

# Finding Representative Interpretations on Convolutional Neural Networks

Peter Cho-Ho Lam<sup>1</sup>, Lingyang Chu<sup>\* 2</sup>, Maxim Torgonskiy<sup>1</sup>, Jian Pei<sup>3</sup>, Yong Zhang<sup>1</sup>, and Lanjun Wang<sup>1</sup>

<sup>1</sup>Huawei Canada Technologies Co., Ltd.

<sup>2</sup>McMaster University

<sup>3</sup>Simon Fraser University

{cho.ho.lam, maxim.torgonskiy, yong.zhang3, lanjun.wang}@huawei.com  
chul9@mcmaster.ca  
jpei@cs.sfu.ca

## Abstract

*Interpreting the decision logic behind effective deep convolutional neural networks (CNN) on images complements the success of deep learning models. However, the existing methods can only interpret some specific decision logic on individual or a small number of images. To facilitate human understandability and generalization ability, it is important to develop representative interpretations that interpret common decision logics of a CNN on a large group of similar images, which reveal the common semantics data contributes to many closely related predictions. In this paper, we develop a novel unsupervised approach to produce a highly representative interpretation for a large number of similar images. We formulate the problem of finding representative interpretations as a co-clustering problem, and convert it into a submodular cost submodular cover problem based on a sample of the linear decision boundaries of a CNN. We also present a visualization and similarity ranking method. Our extensive experiments demonstrate the excellent performance of our method.*

## 1. Introduction

Interpretability becomes more and more important for machine learning [40, 48]. As a fundamental deep learning model, the interpretability of convolutional neural networks (CNNs) [26, 34] has been intensively explored [9, 45, 47, 50, 61, 72]. For example, many interpretation methods [12, 48, 50, 56] have been proposed to interpret the specific decision logic of a CNN on a single input image or a small group of images. However, the challenge of interpretation on CNNs for image classification is still far from being settled. An interpretation accommodating only one individual image or a small group of images is highly sensitive to the noise contained in the interpreted image [20],

<sup>\*</sup>Peter Cho-Ho Lam and Lingyang Chu contribute equally in this work.  
Please refer to [35] for the complete version of this paper.

and thus cannot be robustly reused to interpret the predictions on a large group of unseen images [3, 16, 57].

To tackle the challenge, we advocate seeking *representative interpretations* [7], which interpret the common decision logic of a CNN. A representative interpretation is more convincing, because it can be validly reused to interpret the same decision logic that governs the predictions on a large number of unseen images [7]. Moreover, a representative interpretation also provides a deeper insight into the common semantic of a large number of similar images [21, 32, 70]. Such common semantics are generally believed to be captured by CNNs to achieve excellent prediction performance [18, 72, 73].

However, interpreting the common decision logic of a CNN on a large group of similar images in an unsupervised manner is a novel problem that has not been systematically studied in literature [21]. Can we straight-forwardly extend the existing interpretation methods to produce representative interpretations on a CNN? Unfortunately, many existing methods [9, 12, 45, 50, 48] focus on interpreting the specific decision logic of a CNN on a single input image or a small group of images, and thus cannot produce representative interpretations. More critically, it is difficult to produce a representative interpretation without knowing a large group of similar images that are predicted by a common decision logic of a CNN. As discussed in Section 2, some concept-based interpretation methods summarize the common decision-making concepts from a group of conceptually similar images. However, those methods are either limited to sophisticatedly customized CNN models [11, 70], or require a large set of conceptually annotated images [18, 73] that are too expensive to obtain in most real world applications [21].

In this paper, we propose the novel unsupervised task of finding representative interpretations on convolutional neural networks. The goal is to find and visualize the common decision logic of a CNN that governs the predictions on an input image and a large set of similar images. We make a series of technical contributions as follows.

We model the common decision logic of a CNN on a group of similar images as a decision region formed by a set of pieces of the decision boundaries of the CNN. All images contained in this decision region are predicted as the same class by the same decision logic characterized by the decision boundary pieces. We formulate our task as a co-clustering problem to simultaneously find the largest group of similar images and the corresponding decision region containing them. We further convert the co-clustering problem into an NP-hard submodular cost submodular cover problem, and develop an efficient heuristic method to solve it without requiring any conceptual image annotations or any modifications to CNNs. To produce understandable interpretations for the images contained in a decision region, we visualize the boundaries of the decision region as heat maps to identify important image regions for predictions. We also rank the similar images according to their semantic distances to the input image. Last, we conduct extensive experiments to examine the interpretation quality and reusability of our representative interpretations.

## 2. Related Works

Our work is related to the four major types of existing interpretation methods for CNNs in image classification.

The **gradient-based methods** [9, 45, 50, 52, 66, 67] produce interpretations on an input image by computing and visualizing different types of gradients. The early studies [2, 36, 38, 52, 54, 55, 56, 58, 66, 67] straight-forwardly visualize the gradient of the score of a predicted class with respect to an input image. The class activation mapping (CAM) methods [9, 45, 47, 50, 61, 72] use discriminative localization to produce continuous image regions as interpretations. The relevance score methods [5, 41, 44, 51] produce interpretations by back-propagating a set of relevance scores with respect to an input image.

All those methods focus on producing interpretations that are highly correlated to the prediction on a single input image. However, such interpretations are not representative, because they are highly sensitive to the noise contained in the images that are interpreted [20]. Thus, such interpretations cannot be effectively reused to interpret the predictions on a large group of unseen images [3, 16, 57].

The **model approximation methods** [12, 19, 27, 39, 48, 49, 59, 65] produce interpretations on a deep neural network by approximating it locally or globally with an interpretable agent model, such as decision trees [19] and linear classifiers [12, 48].

The local approximation methods, such as Openbox [12], LIME [48] and Anchors [49], produce interpretations by approximating the predictions of a neural network in a local similarity neighbourhood of a target image for interpretation. The interpretations produced by Openbox [12] and LIME [48] are only applicable to a very small neighbourhood, which largely limits their ability to produce representative interpretations [49]. Interpretations produced by Anchors [49] are applicable to a large neighbourhood. However, Anchors cannot produce effective interpretations for CNNs on images, because the applicable neighbourhoods for Anchors are not well-defined on images [40].

The global approximation methods [19, 27, 59, 65] attempt to approximate the entire neural network with an interpretable agent model. These methods intend to produce interpretations that apply to multiple similar images. However, the agent models are usually too simple to accurately approximate complex deep neural networks [12]. Therefore, most of these methods perform poorly on modern CNNs trained on complicated natural images [60].

The **conceptual interpretation methods** [18, 21, 32, 68, 69, 70, 73] aim to identify a set of concepts that contribute to the predictions on a pre-defined group of conceptually similar images. Here, a concept is either supervision information or learned from data using another model. TCAV [32], Net2Vec [18] and IBD [73] interpret a neural network’s internal state in terms of human-friendly concepts. These methods require a dataset with high-quality conceptual labels, which is difficult, if possible at all, to obtain in real world applications [21]. Some methods [70] use decision trees to learn concepts without using a dataset with conceptual labels. The learned concepts reveal the common decision logic of a group of similar images. However, these methods require sophisticated customization on deep neural networks, and thus suffer from major limitations in applicability to general CNNs.

ACE [21] uses a set of images in the same class to summarize important concepts by clustering similar image segments and removing outlying image segments. A subset of the summarized concepts can be straight-forwardly used to interpret the predictions on an input image. However, as shown in Section 6.3, ACE cannot produce representative interpretations because its interpretations on similar images are quite different due to the high sensitivity of the image segmentation method and the clustering process.

The **example-based methods** [8, 24, 30, 33, 63, 64] find exemplar images to interpret the decision of a deep neural network on a single input image. The exemplar images are similar to the input image, but they do not provide feature level interpretations to reveal the decision logic on the input image or the exemplar images [40]. Therefore, there is no guarantee that the exemplar images are predicted by the same decision logic as the input image.

Instead of finding explanatory example for each single input, the **prototype-based methods** [6, 31] summarize the entire model using a small number of instances as prototypes. However, the selection of prototypes considers very little about the decision process of the model. Thus they do not provide much insight on the model’s decision logic.

As discussed in Section 1, the existing methods cannot produce representative interpretations for CNNs on image classification, which is the challenge tackled by this paper.

## 3. Problem Formulation

In this section, we first discuss the idea of representative interpretations for image classification by CNN [26, 34]. Then, we formulate the problem of finding representative interpretation on CNNs.

### 3.1. Representative Interpretations

In this paper, we consider image classification using CNNs that adopt piecewise linear activation functions, such as MaxOut [23] and the family of ReLU [22, 25, 43], for hidden neurons. Denote by  $\mathcal{X}$  the space of input images. Let  $F : \mathcal{X} \rightarrow \mathbb{R}^C$  be a trained CNN to be interpreted and  $C$  the number of classes of images. We assume a set of unlabeled reference images  $R \subseteq \mathcal{X}$  that are used to generate representative interpretations.

For an input image  $x \in \mathcal{X}$ ,  $F$  produces a vector of prediction scores  $F(x) = [F_1(x), F_2(x), \dots, F_C(x)]$ , where  $F_i(x)$  is the predicted score of the  $i$ -th class before the normalization of the last softmax layer. The predicted class of  $x$  is  $Class(x) = \operatorname{argmax}_i F_i(x)$ .

Roughly speaking, an interpretation of a classification model is a description of the logics used by the model to produce the classification. While different approaches may employ different representations for interpretations, in image classification, for example, a common way to represent an interpretation for an image is a heat map that identifies the important regions leading to the prediction [9, 45, 47, 50, 61, 72]. Since interpretation is often related to generalization, an important idea to make interpretation more understandable is to seek for representative interpretations that are applicable to not only a specific image but also many similar images [7].

**Definition 1.** For an input image  $x \in \mathcal{X}$  predicted as  $Class(x)$  by  $F$ , a **representative interpretation** on  $x$  is an interpretation that reveals the common decision logic of  $F$  on  $x$  as well as a large set of reference images in  $R$ .

In order to construct representative interpretations, let us first consider how to model the decision logic of  $F$ .

For an input image  $x \in \mathcal{X}$ , denote by  $\psi(x)$  the feature map produced by the last convolutional layer of  $F$  and by  $\Omega = \{\psi(x) \mid x \in \mathcal{X}\}$  the space of feature maps. Following the existing studies [9, 47, 50, 61, 70], we model the decision logic of  $F$  on  $x$  by analyzing how the fully connected layers of  $F$ , denoted by  $G : \Omega \rightarrow \mathbb{R}^C$ , use the feature map  $\psi(x)$  to make the prediction on  $x$ . For any input image  $x$ , we have  $G(\psi(x)) = F(x)$ .

Since  $F$  adopts a piecewise linear activation function in all the hidden neurons, the decision logic of  $G$  can be characterized by a piecewise linear decision boundary that consists of connected pieces of decision hyperplanes [42, 46]. We call these hyperplanes **linear boundaries**, and denote by  $\mathcal{P}$  the set of the linear boundaries of  $G$ .

The linear boundaries in  $\mathcal{P}$  partition the space of feature maps  $\Omega$  into a large number of convex polytopes [42, 46], where each convex polytope defines a decision region that predicts all the images contained in the region to be the same class. The linear decision boundaries of the convex polytopes encode the decision logic of the fully connected layers of  $F$  [12].

Inspired by the above insights, we model the decision logic of  $F$  on an input image  $x \in \mathcal{X}$  by a convex polytope in  $\Omega$  that contains  $\psi(x)$ . This convex polytope must be induced by a subset of the linear boundaries of  $G$ , denoted by  $P(x) \subseteq \mathcal{P}$ . The convex polytope defines a decision region

such that the reference images contained in the region are also predicted as  $Class(x)$ .

To keep our presentation concise, we overload symbol  $P(x)$  to denote the decision region, that is, the convex polytope it defines.

When the context is clear, we say a feature map  $\psi(x')$  of an image  $x' \in \mathcal{X}$  is **contained** in  $P(x)$  if  $\psi(x')$  is contained in the convex polytope defined by  $P(x)$ . We also say  $x'$  is **covered** by  $P(x)$  if  $\psi(x')$  is contained in  $P(x)$ .

Since all the reference images covered by  $P(x)$  are predicted as the same class as  $x$  by the same set of decision boundaries, the contained images are sharing the same decision logic as  $x$ . Therefore,  $P(x)$  is a representative interpretation on  $x$  because it reveals the common decision logic of  $G$  when making predictions on  $x$  as well as all the reference images covered by  $P(x)$ .

One issue is that the linear boundaries in  $P(x)$  are in the space  $\Omega$  of feature maps, and thus are not easy to comprehend by human being. To tackle this issue, in Section 5.1, we will develop a method to visualize  $P(x)$  as meaningful heat maps that identify important regions of the covered images.

According to Carvalho et al. [7], the representativeness of an interpretation is the number of images that the interpretation is applicable to. Since the decision logic modeled by  $P(x)$  applies to all the images it covers, we define the **representativeness** of  $P(x)$  as the number of reference images covered by  $P(x)$ .

### 3.2. Finding Representative Interpretations

Given a trained CNN  $F$  and a set of reference images  $R \subseteq \mathcal{X}$ , for an input image  $x \in \mathcal{X}$  whose prediction by  $F$  is to be interpreted, the task of **finding representative interpretation** is to find  $P(x) \subseteq \mathcal{P}$  with the largest representativeness, and visualize  $P(x)$  as human comprehensible heat maps to identify image regions that are important for the prediction. Here, we focus on formulating a co-clustering problem that finds  $P(x)$  with the largest representativeness, and leave visualization of  $P(x)$  to Section 5.1.

Denote by  $P(x) \cap R$  the set of reference images covered by  $P(x)$ .  $|P(x) \cap R|$ , the number of reference images covered by  $P(x)$ , is exactly the representativeness of  $P(x)$ .

Denote by  $D(x) = \{x' \in R \mid Class(x') \neq Class(x)\}$  the set of reference images that are predicted by  $F$  to a class different from  $Class(x)$ . Let  $P(x) \cap D(x)$  be the set of images in  $D(x)$  covered by  $P(x)$ . The task of finding representative interpretations can be formulated as the following co-clustering problem [15].

$$\max_{P(x) \subseteq \mathcal{P}} |P(x) \cap R| \quad (1a)$$

$$\text{s.t. } |P(x) \cap D(x)| = 0 \quad (1b)$$

Here, the objective in Equation (1a) is to maximize the representativeness of  $P(x)$ , and the constraint in Equation (1b) requires  $P(x)$  to be a decision region that covers no reference image in  $R$  that is predicted to a class different from  $Class(x)$ .

The optimization problem in Equation (1) is a co-clustering problem because it simultaneously finds two

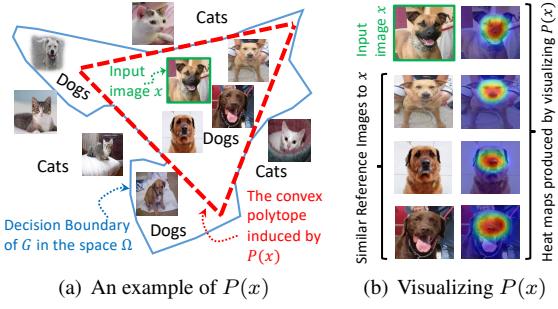


Figure 1. An example showing our ideas. The input image is marked by a green box. The other images are reference images.

clusters to optimize the representativeness of  $P(x)$ . One cluster is the set of similar images in  $P(x) \cap R$  that are predicted by the common decision logic characterized by  $P(x)$ . The other cluster is the set of linear boundaries in  $P(x)$  that characterizes the decision logic of  $G$  on all the images in  $P(x) \cap R$ . By solving the co-clustering problem, we can simultaneously find a maximal set of similar images including the input image and many reference images, as well as a representative interpretation  $P(x)$  that characterizes the common decision logic on these similar images.

**Example 1.** In Figure 1(a), the blue line segments are the piecewise linear decision boundaries of  $G$  in the space of  $\Omega$ . This decision boundary characterizes the decision logic of  $G$  in separating the feature maps of images of dogs and those of cats. We assume  $\Omega$  is a two dimensional space for simplicity.

Given the input image  $x$  of a dog in the green box in the figure, the objective of our co-clustering problem is to find a set of linear boundaries  $P(x)$ , such that the convex polytope induced by  $P(x)$  covers the input image  $x$ , and as many reference images of dogs as possible, but does not cover any reference image of cat. Following this objective, we can see that, in Figure 1(a), the optimal  $P(x)$  consists of the three linear boundaries in red dashed lines, which induce a triangular convex polytope that covers the largest set of images of dogs without covering any image of cat.

The linear boundaries in  $P(x)$  characterize the common decision logic of  $G$  on all the images of dogs covered by  $P(x)$ . Since  $P(x)$  covers a large number of images of dogs, the decision logic characterized by  $P(x)$  is highly representative, and can be generally applied to interpret the predictions on all the images of dogs covered.

Comparing to the existing methods that only interpret the prediction of the input image, our method shows similar reference images with conceptually similar heat maps to the input image. This makes interpretations even more effective and convincing.

## 4. Finding Practical Solutions

The optimization problem in Equation (1) cannot be exactly solved for large CNNs in practice, because the set of linear boundaries  $\mathcal{P}$  of a complex CNN is too large to ob-

tain, and we have to search the power set of  $\mathcal{P}$  to find the optimal solution to the combinatorial problem in Equation (1).

To tackle the challenge, we first obtain a sample  $\mathcal{Q}$  of  $\mathcal{P}$ , and use  $\mathcal{Q}$  to convert the co-clustering problem in Equation (1) into a submodular cost submodular cover (SCSC) problem. Then, we propose a greedy algorithm to find a good solution to the SCSC problem.

### 4.1. Converting into an SCSC Problem

In this section, we first introduce how to sample a subset of linear boundaries  $\mathcal{Q}$  from  $\mathcal{P}$ . Then, we discuss how to use  $\mathcal{Q}$  to convert the co-clustering problem in Equation (1) into an SCSC problem.

Denote by  $t \in \Omega$  a feature map, and by  $\langle W, t \rangle + b = 0$  a hyperplane in  $\Omega$  defined by the coefficients  $W$  and  $b$ . Here,  $\langle W, t \rangle$  is the inner product between the tensors  $W$  and  $t$ .

We can easily derive from [12] that, for any image  $x \in \mathcal{X}$ , the hyperplane induced by the following coefficients

$$\begin{cases} W = \frac{\partial \Delta G(t)}{\partial t} |_{t=\psi(x)} \\ b = \Delta G(t) - \langle W, t \rangle |_{t=\psi(x)} \end{cases} \quad (2)$$

is exactly a linear boundary  $\mathbf{h} \in \mathcal{P}$ . Here,  $\Delta G(t) := \max_1 G(t) - \max_2 G(t)$ , is the difference of the largest two prediction scores;  $\max_i G(t)$  is the  $i$ -th largest value in the vector  $G(t)$ .

To sample a subset of linear boundaries  $\mathcal{Q}$  from  $\mathcal{P}$ , we sample a subset of reference images from  $R$ , and apply Equation (2) on each of the sampled reference images to compute a linear boundary in  $\mathcal{P}$ .

By substituting the set of linear boundaries  $\mathcal{P}$  in Equation (1a) with the sampled set of linear boundaries  $\mathcal{Q}$ , we convert the optimization problem in Equation (1) into

$$\max_{P(x) \subseteq \mathcal{Q}} |P(x) \cap R| \quad (3a)$$

$$\text{s.t. } |P(x) \cap D(x)| \leq \delta, \quad (3b)$$

where  $\delta \geq 0$  is a relaxation parameter to ensure the above problem is feasible. We discuss the effect of  $\delta$  and how to compute  $\delta$  as follows.

Recall that the linear boundaries in  $\mathcal{P}$  partition  $\Omega$  into a set of convex polytopes, and the images covered by the same convex polytope are predicted by  $F$  as the same class [12, 42]. Since  $\mathcal{Q} \subseteq \mathcal{P}$ , the linear boundaries in  $\mathcal{Q}$  also partition  $\Omega$  into a set of convex polytopes. However, due to the missing linear boundaries in  $\mathcal{P} \setminus \mathcal{Q}$ , some images covered by different neighbouring convex polytopes induced by  $\mathcal{P}$  may be covered by the same convex polytope induced by  $\mathcal{Q}$ . Therefore, for some convex polytopes induced by  $\mathcal{Q}$ , the images covered in the same convex polytope may not be predicted by  $F$  as the same class. In consequence,  $|P(x) \cap D(x)|$  may be larger than zero. Thus we use  $\delta$  to keep the constraint in Equation (3b) valid, such that the problem in Equation (3) is feasible.

Given an image  $x \in \mathcal{X}$  to interpret, the minimum value of  $|P(x) \cap D(x)|$ , denoted by  $\tau(x)$ , is the number of images in  $D(x)$  whose feature maps lie on the same side as

---

**Algorithm 1:** A Greedy Algorithm

---

**Input:** An input image  $x \in \mathcal{X}$ , a trained CNN  $F$ , and a set of reference images  $R \subseteq \mathcal{X}$ .  
**Output:** A solution  $P(x)$  to the SCSC problem in Equation (3).

```

1 Initialization:  $P(x) \leftarrow \emptyset, \delta \leftarrow \tau(x)$ .
2 while  $|P(x) \cap D(x)| > \delta$  do
3    $\mathbf{h} \leftarrow \underset{\mathbf{h} \in \mathcal{Q} \setminus P(x)}{\operatorname{argmin}} \frac{|P(x) \cap R| - |(P(x) \cup \{\mathbf{h}\}) \cap R|}{|P(x) \cap D(x)| - |(P(x) \cup \{\mathbf{h}\}) \cap D(x)|}$ .
4   Add linear boundary  $\mathbf{h}$  into the set of linear boundaries  $P(x)$ .
5 end
6 return  $P(x)$ .

```

---

$\psi(x)$  with respect to every linear boundary  $\mathbf{h} \in \mathcal{Q}$ . We can easily compute  $\tau(x)$  by enumeration.

Obviously,  $P(x) = \mathcal{Q}$  is a feasible solution to the problem in Equation (3) when  $\delta \geq \tau(x)$ . We simply set  $\delta = \tau(x)$ , which yields spectacular interpretation performance in our experiments.

We prove that the problem in Equation 3 is a **submodular cost submodular cover (SCSC) problem** [28].

**Theorem 1.** *For a given input image  $x \in \mathcal{X}$ , the problem defined in Equation (3) is a submodular cost submodular cover (SCSC) problem.*

*Proof.* See Appendix A in [35].

## 4.2. Solving the SCSC Problem

An SCSC problem is NP-hard [13], thus we solve our problem in Equation (3) by the simple greedy algorithm [62] summarized in Algorithm 1.

In each iteration of Algorithm 1, a boundary  $\mathbf{h} \in \mathcal{Q} \setminus P(x)$  is selected to minimize

$$\frac{|P(x) \cap R| - |(P(x) \cup \{\mathbf{h}\}) \cap R|}{|P(x) \cap D(x)| - |(P(x) \cup \{\mathbf{h}\}) \cap D(x)|}. \quad (4)$$

The numerator of Equation (4) is the number of reference images that are originally covered by  $P(x)$  but are removed from  $P(x)$  after we add  $\mathbf{h}$  into  $P(x)$ . Since our goal in Equation (3a) is to maximize the number of reference images covered by  $P(x)$ , we need to select an  $\mathbf{h}$  that removes as few images in  $R$  from  $P(x)$  as possible.

Similarly, the denominator is the number of images in  $D(x)$  that are originally covered by  $P(x)$ , but are removed from  $P(x)$  after we add  $\mathbf{h}$  into  $P(x)$ . To quickly find a solution  $P(x)$  that satisfies the constraint in Equation (3b), we need to select an  $\mathbf{h}$  that removes as many images in  $D(x)$  from  $P(x)$  as possible.

Following the above intuition, the greedy algorithm [62] in Algorithm 1 adopts Equation (4) to find an  $\mathbf{h} \in \mathcal{Q}$  that minimizes the numerator and maximizes the denominator. In step 3 of Algorithm 1, when there are multiple linear boundaries that induce a numerator equal to zero, we pick the linear boundary  $\mathbf{h}$  with the largest denominator to add into  $P(x)$ . The denominator is positive before Algorithm 1 converges, because we can always find a linear boundary

to remove some images from  $P(x) \cap D(x)$  until  $|P(x) \cap D(x)| \leq \delta$ .

The iteration continues until the constraint in Equation (3b) is satisfied. Algorithm 1 converges in at most  $|\mathcal{Q}|$  iterations.

## 5. Visualization and Similarity Ranking

In this section, we first introduce how to visualize  $P(x)$  as heat maps on the images covered by  $P(x)$ . Then, we discuss how to rank the images covered by  $P(x)$  according to their semantic distances to the input image  $x \in \mathcal{X}$ .

### 5.1. Visualizing $P(x)$ on Covered Images

Since the heat maps for all the images covered by  $P(x)$  can be produced in the same way, we take the input image  $x$  as an example to introduce how to visualize  $P(x)$ .

To interpret why  $F$  predicts the input image  $x \in \mathcal{X}$  as *Class*( $x$ ), we visualize  $P(x)$  as a heat map that identifies the important regions in  $x$  for the prediction. Since  $F$  predicts every image covered by  $P(x)$  as *Class*( $x$ ), we interpret the decision logic of  $F$  on  $x$  by showing why  $x$  is covered by  $P(x)$ . This is equivalent to showing why  $\psi(x)$  is contained in  $P(x)$ . The key idea is to visualize  $P(x)$  as a heat map to identify the important regions in  $x$  that make  $\psi(x)$  contained in  $P(x)$ .

Recall that every linear boundary in  $P(x)$  contributes to why  $\psi(x)$  is contained in  $P(x)$ . We first visualize every linear boundary in  $P(x)$  as a heat map on  $x$ . Then we produce the heat map of  $P(x)$  by taking an average of the heat maps of all the linear boundaries.

Denote by  $\langle W, t \rangle + b = 0, t \in \Omega$ , the hyperplane of a linear boundary  $\mathbf{h}$  in  $P(x)$ . We visualize  $\mathbf{h}$  as a heat map on  $x$  in a similar way as Grad-CAM [50]. First, if  $\langle W, \psi(x) \rangle + b < 0$ , then we set  $W = -W$  and  $b = -b$  to make sure  $\langle W, \psi(x) \rangle + b > 0$ . Second, we compute the **channel weight**  $\alpha_k$  of the  $k$ -th channel of  $\psi(x)$  by

$$\alpha_k = \frac{1}{Z} \sum_i \sum_j W_{ij}^k, \quad (5)$$

where  $W_{ij}^k$  is the entry on the  $i$ -th row, the  $j$ -th column and the  $k$ -th channel in the tensor  $W$ , and  $Z$  is the total number of entries  $W_{ij}^k$  for every channel  $k$ . Third, we compute a raw heat map by  $raw = \text{ReLU}(\sum_k \alpha_k \psi^k(x))$ , where  $\psi^k(x)$  is the  $k$ -th channel of the feature map  $\psi(x)$ . Last, we rescale and normalize the raw heat map to produce a heat map on  $x$  in the same way as Grad-CAM. This heat map identifies the important regions in  $x$  that show why  $\mathbf{h}$  predicts  $\psi(x)$  to be contained in  $P(x)$ .

Since  $P(x)$  contains many linear boundaries, and every linear boundary contributes to why  $\psi(x)$  is contained in  $P(x)$ , we compute the heat map for  $P(x)$  as the average of the heat maps of all the linear boundaries. Since the images covered by  $P(x)$  are sharing exactly the same set of linear boundaries in  $P(x)$ , the channel weights computed from the  $W$  of each linear boundary in  $P(x)$  are exactly the same for all the images covered by  $P(x)$ . Therefore, we reuse the set of channel weights computed from the input

image  $x$  to produce heat maps for the other images covered by  $P(x)$  in the same way as  $x$ .

As demonstrated later in Section 6.4, due to the high representativeness of  $P(x)$ , the heat maps produced by our method generalize well to unseen images, and they usually highlight conceptually similar parts of different images covered by the same  $P(x)$ .

## 5.2. Ranking Similar Images

Now, let us rank the images covered by  $P(x)$  according to their semantic distances to the input image  $x \in \mathcal{X}$ .

Recall that the linear boundaries in  $P(x)$  identify the common decision logic of  $F$  on all the images covered by  $P(x)$ . The common semantic of the covered images is captured by the **normal vectors** of the hyperplanes of these linear boundaries. Following this insight, we compute the semantic distance between the input image  $x$  and another image  $x'$  covered by  $P(x)$  by first projecting their feature maps  $\psi(x)$  and  $\psi(x')$  onto the normal vectors, and then computing the L1-distance between the projected coordinates.

Denote by  $\vec{W}_h$  the normal vector of the hyperplane of a linear boundary  $h \in P(x)$ . We compute the semantic distance between  $x$  and  $x'$  by

$$Dist(x, x') = \sum_{h \in P(x)} \left| \langle \vec{W}_h, \psi(x) \rangle - \langle \vec{W}_h, \psi(x') \rangle \right|, \quad (6)$$

where  $\langle \vec{W}_h, \psi(x) \rangle$  and  $\langle \vec{W}_h, \psi(x') \rangle$  are the projected coordinates of  $x$  and  $x'$ , respectively. We rank the images covered by  $P(x)$  according to their semantic distance to  $x$  in ascending order, such that a more similar image with a smaller semantic distance is ranked higher in the list.

As demonstrated in our case studies in Section 6.3, showing similar images to the input image with conceptually similar heat maps makes our interpretations more convincing than showing the heat map for the input image alone.

## 6. Experiments

In this section, we present a systematic experimental study. Limited by space, only selected results are reported in this section. Please refer to the Appendix in [35] for more experimental results.

### 6.1. Baseline Methods

We conduct extensive experiments to evaluate the performance of our representative interpretation (RI) method, and compare with the state-of-the-art baselines including Automatic Concept-based Explanation (ACE) [21] and three CAM-based methods, Grad-CAM [50], Grad-CAM++ [9] and Score-CAM [61]. We use the publicly available source code of the baseline methods. The python notebook of RI can be found in [1]. All baseline methods are compared using their optimal parameters. For RI, the number of sampled linear boundaries in  $Q$  does not have much influence on the interpretation performance, thus we simply use  $|Q| = 50$  for all the experiments.

We focus on evaluating the representativeness of interpretations, that is, how well an interpretation for an input image can be reused to interpret the predictions on new images that are similar to the input image [7]. Since the baseline methods cannot directly reuse their interpretations on new images, we propose the following extensions to reuse their interpretations.

The CAM-based methods use the channel weights computed from an input image  $x \in \mathcal{X}$  to generate the interpretation (i.e., heat map) for  $x$ . To reuse the interpretation generated by the CAM-based methods to interpret the prediction on a new image  $x_{new} \in \mathcal{X}$ , we reuse the channel weights computed from the input image  $x$ , and follow the same heat map producing steps of the CAM-based methods to generate the interpretation for  $x_{new}$ . Recall that our method also reuses the set of channel weights computed from the input image  $x$  to generate interpretations for new images covered by  $P(x)$ . It is fair to compare our method with the above extensions of the CAM-based methods.

ACE [21] uses a set of input images in the same class to summarize important concepts. It clusters similar image segments and removes outlying image segments. Every cluster of the similar image segments represents a concept. To reuse these summarized concepts to interpret the prediction on a new image  $x_{new} \in \mathcal{X}$ , we use exactly the same pipeline as ACE to segment  $x_{new}$  and remove outlying image segments. The remaining image segments are assigned to their nearest concepts, and are used to produce the interpretation on  $x_{new}$  in the same way as ACE.

Another issue is that the baseline methods cannot identify the group of similar images that an interpretation can validly apply to. To tackle this issue, following [21, 71], we use Euclidean distance in the space of  $\Omega$  to find the set of similar images whose feature maps are the closest to the feature map of the input image.

### 6.2. Datasets and Evaluation Metrics

We use the following four public datasets to evaluate the performance of all the compared methods.

The ASIRRA dataset [17] contains two classes of images including ‘dog’ and ‘cat’. The original dataset contains over 3 million images and a subset of them was made publicly available through Kaggle. The training data and the testing data contain 25,000 and 12,500 images, respectively.

The Gender Classification (GC) dataset [10] contains two classes of images including ‘male’ and ‘female’. The training data and the testing data contain 47,009 and 11,649 images, respectively.

The Retinal OCT Images (RO) dataset [29] contains four classes of images, including one class of normal retina as ‘NORMAL’, and three classes of retinal disease, ‘CNV’, ‘DME’ and ‘DRUSEN’. The training data and the testing data contain 108,309 and 1,000 images, respectively.

The FOOD dataset [4] consists of two classes of images, ‘food’ and ‘non-food’. The training data and the testing data contain 3,000 and 1,000 images, respectively.

For the large datasets ASIRRA, GC and RO, we randomly sample 2,500 images from each class of the training data as the set of reference images  $R$  to generate representa-

tive interpretations. For the small dataset FOOD, we use all the training images as the set of reference images. We call a set of reference images a **reference dataset**. The ground truth labels of all reference images are not used to generate interpretations.

To evaluate how well an interpretation can be reused to interpret the predictions on unseen images, we randomly sample 1,000 images from the testing data of each dataset to produce an **unseen dataset**.

For each dataset, we train a VGG-19 model [53] as the target model to interpret. Every model is trained using the model parameters pre-trained on ImageNet [14] as initialization.

Next, we introduce how to quantitatively evaluate the representativeness of an interpretation. Denote by  $S \subseteq \mathcal{X}$  the set of unseen evaluation images. For each interpretation method, we first generate some interpretations using the reference dataset, and then reuse these interpretations to identify the important regions in each image in  $S$ . Then, we adopt the evaluation metrics in [9, 47] to evaluate the representativeness of these interpretations. More specifically, we define

$$\text{Average Drop (AD)} = \frac{1}{|S|} \sum_{e \in S} \frac{\max(0, Y_c(e) - Y_c(e'))}{Y_c(e)} \quad (7)$$

and

$$\text{Average Increase (AI)} = \frac{1}{|S|} \sum_{e \in S} \mathbb{I}_{Y_c(e) < Y_c(e')}, \quad (8)$$

where  $Y_c(e)$  is the prediction score for class  $c$  on an image  $e \in S$ , and  $e'$  is a masked image produced by keeping 20% of the most important pixels in  $e$  according to the interpretation on  $e$ . The rest 80% pixels in  $e'$  are masked in black.

Since ACE uses image segments instead of heat maps as interpretations, it is hard to keep exactly 20% pixels in a masked image. Thus, we iteratively select the most important segments until more than 20% pixels are kept.

A small AD and a large AI mean that the interpretation on the input image can be validly reused to accurately identify important regions on the images in  $S$ . This further indicates a good representativeness performance of the interpretation.

### 6.3. A Case Study

In Figure 2, the first row shows an input image and the top-4 similar reference images in  $R$  found by RI. The third row shows the same input image and the top-4 similar reference images found by the baseline methods using Euclidean distance. All images are predicted as ‘female’ with high prediction scores. The top-4 similar images found by RI are all adult females, which are conceptually very similar to the input image. Among the similar images found by the baselines, the fourth image highlighted by the red dashed box is a baby, which is very different from the input image.

RI achieves a superior performance in finding similar images because the proposed co-clustering problem simultaneously finds the interpretation for the input image, as well

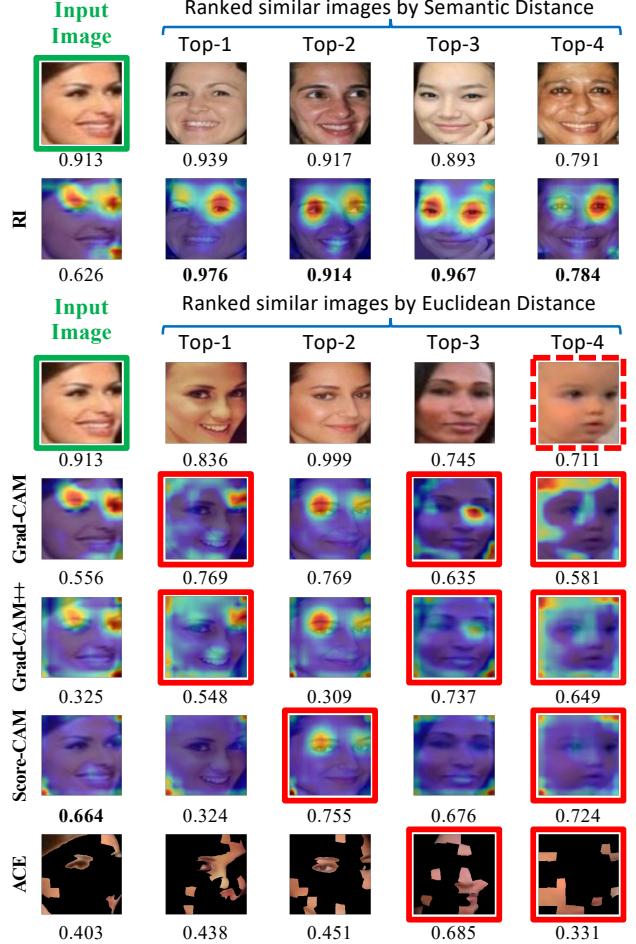


Figure 2. A case study on the GC dataset. The first row shows the input image and the similar reference images in  $R$  found by RI. The third row shows the same input image and the similar reference images found by the baseline methods using Euclidean distance in  $\Omega$ . The other rows are the interpretations of different methods. The numbers under the images in the first and the third rows are the prediction scores of the ‘female’ class. The numbers under the interpretations are the prediction scores of the ‘female’ class on the masked images that are produced by keeping 20% most important pixels. A higher prediction score means a better interpretation.

as a set of images sharing the same interpretation. Since all the images share the same interpretation, they are usually very similar in concepts. The baseline methods are not designed to find similar images sharing the same interpretations. A straight-forward extension is to find the nearest neighbours using Euclidean distance in the space of  $\Omega$ , which, however, may not be effective in finding images with similar concepts.

To examine the representativeness of the interpretations of all the methods, we analyze whether an interpretation on the input image can be successfully reused on similar images to identify conceptually similar parts as the input image. As shown in the second row of Figure 2, the interpretation computed by RI from the input image is highly representative because it consistently identifies the eyes and

eyebrows of the females in the input image and the similar reference images. These parts are significant features that distinguish a female from males in the GC dataset. Finding similar reference images with the same interpretation as the input image makes the interpretations produced by RI highly understandable and effective in practice.

The baseline methods cannot achieve representative interpretations. Consider the images highlighted by the red boxes in Figure 2. The interpretations produced by the baseline methods cannot consistently identify the same conceptual parts for the input image and the similar reference images. Especially, the interpretations produced by ACE are the most inconsistent due to the high sensitivity and randomness of image segmentation and clustering. Such inconsistency makes the interpretations ineffective.

Figure 2 also shows the prediction score of ‘female’ for every masked image produced by keeping 20% of the most important pixels identified by an interpretation. A higher prediction score means a better interpretation quality. Score-CAM achieves the best prediction score on the input image because it is designed to maximize the prediction score of the masked input image. However, as shown in the second last row of Figure 2, the interpretation produced by Score-CAM cannot be effectively reused to achieve a high prediction score on the similar reference images.

RI achieves the second best prediction score on the input image and the best scores on all the similar reference images. This clearly indicates that the representative interpretation produced by RI is much more effective to be reused on similar images than the baselines.

#### 6.4. Representativeness on Unseen Images

We quantitatively evaluate how well interpretations generated on a reference dataset can be reused to interpret predictions on unseen images. For each method, we first generate a set of interpretations  $V$  on the reference dataset. Then, we use metrics AD and AI to evaluate the representativeness of  $V$  on the unseen dataset.

To generate  $V$  for each of RI, Grad-CAM, Grad-CAM++ and Score-CAM, we first apply  $k$ -means [37] on the reference dataset to find  $|V|$  clusters in the space  $\Omega$ . Then, we find the reference images that are the closest to each of the cluster centers, and use them as the input images to produce the interpretations in  $V$ . Since ACE requires a set of reference images as the input to produce one interpretation, we first find the  $|V|$  sets of reference images that form the  $|V|$  clusters produced by the  $k$ -means algorithm, and then use each set of reference images as an input to ACE to produce an interpretation in  $V$ .

The number of interpretations in  $V$  is set to  $|V| = 10 \times$  Number of classes for all methods, that is,  $|V| = 40$  for RO, and  $|V| = 20$  for the other datasets.

To evaluate the representativeness of the group of interpretations  $V$  produced by RI, for each image  $e \in S$ , we first find the polytope in  $v \in V$  that covers  $e$ , and then we use  $v$  to interpret  $e$ . If an unseen image is not covered by the convex polytope of any interpretation in  $V$ , we find its nearest cluster center in the space  $\Omega$ , and use the polytope it generates to interpret  $e$ .

Datasets	Methods	mAD (%)	mAI (%)
ASIRRA	RI	<b>2.00</b> $\pm$ 0.06	30.72 $\pm$ 0.85
	Grad-CAM	2.30 $\pm$ 0.09	<b>32.90</b> $\pm$ 0.39
	Grad-CAM++	3.07 $\pm$ 0.02	29.12 $\pm$ 0.42
	Score-CAM	3.72 $\pm$ 0.09	28.10 $\pm$ 0.28
	ACE	36.27 $\pm$ 0.78	1.36 $\pm$ 0.2
GC	RI	<b>12.70</b> $\pm$ 0.32	<b>6.12</b> $\pm$ 0.19
	Grad-CAM	23.01 $\pm$ 0.50	4.40 $\pm$ 0.33
	Grad-CAM++	29.53 $\pm$ 0.36	3.46 $\pm$ 0.40
	Score-CAM	23.88 $\pm$ 0.37	3.30 $\pm$ 0.38
	ACE	46.37 $\pm$ 0.93	2.26 $\pm$ 0.12
RO	RI	<b>12.57</b> $\pm$ 0.17	<b>27.54</b> $\pm$ 0.50
	Grad-CAM	35.34 $\pm$ 0.78	22.02 $\pm$ 0.33
	Grad-CAM++	21.04 $\pm$ 0.33	23.24 $\pm$ 0.59
	Score-CAM	14.12 $\pm$ 1.39	25.60 $\pm$ 0.34
	ACE	65.36 $\pm$ 0.91	4.44 $\pm$ 0.39
FOOD	RI	<b>8.08</b> $\pm$ 0.33	<b>26.66</b> $\pm$ 0.48
	Grad-CAM	9.22 $\pm$ 0.64	23.34 $\pm$ 1.12
	Grad-CAM++	9.04 $\pm$ 0.27	22.14 $\pm$ 1.34
	Score-CAM	10.44 $\pm$ 1.84	19.56 $\pm$ 0.89
	ACE	39.88 $\pm$ 2.13	24.80 $\pm$ 0.76

Table 1. The mean Average Drop (mAD) and the mean Average Increase (mAI) of all the methods.

Since the baseline methods cannot find the set of unseen images that an interpretation can apply to, we assign each unseen image in the unseen dataset to its nearest cluster center in the space  $\Omega$ .

Table 1 shows the Average Drop (AD) and Average Increase (AI) of all the methods. We run 5 independent experiments for each method, and report the mean and standard deviation of mAD and mAI.

RI achieves the best mAD performance on all the datasets and the best mAI performance on most of the datasets. This clearly demonstrates the superior performance of RI in finding representative interpretations that reveals the common decision logic on the predictions of a large number of similar unseen images.

## 7. Conclusion

In this paper, we develop a novel unsupervised method to produce representative interpretation for many similar images. We formulate a co-clustering problem to simultaneously find the largest group of similar images and the common decision logic of a CNN. To make our interpretations more understandable, we visualize the common decision logic as a heat map to highlight important image regions, and also rank the similar images according to their semantic distances to the input image. Our extensive experiments demonstrate the superior representativeness and quality of the interpretations produced by our method.

## 8. Acknowledgement

Lingyang Chu’s research is supported in part by the start-up grant provided by the Department of Computing and Software of McMaster University. All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies.

## References

- [1] Python Notebook of Representative Interpretation. URL: <https://marketplace.huaweicloud.com/markets/aihub/notebook/detail/?id=8e92ea6c-2f4a-4cff-a89f-bf2905bb7ac0>.
- [2] Julius Adebayo, Justin Gilmer, Ian Goodfellow, and Been Kim. Local explanation methods for deep neural networks lack sensitivity to parameter values. *arXiv preprint arXiv:1810.03307*, 2018.
- [3] David Alvarez-Melis and Tommi S Jaakkola. On the robustness of interpretability methods. *ICML Workshop*, 2018.
- [4] Alexander Antonov. Food-5K image dataset, version 1. Retrieved from <https://www.kaggle.com/trolukovich/food5k-image-dataset>.
- [5] Sebastian Bach, Alexander Binder, Grégoire Montavon, Frederick Klauschen, Klaus-Robert Müller, and Wojciech Samek. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS One*, 10(7), 2015.
- [6] Jacob Bien and Robert Tibshirani. Prototype selection for interpretable classification. *The Annals of Applied Statistics*, 5(4):2403–2424, 2011.
- [7] Diogo V Carvalho, Eduardo M Pereira, and Jaime S Cardoso. Machine learning interpretability: A survey on methods and metrics. *Electronics*, 8(8):832, 2019.
- [8] Guillaume Charpiat, Nicolas Girard, Loris Felardos, and Yuliya Tarabalka. Input similarity from the neural network perspective. In *NeurIPS*, 2019.
- [9] Aditya Chatopadhyay, Anirban Sarkar, Prantik Howlader, and Vineeth N Balasubramanian. Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks. In *WACV*, pages 839–847, 2018.
- [10] Ashutosh Chaudhuri. Gender Classification Dataset, version 1. Retrieved from <https://www.kaggle.com/cashutosh/gender-classification-dataset>.
- [11] Chaofan Chen, Oscar Li, Chaofan Tao, Alina Jade Barnett, Jonathan Su, and Cynthia Rudin. This looks like that: deep learning for interpretable image recognition. In *NeurIPS*, 2019.
- [12] Lingyang Chu, Xia Hu, Juhua Hu, Lanjun Wang, and Jian Pei. Exact and consistent interpretation for piecewise linear neural networks: A closed form solution. In *ACM SIGKDD*, pages 1244–1253, 2018.
- [13] Victoria Crawford, Alan Kuhnle, and My Thai. Submodular cost submodular cover with an approximate oracle. In *ICML*, pages 1426–1435. PMLR, 2019.
- [14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.
- [15] Inderjit S Dhillon, Subramanyam Mallela, and Dharmendra S Modha. Information-theoretic co-clustering. In *ACM SIGKDD*, pages 89–98, 2003.
- [16] Ann-Kathrin Dombrowski, Maximilian Alber, Christopher J Anders, Marcel Ackermann, Klaus-Robert Müller, and Pan Kessel. Explanations can be manipulated and geometry is to blame. In *NeurIPS*, 2019.
- [17] Jeremy Elson, John R Douceur, Jon Howell, and Jared Saul. Asirra: a captcha that exploits interest-aligned manual image categorization. In *CCS*, 2007.
- [18] Ruth Fong and Andrea Vedaldi. Net2vec: Quantifying and explaining how concepts are encoded by filters in deep neural networks. In *CVPR*, pages 8730–8738, 2018.
- [19] Nicholas Frosst and Geoffrey Hinton. Distilling a neural network into a soft decision tree. *arXiv preprint arXiv:1711.09784*, 2017.
- [20] Amirata Ghorbani, Abubakar Abid, and James Zou. Interpretation of neural networks is fragile. In *AAAI*, volume 33 (1), pages 3681–3688, 2019.
- [21] Amirata Ghorbani, James Wexler, James Zou, and Been Kim. Towards automatic concept-based explanations. In *NeurIPS*, 2019.
- [22] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 315–323, 2011.
- [23] Ian Goodfellow, David Warde-Farley, Mehdi Mirza, Aaron Courville, and Yoshua Bengio. Maxout networks. In *ICML*, pages 1319–1327, 2013.
- [24] Yash Goyal, Ziyan Wu, Jan Ernst, Dhruv Batra, Devi Parikh, and Stefan Lee. Counterfactual visual explanations. In *ICML*, pages 2376–2384, 2019.
- [25] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.
- [26] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.
- [27] Mark Ibrahim, Melissa Louie, Ceena Modarres, and John Paisley. Global explanations of neural networks: Mapping the landscape of predictions. In *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, pages 279–287, 2019.
- [28] Rishabh Iyer and Jeff Bilmes. Submodular optimization with submodular cover and submodular knapsack constraints. In *NeurIPS*, 2013.
- [29] Daniel S Kermany, Kang Zhang, and Michael Goldbaum. Large dataset of labeled optical coherence tomography (oct) and chest x-ray images, version 3. In *Mendeley Data*, 2018.
- [30] Rajiv Khanna, Been Kim, Joydeep Ghosh, and Sanmi Koyejo. Interpreting black box predictions using fisher kernels. In *International Conference on Artificial Intelligence and Statistics*, pages 3382–3390, 2019.
- [31] Been Kim, Rajiv Khanna, and Oluwasanmi O Koyejo. Examples are not enough, learn to criticize! criticism for interpretability. In *NeurIPS*, 2016.
- [32] Been Kim, Martin Wattenberg, Justin Gilmer, Carrie Cai, James Wexler, Fernanda Viegas, et al. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, pages 2668–2677, 2018.
- [33] Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *ICML*, pages 1885–1894, 2017.
- [34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 2012.
- [35] Peter Cho-Ho Lam, Lingyang Chu, Maxim Torgonskiy, Jian Pei, Yong Zhang, and Lanjun Wang. Finding representative interpretations on convolutional neural networks. *arXiv preprint arXiv:2108.06384*, 2021. URL: <https://arxiv.org/abs/2108.06384>.
- [36] Alexander Levine, Sahil Singla, and Soheil Feizi. Certifiably robust interpretation in deep learning. *arXiv preprint arXiv:1905.12105*, 2019.
- [37] Stuart Lloyd. Least squares quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982.

- [38] Yang Lu, Wenbo Guo, Xinyu Xing, and William Stafford Noble. Robust saliency maps with decoy-enhanced saliency score. *arXiv preprint arXiv:2002.00526*, 2020.
- [39] Scott Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *NeurIPS*, 2017.
- [40] Christoph Molnar. *Interpretable machine learning*. 2019. <https://christophm.github.io/interpretable-ml-book/>.
- [41] Gr  goire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert M  ller. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222, 2017.
- [42] Guido Mont  far, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *NeurIPS*, pages 2924–2932, 2014.
- [43] Vinod Nair and Geoffrey E Hinton. Rectified linear units improve restricted boltzmann machines. In *ICML*, 2010.
- [44] Woo-Jeoung Nam, Shir Gur, Jaesik Choi, Lior Wolf, and Seong-Whan Lee. Relative attributing propagation: Interpreting the comparative contributions of individual units in deep neural networks. In *AAAI*, volume 34 (3), pages 2501–2508, 2020.
- [45] Daniel Omeiza, Skyler Speakman, Celia Cintas, and Komminist Weldermariam. Smooth grad-cam++: An enhanced inference level visualization technique for deep convolutional neural network models. In *Intelligent Systems Conference*, 2019.
- [46] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *ICML*, pages 2847–2854, 2017.
- [47] Harish Guruprasad Ramaswamy et al. Ablation-cam: Visual explanations for deep convolutional network via gradient-free localization. In *WACV*, pages 983–991, 2020.
- [48] Marco T Ribeiro, Sameer Singh, and Carlos Guestrin. "Why should I trust you?" Explaining the predictions of any classifier. In *ACM SIGKDD*, pages 1135–1144, 2016.
- [49] Marco T Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *AAAI*, volume 32 (1), 2018.
- [50] Ramprasaath R Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, and Dhruv Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, pages 618–626, 2017.
- [51] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. Learning important features through propagating activation differences. In *ICML*, pages 3145–3153, 2017.
- [52] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: Visualising image classification models and saliency maps. *arXiv preprint arXiv:1312.6034*, 2013.
- [53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2015.
- [54] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Vi  gas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- [55] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *ICLR*, 2015.
- [56] Suraj Srinivas and Francois Fleuret. Full-gradient representation for neural network visualization. In *NeurIPS*, 2019.
- [57] Akshayvarun Subramanya, Vipin Pillai, and Hamed Pirsiavash. Fooling network interpretation in image classification. In *ICCV*, pages 2020–2029, 2019.
- [58] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. In *ICML*, pages 3319–3328, 2017.
- [59] Sarah Tan, Rich Caruana, Giles Hooker, Paul Koch, and Albert Gordo. Learning global additive explanations for neural nets using model distillation. In *NeurIPS Workshop*, 2018.
- [60] Alvin Wan, Lisa Dunlap, Daniel Ho, Jihan Yin, Scott Lee, Henry Jin, Suzanne Petryk, Sarah Adel Bargal, and Joseph E Gonzalez. NBDT: neural-backed decision trees. *arXiv preprint arXiv:2004.00221*, 2020.
- [61] Haofan Wang, Zifan Wang, Mengnan Du, Fan Yang, Zijian Zhang, Sirui Ding, Piotr Mardziel, and Xia Hu. Score-cam: Score-weighted visual explanations for convolutional neural networks. In *CVPR Workshops*, pages 24–25, 2020.
- [62] Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.
- [63] Huijun Wu, Chen Wang, Richard Nock, Wei Wang, Jie Yin, Kai Lu, and Liming Zhu. Smint: Toward interpretable and robust model sharing for deep neural networks. *TWEB*, 14(3):1–28, 2020.
- [64] Huijun Wu, Chen Wang, Jie Yin, Kai Lu, and Liming Zhu. Sharing deep neural network models with interpretation. In *TheWebConf*, pages 177–186, 2018.
- [65] Chengliang Yang, Anand Rangarajan, and Sanjay Ranka. Global model interpretation via recursive partitioning. In *HPCC/SmartCity/DSS*, pages 1563–1570, 2018.
- [66] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *ECCV*, pages 818–833, 2014.
- [67] Matthew D Zeiler, Graham W Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *ICCV*, pages 2018–2025, 2011.
- [68] Quanshi Zhang, Ruiming Cao, Feng Shi, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnn knowledge via an explanatory graph. In *AAAI*, volume 32 (1), 2018.
- [69] Quanshi Zhang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. Growing interpretable part graphs on convnets via multi-shot learning. In *AAAI*, volume 31 (1), 2017.
- [70] Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu. Interpreting cnns via decision trees. In *CVPR*, pages 6261–6270, 2019.
- [71] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *CVPR*, pages 586–595, 2018.
- [72] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016.
- [73] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *ECCV*, pages 119–134, 2018.