Explaining Convolutional Neural Networks through Attribution-Based Input Sampling and Block-Wise Feature Aggregation

**Sam Sattarzadeh,[1] Mahesh Sudhakar,[1] Anthony Lem,[2]
Shervin Mehryar,[1] Konstantinos N Plataniotis,[1] Jongseong Jang,[3] Hyunwoo Kim,[3]
Yeonjeong Jeong,[3] Sangmin Lee,[3] Kyunghoon Bae[3]**

[1]The Edward S. Rogers Sr. Department of Electrical & Computer Engineering, University of Toronto
[2]Division of Engineering Science, University of Toronto
[3]LG AI Research
sam.sattarzadeh, mahesh.sudhakar@mail.utoronto.ca; j.jang, hwkim@lgresearch.ai

Abstract

As an emerging field in Machine Learning, Explainable AI (XAI) has been offering remarkable performance in interpreting the decisions made by Convolutional Neural Networks (CNNs). To achieve visual explanations for CNNs, methods based on class activation mapping and randomized input sampling have gained great popularity. However, the attribution methods based on these techniques provide low-resolution and blurry explanation maps that limit their explanation ability. To circumvent this issue, visualization based on various layers is sought. In this work, we collect visualization maps from multiple layers of the model based on an attribution-based input sampling technique and aggregate them to reach a fine-grained and complete explanation. We also propose a layer selection strategy that applies to the whole family of CNN-based models, based on which our extraction framework is applied to visualize the last layers of each convolutional block of the model. Moreover, we perform an empirical analysis of the efficacy of derived lower-level information to enhance the represented attributions. Comprehensive experiments conducted on shallow and deep models trained on natural and industrial datasets, using both ground-truth and model-truth based evaluation metrics validate our proposed algorithm by meeting or outperforming the state-of-the-art methods in terms of explanation ability and visual quality, demonstrating that our method shows stability regardless of the size of objects or instances to be explained.

Introduction

Deep Neural models based on Convolutional Neural Networks (CNNs) have rendered inspiring breakthroughs in a wide variety of computer vision tasks. However, the lack of interpretability hurdles the understanding of decisions made by these models. This diminishes the trust consumers have for CNNs and limits the interactions between users and systems established based on such models. Explainable AI (XAI) attempts to interpret these cumbersome models (Hoffman et al. 2018). The offered interpretation ability has put XAI in the center of attention in various fields, especially where any single false prediction can cause severe consequences (e.g., healthcare) or where regulations force automotive decision-making systems to provide users with explanations (e.g., criminal justice) (Lipton 2018).

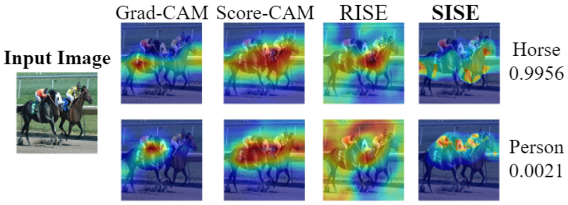

Figure 1: Comparison of conventional XAI methods with SISE (our proposed) to demonstrate SISE's ability to generate class discriminative explanations on a ResNet-50 model.

This work particularly addresses the problem of visual explainability, which is a branch of post-hoc XAI. This field aims to visualize the behavior of models trained for image recognition tasks (Barredo Arrieta et al. 2019). The outcome of these methods is a heatmap in the same size as the input image named "explanation map", representing the evidence leading the model to decide.

Prior works on visual explainable AI can be broadly categorized into 'approximation-based' (Ribeiro, Singh, and Guestrin 2016), 'backpropagation-based', 'perturbation-based', and 'CAM-based' methodologies. In backpropagation-based methods, only the local attributions are represented, making them unable to measure global sensitivity. This drawback is addressed by image perturbation techniques used in recent works such as RISE (Petsiuk, Das, and Saenko 2018), and Score-CAM (Wang et al. 2020). However, feedforwarding several perturbed images in these works makes them very slow. On the other hand, explanation maps produced by CAM-based methods suffer from a lack of spatial resolution as they are formed by combining the feature maps in the last convolutional layer of CNNs, which lack spatial information regarding the captured attributions.

In this work, we delve deeper into providing a solution for interpreting CNN-based models by analyzing multiple layers of the network. Our solution concentrates on mutual utilization of features represented inside a CNN in different semantic levels, achieving class discriminability and spatial resolution simultaneously. Inheriting productive ideas from the aforementioned types of approaches, we formulate a four-phase explanation method. In the first three phases, information extracted from multiple layers of the CNN is represented in their accompanying visualization maps. These maps are then combined via a fusion module to form a unique explanation map in the last phase. The main contributions of our work can be summarized as follows:

- We introduce a novel XAI algorithm that offers both spatial resolution and explanation completeness in its output explanation map by 1) using multiple layers from the "intermediate blocks" of the target CNN, 2) selecting crucial feature maps from the outputs of the layers, 3) employing an attribution-based technique for input sampling to visualize the perspective of each layer, and 4) applying a feature aggregation step to reach refined explanation maps.
- We propose a strategy to select the minimum number of intermediate layers from a given CNN to probe and visualize their discovered features in order to provide the local explanations of the whole CNN. We discuss the applicability of this strategy to all of the feedforward CNNs.
- We conduct thorough experiments on various models trained on object detection and industrial anomaly classification datasets. To justify our method, we employ various metrics to compare our proposed method with other conventional approaches. Therefore, we show that the information between layers can be correctly combined to improve its inference's visual explainability.

Related Work

Backpropagation-based methods In general, calculating the gradient of a model's output to the input features or the hidden neurons is the basis of this type of explanation algorithms. The earliest backpropagation-based methods operate by computing the model's confidence score's sensitivity to each of the input features directly (Simonyan, Vedaldi, and Zisserman 2014; Zeiler and Fergus 2014). To develop such methods, in some preceding works like DeepLift (Shrikumar, Greenside, and Kundaje 2017), IntegratedGradient (Sundararajan, Taly, and Yan 2017) and SmoothGrad (Smilkov et al. 2017), backpropagation-based equations are adapted to tackle the gradient issues. Some approaches such as LRP (Bach et al. 2015), SGLRP (Iwana, Kuroki, and Uchida 2019), and RAP (Nam et al. 2020) modify backpropagation rules to measure the relevance or irrelevance of the input features to the model's prediction. Moreover, FullGrad (Srinivas and Fleuret 2019) and Excitation Backpropagation (Zhang et al. 2018) run by aggregating gradient information from several layers of the network.

Perturbation-based methods Several visual explanation methods probe the model's behavior using perturbed copies of the input. In general, various strategies can be chosen to perform input sampling. Like RISE (Petsiuk, Das, and Saenko 2018), few of these approaches proposed random perturbation techniques to yield strong approximations of explanations. In Extremal Perturbation (Fong, Patrick, and Vedaldi 2019), an optimization problem is formulated to optimize a smooth perturbation mask maximizing the model's output confidence score. Most of the perturbation-based methods' noticeable property is that they treat the model like a "black-box" instead of a "white-box."

CAM-based methods Based on the Class Activation Mapping method (Zhou et al. 2016), an extensive research effort has been put to blend high-level features extracted by CNNs in a unique explanation map. CAM-based methods operate in three steps: 1) feeding the model with the input image, 2) scoring the feature maps in the last convolutional layer, and 3) combining the feature maps using the computed scores as weights. Grad-CAM (Selvaraju et al. 2017) and Grad-CAM++ (Chattopadhay et al. 2018) utilize backpropagation in the second step which causes underestimation of sensitivity information due to gradient issues. Ablation-CAM (Ramaswamy et al. 2020), Smooth Grad-CAM++ (Omeiza et al. 2019), and Score-CAM (Wang et al. 2020) have been developed to overcome these drawbacks.

Despite the strength of CAM-based methods in capturing the features extracted in CNNs, the lack of localization information in the coarse high-level feature maps limits such methods' performance by producing blurry explanations. Also, upsampling low-dimension feature maps to the size of input images distorts the location of captured features in some cases. Some recent works (Meng et al. 2019; Rebuffi et al. 2020) addressed these limitations by amalgamating visualization maps obtained from multiple layers to achieve a fair trade-off between spatial resolution and class-distinctiveness of the features forming explanation maps.

Methodology

Our proposed algorithm is motivated by methods aiming to interpret the model's prediction using input sampling techniques. These methods have shown a great faithfulness in rationally inferring the predictions of models. However, they suffer from instability as their output depends on random sampling (RISE) or random initialization for optimizing a perturbation mask (Extremal perturbation). Also, such algorithms require an excessive runtime to provide their users with generalized results. To address these limitations, we advance a CNN-specific algorithm that improves their fidelity and plausibility (in the view of reasoning) with adaptive computational overhead for practical usage. We term our algorithm as Semantic Input Sampling for Explanation (SISE). To claim such a reform, we replace the randomized input sampling technique in RISE with a sampling technique that relies on the feature maps derived from multiple layers. We call this procedure *attribution-based input sampling* and show that it provides the perspective of the model in various semantic levels, reducing the applicability of SISE to CNNs.

As sketched in Figs. 3 and 5, SISE consists of four phases. In the first phase, multiple layers of the model are selected, and a set of corresponding output feature maps are extracted. For each set of feature maps in the second phase, a subset containing the most important feature maps is sampled with a backward pass. The selected feature maps are then post-processed to create sets of perturbation masks to be utilized in the third phase for attribution-based input sampling and are termed as *attribution masks*. The first three phases are applied to multiple layers of the CNN to output a 2-dimensional saliency map named *visualization map* for each layer. Such obtained visualization maps are aggregated in the last phase to reach the final explanation map.

In the following section, we present a *block-wise* layer selection policy, showing that the richest knowledge in any CNN can be derived by probing the output of (the last layer in) each convolutional blocks, followed by the discussion of the phase-by-phase methodology of SISE.

Block-Wise Feature Explanation

As we attempt to visualize multiple layers of the CNNs to merge spatial information and semantic information discovered by the CNN-based model, we intend to define the most crucial layers for explicating the model's decisions to reach a complete understanding of the model by visualizing the minimum number of layers.

Regardless of the specification of their architecture, all types of CNNs consist of convolutional blocks connected via pooling layers that aid the network to justify the existence of semantic instances. Each convolutional block is formed by cascading multiple layers, which may vary from a simple convolutional filter to more complex structures (e.g., bottleneck or MBConv layers). However, the dimensions of their input and output signal are the same. In a convolutional block, assuming the number of layers to be L, each ith layer can be represented with the function $f_i(.)$, where $i = \{1, ..., L\}$. Denoting the input to each ith layer as y_i, the whole block can be mathematically described as $F(y_1) = f_L(y_L)$. For plain CNNs (e.g., VGG, GoogleNet), the output of each convolutional block can be represented with the equation below:

$$F(y_1) = f_L(f_{L-1}(...(f_1(y_1))) \quad (1)$$

After the emergence of residual networks that utilize skip-connection layers to propagate the signals through a convolutional block in the families as ResNet models, DenseNet, EfficientNet (Tan and Le 2019; Huang et al. 2017; Sandler et al. 2018), and the models whose architecture are adaptively learned (Zoph and Le 2016), it is debated that these neural networks can be represented with a more complicated view. These types of networks can be viewed by the unraveled perspective, as presented in (Veit, Wilber, and Belongie 2016). Based on this perspective as in Fig. 2, the connection between the input and output is formulated as follows:

$$y_{i+1} = f_i(y_i) + y_i \quad (2)$$

and hence,

$$F(y_1) = y_1 + f_1(y_1) + ... + f_L(y_1 + ... + f_{L-1}(y_{L-1})) \quad (3)$$

The unraveled architecture as in Fig. 2 is comprehensive enough to be generalized even to shallower CNN-based models that lack skip-connection layers. For plain networks, the layer functions f_i can be decomposed to an identity function I and a residual function g_i as follows:

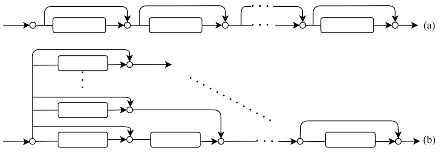

Figure 2: Architecture of the residual convolutional blocks as in (Shen, Ma, and Li 2018). (a) raveled schematic of a residual network, (b) unraveled view of the residual network.

$$f_i(y_i) = I(y_i) + g_i(y_i) \quad (4)$$

Such a decomposition, yields to a similar equation form as equation 2, and consequently, equation 3.

$$y_{i+1} = g_i(y_i) + y_i \quad (5)$$

It can be inferred from the unraveled view that while feeding the model with an input, signals might not pass through all convolutional layers as they may skip some layers and be propagated to the next ones directly. However, this is not the case for pooling layers. Considering they change the signals' dimensions, equation 4 cannot be applied to such layers. To prove this hypothesis, an experiment was conducted in (Veit, Wilber, and Belongie 2016), where the corresponding test errors are reported for removing a layer individually from a residual network. It was observed that a significant degradation in test performance is recorded only when the pooling layers are removed.

Based on this hypothesis and result, most of the information in each model can be collected by probing the pooling layers. Thus, by visualizing these layers, it is possible to track the way features are propagated through convolutional blocks. Therefore, we derive attribution masks from the feature maps in the last layers of all of their convolutional blocks for any given CNN. Then, for each of these layers, we build a corresponding visualization map. These maps are utilized to perform a *block-wise feature aggregation* in the last phase of our method.

Feature Map Selection

As discussed, the first two phases of SISE take responsibility to create multiple sets of attribution masks. In the first phase, we feed the model with an input image to derive sets of feature maps from various layers of the model. Then, we sample the most deterministic feature maps among each set and post-process them to obtain corresponding sets of attribution masks. These masks are utilized for performing attribution-based input sampling.

Assume $\Psi : \mathcal{I} \rightarrow \mathbb{R}$ be a trained model that outputs a confidence score for a given input image, where $\mathcal{I}$ is the space of RGB images $\mathcal{I} = \{I | I : \Lambda \rightarrow \mathbb{R}^3\}$, and $\Lambda = \{1, ..., H\} \times \{1, ..., W\}$ is the set of locations (pixels) in the image. Given any model and image, the goal of an explanation algorithm is to reach an explanation map $S_{I,\Psi}(\lambda)$, that assigns an "importance value" to each location in the image ($\lambda \in \Lambda$). Also, let l be a layer containing N feature

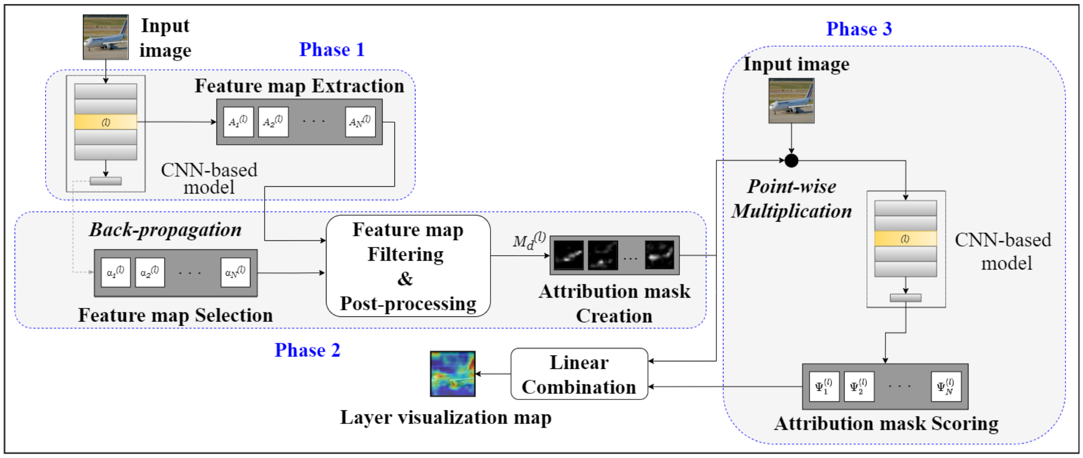
Input image
Phase 1
Feature map Extraction
$A_1^{(l)}$ $A_2^{(l)}$ · · · $A_N^{(l)}$
(l)
CNN-based model
Back-propagation
$\alpha_1^{(l)}$ $\alpha_2^{(l)}$ · · · $\alpha_N^{(l)}$
Feature map Selection
Feature map Filtering & Post-processing
$M_d^{(l)}$
Attribution mask Creation
Phase 2
Phase 3
Input image
Point-wise Multiplication
CNN-based model
$\Psi_1^{(l)}$ $\Psi_2^{(l)}$ · · · $\Psi_N^{(l)}$
Attribution mask Scoring
Linear Combination
Layer visualization map

Figure 3: Schematic of SISE's layer visualization framework (first three phases). The procedure in this framework is applied to multiple layers and is followed by the fusion framework (as in Fig. 5).

maps represented as $A_k^{(l)} (k = \{1, ..., N\})$ and the space of locations in these feature maps be denoted as $\Lambda^{(l)}$. These feature maps are collected by probing the feature extractor units of the model, and a similar strategy is also utilized in (Wang et al. 2020). The feature maps are formed in these units independently from the classifier part of the model. Thus, using the whole set of feature maps does not reflect the outlook of CNN's classifier.

To identify and reject the class-indiscriminative feature maps, we partially backpropagate the signal to the layer l to score the average gradient of model's confidence score to each of the feature maps. These average gradient scores are represented as follows:

$$\alpha_k^{(l)} = \sum_{\lambda^{(l)} \in \Lambda^{(l)}} \frac{\partial \Psi(I)}{\partial A_k^{(l)}(\lambda^{(l)})} \tag{6}$$

The feature maps with corresponding non-positive average gradient scores - $\alpha_k^{(l)}$, tend to contain features related to other classes rather than the class of interest. Terming such feature maps as 'negative-gradient', we define the set of attribution masks obtained from the 'positive-gradient' feature maps, $M_d^{(l)}$, as:

$$M_d^{(l)} = \{\Omega(A_k^{(l)}) | k \in \{1, ..., N\}, \alpha_k^{(l)} > \mu \times \beta^{(l)}\} \tag{7}$$

where $\beta^{(l)}$ denotes the maximum average gradient recorded.

$$\beta^{(l)} = \max_{k \in \{1,...,N\}} (\alpha_k^{(l)}) \tag{8}$$

In equation 7, $\mu \in \mathbb{R}_{\geq 0}$ is a threshold parameter that is 0 by default to discard negative-gradient feature maps while retaining only the positive-gradients. Furthermore, $\Omega(.)$ represents a post-processing function that converts feature maps to attribution masks. This function contains a 'bilinear interpolation,' upsampling the feature maps to the size of the input image, followed by a linear transformation that normalizes the values in the mask in the range $[0, 1]$. A visual comparison of attribution masks and random masks in Fig. 4 emphasizes such advantages of the former.

Attribution-Based Input Sampling

Considering the same notations as the previous section, and according to RISE method, the confidence scores observed for the copies of an image masked with a set of binary masks $(M : \Lambda \rightarrow \{0, 1\})$ are used to form the explanation map by,

$$S_{I,\Psi}(\lambda) = \mathbb{E}_M[\Psi(I \odot m) | m(\lambda) = 1] \tag{9}$$

where $I \odot m$ denotes a masked image obtained by pointwise multiplication between the input image and a mask $m \in M$. The representation of equation 9 can be modified to be generalized for sets of smooth masks $(M : \Lambda \rightarrow [0, 1])$. Hence, we reformat equation 9 as:

$$S_{I,\Psi}(\lambda) = \mathbb{E}_M[\Psi(I \odot m) \cdot C_m(\lambda)] \tag{10}$$

where the term $C_m(\lambda)$ indicates the contribution amount of each pixel in the masked image. Setting the contribution indicator as $C_m(\lambda) = m(\lambda)$, makes equation 10 equivalent to equation 9. We normalize these scores according to the size of perturbation masks to decrease the assigned reward to the background pixels when a high score is reached for a mask with too many activated pixels. Thus, we define this term as:

$$C_m(\lambda) = \frac{m(\lambda)}{\sum_{\lambda \in \Lambda} m(\lambda)} \tag{11}$$

Such a formulation increases the concentration on smaller features, particularly when multiple objects (either from the same instance or different ones) are present in an image.

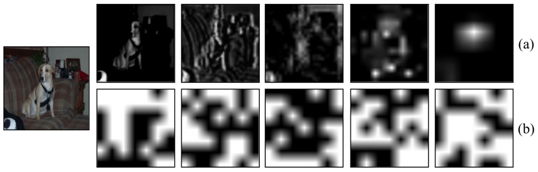

Figure 4: Qualitative comparison of (a) attribution masks derived from different blocks of a VGG16 network as in SISE, with (b) random masks employed in RISE.

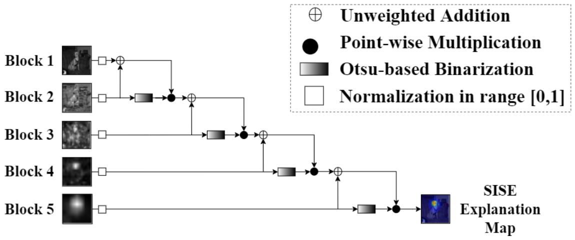

Figure 5: SISE fusion module for a CNN with 5 convolutional blocks.

Putting block-wise layer selection policy and attribution mask selection strategy together with the modified RISE framework, for each CNN containing B convolutional blocks, the last layer of each block is indicated as $l_b \in \{1, ..., B\}$. Using equations 10 and 11, we form corresponding visualization maps for each of these layers by:

$$V^{(l_b)}_{I,\Psi}(\lambda) = \mathbb{E}_{M_d^{(l_b)}}[\Psi(I \odot m) \cdot C_m(\lambda)] \qquad (12)$$

Fusion Module

In the fourth phase of SISE, the flow of features from low-level to high-level blocks are tracked. The inputs to the fusion module are the visualization layers obtained from the third phase of SISE. On the other hand, this module's output is a 2-dimensional explanation map, which is the output of SISE. The fusion block is responsible for correcting spatial distortions caused by upsampling coarse feature maps to higher dimensions and refining the localization of attributions derived from the model.

Our fusion module is designed with cascaded fusion blocks. In each block, the feature information from the visualization maps representing explanations for two consecutive blocks is collected using an "addition" block. Then, the features that are absent in the latter visualization map are removed from the collective information by masking the output of the addition block with a binary mask indicating the activated regions in the latter visualization map. To reach the binary mask, we apply an adaptive threshold to the latter visualization map, determined by Otsu's method (Otsu 1979). By cascading fusion blocks as in Fig. 5, the features determining the model's prediction are represented in a more fine-grained manner while the inexplicit features are discarded.

Experiments

We verify our method's performance on shallow and deep CNNs, including VGG16, ResNet-50, and ResNet-101 architectures. To conduct the experiments, we employed PASCAL VOC 2007[1] (Everingham et al. 2010) and Severstal [2] datasets. The former is a popular object detection dataset containing 4952 test images belonging to 20 object classes. As images with many small object occurrences and multiple instances of different classes are prevalent in this dataset, it is hard for an XAI algorithm to perform well on the whole dataset. The latter is an industrial steel defect detection dataset created for anomaly detection and steel defect segmentation problems. We reformatted it into a defect classification dataset instead, containing 11505 test images from 5 different classes, including one normal class and four different defects classes. Class imbalance, intraclass variation, and interclass similarity are the main challenges of this recast dataset.

Experimental Setup

Experiments conducted on the PASCAL VOC 2007 dataset[3] are evaluated on its test set with a VGG16, and a ResNet-50 model from the TorchRay library (Fong, Patrick, and Vedaldi 2019), trained by (Zhang et al. 2018), both trained for multi-label image classification. The top-5 accuracies of the models on the test set are 93.29% and 93.09%, respectively. On the other hand, for conducting experiments on Severstal, we trained a ResNet-101 model (with a test accuracy of 86.58%) on the recast dataset to assess the performance of the proposed method in the task of visual defect inspection. To recast the Severstal dataset for classification, the train and test images were cropped into patches of size 256×256. In our evaluations, a balanced subset of 1381 test images belonging to defect classes labeled as 1, 2, 3, and 4 is chosen. We have implemented SISE on Keras and set the parameter μ to its default value, 0.

Qualitative Results

Based on explanation quality, we have compared SISE with other state-of-the-art methods on sample images from the Pascal dataset in Fig. 6 and Severstal dataset in Fig. 8. Images with both normal-sized and small object instances are shown along with their corresponding confidence scores. Moreover, Figs. 1 and 7 with images of multiple objects from different classes depict the superior ability of SISE in discriminating the explanations of various classes in comparison with other methods and RISE in particular.

Quantitative Results

Quantitative analysis includes evaluation results categorized into 'ground truth-based' and 'model truth-based' metrics. The former is used to justify the model by assessing the extent to which the algorithm satisfies the users by providing visually superior explanations, while the latter is used to analyze the model behavior by assessing the faithfulness of the algorithm and its correctness in capturing the attributions in line with the model's prediction procedure. The reported results of RISE and Extremal Perturbation in Table 1 are averaged on three runs. The utilized metrics are discussed below.

Ground truth-based Metrics: The state-of-the-art explanation algorithms are compared with SISE based on three distinct ground-truth based metrics to justify the visual quality of the explanation maps generated by our method. Denoting the ground-truth mask as G and the achieved explanation map as S, the evaluation metrics used are:

[1] http://host.robots.ox.ac.uk/pascal/VOC/voc2007

[2] https://www.kaggle.com/c/severstal-steel-defect-detection.

[3] Last date accessed: 20 August 2020

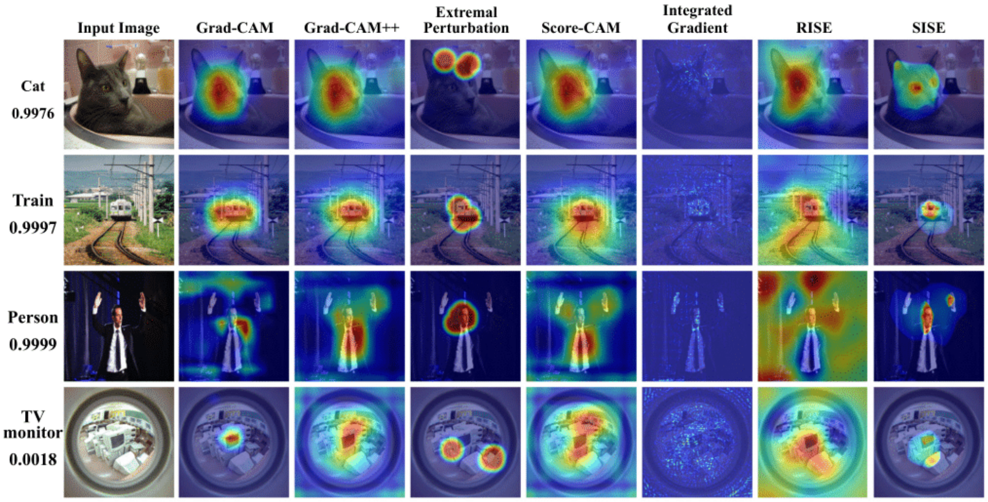

Figure 6: Qualitative comparison of the state-of-the-art XAI methods with our proposed SISE for test images from the PASCAL VOC 2007 dataset. The first two rows are the results from a ResNet-50 model, and the last two are from a VGG16 model.

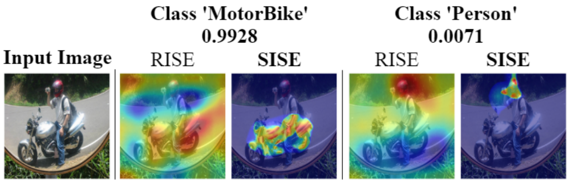

Figure 7: Class discriminative ability of SISE vs. RISE obtained from a VGG16 model

Energy-Based Pointing Game (EBPG) evaluates the precision and denoising ability of XAI algorithms (Wang et al. 2020). Extending the traditional Pointing Game, EBPG considers all pixels in the resultant explanation map S for evaluation by measuring the fraction of its energy captured in the corresponding ground truth G, as $EBPG = \frac{||S \odot G||_1}{||S||_1}$.

mIoU analyses the localization ability and meaningfulness of the attributions captured in an explanation map. In our experiments, we select the top 20% pixels highlighted in each explanation map S and compute the mean intersection over union with their corresponding ground-truth masks.

Bounding box (Bbox) (Schulz et al. 2020) is taken into account as a size-adaptive variant of mIoU. Considering N as the number of ground truth pixels in G, the Bbox score is calculated by selecting the top N pixels in S and evaluating the corresponding fraction captured over G.

Model truth-based metrics: To evaluate the correlation between the representations of our method and the model's predictions, model-truth based metrics are employed to compare SISE with the other state-of-the-art methods. As visual explanation algorithms' main objective is to envision the model's perspective for its predictions, these metrics are considered of higher importance.

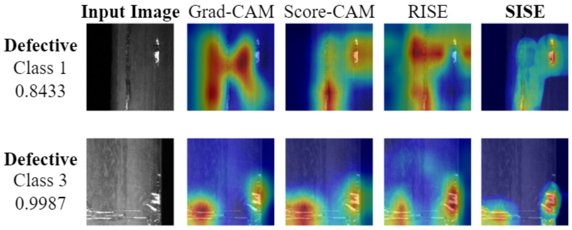

Figure 8: Qualitative comparison of explanation maps by a ResNet-101 model on test images from Severstal dataset.

Drop% and Increase%, as introduced in (Chattopadhay et al. 2018) and later modified by (Ramaswamy et al. 2020; Fu et al. 2020), can be interpreted as an indicator of the positive attributions missed and the negative attribution discarded from the explanation map respectively. Given a model $\Psi(.)$, an input image I_i from a dataset containing K images, and an explanation map $S(I_i)$, the Drop/Increase % metric selects the most important pixels in $S(I_i)$ to measure their contribution towards the model's prediction. A threshold function $T(.)$ is applied on $S(I_i)$ to select the top 15% pixels that are then extracted from I_i using pointwise multiplication and fed to the model. The confidence scores on such perturbed images are then compared with the original score, according to the equations $Drop\% = \frac{1}{K}\sum_{i=1}^{K}\frac{\max(0,\Psi(I_i)-\Psi(I_i \odot T(I_i)))}{\Psi(I_i)} \times 100$ and $Increase\% = \sum_{i=1}^{K} sign(\Psi(I_i \odot T(I_i)) - \Psi(I_i))$.

Discussion

The experimental results in Figs. 1, 6, 7, and 8 demonstrate the resolution, and concreteness of SISE explanation maps, which is further supported by justifying our method via ground truth-based evaluation metrics as in Table 1. Also,

Model	Metric	Grad-CAM	Grad-CAM++	Extremal Perturbation	RISE	Score-CAM	Integrated Gradient	FullGrad	SISE
VGG16	**EBPG**	55.44	46.29	**61.19**	33.44	46.42	36.87	38.72	60.54
	mIoU	26.52	**28.1**	25.44	27.11	27.71	14.11	26.61	27.79
	Bbox	51.7	55.59	51.2	54.59	54.98	33.97	54.17	**55.68**
	Drop	49.47	60.63	43.90	39.62	39.79	64.74	60.78	**38.40**
	Increase	31.08	23.89	32.65	37.76	36.42	26.17	22.73	**37.96**
ResNet-50	**EBPG**	60.08	47.78	63.24	32.86	35.56	40.62	39.55	**66.08**
	mIoU	**32.16**	30.16	26.29	27.4	31.0	15.41	20.2	31.37
	Bbox	60.25	58.66	52.34	55.55	60.02	34.79	44.94	**61.59**
	Drop	35.80	41.77	39.38	39.77	35.36	66.12	65.99	**30.92**
	Increase	36.58	32.15	34.27	37.08	37.08	24.24	25.36	**40.22**

Table 1: Results of ground truth-based and model truth-based metrics for state-of-the-art XAI methods along with SISE (proposed) on two networks trained on the PASCAL VOC 2007 dataset. For each metric, the best is shown in bold, and the second-best is underlined. Except for Drop%, the higher is better for all other metrics. All values are reported in percentage.

XAI method	Drop%	Increase%
Grad-CAM	67.44	12.46
Grad-CAM++	64.1	12.96
RISE	63.25	15.63
Score-CAM	64.29	10.35
FullGrad	77.23	10.26
SISE	**61.06**	**15.64**

Table 2: Results of model truth-based metrics of SISE and state-of-the-art algorithms on a ResNet-101 model trained on Severstal data set.

model truth-based metrics in Tables 1 and 2 prove SISE's supremacy in highlighting the evidence, based on which the model makes a prediction. Similar to the CAM-based methods, the output of the last convolutional block plays the most critical role in our method. However, by considering the intermediate layers based on the block-wise layer selection, SISE's advantageous properties are enhanced. Furthermore, utilizing attribution-based input sampling instead of a randomized sampling, ignoring the unrelated feature maps, and modifying the linear combination step dramatically improves the visual clarity and completeness offered by SISE.

Complexity Evaluation In addition to performance evaluations, a runtime test is carried out to compare the complexity of the methods, using a Tesla T4 GPU with 16GB of memory and the ResNet-50 model. Reported runtimes were averaged over 100 trials using random images from the PASCAL VOC 2007 test set. Grad-CAM and Grad-CAM++ achieved the best runtimes, 19 and 20 milliseconds, respectively. On the other hand, Extremal Perturbation recorded the longest runtime, 78.37 seconds, since it optimizes numerous variables. In comparison with RISE, which has a runtime of 26.08 seconds, SISE runs in 9.21 seconds.

Ablation Study While RISE uses around 8000 random masks to operate on a ResNet-50 model, SISE uses around 1900 attribution masks with μ set to 0, out of a total of 3904 feature maps initially extracted from the same ResNet-50 model before negative-gradient feature maps were removed. The difference in the number of masks allows SISE to operate in around 9.21 seconds. To analyze the effect of reducing the number of attribution masks on SISE's performance, an ablation study is carried. By changing μ to 0.3, a scanty variation in the boundary of explanation maps can be noticed while the runtime is reduced to 2.18 seconds. This shows that ignoring feature maps with low gradient values does not considerably affect SISE outputs since they tend to be assigned low scores in the third phase of SISE anyway. By increasing μ to 0.5, a slight decline in the performance is recorded along with a runtime of just 0.65 seconds. A more detailed analysis of the effect of μ on various evaluation metrics along with an extensive discussion of our algorithm and additional results on MS COCO 2014 dataset (Lin et al. 2014) are provided in the technical appendix of our extended version on arXiv[4].

Conclusion

In this work, we propose SISE - a novel visual explanation algorithm specialized to the family of CNN-based models. SISE generates explanations by aggregating visualization maps obtained from the output of convolutional blocks through attribution-based input sampling. Qualitative results show that our method can output high-resolution explanation maps, the quality of which is emphasized by quantitative analysis using ground truth-based metrics. Moreover, model truth-based metrics demonstrate that our method also outperforms other state-of-the-art methods in providing concrete explanations. Our experiments reveal that mutual utilization of features captured in final and intermediate layers of the model aids in producing explanation maps that accurately locate object instances and reach a greater portion of attributions leading the model to make a decision.

[4] https://arxiv.org/abs/2010.00672

Acknowledgements

This research was supported by LG AI Research. The authors thank all anonymous reviewers for their detailed suggestions and critical comments on the original manuscript that substantially helped to improve the clarity of this paper.

References

Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10(7): e0130140.

Barredo Arrieta, A.; Diaz Rodriguez, N.; Del Ser, J.; Bennetot, A.; Tabik, S.; Barbado González, A.; Garcia, S.; Gil-Lopez, S.; Molina, D.; Benjamins, V. R.; Chatila, R.; and Herrera, F. 2019. Explainable Artificial Intelligence (XAI): Concepts, Taxonomies, Opportunities and Challenges toward Responsible AI. *Information Fusion* doi:10.1016/j.inffus.2019.12.012.

Chattopadhay, A.; Sarkar, A.; Howlader, P.; and 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)*, 839–847. doi:10.1109/WACV.2018.00097.

Everingham, M.; Gool, L.; Williams, C. K.; Winn, J.; and Zisserman, A. 2010. The Pascal Visual Object Classes (VOC) Challenge. *International Journal of Computer Vision* 88(2): 303–338. doi:10.1007/s11263-009-0275-4.

Fong, R.; Patrick, M.; and Vedaldi, A. 2019. Understanding deep networks via extremal perturbations and smooth masks. In *Proceedings of the IEEE International Conference on Computer Vision*, 2950–2958.

Fu, R.; Hu, Q.; Dong, X.; Guo, Y.; Gao, Y.; and Li, B. 2020. Axiom-based Grad-CAM: Towards Accurate Visualization and Explanation of CNNs. In *British Machine Vision Conference*.

Hoffman, R. R.; Mueller, S. T.; Klein, G.; and Litman, J. 2018. Metrics for Explainable AI: Challenges and Prospects. *CoRR* abs/1812.04608. URL http://arxiv.org/abs/1812.04608.

Huang, G.; Liu, Z.; Van Der Maaten, L.; and Weinberger, K. Q. 2017. Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4700–4708.

Iwana, B. K.; Kuroki, R.; and Uchida, S. 2019. Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 4176–4185. IEEE.

Lin, T.-Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; and Zitnick, C. L. 2014. Microsoft coco: Common objects in context. In *European conference on computer vision*, 740–755. Springer.

Lipton, Z. C. 2018. The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability is Both Important and Slippery. *Queue* 16(3): 31–57. ISSN 1542-7730. doi:10.1145/3236386.3241340. URL https://doi.org/10.1145/3236386.3241340.

Meng, F.; Huang, K.; Li, H.; and Wu, Q. 2019. Class Activation Map Generation by Representative Class Selection and Multi-Layer Feature Fusion. *arXiv preprint arXiv:1901.07683* .

Nam, W.-J.; Gur, S.; Choi, J.; Wolf, L.; and Lee, S.-W. 2020. Relative Attributing Propagation: Interpreting the Comparative Contributions of Individual Units in Deep Neural Networks. In *AAAI*, 2501–2508.

Omeiza, D.; Speakman, S.; Cintas, C.; and Weldermariam, K. 2019. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. *arXiv preprint arXiv:1908.01224* .

Otsu, N. 1979. A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics* 9(1): 62–66.

Petsiuk, V.; Das, A.; and Saenko, K. 2018. RISE: Randomized Input Sampling for Explanation of Black-box Models. In *Proceedings of the British Machine Vision Conference (BMVC)*.

Ramaswamy, H. G.; et al. 2020. Ablation-CAM: Visual Explanations for Deep Convolutional Network via Gradient-free Localization. In *The IEEE Winter Conference on Applications of Computer Vision*, 983–991.

Rebuffi, S.-A.; Fong, R.; Ji, X.; and Vedaldi, A. 2020. There and Back Again: Revisiting Backpropagation Saliency Methods. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8839–8848.

Ribeiro, M. T.; Singh, S.; and 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, San Francisco, CA, USA, August 13-17, 2016*, 1135–1144.

Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; and Chen, L.-C. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 4510–4520.

Schulz, K.; Sixt, L.; Tombari, F.; and Landgraf, T. 2020. Restricting the Flow: Information Bottlenecks for Attribution. In *International Conference on Learning Representations*. URL https://openreview.net/forum?id=S1xWh1rYwB.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations From Deep Networks via Gradient-Based Localization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*.

Shen, L.; Ma, Q.; and Li, S. 2018. End-to-end time series imputation via residual short paths. In *Asian Conference on Machine Learning*, 248–263.

Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning Important Features Through Propagating Activation Differences. In Precup, D.; and Teh, Y. W., eds., *Proceedings* *of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 3145–3153. International Convention Centre, Sydney, Australia: PMLR. URL http://proceedings.mlr.press/v70/shrikumar17a.html.

Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. In *Workshop at International Conference on Learning Representations*.

Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: Removing noise by adding noise. arXiv 2017. *arXiv preprint arXiv:1706.03825* .

Srinivas, S.; and Fleuret, F. 2019. Full-gradient representation for neural network visualization. In *Advances in Neural Information Processing Systems*, 4126–4135.

Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 3319–3328. JMLR. org.

Tan, M.; and Le, Q. V. 2019. EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks. *CoRR* abs/1905.11946. URL http://arxiv.org/abs/1905.11946.

Veit, A.; Wilber, M. J.; and Belongie, S. 2016. Residual networks behave like ensembles of relatively shallow networks. In *Advances in neural information processing systems*, 550–558.

Wang, H.; Wang, Z.; Du, M.; Yang, F.; Zhang, Z.; Ding, S.; Mardziel, P.; and 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*, 24–25.

Zeiler, M. D.; and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *European conference on computer vision*, 818–833. Springer.

Zhang, J.; Bargal, S. A.; Lin, Z.; Brandt, J.; Shen, X.; and Sclaroff, S. 2018. Top-Down Neural Attention by Excitation Backprop. *Int. J. Comput. Vision* 126(10): 1084–1102. ISSN 0920-5691. doi:10.1007/s11263-017-1059-x. URL https://doi.org/10.1007/s11263-017-1059-x.

Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2921–2929.

Zoph, B.; and Le, Q. V. 2016. Neural architecture search with reinforcement learning. *arXiv preprint arXiv:1611.01578* .