ReciproCAM: Lightweight Gradient-free Class Activation Map for Post-hoc Explanations

Seok-Yong Byun
Intel Corp.
Seoul, South Korea
mark.byun@intel.com

Wonju Lee
Intel Corp.
Seoul, South Korea
wonju.lee@intel.com

Abstract

To interpret model behavior, AI practitioners have shed light on explainable AI techniques. While visual explanations like class activation maps (CAM) and its derivatives have demonstrated promise, their applicability within post-hoc frameworks is often constrained by architectural limitations, gradient computation capabilities, or slow execution speeds. In this paper, we propose a lightweight gradient-free ReciproCAM by spatially perturbing the internal feature map to exploit the correlation between activations and a model output. From the numerical results, we achieve the gains of 1.78 *to* 3.72% *in the ResNet family compared to ScoreCAM in average drop-coherence-complexity metric, excluding the VGG-16 (*1.39% *drop), while ReciproCAM exhibits* 148 *times faster than ScoreCAM.*

1. Introduction

While deep learning demonstrates exceptional performance, its adoption in mission-critical domains remains cautious due to the opaqueness of its internal decision-making mechanisms. Consequently, there is a growing demand for interpretable or explainable AI (XAI) technology, enabling the analysis of model behavior and the identification of potential bias or errors in a model or data.

Significant advancements have been achieved in the field of computer vision, particularly with the emergence of CAM [24]. Despite exhibiting high performance in saliency map generation and rapid execution, CAM faced limitations in selecting appropriate model architectures. To address this challenge, GradCAM [17] and its variants [2, 7, 12] were developed. However, these methods, relying on gradient calculations, encountered difficulties when applied in post-deployment frameworks such as ONNX [1] or OpenVINO [9].

Meanwhile, black-box XAI algorithms have emerged for analyzing model behavior within such post-deployment frameworks. These algorithms do not necessitate additional calculations during model inference or access to internal activations but instead detect changes in model output through input perturbation and generate saliency maps. RISE [13], in particular, relies on Monte Carlo sampling of a random mask to approximate a true saliency map, typically requiring over thousands of inferences. Moreover, the number of inferences increases with input resolution or desired explanation quality. Extremal perturbation [6] introduces a gradient descent-based optimization method for mask parameters. Nevertheless, this also demands hundreds or thousands of iterations, consuming several seconds of GPU computing time to generate a saliency map for a single input image.

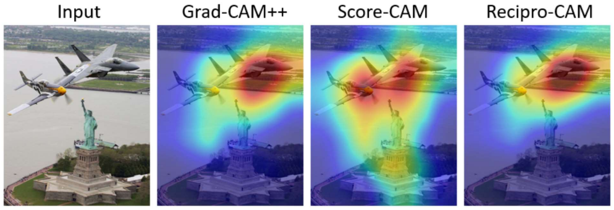

Figure 1. Comparison of resolution capabilities among Grad-CAM++, ScoreCAM, and ReciproCAM. The predicted class is ”aircraft carrier,” while the ground truth is ”warplane.” ScoreCAM fails to differentiate the warplane from nearby objects.

Recently, there has been a development in gray-box XAI algorithms that slightly relax the black-box constraint, allowing access to and perturbation of model internal activations during inference. AblationCAM [4], for instance, perturbs the model by accessing the feature map internally and removing specific channel information. It has firstly demonstrated the ability to outperform GradCAM without relying on gradients. ScoreCAM [21], regarded as the current state-of-the-art XAI algorithm, achieves high-resolution localization performance by forwarding activation-perturbed input. However, due to its reliance on the number of feature maps, ScoreCAM demands considerable computational resources, as it requires inference equivalent to the number of feature maps. Specifically, ScoreCAM is approximately 127× slower than CAM or GradCAM.

Inspired by CAM and RISE, we propose ReciproCAM by perturbing intermediate activations with spatial masks and observing the changes of model output. By generating spatial masks on an intermediate layer with a brute-force manner, we extremely enhance both localization performance and computational efficiency as illustrated in Figure 1.

The main contributions of this paper include:

- We present a novel gray-box XAI algorithm that leverages the reciprocal relationship between perturbed intermediate activations and the model output prediction.
- We provide benchmark results on various XAI performance metrics, i.e., drop/increase, deletion/insertion, and average drop, coherence, and complexity (ADCC). Specifically, ReciproCAM achieves state-of-the-art performance on the ADCC metric for all architectures, except for VGG-16.
- Our method provides 148× faster execution performance than ScoreCAM.

2. Related work

The pioneer work, CAM [24], generates a saliency map that highlights the important regions of an image for a particular class by multiplying a global average pooled activation vector with a fully connected weight vector specific to the class. Essentially, the saliency map S^c for a given class c is obtained by

$$S^c_{\text{CAM}} = \text{ReLU}\left(\sum_{k=1}^{K} w^{k,c} F^k\right) \tag{1}$$

where $w^{k,c}$ represents the weight parameter of the last linear layer connecting channel k to class c, and F^k denotes the channel k of the feature map extracted from the final convolutional layer. CAM, therefore, depends on the configuration of specific layers within the model architecture, specifically requiring a pooling layer followed by a fully-connected layer.

To address the limitation of CAM's architecture-specific requirements, Selvararju et al. [17] have proposed Grad-CAM inspired by gradient visualizations [19, 23]. Grad-CAM achieves class-discriminative localization by weighting the feature map with gradients, allowing for a more generalized approach as

$$S^c_{\text{GradCAM}} = \text{ReLU}\left(\sum_{k=1}^{K} \sum_{u,v} \frac{\partial y^c}{\partial f^k(u,v)} F^k\right), \tag{2}$$

where $f^k(u,v)$ represents the pixel at coordinates (u,v) within the feature map F^k, while y^c denotes the model's prediction result, prior to softmax, for class c. Gradient-based approaches, e.g., GradCAM++ [2], Axiom-based GradCAM [7], and Smooth GradCAM++ [12], have gained popularity as they offer solutions to the limitations of CAM while enhancing interpretability. However, due to their reliance on gradient computations, they are not suitable as visual explanation solutions for post-deployment frameworks like ONNX [1] or OpenVINO [9]. Furthermore, researchers have identified concerns such as saturation and false confidence associated with gradient-based methods [21].

ScoreCAM, introduced by Wang et al. [21], offers a solution to both the saturation and false confidence issues without requiring gradient computations. It achieves this by emphasizing channel-wise importance from a confidence perspective and utilizes the average drop metric concept to enhance confidence. The saliency map can be formulated as

$$S^c_{\text{ScoreCAM}} = \text{ReLU}\left(\sum_{k=1}^{K} h(F^k \odot x)^c F^k\right), \tag{3}$$

where $h(F^k \odot x)^c$ is the channel-wise increase of confidence score with an input image x, Hadamard product $\odot$, and model feedforward function h. While ScoreCAM addresses the needs of post-deployment frameworks, its application involves more than K forward passes, leading to a slowdown in visual explanations. On the other hands, Smooth ScoreCAM [20] and Integrated ScoreCAM [11] have emerged as enhancements, further improving localization performance.

In parallel, numerous studies have explored black-box approaches [3, 10, 13, 14] tailored specifically for post-deployment frameworks. These methods operate solely based on observing model outputs without requiring gradient computation or access to internal activations. Notably, Petsiuk et al. [13] introduced RISE, employing a Monte Carlo sampling approach to perturb input images with randomly generated masks. The saliency map can be represented as

$$S^c_{\text{RISE}} = \sum_{n=1}^{M} h(M_n \odot x)^c M_n, \tag{4}$$

where M^n is the n-th random mask. RISE's approximation of true saliency using confidence-weighted random masks means that its localization performance relies on both the number of masks (equivalent to the number of forward passes) and the perturbation resolution (i.e., the desired size of holes in a mask). Typically, explaining model behavior with RISE necessitates thousands of inferences.

3. ReciproCAM

In order to underscore the importance of individual activations within the feature map concerning the output prediction, we should selectively retain the activation at target pixels. This pixel corresponds to a specific region in the original input image as dictated by the receptive field. In this

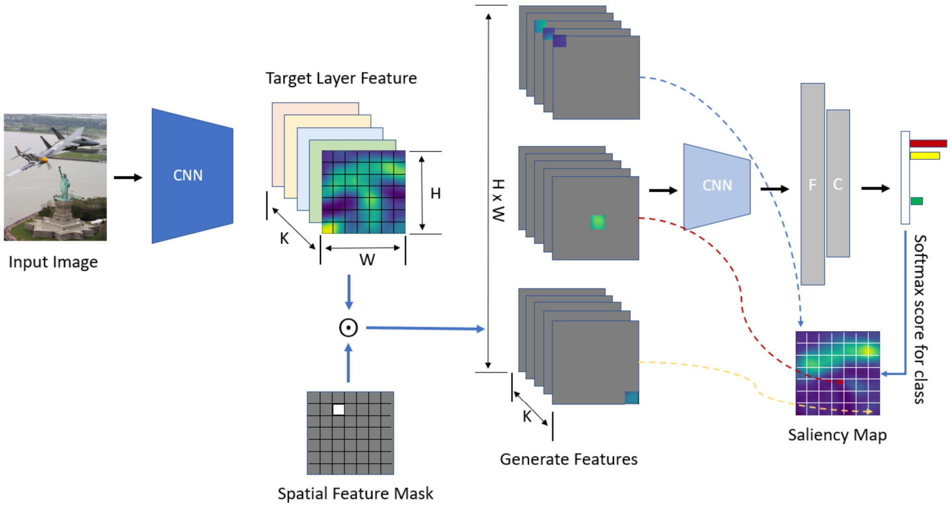

Figure 2. Overview of ReciproCAM. The feature map of the target layer is multiplied element-wise with a set of spatial masks, each having a single non-zero value at a unique spatial position within the feature map. This multiplication generates a new set of feature maps with the same height and width as the original feature map. These new feature maps are then passed as input to the subsequent part of the network. Finally, the predicted scores corresponding to the given class are collected and used to populate each position in the saliency map. Here, K represents the number of channels, H and W denote the height and width of the feature map.

paper, we shed light on the reciprocal relationship between activations and the output prediction by leveraging perturbations on the feature map. It is posited that each perturbed feature map comprehensively encapsulates diverse characteristics of the input image. Consequently, a feature map with multiple channels activated at a singular position is expected to furnish ample information for prediction, thereby minimizing information overlap from adjacent positions.

3.1. Spatially perturbed feature map generation

Unlike RISE, we aim to generate spatial masks in a brute-force manner to specify each single pixels in the feature map. That is, the generation of a spatial mask M^n involves designating a single pixel in the feature map as 1, while setting all other pixels to 0 as described in Figure 2. This process is iteratively applied to every pixel in the feature map, resulting in the creation of N spatial masks, where N is the product of the feature map's height and width, expressed as $N = H \times W$. As a result, each spatial mask is uniquely associated with a specific pixel position in the feature map. Instead of applying perturbation at the input like in RISE, we apply it in the middle of the network, allowing for overlaps according to receptive field in the original input. This approach offers both faster execution and higher saliency resolution.

By generating N distinct masked feature maps, we obtain varied perspectives of the original feature map, each highlighting a unique pixel position. This method enables a more comprehensive analysis of feature map activations, providing insights into the pixels that have the greatest impact on the final output. The n-th masked feature map corresponding to channel k is given by

$$\tilde{F}_n^k = M_n \odot F^k. \tag{5}$$

3.2. Saliency map generation

ReciproCAM formulates a saliency map by partitioning the network into two segments, delineated by the specified layer. The initial part, denoted as f, constitutes the feature extractor, whereas the subsequent layers are represented by g. Consequently, the saliency map S^c for a specific class c can be formulated as

$$S^c_{\text{ReciproCAM}} = \text{reshape}\left(\frac{\mathbf{y}^c - \min(\mathbf{y}^c)}{\max(\mathbf{y}^c) - \min(\mathbf{y}^c)}, (H,W)\right), \tag{6}$$

where the $N \times 1$ confidence vector $\mathbf{y}^c$ composed of $[y_1^c, \ldots, y_N^c]^T$ for the class c is transformed into $H \times W$ matrix with reshape function. Each element y_n^c is computed by

$$y_n^c = g([\tilde{F}_n^1, \cdots, \tilde{F}_n^K])^c. \tag{7}$$

We note that ReciproCAM requires only a single forward pass of f and N times of inference with g. In this paper, we choose a backbone network for f and a single linear classifier for g. It's noteworthy to consider the trade-off between localization performance and speed, which arises from the design choices for both f and g within the overall network. For instance, a smaller receptive field would accentuate individual pixels in the original input more prominently. However, this choice demands more mask generation and forward passes of g.

Table 1. Evaluation of various CAM-based approaches using existing metrics across six different backbone architectures. Evaluation scores for other CAM methods are sourced from [15]. The first rank in ADCC is highlighted in bold, while the second rank is indicated in blue.

	VGG-16							ResNet-18						
Method	Drop	Inc	Del	Ins	Coher	Compl	**ADCC**	Drop	Inc	Del	Ins	Coher	Compl	**ADCC**
	(↓)	(↑)	(↓)	(↑)	(↑)	(↓)	(↑)	(↓)	(↑)	(↓)	(↑)	(↑)	(↓)	(↑)
GradCAM	66.42	5.92	11.12	19.56	69.20	15.65	53.52	42.90	16.63	13.43	41.47	81.03	23.04	69.98
GradCAM++	32.88	20.10	8.82	36.60	89.34	26.33	75.65	17.85	34.46	12.30	44.80	98.18	44.63	74.24
SGradCAM++	36.72	16.11	10.57	31.36	82.68	28.09	71.72	20.67	29.99	12.83	43.13	97.53	43.11	74.20
ScoreCAM	26.13	24.75	9.52	47.00	93.83	20.27	**81.66**	12.81	40.41	10.76	46.01	98.35	41.78	77.30
ReciproCAM	21.51	34.86	9.50	46.88	92.24	27.48	80.27	20.68	36.30	10.19	44.93	97.38	33.60	**79.08**
	ResNet-50							ResNet-101						
GradCAM	32.99	24.27	17.49	48.48	82.80	22.24	75.27	29.38	29.35	18.66	47.47	81.97	22.51	76.40
GradCAM++	12.82	40.63	14.10	53.51	97.84	43.99	75.86	11.38	42.07	14.99	56.65	98.28	43.94	76.34
SGradCAM++	15.21	35.62	15.21	52.43	97.47	42.25	76.19	13.37	37.76	14.32	58.23	97.76	42.61	76.54
ScoreCAM	8.61	46.00	13.33	54.16	98.12	42.05	78.14	7.20	47.93	14.63	59.57	98.37	42.04	78.55
ReciproCAM	15.69	40.54	13.34	55.39	96.68	32.90	**80.84**	15.07	41.39	15.80	59.28	97.21	32.45	**81.38**
	ResNeXt-50							ResNeXt-101						
GradCAM	28.06	29.42	20.73	50.30	82.72	25.57	76.09	24.12	36.37	20.47	61.04	82.94	25.45	77.62
GradCAM++	11.12	41.38	17.07	56.06	97.30	48.66	73.16	9.74	42.63	17.63	62.90	95.05	46.27	74.61
SGradCAM++	12.70	36.58	16.90	56.76	97.32	47.48	73.58	9.49	40.43	17.67	64.16	96.81	49.24	73.03
ScoreCAM	7.20	45.70	15.59	57.92	98.00	46.86	75.38	5.37	47.70	17.30	63.61	97.03	46.83	75.60
ReciproCAM	13.70	40.82	18.94	58.93	96.37	37.36	**79.10**	12.03	42.69	20.25	64.70	97.50	35.62	**80.74**

4. Experiments

We here show the localization performance and the speed of ReciproCAM by quantitative and performance analysis in Sections 4.1 and 4.2, respectively. The qualitative analysis is given in Appendix 4.3.

4.1. Quantitative analysis

For quantitative analysis, we adopt the ADCC metric proposed by [15], following their experimental setup. We evaluate ReciproCAM on the ILSVRC2012 [16] validation set and compare our results with those of [15] across various architectures, including VGG-16 [18], ResNet-18/50/101 [8], ResNeXt-50/101 [22]. Each CAM approach is applied to the last block or convolution layer. The results obtained using ReciproCAM are integrated with those from [15], and the consolidated findings are presented in Table 1.

Table 1 illustrates that ReciproCAM attains SOTA performance on the ADCC metric across five architectures, except for VGG-16. A detailed examination of the ADCC metrics reveals that ScoreCAM achieves the highest average drop score across the five architectures, while ReciproCAM exhibits significantly lower complexity. This difference contributes to the SOTA performance in the ADCC metric. However, for VGG-16, the trend is reversed. While ScoreCAM generally outperforms the proposed method in terms of coherency, the difference is not significant enough to affect the ADCC ranking.

The insertion metric alternates between ReciproCAM and ScoreCAM taking the first and second positions, respectively. Similarly, the deletion metric typically yields results similar to the insertion metric, although in the ResNext model, ScoreCAM outperforms the proposed method noticeably. However, it's important to note that the authors [15] have demonstrated that metrics such as average drop, increase, insertion, and deletion may not adequately evaluate explainability, particularly when using FakeCAM. Our results also support this observation, highlighting the importance of the ADCC metric as a more comprehensive and distinguishable assessment tool.

Table 2. Execution time comparison of five methods. The execution time was measure with 1,000 inputs and calculated the average time, so the time is execution time for single image.

	Time (ms)	FPS	Ratio
ReciproCAM	13.8	72.46	-
GradCAM	16.0	62.50	1.16×
GradCAM++	16.2	62.73	1.17×
SGradCAM++	77.3	12.94	5.60×
ScoreCAM	2039.7	0.49	147.80×

4.2. Performance analysis

To the best of our knowledge, prior white-box XAI research had not addressed post-deployment frameworks, avoiding reliance on gradient computation or sacrificing execution speed. However, visual explanations typically occur with false alarms or missed detections in post-hoc analyses, necessitating consideration of execution time across XAI algorithms. We conducted an average of 1000 explanations on randomly selected images from the ILSVRC2012 dataset using the Torchvision ResNet-50 pretrained model. The hardware setup comprised an Nvidia RTX3090 GPU with an Intel i9-11900 CPU.

From Table 2, ReciproCAM demonstrates the best execution performance, while GradCAM and GradCAM++ exhibit similar performance to ReciproCAM. The slowest performance was observed with ScoreCAM, which is approximately 148 times slower than ReciproCAM.

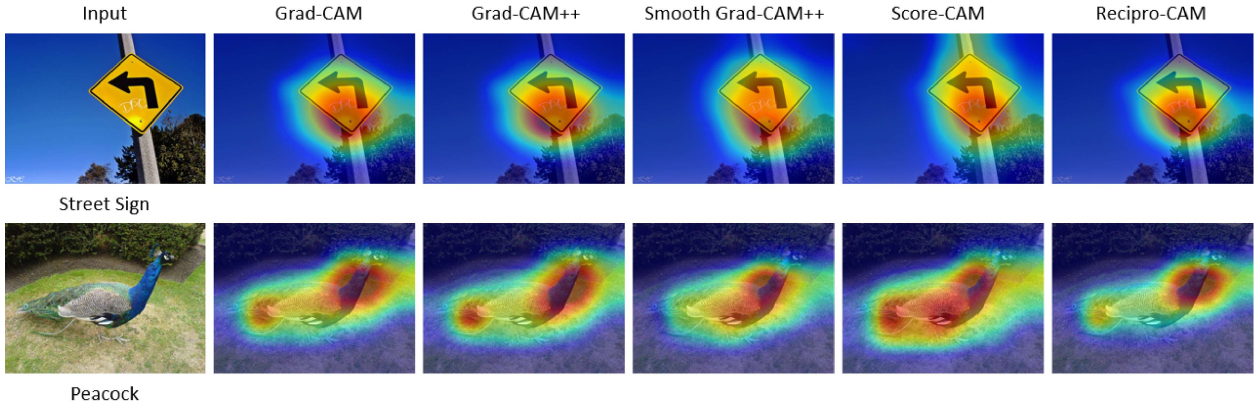

Figure 3. Single object CAM results. Street Sign and Peacock inputs process with GradCAM, GradCAM++, Smooth GradCAM++, ScoreCAM, and ReciproCAM to generate saliency maps.

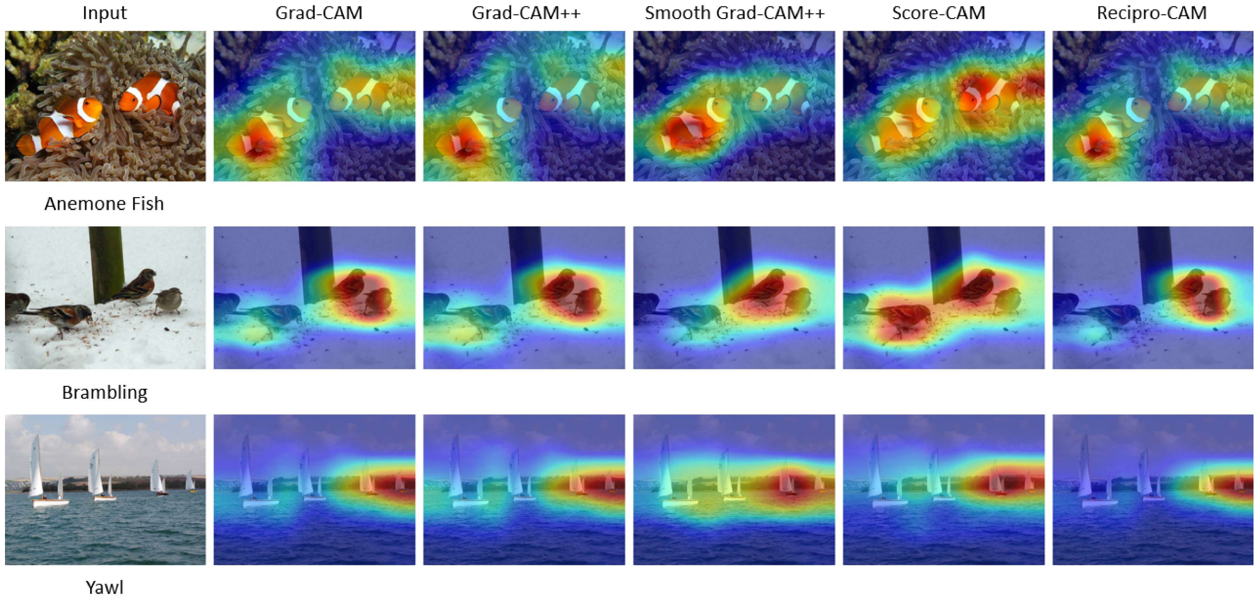

Figure 4. Same multiple objects CAM results. Anemone Fish, Brambling, and Yawl inputs process with GradCAM, GradCAM++, Smooth GradCAM++, ScoreCAM, and ReciproCAM to generate saliency maps.

4.3. Qualitative analysis

For the qualitative analysis of the proposed method compared to other CAM methods, we employed the ResNet-50 backbone architecture and the Torchvision ImageNet pre-trained model. We utilized the TorchCAM libarary [5] to evaluate various saliency maps, including GradCAM, GradCAM++, Smooth GradCAM++, and ScoreCAM. For a comprehensive analysis, we categorized input images from the ILSVRC2012 validation dataset into three groups: single object-only cases, multiple objects with the same identity, and multiple objects with different identities.

Explanations for single objects. In the first case as depicted in Figure 3, we observe notable distinctions among the saliency maps. For instance, when examining a street sign, all methods primarily focus on the bottom edge and nail head. However, ScoreCAM additionally highlights the upper pole, while ReciproCAM emphasizes a smaller saliency area corresponding to treetops.

Similarly, in the peacock example, ReciproCAM exhibits a saliency pattern akin to GradCAM and Grad-CAM++, with variations particularly noticeable in the tail region. Meanwhile, ScoreCAM covers the entire peacock, offering a more comprehensive insight.

Explanations for multiple objects with same identity. Moving to the second case, illustrated in Figure 4, we evaluate the localization capability of these methods. For instance, while ReciproCAM, GradCAM++, and GradCAM display relatively separated saliency maps for anemone fish, ScoreCAM showcases a connected map with strong activation around the coral. Smooth GradCAM exhibits a slightly connected map but with lower saliency scores for the coral

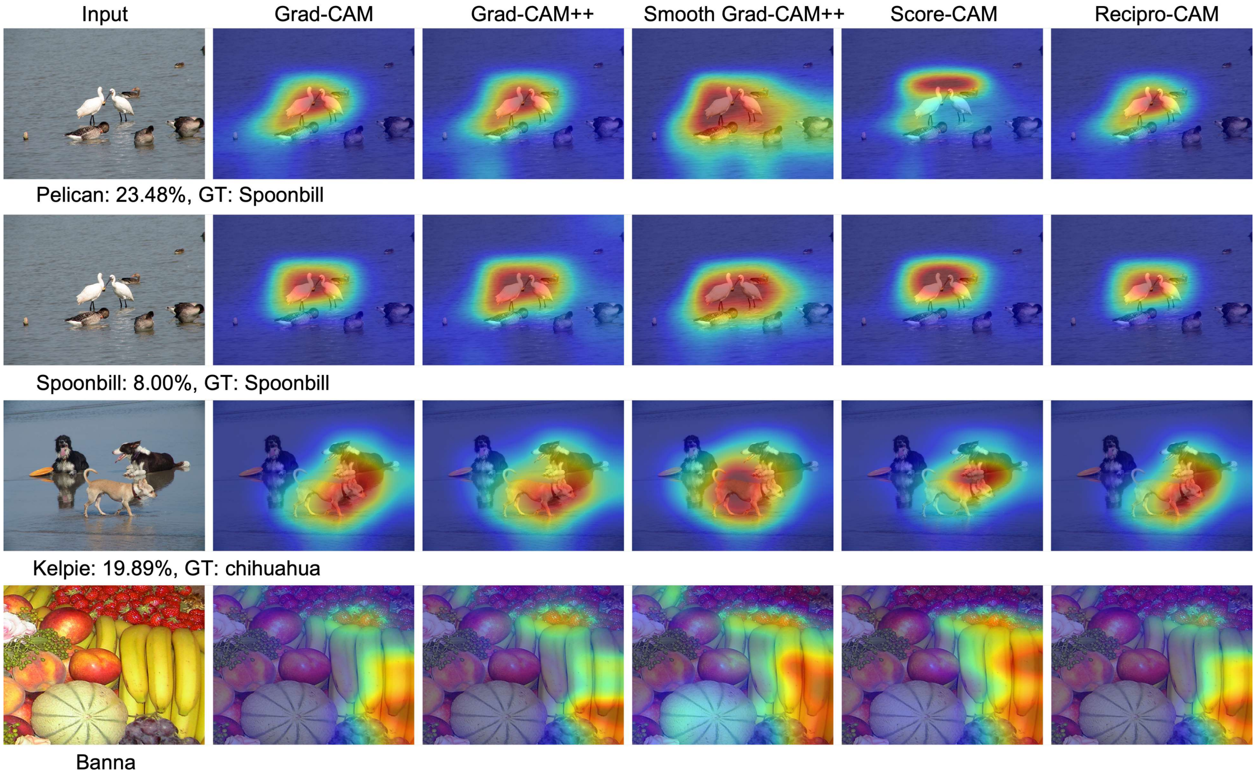
Input
Grad-CAM
Grad-CAM++
Smooth Grad-CAM++
Score-CAM
Recipro-CAM
Pelican: 23.48%, GT: Spoonbill
Spoonbill: 8.00%, GT: Spoonbill
Kelpie: 19.89%, GT: chihuahua
Banna

Figure 5. Multiple objects' saliency map results. Spoonbill, Kelpie, and Banana inputs process with GradCAM, GradCAM++, Smooth GradCAM++, ScoreCAM, and ReciproCAM to generate saliency maps. In this analysis, Smooth GradCAM++ and ScoreCAM showed unusual results.

region. An intriguing observation is the maximum saliency of ScoreCAM being on the right fish, contrasting with other CAMs which peak on the left fish.

In the case of brambling, each CAM exhibits distinct saliency map patterns. ReciproCAM and Smooth GradCAM++ indicate relatively lower saliency scores for the left birds, while GradCAM and GradCAM++ present similar saliency maps, with GradCAM++ showing higher scores for the left birds compared to GradCAM. However, ScoreCAM stands out by assigning high saliency scores to all birds, making it the preferred choice in this scenario.

Similarly, in the last yawl case, ReciproCAM, GradCAM++, and GradCAM produce comparable saliency maps with the highest scores assigned to the fourth yawl from the left. In contrast, Smooth GradCAM++ and ScoreCAM attribute the highest saliency values to the third yawl from the left. Here, Smooth GradCAM++ emerges as the optimal choice as it provides a saliency map that comprehensively covers all yawls.

Explanations for multiple objects with different identities. In the case of multiple objects within the same image, we aim to assess the resolution capability of the methods concerning different objects, as the saliency map should ideally exhibit class-dependent results, with different-class objects being deactivated. The results are presented in Figure 5.

The first row of Figure 5 illustrates an image of a spoonbill, which also contains mallard objects. Despite ResNet-50 predicting it as a pelican with a 23.48% probability, ReciproCAM, GradCAM++, and GradCAM suggest an overfocus on the beak part, potentially contributing to the misclassification. However, Smooth GradCAM++ exhibits the highest saliency score on the left spoonbill body part, also covering the mallards. In this case, it fails to separate different objects from the target class objects. ScoreCAM, on the other hand, highlights the lake surface, providing misleading information.

To delve deeper, we generated a spoonbill saliency map, depicted in the second row. Here, all methods exhibit similar broad coverage on body parts, with Smooth GradCAM++ extending to encompass mallards.

Moving to the third row, we encounter another mispredicted case, where ResNet-50 identifies the image as a Kelpie with a 19.89% probability, while its ground truth is a chihuahua. With all methods except Smooth GardCAM++ indicating saliency between the second and third dogs, it suggests confusion by ResNet-50 due to mixed features. Smooth GradCAM++, however, offers a broad saliency map covering all dogs, with the hottest area aligned with the ground truth, indicating its limitations as debug information.

Finally, the fourth row presents the saliency map for a banana image. Here, all CAM methods exhibit distinct patterns but effectively cover the banana area, except for the Smooth GradCAM++ method, which also encompasses pumpkins. These observations highlight the nuances and challenges associated with object resolution and class-dependent saliency mapping in complex image scenes.

5. Conclusion

We proposed a novel gradient-free saliency map generation method and demonstrated its superiority by quantitative and performance analysis. As a result, the proposed method achieved state-of-the-art results on the ADCC metric and in execution time.

References

[1] J. Bai, F. Lu, K. Zhang, et al. ONNX: Open neural network exchange, 2019. 1, 2

[2] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2018. 1, 2

[3] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. *Neural Information Processing Systems (NeurIPS)*, 2017. 2

[4] S. Desai and H. G. Ramaswamy. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. *IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2020. 1

[5] F.-G. Fernandez. Torchcam: Class activation explorer. 2020. 5

[6] Ruth Fong, Mandela Patrick, and Andrea Vedaldi. Understanding deep networks via extremal perturbations and smooth masks. In *IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 2950–2958, 2019. 1

[7] R. Fu, Q. Hu, X. Dong, Y. Guo, Y. Gao, and B. Li. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. *British Machine Vision Conference (BMVC)*, 2020. 1, 2

[8] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 4

[9] Intel. OpenVINO toolkit, 2019. 1, 2

[10] E. M. Kenny, C. Ford, M. Quinn, and M. T. Keane. Explaining black-box classifiers using post-hoc explanations-by-example: The effect of explanations and error-rates in xai user studies. *Artificial Intelligence*, 2021. 2

[11] R. Naidu, A. Ghosh, Y. Maurya, and S. S. Kundu. Is-cam: Integrated scorecam for axiomatic-based explanations. *arXiv preprint arXiv:2010.03023*, 2020. 2

[12] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224*, 2019. 1, 2

[13] V. Petsiuk, A. Das, and K. Saenko. Rise: Randomized input sampling for explanation of black-box models. *British Machine Vision Conference (BMVC)*, 2018. 1, 2

[14] V. Petsiuk, R. Jain, V. Manjunatha, V. I. Morariu, A. Mehra, V. Ordonez, and K. Saenko. Black-box explanation of object detectors via saliency maps. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020. 2

[15] S. Poppi, M. Cornia, L. Baraldi, and R. Cucchiara. Revisiting the evaluation of class activation mapping for explainability: A novel metric and experimental analysis. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2021. 4

[16] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, (3): 211–252, 2015. 4

[17] R. R. Selvaraju, A. Das, R. Vedantam, M. Cogswell, D. Parikh, and D. Batra. Grad-cam: Why did you say that? visual explanations from deep networks via gradient-based localization. *arXiv preprint arXiv:1610.02391*, 2016. 1, 2

[18] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *International Conference on Learning Representations (ICLR)*, 2015. 4

[19] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806v3*, 2015. 2

[20] H. Wang, R. Naidu, J. Michael, and S. S. Kundu. Ss-cam: Smoothed scorecam for sharper visual feature localization. *arXiv preprint arXiv:2006.14255*, 2020. 2

[21] 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. *IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2020. 1, 2

[22] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He. Aggregated residual transformations for deep neural networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 4

[23] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. *European Conference on Computer Vision (ECCV)*, 2014. 2

[24] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1, 2