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

Haofan Wang, Zifan Wang, Piotr Mardziel
Carnegie Mellon University
{haofanw, zifanw}@andrew.cmu.edu, piotrm@gmail.com

Mengnan Du, Fan Yang, Xia Hu
Texas A&M University
{dumengnan, nacoyang, xiahu}@tamu.edu

Zijian Zhang, Sirui Ding
Wuhan University
zijianzhang0226@gmail.com, sirui_ding@whu.edu.cn

Abstract

Recently, increasing attention has been drawn to the internal mechanisms of convolutional neural networks, and the reason why the network makes specific decisions. In this paper, we develop a novel post-hoc visual explanation method called Score-CAM based on class activation mapping. Unlike previous class activation mapping based approaches, Score-CAM gets rid of the dependence on gradients by obtaining the weight of each activation map through its forward passing score on target class, the final result is obtained by a linear combination of weights and activation maps. We demonstrate that Score-CAM achieves better visual performance and fairness for interpreting the decision making process. Our approach outperforms previous methods on both recognition and localization tasks, it also passes the sanity check. We also indicate its application as debugging tools. Official code has been released[1].

1. Introduction

Explanation of Deep Neural Networks (DNNs) aims to increase the model's transparency to humans so the logic behind the inference can be interpreted in a human understandable way. Among the effort to provide explanations, visualizing a certain quantity of interest, e.g. importance of input features or learned weights, becomes the most straight-forward approach to gain trust from users. Among diverse components in DNNs, the spatial convolution is always the first choice in the feature extraction for both image and language processing. Towards better explanations of convolution operation and Convolutional Neural Network (CNNs), Gradient visualization [15], Perturbation [10], Class Activation Map (CAM) [21] are three of the widely adopted methods.

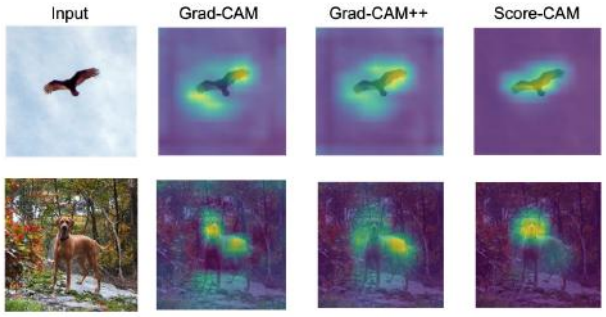

Figure 1. Visualization and comparison between our proposed method, Score-CAM with two related work, Grad-CAM and Grd-CAM++. Score-CAM shows a potential high concentration at the object in the image.

Gradient-based methods backpropagate the gradient of a target class to the input layer to highlight image region that highly influences the prediction. [15] utilizes the derivative of target class score with respect to the input image to generate saliency map. Other works [1, 8, 17, 18, 20] make additional manipulation on original gradient and visually sharpen the saliency map. These maps are generally of low quality and still noisy [8]. Perturbation-Based approaches [3,5,6,9,10,19] work by perturbing original input and observe the change of the prediction of model. To find minimum region, these approaches usually need additional regularizations [6] and are time-costing.

CAM-Based explanation [4,12,21] provide visual explanation for a single input with a linear weighted combination of activation maps from convolutional layers. CAM [21] creates localized visual explanations but is architecture-sensitive, a global pooling layer [7] is required to follow the convolutional layer of interest. Grad-CAM [12] and its variations, *e.g.* Grad-CAM++ [4], intend to generalize CAM to

[1] https://github.com/haofanwang/Score-CAM

models without global pooling layers and finally adopted widely in the community. In this work, we revisit the use of gradient information in GradCAM and discuss our concerns of why gradients may not be an optimal solution to generalize CAM. Further, to address the limitations of gradient-based variations of CAM, we present a new post-hoc visual explanation method, named Score-CAM, where the importance of activation maps are encoded by the global contribution of the corresponding input features instead of the local sensitivity measurement, a.k.a gradient information. We summarize our contributions as follows:

(1) We introduce a novel gradient-free visual explanation method named Score-CAM, which bridges the gap between perturbation-based and CAM-based methods, and represents the weight of activation maps in an intuitively understandable way.

(2) We quantitatively evaluate the fairness of generated saliency maps of Score-CAM on Recognition tasks, specifically Average Drop / Average Increase and Deletion curve / Insertion curve metrics, and show that Score-CAM can find out better evidences of the target class.

(3) We also qualitatively evaluate the visualization and localization performance, and achieve better results on both tasks. Finally, we introduce the effective of its application as a debugging tool to analyze model misbehaviors.

2. Background

Class Activation Mapping (CAM) [21] is a technique for identifying discriminative regions by linearly weighted combination of activation maps of the last convolutional layer before the global pooling layer[2]. To aggregate over multiple channels, CAM incorporates the importance of each channel with the corresponding weight at the following fully connected layer, which generates a score as the class confidence. The biggest restriction of CAM is that not every model is designed with a global pooling layer and even a global pooling layer is present, sometimes more fully connected layers follow before the softmax function, *e.g*. VGG [15]. As a generalization of CAM, Grad-CAM [12] is applicable to a broader range of CNN architectures without requiring a specific architecture. Before we dive into the more analysis about CAM and its variations, we first introduce our notations in this paper.

Notation Consider a CNN neural network $Y = f(X)$ that takes an input $X \in \mathbb{R}^d$ and outputs a probability distribution Y, and we denote Y^c as the probablity of being classified as class c. For a given layer l, let A_l denotes the activations of layer l. Specifically, if l is chosen as a convolutional layer, let A_l^k denote the activation map for the k-th channel. Also denote the weight of the k-th neuron at layer l connecting two layer l and $l+1$ as $w_{l,l+1}$

Definition 1 (Class Activation Map) *Using the notation in Sec 2, consider a model f contains a global pooling layer l that takes the output from the last convolutional layer $l-1$ and feeds the pooled activation to a fully connected layer $l+1$ for classification. For a class of interest c, CAM L_{CAM}^c can be defined as*

$$L_{CAM}^c = ReLU(\sum_k \alpha_k^c A_{l-1}^k) \tag{1}$$

where

$$\alpha_k^c = w_{l,l+1}^c[k] \tag{2}$$

$w_{l,l+1}^c[k]$ is the weight for k-th neuron after global pooling at layer l.

The motivation behind CAM is that each activation map A_l^k contains different spatial information about the input X and the importance of each channel is the weight of the linear combination of the fully connected layer following the global pooling. However, if there is no global pooling layer or there is no (or more than one) fully connected layer(s), CAM will fail due to no definition of α_k^c. To resolve the problem, Grad-CAM extends the definition of α_k^c as the gradient of class confidence Y^c w.r.t. the activation map A_l. Formally, we have the following definition for Grad-CAM:

Definition 2 (Grad-CAM) *Using the notation in Sec 2, consider a convolutional layer l in a model f, given a class of interest c, Grad-CAM $L_{Grad-CAM}^c$ can be defined as*

$$L_{Grad-CAM}^c = ReLU(\sum_k \alpha_k^c A_l^k) \tag{3}$$

where

$$\alpha_k^c = GP(\frac{\partial Y^c}{\partial A_l^k}) \tag{4}$$

$GP(\cdot)$ denoted the global pooling operation [3].

Variations of GradCAM, like GradCAM++ [4], only differentiate in combinations of gradients to represent α_k^c. Therefore, we do not explicitly discuss the definitions but will include GradCAM++ as a comparison in the later sections.

Using the gradient to incorporate the importance of each channel towards the class confidence is a natural choice and it also guarantees that Grad-CAM reduces to CAM if there is only one fully connected layer following the chosen layer. Rethinking the concept of "importance" of each channel in the activation map, we show that Increase of Confidence

[2]Global Maximum Pooling and Global Average Pooling are two possible implementations but in the original CAM paper it shows the average pooling yields better visual explanations

[3]In the original Grad-CAM paper, the authors use Global Average Pooling and normalize the α_k^c to ensure $\sum_k \alpha_k^c = 1$

(definition to follow in Sec 3) is a better way to quantify the channel importance compared to the gradient information. We first discuss some issues regarding the use of gradient to measure importance then we propose our new measurement of channel importance in Sec 3.

2.1. Gradient Issue

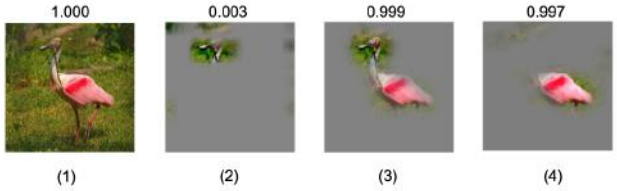

Figure 2. (1) is the input image, (2)-(4) are generated by masking input with upsampled activation maps. The weights for activation maps (2)-(4) are 0.035, 0.027, 0.021 respectively. The values above are the increase on target score given (1)-(4) as input. As shown in this example, (2) has the highest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

Saturation Gradient for a deep neural network can be noisy and also tends to vanish due to the saturation problem for Sigmoid function or the zero-gradient region of ReLU function. One of the consequences is that gradient of the output w.r.t input or the internal layer activation may be noisy visually, which is also one of the issues for Saliency Map method [14] that attributes input feature importance to the output. An example of gradient can be be visually noisy is shown in Fig 4.

False Confidence $L^c_{Grad-CAM}$ is a linear combination of each activation map. Therefore, given two activation map A^i_l and A^j_l, if the corresponding weight $\alpha^c_i \geq \alpha^c_j$, we are supposed to claim that the input region which generates A^i_l is at least as important as another region that generates A^j_l towards target class 'c'. However, it is easy to find counterexamples with false confidence in Grad-CAM: activation maps with higher weights show lower contribution to the network's output compared to a zero baseline. We randomly select activation maps and upsample them into the input size, then record how much the target score will be if we only keep highlighted region in the activation maps. An example is shown in Fig 2. The activation map corresponding to the 'head' part receives the highest weight but cause the lowest increase on the target score. This phenomenon may be caused by the global pooling operation on the top of the gradients and the gradient vanishing issue in the network.

3. Score-CAM: Proposed Approach

In this section, we first introduce the mechanism of proposed Score-CAM for interpreting CNN-based predictions. The pipeline of the proposed framework is illustrated in Fig 3. We first introduce our methodology is then introduced in Sec 3.1. Implementation details are followed in Sec 3.2.

3.1. Methodology

In contrast to previous methods [4, 12], which use the gradient information flowing into the last convolutional layer to represent the importance of each activation map, we incorporate the importance as the *Increase of Confidence*.

Definition 3 (Increase of Confidence) *Given a general function $Y = f(X)$ that takes an input vector $X = [x_0, x_1, ..., x_n]^\top$ and outputs a scalar Y. For a known baseline input X_b, the contribution c_i of $x_i, (i \in [0, n-1])$ towards Y is the change of the output by replacing the i-th entry in X_b with x_i. Formally,*

$$c_i = f(X_b \circ H_i) - f(X_b) \tag{5}$$

where H_i is a vector with the same shape of X_b but for each entry h_j in H_i, $h_j = \mathbb{I}[i = j]$ and $\circ$ denotes Hadamard Product.

Some related work has built similar concepts to Def 3. DeepLIFT [13] uses the difference of the output given an input compared to the baseline to quantify the importance signals propagating through layers. Two similar concepts *Average Drop %* and *Increase in Confidence* are proposed by GradCAM++ [4] to evaluate the performance of localization. We generate Def.3 to Channel-wise Increase of Confidence in order to measure the importance of each activation map.

Definition 4 *Given a CNN model $Y = f(X)$ that takes an input X and outputs a scalar Y. We pick an internal convolutional layer l in f and the corresponding activation as A. Denote the k-th channel of A_l by A^k_l. For a known baseline input X_b, the contribution A^k_l towards Y is defined as*

$$C(A^k_l) = f(X \circ H^k_l) - f(X_b) \tag{6}$$

where

$$H^k_l = s(Up(A^k_l)) \tag{7}$$

$Up(\cdot)$ denotes the operation that upsamples A^k_l into the input size [4] and $s(\cdot)$ is a normalization function that maps each element in the input matrix into [0, 1].

Use of Upsampling CIC first upsamples an activation map that corresponds to a specific region in the original input space, and then perturbs the input with the upsampled activation map. The importance of that activation map is obtained by the target score of masked input. Different from [9], where N masks with a size smaller than image size are generated through Monte Carlo sampling and then

[4] We assume that the deep convolutional layer outputs have smaller spatial size compared to the input.

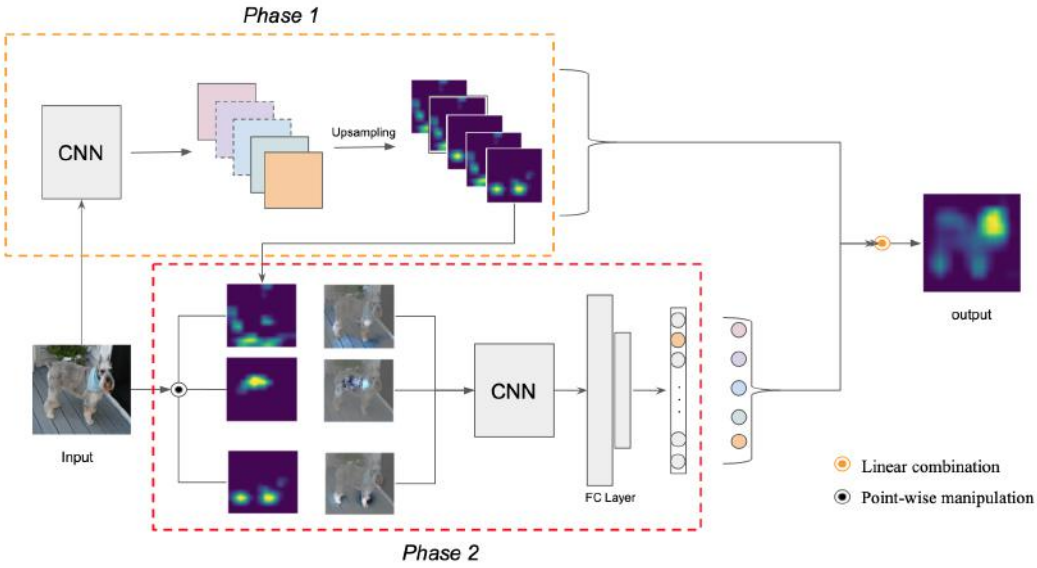

Figure 3. Pipeline of our proposed Score-CAM. Activation maps are first extracted in Phase 1. Each activation then works as a mask on original image, and obtain its forward-passing score on the target class. Phase 2 repeats for N times where N is the number of activation maps. Finally, the result can be generated by linear combination of score-based weights and activation maps. Phase 1 and Phase 2 shares a same CNN module as feature extractor.

upsampled each mask into input size, CIC does not require a process to generate masks. On the contrary, each upsampled activation map not only presents the spatial locations most relevant to an internal activation map, but also can directly work as a mask to perturb the input image.

Smoothing with Normalization *Increase of Confidence* essentially creates a binary mask H_i on the top of the input with only the feature of interest retained in the input. However, the binary mask may not be a reasonable choice when we are not interested in one pixel but a specific region in the input image. Instead of setting all elements to binary values, in order to a generate smoother mask H_L^k for an activation map, we normalize the raw activation values in each activation map into $[0, 1]$. We use the following normalization function in the Algorithm 1 of Score-CAM:

$$s(A_l^k) = \frac{A_l^k - \min A_l^k}{\max A_l^k - \min A_l^k} \tag{8}$$

Finally, we describe our proposed visual explanation method Score-CAM in Def.5. The complete detail of the implementation is described in Algorithm 1.

Definition 5 (Score-CAM) *Using the notation in Sec 2, consider a convolutional layer l in a model f, given a class of interest c, Score-CAM $L^c_{Score-CAM}$ can be defined as*

$$L^c_{Score-CAM} = ReLU(\sum_k \alpha_k^c A_l^k) \tag{9}$$

where

$$\alpha_k^c = C(A_l^k) \tag{10}$$

where $C(\cdot)$ denotes the CIC score for activation map A_l^k.

Algorithm 1: Score-CAM algorithm

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

Similar to [4, 12], we also apply a ReLU to the linear combination of maps because we are only interested in the features that have a positive influence on the class of interest. Since the weights come from the CIC score correspond-

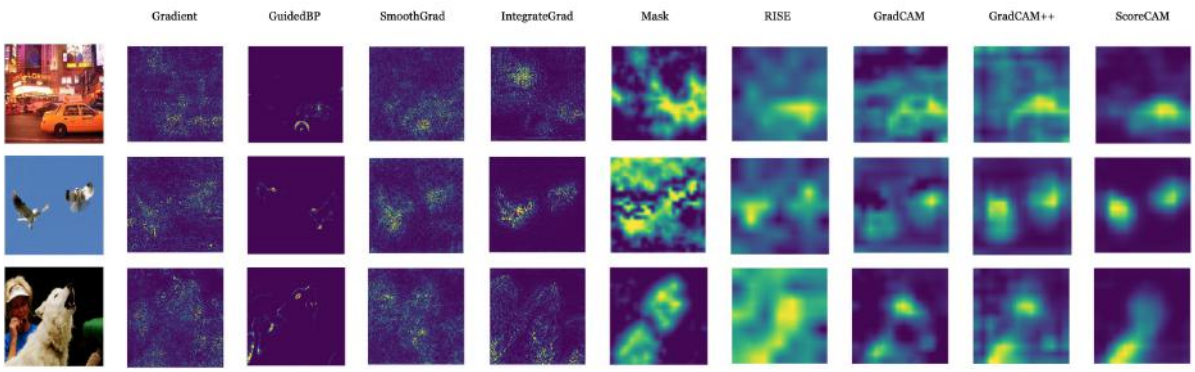

Figure 4. Visualization results of Vanilla Backpropagation [14], Guided Backpropagation [17], SmoothGrad [16], IntegrateGrad [18], Mask [6], RISE [9], Grad-CAM [12], Grad-CAM++ [4] and our proposed Score-CAM. More results are provided in Appendix.

ing to the activation maps on target class, Score-CAM gets rid of the dependence on gradient. Although the last convolution layer is a more preferable choice because it is end point of feature extraction [12], any intermediate convolutional layer can be chosen in our framework.

3.2. Normalization on Score

Each forward passing in neural network is independent, the score amplitude of each forward propagation is unpredictable and not fixed. The relative output value (post-softmax) after normalization is more reasonable to measure the relevance than absolute output value (pre-softmax). Thus, in Score-CAM, we represent weight as post-softmax value, so that the score can be rescaled into a fixed range.

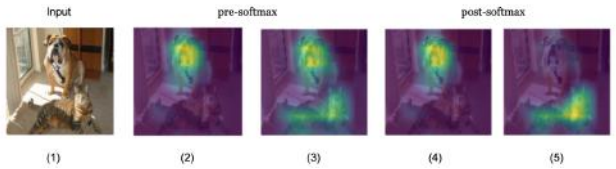

Figure 5. Effect of normalization. (2) and (4) are w.r.t 'boxer dog, (3) and (5) are w.r.t 'tiger cat. As shown, pre-softmax and post-softmax show a difference on class discrimination ability. We adopt post-softmax value in all other section in this paper.

Because of the varied range in each prediction, whether or not using softmax makes a difference. An interesting discovery is shown in Fig 5. The model predicts the input image as 'dog' which can be correctly highlighted no matter which type of score is adopted. But for target class 'cat', Score-CAM highlight both region of 'dog' and 'cat' if using pre-softmax logit as weight. On the contrary, Score-CAM with softmax can well distinguish two different categories, even though the prediction probability of 'cat' is lower than the probability of 'dog'. Normalization operation equips Score-CAM with good class discrimination ability.

4. Experiments

In this section, we conduct experiments to evaluate the effectiveness of the proposed explanation method. First, we qualitatively evaluate our approach via visualization on ImageNet in Sec 4.1. Second, we evaluate the fairness of explanation (how importance the highlighted region is for model decision) on image recognition in Sec 4.2. In Sec 4.3 we show the effectiveness for class-conditional localization of objects in a given image. The sanity check is followed in Sec 4.4. Finally, we employ Score-CAM as a debugging tool to analyze model misbehaviors in Sec 4.5.

In the following experiments, unless stated otherwise, we use pre-trained VGG16 network from the Pytorch model zoo[5] as a base model, more visualization results on other network architectures are provided in Appendix. Publicly available object classification dataset, namely, ILSVRC2012 val [11] is used in our experiment. For the input images, we resize them to $(224 \times 224 \times 3)$, transform them to the range $[0, 1]$, and then normalize them using mean vector $[0.485, 0.456, 0.406]$ and standard deviation vector $[0.229, 0.224, 0.225]$. No further pre-processing is performed.

4.1. Qualitative Evaluation via Visualization

4.1.1 Class Discriminative Visualization

We qualitatively compare the saliency maps produced by 8 state-of-the-art methods, namely gradient-based, perturbation-based and CAM-based methods. Our method generates more visually interpretable saliency maps with less random noises. Results are shown in Fig 4, more examples are provided in Appendix. As shown, in Score-CAM, random noises are much less than Mask [6], RISE [9], Grad-CAM [12] and Grad-CAM++ [4]. Our approach can also generate smoother saliency maps comparing with gradient-based methods.

[5]https://github.com/pytorch/vision

Table 1. Evaluation results on Recognition (lower is better in Average Drop, higher is better in Average Increase).

Method	Mask	RISE	GradCAM	GradCAM++	ScoreCAM
Average Drop(%)	63.5	47.0	47.8	45.5	**31.5**
Average Increase(%)	5.29	14.0	19.6	18.9	**30.6**

Figure 6. Class discriminative result. The middle plot is generated w.r.t 'bull mastiff', and the right plot is generated w.r.t 'tiger cat'.

We demonstrate that Score-CAM can distinguish different classes as shown in Fig 6. The VGG-16 model classifies the input as 'bull mastiff' with 49.6% confidence and 'tiger cat' with 0.2% confidence, our model correctly gives the explanation locations for both of two categories, even though the prediction probability of the latter is much lower than the probability of the former. It is reasonable to expect Score-CAM to distinguish different categories, because the weight of each activation map is correlated with the response on target class, and this equips Score-CAM with good class discriminative ability.

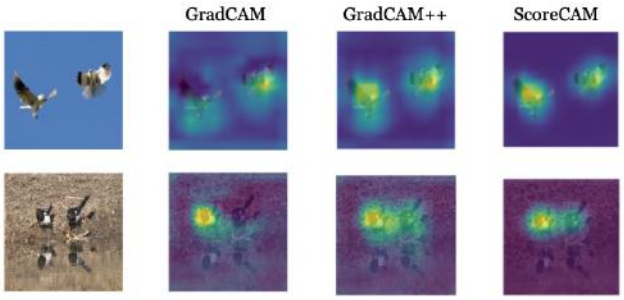

Figure 7. Results on multiple objects. As shown in this example, Grad-CAM only tends to focus on one object, while Grad-CAM++ can highlight all objects. Score-CAM further improves the quality of finding all evidences.

4.1.2 Multi-Target Visualization

Score-CAM can not only locate single object accurately, but also show better performance on locating multiple same class objects than previous works. The result is shown in Fig 7, Grad-CAM [12] tends to only capture one object in the image, Grad-CAM++ [4] and Score-CAM both show ability to locate multiple objects, but the saliency maps of Score-CAM are more focused than Grad-CAM++.

As the weight of each activation map is represented by its score on the target class, each target object with a high confidence score predicted by the model can be highlighted independently. Therefore, all evidences related to target class can get responses and are assembled through linear combination.

4.2. Faithfulness Evaluation via Image Recognition

We first evaluate the faithfulness of the explanations generated by Score-CAM on the object recognition task as adopted in [4]. The original input is masked by point-wise multiplication with the saliency maps to observe the score change on the target class. In this experiment, rather than do point-wise multiplication with the original generated saliency map, we slightly modify by limiting the number of positive pixels in the saliency map (50% of pixels of the image are muted in our experiment). We follow the metrics used in [4] to measure the quality, the Average Drop is expressed as $\sum_{i=1}^{N} \frac{\max(0, Y_i^c - O_i^c)}{Y_i^c} \times 100$, the Increase In Confidence (also denote as Average Increase) is expressed as $\sum_{i=1}^{N} \frac{Sign(Y_i^c < O_i^c)}{N}$, where Y_i^c is the predicated score for class c on image i and O_i^c is the predicated score for class c with the explanation map region as input. $Sign$ presents an indicator function that returns 1 if input is True. Experiment conducts on the ImageNet (ILSVRC2012) validation set, 2000 images are randomly selected. Result is reported in Table 1.

As shown in Table 1, Score-CAM achieves 31.5% average drop and 30.6% average increase respectively, and outperforms other perturbation-based and CAM-based methods by large scale. A good performance on recognition task reveals that Score-CAM can successfully find out the most distinguishable region of the target object, rather than just finding what human think is important. Here we do not compare with gradient-based methods, because of their different visual properties. Results on recognition task demonstrates that Score-CAM could more faithfully reveal the decision making process of the original CNN model than previous approaches.

Furthermore, for a more comprehensive comparison, we also evaluate our method on deletion and insertion metrics which are proposed in [9]. As supplementary to Average Drop and Average Increase metrics, the deletion metric measures a decrease in the probability of the predicted class as more and more important pixels are removed, where the importance of each pixel is obtained from the generated saliency map. A sharp drop and thus a low area under the

Table 2. Comparative evaluation on Energy-Based Pointing Game (higher is better).

	Grad	Smooth	Integrated	Mask	RISE	GradCAM	GradCAM++	ScoreCAM
Proportion(%)	41.3	42.4	44.7	56.1	36.3	48.1	49.3	**63.7**

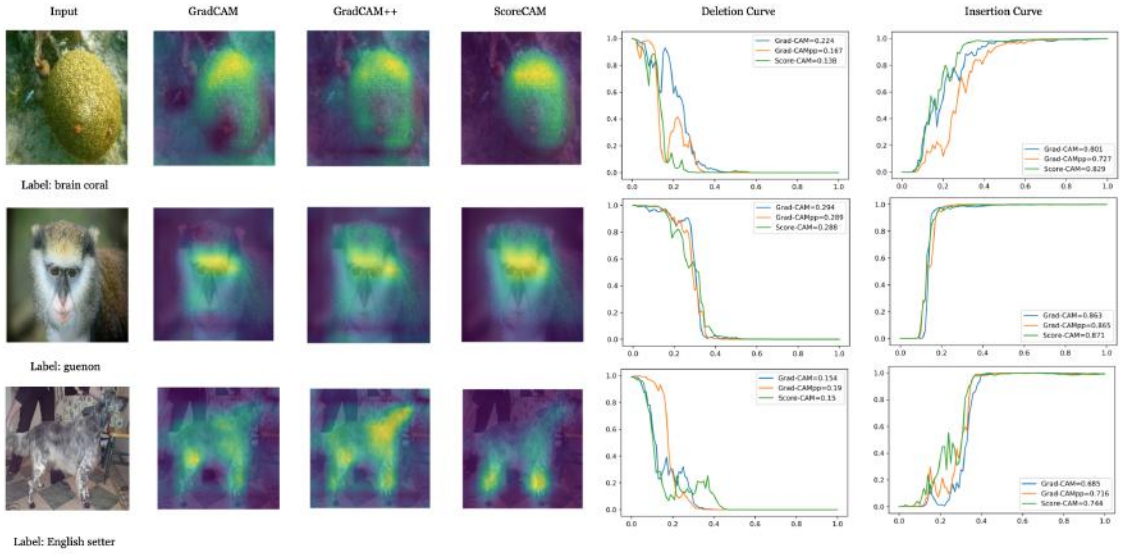

Figure 8. Grad-CAM, Grad-CAM++ and Score-CAM generated saliency maps for representative images with deletion and insertion curves. In deletion curve, a better explanation is expected to drop faster, the AUC should be small, while in increase curve, it is expected to increase faster, the AUC should be large.

probability curve (as a function of the fraction of removed pixels) means a good explanation. The insertion metric, on the other hand, measures the increase in probability as more and more pixels are introduced, with higher AUC indicative of a better explanation.

Table 3. Comparative evaluation in terms of deletion (lower is better) and insertion (higher is better) scores.

	Grad-CAM	Grad-CAM++	Score-CAM
Insertion	0.357	0.346	**0.386**
Deletion	0.089	0.082	**0.077**

There are several ways of removing pixels from an image [5], all of these approaches have different pros and cons. Thus, in this experiment, we simply remove or introduce pixels from an image by setting the pixel values to zero or one with step 0.01 (remove or introduce 1% pixels of the whole image each step). Example are shown in Fig 8. The average result over 2000 images is reported in Table 3, where our approach achieves better performance on both metrics compared with gradient-based CAM methods.

4.3. Localization Evaluation

In this section, we measure the quality of the generated saliency map through localization ability. Extending from pointing game which extracts maximum point in saliency map to see whether the maximum falls into object bounding box, we treat this problem in an energy-based perspective. Instead of using only the maximum point, we care about how much energy of the saliency map falls into the target object bounding box. Specifically, we first binarize the input image with the bounding box of the target category, the inside region is assigned to 1 and the outside region is assigned to 0. Then, we point-wise multiply it with generated saliency map, and sum over to gain how much energy in target bounding box. We denote this metric as $Proportion = \frac{\sum L^c_{(i,j)\in bbox}}{\sum L^c_{(i,j)\in bbox} + \sum L^c_{(i,j)\notin bbox}}$, and call this metric an energy-based pointing game.

As we observe, it is common in the ILSVRC validation set that the object occupies most of the image region, which makes these images not suitable for measure the localization ability of the generated saliency maps. Therefore, we randomly select images from the validation set by removing images where object occupies more than 50% of the whole image, for convenience, we only consider these images with only one bounding box for target class. We experiment on 500 random selected images from the ILSVRC 2012 validation set. Evaluation result is reported in Table 2, which shows that our method outperforms previous works by a large scale, more than 60% energy of saliency map

falls into the ground truth bounding box of the target object. This is also a corroboration that the saliency map generated by Score-CAM comes with less noises. We don't compare with Guided BackProp [17] because it works similar to an edge detector rather than saliency map (heatmap). In addition, it should be more accurate to evaluate on segmentation label rather than object bounding box, we will add it in our future work.

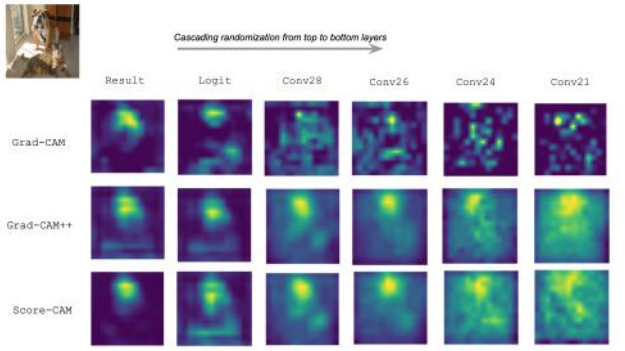

Figure 9. Sanity check results by randomization. The first column is the original generated saliency maps. The following columns are results after randomizing from top the layers respectively. The results show sensitivity to model parameters, the quality of saliency maps can reflect the quality of the model. All three types of CAM pass the sanity check.

4.4. Sanity Check

[2] finds that reliance, solely, on visual assessment can be misleading. Some saliency methods [17] are independent both of the model and of the data generating process. We adopt model parameter randomization test proposed in [2], to compare the output of Score-CAM on a trained model with the output of a randomly initialized untrained network of the same architecture. As shown in Fig 9, as the same as Grad-CAM and Grad-CAM++, Score-CAM also passes the sanity check. The Score-CAM result is sensitive to model parameter and therefore can reflect the quality of model.

4.5. Applications

A good post-hoc visual explanation should not only tell where does the model look at, but also help researchers analyze their models. We claim that much previous work treat visual explanation as a way to do localization, but ignore the usefulness in helping to analyze the original model. In this part, we suggest how to harness the explanations generated by Score-CAM for model analysis, and provide insights for future exploration.

We observe that Score-CAM can work well on localization task even the classification performance of the model is bad, but as the classification performance improves, the noise in saliency map decreases and focuses more on important region. The noise suggests the classification performance. This also can work as a hint to determine whether a model has converged, if the generated saliency map does not change anymore, the model may have converged.

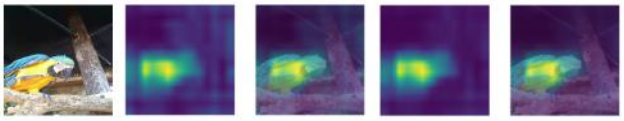

Figure 10. The left is generated by no-finetuning VGG16 with 22.0% classification accuracy , the right is generated by finetuning VGG16 with 90.1% classification accuracy. It shows that the saliency map becomes more focused as the increasing of classification accuracy.

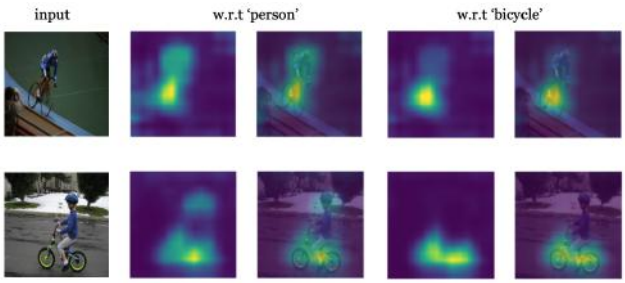

Figure 11. The left column is input example, middle is saliency map w.r.t predicted class (person), right is saliency map w.r.t target class(bicycle).

Besides, Score-CAM can help diagnose why the model makes a wrong prediction and identify dataset bias. The image with label 'bicycle' is classified as 'person' in Fig 11. Saliency maps for both classes are generated. By comparing the difference, we know that 'person' is correlated with 'bicycle' because 'person' appears in most of 'bicycle' images in training set, and 'person' region is the most distractive part that leads to mis-classification.

5. Conclusion

In this paper, we proposed a new kind of CAM variants, named Score-CAM method, as a better visual explanation. Score-CAM incorporates Increase in Confidence in the designing of weight for each activation map, gets rid of the dependence on gradients and has a more reasonable weight representation. We provide an in-depth analysis of motivation, implementation, qualitative and quantitative evaluations. Our method outperforms all previous CAM-based methods and other state-of-the-art methods in recognition and localization evaluation metrics. Future work involves explore the connection of different weight representations in other kind of CAM variants.

References

[1] J. Adebayo, J. Gilmer, I. Goodfellow, and B. Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv preprint arXiv:1810.03307*, 2018.

[2] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim. Sanity checks for saliency maps. In *Advances in Neural Information Processing Systems*, pages 9505–9515, 2018.

[3] C.-H. Chang, E. Creager, A. Goldenberg, and D. Duvenaud. Explaining image classifiers by counterfactual generation. 2018.

[4] A. Chattopadhay, A. Sarkar, P. Howlader, and V. 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. IEEE, 2018.

[5] P. Dabkowski and Y. Gal. Real time image saliency for black box classifiers. In *Advances in Neural Information Processing Systems*, pages 6967–6976, 2017.

[6] R. C. Fong and A. Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3429–3437, 2017.

[7] M. Lin, Q. Chen, and S. Yan. Network in network. *arXiv preprint arXiv:1312.4400*, 2013.

[8] 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.

[9] V. Petsiuk, A. Das, and K. Saenko. Rise: Randomized input sampling for explanation of black-box models. *arXiv preprint arXiv:1806.07421*, 2018.

[10] M. T. Ribeiro, S. Singh, and C. 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*, pages 1135–1144. ACM, 2016.

[11] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015.

[12] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 618–626, 2017.

[13] A. Shrikumar, P. Greenside, and A. Kundaje. Learning important features through propagating activation differences. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3145–3153. JMLR. org, 2017.

[14] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[16] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[17] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[18] M. Sundararajan, A. Taly, and Q. Yan. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 3319–3328. JMLR. org, 2017.

[19] J. Wagner, J. M. Kohler, T. Gindele, L. Hetzel, J. T. Wiedemer, and S. Behnke. Interpretable and fine-grained visual explanations for convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9097–9107, 2019.

[20] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[21] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2921–2929, 2016.

Appendix

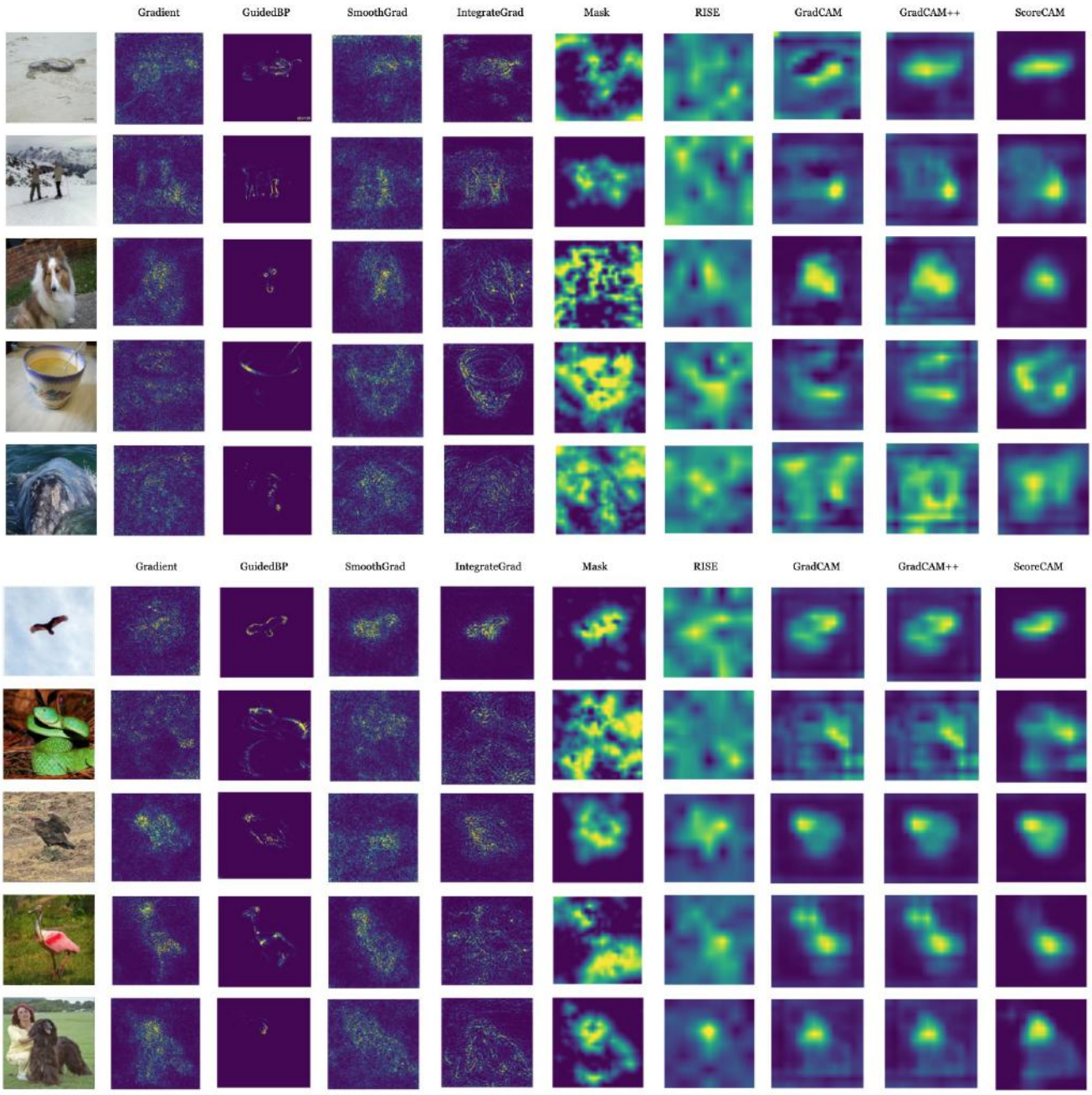

Figure 12. Visualization results on single object.

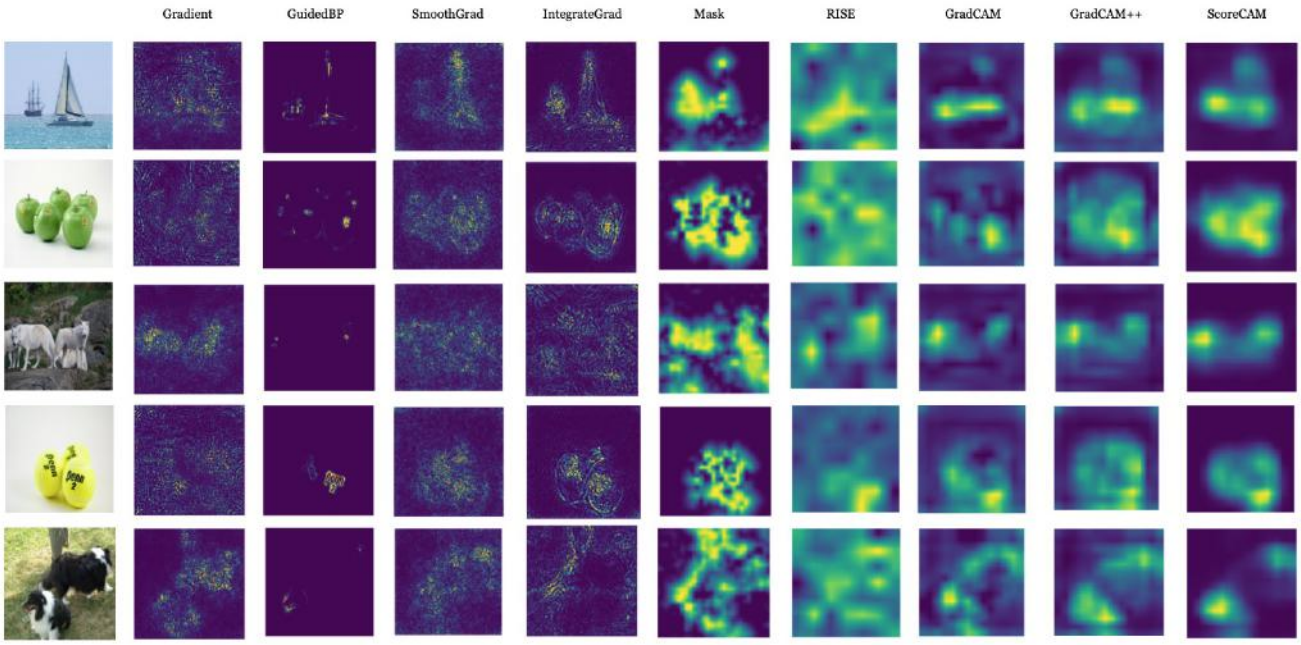

Figure 13. Visualization results on multiple objects.

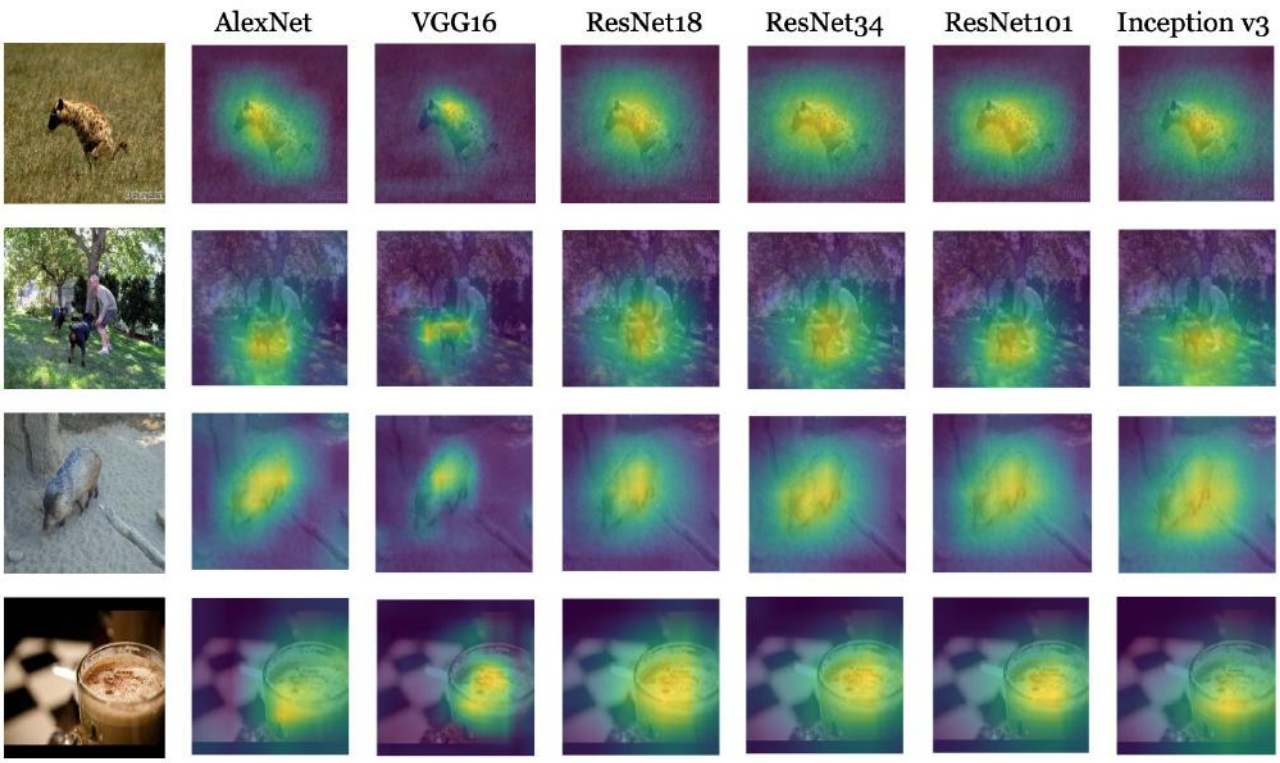

Figure 14. Score-CAM results on other model architectures.