IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 2 October 2022, accepted 16 October 2022, date of publication 19 October 2022, date of current version 1 November 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3215534

METHODS

Collection-CAM: A Faster Region-Based Saliency Method Using Collection-Wise Mask Over Pyramidal Features

YUNGI HA AND CHAN-HYUN YOUN, (Senior Member, IEEE)
School of Electrical Engineering, Korea Advanced Institute of Science and Technology, Daejeon 34141, South Korea

Corresponding author: Chan-Hyun Youn (chyoun@kaist.ac.kr)

This work was supported in part by Samsung Electronics Co., Ltd (SNIC), and in part by the Future Defense Technology R&D Program (No.915027201) of Agency for Defense Development in 2022, and in part by Electronics and Telecommunications Research Institute (ETRI) grant funded by the Korean government. [22ZS1100, Core Technology Research for Self-Improving Integrated Artificial Intelligence System].

ABSTRACT Due to the black-box nature of deep networks, making explanations of their decision-making is extremely challenging. A solution is using post-hoc attention mechanisms with the deep network to verify the decision basis. However, those methods have problems such as gradient noise and false confidence. In addition, existing saliency methods either have limited performance by using only the last convolution layer or suffer from large computational overhead. In this work, we propose the Collection-CAM, which generates an attention map with low computational overhead while utilizing multi-level feature maps. First, the Collection-CAM searches for the most appropriate form of the partition through bottom-up clustering and clustering validation process. Then the Collection-CAM applies different pre-processing procedures on the shallow feature map and final feature map to overcome the false positiveness when applied without distinction. By combining collection-wise masks according to their contribution to the confidence score, the Collection-CAM completes the attention map generation process. Experimental results on ImageNet1k, UC Merced, and CUB dataset and various deep network models demonstrate that the Collection-CAM not only can synthesize a saliency map with a better visual explanation but also requires significantly lower computational overhead compared to those of region-based saliency methods.

INDEX TERMS Visual explanation, deep learning, acceleration, clustering analysis.

I. INTRODUCTION

Deep Neural Networks(DNNs) exhibit superior performance and reproducibility compared to other machine learning algorithms. DNNs have outperformed professional human players in complex strategic games such as Go [1] and showed superior performance in complex tasks such as object recognition [2] and natural language interpretation [3]. Although DNNs show impressive performance in several applications, their nested nonlinear structure of them makes the model opaque, making it unclear which information in the input data serves as the basis for the decision-making. In other words, it is not possible to know in detail which part of the image the basis for DNN's judgment lies, which makes DNNs often referred to as ''Black Box''. This opacity is a clear drawback in that it is difficult to understand and validate the decision process of DNNs in many applications. However, in areas such as medical diagnosis, for the safe utilization of DNNs, the basis for judgment must be provided for experts' interpretation and verification [4]. Also, the interpretable deep model is useful for analyzing its vulnerabilities or selecting models or architecture with similar performance [5]. Accordingly, DNN explainability and human interpretability are prerequisites for ensuring that a DNN is performing correctly while being also essential for improving the functionality of DNNs.

Explanation methods for DNNs can be classified into three main categories: visualization methods, model distillation, and intrinsic methods [6]. Visualization methods illustrate an explanation by highlighting the input's specific part that strongly affects the output of the DNN. A separate

The associate editor coordinating the review of this manuscript and approving it for publication was Xianzhi Wang.

"White-Box" machine learning model is developed in model distillation to pinpoint which features of the input affect the DNN output and the decision rule of the DNN. DNNs with intrinsic methods generate predictions with explanations. In the learning process of DNNs with intrinsic methods, model performance and quality of explanations are optimized jointly. In this work, we discuss the post-hoc attention mechanisms which are classified as visualization methods.

Given a learned DNN and an input image, post-hoc attention mechanisms visualize an intuitive heatmap that shows humans the most relevant part of the DNN's decision-making. Post-hoc attention mechanisms can be divided into activation-based saliency map[1] generation methods [8], [9], [10], [11] and region-based saliency map generation methods [12], [13], [14]. In the case of feature map-based saliency methods, the required computation is very low because it needs one or several times of DNN forward propagation. However, gradient noise leads to an incorrect association between each feature map and target category. Consequently, it generates a saliency map that includes much meaningless information. On the other hand, region-based saliency methods measure the individual mask's contribution to the prediction of the target category and synthesize an attention map with improved explainability based on the measured contribution. However, their computational overhead is much larger than that of feature map-based saliency methods as they require a large execution time due to the amount of computation which is proportional to the number of masks($> 10^3$).

Existing post-hoc attention mechanisms generally utilize DNN's feature maps from the final convolutional layer to generate a saliency map. As the saliency map is a combination of feature maps, the low spatial resolution of the final feature map limits the quality of the resulting attention map. To compensate for this shortcoming, a natural way to make the attention map include more fine-grained information is to utilize shallow feature maps with higher spatial resolution together. Nonetheless, when feature maps are used for attention map synthesis without considering the individual layer's level, it could generate a saliency map with degraded explainability which emphasizes the background more than the object. In Figure 1, we can see that the resulting saliency map with shallow feature maps attends to the background more than it does to the object.

In this work, we present Collection-CAM which reduces the computational overhead of existing region-based saliency methods and generates a more fine-grained attention map by use of shallow feature maps. First, via sequential bottom-up search, we obtain partitions for each feature map level, which groups similar feature maps into a collection. Then, we identify the partition with the maximum dispersion-separation ratio for multiple feature map levels via the clustering validation process. Furthermore, we utilize the pixel-wise gradient of the feature map to generate a collection-wise mask for shallow feature maps. It suppresses the activation from the background while strengthening the fine-grained details of the object. Finally, we get an attention map by combining the collection-wise masks according to the measured importance. Our contributions are summarized as follows.

- We introduce a hybrid visual explanation method, Collection-CAM, which leverages the principles of both gradient-based and region-based saliency methods. It generates a saliency map using masks from multiple-level feature maps in an intuitively understandable way.
- We evaluate generated saliency maps by Collection-CAM quantitatively and qualitatively through extensive experiments. We measure the running time of various saliency methods and verify that it requires significantly less time than other region-based saliency methods. For faithfulness evaluation, using Average Drop / Average Increase and Deletion / Insertion metrics, we demonstrate that Collection-CAM is superior to comparative saliency methods at spotting important parts of the input image. In addition, using the proportion metric, we show that the localization ability of Collection-CAM is preferable to comparative methods.
- We describe the effectiveness of Collection-CAM in other applications. We verify that Collection-CAM can be used as a debugging tool by providing results on sanity checks. In addition, we show that the proposed framework can be used for generating high-quality object proposals for the weakly supervised object detection task.

The remainder of this work is organized as follows. Section 2 reviews existing post-hoc attention mechanisms and issues for them. Section 3 presents the design of the proposed Collection-CAM. In section 4, we validate Collection-CAM via conducting quantitative and qualitative performance comparisons with state-of-the-art methods. Finally, section 5 concludes our work.

II. PROBLEM DESCRIPTION IN RELATED WORKS

We break down saliency methods into three categories: gradient-based methods, activation-based methods, and region-based methods. First, we provide an overview of different types of saliency methods. Also, we discuss the problems that arise when we want to improve visual explanation by the use of multi-level feature maps.

A. EXISTING SALIENCY MAP GENERATION METHODS

Gradient-based saliency methods obtain attributions by applying backpropagation of the output to each layer of the network. Then they produce a saliency map by returning the obtained features to the input. Guided Backpropagation [15] attempted to enhance the quality of the saliency map

[1] In this work, we use a saliency map, an attention map, and visual explanation as the interchangeable term. In salient object detection [7], a saliency map also refers to a means to find the most salient and attention-grabbing object from the input image, which segments the foreground object from the background. However, we limit the use of saliency map, attention map, and a visual explanation to the visualization of the input image region that played a critical role in predicting as $\mathcal{F}^c$ for a classification deep model $\mathcal{F}$.

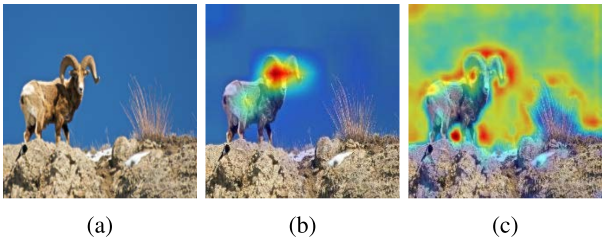


FIGURE 1. Comparison of attention maps for bighorn category image when Grad-CAM applied to ResNet101 classifier; (a) Input image (b) 'layer4' (c) ['layer1', 'layer2', 'layer3', 'layer4'].

by using the ReLU unit to set negative values as zero. Integrated Gradients [16] tried to solve the gradient saturation issue prevalent in gradient-based methods by estimating and utilizing the global importance of each pixel. Excitation Backpropagation [17] utilized a probability model called the probabilistic winner-take-all process to back-project higher-level attribution to lower-level. When calculating gradients, Smoothgrad [18] attempted to alleviate the 'visual diffusion' of 'b' generated by adding noise to the input. Gradient-based saliency methods tend to produce low-quality non-smooth visual explanations with visual noise, as it operates on a pixel-by-pixel basis.

Activation-based saliency methods generate an attention map through a linear combination of weighted feature maps. Typically, these methods utilize output feature maps and corresponding gradients of the final convolutional layer to synthesize the visual explanation. The main difference between these methods lies in how to produce weights. CAM [8] involves structural modification of target CNN architecture that substitutes the fully connected layer with the global average pooling. As a result, it can obtain attention map and prediction through single forward propagation. However, structural modification accompanies the retraining of the deep network. Grad-CAM [9] overcomes this limitation. Grad-CAM multiplies individual feature map and weight obtained by global average pooling for gradient for the target class confidence score $\mathcal{F}^c(I)$. Through the linear combination of them, Grad-CAM generates a visual explanation for the given input image. Similarly to Grad-CAM, Grad-CAM++ [10] and Layer-CAM [11] utilize individual feature maps and gradients for attention map generation. The difference between those methods lies in how to set the importance weight of each feature map. Grad-CAM++ uses the positive partial derivative of the target class for weight generation. Layer-CAM multiplies each feature map's pixel and the corresponding weight generated from the corresponding gradient value for each pixel. They are much faster than region-based saliency methods, as they require only one or several times of forward propagation to obtain the attention map. However, the weak point of those methods is that the gradient itself is not sufficient to measure the importance of each feature map. Noiseness [19] and false confidence [13] are typical examples of the low quality of activation-based saliency methods. False confidence means that the importance weight of the feature map is not proportional to the contribution to the confidence score of the target class. Noisy gradient refers to visual noise induced by gradient explosion and vanish, caused by flat-zero gradient region in activation function such as ReLU and Sigmoid common in many Deep Networks. Furthermore, as the gradient is calculated with weights from connected neighborhood layers, it ignores the relationship between adjacent pixels. As a result, it causes discontinuity between adjacent pixels.

Region-based saliency methods such as RISE [12], Score-CAM [13] and XRAI [14] preserve certain areas of the input image by introducing a mask to measure the importance of each area by propagating the masked input image. With the masks and measured corresponding importance weights, they synthesize an attention map. RISE randomly generates thousands of masks to probe the target deep network with the masked input image. XRAI uses over-segmented image acquired through Felzenswalb's algorithm [20] and pixel-level attribution generated through Integrated Gradients [16] to identify the most important part of the model prediction. Score-CAM measures the importance of individual feature maps through forward propagation of the masked image according to the intensity of each feature map. Specifically, it defines the importance of an individual feature map for the confidence score of the target class as a Channel-wise Increase of Confidence in Equation 1.

$$CIC(A^k) = \mathcal{F}^c(I \odot H^k) - \mathcal{F}^c(X_b) \tag{1}$$

where

$$H^k = s(Up(A^k)) \tag{2}$$

$\odot$ operation in Equation 1 denotes element-wise multiplication. $Up(\cdot)$ in Equation 2 denotes upsampling function for feature map A^k to enlarge it as the same size of the input image I. $s(\cdot)$ normalizes its input into [0, 1] range. Based on those Equations, Score-CAM generates a saliency map:

$$L^c_{Score-CAM} = ReLU(\sum_k \alpha^c_k A^k) \tag{3}$$

where importance weight α^c_k for feature map A^k is substituted with $CIC(A^k)$. Region-based saliency methods synthesize an attention map using mechanisms similar to the Score-CAM, resulting in a better quality attention map compared to the map generated by activation-based saliency methods. However, at the same time, this mechanism results in a much longer execution time as shown in Table 1, due to the large computation overhead proportional to the number of feature maps or generated masks.

B. GENERATING VISUAL EXPLANATION USING SHALLOW FEATURE MAPS

Saliency methods using feature maps usually use feature maps derived from final convolutional layers. It is because the final feature maps are located closest to the prediction and contain the most semantic information with minimum size. To describe how feature maps evolve along the forward

TABLE 1. Preliminary evaluation for faithfulness and running time of different visual explanation generation methods. ResNet-101 and ImageNet1k validation were used as the deep model and the dataset, respectively. A lower average drop(AD) and higher average increase(AI) suggest better faithfulness.

Method	AD(%)	AI(%)	Running time(ms)
Grad-CAM	22.24	35.35	62.23
Grad-CAM++	24.40	32.20	62.21
Score-CAM	18.99	37.60	20584.98

propagation, we give an example. We denote oth penultimate feature map as B^o, kth final feature map generated by applying the activation function f and kth convolution filters to B^o as A_k. We describe the relationship between final feature maps and penultimate feature maps in Equation 4.

$$\sum_{i}\sum_{j} A^k_{i,j} = \sum_{m}\sum_{n} W^k_{m,n} \sum_{o} f(B^o_{m,n}) \tag{4}$$

where $W^k_{m,n}$ denotes kth convolutional filter's weight for location (m, n). With Equation 4, we verify that the total intensity of the activated area is preserved between adjacent feature maps. However, we see that $m \geq i$ and $n \geq j$, which suggests that activated regions are concentrated on a smaller number of pixels due to many convolution filters and poolings along the forward propagation path. Furthermore, activation functions f such as ReLU, Sigmoid, and tanh are non-linear functions that remove fine-grained detail in shallow feature maps and refine final feature maps to contain semantics of the input image. Therefore, final feature maps usually don't show fine-grained details for the target class.

A natural approach that can be utilized to compensate for the weak point of final feature maps is to expand the range of considered feature maps including shallow feature maps that capture fine-grained detail. However, a naive approach that simply expands the target range of feature maps without considering the difference between them causes a worse visual explanation, as shown in Figure 1. This is because shallow feature maps contain fine-grained details for both the target object and background, which results in background noise. To examine the degradation of visual explanation in more detail, we measured the localization performance according to the considered feature map range cases. We used 500 samples from ImageNet1k and ResNet-101 model. In Figure 2 that illustrates the localization performance, we can observe that the localization performance becomes worse as it considers a wider range of feature maps.

III. PROPOSED COLLECTION-CAM

In this section, we provide the details for the proposed Collection-CAM, which generates the enhanced quality of visual explanation while reducing the required execution time. Collection-CAM's attention map generation pipeline is illustrated in Figure 3. First, Collection-CAM performs bottom-up hierarchical clustering using feature map representation vectors. Then, it identifies the most effective partition set among obtained partition hierarchy. After that, Collection-CAM performs post-processing in different ways depending on the level of the layer from which the feature map is extracted, and generates a mask for each collection obtained in the previous step. Finally, it measures how much each mask contributes to the target class score to synthesize an attention map. In III.1, we describe a series of feature map clustering processes. In III.2, we describe the generation of collection-wise masks using clusters and the synthesis of a visual explanation.

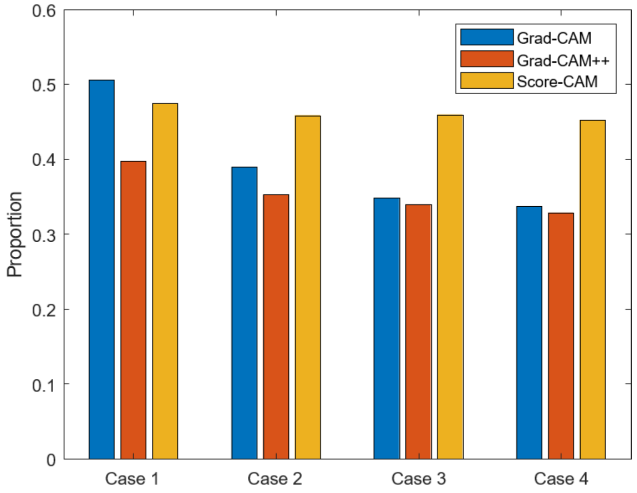


FIGURE 2. Comparison of the conventional visual explanations according to the feature map use cases. A combination with a higher proportion value suggests that it has better localization capability. ResNet101 model and ImageNet1k dataset were used. Feature use map cases are: Case 1 = ['layer4'], Case 2 = ['layer3', 'layer4'], Case 3 = ['layer2', 'layer3', 'layer4'] and Case 4 = ['layer1', 'layer2', 'layer3', 'layer4'].

A. ACQUISITION OF FEATURE MAP COLLECTION

Region-based saliency methods generate a better attention map compared to activation-based saliency methods as previously introduced but require much longer execution time due to large computational overhead. An idea that can be considered to alleviate the computational overhead problem of region-based saliency methods is to cluster similar types of feature maps into a much smaller number of collections than the number of target feature maps to perform their processing. We cluster feature maps of each layer into collections with similar characteristics to generate masks for each collection to improve the quality of the resulting saliency map and reduce the required computation. This subsection describes a specific procedure for securing clustered feature map collection for each layer. To this end, we utilize each feature map A^i's representation vector ψ_i.

We define $\psi_i \in \Psi$ as a representation vector for A^i. It is comprised of (a) Mean of feature Intensity(mi_i), (b) Standard deviation of feature Intensity(si_i), (c) Mean of feature Gradient (mg_i), (d) Mean of feature Gradient (sg_i) as:

$$\psi_i = (mi_i, si_i, mg_i, sg_i) \tag{5}$$

Equation 6 is a square euclidean distance between different feature map presentation vectors, which is used for

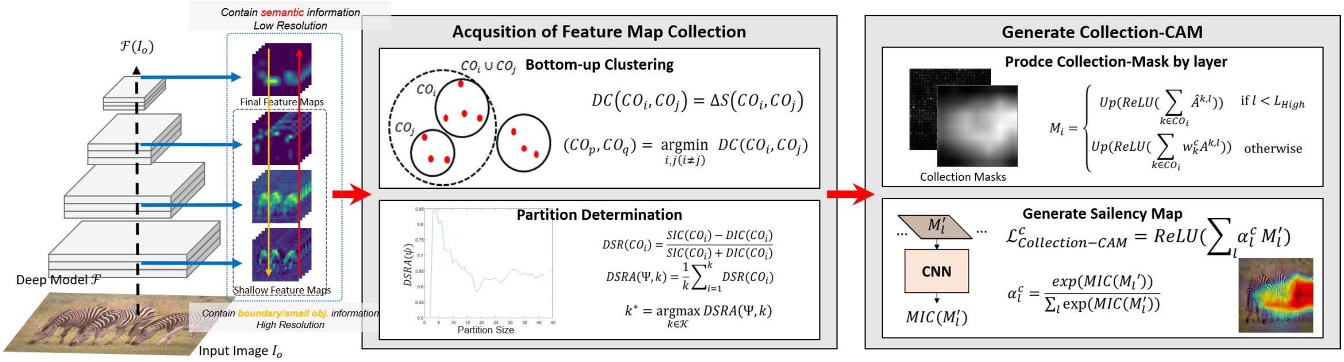

Contain semantic information
Low Resolution
$\mathcal{F}(I_o)$
Final Feature Maps
Shallow Feature Maps
Deep Model $\mathcal{F}$
Contain boundary/small obj. information
High Resolution
Input Image I_o
Acqusition of Feature Map Collection
$CO_i \cup CO_j$ Bottom-up Clustering
CO_i
CO_j
$DC(CO_i, CO_j) = \Delta S(CO_i, CO_j)$
$(CO_p, CO_q) = \underset{i,j(i\neq j)}{\mathrm{argmin}}\ DC(CO_i, CO_j)$
Partition Determination
$DSRA(\Psi)$
Partition Size
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
0
5
10
15
20
25
30
35
40
45
$DSR(CO_i) = \frac{SIC(CO_i) - DIC(CO_i)}{SIC(CO_i) + DIC(CO_i)}$
$DSRA(\Psi, k) = \frac{1}{k}\sum_{i=1}^{k} DSR(CO_i)$
$k^* = \underset{k\in\mathcal{K}}{\mathrm{argmax}}\, DSRA(\Psi, k)$
Generate Collection-CAM
Prodce Collection-Mask by layer
Collection Masks
$M_i = \begin{cases} Up(ReLU(\sum_{k\in CO_i} \hat{A}^{k,l})) & \text{if } l < L_{High} \\ Up(ReLU(\sum_{k\in CO_i} w_k^c A^{k,l})) & \text{otherwise} \end{cases}$
Generate Sailency Map
M_l'
...
...
CNN
$MIC(M_l')$
$\mathcal{L}^c_{Collection-CAM} = ReLU(\sum_l \alpha_l^c M_l')$
$\alpha_l^c = \frac{exp(MIC(M_l'))}{\sum_l \exp(MIC(M_l'))}$

FIGURE 3. **Overall procedure of the proposed Collection-CAM.**

inter-vector dissimilarity measurement.

$$DV(\psi_i, \psi_j) = \|\psi_i - \psi_j\|^2 \tag{6}$$

$m(CO_i)$ is the centroid of the feature map presentation vector belonging to the collection CO_i, as shown in Equation 7.

$$m(CO_i) = \frac{1}{|CO_i|} \sum_{\psi_k \in CO_i} \psi_k \tag{7}$$

We can formulate objective function for a partition which has NC collections as as Equation 8, where NC is within desired number of collections range $\mathcal{K}$.

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{NC} \frac{1}{2|CO_j|} \sum_{\psi_i, \psi_{i'} \in CO_j} DV(\psi_i, \psi_{i'}) \\ \text{subject to} \quad & \bigcup_{i=1}^{NC} CO_i = \mathcal{A}_l, \\ & \text{for } i \neq j, \quad CO_i \cap CO_j = \emptyset \end{aligned} \tag{8}$$

In the Equation 8, if we replace feature map presentation vector ψ_i with a series of points x_i and feature map collection CO_j with cluster C_j, it can be interpreted as an optimization problem of the k-means clustering. The objective function of the k-means clustering is denoted as:

$$\begin{aligned} \text{minimize} \quad & \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2 \\ \text{subject to} \quad & \bigcup_{i=1}^{k} C_i = \mathcal{X}, \\ & \text{for } i \neq j, \quad C_i \cap C_j = \emptyset \end{aligned} \tag{9}$$

Equation 9 is proven to be NP-Complete [21]. To solve this optimization problem, we leverage the hierarchical clustering method that operates in a bottom-up direction to generate partition hierarchy through a series of merging from $|\Psi|$ singleton collections. We perform hierarchical bottom-up clustering based on the dissimilarity between different collections $DC(\cdot)$. We define $DC(\cdot)$ as:

Definition 1 (Dissimilarity Between Collections): Extending $DV(\psi_i, \psi_j)$, we denote squared error sum of CO_i as $S(CO_i)$

$$S(CO_i) = \sum_{\psi_k \in CO_i} \|\psi_k - m(CO_i)\|^2 \tag{10}$$

Leveraging [22], we define the increment of squared error sum when CO_i and CO_j are merged $\Delta S(CO_i, CO_j)$ as dissimilarity between collection $DC(CO_i, CO_j)$, which is denoted as Equation 11.

$$\begin{aligned} DC(CO_i, CO_j) &= \Delta S(CO_i, CO_j) \\ &= S(CO_i \cup CO_j) - S(CO_i) - S(CO_j) \end{aligned} \tag{11}$$

We specify procedures for sequentially acquiring partition hierarchy $\mathcal{PH}$ based on $DC(\cdot)$. First, we measure the $DC(\cdot)$ between the collections initialized as singleton as shown in Equation 11. Among the collections, we search for the collection pair with the minimal $DC(\cdot)$ to merge them into a new collection. Then, $DC(\cdot)$ between the merged collection and the remaining collections are measured. It repeats until a partition with a collection number of 2 is secured.

As a result, we obtain a partition hierarchy $\mathcal{PH} = \{\mathcal{P}_2, \ldots, \mathcal{P}_{|\Psi|}\}$, where k corresponds to the number of collections held by partition $\mathcal{P}_k \in \mathcal{PH}$.

We validate the goodness of each partition $\mathcal{P}_k \in \mathcal{PH}(\forall k \in \mathcal{K})$ to select the partition utilized for collection-wise mask generation. Clustering tries to gather representation vectors belonging to the same collection as similar as possible, while it tries to separate distinct representation vectors into different collections. Clustering quality verification is the task of determining the superiority and inferiority between different partitions. For cluster quality verification for each partition, we define clustering validation indicators. First, we define $DIC(CO_i)$, which is the dispersion of intra-collection for collection $CO_i \in \mathcal{P}_k$. $DIC(CO_i)$ denotes average dissimilarity between $\psi_{i'} \in CO_i$ and $m(CO_i)$.

$$DIC(CO_i) = \frac{1}{|CO_i|} \sum_{\psi_{i'} \in CO_i} \|\psi_{i'} - m(CO_i)\|^2 \tag{12}$$

$SIC(CO_i)$, which is the separation of inter-collection for collection CO_i, is denoted as:

$$SIC(CO_i) = \min_{j \neq i} \|m(CO_i) - m(CO_j)\|^2 \tag{13}$$

From the perspective of $DIC(\cdot)$, the goodness of clustering is to make a collection with small $DIC(\cdot)$ as possible. Meanwhile, from the perspective of $SIC(\cdot)$, the clustering quality for a collection depends on how big it is. Considering these perspectives together, we define the dispersion-separation ratio for collection $DSR(CO_i)$, which is an clustering quality indicator for collection CO_i.

Definition 2 (Dispersion-Separation Ratio): We use the difference and sum ratio of $SIC(CO_i)$ and $DIC(CO_i)$ to evaluate clustering quality of collection CO_i. We regard this as the dispersion-separation ratio, which is denoted as:

$$DSR(CO_i) = \frac{SIC(CO_i) - DIC(CO_i)}{SIC(CO_i) + DIC(CO_i)} \tag{14}$$

As it can be seen from 14, we take $SIC(CO_i)$ and $DIC(CO_i)$ into together to define $DSR(CO_i)$, which is clustering quality indicator for CO_i. We use the difference between $SIC(CO_i)$ and $DIC(CO_i)$ at the numerator of $DSR(CO_i)$ to make $DSR(CO_i)$ an increasing function for the collection CO_i that has small $DIC(CO_i)$ and large $SIC(CO_i)$. In addition, we place the sum of $SIC(CO_i)$ and $DIC(CO_i)$ at the denominator of $DSR(CO_i)$ to prevent $DSR(CO_i)$ from becoming too large. We intend to prevent one collection from having an excessive effect on the clustering quality indicator for partition $DSRA(\mathcal{P}_k)$, which is the average dispersion-separation ratio of partition $\mathcal{P}_k$. We denote it as:

$$DSRA(\mathcal{P}_k) = \frac{1}{k} \sum_{CO_i \in \mathcal{P}_k} DSR(CO_i) \tag{15}$$

We identify the partition with the high clustering quality indicator value based on `BottomUpClustering` function shown in Algorithm 1 and $DSRA(dot)$. We provide detailed description in Lemma 1.

Lemma 1 (Determination of Partition): Let $\mathcal{PH} = \{\mathcal{P}_k\}$ $(\forall k \in \mathcal{K})$ be a generated partition set through `BottomUpClustering` *function in Algorithm 1 when partition size range is given as $\mathcal{K}$. The partition used to create the collection mask, which has the largest clustering indicator value, is denoted as*:

$$\mathcal{P}^* = \underset{\mathcal{P}_k \in \mathcal{PH}}{argmax} DSRA(\mathcal{P}_k) \tag{16}$$

Proof: $DSR(\cdot)$ describes the clustering goodness of a single collection. We regard CO_i to be the better clustering the higher $DSR(CO_i)$ that CO_i has. We evaluate the clustering quality of the partition using $DSRA(\cdot)$, the average value of $DSR(\cdot)$ of the collection belonging to a particular partition. Accordingly, partition $\mathcal{P}_k \in \mathcal{PH}$ having the highest $DSRA(\mathcal{P}_k)$ may be deemed as the best clustering result among partitions in $\mathcal{PH}$. □

B. GENERATE COLLECTION-CAM

We apply the method of generating the collection-wise mask differently depending on the extraction location of the feature map. When the extraction location of the target feature map A^k is shallow layer$(l < L)$, we calculate $\hat{A}^k$ by modifying feature map intensity according to A^k's gradient at spatial location (i,j) $\frac{\partial \mathcal{F}_c(I_o)}{\partial A^k_{i,j}}$ to remove the background detail, which is denoted as:

$$\hat{A}^k_{i,j} = \begin{cases} A^k_{i,j}, & \text{if } \dfrac{\partial \mathcal{F}_c(I_o)}{\partial A^k_{i,j}} > 0 \\ 0, & \text{otherwise} \end{cases} \tag{17}$$

We combine spatial filtered feature map $\hat{A}^k$ in collection CO_s to generate a mask M_s, which is denoted as:

$$M_s = Up(ReLU(\sum_{\hat{A}^k \in CO_s} \hat{A}^k)) \tag{18}$$

The entire process of acquiring masks for shallow feature maps is summarized in `ShallowMapMask` function in Algorithm 1. In the case of the final feature map that focuses on image semantics, we synthesize the collection-wise mask using the global-average pooled gradient value for the feature map, which is denoted as follows:

$$M_f = Up(ReLU(\sum_{k \in CO_f} w^c_k A^k)) \tag{19}$$

$$w^c_k = \frac{1}{Z} \sum_i \sum_j \frac{\partial \mathcal{F}_c(I_o)}{\partial A^k_{i,j}} \tag{20}$$

This process corresponds to the operation of the `FinalMapMask` function in Algorithm 1. As the gradient is in common for the mask generation process of each feature map collection, we apply visual denoising to remove the noise generated from the gradient, according to Equation 21.

$$Denoise(M_l, \theta) = \begin{cases} m_{i,j}, & \text{if } m_{i,j} > PI(M_l, \theta) \\ 0, & \text{otherwise} \end{cases} \tag{21}$$

where $PI(M_l, \theta)$ denotes the θth percentile intensity value of M_l. Then, we obtain M'_l by scaling pixel intensities to [0,1] range via normalization of $Denoise(M_l, \theta)$ as:

$$M'_l = \frac{Denoise(M_l, \theta) - \min(Denoise(M_l, \theta))}{\max(Denoise(M_l, \theta)) - \min(Denoise(M_l, \theta))} \tag{22}$$

How to measure the importance of an individual collection mask for model prediction is to obscure or perturb the rest of the area highlighted by the collection mask. It enables us to estimate how much it affects the decision of the deep network, which is the "Black Box". We define the mask-wise increase of confidence (MIC) to measure the importance of a collection mask in a deep model's decision. $MIC(\cdot)$'s detailed description is provided in Definition 3.

Definition 3 (Mask-wise Increase of Confidence (MIC)): We suppose that CNN $\mathcal{F}$ outputs a prediction $\mathcal{F}_c(I_o)$ for the input image I_o. When replacing baseline image X_b with Hadamard product of the collection mask M'_i and the original image I_o, we can quantify the degree to which it contributes to the confidence score $\mathcal{F}_c(I_o)$. We formally define $MIC(M'_i)$, which is quantified contribution to the the confidence score $\mathcal{F}_c(I_o)$ of $MIC(M'_i)$ as Equation 23.

$$MIC(M'_i) = \mathcal{F}_c(I_o \odot M'_i) - \mathcal{F}_c(X_b) \tag{23}$$

Based on the mask contribution definition, we describe an attention map generation of the proposed saliency method. Lastly, we assemble an attention map using collection-wise masks and corresponding weight as:

$$\mathcal{L}^c_{Collection-CAM} = ReLU(\sum_i \alpha^c_i M'_i) \tag{24}$$

$$\alpha^c_i = \frac{exp(MIC(M'_i))}{\sum_i exp(MIC(M'_i))} \tag{25}$$

In Equation 25, Collection-CAM obtains weight for individual collection mask, which is denoted with $MIC(\cdot)$. Given an arbitrary input, the magnitude of the output score in the layer that immediately precedes softmax is not fixed. A method of utilizing output scores to limit them to fixed categories is to utilize softmax. This motivates us to use the softmax function for weights that are utilized in the linear combination of collection masks. Also, as we are interested only in pixels with a positive influence on the target class prediction, we apply $ReLU(\cdot)$ to eliminate the influence of negative pixels.

We provide complete detail of the implementation in Algorithm 1. We cluster feature maps by layer level from which feature maps are extracted to secure feature map collection. Then, we apply a different collection mask synthesis function for its level. Finally, we measure the mask-wise increase of confidence for each collection mask, generating a saliency map.

IV. EXPERIMENTS

We evaluate the performance of the proposed post-hoc attention method on CNN models designed for image classification tasks. We conduct extensive experiments to answer the following questions.

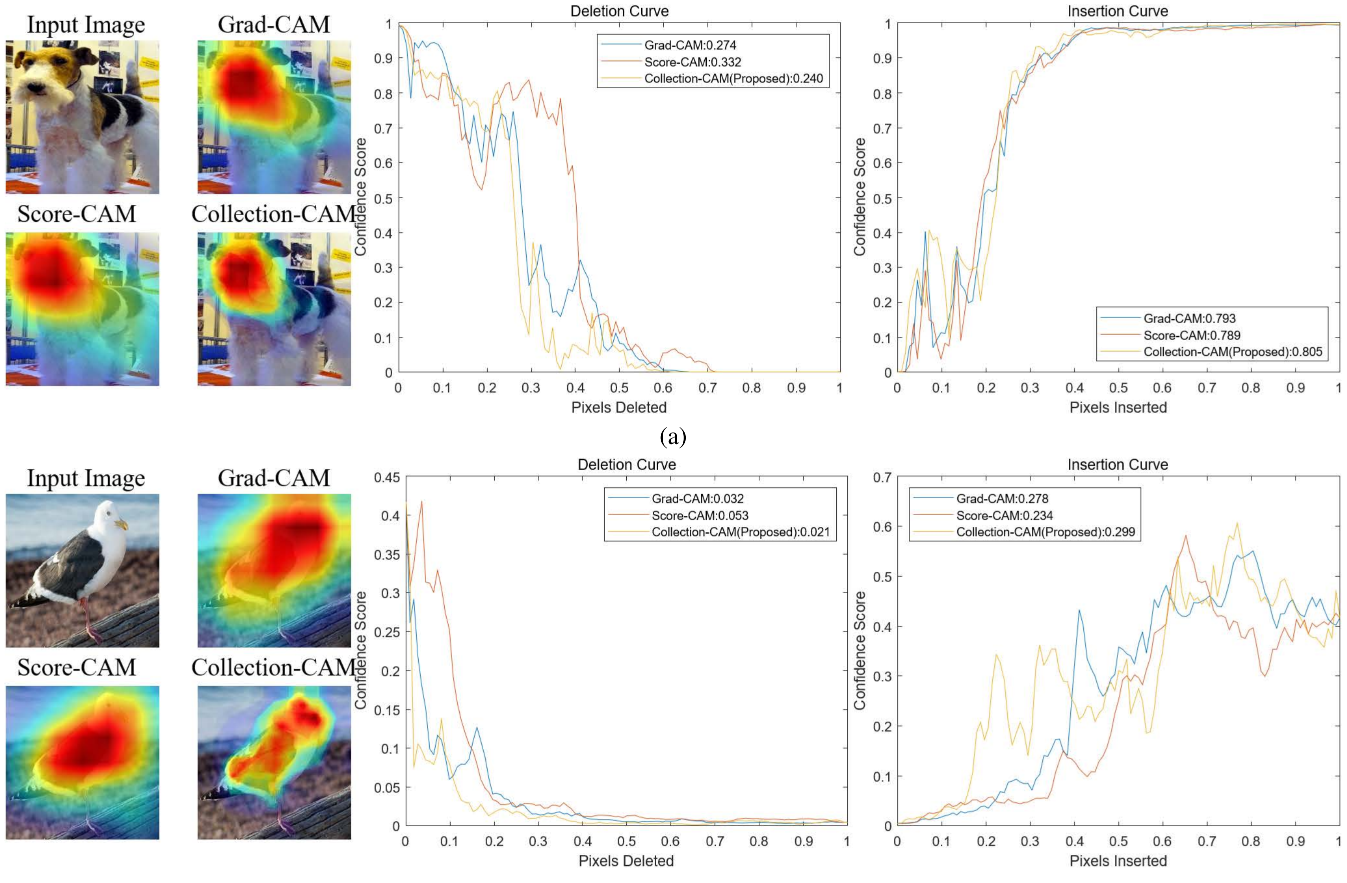
- What qualitative characteristics does the Collection-CAM have?
- How effective is the Collection-CAM in reducing the computation overhead of region-based saliency methods?
- How well does the Collection-CAM highlight the pixels it considers important in the deep model's decision-making?
- How much does the highlighted area by the Collection-CAM correspond to the real object area?
- Does the Collection-CAM reflect the change in model parameters when creating a visual explanation?

Experimental Setup: We tested the Collection-CAM and comparative saliency methods in an environment running publicly available PyTorch 1.8.1 on nodes with Nvidia RTX 3080 GPUs. The datasets used in the experiment are openly accessible ImageNet1k val [23], UC Merced [24], and CUB-200-2011 val [25] dataset. ImageNet1k Val dataset has a total of 50,000 images across 1000 different categories. UC Merced dataset describes land use, which holds 100 images for each category in 21 categories. CUB-200-2011 Val dataset contains 5794 images for 200 different types of birds.

Algorithm 1 Collection-CAM Algorithm

```
INPUT: Deep Network $\mathcal{F}$, Target layer $\mathcal{L}$, Baseline Image $X_b$, Input Image $I_o$, Partition Size Range $\mathcal{KS}$, Percentage Intensity $\theta$
OUTPUT: Saliency Map $\mathcal{L}^c_{Collection-CAM}$
function BottomUpClustering($\Psi$)
  $CO_i = \{\psi_i\}, \mathcal{P}_{|\Psi|} = \{CO_1, \ldots, CO_{|\Psi|}\}$
  $DC(CO_i, CO_{i'}) = \|\psi_i - \psi_{i'}\|^2$
  for $k = |\Psi|$ to 2 do
    $(p, q) \leftarrow \underset{i,j,(i \neq j)}{\text{argmin}} DC(CO_i, CO_j)$
    $CO_r \leftarrow (CO_p \cup CO_q)$
    $\mathcal{P}_{k-1} \leftarrow \mathcal{P}_k \cup \{CO_r\} - \{CO_p, CO_q\}$
    Update $DC(CO_r, CO_j)$ where $CO_j \in \mathcal{P}_{k-1}$
  end for
  $\mathcal{PH} \leftarrow \{\mathcal{P}_2, \ldots, \mathcal{P}_{|\Psi|}\}$
  return $\mathcal{PH}$
end function
function ShallowMapMask($\mathcal{P}, \mathcal{A}$)
  for $k = 1$ to $|\mathcal{A}|$ do
    $\hat{A}^k_{i,j} = \begin{cases} A^k_{i,j}, & \text{if } \frac{\partial \mathcal{F}_c(I_o)}{\partial A^k_{i,j}} > 0 \\ 0, & \text{otherwise} \end{cases}$
 end for
  for $k = 1$ to $|\mathcal{P}|$ do
    $M_k \leftarrow ReLU(\sum_{\hat{A}^i \in CO_k} Up(\hat{A}^i))$
  end for
  return $\{M_1, \ldots, M_{|\mathcal{P}|}\}$
end function
function FinalMapMask($\mathcal{P}, \mathcal{A}$)
  for $k = 1$ to $|\mathcal{P}|$ do
    $M_k \leftarrow ReLU(\sum_{A^i \in CO_k} w^c_i \times Up(A^i))$
  end for
  return $\{M_1, \ldots, M_{|\mathcal{P}|}\}$
end function
Given the target layer $\mathcal{LS} = \{1, \ldots, L\}$ for $\mathcal{F}_c(I_o)$, obtain feature map $\mathcal{AS} = \{\mathcal{A}_1, \ldots, \mathcal{A}_L\}$ and gradient set $\mathcal{GS} = \{\mathcal{G}_1, \ldots, \mathcal{G}_L\}$
$\mathcal{I} \leftarrow \{\}, \mathcal{M} \leftarrow \{\}, \mathcal{M}' \leftarrow \{\}$
for $l = 1$ to $L$ do
  $\mathcal{PH}_l \leftarrow$ BottomUpClustering($\Psi_l$)
  $\mathcal{P}_l \leftarrow \underset{\mathcal{P}_k \in \mathcal{PH}_l}{\text{argmax}} DSRA(\mathcal{P}_k)$
  if $l < L$ then
    $\mathcal{M}_l \leftarrow$ShallowMapMask($\mathcal{P}_l, \mathcal{A}_l$)
  else $l < L$
    $\mathcal{M}_l \leftarrow$FinalMapMask($\mathcal{P}_l, \mathcal{A}_l$)
  end if
  $\mathcal{M} \leftarrow \mathcal{M} \cup \mathcal{M}_l$
end for
for $k = 1$ to $|\mathcal{M}|$ do
  $M'_k \leftarrow \frac{Denoise(M_k,\theta) - \min(Denoise(M_k,\theta))}{\max(Denoise(M_k,\theta)) - \min(Denoise(M_k,\theta))}$
  $\mathcal{I} \leftarrow \mathcal{I} \cup \{M'_k \odot I_o\}, \mathcal{M}' \leftarrow \mathcal{M}' \cup \{M'_k\}$
end for
$MIC(I_k) \leftarrow \mathcal{F}_c(I_k) - \mathcal{F}_c(X_b)$
$\alpha^c_k \leftarrow \frac{exp(MIC(I_k))}{\sum_l exp(MIC(I_k))}$
$\mathcal{L}^c_{Collection-CAM} \leftarrow ReLU(\sum_k \alpha^c_k M'_k)$
```

We consider three model architectures: VGG19, ResNet18, and ResNet101. For ImageNet1k, we use a pre-trained model available in the torchvision library. For CUB-200-2011 and UC Merced, we trained VGG19, ResNet18, and ResNet101 by ourselves. Specifically, we finetune the final fully connected layer with an SGD optimizer by setting the initial learning rate as 0.01. In addition, we decay the learning rate



(a)

(b)

FIGURE 4. Attention maps, deletion curve and insertion curve generated by Grad-CAM, Score-CAM and Collection-CAM for sampled image from (a) ImageNet1k Dataset (b) CUB-200-2011 dataset.

by 0.1 at every 30 epochs. We train the models for a total of 95 epochs. We resize ImageNet1k and CUB-200-2011 as (224 × 224 × 3), while resize UC Merced as (256 × 256 ×3). Also, we set mean vector [0.485,0.456,0.406] and standard deviation vector [0.229, 0.224, 0.225]. Also, we set baseline input X_b and denoise degree θ to a zero matrix of size corresponding to the input image and 0.1, respectively.

Evaluation Metric: We evaluate the saliency methods in terms of faithfulness, localization ability, and quantitative aspects of computation overhead. When modifying the original input image based on the attention map, objective faithfulness is measured through a change in the target score. We employ metrics such as Average Drop(AD) and Average Increase(AI) [10], Insertion and Deletion [12]. AD and AI are defined as:

$$AD = \frac{100}{N}\sum_{i=1}^{N}\frac{max(0, Y_i^c - O_i^c)}{Y_i^c} \tag{26}$$

$$AI = \frac{100}{N}\sum_{i=1}^{N} Sign(Y_i^c > O_i^c) \tag{27}$$

where N refers to the number of data instances that make up the dataset. Y_i^c refers to the softmax output value for the target class c of the ith image. O_i^c refers to the softmax output value when the input image is masked by the generated attention map. $Sign(\cdot)$ is an indicator function, which returns 1 if the input is *True* and 0 if it is *False*. Furthermore, we perform deletion and insertion test proposed in [12] to supplement AD and AI. *deletion* measures the reduction in the confidence score of the target class when removing pixels from the original image according to the generated attention map's intensity. On the contrary, *Insertion* measures the increase in confidence score of the predicted class when introducing the image pixels to the baseline matrix according to the saliency map. While the small area under the curve(AUC) and sharp drop indicate a good explanation for the *Deletion* curve, the large AUC and rapid increase indicate an excellent visual explanation for the *Insertion* curve. In this work, we introduce or remove 224 × 8.3.57%) or 256 × 8.3.13%) pixels to draw the curves. Examples of the curves are given in Figure 4. *Overall* enables a comprehensive understanding of deletion and insertion results, which is defined as:

$$Overall = AUC(Insertion) - AUC(Deletion) \tag{28}$$

To measure the localization ability of each saliency method, we utilize Proportion [13], which is defined as follows:

$$Proportion = 100 \times \frac{\sum L^c_{(i,j)\in bbox}}{\sum L^c_{(i,j)\in bbox} + \sum L^c_{(i,j)\notin bbox}} \tag{29}$$

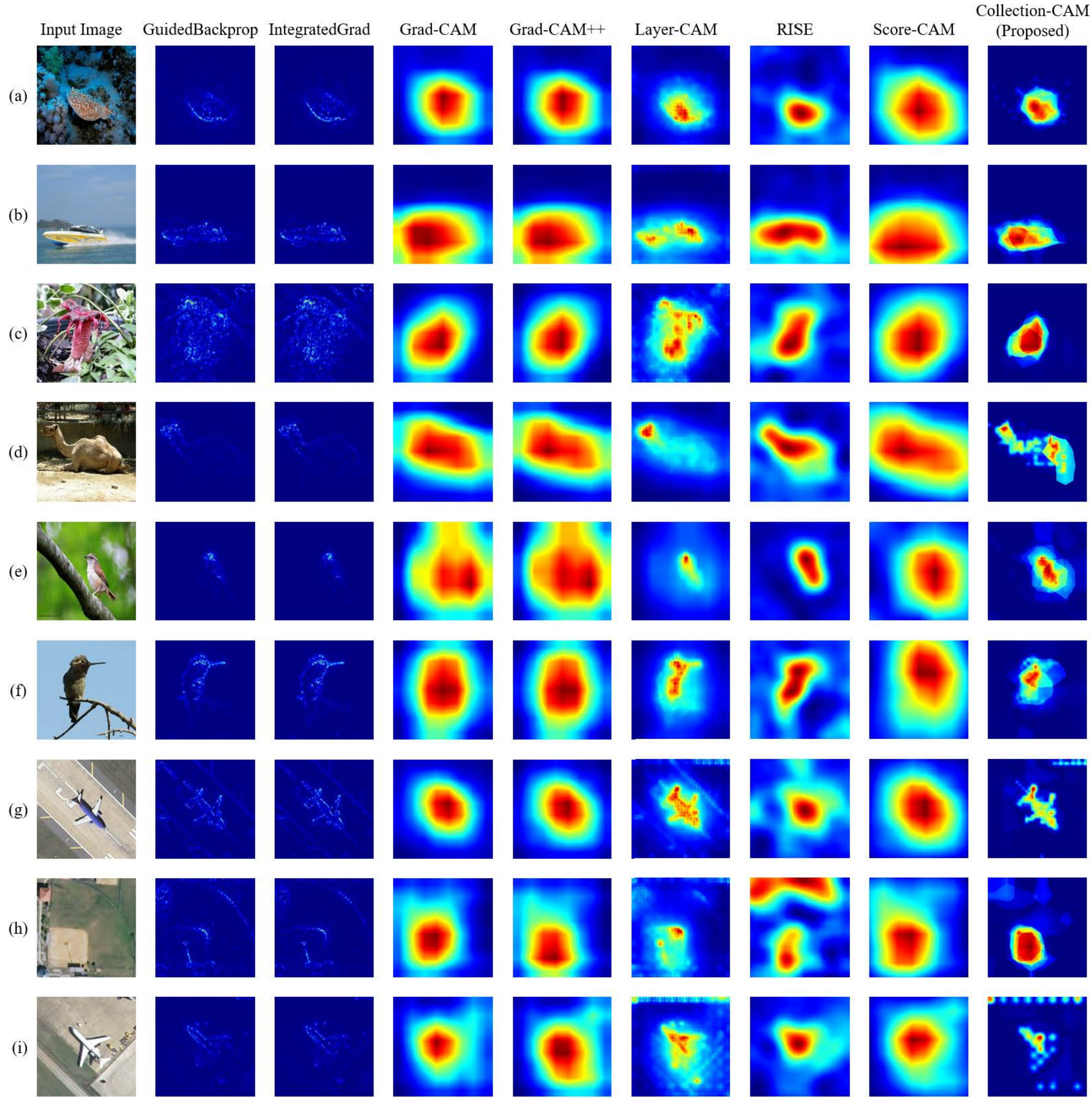


FIGURE 5. Visualization of saliency maps(without overlapping over the input image) on ResNet18 generated by Collection-CAM and baselines. (a) to (d): ImageNet1k, (e),(f): CUB-200-2011, (g),(h),(i): UC merced dataset.

Equation 29 implements the input image as a binarization method that allocates 1 to the inner area and 0 to the outer area, considering how much of the energy of the saliency map flows into the bounding box of the target class.

A. QUALITATIVE EVALUATION VIA VISUALIZATION

In this part, we start with visualizing heatmaps generated by applying our proposed Collection-CAM and the state-of-the art baselines(Guided Backpropagation [15], Integrated Gradients [16], Grad-CAM [9], Grad-CAM++ [10], Layer-CAM [11], RISE [12] and Score-CAM [13] on sample images from different datasets to compare them qualitatively. In the visual examination for the individual attention map, we expect that a saliency method with high explainability has following characteristics:

- It has a low visual noise level.
- Its highlighted area corresponds to the target object's location.
- It reflects the change of the target class well.
- When an input image with multiple objects belonging to the same class is given as input, it locates them well.

In Figure 5, we show heatmaps for sample images from ImageNet1k [23], CUB-200-2011 [25] and UC Merced

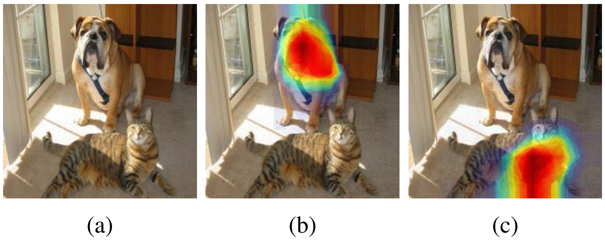


FIGURE 6. Inspection of class discriminability for Collection-CAM. (a) Input image (b) Saliency map with target class 'bull mastiff' (c) Saliency map with target class 'tabby cat'.

[24] dataset on ResNet-18 [2]. Gradient-based methods, Guided Backpropagation, and Integrated Gradients capture target objects' outlines well. However, their heatmaps appear in the form of scattered points and thus contain much noise. Grad-CAM, Grad-CAM++, and Score-CAM generate smooth attention maps. However, they attend to a much larger region than the target object region. Although RISE seems to generally attend to smaller areas than previously described methods, its explainability depends on how well the random mask generation proceeds per the target object. In Figure 5 (h), baseball diamond class, while other methods attend the infield part, RISE attends outfield and contains much noise around the region with high intensity. Layer-CAM synthesizes an attention map similar to the shape of the target object. Nonetheless, its high-intensity region is much small and scattered. Our proposed Collection-CAM attends the target object region with strong intensity. However, like in Figure 5 (i), Collection-CAM sometimes generates a saliency map containing background noises. In the figure, noises with high intensity are accompanied in the top and the bottom part, where the airplane object is not located. This phenomenon is also observed in the saliency map of Layer-CAM, which utilizes shallow feature maps and final feature maps to synthesize the visual explanation. It seems that these visual noises are caused by the mask generated from the shallow layer. Although it sometimes accompanies noises from the shallow masks, the saliency map generated by Collection-CAM captures the object region well.

In Figure 6, we demonstrate that Collection-CAM can discriminate and visualize objects belonging to different classes in the input image. ResNet-101 model classifies the input image as class 'bull mastiff' with a confidence score of 0.582 and class 'tabby cat' with a confidence score of 0.051. As Collection-CAM accurately locates both two different objects, we can confirm its ability to discriminate different objects. This discrimination ability is attributed to the generation of each mask and the application of corresponding weight being relevant to the response to the target class. Therefore, it is reasonable to expect that our method discriminates against different objects in the input image.

Also, in Figure 7, we show that Collection-CAM's localization ability for multiple objects outperforms other methods. In Figure 7, we see that Score-CAM, Layer-CAM, and Collection-CAM capture the location of multiple objects. However, like in Figure 5, Score-CAM attends a much larger area than the objects' region. While Layer-CAM produces saliency maps that have scattered strong intensity pixels, Collection-CAM generates a more concentrated attention map compared to other methods. The reason for Collection-CAM's better explainability is: 1) A collection-wise mask is produced based on the confidence score of the target class and 2) The weight used to assemble them is also obtained based on the contribution to the confidence score of the target class.

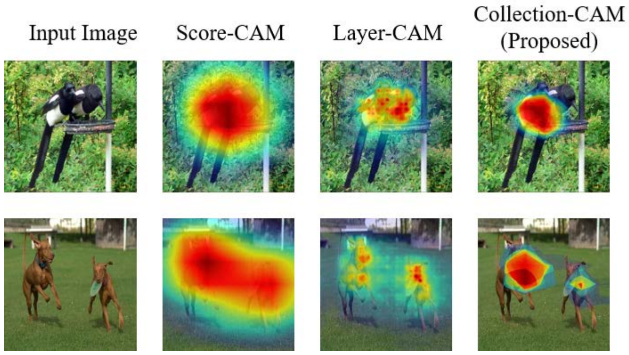


FIGURE 7. Inspection of location capability for multi-objects.

B. SALIENCY MAP GENERATION TIME EVALUATION

We first analyze the computational time model of Grad-CAM [9], Grad-CAM++ [10], Layer-CAM [11], Score-CAM [13], RISE [12] and the proposed Collection-CAM, which are observed with their producing smooth saliency maps in visualization results. We denote the layer set of the deep network ϕ as $\mathcal{LS}^{\phi} = \{CS_1^{\phi}, \ldots, CS_{fin}^{\phi}\}$, the computation time of the forward propagation for a single input image as FP^{ϕ}, and the computation time of backward-propagation to ith layer, namely channel set $CS_i^{\phi} \in \mathcal{LS}^{\phi}$ as $BP^{CS_i^{\phi}}$. Then, we approximate the computation time model of Grad-CAM and Grad-CAM++ as $FP^{\phi} + BP^{CS_{fin}^{\phi}}$. Meanwhile, the computation time model of Layer-CAM, which requires gradient for $\forall CS_i^{\phi} \in \mathcal{LS}^{\phi}$, is approximated as $FP^{\phi} + BP^{CS_1^{\phi}}$. On the other hand, RISE and Score-CAM, which are classified as region-based saliency methods, perform forward propagation by the number of randomly generated random masks $|RM|$ and the number of channels $|CS_{fin}^{\phi}|$ of the final layer, respectively. Besides, RISE requires random mask generation time $|RM| \times MG$. Thereby we approximate computation time model for Score-CAM and RISE as $|CS_{fin}^{\phi}| \times FP^{\phi}$ and $|RM| \times (FP^{\phi} + MG)$, respectively. Finally, our proposed Collection-CAM performs forward propagation as many as the sum of the identified collection number for each layer $\sum_{i=1}^{fin} |\mathcal{P}_i|$. Denoting collection identification time as CT, then the approximated computation time model for the Collection-CAM is $\sum_{i=1}^{fin} |\mathcal{P}_i| \times FP^{\phi} + CT + BP^{CS_1^{\phi}}$.

In Table 2, we report the average execution time when the individual saliency method generates an attention map for a single input image using a single NVIDIA RTX 3080 GPU. Grad-CAM, Grad-CAM++, and Layer-CAM show the shortest average execution time for all listed combinations

TABLE 2. Execution time to generate a saliency map according to deep model and input image resolution.

Setting	Running Time (ms)					
	Grad-CAM	Grad-CAM++	Layer-CAM	Score-CAM	RISE	Collection-CAM (Proposed)
ResNet18, (224×224×3)	13.10	13.72	15.73	1176.12	16328.82	54.15
ResNet18, (256×256×3)	12.96	13.69	15.46	1466.22	22171.15	59.83
ResNet101, (224×224×3)	62.23	62.21	66.55	20584.98	26865.53	322.59
ResNet101, (256×256×3)	58.86	59.09	63.43	20320.01	34880.05	350.55
VGG19, (224×224×3)	18.83	18.16	18.05	1649.61	28003.38	92.63
VGG19, (256×256×3)	20.22	21.04	22.44	2020.42	40003.17	111.38

TABLE 3. Faithfulness evaluation results. (Lower is better for average drop and deletion. higher is better in average increase, insertion and overall. The best figures are in bold.)

Deep Model	Metric	Grad-CAM	Grad-CAM++	Layer-CAM	Score-CAM	RISE	Collection-CAM (Proposed)
ResNet18	AD (%)	34.20	37.80	32.15	32.09	36.98	**25.78**
	AI (%)	27.30	24.00	30.70	29.70	28.60	**37.00**
	Deletion (%)	8.31	8.72	**6.51**	8.60	7.50	8.23
	Insertion (%)	37.61	36.28	38.74	36.23	36.14	**40.83**
	Overall (%)	29.30	27.57	32.24	27.63	28.64	**32.60**
ResNet101	AD (%)	22.24	24.40	25.91	18.99	25.49	**16.73**
	AI (%)	35.35	32.20	33.10	37.60	32.50	**43.25**
	Deletion (%)	12.30	12.79	**10.01**	12.96	11.09	12.75
	Insertion (%)	52.68	51.53	52.69	51.99	50.35	**55.87**
	Overall (%)	40.37	38.74	42.68	39.03	39.26	**43.11**
VGG19	AD (%)	24.64	28.85	22.03	19.15	25.59	**13.78**
	AI (%)	34.65	28.50	40.60	39.85	34.85	**48.20**
	Deletion (%)	9.29	10.54	**8.28**	10.24	9.48	9.17
	Insertion (%)	50.70	47.39	48.03	48.67	46.91	**51.70**
	Overall (%)	41.41	36.86	39.74	38.43	37.43	**42.52**

TABLE 4. Localization ability evaluation on energy-based pointing game. (Higher is better. The best figures are in bold.)

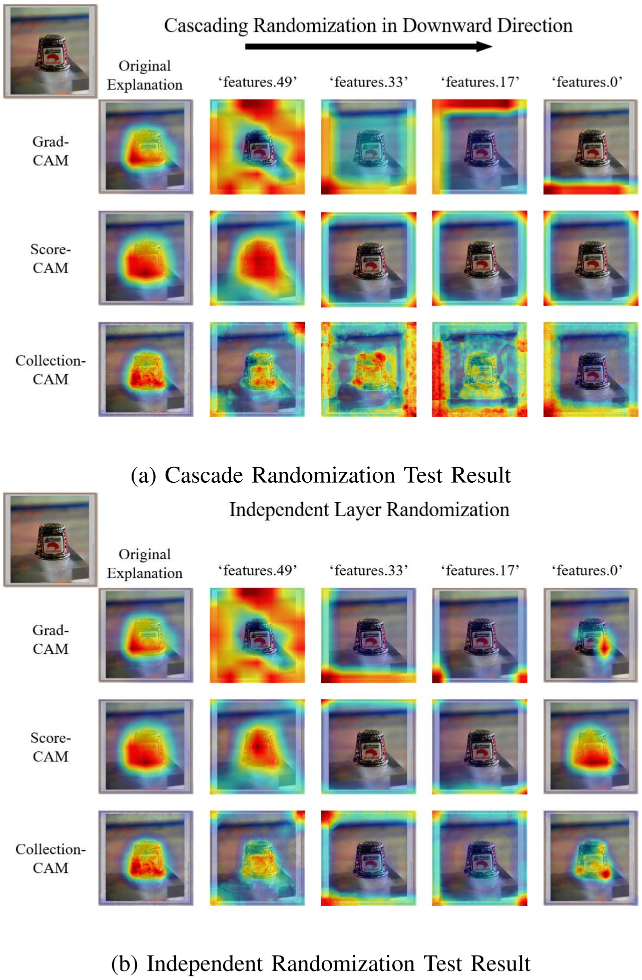
Deep Model	*Proportion (%)*					
	Grad-CAM	Grad-CAM++	Layer-CAM	Score-CAM	RISE	Collection-CAM (Proposed)
ResNet18	49.01	46.43	49.21	42.54	43.73	**60.66**
ResNet101	49.63	47.42	49.85	42.50	46.29	**60.90**
VGG19	52.40	51.14	50.97	52.28	46.30	**61.37**

of (Deep model, Resolution), and there is no significant change in execution time depending on the difference in input image resolution. This is because their attention map generation mechanism utilizes feature maps and gradient-derived weights, thereby requiring only single execution of forward propagation and backward propagation. In contrast, Score-CAM and RISE, which are classified as region-based saliency methods, require a much longer time for attention map generation. This is because Score-CAM requires repetitive forward propagation proportional to the number of feature maps from the target layer(i.e. ResNet18:512, ResNet101:2048, VGG19:512), as can be observed from the computation time models. Similarly, RISE requires a processing time proportional to its large number of masks(>4000). Compared to Score-CAM and RISE, Collection-CAM achieves a significant reduction in the execution time(up to 98.27% reduction from Score-CAM, 99.73% reduction from RISE) by grouping similar feature maps into a collection.

C. FAITHFULNESS EVALUATION

In this part, we randomly sample a total of 2000 image data instances from ImageNet1k, CUB-200-2011, and UC Merced to investigate the faithfulness of explanation by measuring metrics such as AD, AI, and deletion and insertion test results on ResNet18, ResNet101, and VGG19. Considered saliency methods are: Grad-CAM [9], Grad-CAM++ [10], Layer-CAM [11], RISE [12], Score-CAM [13] and Collection-CAM.

As shown in Table 3, Collection-CAM outperforms other methods for various deep model setup. A good performance on the faithfulness evaluation suggests that Collection-CAM not only reduces the execution time of conventional region-based saliency methods but also it can



(a) Cascade Randomization Test Result

(b) Independent Randomization Test Result

FIGURE 8. Sanity check result.

successfully find out the most noticeable parts of the target objects by revealing the decision-making process of the CNN that is currently used.

D. LOCALIZATION ABILITY EVALUATION

In addition to the faithfulness of explanation, we evaluate the localization ability of attention maps generated by different methods through *Proportion*. The testing set is constructed with 500 data randomly sampled from ImageNet1k Validation Split and CUB-200-2011 Val Split both of them provide ground-truth bounding box annotations. At this time, the final value for each technique is expressed as an average percentage and summarized in Table 4. In terms of *Proportion*, the proposed Collection-CAM shows at least 1.17 times up to 1.43 times better performance than other methods, which proves that the noise of the proposed saliency method is much smaller than other methods. Particularly, we observe that Collection-CAM attends to different target objects with more than 60% energy concentrated on various base deep model setups. Gradient-based methods such as Guided Backpropagation [15] and Integrated Gradients [16] are excluded from the baseline methods as their attention maps are represented as edge form as shown in Figure 5, which is far from other methods.

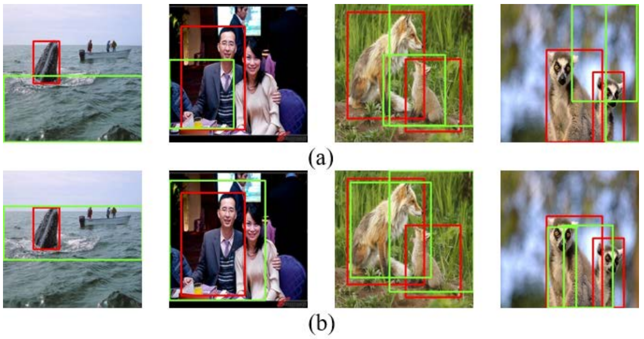


FIGURE 9. Visualization of object proposal on selected images with (a) selective search and (b) selective search+Collection-CAM. Green rectangles and red rectangles refer to object proposals and GT bounding boxes, respectively.

E. SANITY CHECK

We verify whether the Collection-CAM is susceptible to the model parameter through the model parameter randomization test proposed in [26]. To this end, we performed two randomization tests to observe changes in the generated saliency map in the independent layer randomization setting, which randomizes the cascading randomization from the last convolutional layer in the downward direction and fixes the single layer one by one, and the rest to the original model parameter state. As can be seen in Figure 8, in the case of Collection-CAM, it is sensitive to changes in the CNN model parameters to generate different saliency maps, which can be considered to pass the sanity check accordingly.

F. GENERATING OBJECT PROPOSAL

Object detection requires many precise annotations for ground-truth(GT) bounding boxes, but it requires significant human labor to manually prepare such a dataset. To overcome this difficulty, a series of studies uses weakly supervised object detection(WSOD) based on selective search [27]. In this part, we show that the proposed method can be used to improve the quality of object proposals in WSOD. The procedure of object proposal, where Collection-CAM is applied simply, is as follows. First, we generate a saliency map $L^{c_i^*}$ for a image-level label $c_i^* \in C$. Then we obtain a set PL which contains peak intensity pixel positions of $L^{c_i^*}$. Finally, we take the candidate with the highest confidence score in object proposal candidates containing peak $pl \in PL$, as object proposal. As shown in Figure 9, object proposals generated by selective search do not fit well into GT bounding boxes for both single and multi-object images. In contrast, using Collection-CAM together shows object proposals with better quality. This is because determining object proposal only by the confidence score ignores the location information of the object. On the other hand, when Collection-CAM is used with selective search, it is more effective in generating object proposals than using selective search solely as it enables to use of location information of parts considered as objects in the deep network.

V. CONCLUSION

In this paper, we propose the Collection-CAM that generates a saliency map using the feature map of the feature pyramid in a computationally efficient manner. The mechanism of the operation of the Collection-CAM is to combine them with Collection to generate an initial mask, calculate the contribution of each initial mask, and weigh accordingly in a positive manner to generate a saliency map with a linear combination of each mask. Various experiments with multiple deep models (ResNet18, ResNet101, and VGG19) and multiple datasets (ImageNet1k, CUB-200-2011, and UC Merced) demonstrate that the proposed Collection-CAM not only reduces the computation of existing region-based saliency methods by a large margin but also provides enhanced explainability in terms of the faithfulness and the localization ability.

REFERENCES

[1] D. Silver, A. Huang, and C. J. Maddison, "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778.

[3] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," 2014, *arXiv:1406.1078*.

[4] H. Kim, D. C. Jung, and B. W. Choi, "Exploiting the vulnerability of deep learning-based artificial intelligence models in medical imaging: Adversarial attacks," *J. Korean Soc. Radiol.*, vol. 80, no. 2, pp. 259–273, 2019.

[5] S. Lapuschkin, A. Binder, G. Montavon, K.-R. Mü ller, and W. Samek, "Analyzing classifiers: Fisher vectors and deep neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2912–2920.

[6] G. Ras, N. Xie, M. Van Gerven, and D. Doran, "Explainable deep learning: A field guide for the uninitiated," *J. Artif. Intell. Res.*, vol. 73, pp. 329–397, Jan. 2022.

[7] Y. Ji, H. Zhang, Z. Zhang, and M. Liu, "CNN-based encoder–decoder networks for salient object detection: A comprehensive review and recent advances," *Inf. Sci.*, vol. 546, pp. 835–857, Feb. 2021.

[8] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2921–2929.

[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, "Grad-CAM: Visual explanations from deep networks via gradient-based localization," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 618–626.

[10] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. IEEE Winter Conf. Appl. Comput. Vis. (WACV)*, Mar. 2018, pp. 839–847.

[11] P.-T. Jiang, C.-B. Zhang, Q. Hou, M.-M. Cheng, and Y. Wei, "LayerCAM: Exploring hierarchical class activation maps for localization," *IEEE Trans. Image Process.*, vol. 30, pp. 5875–5888, 2021.

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

[13] H. Wang, Z. Wang, M. Du, F. Yang, Z. Zhang, S. Ding, P. Mardziel, and X. Hu, "Score-CAM: Score-weighted visual explanations for convolutional neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2020, pp. 24–25.

[14] A. Kapishnikov, T. Bolukbasi, F. Viegas, and M. Terry, "XRAI: Better attributions through regions," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 4948–4957.

[15] J. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," in *Proc. ICLR (Workshop Track)*, 2015, pp. 1–14.

[16] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 3319–3328.

[17] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, "Top-down neural attention by excitation backprop," *Int. J. Comput. Vis.*, vol. 126, no. 10, pp. 1084–1102, Oct. 2018.

[18] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," 2017, *arXiv:1706.03825*.

[19] D. Balduzzi, M. Frean, L. Leary, J. Lewis, K. W.-D. Ma, and B. McWilliams, "The shattered gradients problem: If resnets are the answer, then what is the question?" in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 342–350.

[20] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vis.*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[21] A. Vattani. (2009). *The Hardness of K-Means Clustering in the Plane*. [Online]. Available: https://cseweb.ucsd.edu/~avattani/papers/kmeans_hardness.pdf

[22] J. H. Ward, Jr., "Hierarchical grouping to optimize an objective function," *J. Amer. Statist. Assoc.*, vol. 58, no. 301, pp. 236–244, 1963.

[23] O. Russakovsky, "Imagenet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[24] Y. Yang and S. Newsam, "Bag-of-visual-words and spatial extensions for land-use classification," in *Proc. 18th SIGSPATIAL Int. Conf. Adv. Geographic Inf. Syst. (GIS)*, 2010, pp. 270–279.

[25] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The caltech-ucsd birds-200–2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[26] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 31, 2018, pp. 1–5.

[27] J. R. R. Uijlings, "Selective search for object recognition," *Int. J. Comput. Vis.*, vol. 104, no. 2, pp. 154–171, Feb. 2013.

YUNGI HA received the B.S. degree in electronic and electrical engineering from Sungkyunkwan University, Suwon, South Korea, in 2013, and the M.S. degree from KAIST, Daejeon, South Korea, in 2014, where he is currently pursuing the Ph.D. degree in electrical engineering. His research interests include online learning and efficient resource management in heterogeneous computing environment.

CHAN-HYUN YOUN (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees in electronics engineering from Kyungpook National University, Daegu, South Korea, in 1981 and 1985, respectively, and the Ph.D. degree in electrical and communications engineering from Tohoku University, Japan, in 1994. Before joining the University, from 1986 to 1997, he was the Head of the High-Speed Networking Team, KT Telecommunications Network Research Laboratories. Since 1997, he has been a Professor at the School of Electrical Engineering, Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, where he was an Associate Vice-President of office of planning and budgets, from 2013 to 2017. He is also the Director of the Grid Middleware Research Center and the XAI Acceleration Technology Research Center, KAIST, where he is developing core technologies that are in the areas of high performance computing, explainable AI system, satellite imagery analysis, and satellite onboard computing with deep learning acceleration system. He wrote a book on *Cloud Broker and Cloudlet for Workflow Scheduling* (Springer, 2017). He was selected to the inaugural class of the IEEE Computer Society Distinguished Contributor, in 2021. He was the General Chair for the 6th EAI International Conference on Cloud Computing (Cloud Comp 2015), KAIST, in 2015. He was also a Guest Editor of the IEEE Wireless Communications, in 2016, and served as a TPC member for many international conferences.