ECAI 2023
K. Gal et al. (Eds.)

doi:10.3233/FAIA230582

Visualization Enhancement of Saliency Methods Based on the Sliding Window Mechanism

Xiaohong XIANG[a], Fuyuan Zhang[a;*], Xin Deng[a] and Xiaoyu Ding[a]

[a]Department of Computer Science and Technology, Chongqing University of Posts and Telecommunications

Abstract. Deep neural networks are widely used in image classification tasks, but their internal decision-making mechanisms are often difficult to explain. While various algorithms have been developed to visualize these mechanisms, many of them produce coarse, noisy results that are not always convincing. To address this issue, we propose a method for enhancing saliency maps produced by saliency methods. Our method uses a fixed-size sliding window to upsample local regions of the input image and feed them into the selected visualization algorithm to generate class-specific saliency maps and probability scores. We then downsample the resulting saliency maps and multiply them by the probability scores to obtain maps with greater detail. We evaluate our method using different saliency methods and network architectures, and demonstrate its effectiveness through both quantitative metrics and intuitive evaluation. Our results show that our method significantly improves the performance of these saliency methods, providing a more valid and reliable means of visualizing the decision mechanisms of deep neural networks. Code is available at https://github.com/LuoLogic/Enhuncement-saliency.

1 Introduction

Currently, deep neural networks shine with their superior performance in tasks such as image classification, object detection, and semantic segmentation. However, how to interpret the decisions made by neural networks has become a challenge in research. In image classification tasks, an intuitive means of interpretation is to find the regions in the input image that the neural network considers important and visualize them with a saliency map, which is called a visualization technique for deep learning.

CAM-based methods [4, 18, 27] are a popular class of methods with many excellent applications. These methods are usually class-sensitive and based on the feature maps output from the internal convolutional layers of CNNs, so they gain the trust of researchers. The quality of the saliency map depends on the position of the selected convolutional layer in the network. Usually, these methods choose a deep layer of the convolutional network, such as the one closest to the output layer, which contains rich class information and leads to a clear class differentiation of the visualized image. However, the feature map of the deep layer is of low resolution and cannot contain more detailed information about the class. The shallow layer has a higher resolution but is full of noise and lacks class information.

Decomposition-based method [3, 5, 11, 14] have a solid theoretical foundation and use DTD(Deep Taylor Decomposition) [14] as their basic theoretical framework. These methods reason backward from the network output to the input, decomposing the decisions made by the current layer of the network into the contributions of the previous layer of the network, until the corresponding elements in the input are reasoned. However, such methods require inference rules for different network types, and some of them lack class sensitivity, and the visualized images they produce suffer from lower resolution and more noise.

This paper addresses the problems of current saliency methods with high noise, low resolution, and lack of detailed feature information by proposing a generic approach that does not seek to modify current visualization methods from within. Thus it can theoretically be applied to any saliency methods with class sensitivity. Specifically, our contributions include:

- We propose an enhancement algorithm applicable to saliency methods with class sensitivity. It does not require internal modification of current saliency methods, has good generality, and can be applied to neural networks with completely different structures, such as CNN-based and Transformer-based networks.
- Experiments show that the method produces richer feature details and can more accurately target regions of interest to the network, while effectively removing noise from the heat maps generated by current saliency methods.
- Experimental results show that our method has a significant enhancement for a wide range of mainstream visualizations, and shows superior performance in perturbation experiments and image segmentation experiments.

2 Related Work

In this section, we introduce the current saliency map methods, which can be broadly classified into the following two categories.

2.1 Backpropagation based saliency methods

Zeiler et al. [25] proposed a visualization method based on deconvolution, which inverse maps the values in the feature map back to the pixel space of the input image, thereby indicating which pixels in the image are involved in the decision. Building on this work, the Guided BP [22] proposes to highlight important features of the visualized target by suppressing inputs and values with gradients less than 0 during backpropagation. Simonyan et al. [19] propose using the input image's gradients as a visual interpretation. This method assumes that some input pixels play a major role in the prediction results of the network. It directly computes gradients of specified probability score

* Corresponding Author. Email: s210231249@stu.cqupt.edu.cn.

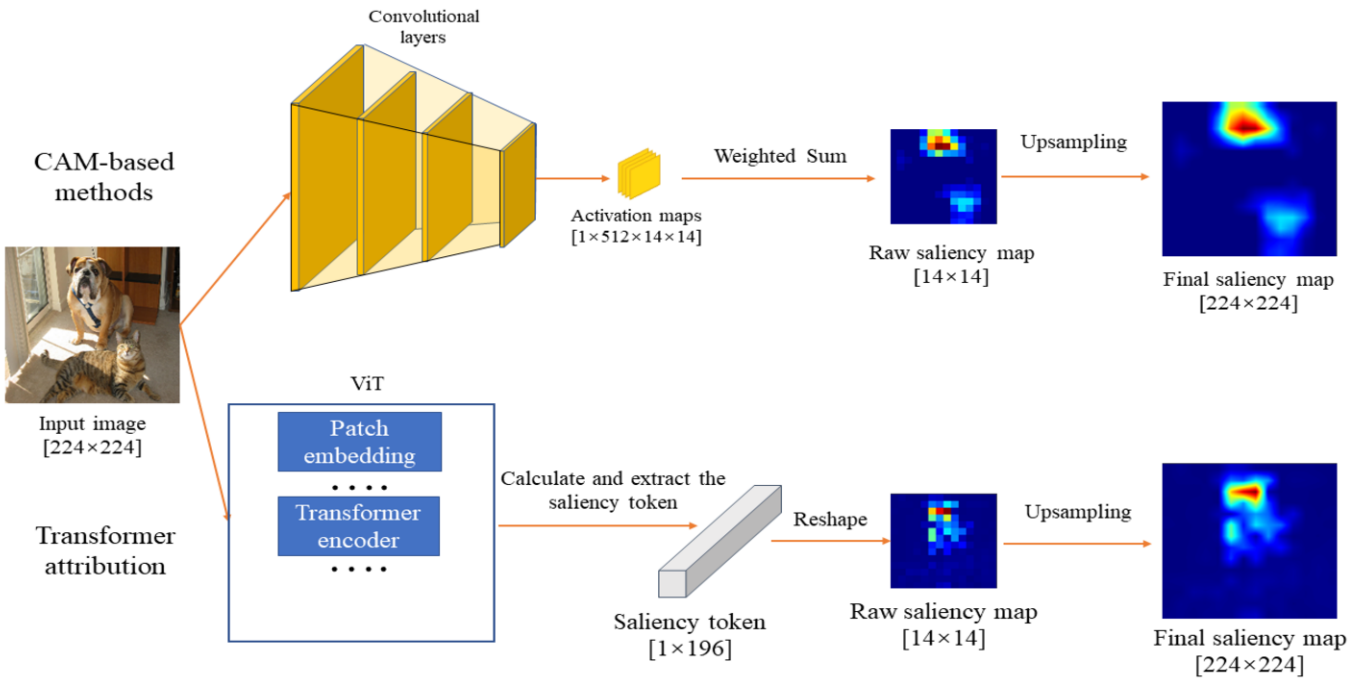

Figure 1. The reason for saliency methods with low resolution and more noise. We show a brief pipeline of two types of saliency methods. In both methods we perform saliency analysis for the dog. The CAM-based methods are limited by the low resolution of the activation maps of the convolutional layer. The saliency method for transformer is limited by the length of the token, which is due to patch embeding.

to the input, but the input gradients contain obvious noise and the visualization is fuzzy. SmoothGrad [21] and VarGrad [1] propose an effective method, which adds noise to the input image several times to generate a group of images containing noise. By averaging the results, the final saliency map is smoother. Although these studies have a solid theoretical basis, their visualization results are not easy to understand and noisy for humans. In addition, many of these methods are class-agnostic, i.e., they do not visually interpret the results for a given class. Some studies [2] point out that the reliability of these methods is questionable, they are not sensitive to network parameters, even without a network trained to get similar results.

There exist several methods that exploit the activation information within a model to produce salient maps. For instance, class activation mapping based approaches, including the first proposed Class Activation Mapping (CAM) [27] and its variants such as Grad-CAM [18], Grad-CAM++ [4], XGrad-CAM [8], etc., aim to generate category-distinct saliency maps by weighting different channels of the feature maps with category-specific gradient information obtained through back-propagation. On the other hand, the Relevance-CAM [13] utilizes the relevance scores obtained from Layer-wise Relevance Propagation (LRP) to weight the feature maps, demonstrating that shallow convolutional layers still retain class-relevant information. The CAMERAS [12] framework, which fuses the feature maps and gradients through multi-stage scaling of the input image, generates a higher resolution saliency map. However, this method is only effective in Convolutional Neural Network (CNN) models with residual structures. In general, the CAM-based methods leverage the feature maps, a crucial data component within a model, resulting in good performance and widespread research attention.

2.2 *Perturbation-based saliency methods*

Perturbation-based saliency methods generate saliency maps without relying on internal model data and instead focus solely on the inputs and outputs of the model. Early approaches, such as the use of a black square [25] to scramble the input image to identify the regions of interest for the model, have been replaced by more sophisticated techniques. For instance, Local Interpretable Model-agnostic Explanations (LIME) [16] employs a proxy model to fit the local decision-making behavior of the target model and thereby assess the sensitivity of various features. RISE [15] utilizes a large number of masks to mask the input image and calculates the probability of the target category as the weight for each mask, which is then combined linearly to produce the saliency map. Another approach, Mask [7] uses gradient descent to identify the mask with the lowest confidence in the target category. Score-CAM [24] and Group-CAM [26] perturb the input image using the feature map as the mask to compute the weight of the feature map and combine it linearly to generate the saliency map. While these methods can provide a direct representation of significant regions in the input image, they often have high computational costs and require optimization.

2.3 *Decomposition based saliency methods*

Decomposition-based algorithms use Deep Taylor Decomposition as the basic theoretical support, and they treat the output of a neural network as a decomposition of the function on the input variables. Here, we focus our attention on Layer-wise Relevance Propagation(LRP) [3] and its related derivative algorithms. Layer-wise Relevance Propagation algorithm is a special case of DTD(Deep Taylor Decomposition), which starts from the prediction class and prop-

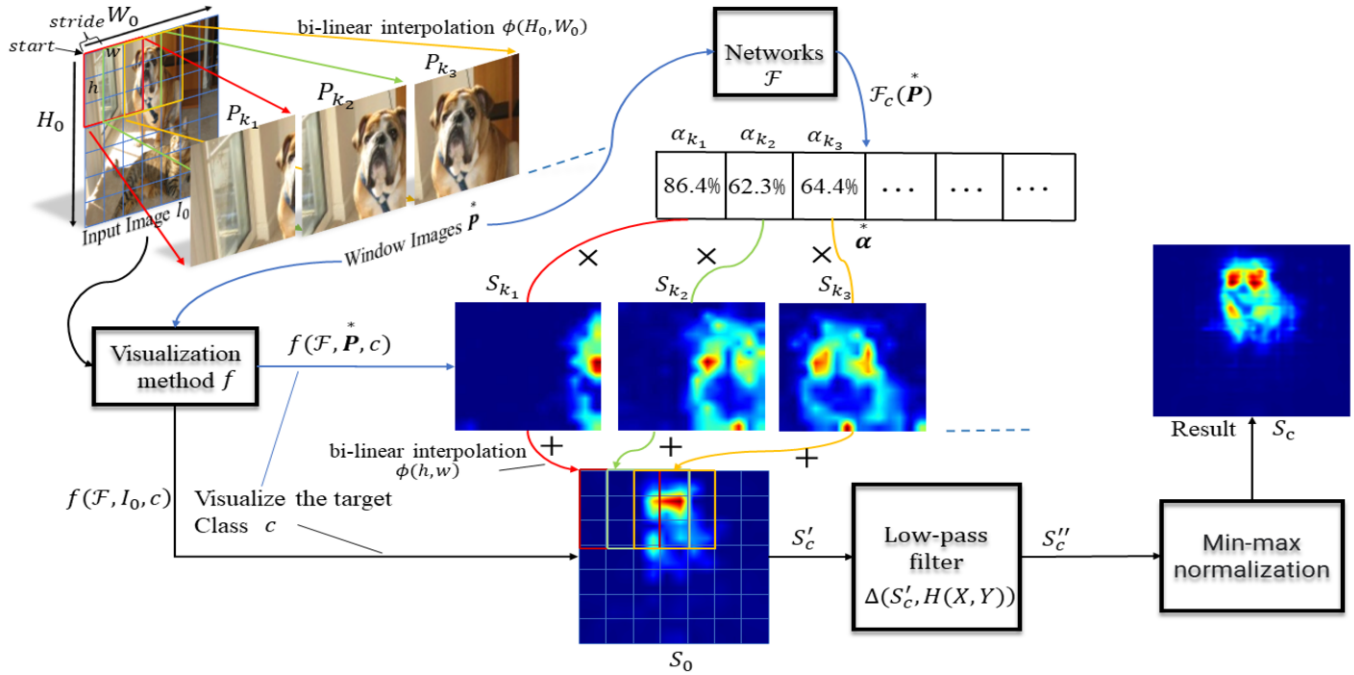

Figure 2. Pipeline of our enhancement method. f can be any of the class-sensitive saliency methods, and there is also no limit to the architecture of the networks.

agates the relevance backward to the input image, and the propagation process follows the rule of relevance conservation, i.e., the sum of the relevance of each layer of neurons in the network is equal. Subsequent derivative algorithms of LRP have followed this basic rule. Examples include Contrast-LRP [9] and Softmax-Gradient-LRP (SGLRP) [11], although both algorithms use information from all other classes to highlight the visualization of a particular class. Hila Chefer et al. [5] develop a new LRP rule for the structure of the Transformer to solve the challenges encountered during propagation in attention layers and skip connections. They solve the problem of visualizing the Transformer in the field of image classification by combining the gradients of each transformer block and its corresponding relevance score. We will cite in our experiments the work of Hila Chefer et al. [5] to demonstrate the generality of our method on transformer-based architecture ViT [6] as well.

3 Proposed Approach

To solve the problems of low resolution, blurred features, and noise in current saliency methods, our method applies a sliding window on the input image and allows the saliency method to enhance the perception of local information in the input image by collecting and fusing the visualization information from different window images. The pipeline of our method is illustrated in Figure 2.

3.1 Motivation

CAM-based saliency methods and transformer attribution method are limited by low resolution and lack the ability to locate detailed feature information. In Figure 1, we show a brief pipeline of most current CAM-based methods such as Grad-CAM [18], Grad-CAM++ [4], XGrad-CAM [8], etc., whose raw saliency map resolution is limited by the resolution of the feature maps extracted from the convolutional layers. Similarly, since transformer structure-based vision models use patch embeding, the original saliency map resolution of proposed saliency methods for such models such as Transformer attribution [5] is also limited by the token length. The raw saliency map contains very limited information and cannot give more detailed feature information. The original saliency map is upsampled to get the final saliency map, which easily generates noise and highlight the areas that are not related to the target class. [12]

3.2 Initialize Window Images

Let $I_0 \in \mathbb{R}^{3\times H_0\times W_0}$ be an original input image. We use a sliding window function $\psi(I, start, h, w, stride)$ to extract I_0 in blocks, where I is the input image; $start$ is the starting point of the sliding window in the coordinate system, whose origin is the upper left corner of the input image in general; h and w denote the window size, which should be smaller than the size of the input image; $stride$ indicates the number of pixels moved by the sliding window each time, and the direction of movement includes right and down.

$$p_{k_n} = \psi(I_0, start, h, w, stride) \tag{1}$$

where p_{k_n} denotes a window image and k_n denotes the coordinates of that image in I_0,

$$\overset{*}{\boldsymbol{p}} = \{p_{k_1}, p_{k_2}, \cdots, p_{k_n}\} \tag{2}$$

where $\overset{*}{\boldsymbol{p}}$ represents the set of all original window images. Then we upsample $\overset{*}{\boldsymbol{p}}$ to the size of I_0 using the bi-linear interpolation function ϕ:

$$\overset{*}{\boldsymbol{P}} = \phi(\overset{*}{\boldsymbol{p}}, H_0, W_0) \tag{3}$$

where $\overset{*}{\boldsymbol{P}} = \{P_{k_1}, P_{k_2}, \cdots, P_{k_n}\}$.

3.3 Get visualizations and weights

Let $\mathcal{F}$ be a pre-trained deep neural network, and $\mathcal{F}_c(I)$ denotes the probability score about class c obtained by inputting image I into $\mathcal{F}$. For the current visualization algorithm, we consider it as a function f of the neural network $\mathcal{F}$, the input image I and the class c. When the input image is I_0, we can get the result of this visualization algorithm for the interpretation of the neural network $\mathcal{F}$:

$$S_0 = f(\mathcal{F}, I_0, c) \tag{4}$$

where $S_0 \in \mathbb{R}^{1\times H_0\times W_0}$. The pixels in S_0 correspond to the pixels in I_0. The larger value in S_0 corresponds to the larger contribution of the pixel points in I_0 to the class c. Then we can get the saliency maps of the window images $\overset{*}{\boldsymbol{P}}$.

$$\begin{aligned} \overset{*}{\boldsymbol{S}} &= f(\mathcal{F}, \overset{*}{\boldsymbol{P}}, c) \\ &= \{S_{k_1}, S_{k_2}, \cdots, S_{k_n}\} \end{aligned} \tag{5}$$

In all CAM-based visualization algorithms, various forms of weighting the feature maps are employed, and this step is essential to measure the contribution of each feature map. To measure the importance of saliency maps generated by different window images, we obtain the probability score of each window image with respect to class c.

$$\begin{aligned} \overset{*}{\boldsymbol{\alpha}} &= \mathcal{F}_c(\overset{*}{\boldsymbol{P}}) \\ &= \{\alpha_{k_1}, \alpha_{k_2}, \cdots, \alpha_{k_n}\} \end{aligned} \tag{6}$$

where α_{k_n} is the probability score obtained from the input of P_{k_n} into $\mathcal{F}$ with respect to the class c.

3.4 Fuse saliency maps

In this step, we fuse these saliency maps $\overset{*}{\boldsymbol{S}}$ with rich details into the saliency map S_0 generated by the input image I_0. First we downsample $\overset{*}{\boldsymbol{S}}$ to the window size:

$$\begin{aligned} \overset{*}{\boldsymbol{s}} &= \phi(\overset{*}{\boldsymbol{S}}, h, w) \\ &= \{s_{k_1}, s_{k_2}, \cdots, s_{k_n}\} \end{aligned} \tag{7}$$

Second, we multiply $\overset{*}{\boldsymbol{s}}$ by their weight and add them to the pixel value on the corresponding coordinates of S_0:

$$\begin{aligned} S'_c &= \sum(\overset{*}{\boldsymbol{s}}\overset{*}{\boldsymbol{\alpha}} + S_0) \\ &= \sum_{k_1}^{k_n}(\alpha_{k_n} s_{k_n} + S_0^{k_n}) \end{aligned} \tag{8}$$

where k_n in $S_0^{k_n}$ denotes the coordinate in S_0, which corresponds to the coordinate in I_0 of the window images.

3.5 Smoothing and Normalizing

Because of the sliding window application, a grid exists on the saliency map, which visually appears not smooth enough. Therefore, we use an ideal low-pass filter $\Delta(S'_c, H(X,Y))$ to optimize it.

$$S''_c = \Delta(S'_c, H(X,Y)) \tag{9}$$

$H(X,Y)$ is the transfer function of the ideal low-pass filter, its definition is:

$$H(X,Y) = \begin{cases} 1 & D(X,Y) \le D_0 \\ 0 & D(X,Y) > D_0 \end{cases} \tag{10}$$

where D_0 is the distance from the cutoff frequency to the center of the spectrum. $D(X,Y) = \sqrt{X^2+Y^2}$ is the distance from the point (X,Y) in the spectrogram of S'_c to the center of the spectrum. The spectrum center is a zero-frequency component.

Finally, after Min-Max normalization we can get a saliency map with more details:

$$S_c = \frac{S''_c - min(S''_c)}{max(S''_c) - min(S''_c)} \tag{11}$$

4 Experiments

In this section, we illustrate the effective enhancement of our method on current saliency methods through qualitative comparison and quantitative evaluation. Among the quantitative evaluation experiments, we follow the work of Hila Chefer et al. [5] for perturbation tests and segmentation tests.

4.1 Setup

4.1.1 Datasets and Models

In this paper, we use two models, one is a CNN-based pre-trained VGG19 [20] and the other is a pre-trained ViT-base model [6] using Transformer blocks, whose input is a fixed 224×224 size image that is decomposed into a sequence of 16×16 size patches after passing through the patch embedding layer. For the perturbation experiments, we use the ImageNet(ILSVRC) 2012 validation set [17], which contains 50,000 images with a total of 1,000 classes. For the segmentation experiments, we use the annotated data from [10], which has a total of 4276 images and contains 445 classes.

4.1.2 Baselines

We select five saliency methods as baselines, among which for ViT-base visualization we chose Grad CAM [18], LRP [3], partial LRP [23], Transformer attribution [5]. The implementation of all these algorithms in ViT follows the work of Hila Chefer et al. [5]. Among them, the Transformer attribution method created by Hila Chefer et al. [5] is the best for the current Transformer-based model visualization. For VGG19 [20], we chose the popular Grad CAM [18] and Grad CAM++ [4], which are both backpropagation and feature map based algorithms that are fast and simple. We choose the last convolutional layer of VGG19 as the source of feature maps because it contains rich semantic information.

4.1.3 Our Method's parameters

In our experiments, all images are resized to 3×224×224. The sliding window parameters are: $start = (0,0), h = 96, w = 96, stride = 32$, the start is the upper left corner of the input image, i.e. the origin of the coordinate system. In the low-pass filter, we set $D_0 = 35$ uniformly.

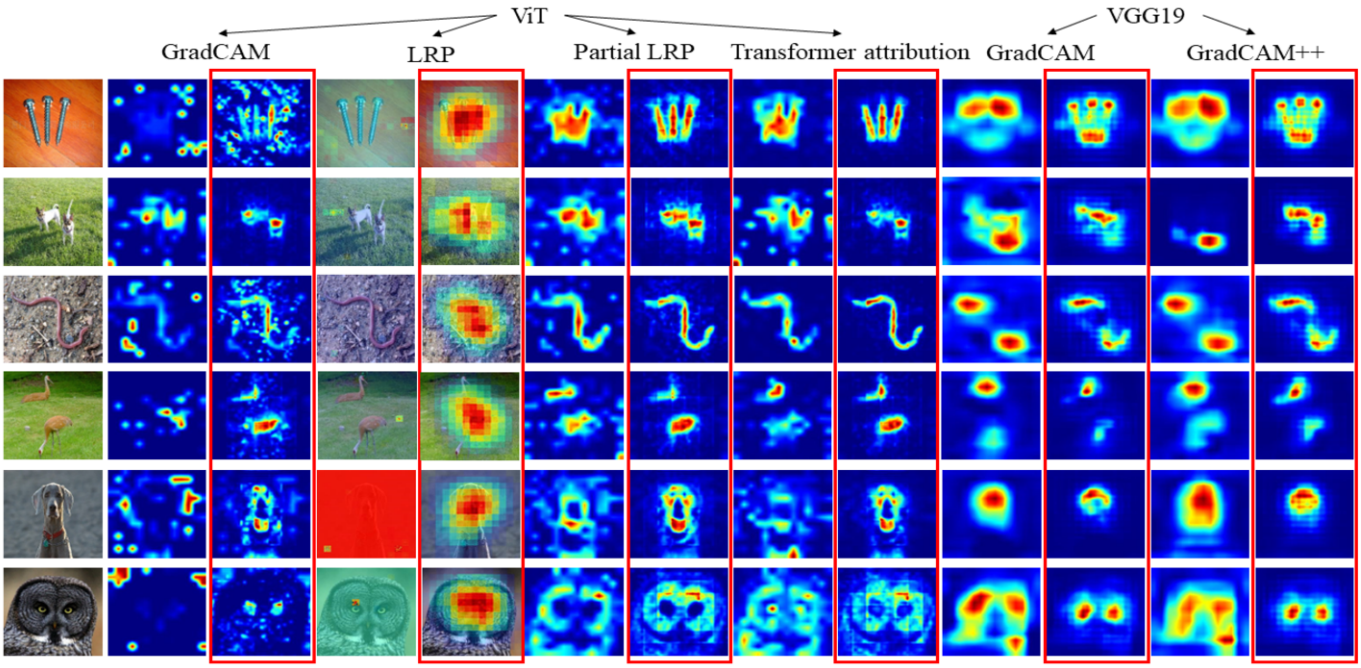

Figure 3. Samples based on different architectures and different algorithms are compared. For one algorithm, the results of our enhancement method are boxed in red.

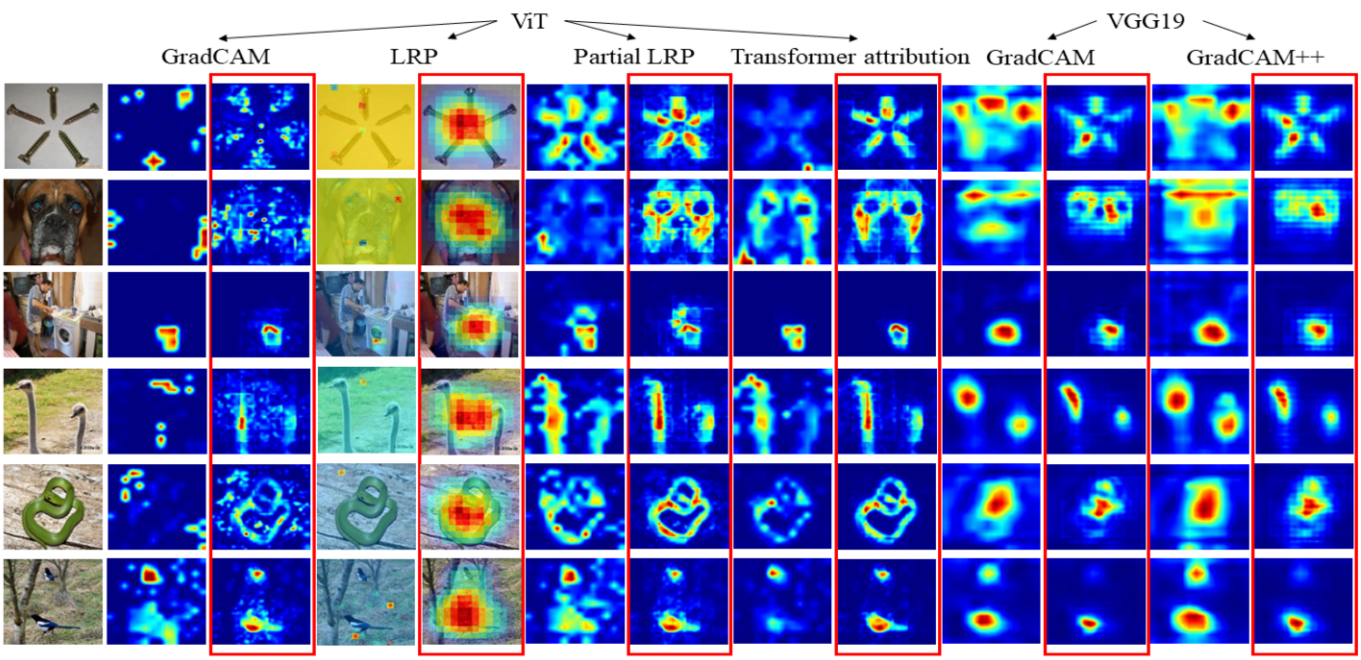

Figure 4. Samples based on different architectures and different algorithms are compared. For one algorithm, the results of our enhancement method are boxed in red.

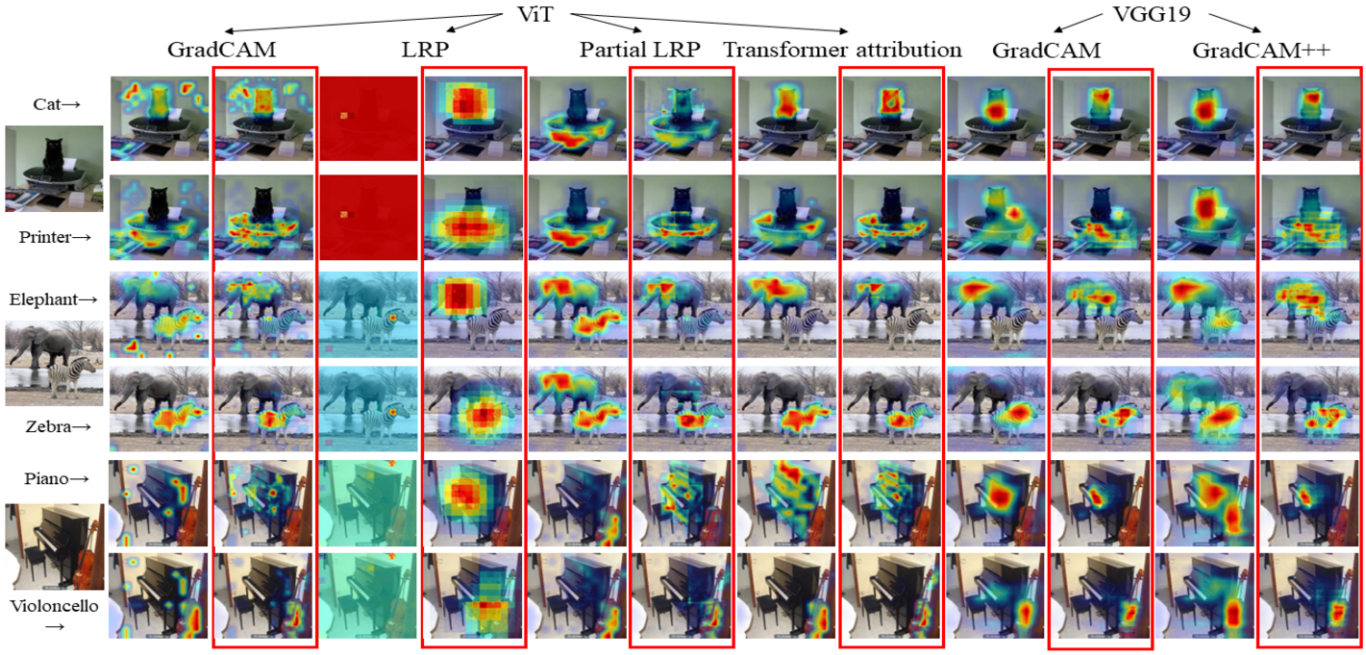

Figure 5. Saliency map comparison of different objects in the image, the results of our enhancement method are boxed in red.

4.2 *Quantitative evaluation*

We test the visual effect of our enhancement method on ViT and VGG19 on existing saliency methods. We do not overlay the saliency maps onto the original input images(except LRP) because this can better show the quality of the saliency maps under different algorithms, especially to facilitate the observation of the noise distribution. In Figure 3 and Figure 4, we randomly select six images and visualize the maximum prediction class with different saliency methods to generate saliency maps separately, where the saliency maps in the red box are obtained by applying our enhancement method to its left side.

A review of Figure 3 and Figure 4 shows that our enhancement method reduces noise and irrelevant highlighting in saliency maps. Our sliding window mechanism extracts subtle class-related evidence from the input image, allowing for identification of multiple occurrences of the same class (lines 1, 2, and 4 in Figure 3, lines 1, 4, and 6 in Figure 4). Our method also effectively highlights detailed features of the target class and key decision-making basis for the network (lines 5 and 6 in Figure 3, lines 2, 3, 5 in Figure 4).

4.3 *Qualitative Evalution*

4.3.1 *Perturbation tests*

A good saliency method accurately identifies key features contributing to a neural network's decision and assigns each pixel in the saliency map a value reflecting its contribution. We evaluate method performance through positive and negative perturbation tests. In the positive perturbation test, we first input a batch of images into the saliency method to generate the saliency map. Next, we replicate the original batch of images to create 10 additional batches, and according to the pixel importance ranking provided by the initial saliency map, we apply different masking ratios to each batch. For instance, the second batch has the top 10% of important pixels masked, the third batch has the top 20% of important pixels masked, and so on, with the first batch remaining unchanged. Finally, we feed these 10 batches of images into the network and calculate the average top-1 accuracy for each batch. The negative perturbation test follows the same steps but starts with unimportant pixels. Our evaluation metric is the area under the curve (AUC) of the average top-1 accuracy in 10% increments from 0%-90%.

In the step of getting saliency maps, we visualize and test the predicted class of the network and its real target class separately. If the saliency method is class-sensitive, its test results on the target class will be better than the predicted class, otherwise, both results are the same.

We propose the $Over-all$ score to comprehensively evaluate the perturbation test results, which is calculated as $AUC(Over-all) = AUC(Negative) - AUC(Positive)$. As can be seen in Table 1, our enhancement method achieves significant improvements in all the $Over-all$ scores. The most significant improvements are observed for the GradCAM and LRP algorithms applied to ViT, which previously lacked the ability to accurately visualize and localize important features. After enhancement using our method, these methods are able to identify important features. Examples of this can be seen in Figure 3, which demonstrates that our method can assist saliency methods in identifying important features of interest to the neural network, regardless of the underlying network structure.

4.3.2 *Segmentation tests*

In Segmentation tests, we calculate the mean value of pixels in each saliency map and set the pixels above the mean value in the saliency map to 1 and the rest to 0. The thresholded saliency map is com-

Table 1. AUC results(percent) for negative and positive perturbations on the ImageNet validation set concerning the predicted and target classes. For 'Negative' and 'Over-all' higher is better, and for 'positive' lower is better, better records are marked as **bold**. For the same method, the line with ✓is the performance after applying our enhancement method.

Model	Method	E	Predicted			Target		
			Negative	Positive	Over-all	Negative	Positive	Over-all
ViT	GradCAM		41.52	34.06	7.46	42.02	33.56	8.46
		✓	**47.54**	**26.99**	**20.55**	**48.46**	**26.49**	**21.97**
	LRP		43.49	41.94	1.55	43.49	41.94	1.56
		✓	**62.69**	**27.51**	**35.18**	**64.76**	**26.45**	**38.31**
	partial LRP		50.49	19.64	30.85	50.49	19.64	30.85
		✓	**55.57**	**18.45**	**37.12**	**56.13**	**18.14**	**37.99**
	Transformer attribution		54.14	17.03	37.11	55.04	**16.04**	39.00
		✓	**57.57**	**16.93**	**40.64**	**58.83**	16.25	**42.58**
VGG19	GradCAM		38.08	12.15	25.93	39.04	11.71	27.33
		✓	**39.07**	**10.20**	**28.87**	**40.09**	**9.78**	**30.31**
	GradCAM++		40.50	11.79	28.71	40.81	11.61	29.20
		✓	**40.95**	**10.09**	**30.86**	**41.49**	**9.81**	**31.68**

Table 2. Segmentation performance on the ImageNet-segmentation, higher is better(percent). For the same method, the line with ✓is the performance after applying our enhancement method.

Model	Method	E	Pixel Acc	mAP	mIoU
ViT	GradCAM		64.44	71.60	40.82
		✓	**70.33**	**77.02**	**47.81**
	LRP		51.09	55.68	32.89
		✓	**69.34**	**80.88**	**50.37**
	partial LRP		76.31	84.67	57.94
		✓	**80.42**	**85.83**	**62.85**
	Transformer attribution		79.74	86.03	62.01
		✓	**81.90**	**86.56**	**64.56**
VGG19	GradCAM		69.03	76.76	48.99
		✓	**73.78**	**79.99**	**53.82**
	GradCAM++		76.77	**85.48**	58.89
		✓	**78.60**	85.42	**60.58**

pared with the ground truth segmentation in the dataset. We use three metrics commonly used in semantic segmentation to measure performance: pixel-accuracy, mean-intersection-over-union (mIoU), and mean-Average-Precision(mAP), where mAP is calculated using the saliency map without thresholding.

The segmentation results are listed in Table 2. From this, we can see that our enhancement method has a significant improvement on all three key metrics of the listed saliency methods, and even Transformer attribution [5] can get a non-negligible improvement, which is currently the best performer on ViT. This indicates that our method can be useful for enhancing the performance of current saliency methods in the segmentation domain.

5 Conclusion

This paper proposes an enhancement method for current saliency methods, which is independent of the structure of networks, as long as the saliency method can give differentiated results for different classes. Our method has a clear optimization effect, and the saliency map can achieve a more accurate localization of the target features while also finding detailed evidence for the neural network to make decisions, as demonstrated in our quantitative experiments.

Currently, our method keeps the sliding window size and stride always constant and is not smart enough. In the future, we will improve it to achieve lower computational costs and more reliable visualization results.

Acknowledgment

This work was supported in part by the Natural Science Foundation of Chongqing under Grant cstc2020jcyj-msxmX0284; in part by the Scientific and Technological Research Program of Chongqing Municipal Education Commission under Grant KJQN202000625.

References

[1] Julius Adebayo, Justin Gilmer, Ian Goodfellow, and Been Kim, 'Local explanation methods for deep neural networks lack sensitivity to parameter values', *ArXiv preprint*, **abs/1810.03307**, (2018).

[2] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian J. Goodfellow, Moritz Hardt, and Been Kim, 'Sanity checks for saliency maps', in *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, eds., Samy Bengio, Hanna M. Wallach, Hugo Larochelle, Kristen Grauman, Nicolò Cesa-Bianchi, and Roman Garnett, pp. 9525–9536, (2018).

[3] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek, 'On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation', *PloS one*, **10**(7), e0130140, (2015).

[4] Aditya Chattopadhay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian, 'Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks', in *2018 IEEE winter conference on applications of computer vision (WACV)*, pp. 839–847. IEEE, (2018).

[5] Hila Chefer, Shir Gur, and Lior Wolf, 'Transformer interpretability beyond attention visualization', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 782–791, (2021).

[6] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby, 'An image is worth 16x16 words: Transformers for image recognition at scale', in *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, (2021).

[7] Ruth C. Fong and Andrea Vedaldi, 'Interpretable explanations of black boxes by meaningful perturbation', in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 3449–3457. IEEE Computer Society, (2017).

[8] Ruigang Fu, Qingyong Hu, Xiaohu Dong, Yulan Guo, Yinghui Gao, and Biao Li, 'Axiom-based grad-cam: Towards accurate visualization and explanation of cnns', in *31st British Machine Vision Conference 2020, BMVC 2020, Virtual Event, UK, September 7-10, 2020*. BMVA Press, (2020).

[9] Jindong Gu, Yinchong Yang, and Volker Tresp, 'Understanding individual decisions of cnns via contrastive backpropagation', in *Asian Conference on Computer Vision*, pp. 119–134. Springer, (2018).

[10] Matthieu Guillaumin, Daniel Küttel, and Vittorio Ferrari, 'Imagenet auto-annotation with segmentation propagation', *International Journal of Computer Vision*, **110**(3), 328–348, (2014).

[11] Brian Kenji Iwana, Ryohei Kuroki, and Seiichi Uchida, 'Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. in 2019 ieee', in *CVF International Conference on Computer Vision Workshop (ICCVW)*, pp. 4176–4185.

[12] Mohammad AAK Jalwana, Naveed Akhtar, Mohammed Bennamoun, and Ajmal Mian, 'Cameras: Enhanced resolution and sanity preserving class activation mapping for image saliency', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16327–16336, (2021).

[13] Jeong Ryong Lee, Sewon Kim, Inyong Park, Taejoon Eo, and Dosik Hwang, 'Relevance-cam: Your model already knows where to look', in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14944–14953, (2021).

[14] Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller, 'Explaining nonlinear classification decisions with deep taylor decomposition', *Pattern recognition*, **65**, 211–222, (2017).

[15] Vitali Petsiuk, Abir Das, and Kate Saenko, 'RISE: randomized input sampling for explanation of black-box models', in *British Machine Vision Conference 2018, BMVC 2018, Newcastle, UK, September 3-6, 2018*, p. 151. BMVA Press, (2018).

[16] Marco Túlio Ribeiro, Sameer Singh, and Carlos Guestrin, '"why should I trust you?": Explaining the predictions of any classifier', in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, eds., Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, pp. 1135–1144. ACM, (2016).

[17] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al., 'Imagenet large scale visual recognition challenge', *International journal of computer vision*, **115**(3), 211–252, (2015).

[18] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra, 'Grad-cam: Visual explanations from deep networks via gradient-based localization', in *IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017*, pp. 618–626. IEEE Computer Society, (2017).

[19] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, 'Deep inside convolutional networks: Visualising image classification models and saliency maps', in *In Workshop at International Conference on Learning Representations*. Citeseer, (2014).

[20] Karen Simonyan and Andrew Zisserman, 'Very deep convolutional networks for large-scale image recognition', in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, eds., Yoshua Bengio and Yann LeCun, (2015).

[21] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg, 'Smoothgrad: removing noise by adding noise', *ArXiv preprint*, **abs/1706.03825**, (2017).

[22] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller, 'Striving for simplicity: The all convolutional net', *arXiv preprint arXiv:1412.6806*, (2014).

[23] Elena Voita, David Talbot, Fedor Moiseev, Rico Sennrich, and Ivan Titov, 'Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned', in *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 5797–5808, (2019).

[24] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu, 'Score-cam: Score-weighted visual explanations for convolutional neural networks', in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 24–25, (2020).

[25] Matthew D Zeiler and Rob Fergus, 'Visualizing and understanding convolutional networks', in *European conference on computer vision*, pp. 818–833. Springer, (2014).

[26] Qinglong Zhang, Lu Rao, and Yubin Yang, 'Group-cam: Group score-weighted visual explanations for deep convolutional networks', *arXiv preprint arXiv:2103.13859*, (2021).

[27] Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba, 'Learning deep features for discriminative localization', in *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pp. 2921–2929. IEEE Computer Society, (2016).