Spatial Sensitive Grad-CAM++: Improved Visual Explanation for Object Detectors via Weighted Combination of Gradient Map

Toshinori Yamauchi
Hitachi, Ltd. Research & Development Group
1-1, Omika-cho 7-chome, Hitachi-shi, Ibaraki-ken, 319-1292, Japan
toshinori.yamauchi.ba@hitachi.com

Abstract

Visual explanations for object detectors are important tasks, and several methods have been proposed. One of the methods is Spatial Sensitive Grad-CAM, which can generate instance-specific heat maps. However, it may generate heat maps that highlight only part of the important regions because it computes the importance of the features from the average of the gradient map. To tackle this problem, we propose Spatial Sensitive Grad-CAM++, which computes the importance of features from the weighted combination of the gradient map, originally introduced in Grad-CAM++ and extended by us to the object detectors. With this improvement, it can generate heat maps that more accurately capture the important regions for the detected results. Through experiments, we confirm that it outperforms other methods, which indicates that it generates higher quality heat maps for the object detectors.

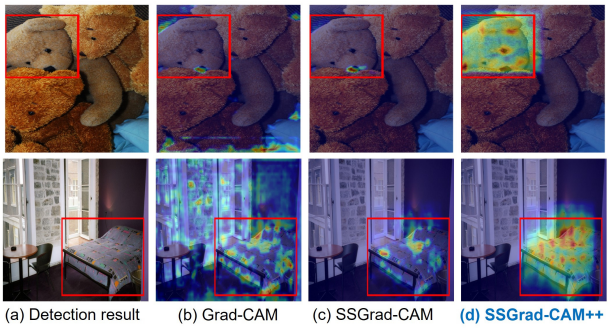

Figure 1. Generated heat maps by each method for the detection result (a). Grad-CAM generates class-specific heat maps (b), which are not appropriate for the object detectors. SSGrad-CAM generates instance-specific heat maps, but only part of the important regions for the detected results are highlighted, as shown in (c) (*e.g.*, highlighting only small regions near the nose of the "teddy bear"). SSGrad-CAM++ more accurately identifies the important regions, as shown in (d) (*e.g.*, strongly highlighting not only the nose but also other features such as eyes and ears).

1. Introduction

In recent years, deep learning has been successfully applied to a variety of tasks [3, 4, 9]. In general, deep learning models have numerous parameters; therefore, it is difficult to understand the reasons for the model's outputs. Visual explanation is the effective method to address it, which leads to a better understanding of the model [2, 10, 14].

Gradient-based, perturbation-based, and class activation map-based (CAM-based) methods are widely used as visual explanations [1, 2, 7, 10–14, 16, 19]. In terms of methods for object detectors, ODAM [18] as gradient-based, D-RISE [8] as perturbation-based, and Spatial Sensitive Grad-CAM (SSGrad-CAM) [15] as CAM-based have been proposed, respectively. Although these methods can generate instance-specific heat maps corresponding to each detected result, there is room for quality improvement. In this study, to generate higher quality heat maps, we propose Spatial Sensitive Grad-CAM++ (SSGrad-CAM++), the CAM-based method for the object detectors, which improves SSGrad-CAM [15].

SSGrad-CAM is based on Grad-CAM [10], which is a well-known CAM-based method and mainly focuses on classification models. Grad-CAM only computes the importance of the features and lacks sensitivity to space; therefore, it generates class-specific heat maps that are not appropriate for object detectors (Fig. 1(b)). To generate instance-specific heat maps, SSGrad-CAM modifies the heat maps generated from Grad-CAM with the space map, which represents the importance of space. In this manner, it can focus on the importance of both features and space and generate instance-specific heat maps [15].

However, SSGrad-CAM may generate heat maps that highlight only part of the important regions for the detected results, as shown in Fig. 1(c). One of the reasons is that it does not correctly reflect the importance of the fea-

tures because it computes the importance of the features by simply averaging the gradient map. For the classification models, Grad-CAM++ mitigates the aforementioned problem by computing the importance of the features using the weighted combination of the gradient map [2]. We believe that, similar to classification models, correctly reflecting the importance of the features is important and this method is effective for object detectors.

To this end, we propose SSGrad-CAM++, which improves SSGrad-CAM in terms of the computation of the importance of the features. SSGrad-CAM++ computes it from the weighted combination of the gradient map, an extension of the computation of Grad-CAM++ [2] for the object detectors, and then computes the importance of space in the same manner as SSGrad-CAM [15]. In this manner, it more accurately identifies the important regions for the detected results, as shown in Fig. 1(d).

Our experiments show that the proposed method is superior to other methods for object detectors, and it can generate higher quality heat maps. These heat maps lead to the proper analysis of the models.

2. Method

Fig. 2 shows an overview of SSGrad-CAM++. In this section, we first explain Grad-CAM and SSGrad-CAM. Then, we describe the improvement of SSGrad-CAM++ from SSGrad-CAM.

2.1. Grad-CAM & SSGradCAM

Grad-CAM first computes the gradient of the score Y^c for class c, with respect to the feature map $A^k \in \mathbb{R}^{H \times W}$, then executes global average pooling for its gradient.

$$w_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y^c}{\partial A_{i,j}^k}, \tag{1}$$

where Z is the number of pixels in the feature map A^k. The weight w_k^c represents the importance of the features, which is captured by the feature map A^k, for score Y^c. Grad-CAM then computes the heat map L^c for the class c as follows:

$$L^c = ReLU\left(\sum_k w_k^c A^k\right), \tag{2}$$

where $ReLU(\cdot)$ is rectified linear unit [6] that eliminates negative values. As represented in Eq. (2), Grad-CAM multiplies the scaler weight w_k^c to all elements of the feature map A^k; therefore, if we apply Grad-CAM to the object detectors, it generates heat maps that highlight the unrelated regions for the detected results [15].

For the aforementioned issue, SSGrad-CAM incorporates sensitivity to space and generates the instance-specific heat map $L^{c,det}$. SSGrad-CAM modifies the heat maps computed by Eq. (2) with the space map that has spatial information for the detected result.

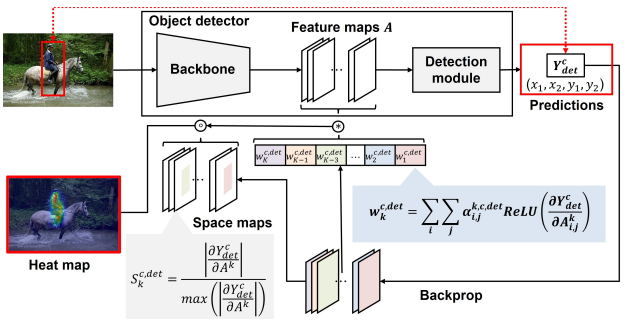

Figure 2. Overview of SSGrad-CAM++. It computes the weight $w_k^{c,det}$ representing the importance of the k-th channel feature map (importance of the features) and the space map $S_k^{c,det}$ representing the relative importance of each neuron in the k-th channel feature map (importance of space). For $w_k^{c,det}$, SSGrad-CAM++ computes it from the weighted combination of the gradient map. Computation of the space map $S_k^{c,det}$ is the same as SSGrad-CAM.

$$L^{c,det} = ReLU\left(\sum_k \left(w_k^{c,det} A^k\right) \circ S_k^{c,det}\right). \tag{3}$$

In the above equation, $w_k^{c,det}$ is the weight representing the importance of the feature map A^k, which is computed as follows:

$$w_k^{c,det} = \frac{1}{Z} \sum_i \sum_j \frac{\partial Y_{det}^c}{\partial A_{i,j}^k}, \tag{4}$$

where Y_{det}^c is the score for the class c corresponding to the predicted bounding box. In Eq. (3), $S_k^{c,det} \in \mathbb{R}^{H \times W}$ is the space map computed as follows:

$$S_k^{c,det} = \frac{\left|\frac{\partial Y_{det}^c}{\partial A^k}\right|}{max\left(\left|\frac{\partial Y_{det}^c}{\partial A^k}\right|\right)} \tag{5}$$

where $max(\cdot)$ is a function to calculate the maximum value. This space map $S_k^{c,det} \in [0, 1]$ indicates the relative numerical magnitude in the absolute value of the gradients in the feature map A^k, and it represents the spatial importance of each neuron for the predicted score Y_{det}^c [15].

As shown in Eq. (3), SSGrad-CAM considers the importance of both features and space, and can generate instance-specific heat maps.

2.2. SSGrad-CAM++

SSGrad-CAM computes the importance of the features by averaging the gradients of the score to feature maps as in

Grad-CAM (Eq. (4)). In this case, it may generate the heat map that highlight only part of important regions. One of the reasons is that the magnitude of the weights $w_k^{c,det}$ depends on the spatial size of the feature [2]. For classification models, Grad-CAM++ mitigates the above issue by computing the weight w_k^c from the weighted combination of the gradient map. SSGrd-CAM++ incorporates this computation by extending it to the object detectors.

Grad-CAM++ defines the weight w_k^c as the weighted combination of the gradient map for classification score [2]. SSGrad-CAM++ emulates Grad-CAM++, and we can defines the weight $w_k^{c,det}$ by simply substituting classification score with Y_{det}^c as follows:

$$w_k^{c,det} = \sum_i \sum_j \alpha_{i,j}^{k,c,det} ReLU\left(\frac{\partial Y_{det}^c}{\partial A_{i,j}^k}\right), \tag{6}$$

where $\alpha_{i,j}^{k,c,det}$ is the weighting coefficients, which controls the magnitude of the weight $w_k^{c,det}$.

Grad-CAM++ computes the weighting coefficients α based on the assumption that the classification score Y^c is written as $Y^c = \sum_k w_k^c \sum_i \sum_j A_{i,j}^k$ [2]. In object detectors, we assume Y_{det}^c as follows, taking into account the architecture of the network.

$$Y_{det}^c = \sum_k w_k^{c,det} \sum_i \sum_j A_{i,j}^k M_{i,j}^{k,det}, \tag{7}$$

where $M^{k,det} \in \mathbb{R}^{H \times W}$ is a binary mask with 1 for related regions to the detected instance and 0 for others. If we set the final layer of backbone to the visualization target, regions with 1 are the ROI pooling area or the corresponding default bounding box area because Y_{det}^c is computed from features from such regions.

Inserting Eq. (6) into Eq. (7), and taking partial derivative with respect to $A_{i,j}^k$ on both sides, we get

$$\begin{aligned}\frac{\partial Y_{det}^c}{\partial A_{ij}^k} =& \sum_a \sum_b \alpha_{a,b}^{k,c,det} \frac{\partial Y_{det}^c}{\partial A_{a,b}^k}\left(M_{i,j}^{k,det}\right) \\ &+ \sum_a \sum_b A_{a,b}^k M_{a,b}^{k,det}\left(\alpha_{i,j}^{k,c,det} \frac{\partial^2 Y_{det}^c}{\left(\partial A_{i,j}^k\right)^2}\right).\end{aligned} \tag{8}$$

Taking a further partial derivative with respect to $A_{i,j}^k$,

$$\begin{aligned}\frac{\partial^2 Y_{det}^c}{\left(\partial A_{i,j}^k\right)^2} =& 2\alpha_{i,j}^{k,c,det} \frac{\partial^2 Y_{det}^c}{\left(\partial A_{i,j}^k\right)^2} M_{i,j}^{k,det} \\ &+ \sum_a \sum_b A_{a,b}^k M_{a,b}^{k,det}\left(\alpha_{i,j}^{k,c,det} \frac{\partial^3 Y_{det}^c}{\left(\partial A_{i,j}^k\right)^3}\right).\end{aligned} \tag{9}$$

From Eq. (9), we get

$$\alpha_{i,j}^{k,c,det} = \frac{\frac{\partial^2 Y_{det}^c}{\left(\partial A_{i,j}^k\right)^2}}{2\frac{\partial^2 Y_{det}^c}{\left(\partial A_{i,j}^k\right)^2} M_{i,j}^{k,det} + \sum_a \sum_b A_{a,b}^k M_{a,b}^{k,det}\left(\frac{\partial^3 Y_{det}^c}{\left(\partial A_{i,j}^k\right)^3}\right)}. \tag{10}$$

As in SSGrad-CAM, SSGrad-CAM++ also computes the space map (Eq. (5)). From the above, SSGrad-CAM++ generates heat maps to follow Eq. (3) using $w_k^{c,det}$, which is computed by substituting Eq. (10) in Eq. (6).

3. Experiments

3.1. Implementation

In this study, we use MS-COCO dataset [5]. We compare the proposed method with Grad-CAM [10], Grad-CAM++ [2], D-RISE [8], SSGrad-CAM [15], and ODAM [18]. To assess whether the generated heat maps capture the important regions, we conduct an evaluation of Deletion and Insertion [2, 7, 8]. For evaluating localization of heat maps, we use Pointing game [2, 17] and its related metrics [14]. Following quantitative experiments, we apply 1,000 masks with a resolution of 16×16 to D-RISE [8], considering the computation time. We apply the same smoothing function to ODAM and the space maps in SSGrad-CAM and SSGrad-CAM++. We implement these methods in Faster R-CNN [9] with the backbones of ResNet-50 [3].

3.2. Deletion and Insertion

Deletion and Insertion evaluate changes in the model's output by deleting or inserting each pixel based on the generated heat map. At each step, a score is calculated to evaluate the similarity between the original output and the output at each step. In this study, we evaluate the following two scores: 1) s_1: the class probability for the predicted class corresponding to the detected result, and 2) s_2: a score that takes into account both the class probability and the bounding box corresponding to the detected result. Specifically, s_2 is the following score introduced in [8].

$$\begin{aligned} s_2(d_t, d_j) &= IoU(B_t, B_j) \cdot \frac{P_t \cdot P_j}{\|P_t\| \|P_j\|} \\ d_i = [B_i, P_i] &= [(x_1^i, y_1^i, x_2^i, y_2^i), (p_1^i, ..., p_C^i)] \end{aligned} \tag{11}$$

where $d_t = [B_t, P_t]$ is the original output, $d_i = [B_i, P_i]$ is the output for image at each step, $B_i = (x_1^i, y_1^i, x_2^i, y_2^i)$ is the bounding box corners, $P_i = (p_1^i, ..., p_C^i)$ is the vector of probability for each class, and $IoU(A, B)$ is a function to calculate the intersection over union (IoU) between the

Table 1. Results of Deletion (Del) and Insertion (Ins). Each result is computed for all detected result with the predicted class probability > 0.6.

	$score = s_1$		$score = s_2$	
Method	Del ↓	Ins ↑	Del ↓	Ins ↑
Grad-CAM [10]	0.201	0.650	0.241	0.655
Grad-CAM++ [2]	0.104	0.853	0.142	0.851
D-RISE [8]	0.154	0.781	0.201	0.766
ODAM [18]	0.113	0.745	0.180	0.731
SSGrad-CAM [15]	0.071	0.916	0.135	0.867
ours	**0.055**	**0.942**	**0.102**	**0.901**

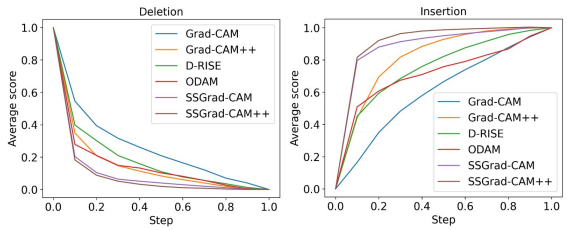

Figure 3. Visualizations of the score curves for Deletion and Insertion using score s_1 (class probability). The X-axis indicates the steps, and the Y-axis shows the average score.

Table 2. Results of Pointing game (P) and energy based Pointing game (eP). These are the results using bounding boxes (b) and instance segmentation masks (m) and evaluated for all detection results with the predicted class probability > 0.6 and $IoU > 0.7$ for the ground truth.

Method	P(b)	P(m)	eP(b)	eP(m)
Grad-CAM [10]	0.389	0.330	0.127	0.08
Grad-CAM++ [2]	0.649	0.563	0.150	0.09
D-RISE [8]	0.624	0.505	0.119	0.070
ODAM [18]	0.916	0.804	0.737	**0.546**
SSGrad-CAM [15]	0.911	0.769	0.726	0.509
ours	**0.981**	**0.880**	**0.743**	0.512

bounding box A and B. We show the results in Tab. 1. We report the area under the curve (AUC). For both scores s_1 and s_2, the proposed method achieves the best values for all metrics. Fig. 3 shows the score curves for Deletion and Insertion using score s_1 (class probability). The curve of SSGrad-CAM++ rapidly decreases (Deletion) or increases (Insertion) at a relatively early step compared to other methods. We confirm the same tendency in the result for the score s_2 (Eq. (11)) as well. These results indicate that SSGrad-CAM++ more accurately identifies important regions for the detected results.

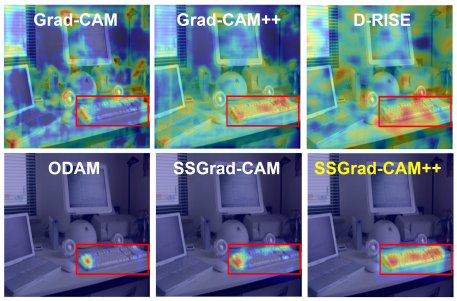

Figure 4. Generated heat maps for the detection of a "keyboard".

3.3. Pointing game

Pointing game [17] calculates the ratio of the maximum value in each heat map that lies within the ground truth (the bounding box or instance mask) for all of heat maps. In this study, we additionally adapt energy-based Pointing game [14] which evaluates the ratio of heat map energy within the ground truth. We show the results in Tab. 2. We compute for all detection results with the predicted class probability > 0.6 and $IoU > 0.7$ for the ground truth. The proposed method achieves the best values for almost all metrics, which indicates that the generated heat maps have high localization ability for the detected instances.

3.4. Heat map comparison

Fig. 4 shows the comparison of heat maps generated by each method. For D-RISE, we adapt 5,000 masks with a resolution of 16×16 to it. Heat maps generated by Grad-CAM and Grad-CAM++ also highlight the unrelated regions due to the lack of sensitivity to space. D-RISE generates noisy heat maps, and the performance may depend on the resolution of the masks. SSGrad-CAM and ODAM generate instance-specific heat maps, but only part of the important regions are highlighted, while SSGrad-CAM++ generates heat maps that more appropriately highlight the important regions.

4. Conclusion

We propose SSGrad-CAM++, CAM-based visual explanation for object detectors that improves SSGrad-CAM in terms of the computation of the importance of the features. In this paper, we applied it to Faster R-CNN, but it can be applied to various type of architectures. Our experiments show that SSGrad-CAM++ generates heat maps that more accurately capture the important regions than other methods. Such heat maps allow us to analyze the model appropriately. We plan to conduct qualitative evaluations to analyze the quality improvement of the heat maps and quantitative evaluations by applying it to other object detectors and using other datasets. These are for our future works.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems (NeuRIPS)*. Curran Associates, Inc., 2018. 1

[2] 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)*, pages 839–847, 2018. 1, 2, 3, 4

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 1, 3

[4] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017. 1

[5] Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft coco: Common objects in context, 2014. 3

[6] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning (ICML)*, pages 807–814, 2010. 2

[7] Vitali Petsiuk, Abir Das, and Kate Saenko. Rise: Randomized input sampling for explanation of black-box models. In *British Machine Vision Conference (BMVC)*, 2018. 1, 3

[8] Vitali Petsiuk, Rajiv Jain, Varun Manjunatha, Vlad I. Morariu, Ashutosh Mehra, Vicente Ordonez, and Kate Saenko. Black-box explanation of object detectors via saliency maps. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11443–11452, 2021. 1, 3, 4

[9] Shaoqing Ren, Kaiming He, Ross B. Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in Neural Information Processing Systems (NeuRIPS)*, pages 91–99, 2015. 1, 3

[10] 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 *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2017. 1, 3, 4

[11] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR*, abs/1312.6034, 2013.

[12] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda B. Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *CoRR*, abs/1706.03825, 2017.

[13] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. In *International Conference on Learning Representations (ICLR) workshop track*, 2015.

[14] 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 Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, 2020. 1, 3, 4

[15] Toshinori Yamauchi and Masayoshi Ishikawa. Spatial sensitive grad-cam: Visual explanations for object detection by incorporating spatial sensitivity. In *IEEE International Conference on Image Processing (ICIP)*, pages 256–260, 2022. 1, 2, 3, 4

[16] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *CoRR*, abs/1311.2901, 2013. 1

[17] Jianming Zhang, Zhe L. Lin, Jonathan Brandt, Xiaohui Shen, and Stan Sclaroff. Top-down neural attention by excitation backprop. *International Journal of Computer Vision*, 126: 1084–1102, 2018. 3, 4

[18] Chenyang Zhao and Antoni B. Chan. Odam: Gradient-based instance-specific visual explanations for object detection. In *IEEE International Conference on Learning Representations (ICLR)*, 2023. 1, 3, 4

[19] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 1