

Semantics for Global and Local Interpretation of Deep Neural Networks

Jindong Gu

The University of Munich
Siemens AG, Corporate Technology
jindong.gu@siemens.com

Volker Tresp

The University of Munich
Siemens AG, Corporate Technology
volker.tresp@siemens.com

Abstract

Deep neural networks (DNNs) with high expressiveness have achieved state-of-the-art performance in many tasks. However, their distributed feature representations are difficult to interpret semantically. In this work, human-interpretable semantic concepts are associated with vectors in feature space. The association process is mathematically formulated as an optimization problem. The semantic vectors obtained from the optimal solution are applied to interpret deep neural networks globally and locally. The global interpretations are useful to understand the knowledge learned by DNNs. The interpretation of local behaviors can help to understand individual decisions made by DNNs better. The empirical experiments demonstrate how to use identified semantics to interpret the existing DNNs.

1 Introduction

Deep neural networks enable many recent advances in artificial intelligence. Due to the lack of their interpretability, it is difficult to explain their decisions to end users. The neural networks operate on features, which do not correspond to human-interpretable concepts. The neural networks work in a feature space E_f spanned by basis vectors e_f corresponding to neural activations. Humans work in a different vector space E_h . The interpretation of the features of neural networks is to map E_f to E_h .

Some existing works associate semantic concepts to individual units (Zeiler and Fergus 2014; Zhou et al. 2014): e.g., in vision one looks for a particular unit e_f^i maximally activated by a specific set of input images. The semantic concept shared by these input images is associated with this particular unit. The specific set of images X satisfy

$$X = \arg \max_{x \in \mathbb{I}} \langle \phi(x), e_f^i \rangle \quad (1)$$

where X is the set of selected images, \mathbb{I} is a held-out set of unseen images, $\phi(x)$ means a deep feature representation of input image x , the one-hot vector $e_f^i \in \mathbb{R}^n$ is a basis vector in the feature space E_f , which is associated with the i -th hidden unit, and \max in equations 1 and 2 means the

first N images that maximally activate the i -th unit. We aim to associate semantic concepts with individual vectors in feature space. Formally speaking, given a named semantic concept c (e.g. wheels or cats), a hold-out image dataset \mathbb{I} and a feature extractor $\phi()$ to obtain deep representations, the primary task is to find a vector $v_c = \sum_i^N b_i \cdot e_f^i$ (linear combination of e_f) such that the images X_c that contain the concept c can be selected from \mathbb{I} using

$$X_c = \arg \max_{x \in \mathbb{I}} f(\phi(x), v_c) \quad (2)$$

where $f()$ is a function measuring the similarity between feature representations and the corresponding semantic vector. The challenge is to find a vector v_c and a meaningful function so that the feature vectors of target images X_c are close to v_c . The method to compute the vector v_c and the choice of the function $f()$ will be introduced in Sec. 3. The vector v_c associated with a semantic concept is called semantic vector (SeVec) in this paper.

The main contribution of this paper is to obtain semantic vectors by solving an optimization problem. With the obtained SeVecs, we interpret the deep neural networks globally. We quantify the relationship between semantic concepts learned by deep neural networks, e.g., how important the concept *stripe* is to the concept *zebra*. Furthermore, we explore the multifacetedness of individual semantic concepts. Another contribution is to generate better saliency maps to explain individual decisions made by DNNs using identified semantic vectors.

The next section reviews related work. Sec. 3 introduces and justifies our method to identify semantic concepts in feature space. Sec. 4 interprets the deep convolutional neural network models globally using the obtained SeVecs. Sec. 5 explains individual decisions of DNNs. The last section concludes this paper.

2 Related Work

The works (Zeiler and Fergus 2014; Simonyan, Vedaldi, and Zisserman 2013) confirm the existence of semantic components in Convolutional Neural Networks (CNNs) by illustrating that some filters response to a few images sharing a common concept. (Zhou et al. 2014) shows that part of

units in the neural network trained for scene classification respond to objects in the scenes. Their work also quantitatively measures the interpretability of deep feature representations by evaluating visualizations. (Bau et al. 2017) developed a scalable method to measure the interpretability of deep representations. They measured the alignment between single units and single interpretable concepts without labor-intensive evaluation. In this work, we associate human-interpretable concepts with vectors in feature spaces.

Many saliency methods have been proposed to explain individual classification decisions by creating saliency maps. The perturbation-based forward propagation approaches perturb individual inputs and observe the impact on later neurons in the network. (Zeiler and Fergus 2014; Zintgraf et al. 2017) understand deep features and classifications by analyzing the difference of neuron activations after marginalizing over or perturbing each input patch. The backpropagation-based approaches propagate a relevant signal from a deep layer back into the input space in a single pass, layer-by-layer. The signal thereof can be vanilla gradients (Simonyan, Vedaldi, and Zisserman 2013), their variants (Springenberg et al. 2014; Zhou et al. 2016; Sundararajan, Taly, and Yan 2017), or the combination of gradients and activations (Bach et al. 2015; Shrikumar, Greenside, and Kundaje 2017).

The evaluation of local explanations (saliency maps) has been an active research topic recently (Adebayo et al. 2018; Hooker et al. 2018). The *Completeness* (Bach et al. 2015), *Input Invariance* (Kindermans et al. 2017), *Implementation Invariance* (Sundararajan, Taly, and Yan 2017), *Robustness* (Alvarez-Melis and Jaakkola 2018) of saliency methods are explored in literature. Another widely studied property of saliency maps is their class-discriminativity. Concretely, the saliency maps produced by DeconvNet, Gradient Visualization, and Guided Backpropagation are proven to be not class-discriminative (Mahendran and Vedaldi 2016). Given a classification decision, they produce almost the same saliency map for different classes. In this work, we improve the class-discriminativity of explanations using the identified semantic vectors and measure the improvement quantitatively by a generalized Pointing Game.

3 Semantics in Deep Neural Networks

In this section, we describe how to overcome the aforementioned challenge. Given a semantic concept c and a feature extractor $\phi(\cdot)$, the goal is to compute the direction \mathbf{v}_c corresponding the concept c in the feature space. \mathbf{I}_c is a set of images containing the concept c . The feature extractor $\phi(\cdot)$ is composed of the first K layers of a deep neural network. The extracted features in the K -th layer is $\phi(x)$.

The works (Agrawal, Girshick, and Malik 2014; Dosovitskiy and Brox 2016) concluded that instead of the precise value, the non-zero patterns of feature representations matter to express the discriminative power and code the semantic meaning. Thus, we search for semantic vectors based on the non-zero patterns of feature representations, i.e., binarized feature representations, which are defined as $\mathbf{a}_i = \mathbf{1}_{\phi(x_i) > 0}$ where $x_i \in \mathbf{I}_c$. In the high-dimensional feature space, the non-zero patterns characterize directions in the space.

Thus, the cosine similarity is taken as the function $f(\cdot)$ to describe the distance between examples in the feature space. The direction of the obtained SeVec \mathbf{v}_c should be as close as possible to that of all the binarized feature vectors of $x_i \in \mathbf{I}_c$. This requirement is formulated as an optimization problem in Equation 3.

$$\begin{aligned} \mathbf{v}_c &= \arg \max_{|\mathbf{v}'|=1} \sum_i^M \cos_sim(\mathbf{a}_i, \mathbf{v}') \\ &= \arg \max_{|\mathbf{v}'|=1} \left(\sum_i^M \hat{\mathbf{a}}_i \right) * \mathbf{v}' \\ &= \arg \max_{|\mathbf{v}'|=1} \mathbf{A} * \mathbf{v}' \end{aligned} \quad (3)$$

where $\hat{\mathbf{a}}_i = \frac{\mathbf{a}_i}{|\mathbf{a}_i|}$, \mathbf{v}' is a linear combination of basis vectors in feature space \mathbb{R}^n and the operation $*$ means dot product. The formula satisfies $\mathbf{A} * \mathbf{v}' \leq |\mathbf{A}| \cdot \cos(\theta)$ where θ is the angle between \mathbf{A} and \mathbf{v}' . When $\theta = 0$, i.e., $\mathbf{v}' = \frac{\mathbf{A}}{|\mathbf{A}|}$, the fomular $\mathbf{A} * \mathbf{v}'$ achieves its maximum. Namely, the optimal solution is $\mathbf{v}_c = \frac{\mathbf{A}}{|\mathbf{A}|}$.

We represent each semantic concept with a single vector in the feature space. However, many semantic concepts are multifaceted. To find the number of facets of a semantic concept c , we cluster the feature representations of images \mathbf{I}_c using cosine distance-based clustering methods. We found that there is always a dominant cluster containing most of the samples, which indicates that the neural networks map all the images of the concept c into a single direction of feature space and learn invariant representations. This conclusion has been drawn many times in the previous publications (Donahue et al. 2014; Oquab et al. 2014). Thus, it is reasonable to represent a semantic concept only using a single vector. The multiple facets of semantic concepts in neural networks will be discussed further in subsection 4.2.

In the optimal solution of equation 3, each element of the computed semantic vector is proportional to the activation rate of the corresponding unit. Namely, the value of an element in a SeVec \mathbf{v}_c implies the relevance of the corresponding unit to the concept c . A unit with a low activation rate can be highly activated in a particular image, which could be caused by a cluttered background. Although the higher the activation rate of a unit is, the more important it is, a single unit itself with a high activation rate is not enough to represent the concept c (see the experiment in Sec. 3.1).

The computed SeVec \mathbf{v}_c corresponds to a single direction in feature space in layer K . The vicinity of \mathbf{v}_c is defined as $B_r(\mathbf{v}_c) = \{\mathbf{v} \in \mathbb{R}^n | \cos_dis(\mathbf{v}, \mathbf{v}_c) < r\}$ using cosine similarity which measures the distance of the two directions in feature space. The feature vector $\phi(x)$ of an image containing the concept c are close to \mathbf{v}_c . With an identified SeVec \mathbf{v}_c , the selected images $\mathbf{X}_c = \arg \max_{x \in \mathbb{I}} \cos_dis(\phi(x), \mathbf{v}_c)$ from a hold-out image set is expected to contain the concept c . By using the cosine distance-based nearest neighbor rule, one can partition the feature space into subspaces, each for one semantic concept. Conversely, for a concept c , each element of the SeVec \mathbf{v}_c specifies the relevance of the corresponding dimension to the concept.

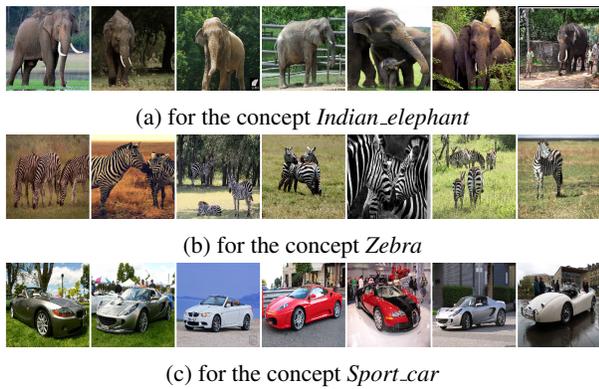


Figure 1: For each semantic concept, the images are selected from a hold-out dataset using the obtained SeVecs.

3.1 Validation of Semantic Vectors

In this subsection, we justify the SeVecs resulting from the optimal solution of Equation 3. In the empirical experiments, we used examples of rectifier neural networks, e.g., the VGG16 network (Simonyan and Zisserman 2015). The pre-trained models are taken from the Pytorch framework. Our experiments are conducted in the feature space corresponding to the first fully-connected ($fc1$) layer of the VGG16 network.

High-level and Low-level Semantic Concepts The training dataset in ILSVRC 2014 (Russakovsky et al. 2015) is labeled with 1000 high-level visual concepts. For each of the 1000 concepts, we compute the corresponding SeVec using the labeled images in the training dataset and select images from a hold-out dataset. For testing, we use the validation dataset as the hold-out dataset. The selected images (top ones) for different concepts are shown in Figure 1. The selected images do contain the concepts that SeVecs correspond to. Although unsurprising, the empirical results show that it is meaningful to represent semantic concepts with vectors in the feature space.

We also validate the SeVecs of low-level visual semantic concepts, such as material, texture, and color. To compute the SeVec of a concept, our algorithm requires a number of images containing the concept. However, such labels are not available in the ImageNet database. We turn to other annotated datasets. For concepts related to texture, material and color, we use the Describable Textures Dataset, Flickr Material Database and Google-512 respectively.

We identify SeVecs of 47 textures, 10 materials and 11 colors using the available labeled images in the three datasets. From the validation dataset of ILSVRC 2014, the images that lie in the vicinity of the corresponding SeVec are selected and shown in Figure 2. For materials and textures, the selected images contain the corresponding semantic concepts in subfigures 2a and 2b. However, the selected images for color concepts do not show the corresponding color concepts in subfigure 2c.

Counter-intuitively, we argue that the color information is not essential in deep representations (e.g., the $fc1$ layer in

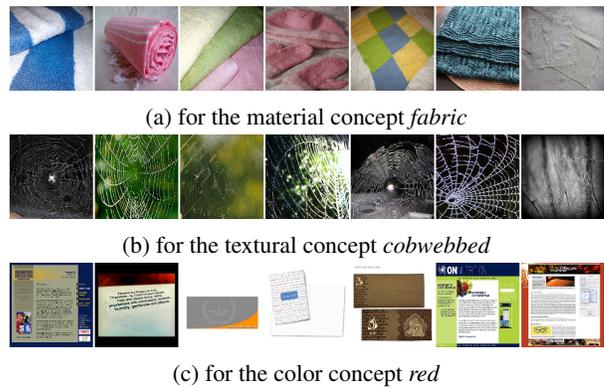


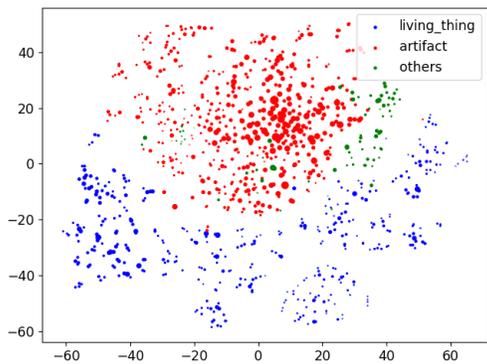
Figure 2: The selected images using the SeVecs corresponding the low-level visual concepts.

Associated Neuron	Random Neuron	SeVec	Permutation of SeVec
1.05e-06	-6.72e-07	0.1929	4.25e-06

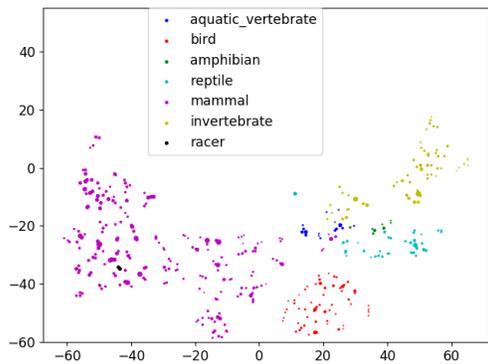
Table 1: The increased scores of the target output units by modifying the representation using the component associated with semantic concepts (average on 1000 concepts of 50K images).

VGG16). We verify this argument with an ablation study on the validation dataset of ILSVRC 2014. We convert the color images into grey ones and duplicate them in three channels to fit the pre-trained models. The classification performance of VGG16 and AlexNet do not drop significantly. The misclassifications thereof are not necessarily caused by the loss of color information. Compared to original images, the new 'grey' input images are translated in each channel. Such translation can potentially lead to misclassification (due to the vulnerability of neural networks (Szegedy et al. 2014; Goodfellow, Shlens, and Szegedy 2015)). The dropped performance is totally recovered after retraining. Hence, we argue that the color information is not essential for the classification in existing deep CNNs. Although VGG16 is not trained with classes in the texture and the material datasets, the obtained SeVecs can still identify the images containing the corresponding concepts.

Single Neuron VS. Semantic Vector Does a single neuron or a distributed vector represent the semantic concept in neural networks? In a deep neural network, we observe the changes of the output probabilities of ground-truth classes in case of perturbing the activations of the K -th layer. Given an image containing a concept c , we identify the single neuron in the K -th layer that is most often activated by the concept c in the forward inference, i.e., the maximal element in SeVec v_c . We modify the representation by assigning a bigger value (1.5 times the biggest activation in the same layer) to the neuron. If it is the single neuron that corresponds to the semantic concept c , the output score of the ground-truth class is supposed to increase. As a comparison, we do the same modification on the activation of a randomly chosen neuron.



(a) 1000 SeVecs of living_things, artifact and others



(b) SeVecs belonging to living_thing (blue points left)

Figure 3: The SeVecs of 1000 semantic concepts (corresponding to 1000 target classes in ImageNet 1k) are visualized using t-SNE (Maaten and Hinton 2008). Each circle corresponds to a semantic concept, and the size of the circle corresponds to the degree of diversity of the semantic concept.

We argue that, instead of individual neurons, the directions in feature space correspond to semantic concepts. We modify the representation by setting it closer to the corresponding SeVec (i.e., multiplying $\mathbf{1}_{v_c > 0.5}$) and observe the changes of the output of the ground-truth class. For a fair comparison, we also modify the representation by multiplying it with a random permutation of $\mathbf{1}_{v_c > 0.5}$.

We conduct experiments on the validation datasets of ILSVRC 2014 containing 50K images. The results of four types of modifications are in Table 1. The positive values therein are the increased probability value of the corresponding ground-truth class, while the negative values mean the dropped one. If the high activation of the associated neuron means the existence of the visual concept, the output probability of the target should increase. However, the output probabilities of target classes hardly change in case of perturbing a single neuron. The modification using the SeVec increases the confidence by about 20%. The comparison group has almost no impact on the final output, which ensures that the increased confidence is caused by the semantic meaning instead of the large modification. The results are consistent with the argument that not individual units, but feature vectors are associated with the semantic concepts (Szegedy et al. 2014).



(a) the multi-faceted concept *scale*



(b) the concept *indigo_bird* with fewer facets

Figure 4: The selected images for the concept with the big or small number of facets.

4 Interpreting Deep CNNs Globally

In this section, we explore global interpretation of neural networks from the perspective of high-level semantic concepts. We answer the following questions: Do CNNs learn semantic-concept hierarchy? How CNNs express the multiple facets of semantic concepts? Do CNNs learn the relationship between high-level concepts and low-level concepts?

4.1 Hierarchy of Semantic Concepts in CNNs

Previous work (Donahue et al. 2014) visualizes feature vectors of images directly and shows that the feature vectors of the images from similar classes lie near to each other. The semantically similar concepts are visually similar. We aim to verify that semantic concepts learned by CNNs also form a hierarchy. Each of the learned semantic concepts is represented by a SeVec. The distance between their SeVecs characterizes the relationship between two semantic concepts. The SeVecs of similar concepts are expected to be near to each other in the feature space.

In the feature space corresponding the *fc1* layer in VGG16. We analyze the 1000 computed SeVecs together. We project the 1000 SeVecs into two-dimensional space using t-SNE (Maaten and Hinton 2008). The visualization is shown in Figure 3 where each point corresponds to a SeVec (a concept in ImageNet 1k database). The database is organized hierarchically. Two or more concepts may belong to the same category in a high level of the hierarchy (e.g., *Wader & Turdus & ...* \rightarrow *Bird* \rightarrow *living_things*).

In the subfigure 3a, we mark each concept using high-level labels, i.e., *living_things*, *artifact* and *others*. It shows that the SeVecs are well clustered. Each cluster corresponds to a high-level semantic concept. Furthermore, we visualize one of clusters *living_things* in subfigures 3b where we mark the semantic concepts with corresponding lower-level labels, i.e., subcategories of *living_things*. The clear subclusters can also be observed. The hierarchical clusters suggest that VGG16 has learned semantic-concept hierarchy.

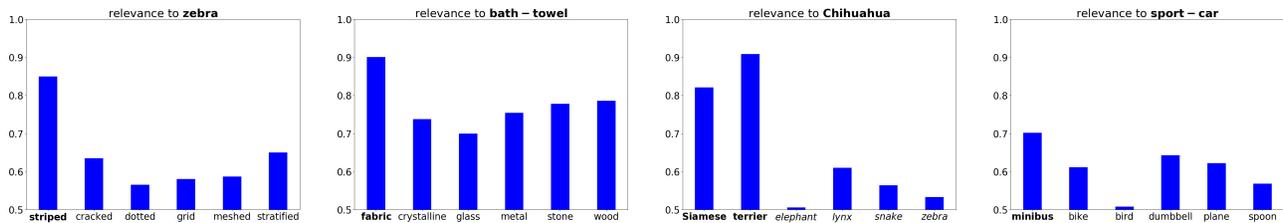


Figure 5: The global interpretation of semantic concepts learned by deep convolutional neural networks. Each figure describes the relationship between semantic concepts. The relevance is calculated on SecVecs using $\cosine(\mathbf{v}_a, \mathbf{v}_b)$.

4.2 Multiple Facets of Semantic Concepts in CNNs

The activation maximization method (Erhan et al. 2009) generates images that maximally activate a neuron without considering its multifacetedness. To understand the multifacetedness of neurons, (Nguyen, Yosinski, and Clune 2016) separately synthesizes each type of image a neuron fires in response to using prior initialization and regularization methods. The multiple facets of a class are formulated as intra-class knowledge in CNN (Wei et al. 2015). Location-variation is expressed in $pool_5$ layer, when content-variation is expressed in f_{c_2} layer in the VGG16 model.

The multiple facets of semantic concepts learned by CNNs are investigated in this work. As discussed before, deep CNNs map most images of a semantic concept into a single direction in the feature space. Clustering in the feature space is not able to define the number of facets of semantic concepts. Thus, instead of the numbers, we start from defining the degree of diversity of semantic concepts. For a semantic concept c , the degree of its diversity is defined as $D_c = 1 - \frac{1}{M} \sum_i^M \cos_sim(\mathbf{a}_i, \mathbf{v}_c)$ where M is the number of images and \mathbf{a}_i is binarized feature representations of an image containing the semantic concept c .

We visualize the degree of diversity of semantic concepts in Figure 3a. Each circle corresponds to a semantic concept, the size of the circle encodes the D_c of semantic concepts. Different semantic concepts show different degrees of the diversity. The bigger the size is, the larger the number of facets of the corresponding semantic concept is. While some concepts in artifacts show very high diversity, the number of facets of Birds-related concepts is relatively small. More concretely, for instance, the concept *scale* is multifaceted to a great degree. The degree of diversity of the concepts *indigo.bird* is very low. The selected images are shown in Figure 4.

Does the classification performance of CNNs correlate to the degree of diversity of the concept? We compute the degree of diversity of 1000 semantic concepts D_c and the classification performance of CNNs on the corresponding 1000 classes. The Pearson correlation coefficient and the p-value for testing non-correlation for the two variables, i.e., the D_c and the classification performance, are shown in the table 2. The table indicates the classification performance of CNNs strongly depends on the degree of multifacetedness of the classes. The higher the degree of the multifacetedness of classes is, the lower the classification performance is.

		Top 1 accuracy	Top 5 accuracy
D_c	Correlation Coefficient	-0.5245	-0.4727
	P-value	9.395e-72	8.485e-57

Table 2: The p-value is close to zero, which means the no-correlation assumption is not held. The correlation coefficients with negative values indicate the degree of diversity of semantic concepts is negatively correlated to the top-1 and top-5 classification performance.

4.3 Explanations in CNNs Beyond Saliency Maps

Most previous methods explain image classification decisions by producing saliency maps. In this experiment, we explain the classification decisions with semantic concepts. The CNNs have learned both high-level and low-level concepts. Do CNNs learn the common sense about the relationship between concepts? The work TCAV (Kim et al. 2018) represents semantic concepts using derivatives of the corresponding local linear classifier trained in a specific feature space. The built TCAVs were used to describe the relationship between different concepts. Similarly, we defined the relevance of the concept a to the concept b as $\cos_sim(\mathbf{v}_a, \mathbf{v}_b)$. We list the relationship between the related concepts in Figure 5. The relationship described in the figure corresponds to our common sense. For example, the texture *stripe* is more important to the concept *zebra* than other texture concepts.

The individual classification decision can also be explained similarly. The individual explanations with low-level concepts are shown in Table 3. The CNN model predicts the object in the first image as *stone.wall* because *it shows cracked texture and its material is stone*. Such explanations with low-level concepts can ensure that the model’s predictions base on correct low-level concepts. We also show inappropriate explanations created by this method (marked by a slash). For instance, the *dalmatian* (a species of dog) shows freckled texture, and it is hard to describe what is the material of a *dalmatian*. None of the concepts in Flickr Material Database can be used to describe the material. Similarly, the texture of a whole streetcar is indescribable using the concepts in DTD. Such a simple and novel explanation with low-level concepts can help to understand individual classification decisions.

							
Texture:	cracked	crystalline	grooved	knitted	bumpy	freckled	porous
Material:	stone	glass	wood	fabric	foliage	plastic	metal
Prediction:	<i>stone_wall</i>	<i>water_bottle</i>	<i>picket_fence</i>	<i>bath_towel</i>	<i>strawberry</i>	<i>dalmatian</i>	<i>streetcar</i>

Table 3: The explanations of the classification decisions by illustrating related low-level visual concepts in input images such as the texture, the material of the recognized objects.

5 Interpreting Deep CNNs Locally

A large number of saliency methods explain local decisions of deep CNNs via backpropagation processes. They propagate a class-relevant signal back through networks until the input layer and visualizes the signals received by inputs in saliency maps. The existing approaches use no explicit high-level semantic information when propagating the signals.

After a glance at an image, if we are asked to find the cat in the image, we will search for a cat with a virtual cat pattern in our mind. This phenomenon is explained in the Biased Competition Theory of cognitive science (Beck and Kastner 2009). In object detection, our visual attention is typically dominated by a goal (high-level semantic information) in a top-down manner. In the feedback loop in brains, the non-relevant neurons are suppressed based on the high-level semantic information.

Inspired by the biased competition theory, we aim to build a similar suppression process using our SeVecs in existing backpropagation-based saliency methods. The obtained SeVecs are applied to suppress the irrelevant internal neurons in backpropagation processes.

Concretely, Guided Backpropagation (GuidedBP) approach combines DeConvNet and Gradient Visualization to produce better saliency maps. The local saliency method describes how the output of the target unit changes for small perturbations around the original input. A global saliency method by multiplying the saliency map with the input describes the marginal effect of a feature on the output with respect to a reference point (Ancona et al. 2018). In rectifier neural networks, the method Gradient*Input is equivalent to ϵ -LRP (Bach et al. 2015) and DeepLIFT (Shrikumar, Greenside, and Kundaje 2017). Without loss of generality, we only consider the local attribution method GuidedBP and the global attribution method Gradient*Input. We will demonstrate our idea on these two simple and representative methods instead of exhaustively generalizing to all existing backpropagation-based saliency methods.

We focus on the *discriminativeness* of explanation maps. For images with multiple objects, a deep CNN have high probabilities for each related class. Given an image x with multiple objects (concepts) $C = \{c_1, c_2, \dots, c_n\}$, each of which corresponds to one class in the output layer, the CNN model makes a prediction for the image. The output probability is $O = \{o_1, o_2, \dots, o_n\}$. We are supposed to create an explanation map for a unit o_i . Explanations should focus on the class-discriminative part of the input image.

Most existing attribution methods suppress the neurons based on their activations or gradients. The Guided Backpropagation and Gradient*Input are built on the gradient values $\frac{\partial o_i}{\partial x}$. In handling ReLU layers in a backward pass, $R^l = R^{l+1} \mathbf{1}_{R^{l+1} > 0 \text{ and } X^l > 0}$ where $R^{n-1} = \frac{\partial o_i}{\partial X^{n-1}}$ and o_i is the i -th class-specific unit, the X^l are the activations before ReLU layer, and $\mathbf{1}$ is the indicator function. However, the neuron activations may caused by cluttered background or other irrelevant concepts. The selection of irrelevant neurons should not depend only on the activations.

We suppress the concept-irrelevant neurons using high-level semantic information. We first compute the SeVec V_i of the class concept o_i in a deep layer. The SeVec we choose is in the feature space of the $pool_5$ in VGG16. The reason for the choice is that the layer maps the spatial information to semantic concepts. This feature space encodes location-variation variance (Wei et al. 2015). The SeVec V_i is applied to suppress the irrelevant neurons to filter the irrelevant information in backward pass, $R^l = R^{l+1} \mathbf{1}_{R^{l+1} > 0 \text{ and } X^l > 0 \text{ and } V_i > 0.5}$.

We only suppress the irrelevant neurons in $pool_5$ layer, which is the most important layer to encode the spatial variation (Wei et al. 2015). The feedback CNN (Cao et al. 2015) suppresses neurons in all layers by maximizing the output o_i . The optimization thereof is an NP-hard problem. The approximated solution proposed in their paper is inefficient, which require many-times backpropagation. Our semantics-based method only requires one backward pass. Our semantics-based method can also be integrated into other saliency methods without much extra cost.

5.1 Qualitative Evaluation

The explanations created by the baseline methods and our semantics-based versions are shown in Figure 6. Each column corresponds to a saliency method. For an given image classified by a CNN, the saliency maps are created separately for two related classes (e.g. *car* and *bike* in the first image). The saliency maps are suppose to identify the features supporting a given class.

For instance, in the first row, the saliency maps should identify the features relevant to *car*. We can observe that the baseline methods (GuidedBP and Gradient*Input) also identify pixels on the *bike*, while our semantic versions (the second and the fourth column) focus more on the pixels on the *car*. Similarly, the second row aims to capture *bike*. While the two baseline approaches visualize both objects

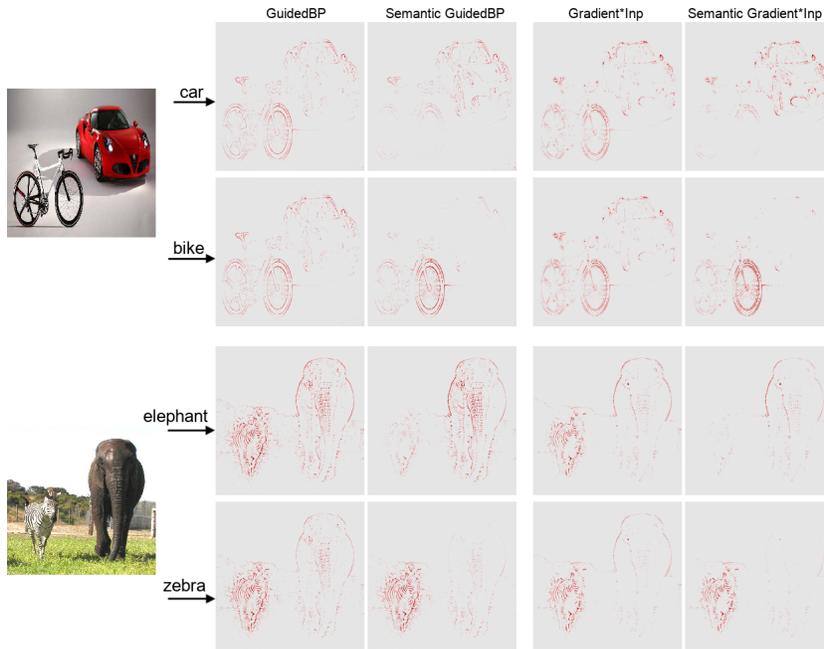


Figure 6: The figure shows explanation maps created by the baseline methods and our semantic versions. Each column corresponds to a saliency method, and each row shows the saliency maps that support classification of a given class.

and produce visually similar maps for two related classes, our two semantics-based methods are able to identify the relevant object accurately.

(Nie, Zhang, and Patel 2018) shows that GuidedBP is essentially doing (partial) image recovery which is unrelated to the network decisions. We leverage SeVecs to improve the discriminativeness of GuidedBP. Our semantic GuidedBP does explain the local decision by creating class-discriminative saliency maps. We only compare with baseline approaches to show the improvement since our goal is not to push the state-of-the-art saliency method.

5.2 Quantitative Evaluation

The various properties of explanatory saliency maps are explored in publications (Adebayo et al. 2018; Hooker et al. 2018). In this experiment, we aim to evaluate the discriminativeness of explanatory saliency maps. To quantitatively evaluate the discriminativeness, (Zhang et al. 2016) proposes a pointing task where the maximum point of the saliency map is extracted and evaluated. A hit is counted if the maximum point lies in the bounding box of the target object, otherwise a miss is counted. The localization accuracy is measured by $Acc = \frac{\#Hits}{\#Hits + \#Misses}$. We found that the naive pointing at the center of the image shows surprisingly high accuracy. SmoothGrad (Smilkov et al. 2017) even clips the maximum point to obtain a better visualization. Hence, we generalize the pointing task into a more comprehensive setting. In the new setting, the first step is to preprocess the saliency map by simply thresholding so that m percent energy is kept. A hit is counted if the remaining foreground

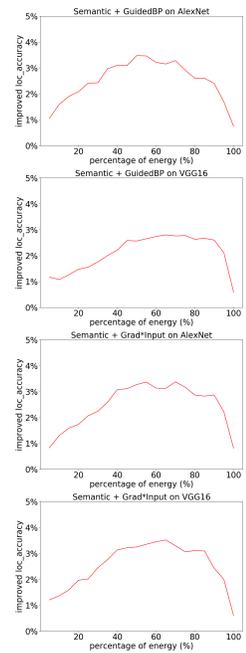


Figure 7: The improved localization accuracy by using our obtained semantic information

area (containing relevant pixels) lies in the bounding box of the corresponding target object, otherwise a miss is counted.

The experiments are conducted on two pre-trained deep CNNs (i.e. AlexNet (Krizhevsky, Sutskever, and Hinton 2012) and VGG16). Using the baseline approaches and ours, we create saliency maps for ground truth class of each image in the validation dataset of ILSVRC. The improved localization accuracy on the two models is shown in Figure 7. When the kept energy is close to 0, the generalized Pointing Game becomes close to the original Pointing Game. In all subplots, the improved localization accuracy is always bigger than zero when the kept energy varies from 0 to 100%. On both models, the localization ability of saliency maps created by our semantic-based method consistently outperforms that of baseline methods. This numerous evidence shows that our SeVecs help to improve the discriminativeness of explanatory saliency maps.

6 Conclusion

In this work, we associate human-interpretable semantic concepts with vectors in a feature space, which is formulated as an optimization problem. We apply the semantic vectors obtained from the optimal solution to interpret the convolutional deep neural networks globally and explain individual classification decisions. In addition to understanding positive aspects of them, we also found limitations of the existing deep CNNs. They do not make full use of color information of input images, and perform much worse on the classes with high multifacetedness. Furthermore, a new interpretable semantics-based architecture is desired when we aim to gain trust from users in real-world applications.

References

- Adebayo, J.; Gilmer, J.; Muelly, M.; Goodfellow, I.; Hardt, M.; and Kim, B. 2018. Sanity checks for saliency maps. In *NeurIPS*, 9505–9515.
- Agrawal, P.; Girshick, R.; and Malik, J. 2014. Analyzing the performance of multilayer neural networks for object recognition. In *ECCV*, 329–344. Springer.
- Alvarez-Melis, D., and Jaakkola, T. S. 2018. On the robustness of interpretability methods. In *2018 Workshop on Human Interpretability in Machine Learning (WHI)*.
- Ancona, M.; Ceolini, E.; Oztireli, C.; and Gross, M. 2018. Towards better understanding of gradient-based attribution methods for deep neural networks. In *ICLR*.
- Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.-R.; and Samek, W. 2015. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PloS one* 10(7):e0130140.
- Bau, D.; Zhou, B.; Khosla, A.; Oliva, A.; and Torralba, A. 2017. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*.
- Beck, D. M., and Kastner, S. 2009. Top-down and bottom-up mechanisms in biasing competition in the human brain. *Vision research* 49(10):1154–1165.
- Cao, C.; Liu, X.; Yang, Y.; Yu, Y.; et al. 2015. Look and think twice: Capturing-down visual attention with feedback convolutional neural networks. In *ICCV*, 2956–2964.
- Donahue, J.; Jia, Y.; Vinyals, O.; Hoffman, J.; Zhang, N.; Tzeng, E.; and Darrell, T. 2014. Decaf: A deep convolutional activation feature for generic visual recognition. In *ICML*, 647–655.
- Dosovitskiy, A., and Brox, T. 2016. Inverting visual representations with convolutional networks. In *CVPR*, 4829–4837.
- Erhan, D.; Bengio, Y.; Courville, A.; and Vincent, P. 2009. Visualizing higher-layer features of a deep network.
- Goodfellow, I. J.; Shlens, J.; and Szegedy, C. 2015. Explaining and harnessing adversarial examples. In *ICLR*.
- Hooker, S.; Erhan, D.; Kindermans, P.-J.; and Kim, B. 2018. Evaluating feature importance estimates. In *2018 Workshop on Human Interpretability in Machine Learning (WHI)*.
- Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *ICML*, 2673–2682.
- Kindermans, P.-J.; Hooker, S.; Adebayo, J.; Alber, M.; Schütt, K. T.; Dähne, S.; Erhan, D.; and Kim, B. 2017. The (un) reliability of saliency methods.
- Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *NeurIPS*, 1097–1105.
- Maaten, L. v. d., and Hinton, G. 2008. Visualizing data using t-sne. *Journal of machine learning research* 9(Nov):2579–2605.
- Mahendran, A., and Vedaldi, A. 2016. Salient deconvolutional networks. In *ECCV*, 120–135. Springer.
- Nguyen, A.; Yosinski, J.; and Clune, J. 2016. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. *arXiv preprint arXiv:1602.03616*.
- Nie, W.; Zhang, Y.; and Patel, A. 2018. A theoretical explanation for perplexing behaviors of backpropagation-based visualizations. In *2018 Workshop on Human Interpretability in Machine Learning (WHI)*.
- Oquab, M.; Bottou, L.; Laptev, I.; and Sivic, J. 2014. Learning and transferring mid-level image representations using convolutional neural networks. In *CVPR*, 1717–1724.
- Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; et al. 2015. Imagenet large scale visual recognition challenge. *ICCV* 115(3):211–252.
- Shrikumar, A.; Greenside, P.; and Kundaje, A. 2017. Learning important features through propagating activation differences. In *ICML*.
- Simonyan, K., and Zisserman, A. 2015. Very deep convolutional networks for large-scale image recognition. In *ICLR*.
- Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2013. Deep inside convolutional networks: Visualising image classification models and saliency maps. In *ICLR*.
- Smilkov, D.; Thorat, N.; Kim, B.; Viégas, F.; and Wattenberg, M. 2017. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*.
- Springenberg, J. T.; Dosovitskiy, A.; Brox, T.; and Riedmiller, M. A. 2014. Striving for simplicity: The all convolutional net. In *ICLR*.
- Sundararajan, M.; Taly, A.; and Yan, Q. 2017. Axiomatic attribution for deep networks. In *ICML*.
- Szegedy, C.; Zaremba, W.; Sutskever, I.; Bruna, J.; Erhan, D.; Goodfellow, I. J.; and Fergus, R. 2014. Intriguing properties of neural networks. In *ICLR*.
- Wei, D.; Zhou, B.; Torralla, A.; and Freeman, W. 2015. Understanding intra-class knowledge inside cnn. *arXiv preprint arXiv:1507.02379*.
- Zeiler, M. D., and Fergus, R. 2014. Visualizing and understanding convolutional networks. In *ECCV*, 818–833. Springer.
- Zhang, J.; Lin, Z.; Brandt, J.; Shen, X.; and Sclaroff, S. 2016. Top-down neural attention by excitation backprop. In *ECCV*, 543–559. Springer.
- Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2014. Object detectors emerge in deep scene cnns. *arXiv preprint arXiv:1412.6856*.
- Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *CVPR*, 2921–2929.
- Zintgraf, L. M.; Cohen, T.; Adel, T.; and Welling, M. 2017. Visualizing deep neural network decisions: Prediction difference analysis. In *ICLR*.