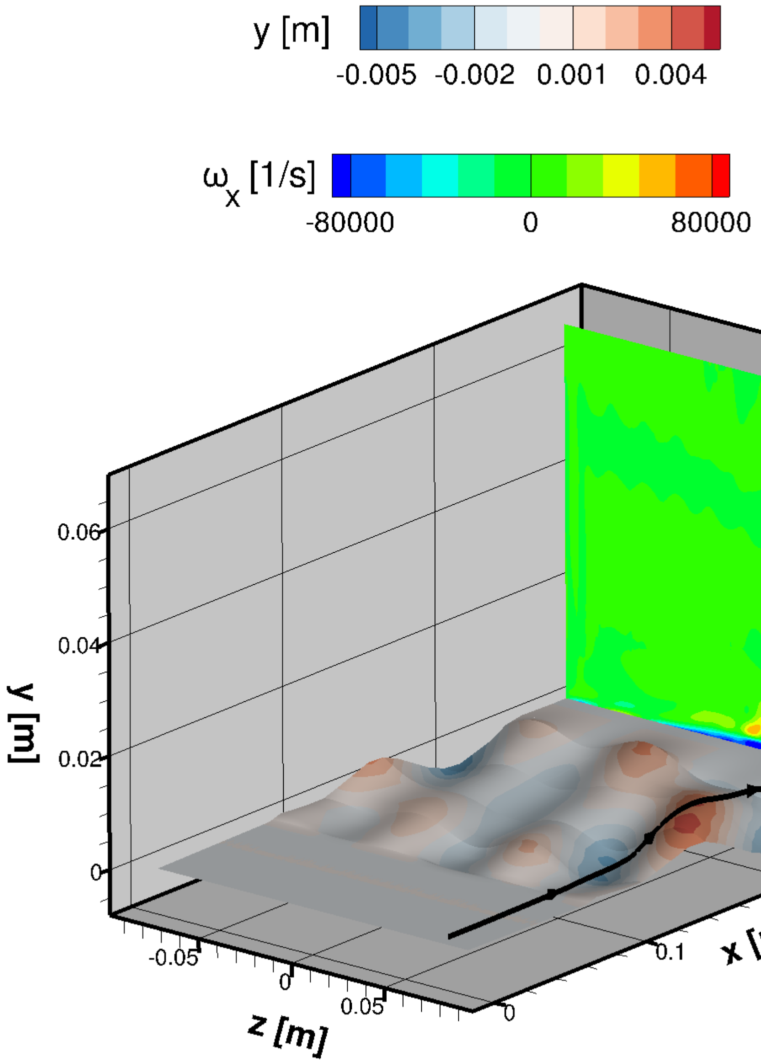
Sanity Checks for Saliency Maps

- Paper showed: Some widely used CAM methods give bad results.
- Sanity Check 1: Model Parameter Randomization Test
- Sanity Check 2: Data Randomization Test
- Result: Some CAM methods only act like edge detectors

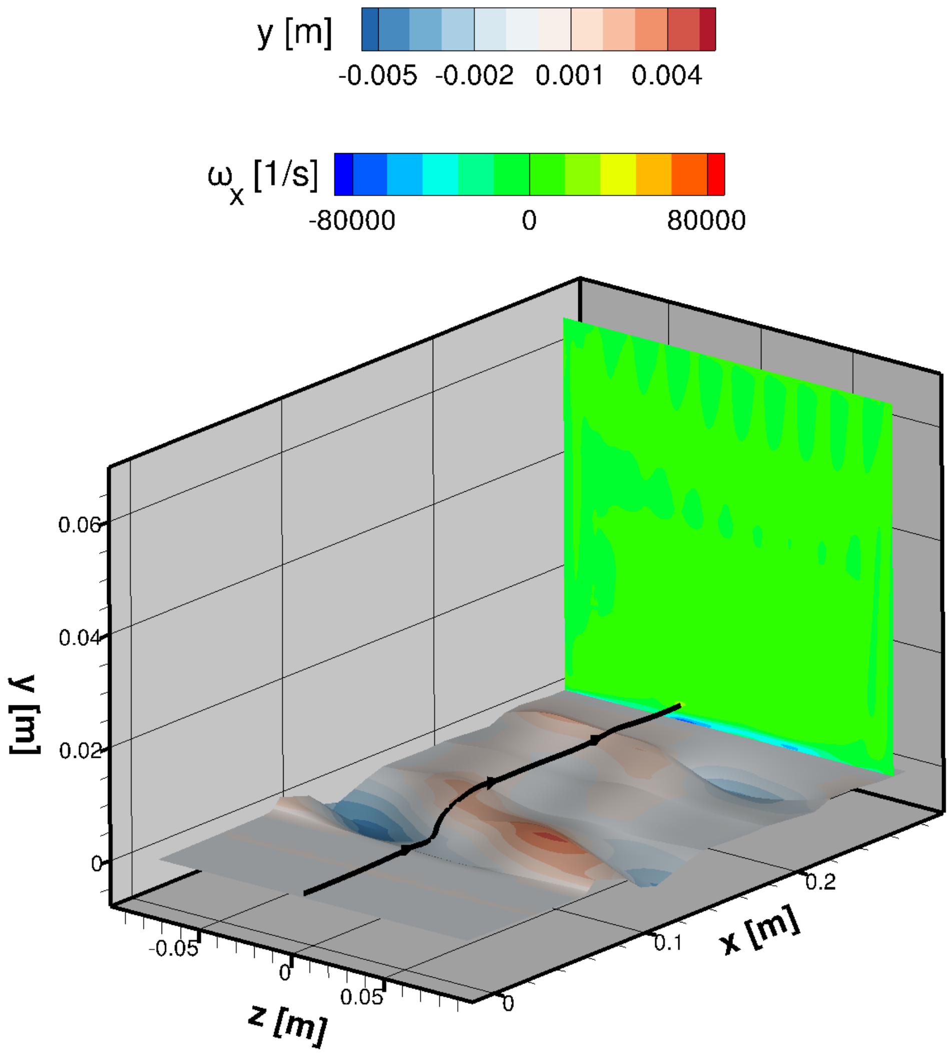
→ Avoid using Guided Backpropagation and Guided Grad-CAM

Where I used saliency maps so far

- For the re-entry of a space capsule into the earth's atmosphere a heat shield is used to slow down and protect the capsule
- Roughness on the surface of the heat shield leads to cross-flow vortices, which lead to faster martial removal on the heat shield
- Given a certain roughness the value of the maximal vortices should be predicted
- Used a train set with 10k of simulations carried out on different roughnesses
- Feeding the discretized roughness into the network to predict the maximum vorticity



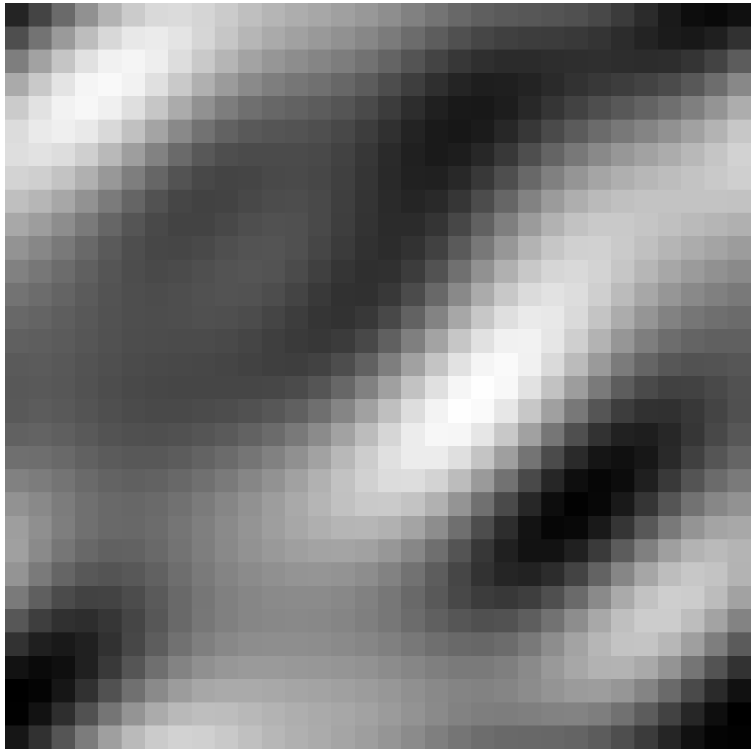
High vorticity



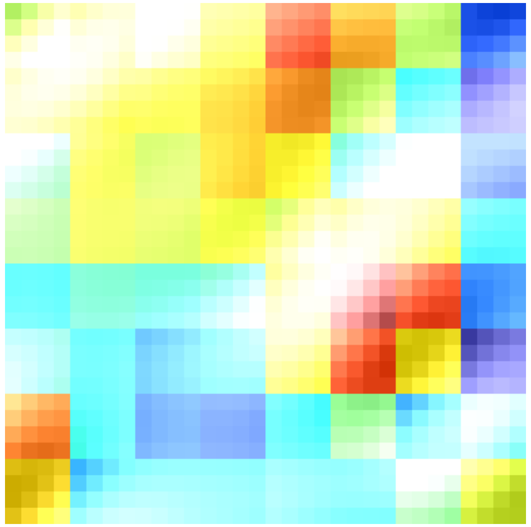
Low vorticity

CAMs (no interpolation)

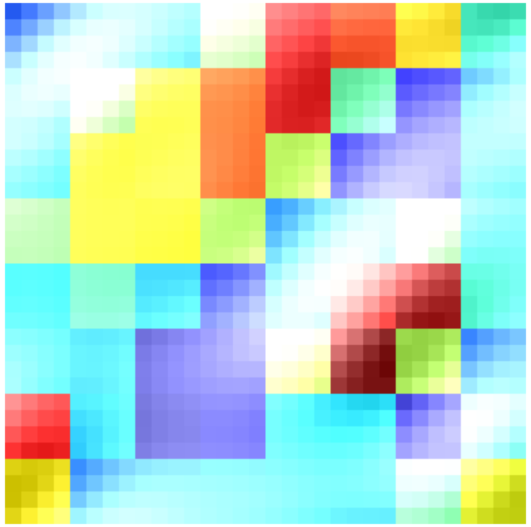
Input roughness



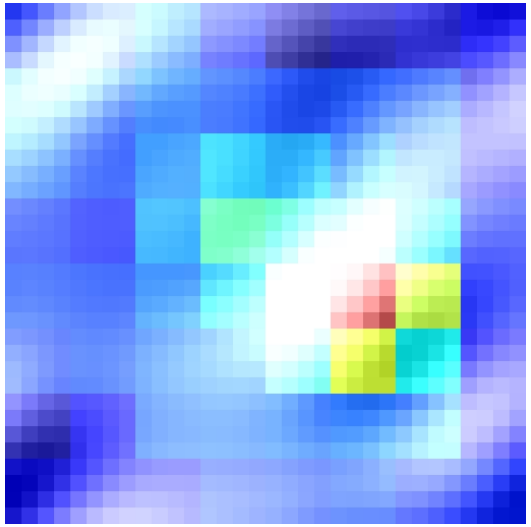
Grad-CAM



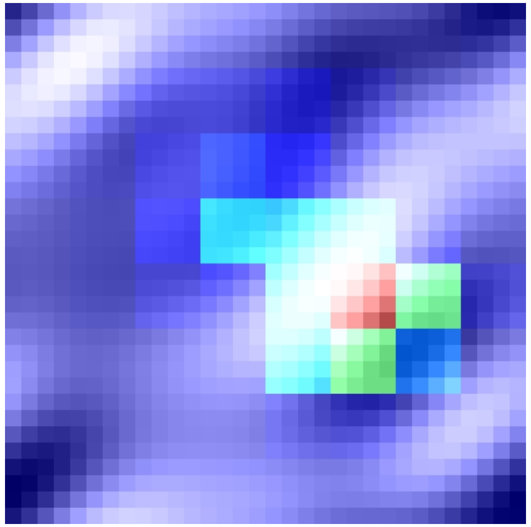
Grad-CAM only pos. derivatives



HiRes-CAM

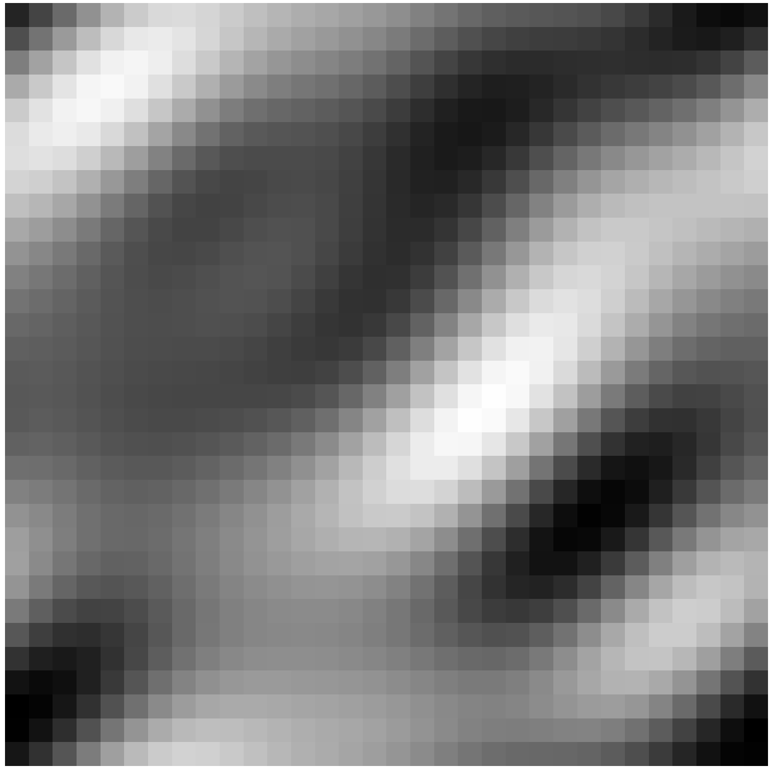


HiRes-CAM only pos. derivatives (layer-CAM)

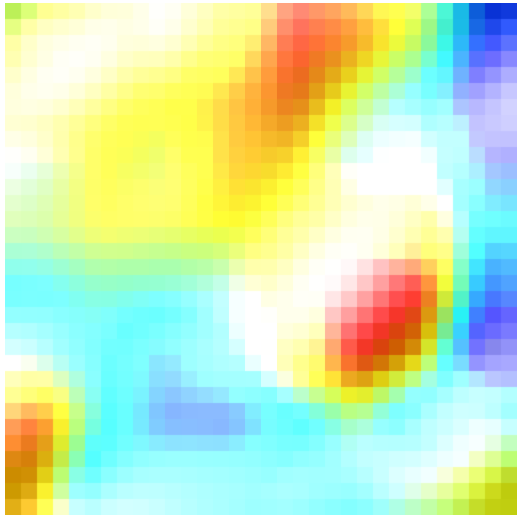


CAMs (interpolation)

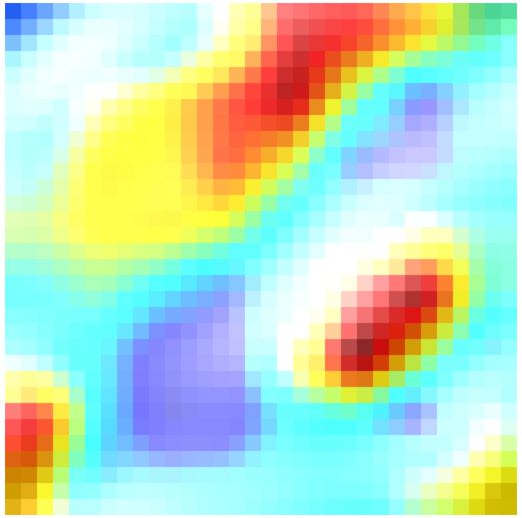
Input roughness



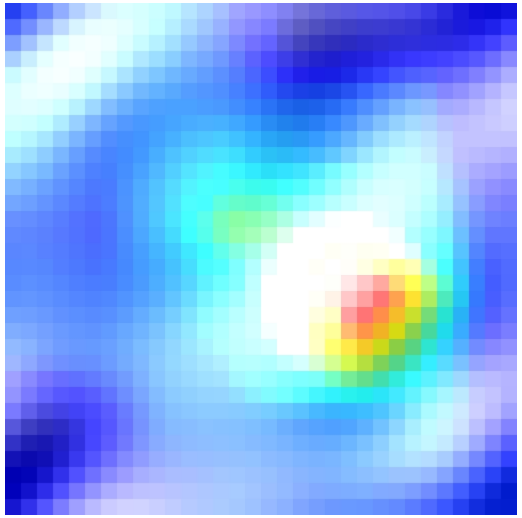
Grad-CAM



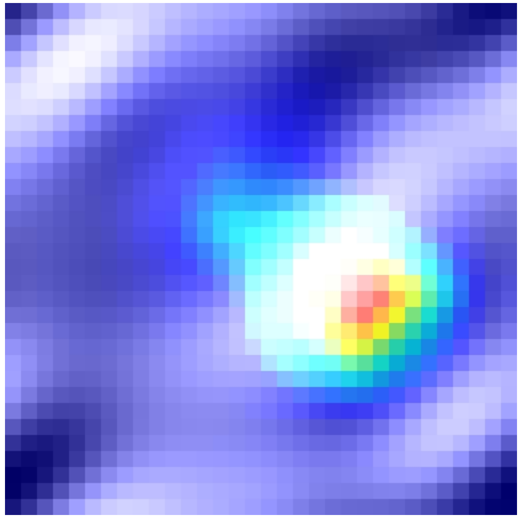
Grad-CAM only pos. derivatives



HiRes-CAM

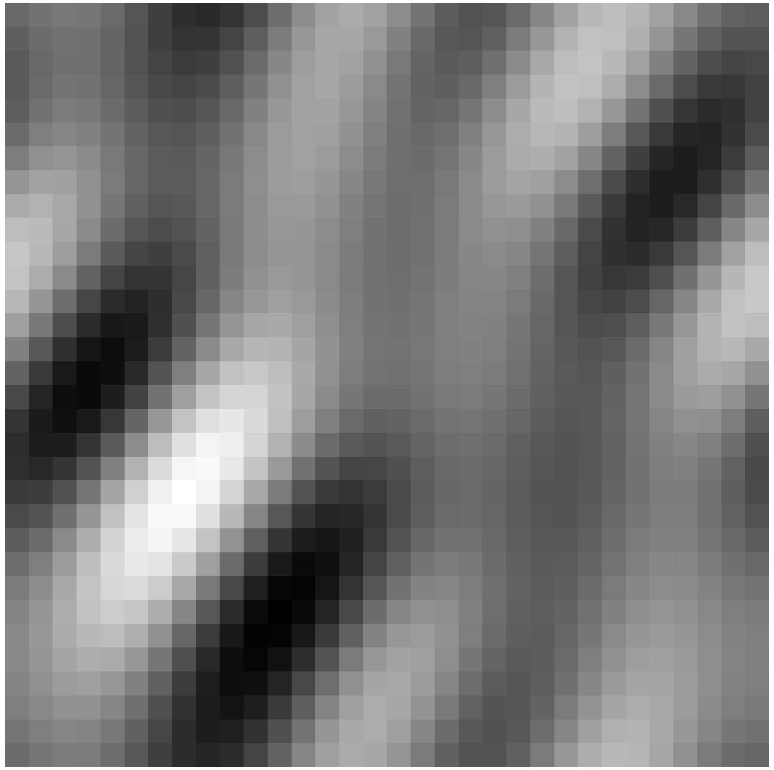


HiRes-CAM only pos. derivatives (layer-CAM)

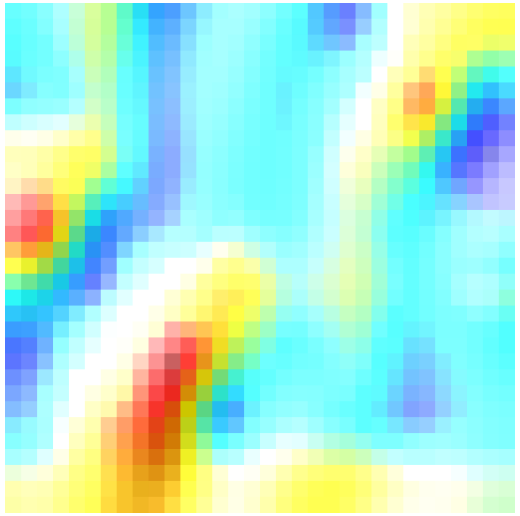


CAMs (interpolation)

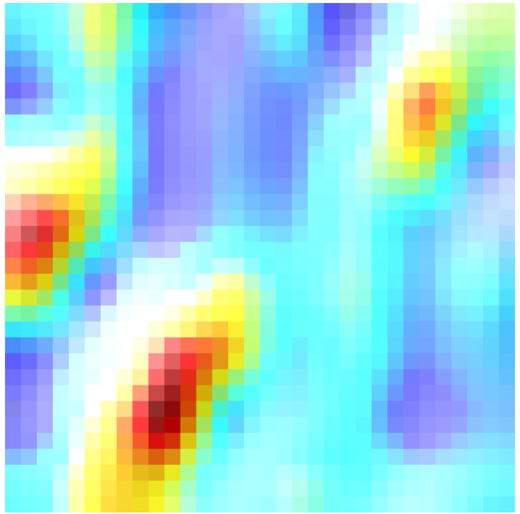
Input roughness



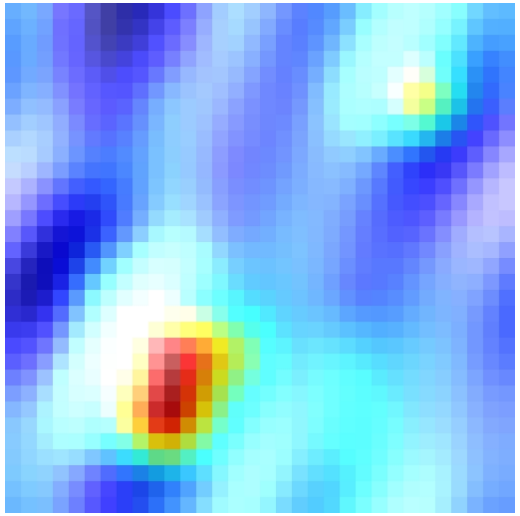
Grad-Cam



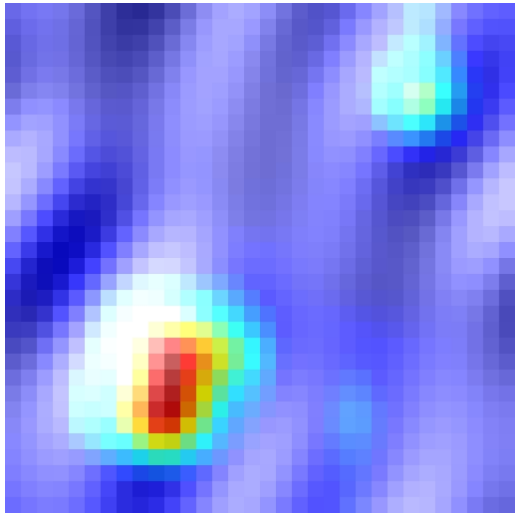
Grad-Cam only pos. derivatives



HiRes-Cam



HiRes-Cam only pos. derivatives (layer-CAM)



References

- Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 2921-2929)
- Draelos, R. L., & Carin, L. (2020). Use hirescam instead of grad-cam for faithful explanations of convolutional neural networks. *arXiv e-prints*, arXiv-2011
- 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 *Proceedings of the IEEE international conference on computer vision* (pp. 618-626)
- Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., ... & Hu, X. (2020). Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 24-25)
- Fahim, M. A. N. I., Saqib, N., Siam, S. K., & Jung, H. Y. (2022). Rethinking Gradient Weight’s Influence over Saliency Map Estimation. *Sensors*, *22*(17), 6516
- Englebert, A., Cornu, O., & De Vleeschouwer, C. (2022). Poly-CAM: High resolution class activation map for convolutional neural networks. *arXiv preprint arXiv:2204.13359*

References

- Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., & Kim, B. (2018). Sanity checks for saliency maps. *Advances in neural information processing systems*, *31*
- Jiang, P. T., Zhang, C. B., Hou, Q., Cheng, M. M., & Wei, Y. (2021). Layercam: Exploring hierarchical class activation maps for localization. *IEEE Transactio*
- Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018, March). Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *2018 IEEE winter conference on applications of computer vision (WACV)* (pp. 839-847). IEEE *ns on Image Processing*, *30*, 5875-5888
- Lerma, M., & Lucas, M. (2022). Grad-CAM++ is Equivalent to Grad-CAM With Positive Gradients. *arXiv preprint arXiv:2205.10838*
- Lu, M. T., Ivanov, A., Mayrhofer, T., Hosny, A., Aerts, H. J., & Hoffmann, U. (2019). Deep learning to assess long-term mortality from chest radiographs. *JAMA network open*, *2*(7), e197416-e197416
- Recasens, A., Kellnhofer, P., Stent, S., Matusik, W., & Torralba, A. (2018). Learning to zoom: a saliency-based sampling layer for neural networks. In *Proceedings of the European Conference on Computer Vision (ECCV)* (pp. 51-66)