DOI: 10.1111/cgf.14302
Eurographics Conference on Visualization (EuroVis) 2021
R. Borgo, G. E. Marai, and T. von Landesberger
(Guest Editors)

Volume 40 (2021), Number 3

A Visual Designer of Layer-wise Relevance Propagation Models

Xinyi Huang[1] and Suphanut Jamonnak[1] and Ye Zhao[1]and Tsung Heng Wu[1] and Wei Xu[2]

[1]Kent State University, [2]Brookhaven National Laboratory

Abstract

Layer-wise Relevance Propagation (LRP) is an emerging and widely-used method for interpreting the prediction results of convolutional neural networks (CNN). LRP developers often select and employ different relevance backpropagation rules and parameters, to compute relevance scores on input images. However, there exists no obvious solution to define a "best" LRP model. A satisfied model is highly reliant on pertinent images and designers' goals. We develop a visual model designer, named as VisLRPDesigner, to overcome the challenges in the design and use of LRP models. Various LRP rules are unified into an integrated framework with an intuitive workflow of parameter setup. VisLRPDesigner thus allows users to interactively configure and compare LRP models. It also facilitates relevance-based visual analysis with two important functions: relevance-based pixel flipping and neuron ablation. Several use cases illustrate the benefits of VisLRPDesigner. The usability and limitation of the visual designer is evaluated by LRP users.

1. Introduction

Deep learning techniques have seen a dominant and pervasive surge in many domains by producing state-of-the-art results with computational solutions based on deep neural networks (DNNs). Nevertheless, a critical problem remains for neural network models which reside in the lack of interpretability and transparency. Explainable deep learning has become an important research topic, while a variety of visualization methods and tools have been developed to "open the black box" [YCN*15, CL18, HKPC19]. In the field of computer vision, many computational methods discover input data components related to decisions based on perturbation, gradient, sensitivity, and relevance scores [SMV*19]. Recently, Layer-wise Relevance Propagation (LRP) methods [BBM*15, MSM18] have become an emerging focus from computer vision researchers [AHM*16,SLSM16,BAL*18,HMK*19,GCS*19, IKU19,KJL19,LWB*19,NIAN19,SKK*19], as they can discover significant input features contributing to the classification or prediction output. LRP also overcomes the weakness of shattered gradients in gradient methods (Grad-CAM) and makes up for the perturbation method (occlusion map).

LRP techniques explain the prediction of a convolutional neural network (CNN) by finding the relevance of input image pixels to the output. By initiating relevance on a selected output class, a backward propagation from the output layer to the lower layers is employed to compute relevance values at each layer and towards the input pixels. In the backpropagation, the distribution of relevance can be computed by using different relevance propagation rules that utilize the forward neuron activations and a set of artificial parameters. Composite LRPs further allow different propagation rules and parameters to be used at different layers selected by users [SBLM17]. With respect to these variations and selections, LRP models create different relevance values which are often shown as heatmaps to explain the contribution of input pixels towards CNN output.

A preferred LRP model depends on different input images and the specific goal of the user. For example, finding salient features for the prediction of "dog" in small scales (e.g. eyes and ears) or large scales (e.g., body shape) will need to configure different LRP models. Therefore, there is no "best" LRP model that a user can create directly. A trial-and-error process often needs to be conducted to investigate different propagation rules and parameter settings. LRP developers have presented suggestions of how to choose the model [MSM18], such as "*if negative relevance is needed, or the heatmaps are too diffuse, replace the rule LRP-*$\alpha1\beta0$ *by LRP-*$\alpha2\beta1$ *in the hidden layers*", "*If the heatmaps obtained with LRP-*$\alpha1\beta0$ *and LRP-*$\alpha2\beta1$ *are unsatisfactory, consider a larger set of propagation rules*". Composite LRPs further confound the situation when various rules are tested on different CNN layers. Novice users are often perplexed in understanding and manipulating this process. Even experienced users need to spend great time and effort in model design, validation, adjustment, and comparison. Unfortunately, in most cases, this exploratory process has to be implemented manually in the coding stage.

In this paper, we develop a visual designer, named as VisLRPDesigner, which helps domain experts and students efficiently design, debug, and compare LRP models. It further integrates two visual analytics functions based on the computed relevance for model validation including: (1) pixel flipping which flips input image pixels to check CNN output changes, and (2) neuron ablation which removes specific neurons to see how that affects performance [VKS20]. The main contributions of this work are as follows:

- We construct an integrated computational framework of different LRP rules. Based on it, we propose a configuration workflow with four parameter-setting steps.
- We identify and utilize segments of CNN layers as basic LRP configuration and computing units. Users thus can flexibly set up and visualize LRP rules and parameters over them.
- We build a visual interface which integrates several coordinated views for multiple segment selection, interactive parameter definition, and LRP result examination. It also facilitates model management and comparison.
- We facilitate users to perform relevance-based visual analysis with popular model explanation tools: neuron ablation and pixel flipping.
- We evaluate he usability of VisLRPDesigner with LRP learners and experts, and discuss the benefits and limitations.

In summary, we present the first visual analytics (VA) system, to the best of our knowledge, that facilitates the easy and intuitive design of LRP models. VisLRPDesigner has two major benefits in promoting wider use of LRP in deep learning explanation: (1) it can mitigate the burden of LRP developers and (2) it can expedite the learning of LRP techniques in education.

2. Related Work

Computational approaches of deep learning explanation have been addressed through a variety of algorithms [SMV*19]. Importance scores of the input features (i.e., saliency map, relevance map) can be computed for model understanding. For instance, perturbation methods [AMJ17], saliency-based methods [SCD*20], and influence functions [KL17] were proposed for these purposes. These approaches were mostly aimed at CNN models to visualize learned features. A general taxonomy classified them into three main categories [GRNT16]: input modification methods, deconvolutional methods, and input reconstruction methods. They often elucidated the internal processes by visualizing input contribution heatmaps. Deconvolutional Networks (DeconvNets) [SVZ14, ZF14], Guided Back Propagation [SDBR15], Class Activation Mapping (CAM) [LCY14, SCD*17, ZKL*16], were the popular approaches. Recently, LRP has become an emerging focus from computer vision researchers [AHM*16, SLSM16, BAL*18, HMK*19, GCS*19, IKU19, KJL19, LWB*19, NIAN19, SKK*19, HMK*19, IKU19, GYT19, LLMX20]. Heatmaps were mostly used in these methods to visualize input pixels' relevance values. An online Interactive LRP Demo System (at *heatmapping.org*) was developed for users to study a few popular LRP rules and see the resultant heatmap. The parameters were changeable through input boxes. However, this demo is very simple and does not support users to customize models, define multi-segments with various rules, and perform model comparison. In comparison, VisLRPDesigner provides a comprehensive visualization tool for LRP design and exploration. It also novelly integrates relevance-based pixel flipping and neuron ablation in the visual system.

Interactive visualization tools have been developed to provide an in-depth understanding of how deep learning models work [CL18, HKPC19, RFFT17]. Some design tools [WSW*18, Kar, STN*16] allowed users to interact with the activation maps and network structures. CNNVis [LSL*17] helped designers in exploring the learned representations in the graph layout. ActiVis [KAKC18] integrated embedding view with multiple coordinates views for visual model exploration. Deep View [ZXZ*17] presented a level-of-detail framework that measured the evolution of the deep neural network both on a local and on a global scale. DeepEyes [PHV*18] supported the identification of layers that learned a stable set of patterns during training. REMAP [CPCS20] discovered a DNN model via visual exploration and rapid experimentation including ablation of neural network architectures. SUMMIT [HPRP20] performed activation aggregation and neuron-influence aggregation, and visualized an attribution graph. For model comparison, DeepCompare [MMD*19] visually compared multiple DNN models for their behaviors and assessed trade-offs among them. A visual genealogy of DNNs helped practitioners understand the behavior and evolution of many existing DNN models [WYC*19]. VATLD [GZL*20] applied representation and adversarial learning to understand the accuracy and robustness of traffic light detectors. CNN Explainer [WTS*21] allowed students to interactively learn and understand high-level structure function and low-level mathematical computation of CNN. However, LRP technologies have not been well utilized in the VA approaches.

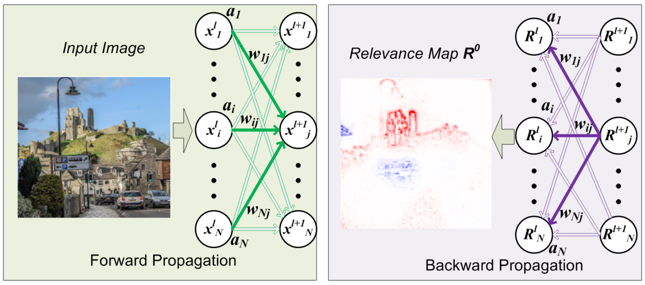


Figure 1: *LRP computational process with two phases: forward propagation of activation and backward propagation of relevance.*

3. LRP Method and Relevance Data

LRP computes relevance values that quantify the contribution of CNN components and input image features to prediction class. Next, we briefly introduce the LRP computing process and rules for a trained CNN.

3.1. LRP Computational Process

LRP computation is implemented in two phases shown in Fig. 1:

- First, a standard forward propagation pass is applied to the network from an input image. The activation a_i of each neuron x^l_i at layer l is collected. The network weight from neuron x^l_i to neuron x^{l+1}_j in its successor layer $l+1$ is also recorded as w_{ij}.
- Second, with a layer-wise backward propagation pass, a relevance map R^l_k is computed to represent the relevance of each neuron k at each layer l. The computation starts from an initial (input) relevance vector R defined at the output layer. Then a backpropagation from layer $l+1$ to layer l is implemented with

Table 1: *Popular LRP rules within the unified formula.*

LRP Rule	Reference	Original Formula	Parameters in the Unified Formula (Eqn. 5)
LRP-0	[BBM*15]	$R_i = \sum_j \frac{a_i w_{ij}}{\sum_i a_i w_{ij}} R_j$	α=1, β=0, ε =0, θ_0=0, θ_1=1, γ_0=0, γ_1=1, γ_{1p}=0, γ_{1n}=0, γ_2=0
LRP-ε	[BBM*15]	$R_i = \sum_j \frac{a_i w_{ij}}{\varepsilon + \sum_i a_i w_{ij}} R_j$	α=1, β=0, $\varepsilon \in [0,1]$, θ_0=0, θ_1=1, γ_0=0, γ_1=1, γ_{1p}=0, γ_{1n}=0, γ_2=0
LRP-γ	[MBL*19]	$R_i = \sum_j \frac{a_i(w_{ij} + \gamma w_{ij}^+)}{\sum_i a_i(w_{ij} + \gamma w_{ij}^+)} R_j$	α=1, β=0, ε =0, θ_0=0, θ_1=1, γ_0=0, γ_1=1, γ_{1p}=1, γ_{1n}=0, γ_2=0
LRP-αβ	[BBM*15]	$R_i = \sum_j (\alpha \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} - \beta \frac{a_i w_{ij}^-}{\sum_i a_i w_{ij}^-}) R_j$	α-β=1, $\beta \geq 0$, ε =0, θ_0=0, θ_1=1, γ_0=0, γ_1=0, γ_{1p}=1 , γ_{1n}=1 , γ_2=0
$LRP - z^+$	[BBM*15]	$R_i = \sum_j \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} R_j$	α=1, β=0, ε =0, θ_0=0, θ_1=1, γ_0=0, γ_1=0, γ_{1p}=1, γ_{1n}=0, γ_2=0
$LRP - w^2$	[MLB*17]	$R_i = \sum_j \frac{w_{ij}^2}{\sum_i w_{ij}^2} R_j$	α=1, β=0, ε =0 , θ_0=1, θ_1=0, γ_0=0, γ_1=0, γ_{1p}=0, γ_{1n}=0, γ_2=1
LRP-flat	[LWB*19]	$R_i = \sum_j \frac{1}{\sum_i 1} R_j$	α=1, β=0, ε =0, θ_0=1, θ_1=0, γ_0=1, γ_1=0, γ_{1p}=0, γ_{1n}=0, γ_2=0

relevance conservation as:

$$R_i^l = \sum_j \mathcal{F}(a_i, w_{ij}) R_j^{l+1}, \quad \sum_i R_i^l = \sum_j R_j^{l+1}. \tag{1}$$

Here $\mathcal{F}$ is the LRP propagation rule. This process stops when a *relevance map* R^0 is achieved on the input image ($l = 0$).

R is typically defined for a target class t from N output classes as:

$$R_i = \begin{cases} c_t, & i = t \;\; i \in [1..N] \\ 0, & \text{otherwise}. \end{cases} \tag{2}$$

Here c_t is the pre-softmax value of class t.

3.2. LRP Propagation Rules

In the original paper of LRP [BBM*15], the propagation rule $\mathcal{F}$ is defined in two popular forms called $LRP - \varepsilon$ and $LRP - \alpha\beta$:

$$LRP\text{-}\varepsilon: \quad R_i^l = \sum_j \frac{a_i w_{ij}}{\varepsilon + \sum_i a_i w_{ij}} R_j^{l+1}, \tag{3}$$

where a small constant ε prevents numerical instability.

$$LRP\text{-}\alpha\beta: \quad R_i^l = \sum_j (\alpha \cdot \frac{a_i w_{ij}^+}{\sum_i a_i w_{ij}^+} - \beta \cdot \frac{a_i w_{ij}^-}{\sum_i a_i w_{ij}^-}) R_j^{l+1}, \tag{4}$$

where $()^+$ and $()^-$ denote the positive weights and the negative weights, and α and β are chosen parameters with $\alpha - \beta = 1$. $LRP - \alpha\beta$ splits the positive and negative activations, while α and β modulate the contributions of excitatory and inhibitory effects (see [MSM18]). The result LRPs are usually referred by the α and β values, such as $LRP - \alpha 2 \beta 1$ for using $\alpha = 2$ and $\beta = 1$, and $LRP - \alpha 1 \beta 0$ for using $\alpha = 1$ and $\beta = 0$.

Researchers have further proposed more LRP forms by manipulating the propagation function $\mathcal{F}$, such as LRP-γ, LRP-w^2, LRP-z^+, LRP-z^β, and so on, which are summarized in Table 1. For example, one popular enhancement approach is LRP-γ which emphasizes positive contributions over negative contributions.

A *composite LRP* strategy is proposed [SBLM17], where different rules are used at different layers. It provides more flexibility for users. For instance, it has been suggested that $LRP - 0$ for top layers, $LRP - \varepsilon$ for middle layers, and $LRP - \gamma$ for lower layers [MBL*19]. However, the flexibility of LRP also imposes a great burden on users. VisLRPDesigner is designed for making this process comfortable and effective.

3.3. Relevance Data

LRP computes *neuron relevance maps* (R_i^l) on every neuron. These neuron maps of a layer can be aggregated (usually by averaging) to form a *layer relevance map*. An input relevance map (R^0) is generated on the input layer. The relevance values in these maps include positive and negative scores. We can compute positive and negative relevance scores for each neuron by summing up the positive and negative relevance values, respectively. These relevance maps and scores can be visualized to show LRP behaviors.

4. VisLRPDesigner Design Overview

4.1. Design Goal

An LRP model is defined with specific LRP rules and parameters over CNN layers. The process of selecting different LRP models and comparing their results can be perplexing and overwhelming without an easy-to-use tool. VisLRPDesigner is designed to provide visual interactions for easy configuration and investigation of different LRP models. The aimed users are domain experts or students who want to understand how different LRP rules and parameters affect the results of relevance score computation.

VisLRPDesigner is developed to overcome several practical challenges, including (1) LRPs have many different propagation rules which need to be tested and selected for different purposes; (2) An LRP model needs to adjust multiple parameters to find good results; (3) different LRP functions and parameters may need to be applied to different CNN layers. Moreover, users may need to utilize the computed relevance in the study of CNN performance so as to investigate different LRP models. VisLRPDesigner thus integrates two popular methods including:

- *Relevance-based noise flipping*: flipping high- or low-relevance pixels to study the change of the CNN prediction.
- *Relevance-based neuron ablation*: cutting off selected neurons based on relevance and evaluating how the CNN's performance changes due to the ablation.

These visual analytics functions enable users to visually study whether the computed relevance scores can successfully discover CNN prediction behaviors. For noise flipping, users can link relevance scores to features on the input images. For neuron ablation, users can investigate neuron-level relevance scores related to prediction output. Then users can justify their selection of different LRP rules and parameters.

In summary, VisLRPDesigner is developed to promote easier and wider LRP application in (1) helping users design, manage, and compare LRP models; and (2) integrating relevance-based visual analytics for optimal LRP model design.

4.2. System Functions

To fulfill the goal, VisLRPDesigner is developed as a VA system integrating modeling (**M1-M2**), visual configuration (**V1-V4**), and visual analysis (**V5-V6**) functions as follows:

- **M1: Unified LRP Formula:** Various LRP rules are integrated into one formula so that users can perform interactive visual configuration easily;
- **M2: LRP Parameter Setting Guide:** LRP parameters are categorized into four groups leading to a four-step workflow with guidance, through which users can control and understand this task better.

- **V1: Visual Definition of LRP Segments:** We propose a concept of "segment" of CNN layers so that multiple LRP rules can be applied to different groups of CNN layers. Users are allowed to interactively split CNN into a few segments, define different rules on them, and compare their relevance results.
- **V2: Interactive Setting of LRP Rules and Parameters:** We develop a visual interface for intuitive LRP rule selection, customization, and parameter adjustment.
- **V3: Relevance Examination:** Users can interactively compute and visualize the relevance data (Sec. 3.3) to examine LRP results.
- **V4: Model Management and Comparison:** A visual manager allows users to create, change, and remove LRP models. A model comparison interface further helps them perform comparison studies on different LRP models.

- **V5: Interactive Pixel Flipping:** A pixel flipping interface supports users to freely brush over relevance heatmaps or input images. The selected pixels are flipped by setting their values to zeros for the new input image to compute the new CNN prediction result so as to understand CNN behavior.
- **V6: Visual Neuron Ablation:** A neuron ablation interface allows users to select CNN layers and visualize their neuron-level relevance data. Then, they can choose individual neurons or groups of neurons for ablation study [LMM18, VKS20], where the new prediction result when the selected neurons are removed can be compared with the original CNN prediction.

5. Unified LRP Formula and Parameters Setting Guide

There exist a variety of LRP propagation rules and users need to adjust their parameters on the fly. In order to support visual interactions in a consistent manner for different rules, we develop a unified formula (for **M1**) to represent LRP rules:

$$R_i^l = \sum_j \left[\alpha \frac{(\theta_0 + \theta_1 a_i)(\gamma_0 + \gamma_1 w_{ij} + \gamma_{1p} w_{ij}^{+} + \gamma_2 w_{ij}^{2})}{\varepsilon + \sum_i (\theta_0 + \theta_1 a_i)(\gamma_0 + \gamma_1 w_{ij} + \gamma_{1p} w_{ij}^{+} + \gamma_2 w_{ij}^{2})} \right. \\ \left. - \beta \frac{(\theta_0 + \theta_1 a_i)(\gamma_0 + \gamma_1 w_{ij} + \gamma_{1n} w_{ij}^{-} + \gamma_2 w_{ij}^{2})}{\varepsilon + \sum_i (\theta_0 + \theta_1 a_i)(\gamma_0 + \gamma_1 w_{ij} + \gamma_{1n} w_{ij}^{-} + \gamma_2 w_{ij}^{2})} \right] R_j^{l+1}. \tag{5}$$

Based on this formula, various LRP rules can be defined and compared uniformly by using different sets of the ten parameters (see Table 1). VisLRPDesigner allows users to easily select heuristic values presented in the literature. In addition, users can try different values: for example, instead of simply using α and β as either 1 or 0, they can also try float values to flexibly combine excitatory and inhibitory effects.

We further design four-parameter setting steps with different tasks following the nature of LRP rules. Users are also guided to control different relevance effects within these steps.

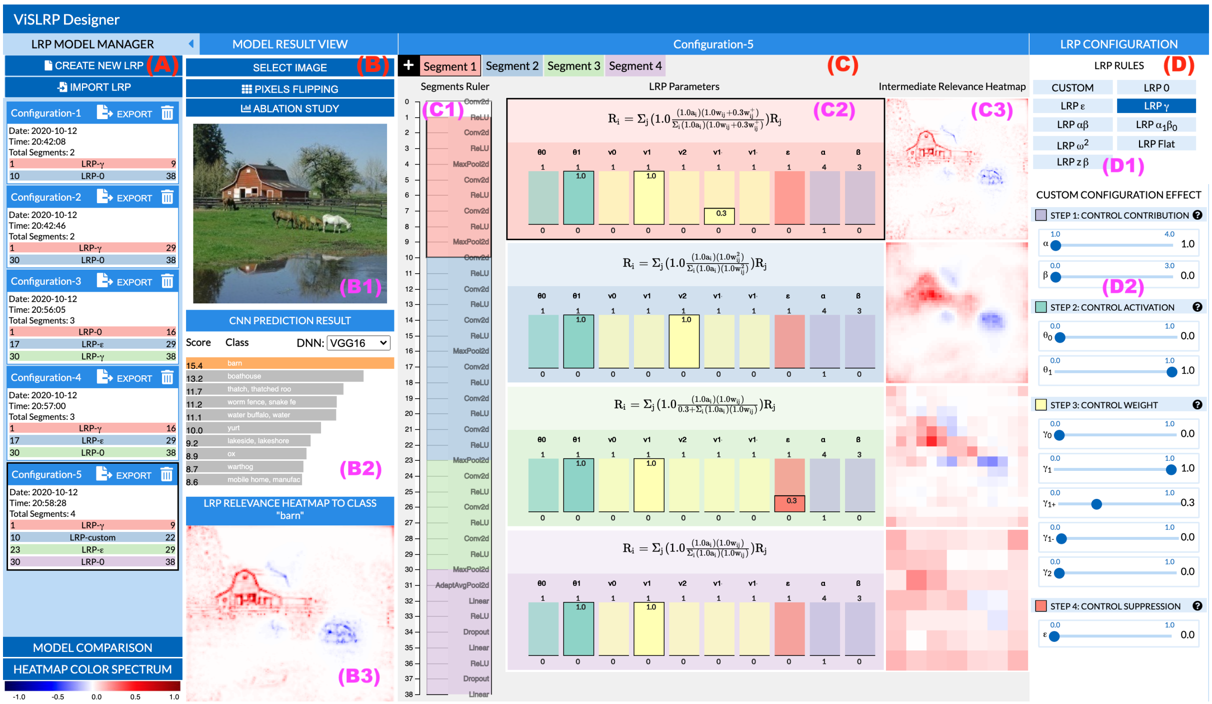
- *Step 1: Control positive/negative contribution with* α *and* β:
  Guide to users: This step is to define the relevance with different contributions from the positive and negative backpropagated relevances. With $\alpha - \beta = 1$, increasing the values (e.g., from $(\alpha = 1, \beta = 0)$ to $(\alpha = 2, \beta = 1)$ can focus on small but important features.
- *Step 2: Control activation effect with* θ_0 *and* θ_1:
  Guide to users: This step is to tune the dependency of activation. In most cases, use $\theta_0 = 0$ and $\theta_1 = 1$ to include activations in LRP. When activations are not involved, use $\theta_0 = 1$ and $\theta_1 = 0$.
- *Step 3: Control weight effect with* γ_0, γ_1, γ_{1p},γ_{1n}, *and* γ_2:
  Guide to users: This step is to tune the dependency of different weights. Use these parameters to define the contributions of CNN filters with original weight (γ_1), positive weight (γ_{1p}), negative weight (γ_{1n}), and squared weight (γ_2); Use γ_0 as a constant if weights are not involved.
- *Step 4: Control suppression effect with* ε :
  Guide to users: This step is to suppress a certain level of noise.

6. VisLRPDesigner System

6.1. Visual Interface

VisLRPDesigner interface is shown in Fig. 2 consisting of four parts (A-D):

- **Model Manager (for V4)**: In Fig. 2A, users can create, edit, and remove multiple LRP models whose names are set by users. Each model shows its configuration information including multi-segment with start and end layers, and the LRP rules used on these segments. Users can choose any model to make it as the active model for investigation.
  Design rationale: An overview of LRP models with segment bars can help users quickly identify their features and directly select multiple models for comparison study. Horizontal segment bars are used here since usually, the number of segments is not large in a range between 1 to 5. It is easy to display LRP rules on the bars together with start and end layers.
- **Model Result View (for V3)**: In Fig. 2B, users can load an image (Fig. 2B1: "barn on lake"), and then check its CNN classification results (Fig. 2B2). By selecting one class ("barn"), the active LRP model is used to compute relevance to this class. Then, users can study its relevance map as a color-encoded heatmap (Fig. 2B3). The heatmap visualizes positive and negative values of relevance with red and blue colors so that users can easily check the contribution of input pixels.
  Design rationale: The VGG model predicts an image with scores for 1000 classes. The top 10 classes (Fig. 2B2) are shown while



Configuration-1 EXPORT
Date: 2020-10-12
Time: 20:42:08
Total Segments: 2

1	LRP-γ	9
10	LRP-0	38

Configuration-2 EXPORT
Date: 2020-10-12
Time: 20:42:46
Total Segments: 2

1	LRP-γ	29
30	LRP-0	38

Configuration-3 EXPORT
Date: 2020-10-12
Time: 20:56:05
Total Segments: 3

1	LRP-0	16
17	LRP-ε	29
30	LRP-γ	38

Configuration-4 EXPORT
Date: 2020-10-12
Time: 20:57:00
Total Segments: 3

1	LRP-γ	16
17	LRP-ε	29
30	LRP-0	38

Configuration-5 EXPORT
Date: 2020-10-12
Time: 20:58:28
Total Segments: 4

1	LRP-γ	9
10	LRP-custom	22
23	LRP-ε	29
30	LRP-0	38

CNN PREDICTION RESULT

DNN: VGG16

Score	Class
15.4	barn
13.2	boathouse
11.7	thatch, thatched roo
11.2	worm fence, snake fe
11.1	water buffalo, water
10.0	yurt
9.2	lakeside, lakeshore
8.9	ox
8.7	warthog
8.6	mobile home, manufac

STEP 1: CONTROL CONTRIBUTION
α (1.0–4.0): 1.0
β (0.0–3.0): 0.0

STEP 2: CONTROL ACTIVATION
θ_0 (0.0–1.0): 0.0
θ_1 (0.0–1.0): 1.0

STEP 3: CONTROL WEIGHT
γ_0 (0.0–1.0): 0.0
γ_1 (0.0–1.0): 1.0
γ_{1+} (0.0–1.0): 0.3
γ_{1-} (0.0–1.0): 0.0
γ_2 (0.0–1.0): 0.0

STEP 4: CONTROL SUPPRESSION
ε (0.0–1.0): 0.0

Figure 2: *VisLRPDesigner interface. (A): Model manager for users to create, edit, and compare multiple LRP models, and to change heatmap color spectrum. (B): Model result view for selecting an image of interest (B1), checking CNN prediction results (B2), and studying LRP computed relevance heatmap (B3). (C): LRP configuration view which includes a segment ruler (C1) for users to drag and define segments, and LRP parameter view (C2) visualizing parameters of four segments, and intermediate relevance heatmaps (C3) after each segment. (D): LRP configuration panel for users to select a predefined LRP rule (D1) and customize LRP parameters in four steps (D2). Please note that VGG16 is used in the examples throughout the paper unless otherwise specified.*

their scores are mapped to the bar lengths, and class names are labelled on the bars for easy observation.

For the relevance heatmap, in default, negative relevance and positive relevance are shown in blue and red, respectively, because this color scheme is the most popular one widely used in LRP literature. The color mapping scheme can be changed from a list of color spectra (at the bottom of Fig. 2A), so as to meet the need of users such as color-blind people.

- **LRP Configuration View (for V1-V3)**: In Fig. 2C, a segment ruler (Fig. 2C1) shows how the CNN layers are divided into segments. Users can easily drag a segment on the ruler to change its effective layers. Here, four segments (Segment1 to Segment4) are shown with different colors (for **V1**). In an LRP parameter view (Fig. 2C2), the parameter values and LRP rules applied to each segment (for **V2**) are visualized. Moreover, the intermediate relevance heatmaps (for **V3**) are shown for each segment (Fig. 2C3). Here, Segment1 is highlighted and its configuration can be adjusted in Fig. 2D.

  Design rationale: The ruler helps users easily add, remove, and edit multiple segments flexibly. Defining LRP rules on CNN segments need to consider the types of layers. Therefore, each layer is labelled (Conv, Pooling, ReLU, etc.) to provide direct hints.

  The parameter view (Fig. 2C2) presents ten bars with different colors (in four groups) for the ten LRP parameters. The bar sizes are fixed while the min and max values allowed by LRP rules are shown. The parameter value is highlighted over its bar. This design enables quick observation of LRP rules at segments and promotes easy comparison of individual parameters over different segments.

  The intermediate relevance heatmaps (Fig. 2C3) are important for people to understand the LRP effect of the segments they define. They are visualized in the same way as in the input image heatmap (Fig. 2B3) for easy understanding and comparison.

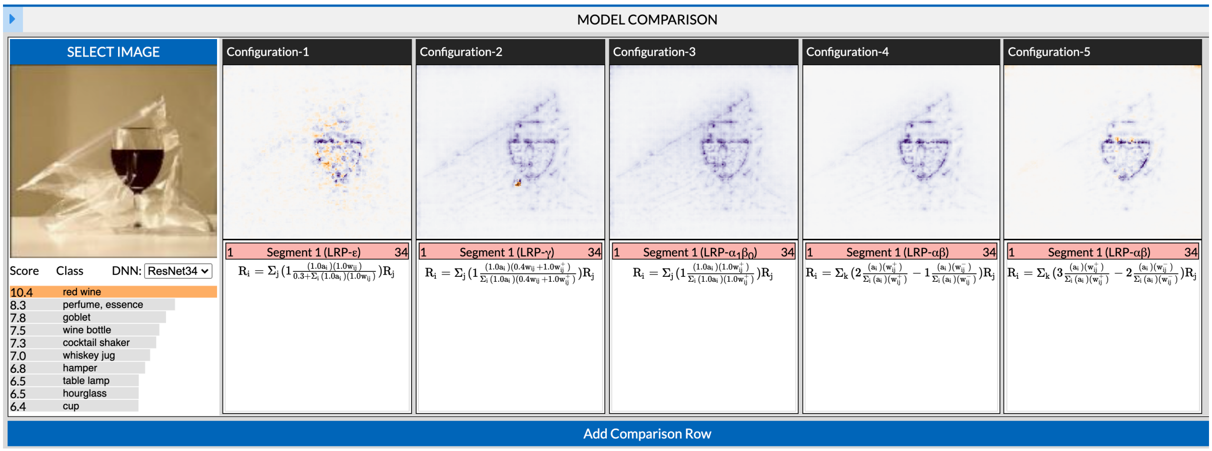
- **LRP Configuration Panel (for V2)**: In Fig. 2D, users can select a popular LRP rule (Fig. 2D1), and then the parameters can be adjusted interactively in their sliders (Fig. 2D2). Moreover, they can also directly customize these parameters to define a preferred LRP model.

  Design rationale: The panel presents existing LRP rules in the predefined buttons for quick selection (Fig. 2D1). Users can directly set the parameter values in four steps (Fig. 2D2). The sliders can be adjusted with the min and max values usually allowed by the LRP rules. Clicking the question marks will show the guide to users in Sec. 5.

These views are coordinated for effective LRP configuration and investigation. The details of this case are discussed in Sec. 7.2.

6.2. LRP Model Comparison

Fig. 3 shows the model comparison interface (for **V4**) where five popular LRP rules (Model1 to Model5) are used in the comparison study. Here, the system works on the CNN architecture of



DNN: ResNet34

Score	Class
10.4	red wine
8.3	perfume, essence
7.8	goblet
7.5	wine bottle
7.3	cocktail shaker
7.0	whiskey jug
6.8	hamper
6.5	table lamp
6.5	hourglass
6.4	cup

Configuration-1: 1 Segment 1 (LRP-ε) 34
$R_i = \Sigma_j (1 \frac{(1.0a_i)(1.0w_{ij})}{0.3+\Sigma_i (1.0a_i)(1.0w_{ij})})R_j$

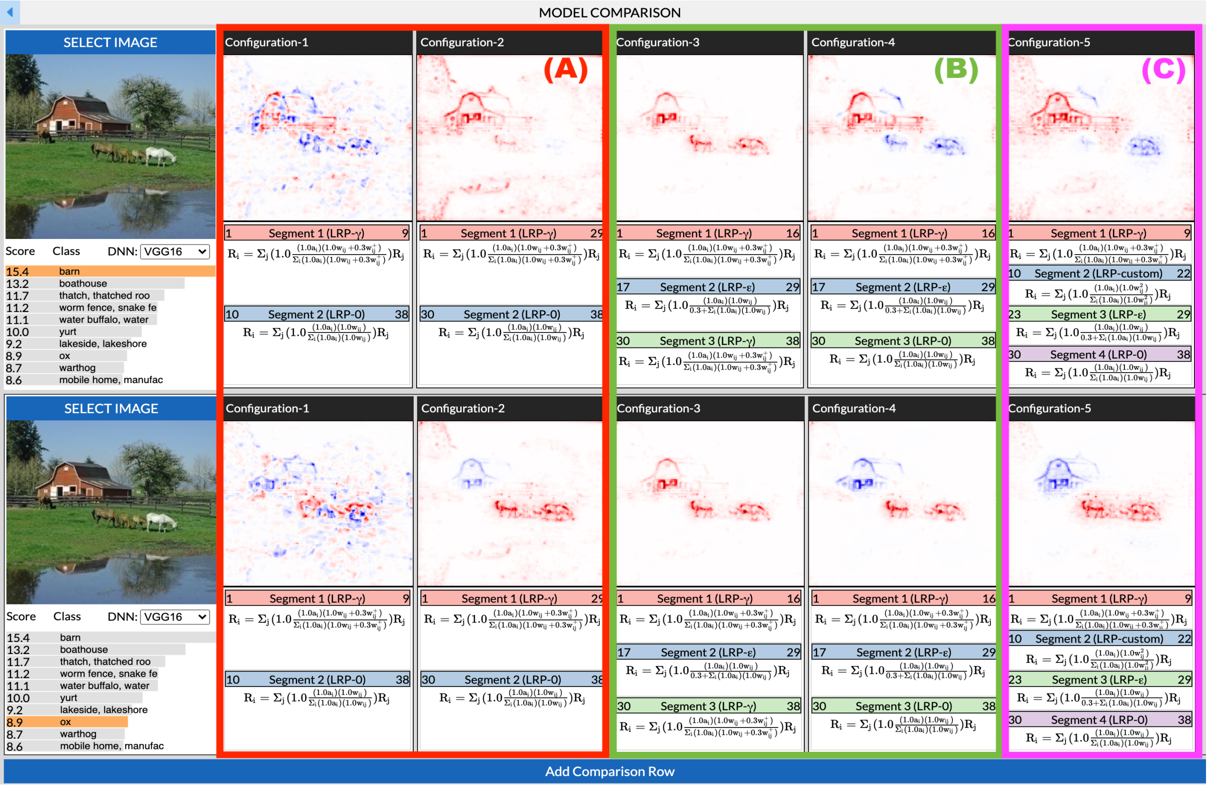
Configuration-2: 1 Segment 1 (LRP-γ) 34
$R_i = \Sigma_j (1 \frac{(1.0a_i)(0.4w_{ij}+1.0w_{ij}^+)}{\Sigma_i (1.0a_i)(0.4w_{ij}+1.0w_{ij}^+)})R_j$

Configuration-3: 1 Segment 1 (LRP-$\alpha_1\beta_0$) 34
$R_i = \Sigma_j (1 \frac{(1.0a_i)(1.0w_{ij}^+)}{\Sigma_i (1.0a_i)(1.0w_{ij}^+)})R_j$

Configuration-4: 1 Segment 1 (LRP-αβ) 34
$R_i = \Sigma_k (2 \frac{(a_i)(w_{ij}^+)}{\Sigma_i (a_i)(w_{ij}^+)} - 1 \frac{(a_i)(w_{ij}^-)}{\Sigma_i (a_i)(w_{ij}^-)})R_j$

Configuration-5: 1 Segment 1 (LRP-αβ) 34
$R_i = \Sigma_k (3 \frac{(a_i)(w_{ij}^+)}{\Sigma_i (a_i)(w_{ij}^+)} - 2 \frac{(a_i)(w_{ij}^-)}{\Sigma_i (a_i)(w_{ij}^-)})R_j$

Figure 3: *Model comparison view of popular LRP rules with an image ("A glass of wine with plastic bag"). Here ResNet34 is used with the top prediction class "red wine". The relevance heatmaps of Model1 to Model5, together with their LRP rule equations, are shown for comparison. Purple/orange color refers to positive/negative relevance pixels.*



Row 1 — DNN: VGG16

Score	Class
15.4	barn
13.2	boathouse
11.7	thatch, thatched roo
11.2	worm fence, snake fe
11.1	water buffalo, water
10.0	yurt
9.2	lakeside, lakeshore
8.9	ox
8.7	warthog
8.6	mobile home, manufac

Configuration-1: 1 Segment 1 (LRP-γ) 9
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
10 Segment 2 (LRP-0) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$

Configuration-2: 1 Segment 1 (LRP-γ) 29
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
30 Segment 2 (LRP-0) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$

Configuration-3: 1 Segment 1 (LRP-γ) 16
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
17 Segment 2 (LRP-ε) 29
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{0.3+\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$
30 Segment 3 (LRP-γ) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$

Configuration-4: 1 Segment 1 (LRP-γ) 16
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
17 Segment 2 (LRP-ε) 29
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{0.3+\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$
30 Segment 3 (LRP-0) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$

Configuration-5: 1 Segment 1 (LRP-γ) 9
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
10 Segment 2 (LRP-custom) 22
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}^2)}{\Sigma_i(1.0a_i)(1.0w_{ij}^2)})R_j$
23 Segment 3 (LRP-ε) 29
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{0.3+\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$
30 Segment 4 (LRP-0) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$

Row 2 — DNN: VGG16

Score	Class
15.4	barn
13.2	boathouse
11.7	thatch, thatched roo
11.2	worm fence, snake fe
11.1	water buffalo, water
10.0	yurt
9.2	lakeside, lakeshore
8.9	ox
8.7	warthog
8.6	mobile home, manufac

Configuration-1: 1 Segment 1 (LRP-γ) 9
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
10 Segment 2 (LRP-0) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$

Configuration-2: 1 Segment 1 (LRP-γ) 29
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
30 Segment 2 (LRP-0) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$

Configuration-3: 1 Segment 1 (LRP-γ) 16
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
17 Segment 2 (LRP-ε) 29
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{0.3+\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$
30 Segment 3 (LRP-γ) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$

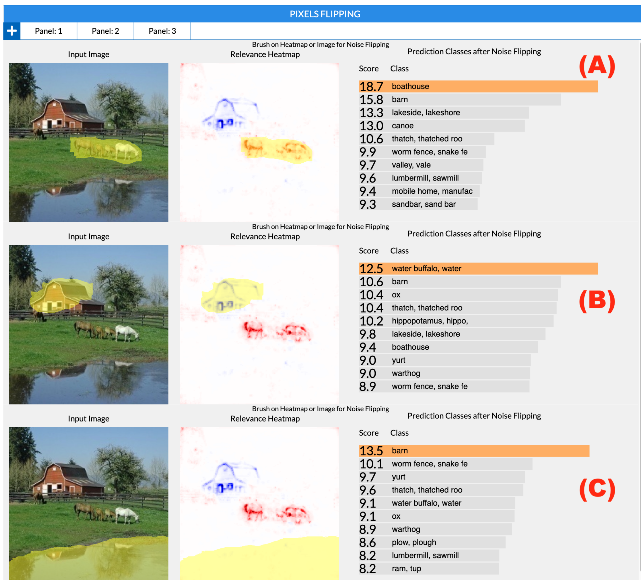
Configuration-4: 1 Segment 1 (LRP-γ) 16
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
17 Segment 2 (LRP-ε) 29
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{0.3+\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$
30 Segment 3 (LRP-0) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$

Configuration-5: 1 Segment 1 (LRP-γ) 9
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)}{\Sigma_i(1.0a_i)(1.0w_{ij}+0.3w_{ij}^+)})R_j$
10 Segment 2 (LRP-custom) 22
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij}^2)}{\Sigma_i(1.0a_i)(1.0w_{ij}^2)})R_j$
23 Segment 3 (LRP-ε) 29
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{0.3+\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$
30 Segment 4 (LRP-0) 38
$R_i = \Sigma_j (1.0 \frac{(1.0a_i)(1.0w_{ij})}{\Sigma_i(1.0a_i)(1.0w_{ij})})R_j$

Figure 4: *Model comparison view of multi-segment LRP models with an image ("barn on lake"). Row1: Study for class "barn". Row2: Study for class "ox". (A) Configuration 1 and 2 have two segments with different sizes; (B) Configuration 3 and 4 have three segments with the same size but different LRP rules at the last segment; (C) Configuration 5 has four segments defined in Fig. 2.*

ResNet34. Users can choose their preferred LRP models from the model manager (as Fig. 2A) and open this interface. Selecting a prediction class (e.g., "red wine") of an image (e.g. "a glass of wine with plastic bag"), the relevance heatmaps of these models are computed and visualized for comparison. Here a different color spectrum is used where purple and orange colors show positive and negative relevance, respectively. Each model also displays its segments and LRP rules. Users can also add multiple rows to study relevance results for different images in one or multiple classes.

Design rationale: First, multiple comparison rows are enabled for users to apply the same models to different images/classes, which is a key method in examining LRP model performance. Second, the heatmaps are shown side-by-side and the same color encoding of relevance is utilized so that their differences can be identified


PIXELS FLIPPING
+
Panel: 1
Panel: 2
Panel: 3
Brush on Heatmap or Image for Noise Flipping
Input Image
Relevance Heatmap
Prediction Classes after Noise Flipping
(A)
(B)
(C)

(A)

Score	Class
18.7	boathouse
15.8	barn
13.3	lakeside, lakeshore
13.0	canoe
10.6	thatch, thatched roo
9.9	worm fence, snake fe
9.7	valley, vale
9.6	lumbermill, sawmill
9.4	mobile home, manufac
9.3	sandbar, sand bar

(B)

Score	Class
12.5	water buffalo, water
10.6	barn
10.4	ox
10.4	thatch, thatched roo
10.2	hippopotamus, hippo,
9.8	lakeside, lakeshore
9.4	boathouse
9.0	yurt
9.0	warthog
8.9	worm fence, snake fe

(C)

Score	Class
13.5	barn
10.1	worm fence, snake fe
9.7	yurt
9.6	thatch, thatched roo
9.1	water buffalo, water
9.1	ox
8.9	warthog
8.6	plow, plough
8.2	lumbermill, sawmill
8.2	ram, tup

Figure 5: *Relevance-based pixel flipping study with image "barn on lake". Users brush on the image or the relevance heatmap to flip pixels, so as to study new CNN prediction results. Multiple rows are added with different flipping operations for comparison study.*

easily. Third, each model's segments and LRP rules are shown for easy observation and comparison of these model details.

6.3. Interactive Pixel Flipping

Based on LRP results (e.g., Fig. 2B), users can open a pixel flipping interface (for **V5**), as illustrated in Fig. 5. Multiple rows, Fig. 5A-Fig. 5C, can be added to compare different pixel flipping operations. At each row, the input image and its relevance heatmap are presented. Users can brush over the input image to flip pixels. They can also brush on the relevance heatmap directly to flip pixels according to the relevance information. The brushed pixels are highlighted on both images. While brushing, new prediction results are automatically computed and visualized for examination. Sec. 7.3 discusses the example in detail.

Design rationale: Pixel flipping is designed to link output to "features" on the input images. It is important to flip different image parts and investigate their prediction results. Their differences help users understand how CNN works with respect to image contents and relevance. Therefore, multiple brushing rows are implemented with the same layout to facilitate immediate comparison. Two brushing options on either the image or the heatmap are synchronized (brushing on either one also highlights the same pixels on another one). This design directly links the relevance and the image content information for effective flipping.

6.4. Visual Neuron Ablation

Fig. 6 shows the interface to perform relevance-based neuron ablation. Fig. 6A is the model result view from an active LRP model. A similar layer ruler (Fig. 6B) is provided for users to choose a convolutional layer. Then, three distribution charts (Fig. 6B) visualize *neuron points* of the selected layer based on their positive relevance scores, negative relevance scores, and CNN activation scores. Users can select individuals or groups of the neurons (points) in either chart. Two neuron matrices (Fig. 6C) further enumerate and visualize the positive and negative relevance scores, where each cell represents a neuron and the selected neurons are highlighted. Users can hover over the cells to see relevance information, and then add/remove specific neurons to/from a selection. The relevance heatmaps of each selected neuron are presented (Fig. 6D), together with its activation map. Finally, a new CNN prediction result after ablating the selected neurons is shown in Fig. 6E for investigation. Please see Sec. 7.4 for the details of this example.

Design rationale: The positive relevance score and negative relevance score of a neuron are two important factors for users to select neurons of interest in a layer. Note that one neuron has both negative and positive scores (see Sec. 3.3). A neuron's activation represents its behavior in CNN. These scores are presented to guide users in ablation. They are separated in the visualizations (Fig. 6B) while the three charts are coordinated for interaction. Users can easily select neurons on one chart and then identify their distributions on all three charts. The matrix view (Fig. 6C) compensates for the charts by providing detailed neuron information and supporting users to adjust individual neurons. The matrix-based visualization is applied because it can easily handle varying numbers of neurons in different layers. Relevance heatmaps (Fig. 6D) are displayed side-by-side with activation heatmaps, which directly represent neurons' behaviors in both forward (CNN) and backward (LRP) processes.

6.5. System Implementation

VisLRPDesigner has been applied on different CNN models, including VGG [SZ15], AlexNet [KSH12], and ResNet [HZRS16]. A Web-based prototype is made publicly available based on a pre-trained VGG-16 by the ImageNet dataset [DDS*09]. The system is built with an Intel i7-9700K CPU and 32GB memory, and an NVidia GTX 1070 GPU with 8GB texture memory. The forward-pass computation is performed through PyTorch [KK17] with CUDA on the GPU. We also leverage GPU computation with CUDA for the implementation of LRP backpropagation. Relevance scores and heatmaps are created and then transferred to the browser for visualization with Node.js [Tai13] and D3.js [BOH11].

7. Use Cases of VisLRPDesigner

7.1. Case 1: Studying Popular LRP Rules

A primary use of VisLRPDesigner is to visually explore popular LRP models. It is demanded by novice users (or students) in learning LRP concepts and gaining firsthand experience of different LRP backpropagation rules. Users can select these rules, change their parameter values, and compare them with different input images. Fig. 3 shows five models with the following rules (Model-1): $LRP-\varepsilon$ with $\varepsilon = 0.3$; (Model-2): $LRP-\gamma$ rule with $\gamma = 0.3$; (Model-3): $LRP-\alpha_1\beta_0$ with $\alpha = 1$ and $\beta = 0$; (Model-4): $LRP-\alpha_2\beta_1$ with $\alpha = 2$ and $\beta = 1$; and (Model-5): $LRP-\alpha_3\beta_2$ with $\alpha = 3$ and $\beta = 2$. Users can observe the differences in their corresponding LRP equations. In Fig. 3, a class “red wine” is selected to study the relevance of “a glass of wine and bag” image. It can be realized that Model-2 $LRP-\gamma$ can discover clearer edges of the glass and bag than the basic rule $LRP-\varepsilon$ in Model-1. Model-2 removes the most of negative contribution pixels in Model-1. In addition, $LRP-\alpha_1\beta_0$ (Model-3) keeps all relevance positive since the weight is positively filtered. Users can further compare different α and β values to understand their effects in LRP results. $LRP-\alpha_2\beta_1$ (Model-4) leads to closer result to $LRP-\gamma$ (Model-2). Comparing with Model-3, it justifies the suggested model behavior in [MSM18]: “*if ... the heatmaps are too diffuse, replace the rule LRP-α1β0 by LRP-α2β1...*”. Using $LRP-\alpha_3\beta_2$ in Model-5 further takes out more small details. In fact, either from Model-3 to Model-1 or from Model-3 to Model-5, the contribution of gradually suppressing the pixels of “plastic bag” in the background is coming from the negative weights kept in the original kernel mixed with positive weight or in the separated inhibitory term.

7.2. Case 2: Exploring Customized LRP Models

LRP designers and experienced users can configure composite LRP models by applying LRP rules on multiple segments of CNN layers. In Fig. 4, five different configurations are shown with the “barn on lake” image. Both Row1 and Row2 use this image but perform relevance study over two different classes: “barn” and “ox”, respectively. It is interesting that for this image, VGG identifies horses as oxen, which might be related to the bowing postures by the horses. Please note that the operation on the “ox” class actually involves horses in the image.

Fig. 4A shows two LRP models designed with two segments, both using $LRP-\gamma$ on Segment1 and $LRP-0$ on Segment2. Configuration-1 defines Segment1 at layers 1-10 and Segment2 at layers 11-38, while Configuration-2 has Segment1 at layers 1-30 and Segment2 at layers 31-38. It can be realized that Configuration-1 has unsatisfied relevance results. Their negative and positive relevance pixels on the heatmaps do not present a meaningful explanation for either “barn” or “ox” classes. In contrast, Configuration-2 performs very well. In Row1, the barn house is discovered while the horses are not emphasized in the relevance heatmap. In Row2, the horses (i.e., ox class) are identified with high positive relevance, and the barn is realized with negative contributions. This example shows that different segment sizes can lead to very different LRP behaviors.

In Fig. 4B, Configuration-3 and Configuration-4 are defined on three fixed segments. In Configuration-3, $LRP-\gamma$, $LRP-\varepsilon$, and $LRP-\gamma$ are applied on Segment1, 2 and 3, respectively. In Configuration-4, $LRP-\gamma$, $LRP-\varepsilon$, and $LRP-0$ are used on Segment1, 2 and 3, respectively. Configuration-3 fails to achieve good class discriminative results, since both “barn” and “ox” classes are related to the pixels of the barn house and horses, although its $LRP-\varepsilon$ in Segment 2 removes many noise pixels in Fig. 4(A). In contrast, Configuration-4 creates very good class discriminative results with a different $LRP-0$ in Segment 3. In Row1, it identifies the barn house as a positive contributor and the horses as a negative contributor. In Row2, it discovers the horses with positive relevance and the barn house with negative relevance. The reason is: Configuration-3 deploys $LRP-\gamma$ on Segment 3 including fully connected layers, which enhances the positive weight by adding $0.3w^+$ and thus suppresses the magnitude of negative weights. This causes “ox” and “barn” are not discriminated against in Row1 and Row2, while $LRP-0$ does not introduce the effect.

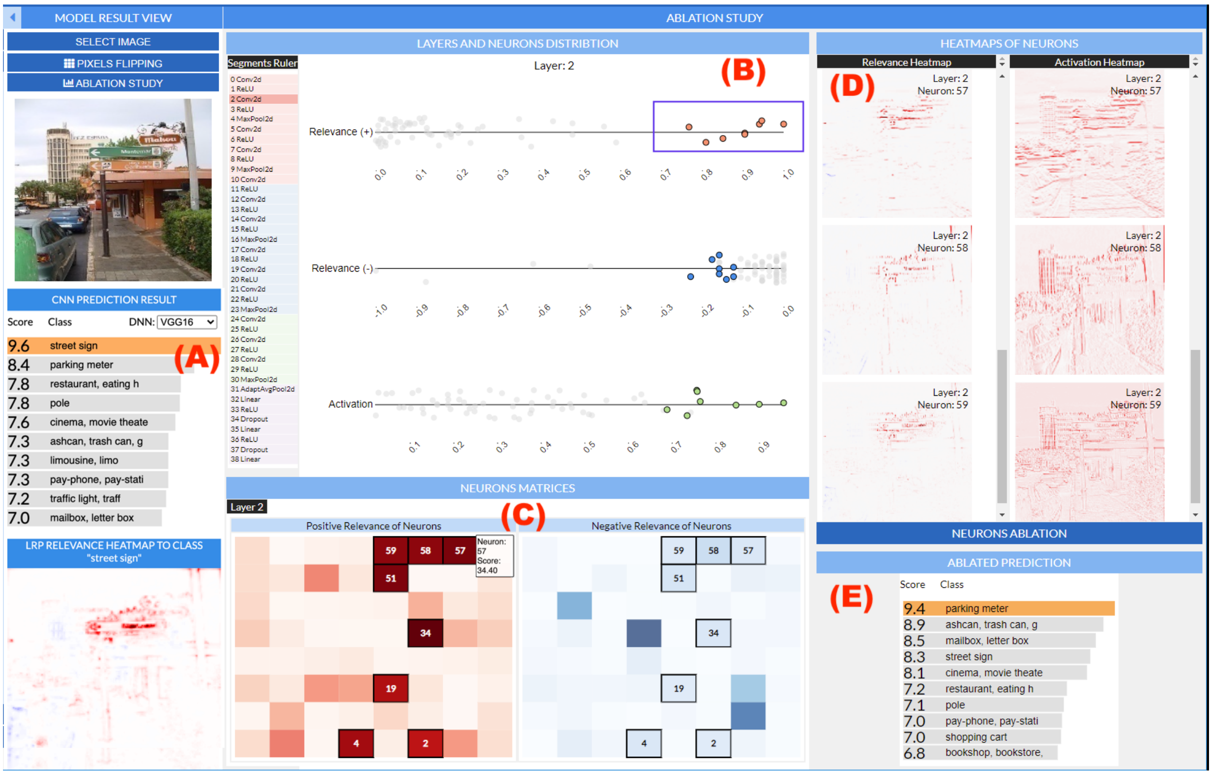
In Fig. 4C, four segments are defined in Configuration-5 which uses $LRP-\gamma$, $LRP-\varepsilon$, and $LRP-0$ in Segment1, Segment3, and Segment4 (similar to Segment 1-3 in Configuration-4). A custom LRP rule is designed and inserted as Segment2. The parameter settings of this configuration are shown in Fig. 2. In comparison to Configuration-4, this model detects more positive pixels to the target class. This example shows the exploratory process of the LRP model design.

7.3. Case 3: Relevance Based Pixel Flipping

VisLRPDesigner allows users to analyze CNN prediction by performing pixel flipping with the help of relevance information. In Fig. 5, users apply three different pixel flipping operations on different parts of the “barn on lake” image. Here the relevance heatmap of “ox” class from Configuration-4 of Fig. 4 is selected. In Fig. 5A, users brush on the relevance heatmap to remove positive contributor pixels. The new CNN prediction result shows the top class as “boathouse”, which reflects the effect of CNN prediction after removing horses. In Fig. 5B, the barn house with negative relevance is removed. The new prediction shows top classes “water buffalo” and “ox”, which helps understand the prediction behavior. Finally, in Fig. 5C, users directly brush on the input image to remove the pond. The new result shows “barn” and “worm fence” as top classes. Here, “boat house”, which is the second class in the original classification (see Fig. 2), is no longer discovered. It indicates how the water surface contributes to the classification.

7.4. Case 4: Relevance Based Neuron Ablation

By selecting a “street view” image, as shown in Fig. 6A, users find the prediction result with the top three classes as “street sign”, “parking meter”, and “restaurant”. By selecting class “street sign”, the relevance heatmap shows high relevance pixels of signs to this class. For the ablation study, users can click on the ruler to select the convolution layer 2. The distribution charts show the neuron points in this layer. Users select a group of neuron points with high positive relevance scores, as shown inside the purple box in Fig. 6B. It can be seen these neurons also have large activations (green points) and small negative relevance values (blue points). Users further explore these neurons in the matrices of layer 2 in Fig. 6C. By observing Fig. 6D, it can be seen that Neuron 57, 58, 59 (more heatmaps of the selected neurons can be observed by scrolling down further) show different neuron activations but their high-relevance pixels are mostly located on the street signs. Since the layer’s LRP heatmap is a result of aggregation by all heatmaps of neurons in that layer, Fig. 6D allows users to visually check the individual heatmaps of neurons to see how the layer’s heatmap is decomposed at the neuron level. That compensates the relevance scores in Fig. 6B by showing their specific spatial distributions. Fig. 6E shows the new prediction result after these neurons in layer 2 are removed from the CNN computation. Now “street sign” is not the top class while “parking meter” becomes the top class. Since the most relevant neurons of



Score	Class
9.6	street sign
8.4	parking meter
7.8	restaurant, eating h
7.8	pole
7.6	cinema, movie theate
7.3	ashcan, trash can, g
7.3	limousine, limo
7.3	pay-phone, pay-stati
7.2	traffic light, traff
7.0	mailbox, letter box

Score	Class
9.4	parking meter
8.9	ashcan, trash can, g
8.5	mailbox, letter box
8.3	street sign
8.1	cinema, movie theate
7.2	restaurant, eating h
7.1	pole
7.0	pay-phone, pay-stati
7.0	shopping cart
6.8	bookshop, bookstore,

Figure 6: *Relevance-based neuron ablation study with an image ("street view"). (A) Relevance result view for a top-class "street sign"; (B) Neuron points from a selected layer 2 showing their positive relevance, negative relevance, and activation distributions. Users select a group of neurons (in a purple rectangle); (C) Neuron matrices for detailed study of positive/negative relevance; (D) Relevance heatmaps and activation heatmaps of selected neurons; (E) Prediction results after ablation with a top-class "parking meter".*

the target class "street sign" are ablated, it makes sense that the prediction score (without softmax function applied) of this class drops. Instead, when low relevant neurons are removed the impact is not significant. Thus, it provides a basic ablation analysis down to the level of neurons to verify the model performance. Advanced methods can be added for further neuron-wise analysis as future work and are summarized in Sec.9.

8. Evaluation by LRP Learners and Experts

User groups: Two LRP user groups with different tasks evaluate the software. Group 1 (G1) included 4 Ph.D. students in CS as LRP learners. They were familiar with DL toolboxes and visualization but had no background in LRP. Group 2 (G2) has three CS Ph.D. students and one professor who had done LRP-related research and published papers in medical image analysis and explainable AI. These LRP experts had abundant knowledge and experience of LRP model design and programming.

Comparison tool: There has no comprehensive VA system like VisLRPDesigner for LRP exploration. An online Interactive LRP Demo System (*abbr.* InterLRP) has been developed in a famous LRP webpage (*heatmapping.org*), which only has limited functions for LRP tutoring. We used it to partly compare with VisLRPDesigner in the level of basic functions.

Table 2: *LRP User Evaluation of InterLRP and VisLRPDesigner.*

Functions 0(poor) - 9(excellent)	InterLRP (Mean/SD)	VisLRPDesigner (Mean/SD)
Part I: LRP Model Design:		
Predefined LRP models	8.4/1.5	9/0
LRP parameter setting	7.8/1.8	8.4/1.5
Image and relevance visualization	7.8/1.8	8.7/0.5
Define segments for composite LRP	N.A.	8.9/0.4
Multi-segment and rule visualization	N.A.	8.7/0.3
Intermediate relevance heatmaps	N.A.	8.6/0.5
User model management	N.A.	9/0
Part II: LRP Model Comparison:		
Multiple model selection	N.A.	8.7/0.7
Images and classes selection	N.A.	8.7/0.8
Comparative visualization of multi-models	N.A.	9.8/0.5
Part III: Relevance-based pixel flipping:		
Brushing on image and heatmap	N.A.	8.7/0.8
Flipping result analysis	N.A.	8/2.6
Comparative visualization of multi-flippings	N.A.	8.7/0.5
Part IV: Relevance-based ablation study:		
Neuron relevance distribution view	N.A.	8.1/0.8
Neuron selection with matrix view	N.A.	8.3/0.9
Ablation result analysis	N.A.	8.4/0.7

Procedure and tasks: For G1, we first introduced LRP and showed them basic LRP codes and InterLRP demo. This step was skipped for G2. Then, we discussed our motivation and introduced VisLRPDesigner. The users were guided to explore InterLRP and VisLRPDesigner through Web browser for at least half an hour with pretrained VGG16 neural network and images from ImageNet.

For G1, they were asked to find LRP settings that can clearly discover input features of given predictions in a few images. They first used popular LRP rules with adjusted parameter, and then designed different composite LRPs with multiple segments. For G2, they first tested the system in a similar way to G1. Then, they were asked to design a new LRP model which provided better results in comparison to popular models.

Both groups employed InterLRP and VisLRPDesigner for the tasks with about half an hour. Then, they evaluated each major function of VisLRPDesigner for its visual design and effectiveness with a score of 0 (poor) to 9 (excellent). These functions were categorized into four parts: model design, model comparison, pixel flipping, and ablation study. These functions were also evaluated if they exist in InterLRP. Moreover, users described their experiences and gave comments for limitations and suggestions.

Scores: The questions and mean scores are shown in Table 2 with standard deviation (SD). InterLRP has three basic functions in model design, and other functions it does not include are shown as N.A. (not available). VisLRPDesigner gained better scores than InterLRP, achieving mean scores of more than 8 out of 9 which indicated satisfaction. We noted that there was no big difference in G1 and G2 for the scores, so we reported mean/SD together.

Feedback: The contribution of VisLRPDesigner were agreed. For G1, the LRP concept was a little hard to consume at the beginning, but they were happy that VisLRPDesigner gave them an intuitive understanding for quickly learning the technology. In G2, the professor with abundant LRP experience said "Brandnew approach to support many new capabilities. This is a profound LRP design tool for the AI community. I haven't seen anything like this before for a model customization." "This is the framework that scientists and researchers are looking for". We summarized the feedback as:

- *Comparison of direct coding, InterLRP, and VisLRPDesigner:* G2 group (who programmed LRP before) mentioned: "it is too tedious and hard to implement and debug". Both G1 and G2 indicated the limitation of InterLRP such as "Demo interface is straightforward and simple, but the functionality of modifying LRP is very limited." "It doesn't allow the user to check intermediate results". They agreed that "VisLRPDesigner is more friendly to let more people learn and use it."
- *VisLRPDesigner interface:* Both groups were satisfied with VisLRPDesigner's comprehensive functionality. They mostly liked: "a clean and clear dashboard". "The bar charts for parameter visualization are very straightforward but effective". "I like the way to change parameters. The layout for target class and model comparison is intuitive and easy to follow." "I really like the feature that the formula is explicitly displayed. And the interactive update of formulas while changing the parameters makes the design very easy to conduct. When comparing with other models, the summary view is also very helpful".
  Limitation: They suggested to provide more guidance and explanation through labels and popup windows, such as "A set of lines connecting the heatmaps of adjacent segments ... help users to see how relevance value propagates.".
- *Application of visual pixel flipping:* This is very important for understanding LRPs. They said: "This is a vital function for model understanding using the LRP. It directly visualizes the impact of changes in the input image." "A very practical feature to include."
  Limitation: They complained that the flipping brushing was fixed and could be improved with finer adjustment. They suggested to add multiple brushes and zooming.
- *Use of ablation study:* The effectiveness of relevance-based ablation was agreed, such as "I find this function very useful and insightful." "With this function, the VisLRPDesigner is a powerful tool for model compression".
  Limitation: They said that the interface needed some learning efforts before using, and the ablation can be further linked to LRP refinement.
- *System Speed:* All agreed that the system well supported interactive operations with real-time response and visualization refreshment.

9. Discussion and Future Work

In addition to the limitations described above, we discuss a few important issues and future directions:

- *LRP parameter space:* The ten LRP parameters and CNN layer segments form a parameter data space. Different parameter combinations create different relevance discoveries (e.g., small structures or large profiles). The space may be analyzed and visualized to help users understand the performance of these combinations, with respect to image classes and CNN models. This is out of the scope of VisLRPDesigner as an LRP design tool but leads to a critical direction in future work.
- *LRP for DNNs:* More technique options could be added in the future, including layers fusing for Batch-Normalization [GHK*20] vs. its bypassing in this system, bias switching, and different attribute-discriminative LRP approaches [NGC*20]. Besides, While designed mostly for CNNs, LRP is also used in other DNNs such as natural language processing [AHM*16], EEG analysis [SLSM16], and audio classification [BAL*18]. VisLRPDesigner is working on CNNs for image datasets but needs to be extended to these various scenarios.
- *LRP inside neurons and layers:* While VisLRPDesigner shows intermediate relevance heatmaps, more visual debugging tools may be developed so that LRP models can be refined in the levels of neurons and layers.
- *Quantitiative metrics:* Relevance study mostly relies on visual observation of heatmaps. A very challenging topic is to develop quantitative metrics and recommendations and incorporate them into the visual system.

10. Conclusion

VisLRPDesigner is a visualization tool to facilitate an easy and efficient design of the emerging LRP approach in deep learning explanation. Users can explore multiple models with relevance data visualization, together with the integrated visual analysis tools of pixel flipping and neuron ablation. VisLRPDesigner has been shared with researchers and general LRP users. We will promote it in the communities of computer vision and deep learning, and perform software enhancement.

Acknowledgement

The authors wish to thank the anonymous reviewers. This work was partly supported by KSU graduate assistantship, BNL LDRD grant 18-009 and ECP CODAR project 17-SC-20-SC.

References

[AHM*16] Arras L., Horn F., Montavon G., Müller K.-R., Samek W.: Explaining predictions of non-linear classifiers in NLP. In *Proceedings of the 1st Workshop on Representation Learning for NLP* (Berlin, Germany, Aug. 2016), Association for Computational Linguistics, pp. 1–7. URL: https://www.aclweb.org/anthology/W16-1601, doi:10.18653/v1/W16-1601. 1, 2, 10

[AMJ17] Alvarez-Melis D., Jaakkola T.: A causal framework for explaining the predictions of black-box sequence-to-sequence models. *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing* (2017), 412–421. URL: http://aclweb.org/anthology/D17-1042, doi:10.18653/v1/D17-1042. 2

[BAL*18] Becker S., Ackermann M., Lapuschkin S., Müller K.-R., Samek W.: Interpreting and Explaining Deep Neural Networks for Classification of Audio Signals. URL: http://arxiv.org/abs/1807.03418, arXiv:1807.03418. 1, 2, 10

[BBM*15] Bach S., Binder A., Montavon G., Klauschen F., Müller K.-R., Samek W.: On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLOS ONE 10*, 7 (2015), e0130140. URL: https://doi.org/10.1371/journal.pone.0130140, doi:10.1371/journal.pone.0130140. 1, 3

[BOH11] Bostock M., Ogievetsky V., Heer J.: D3 Data-Driven Documents. *IEEE Transactions on Visualization and Computer Graphics 17*, 12 (2011), 2301–2309. doi:http://doi.ieeecomputersociety.org/10.1109/TVCG.2011.185. 7

[CL18] Choo J., Liu S.: Visual Analytics for Explainable Deep Learning. *IEEE Computer Graphics and Applications 38*, 4 (2018), 84–92. doi:10.1109/MCG.2018.042731661. 1, 2

[CPCS20] Cashman D., Perer A., Chang R., Strobelt H.: Ablate, Variate, and Contemplate: Visual Analytics for Discovering Neural Architectures. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 863–873. arXiv:1908.00387, doi:10.1109/TVCG.2019.2934261. 2

[DDS*09] Deng J., Dong W., Socher R., Li L., Li K., Li F.-F.: ImageNet: A Large-Scale Hierarchical Image Database. *Computer Vision and Pattern Recognition* (2009), 248–255. doi:10.1002/col.5080170616. 7

[GCS*19] Gu J., Chowdhury M., Shin K. G., Zhu Y., Jeon M., Qian J., Liu H., Guo C.: Tiresias : A GPU Cluster Manager for Distributed Deep Learning. *Proceedings of the 16th USENIX Conference on Networked Systems Design and Implementation* (2019), 485—-500. 1, 2

[GHK*20] Guillemot M., Heusele C., Korichi R., Schnebert S., Chen L.: Breaking batch normalization for better explainability of deep neural networks through layer-wise relevance propagation, 2020. arXiv:2002.11018. 10

[GRNT16] Grün F., Rupprecht C., Navab N., Tombari F.: A Taxonomy and Library for Visualizing Learned Features in Convolutional Neural Networks. *Int. Conf. Mach. Learn. Workshop Vis Deep Learn.* (2016). 2

[GYT19] Gu J., Yang Y., Tresp V.: Understanding Individual Decisions of CNNs via Contrastive Backpropagation. *ACCV Conference, Lecture Notes in Computer Science 11363 LNCS* (2019), 119–134. arXiv:1812.02100, doi:10.1007/978-3-030-20893-6_8. 2

[GZL*20] Gou L., Zou L., Li N., Hofmann M., Shekar A. K., Wendt A., Ren L.: Vatld: A visual analytics system to assess, understand and improve traffic light detection. *IEEE Transactions on Visualization and Computer Graphics* (2020), 1–1. doi:10.1109/TVCG.2020.3030350. 2

[HKPC19] Hohman F., Kahng M., Pienta R., Chau D. H.: Visual Analytics in Deep Learning: An Interrogative Survey for the Next Frontiers. *IEEE Transactions on Visualization and Computer Graphics 25*, 8 (2019), 2674–2693. arXiv:1801.06889, doi:10.1109/TVCG.2018.2843369. 1, 2

[HMK*19] Hoyer L., Munoz M., Katiyar P., Khoreva A., Fischer V.: Grid saliency for context explanations of semantic segmentation. In *Advances in Neural Information Processing Systems* (2019), Wallach H., Larochelle H., Beygelzimer A., d'Alché-Buc F., Fox E., Garnett R., (Eds.), vol. 32, Curran Associates, Inc. URL: https://proceedings.neurips.cc/paper/2019/file/6950aa02ae8613af620668146dd11840-Paper.pdf. 1, 2

[HPRP20] Hohman F., Park H., Robinson C., Polo Chau D. H.: Summit: Scaling Deep Learning Interpretability by Visualizing Activation and Attribution Summarizations. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 1096–1106. arXiv:1904.02323, doi:10.1109/TVCG.2019.2934659. 2

[HZRS16] He K., Zhang X., Ren S., Sun J.: Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2016), pp. 770–778. doi:10.1109/CVPR.2016.90. 7

[IKU19] Iwana B. K., Kuroki R., Uchida S.: Explaining convolutional neural networks using softmax gradient layer-wise relevance propagation. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (2019), 4176–4185. 1, 2

[KAKC18] Kahng M., Andrews P. Y., Kalro A., Chau D. H. P.: ActiVis: Visual Exploration of Industry-Scale Deep Neural Network Models. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 88–97. doi:10.1109/TVCG.2017.2744718. 2

[Kar] Karpathy A.: ConvNetJS: Deep Learning in your browser. *https://cs.stanford.edu/people/karpathy/convnetjs/*. 2

[KJL19] Kang S.-H., Jung H., Lee S.-W.: Interpreting undesirable pixels for image classification on black-box models. *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (2019), 4250–4254. 1, 2

[KK17] Ketkar N., Ketkar N.: Introduction to PyTorch. *Deep Learning with Python* (2017), 195–208. doi:10.1007/978-1-4842-2766-4_12. 7

[KL17] Koh P. W., Liang P.: Understanding black-box predictions via influence functions. In *Proceedings of the 34th International Conference on Machine Learning* (06–11 Aug 2017), Precup D., Teh Y. W., (Eds.), vol. 70 of *Proceedings of Machine Learning Research*, PMLR, pp. 1885–1894. URL: http://proceedings.mlr.press/v70/koh17a.html. 2

[KSH12] Krizhevsky A., Sutskever I., Hinton G. E.: Imagenet classification with deep convolutional neural networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1* (Red Hook, NY, USA, 2012), NIPS'12, Curran Associates Inc., p. 1097–1105. 7

[LCY14] Lin M., Chen Q., Yan S.: Network in network. *CoRR abs/1312.4400* (2014). arXiv:1312.4400. 2

[LLMX20] Li H., Lin Y., Mueller K., Xu W.: Interpreting galaxy deblender gan from the discriminator's perspective. In *Advances in Visual Computing* (Cham, 2020), Springer International Publishing, pp. 239–250. 2

[LMM18] Lillian P. E., Meyes R., Meisen T.: Ablation of a Robot's Brain: Neural Networks Under a Knife. URL: http://arxiv.org/abs/1812.05687, arXiv:1812.05687. 4

[LSL*17] Liu M., Shi J., Li Z., Li C., Zhu J., Liu S.: Towards Better Analysis of Deep Convolutional Neural Networks. *IEEE Transactions* *on Visualization and Computer Graphics 23*, 1 (2017), 91–100. doi:10.1109/TVCG.2016.2598831. 2

[LWB*19] Lapuschkin S., Wäldchen S., Binder A., Montavon G., Samek W., Müller K. R.: Unmasking Clever Hans predictors and assessing what machines really learn. *Nature Communications 10*, 1 (2019). arXiv:1902.10178, doi:10.1038/s41467-019-08987-4. 1, 2, 3

[MBL*19] Montavon G., Binder A., Lapuschkin S., Samek W., Müller K. R.: Layer-Wise Relevance Propagation: An Overview. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 11700 LNCS* (2019), 193–209. doi:10.1007/978-3-030-28954-6_10. 3

[MLB*17] Montavon G., Lapuschkin S., Binder A., Samek W., Müller K.-R.: Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition 65* (2017), 211–222. doi:https://doi.org/10.1016/j.patcog.2016.11.008. 3

[MMD*19] Murugesan S., Malik S., Du F., Koh E., Lai T. M.: DeepCompare: Visual and Interactive Comparison of Deep Learning Model Performance. *IEEE Computer Graphics and Applications 39*, 5 (2019), 47–59. doi:10.1109/MCG.2019.2919033. 2

[MSM18] Montavon G., Samek W., Müller K. R.: Methods for interpreting and understanding deep neural networks. *Digital Signal Processing: A Review Journal 73* (2018), 1–15. arXiv:1706.07979, doi:10.1016/j.dsp.2017.10.011. 1, 3, 8

[NGC*20] Nam W.-J., Gur S., Choi J., Wolf L., Lee S.-W.: Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. *Proceedings of the AAAI Conference on Artificial Intelligence 34*, 03 (Apr. 2020), 2501–2508. URL: https://ojs.aaai.org/index.php/AAAI/article/view/5632, doi:10.1609/aaai.v34i03.5632. 10

[NIAN19] Nikulin D., Ianina A., Aliev V., Nikolenko S.: Free-lunch saliency via attention in atari agents. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (2019), pp. 4240–4249. doi:10.1109/ICCVW.2019.00522. 1, 2

[PHV*18] Pezzotti N., Höllt T., Van Gemert J., Lelieveldt B. P., Eisemann E., Vilanova A.: DeepEyes: Progressive Visual Analytics for Designing Deep Neural Networks. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 98–108. doi:10.1109/TVCG.2017.2744358. 2

[RFFT17] Rauber P. E., Fadel S. G., Falcão A. X., Telea A. C.: Visualizing the Hidden Activity of Artificial Neural Networks. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (2017), 101–110. doi:10.1109/TVCG.2016.2598838. 2

[SBLM17] Samek W., Binder A., Lapuschkin S., Müller K.: Understanding and comparing deep neural networks for age and gender classification. In *2017 IEEE International Conference on Computer Vision Workshops (ICCVW)* (2017), pp. 1629–1638. doi:10.1109/ICCVW.2017.191. 1, 3

[SCD*17] Selvaraju R. R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D.: Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization. *Proceedings of the IEEE International Conference on Computer Vision 2017-Octob* (2017), 618–626. arXiv:1610.02391, doi:10.1109/ICCV.2017.74. 2

[SCD*20] Selvaraju R. R., Cogswell M., Das A., Vedantam R., Parikh D., Batra D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision 128*, 2 (Feb 2020), 336–359. URL: https://doi.org/10.1007/s11263-019-01228-7, doi:10.1007/s11263-019-01228-7. 2

[SDBR15] Springenberg J., Dosovitskiy A., Brox T., Riedmiller M.: Striving for simplicity: The all convolutional net. In *ICLR (workshop track)* (2015). URL: http://lmb.informatik.uni-freiburg.de/Publications/2015/DB15a. 2

[SKK*19] Stergiou A., Kapidis G., Kalliatakis G., Chrysoulas C., Poppe R., Veltkamp R.: Class feature pyramids for video explanation. In *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)* (Los Alamitos, CA, USA, oct 2019), IEEE Computer Society, pp. 4255–4264. URL: https://doi.ieeecomputersociety.org/10.1109/ICCVW.2019.00524, doi:10.1109/ICCVW.2019.00524. 1, 2

[SLSM16] Sturm I., Lapuschkin S., Samek W., Müller K. R.: Interpretable deep neural networks for single-trial EEG classification. *Journal of Neuroscience Methods 274* (2016), 141–145. arXiv:1604.08201, doi:10.1016/j.jneumeth.2016.10.008. 1, 2, 10

[SMV*19] Samek W., Montavon G., Vedaldi A., Hansen L. K., Muller K.-R. (Eds.): *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*. Springer, 2019. doi:10.1007/978-3-030-28954-6. 1, 2

[STN*16] Smilkov D., Thorat N., Nicholson C., Reif E., Viégas F. B., Wattenberg M.: Embedding projector: Interactive visualization and interpretation of embeddings. *In Proc. Neural Inf. Process. Syst. Workshop Interpretable ML Complex Syst.* (2016). 2

[SVZ14] Simonyan K., Vedaldi A., Zisserman A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. *CoRR abs/1312.6034* (2014). 2

[SZ15] Simonyan K., Zisserman A.: Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings* (2015). 7

[Tai13] Taixeira P.: Professional Node.js - Building Javascript Based Scalable Software. *John Wiley & Son, Inc.* (2013), 1–371. doi:10.1007/s13398-014-0173-7.2. 7

[VKS20] Vishnusai Y., Kulakarni T. R., Sowmya Nag K.: Ablation of Artificial Neural Networks. 453–460. doi:10.1007/978-3-030-38040-3_52. 2, 4

[WSW*18] Wongsuphasawat K., Smilkov D., Wexler J., Wilson J., Mané D., Fritz D., Krishnan D., Viégas F. B., Wattenberg M.: Visualizing Dataflow Graphs of Deep Learning Models in TensorFlow. *IEEE Transactions on Visualization and Computer Graphics 24*, 1 (2018), 1–12. doi:10.1109/TVCG.2017.2744878. 2

[WTS*21] Wang Z. J., Turko R., Shaikh O., Park H., Das N., Hohman F., Kahng M., Chau D. H. P.: Cnn explainer: Learning convolutional neural networks with interactive visualization. In *IEEE Transactions on Visualization and Computer Graphics (TVCG)* (2021), IEEE. URL: https://poloclub.github.io/cnn-explainer/. 2

[WYC*19] Wang Q., Yuan J., Chen S., Su H., Qu H., Liu S.: Visual Genealogy of Deep Neural Networks. *IEEE Transactions on Visualization and Computer Graphics* (2019), 1–1. doi:10.1109/tvcg.2019.2921323. 2

[YCN*15] Yosinski J., Clune J., Nguyen A., Fuchs T., Lipson H.: Understanding neural networks through deep visualization. In *Deep Learning Workshop, International Conference on Machine Learning (ICML)* (2015). 1

[ZF14] Zeiler M. D., Fergus R.: Visualizing and understanding convolutional networks. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics) 8689 LNCS*, PART 1 (2014), 818–833. arXiv:1311.2901, doi:10.1007/978-3-319-10590-1_53. 2

[ZKL*16] Zhou B., Khosla A., Lapedriza A., Oliva A., Torralba A.: Learning Deep Features for Discriminative Localization. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition 2016-Decem* (2016), 2921–2929. arXiv:1512.04150, doi:10.1109/CVPR.2016.319. 2

[ZXZ*17] Zhong W., Xie C., Zhong Y., Wang Y., Xu W., Cheng S., Mueller K.: Evolutionary visual analysis of deep neural networks. URL: https://www3.cs.stonybrook.edu/~mueller/papers/Evolutionary%20Visual%20Analysis%20of%20Deep%20Neural%20Networks.pdf. 2