Visual Interpretability for Deep Learning: a Survey

Quanshi Zhang and Song-Chun Zhu
University of California, Los Angeles

Abstract

This paper reviews recent studies in understanding neural-network representations and learning neural networks with interpretable/disentangled middle-layer representations. Although deep neural networks have exhibited superior performance in various tasks, the interpretability is always the Achilles' heel of deep neural networks. At present, deep neural networks obtain high discrimination power at the cost of low interpretability of their black-box representations. We believe that high model interpretability may help people to break several bottlenecks of deep learning, *e.g.* learning from very few annotations, learning via human-computer communications at the semantic level, and semantically debugging network representations. We focus on convolutional neural networks (CNNs), and we revisit the visualization of CNN representations, methods of diagnosing representations of pre-trained CNNs, approaches for disentangling pre-trained CNN representations, learning of CNNs with disentangled representations, and middle-to-end learning based on model interpretability. Finally, we discuss prospective trends in explainable artificial intelligence.

1 Introduction

Convolutional neural networks (CNNs) [LeCun *et al.*, 1998a; Krizhevsky *et al.*, 2012; He *et al.*, 2016; Huang *et al.*, 2017] have achieved superior performance in many visual tasks, such as object classification and detection. However, the end-to-end learning strategy makes CNN representations a black box. Except for the final network output, it is difficult for people to understand the logic of CNN predictions hidden inside the network. In recent years, a growing number of researchers have realized that high model interpretability is of significant value in both theory and practice and have developed models with interpretable knowledge representations.

In this paper, we conduct a survey of current studies in understanding neural-network representations and learning neural networks with interpretable/disentangled representations. We can roughly define the scope of the review into the following six research directions.

- Visualization of CNN representations in intermediate network layers. These methods mainly synthesize the image that maximizes the score of a given unit in a pre-trained CNN or invert feature maps of a conv-layer back to the input image. Please see Section 2 for detailed discussion.
- Diagnosis of CNN representations. Related studies may either diagnose a CNN's feature space for different object categories or discover potential representation flaws in conv-layers. Please see Section 3 for details.
- Disentanglement of "the mixture of patterns" encoded in each filter of CNNs. These studies mainly disentangle complex representations in conv-layers and transform network representations into interpretable graphs. Please see Section 4 for details.
- Building explainable models. We discuss interpretable CNNs [Zhang *et al.*, 2017c], capsule networks [Sabour *et al.*, 2017], interpretable R-CNNs [Wu *et al.*, 2017], and the InfoGAN [Chen *et al.*, 2016] in Section 5.
- Semantic-level middle-to-end learning via human-computer interaction. A clear semantic disentanglement of CNN representations may further enable "middle-to-end" learning of neural networks with weak supervision. Section 7 introduces methods to learn new models via human-computer interactions [Zhang *et al.*, 2017b] and active question-answering with very limited human supervision [Zhang *et al.*, 2017a].

Among all the above, the visualization of CNN representations is the most direct way to explore network representations. The network visualization also provides a technical foundation for many approaches to diagnosing CNN representations. The disentanglement of feature representations of a pre-trained CNN and the learning of explainable network representations present bigger challenges to state-of-the-art algorithms. Finally, explainable or disentangled network representations are also the starting point for weakly-supervised middle-to-end learning.

Values of model interpretability: The clear semantics in high conv-layers can help people trust a network's prediction. As discussed in [Zhang *et al.*, 2018b], considering dataset and representation bias, a high accuracy on testing images still cannot ensure that a CNN will encode correct represen-

tations. For example, a CNN may use an unreliable context—eye features—to identify the "lipstick" attribute of a face image. Therefore, people usually cannot fully trust a network unless a CNN can semantically or visually explain its logic, *e.g.* what patterns are used for prediction.

In addition, the middle-to-end learning or debugging of neural networks based on the explainable or disentangled network representations may also significantly reduce the requirement for human annotation. Furthermore, based on semantic representations of networks, it is possible to merge multiple CNNs into a universal network (*i.e.* a network encoding generic knowledge representations for different tasks) at the semantic level in the future.

In the following sections, we review the above research directions and discuss the potential future of technical developments.

2 Visualization of CNN representations

Visualization of filters in a CNN is the most direct way of exploring visual patterns hidden inside a neural unit. Different types of visualization methods have been developed for network visualization.

First, gradient-based methods [Zeiler and Fergus, 2014; Mahendran and Vedaldi, 2015; Simonyan *et al.*, 2013; Springenberg *et al.*, 2015] are the mainstream of network visualization. These methods mainly compute gradients of the score of a given CNN unit *w.r.t.* the input image. They use the gradients to estimate the image appearance that maximizes the unit score. [Olah *et al.*, 2017] has provided a toolbox of existing techniques to visualize patterns encoded in different conv-layers of a pre-trained CNN.

Second, the up-convolutional net [Dosovitskiy and Brox, 2016] is another typical technique to visualize CNN representations. The up-convolutional net inverts CNN feature maps to images. We can regard up-convolutional nets as a tool that indirectly illustrates the image appearance corresponding to a feature map, although compared to gradient-based methods, up-convolutional nets cannot mathematically ensure that the visualization result exactly reflects actual representations in the CNN. Similarly, [Nguyen *et al.*, 2017] has further introduced an additional prior, which controls the semantic meaning of the synthesized image, to the adversarial generative network. We can use CNN feature maps as the prior for visualization.

In addition, [Zhou *et al.*, 2015] has proposed a method to accurately compute the image-resolution receptive field of neural activations in a feature map. The actual receptive field of neural activation is smaller than the theoretical receptive field computed using the filter size. The accurate estimation of the receptive field helps people to understand the representation of a filter.

3 Diagnosis of CNN representations

Some methods go beyond the visualization of CNNs and diagnose CNN representations to obtain insight understanding of features encoded in a CNN. We roughly divide all relevant research into the following five directions.

Studies in the first direction analyze CNN features from a global view. [Szegedy *et al.*, 2014] has explored semantic meanings of each filter. [Yosinski *et al.*, 2014] has analyzed the transferability of filter representations in intermediate conv-layers. [Lu, 2015; Aubry and Russell, 2015] have computed feature distributions of different categories/attributes in the feature space of a pre-trained CNN.

The second research direction extracts image regions that directly contribute the network output for a label/attribute to explain CNN representations of the label/attribute. This is similar to the visualization of CNNs. Methods of [Fong and Vedaldi, 2017; Selvaraju *et al.*, 2017] have been proposed to propagate gradients of feature maps *w.r.t.* the final loss back to the image plane to estimate the image regions. The LIME model proposed in [Ribeiro *et al.*, 2016] extracts image regions that are highly sensitive to the network output. Studies of [Zintgraf *et al.*, 2017; Kindermans *et al.*, 2017; Kumar *et al.*, 2017] have invented methods to visualize areas in the input image that contribute the most to the decision-making process of the CNN. [Wang *et al.*, 2017; Goyal *et al.*, 2016] have tried to interpret the logic for visual question-answering encoded in neural networks. These studies list important objects (or regions of interests) detected from the images and crucial words in questions as the explanation of output answers.

The estimation of vulnerable points in the feature space of a CNN is also a popular direction for diagnosing network representations. Approaches of [Su *et al.*, 2017; Koh and Liang, 2017; Szegedy *et al.*, 2014] have been developed to compute adversarial samples for a CNN. *I.e.* these studies aim to estimate the minimum noisy perturbation of the input image that can change the final prediction. In particular, influence functions proposed in [Koh and Liang, 2017] can be used to compute adversarial samples. The influence function can also provide plausible ways to create training samples to attack the learning of CNNs, fix the training set, and further debug representations of a CNN.

The fourth research direction is to refine network representations based on the analysis of network feature spaces. Given a CNN pre-trained for object classification, [Lakkaraju *et al.*, 2017] has proposed a method to discover knowledge blind spots (unknown patterns) of the CNN in a weakly-supervised manner. This method grouped all sample points in the entire feature space of a CNN into thousands of pseudo-categories. It assumed that a well learned CNN would use the sub-space of each pseudo-category to exclusively represent a subset of a specific object class. In this way, this study randomly showed object samples within each sub-space, and used the sample purity in the sub-space to discover potential representation flaws hidden in a pre-trained CNN. To distill representations of a teacher network to a student network for sentiment analysis, [Hu *et al.*, 2016] has proposed using logic rules of natural languages (*e.g.* I-ORG cannot follow B-PER) to construct a distillation loss to supervise the knowledge distillation of neural networks, in order to obtain more meaningful network representations.

Finally, [Zhang *et al.*, 2018b] has presented a method to discover potential, biased representations of a CNN. Fig. 1 shows biased representations of a CNN trained for the estima-

Figure 1: Biased representations in a CNN [Zhang *et al.*, 2018b]. Considering potential dataset bias, a high accuracy on testing images cannot always ensure that a CNN learns correct representations. The CNN may use unreliable co-appearing contexts to make predictions. For example, people may manually modify mouth appearances of two faces by masking mouth regions or pasting another mouth, but such modifications do not significantly change prediction scores for the *lipstick* attribute. This figure shows heat maps of inference patterns of the *lipstick* attribute, where patterns with red/blue colors are positive/negative with the attribute score. The CNN mistakenly considers unrelated patterns as contexts to infer the lipstick.

tion of face attributes. When an attribute usually co-appears with specific visual features in training images, then the CNN may use such co-appearing features to represent the attribute. When the used co-appearing features are not semantically related to the target attribute, these features can be considered as biased representations.

Given a pre-trained CNN (*e.g.* a CNN that was trained to estimate face attributes), [Zhang *et al.*, 2018b] required people to annotate some ground-truth relationships between attributes, *e.g.* the *lipstick* attribute is positively related to the *heavy-makeup* attribute, and is not related to the *black hair* attribute. Then, the method mined inference patterns of each attribute output from conv-layers, and used inference patterns to compute actual attribute relationships encoded in the CNN. Conflicts between the ground-truth and the mined attribute relationships indicated biased representations.

4 Disentangling CNN representations into explanatory graphs & decision trees

4.1 Disentangling CNN representations into explanatory graphs

Compared to the visualization and diagnosis of network representations in previous sections, disentangling CNN features into human-interpretable graphical representations (namely *explanatory graphs*) provides a more thorough explanation of network representations. [Zhang *et al.*, 2018a; Zhang *et al.*, 2016] have proposed disentangling features in conv-layers of a pre-trained CNN and have used a graphical model to represent the semantic hierarchy hidden inside a CNN.

As shown in Fig. 2, each filter in a high conv-layer of a CNN usually represents a mixture of patterns. For example, the filter may be activated by both the head and the tail parts of an object. Thus, to provide a global view of how visual knowledge is organized in a pre-trained CNN, studies of [Zhang *et al.*, 2018a; Zhang *et al.*, 2016] aim to answer the following three questions.

- How many types of visual patterns are memorized by each convolutional filter of the CNN (here, a visual pattern may describe a specific object part or a certain texture)?
- Which patterns are co-activated to describe an object part?
- What is the spatial relationship between two co-activated patterns?

Figure 2: Feature maps of a filter obtained using different input images [Zhang *et al.*, 2018a]. To visualize the feature map, the method propagates receptive fields of activated units in the feature map back to the image plane. In each sub-feature, the filter is activated by various part patterns in an image. This makes it difficult to understand the semantic meaning of a filter.

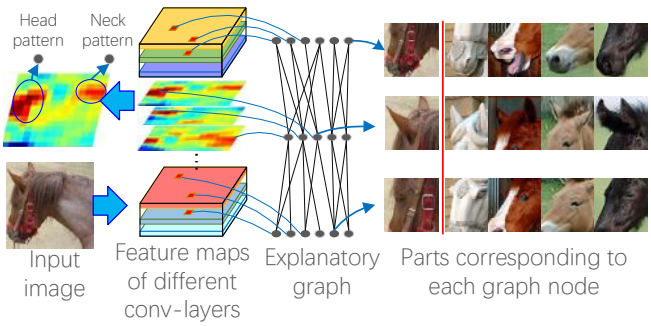

Figure 3: Explanatory graph [Zhang *et al.*, 2018a]. An explanatory graph represents the knowledge hierarchy hidden in conv-layers of a CNN. Each filter in a pre-trained CNN may be activated by different object parts. [Zhang *et al.*, 2018a] disentangles part patterns from each filter in an unsupervised manner, thereby clarifying the knowledge representation.

As shown in Fig. 3, the explanatory graph explains the knowledge semantic hidden inside the CNN. The explanatory graph disentangles the mixture of part patterns in each filter's feature map of a conv-layer, and uses each graph node to represent a part.

- The explanatory graph has multiple layers. Each graph layer corresponds to a specific conv-layer of a CNN.
- Each filter in a conv-layer may represent the appearance of different object parts. The algorithm automatically disentangles the mixture of part patterns encoded in a single filter, and uses a node in the explanatory graph to represent each part pattern.
- Each node in the explanatory graph consistently represents the same object part through different images. We can use the node to localize the corresponding part on

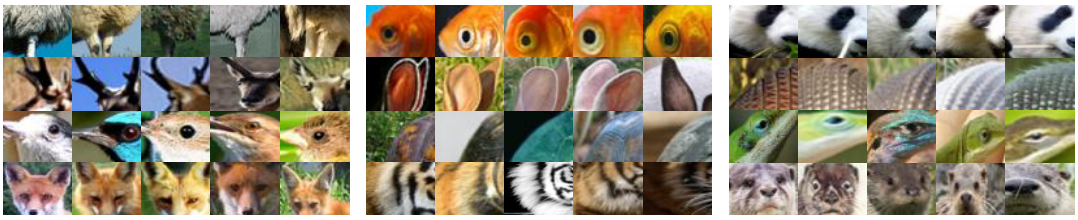

Figure 4: Image patches corresponding to different nodes in the explanatory graph [Zhang *et al.*, 2018a].

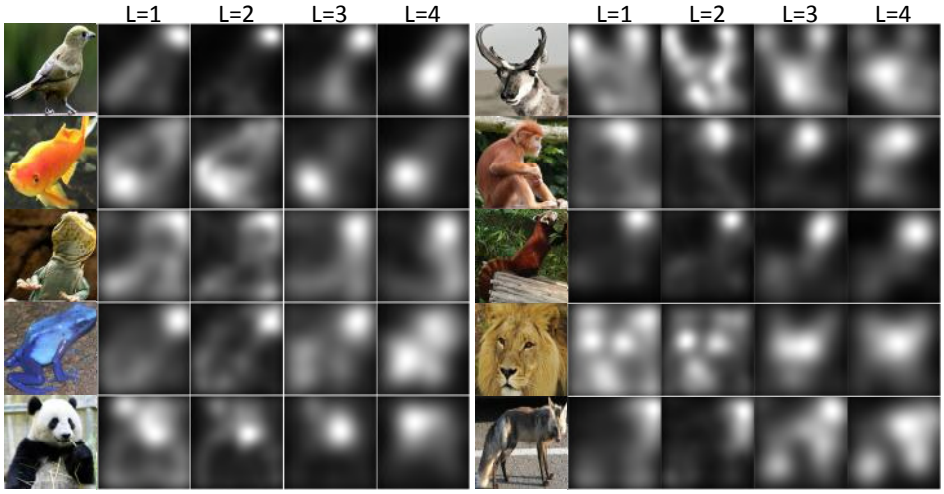

Figure 5: Heat maps of patterns [Zhang *et al.*, 2018a]. A heat map visualizes the spatial distribution of the top 50% patterns in the L-th layer of the explanatory graph with the highest inference scores.

the input image. To some extent, the node is robust to shape deformation and pose variations.

- Each edge encodes the co-activation relationship and the spatial relationship between two nodes in adjacent layers.
- We can regard an explanatory graph as a compression of feature maps of conv-layers. A CNN has multiple conv-layers. Each conv-layer may have hundreds of filters, and each filter may produce a feature map with hundreds of neural units. We can use tens of thousands of nodes in the explanatory graph to represent information contained in all tens of millions of neural units in these feature maps, *i.e.* by which part patterns the feature maps are activated, and where the part patterns are localized in input images.
- Just like a dictionary, each input image can only trigger a small subset of part patterns (nodes) in the explanatory graph. Each node describes a common part pattern with high transferability, which is shared by hundreds or thousands of training images.

Fig. 4 lists top-ranked image patches corresponding to different nodes in the explanatory graph. Fig. 5 visualizes the spatial distribution of object parts inferred by the top 50% nodes in the L-th layer of the explanatory graph with the highest inference scores. Fig. 6 shows object parts inferred by a single node.

Application: multi-shot part localization

There are many potential applications based on the explanatory graph. For example, we can regard the explanatory graph as a visual dictionary of a category and transfer graph nodes to other applications, such as multi-shot part localization.

Given very few bounding boxes of an object part, [Zhang *et al.*, 2018a] has proposed retrieving hundreds of nodes that are related to the part annotations from the explanatory graph, and then use the retrieved nodes to localize object parts in previously unseen images. Because each node in the explanatory graph encodes a part pattern shared by numerous training images, the retrieved nodes describe a general appearance of the target part without being over-fitted to the limited annotations of part bounding boxes. Given three annotations for each object part, the explanatory-graph-based method has exhibited superior performance of part localization and has decreased by about 1/3 localization errors *w.r.t.* the second-best baseline.

4.2 Disentangling CNN representations into decision trees

[Zhang *et al.*, 2018c] has further proposed a decision tree to encode decision modes in fully-connected layers. The decision tree is not designed for classification. Instead, the decision tree is used to quantitatively explain the logic for each CNN prediction. *I.e.* given an input image, we use the CNN to make a prediction. The decision tree tells people which fil-

Figure 6: Image regions inferred by each node in an explanatory graph [Zhang *et al.*, 2018a]. The method of [Zhang *et al.*, 2018a] successfully disentangles object-part patterns from representations of every single filter.

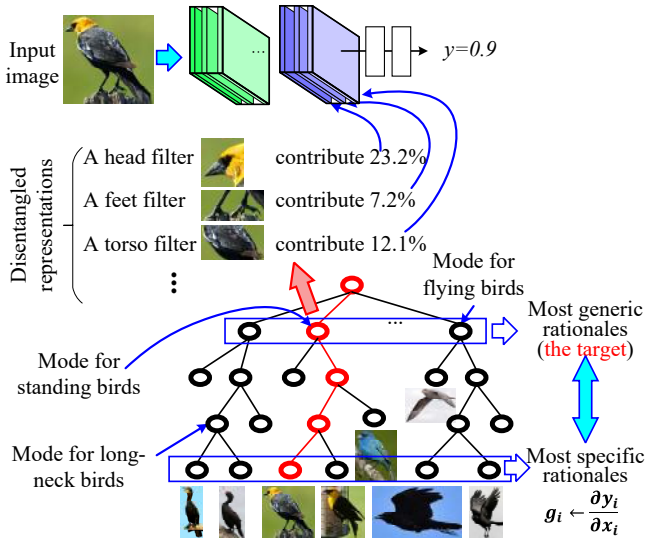

Figure 7: Decision tree that explains a CNN prediction at the semantic level [Zhang *et al.*, 2018c]. A CNN is learned for object classification with disentangled representations in the top conv-layer, where each filter represents a specific object part. The decision tree encodes various decision modes hidden inside fully-connected layers of the CNN in a coarse-to-fine manner. Given an input image, the decision tree infers a parse tree (red lines) to quantitatively analyze rationales for the CNN prediction, *i.e.* which object parts (or filters) are used for prediction and how much an object part (or filter) contributes to the prediction.

ters in a conv-layer are used for the prediction and how much they contribute to the prediction.

As shown in Fig. 7, the method mines potential decision modes memorized in fully-connected layers. The decision tree organizes these potential decision modes in a coarse-to-fine manner. Furthermore, this study uses the method of [Zhang *et al.*, 2017c] to disentangle representations of filters in the top conv-layers, *i.e.* making each filter represent a specific object part. In this way, people can use the decision tree to explain rationales for each CNN prediction at the semantic level, *i.e.* which object parts are used by the CNN to make the prediction.

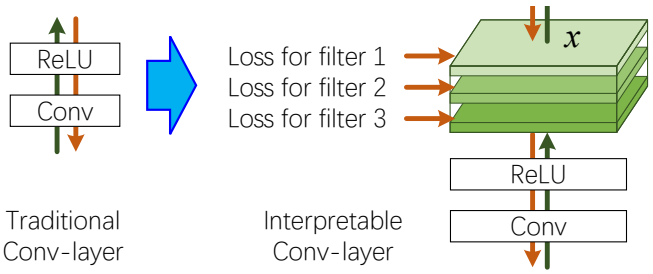

Figure 8: Structures of an ordinary conv-layer and an interpretable conv-layer [Zhang *et al.*, 2017c]. Green and red lines indicate the forward and backward propagations, respectively.

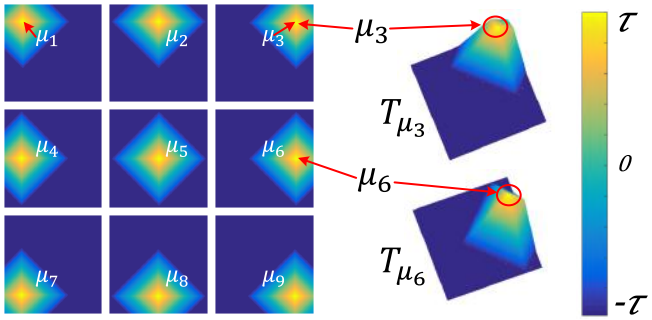

Figure 9: Templates [Zhang *et al.*, 2017c]. Each template T_{μ_i} matches to a feature map when the target part mainly triggers the i-th unit in the feature map.

5 Learning neural networks with interpretable/disentangled representations

Almost all methods mentioned in previous sections focus on the understanding of a pre-trained network. In this section, we review studies of learning disentangled representations of neural networks, where representations in middle layers are no longer a black box but have clear semantic meanings. Compared to the understanding of pre-trained networks, learning networks with disentangled representations present more challenges. Up to now, only a few studies have been published in this direction.

Figure 10: Visualization of interpretable filters in the top conv-layer [Zhang *et al.*, 2017c]. We used [Zhou *et al.*, 2015] to estimate the image-resolution receptive field of activations in a feature map to visualize a filter's semantics. An interpretable CNN usually encodes head patterns of animals in its top conv-layer for classification.

5.1 Interpretable convolutional neural networks

As shown in Fig. 8, [Zhang *et al.*, 2017c] has developed a method to modify an ordinary CNN to obtain disentangled representations in high conv-layers by adding a loss to each filter in the conv-layers. The loss is used to regularize the feature map towards the representation of a specific object part.

Note that people do not need to annotate any object parts or textures to supervise the learning of interpretable CNNs. Instead, the loss automatically assigns an object part to each filter during the end-to-end learning process. As shown in Fig. 9, this method designs some templates. Each template T_{μ_i} is a matrix with the same size of feature map. T_{μ_i} describes the ideal distribution of activations for the feature map when the target part mainly triggers the i-th unit in the feature map.

Given the joint probability of fitting a feature map to a template, the loss of a filter is formulated as the mutual information between the feature map and the templates. This loss encourages a low entropy of inter-category activations. *I.e.* each filter in the conv-layer is assigned to a certain category. If the input image belongs to the target category, then the loss expects the filter's feature map to match a template well; otherwise, the filter needs to remain inactivated. In addition, the loss also encourages a low entropy of spatial distributions of neural activations. *I.e.* when the input image belongs the target category, the feature map is supposed to exclusively fit a single template. In other words, the filter needs to activate a single location on the feature map.

This study assumes that if a filter repetitively activates various feature-map regions, then this filter is more likely to describe low-level textures (*e.g.* colors and edges), instead of high-level parts. For example, the left eye and the right eye may be represented by different filters, because contexts of the two eyes are symmetric, but not the same.

Fig.10 shows feature maps produced by different filters of an interpretable CNN. Each filter consistently represents the same object part through various images.

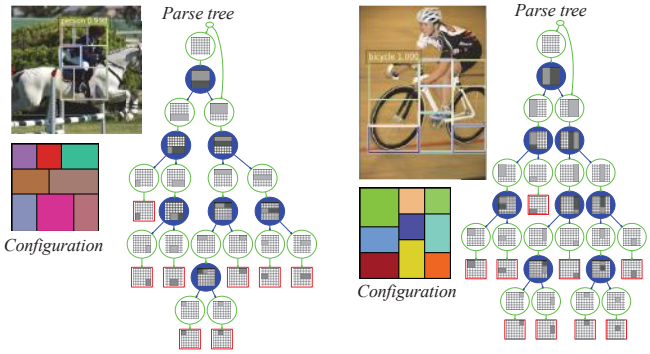

Figure 11: Detection examples of the proposed method [Wu *et al.*, 2017]. In addition to predicted bounding boxes, the method also outputs the latent parse tree and part configurations as the qualitatively extractive rationale in detection. The parse trees are inferred on-the-fly in the space of latent structures, which follow a top-down compositional grammar of an AOG.

5.2 Interpretable R-CNN

[Wu *et al.*, 2017] has proposed the learning of qualitatively interpretable models for object detection based on the R-CNN. The objective is to unfold latent configurations of object parts automatically during the object-detection process. This method is learned without using any part annotations for supervision. [Wu *et al.*, 2017] uses a top-down hierarchical and compositional grammar, namely an And-Or graph (AOG), to model latent configurations of object parts. This method uses an AOG-based parsing operator to substitute for the RoI-Pooling operator used in the R-CNN. The AOG-based parsing harnesses explainable compositional structures of objects and maintains the discrimination power of a R-CNN. This idea is related to the disentanglement of the local, bottom-up, and top-down information components for prediction [Wu *et al.*, 2007; Yang *et al.*, 2009; Wu and Zhu, 2011].

During the detection process, a bounding box is interpreted as the best parse tree derived from the AOG on-the-fly. During the learning process, a folding-unfolding method is used

to train the AOG and R-CNN in an end-to-end manner.

Fig. 11 illustrates an example of object detection. The proposed method detects object bounding boxes. The method also determines the latent parse tree and part configurations of objects as the qualitatively extractive rationale in detection.

5.3 Capsule networks

[Sabour *et al.*, 2017] has designed novel neural units, namely capsules, in order to substitute for traditional neural units to construct a capsule network. Each capsule outputs an activity vector instead of a scalar. The length of the activity vector represents the activation strength of the capsule, and the orientation of the activity vector encodes instantiation parameters. Active capsules in the lower layer send messages to capsules in the adjacent higher layer. This method uses an iterative routing-by-agreement mechanism to assign higher weights with the low-layer capsules whose outputs better fit the instantiation parameters of the high-layer capsule.

Experiments showed that when people trained capsule networks using the MNIST dataset [LeCun *et al.*, 1998b], a capsule encoded a specific semantic concept. Different dimensions of the activity vector of a capsule controlled different features, including 1) scale and thickness, 2) localized part, 3) stroke thickness, 3) localized skew, and 4) width and translation.

5.4 Information maximizing generative adversarial nets

The information maximizing generative adversarial net [Chen *et al.*, 2016], namely InfoGAN, is an extension of the generative adversarial network. The InfoGAN maximizes the mutual information between certain dimensions of the latent representation and the image observation. The InfoGAN separates input variables of the generator into two types, *i.e.* the incompressible noise z and the latent code c. This study aims to learn the latent code c to encode certain semantic concepts in an unsupervised manner.

The InfoGAN has been trained using the MNIST dataset [LeCun *et al.*, 1998b], the CelebA dataset [Liu *et al.*, 2015], the SVHN dataset [Netzer *et al.*, 2011], the 3D face dataset [Paysan *et al.*, 2009], and the 3D chair dataset [Aubry *et al.*, 2014]. Experiments have shown that the latent code has successfully encoded the digit type, the rotation, and the width of digits in the MNIST dataset, the lighting condition and the plate context in the SVHN dataset, the azimuth, the existence of glasses, the hairstyle, and the emotion in the CelebA dataset, and the width and 3D rotation in the 3D face and chair datasets.

6 Evaluation metrics for network interpretability

Evaluation metrics for model interpretability are crucial for the development of explainable models. This is because unlike traditional well-defined visual applications (*e.g.* object detection and segmentation), network interpretability is more difficult to define and evaluate. The evaluation metric of network interpretability can help people define the concept of network interpretability and guide the development of learning interpretable network representations. Up to now, only very few studies have discussed the evaluation of network interpretability. Proposing a promising evaluation metric is still a big challenge to state-of-the-art algorithms. In this section, we simply introduce two latest evaluation metrics for the interpretability of CNN filters, *i.e.* the filter interpretability proposed by [Bau *et al.*, 2017] and the location instability proposed by [Zhang *et al.*, 2018a].

6.1 Filter interpretability

[Bau *et al.*, 2017] has defined six types of semantics for CNN filters, *i.e. objects*, *parts*, *scenes*, *textures*, *materials*, and *colors*. The evaluation of filter interpretability requires people to annotate these six types of semantics on testing images at the pixel level. The evaluation metric measures the fitness between the image-resolution receptive field of a filter's neural activations[1] and the pixel-level semantic annotations on the image. For example, if the receptive field of a filter's neural activations usually highly overlaps with ground-truth image regions of a specific semantic concept through different images, then we can consider that the filter represents this semantic concept.

For each filter f, this method computes its feature maps $\mathbf{X} = \{x = f(I)|I \in \mathbf{I}\}$ on different testing images. Then, the distribution of activation scores in all positions of all feature maps is computed. [Bau *et al.*, 2017] set an activation threshold T_f such that $p(x_{ij} > T_f) = 0.005$, to select top activations from all spatial locations $[i, j]$ of all feature maps $x \in \mathbf{X}$ as valid map regions corresponding to f's semantics. Then, the method scales up low-resolution valid map regions to the image resolution, thereby obtaining the receptive field of valid activations on each image. We use S_f^I to denote the receptive field of f's valid activations *w.r.t.* the image I.

The compatibility between a filter f and a specific semantic concept is reported as an intersection-over-union score $IoU_{f,k}^I = \frac{\|S_f^I \cap S_k^I\|}{\|S_f^I \cup S_k^I\|}$, where S_k^I denotes the ground-truth mask of the k-th semantic concept on the image I. Given an image I, filter f is associated with the k-th concept if $IoU_{f,k}^I > 0.04$. The probability of the k-th concept being associated with the filter f is given as $P_{f,k} = \text{mean}_{I:\text{with k-th concept}} \mathbf{1}(IoU_{f,k}^I > 0.04)$. Thus, we can use $P_{f,k}$ to evaluate the filter interpretability of f.

6.2 Location instability

Another evaluation metric is location instability. This metric is proposed by [Zhang *et al.*, 2018a] to evaluate the fitness between a CNN filter and the representation of an object part. Given an input image I, the CNN computes a feature map $x \in \mathbb{R}^{N \times N}$ of filter f. We can regard the unit $x_{i,j}$ $(1 \le i, j \le N)$ with the highest activation as the location inference of f, where $N \times N$ is referred to as the size of the feature map. We use $\hat{\mathbf{p}}$ to denote the image position that corresponds to the inferred feature map location (i, j), *i.e.* the

[1]The method propagates the receptive field of each activated unit in a filter's feature map back to the image plane as the image-resolution receptive field of a filter.

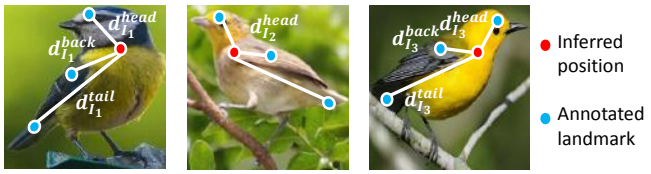

Figure 12: Notation for the computation of a filter's location instability [Zhang *et al.*, 2018a].

center of the unit $x_{i,j}$'s receptive field when we backward propagated the receptive field to the image plane. The evaluation assumes that if f consistently represented the same object part (the object part may not have an explicit name according to people's cognition) through different objects, then distances between the image position $\hat{\mathbf{p}}$ and some object landmarks should not change much among different objects. For example, if filter f represents the shoulder, then the distance between the shoulder and the head should remain stable through different objects.

Therefore, people can compute the deviation of the distance between the inferred position $\hat{\mathbf{p}}$ and a specific ground-truth landmark among different images. The average deviation *w.r.t.* various landmarks can be used to evaluate the location instability of f. As shown in Fig. 12, let $d_I(\mathbf{p}_k, \hat{\mathbf{p}}) = \frac{\|\mathbf{p}_k - \hat{\mathbf{p}}\|}{\sqrt{w^2+h^2}}$ denote the normalized distance between the inferred part and the k-th landmark $\mathbf{p}_k$ on image I. $\sqrt{w^2+h^2}$ denotes the diagonal length of the input image. Thus, $D_{f,k} = \sqrt{\text{var}_I[d_I(\mathbf{p}_k, \hat{\mathbf{p}})]}$ is reported as the relative location deviation of filter f *w.r.t.* the k-th landmark, where $\text{var}_I[d_I(\mathbf{p}_k, \hat{\mathbf{p}})]$ is referred to as the variation of the distance $d_I(\mathbf{p}_k, \hat{\mathbf{p}})$. Because each landmark cannot appear in all testing images, for each filter f, the metric only uses inference results with the top-M highest activation scores on images containing the k-th landmark to compute $D_{f,k}$. In this way, the average of relative location deviations of all the filters in a conv-layer *w.r.t.* all landmarks, *i.e.* $\text{mean}_f \text{mean}_{k=1}^{K} D_{f,k}$, measures the location instability of a CNN, where K denotes the number of landmarks.

7 Network interpretability for middle-to-end learning

Based on studies discussed in Sections 4 and 5, people may either disentangle representations of a pre-trained CNN or learn a new network with interpretable, disentangled representations. Such interpretable/disentangled network representations can further enable middle-to-end model learning at the semantic level without strong supervision. We briefly review two typical studies [Zhang *et al.*, 2017a; Zhang *et al.*, 2017b] of middle-to-end learning as follows.

7.1 Active question-answering for learning And-Or graphs

Based on the semantic And-Or representation proposed in [Zhang *et al.*, 2016], [Zhang *et al.*, 2017a] has developed a method to use active question-answering to semanticize neural patterns in conv-layers of a pre-trained CNN and build a model for hierarchical object understanding.

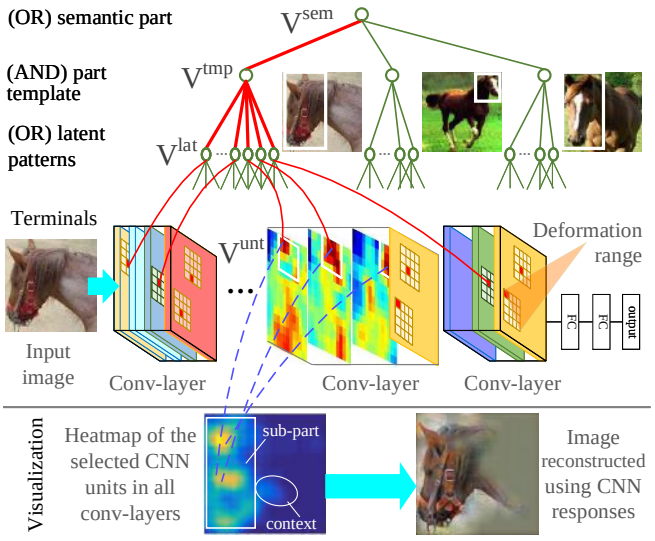

Figure 13: And-Or graph grown on a pre-trained CNN as a semantic branch [Zhang *et al.*, 2017a]. The AOG associates specific CNN units with certain image regions. The red lines indicate the parse graph.

As shown in Fig. 13, the CNN is pre-trained for object classification. The method aims to extract a four-layer interpretable And-Or graph (AOG) to explain the semantic hierarchy hidden in a CNN. The AOG encodes four-layer semantics, ranging across the *semantic part* (OR node), *part templates* (AND nodes), *latent patterns* (OR nodes), and *neural units* (terminal nodes) on feature maps. In the AOG, AND nodes represent compositional regions of a part, and OR nodes encode a list of alternative template/deformation candidates for a local part. The top part node (OR node) uses its children to represent some template candidates for the part. Each part template (AND node) in the second layer uses children latent patterns to represent its constituent regions. Each latent pattern in the third layer (OR node) naturally corresponds to a certain range of units within the feature map of a filter. The latent pattern selects a unit within this range to account for its geometric deformation.

To learn an AOG, [Zhang *et al.*, 2017a] allows the computer to actively identify and ask about objects, whose neural patterns cannot be explained by the current AOG. As shown in Fig. 14, in each step of the active question-answering, the current AOG is used to localize object parts among all the unannotated images. The method actively selects objects that cannot well fit the AOG, namely unexplained objects. The method predicts the potential gain of asking about each unexplained object, and thus determines the best sequence of questions (*e.g.* asking about template types and bounding boxes of unexplained object parts). In this way, the method uses the answers to either refine an existing part template or mine latent patterns for new object-part templates, to grow AOG branches. Fig. 15 compares the part-localization performance of different methods. The QA-based learning exhibits significantly higher efficiency than other baselines. The proposed method uses about 1/6–1/3 of the part annotations for training, but achieves similar or better part-localization per-

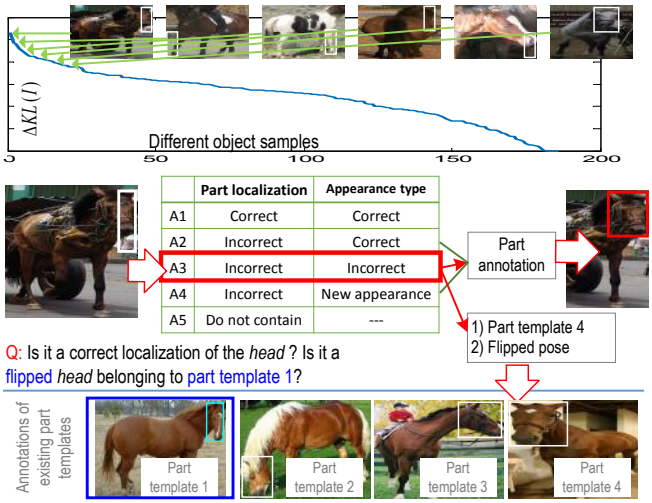

Figure 14: Illustration of the QA process [Zhang *et al.*, 2017a]. (top) The method sorts and selects unexplained objects. (bottom) Questions for each target object.

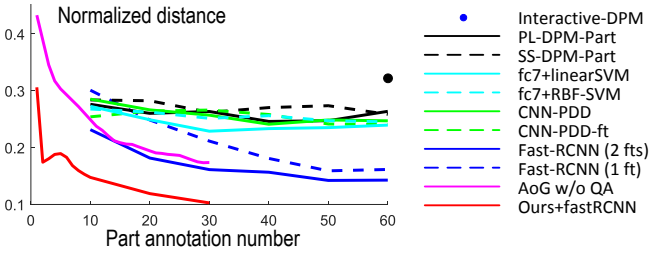

Figure 15: Part localization performance on the Pascal VOC Part dataset [Zhang *et al.*, 2017a].

formance than fast-RCNN methods.

7.2 Interactive manipulations of CNN patterns

Let a CNN be pre-trained using annotations of object bounding boxes for object classification. [Zhang *et al.*, 2017b] has explored an interactive method to diagnose knowledge representations of a CNN, in order to transfer CNN patterns to model object parts. Unlike traditional end-to-end learning of CNNs that requires numerous training samples, this method mines object part patterns from the CNN in the scenario of one/multi-shot learning.

More specifically, the method uses part annotations on very few (e.g. three) object images for supervision. Given a bounding-box annotation of a part, the proposed method first uses [Zhang *et al.*, 2016] to mine latent patterns, which are related to the annotated part, from conv-layers of the CNN. An AOG is used to organize all mined patterns as the representation of the target part. The method visualizes the mined latent patterns and asks people to remove latent patterns unrelated to the target part interactively. In this way, people can simply prune incorrect latent patterns from AOG branches to refine the AOG. Fig. 16 visualizes initially mined patterns and the remaining patterns after human interaction. With the guidance of human interactions, [Zhang *et al.*, 2017b] has exhibited superior performance of part localization.

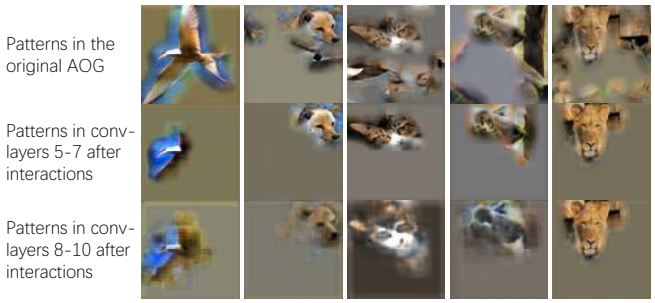

Figure 16: Visualization of patterns for the head part before and after human interactions [Zhang *et al.*, 2017b].

8 Prospective trends and conclusions

In this paper, we have reviewed several research directions within the scope of network interpretability. Visualization of a neural unit's patterns was the starting point of understanding network representations in the early years. Then, people gradually developed methods to analyze feature spaces of neural networks and diagnose potential representation flaws hidden inside neural networks. At present, disentangling chaotic representations of conv-layers into graphical models and/or symbolic logic has become an emerging research direction to open the black-box of neural networks. The approach for transforming a pre-trained CNN into an explanatory graph has been proposed and has exhibited significant efficiency in knowledge transfer and weakly-supervised learning.

End-to-end learning interpretable neural networks, whose intermediate layers encode comprehensible patterns, is also a prospective trend. Interpretable CNNs have been developed, where each filter in high conv-layers represents a specific object part.

Furthermore, based on interpretable representations of CNN patterns, semantic-level middle-to-end learning has been proposed to speed up the learning process. Compared to traditional end-to-end learning, middle-to-end learning allows human interactions to guide the learning process and can be applied with very few annotations for supervision.

In the future, we believe the middle-to-end learning will continuously be a fundamental research direction. In addition, based on the semantic hierarchy of an interpretable network, debugging CNN representations at the semantic level will create new visual applications.

Acknowledgement

This work is supported by ONR MURI project N00014-16-1-2007 and DARPA XAI Award N66001-17-2-4029, and NSF IIS 1423305.

References

[Aubry and Russell, 2015] Mathieu Aubry and Bryan C. Russell. Understanding deep features with computer-generated imagery. *In ICCV*, 2015.

[Aubry *et al.*, 2014] M. Aubry, D. Maturana, A. Efros, B. Russell, and J. Sivic. Seeing 3d chairs: Exemplar part-

based 2d-3d alignment using a large dataset of cad models. *In CVPR*, 2014.

[Bau *et al.*, 2017] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. *In CVPR*, 2017.

[Chen *et al.*, 2016] Xi Chen, Yan Duan, Rein Houthooft, John Schulman, Ilya Sutskever, and Pieter Abbeel. Infogan: Interpretable representation learning by information maximizing generative adversarial nets. *In NIPS*, 2016.

[Dosovitskiy and Brox, 2016] Alexey Dosovitskiy and Thomas Brox. Inverting visual representations with convolutional networks. *In CVPR*, 2016.

[Fong and Vedaldi, 2017] Ruth C. Fong and Andrea Vedaldi. Interpretable explanations of black boxes by meaningful perturbation. *In ICCV*, 2017.

[Goyal *et al.*, 2016] Yash Goyal, Akrit Mohapatra, Devi Parikh, and Dhruv Batra. Towards transparent ai systems: Interpreting visual question answering models. *In arXiv:1608.08974*, 2016.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *In CVPR*, 2016.

[Hu *et al.*, 2016] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric P. Xing. Harnessing deep neural networks with logic rules. *In ACL*, 2016.

[Huang *et al.*, 2017] Gao Huang, Zhuang Liu, Kilian Q. Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *In CVPR*, 2017.

[Kindermans *et al.*, 2017] Pieter-Jan Kindermans, Kristof T. Schütt, Maximilian Alber, Klaus-Robert Müller, Dumitru Erhan, Been Kim, and Sven Dähne. Learning how to explain neural networks: Patternnet and patternattribution. *arXiv: 1705.05598*, 2017.

[Koh and Liang, 2017] PangWei Koh and Percy Liang. Understanding black-box predictions via influence functions. *In ICML*, 2017.

[Krizhevsky *et al.*, 2012] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. *In NIPS*, 2012.

[Kumar *et al.*, 2017] Devinder Kumar, Alexander Wong, and Graham W. Taylor. Explaining the unexplained: A class-enhanced attentive response (clear) approach to understanding deep neural networks. *In CVPR Workshop on Explainable Computer Vision and Job Candidate Screening Competition*, 2017.

[Lakkaraju *et al.*, 2017] Himabindu Lakkaraju, Ece Kamar, Rich Caruana, and Eric Horvitz. Identifying unknown unknowns in the open world: Representations and policies for guided exploration. *In AAAI*, 2017.

[LeCun *et al.*, 1998a] Yann LeCun, Lèon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *In Proceedings of the IEEE*, 1998.

[LeCun *et al.*, 1998b] Yann LeCun, Corinna Cortes, and Christopher JC Burges. The mnist database of handwritten digits. Technical report, 1998.

[Liu *et al.*, 2015] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. *In ICCV*, 2015.

[Lu, 2015] Yao Lu. Unsupervised learning on neural network outputs. *In arXiv:1506.00990v9*, 2015.

[Mahendran and Vedaldi, 2015] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. *In CVPR*, 2015.

[Netzer *et al.*, 2011] Yuval Netzer, Tao Wang, Adam Coates, Alessandro Bissacco, Bo Wu, and Andrew Y. Ng. Reading digits in natural images with unsupervised feature learning. *In NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[Nguyen *et al.*, 2017] Anh Nguyen, Jeff Clune, Yoshua Bengio, Alexey Dosovitskiy, and Jason Yosinski. Plug & play generative networks: Conditional iterative generation of images in latent space. *CVPR*, 2017.

[Olah *et al.*, 2017] Chris Olah, Alexander Mordvintsev, and Ludwig Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[Paysan *et al.*, 2009] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. *In AVSS*, 2009.

[Ribeiro *et al.*, 2016] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. "why should i trust you?" explaining the predictions of any classifier. *In KDD*, 2016.

[Sabour *et al.*, 2017] Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. *In NIPS*, 2017.

[Selvaraju *et al.*, 2017] 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*, 2017.

[Simonyan *et al.*, 2013] Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep inside convolutional networks: visualising image classification models and saliency maps. *In arXiv:1312.6034*, 2013.

[Springenberg *et al.*, 2015] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: the all convolutional net. *ICLR workshop*, 2015.

[Su *et al.*, 2017] Jiawei Su, Danilo Vasconcellos Vargas, and Sakurai Kouichi. One pixel attack for fooling deep neural networks. *In arXiv:1710.08864*, 2017.

[Szegedy *et al.*, 2014] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *In arXiv:1312.6199*, 2014.

[Wang *et al.*, 2017] Peng Wang, Qi Wu, Chunhua Shen, and Anton van den Hengel. The vqa-machine: Learning how

to use existing vision algorithms to answer new questions. *In CVPR*, 2017.

[Wu and Zhu, 2011] Tianfu Wu and Song-Chun Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *International journal of computer vision*, 93(2):226–252, 2011.

[Wu *et al.*, 2007] Tian-Fu Wu, Gui-Song Xia, and Song-Chun Zhu. Compositional boosting for computing hierarchical image structures. *In CVPR*, 2007.

[Wu *et al.*, 2017] Tianfu Wu, Xilai Li, Xi Song, Wei Sun, Liang Dong, and Bo Li. Interpretable r-cnn. *In arXiv:1711.05226*, 2017.

[Yang *et al.*, 2009] Xiong Yang, Tianfu Wu, and Song-Chun Zhu. Evaluating information contributions of bottom-up and top-down processes. *ICCV*, 2009.

[Yosinski *et al.*, 2014] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *In NIPS*, 2014.

[Zeiler and Fergus, 2014] Matthew D. Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. *In ECCV*, 2014.

[Zhang *et al.*, 2016] Q. Zhang, R. Cao, Y. N. Wu, and S.-C. Zhu. Growing interpretable part graphs on convnets via multi-shot learning. *In AAAI*, 2016.

[Zhang *et al.*, 2017a] Quanshi Zhang, Ruiming Cao, Ying Nian Wu, and Song-Chun Zhu. Mining object parts from cnns via active question-answering. *In CVPR*, 2017.

[Zhang *et al.*, 2017b] Quanshi Zhang, Ruiming Cao, Shengming Zhang, Mark Edmonds, Ying Nian Wu, and Song-Chun Zhu. Interactively transferring cnn patterns for part localization. *In arXiv:1708.01783*, 2017.

[Zhang *et al.*, 2017c] Quanshi Zhang, Ying Nian Wu, and Song-Chun Zhu. Interpretable convolutional neural network. *In arXiv:1710.00935*, 2017.

[Zhang *et al.*, 2018a] Q. Zhang, R. Cao, F. Shi, Y.N. Wu, and S.-C. Zhu. Interpreting cnn knowledge via an explanatory graph. *In AAAI*, 2018.

[Zhang *et al.*, 2018b] Q. Zhang, W. Wang, and S.-C. Zhu. Examining cnn representations with respect to dataset bias. *In AAAI*, 2018.

[Zhang *et al.*, 2018c] Quanshi Zhang, Yu Yang, Ying Nian Wu, and Song-Chun Zhu. Interpreting cnns via decision trees. *arXiv:1802.00121*, 2018.

[Zhou *et al.*, 2015] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. *In ICRL*, 2015.

[Zintgraf *et al.*, 2017] Luisa M Zintgraf, Taco S Cohen Tameem Adel, and Max Welling. Visualizing deep neural network decisions: prediction difference analysis. *ICLR*, 2017.