Machine Vision and Applications (2024) 35:89
https://doi.org/10.1007/s00138-024-01567-7

RESEARCH

Poly-cam: high resolution class activation map for convolutional neural networks

Alexandre Englebert[1,2] · **Olivier Cornu**[2,3] · **Christophe De Vleeschouwer**[1]

Received: 19 March 2024 / Revised: 3 June 2024 / Accepted: 7 June 2024 / Published online: 3 July 2024

Abstract
The demand for explainable AI continues to rise alongside advancements in deep learning technology. Existing methods such as convolutional neural networks often struggle to accurately pinpoint the image features justifying a network's prediction due to low-resolution saliency maps (e.g., CAM), smooth visualizations from perturbation-based techniques, or numerous isolated peaky spots in gradient-based approaches. In response, our work seeks to merge information from earlier and later layers within the network to create high-resolution class activation maps that not only maintain a level of competitiveness with previous art in terms of insertion-deletion faithfulness metrics but also significantly surpass it regarding the precision in localizing class-specific features.

Keywords Explainable AI · Deep learning · Convolutional neural networks · Feature localization · High resolution class activation map

1 Introduction

We currently face unprecedented advances in the domain of artificial intelligence, primarily driven by developments in deep neural networks (DNNs). However, in contrast to techniques based on handcrafted features, DNNs often lack transparency and explainability [1]. The need to assess a posteriori the behavior of a model has led to the development of explainable artificial intelligence (XAI) methods, ranging from the development of more transparent model architectures [2] to post-hoc methods (explanation, by example of black-box methods). Those methods have primarily been developed for convolutional neural networks (CNN), without addressing the sampling specificities and transmission issues raised by neuromorphic computing architectures [3–5]. A detailed review of the approaches that have been investigated to explain a CNN is provided in [6].

Saliency maps visualization has been adopted as a convenient approach to identify the image parts justifying the network prediction. Those saliency maps are helpful to check that the predictions of a model are grounded on relevant information. It is indeed known that training convergence alone does not exclude undesired DNN predictions [7], typically because the model has learnt inputs/outputs correlations that do not correspond to the desired meaningful causal relationship. An example of undesired behavior is the one of an image classifier that relies on a source tag present in about one-fifth of the horse figures to detect horses, and inserting the tag on an image of a car changes the classification from car to horse.

Alternatively, when sufficiently accurate, the localization of salient features could convince a user that a model works properly, i.e. uses relevant cues, or could even help in identifying the parts of a signal that are relevant to solve a problem, e.g. help a medical doctor in identifying the X-ray visual cues that permit to anticipate the evolution of a treatment.

Various techniques are available to define class-specific saliency (i.e. specific to a given output class) including perturbation-based analysis [8], gradient-based techniques

✉ Alexandre Englebert
alexandre.englebert@uclouvain.be

Olivier Cornu
olivier.cornu@uclouvain.be

Christophe De Vleeschouwer
christophe.devleeschouwer@uclouvain.be

[1] Information and Communication Technologies, Electronics and Applied Mathematics (ICTEAM), UCLouvain, Louvain-la-Neuve, Belgium

[2] Service de Chirurgie Orthopédique et Traumatologie, Cliniques Universitaires Saint-Luc UCL, Brussels, Belgium

[3] Neuro musculo skeletal Lab (NMSK), UCLouvain, Brussels, Belgium

[9, 10], Class Activation Mapping techniques [11–14], or combination of these techniques [14–16]. As demonstrated in our experimental section, existing Class Activation Maps [11] methods are limited in resolution, while gradient-based techniques are generally subject to noise and thus produce saliency maps composed of a large number of widespread peaky spots.

In a preliminary conference paper, we introduced a method called Poly-CAM [17] to generate high-resolution class activation maps without relying on gradient backpropagation and thus limiting the noisiness of the map. This method is achieved by multiplexing the high-resolution activation maps available in the early layers of the network with oversampled versions of the class-specific activation maps computed in the later layers of the network.

In this paper, we develop the method in more detail, including the application of a novel channel-wise perturbation method to generate the class activation maps weighting factors speeding-up the saliency map computation. Additional results are also provided, with industrial application perspective discussion. The source code to reproduce the proposed method is available at https://github.com/aenglebert/polycam.

The shared elements between the present paper and the conference paper include:

- A perturbation-based approach that is able to generate high-resolution saliency maps without relying on somehow erratic gradient back-propagation.
- An in depth study of our recursive high resolution refinement of saliency maps on the 2012 ILSVRC dataset.

The novel contributions from this paper are:

- The implementation of novel perturbation mechanism that is based on the ablation of feature maps, thereby reducing the complexity of Poly-CAM to the one of a conventional Score-CAM.
- A additional comparison between input and feature maps perturbations on 2012 ILSVRC dataset.
- An exploration of the applicability of the method on industrial images.
- An ablation study on the LNorm operation.

Similarly to previous works addressing the CNN explainability question, our study suffers from the lack of a definitive objective quantitative metric to assess the quality of the generated saliency patterns [18]. However, as depicted in Fig. 1, the visual inspection of the maps reveals that it is able to highlight fine semantically meaningful details that are expected to contribute to the model decision. Moreover, the results obtained in terms of insertion-deletion faithfulness metrics, recommended as the most relevant quantitative metric in earlier works [13, 14, 19, 20], are also in favor of our method.

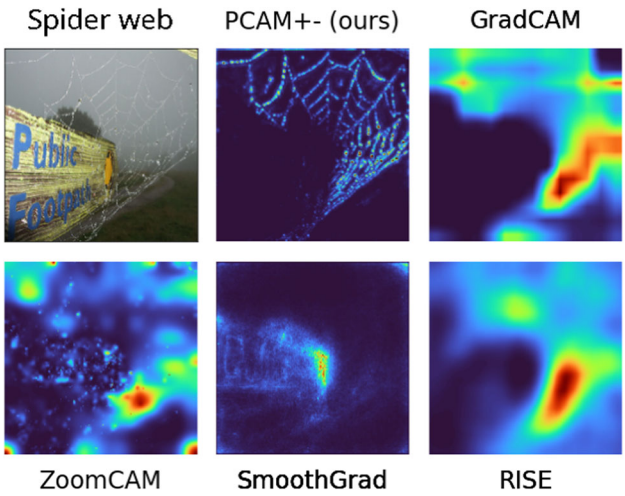

Fig. 1 An example of saliency maps produced for the class "Spider web" by a few XAI methods. The proposed Poly-CAM approach can generate high resolution class activation maps in comparison to previous methods

2 Related work

Various strategies allow visualization of the image features contributing to the class prediction of a CNN.

Perturbation-based methods use multiple perturbations of the input image, and monitor the changes they induce at the output of the model to build a saliency map. Those methods range from occlusion techniques, which recursively occlude patches of the input [8], to more elaborate perturbations such as Randomized Input Sampling for Explanation of Black-box Models (RISE) [19], which generate many random masks normalized between 0 and 1, to be weighted by the class-specific softmax output obtained when the input image is multiplied by the mask. These methods result in smooth masks and are computationally intensive, especially when the resolution of the saliency map increases.

Gradient-based methods use the back-propagated gradient of the neural network to identify regions in the input image that largely impact the prediction. Those methods however suffer from gradient shattering [21], which results in a noisy saliency map. To mitigate this problem, a variety of approaches have been implemented to smooth out the gradient signal. They include Integrated Gradient [9], which integrates the gradient for multiple interpolations between a baseline and the input image, or SmoothGrad [10], which averages gradient for multiple perturbed variations of the input image.

Class Activation Map [11] was initially designed as a linear combination of activations from the last convolu-

tional layer, weighted by the parameters of a fully connected classifier, taking as input the global average pooling of each channel in this last convolutional layer. Multiple methods were introduced based on alternative definitions of the weights. Grad-CAM [12] and derivative methods [13, 22] define the weights based on gradients backpropagated up to the last convolutional layer. Score-CAM defines the weight of a channel based on the softmax output associated to the target class when probing the model with a version of the image masked by the activation channel. In a sense, it mixes CAM methods with perturbations methods [14]. Multiple methods expanding this concept are also described such as SS-CAM [20], IS-CAM [23] or Augmented Score-CAM [24]. Ablation-CAM [25] propose to apply the perturbation inside of the network by alternatively switching off each channel at the last convolutional layer and evaluate the change in classification. Methods such as FD-CAM [26] use a grouped channels switching to compute the scores, this consist of performing the Ablation-CAM switching process on groups of channels instead of isolated channels. The score obtained is also combined with backpropagated gradient. Overall, Class Activation Map methods are less noisy than pure gradient-based methods but are coarser, with a resolution limited to the resolution of the last convolution layer. This can be seen in Fig. 1 and Sect. 4.3.1 with SmoothGrad [10] and IntegratedGrad [9] as examples of gradient methods, and GradCAM [12] and FD-CAM [26] as examples of classical CAM methods.

To produce CAM at higher-resolution, [27] proposed training an expansion network on a segmentation task using CAM supervision to produce a saliency map.

This has the drawback that the definition of the architecture of this expansion network is specific for a given model architecture, thereby requiring specific training for each model and dataset to explain. At the time of writing, this expansion network is only defined for DenseNet [28] and trained on a subset of animal labels of ImageNet [29], available at https://github.com/djib2011/high-res-mapping. Very recently, Layer-CAM [16] and Zoom-CAM [15] have proposed to generate high-resolution CAM by combining back-propagated gradients with activations from multiple layers, using element-wise multiplication. The two methods improve the resolution of the maps but also inherit some noise from the gradients.

In our preliminary conference paper, we introduced Poly-CAM, an original method to compute high-resolution activation maps without using gradients, nor requiring to train a specialized network. The main contribution was the process to leverage the activation from multiple layers to increase the resolution of the Class Activation Map up to the one of the input image.

This paper extends the preliminary version of the method by introducing a new algorithm that uses a perturbation at the channel level in combination to the aforementioned combination of layer activations.

3 Proposed Poly-CAM approach

This section presents the core contribution of our work. Section 3.1 introduces the notations and variables required in the rest of the text, while Sect. 3.2 reviews the formal definition of the conventional Class Activation Map method, which serves as a baseline to our work. Section 3.3 then introduces our Poly-CAM approach, which proposes to generate a high resolution class activation map by recursively multiplexing the high-resolution activation maps available in the early layers of the network with upsampled versions of the class-specific activation maps computed in the last layers of the network. Eventually, Sect. 3.4 introduces three different methods to associate a weight to each layer activation channel by masking/unveiling the input based on the channel activation. Section 3.5 considers a channel switching strategy, meaning that the perturbation is not performed at the input level but directly inside of the network by zeroing specific channels.

3.1 Notations

Let $f_\Theta(X)$ denotes the prediction of a CNN with parameters Θ when the image X is provided as input. In the following, for conciseness and because we are interested in analyzing a trained network (parameters Θ are fixed), we omit Θ, and just use $f(X)$ to refer to the CNN prediction associated to X. $f(X)$ is a vector, defined by the output of a softmax. $f_c(X)$ denotes the component of $f(X)$ corresponding to the class c.

A_l denotes the activation tensor of the l^{th} convolutional layer, $1 \leq l \leq L$, while A_l^k refers to the activations of the k-th channel of layer l.

s_l denotes the subsampling factor of layer l compared to the input. It corresponds to the product of stride and pooling factors encountered between the input and layer l.

$\uparrow_{bi}(M, s)$ defines a bilinear upsampling of a matrix $M \in \mathbb{R}^{m \times n}$ by a factor $s \in \mathbb{N}$.

$\downarrow_{av}(M, s)$ denotes a 2D average pooling on any matrix $M \in \mathbb{R}^{m \times n}$ with a stride $s \in \mathbb{N}$.

$u(M)$ linearly maps the value range of the elements in matrix M to the unit interval.

$\oslash$ denotes the element-wise division operator, while $\odot$ denotes the element-wise product operator.

$LNorm(M, s)$ is a local normalisation operator. It partitions the matrix $M \in \mathbb{R}^{m \times n}$ in a set of non overlapping blocks of size $s \times s$, with $s \in \mathbb{N}$, and divides each matrix element by the mean value of its corresponding block. Formally, using

the above notations,

$$LNorm(M, s) = M \oslash (\uparrow_{bi} (\downarrow_{av} (M, s), s)) . \quad (1)$$

$ReLU$ denotes the rectified linear unit operator [30].

3.2 Activation maps in previous work

In [11] CAM_l^c, the Class Activation Map associated to a target class c and a layer l is defined as

$$CAM_l^c = ReLU\left(\sum_k w_{l,k}(c) A_l^k\right), \quad (2)$$

with A_l^k denoting the k^{th} activation map of the l^{th} convolutional layer, $l \in 1, ..., L$, and $w_{l,k}(c)$ being a scalar weighting factor.

Most of the CAM-based methods [10, 12–14, 20, 23, 25, 26], adopt this formula. They differ in the way they define the weighting factors, and generally only consider it for the last convolutional layer ($l = L$). Alternatively, Zoom-CAM and Layer-CAM have proposed to combine activation maps from multiple layers, using gradients as dense weighting factors. Our work also combines multiple activation maps, but does it without back-propagated gradients, thereby managing to produce high-resolution saliency maps without inheriting the noise from the gradient. As demonstrated by our results in Sect. 4.2 and 4.3.1, this has a huge impact on the visual quality of the saliency maps.

3.3 Our proposed Poly-CAM

It is common that CNNs handle layers at multiple scales/resolutions [31, 32], e.g. to reconstruct images [33] or concatenate features [34].

Our method leverages information from multiple those multiple scale layers to produce a high resolution Class Activation Map. Similar to other CAM-based techniques, it builds on the linear combination of activation maps, but combines them through a backward recursive procedure, as depicted in Fig. 2.

Letting P_l^c denote the class-specific saliency map associated to class c in the l^{th} layer, the recursive process works as follows. In the initial step, the saliency map P_L^c is defined to be equal to the conventional CAM_L^c saliency map, as derived from Eq. (2). Then, at each recursive step, an upsampled version of P_{l+1}^c is tuned (or modulated) by a locally normalized version of the activation map in the l^{th} layer. Mathematically, we have:

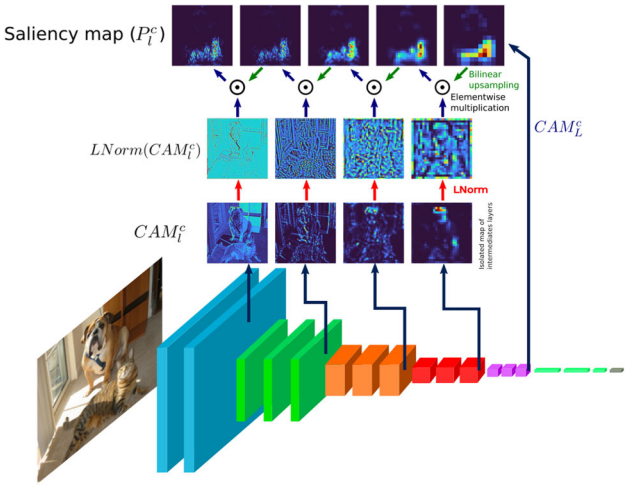

Fig. 2 Our Poly-CAM process: the upsampled version of the saliency map in layer l is tuned based on the class activation map of layer $l-1$. Image samples correspond to the 'cat' class, and are computed from VGG16[35]

$$P_l^c = \begin{cases} CAM_l^c & \text{for } l = L \\ LNorm\left(CAM_l^c, \frac{s_{l+1}}{s_l}\right) \odot \uparrow_{bi} \left(P_{l+1}^c, \frac{s_{l+1}}{s_l}\right) & \text{for } 1 \le l \le L-1 \end{cases}, \quad (3)$$

with CAM_l^c defined in Eq. (2), and s_l defining the subsampling factor of layer l compared to the input. The class-specific weights $w_{l,k}(c)$ involved in Eq. 2 to define CAM_l^c are defined in Sect. 3.4 and 3.5, based on perturbations related to the content of the (l, k) channel.

Intuitively, Eq. (3) can be understood based on the following two observations. First, the element-wise multiplication, between the upsampled (and thus smooth) saliency map of layer $l+1$ and the activation map in layer l, aims at restricting the large saliency values in layer $l+1$ to the locations that are activated in layer l. Second, the local normalization ($LNorm$) of the activation map aims at preserving the spatial distribution of saliency across the layers. It ensures that an image block with large saliency in layer l+1 has also a large saliency in layer l, even if the level of activation in this block is small compared to the rest of the image. This is meaningful since the backpropagated saliency should be predominant to assign a saliency level to a spatial region in layer l, while the activation in layer l should simply control the increase in resolution, by tuning the smooth saliency signal inherited from coarser layers based on the local variations of the activation map.

Our ablation study in Sect. 4.5 confirms the critical role played by the LNorm operator.

This paper also extends the preliminary version of the method by a new algorithm that uses a perturbation at the

level of the channels in combination to the aforementioned combination of layers activation

3.4 Input perturbation for weight definition

This section presents different alternatives to define the weights $w_{l,k}(c)$, used in Eq. (2), by using a masking of the input image.

Channel-wise Increase of Confidence Score-CAM [14], SS-CAM [20] and IS-CAM [23] define $w_{l,k}(c)$ based on the so-called Channel-wise Increase of Confidence (CIC), which estimates how the spatial support of the activation map A_l^k contributes to the softmax output f_c. Formally, the channel-wise increase is denoted $\boldsymbol{w}_{l,k}^{+}(c)$, and is defined as:

$$\boldsymbol{w}_{l,k}^{+}(c) = f_c\left(X \odot u\left(\uparrow_{bi}\left(A_l^k, s_l\right)\right)\right) - f_c\left(X_b\right), \tag{4}$$

with $\odot$ denoting the pixel-wise product, and X_b referring to a baseline image. Previous works have considered baselines that are uniform black, uniform grey, or a blur version of X. In the following, $f_c(X_b)$ is set to zero in all experiments.

Our work proposes two extensions of (4), respectively to measure how the softmax output decreases when masking a fraction of the input, and to sum-up the increase and the decrease associated to the unveiling and the masking of the input. Those new weights are defined as follows.

Channel-wise Decrease of Confidence The Channel-wise Decrease of Confidence (CDC) is a dual notion compared to CIC. Instead of measuring the increase of softmax output when the part of the input corresponding to non-zero A_l^k is unveiled and the remaining is masked, CDC measures the decrease of softmax output when the part of the input corresponding to A_k^l is masked. The intuition is that an important part of the input for any class c not only increase the output when shown, but should also decrease it when hidden. Formally,

$$\boldsymbol{w}_{l,k}^{-}(c) = ReLU\left(f_c(X) - f_c\left(X \odot \left(1 - u\left(\uparrow_{bi}\left(A_l^k, s_l\right)\right)\right)\right)\right) \tag{5}$$

$ReLU$ is applied to only keep the activation maps that decreases the output when removed.

Channel-wise Variation of Confidence By combining the CIC with the CDC, the Channel-wise Variation of Confidence (CVC) is defined. As,

$$\begin{aligned}\boldsymbol{w}_{l,k}^{\pm}(c) = ReLU\Big(f_c\left(X\right) + f_c\left(X \odot u\left(\uparrow_{bi}\left(A_l^k, s_l\right)\right)\right) \\ - f_c\left(X \odot \left(1 - u\left(\uparrow_{bi}\left(A_l^k, s_l\right)\right)\right)\right)\Big).\end{aligned} \tag{6}$$

The Channel-wise Variation of Confidence is influenced by the ability of the activation map to increase the softmax output when inserted, but also by its ability to decrease it when removed.

3.5 Channel perturbation for weight definition

Similarly to FD-CAM [26], we propose to define the weight $w_{l,lk}(c)$ by perturbing the l^{th} layer rather than the input. Therefore, channels of a convolutional layer are grouped together based on their similarities. We computed the cos similarity between an activation map A_l^k and every activation maps $A_l^{k'}$ from the same layer l, with $k \neq k'$.

$$cos(A_l^k, A_l^{k'}) = \frac{v_l^k . v_l^{k'}}{\|v_l^k\| . \|v_l^{k'}\|}, \tag{7}$$

with v_l^k and $v_l^{k'}$ as the vectors obtained after flattening A_l^k and $A_l^{k'}$. For each channel in a layer, we identify a the predefined percentage κ of channels in the same layer with highest cosine similarities. The subset of channels that are similar to channel k in layer l is denoted $S_{k,l}$ (for the sake of simplicity, we omit the dependency on the input image X in the notation). We let $f_{l,S_{k,l}}(X)$ denote the class-specific output of the model f when all the activation maps in layer l are dropped, i.e. all values are set to zero, except for the maps in $S_{k,l}$, which are kept untouched. Similarly, $f_{c,l,\overline{S}_{k,l}}(X)$ denotes the model f_c when the activations maps of the layer l remain untouched, except for ones in $S_{k,l}$ that are set to zero.

We use an approach similar to the one adopted in Eq. 6 to define the $w_{l,k}(c)$. This weight is denoted $\boldsymbol{w}_{l,k}^{\emptyset\pm}(c)$, where $\emptyset$ refers to the fact that it is obtained by setting some channels to zero. Formally,

$$\boldsymbol{w}_{l,k}^{\emptyset\pm}(c) = f_c(X) - f_{c,l,\overline{S}_{k,l}}(X) + f_{l,S_{k,l}}(X). \tag{8}$$

4 Experiments

Four variants of the Poly-CAM method introduced in Sect. 3.3 and Sect. 3.5 are considered, depending on whether $\mathbf{w}_{l,k}$ is defined to be equal to $\boldsymbol{w}_{l,k}^{+}$ (PCAM^{+}), $\boldsymbol{w}_{l,k}^{-}$ (PCAM^{-}), $\boldsymbol{w}_{l,k}^{\pm}$ (PCAM$^{\pm}$) or $\boldsymbol{w}_{l,k}^{\emptyset\pm}$ ($\emptyset$PCAM).

We follow the assessment method described in [13] and [19] to evaluate our proposal. This assessment consists of evaluating a defined number of images using insertion and deletion metrics, as described in Sect. 4.1 and Sect. 4.3. Datasets, networks, and baseline methods are presented in Sect. 4.1. Qualitative and visual assessment is presented in Sect. 4.2, while quantitative assessment of the saliency maps

is considered in Sect. 4.3. Section 4.4 presents the results of the Sanity check and robustness metrics. An LNorm ablation study is carried out in the Sect. 4.5. We assessed the speed of execution in Sect. 4.6. Section 4.2.3 explores the usage of the proposed method to an industrial use case.

4.1 Experimental set-up and saliency map baselines

For these evaluations, 2000 images were randomly selected from the 2012 ILSVRC validation set [29]. The images are scaled to 224x224x3 pixels and normalized to the same mean and standard deviation as the ImageNet [29] training set (mean vector: [0.485, 0.456, 0.406], standard deviation vector [0.229, 0.224, 0.225]). The models used for faithfulness evaluation are VGG16 [35] and ResNet50 [36], both pretrained from the PyTorch model zoo. The analysis considers, as reference baselines, Grad-CAM [12], Grad-CAM++ [13], Smooth Grad-CAM++ [37], Score-CAM [14], SS-CAM [20], IS-CAM [23], Zoom-CAM [15], Layer-CAM [16], Occlusion [8], Input X Gradient [38], FD-CAM [26], Integrated Gradient [9], SmoothGrad [10] and RISE [19]. The implementations for these methods are the ones from Captum [39] for Integrated Gradient, SmoothGrad and Occlusion, from https://github.com/eclique/RISE for RISE, from https://github.com/X-Shi/Zoom-CAM for Zoom-CAM, from https://github.com/crishhh1998/FD-CAM for FD-CAM and from torchcam [40] for all the other CAM-based methods.

For SS-CAM, IS-CAM, FD-CAM, LayerCAM and Zoom-CAM, SmoothGrad and IntegratedGradien, the parameters recommended by the authors or set as default in the reference implementation have been used when available. [1] For FD-CAM, the source code was not compatible with architectures other than VGG at the time of writing without modifications. The experiments were thus limited to VGG16 for this method. For Occlusion, the size of occlusion patch was set to (64, 64) with a stride of (8, 8) as used by [19]. For RISE, 6000 masks were used.

For the Poly-CAM methods (PCAM$^+$, PCAM$^-$, PCAM$^\pm$, ØPCAM), the layers corresponding to a change in resolution were considered to recursively compute the saliency map as depicted in Fig. 2. It corresponds to [block1_conv2, block2_conv2, block3_conv3, block4_conv3, block5_conv3] for VGG16, and [conv1_1, conv2_3, conv3_4, conv4_6, conv5_3] for ResNet50. For ØPCAM, κ was set to 0.05, following previous works[26].

[1] It means 35 input perturbations (with a $\sigma = 2$ Gaussian noise) for SS-CAM, 50 input perturbations (with a $\sigma = 1$ Gaussian noise) for SmoothGrad, 10 interpolation steps for IS-CAM, threshold at 0.95 for FD-CAM and 50 for IntegratedGradient. For Layer-CAM, the layers corresponding to a change in resolution were used, and recommended scaling has been applied to the first two layers. For Zoom-CAM, all the layers/blocks were fused for VGG16 and ResNet50.

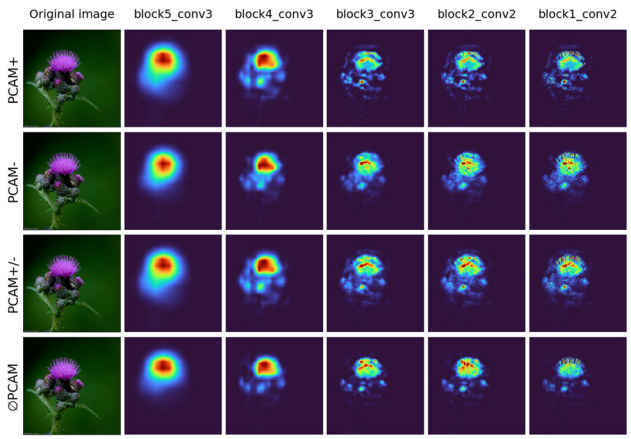

Fig. 3 Layer refinement of PolyCAM method for the four proposed variants. The high frequency details appears progressively during the iterative process

4.2 Visual qualitative assessment

This section assesses our method visually. Saliency maps were generated for all the baseline methods (see Sect. 4.1) on the 2000 selected images using VGG16 model. For the Poly-CAM methods, saliency maps were also generated for each target layer. An interactive interface is provided with the source code as a jupyter notebook, in addition to the source code, to allow an easy visualisation of any of the saliency maps generated in our experiments. Section 4.2.1 compares the three Poly-CAM variants, as a function of the layer index and targeted class. A comparison with previous works is shown in Sect. 4.2.2.

4.2.1 PCAM variants

PCAM produces saliency maps at various resolutions. Figures 2 and 3 show how Poly-CAM progressively refines the last layer saliency map through a backward recursive strategy. We observe that the structures are coarse at block5_conv3, to gain in accuracy when accounting for earlier network layers, doubling the resolution at each step. The elements highlighted by the three variants are similar on the majority of images. However, variations appear on some images, PCAM$^-$ highlighting more frequently contextual elements compared to PCAM$^+$ (and PCAM$^\pm$ sitting between the two). Intuitively, this can be understood by the fact that the appearance of those contextual features in a baseline image does not help in classifying the image correctly, while their removal from the original image hinders the classification. ØPCAM produce similar results with sharper and more thin highlighting.

All Poly-CAM variants are class specific as displayed in Fig. 4, where the saliency maps associated to the the Barn, Alps and the Ox classes are clearly distinct, with a level

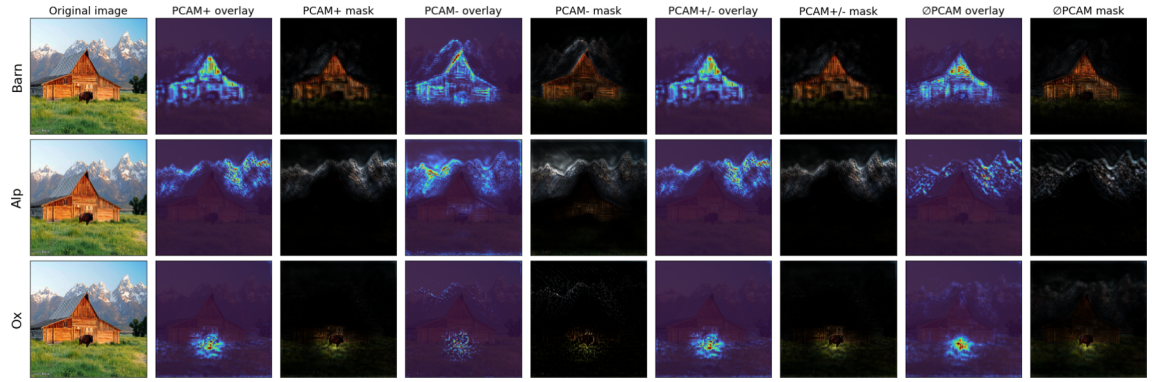

Fig. 4 Class specificity for Poly-CAM. The different classes as correctly determined by the four variants. We can note that PCAM$^-$ is less specific than the other methods (part of the mountain is attributed to the barn), while PCAM$^+$ is the most specific. In general, the different PolyCAM methods perform similarly

of accuracy close to segmentation. It is worth noting that PCAM$^+$ is more specific in highlighting the part of the image related to the class of interest. This is in line with the above observation that PCAM$^-$, and a bit less PCAM$^\pm$, are stronger in highlighting contextual information.

4.2.2 Comparison with previous works

Saliency maps have been produced for all baseline methods listed in Sect. 4.1. A sample of this comparison is presented in Fig. 6. For ease of view, this figure restricts to Grad-CAM, Score-CAM, Integrated Gradient, SmoothGrad, Occlusion, RISE and the three Poly-CAM variants. A comparison with all the methods listed in Sect. 4 can be found in Appendix. We can see that Poly-CAM methods accurately identify quite relevant elements in the image like a spider net or the pipes of an organ. CAM and perturbation methods cannot achieve this level of precision, gradient base methods highlight elements of the image that are not related to the class, while Zoom-CAM and Layer-CAM increase the resolution but are more noisy, halfway between gradient and more classical CAM-based methods. The spotted salamander is highlighted by all methods but the Poly-CAM methods are the only ones to identify the spots. The oranges are identified by Poly-CAM methods while the smile sketched on them is correctly excluded. This is in contrast with other methods that are either too low resolution, or do not exclude the smile, while SmoothGrad seems to give more importance to the smile than to the texture of the orange.

4.2.3 Industrial defect localization

To illustrate the importance of fine details identification in the industry, we also performed an experiment involving defect detection on stone tiles. The used dataset is the Stone Tiles dataset,[2] which consists of 605 labeled images of stone tiles. The labels are either "Good", "Broken", "Damaged" or "Glued". We split the dataset randomly into a validation set (10% of the images) and a training set (90% of the images). We fine tuned a VGG16 model on this dataset, pretrained on ImageNet from the PyTorch model zoo. The model was trained for 50 epochs with a fixed learning rate of 0.001, without early stopping, that's an accuracy of 89.7%. Score-CAM and $PCAM^\pm$ were performed on the validation set images for the classes predicted by the models. We compared the saliency maps produced by ScoreCAM and the proposed method for images that are classified as other than "Good", either correctly or wrongly, to assess if the explanations given by the two methods allow to better grasp the reasons of the classification.

Figure 6 show explanation for correct classifications of stone tiles. Both ScoreCAM and PolyCAM allow to understand the pixels related to a 'Broken' stone tile in the image but PolyCAM is more precise. For 'Damaged' and 'Glued' labels, the area highlighted by Score-CAM are generally very wide since the defects can be distributed on the whole image. ScoreCAM maps thus bring little information about what the model learned to detect theses labels. On the other hand, Poly-CAM show more fine details for those classes and allow to identify that the model use the scratches on the surface of the 'Damaged' tile and the small traces of glue on the 'Glued' tile to make a classification. Figure 7 shows the two samples that are erroneously classified as defectives. For both images, Poly-CAM help to identify more precisely than Score-CAM elements in the images that can be interpreted by the model as 'Broken' or 'Damaged'

[2] Downloaded from the Euresys website: https://www.euresys.com

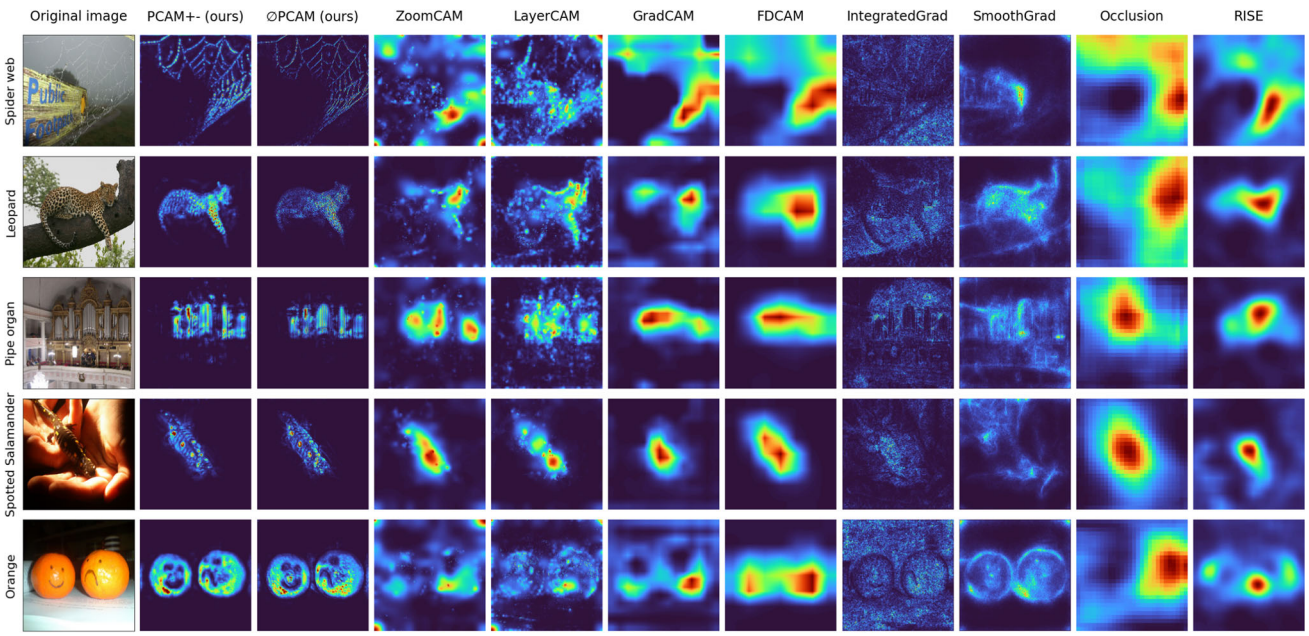

Fig. 5 Visual comparison of methods (see Appendix for full version). The compared methods are ours PCAM$^{\pm}$ and ∅PCAM, Zoom-CAM [15], Layer-CAM [16], Grad-CAM [12], FD-CAM [26], IntegratedGradient [9], SmoothGrad [10], Occlusion [8] and RISE [19]. A description of this figure is available in Sect. 4.2.2

Table 1 Faithfulness metrics for all methods: CAM-based, gradient and perturbation methods

Methods	VGG16			ResNet50		
	Insertion	Deletion	Ins-Del	Insertion	Deletion	Ins-Del
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**
∅PCAM (ours)	0.60	0.17	0.43	0.66	0.27	**0.39**
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
FDCAM	0.60	0.20	0.40	–	–	–
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
IntegratedGradient	0.41	**0.10**	0.31	0.52	**0.16**	0.36
InputXGrad	0.37	0.12	0.26	0.47	0.18	0.28
SmoothGrad	0.54	0.20	0.34	0.62	0.29	0.33
RISE	**0.62**	0.18	0.44	**0.67**	0.28	**0.39**
Occlusion	**0.62**	0.23	0.39	0.66	0.33	0.33

Insertion (higher is better), deletion (lower is better) and Insertion-Deletion (higher is better) with VGG16 and ResNet50 on the 2012 ILSVRC validation set. **Boldface** and underline indicate the best result and the best result amongst CAM-based methods respectively. Comparison of our Poly-CAM methods with gradient methods: Input X Gradient [38], Integrated Gradient [9], SmoothGrad [10], perturbation methods: Occlusion [8] and RISE [19], and CAM methods: Grad-CAM [12], Grad-CAM++ [13], Score-CAM [14], SS-CAM [20], IS-CAM [23], Smooth Grad-CAM++ [37], FD-CAM [26], Zoom-CAM [15] and Layer-CAM [16]. FD-CAM source code was only compatible with VGG16 at the time of writing

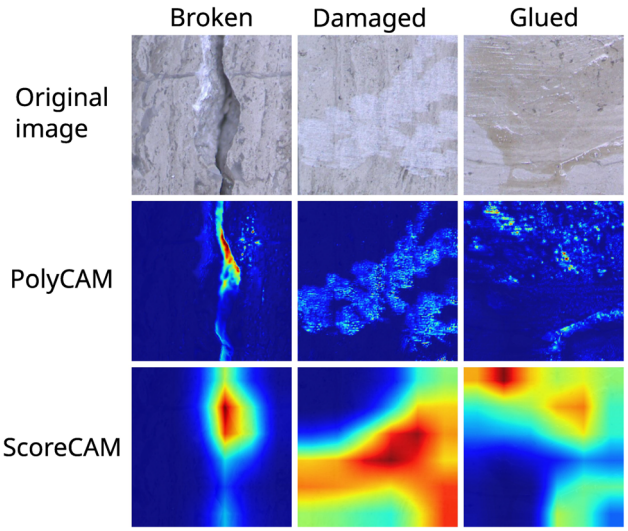

Fig. 6 Correct stone tile classifications explanations. The figure show representative Poly-CAM and Score-CAM explanations of correctly classified images of stone tiles using VGG16. Poly-CAM show a more precise identification of the causal elements of the classification for the three classes, in particular for Damaged and Glued tiles where the structures of the scratches and the glue are more clearly identified

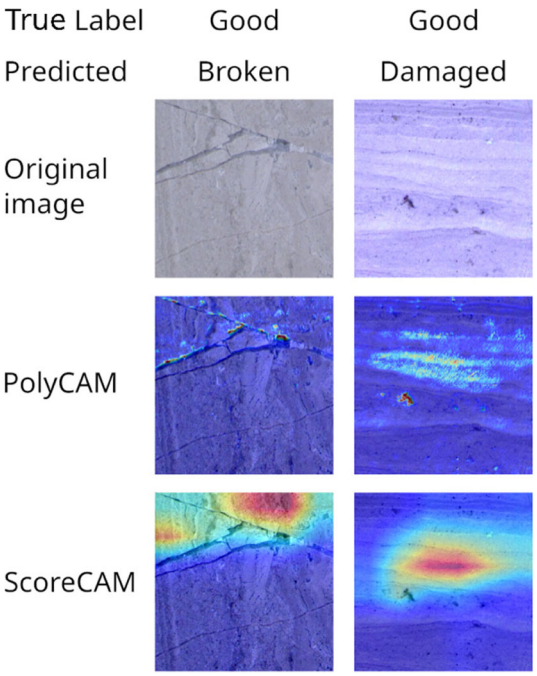

Fig. 7 False positive images of stone tiles. This figure show explanations of images labeled as good but falsely labeled as broken or damaged. Poly-CAM supports, i.e. explains, the wrong decision. We can identify using Poly-CAM that the dark veins are interpreted as fracture lines for the left image, while the lighter areas are shown as scratches for the right images. In comparison, Score-CAM also highlight roughly the veins on the left image, but the explanation on the right image is much less clear

4.3 Faithfulness quantitative assessment

There is still a lack of consensus regarding the metrics to assess the relevance of saliency maps for explainability [41]. The ability to segment the semantic object justifying the class label was used as a proxy to evaluate saliency maps [12], but it has been observed in [19] that the segmentation mask might not be correlated to the discriminant visual features justifying the class label. The same authors also introduced the insertion and deletion metrics to measure the increase (resp. decrease) in the class softmax output when inserting (resp. removing) pixels in decreasing order of saliency. These metrics are widely used in recent works [14, 20, 23]. They are thus considered in this work, even if they remain arguable, as discussed in the subsequent sections.

4.3.1 Metrics definition

Insertion Formalized by [19], the insertion metric measures how fast the model softmax output increases when adding the salient image pixels to a baseline image. The pixels of a baseline (uniform black/grey or highly blurred version of the image) are replaced by the image ones in decreasing order of saliency map value, allowing to define the faithfulness metric as the area under the curve of the class softmax output, observed as a function of the proportion of pixels inserted. The higher the metric the better. A blurred version of the image is generally preferred to a uniform color baseline, since it prevents the introduction of sharp artificial edges that might disrupt the predictions.

Deletion The deletion metric [19] measures the decrease in class softmax output while removing pixels. The pixels are progressively removed from the image and replaced by a baseline using the same logic as the insertion metric. The intuition behind the deletion metric is that removing the more important pixels in the saliency map should more rapidly decrease the softmax output of the target class prediction. The lower the metric, the better.

Insertion - Deletion A good explanation map should have a good insertion map and a good deletion map. A combination of the two scores is thus proposed here by computing the difference between insertion (higher is better) and deletion (lower is better) to obtain a combined score that should be maximized.

It is worth noting that, despite they are widely used, those metrics have some drawbacks. In particular, the insertion and deletion procedures are likely to result in outliers, i.e. in images that do not match the distribution of natural images. To mitigate this problem, as most previous works, we have used blurred baselines when implementing the insertion/deletion process. However, it is not sufficient to ensure

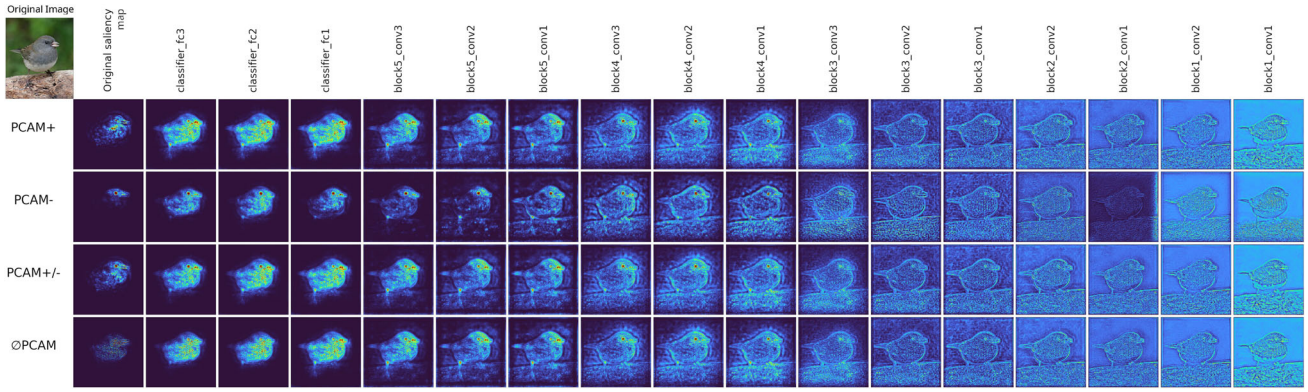

Fig. 8 Cascading randomization of VGG16. Sanity check on Poly-CAM methods [42]. Progression from left to right corresponds to a progressive, layer-wise randomization of the network parameters. It show how the saliency map changes with increasing up to complete randomization of the VGG16 model, starting by the last layer up to the first layer. The methods are sensible to model randomization, which mean they pass this sanity check. It is interesting to note that the class specificity is lost rapidly after randomising the first classifier layer, then more and more features are lost while randomization progress up to the first layer of the network

that a high correlation between the network prediction and the salient pixels means that those pixels exactly correspond to the whole set of class-discriminant visual features. Hence, these quantitative metrics should never be used as a substitute for a visual assessment of the saliency map, which remains the gold standard when comparing saliency map generation methods.

For the above metrics, 224 steps were performed with 224 pixels inserted or removed at each step. The used baseline for all the metrics is a blurred version of the input image using a Gaussian kernel of size 11x11 with a sigma of 5.

4.3.2 Quantitative assessment

Table 1 compares the faithfulness metrics for all methods. Among the Poly-CAM variants, PCAM$^{\pm}$ gives the best results compared to PCAM^{+}, PCAM^{-} and ØPCAM for all metrics on VGG16. On ResNet50, The results are very close for all the variants, PCAM$^{\pm}$ gives a insertion metric similar to PCAM^{+} and slightly superior to PCAM^{-} and ØPCAM, while PCAM^{-} and ØPCAM give a better deletion metric compared to PCAM^{+} and PCAM$^{\pm}$. For insertion-deletion on ResNet50, PCAM$^{\pm}$, ØPCAM and PCAM^{-} are on par.

Interestingly, the insertion metrics of PCAM$^{\pm}$ is systematically better than all other CAM-based approaches. Compared to the non-CAM methods, the PCAM$^{\pm}$ method gives similar or better insertion results than perturbation or gradient methods, respectively.

In terms of deletion, PCAM$^{\pm}$ tends to perform better than most other CAM-based methods, but appears to be weaker than gradient-based methods. InputXGrad and Integrated-Gradient achieve at the same time very poor results on the insertion metric and thus have a poor insertion-deletion. This is not surprising since gradient-based methods give lots of importance to the parts of the input that largely impact the loss and thus the output. As a consequence, the deletion metric is (trivially) good for those methods since this metric measures the decrease of output when important parts are removed from the input. The poor insertion metric however reveal that the parts that are considered as being important by gradient methods are not sufficient to explain the network prediction. This observation reveals the limits of the metrics when applied to dissimilar kinds of techniques.

4.4 Sanity check and robustness

We followed the method in [42] to implement a sanity check of our method. It consists in progressive, iterative, layer-wise randomization of the network parameters while regenerating an explanation after the randomization of each additional layer to evaluate the influence of the model's parameters on the explanation. So, the PCAM saliency maps have been visualized at each step of a cascading randomization of a VGG16 network, from last to first layer. The purpose of this sanity check is to verify that the PCAM methods do not work as edge detectors, and effectively derives class-specific saliency maps. All PCAM variants successfully passed the test, as shown in Fig. 8.

To evaluate the robustness of our explanation method, a sensitivity analysis has also been run, following the methodology introduced in [43] and [44]. The approach consists in assessing the extent of change in the saliency map produced when small perturbations are introduced into the input image. A method with a higher sensitivity is more prone to adversarial attacks. This is important to evaluate in the objective of real world application where adversarial attacks are possible [45, 46]. Results are presented in Table 2. They reveal that PCAM has a small explanation sensitivity, similar to

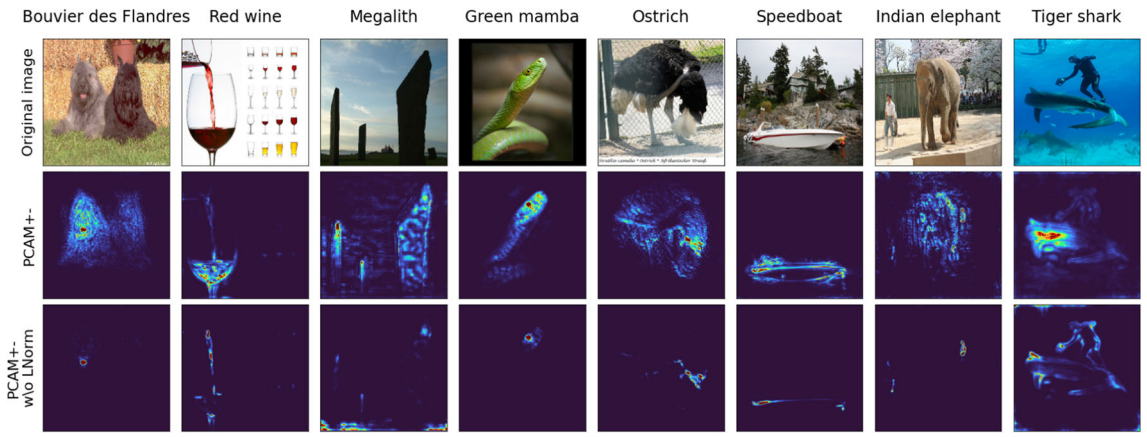

Fig. 9 Visual comparison of PCAM$^{\pm}$ with and without LNorm. Without LNorm the visualisation tends to concentrate on very focal elements of the images like eyes or mouth. Sometime some elements of the image that are not in object of the target class become also highlighted, like an object behind the elephant, or the diver next to the shark

Table 2 Sensitivity max table

Method	Sensitivity max	
	VGG16	ResNet50
IntegratedGradient	0.3576	0.5299
InputXGrad	0.6132	0.7225
SmoothGrad	5.6824	7.7777
RISE	0.7864	0.7841
Occlusion	2.5176	3.4378
GradCAM	0.0625	0.0212
GradCAM++	0.0525	0.0199
SmoothGradCAM++	0.5451	0.1594
ScoreCAM	0.0466	0.0193
ISCAM	0.0433	0.0334
FDCAM	0.0433	-
ZoomCAM	0.0987	0.0485
LayerCAM	0.0937	0.0590
PCAM^{+} (Ours)	0.0650	0.0262
PCAM^{-} (Ours)	0.0837	0.0578
PCAM$^{\pm}$ (Ours)	0.0659	0.0659
ØPCAM (Ours)	0.0783	0.0419

Sensitivity max metric measures maximum sensitivity of an explanation using Monte Carlo sampling-base approximation [44]. Captum implementation was used with 10 perturbations per input and a epsilon radius of a L-Infinity ball set to 0.02 for sampling (defaults parameters from the implementation) [39]. The compared methods are the three Poly-CAM variants proposed in this paper (PCAM^{+}, PCAM^{-}, PCAM$^{\pm}$), Zoom-CAM [15], Layer-CAM [16], Grad-CAM [12], Grad-CAM++ [13], Smooth Grad-CAM++ [37], Score-CAM [14], SS-CAM [20], IS-CAM [23], FD-CAM [26], Input X Gradient [38], IntegratedGradient [9], SmoothGrad [10], Occlusion [8], RISE [19]

Table 3 Execution time per image

Batch size	128	64	32
PCAM^{+} (ours)	4.32	4.38	4.67
PCAM^{-} (ours)	4.38	4.68	4.71
PCAM$^{\pm}$ (ours)	8.48	8.92	9.60
ØPCAM (ours)	1.51	1.61	2.01
ScoreCAM	1.48	1.52	1.64
ISCAM	12.85	13.29	14.18
SSCAM	47.13	48.01	52.12
FDCAM	0.17	0.17	0.17
RISE	OOM*	OOM*	18.50
Occlusion	2.62	2.62	2.62

Execution speed to compute the saliency maps for a VGG16, expressed in second per image. Mean over 100 computations.*OOM: Out of Memory

the ones obtained by other CAM-based methods, and one or two orders of magnitude below the sensitivities obtained by gradient-based and perturbation methods.

4.5 Ablation study on LNorm

The importance of including the LNorm operator in Eq. 3 is challenged in this section. We produced saliency maps using both the complete method and a variant where LNorm has been ablated. Formally, the saliency map of the LNorm ablated method becomes

$$\overline{P}_l^c = \begin{cases} CAM_l^c & \text{for } l = L \\ \left(CAM_l^c, \frac{s_{l+1}}{s_l}\right) \odot \uparrow_{bi} \left(\overline{P}_{l+1}^c, \frac{s_{l+1}}{s_l}\right) & \text{for } 1 \leq l \leq L-1 \end{cases}, \quad (9)$$

Representative examples are presented in Fig. 9 to compare the conventional and ablated PCAM. We clearly observe that the ablated method tends to ignore some of the class-relevant features, and to focus on a limited set of highly contrasted features (such as eye, mouth, or beak).

4.6 Speed of execution

Using the same experimental setup described in the previous sections, we measured the average average execution time per image over the first 100 images of the ILSVRC2012 validation set for both the perturbation-based methods and the subset of CAM-based methods relying on perturbations. Measurements were repeated for batch sizes of 32, 64, and 128 images on an 8Gb nvidia RTX3070M and reported in Table 3.

The results show that the different PolyCAM variants fall between RISE and ScoreCAM in terms of computation time. FDCAM is the fastest method using perturbations, this is understandable since the perturbations only need to be performed for the head of the network and don't require to recompute the whole backbone, but is also limited to the resolution of the last convolutional layer. It is interesting to note that the ØPCAM method, that compute scores similarly to FDCAM, is the fastest of or four variant, with a similar execution speed to ScoreCAM. ØPCAM seems to be the most appropriate compromise in terms of high resolution saliency maps quality and execution time.

5 Conclusion

Our Poly-CAM method produces high resolution saliency maps without relying on gradient backpropagation. Two variants of our Poly-CAM framework are investigated. In the first variant, the values weighting the activation maps are obtained by masking or unveiling image pixels, or both. In the second variant, an approach using perturbations inside of the network by switching off and on convolution channels is implemented. Our experiments reveal that a strategy combining masking and unveiling, either in the pixel space or at the level of channels, provides the more versatile solution. It achieves state of the art performances in term of faithfulness insertion-deletion metrics and outperforms current available methods in term of precision of visualization. As a main original contribution, our method allows for the high-resolution visualization of image regions that contribute to the network prediction. The channel switching strategy having the advantage of being quicker to compute. Despite our work is a valuable step towards a more explainable AI, there is still plenty of room for improvement in this domain. One of the questions raised by this work is related to the way the importance of a pixel should be quantified.

Indeed, the importance of a group of pixels appears to be different when this group is removed or when it is inserted (for example the importance of contextual information is more important when removing it than when inserting it), which can not be properly reflected by a single saliency map.

While our work focuses on post-hoc explanations of classical CNN models using saliency maps, it is worth mentioning that other research directions attempt to give sense to neural network decisions by developing architectures that are naturally suited for interpretability [2].

Appendix A Section title of first appendix

A.1 Visual comparison with previous work

Figure 10 display a visual comparison with all the compared methods.

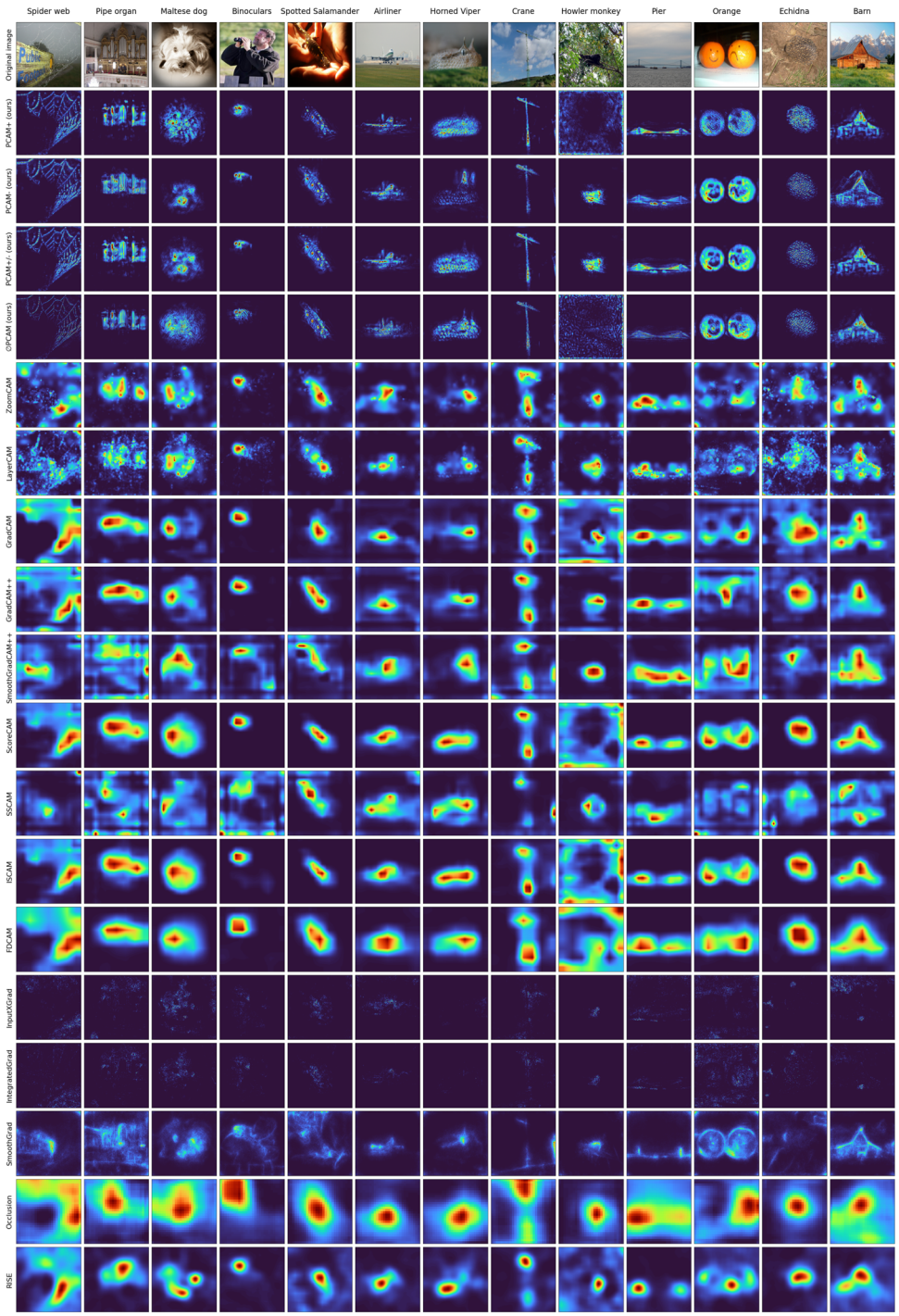

Fig. 10 Visual comparison of methods. The compared methods are the three Poly-CAM variants proposed in this paper (PCAM^{+}, PCAM^{-}, PCAM$^{\pm}$ and ØPCAM), Zoom-CAM [15], Layer-CAM [16], Grad-CAM [12], Grad-CAM++ [13], Smooth Grad-CAM++ [37], Score-CAM [14], SS-CAM [20], IS-CAM [23], Input X Gradient [38], IntegratedGradient [9], SmoothGrad [10], Occlusion [8], RISE [19]

Acknowledgements This work was performed in UCLouvain in Belgium. It was funded by the FNRS, including a FRIA funding to Alexandre Englebert. Computational resources have been provided by the supercomputing facilities of the Université catholique de Louvain (CISM/UCL) and the Consortium des Équipements de Calcul Intensif en Fédération Wallonie Bruxelles (CÉCI) funded by the Fond de la Recherche Scientifique de Belgique (F.R.S.-FNRS) under convention 2.5020.11 and by the Walloon Region We also want to thank the authors of Zoom-CAM for their kind help in using their method.

Author Contributions A.E. made the code and run the experiments. A.E, O.C. and C.D. analysed the results. A.E. and C.D. wrote the manuscript. All the authors reviewed the manuscript.

Funding Fonds De La Recherche Scientifique–FNRS.

Data availability This paper uses the ImageNet dataset https://image-net.org/, and the Stone Tiles dataset from Euresys (https://downloads.

euresys.com/PackageFiles/OPENEVISION/24.02.0.27377WIN/976684208/Deep_Learning_Additional_Resources_24.02.0.27377.zip.)

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

References

1. Adadi, A., Berrada, M.: Peeking inside the black-box: a survey on explainable artificial intelligence (XAI). IEEE Access **6**, 52138–52160 (2018)
2. Wang, W., Han, C., Zhou, T., Liu, D.: Visual recognition with deep nearest centroids (2022). arXiv preprint arXiv:2209.07383
3. Song, X., Wu, N., Song, S., Zhang, Y., Stojanovic, V.: Bipartite synchronization for cooperative-competitive neural networks with reaction-diffusion terms via dual event-triggered mechanism. Neurocomputing **550**, 126498 (2023)
4. Song, X., Sun, P., Song, S., Stojanovic, V.: Quantized neural adaptive finite-time preassigned performance control for interconnected nonlinear systems. Neural Comput. Appl. **35**(21), 15429–15446 (2023)
5. Song, X., Peng, Z., Song, S., Stojanovic, V.: Anti-disturbance state estimation for pdt-switched Rdnns utilizing time-sampling and space-splitting measurements. Commun. Nonlinear Sci. Numer. Simul. **132**, 107945 (2024)
6. Samek, W., Montavon, G., Lapuschkin, S., Anders, C.J., Müller, K.-R.: Explaining deep neural networks and beyond: a review of methods and applications. Proc. IEEE **109**(3), 247–278 (2021)
7. Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., Müller, K.-R.: Unmasking Clever Hans predictors and assessing what machines really learn. Nat. Commun. **10**(1), 1–8 (2019)
8. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: European Conference on Computer Vision. Springer (2014)
9. Sundararajan, M., Taly, A., Yan, Q.: Axiomatic attribution for deep networks. In: International Conference on Machine Learning. PMLR (2017)
10. Smilkov, D., Thorat, N., Kim, B., Viégas, F., Wattenberg, M.: Smoothgrad: removing noise by adding noise (2017). arXiv preprint arXiv:1706.03825
11. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
12. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: visual explanations from deep networks via gradient-based localization. In: Proceedings of the IEEE International Conference on Computer Vision (2017)
13. Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N.: Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV). IEEE (2018)
14. Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., Hu, X.: Score-CAM: score-weighted visual explanations for convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Workshop on Fair, Data Efficient and Trusted Computer Vision (2020)
15. Shi, X., Khademi, S., Li, Y., Gemert, J.: Zoom-cam: generating fine-grained pixel annotations from image labels. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 10289–10296, IEEE (2021)
16. Jiang, P.-T., Zhang, C.-B., Hou, Q., Cheng, M.-M., Wei, Y.: Layercam: exploring hierarchical class activation maps for localization. IEEE Trans. Image Process. **30**, 5875–5888 (2021)
17. Englebert, A., Cornu, O., Vleeschouwer, C.: Backward recursive class activation map refinement for high resolution saliency map. In: 2022 26th International Conference on Pattern Recognition (ICPR). IEEE (2021)
18. Stassin, S., Englebert, A., Albert, J., Nanfack, G., Versbraegen, N., Frénay, B., Peiffer, G., Doh, M., Riche, N., De Vleeschouwer, C.: An experimental investigation into the evaluation of explainability methods for computer vision. Communications in Computer and Information Science (2023)
19. Petsiuk, V., Das, A., Saenko, K.: RISE: Randomized Input Sampling for Explanation of Black-box Models. In: British Machine Vision Conference (BMVC) (2018). http://bmvc2018.org/contents/papers/1064.pdf
20. Wang, H., Naidu, R., Michael, J., Kundu, S.S.: SS-CAM: Smoothed score-CAM for sharper visual feature localization (2020). arXiv preprint arXiv:2006.14255
21. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2014). arXiv preprint arXiv:1409.0473
22. Yamauchi, T., Ishikawa, M.: Spatial sensitive GRAD-CAM: visual explanations for object detection by incorporating spatial sensitivity. In: 2022 IEEE International Conference on Image Processing (ICIP), pp. 256–260, IEEE (2022)
23. Naidu, R., Ghosh, A., Maurya, Y., Kundu, S.S., et al.: IS-CAM: integrated score-CAM for axiomatic-based explanations (2020). arXiv preprint arXiv:2010.03023
24. Ibrahim, R., Shafiq, M.O.: Augmented score-CAM: high resolution visual interpretations for deep neural networks. Knowl. Based Syst. **252**, 109287 (2022)
25. Ramaswamy, H.G., : Ablation-cam: visual explanations for deep convolutional network via gradient-free localization. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 983–991 (2020)
26. Li, H., Li, Z., Ma, R., Wu, T.: FD-CAM: improving faithfulness and discriminability of visual explanation for CNNs (2022). arXiv preprint arXiv:2206.08792
27. Tagaris, T., Sdraka, M., Stafylopatis, A.: High-resolution class activation mapping. In: 2019 IEEE International Conference on Image Processing (ICIP). IEEE (2019)
28. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4700–4708 (2017)
29. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. (IJCV) **115**(3), 211–252 (2015). https://doi.org/10.1007/s11263-015-0816-y
30. Dahl, G.E., Sainath, T.N., Hinton, G.E.: Improving deep neural networks for LVCSR using rectified linear units and dropout. In: 2013 IEEE International Conference on Acoustics, Speech and Signal Processing. IEEE (2013)
31. Ronneberger, O., Fischer, P., Brox, T.: U-net: convolutional networks for biomedical image segmentation. In: International Conference on Medical Image Computing and Computer-assisted Intervention. Springer (2015)

32. Lin, T.-Y., Dollár, P., Girshick, R., He, K., Hariharan, B., Belongie, S.: Feature pyramid networks for object detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 2117–2125 (2017)
33. Collin, A.-S., De Vleeschouwer, C.: Improved anomaly detection by training an autoencoder with skip connections on images corrupted with stain-shaped noise. In: 2020 25th International Conference on Pattern Recognition (ICPR), pp. 7915–7922, IEEE (2021)
34. Liu, D., Cui, Y., Yan, L., Mousas, C., Yang, B., Chen, Y.: Densernet: weakly supervised visual localization using multi-scale feature aggregation. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 35, pp. 6101–6109 (2021)
35. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint arXiv:1409.1556
36. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2016)
37. Omeiza, D., Speakman, S., Cintas, C., Weldermariam, K.: Smooth grad-cam++: an enhanced inference level visualization technique for deep convolutional neural network models (2019). arXiv preprint arXiv:1908.01224
38. Shrikumar, A., Greenside, P., Shcherbina, A., Kundaje, A.: Not just a black box: learning important features through propagating activation differences (2016). arXiv preprint arXiv:1605.01713
39. Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., Yan, S., Reblitz-Richardson, O.: Captum: a unified and generic model interpretability library for PyTorch (2020)
40. Fernandez, F.-G.: TorchCAM: Class Activation Explorer. GitHub, San Francisco (2020)
41. Poursabzi-Sangdeh, F., Goldstein, D.G., Hofman, J.M., Wortman Vaughan, J.W., Wallach, H.: Manipulating and measuring model interpretability. In: Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems (2021)
42. Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B.: Sanity checks for saliency maps (2018). arXiv preprint arXiv:1810.03292
43. Ghorbani, A., Abid, A., Zou, J.: Interpretation of neural networks is fragile. In: Proceedings of the AAAI Conference on Artificial Intelligence (2019)
44. Yeh, C.-K., Hsieh, C.-Y., Suggala, A., Inouye, D.I., Ravikumar, P.K.: On the (in) fidelity and sensitivity of explanations. Adv. Neural Inf. Process. Syst. **32**, 10967–10978 (2019)
45. Cheng, Z., Liang, J., Choi, H., Tao, G., Cao, Z., Liu, D., Zhang, X.: Physical attack on monocular depth estimation with optimal adversarial patches. In: European Conference on Computer Vision, pp. 514–532, Springer (2022)
46. Cheng, Z., Choi, H., Feng, S., Liang, J.C., Tao, G., Liu, D., Zuzak, M., Zhang, X.: Fusion is not enough: single modal attack on fusion models for 3d object detection. In: The 12th International Conference on Learning Representations (2023)

Alexandre Englebert MD, is a PhD candidate at UCLouvain in Belgium, where he received his medical degree in 2017. His current research focuses on the development and application of deep learning techniques in the medical field, with a particular emphasis on explicable AI and multimodal self-supervision. In parallel, he is pursuing a specialization in orthopedic surgery. His research interests lie in the broader field of deep learning, especially in the context of limited annotated data, with a focus on healthcare applications and the potential of AI to improve patient outcomes.

Olivier Cornu is Orthopaedic and Trauma surgeon at the Cliniques Universitaires Saint-Luc UCL in Brussels and Ordinary Professor at UCLouvain. He is member of the Belgian Royal Academy of Medecine. He has been deploying his expertise in the fields of musculoskeletal management of infectious pathology of the musculoskeletal system, to the reconstruction of large bone defects and revision joint replacement surgery. His research focuses on the study of the mechanical and biological properties of bone allografts and bone substitutes, bone reconstruction and implant-related infections, with an interest in treatments against bacterial biofilm. He also pursues numerous works oriented towards orthopaedic and trauma care management. He is author and co-author of more than 150 articles. He is an active member of several national and international scientific associations. He is board member of the Société Royale Belge de Chirurgie Orthopédique (SORBCOT) and past-president of the Belgian Orthopaedic Trauma Association. He is Associate Editor of the "Acta Orthopaedica Belgica" Journal.

Christophe De Vleeschouwer born on the 8th of December 1972, is a Research Director of the Belgian NSF, and a Full Professor at 'Université catholique de Louvain' (UCL). He received the M. Eng. and the Ph. D. degrees from UCL, in 1995 and 1999 respectively. As a young PhD, he has been a senior research engineer with the IMEC Multimedia Information Compression Systems group (1999-2000, working with ERICSSON). Then, he was a Belgian NSF postdoctoral Research Fellow at UC Berkeley (2001-2002, Prof. Avideh Zakhor), a Swiss NSF postdoctoral researcher at Ecole Polytechnique Fédérale de Lausanne (2004-2005, Prof. Pascal Frossard), and a visiting scholar at Carnegie Mellon University (2014-2015, Robotics Institute, Prof. Martial Hebert). He is interested in video and image processing for visual scene interpretation and content management, including transmission, flexible and personalized access, and autonomous acquisition/editing. He is also enthusiastic about deep learning in artificial intelligence, non-linear signal expansion techniques, graph-based formalism, and their use for image signal analysis.