Contents lists available at ScienceDirect

Neurocomputing

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

Understanding contributing neurons via attribution visualization

Rui Shi [a], Tianxing Li [a], Yasushi Yamaguchi [b,*]

[a] *Faculty of Information Technology, Beijing University of Technology, Beijing 100124, China*
[b] *Department of General Systems Studies, The University of Tokyo, Tokyo 153-8902, Japan*

ARTICLE INFO

Article history:
Received 1 February 2023
Revised 29 May 2023
Accepted 21 June 2023
Available online 25 June 2023
Communicated by Zidong Wang

Keywords:
Convolutional neural network
Feature visualization
Feature semantics

ABSTRACT

Understanding contributing neuron features is crucial to explaining convolutional neural network (CNN) decisions. The attribution research provides an effective way to detect contributing neuron features and numerically assign them attribution scores. However, a method to clearly and intuitively represent the implications hidden in neuron attributions is lacking. Attribution scores show the numerical importance of contributing neurons, but the meanings implied by these numerical scores are still not available. To mitigate this gap, we propose an optimization-based visualization method named attribution visualization, which enables an intuitive understanding of neuron attributions. Our approach is distinguished from existing visualization methods by its ability to produce noise-free result, *i.e.*, the ability to remove irrelevant regions from visualizations. We achieve this by introducing an optimizable mask into the visualization process and designing an objective function that simultaneously optimizes the area-constrained mask and visualization. Furthermore, we propose the fractal noise pyramid with diverse and natural frequency spectra as our mask perturbation technique which is key to removing unrelated noise in visualization. We implement several comparisons and user studies with other visual explanations to demonstrate the unique properties of our attribution visualization. We also apply our attribution visualization on two representative CNNs, showcasing its ability to intuitively understand contributing neuron features.

1. Introduction

Convolutional neural networks (CNNs) are being employed on an increasing number of real-world applications, but a good interpretation of network decisions is still lacking [1–3]. Recently, ad hoc and post hoc methods with decision interpretation capabilities have received attention to better understand decision-making. Adhoc interpretation focuses on identifying intrinsic decision modes by building inherently interpretable mechanisms or models, while post hoc methods focus on generating posteriori interpretations for well-structured and pre-trained networks [4,5]. To explain a particular output of a trained CNN, it is crucial to understand contributing neuron feature behaviors under a specific input. In this situation, post hoc attribution methods have been suggested as a good way to detect contributing neurons and assign them attribution scores under a particular output [6,7]. However, there is no effective method to represent the implications behind the numerical attribution scores, where the implications are the meanings implicitly contained in the scores. For example, attribution scores show the numerical importance of contributing neurons (the upper part of Fig. 1), but the meanings implied by these numerical scores are still not available. Users cannot understand what kind of implications are actually reflected by these neuron attribution scores. To this end, we propose a novel viewpoint to achieve a clear and intuitive understanding of contributing neuron features. Specifically, we design an optimization-based attribution visualization method to represent the implications behind attribution scores, as shown in the lower part of Fig. 1.

In the visualization community, there have been many studies aiming at reconstructing neuron or channel features on a feature visualization image which stands for the implication of neurons with a certain degree of diversity [9–13]. These methods concentrated on visualizing the meanings of *feature maps* or any part thereof that can be treated as high-level representations of an input image. On the other hand, *neuron attributions* actually are not representations of the original image, but rather scores indicating the contribution of neuron features. Our research has shown that existing feature visualization methods designed for feature maps are not directly suitable for visualizing the implication of neuron attributions. This is because less-contributing neurons with

[*] Corresponding author.
E-mail address: yama@g.ecc.u-tokyo.ac.jp (Y. Yamaguchi).

https://doi.org/10.1016/j.neucom.2023.126492
0925-2312/© 2023 Elsevier B.V. All rights reserved.

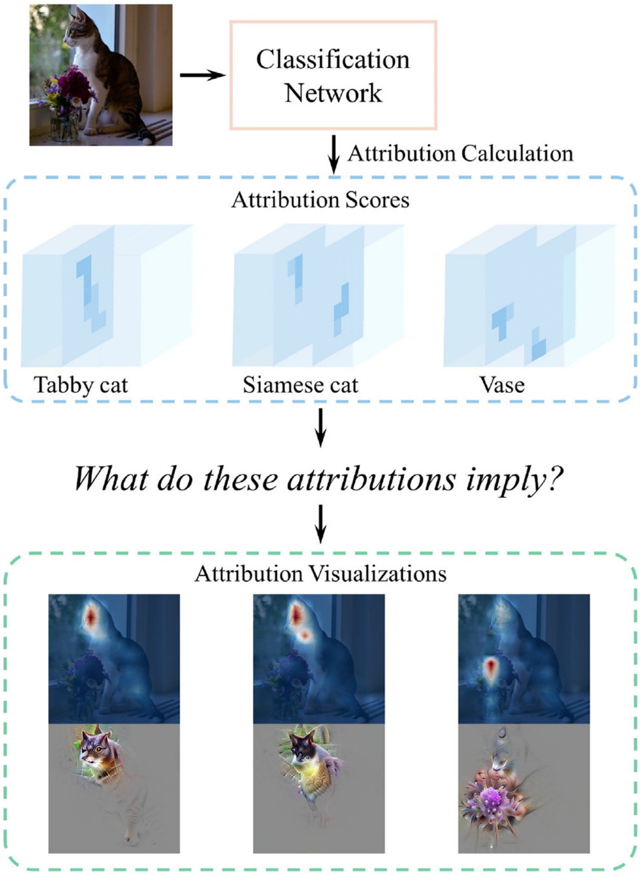


Fig. 1. What do attribution scores imply? The attribution scores calculated using a middle layer of GoogLeNet [8] represent the neuron feature contribution to the classification results. While these scores are statistically valuable, they may not be easily understandable for users what kinds of implications the neuron attributions actually represent. However, by examining the figures presented below, one can gain a better understanding of the implications behind these numerical attribution scores.

zero or small attribution scores can cause meaningless noise in the feature visualization results. For example, as shown in Fig. 2, we visualize neuron attributions related to the "Tabby" class in the mixed4d layer of GoogLeNet [8] using a classical visualization method [10] (slightly modified to accommodate attributions) and our proposal, respectively. Without removing the influence of less-contributing neurons, the noise in the visualization (the left image in Fig. 2), possibly from the irrelevant regions, makes it difficult to understand the contributing features' implications to the "Tabby" class. Therefore, the key challenge in visualizing the implication of neuron attributions lies in developing a method that can accurately represent the features of contributing neurons while eliminating interference from others in the feature space.

Achieving constraints in feature space is non-trivial, because feature space is complex, especially in deep layers, which means we cannot adaptively remove less-contributing neurons during optimization. But attributions show a strong spatial correspondence to image regions, *i.e.*, the combination of contributing neurons spatially corresponds to the target object in image space. That is to say, we can achieve feature-space constraints by introducing a mask in image space. Then, the updating mask can be applied to the updating visualization to filter the noise interference caused by irrelevant or unimportant neurons. With this idea, we introduce the concept of the mask into the visualization process to achieve intuitive and noise-free visualizations of the implications of neuron attributions. As shown in the right image in Fig. 2, the attribution implications associated with the output can be grasped clearly by filtering out noises. The programming implementation of our proposal is available online (https://bit.ly/attr-vis). More specifically, this paper presents the following contributions:

(1) We propose an *attribution visualization* method to represent the implications of neuron attributions, thereby understanding the meanings of contributing neuron features. To eliminate various noise and artifacts generated during the visualization process, we introduce an optimizable mask into visualization and design an objective function of optimizing the mask and the visualization simultaneously.

(2) We find the frequency information of mask perturbations is key to achieving a high-quality mask generation in the case of simultaneous optimization. With this observation, we design a fractal noise pyramid that possesses diverse and natural frequency spectra as real-world images and dynamically apply the fractal noise selected from the pyramid to perturb the visualization image during optimization. Ablation study in Section 4.2 shows our proposal can produce the best visual effect among all tested perturbation alternatives. Further user studies in Section 4.3 also show that our proposal could present semantic information more intuitively than other visual explanations.

2. Related works

2.1. Attribution methods

Attribution methods can be used to understand the relevance between neurons and a particular output. Simonyan et al. [14] used saliency information to explain particular outputs (Saliency). Shrikumar et al. [15] introduced an element-wise multiplication between the signed gradient and the input (GradxInput) as a way of calculating feature contribution scores. Although these vanilla gradient methods can identify the features that can be locally perturbed the least to maximally change the output, they do not help in computing the marginal contribution of a feature. To address this limitation, other methods have been developed that involve designing different backpropagation rules for non-linear operations. Bach et al. [16] proposed Layer-wise Relevance Propagation (LRP) and designed several propagation rules for different network architectures. Shrikumar et al. [17] proposed Deep Learning Important Features (DeepLIFT) Rescale and RevealCancel. However, all above methods break at least one of the self-evident axioms which should be satisfied by any attribution explanation [18–20]. This leads to attribute explanations that seem to be consistent with human intuition but may be unreliable or misleading, and emphasizes the need for researchers to prioritize the reliability of attribution analysis from different sources [21–24].

To mitigate the lack of reliability, some researchers summarized several fundamental axioms as a standard for the design of attribution methods [20,19,18]. Along with this axiomatic idea, the literature on Shapley values [25] in game theory suggests a unique way such that desirable axioms are satisfied [26,20,27,28]. Lundberg et al. [29] proposed Deep Shapley Additive Explanations (DeepSHAP) to estimate Shapley values with a layer-wise chain rule; however, the generalization of the chain rule on Shapley values is not yet clear. Sundararajan et al. [20] proposed Integrated Gradients (IntGrad) that can be regarded as computing Shapley values in infinite games. Chen et al. [30] proposed a Shapley value propagating method to explain a series of networks that is an order of magnitude faster than existing model-agnostic attribution techniques. In this context, we select a recent Aumann–Shapley-based method [28] in our attribution calculation. Resulting neuron attributions

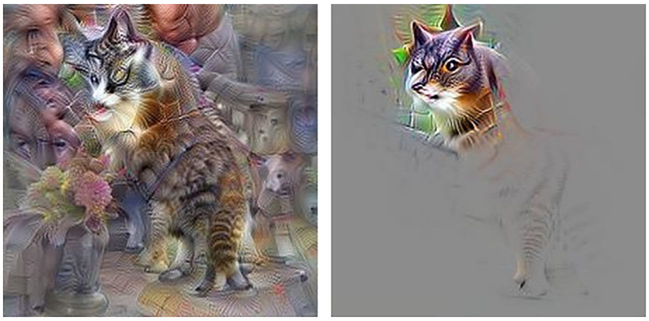


Fig. 2. Left: result generated by modifying [10] to visualize attributions related to the "Tabby" class. Right: our attribution visualization.

$\mathbf{R}^{[\ell]} \in \mathbb{R}^{W \times H \times K}$ stand for the contribution scores of neurons to a particular output in the hidden layer, where W, H, and K are width, height, and channel number of feature maps in the ℓ_{th} layer, *i.e.*, the attribution tensor has the same size with the feature maps.

2.2. Feature visualization methods

Feature visualization is generally based on the fact that a CNN is differentiable with respect to an input image [10]. Thus, derivatives can be used to modify the input image iteratively to look for the kind of input that would cause a certain behavior of neurons. There have been many studies that visualize features by tweaking an image to excite one neuron, one channel in a hidden layer, or specific layer representation [14,9,31–35]. However, these methods of generating a visualization corresponding to a single convolution kernel cannot express the features that are related to a certain behavior under a given input instance. Some studies visualized the implications of feature maps under a given input image or the distribution of intermediate feature maps [36,10,9,12]. Despite this, there are only parts of features in feature maps related to the particular output, the methods without distinguishing the class-targeted features cannot be applied to understand contributing features.

To understand features contributing to the particular output, some methods divide feature maps into several groups according to spatial distribution [11] or neuron attribution factorization [37] to generate class-targeted visualizations. More recently, Singla et al. [13] proposed a detection method to locate contributing or less-contributing neurons, and designed a visualization method to represent these class-targeted neuron features. Although these visualization studies are able to represent the implications of class-targeted features (*i.e.*, contributing features), none of them can filter out the noise interference in visualization results caused by irrelevant or unimportant neuron features as we introduced in Section 1.

3. Attribution visualization

In this section, we start to introduce our attribution visualization method. We first summarize the attribution calculation used in our proposal. Then, we discuss our objective function for attribution visualization and mask area regularization. Finally, we introduce the possible perturbation choices and our fractal noise pyramid. An overview of our proposal is shown in Fig. 3.

3.1. Attribution calculation

To calculate neuron attributions, we use an Aumann–Shapley-based attribution method introduced in [28]. In particular, for a target output, this method computes the attribution scores $\mathbf{R}^{[\ell]} \in \mathbb{R}^{W \times H \times K}$ that represent the importance of neuron features, where ℓ is the layer index, W, H, K are the width, height, and channel number of the feature maps in this layer.

To visualize the attribution scores, we first generate a heatmap by summing the scores along the channel dimension. This heatmap indicates the image regions from which the contributing features are extracted. However, this compression of channels does not reveal the full implications of neuron attributions. To address this, we have developed a visualization method that provides more detailed insights, which we will describe next.

3.2. Objective function for attribution visualization

Our goal is to find a natural-looking visualization to represent the implications of neuron attributions related to the particular output. This can be formulated as a regularized energy minimization problem. Formally, to represent neuron attributions $\mathbf{R}^{[\ell]}$, the loss $\mathscr{L}$ should be maximized as much as possible while ensuring the coordination of spatiality and proportion among neurons by constraining the optimization process. Although there has been a significant body of literature focusing on attribution calculation, in our proposal, we select Aumann–Shapley method to compute $\mathbf{R}^{[\ell]}$ as discussed in Section 2.1. The loss function can be defined as:

$$\mathscr{L}(\mathbf{X}; \mathbf{R}^{[\ell]}) = \sum_{i,j,k}^{W,H,K} \left(f^{[\ell]}(\mathbf{X}) \odot \mathbf{R}^{[\ell]} \right)_{i,j,k}, \tag{1}$$

where $\mathbf{X}$ is the visualization image to be optimized. $\mathbf{R}^{[\ell]}$ stands for the neuron attributions. $f^{[\ell]}(\cdot)$ means the output feature maps in the ℓ_{th} layer. The operator $\odot$ means the Hadamard product. The loss is the sum of the element-wise multiplication of feature maps and neuron attributions. This type of loss functions can be considered as a variation of activation maximization [10]. However, such a function cannot highlight contributing neuron features exactly due to the influence of many irrelevant neurons.

We would like to distinguish neuron features in the visualization image. To achieve this, we introduce the concept of the mask into the visualization loss function to suppress irrelevant neurons. In particular, we want to find a mask $\boldsymbol{M} \in [0,1]^{W^{[0]} \times H^{[0]}}$ where $\boldsymbol{M}(\boldsymbol{n}) = 1$ means that the $\boldsymbol{n}_{th}$ pixel of the visualization image dominate the composed image and $\boldsymbol{M}(\boldsymbol{n}) = 0$ that it does not, $\boldsymbol{n} \in \{1, \ldots, W^{[0]}\} \times \{1, \ldots, H^{[0]}\}$. $W^{[0]}$ and $H^{[0]}$ are width and height of the input image. Note that, the values in the mask are continuous weights. To balance the pixel intensity, we use the mask to induce an image region perturbation operator, denoted as $\boldsymbol{M} \otimes \mathbf{X}$. The detailed discussion of mask perturbation techniques will be introduced in Section 3.4, but for now, it suffices to say that the larger the mask value, the more information of the pixel is preserved, whereas the rest is replaced by noise. We constrain the preserved area ratio of the mask to a fixed value β which can be selected manually or automatically. Then, the goal is to minimize the objective function:

$$\mathbf{X}^*, \boldsymbol{M}^* = \arg\min_{\mathbf{X}, \boldsymbol{M}: f_{mean}(\boldsymbol{M}) = \beta} - \mathscr{L}(\boldsymbol{M} \otimes \mathbf{X}; \mathbf{R}^{[\ell]}), \tag{2}$$

$$\mathscr{L}(\boldsymbol{M} \otimes \mathbf{X}; \mathbf{R}^{[\ell]}) = \sum_{i,j,k}^{W,H,K} \left(f^{[\ell]}(\boldsymbol{M} \otimes \mathbf{X}) \odot \mathbf{R}^{[\ell]} \right)_{i,j,k}, \tag{3}$$

where $\mathbf{X}^*$ is the optimized visualization image. $\boldsymbol{M}^*$ is the optimized mask for determining important regions. f_{mean} is the function to calculate the average of a matrix or a tensor. β is the mask constraint parameter.

In forward propagation, CNNs can obtain the discriminative features to achieve image classification. However, the higher-layer features discard a substantial amount of low-level image features

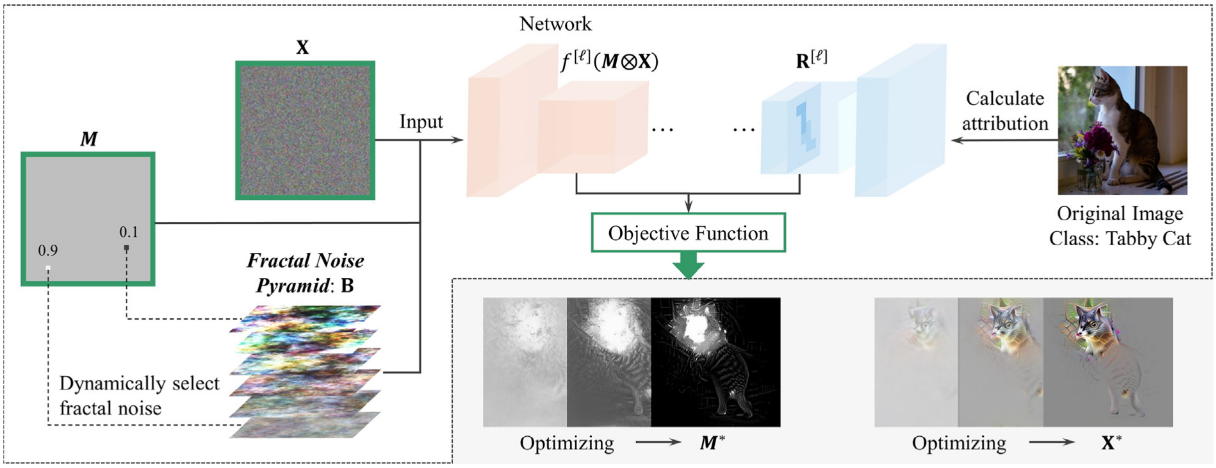

Network
$f^{[\ell]}(\boldsymbol{M}\otimes\mathbf{X})$
X
$\mathbf{R}^{[\ell]}$
M
Input
Calculate attribution
0.1
0.9
Objective Function
Original Image
Class: Tabby Cat
Fractal Noise Pyramid: B
Dynamically select fractal noise
Optimizing → $\boldsymbol{M}^*$
Optimizing → $\mathbf{X}^*$

Fig. 3. The overview of our visualization method. The mask $\boldsymbol{M}$ and the randomly initialized $\mathbf{X}$ marked with bold green borders are variables to be updated simultaneously. Dynamic selection of fractal noise according to mask values can filter out irrelevant information and highlight the meaning of contributing neurons.

such as lines, shapes, angles, and scales that may be not directly relevant to the image classification. Thus, the optimized result $\mathbf{X}^*$ may end up with a nonsensical illusion without regularizing the visualization image. Basically, image regularizers could be divided into three families, *i.e.*, frequency penalization, transformation robustness, and learned priors. Unfortunately, both frequency penalization and learned priors have been shown to discourage legitimate features [34]. Thus, we only consider image regularization techniques of transformation robustness that are often used in dataset enhancement during training consisting of random shearing, padding, jittering, random scaling, and random rotating. Namely, we consider the optimization process:

$$\mathbf{X}^*, \boldsymbol{M}^* = \arg\min_{\mathbf{X},\boldsymbol{M}:f_{mean}(\boldsymbol{M})=\beta} - \mathbb{E}_\tau[\mathscr{L}(f_{trans}(\boldsymbol{M}\otimes\mathbf{X};\tau);\mathbf{R}^{[\ell]})], \tag{4}$$

where $\mathbb{E}$ denotes expectation and τ is a random variable uniformly distributed in the image transformation set. The image regularizer helps to find robust structures and textures of objects during optimization. Because transformation may change the image resolution, padding or cropping to the original size should be the last step before feeding the image into the network. For the sake of clarity in the expression of the following equations, we will omit the image regularizer term in the remainder of the paper, but they are indispensable in generating visualizations.

3.3. Area-constrained mask

Enforcing the area constraint in Eq. (4) is non-trivial; here, we use a regularization term to constrain the mask size by penalizing the mask for which its average deviates from the constraint thresholding value β:

$$\mathbf{X}^*, \boldsymbol{M}^* = \arg\min_{\mathbf{X},\boldsymbol{M}} - \mathscr{L}(\boldsymbol{M}\otimes\mathbf{X};\mathbf{R}^{[\ell]}) + \lambda(f_{mean}(\boldsymbol{M}) - \beta)^2. \tag{5}$$

At a glance, we introduce a new parameter λ to weigh two terms in the objective function. But, the last term does not have an impact on the visualization image $\mathbf{X}$; thus, during optimization, we can simply set λ to increase as the first term increases to achieve the area constraint exactly. To find a β automatically, we use Otsu thresholding method [38] to determine the area ratio adaptively:

$$\boldsymbol{H} = \sum_k^K \mathbf{R}^{[\ell]}_{:,:,k}, \tag{6}$$

$$T^* = \arg\max_T \omega_0(T)\omega_1(T)(\mu_o(T) - \mu_1(T))^2, \tag{7}$$

$$\beta = f_{mean}(\boldsymbol{1}(\boldsymbol{H} > T^*)), \tag{8}$$

where $\boldsymbol{H}$ is the attribution intensity map, which can specify the proportion of contributing neurons. ω_0 and ω_1 are probabilities of two groups that are separated by the thresholding value $T \in [f_{min}(\boldsymbol{H}), f_{max}(\boldsymbol{H})]$. μ_0 and μ_1 are the mean values of the two groups. We set the number of bins for the Otsu method at 200, where the minimum and the maximum of bins are set at the minimum and the maximum of the attribution intensity map $\boldsymbol{H}$. The operator $\boldsymbol{1}(\cdot)$ returns a matrix where the value is 1 when the condition is true. In addition, according to the amount of content we need to display, the mask constraint parameter β can also be selected manually.

3.4. Fractal noise pyramid

In this section, we discuss several common perturbation techniques and define our perturbation operator $\boldsymbol{M}\otimes\mathbf{X}$. The mask here is used to find pixels that do not affect the visualization loss even if these pixels are "deleted". While conceptually simple, there are several problems with this idea. The first one is how to specify a perturbation method to delete information. There are three obvious proxies: blurring the image, replacing the image region with a constant value, and that with random noise. Then these perturbation operators can be defined as follows:

$$(\boldsymbol{M}\otimes\mathbf{X})(\boldsymbol{n}) = \frac{\sum\limits_{\boldsymbol{v}\in\Omega}\kappa(\boldsymbol{n}-\boldsymbol{v};\sigma_{max}(1-\boldsymbol{M}(\boldsymbol{n})))\mathbf{X}(\boldsymbol{v})}{\sum\limits_{\boldsymbol{v}\in\Omega}\kappa(\boldsymbol{n}-\boldsymbol{v};\sigma_{max}(1-\boldsymbol{M}(\boldsymbol{n})))}, \quad \text{where}, \kappa(\boldsymbol{u};\sigma) = e^{-\frac{\|\boldsymbol{u}\|_2}{2\sigma^2}}, \tag{9}$$

$$\boldsymbol{M}\otimes\mathbf{X} = \boldsymbol{M}\odot\mathbf{X} + (\boldsymbol{1} - \boldsymbol{M})C, \tag{10}$$

$$\boldsymbol{M}\otimes\mathbf{X} = \boldsymbol{M}\odot\mathbf{X} + (\boldsymbol{1} - \boldsymbol{M})\odot\mathbf{B}, \tag{11}$$

where $\boldsymbol{M}(\boldsymbol{n})$ is the value of the pixel $\boldsymbol{n}$. Ω is a discrete lattice of the blurring kernel κ defining the kernel size. σ_{max} is the maximum isotropic standard deviation of the kernel. The operator $\|\cdot\|_2$ means 2-norm of a vector. C is the constant value which could be 0 or average color. $\mathbf{B}$ is the random noise with the same size as $\mathbf{X}$. Blurring an image (Eq. (9)) or replacing with a constant value (Eq. (10)) is the most common idea of achieving a mask perturbation. However, in the iterative optimization of visualization, blurring the image of the previous iteration fails to produce stable perturbation to the newly updated visualization, because the visualization itself is also updating in every iteration. As another option, using a constant value may cause the optimization process to fall into a specific mode (*e.g.*, adversarial artifacts), because it is plain in the frequency domain [39]. Thus, we choose pixel replacement with random noise (Eq. (11)) in our visualization method.

Solving the first problem points out the second problem simultaneously, *i.e.*, how to generate the noise. Intuitively, sampling noise from a Gaussian random distribution in the time domain is the most common solution. However, in practice, the visualizations generated based on Gaussian noise often contain discontinuous points and over-saturated colors. According to the study on the influence of frequency information on robustness [40], we find that this problem is due to the fact that the power spectrum of Gaussian noise, which approximately subjects to a uniform distribution, is not common in natural-looking images. Additionally, the uniform distribution can easily lead to adversarial artifacts during visualization optimization.

For random noise generation, there has been another observation in signal processing that many forms of "natural" random images have $\boldsymbol{F}^{-p/2}$ frequency spectra, *i.e.*, their spectra fall off as the inverse of some power of the frequency where the power is related to the fractal dimension. In this context, we generate the noise by simulating natural signals in the frequency domain. In particular, this method first generates random noise in the frequency domain, uses desired $\boldsymbol{F}^{-p/2}$ to adjust the scale, and then performs inverse Fourier transform back to the time domain to obtain a random image that conforms to the spectral representation of natural images.

Although the "natural" spectra provide more reasonable perturbation information, the frequency component is still simple, which also makes the optimization process tend to produce meaningless artifacts. To solve this problem, we further design the *fractal noise pyramid* which enables to select fractal noises with different intensities in the frequency domain to perturb the visualization image according to the mask values. The mask value closing to 0 corresponds to strong noise perturbation; thus, we apply noise in relatively light and correlated sequences with a higher magnitude level, otherwise, perturb visualization image with noise in dark and uncorrelated sequences. With this idea, noise pyramid $\mathbf{B} \in \mathbb{R}^{W^{[0]} \times H^{[0]} \times K^{[0]} \times (D+1)}$ is defined as:

$$\mathbf{B}_d = f_{ift}(g(d,D)), \tag{12}$$

$$g(d,D) = \alpha \mathbf{B}\prime(d,D) \odot \boldsymbol{F}^{-p/2}, \tag{13}$$

where f_{ift} is the inverse Fourier transform function. The number of pyramid layers is $D+1$ with an index $d = 0, \ldots, D$. $\mathbf{B}\prime(d,D)$ is the sampled result from a complex distribution where both the real and imaginary parts are subject to the normal distribution $\mathcal{N}(0,(0.2 - d/(20D))^2)$. Referring to the definitions of pink noise ($p = 2$) and Brownian noise ($p = 4$), we set $p = 3 - d/D$ to reflect the increase in the number of spectral components. $\boldsymbol{F}$ stands for the sample frequencies of the discrete Fourier transform. α is the scaling parameter to adjust the complex coefficients that can be selected based on the image size. The generated noise pyramid $\mathbf{B} \in \mathbb{R}^{W^{[0]} \times H^{[0]} \times K^{[0]} \times (D+1)}$ contains $D+1$ progressively weaker versions of noise images. Then, the perturbation operator can be defined as:

$$(\boldsymbol{M} \otimes \mathbf{X})(\boldsymbol{n}) = (\boldsymbol{M} \odot \mathbf{X})(\boldsymbol{n}) + (1 - \boldsymbol{M}(\boldsymbol{n})) \odot \mathbf{B}(\boldsymbol{n}, \boldsymbol{M}(\boldsymbol{n})). \tag{14}$$

where the last term can be interpreted as that the mask range is divided into $D+1$ intervals, then, the corresponding noise according to the mask value in current pixel $\boldsymbol{n}$ is selected. Finally, the attribution visualization can be optimized using the objective function Eq. (5).

4. Experimental results

To evaluate the major difference between our proposal and previous studies, we first qualitatively compare our attribution visualization with several state-of-the-art visualization methods in Section 4.1. Then, we conduct an ablation study using several alternative mask-perturbation techniques to show better performance of our proposal in Section 4.2. Further, to demonstrate the intuitiveness of our method, we perform user studies comparing with several post hoc visual explanations in Section 4.3. Finally, we show what our visualizations can represent in the last experiment, *i.e.*, attribution visualizations in different layers of GoogLeNet and ResNet-50 [41] and visualizations of grouped attributions in Section 4.4.

The top six images in Fig. 4 licensed under Creative Commons Attributions CC-BY 4.0 that are downloaded from the URL: https://unsplash.com/. The bottom three images in Fig. 4 are selected from Caltech-UCSD Birds 200 (CUB-200) [42] dataset. Our quantitative evaluation experiment utilized these images along with 100 randomly selected images from the ImageNet [43] validation set and 100 bird images from CUB-200. Notably, most of our experiments are performed using a GoogLeNet trained on ImageNet, but the experiments using the bird images are based on a GoogLeNet trained on the CUB-200 dataset.

The optimization algorithm is Adam [44] and the learning rate is decayed exponentially based on loss change. Our image regularizer is empirically set as padding with 12 pixels, jittering 9 pixels, randomly scaling from 0.9× to 1.3×, randomly rotating from -12 to 12 degrees, randomly shearing from -4 to 4 degrees, jittering 4 pixels, and cropping or padding to the original image size. The image padding mode is reflection. The parameter λ to balance the two terms in the objective function Eq. (5) is set at the sum of attributions initially and multiplied by 1.2 at every tenth of the lifetime. We set $D = 5$ to generate a six-layer noise pyramid and apply SVD color decorrelation to the fractal noise to get a color space with better visual effects. The σ_{max} for the perturbation of the

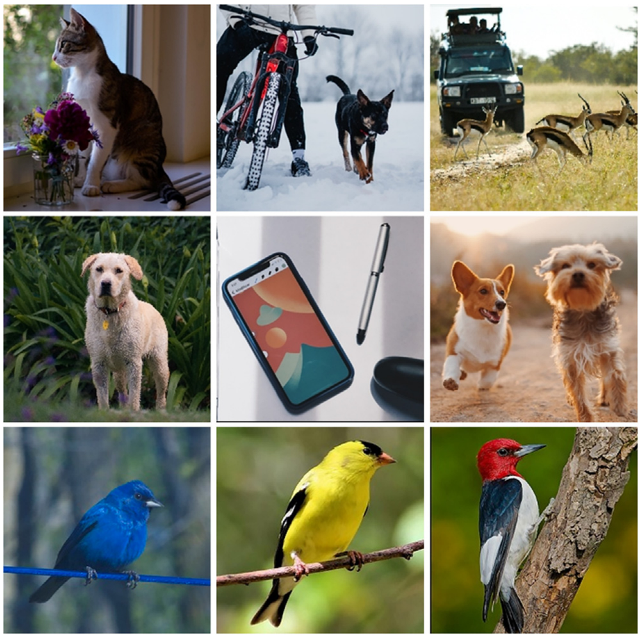


Fig. 4. Image examples used in our experiments. Objects contained in images from left to right: Row 1: tabby cat and vase, Australian kelpie and mountain bike, gazelle and jeep; Row 2: Labrador retriever, cellular telephone and ballpoint, Labrador retriever and cardigan corgi; Row 3: indigo bunting, American goldfinch, red-headed woodpecker.

blurring image is 0.2 and the blurring kernel size is 5 × 5. C is set at 0.5 for the perturbation of replacing with a constant value. Gaussian noise is sampled from the standard normal distribution. The amplitude spectrum results using different perturbation techniques are shown in Fig. 5. We can find only the fractal noise pyramid can produce diverse and natural spectrum among all perturbation techniques. Such advantages are crucial for producing noise-free visualizations, as shown in qualitative comparison results in Section 4.2.

4.1. Comparison of visualization methods

In this section, we compare our visualization method with three state-of-the-art visualization methods of representing feature implications [10,11,13] and our no-mask visualization method. Many previous studies focused on improving visual effects by designing image regularization techniques. However, we directly adopt the most advanced regularizer techniques as discussed in Section 4. For a fair comparison with previous studies, we thus use the same image regularizations for all tested methods. Therefore, the visual effect of visualization results of previous methods we implemented may be slightly different from the results in the original papers. For example, the colors of the visualization results in [13] may look more over-saturated than ours. We believe that a unified visual effect better allows readers to focus on the major difference between our proposal and previous studies, *i.e.*, the ability to generate noise-free visualizations.

Fig. 6 shows the visualizations generated using different visualization methods using a GoogLeNet pre-trained on ImageNet. More experimental results can be found in A. We can find it difficult to distinguish meaningful features in the visualization results from the surrounding noise or artifacts with previous visualizations and our attribution visualization without mask. As an example shown in Fig. 6c, their method [13] distinguishes the feature differentiation between different output classes. But without filtering out irrelevant neuron features during optimization, noise irrelevant to the particular class appears in the visualization results. In the result of the “Tabby” class in Fig. 6c, neurons may be related

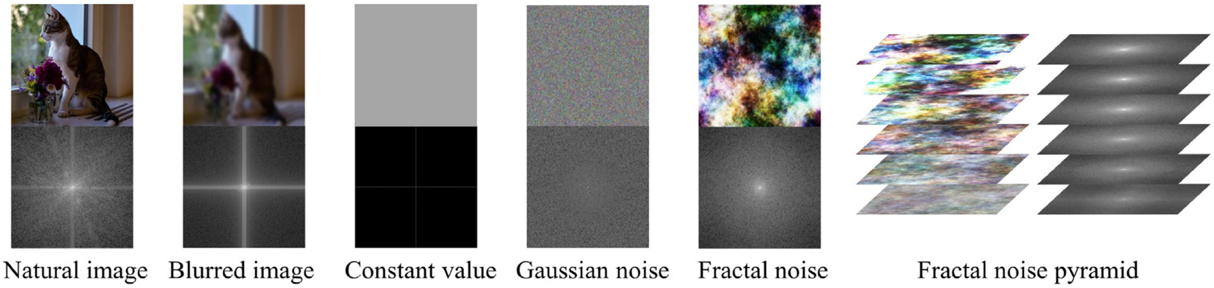


Fig. 5. Amplitude spectrum results using different perturbation techniques. This image is best viewed on screen.

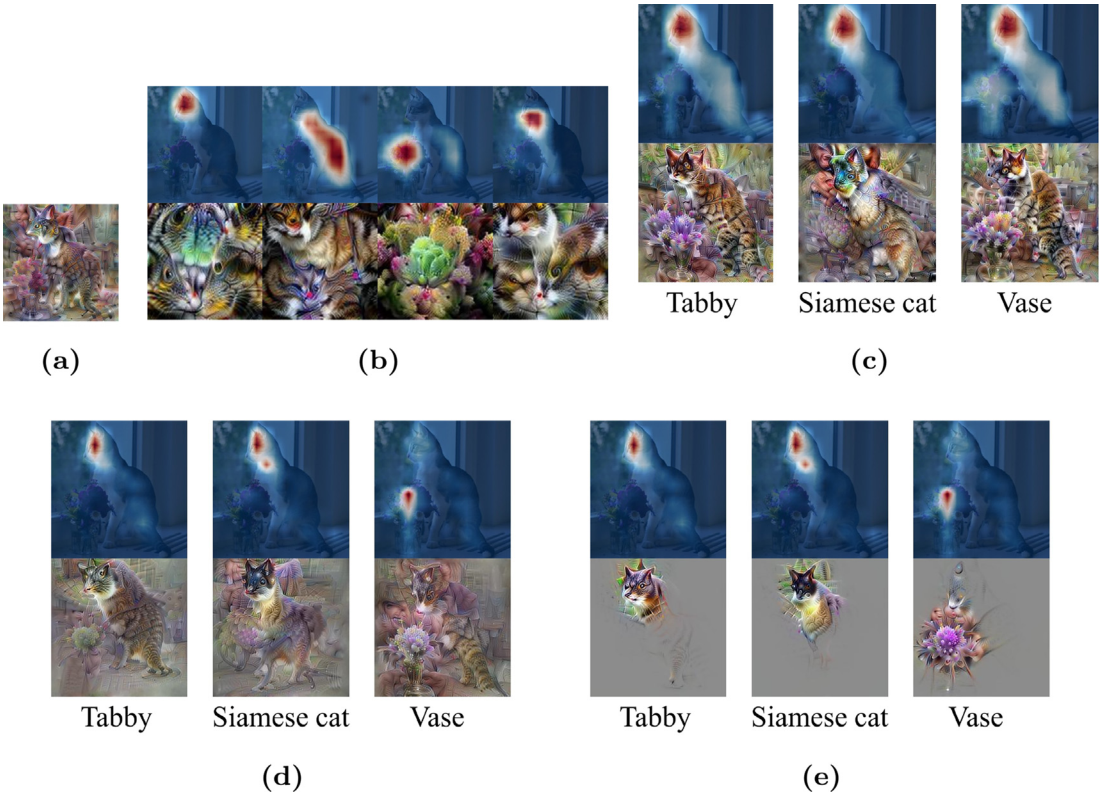


Fig. 6. Visualizations in the mixed4d layer with different methods using the image of the cat and vase in the top left of Fig. 4. (a) Visualizing all feature maps [10]. (b) Visualizing factorized feature maps [11]. (c) Visualizing class-targeted neurons [13]. (d) Visualizing attributions without mask. (e) Visualizing attributions with mask (ours).

to the vase and the background produce a lot of misleading noise, which makes us unable to understand which features really contribute to the cat. Although heatmaps can alleviate this problem, the noise-free visualizations are more intuitive and clearer for analyzing feature implications. For our no-mask case, we apply our visualization technique directly to attributions but remove any mask-related parts, as shown in Fig. 6d. Similar to previous results, the large amount of extraneous information generated prevents us from accurately distinguishing class-targeted contributing features from noise. By proposing the fractal noise pyramid to apply the area-constrained mask to visualization process, our visualizations can remove meaningless noise around the target object.

Fig. 7 shows similar results generated with a GoogLeNet pretrained on CUB-200 using the indigo bunting image in the bottom left corner of Fig. 4. One major benefit of CUB-200 is the inclusion of detailed attributes of the bird in each image. With the attribute information, we further evaluate whether the visualization method is able to demonstrate major attributes. The top five attributes of the indigo bunting are shown in Table 1. While an exact quantification of the assessment may not be available, it is evident that our visualization method is the only one capable of clearly reflecting these attributes.

In addition, the attribution visualizations are inherently class-targeted. Both visualizing feature maps together [10] and visualizing factorized feature maps [11] cannot directly express the relationship between the visualization and the particular output. Note that, if attributions are used to post-process the factorized feature maps, class labels could be assigned to these visualizations. However, since the factorization information has been determined before visualizing, this processing cannot assess the coupling in spatially similar classes. For example, the idea in [11] is to factorize all feature maps according to neuron spatial distributions; but if two cat classes come from the same image region, this method will entangle features related to these two classes. By detecting and visualizing class-targeted neurons [13], Fig. 6c and Fig. 7cx can represent feature meanings related to a particular output. Different from these methods, our attribution visualization is naturally class-targeted, because neuron attributions are essentially neuron contributions to the particular output. As long as the attribution calculation is reliable, our visualization method can accurately represent the meanings of the contributing features.

Table 1
Top attributes of the indigo bunting bird in the bottom left image of Fig. 4.

Attribute Type	Attribute
Bill shape	Cone
Wing color	Blue
Crown color	Blue
Eye color	Black
Upperparts color	Blue

4.2. Ablation study on perturbation techniques

In this section, we evaluate resulting visualizations using our method with different mask perturbation techniques. Different from

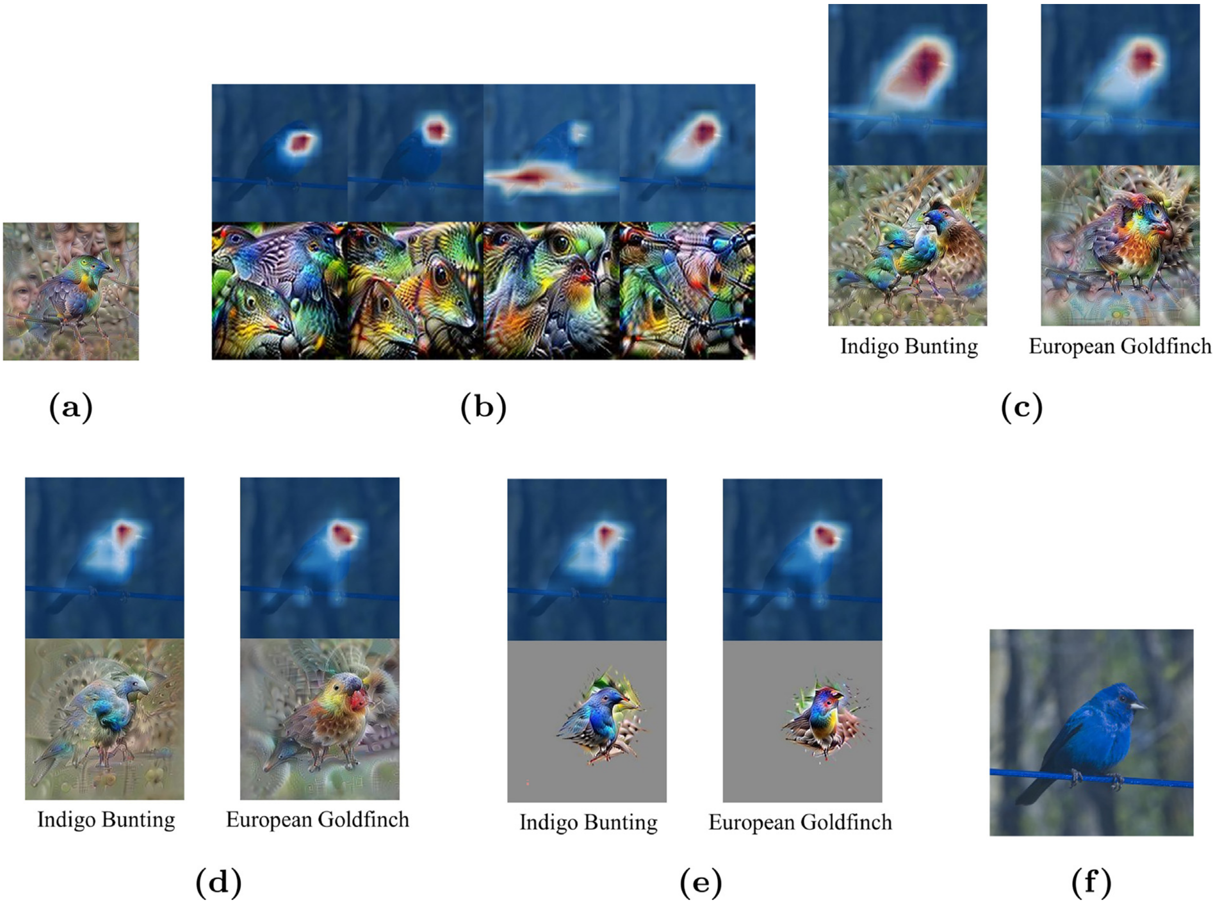


Fig. 7. Visualizations in the mixed4d layer with different methods using the image of the indigo bunting in the bottom left of Fig. 4. (a) Visualizing all feature maps [10]. (b) Visualizing factorized feature maps [11]. (c) Visualizing class-targeted neurons [13]. (d) Visualizing attributions without mask. (e) Visualizing attributions with mask (ours). (f) The original indigo bunting image used in the experiment.

both Gaussian noise and single fractal noise that only contain simple frequency information, our noise pyramid can strengthen the influence of noise on the composed result (*i.e.*, $\boldsymbol{M} \otimes \mathbf{X}$) by strengthening the magnitude when the image is gradually replaced with noise.

Fig. 8 shows the impact of different perturbations on visualizations in the mixed4d layer of GoogLeNet. We find that a single fractal noise cannot express the detailed differences among neuron attributions. For example, by comparing the first three columns

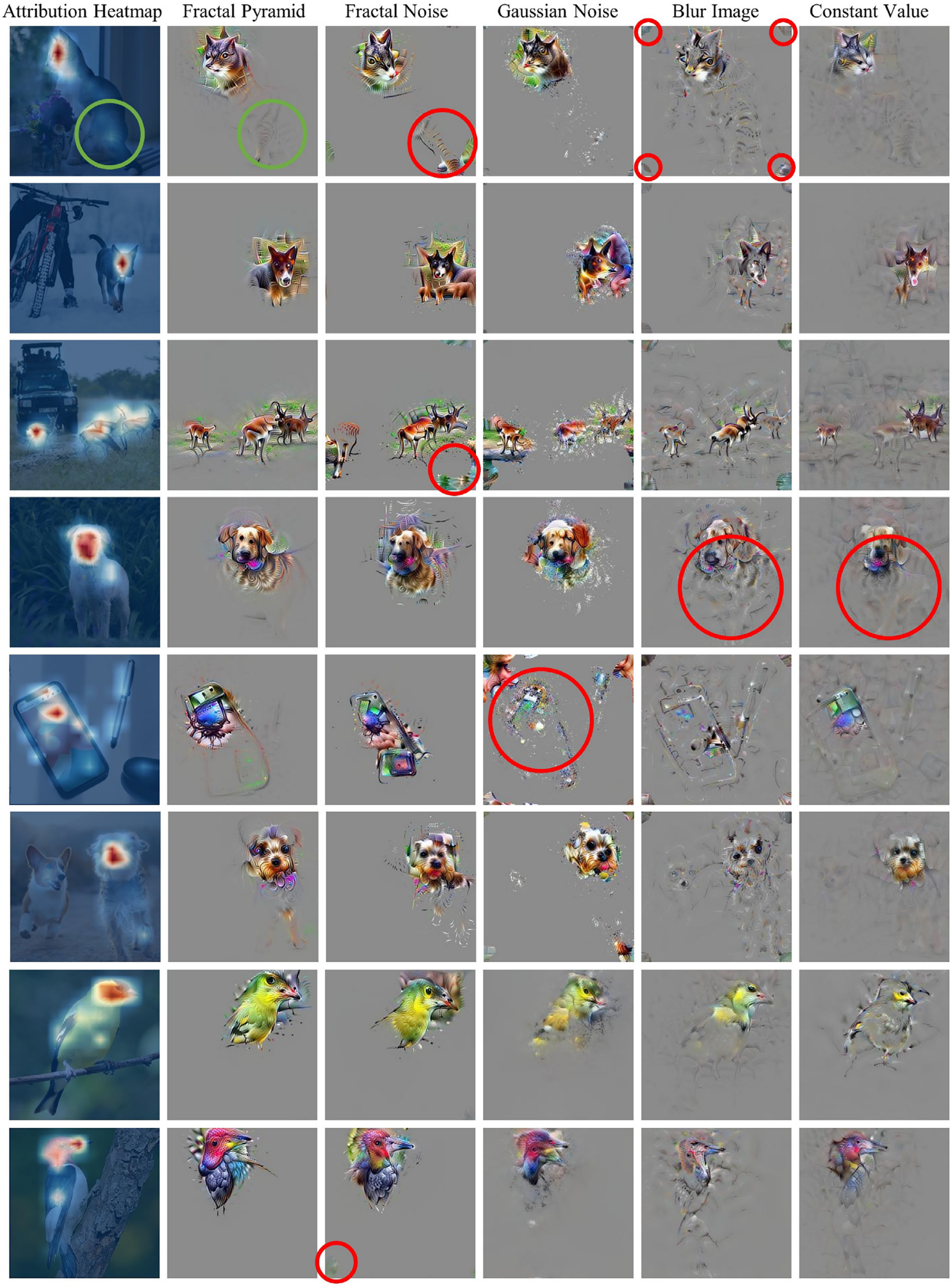


Fig. 8. Visualizations generated using our method with different perturbation techniques in the mixed4d layer.

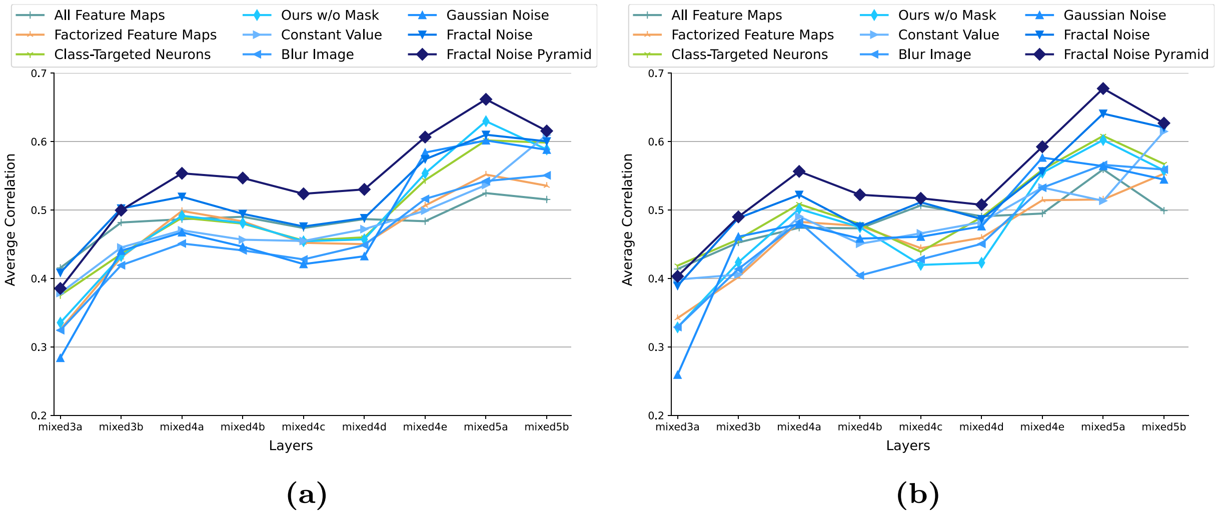


Fig. 9. Visualization correlation results. The PCCs results are calculated in feature space using GoogLeNets pre-trained on (a) ImageNet and (b) CUB-200, respectively. The left three data series in the figure legend are "All Feature Maps", "Factorized Feature Maps", and "Class-Targeted Neurons" corresponding to the methods proposed in [10,11,13], respectively. The right six data series correspond to our visualization methods without mask perturbation and with five different perturbation techniques.

of the top row, we find that the region of the cat's leg has smaller contributions (in green circle), but the visualization results generated using single fractal noise do not weaken this image region (in red circle). In addition, given a particular network, a single magnitude level sometimes produces artifacts in unrelated image regions during optimization, such as the green illusion in the bottom right region (in red circle) in the third column of the third row and of the last row. For Gaussian noise, we find that there are always many discontinuous noise dots in the results, *e.g.*, the red circle in the fourth column of the fifth row. According to the consequence of previous adversarial sample research [40], this type of spectra makes it easier for the network to fall into adversarial artifacts. Fig. 8 also shows that neither blurring image nor constant value can effectively express "natural" perturbation to generate clear visualizations, *e.g.*, red circles in the last two columns of the fourth row. When perturbing by image blurring, depending on the selected padding mode, fuzziness may appear at the corners (in red circle) as shown in the fifth column of the first row. We simply use the reflection mode, but we speculate that a carefully designed method of edge region processing may be able to solve the problem. Compared with all these alternatives, we find that our method can more clearly and intuitively show the implications of features contributing to the particular output.

Based on these visualization results, we next assess these results generated with different perturbation techniques using two metrics, *i.e.*, visualization correlation and classification consistency. The inspiration for this experiment is derived from [10] with some modifications made to the metric details to suit modern CNNs like GoogLeNet. These modifications are discussed in detail below.

Visualization correlation. This metric is used to test whether the visualizations could successfully achieve their goal of representing neuron attributions, *i.e.*, feature maps generated by forwarding a visualization into the network should be proportional to the corresponding attributions. This metric can be considered as a validation of the objective in Eq. (5). The visualization correlation is tested in terms of Pearson correlation coefficients (PCC) between the newly produced feature maps and the attributions at the neuron level. A good visualization should produce a high correlation score as it means that the method can successfully represent neuron attributions.

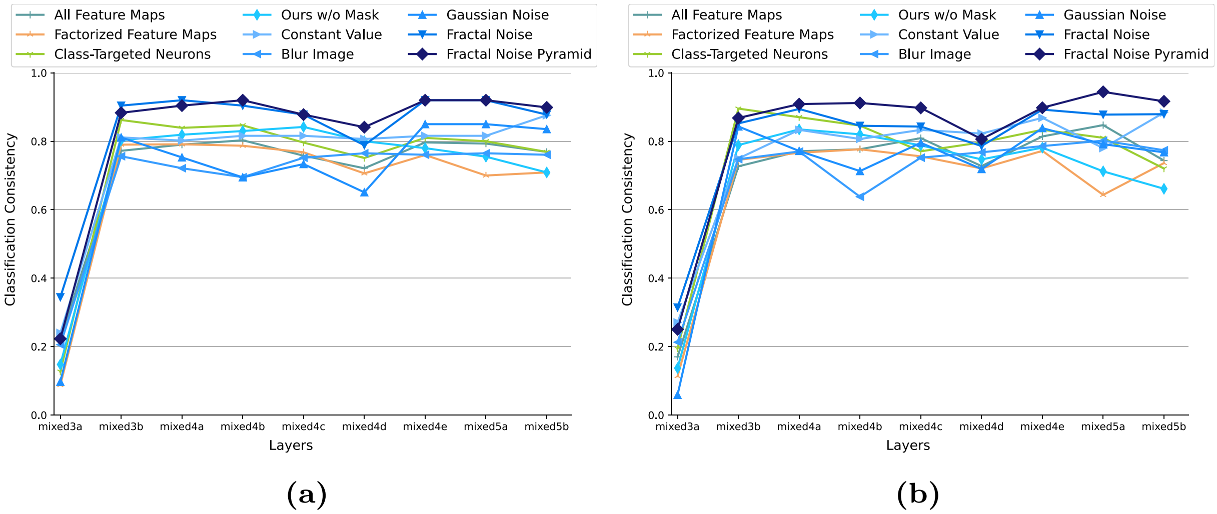


Fig. 10. Classification consistency results, *i.e.*, the fraction of successfully recognizing the particular class when forwarding the visualizations into the network. The results are calculated using GoogLeNets pre-trained on (a) ImageNet and (b) CUB-200, respectively.

The experiment is repeated for all tested layers of GoogLeNet and different perturbation techniques. For the methods using feature maps to generate visualizations, we just evaluate PCCs using the feature maps to replace the corresponding attribution scores. The resulting correlation score is assessed by reporting the average of PCCs over all tested images. Given every tested image, 25 different initializations are used to reduce the interference of random errors. As shown in Fig. 9, the average scores show that using our noise perturbation pyramid can produce the highest correlation score which means our visualizations are the best to represent exact information of attributions.

Classification consistency. With previous visualization correlation results, one question that may arise is whether a correlation of 0.6, or even 0.5, is good enough to represent neuron attributions. Scores above 0.5 can only show that the visualization expresses most of the attribution information rather than perfect representation. To assess to what extent important parts of attributions are represented in our visualizations, we report the classification consistency, which is the fraction of visualizations that the top-5 output classes of the network contain the particular class, as shown in Fig. 10. Note that, as visualizing factorized feature maps [11] generate multiple results, we accumulate the activation values of all these results to make a classification prediction. It is known that attributions stand for the neuron contributions, *i.e.*, given neuron features enhanced in proportion to attribution scores, a high output score should be assigned to the particular class. If the visualization perfectly represents the neuron features, then the classification consistency result of this visualization should be one. It also expresses that the visualization does not drop the semantics of the original object from the network viewpoint.

The results in Fig. 10 show that our proposal outperforms other alternatives. Moreover, despite the low PCCs from the layers mixed3b–mixed4d, the classification consistency results are always good from the layers mixed3b–mixed5b. This may be because the network can produce a high classification score if some of the most contributing neurons are highly activated. But in the mixed3a layer, both PCC score and classification consistency are the worst among all tested layers. To figure it out, we forward the attributions into the network and find that the top outputs are irregular and cannot correspond to the particular class. This shows that the attributions in this layer may not represent the neuron contributions to the output well. The Aumann–Shapley method satisfying several desirable properties aims to calculate the marginal contributions of individual neurons. However, whether the definition of "contribution" should be constant for different layers remains open. For example, since the low-level features are local, then the impact of their combination may be far greater than that of individuals. Thus, in addition to the marginal contribution of an individual neuron, the influence of feature combinations and interactions should probably be assigned a greater weight when calculating the attribution scores. Nevertheless, for attribution visualizations, using the fractal noise pyramid as a perturbation means better visual quality.

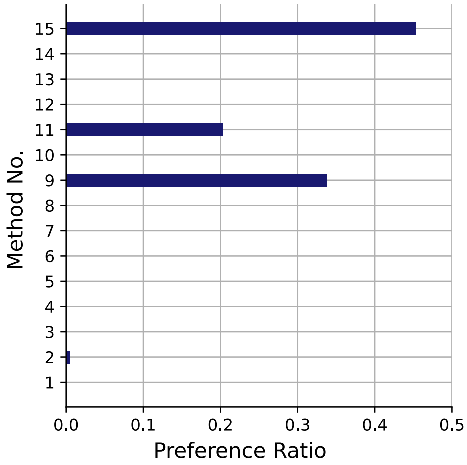
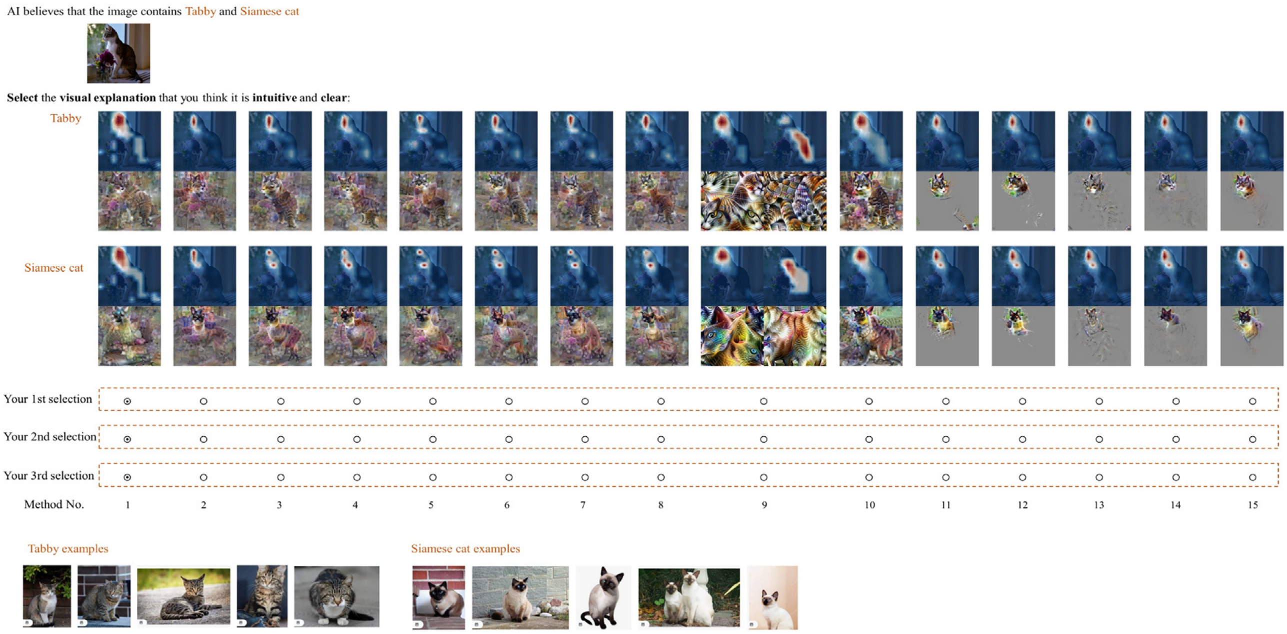


Fig. 12. Preference ratio for different visual explanations.


AI believes that the image contains Tabby and Siamese cat

Select the **visual explanation** that you think it is **intuitive** and **clear**:

Tabby

Siamese cat

Your 1st selection	⊙	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Your 2nd selection	⊙	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Your 3rd selection	⊙	○	○	○	○	○	○	○	○	○	○	○	○	○	○
Method No.	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15

Tabby examples

Siamese cat examples

Fig. 11. An example of the survey of user preference. The 15 Methods include 1) GradCAM [45] 2) LRP [16], 3) DeepLIFT Rescale [17], 4) DeepLIFT RevealCancel [17], 5) GradxInput [15], 6) DeepSHAP [46], 7) NeuronIG [27], 8) IG Noise [47], 9) visualizing class-discriminative feature groups [37], 10) visualizing class-targeted feature maps [13], 11) our visualization method with mask perturbation of fractal noise, 12) Gaussian noise, 13) blurring image, 14) constant value, and 15) fractal noise pyramid. In this example, "Tabby cat" is the top-1 class and "Siamese cat" is the class from top 2 to 5 classes. It is better to zoom in on the screen to see this screenshot of the survey.

4.3. User studies of visual explanations

In this experiment, we implement two user studies to evaluate which post hoc visual explanation is more intuitive and meaningful to humans. Before going deep into the details of the experiment, we first introduce visual explanations used in these user studies and the basic information of participants. In our experiments, we show different visual explanations in *random order*. But for clarity, here in the paper, we introduce 15 methods of generating visual explanations used in the experiment with a fixed order. All visual explanations are generated in the layer mixed4d of GoogLeNet.

Visual explanations of methods 1–8 are generated using seven attribution methods, including GradCAM [45], Layer-wise Relevance Propagation (LRP) [16], Deep Learning Important Features (DeepLIFT) Rescale and RevealCancel [17], GradxInput [15], Deep Shapley Additive Explanations (DeepSHAP) [46], NeuronIG [27], IG Noise [47]. Although other methods except NeuronIG were originally designed to calculate pixel attributions, they can be applied to calculate neuron attributions with small modifications. With these attribution methods, we calculate neuron attributions and generate heatmaps by summing attribution results along the channel dimension. However, for similar classes, only observing attribution heatmaps cannot provide an understandable visual explanation of attribution results. For example, we cannot understand the differences between the features of the two cat classes only using the heatmaps of methods 1–7, as shown in Fig. 11. To make the experiment more assessable, we thus visualize these attributions using our visualization method without a mask, as shown in the figures below the heatmaps. Visual explanations 9 and 10 are generated by visualizing class-discriminative feature groups [37] and class-targeted feature maps [13], respectively. Visual explanations 11–14 are generated using our method with four mask perturbation alternatives, which are fractal noise, Gaussian noise, blurring image, and constant value. The visual explanation 15 is our attribution visualization with a fractal noise pyramid.

In these user studies, we recruited students from three universities to fill out an online survey and received 32 responses. Among these 32 participants, 19 have CNN research experience, 13 are or were computer science majors but not familiar with deep learning.

Preference matters. In this user study, user preference for a visual explanation is evaluated. An example of the survey is shown in Fig. 11. We first generated visual explanations of two classes with the methods mentioned above using 15 images. The interpreted classes were the top-1 output class and one randomly selected class from the second to fifth results. As a reference for the class information, we provided five free images found by a commercial search engine for each of these two classes in the study, as shown at the bottom of Fig. 11. In a user study of each image example, participants were asked to select the top-3 intuitive and clear visual explanations from all 15 pairs of results arranged in random order. The scores assigned to the top-3 selec-

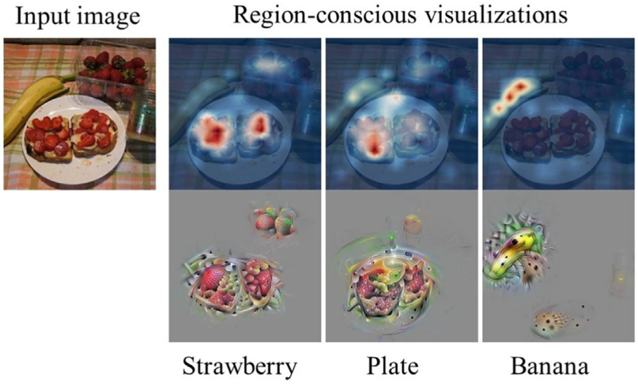


Fig. 14. Attribution visualizations for generating the example of the survey.

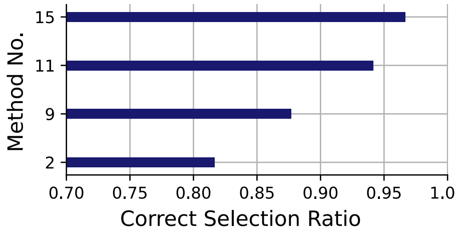
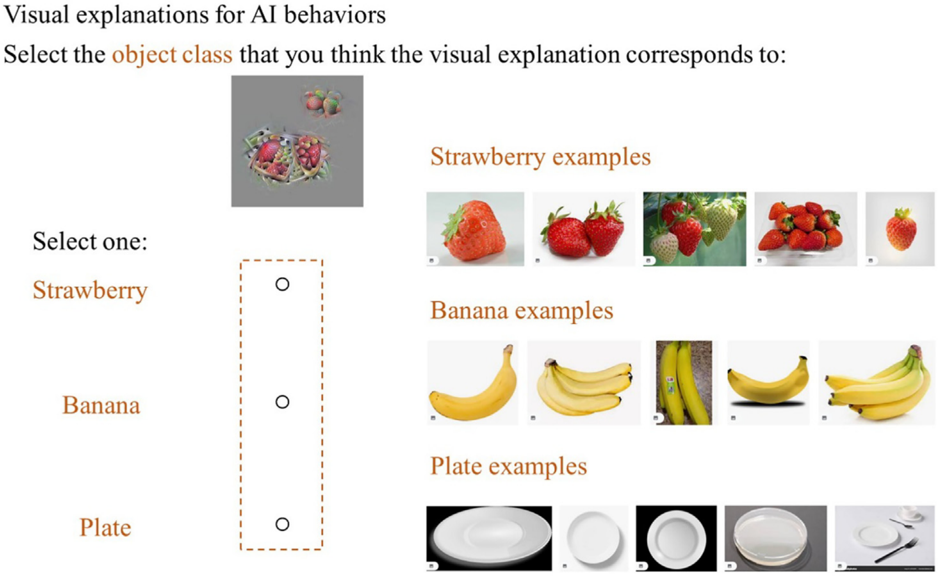


Fig. 15. Ratio of times a visual explanation was found to contain understandable semantic information of an image class. The four Methods include LRP (Method 2), visualizing class-discriminative feature groups (Method 9), our visualization method with mask perturbation of fractal noise (Method 11), and fractal noise pyramid (Method 15).



Visual explanations for AI behaviors

Select the object class that you think the visual explanation corresponds to:

Select one:

Strawberry ○

Banana ○

Plate ○

Fig. 13. An example of the survey of visualization meaning.

tions and the ones not selected are 3, 2, 1, and 0, respectively. Fig. 12 shows the fraction that a visual explanation was found to be intuitive and clear. The top-4 methods considered intuitive are our attribution visualization with the fractal noise pyramid, visualizing class-discriminative feature groups, our visualization with fractal noise, and LRP. Based on these results of preferences, we speculate that our method is preferable to human intuition.

Meaning matters. In this user study, we evaluate whether the visualization contains understandable semantic information about the targeted class. An example of the survey is shown in Fig. 13. The attribution visualizations used for generating the example survey are shown in Fig. 14, but note that the original image and the entire visualizations are not shown in the actual survey. We selected 15 other images and generated visual explanations without heatmaps in this experiment with the resulting top-4 methods in the previous user study. The four methods include LRP, visualizing class-discriminative feature groups, our visualization method with mask perturbation of fractal noise, and fractal noise pyramid.

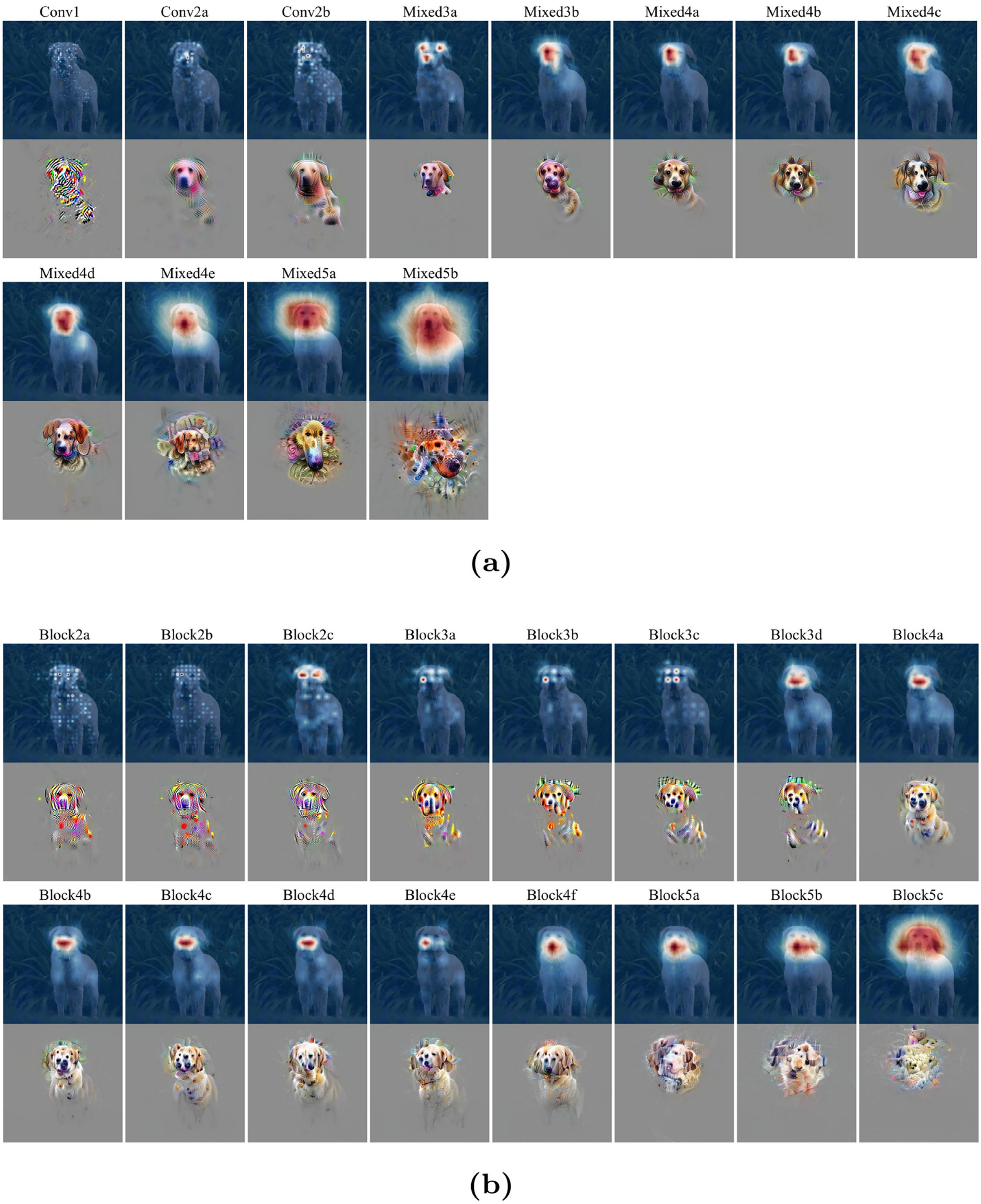


Fig. 16. Visualizations in different layers of (a) GoogLeNet and (b) ResNet-50.

The interpreted classes were the top-1 output class and two randomly selected classes from the second to fifth results. We also provided five images as references of a class like the previous study, as shown in the right part of Fig. 13. In a user study of each image example, participants were asked to select the class label corresponding to this visualization from three options of class labels. Fig. 15 shows the fraction that a visual explanation was found to be meaningful to a class. The results of visualization meaning show our method can more intuitively represent real semantics related to an output class.

4.4. Attribution visualization of different layers

In this section, we analyze the attribution (group) visualizations in different layers using GoogLeNet and ResNet-50 which have been applied to many vision applications as backbone networks. Through layer-wise attribution visualizations, we can intuitively compare the differences between the feature propagation rules of these two networks.

GoogLeNet can be partitioned into five parts, of which conv1 and conv2 extract low-level image features such as various colors, lines, and contours. These simple image features do not yet produce semantic information related to the particular class. Then, as the features are forwarded to deeper layers, the features with intuitive semantics appear in mixed3, mixed4, and mixed5. Fig. 16a shows the attribution visualizations in these layers. The progressive relationship among these layers is remarkable. In the first few layers (conv1–conv2b), the visualizations show Labrador images like crayon or watercolor paintings, which are produced by the arrangement and combination of low-level features, *e.g.*, colors, lines at different angles, and various shapes. These visualizations indicate the convolutional layers that most affect the visual content and styles in neural style transfer, thereby enabling different networks to be applied to style transfer with different architectures [48]. In the layers mixed3a–mixed4e, the visualizations show a relatively complete dog's head. Unlike previous visualizations like watercolor paintings, the head details indicate that low-level features are combined into object-specific features. But, in deeper layers (mixed5a–mixed5b), the geometric information almost disappears, and the abstract feature implications seem to make little sense to human intuition.

ResNet-50 can also be partitioned into five parts, of which conv1, block2, and block3 extract low-level image features. The features with object semantic information appear in block4 and block5. Despite different network architectures, the feature abstraction of ResNet is similar to GoogLeNet, as shown in Fig. 16b. For instance, crayon and watercolor paintings appear at similar stages (block2 and block3). In the following visualizations, the illusion of the dog's head also appears in both mixed4 and block4. But, because there are more layers in ResNet-50, the feature abstraction is achieved more gradually.

The residual architecture of ResNet contains several convolutional layers and residually connects the input and the output of the stage. Therefore, compared with GoogLeNet, features extracted by ResNet-50 seem to be forwarded from the lower layer to the deeper layer more easily. As an example, the heatmaps and the visualizations in the layers block4a–block5b of ResNet-50 are more consistent than that of similar layers (*i.e.*, mixed4a–mixed5a) of GoogLeNet, as shown in Fig. 16. Comparing the visualization details of these two networks, we find that the features used by ResNet-50 for classification could be more reasonable. For instance, the white fur features appear in the layers block4a–block4f, indicating that in addition to the head features, ResNet-50 can also extract useful information from the body region. The visualizations in these layers preserve white fur like the input image. But the visualizations of the layers mixed4a–mixed4e of GoogLeNet repre-

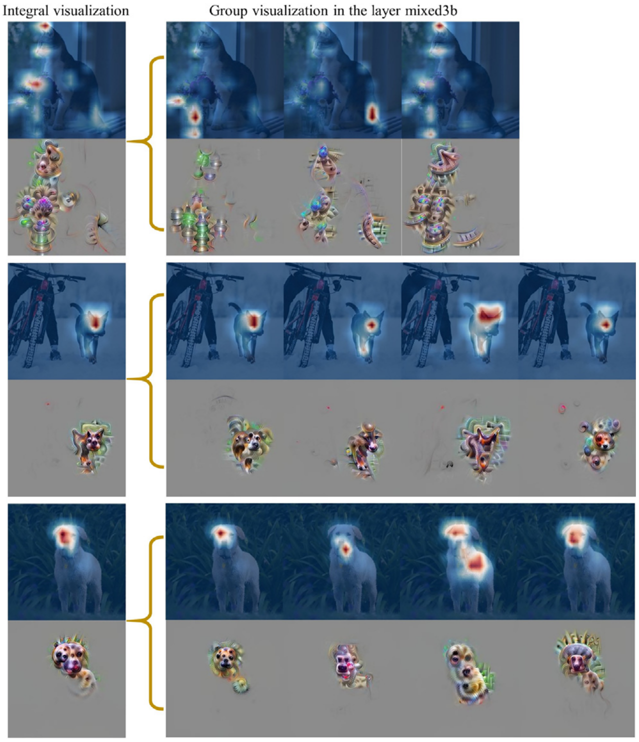


Fig. 17. Visualizations based on all attributions (left) and grouped ones (right) in the mixed3b layer.

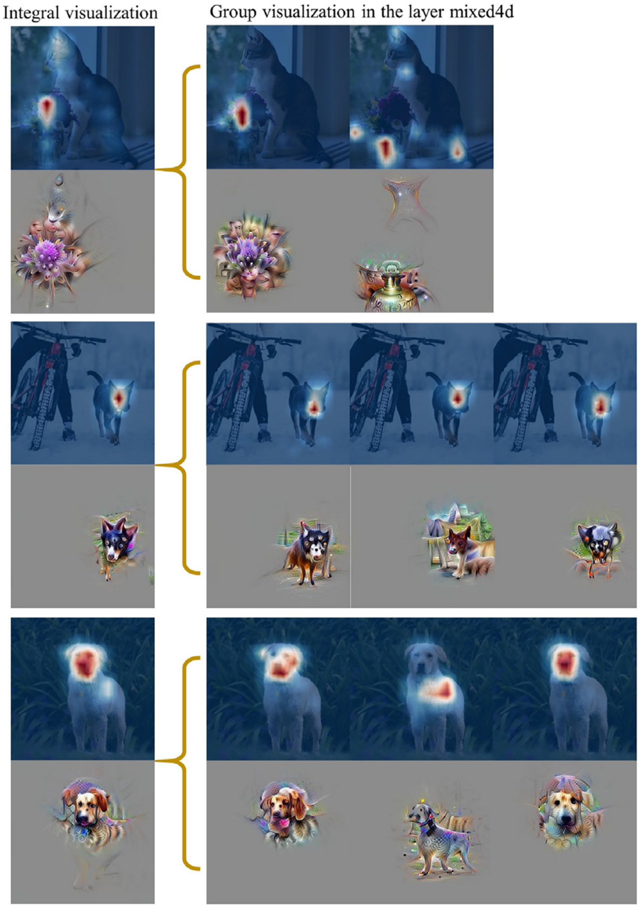


Fig. 18. Visualizations based on all attributions (left) and grouped ones (right) in the mixed4d layer.

sent colored hairs, because most Labrador retrievers in the dataset are black and yellow. This difference may be because there are almost twice more channels in a convolutional layer in ResNet-50 than that in GoogLeNet at a similar convolutional stage which may allow more capacity to memorize white-haired Labrador dogs as well as colored ones. With attribution visualizations, we can intuitively understand the feature abstraction among layers and compare the differences between different layers or different networks.

The visualizations in other experiments are generated using all attributions in a hidden layer together. However, some features with meaningful semantics but low contributions may be ignored in this case. Therefore, we further use factor analysis to decompose attributions with the same class label to obtain the grouping information as the technology proposed in [37]. Given new attributions, we can decompose them into several groups based on the grouping indices and then visualize them separately. The visualization results in the mixed3b and mixed4d layers are shown in Fig. 17 and 18. The visualized classes are "Vase", "Australian kelpie", and "Labrador retriever", respectively. The integral visualization is a combination of several group visualizations. For example, in the bottom row of Fig. 17, we find the integral result seems to be an abstract mix of the four group visualizations on the right.

The major advantage of group visualization is the ability to show some ignored features with small contributions but have meaningful semantics. For example, in the top row of Fig. 18 (the vase image), because the feature contribution from the flower region is much higher than the vase itself, both integral visualization and attribution heatmap cannot effectively display the vase features. However, the group visualization results can clearly show that the vase itself also contributes to the "Vase" class. In the bottom row of Fig. 18 (the Labrador retriever image), the dog's body does not contribute much to the "Labrador retriever" class; thus, there is almost no body part in the integral visualization. However, the decomposed results can highlight and visualize the body features with less contribution.

On the other hand, this advantage may be trivial in the lower layer where semantic information related to the class has not been extracted. Features extracted in these layers are only lines, shapes, *etc.*; thus, the semantic discrimination among visualization results is not obvious. For example, in the top row of Fig. 17, it seems that the shape of the cat's ear contributes to the "Vase" class. Even if we use group visualizations to distinguish these feature regions, we still cannot get more meaningful semantic understandings. A similar problem also appears when feature distribution is highly concentrated. For example, in the second row of Fig. 18, as the features related to the "Australian kelpie" class are all concentrated on the dog's face, group results also cannot show obvious discrimination. We believe that attribution decomposition is more suitable as a supplement, but sometimes it indeed reveals important information that may be ignored in the integral attribution visualization.

5. Conclusion

We propose the attribution visualization method to understand contributing neuron features. The core of our method is to introduce the mask concept into the visualization objective function with an area-constraint regularizer and design the fractal noise pyramid as a dynamic perturbation. According to the mask, fractal noises with different intensities in the spectrum of spatial frequencies are designed to perturb the updating visualization to produce natural-looking visualizations and suppress various noises during optimization. For evaluation, we use two metrics, *i.e.*, visualization correlation and classification consistency, to quantitatively assess the influence of different perturbation techniques. Our method with the fractal noise pyramid shows competitive results on these metrics. We also conduct several qualitative experiments including user studies to verify the effectiveness of our attribution visualizations.

Our work still has some limitations which need to be explored in the future. First, our visualization method is not currently available for reversely improving attribution calculation. It would be valuable to correct the attribution calculation method using visual feedback, especially in some delicate situations such as adversarial samples, which could further improve the reliability of network decisions. Second, our method requires users to manually specify visualization regularizers and select corresponding hyperparameters. An adaptive selection technology for regularizers would be beneficial in this regard. Finally, visualization evaluation metrics are still insufficient, especially the lack of quantitative metrics. Designing a dependable and impartial evaluation metric to enhance the validation of visualization results remains a fascinating subject. As research in this area continues, we plan to extend the proposed visualization method to encompass more sophisticated architectures, *e.g.*, vision systems based on multiple-layer perceptrons (MLPs), vision transformers, and multimodal networks.

CRediT authorship contribution statement

Rui Shi: Conceptualization, Investigation, Methodology, Software, Visualization, Writing - original draft. **Tianxing Li:** Investigation, Methodology, Software, Writing - review & editing. **Yasushi Yamaguchi:** Conceptualization, Formal analysis, Investigation, Funding acquisition, Supervision, Writing - review & editing.

Data availability

I have shared the link to my code in the manuscript.

Declaration of Competing Interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Yasushi YAMAGUCHI reports financial support was provided by Japan Society for the Promotion of Science.

Acknowledgment

We would like to thank our anonymous reviewers for their valuable suggestions. We are indebted to H. Deng for his helpful discussions and feedback. This research was partly supported by Japan Society for the Promotion of Science (JSPS KAKENHI) [Grant No. 20H04203].

Appendix A. Supplementary experimental results

We test the visualization methods stated in Section 4.1 on the other two images from Fig. 4. Fig. A.19 and A.20 show our results can more clearly represent feature implications than other methods without a mask.

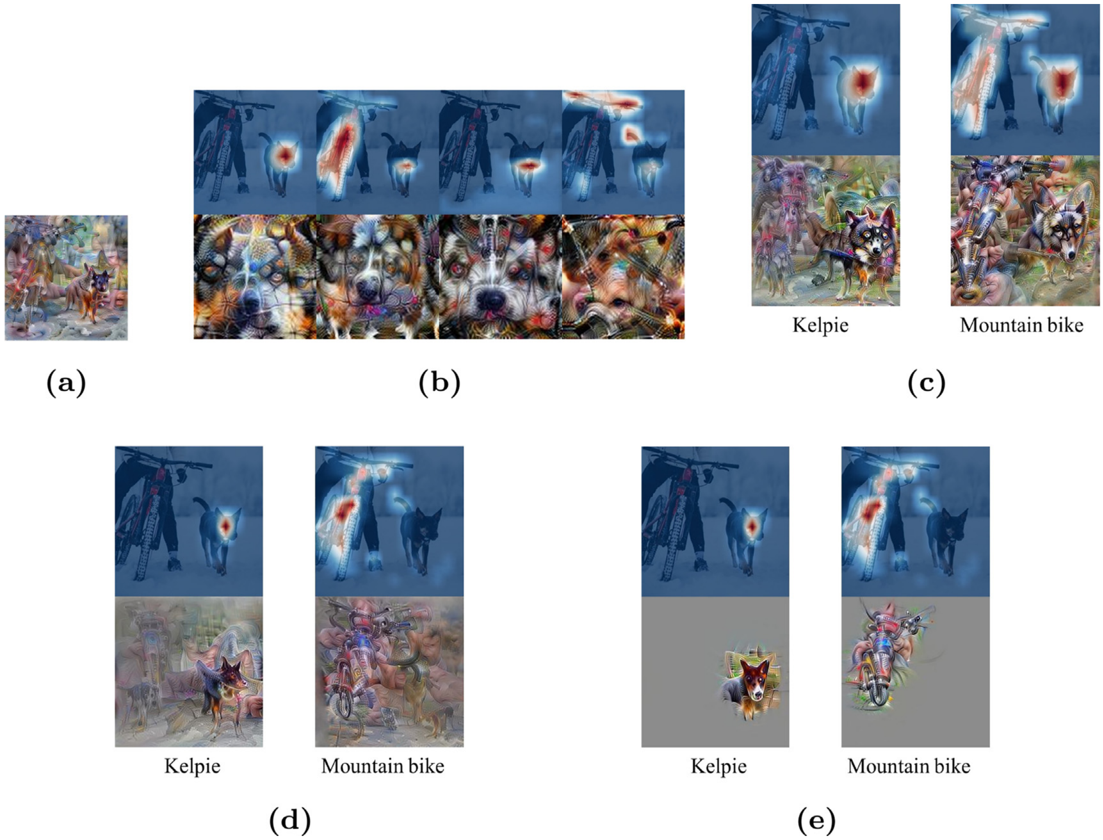


Fig. A.19. Visualizations using an image containing a kelpie and a mountain bike in the mixed4d layer with different methods.

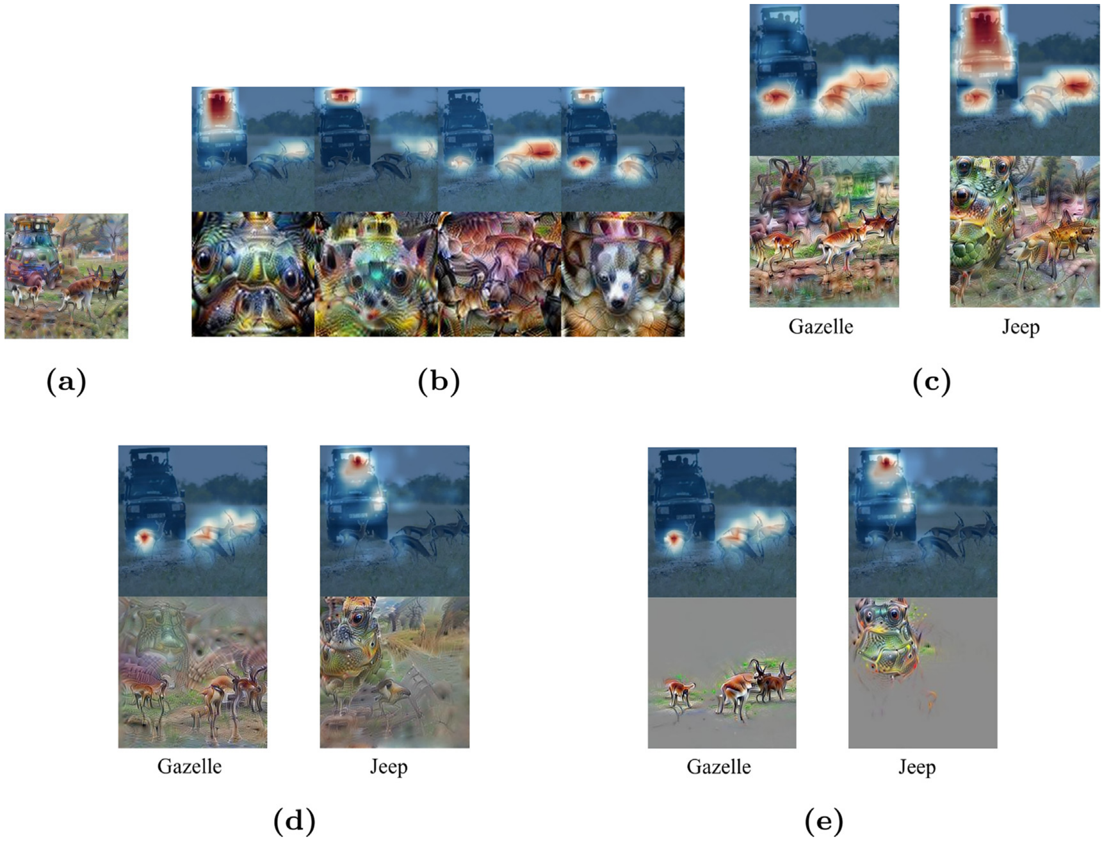


Fig. A.20. Visualizations using an image containing gazelles and a jeep in the mixed4d layer with different methods.

References

[1] L. Alzubaidi, J. Zhang, A.J. Humaidi, A. Al-Dujaili, Y. Duan, O. Al-Shamma, J. Santamaría, M.A. Fadhel, M. Al-Amidie, L. Farhan, Review of deep learning: Concepts, cnn architectures, challenges, applications, future directions, J. Big Data 8 (2021) 1–74.

[2] J. Ren, M. Li, M. Zhou, S.-H. Chan, Q. Zhang, Towards theoretical analysis of transformation complexity of ReLU DNNs, in: Proceedings of the 39th International Conference on Machine Learning, volume 162, PMLR, 2022, pp. 18537–18558. URL: https://proceedings.mlr.press/v162/ren22b.html.

[3] A. Ali, T. Schnake, O. Eberle, G. Montavon, K.-R. Müller, L. Wolf, XAI for transformers: Better explanations through conservative propagation, in: Proceedings of the 39th International Conference on Machine Learning, volume 162, 2022, pp. 435–451. URL: https://proceedings.mlr.press/v162/ali22a.html.

[4] D. Alvarez-Melis, T.S. Jaakkola, Towards robust interpretability with self-explaining neural networks, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, p. 7786–7795.

[5] W. Wang, C. Han, T. Zhou, D. Liu, Visual recognition with deep nearest centroids, in: International Conference on Learning Representations (ICLR), 2023.

[6] Y. Zhang, P. Tio, A. Leonardis, K. Tang, A survey on neural network interpretability, IEEE Trans. Emerg. Top. Comput. Intell. 5 (2021) 726–742.

[7] S. Rao, M. Böhle, B. Schiele, Towards better understanding attribution methods, in: Proceedings of the Conference on Computer Vision and Pattern Recognition, 2022, pp. 10223–10232.

[8] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, A. Rabinovich, Going deeper with convolutions, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 1–9. DOI: 10.1109/CVPR.2015.7298594.

[9] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, H. Lipson, Understanding neural networks through deep visualization, 2015. arXiv:1506.06579.

[10] A. Mahendran, A. Vedaldi, Visualizing deep convolutional neural networks using natural pre-images, Int. J. Comput. Vis. 120 (2016) 233–255.

[11] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, A. Mordvintsev, The building blocks of interpretability, Distill 3 (2018).

[12] H. Yin, P. Molchanov, J.M. Alvarez, Z. Li, A. Mallya, D. Hoiem, N.K. Jha, J. Kautz, Dreaming to distill: Data-free knowledge transfer via deepinversion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020. DOI: 10.1109/CVPR42600.2020.00874.

[13] S. Singla, B. Nushi, S. Shah, E. Kamar, E. Horvitz, Understanding failures of deep networks via robust feature extraction, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2021, pp. 12853–12862.

[14] K. Simonyan, A. Vedaldi, A. Zisserman, Deep inside convolutional networks: Visualising image classification models and saliency maps, in: Proceedings of the International Conference on Learning Representations, 2014. URL: http://dblp.uni-trier.de/db/conf/iclr/iclr2014w.html#SimonyanVZ13.

[15] A. Shrikumar, P. Greenside, A. Shcherbina, A. Kundaje, Not just a black box: Learning important features through propagating activation differences, 2017. arXiv:1605.01713.

[16] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, W. Samek, On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation, PloS one 10 (2015).

[17] A. Shrikumar, P. Greenside, A. Kundaje, Learning important features through propagating activation differences, in: International Conference on Machine Learning, PMLR, 2017, pp. 3145–3153.

[18] M. Ancona, C. Oztireli, M. Gross, Explaining deep neural networks with a polynomial time algorithm for shapley value approximation, in: Proceedings of the 36th International Conference on Machine Learning, 2019, pp. 272–281. URL: http://proceedings.mlr.press/v97/ancona19a.html.

[19] P.-J. Kindermans, S. Hooker, J. Adebayo, M. Alber, K.T. Schütt, S. Dähne, D. Erhan, B. Kim, The (un)reliability of saliency methods, 2017. arXiv:1711.00867.

[20] M. Sundararajan, A. Taly, Q. Yan, Axiomatic attribution for deep networks, in: Proceedings of the 34th International Conference on Machine Learning, 2017, p. 3319–3328.

[21] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, B. Kim, Sanity checks for saliency maps, in: Proceedings of the 32nd International Conference on Neural Information Processing Systems, 2018, pp. 9525–9536. URL: https://dl.acm.org/doi/10.5555/3327546.3327621.

[22] A. Ghorbani, A. Abid, J. Zou, Interpretation of neural networks is fragile, in: Proceedings of the AAAI conference on artificial intelligence, volume 33, 2019, pp. 3681–3688. DOI: 10.1609/aaai.v33i01.33013681.

[23] A. Alqaraawi, M. Schuessler, P. Weiß, E. Costanza, N. Berthouze, Evaluating saliency map explanations for convolutional neural networks: a user study, in: Proceedings of the 25th International Conference on Intelligent User Interfaces, 2020, pp. 275–285. DOI: 10.1145/3377325.3377519.

[24] T. Li, R. Shi, T. Kanai, Detail-aware deep clothing animations infused with multi-source attributes, Computer Graphics Forum 42 (2023) 231–244.

[25] R.J. Aumann, L.S. Shapley, Values of non-atomic games, Princeton University Press, 1974, URL: http://www.jstor.org/stable/j.ctt13x149m.

[26] Y. Sun, M. Sundararajan, Axiomatic attribution for multilinear functions, in: Proceedings of the 12th ACM Conference on Electronic Commerce, 2011, pp. 177–178. DOI: 10.1145/1993574.1993601.

[27] K. Dhamdhere, M. Sundararajan, Q. Yan, How important is a neuron, in: International Conference on Learning Representations, 2019. URL: https://openreview.net/forum?id=SylKoo0cKm.

[28] R. Shi, T. Li, Y. Yamaguchi, Output-targeted baseline for neuron attribution calculation, Image Vis. Computing 124 (2022).

[29] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, Advances in Neural Information Processing Systems, volume 30, Curran Associates Inc, 2017, URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[30] H. Chen, S.M. Lundberg, S.-I. Lee, Explaining a series of models by propagating shapley values, Nature Communications 13 (2022).

[31] A. Nguyen, A. Dosovitskiy, J. Yosinski, T. Brox, J. Clune, Synthesizing the preferred inputs for neurons in neural networks via deep generator networks, in: Proceedings of the 30th International Conference on Neural Information Processing Systems, 2016, pp. 3395–3403.

[32] G. Joshi, R. Natsuaki, A. Hirose, Neural network model for multi-sensor fusion and inverse mapping dynamics for the analysis of significant factors, in: IGARSS 2022–2022 IEEE International Geoscience and Remote Sensing Symposium, 2022, pp. 473–476. DOI: 10.1109/IGARSS46834.2022.9884409.

[33] G. Joshi, R. Natsuaki, A. Hirose, Neural network fusion processing and inverse mapping to combine multisensor satellite data and analyze the prominent features, IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 16 (2023) 2819–2840.

[34] C. Olah, A. Mordvintsev, L. Schubert, Feature visualization, Distill 2 (2017).

[35] E. Protas, J.D. Bratti, J.F. Gaya, P. Drews, S.S. Botelho, Visualization methods for image transformation convolutional neural networks, IEEE Trans. Neural Netw. Learn. Syst. 30 (2018) 2231–2243.

[36] A. Mahendran, A. Vedaldi, Understanding deep image representations by inverting them, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5188–5196. DOI: 10.1109/CVPR.2015.7299155.

[37] R. Shi, T. Li, Y. Yamaguchi, Group visualization of class-discriminative features, Neural Netw. 129 (2020) 75–90.

[38] N. Otsu, A threshold selection method from gray-level histograms, IEEE Trans. Syst. Man Cybern. 9 (1979) 62–66.

[39] T.H. Keitt, Spectral representation of neutral landscapes, Landsc. Ecol. 15 (2000) 479–494.

[40] D. Yin, R. Gontijo Lopes, J. Shlens, E.D. Cubuk, J. Gilmer, A Fourier perspective on model robustness in computer vision, in: Advances in Neural Information Processing Systems, volume 32, 2019. URL: https://proceedings.neurips.cc/paper/2019/file/b05b57f6add810d3b7490866d74c0053-Paper.pdf.

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[42] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, P. Perona, Caltech-UCSD Birds 200, Technical Report CNS-TR-201, Caltech, 2010. URL:/se3/wp-content/uploads/2014/09/WelinderEtal10_CUB-200.pdf, http://www.vision.caltech.edu/visipedia/CUB-200.html.

[43] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, L. Fei-Fei, Imagenet: A large-scale hierarchical image database, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

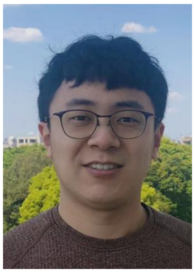
[44] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2017. arXiv:1412.6980.

[45] R.R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, D. Batra, Grad-cam: Visual explanations from deep networks via gradient-based localization, in: IEEE International Conference on Computer Vision, 2017, pp. 618–626. URL: doi: 10.1109/ICCV.2017.74. DOI: 10.1109/ICCV.2017.74.

[46] H. Chen, S.M. Lundberg, S.-I. Lee, Explaining a series of models by propagating shapley values, Nat. Commun. 13 (2022) 4512.

[47] P. Sturmfels, S. Lundberg, S.-I. Lee, Visualizing the impact of feature attribution baselines, Distill 5 (2020).

[48] K. Simonyan, A. Zisserman, Very deep convolutional networks for large-scale image recognition, 2015. arXiv:1409.1556.



Rui Shi received his Ph.D. degree in computer science from the University of Tokyo in 2022. He is currently a lecturer at Beijing University of Technology. His current research interests include explainable artificial intelligence and neural network visualization.

Tianxing Li received her Ph.D. degree in computer science from the University of Tokyo in 2021. She is currently a lecturer at Beijing University of Technology. Her current research interests include computer graphics and pattern recognition.

Yasushi Yamaguchi is a professor of the Graduate School of Arts and Sciences, the University of Tokyo, Tokyo, Japan. He received his Ph.D. in Information Engineering from the University of Tokyo in 1988. His research interests lie in image processing, computer graphics, and visual illusion, including image editing, computer-aided geometric design, visual cryptography, and hybrid image. He was a former president of the International Society for Geometry and Graphics.