Ph.D. DISSERTATION

Inverse-Based Approach to Explaining and Visualizing Convolutional Neural Networks

역연산에 기반한 합성곱신경망의 설명 및 시각화

BY

KWON, Hyuk Jin

AUGUST 2021

DEPARTMENT OF ELECTRICAL AND COMPUTER
ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Inverse-Based Approach to Explaining and Visualizing Convolutional Neural Networks

역연산에 기반한 합성곱신경망의 설명 및 시각화

지도교수 조남익

이 논문을 공학박사 학위논문으로 제출함

2021년 6월

서울대학교 대학원

전기·정보공학부

권혁진

권혁진의 공학박사 학위논문을 인준함

2021년 6월

위원장 ______________________

부위원장 ______________________

위원 ______________________

위원 ______________________

위원 ______________________

Abstract

Interpretability and explainability of machine learning systems have received ever-increasing attention, especially for convolutional neural networks (CNN). Although there are various interpretation techniques for learning algorithms, post-hoc local explanation methods (e.g., the attribution method that visualizes pixel-level contribution of input to its corresponding result) are under great interest because they can deal with the high dimensional parameters and nonlinear operations of CNNs. Therefore, this dissertation presents three new post-hoc local explanation methods to visualize and understand the working mechanisms of CNNs.

At first, this dissertation presents a new method called guided nonlinearity (GNL) that improves the performance of attribution by backpropagating only positive gradients through nonlinear operations. GNL is inspired by the mechanism of action potential (AP) generation in the postsynaptic neuron that depends on the sum of excitatory (EPSP) and inhibitory postsynaptic potentials (IPSP). This dissertation assumes that paths consisting of excitatory synapses faithfully reflect the contributions of inputs to the output. Then this assumption is applied to CNNs by allowing only positive gradients backpropagate through nonlinear operations. Experimental results have shown that GNL outperforms existing methods for computing attributions in terms of the deletion metrics and yields fine-grained and human-interpretable attributions.

However, the attributions from existing methods, including GNL, lack a common theoretical background and sometimes give contradicting results. To address this problem, this dissertation develops the operation-wise inverse method that computes the inverse of prediction in an operation-wise manner by considering that CNNs can be decomposed with four fundamental operations (convolution, max-pooling, ReLU, and fully-connected). The operation-wise inverse process assumes that the forward-pass of CNN is a sequential propagation of physical quantities that indicate the magnitude of specific image features. The inverses of fundamental operations are formulated as constrained optimization problems that inverse results should generate output features consistent with the forward-pass. Then, the inverse of prediction is computed by sequentially applying inverses of fundamental operations of CNN. Experimental results show that the proposed operation-wise approach can be a reference tool for computing attributions because it can provide equivalent visualization results to several conventional methods, and the attributions from the operation-wise method achieve state-of-the-art performances in terms of deletion score.

Although the operation-wise method can provide a reference framework to compute attributions, applying the attribution concept to CNNs with multiple-valued predictions has not yet been addressed because the computation of attribution requires a single scalar value represents the prediction. To address this problem, this dissertation proposes the layer-wise inverse-based approach by decomposing CNNs into a set of layers that process only positive values that can be interpreted as neural activations. Then, the inverses of layers are formulated as constrained optimization problems that identify activations-of-interest in lower-layers. Then, the inverse of prediction is computed by sequentially applying inverses of layers of CNN as in the operation-wise method. Experimental results show that the proposed layer-wise inverse-based method can analyze CNNs for classification and regression in the same framework. Especially for the case of regression, the layer-wise approach showed that conventional CNNs for single image super-resolution overlook a portion of frequency bands that may result in performance degradation.

keywords: Convolutional neural networks, interpretable machine learning, post-hoc local explanation, attribution, inverse approach, image classification, image super-resolution

student number: 2017-37754

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Recently-developed deep CNN shows state-of-the-art performances on various computer vision tasks [1], and the analysis of its high performance becomes an important topic in the field of interpretable machine learning [2]. In general, the characteristics of interpretable machine learning techniques can be categorized as global interpretability and local interpretability. [3].

Global interpretability means that users can globally understand the working mechanism of a machine learning model by describing the structures and parameters of a model (e.g., decision boundary). In contrast, the methods for local interpretability analyze an individual prediction of a given model and try to provide a reason for the decision. In general, it is practically impossible to give global interpretation to CNNs because of their high dimensional parameters [4,5].

The interpretable machine learning algorithms also can be classified into intrinsic interpretability and post-hoc interpretability depending on whether the interpretability is obtained with the design of algorithms or not [3]. Therefore, intrinsic interpretability is usually achieved by constructing self-explanatory models, whereas the post-hoc approach applies external interpretation algorithms to existing models. For CNNs, post-hoc interpretation methods are widely used because of the nonlinear operations of CNNs that make it hard to build self-explanatory models. [6,7].

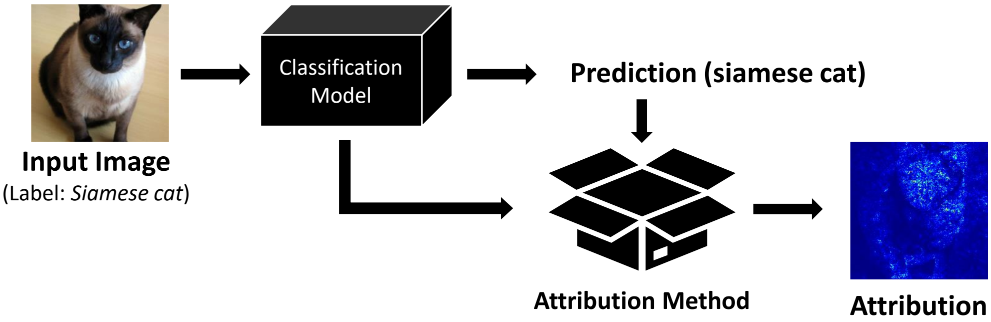


Figure 1.1: The overview of computing attributions.

Hence, many post-hoc local interpretation methods have been developed to understand and explain the inner workings of CNNs. Most of the existing post-hoc local methods target to compute the attribution that a heatmap showing the contribution of each pixel to the prediction as shown in Fig. 1.1 [8–14] .

1.1 Guided Nonlinearity

There has been a variety of methods to compute the attributions of CNNs for classification problems. Although the existing methods have their own advantages, attributions from conventional methods show blurred or perturbed heatmaps. It is hard to interpret the predictions with blurred or perburbed attributions when fine structures in input images have major roles [10]. To overcome this limitation, this dissertation suggests a method called guided nonlinearity (GNL) that computes fine-grained attributions by backpropagating only positive gradients through nonlinear units as shown in Fig 1.2.

Specifically, GNL modifies the backpropagation path of the integrated gradient [11] by restricting only positive valued gradients to be backpropagated for nonlinear operations. The design of GNL is inspired by the generation process of action potentials (AP) in the human visual system that depends on the sum of excitatory postsynaptic potentials (EPSP), which increase the likelihood of AP generation, and inhibitory postsynaptic potentials (IPSP), which decrease the likelihood of AP generation [15–17]. Generally, the computation of attributions can be defined as the pro-

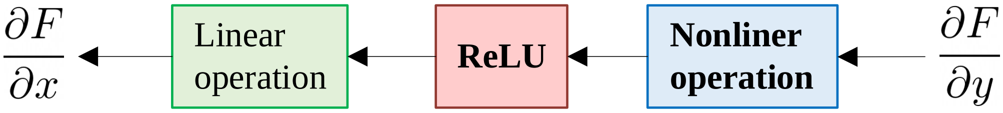

$\frac{\partial F}{\partial x}$
Linear operation
ReLU
Nonliner operation
$\frac{\partial F}{\partial y}$

Figure 1.2: The overview of the proposed guided-nonlinearity method.

cess of evaluating the contribution of pixels to the current output [18] and the outputs of nonlinear operators in neural networks could be considered as the computational modeling of neurotransmitters in the synaptic clefts [15–17]. Based on the above arguments, this dissertation make the following assumption that attributions can be enhanced by focusing on nonlinear hidden units with positive gradient because they can be interpreted as excitatory synapses that increase the likelihood of AP.

Experimental results have demonstrated that GNL yields fine-grained attributions with enhanced visual quality, and the attributions computed by GNL achieve state-of-the-art performance in terms of deletion metric [18]. These results also imply that the classification of internal signals of CNNs, whether they contribute to the prediction or not, could play an essential role in the understanding of the working mechanisms of CNNs.

1.2 Inverse-based approach

A variety of methods, including the proposed GNL, have been developed to compute attributions of CNNs. However, the existing methods for computing attributions lack a common theoretical background and yield attributions having different characteristics. Therefore, attributions from different methods sometimes show contradicting results that make users confused [9–11, 18].

Also, the conventional methods for computing attributions have been usually applied to CNNs for the classification tasks because a single scalar-valued output representing the prediction is necessary in computing attributions [9, 13, 14, 19]. In contrast to this, applying the attribution method to CNNs with multiple-valued outputs (e.g.,

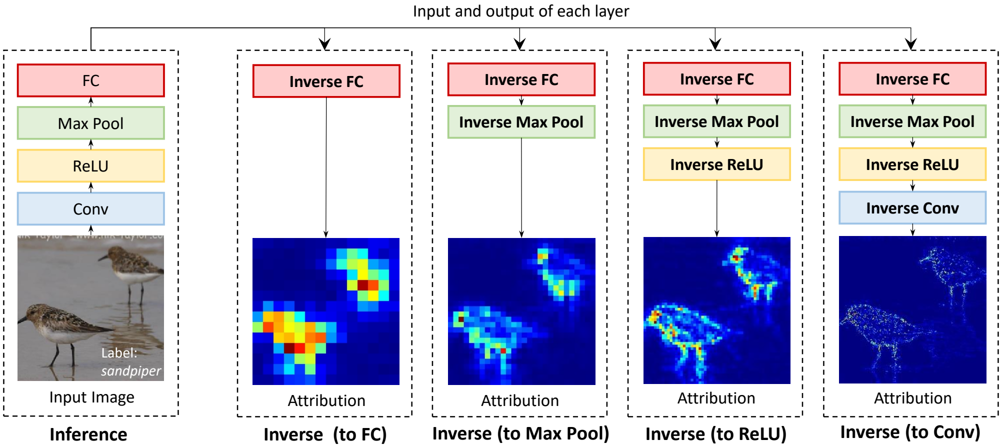


Figure 1.3: The overview of the proposed operation-wise inverse-based method. The inverses of convolution, max-pooling, ReLU, and fully connected (FC) operations are used to inverse the prediction of CNN in a operation-wise manner.

pixel-wise regression) has not been introduced because it is unclear which value could be used for the representative of given prediction if a CNN predicts multiple-valued outputs [20].

For the conventional computer vision algorithms with multiple-valued outputs (e.g., HOG [21], SIFT [22], and small multilayer perceptrons), the inverses of predictions have been tried to visualize features that respond to the algorithms in the input image [23]. However, the high dimensional parameters and nonlinear operations of CNNs make it hard to apply the inverse-based approach to understanding the working mechanisms of CNNs [24].

1.2.1 Operation-wise method

This dissertation proposes an operation-wise inverse-based approach that can interpret the predictions of CNNs in a unified framework to resolve the confusion raised by attributions of conventional methods as shown in Fig. 1.3.

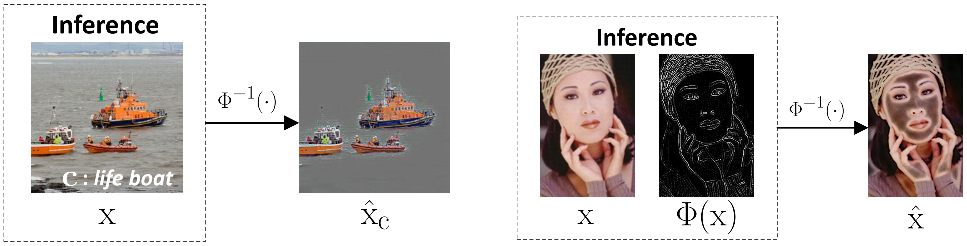
The operation-wise inverse-based method views CNNs as the composition of 4 fundamental operations (convolution, max-pooling, ReLU, and fully-connected) that have a reduced number of parameters than original CNNs. Then, the proposed method computes the inverse of prediction in an operation-wise manner based on the following postulations: (1) forward-pass is a sequential propagation of physical quantities that indicate the magnitudes of specific image features, (2) it is important to find signals that actually contribute to the predictions, (3) a small amount of physical quantities is desirable for human interpretability, and (4) it is possible to approximate the forward-passes of fundamental operations with linear functions around their operating points.

The inverse of each fundamental operation is formulated as a constrained optimization problem that finds a minimum energy (e.g., minimum L_2 norm) solution with only positive values smaller than the original input to explain current outcomes. Specifically, constraints on possible ranges of inverse results mean that negative values are assumed to have no contribution to the current outcomes, and positive values larger than the original input should not happen because the signals from operations are interpreted as magnitudes of image features.

Experimental results have shown that the existing attribution methods can be reproduced with the pixel intensity heatmap of inversed predictions computed by the proposed operation-wise method. Moreover, quantitative analysis has demonstrated that attributions by the proposed operation-wise approach have state-of-the-art performances in terms of deletion metric [18].

1.2.2 Layer-wise method

The operation-wise inverse method can be applied to CNNs with multiple-valued predictions because it does not depend on a single scalar value representing the prediction. However, the outputs of fundamental operations in the operation-wise approach would have different characteristics. For example, the negative-valued outputs from a convolution operation with a negative-valued bias and a convolution operation without bias



(a) (b)

Figure 1.4: Examples of the proposed layer-wise inverse-based method: (a) classification and (b) regression. For the classification task, the inverse of the predicted class ($\hat{x}_c = \Phi^{-1}(e_c)$ for $c = \max_i \Phi(x)$) highlights the object of that class. For the regression task that predicts the high-frequency detail ($y = \Phi(x)$), the inverse of the prediction ($\hat{x} = \Phi^{-1}(y)$) shows the important pixels in the estimation. Here, Φ represents given CNN and e_c is a unit vector that has 1 on the position of c.

can have different physical interpretations.

This dissertation proposes the layer-wise inverse-based approach based on the consideration that CNNs can be described as compositions of layers that have positive-valued activations as their inputs and outputs to resolve issues raised by fundamental operations.

In detail, the layer-wise approach defines a layer as the composition of a linear operation and a following nonlinear operation except the max-pooling operation that is treated as a layer by itself. For example, if a ReLU operation follows the convolution operation, the layer-wise approach treats the convolution and ReLU operations as one layer that processes positive-values. This consideration makes the input and output of each layer to be positive values that can be regarded as neural activations [25]. The inverses of layers are formulated as constrained optimization problems based on the following three postulations: (1) forward-pass is a sequential propagation of neural activations (2) it is important to find neural activations that actually contribute to the predictions, (3) a small amount of neural activation is desirable for human interpretability. Then, the inverse of prediction is computed by applying inverses of layers consist the CNNs.

As a result, the proposed layer-wise method can yield human interpretable inverse results for both classification and regression networks in a same framework as shown in Fig. 1.4. When the layer-wise inverse process is applied to a class label of interest, the inverse result can yield an attribution similar to conventional methods, and if the proposed method is applied to the output of a regression network, the inverse of the prediction can yield a heatmap that shows the contributions of pixels.

Evaluations of the proposed layer-wise inverse method have been performed on VGG16 [26] trained on the ImageNet classification [1] and VDSR [27] that regression network for the single image super-resolution (SISR). Experimental results with VGG16 have shown that the proposed layer-wise method successfully visualizes the input and output relationship and gives attributions comparable to the state-of-the-art methods. Experiments with VDSR have revealed that the enhancement of details by VDSR is concentrated on high-frequency bands, and this high-frequency selectivity may degrade the performance of super-resolution.

In summary, this disseration makes the following contributions.

- This dissertation developed the guided nonlinearity technique that enhances the integrated gradients methods by guiding gradients of nonlinear operations in the CNNs.

- This dissertation proposed the operation-wise inverse-based method that reproduces the existing attribution methods with a single unified framework.

- This dissertation proposed the layer-wise inverse-based method that explains the predictions of CNNs, for both classification and regression problems.

1.3 Outline

The rest of this dissertation is organized as follows. In chapter 2, conventional explanation methods for CNNs are reviewed. The proposed GNL method is described in chapter 3. In chapter 4 and chapter 5, operation-wise and layer-wise inverse-based methods are described, respectively. Finally, this dissertation is concluded in chapter 6.

Chapter 2

Related Work

Numerous methods have been proposed to interpret CNNs and among many methods, the concept of attribution is widely accepted as an effective tool for explaining CNN's behavior. [2, 3, 28]. Methods to compute the attribution can be classified into activation-based, perturbation-based, and backpropagation-based approaches according to their approaches to measuring input-output dependency. Also, there is an inverse based approach to which the proposed operation-wise and layer-wise inverse method belongs.

2.1 Activation-based approach

In CNNs, inputs are transformed by a series of convolution filters, which finally yield feature maps (activation maps) for fully connected layers. Since convolution layers work as a general feature extractor, visualization results can be obtained by computing the channel-wise contribution to a current prediction. Hence, research in this category applied some computations to the fully connected layers, while considering that activation maps obtained by convolutional layers are fixed. Specifically, the channel-level contribution is computed either with or without external networks. For some examples, Class Activation Map (CAM) [29] introduced an external network and used predefined masks to train the network. The introduction of these external networks allows us to apply this idea to a range of CNNs. However, it is unclear whether newly trained networks faithfully reflect the inference processes of the original CNNs. The other approach (e.g., Grad-CAM [9], Grad-CAM++ [30], and Smooth Grad-CAM [31]) did not use external networks. Rather, they used the weights of fully connected layers to compute the local explanations as a linear combination of activation maps.

2.2 Perturbation-based approach

Perturbation-based methods use perturbed inputs in attribution evaluation. Meaningful Perturbation [32] obtains binary attribution maps (object masks) by minimizing a cost function that prefers strong responses with small and smooth region masks. In [33], the authors improved the cost function in [32] so that their algorithm could yield smaller masks. However, both methods suffered from *faulty evidence* (noise-like responses on non-object regions), probably caused by gradient descent-based optimization techniques [34]. To alleviate this problem, FGVis [10] applied the gradient clipping method to compute attribution. Despite the fact that their results were visually improved, it was not clearly discussed why the gradient clipping helped the suppression of the *faulty evidence*.

Instead of using gradient descent optimization, the occlusion-based method [8] repeatedly evaluated partially occluded inputs. For all locations, this algorithm removed the content around a given location and evaluated the changes in the corresponding output. The method then used the difference between the original prediction score and the score obtained with the occluded input to define attribution. This approach reduced *faulty evidence*. However, the sizes and shapes of masks need to be heuristically chosen. In RISE [18], they attempted to alleviate this problem by considering perturbations in a probabilistic way and defined the attribution as the expected value of possible scores. Because of the large size of the sampling space, they used the Monte-

Carlo method: They randomly generated a set of masks, and the attribution of a given pixel is defined as the average score for the masks that include that pixel.

2.3 Backpropagation-based approach

The backpropagation-based approach includes gradient-times-inputs [19], guided backprop [35], Grad-CAM [9], layer-wise relevance propagation [14], DeepLIFT [19], and integrated gradients [11]. These methods were developed based on the observation that gradients represent linear approximations of input and output relations, and that large gradients indicate that these pixels are significant in current classification results.

Gradient-times-inputs [19] defined attributions as the element-wise product of gradients and input intensities. In the study on guided backprop [35] the authors focused on gradient flow, allowing only the flows of positive values, and improved visual qualities. Grad-CAM [9] assumed that the feature map of the final convolution layer could be low resolution attributions and developed a method to compute their linear combination for each class. Layer-wise relevance propagation (LRP) [14] computed attribution by back-propagating the final prediction down to the input space with the layer-wise relevance propagation framework. The study on DeepLIFT [19] followed the same approach to LRP. However, to make the attribution satisfy the *summation to delta* property, which means the sum of attribution should be equal to the difference between two scores (a score from a given input and a score from a baseline image), the authors developed a method by separating positive and negative gradients. integrated gradients (IG) [11] used the line integral of gradients from baselines to calculate attributions. This method, in addition to the *summation to delta* property, satisfies the *implementation invariance* axiom, which means that attributions should be the same for functionally equivalent networks.

2.4 Inverse-based approach

In inverse-based methods, input and output relationships are visualized by computing the inverse functions of CNNs, with additional assumptions. Studies conducted by [24] and [8] are most relevant to the proposed methods among these inverse-based methods. [24] inverted the predictions of CNNs by treating an entire CNN as a single function. The authors formulated this problem as an optimization task by employing the sum of total variation and L_2 norm as a regularization term. Their method successfully localized features in input images, however, the visualizations of inverse maps are blurred. DeconvNet [8] visualizes features in layers by applying deconvolution repeatedly. To this end, the authors approximate inverse-operations for max-pool and convolutions. They applied inverse operations in a layer-wise manner to provide intuitive feature visualization. However, this method was based on heuristics and can only be used to visualize a sparse feature.

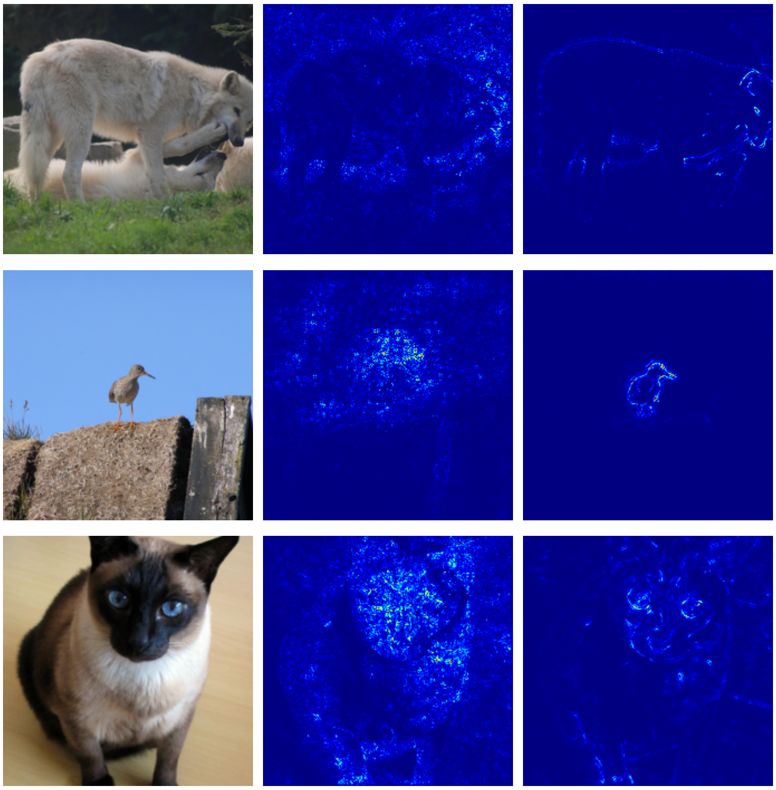
Chapter 3

Guided Nonlinearity

3.1 Motivation and Overview

Convolutional neural networks (CNNs) have shown remarkable performance in numerous computer vision tasks [36]. It is due to many recent technological and scientific progress, including the development of high-performance hardware enabling parallel processing, the availability of large-scale datasets, and the improvement of optimization methods, and deep network architectures [1, 37, 38]. In terms of building blocks for deep neural networks, the most important advance is probably the use of new nonlinear operators such as rectified linear units (ReLUs) and the pooling with maximum elements (max-pooling) [39]. Specifically, ReLU operators alleviated the vanishing gradient problem, and the max-pooling operation successfully reduced the resolutions of feature vectors (or equivalently enabled large receptive fields) while preserving the relevant information for the correct prediction.

Along with performance improvements, demands to understand neural networks (explainability and interpretability) have increased, and thus many methods have been proposed for the interpretation of networks and decision results. However, the high dimensionality and the nonlinear nature of neural networks makes it difficult to interpret the mechanism of CNNs directly (e.g., decision boundary visualization). Rather,



(a) Input (b) IG (c) Proposed

Figure 3.1: Attribution heatmap obtained by integrated gradients (IG) [11] and the proposed method on VGG16 (top, lable: *white wolf*) [26], ResNet50 (middle, label: *redshank*) [40], and GoogleNet (bottom, label: *siamese cat*) [41]. Attribution allows users to interpret decision results. The proposed GNL method provides more target-focused results compared with the IG.

researchers focused on indirect methods (e.g. post-hoc explanation) to understand the process of CNNs [2,3,6,7,28,42,43], and computed the attribution that can be visualized as a heatmap showing the importance of each pixel in the prediction as illustrated in Fig. 3.1. It is possible to understand the reason why a neural network yields such classification results and use this insight for error analysis and model improvements by using attribution [18].

Among a number of methods to compute attribution [8,10,11,18,35], the integrated gradients (IG) method has received much attention due to its non ad-hoc design [11, 44,45]. IG defines attribution with a path integral value of gradients (differentiation of the prediction result with respect to the input image) from the baseline image to the input image. The sum of the attribution equals to the difference between two outputs of the baseline image and the input image [11].

Since the goal of attribution is to measure the contribution of each pixel to the output, using gradients is a simple and intuitive choice. That is, gradients reflect linear relations between inputs and outputs, and it is a good measure to represent their dependencies. However, this dissertation assumes that this measure (gradients) used in the computation of attribution can be improved. To be precise, nonlinear operators in neural networks [15–17] could be considered as the computational modeling of neurotransmitter release that controls action potential firing in neurons [46]. The rate of action potential generation in postsynaptic neurons depends on the sum of excitatory (EPSP) and inhibitory postsynaptic potentials (IPSP) that are induced by neurotransmitters [47]. In this context, what users want to know is which pixels are supporting the current output, and this dissertation assumes that paths consisting of excitatory synapses reflect the contribution more faithfully. Based on this observation, this dissertation proposes a method called guided-nonlinearity (GNL) that makes only positive gradients backpropagate through nonlinear units.

The experimental results have shown that attributions obtained by GNL are fine-grained, as shown in Fig. 3.1 (See also experimental section). For the objective evaluation, this dissertation adopted deletion and insertion metrics [18] and compared the results with several existing methods on popular CNN structures. The comparison shows that the proposed method outperforms IG in both metrics and achieves state-of-the-art results in the deletion score.

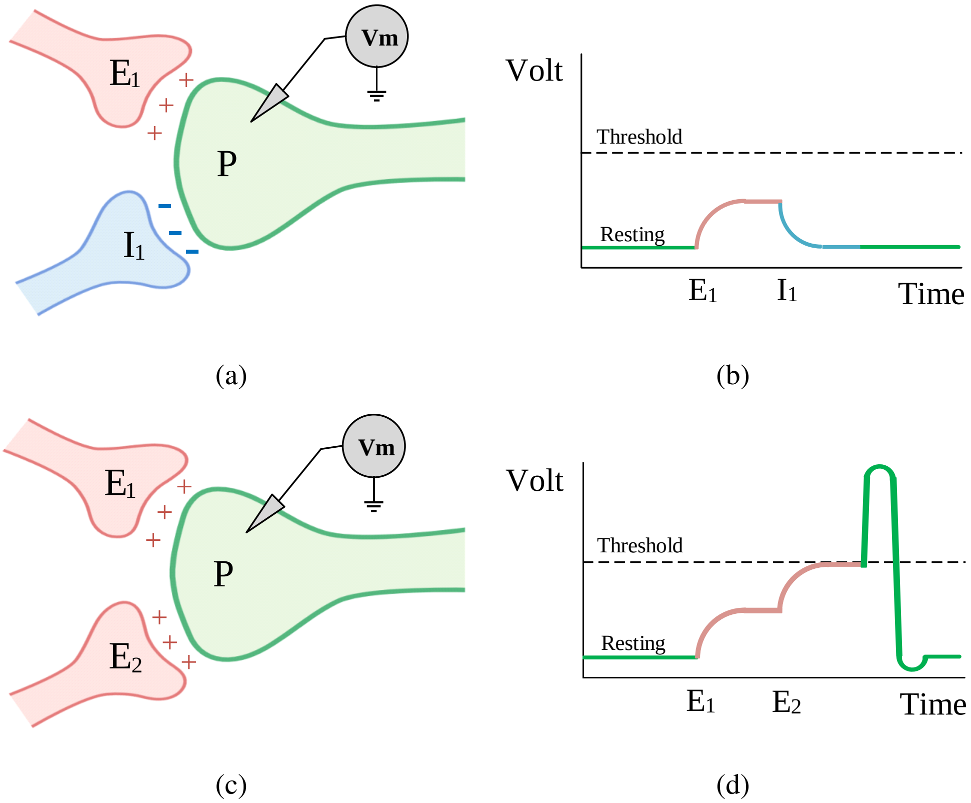


Figure 3.2: Diagrams of synapses with different postsynaptic potentials: (a) illustrates that a synaptic cleft consists of two presynaptic neurons (one generates EPSP (E_1) and the other generates IPSP (I_1) to the postsynaptic cell (P)) so that the postsynaptic cell (P) remains in the resting state as shown in (b). (c) and (d) illustrate the firing of an action potential in postsynaptic neuron (P) with two EPSPs generated by E_1 and E_2.

3.2 Proposed Guided Non-linearity

Although gradients are mathematically sound and easy to compute, this dissertation supposes that gradients obtained by backpropagation lack some properties to analyze input-output relations in highly nonlinear systems [9, 13, 35]. For the presentation of GNL, this dissertation first reviews the IG method [11] and presents postulations and details of the proposed method.

3.2.1 Integrated Gradients

To make attribution of an input x for a given CNN (denoted as F), IG used a line integral of gradients along the path from a baseline image x' to the given input x:

$$\mathrm{IG}_i(x) = (x_i - x'_i) \times \int_0^1 \frac{\partial F(x' + \alpha \times (x - x'))}{\partial x_i} d\alpha \tag{3.1}$$

where i is used as a pixel index. Unlike [48], this formulation use gradients to approximate only small intervals (it becomes clearer when considering its discrete version in (3.3)). The authors also showed that (3.1) satisfies *the completeness property*:

$$\sum_i \mathrm{IG}_i(x) = F(x) - F(x'), \tag{3.2}$$

which means that the sum of all attribution equals to the (network) output distance between x and x'. This property is also called *the summation to delta* [49] and *efficiency* in the literature [50, 51]. For the implementation, (3.1) is approximated with its discrete version:

$$\mathrm{IG}_i(x) \approx \frac{1}{m} \sum_{k=1}^{m} \frac{\partial F(x' + \frac{k}{m} \times (x - x'))}{\partial x_i}. \tag{3.3}$$

Intuitively, $\mathrm{IG}_i(x)$ is the sum of incremental contributions of the i-th pixel to the output (along the path from x' and x), where each contribution was evaluated with gradients.

3.2.2 Postulations

Neural networks are based on the modeling of neurons [15–17] that have linear and nonlinear parts. The nonlinear operators in neural networks could be considered axonal terminals that control the generation of action potentials in postsynaptic cells by releasing neurotransmitters [46, 52, 53]. In the human visual system, neurotransmitters induce EPSPs and IPSPs on postsynaptic neurons according to their types (e.g. glutamate and GABA produce EPSP and IPSP respectively) and the likelihood of action potential firing in postsynaptic cells increases with EPSPs and decreases with IPSPs [54,55]. Actually, the spatial and temporal sum of EPSPs and IPSPs determines the depolarization of postsynaptic cells [47] as illustrated in Fig. 3.2. This dissertation postulates that nonlinear units (e.g., ReLU and max-pooling) with positive gradients operate as EPSPs and neurons that generated EPSPs should be focused on finding the chain of fired neurons.

3.2.3 Proposed method

Based on the above observations, it is natural to focus on the positive gradients in nonlinear units (corresponding to axonal terminals) for attribution. In other words, positive gradients should be focused to find the cause of the current prediction, because neurons yielding IPSPs are against the current prediction results.

This dissertation computationally achieves this goal by clipping negatively valued gradients in nonlinear units to zero (Fig. 3.3) and use these new gradients in the path integral of IG. As an example, it is possible to express a ReLU unit as,

$$y = \mathrm{relu}(x) = x \odot I(x > 0) \tag{3.4}$$

where $I(\cdot)$ is an indicator function and $\odot$ mean the elementwise product. The conventional backpropagation is given by

$$\frac{\partial F(\cdot)}{\partial x} = \frac{\partial F(\cdot)}{\partial y} \odot I(x > 0). \tag{3.5}$$

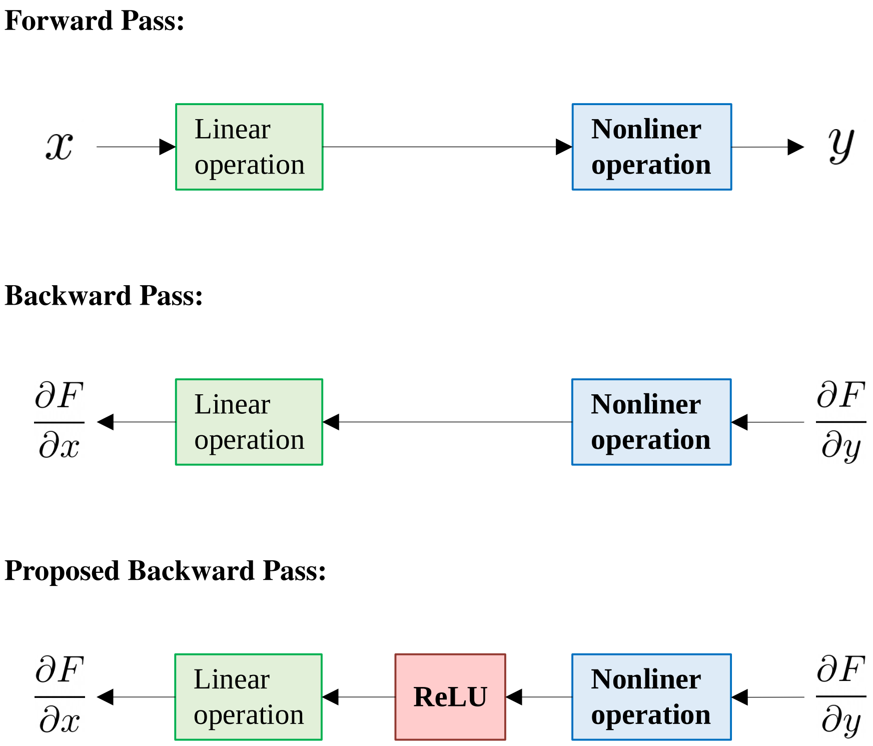


Figure 3.3: Comparison of the proposed guided non-linearity with the normal backpropagation. Both methods have the same forward pass for all operations. For backward pass, however, the proposed GNL method apply ReLU to clip negatively valued gradients to zero.

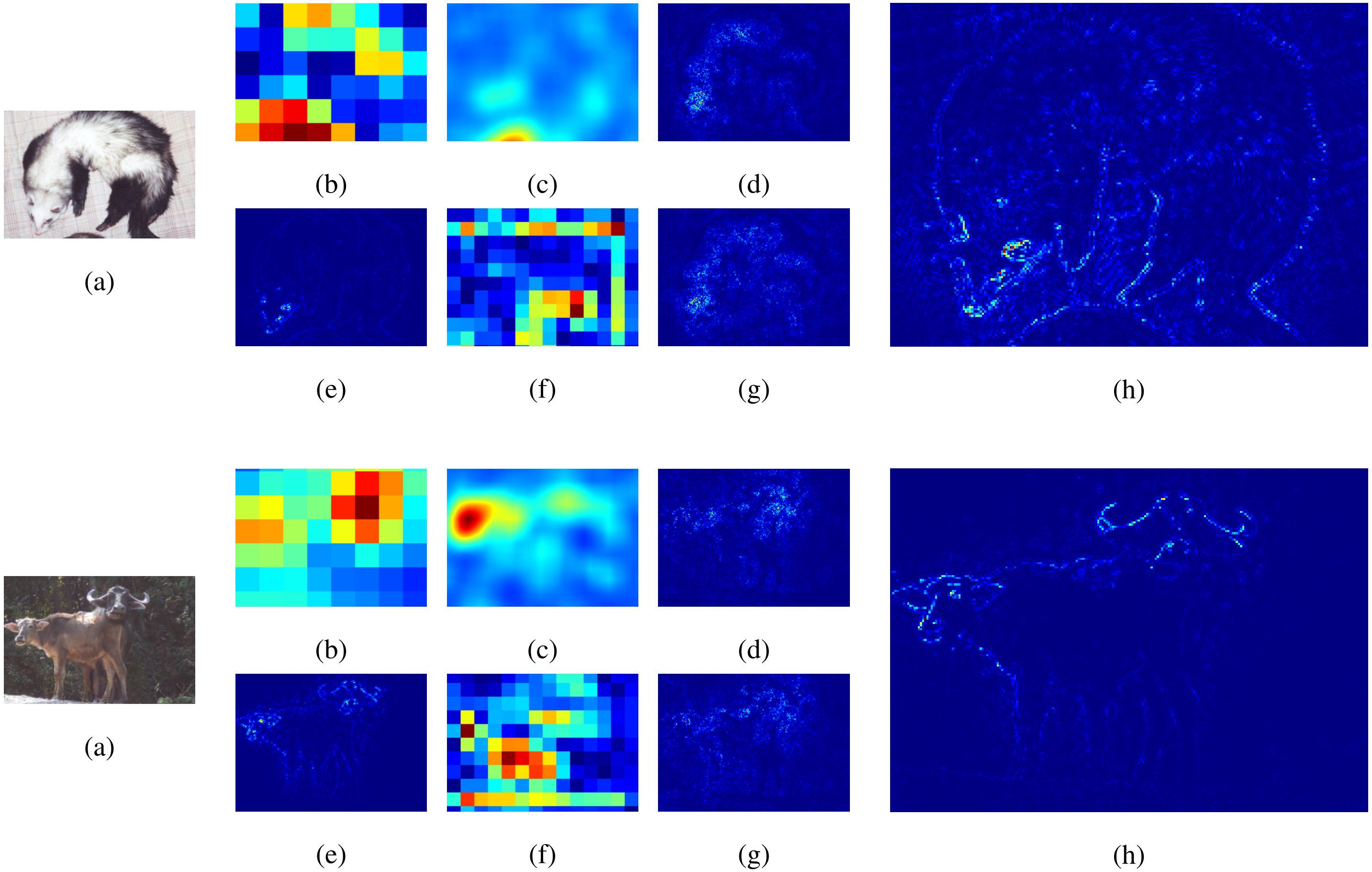


Figure 3.4: Comparison of attribution heatmaps using VGG16 [26]: (a) Input image (label: *ferret, water buffalo*), (b) Occlusion [8], (c) RISE [18], (d) Gradients [48], (e) Guided Backprop (GB) [35], (f) Grad-CAM [9], (g) Integrated Gradients (IG) [11], and (h) GNL. Note that perturbation-based methods ((b) and (c)) provide coarse localization.

Howerver, to make gradients propagate along neurons generating EPSPs, this dissertation propose to use

$$\frac{\partial F(\cdot)}{\partial x} = \text{relu}\left(\frac{\partial F(\cdot)}{\partial y} \odot I(x > 0)\right) \tag{3.6}$$

instead of (3.5). For a max-pooling operator,

$$y_i = \text{max}_j x_{ij} \tag{3.7}$$

where i is the index for the output of max-pooling and j is the index for the input, the conventional backpropagation is

$$\frac{\partial F(\cdot)}{\partial x_{ij}} = \frac{\partial F(\cdot)}{\partial y_i} \odot I(x_{ij} = y_i). \tag{3.8}$$

Then GNL propose to use

$$\frac{\partial F(\cdot)}{\partial x_{ij}} = \text{relu}\left(\frac{\partial F(\cdot)}{\partial y_i} \odot I(x_{ij} = y_i)\right). \tag{3.9}$$

This gradient clipping method is also used in the guided backpropagation (GB) [35] for ReLU operation. However, it was heuristically designed only for ReLU operators and evaluated subjectively.

3.3 Experimental Results

GNL has implemented with PyTorch [56]. Fig 3.4 shows that attribution heatmaps of GNL are fine-grained and more human-interpretable than the heatmaps of others, including the baseline method IG. The first row of Fig. 3.4 shows that the proposed method captures the outline of*ferret* better than others, and the second row shows that GNL successfully delineates multiple objects (*water buffalo*).

This dissertation also compared the results with 6 existing methods on 5 popular CNN models objectively [9, 11, 18, 35, 48, 57]. Since the objective evaluation of attribution is not straightforward, two quantitative measures, insertion and deletion scores, are explained, and experimental results will be presented in terms of them.

Table 3.1: Comparison of deletion (lower is better, ↓) and insertion (higher is better, ↑) metrics for 5 networks.

	VGG16		VGG19		ResNet34		ResNet50		GoogleNet	
Method	Deletion ↓	Insertion ↑	Deletion ↓	Insertion ↑	Deletion ↓	Insertion ↑	Deletion ↓	Insertion ↑	Deletion ↓	Insertion ↑
Occlusion [8]	0.1577	0.5755	0.1616	0.5770	0.1874	0.5914	0.2141	0.6309	0.1350	0.4667
LIME [57]	0.1014	0.6167	-	-	-	-	0.1217	**0.6940**	-	-
RISE [18]	0.0964	**0.6048**	0.0998	**0.6070**	0.1028	0.6308	0.1121	0.6762	0.0684	0.4995
Gradients [48]	0.0672	0.3270	0.0791	0.3423	0.1268	0.4221	0.1134	0.4234	0.0745	0.3574
GB [35]	0.0526	0.5279	0.0567	0.5445	0.0826	0.6141	0.0755	0.6460	0.0639	**0.5124**
GradCam [9]	0.1605	0.4305	0.1520	0.4578	0.1557	**0.6333**	0.1887	0.6715	0.1156	0.5086
IG [11]	0.0543	0.3621	0.0640	0.3792	0.1030	0.4575	0.0931	0.4589	0.0634	0.3936
Proposed	**0.0495**	0.5151	**0.0532**	0.5295	**0.0763**	0.5932	**0.0721**	0.6295	**0.0601**	0.4912

3.3.1 Evaluation Metrics

For the quantitative evaluation, this dissertation has used the deletion and insertion metrics proposed in [18], which was designed to evaluate the quality of attribution without human intervention. Basically, the methods sort pixels according to their importance and evaluate the quality of sorting results in two complementary ways.

For the evaluation of the deletion metric, pixels are deleted (pixel values are replaced with 0) according to the descending order of importance (the more important it is, the earlier it is removed). When the sorting order faithfully reflects the importance of pixels, the classification results will be quickly degraded (and vice versa). Therefore, the area under the curve whose horizontal axis means the number of deleted pixels and the vertical axis is the softmax output of the original class shows the quality of attribution. An example of curves can be found in Fig. 3.5-(c). Since sharp drops indicate better attribution, the lower deletion metric means better performance.

The insertion metric is also defined as the area under the curve. However, the curve is defined in the opposite way. From an initially blurred image, true pixel values are inserted (blurred pixel values are replaced with true values) according to the descending order of importance. If the importance of pixels is well-reflected in this order, the softmax output of the original class will increase sharply. Therefore, better attribution shows a larger insertion metric. This dissertation used the same baseline image in [18] for the fair comparison.

3.3.2 Experiment details

GNL is evaluated with 5 CNN architectures (VGG16 , VGG19 [26], ResNet34, ResNet 35 [40] and GoogleNet [41]) on 5,000 linearly sampled images from the validation split of ImageNet classification database [1]. The Captum library [58] is used to evaluate the performance of existing methods: Occlusion [8], Gradients [48], GB [35], GradCAM [9] and IG [11]. For the methods that are not supported by the Captum library, network architectures are replaced based on the authors' code [18] or used the numbers reported in [44]. For a multi-channel attribution (including GNL), scalar-value-importance is calculated by the absolute value of the sum of attribution along with the channel dimension [10, 44].

3.3.3 Results and Discussions

The quantitative results are summarized in TABLE. 3.1. As shown, GNL outperforms IG in both metrics. Two typical examples of curves (used in the metric evaluation) are also shown in Fig. 3.5. As shown, attribution by GNL focus on important pixels and yields better deletion/insertion curves. This clearly means that GNL improves the IG method. Also, the proposed method outperforms all other methods in terms of the deletion metric and gets the insertion metric score comparable to the state-of-the-art results. Although the proposed method yields a little lower insertion metric compared with perturbation methods, these methods lack the power to localize targets as can be seen in Fig 3.4-(b) and (c).

3.4 Summary

In this chapter, this dissertation have proposed GNL that generates human interpretable attribution. GNL modifies the backpropagation methods on ReLU and max-pooling nonlinearity used in the path integral of integrated gradients. This design was inspired by the mechanism of action potential generation in postsynaptic neurons of the human visual system. Extensive experiments show that GNL yields visually pleasing and more human interpretable results, and quantitative evaluation also indicates that GNL achieves the state-of-the-art deletion score and outperforms the IG method.

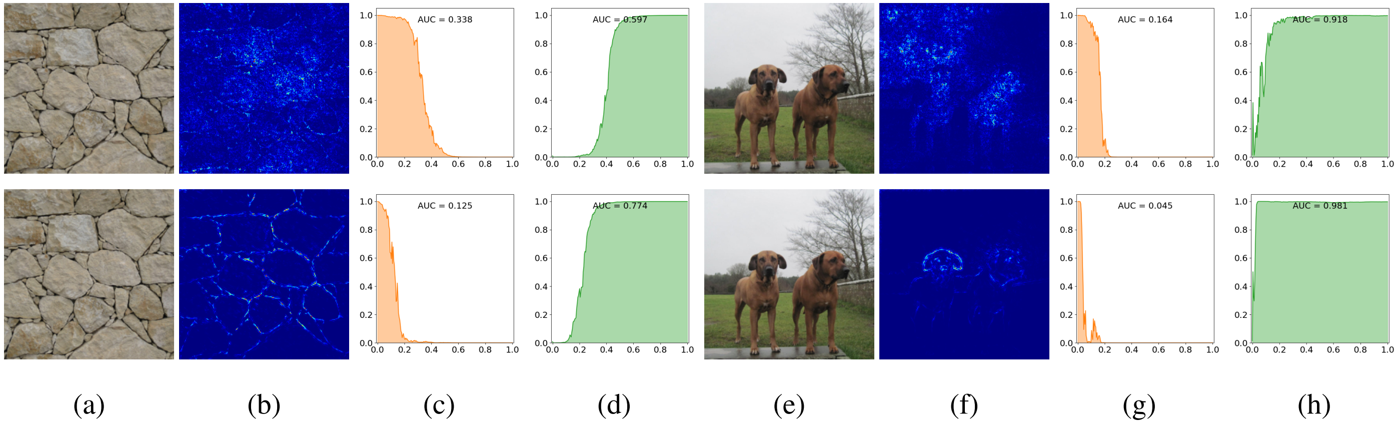


Figure 3.5: Illustrations of the deletion and insertion metrics for IG (upper) and GNL (below) using ResNet50: (a), (e) input images (label: *stone wall*, *Rhodesian ridgeback*), (b), (f) attribution, (c), (g) curves for the deletion metric (IG: AUC=0.338/0.164, GNL: AUC=0.125/0.045 respectively), (d), (h) curves for insertion metric (IG: AUC=0.597/0.774, GNL: AUC=0.918/0.981 respectively).

Chapter 4

Operation-wise Approach

4.1 Motivation and Overview

Convolutional Neural Networks (CNNs) have been used in many computer vision applications with state-of-the-art performance. Moreover, the performance is continuously improving as new architectures, better training methods, and larger/realistic training sets are being developed [26, 36, 39]. Along with these performance-centric results, researchers have also attempted to understand and explain the inner workings of CNNs [4, 5, 42].

There are roughly three approaches to provide human-understandable explanations to machine learning models [59]. First, the example-based approach aims to provide users with relevant training samples to a prediction result, so they can get clues to understanding the system. Second, the global approach focused on the trained model itself and tried to provide a set of understandable rules that simulate the inference processes. Although both approaches are intuitive, they are not easily applicable to deep-learning models due to their high complexity. Hence, researchers are recently focusing on the third approach, i.e., local explanations. The local approach is to explain a prediction result by showing the changes of predictions according to small changes in inputs, where the changes are usually described in images (e.g., heat-maps). In this category, numerous methods have been proposed, which are greatly improving the understanding of CNNs by highlighting important clues in inputs that can explain the behaviors of CNN [8,9,11,13,60,61]. However, they are based on somewhat heuristic assumptions, and they lack common backgrounds.

This chapter addresses the CNN visualization problem by developing the inverse operations of a feed-forward (inference) pass. With the proposed inverse operations, many conventional local methods can be explained in a single framework and obtain a range of visualization results by selecting corresponding settings as illustrated in Fig. 4.1. Specifically, the proposed method encompasses the existing analyses, in that Figs. 4.1(a)-(d) respectively correspond to:

(a) The inverse of fully connected layers yields equivalent results to [9].

(b) The proposed method can visualize active neurons in intermediate layers as in [13].

(c) The proposed method can evaluate the importance/contribution of input pixels (attributions) like [11].

(d) The proposed method can visualize input patterns that make a specific neuron active as [8].

This dissertation focuses on the physical interpretations of neuron activations and designs the corresponding processes to develop an inverse-based framework. The following three assumptions are based on the general properties of neural networks.

First, this dissertation assumes that the inverse should be computed only with neurons that contributed to the inference (i.e., active neurons only). Some neurons were off in the forward-pass, and these neurons should not be used in the inverse process.

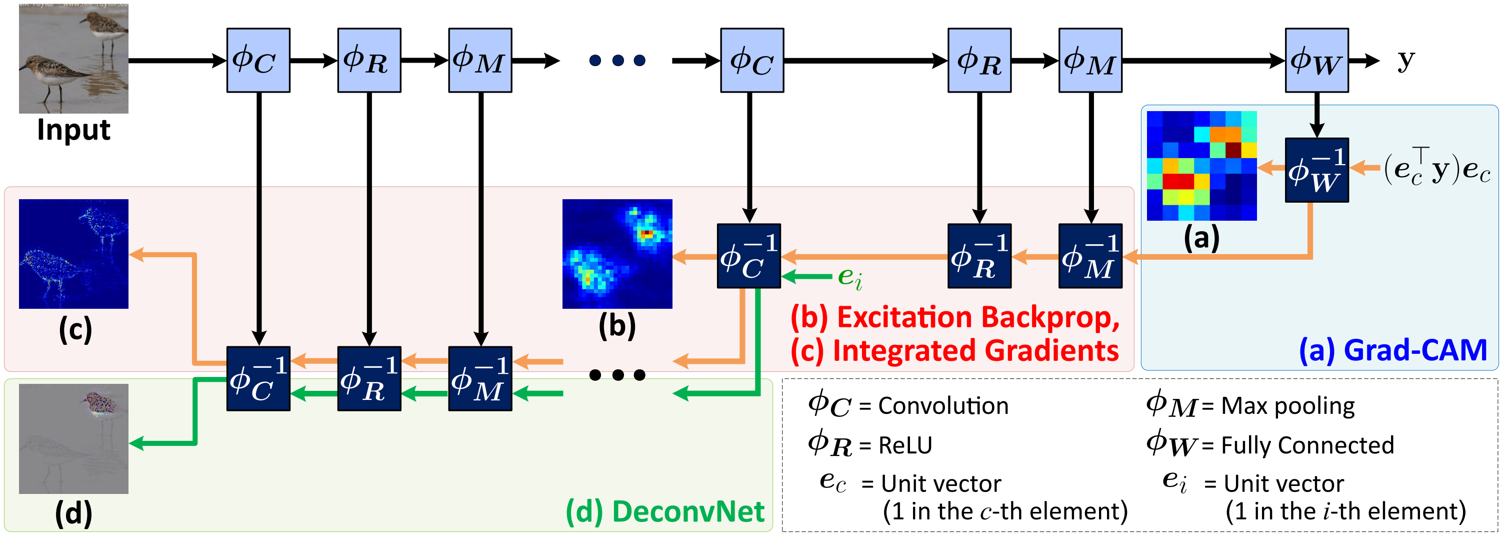


Figure 4.1: The overview of the proposed operation-wise inverse of CNNs, where e_c (e_i) is a unit vector having 1 in the c-th (i-th) element. By applying the inverse procedures, a range of visualization results can be computed to understand given neural networks.

Second, although non-linear units can be problematic in finding the inverse, they are generally simple (ReLU and max-pooling) to be linearized near operating points. Third, this dissertation imposes constraints on internal values (activations). In the proposed framework, inputs and outputs of operations indicate the levels of neuron activations, and this dissertation assumes that they should be within valid ranges. For instance, the negative (internal) activations are not allowed in explaining current outcomes because negative neural activations do not have physical meanings.

Based on these ideas, this dissertation formulates the inverse of each operation as a constrained optimization problem. As illustrated in Fig. 4.1, the proposed method can handle networks with an arbitrary number of layers by applying the inverse sequentially. Experiments show that the proposed method can reproduce equivalent results to several conventional methods [8, 9, 11, 13]. Although all results look similar to that of the corresponding conventional work, it needs to be noted that they are achieved with a single unified framework. Quantitative results on AlexNet [39], VGG16, and VGG19 [26] also show that the proposed method achieves the best deletion score [18] for VGG16 and VGG19.

4.2 Proposed Method

In a forward-pass of CNNs, input signals generate sequential activations of neurons, and activations in lower layers are the causes of the activations in higher layers. Therefore, when analyzing the activation of a certain neuron, it is necessary to compute the inverse of its overall forward-pass [24]. In this section, this dissertation explains the designs of inverse processes.

4.2.1 Problem statement

The inference of a neural network can be denoted as $\mathrm{y} = \Phi(\mathrm{x})$ with input x and output y, which can be written as a cascade of operations as:

$$\mathrm{y} = \Phi(\mathrm{x}) = (\phi_{\mathrm{N}} \circ \cdots \circ \phi_2 \circ \phi_1)(\mathrm{x}). \tag{4.1}$$

where ϕ_i, $i = 1, 2, \ldots, N-1$, is the i-th operation in the network consisted of convolution, ReLU, and pooling, and the last one (ϕ_N) corresponds to the fully connected operation. Then, the goal of the proposed method is to develop a method to compute the inverse of $\Phi(\cdot)$ by designing operation-wise inverses, i.e., $\phi_i^{-1}(\cdot)$, $i = 1, 2, \ldots, N$. This operation-wise inverse not only allows users to explicitly consider internal activations but also enables a range of ways to visualize CNNs as illustrated in Fig. 4.1.

However, in many cases, $\phi_i(\cdot)$ is a many-to-one function and thus finding $\phi_i^{-1}(\cdot)$ satisfying

$$\phi(\mathrm{x}) = \mathrm{y} \longleftrightarrow \mathrm{x} = \phi^{-1}(\mathrm{y}) \tag{4.2}$$

is not a well-defined problem. This dissertation develops constraints related to motivations and interpretations of neural networks to address this challenge. That is, nonlinearities in neural networks are employed to reflect on/off behaviors of neurons, and the amount of activation indicates the strength (presence) of corresponding features in inputs.

4.2.2 Proposed constraints

Specifically, the proposed constraints are listed as

(1) *Linear approximation around the operating point*: Approximating the forward process with a linear function, around the operating points of neurons in a forward-pass.

(2) *Partial activation*: When a neuron activation in a given layer is $\bar{x}$, the activation computed in the inverse process should satisfy $0 \preceq x \preceq \bar{x}$, where $\preceq$ denotes the

element-wise inequality.

(3) *Sparse activation*: When the inverse process is under-determined (there are many solutions), sparse activations are preferred.

To be precise with the motivation behind the above constraints, this dissertation notes that a forward function generally changes dramatically, even for small input changes due to the switching behavior of non-linear units. Therefore, if non-linear units are used in different modes (on → off, off → on), the inverse function cannot be considered the inverse of the original forward-pass. Hence, the inverse should be performed on the operating point of the forward-pass for successfully analyzing a given case [10]. Similarly, neurons that were off in the forward-pass should not be used in its inverse. This dissertation also supposes that neither negative activation nor stronger activation than the forward-pass has physical meanings [62]. Finally, when there are many solutions, sparse activations are preferred from the ideas of [63, 64].

4.2.3 Mathematical formulation

For a operation of neural network described in eq.(4.1), its linearly approximated version $\phi'(\mathrm{x})$ around $\bar{\mathrm{x}}$ is formaulated as [65–67]:

$$\phi'(\mathrm{x}) = \left.\frac{\partial\phi(\mathrm{x})}{\partial\mathrm{x}}\right|_{\mathrm{x}=\bar{\mathrm{x}}}(\mathrm{x}-\bar{\mathrm{x}}) + \phi(\bar{\mathrm{x}}). \tag{4.3}$$

Then, the inverse process is defined as

$$\phi^{-1}(\mathrm{y}) = \arg\min_{\mathrm{x}} \|\phi'(\mathrm{x}) - \mathrm{y}\|_2 + \mathrm{S}(\mathrm{x}) \tag{4.4}$$

or

$$\phi^{-1}(\mathrm{y}) = \arg\min_{\mathrm{x}} \mathrm{S}(\mathrm{x}) \;\text{ subject to }\; \phi'(\mathrm{x}) = \mathrm{y}. \tag{4.5}$$

Note that both eq.(4.4) and eq.(4.5) are subject to

$$0 \preceq \mathrm{x} \preceq \bar{\mathrm{x}}, \tag{4.6}$$

and that eq.(4.5) is used when a closed form solution is available. The regularization term $S(\mathrm{x})$ is designed to penalize the amount of activation defined as:

$$S(\mathrm{x}) = \lambda_1 \|\mathrm{x}\|_1 + \lambda_2 \|\mathrm{x}\|_2 \tag{4.7}$$

where $\|\cdot\|_1$ and $\|\cdot\|_2$ are L1 and L2 norm, respectively.

4.3 Implementation details

In this section, this dissertation presents the proposed inverse processes of building blocks defined in eq.(4.1).

4.3.1 Inverse of ReLU and Max Pooling

Among four components in eq.(4.1), convolution and fully connected operations are linear, and ReLU and max-pooling introduce non-linearity. In the forward-pass, ReLU can be represented with binary matrices around the operating point as eq.(4.3). For example, $ReLU([a,b,c]^\top) = [a,b,0]^\top$ when $a,b > 0$ and $c < 0$, and this ReLU function can be represented around its operating point as a matrix multiplication:

$$\phi'\left(\begin{bmatrix} a \\ b \\ c \end{bmatrix}\right) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} \tag{4.8}$$

As ReLU can be considered a multiplication with a binary matrix R, it is possible to compute the Moore-Penrose pseudo-inverse (R^+) of R. With the formulation of eq.(4.5), the solution becomes the right inverse of R: $R^+ = R^\top$.

The output of this inverse process naturally satisfies eq.(4.6) because the input (y) satisfies the *partial activation* constraint. Similar to ReLU, max-pooling can be described as a matrix multiplication with a binary matrix M. Therefore, the exact solution of eq.(4.5) is given by $M^+ = M^\top$. This result also satisfies eq.(4.6).

Algorithm 1 Gradient Projection Algorithm (GPA) to solve eq.(4.4): τ determines termination condition, N is decay interval, and γ ($\in (0, 1)$) is a constant for decaying the step size r_n.

```
procedure GPA($\phi, \bar{\mathrm{x}}$)
    $f(\mathrm{x}) := \|\phi'(\mathrm{x}) - \mathrm{y}\|_2 + S(\mathrm{x})$
    $\mathrm{x}_o$ = blurred version of $\bar{\mathrm{x}}$
    $\tau_c = \tau f(\mathrm{x}_o)$
    while $\|c_n - c_{n-1}\| > \tau_c$ do
        $c_n \leftarrow f(\mathrm{x}_n)$
        $\mathrm{x}_n \leftarrow \mathrm{x}_n - r_n \nabla c_n$
        $\mathrm{x}_n \leftarrow$ clipping $\mathrm{x}_n$ with $0 \leq \mathrm{x}_n \leq \bar{\mathrm{x}}$
        if $\mod (n, N) = 0$ then
            $r_n \leftarrow \gamma r_n$
        end if
    end while
end procedure
```

4.3.2 Inverse of Fully Connected and Convolution Layers

Although a forward-pass of fully connected and convolution layers is linear, its inverse given by eq.(4.4) is difficult to evaluate directly, due to a huge number of weights and corresponding constraints. Therefore, their inverses are computed iteratively by minimizing a cost function with Gradient Projection Algorithm (GPA) [68] as summarized in Algorithm 1. Fortunately, gradient computations are efficiently supported by modern deep learning libraries like TensorFlow [69] and PyTorch [56].

4.4 Experimental Settings

To show the performance of the proposed method, experiments are conducted for $5,000$ images in ImageNet classification dataset. AlexNet, VGG16, and VGG19, which were trained for the ImageNet classification task are used for experiments.

For fully connected layers, the regularization term is not used ($\lambda_1 = \lambda_2 = 0$) because they are likely to be over-determined. For convolutional layers, λ_1 and λ_2 are set as 0.0 and 0.9 for experiments with the deconvolution method [8] and as 0.1 and 0.9 for other experiments. Although convolution layers can be either over-determined or under-determined, it is difficult to determine their types. Also, input images are normalized with a mean and standard deviation [1].

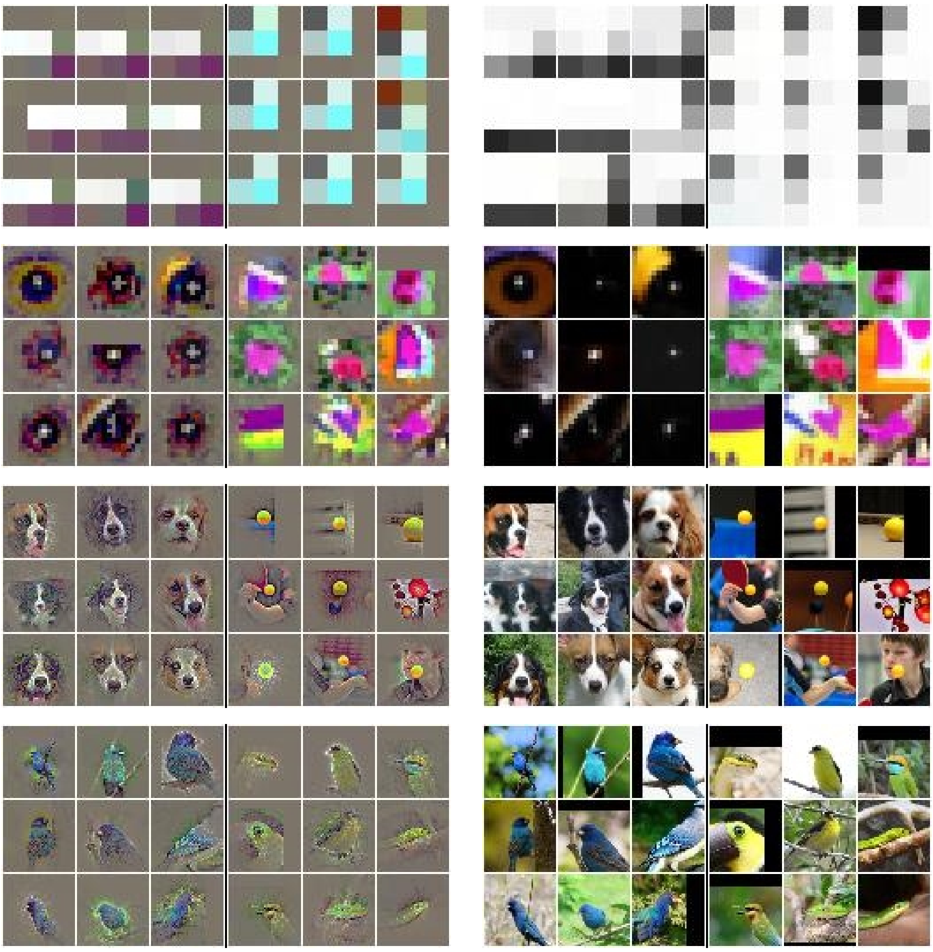
4.4.1 Qualitative results

The proposed method is closely related to many conventional methods [8,9,11,13], and thus their results can be reproduced by applying corresponding settings to the proposed framework. For visualization, local explanation masks are computed by summing up absolute values of final results across channels.

Deconvolution method

In [8], features learned in convolutional layers were visualized by computing the inverse (deconvolution) of an activation in a given feature map. For this, the authors designed inverse processes for convolution and max-pooling. Compared with the proposed method, they did not consider the inverse of ReLU layers and used the transposed convolution for the inverse of convolutional layers. Also, they did not impose any restrictions on strong activations, which may prevent the physical interpretation of deconvolution results. Empirical results also showed that this method works only when a sufficient number of max-pooling layers are used in the networks [35, 70].

The goal of the deconvolution method is to compute input patterns that fired the



(a) (b)

Figure 4.2: Input patterns that induced top 9 activations in selected channels for given layers: (a) Patterns obtained by the proposed method and (b) Corresponding patches in input images.

activation of a specific neuron. This process is naturally represented as an inverse function, and the proposed method can be used in the feature visualization as illustrated in Fig. 4.1(d). Similar to [8], this dissertation randomly selects channels from the feature map of each layer and makes a one-hot encoded vector using the top 9 activations in the selected channels. Then, these one-hot vectors are inversed to input spaces. As shown in Fig 4.2, the proposed method can visualize patterns that have activated a specific neuron, showing that the method in [8] can be explained with the proposed approach.

Grad-CAM method

Grad-CAM was proposed to handle fully connected layers [9], and it computed the contributions of individual channels in classifying an input as the c-th class. When a class-discriminative localization map (attribution) is denoted as L^c, their results can be expressed as

$$L^c = ReLU(\sum_k \alpha_k^c A^k) \tag{4.9}$$

where k is a channel index, A^k is the k-th channel in the feature map in the last convolution layer, and α_k^c is the average of $\frac{\partial(e_c^\top \Phi(\mathrm{x}))}{\partial A^k}$. The authors [9] claimed that the ReLU in eq.(4.9) was introduced to reflect positive influence only on the prediction.

In the proposed framework, the process in [9] can be considered as the inverse of fully connected layers of one-hot encoded prediction as illustrated in Fig. 4.1(a). As an example, for a classification result c, i.e.,

$$c = \arg\max_k \left(e_k^\top \mathrm{y}\right) \tag{4.10}$$

values corresponding to α_k^c are obtained by averaging the k-th channel of inverse of fully connected layers for $(e_c^\top \mathrm{y})e_c$. As shown in Fig 4.3, the proposed method gives similar results to Grad-CAM. Since the resolution of the last convolution layers is usually low, bicubic interpolation is used to get smooth heat maps. More examples can be found in Fig. 4.6(a), (b), and (d).

Excitation Backprop

In [13], neuron activations were visualized in a probabilistic way. The contribution of the j-th activation in a given layer to the prediction is defined as probability $P(a_j) = \sum_{a_i \in \mathcal{S}_\mathrm{j}} P(a_j|a_i)P(a_i)$ ($\mathcal{S}_\mathrm{j}$ is the parent node set of a_j in top-down order). For each a_j

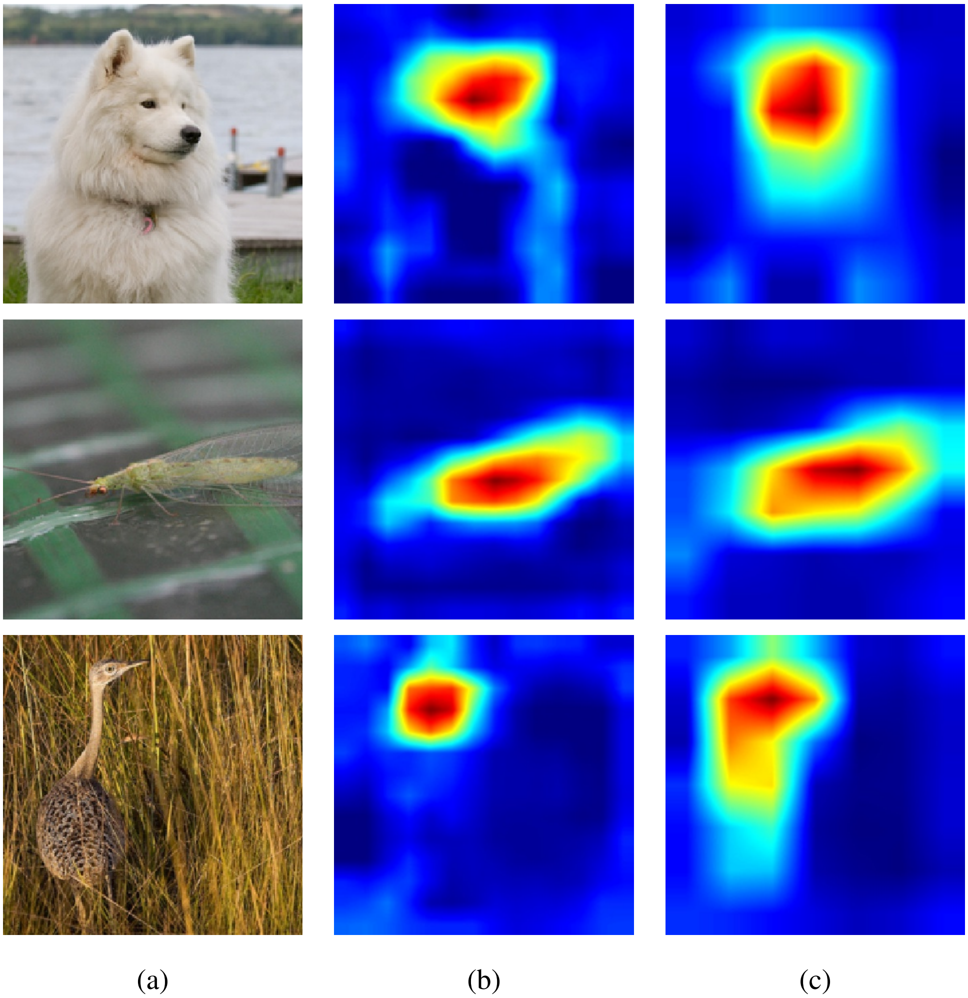


Figure 4.3: Comparison of Grad-CAM and the reproduction by the proposed method in VGG16: (a) Input images (labels: *samoyed* (upper), *lacewing* (middle) and *bustard* (lower)), (b) Grad-CAM, and (c) Proposed.

in child node set of a_i, $P(a_j|a_i)$ is calculated in a layer-wise manner:

$$P(a_j|a_i) = \begin{cases} Z_i \hat{a_j} w_{ji} & w_{ji} \geqslant 0 \\ 0 & w_{ji} < 0 \end{cases} \tag{4.11}$$

where Z_i is a normalization factor for $\sum_{a_j} P(a_j|a_i) = 1$ and w_{ji} is the element of a weight matrix in $\hat{a_i} = ReLU(\sum_j w_{ji}\hat{a_j} + b_i)$. In other words, $P(a_j)$ is a function of parent nodes in the preceding operations that can be conputed by top-down operation-wise manner.

Although the proposed method takes a deterministic approach, it can be used in back-propagating activations as illustrated in Fig. 4.1(b). Therefore, the proposed method can make attributions simillar to the results of [13] as visualized in Fig. 4.4.

Gradient-based attribution method

IG method [11] has been widely used to compute attribution [44, 45]. Also, it has desirable properties such as *implementation invariant* property and *summation to delta* property [14]. Precisely, given an input image x and a baseline image x^{o} (usually set to zero images), the method defines the attribution as

$$\begin{aligned} \mathrm{IG}_i(\mathrm{x}) &= (\mathrm{x}_i - \mathrm{x}_i^{\mathrm{o}}) \times \int_0^1 \frac{\partial \left(e_c^\top \Phi(\mathrm{x}^{\mathrm{o}} + \alpha \times (\mathrm{x} - \mathrm{x}^{\mathrm{o}})) \right)}{\partial \mathrm{x}_i} d\alpha \\ &\approx \frac{1}{m} \sum_{k=1}^{m} \frac{\partial \left(e_c^\top \Phi(\mathrm{x}^{\mathrm{o}} + \frac{k}{m} \times (\mathrm{x} - \mathrm{x}^{\mathrm{o}})) \right)}{\partial \mathrm{x}_i} \end{aligned} \tag{4.12}$$

where i is a pixel index, c is defined as eq.(4.10), and m is a user-defined constant. Naturally, $\mathrm{IG}_i(\mathrm{x})$ has a large value when this i-th pixel makes a large contribution for the classification.

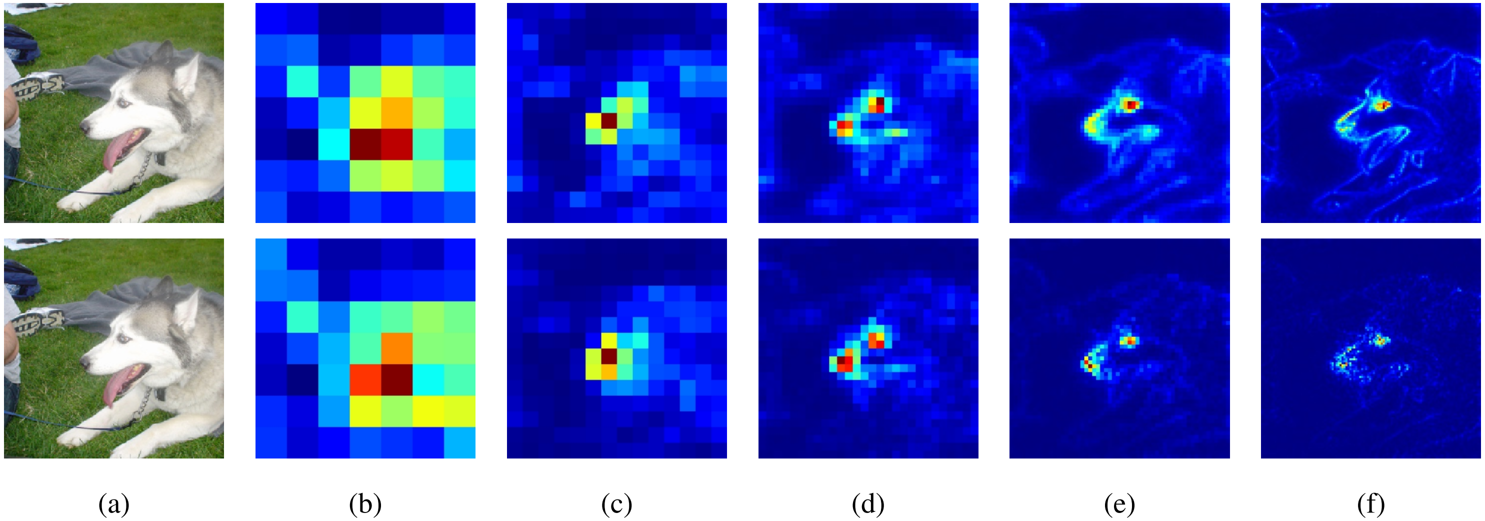


Figure 4.4: Visualization of activation from max pools in VGG16 by Excitation backprop [13] (upper) and the proposed method (lower): (a) Input image (label: *Siberian Husky*), (b) Pool5, (c) Pool4, (d) Pool3, (e) Pool2 and (f) Pool1.

The proposed method is not developed to evaluate the importance of input pixels, but it can compute the lower-layer activations that result in a given prediction. However, the inverse of a classification result, $\Phi^{-1}\left((e_c^\top \mathrm{y})e_c\right)$, can be considered an attribution. Note that (a) input pixel values can be considered activations for the first layer of neural networks and (b) the inverse of $(e_c^\top \mathrm{y})e_c$ yields activations that yield the c-th class. Therefore, the proposed method naturally yields the importance values of the current predictions as the attribution computed by IG. Fig. 4.5, Fig. 4.6(a), (c) and (e) show attributions obtained by IG method and the proposed method. As can be seen, the proposed method captures fine-grained details better than [11]. Especially Figs. 4.6(c) and (e) show that the proposed method captures all objects that contribute to the final prediction even when there is more than one object.

4.4.2 Quantitative Results

For the objective evaluation, this dissertation have computed deletion scores and compared them with conventional methods [10, 18, 44, 71]. The deletion score is defined as the area under a curve that represents the probabilities of an originally-predicted class for modified inputs. Here, modified inputs are made by removing pixels in input images according to the descending order of importance. If the importance (attribution) is computed correctly, the deletion of highly-ranked-pixels will quickly drop the class probability, and the deletion score should be small. Therefore, this metric can evaluate the quality of local explanation maps in a human-independent manner. Curve examples are shown in Fig. 4.5(d) and (e). In the graphs, the horizontal axis represents the proportion of replaced pixels, and the vertical axis is the probability of an originally-predicted class. As shown in Fig. 4.5(b) and (c), the proposed method more focuses on objects and shows better deletion scores.

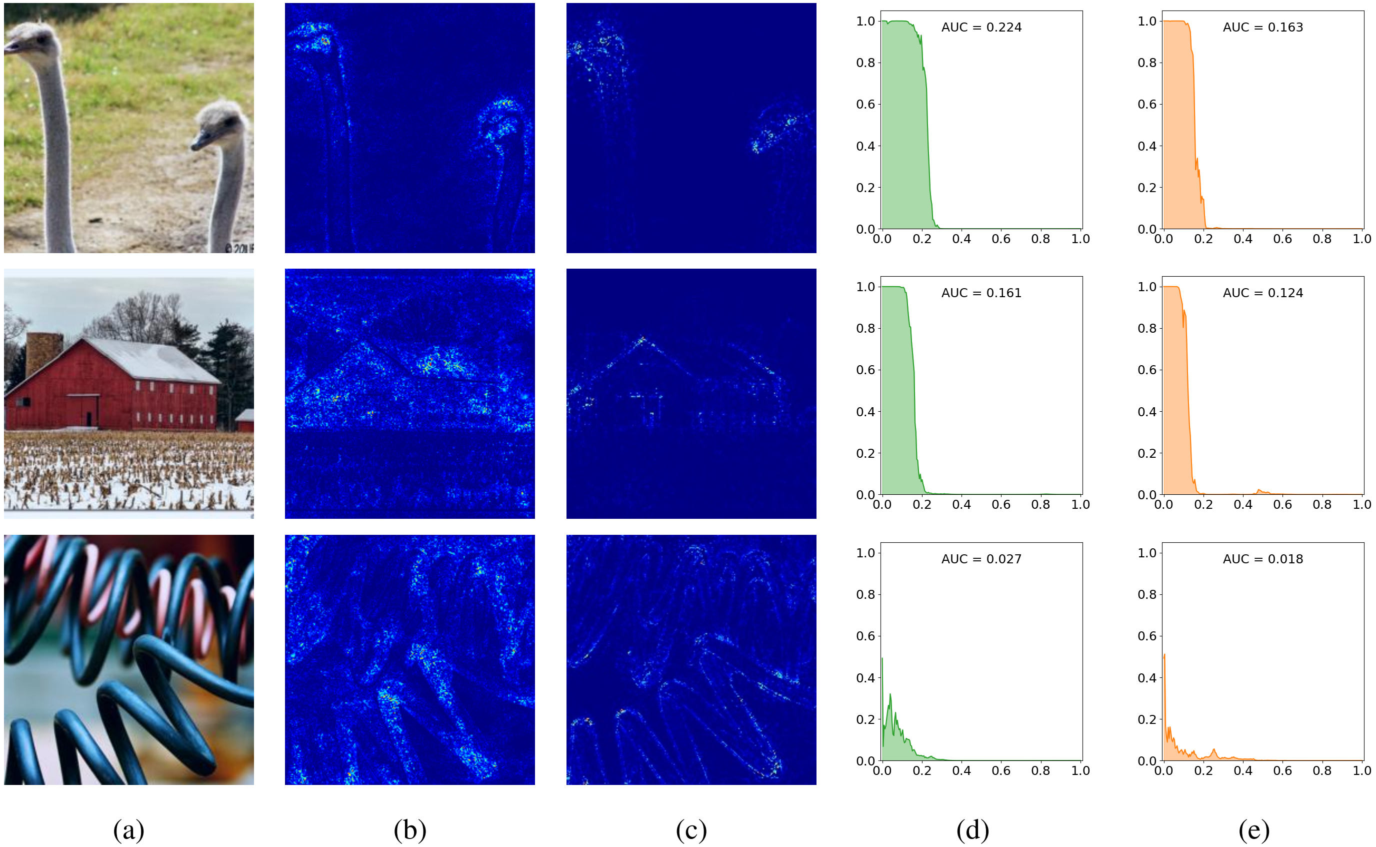


Figure 4.5: Comparison of Integrated Gradients (IG) [11] and the proposed method for VGG16: (a) Input images (labels: *ostrich* (upper), *barn* (middle) and *coil* (lower)), (b) Attribution heatmap of IG, (c) Attribution heatmap by the proposed method, (d) Deletion score curve of IG, and (e) Deletion score curve of the proposed method.

Table 4.1: The deletion scores computed on the ImageNet validation dataset. The rank of the top 3 methods are denoted after the score.

Method	AlexNet	VGG16	VGG19	Average
Occlusion [8]	0.0781	0.1577	0.1616	0.1325
LIME [57]	-	0.1014	-	-
RISE [18]	0.0801	0.0964	0.0998	0.0921
Guided BP [35]	**0.0452 (2)**	**0.0526 (2)**	**0.0567 (2)**	**0.0515 (3)**
Grad-Cam [9]	0.1000	0.1512	0.1524	0.1345
Excitation BP [13]	0.0852	0.0926	0.0967	0.0915
Int. Grad. [11]	**0.0347 (1)**	**0.0543 (3)**	**0.0640 (3)**	**0.0510 (2)**
FGVis [10]	-	0.0644	-	-
GNL [72]	**0.0474 (3)**	**0.0471 (1)**	**0.0495 (1)**	**0.0480 (1)**
Proposed	**0.0474 (3)**	**0.0471 (1)**	**0.0495 (1)**	**0.0480 (1)**

Therefore, the average deletion scores for AlexNet, VGG16, and VGG19 on linearly sampled 5,000 images from ImageNet validation split are compared with the existing methods for the quantitative comparison. The results are summarized in Tab. 4.1. As shown, the proposed method achieves state-of-the-art performance in VGG16 and VGG19, and takes the third place in AlexNet, following [11] and [35]. Notably, the proposed method shows stable performance for a range of networks.

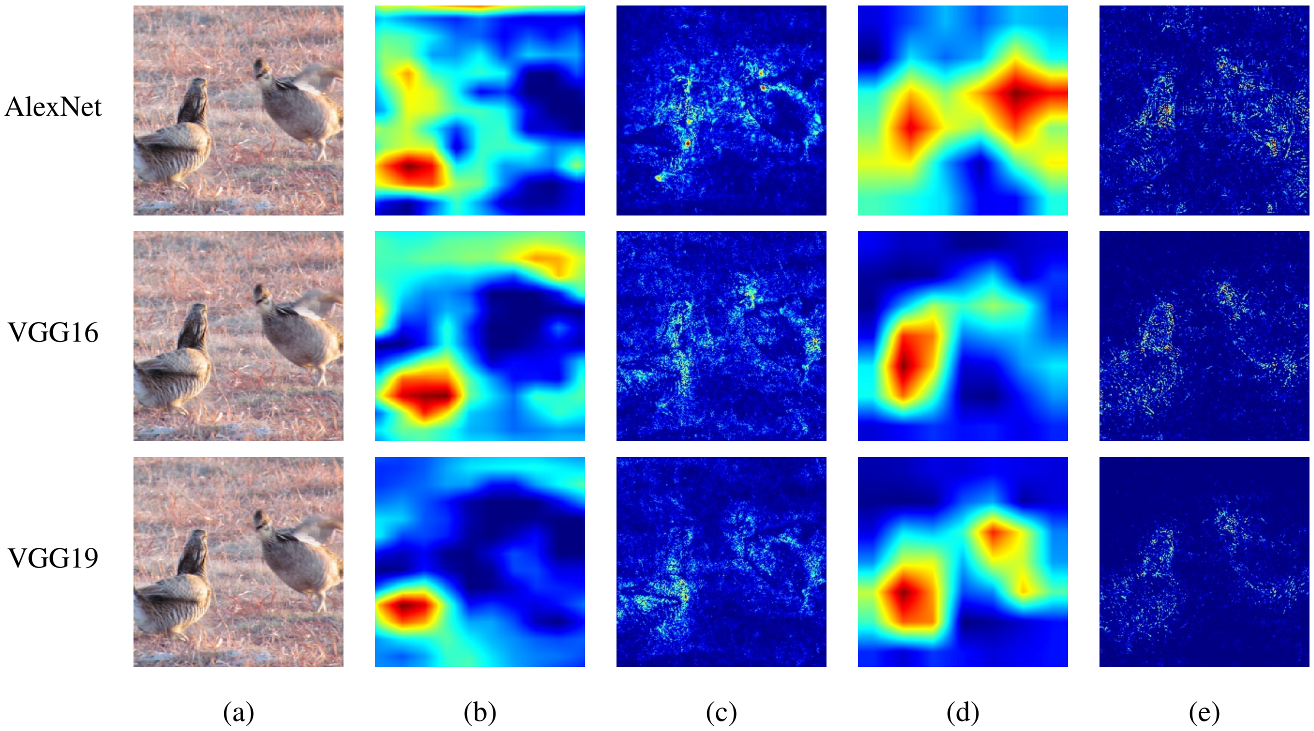


Figure 4.6: Comparison of local explanation masks for three neural networks (AlxeNet (upper), VGG16 (middle), and VGG19 (lower)): (a) Input images (label: *prairie chicken*), (b) Grad-CAM [9], (c) Integrated Gradients [11], (d) Proposed (inverse processes are applied to FC layers) and (e) Proposed (inverse processes are applied to all layers).

4.5 Summary

In this chapter, this dissertation has proposed a framework that explains CNNs with the inverse operations, and used this framework to visualize the inner workings of CNNs. Specifically, based on the physical interpretation of neural networks, this dissertation has proposed constraints on inverse operations and formulated the inverse process as constrained optimization problems. From experiments, it has been demonstrated that the proposed method shows the state-of-the-art performance in terms of deletion metric. Also, qualitative results show that the attributions of conventional methods can be similarly obtained within the proposed framework.

Chapter 5

Layer-wise Approach

5.1 Motivation and Overview

As convolutional neural networks (CNNs) are achieving significant performance gains across numerous computer vision tasks, research interests in understanding CNNs are ever increasing [2, 6, 73]. Obtaining an attribution (an image showing the importance of input pixels to the final decision) of inputs has been considered an effective analysis tool for CNNs among many approaches [3, 28]. Methods for computing attributions can be categorized into perturbation-based and gradient-based approaches. The perturbation-based methods compute attributions by evaluating output changes in response to input perturbations [8,18,32,33], whereas the backpropagation-based approaches compute attributions by propagating prediction-related signals into a backward-pass (computing gradients can be an example) [9,11,13,19,35].

The inverse of predictions were also used to visualize feature vectors, like HOG [21], SIFT [22] and small multilayer perceptrons [23]. However, for CNNs, there are limited studies that exploit inverse operations to understand their working mechanisms owing to the high dimensional parameters and nonlinear operations of CNNs. Furthermore, most of the existing methods are based on heuristics [8, 65] and focused on classification networks [24].

Although these conventional methods have been successfully applied to image classification problems [26, 40], applying them to regression problems is more difficult because they require a global scalar value to measure the output change. The logit value corresponding to a class of interest is usually used as the global scalar value for the classification problem. However, when networks yield multiple predictions simultaneously, such as pixel-wise regression (e.g., super-resolution [27, 74–76]), it is not clear which value should be used as the global scalar value [77]. As a result, using conventional methods to analyze CNNs for various purposes in a common framework is difficult.

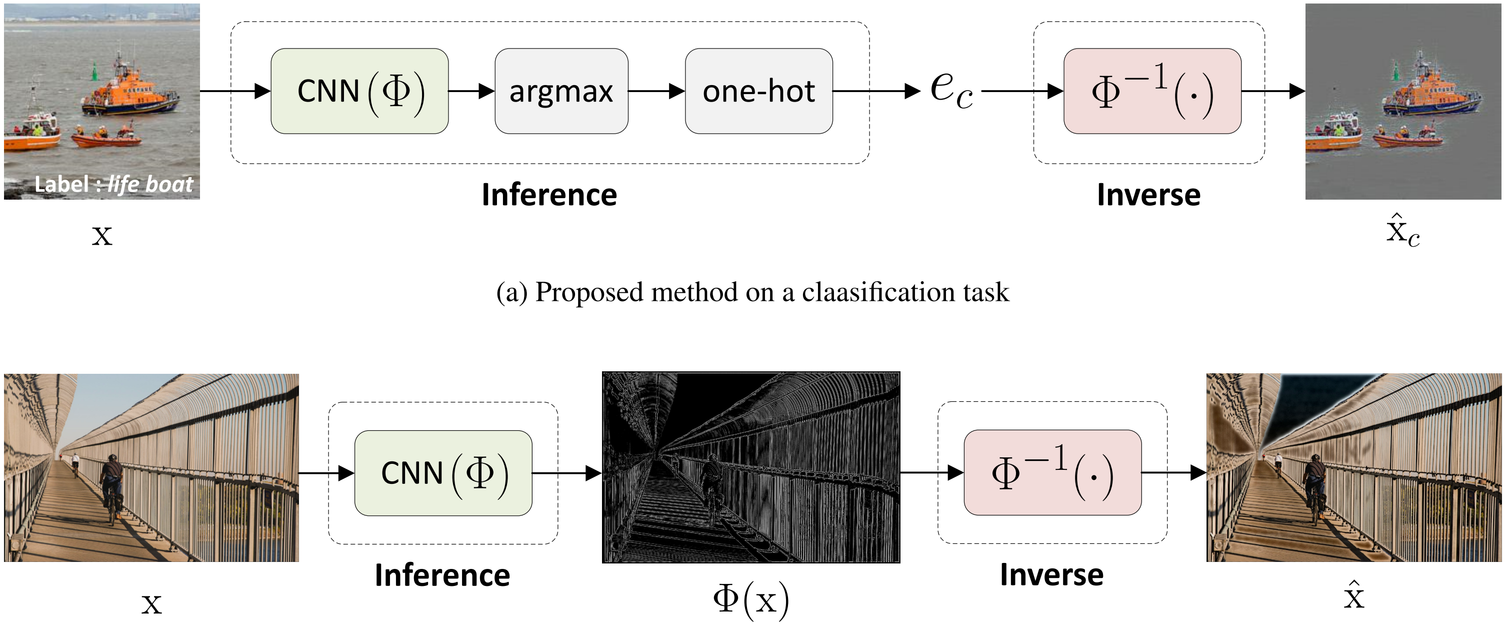
To address these problems, this dissertation propose a new inverse-based approach in which a forward-pass is viewed as a sequential propagation of activations across all network types. In the proposed framework, the activations in lower layers that have caused current output activations are idetified to understand the outputs of neural networks. A network $\Phi(\cdot)$ that outputs y for an input x can be espressed as:

$$\mathrm{y} = \Phi(\mathrm{x}). \tag{5.1}$$

This function is generally not invertible, *i.e.*, $\Phi^{-1}(\mathrm{y}) = \{\mathrm{x}|\Phi(\mathrm{x}) = \mathrm{y}\}$ can have multiple elements. Hence, the goal of the proposed method is to find a human-interpretable $\hat{\mathrm{x}}$,

$$\hat{\mathrm{x}} \in \Phi^{-1}(\mathrm{y}), \tag{5.2}$$

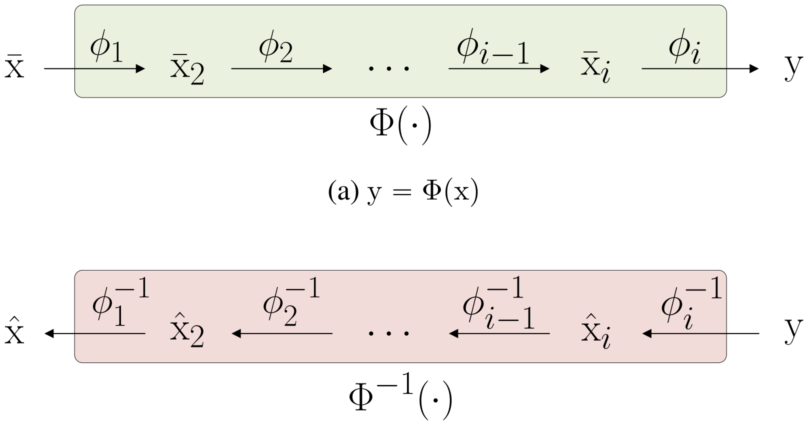
by computing $\Phi^{-1}(\cdot)$ with the following constraints. First, sequential activations caused by $\hat{\mathrm{x}}$ should match those caused by the original input x. This dissertation assumes that neural activations to be physical quantities that indicate the presence (strength) of specific image features. Therefore, when two internal activation patterns are inconsistent, $\hat{\mathrm{x}}$ cannot provide meaningful information for understanding $\mathrm{y} = \Phi(\mathrm{x})$, even though $\Phi(\hat{\mathrm{x}}) \simeq \Phi(\mathrm{x})$. Second, for human-interpretability, the amount of activations should be kept to a minimum as human-interpretability is maximized when the amount of activations is kept to a minimum [3].



(a) Proposed method on a claasification task

(b) Proposed method on a regression task

Figure 5.1: Examples of the proposed inverse approach: (a) classification (ImageNet classification task) and (b) regression (super-resolution). For the classification task, the predicted class ($\Phi^{-1}(e_c)$, where $c = \max_i \Phi(\mathrm{x})$, and e_c is a unit vector with 1 on the position of c) is inversed, and the result highlights all objects belonging to that class. For the regression task, $\hat{\mathrm{x}} \, (\in \Phi^{-1}(\mathrm{y}))$ shows important pixels in estimating the high-frequency details.



(a) $y = \Phi(x)$

(b) $\hat{x} \in \Phi^{-1}(y)$

Figure 5.2: Forward-pass of a CNN ($\Phi(\cdot)$) and its layer-wise inverse process: (a) $y = \Phi(x)$ and (b) $\hat{x} \in \Phi^{-1}(y)(y = \hat{x}_{i+1})$.

Unlike conventional methods, the proposed method yields human interpretable inverse results without a single global score and users can apply the proposed method to classification and regression networks in the same framework as shown in Fig. 5.1. When the proposed inverse process is applied to a class label of interest, it can yield an attribution similar to conventional methods. Similarly, a map showing the contributions of pixels is obtained when the proposed method is applied to the output of a single-image super-resolution network such as VDSR [27].

The majority of existing performance metrics (e.g., insertion and deletion scores) for explanation methods were based on the evaluation of pixel importance and their orders [18, 78]. These metrics have limitations when it comes to reflecting human interpretability and comparing different approaches. To address them, this dissertation develops a novel plot that shows the trade-off between the amount of activations and the rate of class re-identification. The horizontal axis of the plot indicates the amount of pixels used in the visualized attributions, while the vertical axis represents the quality of reproduced output, *i.e.*, its similarity to the original output. Therefore, this plot shows the efficiency with which the method identifies the attributed pixels that produce the same output without other surrounding or unrelated pixels.

The proposed method has been evaluated on VGG16 [26] trained on the ImageNet classification [1] and VDSR [27] for single image super-resolution (SISR). Experimental results have shown that the proposed approach successfully visualizes the input and output relationship and yields the best output-reconstruction performance for a given amount of pixels in VGG16. In the experiment with VDSR, it has shown that the inverse of the residual map from VDSR using the proposed method yields a similar super-resolution result from the original input. The inverse of residual also reveals that VDSR focuses on high-frequency bands and that this high-frequency selectivity may degrade the performance of super-resolution (SR).

In short, this dissertation makes the following contributions.

- Layer-wise inverse technique developed that explains the predictions of CNNs, for both classification and regression problems.
- A graphical plot is proposed that shows the trade-off between the amount of activations and the output-reconstruction quality.
- Human interpretable inverses for VGG16 trained for the ImageNet task is computed, as an example of the classification task.
- The inverses of a regression CNN, specifically an SR CNN, is computed and it is discovered that the SR CNN focuses on certain frequency bands, potentially degrading SR performance.

5.2 Formulation of the Proposed Inverse Approach

In a forward-pass of CNNs, the input signal makes a sequential propagation of neuron activations. From this viewpoint, the inverse of given neuron's activation to lower layers should be computed to analyze this activation. For the presentation of the proposed method, a neural network is considered as consisting of n units, where each unit $\phi_i(\cdot)$

has neural activations as its input and output,

$$\mathrm{y} = \Phi(\mathrm{x}) = (\phi_1 \circ \phi_2 \circ \cdots \circ \phi_n)(\mathrm{x}), \tag{5.3}$$

where $\circ$ indicates the function composition. The goal of the proposed method is to develop an optimization method to compute the inverse of y:

$$\hat{\mathrm{x}} \in \left(\phi_n^{-1} \circ \phi_{n-1}^{-1} \circ \cdots \circ \phi_1^{-1}\right)(\mathrm{y}), \tag{5.4}$$

which has human-interpretability with following requirements.

5.2.1 Activation range

Activations in convolutional layers indicate the strengths of corresponding features, and negative activation should not happen in the inversion process when ReLU or softmax are used as activation functions [62]. Similarly, neuron activations in the inverse process should not exceed the activation level of its corresponding forward-pass activation [10]. That is, when looking for causes (lower layer activations) that lead to a specific result (higher layer activations), possible candidates for explanations are activations that actually occurred during the forward-pass. This observation is represented with the *activation range constraint*:

activation range constraint: When a forward-pass is given by $\bar{\mathrm{y}} = \phi(\bar{\mathrm{x}})$, the domain and co-domain of its inverse is $\{y | 0 \preceq y \preceq \bar{y}\}$ and $\{x | 0 \preceq x \preceq \bar{x}\}$, respectively, where $\preceq$ is an element-wise inequality.

5.2.2 Minimal activation

Furthermore, using a minimal amount of activations is preferred for human-interpretability [3, 79, 80]. However, unlike the *activation range* constraint, this is a trade-off between output-reconstruction quality and human-interpretability rather than a hard constraint. Hence, this observation is implemented as:

minimal activations constraint: Regularization terms are used to penalize the amount of activations.

5.2.3 Linear approximation

Although it is sometimes preferable to deal directly with non-linear units (without approximations), their linear approximations are sometimes required. In this case, the proposed method used the linear approximation around operating points, which allows finding inverses without using off-neurons. The linear approximation converges to multiplication with a matrix filled with “0” or “1”, and this observation is summarized as:

linear approximation constraint: When necessary, the proposed method approximate a forward process with a linear function, around the operating points in the forward-pass.

5.2.4 Layer-wise inverse

To find inverses satisfying the proposed *activation range constraint*, all activations should be considered during the optimization. To handle a modern huge network at the same time is a infeasible task, and therefore a layer-wise inverse method is proposed as shown in Fig. 5.2. To be precise, when a forward-pass consists of a set of $\{\phi_i(\cdot)\}$ that generates features through an activation function (e.g. ReLU), the inverse of $\hat{\mathrm{x}}_{i+1}$ is given by

$$\hat{\mathrm{x}}_i = \arg\min_{\mathrm{x}_i} \left(\|\phi_i(\mathrm{x}_i) - \hat{\mathrm{x}}_{i+1}\|_2 + \lambda R(\mathrm{x}_i)\right) \tag{5.5}$$

which is subject to $0 \preceq \mathrm{x}_i \preceq \bar{\mathrm{x}}_i$, where $\bar{\mathrm{x}}_i$ is an activation in the forward-pass, and $R(\cdot)$ is a regularization term that penalizes the large amount of activations. In theory, λ should be 0 for over-determined systems. However, determining whether a system is over-determined is difficult, especially when there are a large number of weights. Therefore, the regularization term is used for convolutional and fully connected layers. Finally, the inverse of the whole process in Eq. (5.4) was given by the recursive use of Eq. (5.5).

Algorithm 2 Gradient projection algorithm (GPA) to compute Eqs. (5.5) and (5.12), where $\sigma(\cdot)$ is the standard deviation for the given sequence, and $h(J(\cdot), \mathrm{x})$ is the maximum diagonal element of Hessian of $J(\cdot)$ near x.

```
procedure GPA($\bar{\mathrm{x}}_i, \hat{\mathrm{x}}_{i+1}, \phi_i(\cdot)$)
    if $\phi_i(\cdot)$ has a bias then
        $\hat{\mathrm{x}}_{i+1} \leftarrow \frac{\|\bar{\mathrm{x}}_i\|_2}{\|\hat{\mathrm{x}}_{i+1}\|_2}\hat{\mathrm{x}}_{i+1}$
    end if
    $J(\cdot) = \|\phi_i(\cdot) - \hat{\mathrm{x}}_{i+1}\|_2 + \lambda R(\cdot)$
    $\mathrm{x}_o =$ blurred version of $\bar{\mathrm{x}}_i$
    $\mathrm{x}_1 = \mathrm{x}_o$
    $\beta_o = 0$
    $r_o = 1$
    while $\sigma(\{c_n\}) < \tau$ do
        $\mathrm{t}_n \leftarrow \mathrm{x}_n + \beta_n(\mathrm{x}_n - \mathrm{x}_{n-1})$
        $c_n \leftarrow J(\mathrm{t}_n)$
        $\mathrm{z}_n \leftarrow \mathrm{t}_n - r_n \nabla_{\mathrm{t}_n} c_n$
        $\mathrm{x}_n \leftarrow$ clipping $\mathrm{z}_n$ with $0 \leq z_n \leq \bar{\mathrm{x}}_i$
        $\beta_n \leftarrow \frac{n-1}{n+2}$
        $r_n \leftarrow \min\left(r_{n-1}, \frac{0.5}{h(J(\cdot), \mathrm{x}_n)}\right)$
    end while
    return $\mathrm{x}_n$
end procedure
```

5.3 Details of inverse computation

In this section, the details of computing layer-wise inverses is presented. The proposed method deals with four types of layers in CNNs: convolution block (linear part) + ReLU, max-pooling layer, fully connected block (linear part) + ReLU, and fully connected block (linear part) + Softmax. Because of the large number of parameters, the proposed method optimizes Eq. (5.5) using the gradient projection algorithm (GPA) [68], when a closed-form solution is not available. The GPA method is summarized in Algorithm 2.

5.3.1 Convolution block (linear part) + ReLU

Because the output of a linear convolutional block can have negative values, the inverse of the linear block alone is not desirable in the proposed framework. Instead, this dissertation considers a linear convolutional block and non-linear unit as a single block and performs the inverse based on Eq. (5.5).

Activation regularization

While computing the inverse, regularization terms are often used to address the ill-posed nature of inverses [24, 32, 33, 44]. These regularization terms are expected to have a role in computing interpretable inverse result by imposing natural image priors (e.g., minimizing the total variation of inverse results) [24]. However, the input to a layer is another activation from the previous layer, not the images, with the exception of $\phi_1(\cdot)$. Therefore, a regularization method for activations based on the local connectivity of convolution is developed: when the activation of a given position is small, the corresponding local region's activations in lower layers should be small.

For the presentation of the proposed regularization term, this dissertation assume that the proposed method compute the inverse of $\hat{\mathrm{x}}_{i+1}$. Then, r_{i+1} is computed using (1) summing the activations of $\hat{\mathrm{x}}_{i+1}$ along a channel axis and (2) linearly normalizing the map with $\max \hat{\mathrm{x}}_{i+1}$, so that $\mathrm{r}_{i+1} \in [0,1]^{W \times H}$, where W and H are the width and height of the map, respectively. Intuitively, s_{i+1} indicates the region-of-interest map for $\hat{\mathrm{x}}_{i+1}$. Because of the local connectivity of convolutional layers, it is desirable to penalize the activations outside this region. Hence, the activation regularization term is defined as

$$R(\mathrm{x}) = G(\mathrm{x}, \hat{\mathrm{x}}_{i+1}) = \|\mathrm{x} \odot (1 - \mathrm{r}_{i+1})\|_2 \tag{5.6}$$

where $\odot$ is an element-wise product. The r_{i+1} should be up-sampled to the size of x if its resolution is smaller than x.

5.3.2 Max-pooling layer

The inverse of max-pooling operation is under-determined, i.e., there are many solutions satisfying the first part of Eq. (5.5). Thus, the inverse of max-pooling becomes the solution of optimization problem that minimize the regularization term while satisfying $\phi_i(\mathrm{x}_i) = \hat{\mathrm{x}}_{i+1}$. In this case, the proposed method approximate the forward-pass with the matrix multiplication $\phi_i(\mathrm{x}_i) = M_i \mathrm{x}_i$. As an example, for a 2×2 max-pooling without overlaps, the forward operation for each 2×2 block can be represented by the matrix multiplication(here, the last element is assumed as the maximum):

$$\phi\left(\begin{bmatrix} x_{i,0} \\ x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{bmatrix}\right) = \begin{bmatrix} 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{i,0} \\ x_{i,1} \\ x_{i,2} \\ x_{i,3} \end{bmatrix}. \tag{5.7}$$

Since the optimization problem for an under-determined system is

$$\hat{\mathrm{x}}_i = \arg\min_{\mathrm{x}_i} G(\mathrm{x}_i, \hat{\mathrm{x}}_{i+1}) \tag{5.8}$$

subject to $\phi_i(\mathrm{x}_i) = M_i \mathrm{x}_i = \hat{\mathrm{x}}_{i+1}$, it is the same as the pseudo inverse problem that finds the minimum norm solution. Hence, it is possible to have a closed-form solution

$$\hat{\mathrm{x}}_i = (M_i)^{+} \hat{\mathrm{x}}_{i+1}, \tag{5.9}$$

where $(M_i)^+$ is the Moore-Penrose inverse of M_i [81]. Note that the *activation range* constraint is automatically satisfied for the solution of Eq. (5.9). In fact, this result is the same as [8], which was designed intuitively.

5.3.3 Fully connected block (linear part) + ReLU

The inverse of a fully connected layer is similar to that of convolutional blocks, with the exception of the regularization term in Eq. (5.6). Because fully connected layers do not have local connectivity properties, only the L_2 norm described as

$$R(\mathrm{x}_i) = \|\mathrm{x}_i\|_2, \tag{5.10}$$

is used for the regularization term and the GPA algorithm is applied for its optimization.

5.3.4 Fully connected block (linear part) + Softmax

For the inverse of top layers in the classification problem, it is need to deal with softmax layers. The proposed method focus on output-reconstruction and do not use a regularization term in this context. Specifically, if a fully connected block is described as $\mathrm{w} = W\mathrm{x}_i$, and a softmax layer is described as

$$\mathrm{s} = \left[\frac{\exp(\mathrm{w}^1)}{\sum \exp(\mathrm{w}^i)} \quad \frac{\exp(\mathrm{w}^2)}{\sum \exp(\mathrm{w}^i)} \quad \cdots \quad \frac{\exp(\mathrm{w}^C)}{\sum \exp(\mathrm{w}^i)} \right]^\top \tag{5.11}$$

where w^i is the i-th element in a vector w, the inverse of the final classification result (i.e., a unit vector $\mathrm{e_c}$) becomes

$$\begin{aligned} \arg\min_{\mathrm{s}_i} \|\mathrm{z} - \mathrm{e}_c\|_2 &= \arg\max_{\mathrm{x}_i} \left(\frac{\exp(\mathrm{w}^c)}{\sum \exp(\mathrm{w}^i)} \right) \\ &= \arg\min_{\mathrm{x}_i} \left(\sum \exp(\mathrm{w}^i - \mathrm{w}^c) \right) \\ &\simeq \arg\min_{\mathrm{x}_i} \left(1 + \max_{i \neq c} (\mathrm{w}^i - \mathrm{w}^c) \right) \\ &= \arg\min_{\mathrm{x}_i} \left(\max_{i \neq c} (\mathrm{w}^i) - \mathrm{w}^c \right). \end{aligned} \tag{5.12}$$

Subsequently, the above problem is optimized by the GPA algorithm with $0 \preceq \mathrm{x}_i \preceq \bar{\mathrm{x}}_i$.

5.4 Application to the ImageNet classification task

The proposed method is applied to the ImageNet classification task. Because the proposed method differs from conventional approaches in several aspects, this dissertation first discuss possible evaluation methods. Then, the hyper-parameter λ in Eq. (5.5), which achieves a trade-off between the reconstruction and interpretability, is selected. Finally, the proposed method is applied to VGG16 [26] and compared the result to those obtained using conventional methods.

5.4.1 Evaluation of output-reconstruction in terms of input-simplicity

The proposed method aims to obtain a human interpretable inverse $\hat{\mathrm{x}}$, and hence experimental results are evaluated from two viewpoints: (1) output-reconstruction performance (whether $\Phi(\hat{\mathrm{x}})$ is similar to the original prediction result) and (2) human interpretability (whether $\hat{\mathrm{x}}$ is human interpretable). That is, if an algorithm can visualize the inverse or find an attribution for a given prediction, it should be able to generate a simpler input that yields a similar result to the original prediction [45,82]. Otherwise, it is unclear whether the algorithm successfully localizes regions used for CNN prediction. The proposed method naturally provides such an input, i.e., $\hat{\mathrm{x}}$. However, most conventional methods focus on attributions, and images corresponding to $\hat{\mathrm{x}}$ are not directly available.

This dissertation define an attribution-weighted input as

$$\tilde{\mathrm{x}} = \alpha \odot \mathrm{x} \tag{5.13}$$

where $\alpha \in [0, 1]^{W \times H}$ is a linearly-scaled attribution and x is an original input. Because attributions are designed to have large values for important pixels, the attribution-weighted input $\tilde{\mathrm{x}}$, which can also be considered an important-region masked input, should yield a similar result to the original. Precisely, let $I(\cdot, \cdot)$ as $I(\mathrm{x}_1, \mathrm{x}_2) = 1$ if x_1 and x_2 give the same classification results, and $I(\mathrm{x}_1, \mathrm{x}_2) = 0$ otherwise. The output-reconstruction performance is calculated by $I(\mathrm{x}, \hat{\mathrm{x}})$ for the proposed method and $I(\mathrm{x}, \tilde{\mathrm{x}})$ for conventional methods that have only attributions.

However, because there is a trade-off between output-reconstruction performance and input simplicity, the output-reconstruction performance alone cannot be used in the evaluation [24, 78]. To reflect this trade-off, $\|\alpha\|_1$, the L_1 norm of α, is considered as a measure of the complexity of $\tilde{\mathrm{x}}$. When α has binary values, $\|\alpha\|_1$ is the amount of pixels used in a new input in Eq. (5.13). Even when α has continuous values, this value can be considered as the amount of input pixels. In the proposed approach, attributions were not directly available, and α is obtained with element-wise division of $\hat{\mathrm{x}}$ by x according to Eq. (5.13) for a fair comparison with other methods.

For the brevity of the presentation, $\|\alpha\|_1'$ is considered as an area-normalized value. When $\|\alpha\|_1' = 1$, output-reconstruction becomes 1.0 (the same result as the original prediction). When $0 \leqslant \|\alpha\|_1' \leqslant 1$, it shows a trade-off between good output-reconstruction performance and simpler inputs.

5.4.2 Deletion and insertion scores

Deletion and insertion scores are often employed for the evaluation of attribution-computing methods [10, 18, 44]. Both scores compute the area under the curve representing the probability of being an original class for synthesized inputs. Synthesized inputs are obtained by deleting/inserting pixels based on their importance (attributions). Therefore, a lower deletion score and a higher insertion score indicate that current attributions accurately reflect the importance orders of pixels.

Although deletion and insertion scores are reasonable measures for evaluating the quality of importance orders, they have drawbacks in representing human interpretability in at least two aspects. First, some algorithms can achieve high scores by selecting scattered pixels (rather than regions). It is known, for example that strong activations can be obtained with noise-like sparse inputs, and better deletion scores can be achieved only with *faulty evidence* [34, 44, 83]. Second, both measures focus only

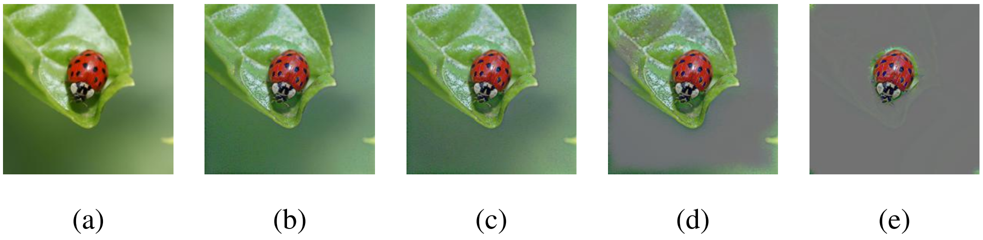


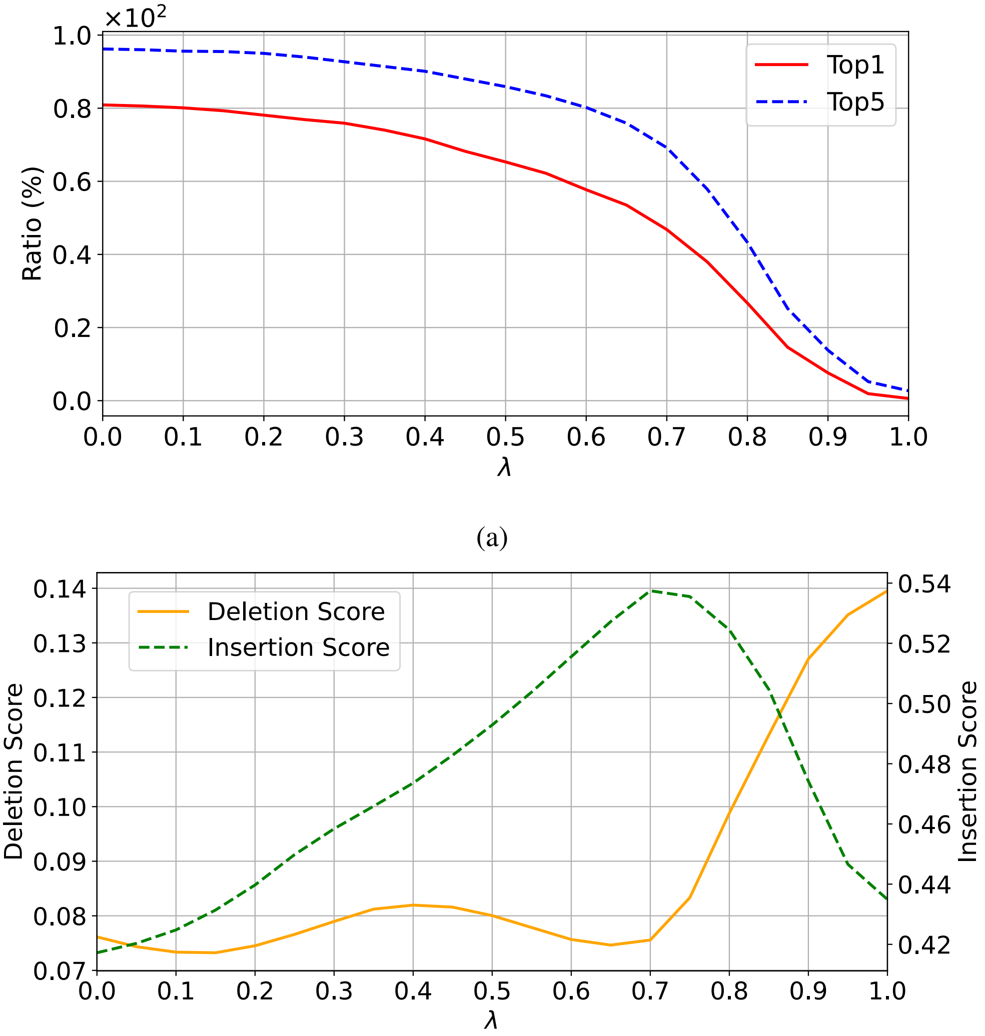
Figure 5.3: Illustration of $\hat{x}_c$ ($c = \textit{ladybug}$) for several λ values on VGG16.: (a) Input (x), (b) $\hat{x}_c$ with λ = 0.0, (c) $\hat{x}_c$ with λ = 0.2, (d) $\hat{x}_c$ with λ = 0.4, and (e) $\hat{x}_c$ with λ = 0.8. As the value of λ increases, $\hat{x}_c$ concentrates on the target object (*ladybug*) and loses the background information that could assist the prediction.

on the original class, with no consideration given to whether the most likely class has shifted from the original prediction to another.

5.4.3 Selection of the regularization term weight

As shown in Fig. 5.3, λ in Eq. (5.5) is a trade-off parameter between the reconstruction quality and human-interpretability. To see the trade-off between them quantitatively, graphs for various λ values are plotted in Fig. 5.4. The graph clearly shows a sharp drop in output-reconstruction performance around $\lambda = 0.7$. As illustrated in Fig. 5.3, the regularization term penalizes the amount of activations, and the support of valid regions decreases as λ increases. This has no affect on output-reconstruction performance as long as the region contains all relevant objects. However, the output-reconstruction performance starts to drop as the support becomes smaller than objects. Thus, this dissertation assume that a reasonable λ can be chosen by locating a sharp drop point.

Interestingly, this λ yields the best deletion and insertion scores. For this λ, activations are only allowed in object regions, and the classification results become sensitive to the insertion and deletion of pixels. Based on this result, this dissertation conclude that, insertion and deletion scores are adequate measures of human-interpretability, despite their limitations.



(a)

(b)

Figure 5.4: The horizontal axis represents λ in Eq. (5.5) in both figures, and the vertical axis of (a) indicates the ratio of yielding the same class to the original or among top 5 predictions. The vertical axis of (b) indicates (left) Deletion score and (right) Insertion score (see the text for details).

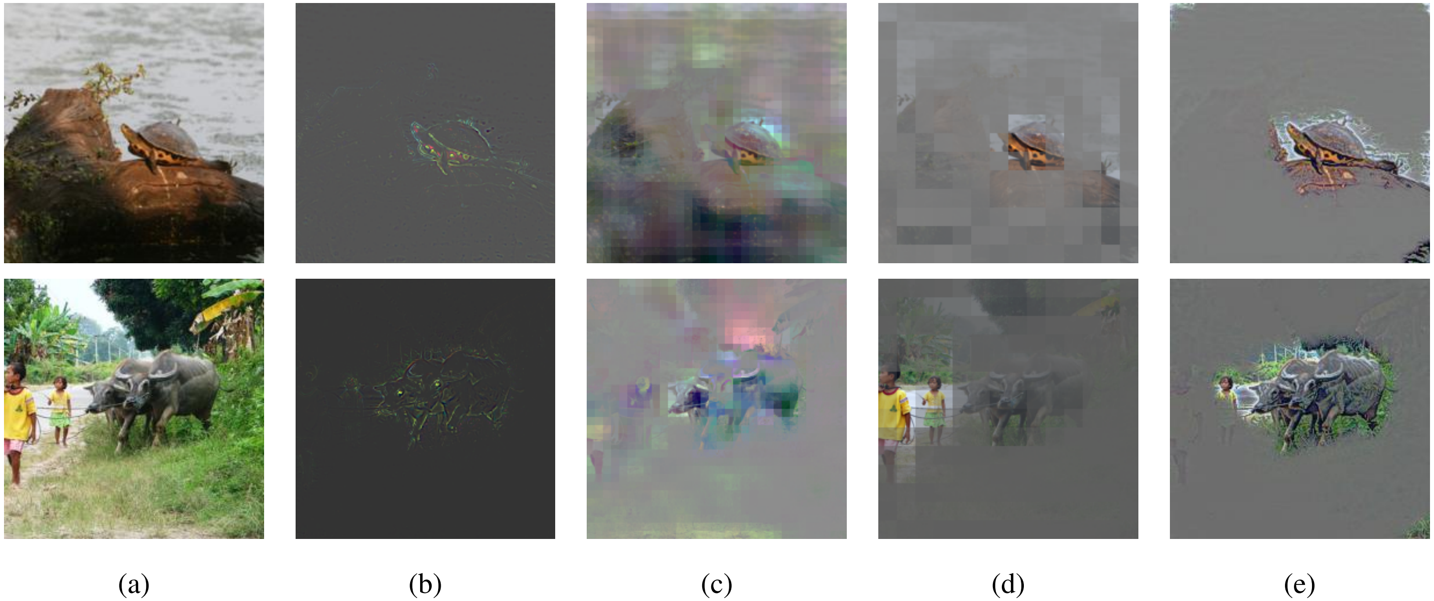


Figure 5.5: Comparison of input-reconstruction results: (a) Input image (top: *terrapin*, bottom: *water buffalo*), (b) Guided Backprop [35], (c) Occlusion based method [8], (d) Grad-CAM [9], and (e) Proposed. The results of the proposed method are obtained with $\lambda = 0.7$. For conventional methods, images are computed by Eq. (5.13).

Table 5.1: Deletion/insertion scores, output-reconstruction performance (%), and processing time (s) on the validation split of ImageNet.

Mehods	Deletion (↓)	Insertion (↑)	Top 1 (↑)	Top 5 (↑)	Time (↓)
Guided Backprop [35]	0.051	**0.550**	1.9	5.2	0.008
Gradients × Input	0.068	0.377	0.0	0.4	0.006
Deep Lift [19]	0.052	0.466	0.1	5.8	0.028
Integrate Gradients [11]	0.054	0.417	0.0	0.5	0.055
Gradient SHAP	**0.050**	0.385	0.0	0.1	0.022
Occlusion [8]	0.095	0.485	9.7	21.9	17.065
Grad-CAM [9]	0.163	0.434	27.4	45.1	**0.005**
Proposed	0.075	0.537	**46.8**	**69.2**	6.725

5.4.4 Comparison to Existing Methods

The proposed method is compared to conventional methods, and the results are summarized in Tab. 5.1. The images shown in Fig. 5.5 are obtained with Eq. (5.13). When necessary, α is linearly normalized to $[0, 1]$ and attributions are resized with the nearest-neighbor interpolation.

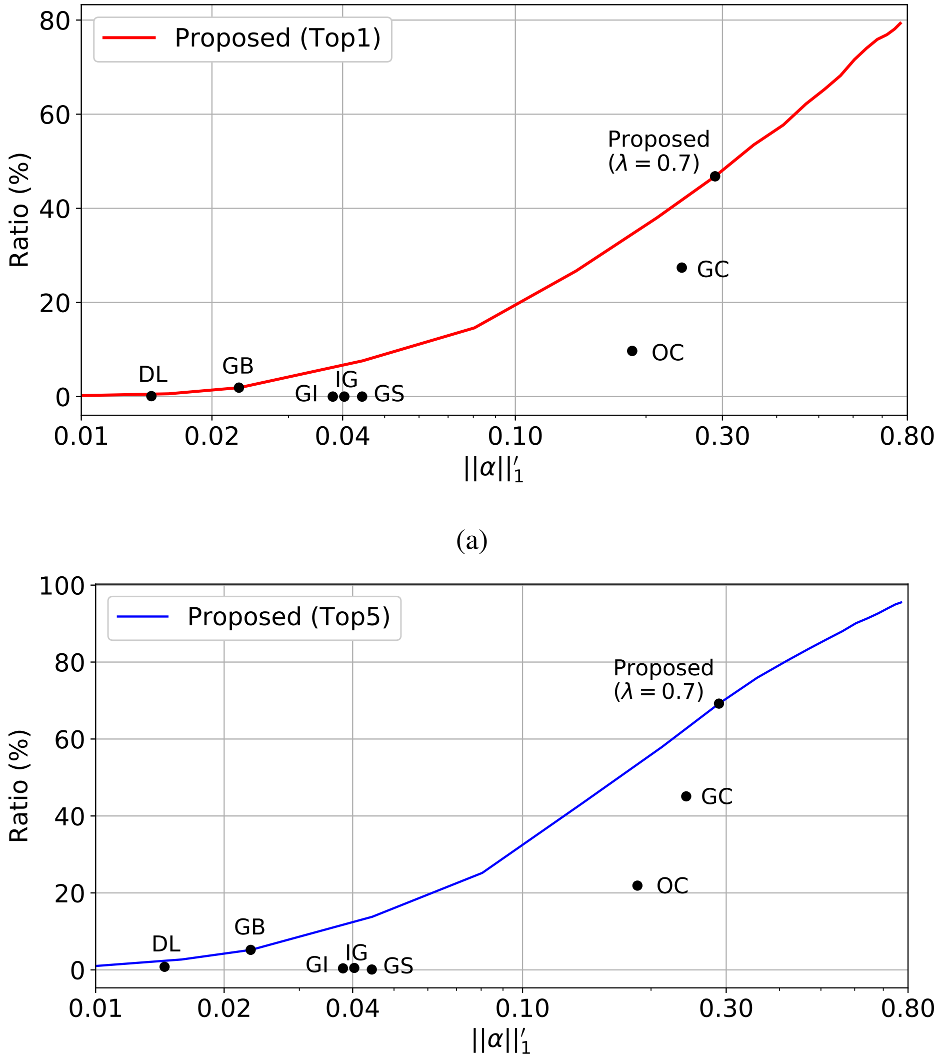
As shown in Figs. 5.3 and 5.5, the proposed method yields human-interpretable results. Because of the activation regularization term that takes into account local connectivities, the proposed method generates new images with interpretable regions. Tab. 5.1 summarizes the average value of output-reconstruction performance, deletion/insertion scores, and average processing time computed on the $1,000$ linearly sampled images from the validation split of ImageNet. The table shows that the proposed method outperforms conventional methods in terms of output-reconstruction performance within a reasonable processing time, implying that the proposed method faithfully and efficiently computes the inverse of the forward-pass.

Overall, the proposed method yields lower deletion scores than the four conventional methods. However, as shown in Fig. 5.5, conventional algorithms lack the ability to provide reasonable input images, resulting in very low output-reconstruction performance. When the insertion and deletion metrics are considered together, the proposed method compares favorably to conventional methods. Fig. 5.5 also show that, when compared to other methods, $\hat{\mathrm{x}}$ of the proposed method successfully focuses on target objects. The bottom row (label: *water buffalo*) shows that the proposed method correctly localizes multiple targets.

5.4.5 Output-reconstruction versus input-simplicity plot

As an alternative evaluation method, this dissertation proposes the plot shown in Fig. 5.6, in which the vertical axis indicates output-reconstruction performance, and the horizontal axis represents $\|\alpha\|_1'$. Because the proposed method has a trade-off parameter λ, its results are represented by curves, while conventional methods are represented by dots. As shown, there are two groups: the left-hand-side group (DL, GB, GI, GS, and IG methods) tries to explain CNN outputs with a very little amount of pixels, and the results are naturally focused on object boundaries as shown in Fig. 5.5. By contrast, the right-hand-side group (GC, OC, and the proposed method with $\lambda = 0.7$) attempts to explain the result with approximately 30% of the input pixels so that a human can easily recognize objects and output-reconstruction performance is also high.

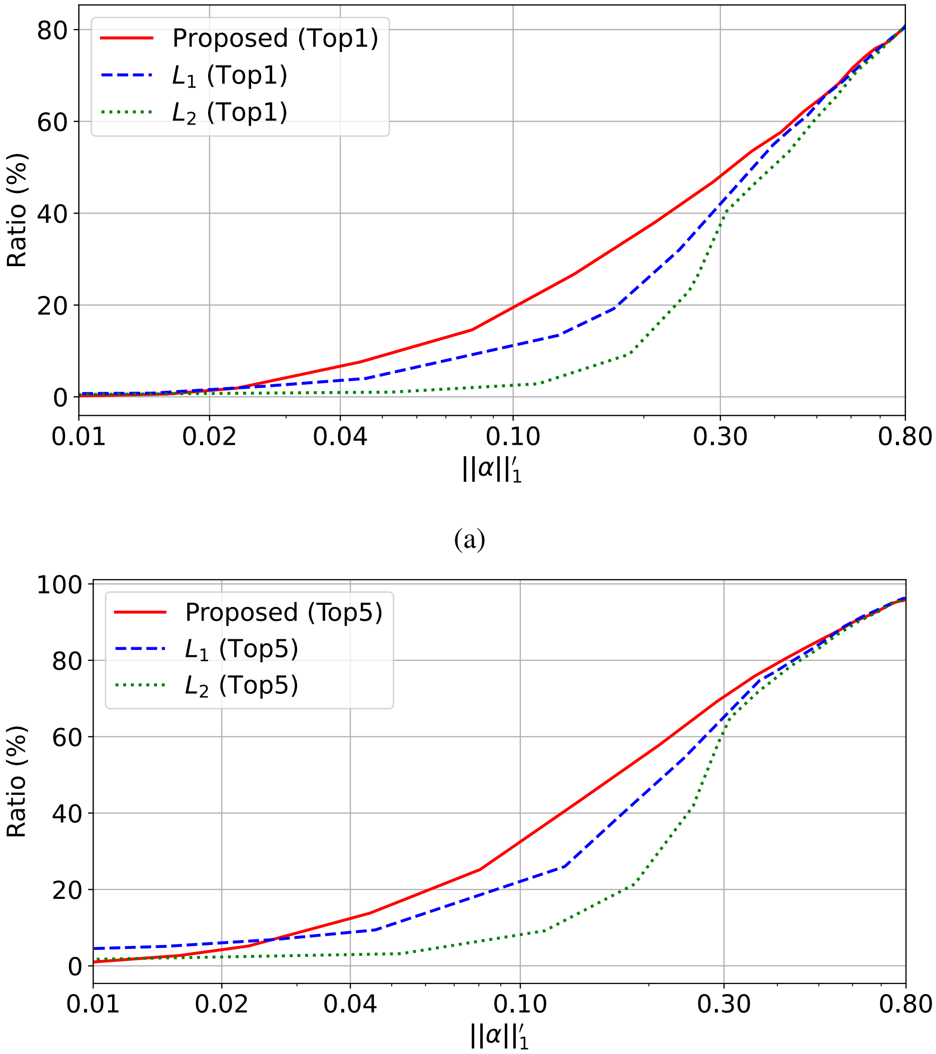
This dissertation suggests that the proposed output-reconstruction versus input-simplicity plot provides a two-dimensional view of algorithm performance that will aid in comparisons.



(a)

(b)

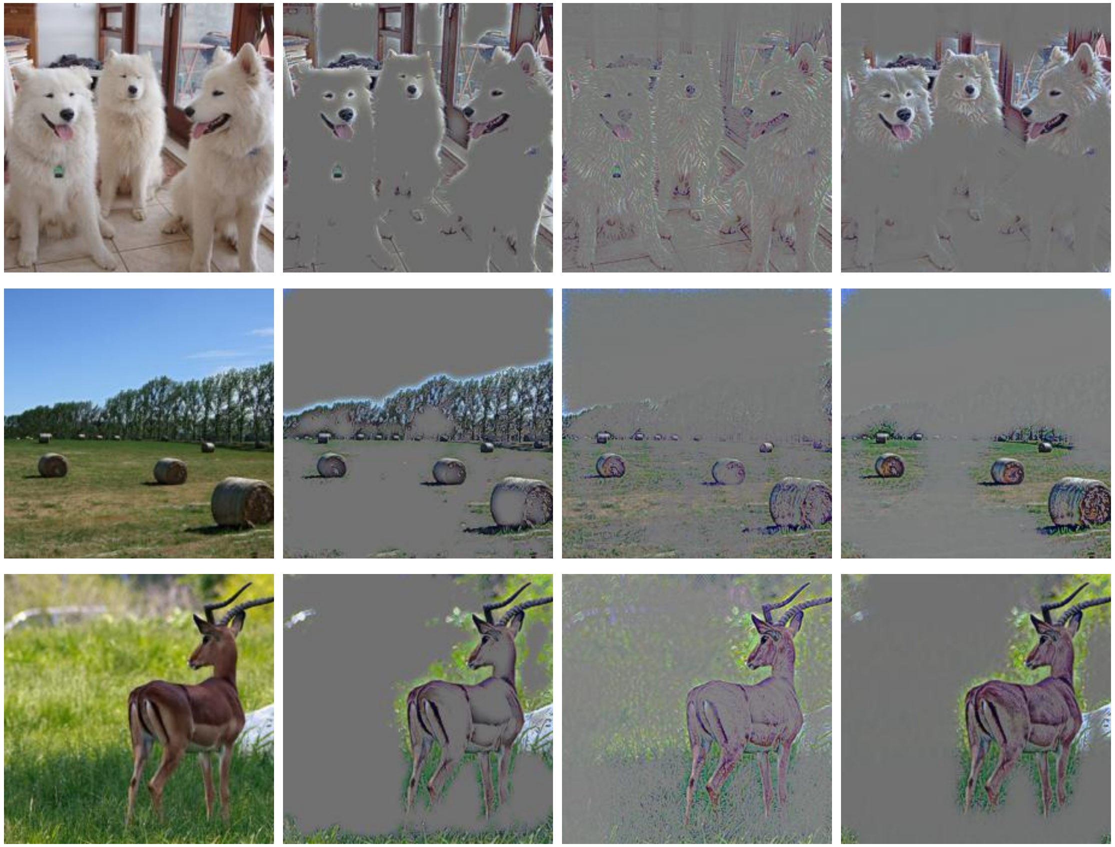
Figure 5.6: The horizontal axis represents average $\|\alpha\|_1'$ on test images, and the vertical axis indicates the ratio of yielding (a) the same class to the original or (b) the same one among top 5 predictions. The results from gradients times inputs (GI) [84], DeepLift (DL) [19], Guided Backprop (GB) [35], Integrated Gradients (IG) [11], Grad-CAM (GC) [9], Gradient SHAP (GS) [85] and occlusion based method (OC) [8] are denoted as points in (e) and (f).



(a)

(b)

Figure 5.7: The output reconstruction performance of $\hat{\mathrm{x}}_c$ for L_1, L_2, and the proposed activation regularization terms on VGG16 is plotted. The horizontal axis represents average $\|\alpha\|_1'$ on test images, and the vertical axis indicates the ratio of yielding (a) the same class to the original or (b) the same class among the top 5 predictions.



(a) (b) (c) (d)

Figure 5.8: Illustration of $\hat{\mathrm{x}}_c$ with L_1, L_2, and the proposed activation regularization on VGG16: (a) Input image (top: *samoyed*, middle: *hay*, bottom: *impala*), (b) $\hat{\mathrm{x}}_{\mathrm{c}}$ with L_1, (c) $\hat{\mathrm{x}}_{\mathrm{c}}$ with L_2, and (d) $\hat{\mathrm{x}}_{\mathrm{c}}$ with proposed activation regularization.

5.4.6 Ablation study of the activation regularization

The performance of the activation regularization term is evaluated along with conventional L_1 and L_2 norm regularizers. Fig. 5.7 shows the output-reconstruction versus the input-simplicity plot for $\|\alpha\|_1' \in [0.0, 0.8]$ ($\|\alpha\|_1' = 0.8$ is obtained when $\lambda = 0.0$). As shown in Fig. 5.7, the activation regularization term shows the best top 1 output-reconstruction performances for all $\|\alpha\|_1'$ values. For the top 5 output-reconstruction performances, the activation regularization term shows slightly poor performance compared to the L_2 norm regularizer for $\|\alpha\|_1' \in [0.0, 0.3]$. However, the activation regularization term shows the best performance for $\|\alpha\|_1' \in [0.3, 0.8]$.

For qualitative comparison, this dissertation visualized $\hat{\mathrm{x}}_\mathrm{c}$ obtained by three regularization terms in Fig. 5.8. The λ for the activation regularization term was set as 0.7, and λ for L_1 and L_2 was set to have similar $\|\alpha\|_1'$ values. As shown, $\hat{\mathrm{x}}_\mathrm{c}$ with the activation regularization term describes target objects accurately while suppressing irrelevant background objects. Fig. 5.8 also shows the characteristics of each regularization method. The result from L_1 focuses on selecting relevant pixels, whereas the results from L_2 degrade the intensities of irrelevant pixels. The proposed activation regularization term gives similar results to the L_1 case. However, it preserves textural details of target objects in input images. It seems that the element-wise product and L_2 penalty in Eq. (5.6) balance the amount of pixels and their intensities in $\hat{\mathrm{x}}_c$.

5.5 The inverse of single image super-resolution network

As an example of analyzing a regression task, this dissertation applied the proposed framework to VDSR [27], a well-known CNN architecture for the SISR.

5.5.1 Experimental setting

VDSR [27] consists of 20 convolution layers with ReLU activation. VDSR predicts the residual ($\Phi(\mathrm{x})$) from the luminance channel of a low resolution (LR) image (x).

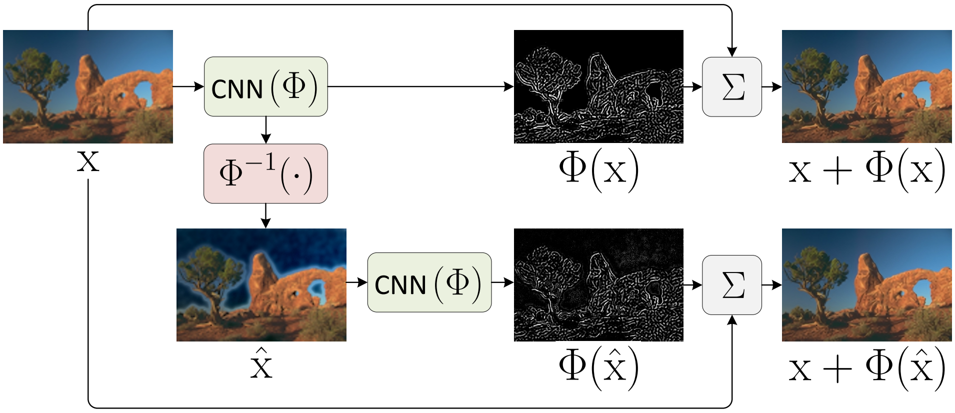


Figure 5.9: Experimental setting for VDSR, illustrating the computation of the residual image ($\Phi(x)$) from the input LR image (x), HR image ($x + \Phi(x)$), the inverse of residual image ($\hat{x}$), and HR image from the inverse result ($x + \Phi(\hat{x})$).

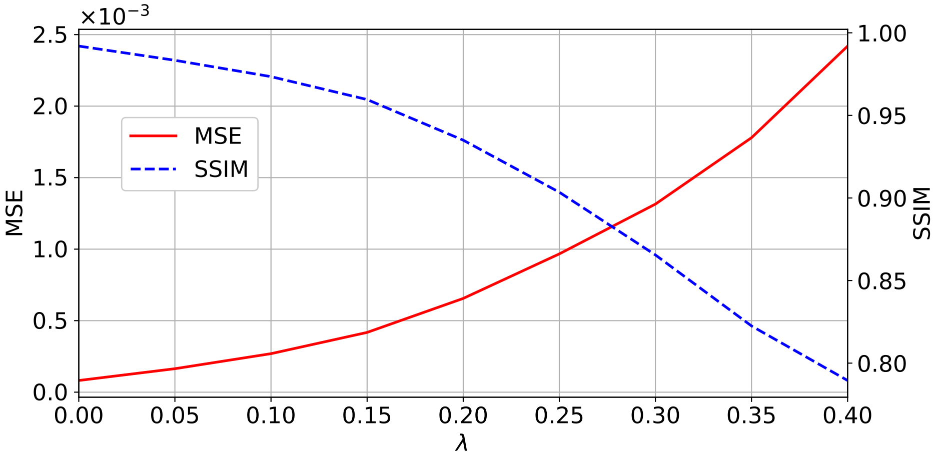


Figure 5.10: Experiment on Set5 with scale factor $\times 2$ for $\lambda \in [0.0, 0.4]$. The horizontal axis represents λ in Eq. (5.5), the left vertical axis represents the mean square error (MSE) between $\Phi(x)$ and $\Phi(\hat{x})$, and the right vertical axis is structural similarity (SSIM) between $x + \Phi(x)$ and $x + \Phi(\hat{x})$.

Table 5.2: Notations for pairs used in SR experiments. ('g' means the ground truth HR image.)

Name	Distance Pair
d_A	$(\mathrm{x}, \hat{\mathrm{x}})$
d_B	$(\Phi(\mathrm{x}), \Phi(\hat{\mathrm{x}}))$
d_C	$(\mathrm{x} + \Phi(\mathrm{x}), \mathrm{x} + \Phi(\hat{\mathrm{x}}))$
d_D	$(\mathrm{x} + \Phi(\mathrm{x}), \mathrm{g})$
d_E	$(\mathrm{x} + \Phi(\hat{\mathrm{x}}), \mathrm{g})$

Subsequently, output high resolution (HR) images are given by the sum of residuals and LR images (i.e., $\mathrm{x} + \Phi(\mathrm{x})$), as shown in Fig. 5.9. As in [27], colors of $\hat{\mathrm{x}}$, $\mathrm{x} + \Phi(\mathrm{x})$ and $\mathrm{x} + \Phi(\hat{\mathrm{x}})$ are reconstructed from chrominance channels of x for color visualization.

5.5.2 Selection of the regularization term weight

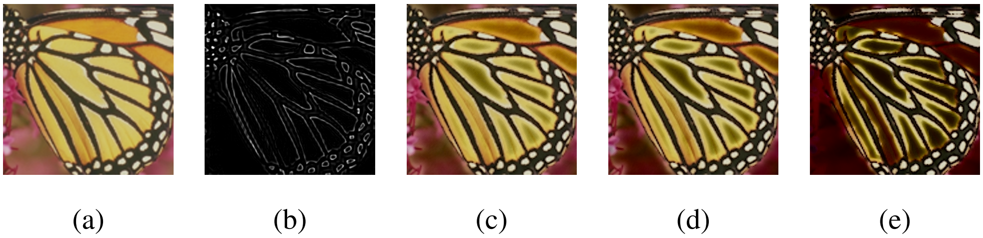

(a) (b) (c) (d) (e)

Figure 5.11: Inverse results ($\hat{\mathrm{x}}$) for three λ values: (a) Input (x), (b) residual image ($\Phi(\mathrm{x})$), and (c) $\hat{\mathrm{x}}$ with λ = 0.1, (d) $\hat{\mathrm{x}}$ with λ = 0.15, (e) $\hat{\mathrm{x}}$ with λ = 0.4.

Table 5.3: Summary of distances measured with MSE, PSNR, and SSIM. The best numbers are shown in bold. Results show that $\Phi(\mathrm{x}) \simeq \Phi(\hat{\mathrm{x}})$ (d_B and d_C), even with the large difference in the d_A.

Dataset	Scale	d_A (MSE/PSNR/SSIM)	d_B (MSE/PSNR/SSIM)	d_C (MSE/PSNR/SSIM)	d_D (MSE/PSNR/SSIM)	d_E (MSE/PSNR/SSIM)
Set5	×2	0.0115/21.09/0.7845	0.0004/35.44/0.8943	0.0004/35.44/**0.9596**	**0.0002/37.54**/0.9596	0.0006/32.97/0.9227
	×3	0.0145/19.25/0.8031	0.0008/34.12/0.8238	0.0008/**34.12/0.9438**	**0.0005**/33.72/0.9233	0.0012/30.83/0.8753
	×4	0.0109/20.78/0.8190	0.0012/32.42/0.7304	0.0012/**32.42/0.9227**	**0.0009**/31.39/0.8858	0.0020/28.80/0.8190
Set14	×2	0.0228/20.58/0.7713	**0.0005/33.98**/0.8866	**0.0005/33.98/0.9543**	0.0007/33.14/0.9140	0.0012/29.93/0.8763
	×3	0.0240/19.86/0.7891	**0.0006/33.27**/0.8504	**0.0006/33.27/0.9479**	0.0014/29.92/0.8346	0.0020/27.98/0.7969
	×4	0.0239/19.99/0.8015	**0.0008/32.47**/0.7918	**0.0008/32.47/0.9395**	0.0021/28.13/0.7705	0.0029/26.51/0.7223
BSDS100	×2	0.0150/21.03/0.7875	**0.0004/34.82**/0.8996	**0.0004/34.82/0.9604**	0.0010/31.91/0.8968	0.0014/29.76/0.8683
	×3	0.0202/20.58/0.7984	**0.0004/34.74**/0.8720	**0.0004/34.74/0.9560**	0.0019/28.84/0.7991	0.0023/27.53/0.7700
	×4	0.0169/21.21/0.8230	**0.0005/34.75**/0.8207	**0.0005/34.75/0.9514**	0.0026/27.29/0.7261	0.0030/26.27/0.6874
Urban100	×2	0.0283/16.91/0.7534	**0.0007/31.97**/0.8797	**0.0007/31.97/0.9506**	0.0013/30.80/0.9151	0.0020/27.96/0.8760
	×3	0.0290/16.89/0.7718	**0.0011/30.58**/0.8003	**0.0011/30.58/0.9316**	0.0029/27.17/0.8297	0.0039/25.23/0.7768
	×4	0.0297/17.09/0.7886	**0.0015/29.54**/0.7129	**0.0015/29.54/0.9088**	0.0043/25.20/0.7542	0.0058/23.54/0.6841
Average		0.0206/19.61/0.7909	**0.0007/33.17**/0.8302	**0.0007/33.17/0.9439**	0.0016/30.42/0.8507	0.0024/28.11/0.8063

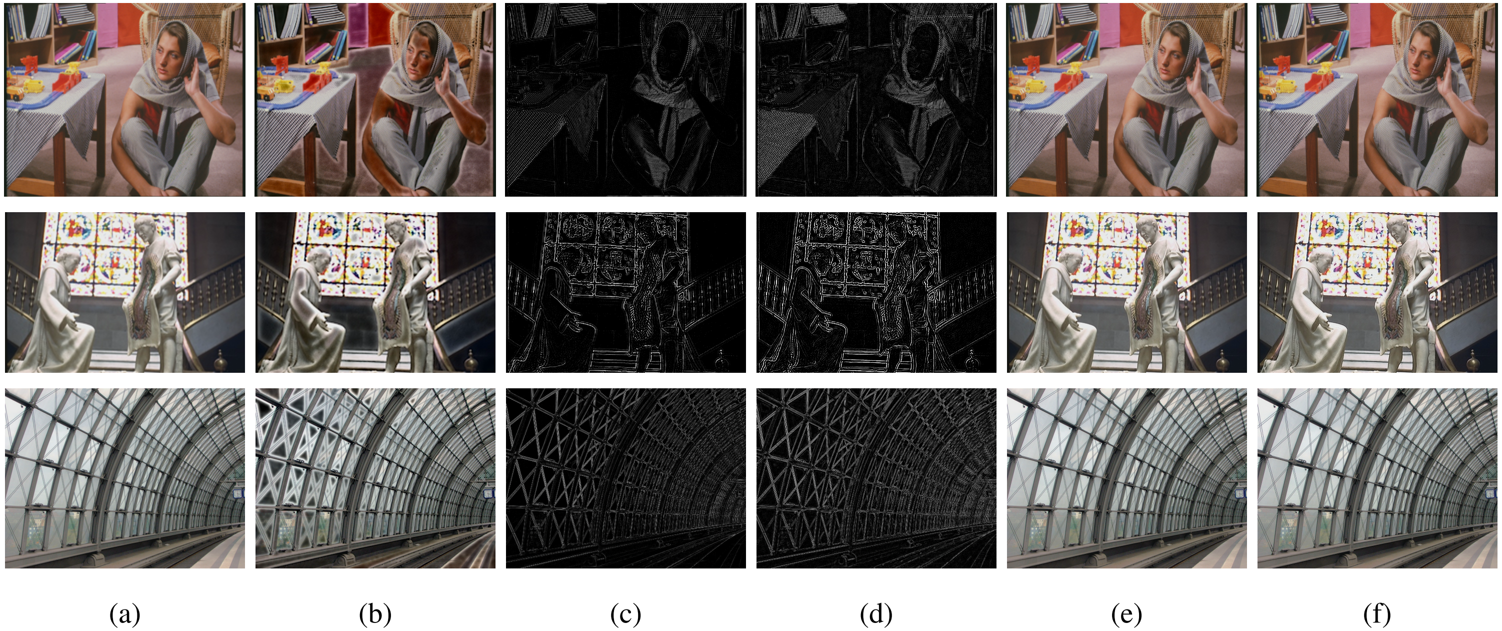


Figure 5.12: Experimental results on Set14 (top), BSDS100 (middle), and Urban100 (bottom) with the scale factor of 2: (a) Input image (x), (b) inverse of residual image ($\hat{\mathrm{x}}$), (c) residual image ($\Phi(\mathrm{x})$), (d) residual image from $\hat{\mathrm{x}}$ ($\Phi(\hat{\mathrm{x}})$), (e) HR image from $\Phi(\mathrm{x})$ ($\mathrm{x} + \Phi(\mathrm{x})$), and (f) HR image from $\Phi(\hat{\mathrm{x}})$ ($\mathrm{x} + \Phi(\hat{\mathrm{x}})$).

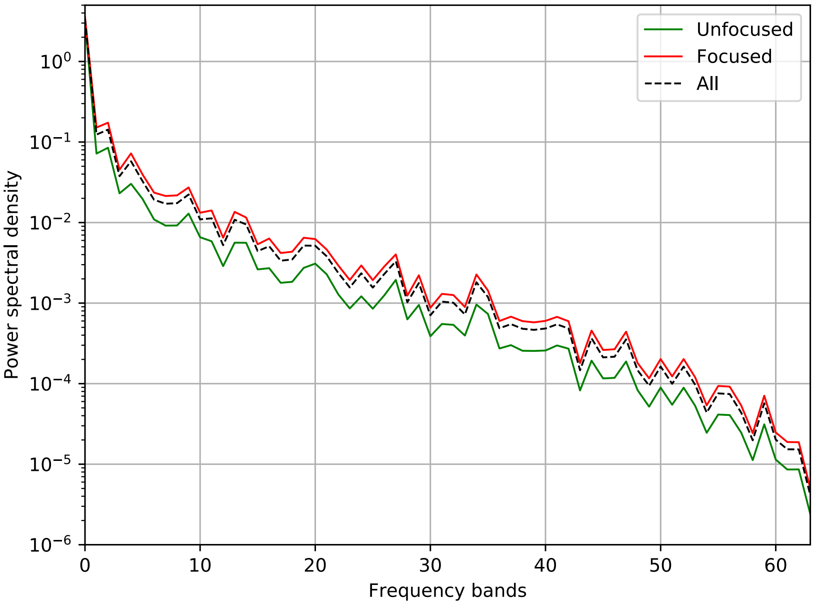
Similar to the previous section, this dissertation plotted a graph whose vertical axis is the output-reconstruction performance ($\mathrm{x} + \Phi(\mathrm{x})$ vs. $\mathrm{x} + \Phi(\hat{\mathrm{x}})$) and the horizontal axis is λ. This dissertation evaluated the MSE and SSIM [86] for the output-reconstruction performance. As shown in Fig. 5.10, the reconstruction performance drops as λ increases. However, unlike the classification case, work related to the visual explanations of regression network are few. As a result, λ is subjectively selected, taking into account the trade-off between the output-reconstruction and visually evaluated input-simplicity. This dissertation set λ to 0.15, although a range of parameters can be possible choices, as shown in the example Fig. 5.11.

5.5.3 Evaluation of the proposed inverse process

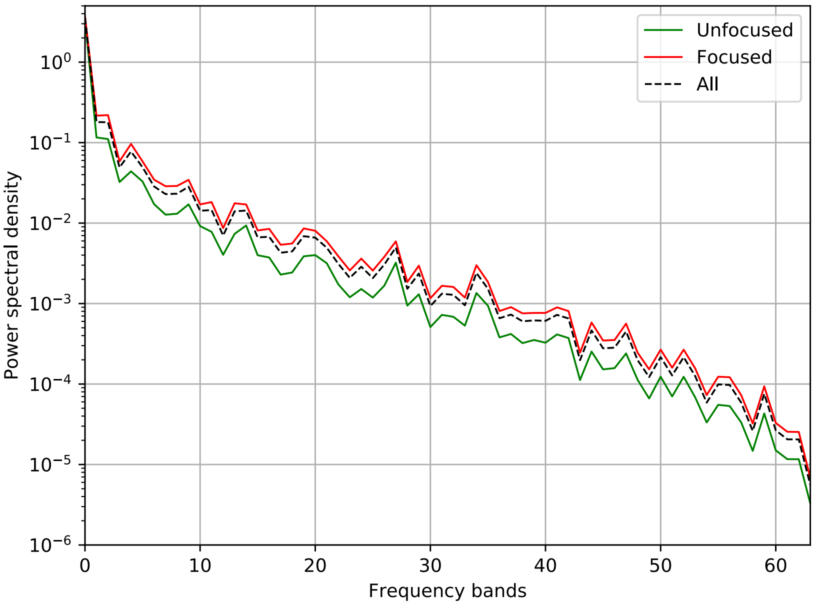
To evaluate the proposed inverse method objectively, this dissertation measured the differences/similarities for the five pairs that are summarized in Tab. 5.2.

The proposed method is evaluated on Set5, Set14, BSDS100, and Urban100 with scale factors $\times 2$, $\times 3$, and $\times 4$. As shown in Tab. 5.3, the proposed method successfully reconstructs original outputs (small d_B and d_C) with simplified inputs (large d_A) in terms of three metrics (MSE, PSNR, and SSIM). Fig. 5.12 shows experimental results from Set14, BSDS100, and Urban100 with the scale factor $\times 2$. As shown in the first and second columns, homogeneous regions (those with little textual information) are naturally faded out in $\hat{\mathrm{x}}$, and the proposed method computes simplified inputs by suppressing them. However, as shown in the last two columns, simplified inputs yield outputs that are similar to the original inference results.

5.5.4 Frequency domain analysis of attribution

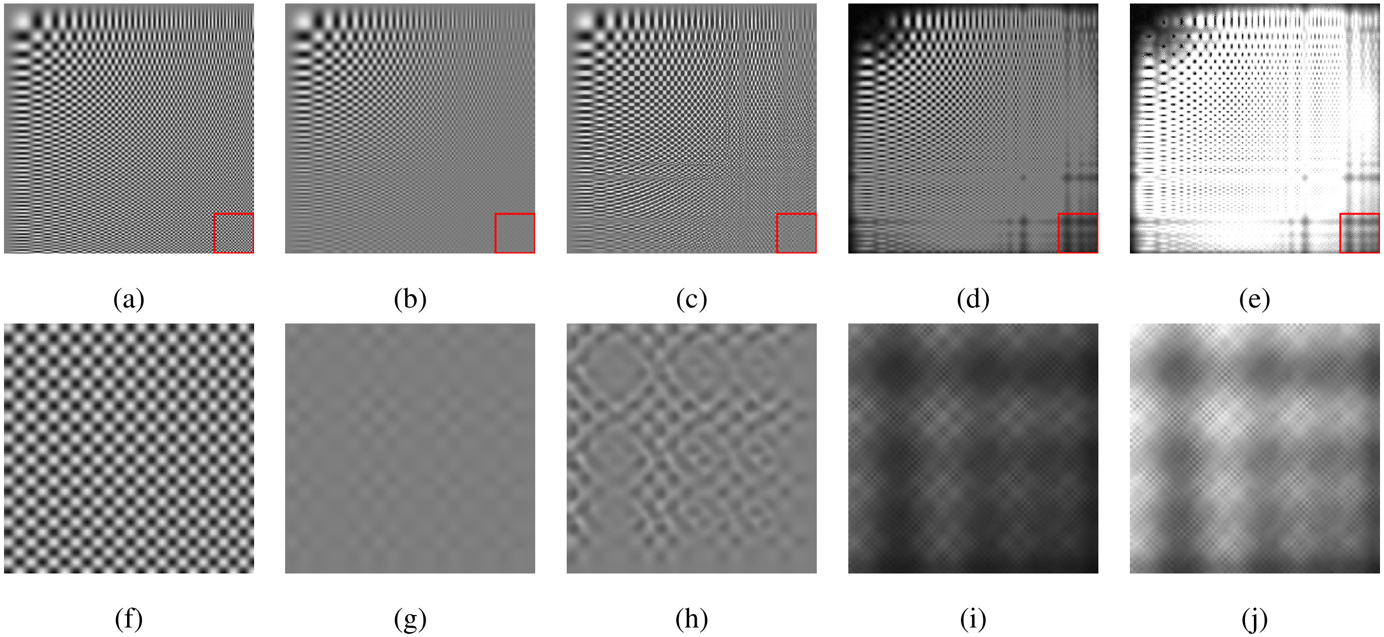


(a)



(b)

Figure 5.13: Average PSDs of *focused* (red), *unfocused* (green), and all (black) 8×8 patches from BSDS100 (a) and Urban100 (b) with scale factor $\times 4$.


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

Figure 5.14: Experiments with an artificial image generated by $\sin\left(\frac{x^2y^2}{1024}\right)$. The first row shows the whole image of each result, and the second row shows high-frequency regions near the bottom right corner that are denoted by red boxes in the first row: ((a), (f)) ground truth HR image, ((b), (g)) input image (x), ((c), (h)) HR image from $\Phi(\mathrm{x})$ ($\mathrm{x} + \Phi(\mathrm{x})$), ((d), (i)) inverse of $\Phi(\mathrm{x})$ ($\hat{\mathrm{x}}$), and ((e), (j)) attribution from $\hat{\mathrm{x}}$ (α).

The $\hat{\mathrm{x}}$ in Fig. 5.12 shows that VDSR has a preference for certain frequency bands. To understand the frequency domain characteristics, this dissertation compared the power spectral densities (PSD) of focused and unfocused regions as follows. First, this dissertation computed α from the $\hat{\mathrm{x}}$ in a test set using Eq. (5.13). Then, x and α were partitioned into 8×8 size non-overlapping patches respectively. Because $\|\alpha\|_1$ of each patch indicates the level of focus, two sets of patches were built based on $\|\alpha\|_1$ values: *Focused* is a set of patches having large $\|\alpha\|_1$ values (greater than the average), whereas *Unfocused* is a set of other patches. The average PSD was then computed using discrete cosine transform (DCT) coefficients. The average PSDs were plotted in 1D by zigzag scanning of averaged 2D coefficients (as in the JPEG encoding), and they were compared to the average PSD computed from all patches. Fig. 5.13 shows that, with the exception of DC components, the *focused* patches have higher PSD than *unfocused* patches in all frequency bands.

The focus on the high-frequency regions may appear natural for SR CNNs because degradations in high-frequency details are more noticeable [27, 75]. However, experiments with an image generated by $\sin\left(\frac{x^2y^2}{1024}\right)$ show that the concentration of VDSR on high-frequency regions of LR image can degrade SR performance. In experiments on this image, this dissertation created the LR input image (x) by bicubic interpolation of the downsampled image (scale factor 4) and visualized the HR image from the x ($\mathrm{x} + \Phi(\mathrm{x})$), the inverse of $\Phi(\mathrm{x})$ ($\hat{\mathrm{x}}$), and attribution from $\hat{\mathrm{x}}$ (α) in Fig. 5.14.

As shown in Fig. 5.14-(b), the input LR image loses its spatial details except for the low-frequency regions near the top left corner. Ideal SR CNNs should be able to recover degraded regions. However, the outputs of VDSR exhibit unfavorable results in the high spatial frequency band (bottom right corner): $\mathrm{x} + \Phi(\mathrm{x})$ shows aliasing as shown in Figs. 5.14-(c) and (h), $\hat{\mathrm{x}}$ shows that VDSR does not focus on some regions, as shown in Figs. 5.14-(d) and (i). Especially, smoothed regions in the bottom right corner of x, which has high frequency details in the input image, are surrounded by high-frequency checkerboard patterns as in Fig. 5.14-(g). Although the high-frequency details in these regions should be reconstructed, VDSR ignores them as shown in Figs. 5.14-(i) and (j). Therefore, this dissertation suggest that SR CNNs should be designed and trained to recognize low-frequency regions, at least when high-frequency regions surround them.

5.6 Summary

In this chapter, this dissertation has proposed an inverse-based approach to explain CNNs. The proposed method performs the inverse operation of a forward-pass in a layer-wise manner, which is designed based on two observations: (1) inverse results should show consistent internal activations to the original forward-pass, and (2) a small amount of activation is preferable. These observations are incorporated into a constrained optimization problem. The proposed method was applied to VGG16 trained for ImageNet classification and VDSR trained for single image super-resolution (SISR). Experimental results have shown that the proposed method can be used to understand predictions for both classification and regression tasks. This dissertation suggests that the understanding of CNNs for other regression tasks using the proposed inverse approach can be an important research topic.

Chapter 6

Conclusions

This dissertation has proposed three new methods to explaining and visualizing the working mechanisms of CNNs.

First, this dissertation has proposed GNL that improves the attribution from integrated gradients by backpropagating only positive-valued gradients of ReLU and max-pool nonlinearity. The backpropagation of positive-valued gradients was inspired by the action potential generation in postsynaptic neurons of the human visual system. Experiments have demonstrated that attributions by GNL show enhanced visual quality and achieve state-of-the-art deletion score.

Next, the operation-wise inverse approach has been presented based on the observation that CNNs can be decomposed into four fundamental operations. The inverses of fundamental operations are formulated as constrained optimization problems based on the postulation that inverse results should generate output features consistent with forward-pass. The experimental results have shown that attributions computed by the proposed operation-wise inverse method show state-of-the-art performances in terms of deletion metric, and the results of conventional methods can be similarly obtained by the operation-wise inverse method.

Lastly, this dissertation has presented the layer-wise inverse method. The layer-wise approach considers CNNs as the composition of layers that process positive-valued neural activations. The inverses of layers are computed to be consistent with internal activations to the original forward-pass with the least amount of activation. Experimental results have shown that the layer-wise inverse method can compute the inverse of a prediction given by CNN for classification or regression tasks in the same framework. Especially, the inverse result of VDSR has revealed that VDSR focuses on the high-frequency bands of input images and should be designed to enhance the textural details in the low-frequency regions.

Bibliography

[1] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, 2015.

[2] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, and D. Pedreschi, "A survey of methods for explaining black box models," *ACM Comput. Surv.*, vol. 51, no. 5, p. 93, 2019.

[3] M. Du, N. Liu, and X. Hu, "Techniques for interpretable machine learning," *Commun. ACM*, vol. 63, no. 1, pp. 68–77, 2019.

[4] F. Dalvi, N. Durrani, H. Sajjad, Y. Belinkov, D. A. Bau, and J. Glass, "What is one grain of sand in the desert? analyzing individual neurons in deep nlp models," in *Proc. Assoc. Advancement Artif. Intell. Conf. (AAAI)*, vol. 33, 2019, pp. 6309–6317.

[5] D. Marcos, S. Lobry, and D. Tuia. (2019) Semantically interpretable activation maps: what-where-how explanations within cnns. [Online]. Available: https://arxiv.org/abs/1909.08442

[6] J. Townsend, T. Chaton, and J. M. Monteiro, "Extracting relational explanations from deep neural networks: A survey from a neural-symbolic perspective," *IEEE Trans. Neural Netw. Learn. Syst.*, 2019.

[7] R. M. Byrne, “Counterfactuals in explainable artificial intelligence (xai): Evidence from human reasoning.” in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 2019, pp. 6276–6282.

[8] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” in *Proc. Eur. Conf. Comput. Vis. (ECCV)*. Springer, 2014, pp. 818–833.

[9] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual explanations from deep networks via gradient-based localization,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2017, pp. 618–626.

[10] J. Wagner, J. M. Kohler, T. Gindele, L. Hetzel, J. T. Wiedemer, and S. Behnke, “Interpretable and fine-grained visual explanations for convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 9097–9107.

[11] M. Sundararajan, A. Taly, and Q. Yan, “Axiomatic attribution for deep networks,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3319–3328.

[12] S. Odense and A. d. Garcez. (2020) Layerwise knowledge extraction from deep convolutional networks. [Online]. Available: https://arxiv.org/abs/2003.09000

[13] J. Zhang, S. A. Bargal, Z. Lin, J. Brandt, X. Shen, and S. Sclaroff, “Top-down neural attention by excitation backprop,” *Int. J. Comput. Vis.*, vol. 126, no. 10, pp. 1084–1102, 2018.

[14] A. Binder, G. Montavon, S. Lapuschkin, K.-R. Müller, and W. Samek, “Layerwise relevance propagation for neural networks with local renormalization layers,” in *Proc. Int. Conf. Artif. Neural Netw. (ICANN)*. Springer, 2016, pp. 63–71.

[15] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, “Robust object recognition with cortex-like mechanisms,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, 2007.

[16] M. Riesenhuber and T. Poggio, “Hierarchical models of object recognition in cortex,” *Nature neuroscience*, vol. 2, no. 11, pp. 1019–1025, 1999.

[17] I. Kuzovkin, R. Vicente, M. Petton, J.-P. Lachaux, M. Baciu, P. Kahane, S. Rheims, J. R. Vidal, and J. Aru, “Activations of deep convolutional neural networks are aligned with gamma band activity of human visual cortex,” *Communications biology*, vol. 1, no. 1, pp. 1–12, 2018.

[18] V. Petsiuk, A. Das, and K. Saenko, “RISE: Randomized input sampling for explanation of black-box models,” in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2018.

[19] A. Shrikumar, P. Greenside, and A. Kundaje, “Learning important features through propagating activation differences,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3145–3153.

[20] P. N. Michelini, H. Liu, Y. Lu, and X. Jiang, “A tour of convolutional networks guided by linear interpreters,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 4753–4762.

[21] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, “Hoggles: Visualizing object detection features,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 1–8.

[22] P. Weinzaepfel, H. Jégou, and P. Pérez, “Reconstructing an image from its local descriptors,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*. IEEE, 2011, pp. 337–344.

[23] B.-L. Lu, H. Kita, and Y. Nishikawa, “Inverting feedforward neural networks using linear and nonlinear programming,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 10, no. 6, pp. 1271–1290, 1999.

[24] A. Mahendran and A. Vedaldi, "Understanding deep image representations by inverting them," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 5188–5196.

[25] C. M. Bishop, *Pattern recognition and machine learning*. springer, 2006.

[26] K. Simonyan and A. Zisserman. (2014) Very deep convolutional networks for large-scale image recognition. [Online]. Available: https://arxiv.org/abs/1409.1556

[27] J. Kim, J. K. Lee, and K. M. Lee, "Accurate image super-resolution using very deep convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 1646–1654.

[28] L. H. Gilpin, D. Bau, B. Z. Yuan, A. Bajwa, M. Specter, and L. Kagal, "Explaining explanations: An overview of interpretability of machine learning," in *Proc. IEEE 5th Int. Conf. on Data Sci. and Adv. Analytics (DSAA)*, 2018, pp. 80–89.

[29] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, "Learning deep features for discriminative localization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 2921–2929.

[30] A. Chattopadhay, A. Sarkar, P. Howlader, and V. N. Balasubramanian, "Grad-CAM++: Generalized gradient-based visual explanations for deep convolutional networks," in *Proc. Winter Conf. Appl. Comput. Vis. (WACV)*, 2018.

[31] D. Omeiza, S. Speakman, C. Cintas, and K. Weldermariam. (2019) Smooth Grad-CAM++: An enhanced inference level visualization technique for deep convolutional neural network models. [Online]. Available: https://arxiv.org/abs/1908.01224

[32] R. C. Fong and A. Vedaldi, "Interpretable explanations of black boxes by meaningful perturbation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 3429–3437.

[33] R. Fong, M. Patrick, and A. Vedaldi, "Understanding deep networks via extremal perturbations and smooth masks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2019, pp. 2950–2958.

[34] P. Dabkowski and Y. Gal, "Real time image saliency for black box classifiers," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2017, pp. 6967–6976.

[35] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. (2014) Striving for simplicity: The all convolutional net. [Online]. Available: https://arxiv.org/abs/1412.6806

[36] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[37] P. Du, R. Weber, P. Luszczek, S. Tomov, G. Peterson, and J. Dongarra, "From CUDA to OpenCL: Towards a performance-portable solution for multi-platform GPU programming," *Parallel Computing*, vol. 38, no. 8, pp. 391–407, 2012.

[38] S. Ruder. (2016) An overview of gradient descent optimization algorithms. [Online]. Available: https://arxiv.org/abs/1609.04747

[39] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2012, pp. 1097–1105.

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

[41] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015.

[42] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (XAI)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[43] J. Adebayo, J. Gilmer, M. Muelly, I. Goodfellow, M. Hardt, and B. Kim, "Sanity checks for saliency maps," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 9505–9515.

[44] Z. Qi, S. Khorram, and F. Li, "Visualizing deep networks by optimizing with integrated gradients," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR Workshops)*, 2019, pp. 1–4.

[45] A. Kapishnikov, T. Bolukbasi, F. Viégas, and M. Terry, "XRAI: Better attributions through regions," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 4948–4957.

[46] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[47] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. Hudspeth, *Principles of neural science*. McGraw-hill New York, 2000, vol. 4.

[48] A. Shrikumar, P. Greenside, and A. Kundaje, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3145–3153.

[49] ——, "Learning important features through propagating activation differences," in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2017, pp. 3145–3153.

[50] A. E. Roth, *The Shapley value: essays in honor of Lloyd S. Shapley*. Cambridge University Press, 1988.

[51] M. Ancona, E. Ceolini, C. Öztireli, and M. Gross. (2017) Towards better understanding of gradient-based attribution methods for deep neural networks. [Online]. Available: https://arxiv.org/abs/1711.06104

[52] P. R. Huttenlocher, C. de Courten, L. J. Garey, and H. Van der Loos, "Synaptogenesis in human visual cortex—evidence for synapse elimination during normal development," *Neuroscience letters*, vol. 33, no. 3, pp. 247–252, 1982.

[53] J. B. Demb and J. H. Singer, "Mind the gap junctions: The importance of electrical synapses to visual processing," *Neuron*, vol. 90, no. 2, pp. 207–209, 2016.

[54] M. A. Goodale, J. P. Meenan, H. H. Bülthoff, D. A. Nicolle, K. J. Murphy, and C. I. Racicot, "Separate neural pathways for the visual analysis of object shape in perception and prehension," *Current Biology*, vol. 4, no. 7, pp. 604–610, 1994.

[55] D. Milner and M. Goodale, *The visual brain in action*. OUP Oxford, 2006, vol. 27.

[56] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," in *Proc. Adv. Neural Inf. Process. Syst. Autodiff Workshop (NeurIPS Autodiff Workshop)*, 2017.

[57] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Assoc. Comput. Mach. Special Interest Group Knowl. Discovery Data (ACM SIGKDD)*. ACM, 2016, pp. 1135–1144.

[58] S. Raschka, J. Patterson, and C. Nolet. (2020) Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence. [Online]. Available: https://arxiv.org/abs/2002.04803

[59] G. Plumb, D. Molitor, and A. S. Talwalkar, “Model agnostic supervised local explanations,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2018, pp. 2515–2524.

[60] K. Oh, S. Kim, and I.-S. Oh, “Salient explanation for fine-grained classification,” *IEEE Access*, vol. 8, pp. 61 433–61 441, 2020.

[61] D. Seo, K. Oh, and I.-S. Oh, “Regional multi-scale approach for visually pleasing explanations of deep neural networks (december 2019),” *IEEE Access*, 2019.

[62] V. Nair and G. E. Hinton, “Rectified linear units improve restricted boltzmann machines,” in *Proc. Int. Conf. Mach. Learn. (ICML)*, 2010, pp. 807–814.

[63] B. Liu, M. Wang, H. Foroosh, M. Tappen, and M. Pensky, “Sparse convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2015, pp. 806–814.

[64] S. Changpinyo, M. Sandler, and A. Zhmoginov. (2017) The power of sparsity in convolutional neural networks. [Online]. Available: https://arxiv.org/abs/1702.06257

[65] P. N. Michelini, H. Liu, and D. Zhu, “Multigrid backprojection super–resolution and deep filter visualization,” in *Proc. Assoc. Advancement Artif. Intell. Conf. (AAAI)*, vol. 33, 2019, pp. 4642–4650.

[66] P. N. Michelini, H. Liu, Y. Lu, and X. Jiang, “A tour of convolutional networks guided by linear interpreters,” in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2019, pp. 4753–4762.

[67] C.-C. J. Kuo, “Understanding convolutional neural networks with a mathematical model,” *J. Vis. Commun. Image Represent*, vol. 41, pp. 406–413, 2016.

[68] D. P. Bertsekas and A. Scientific, “Additional algorithmic topics,” in *Convex optimization algorithms*. Belmont, MA, USA: Athena Scientific, 2015, ch. 6, pp. 301–442.

[69] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: A system for large-scale machine learning,” in *12th USENIX Symp. Operating Syst. Des. Implementation (OSDI 16)*, 2016, pp. 265–283. [Online]. Available: https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

[70] W. Nie, Y. Zhang, and A. Patel. (2018) A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. [Online]. Available: https://arxiv.org/abs/1805.07039

[71] W. Samek, A. Binder, G. Montavon, S. Lapuschkin, and K.-R. Müller, “Evaluating the visualization of what a deep neural network has learned,” *IEEE transactions on neural networks and learning systems*, vol. 28, no. 11, pp. 2660–2673, 2016.

[72] H. J. Kwon, H. I. Koo, and N. I. Cho, “Improving explainability of integrated gradients with guided non-linearity,” in *Proc. Int. Conf. Pattern Recognit. (ICPR)*. IEEE, 2021, pp. 385–391.

[73] F. Doshi-Velez and B. Kim. (2017) Towards a rigorous science of interpretable machine learning. [Online]. Available: https://arxiv.org/abs/1702.08608

[74] C. Dong, C. C. Loy, K. He, and X. Tang, “Image super-resolution using deep convolutional networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2015.

[75] B. Lim, S. Son, H. Kim, S. Nah, and K. Mu Lee, “Enhanced deep residual networks for single image super-resolution,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshops (CVPR Workshops)*, 2017, pp. 136–144.

[76] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang *et al.*, “Photo-realistic single image super-resolution using a generative adversarial network,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2017, pp. 4681–4690.

[77] E. Protas, J. D. Bratti, J. F. Gaya, P. Drews, and S. S. Botelho, “Visualization methods for image transformation convolutional neural networks,” *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 30, no. 7, pp. 2231–2243, 2018.

[78] S. Hooker, D. Erhan, P.-J. Kindermans, and B. Kim, “A benchmark for interpretability methods in deep neural networks,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 9734–9745.

[79] Q. Zhang, Y. Nian Wu, and S.-C. Zhu, “Interpretable convolutional neural networks,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2018, pp. 8827–8836.

[80] A. Ghorbani, J. Wexler, J. Y. Zou, and B. Kim, “Towards automatic concept-based explanations,” in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, 2019, pp. 9277–9286.

[81] A. Ben-Israel and T. N. Greville, *Generalized inverses: theory and applications*. Springer Science & Business Media, 2003, vol. 15.

[82] M. Yang and B. Kim. (2019) Benchmarking attribution methods with relative feature importance. [Online]. Available: https://arxiv.org/abs/1907.09701

[83] I. J. Goodfellow, J. Shlens, and C. Szegedy. (2014) Explaining and harnessing adversarial examples. [Online]. Available: https://arxiv.org/abs/1412.6572

[84] A. Shrikumar, P. Greenside, A. Shcherbina, and A. Kundaje. (2016) Not just a black box: Learning important features through propagating activation differences. [Online]. Available: https://arxiv.org/abs/1605.01713

[85] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Proc. Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, 2017, pp. 4765–4774.

[86] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, 2004.

초록

해석 가능한 기계학습 알고리즘들은 최근 많은 관심을 받고 있으며, 이 중 합성곱신경망 (CNN)의 설명 및 시각화는 주요한 연구주제로서 취급되고 있다. 기계학습 알고리즘을 이해하기 위한 다양한 방법 중 특히 주어진 알고리즘의 예측 결과에 대한 입력의 기여도를 시각화하는 귀인 (attribution)과 같은 사후검정 (post-hoc) 국소설명 (local explanation) 방법은 고차원 매개 변수를 가진 비선형 함수에 적용할 수 있어서 CNN의 설명 및 시각화의 주요한 방법으로 사용되고 있다. 이에 따라 본 논문은 CNN의 작동 원리를 시각화하고 이해하는데 사용될 수 있는 세 가지 사후검정 국소설명 방법들을 제시한다.

첫 번째로, 본 논문은 비선형 연산의 양의 기울기 (positive valued gradient)만 역전파 (backpropagation)하여 귀인 성능을 향상시키는 유도된비선형법 (guided non-linearity method)을 제시한다. 유도된비선형법의 설계는 흥분성 및 억제성 시냅스 후 전위의 합에 의존하는 시냅스 후 뉴런의 활동 전위 생성 메커니즘으로부터 비롯되었다. 본 논문은 흥분성 시냅스로 구성된 경로가 출력에 대한 입력의 기여도를 충실하게 반영하고 있다고 가정하였다. 그 후, 본 논문은 비선형 연산의 양의 기울기만 역전파 되도록 허용함으로써 이 가정을 CNN의 설명 및 시각화에 적용할 수 있도록 구현하였다. 본 논문은 실험을 통해, 제안된 유도된비선형법이 삭제척도 (deletion metric) 측면에서 기존의 방법들보다 향상된 성능을 보이며 해석 가능하고 세밀한 (fine-grained) 귀인을 산출함을 보였다.

그러나 유도된비선형법을 포함한 기존의 귀인 방법들은 서로 다른 이론을 기반으로 설계되었으며, 이로 인하여 서로 모순되는 귀인들을 계산하는 때도 있다. 이

문제를 해결하기 위해 본 논문에서는 CNN이 합성곱 (convolution), 최대풀링 (max-pooling), ReLU, 전연결 (full-connected)의 4가지 기본 연산들의 합성함수로 표현될 수 있다는 점에 기반하여, CNN을 통한 예측의 역상 (inverse image)을 기본 연산들의 역연산을 통해 계산하는 연산별역연산법 (operation-wise inverse-based method)을 제안한다. 연산별역연산법은 CNN의 정방향진행 (forward-pass)을 특정 이미지 특징 (image feature)의 크기를 의미하는 물리량의 순차적 전파로 가정한다. 이 가정하에 연산별역연산법은 계산된 역상이 기존의 정방향진행 결과와 모순되지 않도록 설계된 제한된 최적화 문제 (constrained optimization problem)를 통해 기본 연산의 역연산을 계산한다. 본 논문은 실험을 통해 연산별역연산법이 기존의 여러 귀인 방법들보다 삭제척도 측면에서 향상되었으면서도 질적 측면에서 유사한 시각화 결과를 제공하는 것을 보임으로써 연산별역연산법이 귀인계산의 공통 프레임 워크 (reference framework)로 사용될 수 있음을 보였다.

한편, 영상 분류 문제와 같이 단일 예측을 대상으로 한 CNN과는 달리 복수의 예측값을 가지는 CNN에 대하여 귀인계산을 시도한 연구는 현재까지 보고되지 않았다. 이는 기존의 귀인 계산방법들은 CNN에 대하여 단일 스칼라 (scalar) 값을 출력하도록 요구하기 때문이다. 이 문제를 해결하기 위해 본 논문에서는 계층별역연산법 (layer-wise inverse-based method)을 제안한다. 계층별역연산법은 CNN을 인공 뉴런의 활성값 (neural activation)으로 해석할 수 있는 양의 실수들을 입출력으로 하는 계층 (layer)으로 분해하고, 제한된 최적화 문제로 정의되는 각 계층의 역연산을 정방향진행 결과에 순차적으로 적용함으로써 CNN을 통한 예측의 역상을 계산한다. 본 논문은 실험을 통해, 제안된 계층별역연산법이 영상 분류 및 회기를 대상으로 한 CNN들의 설명 및 시각화를 동일한 프레임 워크 (common framework)에서 처리할 수 있음을 확인하였다. 또한, 본 논문은 계층별역연산법을 통해 단일 영상 고해상화 (single image super-resolution)를 대상으로 한 CNN인 VDSR이 입력 영상의 주파수 대역의 일부를 간과하고 있고 이는 VDSR을 통한 고해상화시 특정 주파수 대역에서 영상 품질의 하락을 유발할 수 있음을 보였다.

주요어: 합성 곱 신경망, 해석가능한 기계학습, 역연산, 영상 분류, 영상 고해상화
학번: 2017-37754