Generating Self-Attention Activation Maps for Visual Interpretations of Convolutional Neural Networks

Yu Liang[a], Maozhen Li[b,*] and Changjun Jiang[a]

[a]Department of Computer Science and Technology, Tongji University, Shanghai 201804, China
[b]Department of Electronic and Electrical Engineering, Brunel University London, Uxbridge, UB8 3PH, UK.

ARTICLE INFO

Keywords:
Interpretable machine learning
Black-box models
Transparent models
Deep learning
Explainable artificial intelligence

ABSTRACT

In recent years, many interpretable methods based on class activation maps (CAMs) have served as an important judging basis for the predictions of convolutional neural networks (CNNs). However, these methods still suffer from the problems of gradient noise, weight distortion, and perturbation deviation. In this work, we present self-attention class activation map (SA-CAM) and shed light on how it uses the self-attention mechanism to refine the existing CAM methods. In addition to generating basic activation feature maps, SA-CAM adds an attention skip connection as a regularization item for each feature map which further refines the focus area of an underlying CNN model. By introducing an attention branch and constructing a new attention operator, SA-CAM greatly alleviates the limitations of the CAM methods. The experimental results on the ImageNet dataset show that SA-CAM can not only generate highly accurate and intuitive interpretation but also have robust stability in adversarial comparison with the state-of-the-art CAM methods.

1. Introduction

The complexity of convolutional neural networks (CNNs) in the training process has led to uncertainty and unreliability of model prediction, especially in sensitive realms such as medical imaging [8] and autonomous driving [2]. With an increasing demand for transparency in CNNs, many interpretable works [19, 22, 16] have been published. Existing interpretable work on CNN can be divided into data-driven and model-driven interpretable methods. A data-driven interpretable method employs perturbation data to compare the prediction results generated by the same CNN model and extracts the content that needs to be explained in the CNN model. A model-driven interpretable method evaluates the gradients in CNN and selects data points with higher explanatory power through threshold division to generate an interpretation [13].

As one of the influential model-driven methods, the core idea of CAM [25] is the accumulation of weights dot product feature maps. Through the cumulative heating maps provided by the weights dot product feature maps, CAM can obtain the key information learned by a CNN model during the training process. However, CAM uses the global pooling layer to extract feature information which requires proper modification of the original model structure.

Grad-CAM [19] uses the gradient of the last convolutional layer to assign weights to each feature map to visualize CNNs of any structure without modifications. To further solve the gradient-based limitations of Grad-CAM, Score-CAM [22] and its acceleration method Group-CAM [24] use data-driven methods to generate CAM interpretations. Score-CAM extracts feature maps from the last convolutional layer, and generates corresponding prediction scores through perturbation data instead of the weights generated by gradients.

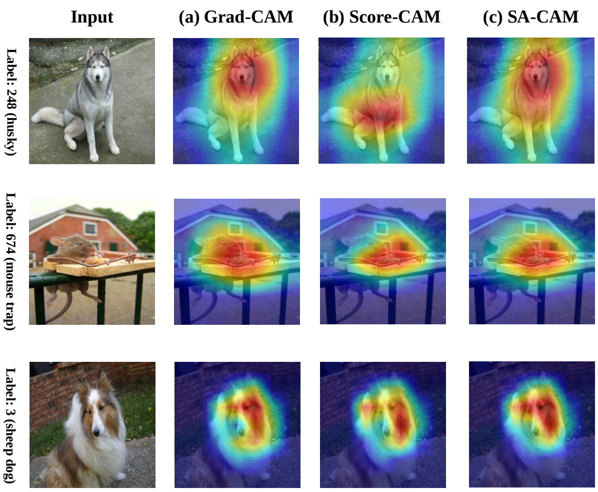


Figure 1: Heatmap visualized by SA-CAM, Score-CAM and Grad-CAM using CAM-toolbox [9].

Although Score-CAM can reduce weight distortion, it still introduces perturbation-based errors. The perturbation-based method usually uses different mask types to cover the up-sampling feature maps for re-prediction. The interference information brought by the mask may cause new errors in the model. To address these limitations, there are two ideas worth trying. The first is to make gradient-based methods and perturbation-based methods supervise each other to focus on key information of common concern. The other is to perform a perturbation-based method without introducing a new redundant mask.

In this paper, we revisit the self-attention mechanism and propose a self-attention mechanism-based CAM generation architecture called SA-CAM. SA-CAM improves the calcu-

*Corresponding authors
✉ maozhen.li@brunel.ac.uk (M. Li)
ORCID(s): 0000-0002-5095-4131 (Y. Liang)

lation method of perturbation-based scores by removing redundant masks and introduces a self-attention gradient term which provides an error correction for the original perturbation-based weights to focus on key information of common concern. It can be observed from the three examples visualized in Figure 1 that the interpretations generated by SA-CAM are more robust and more in line with human intuition than the existing Score- CAM and Grad-CAM methods. Specifically, our contributions are as follows:

(1) We propose a CAM framework based on the self-attention mechanism. This framework alleviates the weight distortion and perturbation deviation problem by introducing a gradient-based skip connection to effectively limit the perturbation-based weights and make the interpretation focus on key information of common concern. This framework is suitable for most deep neural network models with convolutional layer structure as the core and does not require additional components.

(2) We design a new score function for the perturbation-based framework, which can further alleviate perturbation deviation without introducing a new mask. Since the perturbation-based self-attention mechanism is only used to limit the weight provided by the gradient-based score in SA-CAM, it can minimize perturbation-based errors and improve robustness through this score function.

(3) We qualitatively evaluate the effectiveness of SA-CAM on visualization and adversarial interpretation based on the ImageNet dataset. The results show that SA-CAM does not incur both gradient and perturbation based errors, and can generate more robust interpretations. In multi-target tasks and adversarial interpretation tasks, SA-CAM performs well due to its robust interpretation.

The remainder of this work is organized as follows. Section 2 reviews related work on CAMs. Section 3 presents the design of the SA-CAM framework. Section 4 conducts comprehensive experiments and validates the performance of SA-CAM in comparison with state-of-the-art results. Section 5 concludes the paper and points out some future work.

2. Related Work

This section reviews related work from the aspects of gradient-based CAM and perturbation-based CAM.

2.1. Gradient-based CAM

CAM is a visualization method first proposed in [25]. This method displays its decision-making basis in the form of a heatmap. For the training process of a CNN model, the last convolutional layer has the most abundant spatial and semantic information after multiple iterations of convolution and pooling operations. However, this information contained in the last convolutional layer is difficult for human beings to understand and display in a visual way. Therefore, in order for the CNN to give a reasonable explanation for its classification results, it is necessary to make full use of the last convolutional layer. To explain why the result of the classification is c, we take all the weights w^c corresponding to the c and find the weighted sum of their corresponding feature maps.

Let function $F_l^c(I)$ be a CNN training process which takes an input $I \in \mathbb{R}$ and F be a CNN model, $f(I)$ represents the predicted probability of image I in $F_l^c(I)$and $A^l(I)$ represents feature maps in the layer l of a CNN. For a class of interest c, the CAM explanation, written as CAM^c can be defined as:

$$CAM^c = ReLU\left(\sum_n w_l^c A^l(I)\right) \tag{1}$$

where

- w_l^c represents the weight of each neuron in layer l based on category c.
- n represents the number of channels in l.

Eq. (1) calculates each feature map and generates a heatmap from the final weighted sum. Since the size and feature map of this result is consistent, we only need to upsample it and overlay it on the original image to get a complete class activation map.

However, the CAM method proposed in [25] is limited in that it requires modification of the structure of the original model, which leads to the need to re-train the model. As an improvement of CAM, Grad-CAM [19] can be applied to many types of CNN models without modifying the original model structure.

The Grad-CAM is a CAM interpretation method that uses the global average of the gradient to calculate the weights. For a class of interest c, the CAM interpretation, written as CAM$^c_{Grad}$ can be defined as:

$$CAM^c_{Grad} = ReLU\left(\sum_n mean(\frac{\partial y^c}{\partial A^l(I)})A^l(I)\right) \tag{2}$$

where

- $mean(\cdot)$ is an averaging function which represents the global pooling operation.
- y^c is the value before SoftMax function.

2.2. Perturbation-based CAM

Mask perturbation refers to an interpretation that changes the original data by masking the part of input data and systematically analyzes the contrast result. Most of the perturbation-based methods [23, 6, 4, 21] are all based on the smallest sufficient region (SSR) or the small destroying region (SDR) proposed in [4] which suffers from the shortcomings of the perturbation-based methods.

The SSR written as I_{ssr} is the smallest region of I that allows the $max(P)$ where $P = f(I_{ssr})$ represents the classification accuracy of I_{ssr} in category c. The SDR written as I_{sdr}

is the smallest region of I that allows the $max(P)$ where $P = f(I - I_{sdr})$ represents the classification accuracy of $I - I_{sdr}$ in category c.

Based on the principles of SSR and SDR, Score-CAM provides mask perturbation for the feature maps of the last layer of CAM and thus gives up the weights provided by the gradient. For each feature map, Score-CAM first performs an up-sampling operation and masks some areas in the up-sampled image through mask perturbation. Through the probability prediction of the masked up-sampled image, Score-CAM generates a corresponding score as the new weight of each up-sampling map. For a class of interest c, the Score-CAM interpretation written as CAM^c_{Score} can be defined as:

$$CAM^c_{Score} = ReLU\left(\sum_n S(A^l(U_{mask}))A^l(I)\right) \tag{3}$$

where

$$U_{mask} = Mask * I + (1 - Mask) * \mu \tag{4}$$

- $S(\cdot)$ is the SoftMax function.
- U is the up-sampling of feature maps.
- $Mask$ is a mask with a value between 0 and 1.
- μ is an average colour.

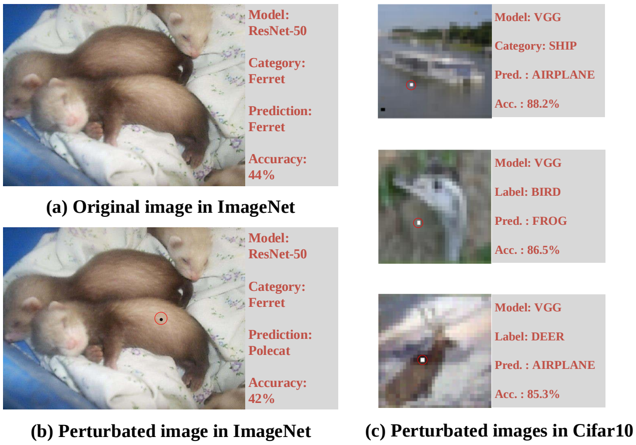
Group-CAM [24] is a lightweight method of Score-CAM. Its essence is to group preprocessed feature maps to calculate scores and perform the same accumulation operation as Score-CAM.

2.3. Discussions

The Score-CAM methods are completely different from gradient-based CAM methods. The core differences of these methods are mainly reflected in the calculation methods of weights after upsampling feature maps which directly leads to the difference in the form of expression of CAM.

Gradient-based CAM methods accumulate the up-sampled feature maps based on the weights generated by gradients. However, due to the existence of nonlinear activation functions such as ReLU, a gradient-based CAM is prone to errors. Take Grad-CAM in Figure 1 as an example, Grad-CAM may focus on some background that has nothing to do with the target. As an improvement of the Grad-CAM, Score-CAM does not rely on gradient generation. Score-CAM is more close to the data-driven interpretation method but uses the feature maps in the last layer of the model to replace the original image as an input.

Perturbation-driven interpretation methods suffer from the introduction of redundant information when masking a part of the image, which may distort the prediction result. Figure 2 shows multiple examples of misleading the CNN model to make incorrect judgments through a single pixel perturbation. Figure 2 (a) and Figure 2 (b) respectively show the original example based on the ImageNet validation dataset and the result after black pixel perturbation. Figure 2(c) shows the prediction results of the perturbed image after white pixel perturbation on the Cifar10 [12]. We can find from Figure 2 that just after introducing a point or pixel, the prediction results of the model have changed. Therefore, when introducing a large-scale perturbation mask, there is also a high probability that the model will misjudge and cause errors.



(a) Original image in ImageNet

(b) Perturbated image in ImageNet

(c) Perturbated images in Cifar10

Figure 2: Misleading the CNN model to make incorrect judgments through a single pixel perturbation.

By comparing perturbation-based and gradient-based CAM methods as shown in Figure 1, 2, 4 and 7, we observe that these two types of methods are susceptible to gradient and perturbation errors respectively, which are not conducive to generating robust interpretations. How to alleviate the problems of perturbation-based and gradient-based CAM has become the key to further improvement.

3. The Design of SA-CAM

In this section, we present SA-CAM which combines perturbation-based and gradient-based CAM methods by a self-attention regularization term to generate a more robust interpretation.

To alleviate the limitations mentioned in Section 2.3, a more reasonable idea is to allow the gradient-based and perturbation-based methods mutually supervised to avoid respective defects. Based on this idea, we propose a self-attention interpretation CAM method that combines gradient-based and perturbation-based CAM methods. Specifically, for a class of interest c, the self-attention interpretation written as CAM^c_{SA} can be defined as:

$$CAM^c_{SA} = ReLU\left(\sum_n S(\cdot) \odot K(\cdot)A^l(I)\right) \tag{5}$$

- $S(\cdot)$ represents the score generated by Score-CAM.

- $K(\cdot)$ represents the self-attention weight generated from the internal gradient of CNN.

$K(\cdot)$ performs the mean operation of the gradient for the feature maps in the last layer of a CNN and provides a mutual supervision weight for $S(\cdot)$ by $K(\cdot)$. Eq. (5) enables SA-CAM to focus on the common concern areas of gradient-based and perturbation-based CAM methods while ignoring the errors provided by $K(\cdot)$ and $S(\cdot)$ respectively.

Since Eq. (5) only provides a corresponding weight for each feature map and does not calculate the priority of each upsampled pixel, we further propose a vector $Q(\cdot)$ to represent the priority of each pixel in the feature maps which can be defined in Eq. (6).

$$CAM^{c}_{SA} = ReLU\left(\sum_{n} S(\cdot) \odot K(\cdot) \odot Q(\cdot) A^{l}(I)\right) \quad (6)$$

- $Q(\cdot)$ represents the priority of each pixel in the feature maps.

Eq. (6) provides a more robust weight and refines the weight of each pixel to make the interpretation extremely accurate.

Although Eq. (6) alleviates the errors of perturbation-based and gradient-based CAM methods through mutual supervision mechanism, this continuous dot multiplication mechanism can easily lead to new problems of weight attenuation.

The reason for weight attenuation is that after the CAM method completes the dot multiplication of the weight and activation values, it will re-normalize through Eq. (7) which leads to the values of $S(\cdot)$, $K(\cdot)$ and $Q(\cdot)$ varying between 0 and 1. If $\sum\limits_{n} S(\cdot) \odot K(\cdot) \odot Q(\cdot)$ is used as the corresponding weights, Eq. (7) may generate a small gap between the value $Max(A^{l}(I) \odot w^{c}_{l}))$ and $Min(A^{l}(I) \odot w^{c}_{l}))$ that may make the interpretation lose discernment when we expand it to a multiple of 255 colour gamut. Therefore, the operators provided by Eq.(6) are not suitable for nesting into a unified CAM framework to generate more robust interpretations, we have to carry out a formal transformation to the Eq.(6) calculation method.

$$CAM_{normal} = \frac{A^{l}(I) \odot w^{c}_{l} - Min(A^{l}(I) \odot w^{c}_{l})}{Max(A^{l}(I) \odot w^{c}_{l})} \quad (7)$$

To better focus on the areas that both gradient-based and perturbation-based CAMs pay attention to while avoiding the risk of weight attenuation, we can approximate the matrix dot multiplication paradigm to matrix addition. For normalized $S(\cdot)$ and $K(\cdot)$, the accumulation of $S(\cdot)$ and $K(\cdot)$ not only produces an effect similar to $S(\cdot) \odot K(\cdot)$ but also avoids weight attenuation.

We use the attention weight generated by the gradient as a Skip Connection in Score-CAM to accumulatively generate the explanation. For a class of interest c, the CAM^{c}_{SA} can be re-defined as Eq. (8):

$$CAM^{c}_{SA} = ReLU\left(\sum_{n}(S(\cdot) + K(\cdot)) \odot Q(\cdot) A^{l}(I)\right) \quad (8)$$

Eq. (8) overcomes the shortcomings in Eq. (6) while retaining the advantages of $S(\cdot)$ and $K(\cdot)$, and the weight of $Q(\cdot)$ further reduces the error in each feature map.

Based on the closed-loop of the SA-CAM and the simplicity of expression, we further re-write Eq. (8) and only introduce a single regularization term $SA(I)$ as shown in Eq. (9).

$$CAM^{c}_{SA} = ReLU\left(\sum_{n}(S(\cdot) + SA(\cdot)) A^{l}(I)\right) \quad (9)$$

The $SA(I)$ can be expressed as:

$$\begin{aligned} SA(I) &= \lambda_1 K(I) + \lambda_2 Q(I) \\ &= \sum_{i}^{a} \sum_{j}^{b} \left(\frac{\lambda_1}{i \times j} \frac{\partial y^{c}}{\partial A^{l}(I)} + \alpha^{c}_{ij} ReLU\left(\frac{\partial y^{c}}{\partial A^{l}(I)}\right) \right) \end{aligned} \quad (10)$$

where

$$\alpha^{c}_{ij} = \frac{\lambda_2 \cdot \frac{\partial^2 y^{c}}{(\partial A^{l}_{ij})^2}}{2 \frac{\partial^2 y^{c}}{(\partial A^{l}_{ij})^2} + \sum_a \sum_b A^{l}_{ab} \left\{ \frac{\partial^3 y^{c}}{(A^{l}_{ij})^3} \right\}} \quad (11)$$

- λ is the impact factor of regularization.
- (i, j) are the length and height of A^{l}
- (i, j) and (a, b) are iterators over the same activation map A^{l}.

Based on Eq. (9), we can get the overall framework of SA-CAM. As shown in Figure 3, the process of generating a heatmap in SA-CAM consists of two parts in total. The first part is the accumulation of the score function-Score(FP+M) which represents the $S(\cdot)$ part in Eq. (9). The second part is Attention(input) which gets the attention value through extracting $SA(\cdot)$ in the last convolutional layer of the CNN model and adding the results of the first part point by point.

For the calculation method of the score function, we also provide corresponding solutions to overcome the limitations of the work [22]. As mentioned in Section 2, most of the existing calculation methods for the score function are to add additional masks to highlight the core area in the upsampling maps. But we can find from Figure 2 that adding additional masks may bring additional judgment evidence to the model which leads to deviations in the results of the score function.

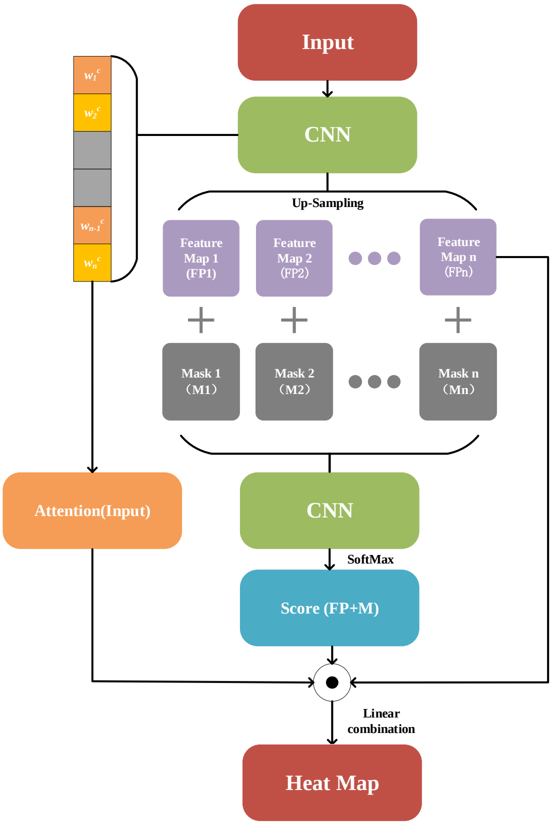


Figure 3: The pipeline of SA-CAM.

To alleviate this problem, we first extract the key areas in the up-sampling maps using image segmentation to direct prediction without introducing any masks. Let the original up-sampling map be U_{org}, the heat maps of feature maps be U_{heat} and the segmented maps be U_{seg}. U_{seg} is the segmentation result based on the heat map generated by feature maps, and U_{seg} can be obtained through Algorithm 1.

Algorithm 1 : The key area extraction algorithm (KAE).

```
Input:
    The original up-sampling map $U_{org}$, the heat maps $U_{heat}$
Output:
    The key areas in the up-sampling maps $U_{seg}$
FOR $(i, j), range(U_{heat}.shape[:])$ DO;
    $px = U_{heat}[i, j]$
    IF $px[0] > threshold$ :
        $U_{org}[i, j] = [255, 255, 255]$
FOR $(x, y), range(U_{org}.shape[:])$ DO;
    IF $pixdata[x, y][:] > 240$ :
        $pixdata[x, y] = (255, 255, 255, 0)$
$U_{seg} \leftarrow U_{org}$
```

By using U_{seg}, a reliable score can be generated without introducing a mask, but U_{seg} will change in the transformation process due to its irregularities. To further reduce the error of score, we also combine perturbation operators in [6] and U_{seg} as a restricted mask-based score $(1 + f(U_{heat})$ to mitigate the transformation errors. Since the value of U_{heat} is less than 1, $(1 + f(U_{heat})$ has little effect on the final score compared with U_{seg}, but it can improve the robustness of the final score.

$$Score = \begin{cases} f(U_{seg}) * (1 + f(U_{constant})) \\ f(U_{seg}) * (1 + f(U_{noise}) \\ f(U_{seg}) * (1 + f(U_{blur})) \end{cases} \tag{12}$$

where

$$\begin{cases} U_{constant} = m \odot U_{org} + (1 - m) \odot \mu_0 \\ U_{noise} = m \odot U_{org} + (1 - m) \odot \eta(u) \\ U_{blur} = \int g_{\sigma_0 m(u)}(v - u) U_{org}(v) dv \end{cases} \tag{13}$$

- $m : \Lambda \rightarrow [0, 1]$ represents a mask.
- u represents a pixel point in U_{org}
- σ_0 is the maximum isotropic standard deviation of the Gaussian blur kernel g_σ.

The complete detail of the implementation is described in Algorithm 2. In lines 1-3, we initialize the heat map that SA-CAM needs to generate through up-sampling and Eq. (7). Lines 4-9 execute a for loop which extracts the U_{seg} and $U_{constant}$ involved in Eq. (12). Line 10 and 11 respectively calculate the score based on Eq. (12) and Eq. (9). Line 12 provides the normalization of the weights and Line 13 is used to generate the interpretation provided by SA-CAM.

Algorithm 2 : SA-CAM algorithm.

```
Input:
    Image $I$, Model $F(I)$, class $c$, target layer $l$, the number of channels $n$
Output:
    $CAM^c_{SA-CAM}$
 1: $U_{org} \leftarrow [U_l^1, ..., U_l^n]$
 2: $U_{heat}[\cdot] \leftarrow the\ heat\ map\ of\ \mathrm{U}_{org}$
 3: $U_{mask}[\cdot] \leftarrow the\ masked\ of\ \mathrm{U}_{org}$
 4: FOR $i$ in $(0,n)$ DO;
 5:     $U_l^i \leftarrow Upsample(F_l^c(I))$
 6:     $U^i_{heat} \leftarrow Normalization(\mathrm{U}^i_{org})$
 7:     $U^i_{mask} = U_l^i \odot \mathrm{U}^i_{heat}$
 8:     $U_{blur}, U_{constant} \leftarrow blur(\mathrm{U}^i_{mask})$ or constant($U^i_{mask}$)
 9:     $U_{seg} \leftarrow KAE(\mathrm{U}^i_{mask})$
10: $Score^c = F(U_{seg}) * (1 + F(U_{blur})$
11: $SA^c = Score^c + \lambda_1 K(I) + \lambda_2 Q(I)$
12: $w_i^c \leftarrow \frac{exp(SA_i^c)}{\sum_{i=1}^n exp(SA_i^c)}$
13: $CAM^c_{SA-CAM} \leftarrow ReLU(\sum_{i=1}^n (w_i^c \odot F_l^c(I)))$
```

4. Experimental Results

In this section, we verify the effectiveness of the proposed SA-CAM method. In the following experiments, we

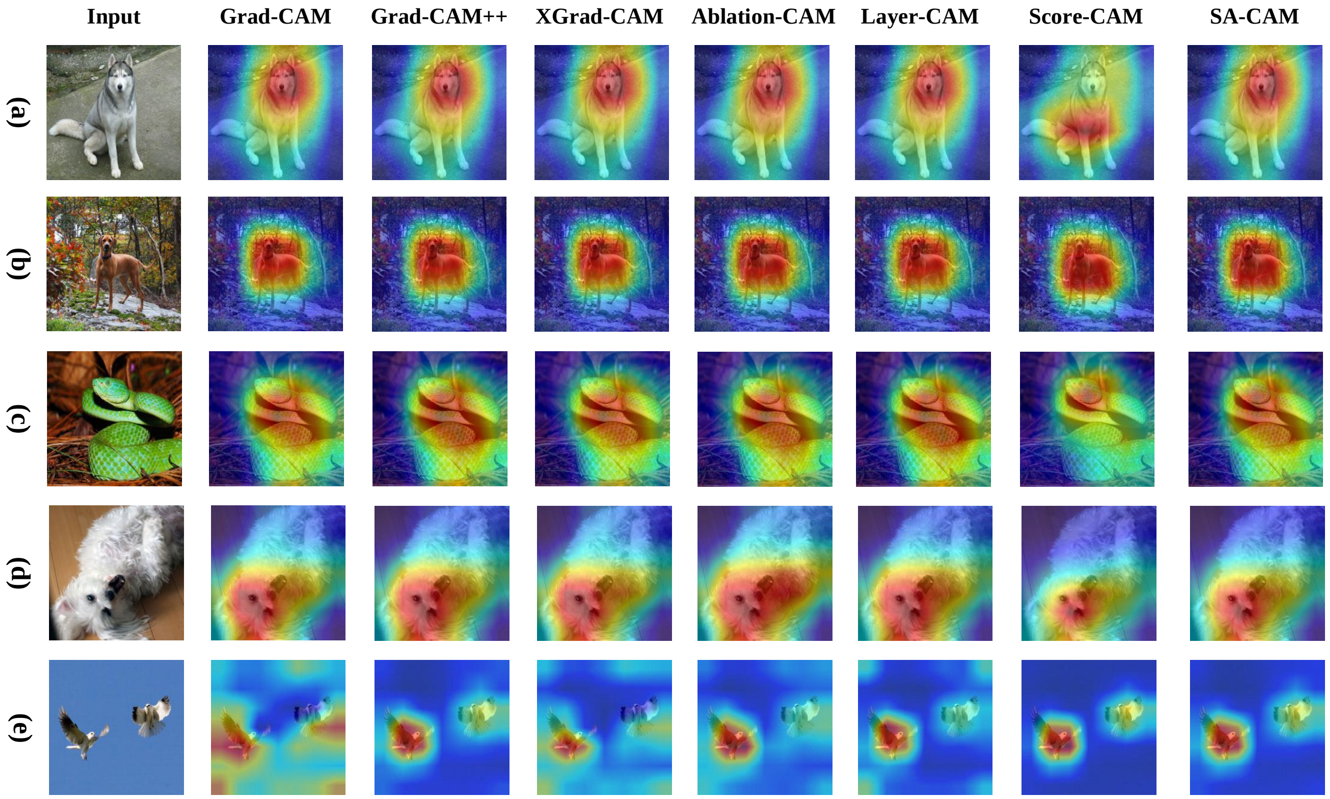


Figure 4: Demonstration of seven CAM-based methods on ResNet101.

employed VGGNet [20], ResNet [10] and MobileNet [18] as the baseline network from the Pytorch pre-trained model. Publicly available dataset ImageNet [5] and CAM tool pytorchcam [9] were also used in our experiments. For the input images, we resized them to (224 × 224 × 3), transformed them to the range [0, 1], and then normalized them using wildly accepted mean vector [0.485, 0.456, 0.406] and standardized deviation vector [0.229, 0.224, 0.225].

4.1. Visualization

This section provides a visual evaluation of SA-CAM on saliency maps in comparison with six CAM methods [3, 19, 7, 11, 22, 15] as shown in Figure 4. In this experiment, all other methods are gradient-based methods except for SA-CAM and Score-CAM. For the fairness of comparison, SA-CAM used the same mask strategy as Score-CAM to process the same input data. We tested five images which were extracted from the ImageNet validation dataset. The labels of Figure 4 (a-e) are "husky", "ridgeback dog", "green snake", "maltese dog" and "haliaeetus leucocephalus" respectively.

Figure 4 (a) shows the interpretation results of CAM methods for "husky". The interpretations provided by the gradient-based methods provide an accurate positioning which all focus on the critical part of the head in the image. However, the interpretation provided by Score-CAM does not focus on the head but the texture part. Through the correction of the weight by $SA(\cdot)$, SA-CAM not only focuses on head area of "husky" compared with Score-CAM but also focuses on more texture features than the gradient-based methods.

Figure 4 (b) shows the interpretation results of seven CAM methods for "ridgeback dog". Compared with Score-CAM and gradient-based CAMs, SA-CAM pays more attention to the head area and lower limb area which provides a more reliable interpretation. Figure 4 (c) and Figure 4 (d) respectively show the interpretations of CAM methods for "green snake" and "maltese dog". We can find that the gradient-based methods cover a wider range, but do not pay attention to the complete head. Score-CAM focuses more on the head area, but does not provide a wider coverage. SA-CAM covers a wider range while focusing on the head area.

Figure 4 (e) shows the interpretation results of CAM methods for "haliaeetus leucocephalus". The performance of gradient-based methods is not as good as that of Score-CAM and SA-CAM. Intuitively speaking from the above five test results, SA-CAM has high robustness and accuracy in that it alleviates the noises incurred in other methods.

4.2. Multi-Model Testing

In Section 4.1, we showed CAM-based interpretations on ResNet101. Since the interpretation of CAM changes with different types of CNN models, we have further compared the aforementioned seven methods under different models to verify the robustness of SA-CAM which can be observed in Figure 5.

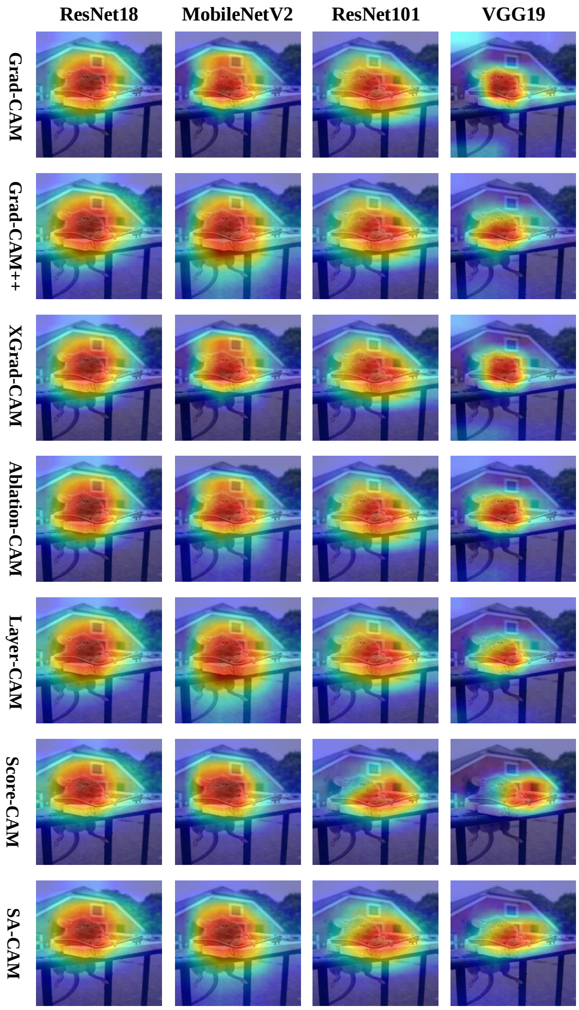


Figure 5: Demonstration of seven CAM-based methods on CNN models.

Figure 5 shows an image that contains two labels and the labels are related to each other. The correct label is "mouse trap". We fed this image into ResNet18, MobileNet, ResNet101 and VGG19 respectively and used seven methods for comparative analysis. We can find that the interpretations given by CAMs under different models are not the same. On ResNet18 and MobileNet, all the CAM methods did not produce the satisfactory interpretation in that they focus more on the "mouse" rather than the "mouse trap".

However, all the CAM methods on ResNet101 and VGG19 produced better results. On ResNet101 ,all the CAM methods recognized relationship between the mouse trap and the mouse. Among the best performers are SA-CAM and Score-CAM which payed attention directly to the mouse trap.

4.3. Multi-Target Positioning

Figure 6 shows an example in which an input image contains a cat and dog. It can be observed that the SA-CAM accurately locks the cat's head and neck areas when we set the corresponding prediction label to "Tiger Cat". When we set the corresponding prediction label to “Bull Mastiff”, SA-CAM accurately locks the dog's head and body parts.

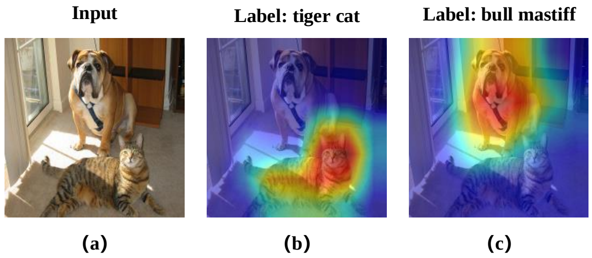


Figure 6: An example of multi-target positioning.

In addition to positioning test, we also tested the positioning effect of multiple targets of the same class. As shown in Figure 7, we show the positioning effect of multiple targets of the same class.

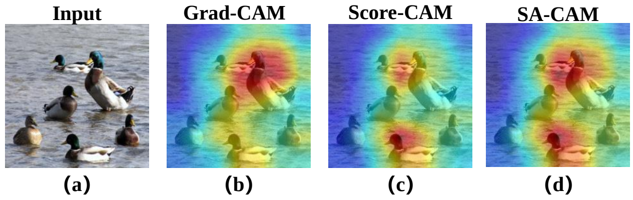


Figure 7: A comparison of Grad-CAM, Score-CAM and SA-CAM on multi-target positioning of the same class.

Figure 7(a) shows the image labeled "drake". In this image, there are multiple objects of the same category. As shown in Figure 7(b) to Figure 7(d), we can find that although both Grad-CAM and Score-CAM can lock a target for the same model, neither of these two methods can lock multiple identical targets whereas the SA-CAM does.

4.4. Sanity Check

The work in [1] emphasized that relying only on visual assessment can be misleading, certain interpretability methods generate explanations that can be independent of models and data. Following [1], we designed parameter randomization test to compare the up-sampling of SA-CAM on the pretrained VGG16 model with the up-sampling of the randomly initialized untrained network of the same architecture.

As shown in Figure 8, SA-CAM passed the sanity check like Grad-CAM and Score-CAM. The first column is the interpretation generated by the three methods. The following columns respectively show the results of the 21th - 28th layer randomization on VGG16. The results of SA-CAM are sensitive to model parameters reflecting the good quality of the model.

4.5. Deletion and Insertion

Following [14, 22], we performed deletion and insertion tests on the Grad-CAM, Score-CAM, and SA-CAM methods based on the ImageNet validation set. The purpose of

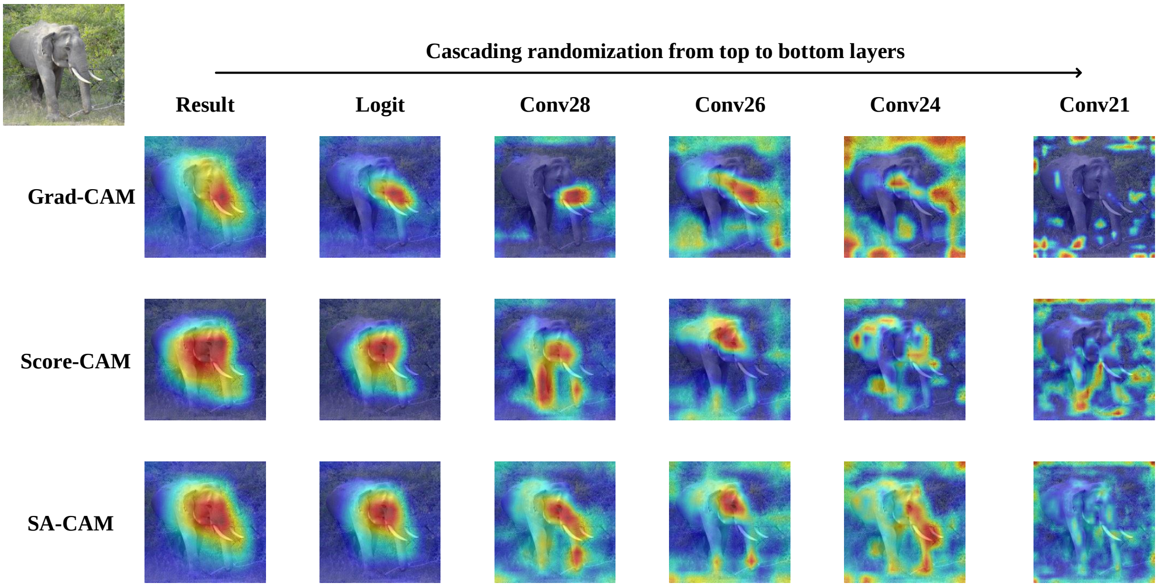


Figure 8: Sanity check results through randomization.

the tests was to test the corresponding accuracy of the three CAM methods when fading out and fading in the red mark part of an input image. Figure 9 shows two examples of deletion and insertion experiments.

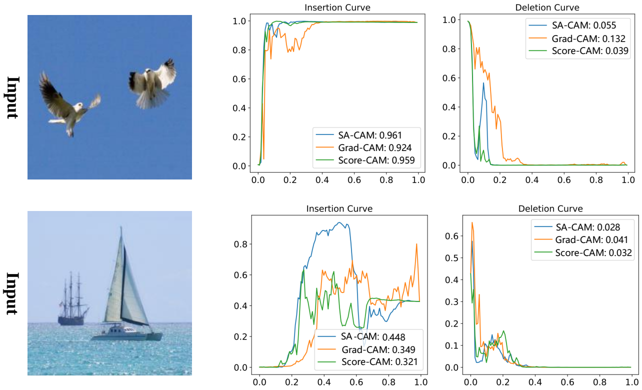


Figure 9: The fade-in and fade-out test curves of Grad-CAM, Score-CAM and SA-CAM.

Figure 9 shows the curves and AUC values generated by Grad-CAM, Score-CAM, and SA-CAM based on fade-in and fade-out tests. For the the deletion curve, a better performances on interpretation is that the AUC curves should be expected to fall fast and the AUC values should be small. However, for the insertion curve, the growth is expected to be fast and the AUC value should be large. The overall AUC value is the difference between the insertion AUC and the deletion AUC which reflects the quality of CAM methods.

For comparison fairly, we divided the images extracted from the ImageNet validation set into five groups. Table 1, Table 2 and Table 3 show the five sets of AUC scores for fade-in, fade-out and overall tests respectively.

Table 1
The AUC results of the insertion (fade-in) curves.

	Grad-CAM	Score-CAM	SA-CAM
Group-1	0.342	0.357	0.362
Group-2	0.43	0.467	0.462
Group-3	0.426	0.441	0.447
Group-4	0.429	0.442	0.445
Group-5	0.297	0.294	0.319
Total	1.924	2.001	2.035

Table 2
The AUC results of the deletion (fade-out) curves.

	Grad-CAM	Score-CAM	SA-CAM
Group-1	0.077	0.043	0.051
Group-2	0.101	0.056	0.064
Group-3	0.128	0.052	0.066
Group-4	0.096	0.057	0.058
Group-5	0.062	0.061	0.062
Total	0.464	0.269	0.301

From Table 1, it can be found that the AUC score of SA-CAM are more advantageous. From Table 2, we can observe that SA-CAM performs very closely to the Score-CAM. It is worth noting in Table 3 that SA-CAM achieves the best score of 1.734 when performing the deduction on the two sets of AUC scores.

It should be point out that the evaluations of fade-out and fade-in operations highly depend on data perturbation. As a result this evaluation can only be considered as an auxiliary

Table 3
The overall AUC results..

	Grad-CAM	Score-CAM	SA-CAM
Insertion AUC	1.924	2.001	**2.035**
Deletion AUC	0.464	**0.269**	0.301
Overall AUC	1.46	1.732	**1.734**

element because it is in favor of perturbation-based interpretation methods. In addition, this evaluations is too sensitive to the underlying CNN models which leads to inconsistent result as shown in Figure 10.

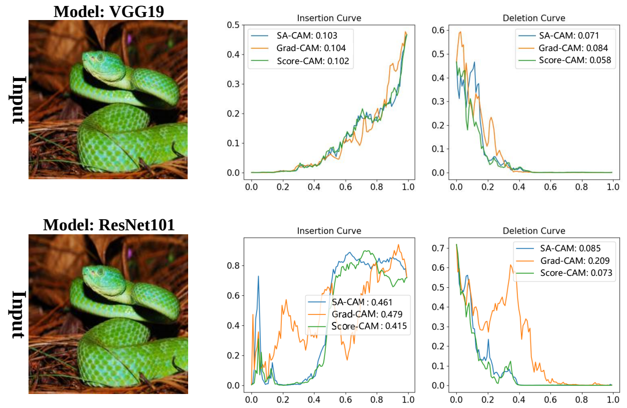


Figure 10: An example of inconsistent performance of Grad-CAM, Score-CAM and SA-CAM on ResNet101 and VGG19.

4.6. Adversarial Verification

To further evaluate the robustness of SA-CAM, we conducted adversarial tests on Grad-CAM, Score-CAM, and SA-CAM in multiple scenarios based on [17] and the adversarial examples in Figure 2. The changes in the overall environment of an image have an impact on the results of the underlying CNN model. In this section, we further verify the robustness of SA-CAM in adversarial verification. Table 3 compares the accuracy result of three adversarial images using VGG19.

Figure 11 shows the pipeline of adversarial verification. We first extracted the core interpretation area and randomly embed it into multiple adversarial images to test the accuracy. The evaluation index of adversarial detection is the comprehensive expectation of multiple adversarial images. We conducted adversarial verification on Grad-CAM, Score-CAM and SA-CAM based on the images in the ImageNet validation set. The experimental results of accuracy in Table 4 show the accuracy results that SA-CAM is more robust than both the Grad-CAM and Score-CAM methods.

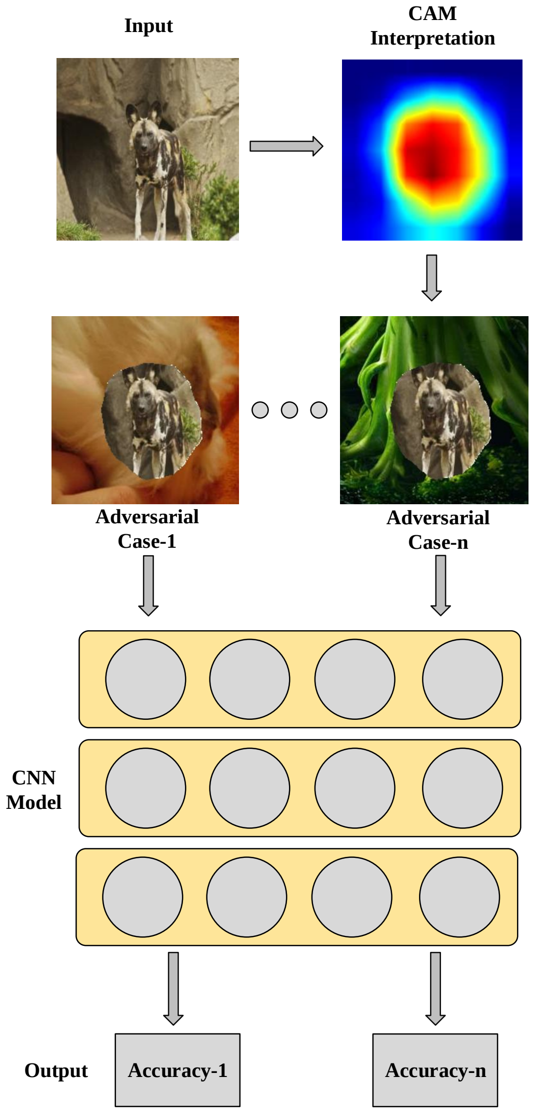


Figure 11: The pipeline of an adversarial verification.

Table 4
A comparison of adversarial results on accuracy (%).

	Grad-CAM	Score-CAM	SA-CAM
Case-1	18.40	22.31	**22.35**
Case-2	21.59	24.18	**29.26**
Case-3	29.73	28.11	**34.67**

We also discussed the fading rate of the three methods in adversarial verification to further prove the robustness of SA-CAM. The fading rate mainly describes the degradation degree of the original image after it is substituted into the adversarial environment. In the experiment, we use the three adversarial cases in Table 4 to test the fading rate. Faced with large amounts of data, The lower the fading rate, the better the performance. The experimental results of fading rate in Table 5 also show the accuracy results that SA-CAM is more robust than both the Grad-CAM and Score-CAM methods.

Table 5
A comparison of adversarial results on fading rate(%).

	Grad-CAM	Score-CAM	SA-CAM
Case-1	75.56	70.37	**70.31**
Case-2	71.32	67.88	**61.14**
Case-3	60.51	62.66	**53.95**

5. Conclusion

In this paper, we have presented SA-CAM, a novel CAM method building on the self-attention mechanism. Compared with the state-of-the-art results, SA-CAM generates more robust interpretations on CNN models and performs well in multi-target category positioning, fade in-out testing and adversarial testing.

SA-CAM can be used to interpret the decision making process of multiple computer vision tasks such as target detection and monocular depth estimation. It can also be used to enhance the interpretability of CNN models on medical imaging and autonomous driving, which has a very wide range of uses.

One immediate future work will be to deploy SA-CAM on weak-supervised localization improvement and to further introduce SA-CAM into a deep Bayesian neural network for enhancement of trustiness in machine learning.

Acknowledgement

This research is supported by 2018YFB2100801, the Director Foundation Project of National Engineering Laboratory for Public Safety Risk Perception and Control by Big Data (PSRPC), and the Fundamental Research Funds for the Central Universities.

References

[1] Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., Kim, B., 2018. Sanity checks for saliency maps, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, pp. 9525–9536.

[2] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O., 2020. nuscenes: A multimodal dataset for autonomous driving, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 11621–11631.

[3] Chattopadhay, A., Sarkar, A., Howlader, P., Balasubramanian, V.N., 2018. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks, in: 2018 IEEE Winter Conference on Applications of Computer Vision (WACV), IEEE. pp. 839–847.

[4] Dabkowski, P., Gal, Y., 2017. Real time image saliency for black box classifiers, in: Advances in Neural Information Processing Systems, pp. 6967–6976.

[5] Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L., 2009. Imagenet: A large-scale hierarchical image database, in: 2009 IEEE conference on computer vision and pattern recognition, Ieee. pp. 248–255.

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

[7] Fu, R., Hu, Q., Dong, X., Guo, Y., Gao, Y., Li, B., 2020. Axiom-based grad-cam: Towards accurate visualization and explanation of cnns. arXiv preprint arXiv:2008.02312 .

[8] Gao, X., Lin, S., Wong, T.Y., 2015. Automatic feature learning to grade nuclear cataracts based on deep learning. IEEE Transactions on Biomedical Engineering 62, 2693–2701.

[9] Gildenblat, J., contributors, 2021. Pytorch library for cam methods. https://github.com/jacobgil/pytorch-grad-cam.

[10] He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: Computer Vision and Pattern Recognition, pp. 770–778.

[11] Jiang, P.T., Zhang, C.B., Hou, Q., Cheng, M.M., Wei, Y., 2021. Layercam: Exploring hierarchical class activation maps for localization. IEEE Transactions on Image Processing 30, 5875–5888.

[12] Krizhevsky, A., Hinton, G., et al., 2009. Learning multiple layers of features from tiny images .

[13] Liang, Y., Li, S., Yan, C., Li, M., Jiang, C., 2021. Explaining the black-box model: A survey of local interpretation methods for deep neural networks. Neurocomputing 419, 168–182.

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

[15] Ramaswamy, H.G., et al., 2020. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization, in: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pp. 983–991.

[16] Ribeiro, M.T., Singh, S., Guestrin, C., 2016. 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, ACM. pp. 1135–1144.

[17] Ribeiro, M.T., Singh, S., Guestrin, C., 2018. Anchors: High-precision model-agnostic explanations, in: Thirty-Second AAAI Conference on Artificial Intelligence.

[18] Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., Chen, L.C., 2018. Mobilenetv2: Inverted residuals and linear bottlenecks, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4510–4520.

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

[20] Simonyan, K., Zisserman, A., 2014. Very deep convolutional networks for large-scale image recognition. Computer Science .

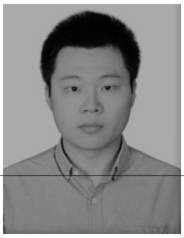
[21] Singla, S., Wallace, E., Feng, S., Feizi, S., 2019. Understanding impacts of high-order loss approximations and features in deep learning interpretation. arXiv preprint arXiv:1902.00407 .

[22] Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., Hu, X., 2020. 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.

[23] Zeiler, M.D., Fergus, R., 2014. Visualizing and understanding convolutional networks, in: European conference on computer vision, Springer. pp. 818–833.

[24] Zhang, Q., Rao, L., Yang, Y., 2021. Group-cam: Group score-weighted visual explanations for deep convolutional networks. arXiv preprint arXiv:2103.13859 .

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



Yu Liang received the Bachelor degree in Computer Science and Technology from Shandong Agricultural University, China in 2015. He received the Master degree in Software Engineering from Shandong University of Science and Technology, China in 2019. He is currently a PhD student in the Department of Computer Science and Technology, Tongji University, Shanghai, China. His research interests include lightweight deep neural networks (DNNs), interpretation methods for DNNs and graph DNNs.

Maozhen Li is a Professor in the Department of Electronic and Electrical Engineering, Brunel University London, UK. He received the PhD from the Institute of Software, Chinese Academy of Sciences in 1997. His main research interests include high performance computing, big data analytics and intelligent systems with applications to smart grid, smart manufacturing and smart cities. He has over 180 research publications in these areas including 4 books. He has served over 30 IEEE conferences and is on the editorial board of a number of journals. He is a Fellow of the British Computer Society (BCS) and the Institute of Engineering and Technology (IET).

Changjun Jiang is a Professor in the the Department of Computer Science and Technology, Tongji University, Shanghai, China. He received the Ph.D. degree from the Institute of Automation, Chinese Academy of Science, Beijing, China in 1995. He is currently the Director of the Key Laboratory of the Ministry of Education on Embedded System and Service Computing, Tongji University, Shanghai, China. His research interests include concurrency theory, Petri nets, formal verification of software, machine learning, intelligent transportation systems, and service oriented computing. He has published more than 300 papers in journals and conference proceedings in these areas. He has led over 30 research projects sponsored by the National Natural Science Foundation of China, the National High Technology Research and Development Program of China, and the National Basic Research Developing Program of China. He is a Fellow of the Chinese Association for Artificial Intelligence (CAAI) and also a Fellow of the Institute of Engineering and Technology (IET).