Deeply Explain CNN via Hierarchical Decomposition

Ming-Ming Cheng[1*†], Peng-Tao Jiang[1†], Ling-Hao Han[1], Liang Wang[2] and Philip Torr[3]

[1*]TMCC, Nankai University, Tianjin, China.
[2]NLPR, Beijing, China.
[3]University of Oxford, UK.

*Corresponding author(s). E-mail(s): cmm@nankai.edu.cn;
†The first two authors contributed equally to this work.

Abstract

In computer vision, some attribution methods for explaining CNNs attempt to study how the intermediate features affect network prediction. However, they usually ignore the feature hierarchies among the intermediate features. This paper introduces a hierarchical decomposition framework to explain CNN's decision-making process in a top-down manner. Specifically, we propose a gradient-based activation propagation (gAP) module that can decompose any intermediate CNN decision to its lower layers and find the supporting features. Then we utilize the gAP module to iteratively decompose the network decision to the supporting evidence from different CNN layers. The proposed framework can generate a deep hierarchy of strongly associated supporting evidence for the network decision, which provides insight into the decision-making process. Moreover, gAP is effort-free for understanding CNN-based models without network architecture modification and extra training process. Experiments show the effectiveness of the proposed method. The data and source code will be publicly available at https://mmcheng.net/hdecomp/.

Keywords: Explaining CNNs, hierarchical decomposition

1 Introduction

Deep convolutional neural networks (CNN) have made significant improvements on various computer vision tasks, such as image recognition (Simonyan and Zisserman, 2015; He et al, 2016; Huang et al, 2017), object detection (Girshick et al, 2014; Girshick, 2015; Ren et al, 2015), semantic segmentation (Long et al, 2015; Chen et al, 2017; Zhao et al, 2017; Lin et al, 2017), traffic environment analysis (Zhu et al, 2016; Hou et al, 2019), medical image understanding (Ronneberger et al, 2015; Litjens et al, 2017), and weakly supervised segmentation (Papandreou et al, 2015; Hou et al, 2018; Jiang et al, 2022). Despite the high performance, CNNs are usually used as black boxes as their internal decision process is unclear. Moreover, plenty of recent research (Goodfellow et al, 2014; Kurakin et al, 2017; Athalye et al, 2018), has pointed out that the previous successful CNN models can still be fooled by adversarial examples where the changes can not even be noticed by human eyes. With the above prior, it is difficult for human beings to trust the good-performing yet opaque CNN models. Therefore, the interpretability of CNNs is as crucial as their performance, especially in some critical applications.

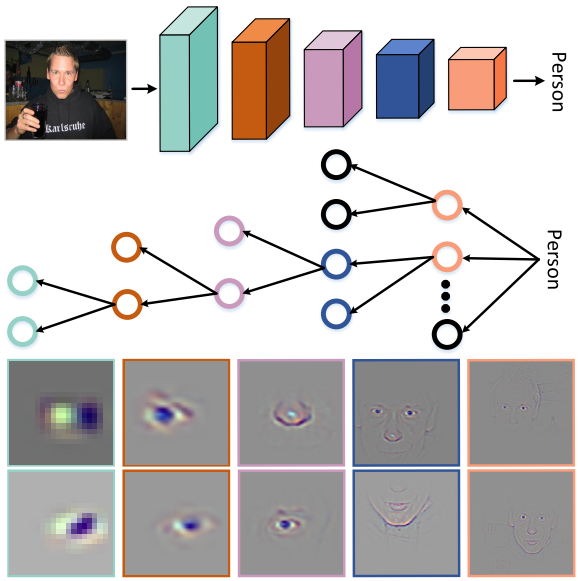
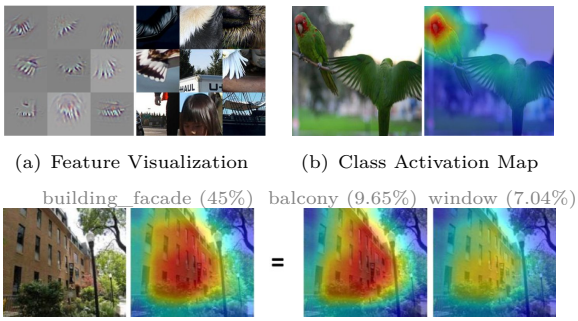


Fig. 1 Overview of our approach. The middle is an evidence pyramid for the network prediction, and the bottom shows the important features from different stages of VGG-16. The colored circles represent the features. We detect interactions among the features and show how they are combined at different hierarchies in the decision-making process.

A fully interpretable convolutional neural network is a long-standing holy grail for deep learning researchers. To this end, researchers have proposed a wide range of techniques. Feature attribution (or saliency) methods (Sundararajan et al, 2017; Shrikumar et al, 2017; Simonyan et al, 2014) provide a powerful tool for interpretability. They attribute an output prediction of CNN to the input image, where the generated saliency map can tell us which pixels are important to the prediction. Such ability helps humans to understand how the input affects the prediction. Another set of feature attribution methods (Dhamdhere et al, 2019; Leino et al, 2018) measures the importance of intermediate features towards a prediction. They further select important features and study their impact on the prediction. Apart from generating the feature importance, the relationships among intermediate features (Olah et al, 2020) are also important to understand the predictions but receive little attention.

CNNs have demonstrated a strong ability to gradually abstract image contents and generate features at different semantic levels, *e.g.,* blobs/edges, textures, and object parts (Zeiler and Fergus, 2014). While discovering important features can provide a rich set of evidence for the output prediction, isolated evidence is less convincing and informative than the evidence chain (Giannelli, 1982) or the evidence pyramid (Murad et al, 2016). According to the feature integration theory developed by Treisman and Gelade (1980), the human brain first extracts basic features and then utilizes attention to combine individual features to perceive the object. Ideally, we would expect a hierarchical evidence tree as demonstrated in Fig. 1, which attributes a CNN decision to multiple key features, and each of them can be recursively attributed to more basic features. By associating intermediate features like 'head', 'face', 'eye', 'nose', and 'edge' in this example, a group of strongly associating evidence corresponding to the network's inner state is emerging, reviewing real-world facts of the human perception decision.

There are two major challenges for existing feature attribution methods to achieve the hierarchical decomposition. Firstly, directly decomposing millions of feature responses in all channels and all spatial locations is both computationally infeasible and cognitively overloading for humans. Meanwhile, feature attribution methods such as (Dhamdhere et al, 2019; Leino et al, 2018) are quite time-consuming because they need to repeat the backpropagation process many times. Secondly, some attribution methods (Zhou et al, 2016; Selvaraju et al, 2020) generate an attention map for the whole layer, rather than a group of attention maps for each feature channel. The channel-wise attention maps are crucial for the iterative decomposition process as they can indicate the most important neuron in a feature channel to be decomposed. To alleviate these issues, we propose an efficient gradient-based Activation Propagation (gAP) module, which decomposes a feature response at any CNN location to its lower layer. As the gAP module generates an activation map for each feature channel, we can easily select a few mostly activated feature channels as crucial evidence, obtaining human-scale explanations. For each of those selected feature channels, the CNN feature at the most activated spatial position can be iteratively decomposed. By avoiding decomposing features at too many spatial locations, we can further reduce the number of potential visualizations to the human scale.



(a) Feature Visualization (b) Class Activation Map

(c) High-level Decomposition

Fig. 2 Illustration of different kinds of interpretative methods.

The proposed hierarchical decomposition framework can effectively generate hierarchical explanations (see Fig. 1), which builds relationships among crucial intermediate features. We have conducted extensive experiments on several aspects, including a sanity check of the gAP module and understanding the network decisions. Experiments show the effectiveness of our framework to explain network decisions. In summary, we make two major contributions:

- We propose an efficient gradient-based Activation Propagation (gAP) module, which decomposes the network decision and intermediate features to find their key supporting evidence from previous layers.
- We propose a hierarchical decomposition framework, which builds relationships among important intermediate features, enabling hierarchical explanations with human-scale supporting evidence.

2 Related Work

The interpretability of CNNs has been actively studied, with major progress in four main areas, including feature attribution, feature visualization, knowledge distillation, and intrinsic interpretable models.

2.1 Feature Attribution

Feature attribution methods typically generate a saliency map to locate the input locations important to the output. We classify them into three categories: backpropagation-based methods, perturbation-based methods, and activation-based methods.

Backpropagation-based methods. In the early days, Sung (1998) learn to rank the importance of input for the backpropagation networks by several tools such as sensitivity analysis. Baehrens et al (2010) identify the feature importance for a particular instance by computing the gradients of the decision function. Simonyan et al (2014) backpropagate the gradients of the output prediction *w.r.t.* the input image and generate a saliency map that indicates the importance of each pixel in the image. All the above methods utilize the partial derivative of the output to the input. Guided Backpropagation (Springenberg et al, 2015) and Deconvnet (Zeiler and Fergus, 2014) utilize different backpropagation logics through ReLU, where they both zero out the negative gradients. Sundararajan et al (2017) consider the saturation and thresholding problem. They compute the saliency map by accumulating the gradients along a path from the base image to the input image. Another set of methods, such as LRP (Bach et al, 2015), DeepTayor (Montavon et al, 2017), RectGrad (Kim et al, 2019), DeepLift (Shrikumar et al, 2017), FullGrad (Srinivas and Fleuret, 2019), PatternAttribution (Kindermans et al, 2018), and Excitation Backprop (Zhang et al, 2016), utilize different top-down relevance propagation rules. Yang et al (2020) attempt to learn the propagation rule automatically for attribution map generation. SmoothGrad (Smilkov et al, 2017) sharpen the gradient-based salience maps to reduce visual noise. Zintgraf et al (2017) not only identify the important regions supporting the network decision but also identify the regions against the decision. Moreover, some methods (Dhamdhere et al, 2019; Leino et al, 2018) measure the importance of the hidden unit to the prediction based on the backpropagation. These methods can find out the most important features from different layers of deep networks. Kim et al (2018) study the high-level concepts instead of low-level features for interpreting the internal state of the neural network. They utilize the directional derivatives to quantify the importance of high-level concepts to a classification result.

Perturbation-based methods. These methods perturb the input to observe the output changes. Zeiler and Fergus (2014) occlude the input image by sliding a gray square and use the change of

the output as the importance. Petsiuk et al (2018) randomly sampled a masked region. Ribeiro et al (2016) utilize the super-pixel to select occluded image regions. They learn a local linear model to compute the contribution of each super-pixel. Besides, the recent methods (Fong and Vedaldi, 2017; Fong et al, 2019; Dabkowski and Gal, 2017) learn a perturbation map, where the map applied to the input image can maximumly affect the prediction. Fong et al (2019) also apply the input attribution method to study the salient channels of deep networks.

Activation-based methods. These methods (Zhou et al, 2016; Selvaraju et al, 2020; Chattopadhay et al, 2018) generate a coarse class activation map by linearly combining the feature channels from the convolutional layer. The class activation map is upsampled to the size of the input image and provides image-level evidence that is important for the network prediction, as demonstrated in Fig. 2(b). Zhou et al (2016) propose Class Activation Mapping (CAM). They need a specific network with the global average pooling layer to generate class activation maps. Later, Grad-CAM (Selvaraju et al, 2020) and Grad-CAM++ (Chattopadhay et al, 2018) generalize the CAM method to other tasks by utilizing the task-specific gradients as weights. Besides, LayerCAM (Jiang et al, 2021) generate reliable class activation maps for both deep and shallow layers. Unlike Grad-CAM, Score-CAM (Wang et al, 2020) utilize the forward passing score on the target class to obtain the weight for each activation. Recently, Zhou et al (2018) attempt to decompose the network decision into several semantic components and study each component's contribution. As shown in Fig. 2(c), the class activation map is decomposed into several semantic components.

The aforementioned attribution methods mostly focus on generating saliency/activation maps to study how the input affects the output prediction. Although some attribution methods can measure the importance of intermediate features to the output prediction, they usually neglect to study the relationships among different intermediate features. As pointed by Olah et al (2020), the relationships among different intermediate features are also important to interpret a prediction. We decompose not only the network decision but also the intermediate features to find their supporting evidence from previous layers, explaining how these associated intermediate features affect each other. While LRP (Bach et al, 2015) method propagates feature importance to intermediate features, the feature importance for different channels is coupled in the back-propagation process. This method generates simple explanations for the entire network behavior rather than hierarchical explanations.

2.2 Feature Visualization

Visualizing the CNN features of the intermediate layers can provide insight into what these layers learn. For the first layer of the CNN, we can directly project its three-channel weights into the image space. To visualize the features from higher layers, researchers have proposed many alternative approaches. Among them, Erhan et al (2009) and Simonyan et al (2014) utilize the gradient ascent algorithm to find the optimal stimuli in the image space that maximizes the neuron activations. Other methods (Zeiler and Fergus, 2014; Springenberg et al, 2015; Zhou et al, 2015) identify the image patches from the dataset that maximize the neuron activation of the CNN layers, as shown in Fig. 2(a). Guided Backpropagation (Springenberg et al, 2015) and Deconvnet (Zeiler and Fergus, 2014) also utilize the top-down gradients to discover the patterns that the intermediate layers learn. Using the natural image prior, feature inversion methods (Mordvintsev et al, 2015; Yosinski et al, 2015; Mahendran and Vedaldi, 2015; Dosovitskiy and Brox, 2016; Olah et al, 2017) learn an image to reconstruct the neuron activation. Furthermore, the recent methods (Bau et al, 2017; Fong and Vedaldi, 2018; Bau et al, 2020) attempt to detect the concepts learned by intermediate CNN layers. The above feature visualization methods explore what the intermediate features detect, but they do not answer how the network assembles individual features to make a prediction.

2.3 Knowledge Distillation

Recently, another research line has attempted to transfer the powerful ability of CNN to explainable models, such as the decision tree or linear model, to approximate the behavior of the original

model. Chen et al (2019b) distill the knowledge into an explainable additive model. Ribeiro et al (2016) utilize a local linear model to approximate the original model, studying how the input affects any classifier's decisions. Frosst and Hinton (2017) and Liu et al (2018) distill the learned knowledge of CNN into the decision tree. These methods only bridge the network decision and the input. They cannot help the user understand how the internal features of CNNs affect the network decision and each other. Our hierarchical decomposition is also an approximation to the original model. Unlike the above methods, our hierarchical decomposition not only highlights the important features for the network decision but builds the relationships among the feature channels from different layers. From our method, we can obtain the states of the internal features and how the internal features affect each other and the network decision.

2.4 Intrinsic Interpretable Models

Except for the post-hoc interpretability analysis for a trained CNN, some researchers have attempted to explore inherently interpretable models. Chen et al (2019a) propose a deep network architecture called prototypical part network. The network has a transparent reasoning process that first computes the similarity scores between the image patches and the learned prototypes. Then the network makes predictions based on a weighted sum of the similarity scores. Concept bottleneck models (Koh et al, 2020; Kumar et al, 2009; Lampert et al, 2009) are also inherently interpretable. Unlike those post-hoc methods (Bau et al, 2017; Fong and Vedaldi, 2018) that utilize human-specific concepts to generate explanations, they directly predict a set of human-specific concepts at training time and then use these concepts to make predictions, where the reasoning process is interpretable. Some recent intrinsic interpretable models (Koh et al, 2020; Chen et al, 2019a) utilize VGG (Simonyan and Zisserman, 2015) or ResNet (He et al, 2016) to extract high-level features first and perform the reasoning process on the high-level features. Our method is complementary to these CNN-based intrinsic interpretable models because one can use the hierarchical decomposition to provide more hierarchical evidence from the feature extractor if needed.

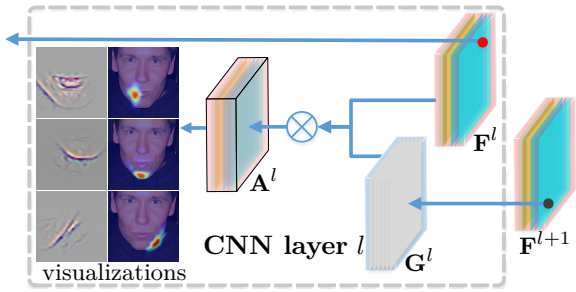


Fig. 3 Our gradient-based activation propagation (gAP) method explains a decision of interest $\mathbf{F}^{l+1}_{k',x,y} \in \mathbb{R}$, *i.e.,* the CNN feature illustrated by the black dot, by localizing the most related neuron activations in its previous CNN layer.

3 Methodology

3.1 Gradient-based Activation Propagation

We begin by defining the notation for the CNN, as illustrated in Fig. 3. In the l^{th} CNN layer, the **features** $\mathbf{F}^l$, partial **gradients** $\mathbf{G}^l$, and corresponding neuron **activations** $\mathbf{A}^l$ are 3D tensors with the same size, *i.e.,* $\mathbf{G}^l, \mathbf{A}^l, \mathbf{F}^l \in \mathbb{R}^{K^l \times H^l \times W^l}$, where K^l is the number of channels and $H^l \times W^l$ is the spatial size in the CNN layer l. To find supporting evidence for the final CNN decision or any intermediate feature response, we propose a gradient-based activation propagation (gAP) method. Using the gAP module, we can understand *a decision of interest at a CNN layer by localizing the most related evidence in its previous layer.*

As shown in Fig. 3, we decompose a CNN feature (*i.e.,* a decision of interest) $\mathbf{F}^{l+1}_{k',x,y}$ at the convolutional layer $l+1$, channel k', and spatial position (x, y), to find the supporting evidence in its previous convolutional layer l. In this work, we are interested in understanding the strong feature response $\mathbf{F}^{l+1}_{k',x,y}$, where $\mathbf{F}^{l+1}_{k',x,y} > 0$. In typical CNNs, a certain feature at layer $l+1$ is computed as a linear combination of features from its previous layer l and a ReLU. For the strong feature $\mathbf{F}^{l+1}_{k',x,y}$, we have

$$\begin{aligned}\mathbf{F}^{l+1}_{k',x,y} &= \mathrm{ReLU}(\sum_k \sum_i \sum_j \mathbf{w}^{\mathbf{l}}_{\mathbf{k},\mathbf{i},\mathbf{j}} \cdot \mathbf{F}^{\mathbf{l}}_{\mathbf{k},\mathbf{i},\mathbf{j}}) \\ &= \sum_k \sum_i \sum_j \mathbf{w}^{\mathbf{l}}_{\mathbf{k},\mathbf{i},\mathbf{j}} \cdot \mathbf{F}^{\mathbf{l}}_{\mathbf{k},\mathbf{i},\mathbf{j}},\end{aligned} \tag{1}$$

where $\mathbf{w}^{\mathbf{l}}_{\mathbf{k,i,j}}$ is the linear weight for combining the feature $\mathbf{F}^l_{k,i,j}$. To obtain the weight, we first use backpropagation to compute the partial gradient map $\mathbf{G}^l_k$ of the feature $\mathbf{F}^{l+1}_{k',x,y}$ *w.r.t.* the feature map $\mathbf{F}^l_k$ by

$$\mathbf{w}^{\mathbf{l}}_{\mathbf{k}} = \mathbf{G}^{\mathbf{l}}_{\mathbf{k}} = \underbrace{\frac{\partial \mathbf{F}^{l+1}_{k',x,y}}{\partial \mathbf{F}^l_k}}_{\text{gradients via backprop}}. \tag{2}$$

The gradient map $\mathbf{G}^l_k$ captures the 'importance' of the feature map $\mathbf{F}^l_k$ for the decision $\mathbf{F}^{l+1}_{k',x,y}$.

We employ the gradient map $\mathbf{G}^l_k$ to generate an activation map

$$\mathbf{A}^l_k = \mathbf{G}^l_k \cdot \mathbf{F}^l_k. \tag{3}$$

The activation map indicates the contribution of each feature in $\mathbf{F}^l_k$ to the decision $\mathbf{F}^{l+1}_{k',x,y}$. Based on its corresponding activation map, each channel's contribution to the decision can be computed by

$$\alpha^l_k = \frac{1}{Z^l} \sum_x^{H^l} \sum_y^{W^l} \mathbf{A}^l_{k,x,y}, \tag{4}$$

where $Z^l = H^l \times W^l$ denotes the number of spatial positions in the activation map $\mathbf{A}^l_k$. We can also identify the feature $\mathbf{F}^l_{k,\hat{x},\hat{y}}$ in k^{th} feature channel that contributes the most to the decision in which

$$(\hat{x}, \hat{y}) = \arg\max_{(x,y)} \mathbf{A}^l_{k,x,y}. \tag{5}$$

Thus, for each decision at layer $l+1$, we can find the most important feature channel $\mathbf{F}^l_k$ at layer l according to the contribution α^l_k computed by Eqn. (4). In the most important channel, we can also identify the feature $\mathbf{F}^l_{k,\hat{x},\hat{y}}$ that contributes most to the decision according to Eqn. (5). In the top row of Fig. 4, we show the three most important activation maps $\mathbf{A}^4_{131}$, $\mathbf{A}^4_{255}$, and $\mathbf{A}^4_{452}$ in layer `conv4_3` for the decision from $\mathbf{F}^5_{277}$. These activation maps provide spatial channel responses to the decision, benefiting human understanding. Using Guided Backpropagation (Springenberg et al, 2015), we visualize the most contributing feature by generating sharp visualizations, which highlight the associated input. An example is shown in the bottom row of Fig. 4.

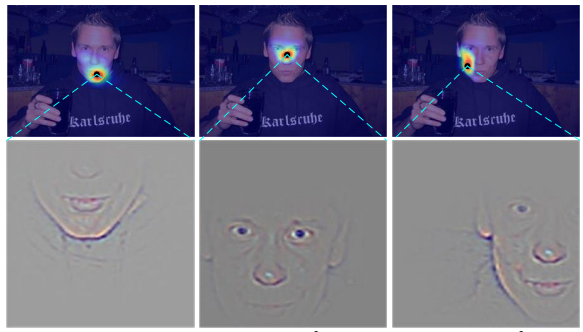


Fig. 4 Example of the most significant activation maps (upper row) and their corresponding visualizations (lower row) for layer `conv4_3`, which contains 512 channels. The black dot denotes the peak location in the activation map.

Note that gAP itself does not include Guided Backpropagation. We only utilize Guided Backpropagation Springenberg et al (2015) to generate sharp visualizations of the selected top activations by gAP.

Discussion. Our gAP module is inspired by CAM (Zhou et al, 2016) and Grad-CAM (Selvaraju et al, 2020), which explain CNN decisions by class activation localization. To explain the relation and difference to our gAP module, we first revisit CAM and Grad-CAM. Selvaraju et al (2020) have proofed that Grad-CAM is a strict generalization of CAM. Without loss of generality, we consider the same network discussed in (Zhou et al, 2016). For an image classification CNN, the CNN features $\mathbf{F}^L$ of the last convolutional layer[1] are spatially pooled using the global average pooling layer to obtain feature vectors. The network performs a **linear combination** of the feature vectors by feeding them into a fully connected layer before the softmax. Let C denote the number of classes. The classification score before softmax S^c for each class $c \in \{1, 2, \ldots, C\}$ is

$$\begin{aligned} S^c &= \sum_k^{K^L} w^c_k \overbrace{\frac{1}{Z^L} \sum_x^{H^L} \sum_y^{W^L}}^{\text{global average pooling}} \mathbf{F}^L_{k,x,y} \\ &= \frac{1}{Z^L} \sum_x^{H^L} \sum_y^{W^L} \sum_k^{K^L} w^c_k \mathbf{F}^L_{k,x,y}, \end{aligned} \tag{6}$$

[1] We index the last convolutional layer as L.

where w_k^c is the weight connecting the k^{th} feature map with the c^{th} class. The contribution of a feature $\mathbf{F}_{k,x,y}^L$ to S^c is $w_k^c \mathbf{F}_{k,x,y}^L$. CAM generates a class activation map M^c by summing over all feature maps,

$$\mathbf{M}^c = \sum_k^{K^L} w_k^c \mathbf{F}_k^L, \tag{7}$$

where each value in $\mathbf{M}^c$ indicates the contribution of each spatial location to S^c.

For the linear function, the importance weight is also equal to the gradient. Thus, we can also obtain the weight by computing the back-propagating gradients,

$$w_k^c = \sum_x^{H^L} \sum_y^{W^L} \frac{\partial S^c}{\partial \mathbf{F}_{k,x,y}^L}. \tag{8}$$

The detailed derivations of w_k^c is depicted in (Selvaraju et al, 2020). Eqn. (8) is also the way that Grad-CAM computes the weight w_k^c. A little difference is that Grad-CAM multiplies w_k^c by a proportionality constant, *i.e.,*

$$w_k^c = \frac{1}{Z^L} \sum_x^{H^L} \sum_y^{W^L} \frac{\partial S^c}{\partial \mathbf{F}_{k,x,y}^L}, \tag{9}$$

where the proportionality constant $1/Z^L$ will be normalized.

Considering the scores $\{S^c\}$ as CNN features with C channels and spatial size 1×1, *i.e.,* $\mathbf{F}_{c,1,1}^{L+1} = S^c$, we can plug Eqn. (2) into Eqn. (9) and get

$$w_k^c = \frac{1}{Z^L} \sum_x^{H^L} \sum_y^{W^L} \mathbf{G}_{k,x,y}^L. \tag{10}$$

Due to the global average pooling layer, the gradient of each element in $\mathbf{F}_k^L$ is the same, *i.e.,* $\forall_{x,y}, \mathbf{G}_{k,x,y}^L = w_k^c$. The class activation map $\mathbf{M}^c$ can be written as

$$\mathbf{M}^c = \sum_k^{K^L} \mathbf{G}_k^L \cdot \mathbf{F}_k^L = \sum_k^{K^L} \mathbf{A}_k^L. \tag{11}$$

Eqn. (11) suggests that the activation map $\mathbf{M}^c$ for Grad-CAM can be generated by simply adding the activations maps $\mathbf{A}_k^L$ from our gAP.

The differences between gAP and Grad-CAM/CAM are:

- Grad-CAM/CAM combine all activation maps to generate a single class activation map $\mathbf{M}^c$, which highlights important regions supporting the prediction. Our gAP method explains a decision of interest by generating a group of activation maps $\{\mathbf{A}_k\}$. Each activation map corresponds to a feature channel, which is crucial for our iterative decomposition process.
- Grad-CAM/CAM generate class activation maps from the last convolutional layer to explain the prediction. Our gAP generalizes this idea and iteratively decomposes a decision of any CNN layer to its lower layer.

While the above derivations apply to adjacent layers, we empirically find that satisfactory decomposition results can also be obtained when applying the gAP module between two layers from different stages of CNN (see Sec. 4.1). In the following, we will describe how we build hierarchical explanations for the network decisions.

3.2 Hierarchical Decomposition

In Fig. 5, we demonstrate an example of the hierarchical decomposition process. First, we decompose the network decision to the last convolutional layer and find the top few most crucial supporting features. Then, we decompose each of the supporting features to their previous layer and iteratively repeat the decomposition process until the bottom layer. As mentioned in Sec. 1, the key challenge is that naively building the hierarchical decomposition will generate too many visualizations, which will be a cognitive burden for humans. Even if we only decompose a single maximal contributed feature in each channel (see also Eqn. (5)), directly decomposing all channels in VGG-16 will generate $512^3 \times 512^3 \times 256^3 \times 128^2 \times 64^2 \approx 2.5 \times 10^{22}$ visualizations.

To obtain human-scale visualizations, we propose two strategies to reduce the number of visualizations. Firstly, we only decompose the top few most important feature at each layer. Experiment (see Sec. 4.1) has verified that a small subset of feature channels in a layer accounts for the majority of the contributions to a decision. Thus, we select the top few most important channels. We simplify the top-down decision decomposition

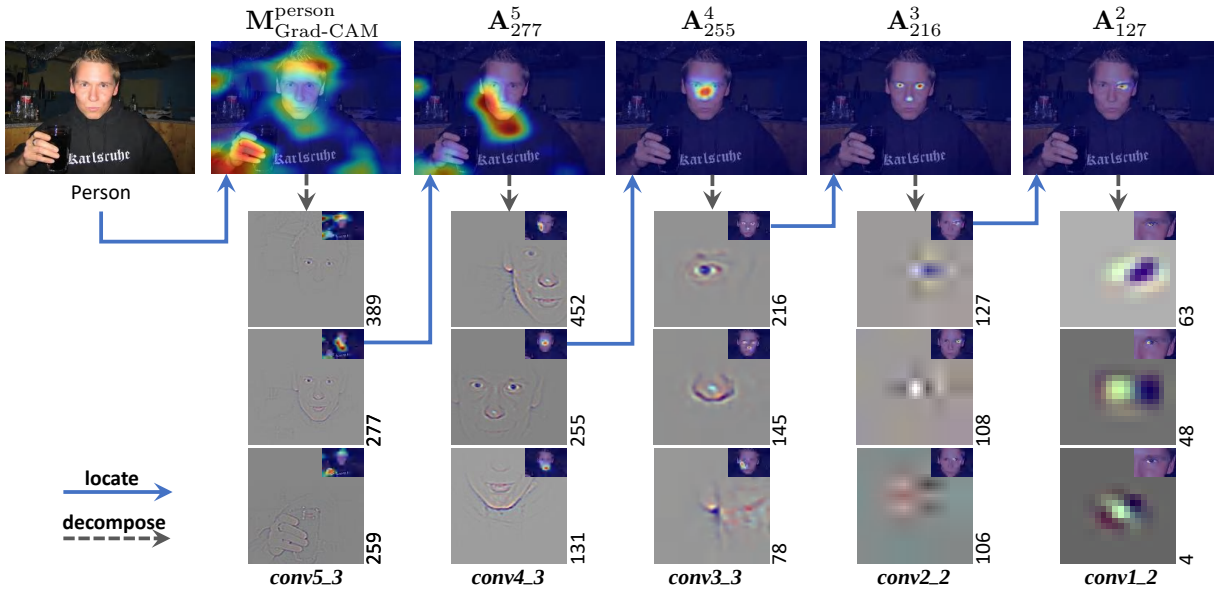


Fig. 5 Illustration of our hierarchical decomposition process. The number for each visualization denotes the channel id of VGG-16 (Simonyan and Zisserman, 2015). At each stage, we decompose one of the top-3 most important features to the lower layer. Following the blue line, we zoom in an activation map for the decision. The gray dash line represents the decomposition of the feature response corresponding to the maximal activation. Additionally, we also make the decomposition process interactive. At each stage, the user can select any decision and decompose it.

process by utilizing the last convolutional layer of each stage. Current popular CNNs (Simonyan and Zisserman, 2015; He et al, 2016) usually reduce the spatial size of feature maps after each stage, where a stage is composed of a set of convolution layers with the same output resolution. Each stage learns different patterns, such as blobs/edges, textures, and object parts (Springenberg et al, 2015; Zeiler and Fergus, 2014). Experiments verify that when using the gAP module between two layers from two consecutive stages, we can obtain visually meaningful decomposition results (see Fig. 5). By these two strategies, we can largely reduce the number of visualizations to obtain human-scale explanations.

An example of the VGG-16 classification network is shown in Fig. 5. We select `conv1_2`, `conv2_2`, `conv3_3`, `conv4_3`, `conv5_3` and index these layers as $\{1, 2, \ldots, L\}$, where $L = 5$. The network output before softmax could be considered as the 6^{th} CNN layer, with features $\mathbf{F}^6 \in \mathbb{R}^{C\times 1\times 1}$. The decomposition process starts from the CNN decision $\mathbf{F}^6_c$, where c corresponds to the 'person' class. Using gAP, we first decompose the CNN decision $\mathbf{F}^6_c$ to 5^{th} layer. The decomposition generates a set of activation maps $\{\mathbf{A}^L\}$ at 5^{th} layer for $\mathbf{F}^6_c$. We use Eqn. (4) to select the top N (*e.g.*, N=3) important activation maps, *i.e.*, $\mathbf{A}^5_{389}$, $\mathbf{A}^5_{277}$, and $\mathbf{A}^5_{259}$. We continue to decompose the decisions from $\mathbf{F}^5_{389}$, $\mathbf{F}^5_{277}$, $\mathbf{F}^5_{259}$, and find the top N most important activation maps at 4^{th} layer for them, respectively. However, directly decomposing the feature map is not easy. Because not all of the features in a feature map contribute to decision (see the activation maps in the top row of Fig. 4). We select the most representative feature that contributes most to a decision and decompose this feature. We utilize Eqn. (5) to find the feature $\mathbf{F}^l_{k,\hat{x},\hat{y}}$ corresponding to the maximum activation. Then we decompose it to layer $l-1$ using gAP. This hierarchical decomposition process recursively runs until we decompose the CNN decision to the lowest layer.

The number of visualizations N is a flexible parameter, which controls how many top response feature channels will be selected during each decomposition. To make human cognition easier, N is set to 3 in Fig. 5. Moreover, we make the hierarchical decomposition interactive, so that the users can choose the features to be decomposed, easily accessing the information they need. We also provide a video about the interactive demo, shown in supplementary materials. In Fig. 5, we can see that the features

Table 1 The Pearson correlation coefficient (PCC) of different settings. → denotes the decomposition. **S5-S1** denotes the last convolutional layer of 5 different stages in VGG-16 (Simonyan and Zisserman, 2015). **AA**: Average Activation. **MA**: Maximum Activation. **AG**: Average Gradient. **MG**: Maximum Gradient. **T**: target category. Average activation achieves the best result.

ILSVRC	T → S5	S5 → S4	S4 → S3	S3 → S2	S2 → S1
AA	0.985	0.959	0.933	0.898	0.895
MA	0.897	0.912	0.894	0.864	0.890
AG	0.623	0.421	0.497	0.545	0.472
MG	0.454	0.456	0.567	0.594	0.606

VOC	T → S5	S5 → S4	S4 → S3	S3 → S2	S2 → S1
AA	0.987	0.961	0.932	0.899	0.893
MA	0.917	0.913	0.892	0.856	0.897
AG	0.702	0.492	0.525	0.564	0.480
MG	0.575	0.525	0.536	0.583	0.669

detected in high-level layers can be decomposed to different parts detected in low-level layers. The hierarchical decomposition process tracks important features and recursively explains the evidence using evidence from lower layers. For instance, the classification results of 'person' have been decomposed to 'face' and 'hand' evidence. The 'face' evidence is then decomposed to 'eye', 'nose', and 'lower jaw'. This process continues until we reach the lowest layer, which usually detects edge and blob features.

Advantages over other attribution methods. First, the hierarchical decomposition can provide more valuable stronly-correlated evidence from different layers, which other current attribution methods cannot provide. Second, the hierarchical decomposition is an interactive tool, where the users can freely select the top activations in different channels to be decomposed. The hierarchical decomposition helps the user easily analyze the state of the intermediate features.

4 Experiments

In this section, we first conduct experiments to verify the correctness and efficiency of the decision decomposition. Then, we use the hierarchical decomposition process to analyze network characteristics and explain network decisions. We conduct experiments on two popular datasets, ILSVRC (Russakovsky et al, 2015) and PASCAL VOC (Everingham et al, 2015). On the PASCAL VOC dataset, the augmented training set containing 10582 training images is used to fine-tune different classification networks. All the experiments are tested on a single RTX 2080Ti GPU.

4.1 Sanity Check for gAP

The effectiveness of gAP. We have shown that the gradient-based Activation Propagation (gAP) module helps to decompose the network decision hierarchically for the CNN-based models. During the decomposition process, what matters most is the accuracy of the channel contributions calculated by the gAP module. Thus, we first examine the accuracy of the channel contributions to the decision of interest. Following (Dhamdhere et al, 2019; Zhang et al, 2019; Bau et al, 2020), we take the decision score drop, when removing a feature channel at a time, as the ground truth of the channel's contribution. Specifically, given an input image I, let f^{l+1} be a decision score at the $l+1^{th}$ layer. $\hat{f}^{l+1}$ denotes the decision score when setting the k^{th} feature channel in the l^{th} layer to the average activation. The score drop $\hat{\alpha}_k^l = f^{l+1} - \hat{f}^{l+1}$ denotes the k^{th} channel's ground truth contribution to this decision.

The Pearson Correlation Coefficient (PCC) metric (Benesty et al, 2009) is utilized to measure the linear correlations between ground truth contribution $\hat{\alpha}^l \in \mathbb{R}^{K^l}$ and the contribution $\alpha^l \in \mathbb{R}^{K^l}$ estimated by Eqn. (4). When the PCC value

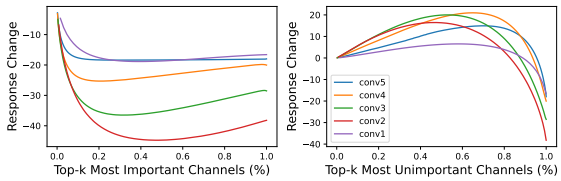


Fig. 6 Comparisons of the feature response change when removing top-k most important feature channels and top-k most unimportant feature channels, respectively.

equals 1, there are linear correlations between the two variables (0 denotes no linear correlations, -1 denotes total negative linear correlations). The PCC metric is computed by

$$\rho = \frac{\mathbb{E}\left[(\alpha^l - \mu_{\alpha^l})(\hat{\alpha}^l - \mu_{\hat{\alpha}^l})\right]}{\sigma_{\alpha^l} \cdot \sigma_{\hat{\alpha}^l}}, \tag{12}$$

where μ and σ denote the mean and the standard deviation, respectively.

As shown in Tab. 1, we study several strategies of calculating the contribution of a feature channel to the decision of interest. It can be seen that the contribution computed by averaging activations (*i.e.,* Eqn. (4)) obtains the highest PCC value with the ground truth. For all stages in VGG-16, there are strong linear correlations between the computed contributions and the ground truth. This high correlation verifies the effectiveness of the gAP module.

Besides, we design another experiment to verify the accuracy of the contributions computed by gAP. Specifically, we present the feature response change when masking the top-k most important channels and top-k most unimportant channels. As shown in Fig. 6, when masking top-k most important channels, the feature response decreases a lot. However, when masking top-k most unimportant channels, the feature response increases. This indicates that the channel importance computed by gAP is accurate.

Taking the computational efficiency into account, it is rather a time-consuming style of measuring channel contributions by calculating the score drop when removing feature channels iteratively in a layer (Dhamdhere et al, 2019; Zhang et al, 2019; Bau et al, 2020). In comparison, only one backpropagation process is needed when gAP calculates the channel contributions to a decision. On a VGG-16 backbone, calculating the ground truth channel contributions of an

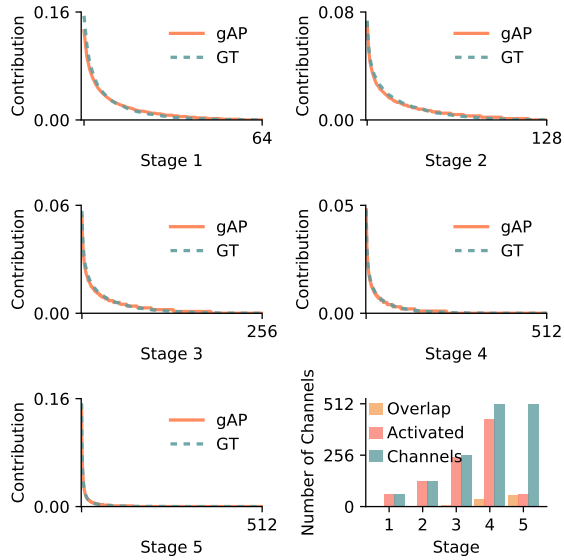
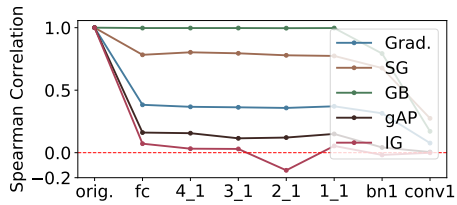


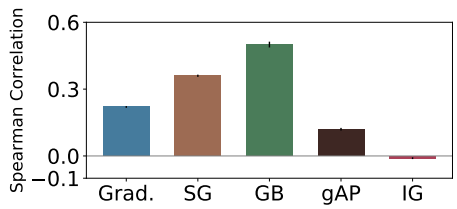
Fig. 7 (a) The first five figures plot the contributions of each channel in a CNN layer to the decision. The contribution of a channel to a decision denotes how much it affects the decision. gAP denotes the proposed gradient-based activation propagation method. GT denotes the method that removes feature channels. 'Stage 1-5' denotes the last convolutional layer of each stage. The channel contributions are sorted in descending order. The contribution distribution calculated by gAP keeps almost the same as that of the ground truth. Besides, the contribution distribution in a layer is long-tailed. (b) The last chart plots the number of the activated channels and the number of all channels at different layers. In high-level layers, there are many activated channels with similar effects to a decision.

image takes about 10s, while the gAP module only takes about 50ms, nearly 200x faster. With the efficiency advantages of the gAP module, our hierarchical decomposition process can immediately yield detailed explanations of a network decision.

The distribution of contributions. As shown in the first five curves of Fig. 7, we can observe that the distribution of the channel contributions in a CNN layer is long-tailed. A small number of feature channels play the most important role for a decision of interest. With deeper layers of the networks, the proportion of the important feature channels decreases. In high-level layers, the feature channels are usually more discriminative. This fact is in line with the accepted notion (Zeiler and Fergus, 2014). Besides, we also check how many feature channels at a CNN layer work together to determine a decision for the higher CNN layers. We call the channel with $\alpha_k^l > 0$ as



(a) model randomization test



(b) data randomization test

Fig. 8 Sanity check for different attribution methods using cascading model parameter and data randomization test. SG: SmoothGrad (Smilkov et al, 2017), GB: Guided Backprop (Springenberg et al, 2015), IG: Integrated Gradient (Sundararajan et al, 2017). The spearman rank correlation metric (Sedgwick, 2014) is used to measure the correlation between the attribution maps of the original model and the randomizing model. Low correlation means the attribution method is sensitive to the model parameters and the data labeling, and thus suitable for explaining the model decisions. Our gAP obtains low correlation values in these two tests. Best viewed with zoom in.

the activated channels and compute the number of the activated channels in the decision decomposition process. As shown in the last chart of Fig. 7, when decomposing a decision from layer `conv2_2` to layer `conv1_2`, nearly all channels in layer `conv1_2` are found activated. However, for the decomposition from the final decision to layer `conv5_3`, we can see that the activated channels' number is much less than the number of all channels in layer `conv5_3`.

Channel-effect overlaps. Using the gAP module, we observe that the activation maps of some channels decomposed from the same decision often have strong activations in similar spatial locations. Such spatial locations usually denote an underlying concept (Bau et al, 2017; Fong and Vedaldi, 2018), contributing to the decision. When presenting visualizations of the hierarchical decomposition, we will merge these duplicate channels with similar effects for human better understanding. Specifically, when decomposing a decision of interest into the lower layer, we will obtain activation maps corresponding to each channel in this layer. We first threshold the activation maps into binary masks and then compute Intersection-over-Union (IoU) between them. Then we apply the non-maximum suppression algorithm (Neubeck and Van Gool, 2006) to suppress activation maps with an IoU score larger than 0.9, where the activation maps are sorted using the contribution scores by Eqn. (4). As shown in Fig. 7, we present how many activated channels have large overlaps with each other. In low-level layers, the number of activated channels with large overlaps is very small. But in high-level layers, there are many activated channels with similar effects to a decision.

Sanity checks for gAP. Adebayo et al (2018) propose the model parameter and data randomization test for sanity check for visual attribution methods. These two tests are used to check whether the attribution method is sensitive to the model parameters and the labeling of the data. An attribution method insensitive to the model parameters and data labels is inadequate for debugging the model and explaining the mechanism that depends on the relationship between the instances and the labeling of the data. To generate saliency maps from our gAP, we hierarchically decompose the decision until the data layer and sum all the gradients from each decomposition. We do the model parameter randomization test on the pretrained ResNet-18 model (He et al, 2016) and randomly initialize the model parameters from the top layer to the bottom layer in a cascading manner. We utilize the spearman rank correlation metric to compute the difference between the attribution maps from the original model and the randomly initialized model. Besides, we do the data randomization test by comparing the saliency maps from CNNs trained with true labels and permuted labels, respectively.

In Fig. 8(a), the low spearman metric indicates that the attribution maps from the original model and the randomly initialized model differ substantially, which demonstrates that gAP is sensitive to model parameters. In Fig. 8(b), the low spearman metric also indicates gAP is sensitive to the labeling of the data. The experimental results verify that our method can be used for debugging models. The visual comparisons are shown in supplementary materials.

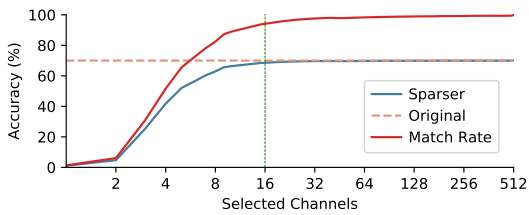


Fig. 9 The classification accuracy of the sparser model generated by gAP and the original model. The match rate denotes the prediction agreement between the sparser model and the original model.

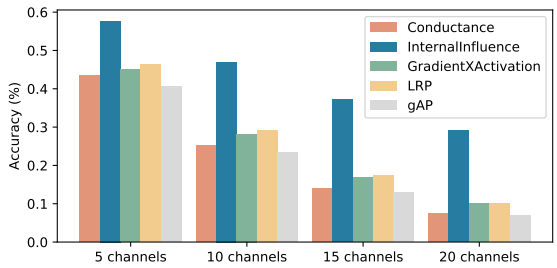


Fig. 10 Comparisons of the classification accuracy after removing the top few most important feature channels (the lower the better). The y-axis is the classification accuracy on the ILSVRC validation set (Russakovsky et al, 2015). The x-axis means the number of important feature channels to ablate. Conductance: Dhamdhere et al (2019), InternalInfluence: Leino et al (2018), LRP: Bach et al (2015).

Is the top-k decomposition a good approximation to the original model? We have tested the classification accuracy of the sparser surrogate model generated by gAP. Moreover, we measure the match rate by comparing the predictions between the sparser surrogate model and the original model. Specifically, we decompose from the decision of the predicted category to the bottom layers and select the top-k important features in each decomposition to make predictions. We mask those unimportant feature channels from different layers and reinput the image to obtain new predictions. As shown in Fig. 9, when using top-16 decomposition, the sparser surrogate model has a similar classification accuracy to the original model. According to the match rate, when using top-16 decomposition, the predictions of the sparser surrogate model and the original model are consistent on almost all samples. The sparser surrogate model selecting a small number of feature channels can make a good approximation to the original model.

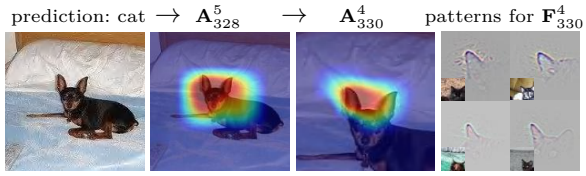
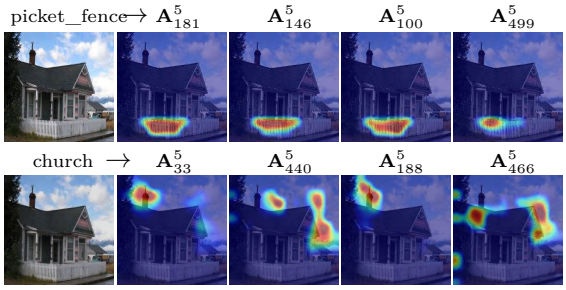


Fig. 11 Analysis of a failure example. The leftmost is the dog image that is misclassified to the *cat* category. We decompose the network decision to layer `conv4_3`. The rightmost is the patterns that maximumly activate the 330^{th} channel. For this example, the channels sensitive to the *cat* category's attribute have strong activations, causing VGG-16 to make a wrong decision.

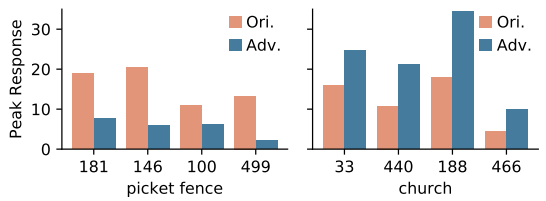
Comparison with individual-based methods. The individual-based methods (Dhamdhere et al, 2019; Leino et al, 2018) compute the importance of each channel from different layers to the final network decision. Compared with individual-based methods, gAP can help us explore the relationships among different feature channels. To directly compare with them, we propagate the importance of each selected channel of the top layer to the shallow layer. We select the top-N most important channels from different layers of VGG-16 and ablate them to watch the change of the classification accuracy. We conduct experiments on the ILSVRC validation set (Russakovsky et al, 2015). As shown in Fig. 10, when removing the top few most important feature channels, gAP obtains lower classification accuracy than other individual-based methods. We analyze that gAP only propagates the contributions of those important feature channels to lower layers, which reduces the interference of other feature channels. Compared with the individual-based methods, gAP can not only effectively detect the important features but also how these features affect each other.

4.2 Diagnosing CNN

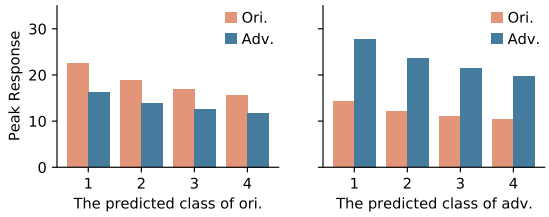
Analyzing failure predictions of CNN. Previous work (Selvaraju et al, 2020) can generate class activation maps for the network predictions, highlighting the most important image regions supporting the network decision. However, such an explanation is not informative enough. The hierarchical decomposition can further provide a more detailed explanation for the network decision. We decompose the network's decision iteratively to the low-level layers and find the most



(a) Decomposition



(b) Peak Response



(c) Average Peak Response

Fig. 12 Example of the adversarial attacks. (a) The top is the original image, and the bottom is the adversarial image. $\rightarrow$ denotes the decomposition. (b) plots the peak feature responses of the most important channels for the network decision in the original image and adversarial image. (c) plots the average peak feature responses of the top 4 most important channels for the network decision in the original image and adversarial image over the whole ILSVRC validation dataset (Russakovsky et al, 2015). The peak values of the important channels for the correct category largely decrease, and those for the wrong category increase by a large margin.

important feature channels at different layers. We can see each channel's contribution to the network decision. Further, important channels and their corresponding activation maps can also be studied.

As shown in Fig. 11, we use the hierarchical decomposition to examine the CNN's wrong decision. Fig. 11 demonstrates a failure case. A dog image misclassified to the *cat* category with a probability of 99%. We first decompose the network decision to layer `conv5_3` and find the most important channel, *i.e.,* the 328^{th} channel, with a 32.3% contribution. We further present the decomposition from channel 328^{th} to layer `conv4_3` and find the most important channel, *i.e.,* the 330^{th} channel. The activation map $\mathbf{A}^4_{330}$ has strong activations at the ear region. Moreover, the patterns that maximumly activate the 330^{th} channel are the ear image patches of the *cat* category. We find the dog's ear of this example has a similar shape to those ear image patches of the *cat* category. We further occlude the image region of the dog's ear and observe that the CNN correctly predicts the *dog* category with a probability of 65%. With the hierarchical decomposition, we found that CNN makes the wrong decision because it takes the dog's ear as the cat's ear in this example.

Analyzing adversarial attacks. Current CNN models are vulnerable to adversarial attacks. When the adversarial attack algorithms add a small perturbation to the original images, these CNN models easily misclassify them. To understand how the adversarial images successfully fool the CNN models, following (Bau et al, 2020), we study the change of the feature responses for important channels. As shown in Fig. 12(a), we present the original image (top row) and the adversarial image (bottom row). The adversarial image is generated by a popular attack algorithm (Madry et al, 2018). VGG-16 classifies the original image to the *picket_fence* category (probability 92%) and the adversarial image to the *church* category (probability 100%). Through our decomposition from the network decision to layer `conv5_3`, we find the top few most important feature channels for the *picket_fence* and *church* category, respectively.

As shown in Fig. 12(b), when comparing the adversarial image to the original image, we observe that the peak feature responses of important channels for the *picket_fence* category, *i.e.,* the 181^{st}, 146^{th}, 100^{th}, 499^{th} channels, largely decrease by 11.3, 14.5, 4.7, and 11.1. However, the peak feature responses of important channels for the *church* category, *i.e.,* the 33^{rd}, 440^{th}, 188^{th}, 466^{th} channels, largely increase by 8.7, 10.4, 16.4, and 5.5.

As shown in Fig. 12(c), we also compute the average peak responses of important channels on the whole ILSVRC validation dataset

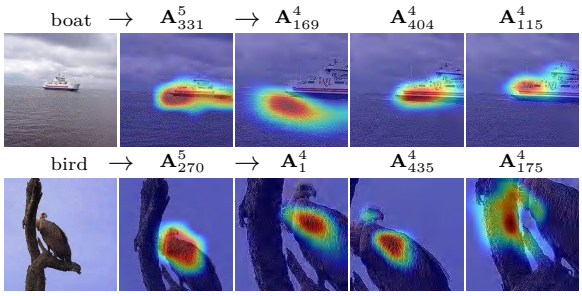


Fig. 13 The context information in the activation maps. → denotes the decomposition.

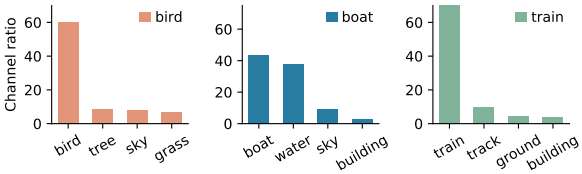


Fig. 14 Context information for each category in the PASCAL VOC dataset.

(Russakovsky et al, 2015). The adversarial attack algorithms change the feature responses of important channels to affect the final network decision. For the important channels, they reduce the correct category's feature responses and increase the wrong category's feature responses.

The context in activation maps. Context information (Oquab et al, 2015; Kumar and Hebert, 2005) is crucial for recognition. A known prior is that the target category usually appears in a specific context. For example, the boats usually appear in the seas or lakes, and the birds often stand on the tree branch. Through our decision decomposition, we find some context in the activation maps to support the CNN prediction. Fig. 13 shows that the 331^{st} channel in layer `conv5_3` has strong responses to the image's 'boat' region. We decompose the peak point indicated by the activation map to layer `conv4_3`. The 169^{st}, 404^{th}, and 115^{th} channels are the top-3 most important channels. The most important channel is the 169^{st} channel, whose corresponding activation map locates the sea.

To quantitatively analyze the context information contained in the activation maps, we utilize the PASCAL-Context dataset (Mottaghi et al, 2014) for evaluation. We select the images with context annotations from the PASCAL VOC validation set (Everingham et al, 2015) and compute the most frequent context labels for each category. Specifically, we perform the hierarchical decomposition to layer `conv4_3`, obtaining the activation map for each selected channel. The activation map is first thresholded to a binary map. Then we compute the IoU between the binary activation map and each context region. The activation map is assigned with the label of the context region corresponding to the largest IoU. In Fig. 14, we have shown the top few most frequent context labels for three categories, *i.e.,* bird, boat, and train. These categories usually appear in a specific environment. This fact suggests that the context of the objects is critical for recognition. The context information of other categories and the qualitative examples are shown in the supplementary materials.

Channel discrimination analysis. We utilize the hierarchical decomposition to explore the discriminative information of the channels in different layers. Specifically, we define a discriminative degree D to measure the discriminative information of a channel. When performing the hierarchical decomposition process for the images with label c, we count the number of times N_c for channel k when its contribution to a decision ranking top-3. N_c is summed on all images from the validation set. Then the discriminative degree D is computed by

$$D = \frac{\max_c N_c}{\sum_{c=1}^{C} N_c}, \tag{13}$$

where C denotes the number of categories in the dataset. When the feature in channel k is only decomposed from one single category, the discriminative degree $D = 1$. Besides, we can get the minimum value of D when the feature decomposed from each category with equal times: $D = 1/C$.

We apply the hierarchical decomposition to different CNNs. As shown in Fig. 15, the channels' discriminative degrees in low-level layers are very small. They usually have strong activations for multiple categories. This fact indicates that the basic features detected by channels in low-level layers are shared among different categories, which lacks discriminative information for classification. However, in high-level layers of CNNs, the

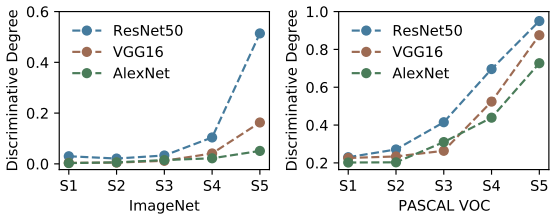


Fig. 15 The discriminative degrees of the disentangled channels from different layers of different CNNs on the validation set of PASCAL VOC (Everingham et al, 2015) and ILSVRC (Russakovsky et al, 2015).

channels' discriminative degrees are much larger than those in low-level layers. Because the high-level layers in CNNs gradually combine basic features from low-level layers to form more discriminative features. In high-level layers, different categories tend to highlight their own discriminative channels. These results provide additional evidence for the conclusion found by Zeiler and Fergus (2014).

Moreover, for the high-level layers of different CNNs, the discriminative degrees of the channels gradually increase with the growth of the network depth (ResNet-50 (He et al, 2016) > VGG-16 (Simonyan and Zisserman, 2015) > AlexNet (Krizhevsky et al, 2012)). Such difference suggests that the high-level layers of ResNet-50 have a stronger discriminative ability. The strong discriminative ability of the channels can effectively reduce confusion among different categories, which helps ResNet-50 achieve higher classification accuracy than VGG-16 and AlexNet.

5 Limitation

The proposed hierarchical decomposition method explains the individual decision by selecting a set of strongly correlative channels from different layers of CNN. These feature channels provide a rich hierarchy of evidence. However, the feature channels are less confident for an unprofessional user to understand the network's reasoning process because not all examples are as easy to understand as the person image. So in the future, we will attempt to build connections between the selected feature channels and human-specific concepts for better human understanding.

Besides, following (Dhamdhere et al, 2019; Bau et al, 2020), we have removed channels individually to study their contributions. However, as verified in (Fong and Vedaldi, 2018; Leavitt and Morcos, 2020), the representations are usually distributed among multiple channels. We observe that the activation maps of some channels decomposed from the same decision often have strong activations in similar spatial locations. This phenomenon suggests that multiple feature channels produce class responses together. One possible solution to the flaw of removing channels individually is that we can first find those feature channels with similar effects by measuring the overlap between their corresponding activation maps. Then we analyze these feature channels together to the network decision. In this paper, we focus on building the evidence hierarchy. The issue of removing individual channels will be our future work.

The proposed hierarchical decomposition method selects and decomposes the most representative feature in each channel, which may miss some important features. In most cases, the representative features are enough for decision explanations. However, some potentially important evidence might be missed. How to measure the number of missed features and evaluate their importance to the decision will be our future work.

6 Conclusion

We present a novel gradient-based activation propagation (gAP) scheme that can decompose any CNN layer's decision to its lower layers. Based on the gAP, the network decision can be hierarchically decomposed to a rich set of the evidence pyramid associated with all layers of the CNN model. Our method allows users to delve deep into the CNN's decision-making process in a top-down manner. We have experimentally verified the effectiveness of our method and demonstrated its ability to understand and diagnose CNN predictions. While currently mostly focus on explaining CNN-based image classifiers, we will study how to generalize the framework to other tasks and other deep learning models in the future. The source code and interactive demo website will be made publicly available.

Acknowledgment

This research was supported by the Major Project for New Generation of AI under Grant No. 2018AAA0100400, NSFC (61922046), and the Fundamental Research Funds for the Central Universities (Nankai University, NO. 63223050).

References

Adebayo J, Gilmer J, Muelly M, et al (2018) Sanity checks for saliency maps. In: Adv. Neural Inform. Process. Syst.

Athalye A, Engstrom L, Ilyas A, et al (2018) Synthesizing robust adversarial examples. In: Int. Conf. Mach. Learn.

Bach S, Binder A, Montavon G, et al (2015) On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one 10(7):e0130,140

Baehrens D, Schroeter T, Harmeling S, et al (2010) How to explain individual classification decisions. The Journal of Machine Learning Research 11:1803–1831

Bau D, Zhou B, Khosla A, et al (2017) Network dissection: Quantifying interpretability of deep visual representations. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 6541–6549

Bau D, Zhu JY, Strobelt H, et al (2020) Understanding the role of individual units in a deep neural network. Proceedings of the National Academy of Sciences

Benesty J, Chen J, Huang Y, et al (2009) Pearson correlation coefficient. In: Noise reduction in speech processing. Springer, p 1–4

Chattopadhay A, Sarkar A, Howlader P, et al (2018) Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In: IEEE Winter Conf. Appl. Comput. Vis., pp 839–847

Chen C, Li O, Tao D, et al (2019a) This looks like that: Deep learning for interpretable image recognition. In: Adv. Neural Inform. Process. Syst., pp 8930–8941

Chen LC, Papandreou G, Kokkinos I, et al (2017) Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Trans Pattern Anal Mach Intell

Chen R, Chen H, Ren J, et al (2019b) Explaining neural networks semantically and quantitatively. In: Int. Conf. Comput. Vis., pp 9187–9196

Dabkowski P, Gal Y (2017) Real time image saliency for black box classifiers. In: Adv. Neural Inform. Process. Syst.

Dhamdhere K, Sundararajan M, Yan Q (2019) How important is a neuron? Int Conf Learn Represent

Dosovitskiy A, Brox T (2016) Inverting visual representations with convolutional networks. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 4829–4837

Erhan D, Bengio Y, Courville A, et al (2009) Visualizing higher-layer features of a deep network. University of Montreal 1341(3):1

Everingham M, Eslami SA, Van Gool L, et al (2015) The pascal visual object classes challenge: A retrospective. Int J Comput Vis

Fong R, Vedaldi A (2018) Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 8730–8738

Fong R, Patrick M, Vedaldi A (2019) Understanding deep networks via extremal perturbations and smooth masks. In: Int. Conf. Comput. Vis., pp 2950–2958

Fong RC, Vedaldi A (2017) Interpretable explanations of black boxes by meaningful perturbation. In: Int. Conf. Comput. Vis., pp 3429–3437

Frosst N, Hinton G (2017) Distilling a neural network into a soft decision tree. In: CEX workshop at AIIA

Giannelli PC (1982) Chain of custody and the handling of real evidence. Am Crim L Rev 20:527

Girshick R (2015) Fast r-cnn. In: Int. Conf. Comput. Vis., pp 1440–1448

Girshick R, Donahue J, Darrell T, et al (2014) Rich feature hierarchies for accurate object detection and semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 580–587

Goodfellow IJ, Shlens J, Szegedy C (2014) Explaining and harnessing adversarial examples. In: Int. Conf. Learn. Represent.

He K, Zhang X, Ren S, et al (2016) Deep residual learning for image recognition. In: IEEE Conf. Comput. Vis. Pattern Recog.

Hou Q, Jiang P, Wei Y, et al (2018) Self-erasing network for integral object attention. In: Adv. Neural Inform. Process. Syst.

Hou Y, Ma Z, Liu C, et al (2019) Learning lightweight lane detection cnns by self attention distillation. In: Int. Conf. Comput. Vis., pp 1013–1021

Huang G, Liu Z, Van Der Maaten L, et al (2017) Densely connected convolutional networks. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 4700–4708

Jiang PT, Zhang CB, Hou Q, et al (2021) Layercam: Exploring hierarchical class activation maps for localization. IEEE Trans Image Process 30:5875–5888

Jiang PT, Han LH, Hou Q, et al (2022) Online attention accumulation for weakly supervised semantic segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 44(10):7062–7077. https://doi.org/10.1109/TPAMI.2021.3092573

Kim B, Wattenberg M, Gilmer J, et al (2018) Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: Int. Conf. Mach. Learn., PMLR, pp 2668–2677

Kim B, Seo J, Jeon S, et al (2019) Why are saliency maps noisy? cause of and solution to noisy saliency maps. In: IEEE ICCVW, IEEE, pp 4149–4157

Kindermans PJ, Schütt KT, Alber M, et al (2018) Learning how to explain neural networks: Patternnet and patternattribution. In: Int. Conf. Learn. Represent.

Koh PW, Nguyen T, Tang YS, et al (2020) Concept bottleneck models. In: Int. Conf. Mach. Learn., pp 5338–5348

Krizhevsky A, Sutskever I, Hinton GE (2012) Imagenet classification with deep convolutional neural networks. In: Adv. Neural Inform. Process. Syst., pp 1097–1105

Kumar N, Berg AC, Belhumeur PN, et al (2009) Attribute and simile classifiers for face verification. In: Int. Conf. Comput. Vis., pp 365–372

Kumar S, Hebert M (2005) A hierarchical field framework for unified context-based classification. In: Int. Conf. Comput. Vis., IEEE, pp 1284–1291

Kurakin A, Goodfellow I, Bengio S (2017) Adversarial examples in the physical world. In: Int. Conf. Learn. Represent. Worksh.

Lampert CH, Nickisch H, Harmeling S (2009) Learning to detect unseen object classes by between-class attribute transfer. In: IEEE Conf. Comput. Vis. Pattern Recog., IEEE, pp 951–958

Leavitt ML, Morcos AS (2020) Selectivity considered harmful: evaluating the causal impact of class selectivity in dnns. In: Int. Conf. Learn. Represent.

Leino K, Sen S, Datta A, et al (2018) Influence-directed explanations for deep convolutional networks. In: 2018 IEEE International Test Conference (ITC), IEEE, pp 1–8

Lin G, Milan A, Shen C, et al (2017) Refinenet: Multi-path refinement networks with identity mappings for high-resolution semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog.

Litjens G, Kooi T, Bejnordi BE, et al (2017) A survey on deep learning in medical image analysis. Medical image analysis 42:60–88

Liu X, Wang X, Matwin S (2018) Improving the interpretability of deep neural networks with knowledge distillation. In: IEEE Int. Conf. Data Mining Worksh., IEEE, pp 905–912

Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE Conf. Comput. Vis. Pattern Recog.

Madry A, Makelov A, Schmidt L, et al (2018) Towards deep learning models resistant to adversarial attacks. In: Int. Conf. Learn. Represent.

Mahendran A, Vedaldi A (2015) Understanding deep image representations by inverting them. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 5188–5196

Montavon G, Lapuschkin S, Binder A, et al (2017) Explaining nonlinear classification decisions with deep taylor decomposition. Pattern Recognition 65:211–222

Mordvintsev A, Olah C, Tyka M (2015) Inceptionism: Going deeper into neural networks

Mottaghi R, Chen X, Liu X, et al (2014) The role of context for object detection and semantic segmentation in the wild. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 891–898

Murad MH, Asi N, Alsawas M, et al (2016) New evidence pyramid. BMJ Evidence-Based Medicine 21(4):125–127

Neubeck A, Van Gool L (2006) Efficient non-maximum suppression. In: 18th International Conference on Pattern Recognition (ICPR'06), IEEE, pp 850–855

Olah C, Mordvintsev A, Schubert L (2017) Feature visualization. Distill 2(11):e7

Olah C, Cammarata N, Schubert L, et al (2020) Zoom in: An introduction to circuits. Distill 5(3):e00,024–001

Oquab M, Bottou L, Laptev I, et al (2015) Is object localization for free?-weakly-supervised learning with convolutional neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 685–694

Papandreou G, Chen LC, Murphy KP, et al (2015) Weakly-and semi-supervised learning of a deep convolutional network for semantic image segmentation. In: Int. Conf. Comput. Vis., pp 1742–1750

Petsiuk V, Das A, Saenko K (2018) Rise: Randomized input sampling for explanation of black-box models. In: Brit. Mach. Vis. Conf.

Ren S, He K, Girshick R, et al (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Adv. Neural Inform. Process. Syst., pp 91–99

Ribeiro MT, Singh S, Guestrin C (2016) "Why should i trust you?" Explaining the predictions of any classifier. In: ACM SIGKDD, pp 1135–1144

Ronneberger O, Fischer P, Brox T (2015) U-net: Convolutional networks for biomedical image segmentation. In: Int. Conf. Medical image computing and computer-assisted intervention, pp 234–241

Russakovsky O, Deng J, Su H, et al (2015) Imagenet large scale visual recognition challenge. Int J Comput Vis 115(3):211–252

Sedgwick P (2014) Spearman's rank correlation coefficient. Bmj 349

Selvaraju RR, Cogswell M, Das A, et al (2020) Grad-cam: Visual explanations from deep networks via gradient-based localization. Int J Comput Vis 128(2):336–359

Shrikumar A, Greenside P, Kundaje A (2017) Learning important features through propagating activation differences. In: Int. Conf. Mach. Learn.

Simonyan K, Zisserman A (2015) Very deep convolutional networks for large-scale image recognition. In: Int. Conf. Learn. Represent.

Simonyan K, Vedaldi A, Zisserman A (2014) Deep inside convolutional networks: Visualising image classification models and saliency maps. In: Int. Conf. Learn. Represent. Worksh.

Smilkov D, Thorat N, Kim B, et al (2017) Smoothgrad: removing noise by adding noise. In: Int. Conf. Mach. Learn. Worksh.

Springenberg JT, Dosovitskiy A, Brox T, et al (2015) Striving for simplicity: The all convolutional net. In: Int. Conf. Learn. Represent. Worksh.

Srinivas S, Fleuret F (2019) Full-gradient representation for neural network visualization. In: Adv. Neural Inform. Process. Syst., pp 4124–4133

Sundararajan M, Taly A, Yan Q (2017) Axiomatic attribution for deep networks. In: Int. Conf. Mach. Learn.

Sung A (1998) Ranking importance of input parameters of neural networks. Expert systems with Applications 15(3-4):405–411

Treisman AM, Gelade G (1980) A feature-integration theory of attention. Cognitive psychology 12(1):97–136

Wang H, Wang Z, Du M, et al (2020) Score-cam: Score-weighted visual explanations for convolutional neural networks. In: IEEE Conf. Comput. Vis. Pattern Recog. Worksh., pp 24–25

Yang Y, Qiu J, Song M, et al (2020) Learning propagation rules for attribution map generation. In: Eur. Conf. Comput. Vis., pp 672–688

Yosinski J, Clune J, Nguyen A, et al (2015) Understanding neural networks through deep visualization. In: Int. Conf. Mach. Learn. Worksh.

Zeiler MD, Fergus R (2014) Visualizing and understanding convolutional networks. In: Eur. Conf. Comput. Vis., Springer

Zhang J, Lin Z, Brandt J, et al (2016) Top-down neural attention by excitation backprop. In: Eur. Conf. Comput. Vis.

Zhang Q, Yang Y, Ma H, et al (2019) Interpreting cnns via decision trees. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 6261–6270

Zhao H, Shi J, Qi X, et al (2017) Pyramid scene parsing network. In: IEEE Conf. Comput. Vis. Pattern Recog.

Zhou B, Khosla A, Lapedriza A, et al (2015) Object detectors emerge in deep scene cnns. In: Int. Conf. Learn. Represent.

Zhou B, Khosla A, Lapedriza A, et al (2016) Learning deep features for discriminative localization. In: IEEE Conf. Comput. Vis. Pattern Recog.

Zhou B, Sun Y, Bau D, et al (2018) Interpretable basis decomposition for visual explanation. In: Eur. Conf. Comput. Vis., pp 119–134

Zhu Z, Liang D, Zhang S, et al (2016) Traffic-sign detection and classification in the wild. In: IEEE Conf. Comput. Vis. Pattern Recog., pp 2110–2118

Zintgraf LM, Cohen TS, Adel T, et al (2017) Visualizing deep neural network decisions: Prediction difference analysis. In: Int. Conf. Learn. Represent.