Received September 18, 2020, accepted October 18, 2020, date of publication October 29, 2020,
date of current version November 10, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034625

Measuring Feature Importance of Convolutional Neural Networks

XIAOHANG ZHANG, (Member, IEEE), AND JIUYI GAO
School of Modern Post, Beijing University of Posts and Telecommunications, Beijing 100876, China
Corresponding author: Xiaohang Zhang (zhangxiaohang@bupt.edu.cn)

ABSTRACT Convolutional neural networks have demonstrated powerful abilities to achieve state-of-the-art results in many classification tasks, while the interpretability and reliability of these complicated models are also a non-negligible problem. Understanding how these networks arrive at their final decisions becomes more and more indispensable, so this article puts forward an interpretive method to obtain feature importance, which indicates to what extent an input feature can discriminate different classes. The proposed method utilizes the attribution maps of multiple-class predictions and can decomposes the feature importance into individual and co-variation effects. Some properties of the method are justified theoretically. Furthermore, a visualization method is proposed to sketch the silhouette of the target object. And to improve computational efficiency, some practical tricks are applied. For the purpose of evaluating this method, some comparative experiments are performed; the results testify that the proposed method can identify important features and improve the visualization effects.

INDEX TERMS Class discriminability, convolutional neural networks, model interpretation, feature importance, visualization map.

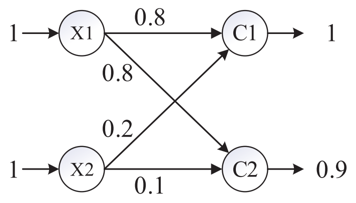
I. INTRODUCTION

In recent years, CNNs have shown promising performance in various tasks, such as image classification [1], [2], face recognition [3], [4], semantic segmentation [5], [6], and object detection [7], [8]. Yet, due to the sophisticated internal structures and operation mechanisms, it is problematic to interpret their predictions. Consequently, how to clarify the behavior of these black-box models and make them more credible and understandable is of considerable significance in areas of healthcare, autonomous driving, bank loans, etc. For example, when doctors use a CNN model to diagnose a patient's medical image, knowing which parts of the image are used to make decisions is the key to judging whether the model diagnosis is correct or wrong. Another case in point is that the generalization ability of a model can be effectively measured by analyzing which input features are used to predict, e.g. the model of using snow to distinguish huskies from wolves does not have strong generalization [9].

With the increasing attention on the interpretation of CNNs, large quantities of researchers have proposed various methods to resolve this issue. Plenty of works have concentrated on explaining an individual prediction, which can be realized by computing the attribution of each input feature to the score of a target class. The approaches include perturbation-based methods [9], [10], backpropagation-based methods [11], [12], gradients-based methods [13], [14], and so on. Most of the attribution methods satisfy the *Conservation theorem* [11] that information will not be lost when it spreads between hidden layers, i.e. the target class score can be completely converted to the sum of the attributions of all input features. However, a common drawback of the attribution methods is that the feature attributions are calculated based on only a single target class that is usually the real class when it is known or the class with maximum prediction probability. Therefore, the attribution methods only use part of the information in neural networks to evaluate feature contributions.

The feature attributions can be used to interpret the model predictions; however, in some cases they cannot give proper interpretation. For example, given a simple neural network model without hidden layers (left in Fig 1), in which the two class outputs are only the linear combination of the two inputs, we can calculate the feature attributions by using the LRP, DeepLift (the baseline is set to 0), perturbation (the baseline is set to 0), input*gradient and integrated gradients techniques, all of which obtain the same feature attributions for the instance (1,1). As shown in right table in Fig 1,

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



	X1	X2
C1	0.8	0.2
C2	0.8	0.1

FIGURE 1. A simple neural network model (left) and the feature attributions (right).

the feature X1 plays more import role than X2 for interpreting the predictions of C1 and C2 because the feature attribution of X1 is larger than that of X2. However, it is obvious that X2 can determine the final decision because X1 contributes equally to the predictions of C1 and C2, and X2 contributes more to C1 than to C2.

This article mainly focuses on how to measure and understand the importance of input features in classification tasks by utilizing all class information in CNNs. The feature importance score indicates to what extent an input feature can discriminate different classes. The main contributions of this research are summarized as follows: 1)

1) The proposed feature importance score is the attribution to the variation of multiple-class predictions instead of the attribution to a single target class prediction; the feature importance score ground on the attribution methods has an improvement in the visualization effect compared with the existing approaches (see Fig 2 (STD)).
2) The feature importance can be decomposed into individual effects and co-variation effects, which solve the limitations of most of current methods that cannot give co-variation effects between features. Moreover, some practical tricks are introduced into the realization of feature importance maps to enhance computational efficiency.
3) A simple visualization method based on the generated feature importance score map is proposed, which can sketch the silhouette of the object related to the prediction in the picture (Fig 2 (Silhouette)).

This article is organized as follows: Section 2 reviews some previous techniques that explain the individual predictions of CNN classification networks; in Section 3, the proposed method in this article is illustrated in detail; to evaluate the quality of this method, some experiments are elaborated in Section 4; finally, Section 5 includes the conclusions and the attention of future work.

II. RELATED WORKS

This section firstly describes some properties that local interpretation methods may have. Then some existing methods of calculating feature attributions to single target class prediction are summarized.

A. PROPERTIES OF LOCAL INTERPRETATION

Because the effects of local interpretation methods are not easy to evaluate based on some quantified criterion, researchers propose some properties that a good local interpretations method should satisfies. We list the properties as in [15].

1) *Sensitivity* (also called *Dummy* in [16] and *Missingness* in [17]). If a model does not depend on some feature, then the attribution of that feature is always zero.
2) *Completeness* (also called *Local Accuracy* in [17], *Efficiency* in [18]). The feature attributions add up to the prediction of model at any input.
3) *Linearity* (also called *Additivity* in [18]). The attributions for a model $\alpha f + \beta g$, a linear combination of two models f and g, should be the weighted sum of the attributions for f and g with weights α and β, respectively.
4) *Symmetry-Preserving* (also called *Symmetry* in [18]). An attribution method is symmetry-preserving, if for all inputs that have identical values for symmetric features, the symmetric features receive identical attributions.
5) *Implementation Invariance*. The attributions are always identical for two functionally networks, whose outputs are equal for all inputs, despite having different implementations.

B. LOCAL INTERPRETATION METHODS

1) PERTURBATION-BASED METHODS

These approaches make ablation to some input features (e.g. some pixels of the input image) and observe the change of the activation of the interested output neuron. *Deconvolutional Network (DN)* [19] occluded some parts of the input image, and the change of the output decision showed which part of the image was more important for prediction. *Prediction Difference Analysis (PDA)* [10] removed multiple input features simultaneously and measured the change of target class score by the difference between $p(c|\mathbf{x})$ and $p(c|\mathbf{x}_{\setminus i})$ under the assumption that the input features are correlated, where $p(c|\mathbf{x})$ represents the probability that the prediction of the input image $\mathbf{x}$ is c and $p(c|\mathbf{x}_{\setminus i})$ represents the probability that the input image $\mathbf{x}$ is predicted as class c when the feature (or pixel) i is removed

2) BACKPROPAGATION-BASED METHOD

These approaches propagate back the class score of a specific output neuron through each hidden layer down to the input variables.

Deconvolutional Network [19] proposed a deconvolution visualization framework where the feature map of each layer could be visualized by mapping back to the input space through unpooling, rectification, and filtering. *Layer-wise Relevance Propagation (LRP)* [11] explained the contribution of each input variable to the target output neuron activation. The attribution map was realized by layer-wise linear proportional distribution with one forward and one backward propagation. *Deep Taylor Decomposition (DTD)* [20] decomposed the activation value for each neuron according to the contribution of its input based on layer-wise Taylor's first-order

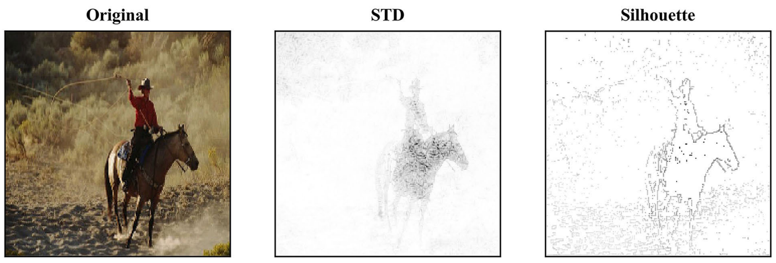


FIGURE 2. **Visualization of the proposed method. The original image is sampled from ImageNet dataset (Original), the feature importance map is obtained by the proposed method (STD), and the silhouette map is based on the feature importance map (Silhouette).**

expansion, but it ignored the negative impact of input elements on the target output score. *DeepLIFT (DL)* [12] compared the activation value of each neuron with its reference value, and converted discrete gradients into feature attributions based on gradient chain rule. The methods of LRP, DTD and DL solve the *saturation* problem and satisfy *sensitivity* ground on explaining the individual prediction without considering the class separation of input features. *PatternNet and PatternAttribution* [21] separated the signals from the distractors during the backward pass, trained a signal estimator for each neuron, and substituted the informative direction generated by the signal estimator for the weight.

3) GRADIENTS-BASED METHODS

These methods construct a saliency map by calculating the derivative of the output class function concerning each input pixel.

Saliency maps (SM) [13] were acquired by differentiating the class scores with respect to the individual input pixel and taking the absolute value. *SmoothGrad (SG)* [14] was achieved by repeatedly adding random Gaussian noises to the input image and averaging the resulting saliency maps. This process produces less noise and better performance in visual coherence and discriminability. *Guided Backpropagation (GB)* [22] proposed that the generated map contained both forward and backward propagation information by combining the gradients-based method and the deconvolution framework to operate ReLU activations in the process of backpropagation. Nevertheless, the essence of SM, SG and GB is to calculate the gradients, so when saturation problem [23] occurs, i.e. the prediction flattens in vicinity of input, the gradient violates *sensitivity* [15].

*Input*Gradients (Inp.*Grad.)* [24] proved that if only linear or approximately linear (e.g. ReLU) functions were used to activate neurons and bias terms were included in activation vectors propagated between hidden layers, then LRP reduced to Inp.* Grad.; this technique improves the resolution of attribution maps and leverages the information of inputs, but dose not solve the saturation problem of Gradients-based methods. *Integrated Gradients* [15] hypothesized that the reference value and the original input value fell on a straight line for each input feature; then calculating the gradient for each point on the line and integrating all the gradients could obtain the input feature relevance score; this method addresses the *saturation* problem and satisfies the *sensitivity* and *implementation invariance* [15].

*Gradient * Input* [25] is a unified framework, under which the occlusion method of DN, ε-LRP, SM, DL, Integrated Gradients can all be realized, and each method has a particular expression corresponding to the gradient of the activation function.

4) ADDITIVE MODEL-BASED METHODS

Local Interpretable Model-agnostic Explanations (LIME) [9] trained a linear separable explainer with interpretable features near an input point through perturbating the explainable feature representations, which locally approximated the behavior of the original model. This method can be widely applied to many machine learning models. *SHapley Additive exPlanations (SHAP)* [17], [18] is based on Shapley value of cooperative game, and can be employed in explaining various models of machine learning and deep learning, and also can produce a unique explanation. SHAP is subject to the properties of *consistency*, *missingness*, and *additivity*.

III. METHODOLOGY

To calculate feature importance, feature attribution maps need to be obtained firstly by some methods mentioned in Section II; then, the attribution maps ground on multiple class scores are utilized to measure the class discriminability of each input feature. In addition, extracting some more essential features from the feature importance map can sketch the silhouette of the target object; thus, the feature selection mechanisms of CNN models can be displayed more visually in prediction processes.

A. FEATURE IMPORTANCE

First of all, let $\mathbf{A}$ be a $K \times N$ matrix of K classes and N features as shown in Table 1. Each row j, denoted as $[A_{j1}, A_{j2}, \ldots, A_{jN}]$ corresponds to an attribution map which is achieved by an attribution method regarding the target class C_j. Each column i, denoted as $[A_{1i}, A_{2i}, \ldots, A_{Ki}]^T$, corresponds to the i^{th} input feature's attributions to K classes.

TABLE 1. **The form of matrix A.**

	X_1	$\cdots$	X_i	$\cdots$	X_N	sum
C_1	A_{11}		A_{1i}		A_{1N}	$A_{1.}$
$\vdots$						
C_j	A_{j1}		A_{ji}		A_{jN}	$A_{j.}$
$\vdots$						
C_K	A_{K1}		A_{Ki}		A_{KN}	$A_{K.}$
sum	$A_{.1}$		$A_{.i}$		$A_{.N}$	A

Moreover, let $A_{j.} = \sum_{i=1}^{N} A_{ji}$, $A_{.i} = \sum_{j=1}^{K} A_{ji}$ and $A = \sum_j \sum_i A_{ji}$.

Definition 1: Feature attribution. A_{ji} is defined as the attribution of feature X_i on class C_j.

Feature attribution can be calculated by LRP, Inp.* Grad., DL and other attribution methods described in Section II. If the attribution map satisfies the property of *Completeness*, then for each row j in $\mathbf{A}$, $\sum_{i=1}^{N} A_{ji} = \text{pred}(C_j)$, where $\text{pred}(C_j)$ refers to either pre-softmax or after-softmax value of the j^{th} class.

Definition 2: Variation of class predictions.The variation of classes $(C_1, C_2, \ldots, C_K)$ is defined as

$$V[C] = \sum_{j=1}^{K} \left(\text{pred}(C_j) - \frac{1}{K} \sum_{i=1}^{K} \text{pred}(C_i) \right)^2$$

$V[C]$ suggests the intensity of change of K class scores. The bigger the value of $V[C]$, the greater the degree of distinction between classes is.

Definition 3: Individual effect of feature. The individual effect of feature X_i's attribution on classes $(C_1, C_2, \ldots, C_K)$ is defined as

$$V[X_i] = \sum_{j=1}^{K} (A_{ji} - \frac{1}{K} A_{.i})^2$$

$V[X_i]$ implies the variation of attributions of X_i to K class scores, and the magnitude of $V[X_i]$ illustrates the capacity of X_i to distinguish classes. Additionally, the standard deviation of feature X_i's attribution on classes $(C_1, C_2, \ldots, C_K)$ is defined as

$$STD[X_i] = \sqrt{V[X_i]} \tag{1}$$

Definition 4: Co-variation effect of two features.The co-variation of features X_i and X_s's attributions is defined as

$$COV[X_i, X_s] = \sum_{j=1}^{K} \left(A_{ji} - \frac{1}{K} A_{.i} \right) \left(A_{js} - \frac{1}{K} A_{.s} \right) \tag{2}$$

The larger the value of $COV[X_i, X_s]$, the stronger the co-variation between the two features in the prediction process. When $i = s$, $COV[X_i, X_s] = V[X_i]$.

Definition 5: Feature importance. The feature importance of feature X_i is defined as

$$\phi_i = V[X_i] + \sum_{s=1, s\neq i}^{N} COV[X_i, X_s] \tag{3}$$

The first term on the right hand side of (3) denotes the individual effects of feature X_i; the second term denotes the co-variation effects of feature X_i with the other features. If features are independent of each other, then the feature importance of feature X_i is simplified as $\phi_i = V[X_i]$. In fact, ϕ_i represents the ability of X_i to distinguish between classes.

Proposition 1: The feature importance defined in (3) satisfies the properties of Dummy, Implementation Invariance, Symmetry-Preserving if the attribution method satisfies these properties.

The proofs of the propositions throughout the paper are shown in Appendix A.

Proposition 2: The sum of feature importance equals the variation of classes, i.e. $V[C] = \sum_{i=1}^{N} \phi_i$ if the feature attribution map satisfies the property of Completeness.

Proposition 2 illustrates that the feature importance add up to the variation of class scores, other than adding up to a single class score like the feature attributions.

Proposition 3: For a linear combination $f + g$ of two models f and g, $\phi_i(f + g) = \phi_i(f) + \phi_i(g) + \phi_i(f * g)$ where

$$\phi_i(f * g) = \sum_{s=1}^{N} \sum_{j=1}^{K} \left(A_{ji} - \frac{1}{K} A_{.i} \right) \left(B_{js} - \frac{1}{K} B_{.s} \right) + \sum_{s=1}^{N} \sum_{j=1}^{K} \left(B_{ji} - \frac{1}{K} B_{.i} \right) \left(A_{js} - \frac{1}{K} A_{.s} \right)$$

where A and B denote the attribution matrices of models f and g, respectively.

For a linear combination of models, the feature importance takes into account the co-variation between model f and g, $\phi(f * g)$. However, the attribution methods that satisfies the property of *Linearity*, such as LRP and Inp.*Gra., assume that there are no co-variation effects between two models.

B. SILHOUETTE

Based on the obtained STD map of input features defined in (1), this article presents a novel visualization method of important features, which can sketch the silhouette of the target object related to prediction in the image.

For each element of the STD map, the ratios of the element to its neighboring elements in its neighborhood of size r are calculated. If the maximum value of all ratios is greater than a specified threshold θ, then the target element is retained in the Silhouette map; otherwise, it is set to 0.

The process of the Silhouette map generation is equivalent to filtering features with far less important in the STD map and only preserving features that are essential for the class separation. Algorithm 1 expounds the process of generating a Silhouette map.

```
Algorithm 1 Generating Silhouette Map
Input: the STD_map, $S$, of size $H \times W \times C$; the neighbor
    size $r$; threshold $\theta$
Output: the Silhouette map $M$
 1: for each element $S_{ijk}$ in $S$ do
 2:     $M_{ijk} = 0$
 3:     $N(S_{ijk}, r) = \{X_{lmk} \in S : |l - i| \leq r$ and |m-j| $\leq r\}$
 4:     for each neighbor s in $N(S_{ijk}, r)$ do
 5:         if $S_{ijk}/s \geq \theta$ then
 6:             $M_{ijk} = S_{ijk}$
 7:             exit for
 8:         end if
 9:     end for
10: end for
    return $M$
```

C. PRACTICAL ESTIMATION

To enhance computational efficiency, three practical tricks are employed: first, top-k ($k < K$) class scores, instead of all K classes, are used to calculate feature importance; second, the attribution maps of multiple-class scores can be obtained by Jacobian matrix in the framework of *TensorFlow*; third, the co-variation between input features can be efficiently calculated based on matrix operations.

1) TOP-k SELECTION

In Section 3.1, all class scores (K classes) of a neural network model are used to obtain $\mathbf{A}$ that is further used to calculate feature importance in (3). Although for an input feature, calculating the variation of its attributions for all classes can better evaluate its class discriminability, the calculation cost will be prohibitive if the model has a great many of output neurons, such as 1,000 classes in VGG16. Therefore, in order to ensure high computational efficiency, the top-k ($k < K$) classes are selected for computing feature importance in this study.

To prove that the utilization of top-k classes will not have a significant impact on the feature importance results and visualization effects, this article employs VGG16 and ImageNet database to perform two sets of verification experiments. The premise of these experiments is to randomly select 500 images from the ImageNet database and apply Inp.* Grad. to calculate $\mathbf{A}$, then for each input image:

1) The first experiment sets three threshold values: $min_p_1 = $ 1e-5, $min_p_2 = $ 1e-6, $min_p_3 = $ 1e-7; then selects k_i classes with prediction probabilities larger than $min\_p_i, i = (1, 2, 3)$ to calculate the feature importance map, respectively. The Pearson correlation coefficient between the importance map of k_i classes and the map obtained by 1,000 classes is calculated; furthermore, this experiment draws the frequency distribution histogram of k_i classes (first row in Fig 3 (a)) and the scatter diagram that describes the relationship between

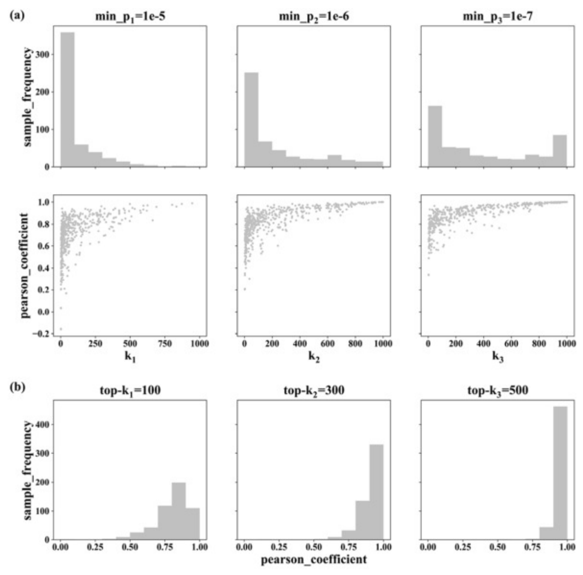


FIGURE 3. Correlation between feature importances of top-k classes and all K classes.

the correlation coefficient and the number of selected classes k_i (second row in Fig 3 (a)).

2) The second experiment sets up three threshold values: top-$k_1 = 100$, top-$k_2 = 300$, top-$k_3 = 500$; next, calculates directly feature importance map by top-k_i, $i = (1, 2, 3)$ class scores. Meanwhile, this experiment computes the Pearson correlation coefficient between the importance map of top-$k_i, i = (1, 2, 3)$ classes and the map acquired by 1,000 classes; on this basis, the frequency distribution histogram of the correlation coefficient is plotted (Fig 3 (b)).

Fig 3 (a) suggests that one probability threshold, *min_p*, can result in a large range of k; and for some images, the feature importance maps with small k can be very close to the results obtained by all classes scores, i.e. the correlation coefficients are close to 1. However, even if the probability threshold is set to 1e-7, for some images, k are not large enough such that the correlation coefficients are small. Therefore, determining top-k classes by setting the probability thresholds has certain limitations. As can be seen from Fig 3 (b), as the number of selected classes increases, the correlation coefficients of more samples fall into the range of [0.8, 1], which means that fixing the value of k can control the results easily.

Furthermore, Fig 4 compares the visualization effects realized by top-k class scores with those obtained by all class scores based on Inp.* Grad., which demonstrate that the utilization of top-k classes almost does not affect the effects of visualization while improving the computational efficiency.

2) GRADIENTS CALCULATION

Shrikumar *et al.* [24] first demonstrated that LRP was equivalent to Inp.* Grad. under certain conditions; then the viewpoint presented in [12] suggested that Inp.* Grad. took

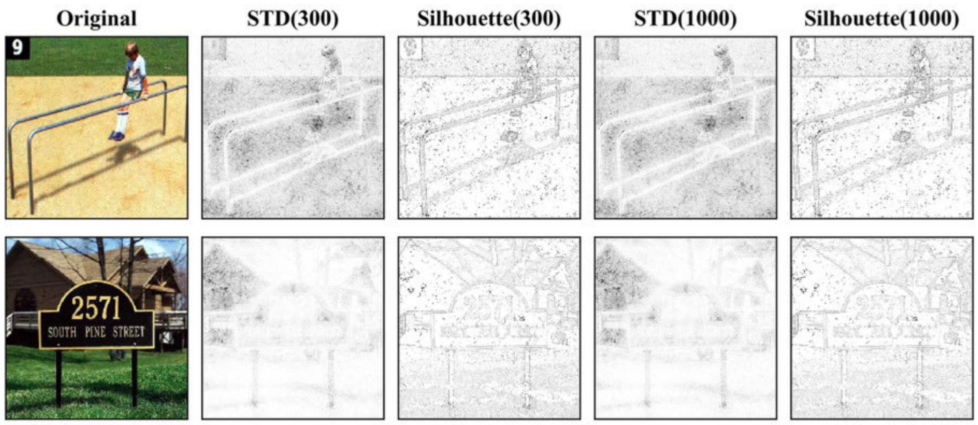


FIGURE 4. Comparison of the visualization effects produced by using different values of top-*k*. The second and third columns are the results calculated by top-300 classes, and the last two columns are the heatmaps generated by all classes. Furthermore, the results are realized by VGG16 and ImageNet dataset.

advantage of information about the signs and values of input features and was more desirable than the methods of merely calculating the gradients. Afterwards Ancona *et al.* [25] argued that Inp.* Grad. belonged to global attribution methods, which made the contributions of input features more reasonable and improved the visualization effects of attribution maps.

With that in mind, this study adopts Inp.* Grad. attribution method and obtains the gradients of multiple-class scores on input features in parallel through Jacobian matrix that is suitable for first-order differential of vectors, which can greatly improve the computational efficiency. If class scores vector is denoted as $\mathbf{S_c} = [S_1, S_2, \ldots, S_K]^T$ and input features vector is $\mathbf{X} = [X_1, X_2, \ldots, X_N]^T$, then the partial derivative of $\mathbf{S_c}$ with respect to $\mathbf{X}$ can be expressed as matrix $\mathbf{J}$:

$$\mathbf{J} = \frac{\partial \mathbf{S_C}}{\partial \mathbf{X}} = \begin{bmatrix} \frac{\partial S_1}{\partial \mathbf{X}} \\ \frac{\partial S_2}{\partial \mathbf{X}} \\ \vdots \\ \frac{\partial S_K}{\partial \mathbf{X}} \end{bmatrix} = \begin{bmatrix} \frac{\partial S_1}{\partial X_1} & \frac{\partial S_1}{\partial X_2} & \cdots & \frac{\partial S_1}{\partial X_N} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial S_K}{\partial X_1} & \frac{\partial S_K}{\partial X_2} & \cdots & \frac{\partial S_K}{\partial X_N} \end{bmatrix}$$

and matrix $\mathbf{J}$ can be easily calculated by *tf.jacobian* in the framework of *Tensorflow*. The matrix $\mathbf{A}$ defined in Section 3.1 is calculated by $\mathbf{X} \otimes \mathbf{J}$ where $\otimes$ denotes element-wise product.

3) CO-VARIATION CALCULATION

To calculate all features' importance, the number of co-variations needed to be calculated is $N(N-1)/2$, whose computation cost is very high when N is large. For example, for an image with size of $224 \times 224 \times 3$, the number of co-variations is more than 11 billion. Therefore, it is necessary to find an efficient way to calculate co-variations.

Proposition 4: Let $A_c = [A_{.1}, A_{.2}, \ldots, A_{.N}]$, $A_r = [A_{1.}, A_{2.}, \ldots, A_{K.}]^T$*. The co-variation term in feature importance can be calculated by*

$$\mathbf{COV} = \text{sum}_c \left((\mathbf{A} - \frac{1}{K} A_c) \otimes [(A_r - \frac{1}{K} A) - (\mathbf{A} - \frac{1}{K} A_c)] \right) \quad (4)$$

where $\otimes$ denotes element-wise product; sum$_c(\mathbf{X})$ denotes the column sum of matrix $\mathbf{X}$. In this equation, a matrix minus a column (row) vector means that each column (row) in the matrix minus the column (row) vector; a matrix minus a scalar value means each element in the matrix minus the scalar value.

Since the calculation in (4) adopts the matrix operation, the co-variations of all input features can be calculated in parallel, which greatly improves the calculation efficiency.

IV. EXPERIMENTS

In this Section, four comparative experiments and results are elaborated: (1) *STD and Silhouette* shows the improvement of the proposed method over Inp.* Grad. through the visualization of STD maps and Silhouette maps; (2) *Occlusion* objectively measures the correctness of the proposed method by blocking multiple input features simultaneously in descending order of the importance scores and observing the changes of the predictions; (3) *Model Comparison* further explores the characteristics of three CNN models through comparing the diversity of input features concerned by different models in predictions; (4) *Method Comparison* compares the implementation effects by using different attribution methods to calculate feature importance scores.

A. EXPERIMENT SETUP

In the experiments, this study adopts three popular pre-trained CNN models: VGG16 [26], ResNet50 [2], and Inception

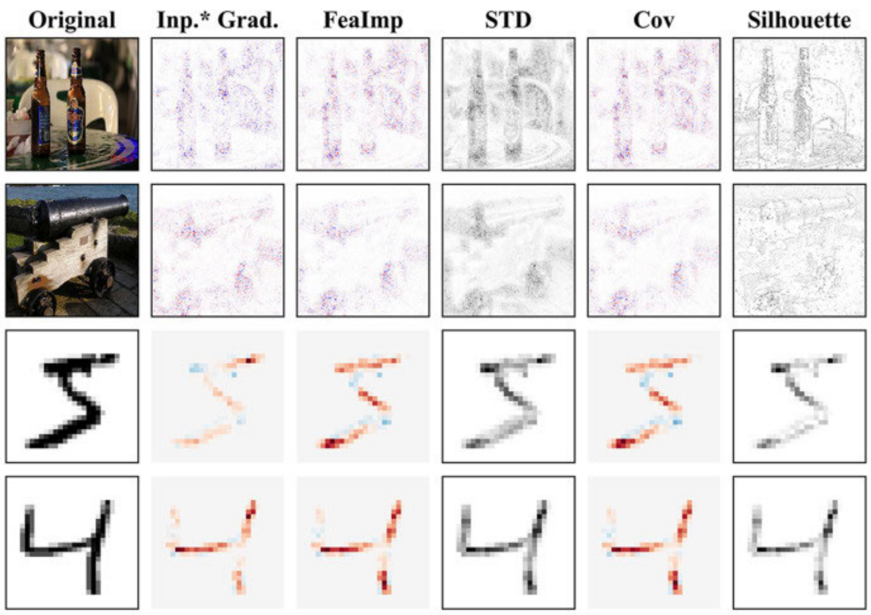


FIGURE 5. Visualization of the effects of Inp.* Grad. and the proposed method. The third column refers to the results of the feature importance scores calculated in (3).

TABLE 2. The structure of MNIST_CNN.

Conv2D (5×5, 20 kernels)
Max-pooling (2×2)
Conv2D (5×5, 50 kernels)
Max-pooling (2×2)
Dense (500)
Dense (10, activation functions: softmax)
The activation functions (except output layer): ReLUs
Total params: 1,256,080

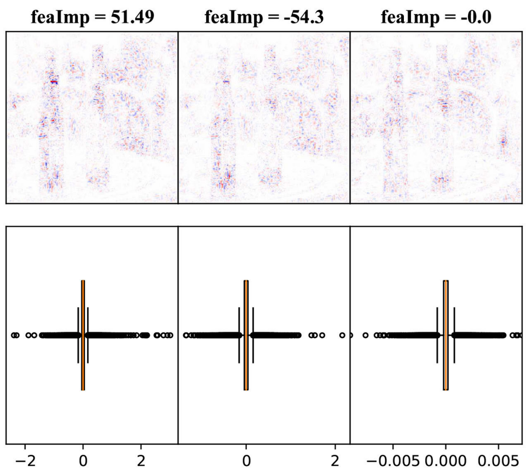


FIGURE 6. Visualization of the co-variation effects for three chosen pixels from top-left image in Fig 5. The columns from left to right correspond to the pixels with maximum, minimum and almost zero feature importance, respectively. The first and second rows are the visualization and distribution (box plots) of co-variation effects, respectively.

V3 [27]; Furthermore, this article trains a simple CNN model for the MNIST dataset [28], which is called MNIST_CNN. MNIST_CNN refers to the structure of LeNet and parameters setting proposed in [29], and achieves 98.89% testing accuracy. The structure and parameters of MNIST_CNN are detailed in TABLE 2. To measure the quality of this proposed method, ImageNet dataset [30] and MNIST dataset are chosen; and for these two data sets, the four experiments respectively select top-300 and all ten class scores to obtain matirx $\mathbf{A}$, respectively. Furthermore, the first three experiments apply Inp.* Grad. to calculate feature importance maps and the last experiment compares different attribution methods.

B. RESULTS

1) STD AND SILHOUETTE

This experiment randomly samples some images, which are correctly predicted by VGG16 and MNIST_CNN, from ImageNet dataset and MNIST dataset, respectively. To generate Silhouette maps, the neighbor size r is set up to 3 and the default value of threshold θ is 3 in Algorithm 1. In addition, for Inp.* Grad. attribution maps, only the maximum class score is propagated backward.

As shown in Fig 5, the results of this experiment suggests that the attribution maps based on Inp.* Grad., the feature importance maps (FeaImp) and the co-variation maps (Cov) have high similarities. Furthermore, the STD maps can highlight the important input features more clearly and the Silhouette maps can sketch the silhouettes of the objects. Thus, owing to the competitive performance of visualization, the following experimental results are mainly based on STD maps. In particular, a large number of tests have proven that

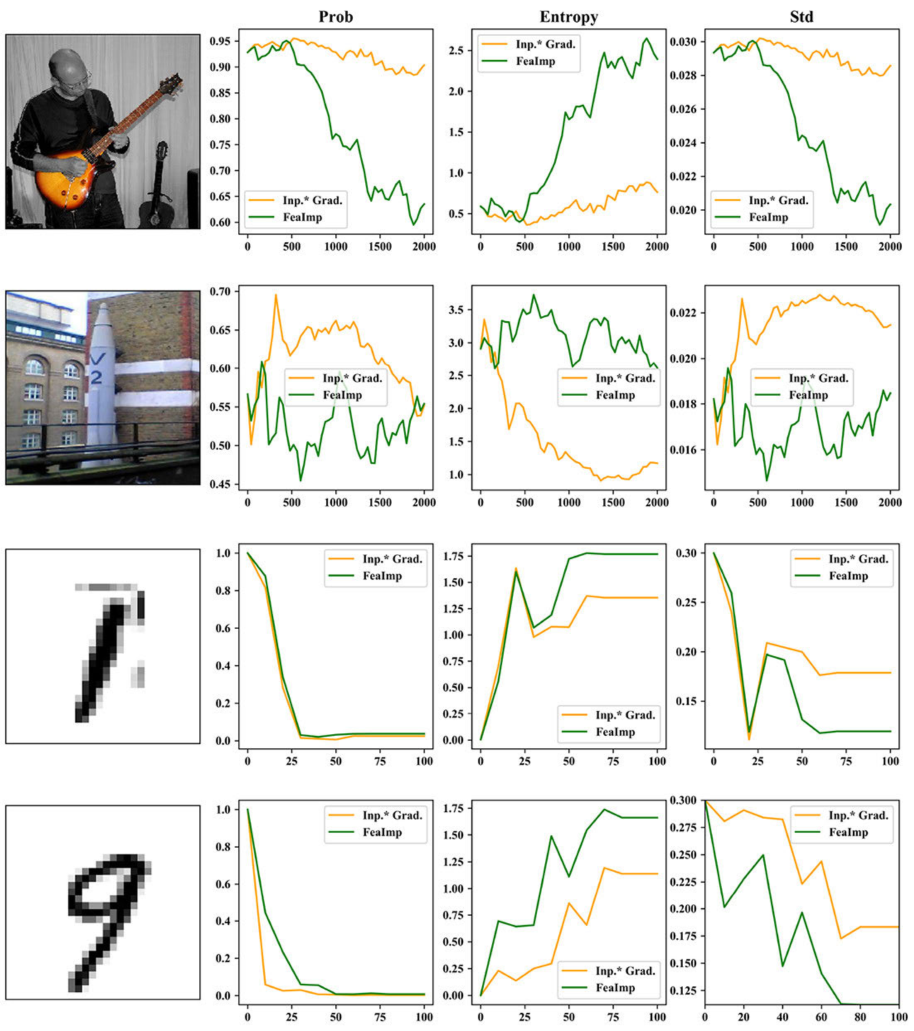


FIGURE 7. The occlusion results of Inp.* Grad. and the proposed method. The metrics, including prediction of the label, the entropy and standard deviation of interested classes, are used to evaluate methods. The vertical axis of the line chart indicates the value of the metrics, and the horizontal axis represents the number of occluded pixels. The pictures are sampled from ImageNet dataset and MNIST dataset.

STD maps and Silhouette maps have a better effect on images with more complex textures (see Appendix A).

To illustrate the co-variation effects, we choose three pixels with maximum, minimum and almost zero feature importance, respectively, from the top-left image in Fig 5. We calculate the co-variation effects in (2) of the chosen pixels with the other pixels and show them in the first row of Fig 6. And the second row of Fig 6 shows the distributions of co-variations. We can find that all of the co-variation effects for the three pixels highlight the main objects – bottles in the image. However, the distribution of co-variation effects are different greatly.

2) OCCLUSION

Aiming to objectively evaluate the proposed method, this article adopts the region perturbation method [31]. In the first place, this experiment selects some images which are classified accurately by Inception V3 and MNIST_CNN from ImageNet dataset and MNIST dataset. Second, the feature importance maps and the Inp.* Grad. score maps generated from Inception V3 and MNIST_CNN are sorted in descending order concerning the score values. Next, each iteration will further ablate a certain number of pixels according to the order, while keeping the occluded pixels of last iterative step. Finally, for each iteration, the experiment observes the values of three metrics, including the prediction of the target class, entropy and standard deviation of selected multiple-class scores. The entropy of top-k class scores has the mathematical form:

$$E[\mathbf{p}] = -\sum_{c=1}^{k} p_c \log(p_c)$$

where $\mathbf{p}$ represents the vector of k class scores and p_c denotes the c^{th} element in $\mathbf{p}$.

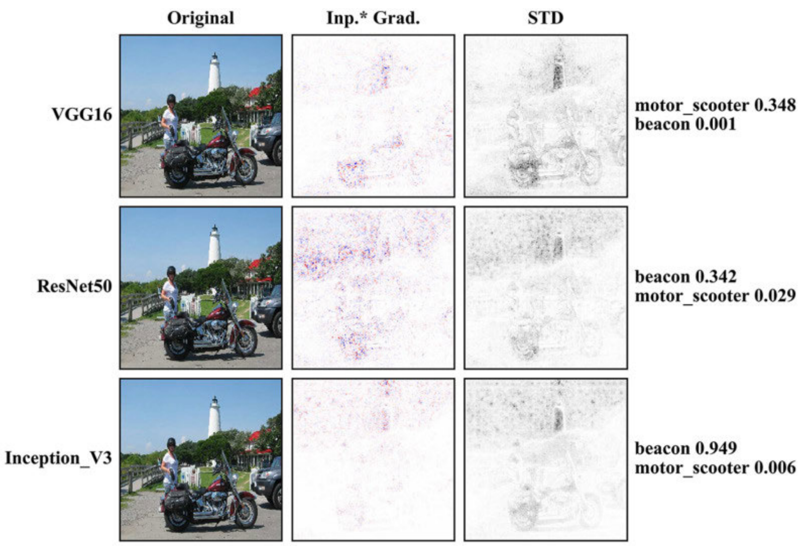


FIGURE 8. **Visualization of the important features extracted from three CNN models. The right side of the figure is marked with the predicted classes and probabilities of the corresponding model for the target objects in the sample.**

The three metrics stand for different meanings: the change of the target class prediction depicts the influence of occluded pixels on the prediction; the entropy refers to the degree of information confusion, and standard deviation describes the variation range of the data. Because the proposed method focus on variation of multiple-class predictions, this experiment pays attention to the entropy and the standard deviation. Intuitively, when the pixels with higher class discriminability are occluded, the divergence between the decisions of multiple classes tends to be smaller, so the standard deviation of these class scores will decrease; in accordance with it, the entropy of these scores will increase.

Fig 7 displays the results of this experiment. Each occlusion processes the pixels that have positive importance (relevance) scores. As the number of occluded pixels increases, in general, the entropy of multiple-class scores will grow and the standard deviation of these predictions will decline; moreover, the values of the entropy and standard deviation calculated by the feature importance scores change more dramatically than those from Inp.* Grad.. This phenomenon reveals that the pixels with strong class separation capacity are ablated when the image is continuously perturbed, and compared with Inp.* Grad., the proposed method embodies the class discriminability of input features more prominently.

3) MODEL COMPARISON

We may want to know if different networks extract distinctive crucial features in the classification tasks. As network structures become more complicated, the features that are extracted by the last convolutional layer will contain more spatial and semantic information. In an attempt to understand which features are used to classify in different CNN networks, this experiment compares the heat maps calculated by the proposed method and Inp.* Grad., separately, with three network structures. The prerequisites of this experiment is to choose some pictures which have different decisions relying on the network structures from ImageNet dataset.

As illustrated in Fig 8, VGG16 predicts that the top-1 class is the motor scooter (prob = 0.348); ResNet50 and Inception V3 forecast that the top-1 class is the beacon with the probability of 0.342 and 0.949, respectively. It is consistent that the STD map of VGG16 highlights the motor scooter in the picture, while the other models only emphasize the beacon in the image.

4) METHOD COMPARISON

In the previous experiments, the feature attribution maps, which are used to calculate the proposed STD and Silhouette maps, are obtained based on Inp.* Grad.. In this experiment, the results based on some other attribution methods, including ε-LRP, SM, Inp.* Grad., DL, Integrated Gradients, are compared. In this experiment, some samples correctly predicted by VGG16 and MNIST_CNN are randomly drawn from ImageNet dataset and MNIST dataset, respectively.

The visualization results (Fig 9) demonstrate that the STD and Silhouette maps on different attribution methods are roughly similar except the maps generated from SM, which confirms the conclusion drawn by Ancona *et al.* [25] that ε-LRP, SM, DL, Integrated Gradients could be unified in Inp.* Grad. framework. Moreover, the performance of SM is undesirable, probably because just calculating the gradients will produce more noise.

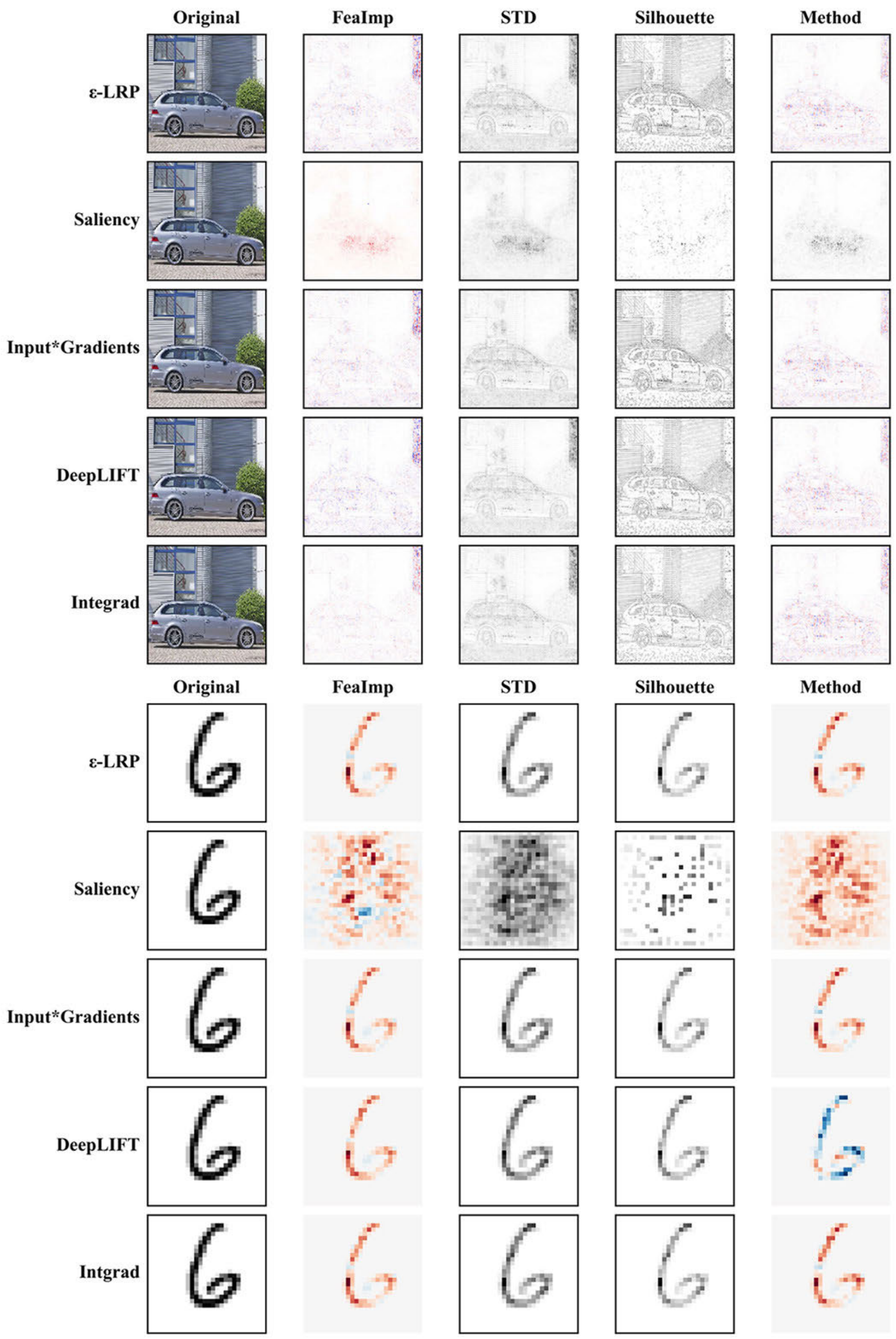


FIGURE 9. **Implementation effects of different attribute methods. The last column are heat maps generated directly from the attribution methods.**

V. CONCLUSION

For complex nonlinear models (e.g. CNNs), it is of great practical significance to understand their internal operation mechanisms and make their decisions more reliable. Preferable performance combined with comprehensible interpretation processes has a profound and positive impact on the future applications in many fields.

The method proposed in this article can improve the interpretability and visualization effects of existing attribution methods for classification tasks of CNNs. Specifically, STD maps achieved by backpropagating multiple predictions can concentrate on the class discriminability of input pixels. Silhouette maps generated from the STD maps can approximately sketch the silhouettes of the target objects. At the same time, three practical tricks including selection of top-k classes, calculation of Jacobian matrix and matrix operation of co-variations can ensure the computational efficiency.

The proposed method can also be applied to some other complex models, not limiting to CNNs, if we can obtain the attribution matrix. Fortunately, some model-agnostic

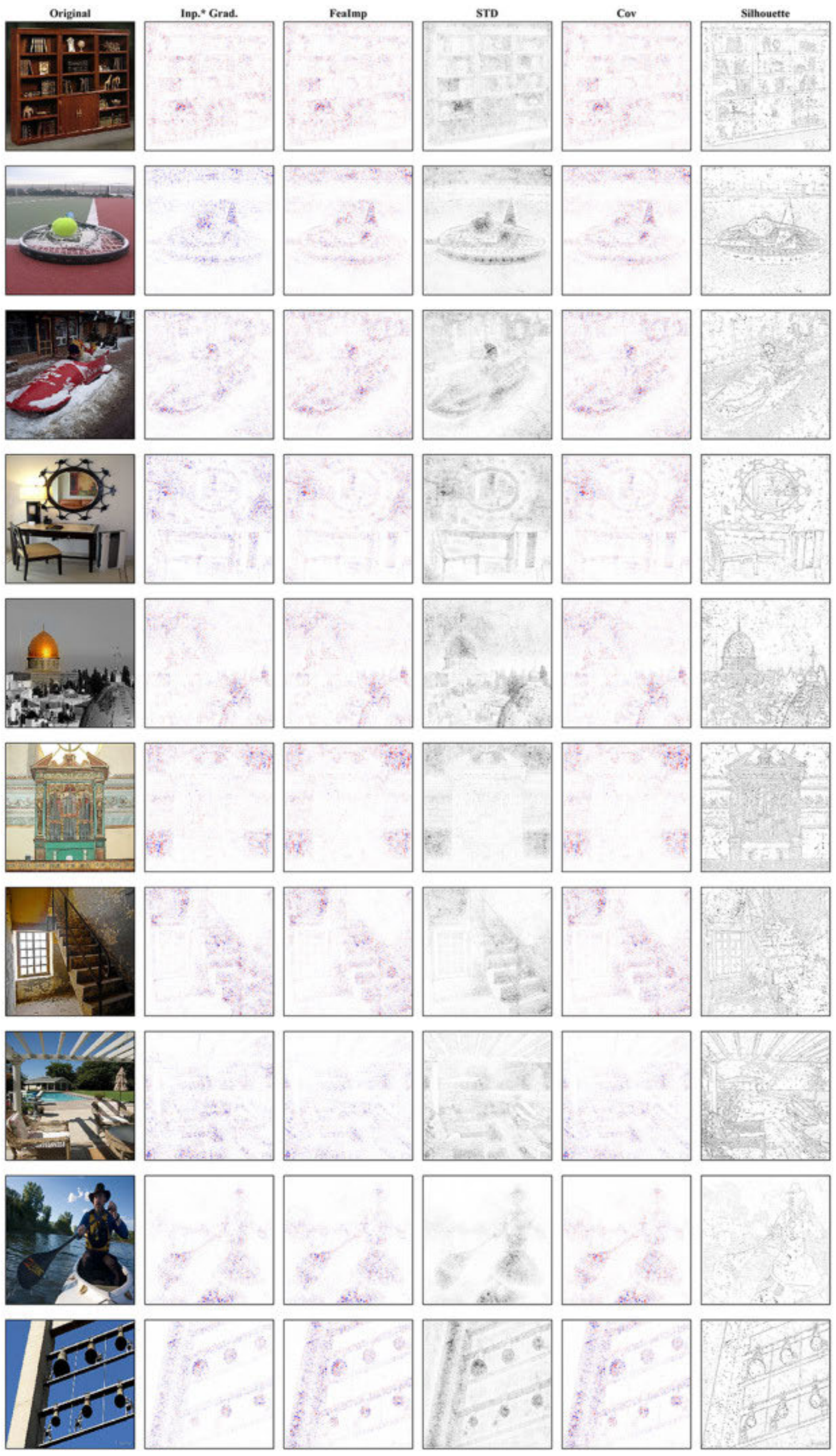


FIGURE 10. Heatmaps produced by the proposed method and Inp.* Grad., which are randomly sampled from ImageNet dataset.

methods, such as LIME and SHAP, can help obtain the attribution matrix for almost any model in classification tasks.

The future work is to study the application of the proposed method on some other network structures, such as recurrent neural networks, or more diversiform data sets, such as natural language data.

APPENDIX A
PROOF OF PROPOSITIONS

A. PROOF OF PROPOSITION 1

If the attribution method satisfies the property of *Dummy*, the attribution to the feature not contained in model is zero for any class. Thus, the variation and co-variation effects of the feature are zero; and further its feature importance is zero. If the attribution method satisfies the property of *Implementation Invariance*, then the attribution matrix (Table 1) keeps invariant for different functionally models with the same implementation; thus, the feature importance is implementation invariant. If a model is *symmetry-Preserving* for two feature X_i and X_j, then the columns corresponding to X_i and X_j in attribution matrix (Table 1) are identical; thus, the feature importance of X_i and X_j are identical.

B. PROOF OF PROPOSITION 2

If the feature attribution map satisfies the property of *Completeness*, then $A_{j.} = \text{pred}(C_j)$. Eq. (3) can be transformed into another representation:

$$\phi_i = V[X_i] + \sum_{s=1, s\neq i}^{N} COV[X_i, X_s] = \sum_{s=1}^{N} COV[X_i, X_s]$$

So,

$$\begin{aligned}\sum_{i=1}^{N}\phi_i &= \sum_{i=1}^{N}\sum_{s=1}^{N}\sum_{j=1}^{K}\left[\left(A_{ji} - A_{.i}\right)\left(A_{js} - \frac{1}{K}A_{.s}\right)\right]\\ &= \sum_{j=1}^{K}\left[\sum_{i=1}^{N}\left(A_{ji} - \frac{1}{K}A_{.i}\right)\sum_{s=1}^{N}\left(A_{js} - \frac{1}{K}A_{.s}\right)\right]\\ &= \sum_{j=1}^{K}\left[\left(A_{j.} - \frac{1}{K}A\right)\left(A_{j.} - \frac{1}{K}A\right)\right]\\ &= \sum_{j=1}^{K}\left(\text{pred}(C_j) - \frac{1}{K}\sum_{i=1}^{K}\text{pred}(C_i)\right)^2\\ &= V[C]\end{aligned}$$

C. PROOF OF PROPOSITION 4

Let $\hat{\mathbf{A}} = \mathbf{A} - A_c$, $\hat{a} = A_r - A$, and $\hat{\mathbf{A}}_i$ denotes the i^{th} column of $\hat{\mathbf{A}}$. It is obvious that $\hat{a} = \sum_{i=1}^{N}\hat{\mathbf{A}}_i$, and the i^{th} element of **COV** in (4) can be written as:

$$\mathbf{COV}_i = \left(\hat{\mathbf{A}}_i\right)^T\left(\hat{a} - \hat{\mathbf{A}}_i\right), \quad i = 1, 2, \ldots, N$$

In (2),

$$\begin{aligned}COV[X_i, X_s] &= \sum_{j=1}^{K}(A_{ji} - A_{.i})(A_{js} - A_{.s})\\ &= (\hat{\mathbf{A}}_s)^T\hat{\mathbf{A}}_i\end{aligned}$$

So, the second term in (3) is

$$\begin{aligned}\sum_{s=1, s\neq i}^{N} COV[X_i, X_s] &= \sum_{s=1, s\neq i}^{N}(\hat{\mathbf{A}}_i)^T\hat{\mathbf{A}}_s\\ &= (\hat{\mathbf{A}}_i)^T\sum_{s=1, s\neq i}^{N}\hat{\mathbf{A}}_s\\ &= (\hat{\mathbf{A}}_i)^T(\hat{a} - \hat{\mathbf{A}}_i)\end{aligned}$$

APPENDIX B
THE VISUALIZATION EFFECTS OF IMAGES WITH COMPLEX TEXTURES

Fig. 10 shows some results of the proposed method, which further testify the visualization effects of images with complex textures.

REFERENCES

[1] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 4700–4708.

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

[3] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the gap to human-level performance in face verification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1701–1708.

[4] Y. Sun, X. Wang, and X. Tang, "Deep learning face representation from predicting 10,000 classes," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1891–1898.

[5] L.-C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "DeepLab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected CRFs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 4, pp. 834–848, Apr. 2018.

[6] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Intervent.* Berlin, Germany: Springer, 2015, pp. 234–241.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[8] T.-Y. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 2117–2125.

[9] M. Ribeiro, S. Singh, and C. Guestrin, "'Why should i trust you?': Explaining the predictions of any classifier," in *Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics, Demonstrations*, 2016, pp. 1135–1144.

[10] L. M Zintgraf, T. S Cohen, T. Adel, and M. Welling, "Visualizing deep neural network decisions: Prediction difference analysis," 2017, *arXiv:1702.04595*. [Online]. Available: http://arxiv.org/abs/1702.04595

[11] S. Bach, A. Binder, G. Montavon, F. Klauschen, K.-R. Müller, and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation," *PLoS ONE*, vol. 10, no. 7, Jul. 2015, Art. no. e0130140.

[12] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3145–3153.

[13] K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep inside convolutional networks: Visualising image classification models and saliency maps," 2013, *arXiv:1312.6034*. [Online]. Available: http://arxiv.org/abs/1312.6034

[14] D. Smilkov, N. Thorat, B. Kim, F. Viégas, and M. Wattenberg, "SmoothGrad: Removing noise by adding noise," 2017, *arXiv:1706.03825*. [Online]. Available: http://arxiv.org/abs/1706.03825

[15] M. Sundararajan, A. Taly, and Q. Yan, "Axiomatic attribution for deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, 2017, pp. 3319–3328.

[16] E. J. Friedman, "Paths and consistency in additive cost sharing," *Int. J. Games Theory*, vol. 32, no. 4, pp. 501–518, Aug. 2004.

[17] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4765–4774.

[18] E. Strumbelj and I. Kononenko, "An efficient explanation of individual classifications using game theory," *J. Mach. Learn. Res.*, vol. 11, pp. 1–18, Jan. 2010.

[19] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *Proc. Eur. Conf. Comput. Vis.* Berlin, Germany: Springer, 2014, pp. 818–833.

[20] G. Montavon, S. Lapuschkin, A. Binder, W. Samek, and K.-R. Müller, "Explaining nonlinear classification decisions with deep Taylor decomposition," *Pattern Recognit.*, vol. 65, pp. 211–222, May 2017.

[21] P.-J. Kindermans, K. T. Schütt, M. Alber, K.-R. Müller, D. Erhan, B. Kim, and S. Dähne, "Learning how to explain neural networks: PatternNet and PatternAttribution," 2017, *arXiv:1705.05598*. [Online]. Available: http://arxiv.org/abs/1705.05598

[22] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller, "Striving for simplicity: The all convolutional net," 2014, *arXiv:1412.6806*. [Online]. Available: http://arxiv.org/abs/1412.6806

[23] M. Sundararajan, A. Taly, and Q. Yan, "Gradients of counterfactuals," 2016, *arXiv:1611.02639*. [Online]. Available: http://arxiv.org/abs/1611.02639

[24] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences," 2016, *arXiv:1605.01713*. [Online]. Available: http://arxiv.org/abs/1605.01713

[25] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross, "Towards better understanding of gradient-based attribution methods for deep neural networks," 2017, *arXiv:1711.06104*. [Online]. Available: http://arxiv.org/abs/1711.06104

[26] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," 2014, *arXiv:1409.1556*. [Online]. Available: http://arxiv.org/abs/1409.1556

[27] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 2818–2826.

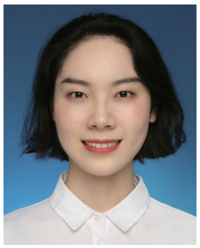
[28] C. J. B. Yann LeCun and C. Cortes. (1998). *The Mnist Database of Handwritten Digits*. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[29] Y. Lecun and Y. Bengio, "Convolutional networks for images, speech, and time series," in *The Handbook of Brain Theory and Neural Networks*. Cambridge, MA, USA: MIT Press, 1998, pp. 255–259.

[30] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[31] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Muller, "Evaluating the visualization of what a deep neural network has learned," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 28, no. 11, pp. 2660–2673, Nov. 2017.

XIAOHANG ZHANG (Member, IEEE) received the Ph.D. degree in management science and engineering from the Beijing University of Posts and Telecommunications, in 2003. From 2015 to 2017, he was a Research Fellow with the Department of Statistics, University of Michigan at Ann Arbor, Ann Arbor. He is currently a Professor of Information Systems with the Beijing University of Posts and Telecommunications. His recent research interests include machine learning, business intelligence, and credit portfolio risk modeling.



JIUYI GAO received the B.Admin. degree in management engineering and e-commerce from Zhejiang Gongshang University, in 2018. She is currently pursuing the master's degree in information management and information system with the Beijing University of Posts and Telecommunication. Her research interest includes the interpretation of neural networks.