Interpretable Basis Decomposition for Visual Explanation

Bolei Zhou[1]*, Yiyou Sun[2]*, David Bau[1]*, Antonio Torralba[1]

[1]MIT CSAIL [2]Harvard
{bzhou,davidbau,torralba}@csail.mit.edu, sunyiyou@seas.harvard.edu
* indicates equal contribution

Abstract. Explanations of the decisions made by a deep neural network are important for human end-users to be able to understand and diagnose the trustworthiness of the system. Current neural networks used for visual recognition are generally used as black boxes that do not provide any human interpretable justification for a prediction. In this work we propose a new framework called Interpretable Basis Decomposition for providing visual explanations for classification networks. By decomposing the neural activations of the input image into semantically interpretable components pre-trained from a large concept corpus, the proposed framework is able to disentangle the evidence encoded in the activation feature vector, and quantify the contribution of each piece of evidence to the final prediction. We apply our framework for providing explanations to several popular networks for visual recognition, and show it is able to explain the predictions given by the networks in a human-interpretable way. The human interpretability of the visual explanations provided by our framework and other recent explanation methods is evaluated through Amazon Mechanical Turk, showing that our framework generates more faithful and interpretable explanations[1].

1 Introduction

As deep networks continue to prove their capabilities on an expanding set of applications in visual recognition such as object classification [19], scene recognition [29], image captioning [24], and visual question answering [1], it is increasingly important not only for a network to make accurate predictions, but also to be able to explain why the network makes each prediction.

A good explanation of a deep network should play two roles: first, it should be a faithful representation of the operation of the network; and second, it should be simple and interpretable enough for a human to understand. There are two approaches for creating human-understandable explanations for the internals of a deep network. One is to identify the *evidence* that a network uses to make a specific decision by creating a heatmap that indicates which portions of an input are most salient to the decision [28, 2, 20]. Such heatmaps can be created using

[1] The code and data are available at https://github.com/CSAILVision/IBD

a variety of techniques and can be applied to identify the most salient parts of images and training sets. A second approach is to identify the *purpose* of the internal representations of a network by identifying the concepts that each part of the network detects [3, 27, 7]. Such concept dictionaries can be created by matching network units to a broad concept data set, by generating or sampling example inputs that reveal the sensitivity of a unit, or by training parts of the network to solve interpretable subproblems.

In this paper we describe a framework called *Interpretable Basis Decomposition (IBD)*, for bringing these two approaches together to generate explanations for visual recognition. The framework is able to decompose the evidence for a prediction for image classification into semantically interpretable components, each with an identified purpose, a heatmap, and a ranked contribution, as shown in Fig. 1. In addition to showing where a network looks, we show which concepts a network is responding to at each part of the input image.

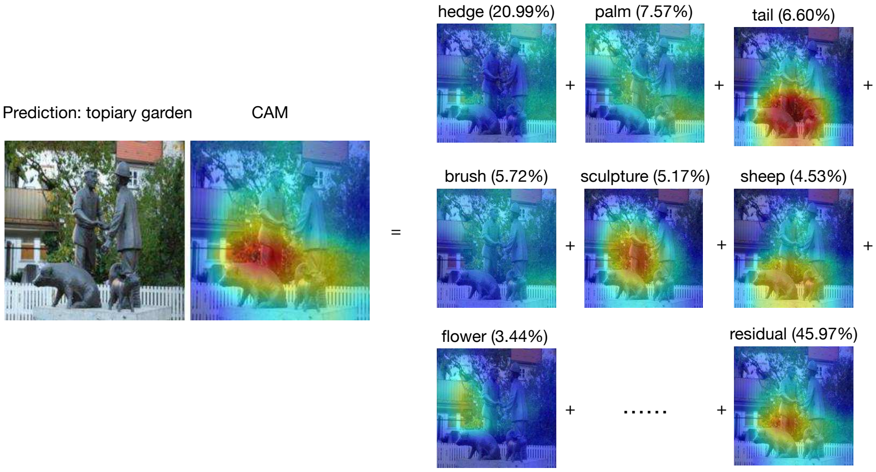


Fig. 1. Interpretable Basis Decomposition provides an explanation for a prediction by decomposing the decision into the components of interpretable basis. Top contributing components are shown with a label, contribution, and heatmap for each term.

Our framework is based on the insight that good explanations depend on context. For example, the concepts to explain what makes up a 'living room' are different from the concepts to explain an 'airport'. A overstuffed pillow is not an airliner, nor vice-versa. We formalize the idea of a salient set of concepts as a choice of a interpretable basis in the feature space, and describe how to construct a context-specific concept basis as the solution to a least-squares problem.

Each explanation we describe is both a visualization and a vector decomposition of a layer's internal state into interpretable components. As a vector decomposition, each explanation is faithful to the network, quantifying the contribution of each component and also quantifying any uninterpreted residual.

The framework also provides explanations that are simple enough for a person to understand. We conduct human evaluations to show that the explanations give people accurate insights about the accuracy of a network.

We summarize our contributions as follows: 1) A new framework called Interpretable Basis Decomposition to provide semantic explanations with labels and heatmaps for neural decision making. 2) Application of the proposed framework on a wide range of network architectures, showing its general applicability. 3) Human evaluations to demonstrate that the explanations are understandable to people, outperforming previous heatmap and unit-based explanation methods.

1.1 Related Work

Visualizing neural networks. A number of techniques have been developed to visualize the internal representations of convolutional neural networks. The behavior of a CNN can be visualized by sampling image patches that maximize activation of hidden units [25], and by backpropagation to identify or generate salient image features [16, 21]. An image generation network can be trained to invert the deep features by synthesizing the input images [5]. The semantics of visualized units can be annotated manually [27] or automatically [3] by measuring alignment between unit activations and a predefined dictionary of concepts.

Explaining neural network decisions. Explanations of individual network decisions have been explored by generating informative heatmaps such as CAM [28] and grad-CAM [20], or through back-propagation conditioned on the final prediction [21] and layer-wise relevance propagation [2]. The attribution of each channel to the final prediction has been studied [18]. Captioning methods have been used to generate sentence explanations for a fine-grained classification task [9]. The limitation of the heatmap-based explanation methods is that the generated heatmaps are qualitative and not informative enough to tell which concepts have been detected, while the sentence-based explanation methods require an ad-hoc corpus of sentence description in order to train the captioning models. Our work is built upon previous work interpreting the semantics of units [3] and on heatmaps conditioned on the final prediction [20, 28]. Rather than using the semantics of activated units to build explanations as in [26], we learn a set of interpretable vectors in the feature space and decompose the representation in terms of these vectors. We will show that the proposed method is able to generate faithful explanations which are more informative than the previous heatmap-based and unit-activation methods.

Component analysis. Understanding an input signal by decomposing it into components is an old idea. Principal Component Analysis [12] and Independent Component Analysis [11] have been widely used to disentangle a low-dimensional basis from high-dimensional data. Other decomposition methods such as Bilinear models [23] and Isomap [22] are also used to discover meaningful subspaces and structure in the data. Our work is inspired by previous work on component decomposition. Rather than learning the components unsupervised, we learn the set of components from a fully annotated dataset so that we have a ground-truth label for each component. After projecting, the labeled components provide interpretations, forming human-understandable explanations.

Concurrent work [14] proposes examining the behavior of representations in the direction of a set of semantic concept vectors learned from a pre-defined dataset. Those Concept Activation Vectors play a similar role as our Interpretable Basis Vectors, but while that work focuses on using a single feature at a time for retrieval and scoring of samples, our work uses basis sets of vectors to create explanations and decomposed heatmaps for decisions.

2 Framework for Interpretable Basis Decomposition

The goal of Interpretable Basis Decomposition is to decode and explain every bit of information from the activation feature vector in a neural network's penultimate layer. Previous work has shown that it is possible to roughly invert a feature layer to recover an approximation to the original input image using a trained feature inversion network [5]. Instead of recovering the input image, our goal is to decode the meaningful nameable components from the feature vector so that we can build an explanation of the final prediction.

We will describe how we decompose feature vectors in three steps. We begin by describing a way to decompose an output class k into a set of interpretable components c. In our decomposition, both the class and the concepts are represented as vectors w_k and q_c that correspond to linear classifiers in the feature space, and the decomposition is expressed as an optimal choice of basis for w_k. The result of this step is a set of elementary concepts relevant to each class.

Next, we describe how to derive vectors q_c corresponding to a broad dictionary of elementary interpretable concepts c. Each q_c is learned by training a linear segmentation model to locate the concept within the feature space.

Finally, we describe how to create explanations of instance decisions. This is done by projecting the feature vector into the learned interpretable basis and measuring the contribution of each interpretable component. An explanation consists of a list of concepts that contribute most to the final score, together with a heatmap for each concept that shows where the contributions arise for the final prediction. The framework is illustrated in Fig. 2.

2.1 Defining an Interpretable Basis

Explaining a layer can be done by choosing an interpretable basis for the layer's input representation. To see why, set $f(x) \in \mathbb{R}^K$ as a deep net with K output dimensions, considered without the final softmax. We are interested in explaining properties of x which determine the score $f_k(x)$ for a particular class $k \leq K$: for example, we may wish to know if a concept c such as crowds of people tends to cause the input to be classified as an output class k such as airports.

We can express our question in terms of an intermediate representation. Write $f(x) = h(g(x))$ where $h(a)$ is the top of the network and $a = g(x) \in \mathbb{R}^D$ is a point in the representation space of the layer of interest. Then to investigate the

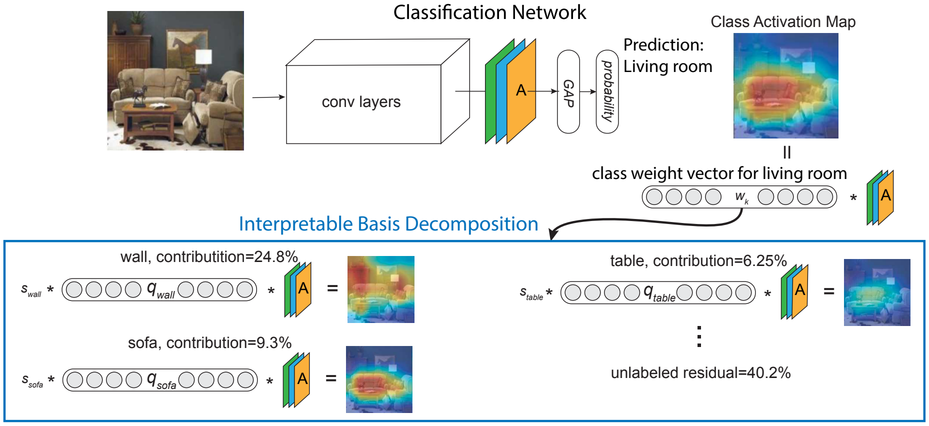


Fig. 2. Illustration of Interpretable Basis Decomposition. The class weight vector w_k is decomposed to a set of interpretable basis vectors $\sum s_{c_i} q_{c_i}$, each corresponding to a labeled concept c_i as well as a projection $q_{c_i}^T A$ that reveals a heatmap of the activations. An explanation of the prediction k consists of the concept labels c_i and the corresponding heatmaps for the most significant terms in the decomposition of $w_k^T a$. For this particular example, wall, sofa, table (and some others are not shown) are labels of the top contributing basis elements that make up the prediction of living room.

properties of x that determine $f_k(x)$, we can ask about the properties of the intermediate representation $a = g(x)$ that determine $h_k(a)$.

Let us focus on the simple case where $a = g(x)$ is the output of the second-to-last layer and $h(a)$ is a simple linear operation done by the last layer. Then h_k is a linear function that scores a according to the angle between a and $w_k \in R^D$:

$$h(a) \equiv W^{(h)} a + b^{(h)} \tag{1}$$

$$h_k(a) = w_k^T a + b_k \tag{2}$$

Not all directions in the representation space R^D are equally interpretable. Suppose we have a set of directions $q_{c_i} \in \mathbb{R}^D$ that each correspond to elementary interpretable concepts c_i that are relevant to class k but easier to understand than k itself. Then we can explain w_k by decomposing it into a weighted sum of interpretable components q_{c_i} as follows.

$$w_k \approx s_{c_1} q_{c_1} + \cdots + s_{c_n} q_{c_n} \tag{3}$$

Unless w_k lies exactly in the space spanned by the $\{q_{c_i}\}$, there will be some residual error in the decomposition. Gathering the q_{c_i} into columns of a matrix C, we can recognize that minimizing this error is a familiar least-squares problem:

$$\begin{aligned} \text{Find } s_{c_i} \text{ to minimize } ||r|| \text{ where } w_k &= s_{c_1} q_{c_1} + \cdots + s_{c_n} q_{c_n} + r && (4)\\ &= Cs + r && (5) \end{aligned}$$

The optimal s is given by $s = C^+ w_k$ where C^+ is the pseudoinverse of C.

When interpreting the decomposition of w_k, negations of concepts are not as understandable as positive concepts, so we seek decompositions for which each coefficient $s_{c_i} > 0$ is positive. Furthermore, we seek decompositions with a small number of concepts.

We build the basis q_{c_i} in a greedy fashion, as follows. Suppose we have already chosen a set of columns $C = [q_{c_1}|\cdots|q_{c_n}]$, and the residual error is in (4) is $\epsilon = ||w_k - Cs||$. Then we can reduce the residual by adding an $(n+1)$th concept to reduce error. The best such concept is the one that results in the minimum residual while keeping the coefficients positive:

$$\underset{c \in \mathcal{C}}{\operatorname{argmin}} \min_{s, s_i > 0} ||w_k - [C|q_c]s|| \tag{6}$$

where $[C|q_c]$ indicates the matrix that adds the vector q_c for the candidate concept c to the columns of C.

2.2 Learning the Interpretable Basis from Annotations

For explaining an image classification task, we build the universe of candidate concepts $\mathcal{C}$ using the Broden dataset [3]. Broden includes pixel-level segmentations for a broad range of both high-level visual concepts such as objects and parts, as well as low-level concepts such as colors and materials. For each candidate concept c in Broden, we compute an embedding $q_c \in \mathcal{C} \subset \mathbb{R}^D$ as follows.

Since Broden provides pixel-level segmentations of every concept, we train a logistic binary classifier $h_c(a) = \text{sigmoid}(w_c^T a + b_c)$ to detect the presence of concept c. Training is done on a mix of images balancing c present or absent at the center, and hard negative mining is used to select informative negative examples during the training progress; the training procedure is detailed in Sec. 3.1. The learned w_c captures the features relevant to class c, but it is scaled in a way that is sensitive to the training conditions for c. To eliminate this arbitrary scaling, we standardize q_c as the normalized vector $q_c = (w_c - \overline{w}_c)/||w_c - \overline{w}_c||$.

2.3 Explaining a Prediction via Interpretable Basis Decomposition

The decomposition of any class weight vector w_k into interpretable components $C_k \subset \mathcal{C} \subset \mathbb{R}^D$ allows us to decompose the scoring of activations a into components of C_k in exactly the same way as we decompose w_k itself. This decomposition will provide an interpretable explanation of the classification.

Furthermore, if we include define a larger basis $C_k^* \supset C_k$ that adds the residual vector $r = w_k - C_k s$, we can say something stronger: projecting a into the basis of C_k^* captures the entire linear relationship described by the network's final layer score $h_k(a)$ for class k.

$$\begin{aligned} h_k(a) &= w_k^T a + b_k \qquad (7) \\ &= (C_k^* s)^T a + b_k \qquad (8) \\ &= s_1 q_{c_1}^T a + \cdots + \underbrace{s_i q_{c_i}^T a}_{\text{contribution of concept } c_i} + \cdots + s_n q_{c_n}^T a + \underbrace{r^T a}_{\text{residual contribution}} + b_k \qquad (9) \end{aligned}$$

Thus we can decompose the score into contributions from each concept, and we can rank each concept according to its contribution. When the activation $a = \text{pool}(A)$ is derived by global average pooling of a convolutional layer A, we can commute the dot product inside the pooling operation to obtain a picture that localizes the contribution of concept c_i.

$$s_i q_{c_i}^T a = s_i q_{c_i}^T \,\text{pool}(A) \tag{10}$$

$$= \text{pool}(s_i \underbrace{q_{c_i}^T A}_{\text{heatmap for concept } c_i}) \tag{11}$$

The explanation we seek consists of the list of concepts c_i with the largest contributions to $h_k(a)$, along with the heatmaps $q_{c_i}^T A$ for each concept. The IBD heatmaps $q_{c_i}^T A$ are similar to the CAM heatmap $w_k^T A$ and can be used to reconstruct the CAM heatmap if they are all summed. However, instead of summarizing the locations contributing to a classification all at once, the interpretable basis decomposition separates the explanation into component heatmaps, each corresponding to a single concept that contributes to the decision.

Decomposing gradients for GradCAM: Grad-CAM is an extension of CAM [28] to generate heatmap for networks with more than one final nonconvolutional layers. Starting with the final convolutional featuremap $a = g(x)$, the Grad-CAM heatmap is formed by multiplying this activation by the pooled gradient of the higher layers $h(a)$ with respect class k.

$$w_k(a) = \frac{1}{Z} \sum_i \sum_j \nabla_a h_k(a) \tag{12}$$

Here the vector $w_k(a)$ plays the same role as the constant vector w_k in CAM: to create an interpretable basis decomposition, $w_k(a)$ can be decomposed as described in Eqs. 4-6 to create a componentwise decomposition of the Grad-CAM heatmap. Since $w_k(a)$ is a function of the input, each input will have its own interpretable basis.

3 Experiments

In this section, we describe how we learn an interpretable basis from an annotated dataset. Then we will show that the concepts of the interpretable basis that are associated with each prediction class of the networks sheds lights on the abstractions learned by each network. After that we use the interpretable basis decomposition to build explanations for the predictions given by the popular network architectures: AlexNet [15], VGG [13], ResNet (18 and 50 layers) [8], each trained scratch on ImageNet [4] and Places365 [29]. Finally we evaluate the fidelity of the explanations given by our method through Amazon Mechanical Turk and compare with other visual explanation generation methods.

3.1 Interpretable Basis Learned from Broden

We derive an interpretable basis from the fully annotated image dataset Broden [26]. Because our focus is to explain high-level features of the neural networks in terms of human interpretable concepts, we take a subset of the Broden dataset consisting of object and part concepts. The annotations of the objects and parts in Broden dataset originally come from the datasets ADE20K [30], Pascal Context [17], and Pascal Parts [6]. We filter out the concepts with fewer than 10 image samples, resulting to 660 concepts from 30K images used for training and testing.

For each concept in the Broden dataset, we learn a logistic binary classifier. The input of the classifier is a feature vector $a^{(i,j)} \in \mathbb{R}^D$ in activation $A \in \mathbb{R}^{D\times H\times W}$, and the output is the prediction of the probability of the concept appearing at $(i, j) \in (\text{range}(H), \text{range}(W))$. Our ground truth labels for the segmentations are obtained by downsampling the original concept masks to $H \times W$ size using nearest neighbor. Note that Broden provides multi-labeled segmentations, and there are often several concepts present in each downsampled pixel. Therefore it is appropriate for each concept classifier to be trained independent of each other. Because the number of positive samples and the number of negative samples for some concepts are highly unbalanced, we resample the training set to keep the ratio of positive and negative examples of each class fixed at 1 : 20 and use five rounds of hard negative mining.

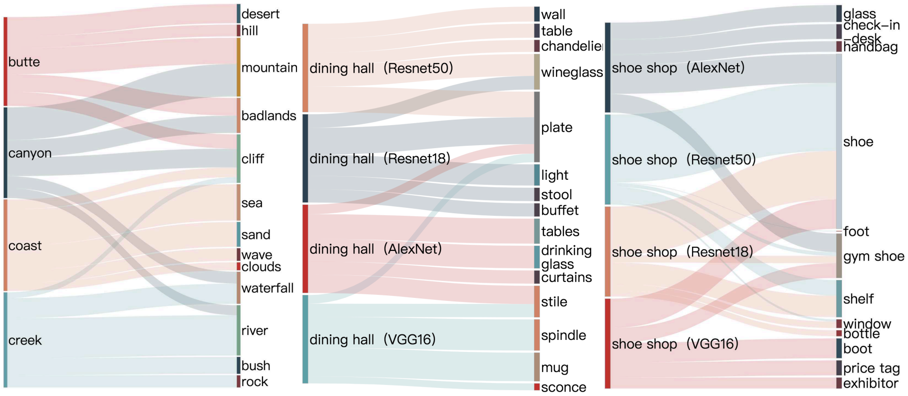
We evaluate the accuracy of the deep features learned from several networks as shown in Table 1. All models are evaluated with mAP on a fixed validation set of Broden dataset.

model	AlexNet	VGG16	Resnet18	Resnet50
mAP	0.625	0.691	0.784	0.804

Table 1. The mAP of the learned concept classifiers for the object and part concepts in the Broden dataset. The features used are the activations at the final convolutional layer of the network trained from scratch on Places365.

3.2 Explaining Classification Decision Boundaries

Interpretable Basis Decomposition assigns a basis of interpretable concepts for each output class. This basis can be seen as a set of *compositional rules* between the output classes and the elementary concepts in the Broden datasets. Different network learns a different set of such semantic rules for a prediction, thus by directly examining the interpretable basis decomposition of a network we can gain insight about the decision boundaries learned by each network for each class.



(a) Comparing different concepts that Resnet18 utilizes to make different predictions.

(b) Comparing different concepts that different networks (Resnet18, Resnet50, AlexNet-CAM, VGG16-CAM) utilize to make predictions (places365 label: dining hall, shoe shop).

Fig. 3. Visualizing how different networks compose the final prediction classes using the Broden concepts. The left labels in each graph show the classes of Places365 and the right labels are the concepts of Broden. The thickness of each link between a class and a concept indicates the magnitude of the coefficient s_{c_i}.

Specifically, our method decomposes each weight vector w_k of class k in the last layer[2] as the sum $w_k = s_{c_1} q_{c_1} + \cdots + s_{c_n} q_{c_n} + r$, where q_{c_i} represents the embedding vector for concept c_i and s_{c_i} is the coefficient indicating its contribution to the overall class k. This decomposition indicates a relationship between the output class k and the concept c_i described by the coefficient s_{c_i}. In Fig. 3, we visualize a subset of Places365 classes k and how they are decomposed into Broden concepts c_i by different networks. The left column of the figure is the list of Places365 classes to be decomposed. The right column shows the related concepts from the Broden dataset. The thicknesses of the arcs between classes and concepts are drawn to show the magnitude of the coefficients s_{c_i}. The larger s_{c_i}, the more important concept c_i is to the prediction of class k.

In Fig. 3.(a), it can be seen how a single network composes concepts to constitute a variety of different prediction classes. Note that all the classes shown in (a) share the same concept "cliff" but differ in the importance given to this concept, which can be seen as different s_{c_i}. Fig. 3.(b), shows the different compositional rules that different networks use to make the same prediction for a class. For example, in the prediction class "shoe shop", all networks agree that "shoe" is a key element that contributes to this prediction, while they disagree on other elements. VGG16 treats "boot" and "price tag" as important indicators of a "shoe shop," while and AlexNet decomposes "shoe shop" into different concepts such as "glass" and "check-in-desk."

[2] For this experiment, we replace the fc layers in AlexNet and VGG16 with a GAP layer and retrain them, similar to [28]

3.3 Explaining Image Predictions

Given the interpretable basis decomposition $w_k = s_{c_1} q_{c_1} + \cdots + s_{c_n} q_{c_n} + r$, the instance prediction result $w_k^T a$ is decomposed as $w_k^T a = s_{c_1} q_{c_1}^T a + \cdots + s_{c_n} q_{c_n}^T a + r^T a$ where each term $s_{c_i} q_{c_i}^T a$ can be regarded as the contribution of concept i to the final prediction. We rank the contribution scores and use the concept labels of the top contributed basis as an explanation for the prediction. Each term also corresponds to a contribution to the CAM or Grad-CAM salience heatmap.

Fig. 4 shows qualitative results of visual explanations done by our method. For each sample, we show the input image, its prediction given by the network, the heatmaps generated by CAM [28] for Resnet18 and Resnet18, and the heatmaps generated by Grad-CAM heatmap [20] for AlexNet and VGG166, and the top 3 contributing interpretable basis components with their labels and numerical contribution.

In Fig. 4(a), we select three examples from Places365 in which VGG16 and ResNet18 make the same correct predictions. In two of the examples, the explanations provide evidence that VGG16 may be *right for the wrong reasons* in some cases: it matches the *airplane* concept to contribute to the *crosswalk* prediction, and it matches the sofa concept to contribute to its *market* prediction. In contrast, ResNet18 appears to be sensitive to more relevant concepts.

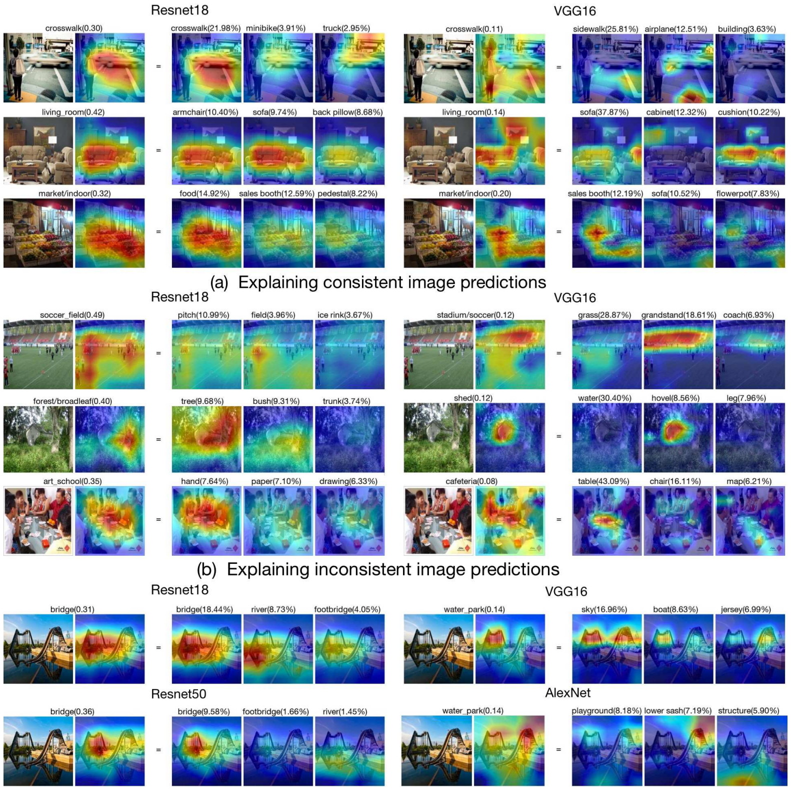
In Fig. 4(b), we show how our method can provide insight on an inconsistent prediction. ResNet18 classifies the image in last row as an *art school* because it sees features described as *hand* and *paper* and *drawing*, while VGG16 classifies the image as a *cafeteria* image because VGG16 it is sensitive to *table* and *chair* and *map* features. Both networks are incorrect because the table is covered with playing cards, not drawings or maps, and the correct label is *recreation room.*

In Fig. 4(c), we show the variations generated by different models for the same sample.

3.4 Human Evaluation of the Visual Explanations

To measure whether explanations provided by our method are reasonable and convincing to humans, we ask AMT raters to compare the quality of two different explanations for a prediction. We create explanations of decisions made by four different models (Resnet50, Resnet18, VGG16, and AlexNet, trained on Places365) using different explanation methods (Interpretable Basis Decomposition, Network Dissection, CAM and Grad-CAM).

The evaluation interface is shown in Fig. 5. In each comparison task, raters are shown two scene classification predictions with identical outcomes but with different explanations. One explanation is identified as Robot A and the other as Robot B, and raters are asked to decide which robot is more reasonable on a five-point Likert scale. Written comments about the difference are also collected. In the interface, heatmaps are represented as simple masks that highlight the top 20% of pixels in the heatmap; explanations are limited to four heatmaps; and each heatmap can be labeled with a named concept.

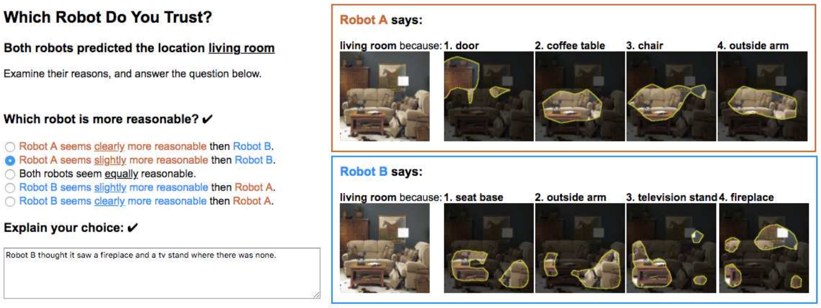


(a) Explaining consistent image predictions

(b) Explaining inconsistent image predictions

(c) Showing variations for different models

Fig. 4. Explaining specific predictions. The first image pair in each group contains original image (left) and single heatmap (right), with the predicted label and normalized prediction score in parentheses. Single heatmaps are CAM for ResNet and Grad-CAM for Alexnet and VGG. This is followed by three heatmaps corresponding to the three most significant terms in the interpretable basis decomposition for the prediction. The percentage contribution of each component to the score is shown. (a) Examples where two networks make the same prediction. (b) Explanations where two networks make different predictions. (c) Comparisons of different architectures.



Which Robot Do You Trust?

Both robots predicted the location living room

Examine their reasons, and answer the question below.

Which robot is more reasonable? ✔

○ Robot A seems clearly more reasonable then Robot B.
● Robot A seems slightly more reasonable then Robot B.
○ Both robots seem equally reasonable.
○ Robot B seems slightly more reasonable then Robot A.
○ Robot B seems clearly more reasonable then Robot A.

Explain your choice: ✔

Robot B thought it saw a fireplace and a tv stand where there was none.

Robot A says:

living room because: 1. door 2. coffee table 3. chair 4. outside arm

Robot B says:

living room because: 1. seat base 2. outside arm 3. television stand 4. fireplace

Fig. 5. Interface for human evaluations. Two different explanations of the same prediction are presented, and human raters are asked to evaluate which is more reasonable.

Baseline CAM, Grad-CAM, and Network Dissection explanations We compare our method to several simple baseline explanations. The first baselines are CAM [28] and Grad-CAM [20], which consist of a single salience heatmap for the image, showing the image regions that most contributed to the classification. Using the notation of Section 2.1, the CAM/Grad-CAM heatmap is given by weighting the pixels of the penultimate feature layer A according to the classification vector w_k, or to the pooled gradient $w_k(A)$:

$$\text{CAM}_k(A) \equiv w_k^T A \qquad\qquad \text{Grad-CAM}_k(A) \equiv w_k(A)^T A \tag{13}$$

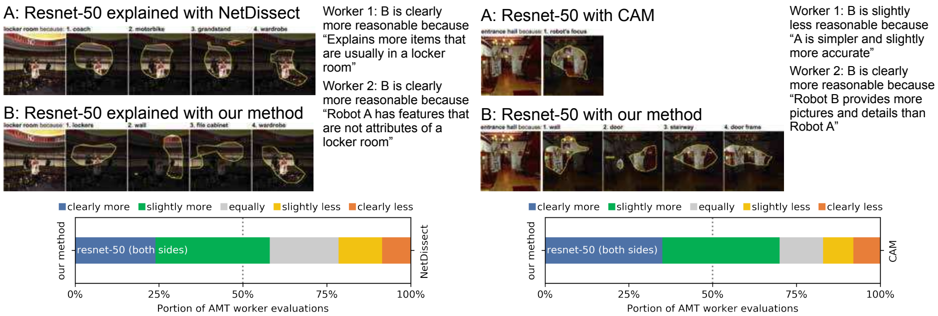
The second baseline is a simple unit-wise decomposition of the heatmap as labeled by Network Dissection. In this baseline method, every heatmap corresponds to a single channel of the featuremap A that has an interpretation as given by Network Dissection [26]. This baseline explanation ranks channels according to the components i that contribute most to $w_k^T a = \sum_i w_{ki} a_i$. Using the notation of Section 2.1, this corresponds to choosing a fixed basis C where each concept vector is the unit vector in the ith dimension $q_{c_i} = e_i$, labeled according to Network Dissection. Heatmaps are given by:

$$\text{NetDissect}_{k,i}(A) \equiv e_i^T A, \qquad \text{ranked by largest } w_{ki} a_i \tag{14}$$

CAM and the Network Dissection explanations can be thought of as extremal cases of Interpretable Basis Decomposition: CAM chooses no change in basis and visualizes the contributions from the activations directly; while Network Dissection always chooses the same unit-wise basis.

Comparing Explanation Methods Directly. In the first experiment, we compare explanations generated by our method head-to-head with explanations generated by Network Dissection [26] and CAM [28] and Grad-CAM [20]. In this experiment, both Robot A and Robot B are the same model making the same decision, but the decision is explained in two different ways. For each network and pair of explanation methods, 200 evaluations of pairs of explanations are done by at least 40 different AMT workers. Fig. 8 summarizes the six pairwise comparisons. Across all tested network architectures, raters find our method more reasonable, on average, than then explanations created by CAM, Grad-CAM, and Network Dissection.

Representative samples with comments from the evaluation are shown in Fig. 6. Raters have paid attention to the quality and relevance of the explanatory regions as well as the quality and relevance of the named concepts. When comparing the single-image explanations of CAM and Grad-CAM with our multiple-image explanations, some raters express a preference for shorter explanations and others prefer the longer ones. Since is generally assumed that humans have a strong bias towards simpler explanations ([10]), it is interesting to find that, on average, human raters prefer our longer explanations. The second experiment, described next, controls for this bias by evaluating only comparisons where raters see the same type of explanation for both Robot A and Robot B.



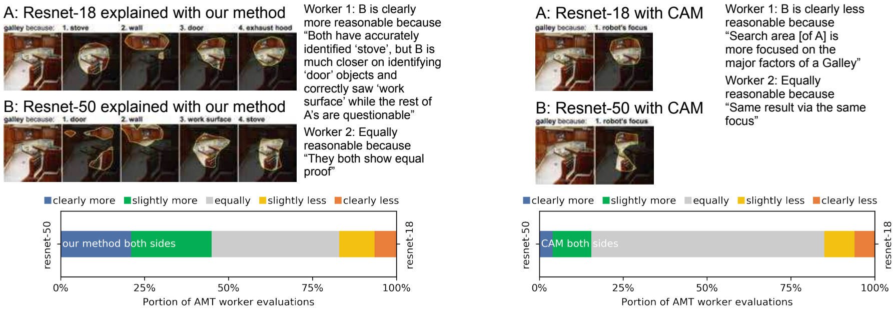
Worker 1: B is clearly more reasonable because "Explains more items that are usually in a locker room"

Worker 2: B is clearly more reasonable because "Robot A has features that are not attributes of a locker room"

Worker 1: B is slightly less reasonable because "A is simpler and slightly more accurate"

Worker 2: B is clearly more reasonable because "Robot B provides more pictures and details than Robot A"

Fig. 6. Representative examples of human feedback in head-to-head comparisons of methods. For each image, one comparison is done. At left, explanations using Net Dissection and our method are compared on same ResNet50 decision. At right, explanations using CAM and our method are compared on another ResNet50 decision.



Worker 1: B is clearly more reasonable because "Both have accurately identified 'stove', but B is much closer on identifying 'door' objects and correctly saw 'work surface' while the rest of A's are questionable"

Worker 2: Equally reasonable because "They both show equal proof"

Worker 1: B is clearly less reasonable because "Search area [of A] is more focused on the major factors of a Galley"

Worker 2: Equally reasonable because "Same result via the same focus"

Fig. 7. Representative examples of human feedback in trust comparison. For each image, two independent comparisons are done. At left, a decision of ResNet50 and ResNet18 are compared using our method of explanation. At right, the same pair of decisions is compared using a CAM explanation.

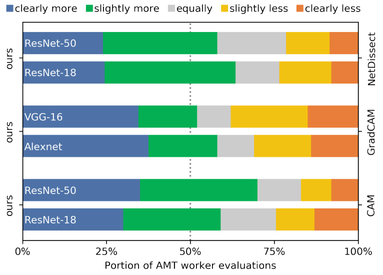


Fig. 8. Comparing different explanation methods side-by-side. Each bar keeps the network the same and compares our explanations to another method. Blue and green indicate ratings of explanations of our method that are clearly or slightly more reasonable, and yellow and orange indicate ratings for where our method is slightly or clearly less reasonable than a different explanation method.

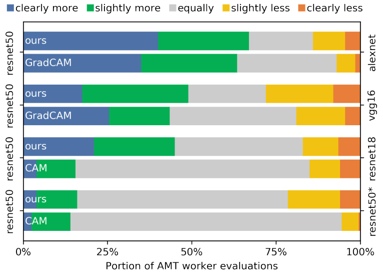


Fig. 9. Comparing ability of users to evaluate trust using different explanation methods. Each bar keeps the explanation method the same and compares ResNet50 to another model. Blue and green indicate evaluations where ResNet50 explanations are rated clearly and slightly more reasonable, and yellow and orange indicate explanations where ResNet50 is slightly and clearly less reasonable.

Comparing Evaluations of Model Trust. The second experiment evaluates the ability of users to evaluate trustworthiness of a model based on only a single pair of explanations. The ordinary way to evaluate the generalization ability of a model is to test its accuracy on a holdout set of many inputs. This experiment tests whether a human can compare two models based on a single comparison of explanations of identical decisions made by the models on one input image.

In this experiment, as shown in Fig. 7, explanations for both Robot A and Robot B are created using the same explanation method (either our method or CAM), but the underlying networks are different. One is always Resnet50, and the other is either AlexNet, VGG16, Resnet18, or a mirrored version of Resnet50 (resnet50*) where all the convolutions are horizontally flipped. Only explanations where both compared networks make the same decision are evaluated: as can be seen in the feedback, our explanation method allow raters to discern a quality difference between deeper and shallower methods, while the single-image CAM heatmap makes the two networks seem less different.

Fig. 9 summarizes results across several different network architectures. With our explanation method, raters can identify that Resnet50 is more trustworthy than Alexnet, VGG16 and Resnet18; the performance is similar to or marginally better than Grad-CAM, and it outperforms CAM. Comparisons of two Resnet50 with each other are evaluated as mostly equivalent, as expected, under both methods. It is interesting to see that it is possible to discern the difference between shallower and deeper networks despite a very narrow difference in validation accuracy between the models, even after observing only a single case on which two different models perform identical predictions.

4 Discussion and Conclusion

The method has several limitations: first, it can only identify concepts in the dictionary used. This limitation can be quantified by examining the magnitude of the residual. For scene classification on ResNet50, explanations derived from our dataset of 660 concepts have a mean residual of 65.9%, suggesting most of the behavior of the network remains orthogonal to the explained concepts. A second limitation is that the residual is not guaranteed to approach zero even if the concept dictionary were vast: decisions may depend on visual features that do not correspond to *any* natural human concepts. New methods may be needed to to characterize what those features might be.

We have proposed a new framework called Interpretable Basis Decomposition for providing visual explanations for the classification networks. The framework is able to disentangle the evidence encoded in the activation feature vector and quantify the contribution of each part of the evidence to the final prediction. Through crowdsourced evaluation, we have verified that the explanations are reasonable and helpful for evaluating model quality, showing improvements over previous visual explanation methods.

Acknowledgement: The work was partially funded by DARPA XAI program FA8750-18-C0004, the National Science Foundation under Grants No. 1524817, and the MIT-IBM Watson AI Lab. B.Z is supported by a Facebook Fellowship.

References

1. Antol, S., Agrawal, A., Lu, J., Mitchell, M., Batra, D., Lawrence Zitnick, C., Parikh, D.: Vqa: Visual question answering. In: Proc. CVPR (2015)
2. 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), e0130140 (2015)
3. Bau, D., Zhou, B., Khosla, A., Oliva, A., Torralba, A.: Network dissection: Quantifying interpretability of deep visual representations. In: Proc. CVPR (2017)
4. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: ImageNet: A Large-Scale Hierarchical Image Database. In: CVPR09 (2009)
5. Dosovitskiy, A., Brox, T.: Generating images with perceptual similarity metrics based on deep networks. In: Advances in Neural Information Processing Systems. pp. 658–666 (2016)
6. Everingham, M., Eslami, S.M.A., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision **111**(1), 98–136 (Jan 2015)
7. Gonzalez-Garcia, A., Modolo, D., Ferrari, V.: Do semantic parts emerge in convolutional neural networks? International Journal of Computer Vision pp. 1–19 (2017)
8. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. arXiv preprint arXiv:1512.03385 (2015)
9. Hendricks, L.A., Akata, Z., Rohrbach, M., Donahue, J., Schiele, B., Darrell, T.: Generating visual explanations. In: European Conference on (ECCV) (2016)
10. Herman, B.: The promise and peril of human evaluation for model interpretability. arXiv preprint arXiv:1711.07414 (2017)
11. Hyvärinen, A., Oja, E.: Independent component analysis: algorithms and applications. Neural networks **13**(4-5), 411–430 (2000)
12. Jolliffe, I.T.: Principal component analysis and factor analysis. In: Principal component analysis, pp. 115–128. Springer (1986)
13. K. Simonyan, A.Z.: Very deep convolutional networks for large-scale image recognition (2014)
14. Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., et al.: Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In: International Conference on Machine Learning (2018)
15. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems 25, pp. 1097–1105. Curran Associates, Inc. (2012), http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf
16. Mahendran, A., Vedaldi, A.: Understanding deep image representations by inverting them. Proc. CVPR (2015)
17. Mottaghi, R., Chen, X., Liu, X., Cho, N.G., Lee, S.W., Fidler, S., Urtasun, R., Yuille, A.: The role of context for object detection and semantic segmentation in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2014)
18. Olah, C., Satyanarayan, A., Johnson, I., Carter, S., Schubert, L., Ye, K., Mordvintsev, A.: The building blocks of interpretability. Distill (2018). https://doi.org/10.23915/distill.00010, https://distill.pub/2018/building-blocks

19. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al.: Imagenet large scale visual recognition challenge. Int'l Journal of Computer Vision (2015)
20. Selvaraju, R.R., Das, A., Vedantam, R., Cogswell, M., Parikh, D., Batra, D.: Grad-cam: Why did you say that? arXiv preprint arXiv:1611.07450 (2016)
21. Simonyan, K., Vedaldi, A., Zisserman, A.: Deep inside convolutional networks: Visualising image classification models and saliency maps. International Conference on Learning Representations Workshop (2014)
22. Tenenbaum, J.B., De Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. science **290**(5500), 2319–2323 (2000)
23. Tenenbaum, J.B., Freeman, W.T.: Separating style and content. In: Advances in neural information processing systems. pp. 662–668 (1997)
24. Vinyals, O., Toshev, A., Bengio, S., Erhan, D.: Show and tell: A neural image caption generator. In: Proc. CVPR. IEEE (2015)
25. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. Proc. ECCV (2014)
26. Zhou, B., Bau, D., Oliva, A., Torralba, A.: Interpreting deep visual representations via network dissection. In: IEEE Trans. on Pattern Analysis and Machine Intelligence (2018)
27. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Object detectors emerge in deep scene cnns. International Conference on Learning Representations (2015)
28. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., Torralba, A.: Learning deep features for discriminative localization. In: Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on. pp. 2921–2929. IEEE (2016)
29. Zhou, B., Lapedriza, A., Xiao, J., Torralba, A., Oliva, A.: Learning deep features for scene recognition using places database. In Advances in Neural Information Processing Systems (2014)
30. Zhou, B., Zhao, H., Puig, X., Fidler, S., Barriuso, A., Torralba, A.: Scene parsing through ade20k dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (2017)