Machine Learning (2023) 112:1627–1662
https://doi.org/10.1007/s10994-023-06312-1

Cross-model consensus of explanations and beyond for image classification models: an empirical study

Xuhong Li[1] · Haoyi Xiong[1] · Siyu Huang[2] · Shilei Ji[1] · Dejing Dou[1]

Received: 10 December 2021 / Revised: 2 June 2022 / Accepted: 29 January 2023 /
Published online: 23 February 2023

Abstract

Existing explanation algorithms have found that, even if deep models make the same correct predictions on the same image, they might rely on different sets of input features for classification. However, among these features, some common features might be used by the majority of models. In this paper, we are wondering *what the common features used by various models for classification are* and *whether the models with better performance may favor those common features*. For this purpose, our work uses an explanation algorithm to attribute the importance of features (e.g., pixels or superpixels) as explanations and proposes the *cross-model consensus of explanations* to capture the common features. Specifically, we first prepare a set of deep models as a *committee*, then deduce the explanation for every model, and obtain the *consensus* of explanations across the entire committee through *voting*. With the cross-model consensus of explanations, we conduct extensive experiments using 80+ models on five datasets/tasks. We find three interesting phenomena as follows: (1) the consensus obtained from image classification models is aligned with the ground truth of semantic segmentation; (2) we measure the similarity of the explanation result of each model in the committee to the consensus (namely *consensus score*), and find positive correlations between the consensus score and model performance; and (3) the consensus score potentially correlates to the interpretability.

Keywords Interpretability · Explanations of deep neural networks · Semantic segmentation · and Visualization

Editors: Krzysztof Dembczynski and Emilie Devijver.

✉ Haoyi Xiong
xionghaoyi@baidu.com

Extended author information available on the last page of the article

1 Introduction

Deep models are well-known for their excellent performance achieved in many challenging domains, as well as their black-box nature. To interpret the prediction of a deep model, a number of explanation algorithms (Bach et al., 2015; Lundberg & Lee, 2017; Ribeiro et al., 2016; Smilkov et al., 2017; Sundararajan et al., 2017; Zhou et al., 2016) have been recently proposed to attribute the importance of every input feature in a given sample with respect to the model's output. For example, given an image classification model, LIME (Ribeiro et al., 2016) and SmoothGrad (Smilkov et al., 2017) could attribute the importance score to every superpixel/pixel in an image with respect to the model's prediction. In this way, one can gain insights into models' behaviors by visualizing the important features used by the model for prediction.

We take image classification models as the research target. The use of interpretation tools finds that, even though these deep models make the same and correct predictions on the same image, they may rely on different sets of input features to solve the task. Our work uses LIME (or SmoothGrad similarly) to explain a number of image classification models trained on the same set of images, all of which make the correct predictions. The explanation algorithm obtains (slightly to moderately) different explanations for these models, with examples latterly shown in Figs. 2 and 3. While the features used by these models are not exactly the same, we can still find a set of features that the majority of models might use. We name them as common features. In this way, we are particularly interested in two research questions as follows: (1) *What are the common features used by various models in an image?* (2) *Whether the models with better performance favor those common features?*, to better understand the behaviors behind the black-box models.

To answer these two questions, we propose to study the common features across deep models and measure the similarity between the set of common features and the one used by individual models. Specifically, as illustrated in Fig. 1, we generalize an electoral system to first form a *committee* with a number of deep models, obtain the explanations for a given image based on one trustworthy explanation algorithm, then call for *voting* to get the *cross-model consensus of explanations*, or shortly *consensus*, and finally

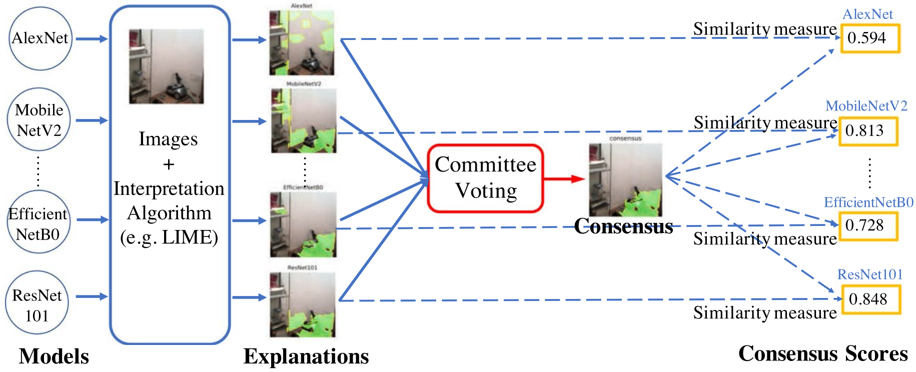


Fig. 1 Illustration of the proposed framework that consists of the three steps: (1) preparing a set of trained models as committee, (2) aggregating explanation results across the committee to get the consensus, and (3) computing the similarity score of each explanation to the consensus

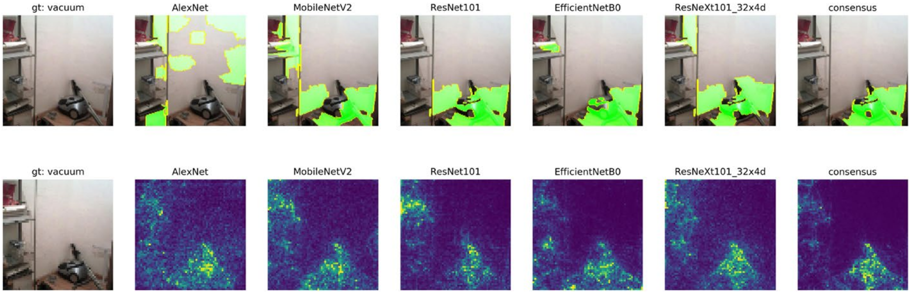


Fig. 2 Visual comparisons between consensus and the interpretation results of CNNs using LIME (in the upper line) and SmoothGrad (in the lower line) based on an image from ImageNet, where the ground truth of segmentation is not available

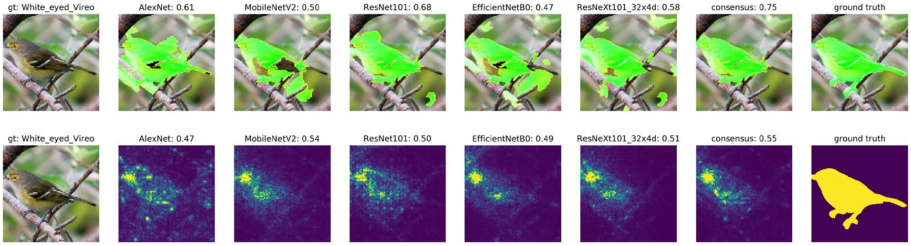


Fig. 3 Visual comparisons between consensus and the explanation results of deep models using LIME (in the upper line) and SmoothGrad (in the lower line) based on an image from CUB-200-2011, where the ground truth of segmentation is available as pixel-wise annotations and the mean Average Precision (mAP) are measured

compute a similarity score between the consensus and the explanation result for each deep model, denoted as *consensus score*. Through extensive experiments using 80+ models on five datasets/tasks, we find that (1) the consensus is aligned with the ground truth of image semantic segmentation; (2) a model in the committee with a higher consensus score usually performs better in terms of testing accuracy; and (3) models' consensus scores potentially correlate to their interpretability.

The contributions of this paper can be summarized as follows. To the best of our knowledge, this work is the first to investigate the common features used and shared by a large number of deep models for image classification through incorporating trustworthy explanation algorithms. We propose the cross-model consensus of explanations to characterize the common features and connect the consensus score to the model performance and interpretability. Finally, we obtain the three observations from the experiments with thorough analyses and discussions.

2 Related work

In this section, we first review the explanation algorithms and the approaches to evaluating their trustworthiness. Then we introduce some works related to our observations on the positive correlation between model performance and the proposed consensus score.

2.1 Explanation algorithms and evaluations

Many algorithms have been proposed to visualize the activated regions of feature maps in the intermediate layers (Chattopadhay et al., 2018; Selvaraju et al., 2020; Wang et al., 2020; Zhou et al., 2016), to gain insights for understanding the internals of convolutional networks. Apart from investigating the inside of complex deep networks, simple linear or tree-based surrogate models have been used as "out-of-box explainers" to explain the predictions made by the deep model over the dataset through local or global approximations (Ahern et al., 2019; Ribeiro et al., 2016; van der Linden et al., 2019; Zhang et al., 2019). Instead of using surrogates for deep models, investigations on the gradients for differentiable models have also been proposed to estimate the input feature importance with respect to the model predictions, such as SmoothGrad (Smilkov et al., 2017), Integrated Gradients (Sundararajan et al., 2017), DeepLIFT (Shrikumar et al., 2017) etc. Note that there are many other explanation algorithms (Afrabandpey et al., 2020; Atanasova et al., 2020; Bach et al., 2015; Kim et al., 2018; Looveren & Janis, 2020) and we mainly focus on those that are related to feature attributions and suitable for image classification models in this work.

There are few works of analyzing the explanations across models. For example, Fisher et al. (2019) theoretically studied the variable importance for machine learning models of the same family. Agarwal et al. (2021) aggregates rankings from several classifiers for time series classification tasks, instead of averaging explanation results. In this work, we investigate the cross-model explanations for image classification models and relate the consensus to the ability of localizing visual objects, model performance and interpretability.

Evaluations on the trustworthiness of explanation algorithms are of objective to qualify their fidelity to the models and avoid the misunderstanding of models' behaviors. For example, Adebayo et al. (2018) have found that some algorithms are independent both of the model and the data generating process, which should be avoided for not explaining the model, and thus proposed a sanity-check framework through perturbing parts parameters of models. Quantitative metrics for the trustworthiness evaluations include measuring the performance drop by perturbation of important features (Hooker et al., 2019; Petsiuk et al., 2018; Samek et al., 2016; Vu et al., 2019), model trojaning attacks (Chen et al., 2017; Gu et al., 2017; Lin et al., 2020), infidelity and sensitivity (Ancona et al., 2018; Yeh et al., 2019) to similarity samples in the neighborhood, through crafted datasets (Yang & Kim, 2019), and user-study experiments (Jeyakumar et al., 2020; Lage et al., 2019).

Towards building more interpretable and explainable AI systems, trustworthy explanation algorithms are the first step. Evaluations on the model interpretability, indicating which model is more interpretable, are also urged. However, such evaluations across deep models are scarce. Bau et al. (2017) proposed *Network Dissection* to build an additional dataset with dense annotations of a number of visual concepts for evaluating the interpretability of convolutional neural networks. Given a convolutional model, Network Dissection recovers the intermediate-layer feature maps used by the model for the classification. It then measures the overlap between the activated regions in the feature maps with the densely human-labeled visual concepts to estimate the interpretability of the model. Note that elaborately designed user-study experiments are also a common solution to evaluating deep model interpretability.

In this work, we do not directly evaluate the interpretability across deep models. Instead, we experimentally show that the consensus score is positively correlated to the generalization performance of deep models and related to the interpretability. We will discuss more details with analyses later. Based on the explanations, our proposed framework and the consensus score could help to understand the deep models better.

2.2 Explanation and semantic segmentation

Explanations are also useful to improve model performance (Kim et al., 2020), robustness (Ross et al., 2018) or interpretability (Chen et al., 2019). One related direction is weakly supervised semantic segmentation, which trains a deep model with image-wise annotations and makes efforts to predict pixel-wise segmentation results. The explanation results are beneficial to connect the bridge between two levels of labels (Jo et al., 2021; Wang et al., 2020). The proposed consensus, obtained across models from another aspect, confirms this connection as described in our first observation.

3 Framework of cross-model consensus of explanations

In this section, we first recall the two explanation algorithms that are used to validate our proposed framework, i.e., LIME (Ribeiro et al., 2016) and SmoothGrad (Smilkov et al., 2017). Then we introduce the proposed approach that generalizes the electoral system to provide the consensus of explanations across various deep models.

3.1 Recall LIME and SmoothGrad

Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al., 2016) searches an interpretable model, usually a linear one, to approximate the output of a deep model for an individual data point, such that LIME obtains a weighted linear combination of features including the importance of every feature for classifying the data point. To explain a deep model on vision tasks, LIME (Ribeiro et al., 2016) first performs a superpixel segmentation (Vedaldi et al., 2008) for a given image to reduce the number of input features (pixels aggregated into superpixels), then generates interpolated samples by randomly masking some superpixels and computing the prediction outputs of the generated samples through the original model, and finally uses a linear regression model to fit the outputs with the presence/absence of superpixels as inputs. The linear weights are then used to indicate the contributions of superpixels as the explanation results.

LIME is model-agnostic, without any requirement on the inside of models. Instead, another family of explanation algorithms is based on gradients, thus requires the models be differentiable. The gradients of model output w.r.t. input can partly identify influential pixels, but due to the saturation of activation functions in the deep networks, the vanilla gradient is usually noisy. SmoothGrad (Smilkov et al., 2017) reduces the visual noises by repeatedly adding small random noises on the image to get the gradients corresponding to the noised inputs, and then averages these gradients to smooth out noises for obtaining the final explanation result. In this work, based on LIME and SmoothGrad, we conduct experiments to validate our proposed approach. Note that other post-hoc explanation algorithms for interpreting individual samples are also available for our proposed framework.

3.2 Steps to computing cross-model consensus of explanations

Based on one of the explanation algorithms, our proposed framework computes the cross-model consensus of explanations and the consensus score, with the three specific steps as follows.

Algorithm 1 Cross-Model of Explanations Pseudocode

```
Require: $\mathcal{D}$, a dataset; $\mathcal{A}$, an explanation algorithm; $m$, the number of deep
    models that form the committee.
Ensure: Output the consensus score for each model in $\mathcal{M}$.
$\boldsymbol{S} \leftarrow$ An empty matrix for storing the similarity scores.
▷ Step 1: Committee Formation with Deep Models
$\mathcal{M} \leftarrow m$ models trained on $\mathcal{D}$
for each example $d_i$ in $\mathcal{D}$ do
    ▷ Step 2: Committee Voting for Consensus Achievement
    $\boldsymbol{L} \leftarrow$ An empty matrix for storing the explanations.
    for each model $\boldsymbol{M}_j$ in $\mathcal{M}$ do
        $\boldsymbol{L}_j \leftarrow \textsc{interpret}(\mathcal{A}, d_i, \boldsymbol{M}_j)$
    end for
    $\boldsymbol{c} \leftarrow \textsc{reach\_consensus}(\boldsymbol{L})$
    ▷ Step 3: Consensus-based Similarity Score
    for each model $\boldsymbol{M}_j$ in $\mathcal{M}$ do
        $\boldsymbol{S}_{ij} \leftarrow \textsc{similarity}(\boldsymbol{L}_j, \boldsymbol{c})$, as the score of $\boldsymbol{M}_j$ for $d_i$.
    end for
end for
Return $\frac{1}{|\mathcal{D}|} \sum_i \boldsymbol{S}_i$, the consensus score for each model in $\mathcal{M}$.
```

Algorithm 2 Definitions of Functions in Algorithm 1

```
procedure INTERPRET($\mathcal{A}$, $d$, $\boldsymbol{M}$)
    Require: $\mathcal{A}$, an explanation algorithm; $d$, a data sample $d$; $\boldsymbol{M}$, a trained model.
    ▷ Most algorithms are applicable for $\mathcal{A}$, e.g. LIME and SmoothGrad.
    Return $\boldsymbol{L}_d$, the explanation result of $\boldsymbol{M}$ on $d$ by $\mathcal{A}$.
end procedure

procedure REACH_CONSENSUS($\boldsymbol{L}$)
    Require: $\boldsymbol{L}$, a collection of interpretations of $m$ models for one given data sample.
    $\boldsymbol{c} \leftarrow$ An empty matrix for storing the consensus results.
    for each dimension $k$ in the explanation do
        $\boldsymbol{c}_k \leftarrow \frac{1}{m}\sum_{j=1}^{m} \frac{\boldsymbol{L}_{jk}^2}{\|\boldsymbol{L}_j\|}$ for LIME explanations
        $\boldsymbol{c}_k \leftarrow \frac{1}{m}\sum_{j=1}^{m} \frac{\boldsymbol{L}_{jk} - min(\boldsymbol{L}_j)}{max(\boldsymbol{L}_j) - min(\boldsymbol{L}_j)}$ for SmoothGrad explanations.
    end for
    Return $\boldsymbol{c}$, the cross-model consensus of explanations.
end procedure

procedure SIMILARITY($\boldsymbol{a}$, $\boldsymbol{b}$)
    Require: Two vectors $\boldsymbol{a}$ and $\boldsymbol{b}$.
    $s \leftarrow \frac{<\boldsymbol{a},\, \boldsymbol{b}>}{\|\boldsymbol{a}\|\|\boldsymbol{b}\|}$ for LIME explanations;
    $s \leftarrow e^{-\frac{(\|\boldsymbol{a}-\boldsymbol{b}\|/\sigma)^2}{2}}$ for SmoothGrad explanations.
    Return $s$, similarity score between $\boldsymbol{a}$ and $\boldsymbol{b}$.
end procedure
```

Step 1: Committee formation with deep models

Given m deep models that are trained for solving a target task (image classification task in our experiments) on a visual dataset where each image contains one primary object, the approach first forms the given deep models as a *committee*, noted as $\mathcal{M}$, and then considers the variety of models in the committee that would establish the consensus for comparisons and evaluations.

Step 2: Committee voting for consensus achievement

With a committee of deep models for the image classification task, our proposed framework leverages a trustworthy interpretation tool $\mathcal{A}$, e.g. LIME (Ribeiro et al., 2016) or SmoothGrad (Smilkov et al., 2017), to obtain the explanation on each image in the dataset. Given some sample, denoted as d_i, from the dataset, we note the obtained explanation results of all models in the committee as $\boldsymbol{L}$. Specifically, $\boldsymbol{L}_j$ indicates the explanation given by the j-th model. Then, we propose a *voting* procedure that aggregates $\{\boldsymbol{L}_j\}_{j=1,\ldots,m}$ to reach the cross-model consensus of explanations, i.e., the *consensus*, $\boldsymbol{c}$ for d_i. Specifically, the k-th element of the consensus $\boldsymbol{c}$ is $\boldsymbol{c}_k = \frac{1}{m}\sum_{j=1}^{m} \frac{\boldsymbol{L}_{jk}^2}{\|\boldsymbol{L}_j\|}$, $\forall 1 \le k \le K$ for LIME, where K refers to the dimension of an explanation result and $\boldsymbol{c}_k = \frac{1}{m}\sum_{j=1}^{m} \frac{\boldsymbol{L}_{jk} - min(\boldsymbol{L}_j)}{max(\boldsymbol{L}_j) - min(\boldsymbol{L}_j)}$, $\forall 1 \le k \le K$ for SmoothGrad, following the conventional normalization-averaging procedure (Ahern et al., 2019; Ribeiro et al., 2016; Smilkov et al., 2017). To the end, the consensus has been reached for every sample in the target dataset based on committee voting.

Step 3: Consensus-based similarity score

Given the reached consensus, our approach calculates the similarity score between the explanation result of every model in the committee and the consensus, as the *consensus score*. Specifically, for the explanations and the consensus based on LIME (visual feature importance in superpixel levels), cosine similarity between the flattened vector of explanation of each model and the consensus is used. For the results based on SmoothGrad (visual feature importance in pixel levels), a similar procedure is followed, where the proposed algorithm uses Radial Basis Function (RBF, $exp(-\frac{1}{2}(||\boldsymbol{a}-\boldsymbol{b}||/\sigma)^2)$) for the similarity measurement. The difference in similarity computations is due to that (1) the dimensions vary across data samples for LIME explanations while do not change for SmoothGrad, and (2) the scales of LIME explanation results vary much larger than SmoothGrad. Thus cosine similarity is more suitable for LIME while RBF is for SmoothGrad. Eventually, the framework computes a quantitative but relative score for each model in the committee using their similarity to the consensus.

In summary, the proposed method first selects a number of deep models as a committee. Then given a data sample from the dataset, the proposed method computes the explanation using one interpretation algorithm (e.g. LIME or SmoothGrad) for each deep model in the committee, and obtains the cross-model *consensus* of explanations through a voting process. Finally, given the reached consensus, our approach calculates the similarity score for each model, as the *consensus score*, between the explanation result of that model and the consensus. For each data sample, we can compute such a consensus score and the final one is averaged across all data samples from the dataset. For further clarity, these three steps of the proposed framework are illustrated in Fig. 1 and formalized in Algorithm 1, while the definitions of elementary functions (i.e., the interpretation algorithm, the voting procedure, and the similarity score) are omitted in Algorithm 1 but given in Algorithm 2.

4 Overall experiments and results

In this section, we start by introducing the experiment setups. We use the image classification as the target task and follow the proposed framework to obtain the consensus and compute the consensus scores. Through the experiments, we have found (1) the good alignment between the consensus and image semantic segmentation, (2) positive correlations between the consensus score and model performance, and (3) potential correlations between the consensus score and model interpretability. We end this section with robustness analyses of the framework.

4.1 Evaluation setups

4.1.1 Datasets

For overall evaluations and comparisons, we use ImageNet (Deng et al., 2009) for general visual object recognition and CUB-200-2011 (Welinder et al., 2010) for bird recognition respectively. Note that ImageNet provides the class label for every image, and the CUB-200-2011 dataset includes the class label and pixel-level segmentation for the bird in every image, where the pixel annotations of visual objects are found to align with the consensus.

4.1.2 Committee formation with deep models

Our main experiments and results include models of the two committees based on ImageNet and CUB-200-2011, respectively. Both of them target the *image classification* task, with each image being labeled to one category. For complete comparisons, we use more than 80 deep models trained on ImageNet that are publicly available.

There are over 100 deep models[1] at the moment we initiate the experiments. We first exclude some very large models that take much more computation resources. Then for the consistency of computing superpixels, we include only the models that take images of size 224×224 as input, resulting 81 models for the committee based on ImageNet. Note that other models can be also included by an additional step of aligning the superpixels in images of different sizes. However, in our experiments, we choose to ignore this small set of models since a large number of models are available.

As for CUB-200-2011 (Welinder et al., 2010), similarly we first exclude the very large models. Then we follow the standard procedures (Sermanet et al., 2014) for fine-tuning ImageNet-pretrained models on CUB-200-2011. We choose the default hyper-parameter setting to conduct the fine-tuning experiments on CUB-200-2011 for all models, i.e., learning rate 0.01, batch size 64, SGD optimizer with momentum 0.9, resize to 256 being the short edge, randomly cropping images to the size of 224×224, and obtain 85 models that are well trained. Different hyper-parameters may help to improve the performance of some specific networks, but for a fair comparison across model structures and economical reasons, we choose not to do the hyper-parameter tuning.

4.1.3 Explanation algorithms

As we previously introduced, we consider two explanation algorithms, LIME and SmoothGrad. Specifically, LIME surrogates the explanation as the assignment of visual feature importance to superpixels, and SmoothGrad outputs the explanations as the visual feature importance over pixels. In this way, we can validate the flexibility of the proposed framework over explanation results from diverse sources (i.e., linear surrogates vs. input gradients) and in multiple granularities (i.e., feature importance in superpixel/pixel-levels).

4.1.4 Computation costs

We here report the computation costs of preparing the committee and the explanation for reference, tested on one V100 GPU. Each fine-tuned model on CUB-200-2011 takes one hour more or less depending on the model size. LIME takes around 15 s in average (variant across models) per sample and SmoothGrad takes around 3 s per sample. For practical usages of cross-model consensus explanation, 15 models are suggested while a smaller scale of 5 may work as well.

[1] https://github.com/PaddlePaddle/models/blob/release/1.8/PaddleCV/image_classification/README_en.md#supported-models-and-performances.

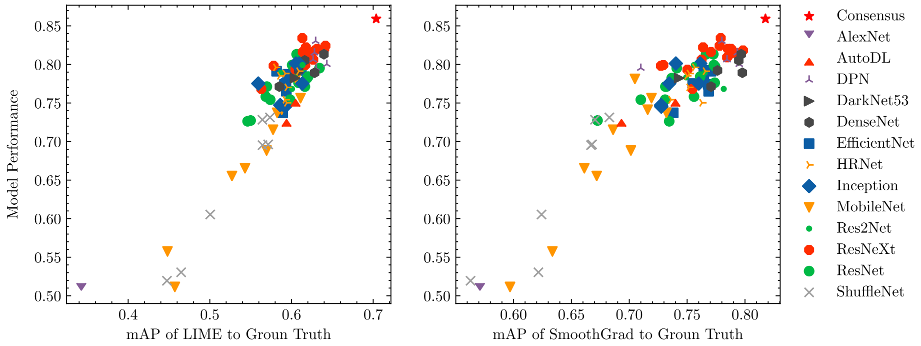


Fig. 4 Correlation between model performance and mAP scores to the segmentation ground truth using LIME (left) and SmoothGrad (right) on CUB-200-2011 over 85 models. Pearson correlation coefficients are 0.927 (with p value 4e−37) for LIME and 0.916 (p value 9e−35) for SmoothGrad. Take the "AlexNet" as example, this model gets 0.507 accuracy score on CUB-200-2011, and the alignment between its LIME explanations and the ground truth of semantic segmentation is measured by the mAP score of 0.343 (and 0.571 for SmoothGrad). These numeric results are reported in the Appendix. Moreover, the points "Consensus" here refer to the testing accuracy of the ensemble of networks in the committee by probabilities averaging and voting (in y-axis), as well as the mAP between the consensus and the ground truth (in x-axis). For the concise purpose, models in the same family are represented by the same symbol. Best viewed in color and with zoom-in (Color figure online)

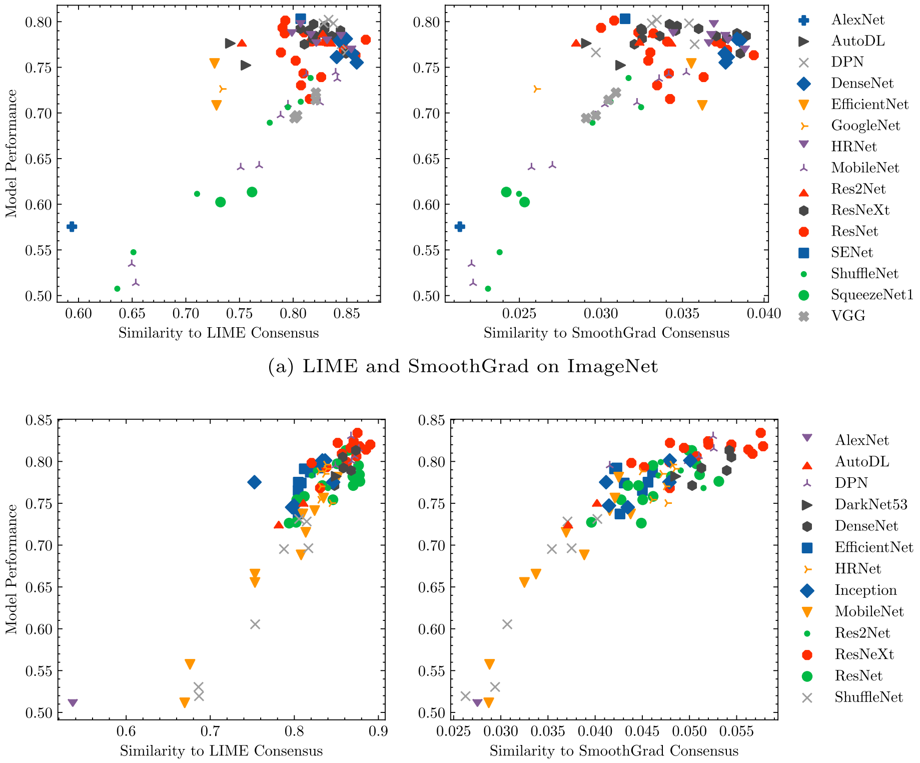
4.2 Alignment between the consensus and image segmentation

The image segmentation task searches the pixel-wise classifications of images. The cross-model consensus of explanations for image classification are well aligned to image segmentation, especially when only one object is contained in the image. This observation partially demonstrates the effectiveness of most deep models in extracting visual objects from input images. We show two examples using both LIME and SmoothGrad in Figs. 2 and 3 from ImageNet and CUB-200-2011 respectively. For both examples, we can find that the explanation algorithms reveal the models' predictions by highlighting some parts of the target objects, while the cross-model consensus shows a much better alignment with the objects than individual models. This observation can be found in more examples, as shown in the Appendix.

We confirm this alignment using the Average Precision (AP) score between the cross-model consensus of explanations and the image segmentation ground truth, where the latter is available on CUB-200-2011. We take the mean of AP scores (mAP) over the dataset to compare with the overall consensus scores. Higher mAP scores indicate better alignment between the explanations and the image segmentation ground truth. The quantitative results are shown in Fig. 4, where the consensus achieves higher mAP scores than any individual network. Both visual comparisons and quantitative results validate the closeness of consensus to the ground truth of image segmentation.

4.3 Positive correlations between consensus scores and model performance

Raw input features are not always useful. Some are discriminative while others are not. We use discriminative features to indicate those that can be used by models to well separate



(a) LIME and SmoothGrad on ImageNet

(b) LIME and SmoothGrad on CUB

Fig. 5 Model performance vs. consensus scores using LIME (**a**, left) and SmoothGrad (a, right) over 81 models on ImageNet (**b**, left) and 85 models on CUB-200-2011 (b, right). Pearson correlation coefficients are **a** 0.809 and 0.783, **b** 0.908 and 0.880. For concise purposes, networks in the same family are represented by the same symbol. Best viewed in color and with zoom-in (Color figure online)

samples from different categories. Usually they are the important features for solving the learning task. Based on this, we could reasonably assume that (1) for classification tasks of single-object images, the discriminative features are the pixels/superpixels of the target object in the image, and (2) if the key features used by the deep model (that can be revealed by trustworthy explanation algorithms) are aligned with the discriminative ones, the model is more likely to produce correct predictions and thus better performance. We presented previously that the cross-model consensus of explanations is aligned with object segmentation, implicitly indicating that the common features may be aligned with the discriminative ones. Here we show the positive correlations between the consensus score and model performance.

Specifically, in Fig. 5, we present the consensus scores (in x-axis) using LIME (left) and SmoothGrad (right) on ImageNet (a) and CUB-200-2011 (b), against model performance (in y-axis). High correlation coefficients are observed across the dataset-explanation combinations, though in some local areas of Fig. 5 (a, right), the correlation between the consensus score and model performance is weaker. In this way, we could conclude that, in an overall manner, the evaluation results based on the consensus score using both LIME and SmoothGrad over the two datasets are correlated to model performance with significance.

Table 1 Rankings (and scores) of five deep models, evaluated by Network Dissection Bau et al. (2017) (Net.Dis.), user-study evaluations (User-Study), and Consensus with LIME and SmoothGrad (C.LIME and C.SG respectively).

	DenseNet161	ResNet152	VGG16	GoogleNet	AlexNet
Net.Dis.	2	1	3	4	5
User-Study	1 (1.715)	2 (1.625)	3 (1.585)	4 (1.170)	5 (0.840)
C.LIME	1 (0.849)	2 (0.846)	3 (0.821)	4 (0.734)	5 (0.594)
C.SG.	1 (0.038)	2 (0.037)	3 (0.030)	4 (0.026)	5 (0.021)

This can further be supported by experiments on other datasets with random subsets of deep models, as shown in Fig. 11 (Appendix 2).

4.4 Potential correlations between consensus scores and model interpretability

Deep model interpretability measures *the ability to present in understandable terms to a human* (Doshi-Velez & Kim, 2017). Network Dissection (Bau et al., 2017) and user-study experiments are two possible methods to measure the interpretability of deep models quantitatively. Network Dissection (Bau et al., 2017) provided a dataset named Broden, which pre-defines a set of semantics, including colors, patterns, materials, textures, object parts etc, and provides the manually annotated pixel-wise labels in each image. Network Dissection benefits this dataset to count the number of semantic neurons in the intermediate layers of deep models as the interpretability. User-study evaluations measure the interpretability through designed experiments with humans' interactions and the collected statistics.

This subsection shows that the consensus scores are correlated with interpretability scores, measured by Network Dissection and user-study evaluations. Note that the consensus scores are computed based on explanation results, but they are not a direct estimator or metric of the model interpretability.

4.4.1 Consensus versus network dissection

We compare the results of the proposed framework with the ones from Network Dissection (Bau et al., 2017). Based on the Broden dataset, Network Dissection reported a ranking list of five models (w.r.t. the model interpretability), shown in Table 1, through counting the *semantic* neurons, where a neuron is defined as semantic if its activated feature maps overlap with human-annotated visual concepts. With our proposed framework, we report the consensus scores using LIME and SmoothGrad in Table 1, which are consistent to Fig. 5 (a, LIME) and (a, SmoothGrad). The three ranking lists are almost identical, except the comparisons between DenseNet161 and ResNet152, and in both lists based on the consensus scores, DenseNet161 is similar to ResNet152 with marginally elevated consensus scores, while Network Dissection considers ResNet152 is more interpretable than DenseNet161.

We believe the results from our proposed framework and Network Dissection are close enough from the perspectives of ranking lists. The difference may be caused by the different ways that our framework and Network Dissection perform the evaluations. The consensus score measures the similarity to the consensus on images, while Network Dissection counts the number of neurons in the intermediate layers activated by all the visual concepts, including objects, object parts, colors, materials, textures, and scenes. Furthermore, Network Dissection evaluates the interpretability of deep models using the Broden dataset with densely labeled visual objects and patterns (Bau et al., 2017), while the consensus score does not need additional datasets or the ground truth of semantics. In this way, the results of our proposed framework and Network Dissection might be slightly different.

4.4.2 Consensus versus user-study evaluations

In order to further validate the effectiveness of the proposed framework, we have also conducted user-study experiments on these five models and report the results on the second row of Table 1. The experimental settings of the user-study evaluations are as followed. For each image, we randomly choose two models from the five models and present the LIME (or SmoothGrad respectively) explanations of the two models, without giving the model information to users. Users are then requested to choose which one helps better to reveal the model's reasoning of making predictions according to their understanding, or equal if the two interpretations are equally bad or good. Each pair of models is repeated three times and represented to different users. The better one in each pair will get three points and the other one will get zero; in the equal case, both get one point. Finally, a normalization of dividing the number of images and the number of repeats (i.e. 3) is performed for each model. The user-study evaluations yield the scores indicating the model interpretability, as shown in Table 1. The results confirm that our proposed framework is capable of approximating the model interpretability.

We note that it is a small-scale user study with around thirty users. Since there is a ranking list of only five models available in Network Dissection which we can compare with, our experiments here aim to validate the effectiveness of the proposed framework by approximately evaluating model interpretability. The scores obtained in the user study may not be such accurate but the ranking list is roughly valid.

4.5 Robustness analyses of consensus

In this subsection, we investigate several factors that might affect the evaluation results with consensus, including basic explanation algorithms (e.g., LIME and SmoothGrad), the size of the committee, and the candidate pool for models in the committee.

4.5.1 Consistency between LIME and SmoothGrad

Even though the granularity of explanation results from LIME and SmoothGrad are different, which causes mismatching in mAP scores to segmentation ground truth, the consensus scores based on the two algorithms are generally consistent. The consistency has been confirmed by Fig. 6, which shows the consensus scores based on LIME and those based on SmoothGrad. The correlation coefficients are 0.825 and 0.854 respectively, indicating the strong correlations over all models on both datasets. This shows that the proposed framework can work well with a broad spectrum of basic explanation algorithms.

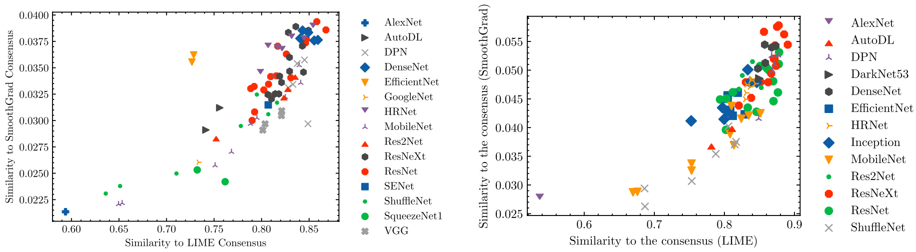


Fig. 6 Consistency between LIME and SmoothGrad. This figure shows the similarity to the consensus of SmoothGrad interpretations vs. the similarity to the consensus of LIME interpretations on the ImageNet committee (**a**) and CUB-200-2011 committee (**b**). Pearson correlation coefficients are **a** 0.825 and **b** 0.854. For concise purposes, networks in the same family are represented by the same symbol. Best viewed in color and with zoom-in (Color figure online)

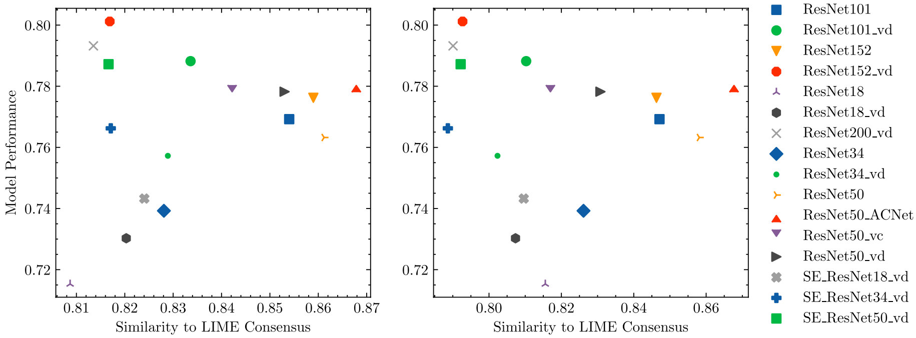


Fig. 7 Model performance vs. similarity to the consensus of LIME on ResNet family. The consensus in the left plot is voted by the ResNet family (16 models) while the right is by complete committee on ImageNet (81 models). Best viewed in color and with zoom-in (Color figure online)

4.5.2 Consistency across committees

In real-world applications, the committee-based estimations and evaluations may make inconsistent results in a committee-by-committee manner. In this work, we are interested in whether the consensus score estimations are consistent against the change of committee. Given 16 ResNet models as the targets, we form 20 independent committees by combining the 16 ResNet models with 10–20 models randomly drawn from the rest networks. In each of these 20 independent committees, we compute the consensus scores of the 16 ResNet models. We then estimate the Pearson correlation coefficients between any of these 20 results and the one in Fig. 5 (a, LIME), where the mean correlation coefficient is 0.96 with the standard deviation of 0.04. To visually show the low variance, we present the consensus scores and performance[2] of these 16 ResNet models based on the complete committee (81 models) and themselves (16 ResNet models) in Fig. 7. No large difference is observed

[2] The positive correlation between the model performance and the consensus scores does not exist in the ResNet family, as we explained before that in some local areas, especially when models are extremely large, the correlation is not always positive.

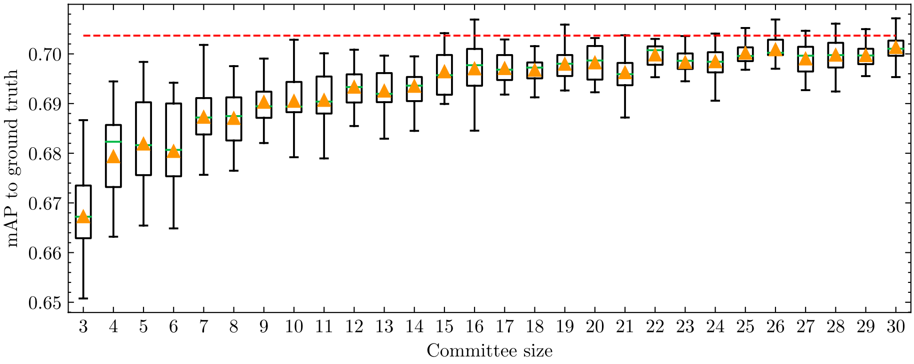


Fig. 8 Convergence of mAP between the ground truth and the consensus results based on committees of increasing sizes, using LIME on CUB-200-2011. The green lines and orange triangles are, respectively, the mean values and the median values of 20 random trials. The red dashed line is the mAP of the consensus reached by the complete committee of the original 85 models

for these two extreme cases. Thus, we can say the consensus score evaluation would be consistent against randomly picked committees.

4.5.3 Convergence over committee sizes

To understand the effect of the committee size on the consensus score estimation, we run the proposed framework using committees of various sizes formed by deep models that are randomly picked up from the pools. In Fig. 8, we plot and compare the performance of the consensus with increasing committee sizes, where we estimate the mAP between the ground truth and the consensus reached by the random committees of different sizes and 20 random trials have been done for every single size independently. It shows that the curve of mAP would quickly converge to the complete committee, while the consensus based on a small proportion of committee (e.g., 15 networks) works well enough even compared to the complete committee of 85 networks.

5 Discussions: limitations and strengths with future works

In this section, we discuss several limitations and strengths in our studies, with interesting directions for future works.

5.1 Limitations

First of all, our studies are based on the explanation algorithms. We propose to study the features used by deep models using the explanation results (i.e., the importance of superpixels/pixels in the image for prediction). The correctness of these explanation algorithms might affect our results. However, we use two independent algorithms, including LIME (Ribeiro et al., 2016) and SmoothGrad (Smilkov et al., 2017), which attribute feature importance in two different scales i.e., superpixels and pixels. Both algorithms lead to the same observations and conclusive results (see Sect. 4.5 for the consistency between results

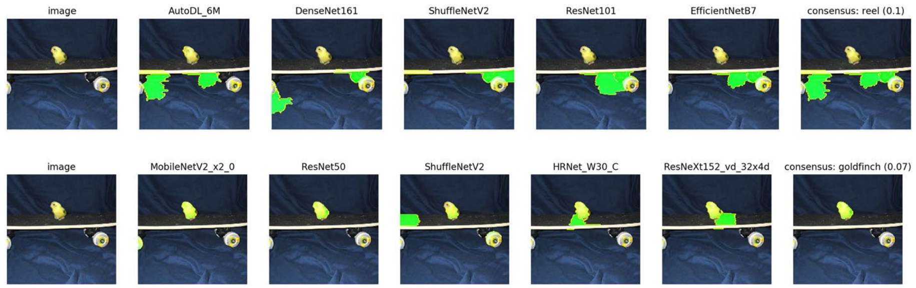


Fig. 9 Visualization of an image from the MS-COCO dataset (Lin et al., 2014) for showing the strengths of cross-model consensus of explanations, where the predicted label with probability is noted. Models are not adapted to the COCO dataset

obtained by LIME and SmoothGrad). Thus, we believe the explanation algorithms here are (almost) trustworthy and it is appropriate to use explanation results as a proxy to analyze features. For future research, we would include more advanced explanation algorithms to confirm our observations.

We obtain some interesting observations from our experiments and make conclusions using multiple datasets. However, the image classification datasets used in our experiments have some limitations—every image in the dataset only consists of one visual object for classification. It is reasonable to doubt that when multiple visual objects (rather than the target for classification) and complicated visual patterns for background (Chen et al., 2017; Koh & Liang, 2017) co-exist in an image, the cross-model consensus of explanations may no longer overlap to the ground truth semantic segmentation. To showcase our approach, we include an example from the COCO dataset (Lin et al., 2014) in Fig. 9, where multiple objects co-exist in the image and the consensus *partly* matches the segmentation. To address this issue, our future work would focus on the datasets with multiple visual objects and complicated background for object detection, segmentation, and multi-label classification tasks.

Finally, only well-known models with good performance have been included in the committee. It would probably bring some bias in our analysis, but it would not cause many problems in practice because these models would be one of the first choices or frequently used in many applications for relevance. Moreover, if the committee consists of a large number of random-guess models, the consensus would become a constant matrix. To avoid this case and simplify the analyses, we consider well-known models with good performance in this work. In our future work, we would include more models with diverse performances to seek more observations and will try to explain more complicated models such as Transformers (Yuan et al., 2021).

5.2 Strengths

In addition to the limitations, we demonstrate several strengths of cross-model consensus of explanations for further studies.

As was shown in Fig. 8, with a larger committee, the consensus would slowly converge to a stable set of common features that aligns with the segmentation ground truth of the dataset. This experiment further demonstrates the capacity of consensus to precisely position the visual objects for classification. Thus, in our future work, we would like to use

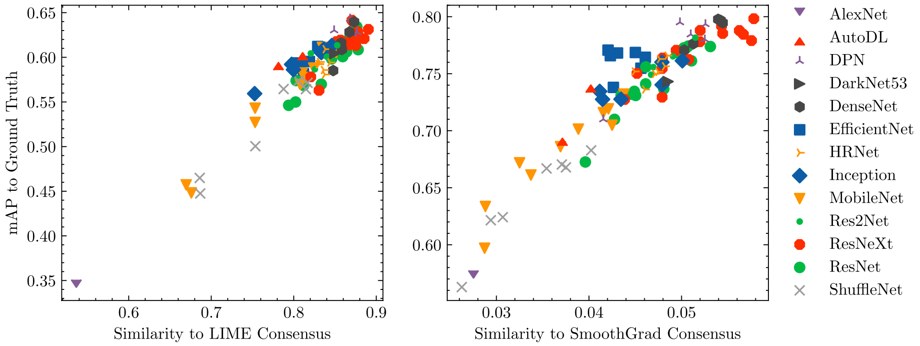


Fig. 10 Correlation between mAP scores to the segmentation ground truth and the consensus scores using **a** LIME and **b** SmoothGrad with the CUB-200-2011 dataset over 85 models (of the committee). Pearson correlation coefficients are 0.885 (p value 3e−29) for LIME and 0.906 (p value 8e−33) for SmoothGrad. For concise purposes, networks in the same family are represented by the same symbol. Best viewed in color and with zoom-in (Color figure online)

consensus based on a committee of image classification models to detect the visual objects in the image.

Model performance is one of the most critical metrics in most practical scenarios. Estimations of model performance are needed in these situations, especially when there are no (or few) validation samples. Our second observation that a model in the committee with a higher consensus score usually performs better in terms of testing accuracy, would be helpful to relatively estimate the performance of models.

We believe that the cross-model consensus of explanations, or the common features, is an explanation of **data**, instead of explanations for individual models, that aim to explain the **model**. Informally, we consider the explanations as a conditional probability (of importance) of features f given a trained deep model $\boldsymbol{M}$, denoted as $p(f|\boldsymbol{M})$. Higher values of $p(f|\boldsymbol{M})$ indicate that the features f are (supposed to be) more important to solve the task, from the view of the given model $\boldsymbol{M}$. Then intuitively, the cross-model consensus of explanations is to marginalize out the variable of models, i.e., $p(f) = \int_{\boldsymbol{M}} p(f|\boldsymbol{M})p(\boldsymbol{M})\mathrm{d}\boldsymbol{M}$, to indicate the feature importance from the view of data. In practice, the intractable integration is approximated by the discrete summation (15 models approximate well, cf Sect. 4.5). The common features found by the consensus are approximately equivalent to the features that are discriminative for solving the task, which resides in data. Therefore, the cross-model consensus of explanations would be capable of identifying the discriminative features.

Following the previous notations, individual models are supposed to use them to achieve good performance and thus the consensus score measures the quantity of discriminative features that the model uses to make predictions. A higher consensus score indicates that the model is more likely to achieve good performance. More investigations on this may lead to theoretical proofs of our second observation.

Furthermore, our experiments with both explanation algorithms on all datasets have found that consensus scores are correlated to the interpretability scores of the models, even though interpretability scores were evaluated through totally different ways—network dissections (Bau et al., 2017) and user studies. Actually, network dissections evaluate the interpretability of a model through matching its activation maps in intermediate layers with the ground truth segmentation of visual concepts in the image. A model with higher interpretability should have more convolutional filters activated at the visual patterns/objects for classification. In this way, we particularly measure the similarity between the explanation results obtained for every model and the segmentation ground truth of images. We found that the models' segmentation-explanation similarity significantly correlates to their consensus scores (see Fig. 10). This observation might encourage us to further study the connections between interpretability and consensus scores in the future work.

6 Conclusion

In this paper, we study the common features shared by various deep models for image classification. Specifically, given the explanation results obtained by explanation algorithms, we propose to aggregate the explanation results from different models and obtain the cross-model consensus of explanations through voting. To understand features used by every model and the common ones, we measure the consensus scores as the similarity between the consensus and the explanation of individual models.

Our empirical studies based on comprehensive experiments using 80+ deep models on five datasets/tasks find that (i) the consensus aligns with the ground truth semantic segmentation of the visual objects for classification; (ii) models with higher consensus scores would enjoy better testing accuracy; and (iii) the consensus scores correlate to the interpretability scores. In addition to the main claims, we also include additional experiments to demonstrate robustness and consistency of the proposed cross-model consensus of explanations in explanation algorithms, formed committees, the committee size, and random selections on various datasets. All these studies confirm the applicability of consensus as a proxy to study and analyze the common features shared by different models in our research. Furthermore, several open issues and strengths have been discussed, with future directions introduced. Hereby, we are encouraged to adopt the consensus and consensus scores for better understanding the behaviors of deep models.

Appendix 1: Definitions of metrics

Average precision

Average Precision (AP)[3] computes the area under the precision-recall curve:

$$AP = \sum_{n}(R_n - R_{n-1})P_n, \tag{1}$$

where R_n and P_n are the recall and precision values at the n^{th} threshold.

Pearson correlation coefficient

The Pearson correlation coefficient[4] measures the linear relationship between two variables by

[3] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.average_precision_score.html.

[4] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.pearsonr.html.

$$r = \frac{\sum(x-\bar{x})(y-\bar{y})}{\sqrt{(x-\bar{x})^2(y-\bar{y})^2}}, \tag{2}$$

where $\bar{x}$ is the mean of x, $\bar{y}$ is the mean of y.

Spearman rank-order correlation coefficient

The Spearman rank-order correlation coefficient[5] is a nonparametric measure of the monotonicity of the relationship between two variables by

$$r = \frac{Cov(R(X), R(Y))}{\sigma_{R(X)}\sigma_{R(Y)}}, \tag{3}$$

where X and Y are two variables, $R(X)$ and $R(Y)$ are the ranking variables related to X and Y respectively, $\sigma_{R(X)}$ and $\sigma_{R(Y)}$ are the standard deviations of the rank variables.

Appendix 2: Applicability with random committees over more datasets

To demonstrate the applicability of the proposed framework, we extend our experiments using networks randomly picked up from the pool to other datasets, including Stanford Cars 196 (Krause et al., 2013), Oxford Flowers 102 (Nilsback & Zisserman, 2008) and Foods 101 (Bossard et al., 2014). These datasets respectively contain 196 types of automobiles, 102 categories of flowers and 101 kinds of foods, with 8144, 1030, 75,750 training samples. The fine-tuning procedure on these three datasets is the same as on CUB-200-2011, to prepare the committee. However, given the convergence over committee size (Fig. 8), which suggests a committee of more than 15 models, we randomly train 20 models for each of the three datasets. The results in Fig. 11 confirm that the positive correlations between the consensus score and model performance exist for a wide range of models on ubiquitous datasets/tasks.

Appendix 3: References of network structures

Most frequently-used structures of deep models have been evaluated in this paper, including AlexNet (Krizhevsky et al., 2012), ResNet (He et al., 2016), ResNeXt (Xie et al., 2017), SEResNet (Hu et al., 2018), ShuffleNet (Zhang et al., 2018; Ma et al., 2018), MobileNet (Howard et al., 2017; Sandler et al., 2018; Howard et al., 2019), VGG (Simonyan & Zisserman, 2015), GoogleNet (Szegedy et al., 2015), Inception (Szegedy et al., 2015), Xception (Chollet, 2017), DarkNet (Redmon et al., 2016, 2018), DenseNet (Huang et al., 2017), DPN (Chen et al., 2017), SqueezeNet (Iandola et al., 2016), EfficientNet (Tan & Le, 2019), Res2Net (Gao et al., 2019), HRNet (Wang et al., 2020), Darts (Liu et al., 2018), AcNet (Ding et al., 2019) and their variants.

[5] https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.spearmanr.html.

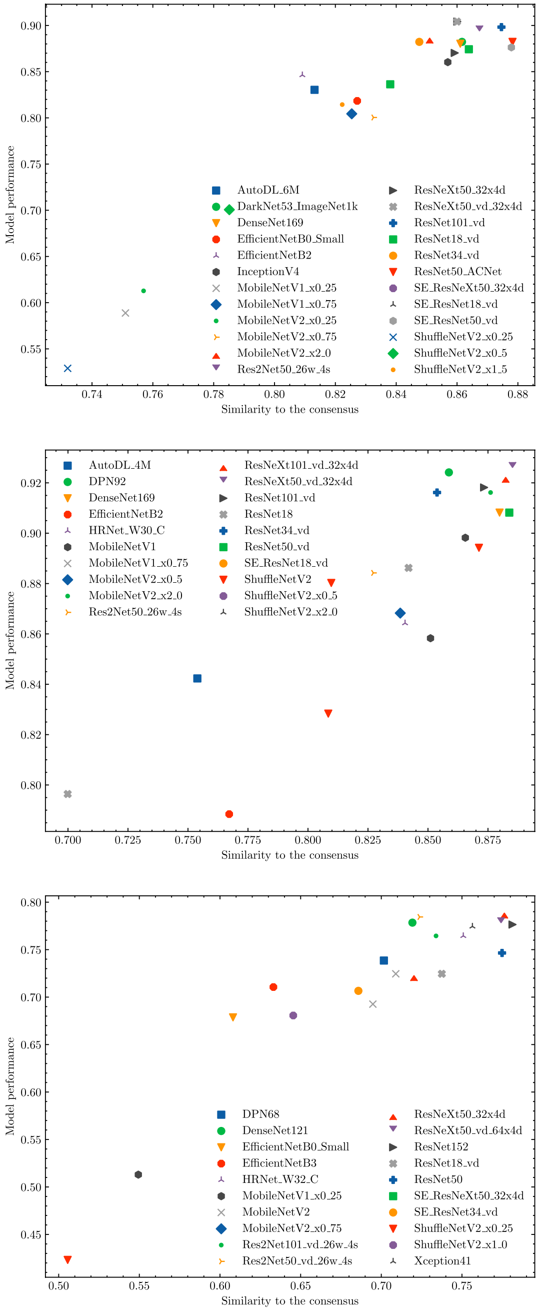


Fig. 11 Model performance vs. the consensus scores using LIME on Stanford Cars 196 (Krause et al., 2013), Oxford Flowers 102 (Nilsback & Zisserman, 2008) and Foods 101 (Bossard et al., 2014). Pearson correlation coefficients are 0.952, 0.879 and 0.913 respectively. Best viewed in color and with zoom-in (Color figure online)

Appendix 4: Numerical report of main plots

Table 2 Numerical report of model performance and similarity to the consensus using LIME and SmoothGrad over 81 models on ImageNet in sub-table(a) corresponding to Figs. 5a, c and 6a, and over 85 models on CUB-200-2011 in sub-table(b) corresponding to Figs. 4 and 5b, d, Figs. 6b and 10

	perf.	Consensus scores w/ LIME	Consensus scores w/ Smooth-Grad
(a) *On ImageNet*			
AlexNet	0.575	0.594	0.0214
AutoDL_4M	0.752	0.756	0.0312
AutoDL_6M	0.776	0.741	0.0291
DPN107	0.798	0.828	0.0331
DPN131	0.802	0.833	0.0334
DPN68	0.766	0.849	0.0297
DPN92	0.775	0.845	0.0357
DPN98	0.798	0.837	0.0354
DenseNet121	0.755	0.859	0.0376
DenseNet161	0.781	0.849	0.0383
DenseNet169	0.765	0.855	0.0376
DenseNet201	0.779	0.843	0.0385
DenseNet264	0.761	0.841	0.0378
EfficientNetB0	0.754	0.727	0.0355
EfficientNetB0_Small	0.708	0.729	0.0362
GoogleNet	0.726	0.734	0.0260
HRNet_W18_C	0.766	0.854	0.0388
HRNet_W30_C	0.776	0.832	0.0378
HRNet_W32_C	0.781	0.845	0.0376
HRNet_W40_C	0.773	0.822	0.0366
HRNet_W44_C	0.782	0.817	0.0368
HRNet_W48_C	0.794	0.807	0.0369
HRNet_W64_C	0.784	0.799	0.0344
MobileNetV1	0.711	0.825	0.0322

Table 2 (continued)

	perf.	Consensus scores w/ LIME	Consensus scores w/ Smooth-Grad
MobileNetV1_x0_25	0.513	0.653	0.0222
MobileNetV1_x0_5	0.640	0.751	0.0257
MobileNetV1_x0_75	0.697	0.788	0.0297
MobileNetV2	0.742	0.812	0.0342
MobileNetV2_x0_25	0.534	0.650	0.0221
MobileNetV2_x0_5	0.642	0.768	0.0270
MobileNetV2_x0_75	0.709	0.795	0.0302
MobileNetV2_x1_5	0.737	0.841	0.0336
MobileNetV2_x2_0	0.744	0.840	0.0352
Res2Net101_vd_26w_4s	0.780	0.752	0.0285
Res2Net50_14w_8s	0.781	0.823	0.0324
Res2Net50_26w_4s	0.780	0.835	0.0343
Res2Net50_vd_26w_4s	0.790	0.828	0.0332
ResNeXt101_32x4d	0.784	0.843	0.0371
ResNeXt101_vd_32x4d	0.795	0.830	0.0347
ResNeXt101_vd_64x4d	0.784	0.821	0.0336
ResNeXt152_32x4d	0.782	0.842	0.0377
ResNeXt152_64x4d	0.787	0.828	0.0383
ResNeXt152_vd_32x4d	0.792	0.807	0.0325
ResNeXt152_vd_64x4d	0.790	0.814	0.0325
ResNeXt50_32x4d	0.765	0.849	0.0385
ResNeXt50_64x4d	0.784	0.836	0.0389
ResNeXt50_vd_32x4d	0.790	0.844	0.0346
ResNeXt50_vd_64x4d	0.792	0.829	0.0360
ResNet101	0.769	0.847	0.0377

Table 2 (continued)

	perf.	Consensus scores w/ LIME	Consensus scores w/ Smooth-Grad
ResNet101_vd	0.788	0.810	0.0323
ResNet152	0.776	0.846	0.0374
ResNet152_vd	0.801	0.793	0.0308
ResNet18	0.715	0.816	0.0342
ResNet18_vd	0.730	0.807	0.0334
ResNet200_vd	0.793	0.790	0.0300
ResNet34	0.739	0.826	0.0363
ResNet34_vd	0.757	0.802	0.0329
ResNet50	0.763	0.858	0.0394
ResNet50_ACNet	0.780	0.868	0.0386
ResNet50_vc	0.778	0.817	0.0370
ResNet50_vd	0.778	0.831	0.0341
SENet154_vd	0.803	0.807	0.0315
SE_ResNeXt101_32x4d	0.781	0.818	0.0325
SE_ResNeXt50_32x4d	0.775	0.810	0.0321
SE_ResNeXt50_vd_32x4d	0.797	0.819	0.0342
SE_ResNet18_vd	0.743	0.810	0.0342
SE_ResNet34_vd	0.766	0.789	0.0330
SE_ResNet50_vd	0.787	0.792	0.0332
ShuffleNetV2	0.706	0.795	0.0325
ShuffleNetV2_x0_25	0.507	0.636	0.0231
ShuffleNetV2_x0_33	0.547	0.651	0.0238
ShuffleNetV2_x0_5	0.611	0.710	0.0250
ShuffleNetV2_x1_0	0.689	0.778	0.0295
ShuffleNetV2_x1_5	0.712	0.807	0.0306

Table 2 (continued)

	perf.	Consensus scores w/ LIME	Consensus scores w/ Smooth-Grad		
ShuffleNetV2_x2_0	0.738	0.816	0.0317		
SqueezeNet1_0	0.602	0.732	0.0253		
SqueezeNet1_1	0.613	0.762	0.0242		
VGG11	0.694	0.801	0.0291		
VGG13	0.697	0.804	0.0297		
VGG16	0.714	0.821	0.0305		
VGG19	0.722	0.821	0.0309		
	perf.	Consensus Scores w/ LIME	Consensus Scores w/ Smooth- Grad	mAP between g.t. of segmentation and LIME explanation	mAP between g.t. of segmentation and SmoothGrad explanation
(b) *On CUB-200-2011*					
AlexNet	0.507	0.536	0.0275	0.343	0.571
AutoDL_4M	0.728	0.781	0.0371	0.594	0.693
AutoDL_6M	0.754	0.811	0.0402	0.605	0.740
DPN107	0.830	0.867	0.0525	0.630	0.780
DPN131	0.800	0.868	0.0498	0.643	0.795
DPN68	0.795	0.849	0.0415	0.630	0.710
DPN92	0.806	0.872	0.0510	0.626	0.784
DPN98	0.815	0.877	0.0526	0.628	0.793
DarkNet53_ImageNet1k	0.782	0.850	0.0485	0.604	0.743
DenseNet121	0.771	0.848	0.0503	0.585	0.771
DenseNet161	0.813	0.873	0.0542	0.640	0.797
DenseNet169	0.792	0.858	0.0513	0.609	0.776
DenseNet201	0.805	0.858	0.0544	0.616	0.795

Table 2 (continued)

	perf.	Consensus Scores w/ LIME	Consensus Scores w/ Smooth- Grad	mAP between g.t. of segmentation and LIME explanation	mAP between g.t. of segmentation and SmoothGrad explanation
DenseNet264	0.789	0.868	0.0540	0.628	0.798
EfficientNetB0	0.765	0.805	0.0450	0.594	0.769
EfficientNetB0_Small	0.737	0.805	0.0426	0.589	0.738
EfficientNetB1	0.775	0.805	0.0456	0.593	0.755
EfficientNetB2	0.787	0.819	0.0461	0.595	0.764
EfficientNetB3	0.791	0.812	0.0421	0.582	0.771
EfficientNetB4	0.792	0.829	0.0423	0.612	0.766
EfficientNetB5	0.774	0.808	0.0431	0.591	0.768
HRNet_W18_C	0.754	0.831	0.0461	0.592	0.736
HRNet_W30_C	0.770	0.832	0.0475	0.595	0.752
HRNet_W32_C	0.785	0.836	0.0471	0.586	0.750
HRNet_W40_C	0.750	0.844	0.0476	0.594	0.763
HRNet_W44_C	0.788	0.830	0.0449	0.592	0.752
HRNet_W48_C	0.796	0.838	0.0482	0.581	0.757
HRNet_W64_C	0.791	0.838	0.0485	0.609	0.766
InceptionV4	0.745	0.797	0.0435	0.592	0.728
MobileNetV1	0.741	0.824	0.0415	0.588	0.716
MobileNetV1_x0_25	0.557	0.676	0.0288	0.448	0.634
MobileNetV1_x0_5	0.655	0.753	0.0325	0.527	0.672
MobileNetV1_x0_75	0.688	0.808	0.0388	0.569	0.701
MobileNetV2	0.737	0.810	0.0438	0.582	0.732
MobileNetV2_x0_25	0.511	0.670	0.0287	0.457	0.597
MobileNetV2_x0_5	0.665	0.753	0.0337	0.543	0.661
MobileNetV2_x0_75	0.715	0.814	0.0369	0.577	0.686

Table 2 (continued)

	perf.	Consensus Scores w/ LIME	Consensus Scores w/ Smooth- Grad	mAP between g.t. of segmentation and LIME explanation	mAP between g.t. of segmentation and SmoothGrad explanation
MobileNetV2_x1_5	0.756	0.835	0.0421	0.611	0.719
MobileNetV2_x2_0	0.781	0.851	0.0425	0.605	0.705
Res2Net101_vd_26w_4s	0.799	0.853	0.0470	0.613	0.756
Res2Net50_14w_8s	0.789	0.826	0.0491	0.587	0.765
Res2Net50_26w_4s	0.768	0.840	0.0515	0.601	0.782
Res2Net50_vd_26w_4s	0.783	0.821	0.0467	0.604	0.749
ResNeXt101_32x4d	0.818	0.877	0.0578	0.629	0.798
ResNeXt101_32x8d_wsl	0.768	0.831	0.0479	0.563	0.755
ResNeXt101_vd_32x4d	0.816	0.867	0.0494	0.614	0.771
ResNeXt101_vd_64x4d	0.824	0.871	0.0520	0.642	0.778
ResNeXt152_32x4d	0.815	0.872	0.0543	0.619	0.792
ResNeXt152_64x4d	0.834	0.875	0.0576	0.613	0.779
ResNeXt152_vd_32x4d	0.820	0.872	0.0520	0.640	0.788
ResNeXt152_vd_64x4d	0.822	0.852	0.0479	0.618	0.764
ResNeXt50_32x4d	0.809	0.856	0.0567	0.619	0.785
ResNeXt50_64x4d	0.814	0.885	0.0562	0.621	0.788
ResNeXt50_vd_32x4d	0.806	0.874	0.0508	0.627	0.762
ResNeXt50_vd_64x4d	0.820	0.890	0.0544	0.631	0.785
ResNet101	0.784	0.878	0.0511	0.620	0.761
ResNet101_vd	0.813	0.864	0.0499	0.606	0.766
ResNet152	0.799	0.859	0.0506	0.601	0.773
ResNet152_vd	0.797	0.851	0.0507	0.613	0.774
ResNet18	0.726	0.794	0.0449	0.546	0.735
ResNet18_vd	0.754	0.846	0.0428	0.598	0.710

Table 2 (continued)

	perf.	Consensus Scores w/ LIME	Consensus Scores w/ Smooth- Grad	mAP between g.t. of segmentation and LIME explanation	mAP between g.t. of segmentation and SmoothGrad explanation
ResNet200_vd	0.813	0.861	0.0502	0.618	0.773
ResNet34	0.758	0.812	0.0461	0.569	0.756
ResNet34_vd	0.771	0.833	0.0435	0.570	0.731
ResNet50	0.776	0.878	0.0531	0.609	0.774
ResNet50_ACNet	0.782	0.870	0.0481	0.619	0.737
ResNet50_vd	0.795	0.876	0.0461	0.634	0.741
SE_ResNeXt101_32x4d	0.793	0.838	0.0452	0.605	0.750
SE_ResNeXt50_32x4d	0.798	0.821	0.0438	0.578	0.727
SE_ResNeXt50_vd_32x4d	0.799	0.863	0.0479	0.617	0.729
SE_ResNet18_vd	0.727	0.802	0.0396	0.550	0.673
SE_ResNet34_vd	0.754	0.803	0.0450	0.574	0.731
SE_ResNet50_vd	0.771	0.870	0.0446	0.616	0.732
ShuffleNetV2	0.696	0.817	0.0375	0.571	0.668
ShuffleNetV2_x0_25	0.519	0.687	0.0263	0.448	0.563
ShuffleNetV2_x0_33	0.530	0.686	0.0294	0.465	0.622
ShuffleNetV2_x0_5	0.605	0.753	0.0307	0.500	0.624
ShuffleNetV2_x1_0	0.695	0.788	0.0354	0.564	0.667
ShuffleNetV2_x1_5	0.728	0.815	0.0371	0.564	0.670
ShuffleNetV2_x2_0	0.731	0.806	0.0402	0.574	0.683
Xception41	0.801	0.833	0.0501	0.605	0.761
Xception41_deeplab	0.775	0.753	0.0412	0.559	0.734
Xception65	0.801	0.837	0.0479	0.609	0.740
Xception65_deeplab	0.747	0.800	0.0415	0.586	0.728

Table 2 (continued)

	perf.	Consensus Scores w/ LIME	Consensus Scores w/ Smooth- Grad	mAP between g.t. of segmentation and LIME explanation	mAP between g.t. of segmentation and SmoothGrad explanation
Xception71	0.775	0.846	0.0479	0.613	0.760
consensus	0.859	N/A	N/A	0.704	0.818

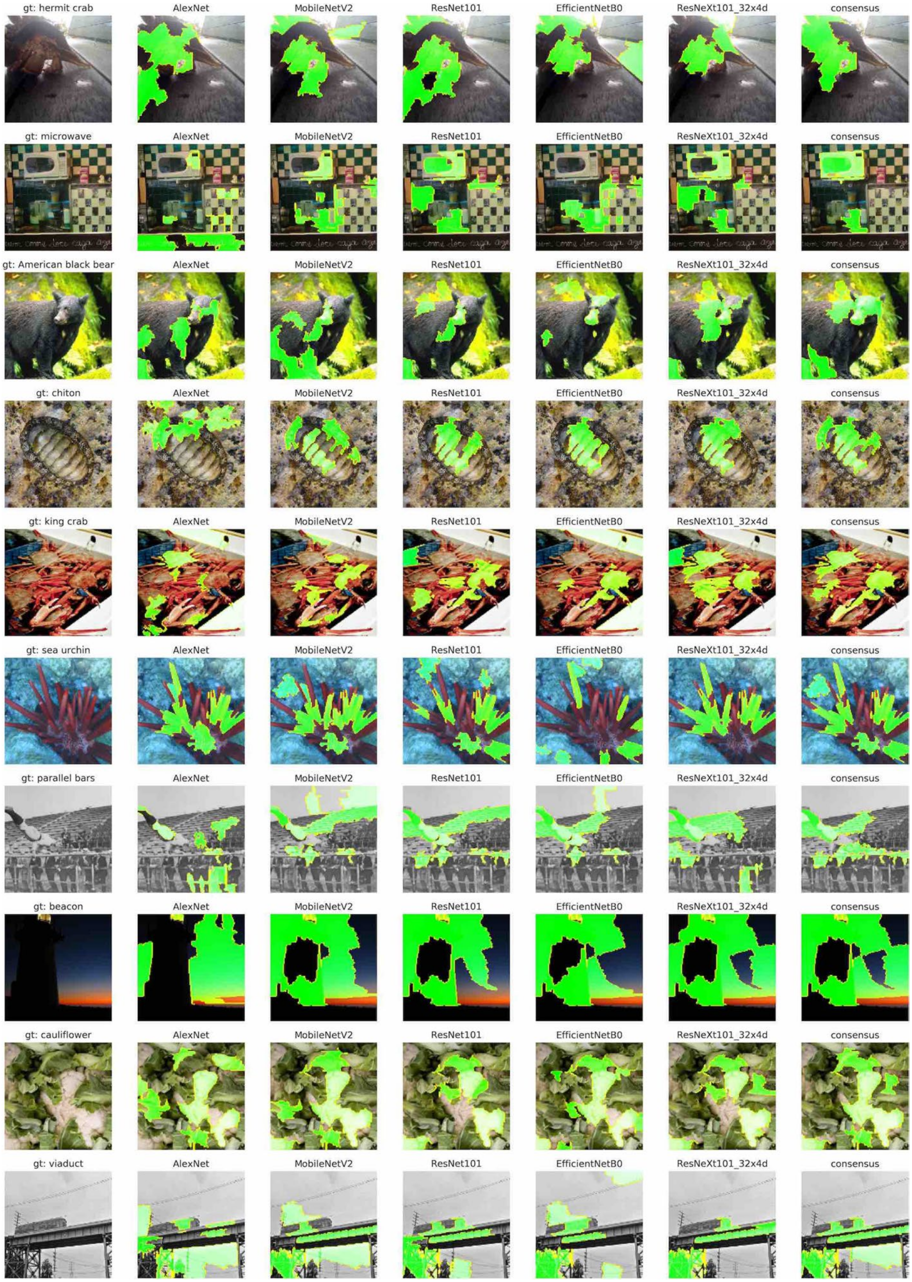


Fig. 12 More visual comparisons between the consensus and the explanations of deep models with LIME on samples from ImageNet. Note that the consensus (last column) is the cross-model consensus of explanations

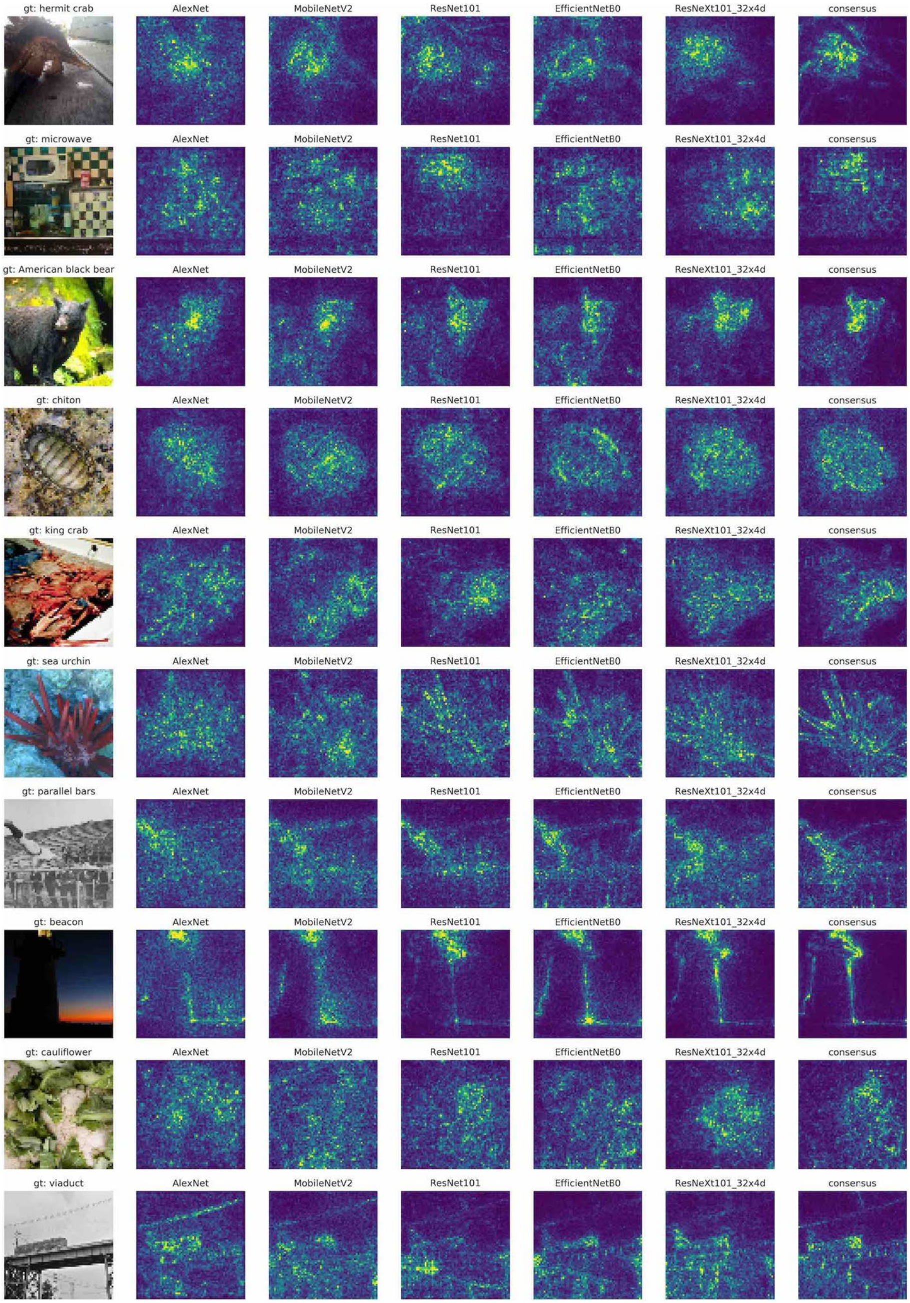


Fig. 13 More visual comparisons between the consensus and the explanations of deep models with SmoothGrad on samples from ImageNet. Note that the consensus (last column) is the cross-model consensus of explanations

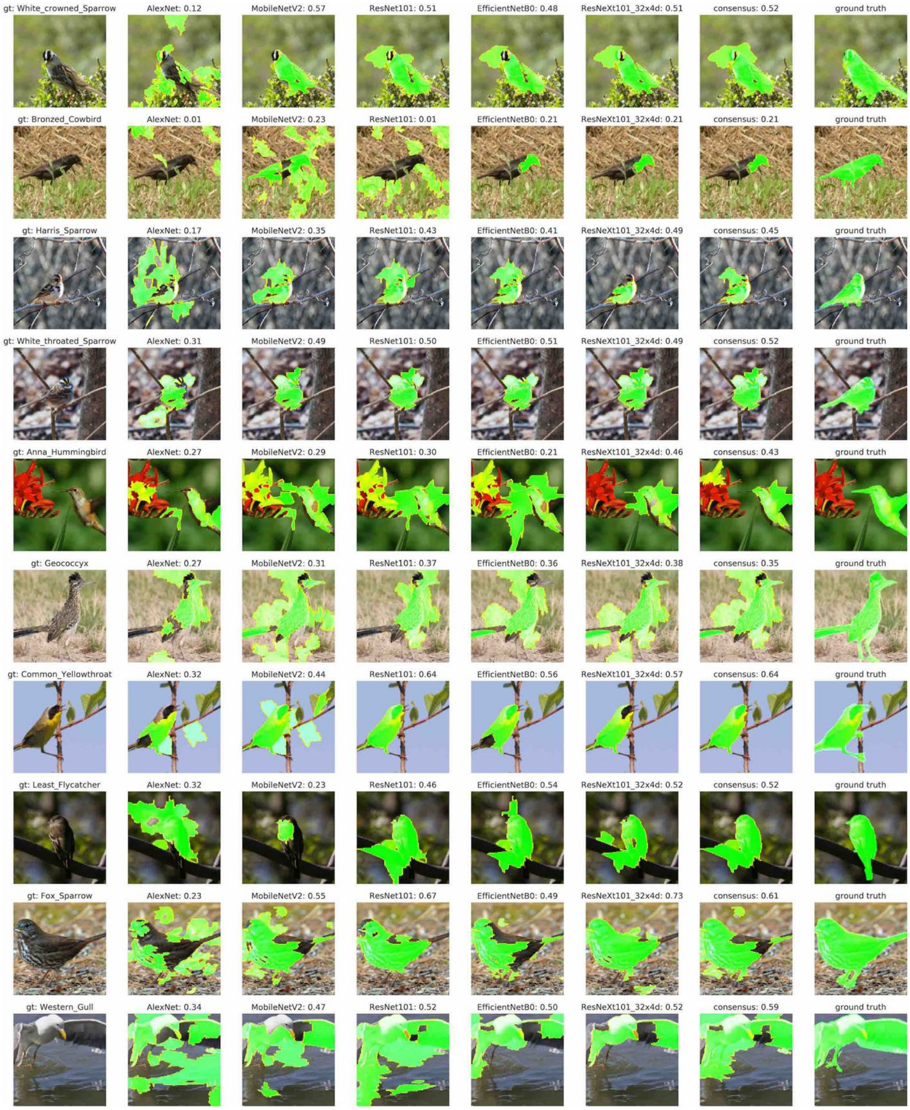


Fig. 14 More visual comparisons between the consensus and the explanations of deep models with LIME on samples from CUB-200-2011, where the pixel-wise annotations of image segmentation are available and the mAPs are measured for the similarity to the segmentation ground truth. Note that the consensus (second last column) is the cross-model consensus of explanations

Due to the large number of deep models evaluated, Figs. 4, 5, 6 and 10 grouped some that are of the same architecture. Here, we report all of the corresponding numerical results in Table 2 with a smaller scale.

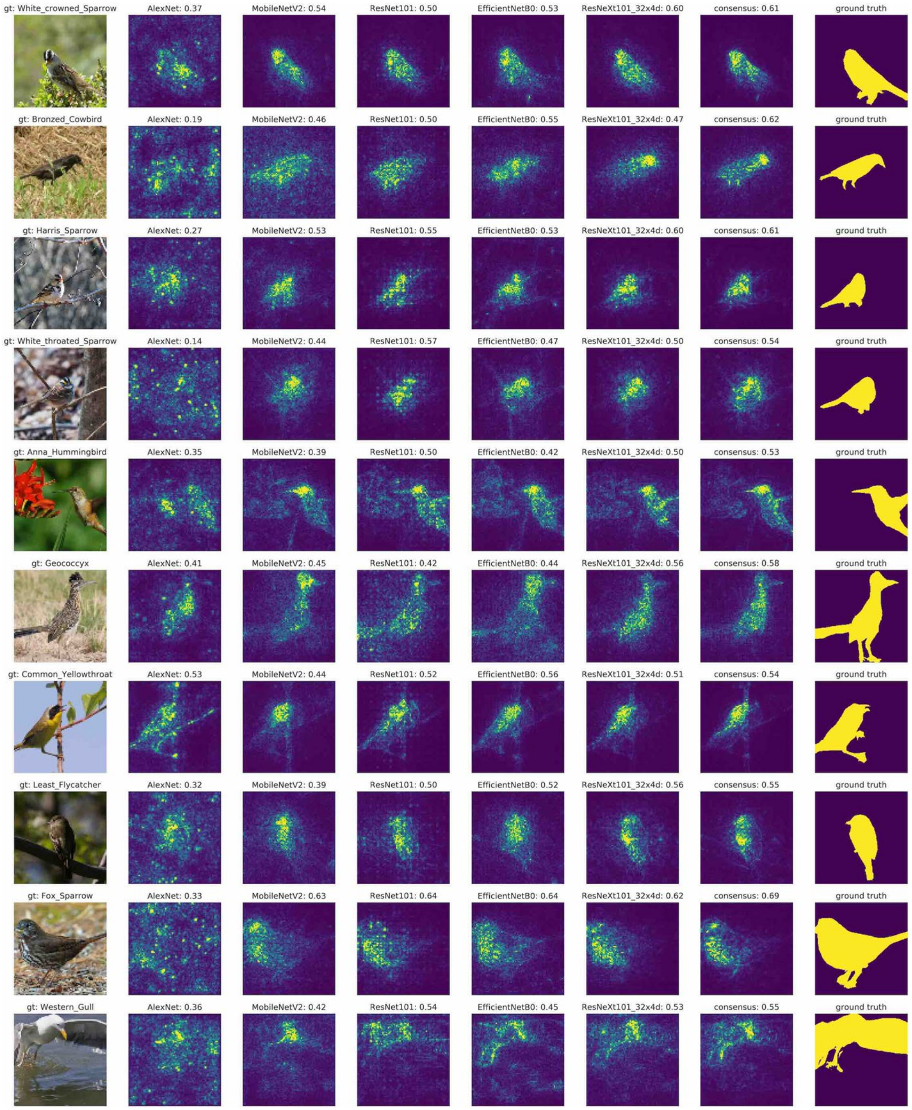


Fig. 15 More visual comparisons between the consensus and the explanations of deep models with SmoothGrad on samples from CUB-200-2011, where the pixel-wise annotations of image segmentation are available and the mAPs are measured for the similarity to the segmentation ground truth. Note that the consensus (second last column) is the cross-model consensus of explanations

Appendix 5: More visualization results

We present more visualization results of cross-model consensus of explanations obtained from LIME and SmoothGrad in Figs. 12, 13, 14 and 15, where the samples are from ImageNet and CUB-200-2011.

Author contributions Dr. Xuhong Li contributed in the original research ideas, implemented algorithms, conducted parts of experiments, and wrote parts of the manuscript. Dr. Haoyi Xiong formulated the research problems, proposed part of algorithms, wrote parts of the manuscript and involved in discussions. Dr. Siyu Huang wrote parts of the manuscript and involved in discussions. Mr. Shilei Ji involved in discussions. Prof. Dejing Dou involved in discussions and oversaw the research progress.

Funding Xuhong Li and Haoyi Xiong are supported by National Key R&D Program of China (No. 2021ZD0110303).

Availability of data and materials All experiments were based on public available open-source datasets.

Code availability The source codes could be found at https://github.com/PaddlePaddle/InterpretDL.

Declarations

Conflict of interest Not applicable.

Ethics approval Not applicable (all based on public available open-source datasets).

Consent to participate Not applicable (all based on public available open-source datasets).

Consent for publication Not applicable (all based on public available open-source datasets).

References

Adebayo, J., Gilmer, J., Muelly, M. Goodfellow, I., Hardt, M., & Kim, B. (2018). Sanity checks for saliency maps. In *Advances in neural information processing systems (NeurIPS)* (pp. 9505–9515).

Afrabandpey, H., Peltola, T., Piironen, J., Vehtari, A., & Kaski, S. (2020). A decision-theoretic approach for model interpretability in bayesian framework. *Machine Learning, 109*, 1855–1876.

Agarwal, S., Nguyen, T. T., Nguyen, T. L., & Ifrim, G. (2021). Ranking by aggregating referees: Evaluating the informativeness of explanation methods for time series classification. In *International workshop on advanced analytics and learning on temporal data* (pp. 3–20). Springer

Ahern, I., Noack, A., Guzman-Nateras, L., Dou, D., Li, B., & Huan, J. (2019). Normlime: A new feature importance metric for explaining deep neural networks. arXiv:1909.04200

Ancona, M., Ceolini, E., Öztireli, C., & Gross, M. (2018). Towards better understanding of gradient-based attribution methods for deep neural networks. In *International conference on learning representations (ICLR)*.

Atanasova, P., Simonsen, J. G., Lioma, C., & Augenstein, I. (2020). Generating fact checking explanations. In *Proceedings of the 58th annual meeting of the association for computational linguistics, ACL 2020*, Online, July 5–10, 2020.

Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K.-R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE, 10*(7), 0130140.

Bau, D., Zhou, B., Khosla, A., Oliva, A., & Torralba, A. (2017). Network dissection: Quantifying interpretability of deep visual representations. In *IEEE transactions on pattern analysis and machine intelligence (TPAMI)* (pp. 6541–6549).

Bossard, L., Guillaumin, M., & Van Gool, L. (2014). Food-101–mining discriminative components with random forests. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 446–461). Springer.

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

Chen, C., Li, O., Tao, D., Barnett, A., Rudin, C., & Su, J. K. (2019). *This looks like that: Deep learning for interpretable image recognition*.

Chen, X., Liu, C., Li, B., Lu, K., & Song, D.(2017). Targeted backdoor attacks on deep learning systems using data poisoning. arXiv:1712.05526

Chen, Y., Li, J., Xiao, H., Jin, X., Yan, S., & Feng, J. (2017). Dual path networks. In *Advances in neural information processing systems (NeurIPS)* (pp. 4467–4475).

Chollet, F. (2017). Xception: Deep learning with depthwise separable convolutions. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1251–1258).

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). Imagenet: A large-scale hierarchical image database. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 248–255).

Ding, X., Guo, Y., Ding, G., & Han, J. (2019). Acnet: Strengthening the kernel skeletons for powerful cnn via asymmetric convolution blocks. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1911–1920).

Doshi-Velez, F., & Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv:1702.08608

Fisher, A., Rudin, C., & Dominici, F. (2019). All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously. *Journal of Machine Learning Research, 20*(177), 1–81.

Gao, S., Cheng, M.-M., Zhao, K., Zhang, X.-Y., Yang, M.-H., &Torr, P. H. (2019). Res2net: A new multiscale backbone architecture.

Gu, T., Dolan-Gavitt, B., & Garg, S. (2017). Badnets: Identifying vulnerabilities in the machine learning model supply chain. arXiv:1708.06733

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *IEEE conference on computer vision and pattern recognition (CVPR).*

Hooker, S., Erhan, D., Kindermans, P.-J., & Kim, B. (2019). A benchmark for interpretability methods in deep neural networks. In *Advances in neural information processing systems (NeurIPS)* (pp. 9737–9748).

Howard, A., Sandler, M., Chu, G., Chen, L.-C., Chen, B., Tan, M., Wang, W., Zhu, Y., Pang, R., Vasudevan, V., & Le, Q. V. (2019). Searching for mobilenetv3. In *IEEE conference on computer vision and pattern recognition (CVPR).*

Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., Andreetto, M., & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv:1704.04861

Hu, J., Shen, L., & Sun, G. (2018). Squeeze-and-excitation networks. In *IEEE conference on computer vision and pattern recognition (CVPR).*

Huang, G., Liu, Z., Van Der Maaten, L., & Weinberger, K. Q. (2017). Densely connected convolutional networks. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 4700–4708).

Iandola, F. N., Han, S., Moskewicz, M. W., Ashraf, K., Dally, W. J., & Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and< 0.5 mb model size. arXiv:1602.07360

Jeyakumar, J. V., Noor, J., Cheng, Y.-H., Garcia, L., & Srivastava, M. (2020). How can i explain this to you? An empirical study of deep neural network explanation methods. In *Advances in neural information processing systems (NeurIPS).*

Jo, S., & Yu, I.-J. (2021). Puzzle-cam: Improved localization via matching partial and full features. arXiv:2101.11253

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., & Viegas, F. (2018). Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning (ICML)* (pp. 2668–2677).

Kim, J.-H., Choo, W., & Song, H.O. (2020). Puzzle mix: Exploiting saliency and local statistics for optimal mixup. In *Proceedings of the international conference on machine learning*.

Koh, P. W., & Liang, P. (2017). Understanding black-box predictions via influence functions. In *International conference on machine learning (ICML)* (pp. 1885–1894). PMLR.

Krause, J., Stark, M., Deng, J., & Fei-Fei, L. (2013). 3D object representations for fine-grained categorization. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 554–561).

Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NeurIPS).*

Lage, I., Chen, E., He, J., Narayanan, M., Kim, B., Gershman, S., & Doshi-Velez, F. (2019).An evaluation of the human-interpretability of explanation. arXiv:1902.00006

Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., & Zitnick, C. L. (2014). Microsoft COCO: Common objects in context. In *Proceedings of the European conference on computer vision*.

Lin, Y.-S., Lee, W.-C., & Celik, Z. B.(2020). What do you see? Evaluation of explainable artificial intelligence (xai) interpretability through neural backdoors. arXiv:2009.10639

Liu, H., Simonyan, K., & Yang, Y. (2018). Darts: Differentiable architecture search. arXiv:1806.09055

Looveren, A. V., & Janis, K. (2020). *Interpretable counterfactual explanations guided by prototypes*.

Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in neural information processing systems* (NeurIPS) (pp. 4765–4774).

Ma, N., Zhang, X., Zheng, H.-T., & Sun, J. (2018). Shufflenet v2: Practical guidelines for efficient cnn architecture design. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 116–131).

Nilsback, M.-E., &Zisserman, A. (2008). Automated flower classification over a large number of classes. In *Sixth Indian conference on computer vision, graphics & image processing* (pp. 722–729). IEEE.

Petsiuk, V., Das, A., & Saenko, K.(2018). Rise: Randomized input sampling for explanation of black-box models. In *Proceedings of the British machine vision conference (BMVC)*.

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

Redmon, J., & Farhadi, A. (2018). Yolov3: An incremental improvement. arXiv:1804.02767

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 1135–1144).

Ross, A., & Doshi-Velez, F. (2018). Improving the adversarial robustness and interpretability of deep neural networks by regularizing their input gradients. In *Proceedings of the AAAI conference on artificial intelligence*.

Samek, W., Binder, A., Montavon, G., Lapuschkin, S., & Müller, K.-R. (2016). Evaluating the visualization of what a deep neural network has learned. *IEEE Transactions on Neural Networks and Learning Systems, 28*(11), 2660–2673.

Sandler, M., Howard, A., Zhu, M., Zhmoginov, A., & Chen, L.-C. (2018). Mobilenetv2: Inverted residuals and linear bottlenecks. In *IEEE conference on computer vision and pattern recognition (CVPR)*.

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2020). Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision (IJCV), 128*(2), 336–359.

Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., & LeCun, Y. (2014). Overfeat: Integrated recognition, localization and detection using convolutional networks. In *International conference on learning representations (ICLR)*.

Shrikumar, A., Greenside, P., & Kundaje, A. (2017). Learning important features through propagating activation differences. In *International conference on machine learning (ICML)* (pp. 3145–3153).

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. In *International conference on learning representations (ICLR)*.

Smilkov, D., Thorat, N., Kim, B., Viégas, F., & Wattenberg, M. (2017). Smoothgrad: Removing noise by adding noise. In *ICML workshop on visualization for deep learning*.

Sundararajan, M., Taly, A., & Yan, Q. (2017). Axiomatic attribution for deep networks. In *International conference on machine learning (ICML)*.

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., & Rabinovich, A. (2015). Going deeper with convolutions. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 1–9).

Tan, M., & Le, Q. V. (2019). Efficientnet: Rethinking model scaling for convolutional neural networks. In *International conference on machine learning (ICML)*.

van der Linden, I., Haned, H., & Kanoulas, E. (2019). Global aggregations of local explanations for black box models. In *FACTS-IR: Fairness, accountability, confidentiality, transparency, and safety—SIGIR 2019 workshop*.

Vedaldi, A., & Soatto, S. (2008). Quick shift and kernel methods for mode seeking. In *Proceedings of the European conference on computer vision (ECCV)* (pp. 705–718). Springer.

Vu, M. N., Nguyen, T. D., Phan, N., Gera, R., & Thai, M. T. (2019). Evaluating explainers via perturbation. arXiv:1906.02032

Wang, H., Wang, Z., Du, M., Yang, F., Zhang, Z., Ding, S., Mardziel, P., & Hu, X. (2020). Score-cam: Score-weighted visual explanations for convolutional neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops* (pp. 24–25).

Wang, J., Sun, K., Cheng, T., Jiang, B., Deng, C., Zhao, Y., Liu, D., Mu, Y., Tan, M., Wang, X., Liu, W. (2020). Deep high-resolution representation learning for visual recognition.

Wang, Y., Zhang, J., Kan, M., Shan, S., & Chen, X. (2020). Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation. In *2020 IEEE/CVF conference on computer vision and pattern recognition, CVPR 2020*, Seattle, WA, USA, June 13–19, 2020.

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S., & Perona, P. (2010). *Caltech-UCSD birds 200*. Technical Report CNS-TR-2010-001, California Institute of Technology.

Xie, S., Girshick, R., Dollár, P., Tu, Z., & He, K. (2017). Aggregated residual transformations for deep neural networks. In *IEEE conference on computer vision and pattern recognition (CVPR).*

Yang, M., & Kim, B. (2019). Benchmarking attribution methods with relative feature importance. arXiv:1907.09701

Yeh, C.-K., Hsieh, C.-Y., Suggala, A. S., Inouye, D. I., & Ravikumar, P. (2019). On the (in) fidelity and sensitivity for explanations. In *Advances in Neural Information Processing Systems (NeurIPS).*

Yuan, T., Li, X., Xiong, H., Cao, H., & Dou, D. (2021). Explaining information flow inside vision transformers using markov chain. In *eXplainable AI approaches for debugging and diagnosis*.

Zhang, Q., Yang, Y., Ma, H., & Wu, Y. N. (2019). Interpreting cnns via decision trees. In *IEEE conference on computer vision and pattern recognition (CVPR)* (pp. 6261–6270).

Zhang, X., Zhou, X., Lin, M., & Sun, J. (2018). Shufflenet: An extremely efficient convolutional neural network for mobile devices. In *IEEE conference on computer vision and pattern recognition (CVPR)*

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

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Xuhong Li[1] · Haoyi Xiong[1] · Siyu Huang[2] · Shilei Ji[1] · Dejing Dou[1]

Xuhong Li
lixuhong@baidu.com

Siyu Huang
huangsiyutc@gmail.com

Shilei Ji
jishilei@baidu.com

Dejing Dou
doudejing@baidu.com

[1] Baidu Inc., Beijing, China

[2] Harvard University, Cambridge, MA, USA