Neural Networks 178 (2024) 106473

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Full Length Article

Cluster-CAM: Cluster-weighted visual interpretation of CNNs' decision in image classification

Zhenpeng Feng [a], Hongbing Ji [a,*], Miloš Daković [c], Xiyang Cui [a], Mingzhe Zhu [a,b], Ljubiša Stanković [c]

[a] School of Electronic Engineering, Xidian University, Xi'an, China
[b] Kunshan Innovation Institute of Xidian University, School of Electronic Engineering, Xidian University, China
[c] Faculty of Electrical Engineering, University of Montenegro, Podgorica, Montenegro

ARTICLE INFO

Dataset link: https://www.image-net.org/challenges/LSVRC/

Keywords:
Explainable artificial intelligence
Class activation mapping
Clustering algorithm
Image classification

ABSTRACT

Despite the tremendous success of convolutional neural networks (CNNs) in computer vision, the mechanism of CNNs still lacks clear interpretation. Currently, class activation mapping (CAM), a famous visualization technique to interpret CNN's decision, has drawn increasing attention. Gradient-based CAMs are efficient, while the performance is heavily affected by gradient vanishing and exploding. In contrast, gradient-free CAMs can avoid computing gradients to produce more understandable results. However, they are quite time-consuming because hundreds of forward interference per image are required. In this paper, we proposed Cluster-CAM, an effective and efficient gradient-free CNN interpretation algorithm. Cluster-CAM can significantly reduce the times of forward propagation by splitting the feature maps into clusters. Furthermore, we propose an artful strategy to forge a cognition-base map and cognition-scissors from clustered feature maps. The final salience heatmap will be produced by merging the above cognition maps. Qualitative results conspicuously show that Cluster-CAM can produce heatmaps where the highlighted regions match the human's cognition more precisely than existing CAMs. The quantitative evaluation further demonstrates the superiority of Cluster-CAM in both effectiveness and efficiency.

1. Introduction

Convolutional neural networks (CNNs) have provided a basis for numerous remarkable achievements in various computer vision tasks, like image classification (He, Zhang, Ren, & Sun, 2016; Krizhevsky, Sutskever, & Hinton, 2012; Liu, Meng, Li, Mao, & Chen, 2022; Srinivas et al., 2021), object detection (Cao, Pang, Han, & Li, 2019; Liu, Zhang, Zhou, & Wang, 2023; Redmon, Divvala, Girshick, & Farhadi, 2016; Zhu et al., 2017), and semantic segmentation (Chen, Jin, Jin, Zhu, & Chen, 2022; Liang, Hu, Zhang, Lin, & Xing, 2018; Liu, Mao, et al., 2022). Despite CNNs' extraordinary performance, they still lack a clear interpretation of the inner mechanism (Lapuschkin et al., 2019; Saleem, Yuan, Kurugollu, Anjum, & Liu, 2022; Zhu et al., 2022). This lack of transparency can indeed be a disqualifying factor in some peculiar scenarios where mistakes can jeopardize human life and health, like medical image processing or autonomous vehicles (Ren, Li, Liu, & Zhang, 2021; Townsend, Chaton, & Monteiro, 2020; Vlahek & Mongus, 2021; Zhao, Xie, Wang, Liu, Shi, & Du, 2019). Therefore, it is highly desirable to understand and explain what exactly CNNs have learned during the training process (Macpherson et al., 2021; Spinelli, Scardapane, & Uncini, 2022; Tan, Gao, Khan, & Guan, 2022).

Recently, Class Activation Mapping (CAM), a visual interpretation technique, has drawn increasing attention (Sun, Song, Cai, Du, & Guizani, 2022; Tu, Zhou, Gan, Jiang, Hussain, & Luo, 2021). CAM aims at highlighting salience regions of an input image for CNN's decision by using a linearly weighted combination of feature maps. Vanilla CAM directly utilizes the weight of each feature map after global average pooling (GAP) corresponding to the target class, so it is only available for CNNs with GAP layers (Zhou, Khosla, Lapedriza, Oliva, & Torralba, 2016). To further extend CAM to more generic CNN structures, numerous modified CAMs are proposed and they can be broadly categorized as: (1) gradient-based CAMs, and (2) gradient-free CAMs. Gradient-based CAMs (e.g. Grad-CAM (Selvaraju et al., 2017), Grad-CAM++ (Chattopadhay, Sarkar, Howlader, & Balasubramanian, 2018), SmoothCAM++ (Omeiza, Speakman, Cintas, & Weldermariam, 2019), XGrad-CAM (Fu et al., 2020), Self-Matching CAM (Feng, Zhu, Stanković, & Ji, 2021), SC-SM CAM (Feng, Ji, Stanković, Fan, & Zhu, 2021)) define the weights by using the average partial gradient of

* Corresponding author.
E-mail address: hbji@xidian.edu.cn (H. Ji).

https://doi.org/10.1016/j.neunet.2024.106473
Received 19 January 2023; Received in revised form 13 June 2024; Accepted 16 June 2024
Available online 20 June 2024

CNN's prediction with respect to the feature maps. Gradient-based CAMs are usually computed efficiently. However, their weights lack reasonable explanation and are easily impeded by gradient exploding or vanishing. To address this limitation, some gradient-free CAMs are proposed. They define an intuitive impact of each feature map on the predicted score instead of using the gradient. Examples of gradient-free CAMs are Ablation CAM (Ramaswamy et al., 2020), Score-CAM (Wang et al., 2020), LIFT-CAM (Jung & Oh, 2021), and SHAP-CAM (Zheng, Wang, Zhou, & Lu, 2022). Gradient-free CAMs can provide a more explainable weight definition than gradient-based CAMs in most cases. Fig. 1 (from the second to the eighth columns) shows the salience heatmaps produced by several aforementioned CAMs.

Although gradient-free CAMs define the weights more reasonably, they are usually very time-consuming since hundreds of forward propagations are required per image. To improve the efficiency of gradient-free CAMs, Q. Zhang et al. proposed Group-CAM where feature maps are split into several groups (Zhang, Rao, & Yang, 2021). In this case, only several forward propagations are needed in computing the weights. Nonetheless, the feature maps are split without any regulation in the Group-CAM. In fact, feature maps have learned different semantic concepts relevant/irrelevant to the object. Therefore, those feature maps with similar semantics should be divided to the same group. Z. Feng et al. proposed SC-SM CAM by using spectral clustering to accomplish this goal, but this method is only available for synthetic aperture radar (SAR) images (Feng, Ji, et al., 2021), and no further selection of clustered features is mentioned in the SC-SM CAM (Feng, Ji, et al., 2021; Feng et al., 2021).

In this paper, we propose a Cluster-CAM, an effective and efficient gradient-free CAM, based on spectral clustering. In Cluster-CAM, a clustering technique, K-means/spectral clustering is adopted to split feature maps into several clusters. Subsequently, we provide an artful strategy to merge those feature maps into a cognition-base map and a cognition-scissors map. Finally, they will be merged as the salience heatmap. The highlights of this paper are as follows:

- We propose Cluster-CAM, as the first attempt to provide a cluster-weighted CAM framework via spectral clustering based on optical images.
- We provide a novel and artful weight-forming strategy to merge the cognition-base map and cognition-scissors map. These two maps greatly match the human's cognition and intuition, i.e., the weights of feature maps are reasonable and understandable.
- Cluster-CAM is effective and efficient, outperforming existing gradient-free CAMs in performance in most cases with significantly lower computing costs.

The rest of this paper is organized as follows. Section 2 introduces the basic knowledge of some existing CAMs as well as their advantages and disadvantages. Section 3 elaborates on how to generate salience heatmaps by Cluster-CAM. In Section 4, experiments are implemented to demonstrate the validity of Cluster-CAM and results are further analyzed from different perspectives. Section 5 concludes this paper.

2. Related work

As discussed in Section 1, the key issue in CAMs is defining a set of reasonable weights for feature maps. Based on definitions of these weights, CAMs are further categorized as: vanilla CAM, gradient-based CAMs and gradient-free CAMs. Gradient-based CAMs mainly include Grad-CAM (Selvaraju et al., 2017), Grad-CAM++ (Selvaraju et al., 2017), Smooth Grad-CAM++, Self-Matching CAM (Feng et al., 2021), and SC-SM CAM (Feng, Ji, et al., 2021). Gradient-free CAMs mainly include Ablation-CAM (Ramaswamy et al., 2020), Grad-CAM++ (Wang et al., 2020), SHAP-CAM (Zheng et al., 2022), and LIFT-CAM (Jung & Oh, 2021). In the following context, we will firstly discuss aforementioned 10 CAMs in Sections 2.1 and 2.2. Subsequently, we will analyze the advantages and disadvantages of each CAM, and provide a picture in Section 2.3 to show the most challenging problems that our proposed scheme tries to solve.

Vanilla CAM: Vanilla CAM is proposed by B. Zhou et al. in Zhou et al. (2016) to produce a saliency heatmap by a linerly weighted combination of feature maps in the last convolutional layer. Denote $\mathbf{F}_n$ as feature maps in the last convolutional layer followed with GAP layer. $F_n(i,j),\ n=1,2,\dots,N$, where N refers to the number of filters in this layer. The saliency heatmap, $M_c(i,j)$, is defined as:

$$M_c(i,j)=\sum_n \alpha_n^c F_n(i,j) \tag{1}$$

$$S_c=\sum_n \omega_n^c \sum_{i,j} F_n(i,j)=\sum_n \alpha_n^c \sum_{i,j} F_n(i,j), \tag{2}$$

where S_c denotes the predicted score for the target class c. The weight of each feature map, α^c, is determined by the last layer's fully-connected weights corresponding to the target class. Therefore, vanilla CAM is restricted to GAP-CNNs, i.e, the penultimate layer is constrainted to be a GAP layer. To extend CAM to generic CNN structures, gradient-based CAMs and gradient-free CAMs define the weights from different perspectives to replace α^c in .

2.1. Gradient-based class activation mapping

Grad-CAM: Selvaraju et al. proposed Grad-CAM (Selvaraju et al., 2017) to visualize any classification CNN architectures by weighting the feature maps with the gradients of the predicted score, $\mathbf{s}_c$ with respect to $\mathbf{F}_n$, as

$$\alpha_n^{c,Grad}=\sum_{i,j}\frac{\partial \mathbf{s}_c}{\partial F_n(i,j)}, \tag{3}$$

where different from (2), $\mathbf{s}_c$ is a sparse vector whose elements are zeros except the cth element which is equal to S_c. However, the highlighted regions generated by Grad-CAM are usually smaller than the object.

Grad-CAM++: A. Chattopadhyay et al. further proposed Grad-CAM++ (Chattopadhay et al., 2018) to produce more precise highlighted locality of the object. Grad-CAM++ assumes different elements in feature maps should have different contributions to CNN's prediction, thus an extra factor is introduced to realize this assumption by using higher order partial gradients as:

$$\boldsymbol{\alpha}_n^{c,Grad++}=\frac{\frac{\partial^2 \mathbf{s}_c}{\partial (F_n(i,j))^2}}{2\frac{\partial^2 \mathbf{s}_c}{\partial (F_n(i,j))^2}+\sum_{a,b}F_n(a,b)\frac{\partial^3 \mathbf{s}_c}{(F_n(i,j))^3}}\sum_{i,j}\frac{\partial \mathbf{s}_c}{\partial F_n(i,j)}. \tag{4}$$

However, the gradient, $\frac{\partial \mathbf{s}_c}{\partial \mathbf{F}_n}$, is sometimes heavily noised or even all-zero. It is probably because (1) CNN is trained to learn a generalized capability to classify a general concept rather than a specific object. (2) some unreasonable collapse emerged in CNN's training, like gradient vanishing and gradient exploding.

Smooth Grad-CAM++ To produce better visual interpretation of CNNs' decision on fine-grained images, D. Omeiza et al. proposed Smooth Grad-CAM++ (Omeiza et al., 2019) where the weights are defined using the average of the gradients as:

$$\alpha_n^{c,Smooth}=\frac{\frac{1}{m}\sum_{m=1}^{M}D_1^n\left(\frac{1}{m}\sum_{m=1}^{M}D_1^n\right)}{2\frac{1}{m}\sum_{m=1}^{M}D_2^n+\sum_{a,b}F_n(a,b)\frac{1}{m}\sum_{m=1}^{M}D_3^n} \tag{5}$$

where $D_1^n=\sum_{i,j}\frac{\partial \mathbf{s}_c}{\partial F_n(i,j)}$, $D_2^n=\sum_{i,j}\frac{\partial^2 \mathbf{s}_c}{\partial (F_n(i,j))^2}$, and $D_3^n=\sum_{i,j}\frac{\partial^3 \mathbf{s}_c}{\partial (F_n(i,j))^3}$ when the input is added with random noise for M times (M is a constant integer). This smoothing strategy is intuitive but still rough and not understandable enough for some complex CNN structures.

XGrad-CAM: To allay above drawbacks, R. Fu et al. proposed XGrad-CAM (Fu et al., 2020) by introducing two completely explainable axioms to form the weight:

$$\alpha_n^{c,XGrad}=\sum_{i,j}\frac{F_n(i,j)}{\sum_{i,j}F_n(i,j)}\frac{\partial \mathbf{s}_c}{\partial F_n(i,j)}. \tag{6}$$

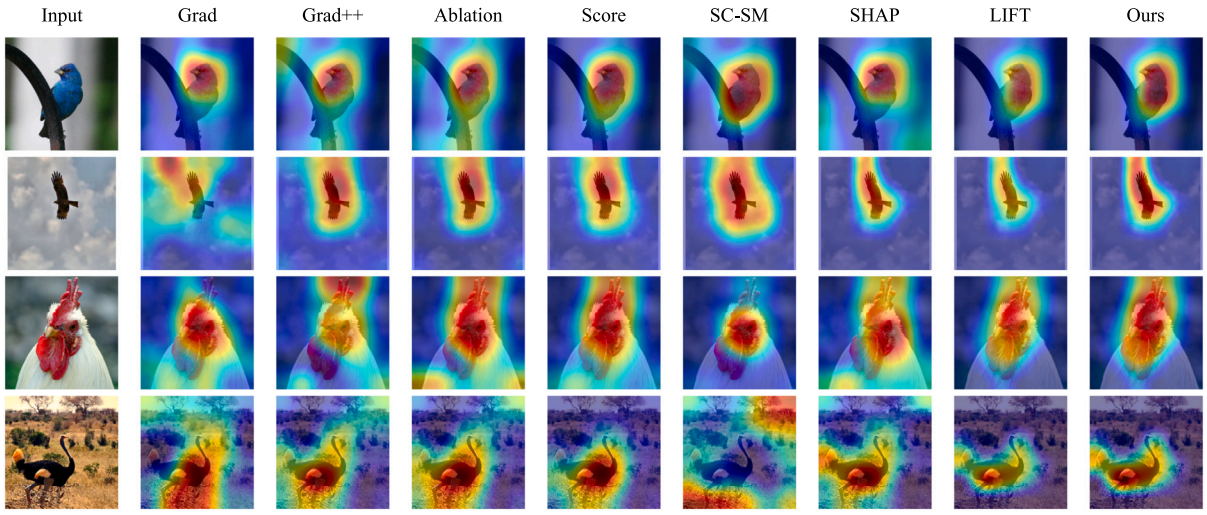

Fig. 1. Visual comparison of Cluster-CAM and other CAMs. The first column is input images (indigo finch, eagle, rooster, and ostrich from the first to the fourth row). The second to last columns are heatmaps produced by Grad-CAM, Grad-CAM++, Ablation-CAM, Score-CAM, SC-SM CAM, SHAP-CAM, LIFT-CAM, and Cluster-CAM.

It is worth noting that Smooth GradCAM++ and XGrad-CAM are variants of Grad-CAM++ with the same aim: to enhance the performance in fine-grained image classification rather than suppress the abnormal gradients. So we only choose Grad-CAM and Grad-CAM++ as comparative gradient-based CAMs in the following experiments.

Self-Matching CAM: It is imperative to acknowledge that the CAMs mentioned above may encounter diminished efficacy when employed on peculiar data, like SAR images. To address this issue, Z. Feng et al. propose Self-Matching CAM (Feng et al., 2021) based on SAR imaging mechanism. Self-Matching CAM augments the object-relevant information in feature maps by obtaining the Hadamard product of feature maps and the input SAR image. The modified feature maps in Self-Matching CAM algorithm are formulated as follows:

$$\mathbf{F}_n^{\mathrm{SM}} = \mathbf{F}_n \circ \mathbf{X} \tag{7}$$

where $\mathbf{X}$ is the input SAR image and $\circ$ denotes the element-wise multiplication. In the saliency maps produced by Self-Matching CAM, the highlighted regions tend to exhibit a higher consistency to the shape of SAR objects than other CAMs. Note this operation is not applicable for common optical images because some complex edges and textures of life/nature scenes probably introduce negative influences on feature maps.

SC-SM CAM: It is demonstrated that many redundant filters exist in CNNs (Zhang et al., 2021). Consequently, Z. Feng et al. further proposed SC-SM CAM (Feng, Ji, et al., 2021), a variant of Self-Matching CAM. SC-SM CAM is also designed for SAR images, but employs spectral clustering to categorize the feature maps, thereby reducing the number of feature maps and also facilitating a slight improvement in the saliency heatmaps. SC-SM CAM is similar to our proposed Cluster-CAM because both of them utilize spectral clustering to divide feature maps into groups, however, they are essentially different. The distinct differences between SC-SM CAM and Cluster-CAM are elaborated in detail in the ending of Section 3.2.

2.2. Gradient-free class activation mapping

Ablation-CAM: Ablation-CAM (Ramaswamy et al., 2020) formulates the weights of feature maps by referring to the direct change of CNN's prediction after a certain feature map is occluded. The weights of Ablation-CAM is defined as:

$$\alpha_n^{c,Ablation} = \frac{S_c - S_{c,n}}{S_c}, \tag{8}$$

where $S_{c,n}$ denotes the predicted score for class c when nth feature map is set to zero. In this case, a large weight will be assigned to the current feature map if removing it can lead to a sharp drop of the predicted score ($S_c - S_{c,n}$ is a large value) and vice versa. The authors argue that Ablation-CAM is immune to both saturation (marking a filter as important although it is not important) and explosion (marking a small influent filter as high importance.).

Score-CAM: Different from Ablation-CAM, Score-CAM considers measuring the impact of each feature map by introducing the input image, $\mathbf{X}$, as

$$\alpha_n^{c,Score} = S_c(\mathbf{X} \circ \mathbf{H}_n) - S_c(\mathbf{X}_b) \tag{9}$$

$$\mathbf{H}_n = s(\mathrm{U_p}(\mathbf{F}_n)), \tag{10}$$

where $\mathbf{X}_b$ is a baseline image which can be set the input image itself, $\mathrm{U_p}(\cdot)$ denotes the operation that upsamples $\mathbf{F}_n$ into the input size and $s(\cdot)$ is a normalization function that maps elements in the input into $[0, 1]$. $\mathbf{X} \circ \mathbf{H}_n$ can be deemed as filtering which only passes elements in $\mathbf{X}$ masked by $\mathbf{H}_n$, thus a large weight will be assigned if most target-discriminative parts are preserved by the current feature map, i.e. $S_c(\mathbf{X} \circ \mathbf{H}_n)$ is higher than $S_c(\mathbf{X}_b)$ and vice versa. Currently, gradient-free CAMs have drawn more attention than gradient-based CAMs due to their superior performance and explainable definition of weights. However, gradient-free CAMs are much more time-consuming than gradient-based CAMs because hundreds or even thousands of forward interference are required while gradient-based CAMs only require one forward interference.

LIFT-CAM: Different from Score-CAM, H. Jung et al. formulate a CNN explanation model as a linear function of binary variables representing activation map existence (Jung & Oh, 2021), allowing for an analytical determination using additive feature attribution methods. They further validate the use of SHAP values as weights for feature maps in CAM, and introduce an efficient approximation method, LIFT-CAM (Jung & Oh, 2021), based on DeepLIFT, which achieves accurate estimation of SHAP values for feature maps. Firstly, a binary vector is defined as $a \in \{0, 1\}^N$ corresponding to feature maps. Then, the weights of feature maps in LIFT-CAM are defined as:

$$\alpha_n^{c,\mathrm{LIFT}} = \sum_{a' \subset \mathbf{F}'_n} \frac{(N - |a'|)!(|a'| - 1|)!}{N!} [S_c(h_F(a')) - S_c(h_F(a' \backslash n))] \tag{11}$$

where h_F is a mapping function that converts a' into the embedding space of $\mathbf{F}_n$: it satisfies $\mathbf{F} = h_F(\mathbf{F}')$, where $\mathbf{F}'$ is a vector of ones. The above equation implies that $\alpha_n^{c,\mathrm{LIFT}}$ can be obtained by averaging marginal prediction differences between presence and absence of $\mathbf{F}_n$ across a set of all possible feature orderings of $\{1, 2, \ldots, N\}$.

Table 1
Advantages and disadvantages of existing CAMs. In this table, we conclude the most distinct advantages and disadvantages of existing CAMs to provide a clear comparison.

Vanilla CAM (Zhou et al., 2016) (only available for CNN with GAP layers)					
Gradient-based CAMs			Gradient-free CAMs		
Method	Advantages	Disadvantages	Method	Advantages	Disadvantages
Grad-CAM (Selvaraju et al., 2017)	GAP layer is not required	low robustness to abnormal gradients	Ablation-CAM (Ramaswamy et al., 2020)	avoid gradients and better localizing objects	low efficiency
Grad-CAM++ (Chattopadhay et al., 2018)	better localizing multiple objects	low robustness to abnormal gradients	Score-CAM (Wang et al., 2020)	avoid gradients and better localizing objects	low efficiency
XGrad-CAM (Fu et al., 2020)	analytican and understandable	only available for ReLU-CNN	LIFT-CAM (Jung & Oh, 2021)	involve the interaction between feature maps	extremely time-consuming
Self-Matching CAM (Feng et al., 2021)	perfectly matching SAR objects' edges	only available for SAR images	SHAP-CAM (Zheng et al., 2022)	involve the interaction among elements in feature maps	extremely time-consuming
SC-SM CAM (Feng, Ji, et al., 2021)	faster than Self-Matching CAM	semantic-chaos due to unreasonable clusters			

SHAP-CAM: LIFT-CAM only considers the marginal prediction differences with all elements are occluded or masked in a feature map, without considering the interaction between these elements. To mitigate this issue, Q. Zheng et al. proposed SHAP-CAM (Zheng et al., 2022) by introducing Shapley value (i.e., marginal contributions to classification between feature map elements) to construct weights. Assume $\mathbb{P} = \{(i,j)|i = 1,\dots,h;\ j = 1,\dots,w\}$ be the set of elements in nth feature map, $\mathbf{F}_n$, where w and h denote the height and width of $\mathbf{F}_n$, i.e., $w \times h = z$ elements in total. The weights of Shap-CAM is defined as:

$$\alpha_n^{c,\mathrm{SHAP}} = \sum_{\mathbb{S}\subseteq\mathbb{P},i,j\notin\mathbb{S}} \frac{(z-s-1)!s!}{z!}[S_c(\mathbb{S}\cup\{(i,j)\}) - S_c(\mathbb{S})] \tag{12}$$

where the subset $\mathbb{S} \subseteq \mathbb{P}$, $S_c(\mathbb{S})$ represents the output probability of class c when only the elements in the set $\mathbb{S}$ remain and the others are set to the average value of the entire feature map. It is essential to highlight that both LIFT-CAM and SHAP-CAM pose significant computational challenges because exact calculating Shapley values demands $O(2^n)$ complexity which is infeasible for real problems. Although some sampling techniques are used to reduce computation complexity to $O(mn)$ (where m is the number of samples in a permutation (Zheng et al., 2022)), millions of forward propagation are still required per image.

2.3. Analysis of existing CAMs

To provide a clear comparison of existing CAMs, Table 1 briefly concludes the most distinct advantages and disadvantages of them. Evidently, gradient-based CAMs demonstrate efficiency but exhibit vulnerability to abnormal gradients, whereas gradient-free CAMs, albeit avoiding gradient-related issues, are confronted with heavy computation burdens. Consequently, we endeavor to strike a judicious balance between efficiency and efficacy. Note that the reason why gradient-free CAMs are inefficient is that forward propagation is required for each feature map and usually there are hundreds or even thousands of feature maps in a CNN. We are further inspired by SC-SC CAM (Feng, Ji, et al., 2021) where feature maps can be categorized into groups by spectral clustering to reduce the number of forward propagations. However, we notice that there are some semantic-agnostic clustered feature maps (as shown in Fig. 2) which could lead to semantic-chaos in final saliency heatmap (e.g., the heatmap for ostrich produced by SC-SM CAM in Fig. 1, the examples in Figs. 4 and 6). Fig. 2 visually explains the above phenomenon which is the main challenge that our scheme tries to solve.

3. Methodology

In this section, we will first introduce some basic concepts on graph-based spectral clustering and K-means. Then we present the detailed procedures of Cluster-CAM.

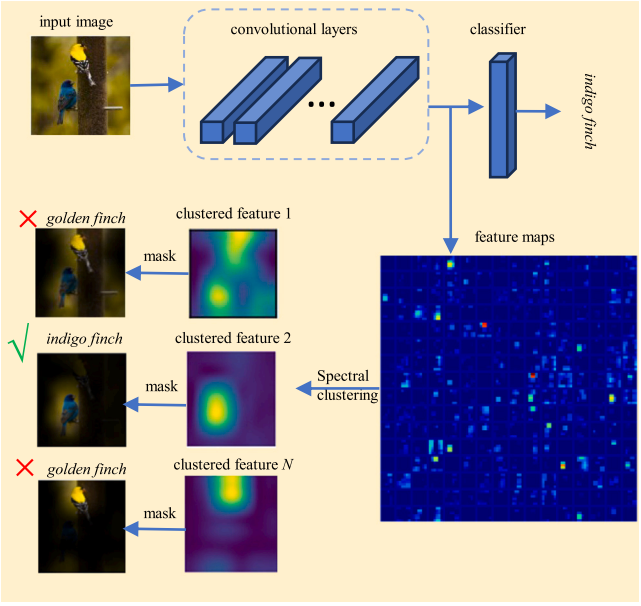

Fig. 2. The visual explanation of semantic-agnostic feature maps produced by spectral clustering. The input image includes two objects: indigo finch and golden finch. Spectral clustering is utilized to divide feature maps into several groups (only three clusters are shown). It is obvious that the first and the third clustered feature maps are semantic-agonistic to the current label. This issue cannot be solved in Feng, Ji, et al. (2021) and Zhang et al. (2021), which our proposed scheme tries to solve in this paper.

3.1. Spectral clustering and K-means

Spectral clustering is a widely-used unsupervised clustering algorithm based on graph signal processing (Ma, Zhang, Pena-Pena, & Arce, 2021; Scalzo, Stanković, Daković, Constantinides, & Mandic, 2023; Stankovic, Dakovic, & Sejdic, 2017; Stankovic et al., 2019). Specifically, the processed data (feature maps, $\mathbf{F}_n$) are regarded as vertices in a graph topology. Then the elements, $S(i,j)$, of the similarity matrix, $\mathbf{S}$, can be defined:

$$S(i,j) = \mathrm{similarity}(\mathbf{F}_i, \mathbf{F}_j), \tag{13}$$

where $\mathrm{similarity}(\cdot)$ refers to a function that measures the similarity between two vertices (feature maps). If we use the structural similarity index (SSIM), then it ranges from 0 (no similarity) to 1 (identical feature maps). The elements of the weighted adjacency matrix, $\mathbf{A}$, can be defined as:

$$\begin{cases} A(i,j) = \exp(-(1-S(i,j))/\sigma), & \text{if } S(i,j) > \theta, \\ A(i,j) = 0, & \text{else,} \end{cases} \tag{14}$$

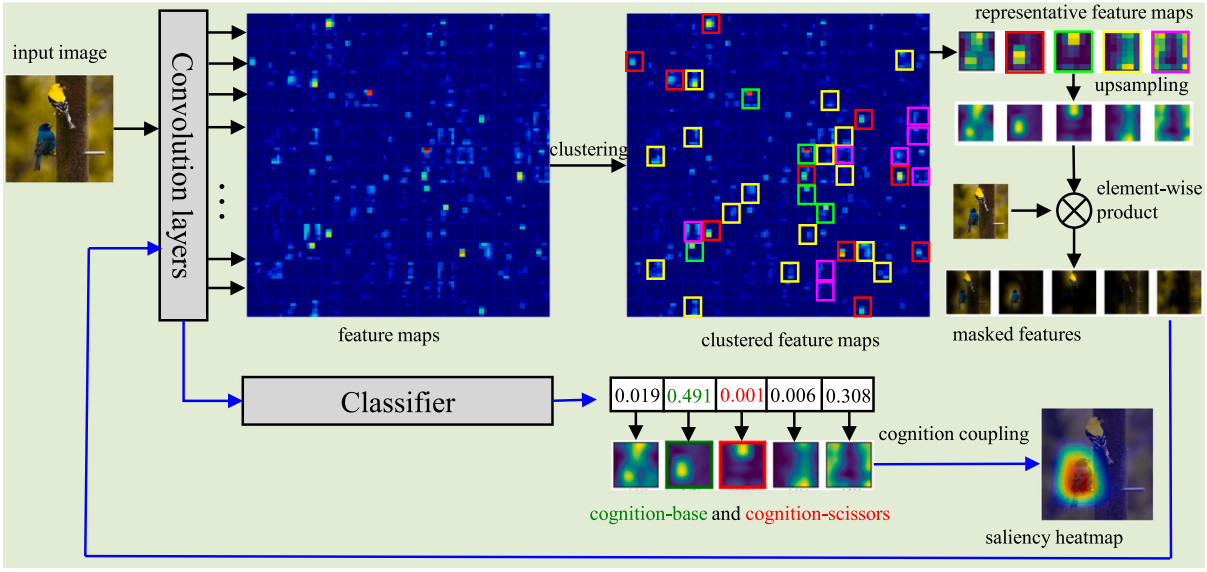

Fig. 3. The overview of Cluster-CAM.

where θ is a threshold to keep the direct edge in the corresponding graph for two neighboring vertices and σ is an adjusting parameter. Note that, by definition of similarity, this adjacency matrix is a symmetric matrix resulting in an undirected graph, that is, $A(i,j) = A(j,i)$.

The similarity can be defined using the difference between two vertices (feature maps), $d(i,j) = \|\mathbf{F}_i - \mathbf{F}_i\|$. Then the weighted adjacency matrix is defined by

$$\begin{cases} A(i,j) = \exp(-d^2(i,j)/\sigma^2), & \text{if } S(i,j) > \theta, \\ A(i,j) = 0, & \text{else,} \end{cases} \tag{15}$$

where θ and σ have the same role as in (14).

In order to produce the vectors for spectral clustering, now we compute the graph Laplacian matrix, $\mathbf{L}$, as

$$\mathbf{L} = \mathbf{D} - \mathbf{A} \tag{16}$$

where $D(i,i) = \sum_j A(i,j)$ are the elements of the degree matrix $\mathbf{D}$ which is diagonal.

In practice, the graph Laplacian matrix usually can be normalized, as

$$\mathbf{L}_N = \mathbf{D}^{-\frac{1}{2}}\mathbf{L}\mathbf{D}^{-\frac{1}{2}} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}. \tag{17}$$

The clustering results obtained using these two matrices are very similar.

The eigendecomposition of the graph Laplacian

$$\mathbf{L} = \mathbf{U}^T \boldsymbol{\Lambda} \mathbf{U}. \tag{18}$$

Results in eigenvectors $\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_N$ that are the columns of matrix $\mathbf{U}$. The smoothness index of these vectors is equal to the corresponding eigenvalue λ_i. When clustering the data into two clusters, only the eigenvector $\mathbf{u}_2$ is used (Fiedler vector) since the vector $\mathbf{u}_1$ is omitted as its elements are constant.

If we want to get a few clusters (Q clusters) then we can use the smoothest K eigenvectors, $\mathbf{u}_2, \mathbf{u}_3 \cdots, \mathbf{u}_{K+1}$, written in the matrix form as

$$\mathbf{B} = \begin{bmatrix} \mathbf{u}_2 & \mathbf{u}_3 & \cdots & \mathbf{u}_{K+1} \end{bmatrix} = \begin{bmatrix} u_{12} & u_{13} & \cdots & u_{1(K+1)} \\ u_{22} & u_{23} & \cdots & u_{2(K+1)} \\ \vdots & \vdots & \ddots & \vdots \\ u_{N2} & u_{N3} & \cdots & u_{N(K+1)} \end{bmatrix}, \tag{19}$$

where N features with K dimension are considered.

The clusters are determined based on the K-dimensional spectral similarity vectors, $\mathbf{q}_1 = [u_{12}, u_{13}, \ldots, u_{1(K+1)}]$, $\mathbf{q}_2 = [u_{22}, u_{23}, \ldots, u_{2(K+1)}]$, $\ldots$, $\mathbf{q}_N = [u_{N2}, u_{N3}, \ldots, u_{N(K+1)}]$, defined for vertices (features $\mathbf{F}_1$, $\mathbf{F}_2$, $\ldots$, $\mathbf{F}_N$).

In this way, the dimension of the measuring distance is significantly reduced from the original N dimensional space in $d(i,j) = \|\mathbf{F}_i - \mathbf{F}_i\|$ to a very low K-dimensional spaces of spectral vectors $\mathbf{q}_n$.

Finally, the clustering result (the data grouped into Q clusters) can be refined using K-means and the Euclidean distance $d(i,j) = \|\mathbf{F}_i - \mathbf{F}_i\|$.

Note that the traditional K-means algorithm can be used with an initial random clustering of feature maps into Q clusters, with a slower convergence due to random initialization. In this case, all the feature maps are grouped into Q initial clusters, $\mathbb{Q}_q$, $q = 1, 2, \ldots, Q$. Means of the feature maps are calculated for each cluster, $M_q = \text{mean}(\mathbf{F}_n, n \in \mathbb{Q}_q)$. The distance of each feature map is checked with respect to each of the mean M_q. The feature map is reassigned to the cluster whose mean is the closest to the considered feature map. After all feature maps are considered, the means are recalculated for the new clusters. The procedure is repeated until no cluster changes its feature maps.

3.2. Cluster-CAM

Now we are ready to introduce spectral clustering and K-means in Cluster-CAM. Here the feature maps, $\mathbf{F}$, represent the vertices in (13). Take Euclidean distance as similarity measurement, (13) can be expressed as:

$$S(i,j) = \exp\{-\|\mathbf{F}_i - \mathbf{F}_j\|\}, \tag{20}$$

where a shorter distance means a higher similarity. By substituting (20) into (14), (16), (17), and (19), we can split N feature maps into Q clusters, $\mathbb{Q}_q$, $q = 1, 2, \ldots, Q$, $Q \ll N$. Then we can obtain the Q representative feature maps, $\tilde{\mathbf{F}} = [\tilde{\mathbf{F}}_1, \tilde{\mathbf{F}}_2, \ldots, \tilde{\mathbf{F}}_Q]$, by calculating the mean of feature maps in each cluster, as

$$\tilde{\mathbf{F}}_q = \text{mean}\{\mathbf{F}_n,\ n \in \mathbb{Q}_q\}, \quad q = 1, \ldots, Q. \tag{21}$$

Next we obtain the Hadamard product of $\tilde{\mathbf{F}}$ and $\mathbf{X}$ ($\tilde{\mathbf{F}}$ will be upsampled to the same size of $\mathbf{X}$). This processing can be deemed as filtering that mainly passes those elements corresponding to large values in $\tilde{\mathbf{F}}$. The predicted score of each masked image is computed as:

$$\begin{aligned} \mathbf{y} &= [y_1, y_2, \ldots, y_Q]^T \\ &= [S_{c,1}(\tilde{\mathbf{F}}_1 \circ \mathbf{X}), \ldots, S_{c,Q}(\tilde{\mathbf{F}}_Q \circ \mathbf{X})]. \end{aligned} \tag{22}$$

In this case, we can obtain the cognition-base map and cognition scissors as

$$\tilde{\mathbf{F}}_{\text{base}} = \mathbf{F}_{q_{\max}}, \quad q_{\max} = \arg\max_{q}(\mathbf{y}) \tag{23}$$

$$\tilde{\mathbf{F}}_{\text{scissors}} = \mathbf{F}_{q_{\min}}, \quad q_{\min} = \arg\min_{q}(\mathbf{y}). \tag{24}$$

Next, we can semantically couple the cognition-base map and cognition-scissors to form the salience heatmap, as:

$$\mathbf{H}^{Cluster} = \beta\tilde{\mathbf{F}}_{\text{base}} - (1-\beta)\tilde{\mathbf{F}}_{\text{scissors}}, \tag{25}$$

where $\beta \in [0, 1]$ is a balance factor to adjust the importance of cognition-base map and cognition-scissors. Fig. 3 provides the overview of Cluster-CAM. Note that though spectral-clustering is an unsupervised clustering method, Cluster CAM is essentially not unsupervised because the feature maps are derived from a pre-trained model, implying that there is a supervised component involved in the initial stages of the Cluster-CAM method.

It should be clarified clearly that Cluster-CAM is inspired by our previous work, SC-SM CAM (Feng, Ji, et al., 2021), however, there exist significant differences between SC-SM CAM and Cluster-CAM, which can be summarized as:

- **Different Application Scenarios**: The SC-SM CAM is proposed to locate the object (ground vehicles in MSTAR dataset) in SAR images, i.e., SC-SM CAM is only available for CNNs trained on SAR image datasets. In comparison, Cluster-CAM is dedicated to provide a visual interpretation of CNNs trained on normal optical images.
- **Different Feature Processing**: The SC-SM CAM can directly generate the final heatmaps by computing the Hadamard product of weighted feature maps and the input SAR image, termed "Self-Matching" due to three peculiar characteristics of SAR images: (1) the background is simpler than that in optical images (2) the objects are centrally cropped. (3) they are grey-scale images with a single channel. In contrast, the Cluster-CAM does not contain this step.
- **Different Feature Fusion**: The SC-SM CAM does not consider the semantic-chaos in clustered feature maps since it seldom occurs in SAR images while it is common for optical images, thus Cluster-CAM proposes semantic-scissors to restrain the unreasonable feature maps after spectral clustering.

4. Experiments

In this section, we will present and analyze the performance of Cluster-CAM from various perspectives. Firstly we will briefly describe the dataset used in our experiments. Then we verify the superiority of Cluster-CAM to other existing CAMs in terms of discrimination, localization, explanation and multiple ablation study. As discussed in Section 2.2, the computation complexity of LIFT-CAM and SHAP-CAM still remains $O(mn)$ after sampling some permutations of the subset orders, $\mathbb{S}$, (m is the number of samples in a permutation). Taking ResNet-18 as an example, there are 512 feature maps in shape of 7×7 in the last convolutional layer. Even though we only sample 40 potential subsets in each permutation ($m = 40$), at least $512\times7\times7\times40 = 1,003,520$ times of forward propagation is needed to produce a saliency heatmap per image. It means approximately 10 minutes to produce one saliency heatmap in our device. Therefore, it is impractical to compute the numerical metrics through the whole ILSVRC validation set, whose time is $50,000 \times 10 = 500,000$ minutes $\approx$ 347 days. Nevertheless, it is necessary to provide an intuitive comparison of Cluster-CAM and SHAP-CAM & LIFT-CAM, so we produce the saliency heatmaps of several images in Fig. 1.

4.1. Experimental setup

Dataset: In the following experiments, CNNs are trained on a benchmark dataset, i.e., ILSVRC (Deng, Dong, Socher, Li, Li, & Fei-Fei, 2009). In ILSVRC, there are around 1.2 million images with 1000 categories for training, and 50 thousand images with 1000 categories for validation. 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 $[0.4850.456, 0.406]$ and standard deviation vector $[0.229, 0.224, 0.225]$. No further pre-processing is performed.

Network Structure: In this paper, VGG-16 (Simonyan & Zisserman, 2015), ResNet-18 (He et al., 2016), and InceptionV3 (Szegedy, Vanhoucke, Ioffe, Shlens, & Wojna, 2016), are used as classification CNNs. VGG-16 is a very deep CNN with approximately 134M trainable parameters. ResNet-18, a relatively lightweighted architecture with 25.6M parameters. InceptionV3 has about 23.8M parameters. Cluster-CAM is dedicated to provide an intuitive and understandable visual interpretation of generic CNN's mechanism rather than interpreting artful tricks of a state-of-the-art (SOTA) network. The three mentioned classic models represent fundamental architectures in the development of CNNs in the visual domain. Therefore, referring to the configuration in Wang et al. (2020), while these CNNs are not currently SOTA, they are strategically chosen as backbone networks to assess the performance of Cluster-CAM in interpreting CNNs.

Implementation Details: Note a consistent configuration of experiments is necessary for a fair and convicing comparison. In the following experiments, unless stated otherwise, the configuration of each CAM is the same as the experimental setup in Wang et al. (2020). We use the pre-trained VGG16, ResNet-18, and InceptionV3 from the Pytorch model zoo as base models in torchvision. All experiments are implemented in Pytorch 1.8.0+cudnn11.1, NVIDIA RTX-3070. The processor is AMD Ryzen 7 5800H with Radeon Graphics (the clock speed is 3.20 GHz).

4.2. Performance of class discrimination

Fig. 1 shows the salience heatmaps of input images with different objects (indigo finch, eagle, rooster, and ostrich) by Grad-CAM, Grad-CAM++, Ablation-CAM, Score-CAM, SC-SM CAM, SHAP-CAM, LIFT-CAM, and Cluster CAM. Visually, in comparison to existing CAMs, the highlighted region produced by Cluster-CAM mostly matches human's intuitive understanding of the discriminative part of the object. Take the indigo finch as an example, Grad-CAM only highlights the head of the bird, whereas Grad-CAM++ and Ablation-CAM also highlight the branch (irrelevant to the label). Score-CAM, SHAP-CAM, LIFT-CAM, and Cluster-CAM all highlight the finch body without the branch. But obviously, the region produced by Cluster-CAM matches the edges of the finch more precisely than Score-CAM. Though SHAP-CAM and LIFT-CAM can produce highlighted regions similar in shape to the object as Cluster-CAM, our method can achieve most concentrated salient regions (dark red elements) inside the object's profile. Note that the highlighted regions generated by SC SM CAM usually exceed the profile of the object or sometimes exhibit semantic-chaos (e.g., ostrich in Fig. 1). This issue arises due to the lack of channel selection of clustered feature maps, which motivates Cluster-CAM to introduce cognition-scissors to restrain those unreasonable feature maps.

4.3. CNN's cognitive explanation

Cognition Analysis of Multi-objects Images: Images of multiple objects are optimal samples to verify the rationality of the cognition-base map and semantic-scissors in (23) and (24), respectively. As discussed in Section 3, a reasonable cognition-base map should incorporate the object-relevant information as much as possible, while the corresponding cognition-scissors should include such information as less as better. Therefore, it is necessary to check whether the cognition-base map

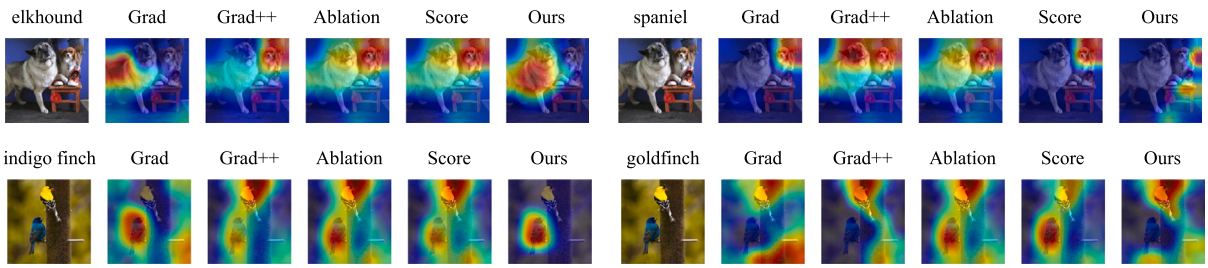

Fig. 4. Visual comparison of Cluster-CAM and other CAMs for mutiple-object images. The heatmaps produced by different CAMs according to the label elkhound (top-left subfigure). The results with the label spaniel are organized in the same structure (top-right subfigure). The indigo finch and goldfinch are shown in the bottom-left and bottom-right subfigures, respectively.

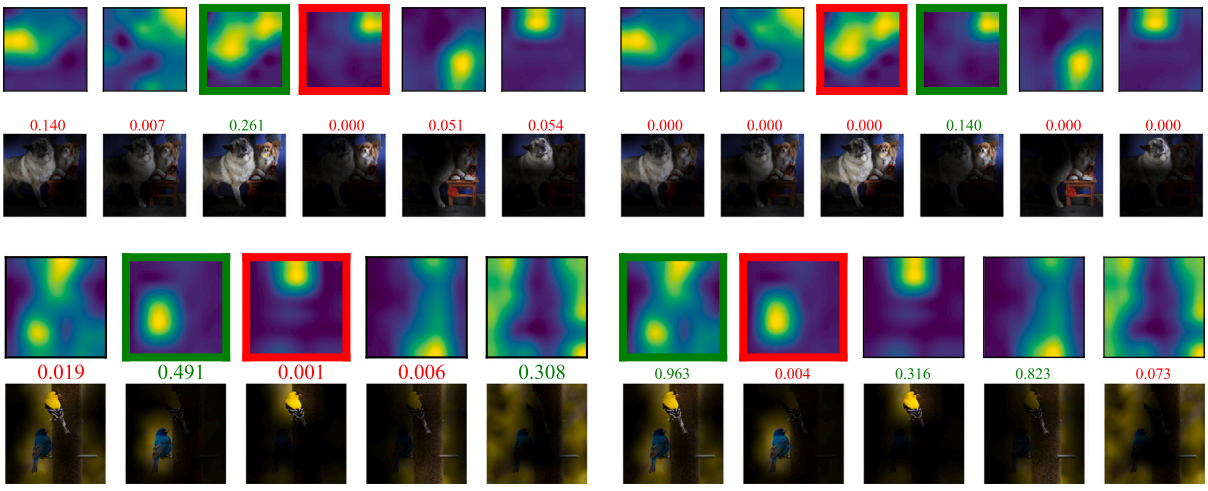

Fig. 5. The analysis of feature maps for images of multiple objects. The clustered feature maps (the first row and the third row) as well as corresponding masked images (the second row and the fourth row). Note that the cognition-base map and cognition-scissors are marked by green and red squares, respectively. The predicted score for the current class is provided above each masked image.

and cognition-scissors can interchange in a multi-objects image if the target label is changed to each other. Fig. 4 shows the visual comparison of Cluster-CAM and other CAMs for two multiple-object images. There are two kinds of dogs in the first image, i.e. elkhound (the big gray dog on the left) and spaniel (the tiny brown dog on the right). The second image includes two finches with different colors, i.e., indigo finch and golden finch. It is evident that Cluster-CAM shows most sensitivity to the change of target labels. Fig. 5 shows the cognition-base map and cognition-scissors of two muti-objects images as well as the corresponding masked images. When the target class is elkhound, the third and the fourth clustered feature map in each subfigure are cognition-base map and cognition scissors, respectively (marked by green and red squares). It matches human's cognition because cognition-base map incorporates both objects while cognition-scissors only selects the spaniel, thus the highlighted region will only be concentrated on the elkhound, as shown in the top-left subfigure in Fig. 4. When the target class is changed to the spaniel, the cognition-base map and cognition-scissors are also interchanged, as shown in the third column in the top-left subfigure in Fig. 5. The same phenomenon also emerges in indigo finch and goldfinch, as shown in the bottom subfigures in Fig. 5. In this case, Cluster-CAM provides solid evidence that CNN's recognition mechanism is similar to human cognition in multiple objects classification.

Cognition Analysis of Fine-grained Images: To further understand how CNN utilizes the learned information to make decisions, we can use CAM to interpret CNNs in fine-grained image classification. Fine-grained classification aims to distinguish subordinate categories within general categories. Examples include recognizing species of birds such as northern cardinal or indigo bunting; monkeys such as guenon or langur. Fine-grained classification often requires much more detailed information compared with generic object recognition, like the texture of the skin, the thickness of the fur, etc., so CAMs on fine-grained images can tell whether the information is reasonably learned by CNN for classification. Fig. 6 (the first row) shows the heatmaps generated by several mentioned CAMs given the input image of a guenon. Interestingly, they focus on completely different parts of the guenon. Grad-CAM and Grad-CAM++ highlight the guenon's eyes and cheek, respectively. Ablation CAM and Score CAM both highlight the guenon's face, whereas Cluster-CAM only highlights the guenon's forehead. Intuitively, Ablation CAM and Score CAM seem the most reasonable but the cognition-base map and cognition-scissors clearly show that the forehead is the most discriminative part, while the face is negative for guenon's classification. It will be understood if we further study the difference in species between guenon and langur. Guenon is characterized by blond hair on the forehead and a busty white lip, in contrast, langur is characterized by a completely black face. We mark their characteristics by green and red circles in the third row in Fig. 6. It is the reason why the third feature map (face) is deemed as cognition-scissors, i.e., the black face is an interference factor for guenon's categorizing. This example perfectly demonstrates the rationality of Cluster-CAM, especially the validity of cognition-scissors.

4.4. Ablation study

Analysis of Different Layers: Most CNNs are constructed by a cascade of convolutional blocks (a block consists of convolutional layers, nonlinear activation, pooling operation, etc.). Fig. 7 shows the salience heatmaps of different convolutional blocks in VGG-16. The results basically match human's intuition that the shallow layers mainly capture

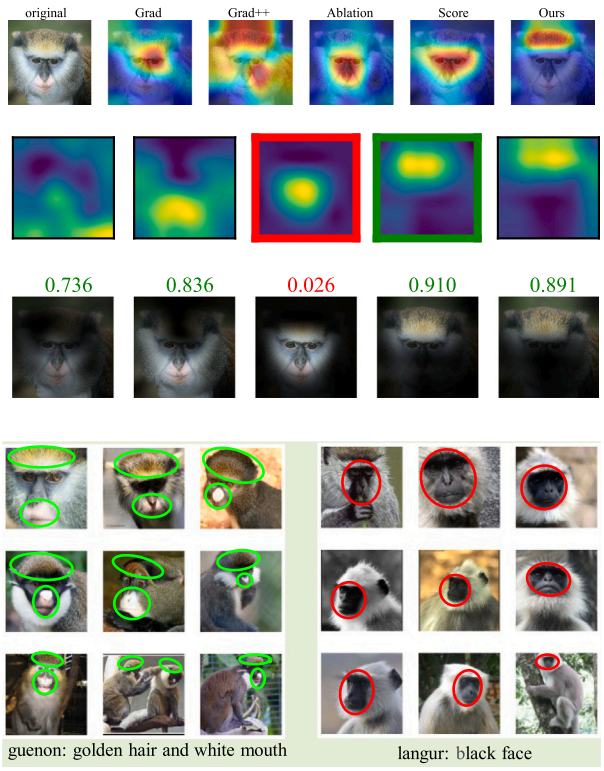

Fig. 6. The cognition-base map (green square) and cognition-scissors (red square) in merged feature maps (top). The images are masked by corresponding feature masks as well as the predicted score (middle). Nine images of guenon and nine images of langur (bottom). Note that the discriminative characteristics of guenon (golden hair and white mouth) and langur (black face) are labeled with green and red circles, respectively.

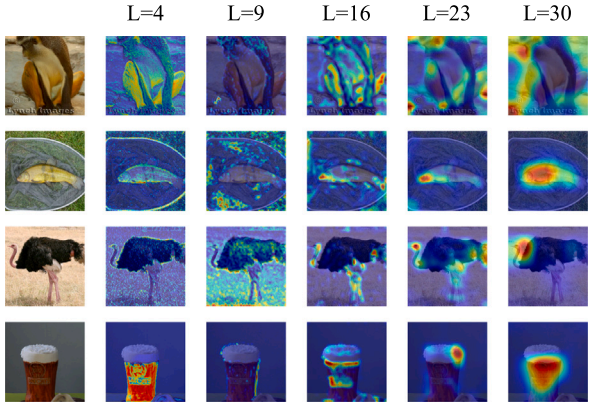

Fig. 7. The heatmaps produced by Cluster-CAM with different layers of VGG-16. *L* refers to the indices of layers in VGG-16.

some detailed information (e.g., texture and edge), whereas deep layers concentrate on those parts with clearer semantics.

Number of Clusters and CNN Structures: The number of clusters usually plays a critical role in clustering algorithms. Here we vary the cluster number from 2 to 8 and present the corresponding salience heatmaps for AlexNet and VGG-16 in Fig. 8. It shows the salience heatmaps are sensitive to the number of clusters and are different with CNN models. For AlexNet, the highlighted region in heatmaps could be semantic chaos if the feature maps are split into too many or too few clusters. It is probably because a large number of clusters may introduce too many detailed patterns of the object and a small number of clusters may directly include background information. Therefore, the number of clusters should be selected as a median value. Note it is only an

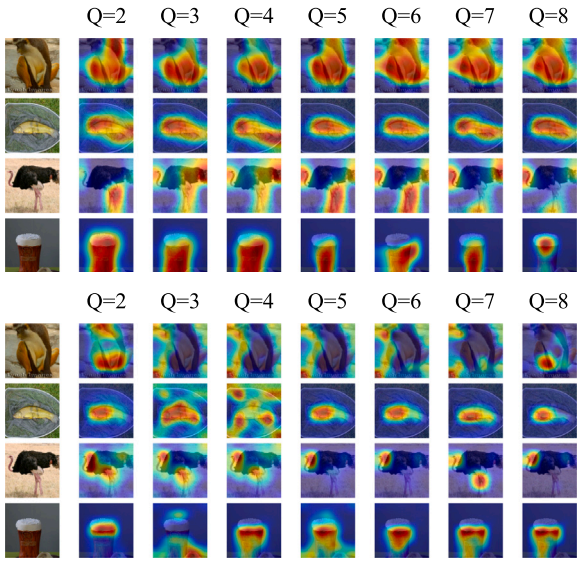

Fig. 8. The heatmaps produced by Cluster-CAM with different numbers of clusters for AlexNet (top) and VGG-16(bottom).

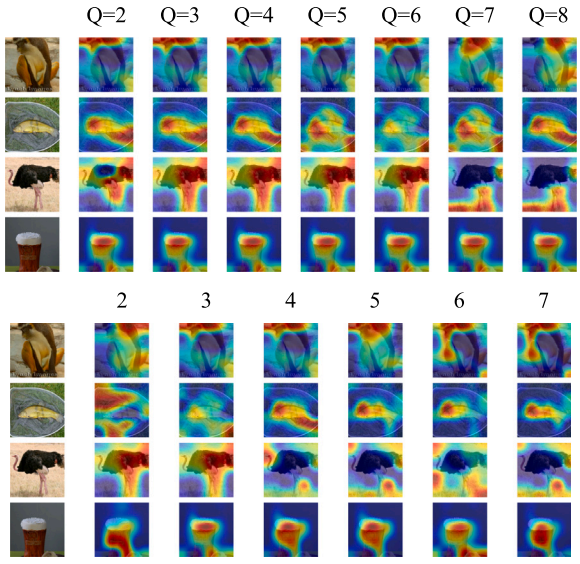

Fig. 9. The heatmaps produced by Cluster-CAM with different numbers of clusters by spectral clustering (top) and heatmaps produced by Cluster-CAM with different numbers of eigenvectors (bottom).

empirical conclusion and exclusion exists that the optimal value is 2 for the fourth row (beer) in Fig. 8. It is probably because the object is simple and in regular shape, thus only two clusters are enough to represent all necessary information. For VGG-16, the highlighted region is more concentrated on a specific part of the object than AlexNet. It is probably because more detailed discriminative information could be captured in VGG-16 with much deeper layers than AlexNet.

Clustering Method: In Section 3, we introduced two clustering algorithms, i.e., K-means and spectral clustering. Here we take each feature map as a vertex in the graph and use distance to construct the similarity matrix, adjacent matrix, degree matrix, and Laplacian matrix using (20), (13), (14), and (17). Fig. 9 shows the salience heatmaps produced by spectral clustering with different clusters and different eigenvectors. It can be observed that the heatmaps are highly related to the number of eigenvectors rather than the clusters. A disadvantage of Cluster-CAM should be clarified that both the number of eigenvectors and the

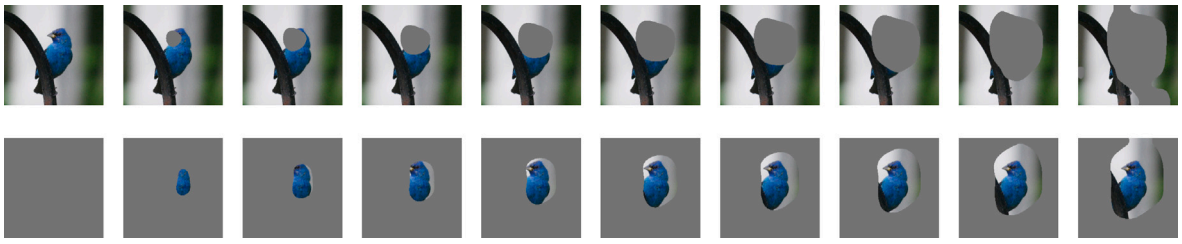

Fig. 10. The upper row delineates the deletion metric, wherein salient pixels are gradually masked from the image. Specifically, this metric measures a decrease in the probability of the predicted class as more and more important pixels are removed. Conversely, the lower row elucidates the insertion metric.

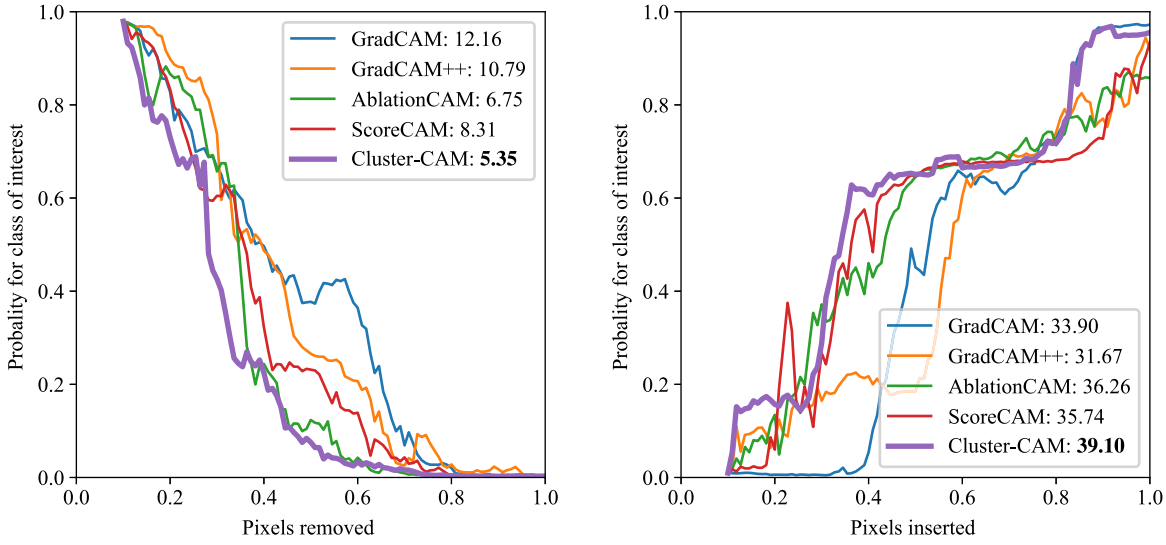

Fig. 11. Deletion and insertion curves of Cluster-CAM and other CAMs. The left subfigure shows the deletion curves of Grad-CAM, Grad-CAM++, Ablation-CAM, Score-CAM and Cluster-CAM. The AUC of each deletion curve is presented following the corresponding legend and obviously Cluster-CAM obtains the best performance (the minimal AUC). The right subfigures shows the insertion curves of these CAMs in the same layout. Note Cluster-CAM also performs the best (the maximal AUC).

number of clusters in K-means are critical parameters for saliency maps. Hence, the optimal value of above two parameters should be adjusted carefully for objects with different classes of interest.

4.5. Quantitative evaluation

4.5.1. Performance evaluation

To quantitatively evaluate the interpreting performance, two widely used evaluation metrics are adopted in this paper, i.e., confidence drop and increase number (Zeiler & Fergus, 2014; Zhou & Kainz, 2018). First of all, let us think about what kinds of heatmap can be regarded as a good interpretation of CNN. A natural and intuitive idea is to measure how much the confidence (predicted score) of the target class will drop when the original image is partly occluded according to the heatmap. Specifically, for each image, a corresponding explanation map, $\mathbf{H}$, is generated by element-wise multiplication of the heatmaps and the current image as in (9) and (10).

Confidence drop: This metric compares the average drop of the model's confidence for a particular class in an image after occlusion as:

$$\text{confidence\_drop} = \frac{S_c(\mathbf{X}) - S_c(\mathbf{X} \circ \mathbf{H})}{S_c(\mathbf{X})}, \tag{26}$$

For example, assume that CNN predicts an object indigo finch in an image $\mathbf{X}$ with confidence 0.8. When we input the explanation map, $\mathbf{X} \circ \mathbf{H}$, of this image, the CNN's confidence in the class indigo finch falls to 0.6. Then the confidence_drop would be 25%. It means that the most discriminative part (75%) is included in the highlighted region. Confidence drop is expected to be lower for a better CAM and is usually averaged over many images.

Increase number measures how many times the CNN's prediction score for c increased when the input image is masked. Specifically, it happens sometimes that the object is entirely included and object-irrelevant parts are occluded (e.g., other objects or background) in the highlighted region. In this case, there could be an increase in the CNN's predicted score for the class (i.e., confidence drop < 0). This value is computed as a percentage through the whole dataset.

Table 2 shows two evaluation metrics of VGG-16, ResNet-18, and InceptionV3 on the entire validation set in ILSVRC dataset (↓ means the lower value is better and ↑ means the higher value is better). Generally, Cluster-CAM outperforms other CAMs in both metrics. Gradient-free CAMs always obtain better performance than gradient-based CAMs (Grad-CAM and Grad-CAM++), but it is at cost of computation burden. Note that SC-SM CAM performs worse than any other CAMs. It is because the clustered feature maps are directly merged without considering which channels should be preserved or deleted. Such straightforward mechanism is suitable only for SAR images with relatively pure backgrounds but falls short when applied to optical images in ILSVRC. This result further indicates the necessity of cognition base and cognition-scissors in Cluster-CAM for CNNs trained on optical datasets. These two metrics clearly demonstrate the superiority of Cluster-CAM to other existing CAMs.

Insertion and Deletion In accordance with Petsiuk, Das, and Saenko (2018), we employ two automated assessment metrics for evaluating CAM's interpretative capabilities: deletion and insertion. The deletion metric gauges the decline in class probability as salient pixels are

Table 2
The comparison of Cluster-CAM and other state of the art methods in discrimination.

Method	Backbone	Discrimination metrics	
		Confidence drop↓	Increase number ↑
Grad	VGG-16	17.94	19.15
Grad++	VGG-16	18.44	19.75
Ablation	VGG-16	12.38	24.67
Score	VGG-16	12.21	25.48
SC-SM	VGG-16	17.52	18.33
Cluster	VGG-16	**11.60**	**26.10**
Grad	ResNet-18	19.32	23.47
Grad++	ResNet-18	18.95	22.54
Ablation	ResNet-18	10.36	27.30
Score	ResNet-18	9.46	28.75
SC-SM	ResNet-18	19.47	21.39
Cluster	VGG-16	**8.26**	**30.73**
Grad	InceptionV3	15.12	25.39
Grad++	InceptionV3	14.68	26.05
Ablation	InceptionV3	10.76	27.92
Score	InceptionV3	11.20	27.16
SC-SM	InceptionV3	16.83	24.77
Cluster	InceptionV3	**7.94**	**33.35**

Table 3
The comparison of Cluster-CAM and other state of the art methods in localization.

Method	Backbone	*Loc.Acc*		
		Top-1	Top-5	*Gt − Known*
Grad	VGG-16	43.5	53.6	59.4
Grad++	VGG-16	44.7	52.5	60.0
Ablation	VGG-16	47.7	58.6	**62.5**
Score	VGG-16	47.4	58.3	61.8
SC-SM	VGG-16	45.2	53.0	60.9
Cluster	VGG-16	**48.5**	**59.4**	61.2
Grad	ResNet-18	44.8	54.3	60.2
Grad++	ResNet-18	43.7	54.0	60.4
Ablation	ResNet-18	46.8	57.3	**63.5**
Score	ResNet-18	46.5	**57.5**	63.2
SC-SM	ResNet-18	45.8	54.3	60.0
Cluster	ResNet-18	**49.2**	57.2	60.4
Grad	InceptionV3	46.3	58.2	62.7
Grad++	InceptionV3	48.5	60.4	64.2
Ablation	InceptionV3	50.5	61.9	65.3
Score	InceptionV3	51.2	62.3	**66.4**
SC-SM	InceptionV3	47.7	59.5	62.3
Cluster	InceptionV3	**53.5**	**62.8**	64.5

gradually removed from the image, as shown in Fig. 10 (top). A sharp decrease serves as an indicator of a proficient explanation (leading to a minimized Area Under Curve AUC). In contrast, the insertion metric assesses the significance of pixels in terms of their ability to synthesize an image. This is determined by the elevation in the probability of the target class when pixels are incrementally inserted based on the saliency map, as shown in Fig. 10 (bottom). It quantifies the elevation in probability with the gradual introduction of pixels, and a higher AUC signifies a more effective and comprehensive explanation. Fig. 11 shows the deletion and insertion curves of Cluster-CAM and other CAMs. Obviously, Cluster-CAM outperforms other CAMs in both deletion and insertion metrics. Specifically, the deletion AUC of Cluster-CAM (5.35) is lower than the second best method, Score-CAM (8.31), by over 35.6%. The insertion AUC of Cluster-CAM stands at 39.10, surpassing the second-best method, Ablation-CAM whose AUC reaches 36.26, signifying an approximately 8% superiority.

Localization Top-1 and Top-5 Localization Accuracy (Top-1/Top-5 *Loc.Acc*) and Localization Accuracy with Ground Truth (*Gt-Known Loc.Acc*) are used as metrics for evaluating localization performance. Note a positive prediction of *Loc.Acc* means it meets the following criteria: the predicted classification is correct, and the predicted bounding boxes have an intersection over union (IoU) exceeding 50% with at least one of the ground-truth boxes. *Gt-Known* signifies that it considers localization regardless of classification. The numerical results in Table 3 demonstrate that Cluster-CAM surpasses other CAMs in Top-1/Top-5 *Loc.Acc*, while inferior to other Score-CAM or Ablation-CAM in *Gt-Known*. It is worth noting that Cluster-CAM is tailored to alleviate "semantic-chaos" which usually occurs in clustered feature maps. Unlike other CAMs, Cluster-CAM focuses more on correcting semantic-chaos in heatmaps rather than generating heatmaps that cover the entire object as much as possible. This is why Cluster-CAM performs better on interpretability metrics than other CAMs, but its performance on localization metrics is not necessarily the best. The example in Fig. 6 further demonstrates that the highlighted forehead region of the monkey, despite yielding a lower IoU, is highly indicative of the guenon classification.

4.5.2. Efficiency evaluation

We present two efficiency metrics in Table 4, i.e., the average computing time (ACT) and the number of forward propagation (FPn) per image. Note that cluster computation time (CCT) is included in ACT as a fair comparison. Table 4 shows Cluster-CAM greatly reduces the number of FP compared with Ablation-CAM and Score-CAM. Naturally, a significant improvement in efficiency emerges from Cluster-CAM, i.e., Cluster-CAM is 5.7 times faster than Ablation CAM and 12.1 times faster than Score-CAM. Therefore, Cluster-CAM can obtain better visualization and interpretation performance than gradient-based and gradient-free CAMs with efficiency closer to gradient-based CAMs. Besides, Table 5. presents the relation between ACT of Cluster-CAM and two hypterparameters in Spectral Clustering (i.e., cluster number, Q, and eigenvector number, k). The results show the number of clusters plays a significant role in impacting ACT, whereas the count of eigenvectors exerts a comparatively minor effect on it.

Table 4
Efficiency Evaluation Metrics: Cluster computation time (CCT), average computing time (ACT) and the number of forward propagation (FPn). Note the CCT is included in ACT for Cluster-CAM with $Q = 6$.

Method	Backbone	Efficiency		
		CCT (s)	ACT (s)	FPn
Grad	VGG-16	–	0.078	1
Grad++	VGG-16	–	0.141	1
Ablation	VGG-16	–	2.206	256
Score	VGG-16	–	4.647	256
SC-SM	VGG-16	0.130	0.215	6
Cluster	VGG-16	0.133	0.382	6
Grad	ResNet-18	44.8	54.3	60.2
Grad++	ResNet-18	43.7	54.0	60.4
Ablation	ResNet-18	46.8	57.3	63.5
Score	ResNet-18	46.5	57.5	63.2
SC-SM	ResNet-18	37.8	42.3	45.5
Cluster	ResNet-18	**49.2**	59.2	60.4
Grad	InceptionV3	46.3	58.2	62.7
Grad++	InceptionV3	48.5	60.4	64.2
Ablation	InceptionV3	50.5	61.9	65.3
Score	InceptionV3	51.2	62.7	66.0
SC-SM	InceptionV3	42.7	45.5	52.3
Cluster	InceptionV3	**53.5**	**64.4**	65.5

Table 5
The ACT of Cluster-CAM with different pairs of Cluster numbers, Q, and eigenvector numbers, k.

k \ Q	2	3	4	5	6	7	8
2	0.22	0.26	0.29	0.35	0.38	0.39	0.43
3	0.22	0.27	0.30	0.35	0.37	0.41	0.43
4	0.23	0.26	0.31	0.34	0.38	0.40	0.43
5	0.21	0.26	0.31	0.36	0.38	0.39	0.44
6	0.22	0.26	0.29	0.35	0.37	0.40	0.42
7	0.22	0.25	0.30	0.35	0.37	0.40	0.43

5. Conclusion

In this paper, we proposed Cluster-CAM, an effective and efficient CNN interpretation technique based on clustering algorithms. Cluster-CAM is the first attempt to comprehensively analyze how to split feature maps into different groups and provide an artful strategy to remove the object-irrelevant elements by defining cognition-scissors. In Cluster-CAM, only several times of forward propagation is required per image while it is usually more than hundreds for other gradient-free CAMs. We acknowledge two imperfections of Cluster-CAM: (1) for different objects, careful adjustment of two hyper-parameters is demanded; (2) in situation without semantic-chaos, Cluster-CAM cannot always guarantee the best localization performance. However, it still provides a more understandable and more efficient visual interpretation for CNNs' mechanism in most cases. Qualitative and quantitative experimental results further verified Cluster-CAM can obtain better performance in CNNs' interpretation than gradient-free CAMs with much lower computing time.

CRediT authorship contribution statement

Zhenpeng Feng: Writing – review & editing, Writing – original draft, Methodology, Conceptualization. **Hongbing Ji:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Funding acquisition. **Miloš Daković:** Software, Formal analysis. **Xiyang Cui:** Visualization, Software, Methodology, Investigation. **Mingzhe Zhu:** Supervision, Resources, Funding acquisition. **Ljubiša Stanković:** Writing – review & editing, Writing – original draft, Supervision, Project administration, Methodology, Investigation.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Hongbing Ji reports financial support was provided by National Natural Science Foundation of China.

Data availability

ILSVRC dataset can be downloaded from the website https://www.image-net.org/challenges/LSVRC/.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (No. 62276204).

References

Cao, J., Pang, Y., Han, J., & Li, X. (2019). Hierarchical shot detector. In *Proceedings of the IEEE/CVF international conference on computer vision* (pp. 9705–9714).

Chattopadhay, A., Sarkar, A., Howlader, P., & Balasubramanian, V. N. (2018). Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks. In *Proceedings of 2018 IEEE winter conference on applications of computer vision* (pp. 839–847). IEEE.

Chen, H., Jin, Y., Jin, G., Zhu, C., & Chen, E. (2022). Semisupervised semantic segmentation by improving prediction confidence. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(9), 4991–5003. http://dx.doi.org/10.1109/TNNLS.2021.3066850.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A large-scale hierarchical image database. In *Proceedings of 2009 IEEE conference on computer vision and pattern recognition* (pp. 248–255). http://dx.doi.org/10.1109/CVPR.2009.5206848.

Feng, Z., Ji, H., Stanković, L., Fan, J., & Zhu, M. (2021). SC-SM CAM: An efficient visual interpretation of CNN for SAR images target recognition. *Remote Sensing*, *13*(20), 4139.

Feng, Z., Zhu, M., Stanković, L., & Ji, H. (2021). Self-matching CAM: A novel accurate visual explanation of CNNs for SAR image interpretation. *Remote Sensing*, *13*(9), 1772.

Fu, R., Hu, Q., Dong, X., Guo, Y., Gao, Y., & Li, B. (2020). Axiom-based Grad-CAM: Towards accurate visualization and explanation of CNNs. In *Proceedings of the 2020 British machine vision conference*.

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of 2016 IEEE conference on computer vision and pattern recognition* (pp. 770–778). http://dx.doi.org/10.1109/CVPR.2016.90.

Jung, H., & Oh, Y. (2021). Towards better explanations of class activation mapping. In *Proceedings of the 2021 IEEE/CVF international conference on computer vision* (pp. 1316–1324).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). ImageNet classification with deep convolutional neural networks. In F. Pereira, C. Burges, L. Bottou, & K. Weinberger (Eds.), *Advances in neural information processing systems, vol. 25*. Curran Associates, Inc.

Lapuschkin, S., Wäldchen, S., Binder, A., Montavon, G., Samek, W., & Müller, K. R. (2019). Unmasking clever hans predictors and assessing what machines really learn. *Nature Communications*, *10*(1), 1–8.

Liang, X., Hu, Z., Zhang, H., Lin, L., & Xing, E. P. (2018). Symbolic graph reasoning meets convolutions. *Advances in Neural Information Processing Systems*, *31*.

Liu, Z., Mao, H., Wu, C. Y., Feichtenhofer, C., Darrell, T., & Xie, S. (2022). A convnet for the 2020s. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 11976–11986).

Liu, K., Meng, R., Li, L., Mao, J., & Chen, H. (2022). SiSL-Net: Saliency-guided self-supervised learning network for image classification. *Neurocomputing*, *510*, 193–202. http://dx.doi.org/10.1016/j.neucom.2022.09.029.

Liu, J., Zhang, F., Zhou, Z., & Wang, J. (2023). BFMNet: Bilateral feature fusion network with multi-scale context aggregation for real-time semantic segmentation. *Neurocomputing*, *521*, 27–40. http://dx.doi.org/10.1016/j.neucom.2022.11.084.

Ma, X., Zhang, S., Pena-Pena, K., & Arce, G. R. (2021). Fast spectral clustering method based on graph similarity matrix completion. *Signal Processing*, *189*, Article 108301. http://dx.doi.org/10.1016/j.sigpro.2021.108301.

Macpherson, T., Churchland, A., Sejnowski, T., DiCarlo, J., Kamitani, Y., Takahashi, H., et al. (2021). Natural and Artificial Intelligence: A brief introduction to the interplay between AI and neuroscience research. *Neural Networks*, *144*, 603–613. http://dx.doi.org/10.1016/j.neunet.2021.09.018.

Omeiza, D., Speakman, S., Cintas, C., & Weldermariam, K. (2019). Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. arXiv preprint arXiv:1908.01224.

Petsiuk, V., Das, A., & Saenko, K. (2018). RISE: Randomized input sampling for explanation of black-box models. arXiv preprint arXiv:1806.07421v3.

Ramaswamy, H. G., et al. (2020). Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *Proceedings of the IEEE winter conference on applications of computer vision* (pp. 983–991).

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of 2016 IEEE conference on computer vision and pattern recognition* (pp. 779–788). http://dx.doi.org/10.1109/CVPR.2016.91.

Ren, J., Li, M., Liu, Z., & Zhang, Q. (2021). Interpreting and disentangling feature components of various complexity from DNNs. In *Proceedings of international conference on machine learning* (pp. 8971–8981). PMLR.

Saleem, R., Yuan, B., Kurugollu, F., Anjum, A., & Liu, L. (2022). Explaining deep neural networks: A survey on the global interpretation methods. *Neurocomputing*, *513*, 165–180. http://dx.doi.org/10.1016/j.neucom.2022.09.129.

Scalzo, B., Stanković, L., Daković, M., Constantinides, A. G., & Mandic, D. P. (2023). A class of doubly stochastic shift operators for random graph signals and their boundedness. *Neural Networks*, *158*, 83–88. http://dx.doi.org/10.1016/j.neunet.2022.10.035.

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 2017 IEEE international conference on computer vision* (pp. 618–626).

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *3rd international conference on learning representations* (pp. 1–14).

Spinelli, I., Scardapane, S., & Uncini, A. (2022). A meta-learning approach for training explainable graph neural networks. *IEEE Transactions on Neural Networks and Learning Systems*, 1–9. http://dx.doi.org/10.1109/TNNLS.2022.3171398.

Srinivas, A., Lin, T.-Y., Parmar, N., Shlens, J., Abbeel, P., & Vaswani, A. (2021). Bottleneck transformers for visual recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition* (pp. 16519–16529).

Stankovic, L., Dakovic, M., & Sejdic, E. (2017). Vertex-frequency analysis: A way to localize graph spectral components [lecture notes]. *IEEE Signal Processing Magazine*, *34*(4), 176–182. http://dx.doi.org/10.1109/MSP.2017.2696572.

Stankovic, L., Mandic, D. P., Dakovic, M., Kisil, I., Sejdic, E., & Constantinides, A. G. (2019). Understanding the basis of graph signal processing via an intuitive example-driven approach. *IEEE Signal Processing Magazine*, *36*(6), 133–145. http://dx.doi.org/10.1109/MSP.2019.2929832.

Sun, S., Song, B., Cai, X., Du, X., & Guizani, M. (2022). CAMA: Class activation mapping disruptive attack for deep neural networks. *Neurocomputing*, *500*, 989–1002. http://dx.doi.org/10.1016/j.neucom.2022.05.065.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., & Wojna, Z. (2016). Rethinking the inception architecture for computer vision. In *2016 IEEE conference on computer vision and pattern recognition* (pp. 2818–2826).

Tan, R., Gao, L., Khan, N., & Guan, L. (2022). Interpretable artificial intelligence through locality guided neural networks. *Neural Networks*, *155*, 58–73. http://dx.doi.org/10.1016/j.neunet.2022.08.009.

Townsend, J., Chaton, T., & Monteiro, J. M. (2020). Extracting relational explanations from deep neural networks: A survey from a neural-symbolic perspective. *IEEE Transactions on Neural Networks and Learning Systems*, *31*(9), 3456–3470. http://dx.doi.org/10.1109/TNNLS.2019.2944672.

Tu, Z., Zhou, A., Gan, C., Jiang, B., Hussain, A., & Luo, B. (2021). A novel domain activation mapping-guided network (DA-GNT) for visual tracking. *Neurocomputing*, *449*, 443–454. http://dx.doi.org/10.1016/j.neucom.2021.03.056.

Vlahek, D., & Mongus, D. (2021). An efficient iterative approach to explainable feature learning. *IEEE Transactions on Neural Networks and Learning Systems*, 1–13. http://dx.doi.org/10.1109/TNNLS.2021.3107049.

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., et al. (2020). Score-CAM: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition (CVPR) workshops* (pp. 24–25).

Zeiler, M. D., & Fergus, R. (2014). Visualizing and understanding convolutional networks. In *European conference on computer vision* (pp. 818–833). Springer.

Zhang, Q., Rao, L., & Yang, Y. (2021). Group-CAM: Group score-weighted visual explanations for deep convolutional networks. arXiv preprint arXiv:2103.13859.

Zhao, Z., Xie, X., Wang, C., Liu, W., Shi, G., & Du, J. (2019). Visualizing and understanding of learned compressive sensing with residual network. *Neurocomputing*, *359*, 185–198. http://dx.doi.org/10.1016/j.neucom.2019.05.043.

Zheng, Q., Wang, Z., Zhou, J., & Lu, J. (2022). Shap-CAM: Visual explanations for convolutional neural networks based on Shapley value. In *Proceedings of the 2022 17th European conference on computer vision* (pp. 459–474).

Zhou, K., & Kainz, B. (2018). Efficient image evidence analysis of cnn classification results. arXiv preprint arXiv:1801.01693.

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

Zhu, M., Feng, Z., Stanković, L., Ding, L., Fan, J., & Zhou, X. (2022). A probe-feature for specific emitter identification using axiom-based grad-CAM. *Signal Processing*, Article 108685.

Zhu, Y., Zhao, C., Wang, J., Zhao, X., Wu, Y., & Lu, H. (2017). Couplenet: Coupling global structure with local parts for object detection. In *Proceedings of the IEEE international conference on computer vision* (pp. 4126–4134).

Zhenpeng Feng was born in Xianyang, Shaanxi, China in 1996. He received a B.E. degree in School of Electronic Engineering, Xidian University in 2019. He is currently a Ph.D. student in explainable artificial intelligence at School of Electronic Engineering, Xidian University. He is also a visiting student in the University of Montenegro, working with Prof. Ljubiša Stanković's research team. His research interests include interpreting deep neural networks and signal processing.

Hongbing Ji received a B.S. degree in radar engineering, an M.S. degree in circuit, signals, and systems, and the Ph.D. degree in signal and information processing from Xidian University, Xi'an, China, in 1983, 1989, and 1999, respectively. He is currently a full professor at Xidian University and a senior member of IEEE. His research interests include pattern recognition, radar signal processing, and multi-sensor information fusion.

Miloš Daković was born in 1970 in Nikšić, Montenegro. He received a B.S. in 1996, an M.S. in 2001, and a Ph.D. in 2005, all in electrical engineering from the University of Montenegro. He is a full professor at the University of Montenegro. His research interests are in signal processing, time-frequency signal analysis, compressive sensing, radar signal processing, and graph signal processing.

Xiyang Cui was born in Handan, Hebei, China in 1997. He received the B.E. degree and M.E. degree in Electronic Information Engineering and Electrical Circuit System from School of Electronic Engineering, Xidian University in 2019 and 2021, respectively. He is currently an investigator of an electronic company and collaborates with Zhenpeng Feng and Prof. Ljubiša Stanković in scientific research. His research interests include electrical circuit design and image processing.

Mingzhe Zhu was born in China in 1982. He received a B.S. degree in signal and information processing, a Ph.D. degree in pattern recognition and intelligent system from Xidian University in 2004 and 2010, respectively. He is currently an associate professor at School of Electronic Engineering, Xidian University. His research interests include non-stationary signal processing, time-frequency analysis, and target recognition.

Ljubiša Stanković was born in Montenegro, 1960. He was at the Ruhr University Bochum, 1997–1999, supported by the AvH Foundation. Stankovic was the Rector of the University of Montenegro 2003–2008, the Ambassador of Montenegro to the UK, 2011–2015, and a visiting academic to the Imperial College London, 2012–2013. He published almost 200 journal papers. He is a member of the National Academy of Science and Arts (CANU) and the Academia Europaea. Stanković won the Best paper award from the EURASIP in 2017 and the IEEE SPM Best Column Award for 2020. Stanković is a professor at the University of Montenegro and a Fellow of the IEEE.