Frontiers of Information Technology & Electronic Engineering
www.jzus.zju.edu.cn; engineering.cae.cn; www.springerlink.com
ISSN 2095-9184 (print); ISSN 2095-9230 (online)
E-mail: jzus@zju.edu.cn

Review:

Visual interpretability for deep learning: a survey*

Quan-shi ZHANG‡, Song-chun ZHU

University of California, Los Angeles, California 90095, USA

E-mail: zhangqs@ucla.edu; sczhu@stat.ucla.edu

Received Dec. 2, 2017; Revision accepted Jan. 25, 2018; Crosschecked Jan. 28, 2018

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, interpretability is always Achilles' heel of deep neural networks. At present, deep neural networks obtain high discrimination power at the cost of a low interpretability of their black-box representations. We believe that high model interpretability may help people break several bottlenecks of deep learning, e.g., learning from a few annotations, learning via human–computer communications at the semantic level, and semantically debugging network representations. We focus on convolutional neural networks (CNNs), and 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.

Key words: Artificial intelligence; Deep learning; Interpretable model
https://doi.org/10.1631/FITEE.1700808 **CLC number:** TP391

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 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 five research directions:

1. Visualization of CNN representations in intermediate network layers. These methods either synthesize mainly 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 discussions.

2. 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.

3. Disentanglement of 'the mixture of patterns' encoded in each filter of CNNs. These studies disentangle mainly complex representations in conv-layers

‡ Corresponding author

* Project supported by the ONR MURI project (No. N00014-16-1-2007), the DARPA XAI Award (No. N66001-17-2-4029), and NSF IIS (No. 1423305)

ORCID: Quan-shi ZHANG, http://orcid.org/0000-0002-6108-2738

and transform network representations into interpretable graphs. Please see Section 4 for details.

4. Building explainable models. We discuss interpretable CNNs (Zhang et al., 2018d), capsule networks (Sabour et al., 2017), interpretable R-CNNs (Wu et al., 2017), and InfoGAN (Chen et al., 2016) in Section 5.

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 a weak supervision. Section 7 introduces methods to learn new models via human–computer interactions (Zhang et al., 2017b) and active question-answering with a 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 more challenges to the state-of-the-art algorithms. Finally, explainable or disentangled network representations are also the starting point for weakly-supervised middle-to-end learning.

The clear semantics in high conv-layers can help people trust a network's prediction. As discussed in Zhang et al. (2018a), considering dataset and representation bias, a high accuracy on testing images still cannot ensure that a CNN will encode correct representations. 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 significantly reduce the requirements 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.

2 Visualization of convolutional neural network representations

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

First, gradient-based methods (Simonyan et al., 2013; Zeiler and Fergus, 2014; Mahendran and Vedaldi, 2015; Springenberg et al., 2015) are the mainstream of network visualization. These methods compute mainly 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) 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 the up-convolutional net as a tool that indirectly illustrates the image appearance corresponding to a feature map, although compared to gradient-based methods, the up-convolutional net cannot ensure mathematically that the visualization results exactly reflect actual representations in CNN. Similarly, Nguyen et al. (2017) 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) 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 understand the representation of a filter.

3 Diagnosis of convolutional neural network representations

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

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

2. The second research direction extracts image regions that directly contribute to the network output for a label/attribute to explain CNN representations of the label/attribute. This is similar to the visualization of CNNs. Fong and Vedaldi (2017) and Selvaraju et al. (2017) proposed methods 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 by Ribeiro et al. (2016) extracts image regions that are highly sensitive to the network output. Zintgraf et al. (2017), Kindermans et al. (2017), and Kumar et al. (2017) invented methods to visualize areas in the input image that contribute the most to the decision-making process of CNN. Wang et al. (2017) and Goyal et al. (2016) tried to interpret the logic for visual question-answering encoded in neural networks. These studies have listed important objects (or regions of interests) detected from the images and crucial words in questions as the explanation of output answers.

3. The estimation of vulnerable points in the feature space of a CNN is also a popular direction for diagnosing network representations. Approaches proposed by Su et al. (2017), Koh and Liang (2017), and Szegedy et al. (2014) were 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 by 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.

4. 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) proposed a method to discover knowledge blind spots (unknown patterns) of CNN in a weakly-supervised manner. This method groups all sample points in the entire feature space of a CNN into thousands of pseudo-categories. It assumes 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) proposed a method of 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, to obtain more meaningful network representations.

5. Finally, Zhang et al. (2018a) presented a method to discover potential, biased representations of a CNN. Fig. 1 shows biased representations of a CNN trained to estimate face attributes. When an attribute usually co-appears with specific visual features in training images, CNN may use such co-appearing features to represent the attribute. When the co-appearing features used are not semantically related to the target attribute, these features can be considered as biased representations.

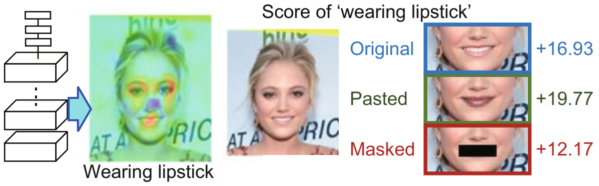

Fig. 1 Biased representations in a convolutional neural network (Zhang et al., 2018a)

Considering potential dataset bias, a high accuracy on testing images cannot always ensure that a convolutional neural network (CNN) learns correct representations. CNN may use unreliable co-appearing contexts to make predictions. For example, people may modify mouth appearances of two faces manually by masking mouth regions or pasting another mouth; however, such modifications do not significantly change prediction scores for the 'lipstick' attribute. Fig. 1 shows the heat maps of inference patterns of the 'lipstick' attribute, where red/blue patterns are positive/negative with the attribute score. CNN mistakenly considers unrelated patterns as contexts to infer the lipstick. References to color refer to the online version of this figure

Given a pre-trained CNN (e.g., a CNN that was trained to estimate face attributes), Zhang et al. (2018a) required people 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 mines inference patterns of each attribute output from conv-layers, and uses inference patterns to compute actual attribute relationships encoded in CNN. Conflicts between the ground-truth and the mined attribute relationships indicate biased representations.

4 Disentangling convolutional neural network representations into explanatory graphs and decision trees

4.1 Disentangling convolutional neural network representations into explanatory graphs

Compared with the visualization and diagnosis of network representations in Sections 2 and 3, disentangling CNN features into human-interpretable graphical representations (namely 'explanatory graphs') provides a more thorough explanation of network representations. Zhang et al. (2016, 2018b) proposed disentangling features in conv-layers of a pre-trained CNN and 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 tail parts of an object. Thus, to provide a global view of how visual knowledge is organized in a pre-trained CNN, Zhang et al. (2016, 2018b) aimed to answer the following three questions:

1. How many types of visual patterns are memorized by each convolutional filter of CNN (here, a visual pattern may describe a specific object part or a certain texture)?

2. Which patterns are co-activated to describe an object part?

3. What is the spatial relationship between two co-activated patterns?

As shown in Fig. 3, the explanatory graph explains the knowledge semantic hidden inside 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:

1. The explanatory graph has multiple layers. Each graph layer corresponds to a specific conv-layer of a CNN.

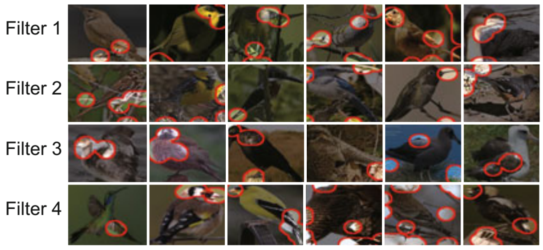

Fig. 2 Feature maps of a filter obtained using different input images (Zhang et al., 2018b)

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. References to color refer to the online version of this figure

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

3. Each edge encodes the co-activation and spatial relationships between two nodes in adjacent layers.

4. 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.

5. Just like a dictionary, each input image can trigger only a small subset of part patterns (nodes) in the explanatory graph. Each node describes a common part pattern with a 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.

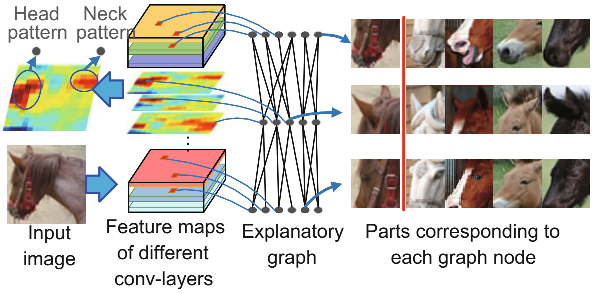
Head pattern
Neck pattern
Input image
Feature maps of different conv-layers
Explanatory graph
Parts corresponding to each graph node

Fig. 3 Explanatory graph (Zhang et al., 2018b)

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. (2018b) disentangles part patterns from each filter in an unsupervised manner, thereby clarifying the knowledge representation. References to color refer to the online version of this figure

4.1.1 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 a few bounding boxes of an object part, Zhang et al. (2018b) proposed a method of retrieving hundreds of nodes that are related to part annotations from the explanatory graph, and then using 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 exhibits superior performance of part localization and decreases by about 1/3 localization errors w.r.t. the second best baseline.

4.2 Disentangling convolutional neural network representations into decision trees

Zhang et al. (2018c) further proposed a decision tree to encode decision modes in fully connected layers. The decision tree is not designed for classification. Instead, it is used to quantitatively explain

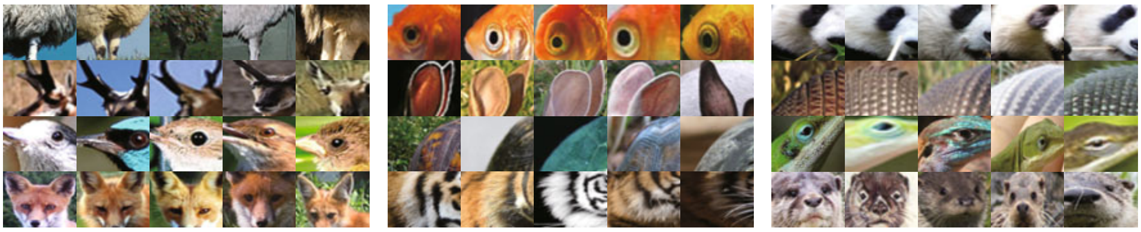

Fig. 4 Image patches corresponding to different nodes in the explanatory graph (Zhang et al., 2018b)

References to color refer to the online version of this figure

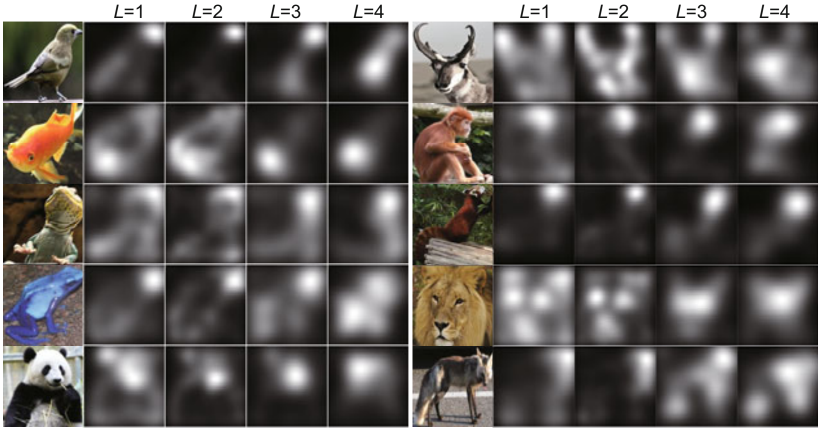

Fig. 5 Heat maps of patterns (Zhang et al., 2018b)

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. References to color refer to the online version of this figure

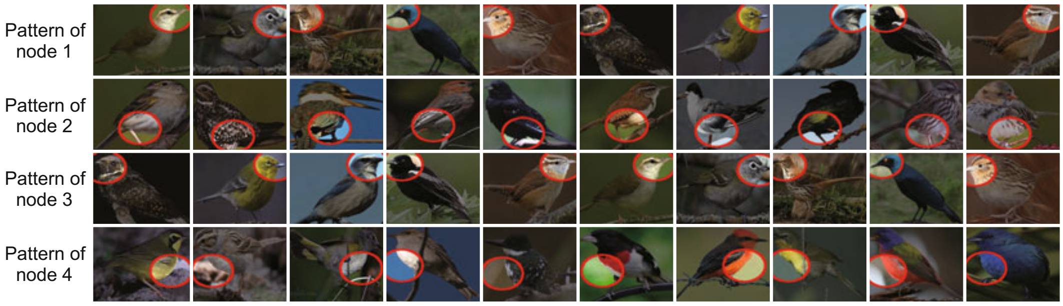

Fig. 6 Image regions inferred by each node in an explanatory graph (Zhang et al., 2018b)

The method proposed by Zhang et al. (2018b) successfully disentangles object-part patterns from representations of every single filter. References to color refer to the online version of this figure

the logic for each CNN prediction; i.e., given an input image, we use CNN to make a prediction. The decision tree tells people which filters 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 proposed by Zhang et al. (2018d) 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 CNN to make the prediction.

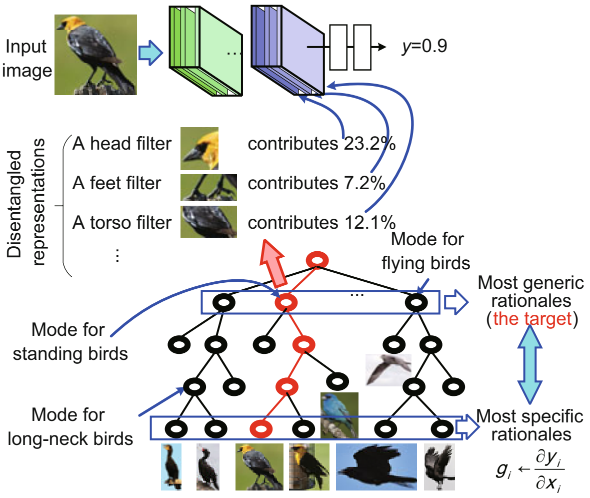

Fig. 7 Decision tree that explains a convolutional neural network (CNN) prediction at the semantic level (Zhang et al., 2018c)

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 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. References to color refer to the online version of this figure

5 Learning neural networks with interpretable/disentangled representations

Almost all methods mentioned in Sections 2–4 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 with 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.

5.1 Interpretable convolutional neural networks

As shown in Fig. 8, Zhang et al. (2018d) 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 $\boldsymbol{T}_{\mu_i}$

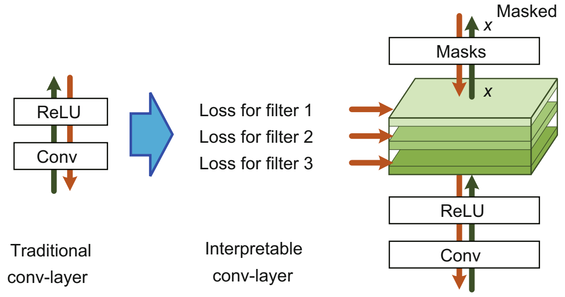

Fig. 8 Structures of an ordinary conv-layer and an interpretable conv-layer (Zhang et al., 2018d)

Green and red lines indicate the forward and backward propagations, respectively. References to color refer to the online version of this figure

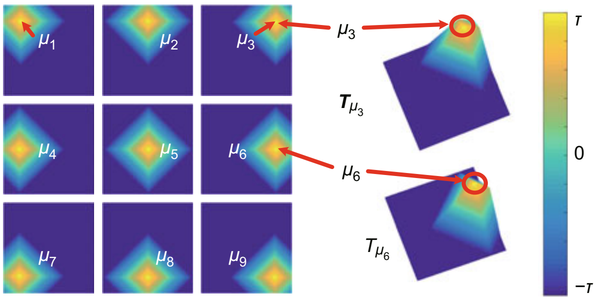

Fig. 9 Templates designed by Zhang et al. (2018d)

Each template $\boldsymbol{T}_{\mu_i}$ matches a feature map when the target part triggers mainly the i^{th} unit in the feature map. References to color refer to the online version of this figure

is a matrix with the same size of feature map. $\boldsymbol{T}_{\mu_i}$ describes the ideal distribution of activations for the feature map when the target part triggers mainly 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 encourages a low entropy of spatial distributions of neural activations; i.e., when the input image belongs to 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.

Zhang et al. (2018d) assumed 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 with different filters of an interpretable CNN. Each filter consistently represents the same object part through various images.

5.2 Interpretable region-based convolutional neural networks

Wu et al. (2017) proposed the learning of qualitatively interpretable models for object detection based on the region-based convolutional neural network (R-CNN) 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) used 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 R-CNN. The AOG-based parsing harnesses explainable compositional structures of objects and maintains the discrimination power of an 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 AOG on the fly. During the learning process, a folding-unfolding method is used to train AOG and R-CNN in an end-to-end manner.

Fig. 11 illustrates an example of object detection proposed by Zhang et al. (2018d). This method detects object bounding boxes. It 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) designed novel neural units, namely 'capsules', 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.

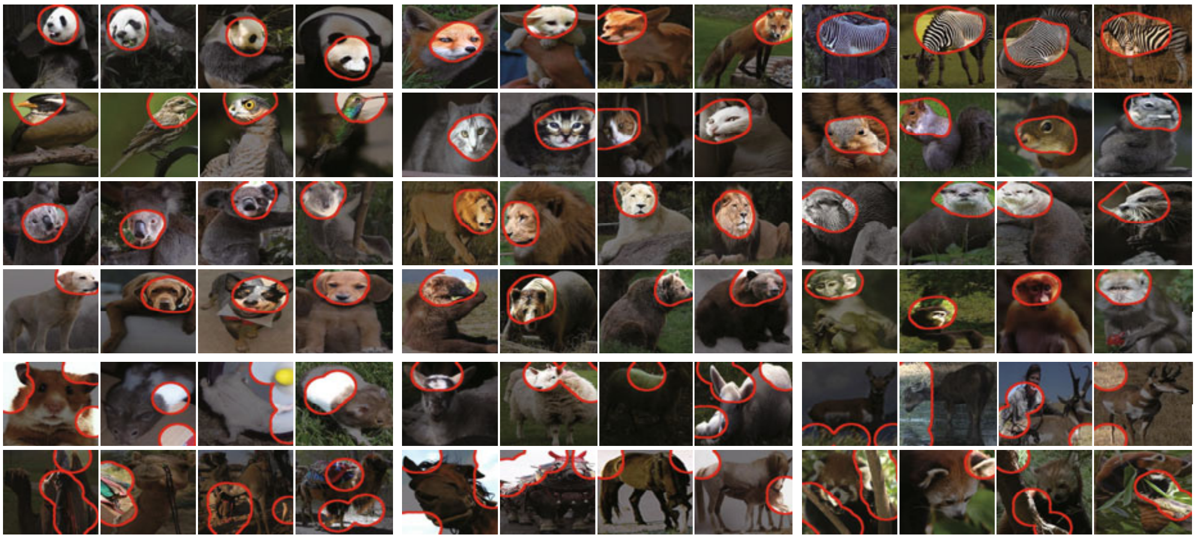

Fig. 10 Visualization of interpretable filters in the top conv-layer (Zhang et al., 2018d)

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. References to color refer to the online version of this figure

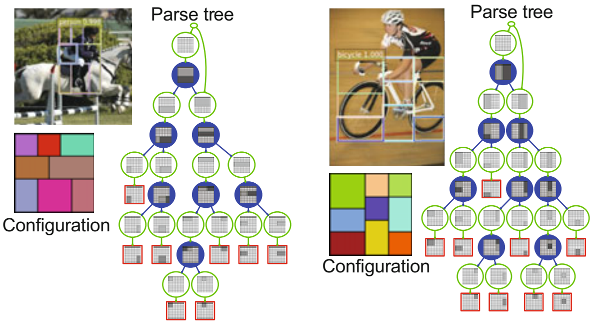

Fig. 11 Detection examples of the method proposed by Wu et al. (2017)

In addition to predicted bounding boxes, the method 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 And-Or graph (AOG)

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 train 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, (4) localized skew, and (5) 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. InfoGAN maximizes the mutual information between certain dimensions of the latent representation and the image observation. InfoGAN separates input variables of the generator into two types, i.e., incompressible noise z and latent code c. This study aims to learn latent code c to encode certain semantic concepts in an unsupervised manner.

InfoGAN was 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 successfully encodes the digit type, rotation, and width of digits in the MNIST dataset, the lighting condition and plate context in the SVHN dataset, the azimuth, existence of glasses, hairstyle, emotion in the CelebA dataset, and 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 a 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. (2018b).

6.1 Filter interpretability

Bau et al. (2017) defined six types of semantics for CNN filters, i.e., 'objects', 'parts', 'scenes', 'textures', 'materials', and 'colors'. The evaluation of filter interpretability requires people 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 (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) and the pixel-level semantic annotations on the image. For example, if the receptive field of a filter's neural activations usually overlaps highly 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 $\boldsymbol{X} = \{x = f(I) | I \in \boldsymbol{I}\}$ 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 $[i, j]$'s of all feature maps $x \in \boldsymbol{X}$ 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. image I.

The compatibility between a filter f and a specific semantic concept is reported as an intersection-over-union (IoU) score $\mathrm{IoU}_{f,k}^I = \frac{\|S_f^I \cap S_k^I\|}{\|S_f^I \cup S_k^I\|}$, where S_k^I denotes the ground-truth mask of the k^{th} semantic concept on image I. Given an image I, filter f is associated with the k^{th} concept if $\mathrm{IoU}_{f,k}^I > 0.04$. The probability of the k^{th} concept being associated with filter f is given as $P_{f,k} = \mathrm{mean}_{I:\text{with } k^{\text{th}} \text{ concept}} \mathbf{1}(\mathrm{IoU}_{f,k}^I > 0.04)$. 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 was proposed by Zhang et al. (2018b) to evaluate the fitness between a CNN filter and the representation of an object part. Given an input image I, CNN computes a feature map $x \in \mathbb{R}^{N \times N}$ of filter f. We can regard 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 the size of the feature map. We use $\hat{\boldsymbol{p}}$ to denote the image position that corresponds to the inferred feature map location (i, j), i.e., the center of 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 represents 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 $\hat{\boldsymbol{p}}$ 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 $\hat{\boldsymbol{p}}$ 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 $d_I(\boldsymbol{p}_k, \hat{\boldsymbol{p}}) = \frac{\|\boldsymbol{p}_k - \hat{\boldsymbol{p}}\|}{\sqrt{w^2 + h^2}}$ denote the normalized distance between the inferred part and the k^{th} landmark $\boldsymbol{p}_k$ on image I, and $\sqrt{w^2 + h^2}$ denotes the diagonal length of the input image. Thus, $D_{f,k} = \sqrt{\mathrm{var}_I[d_I(\boldsymbol{p}_k, \hat{\boldsymbol{p}})]}$ is reported as the relative location deviation of filter f w.r.t. the k^{th} landmark, where $\text{var}_I[d_I(\boldsymbol{p}_k, \hat{\boldsymbol{p}})]$ is referred to as the variation of distance $d_I(\boldsymbol{p}_k, \hat{\boldsymbol{p}})$. Because each landmark cannot appear in all testing images, for each filter f, the metric uses only 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.

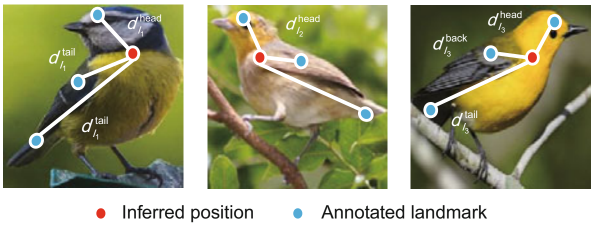

Fig. 12 Notation for the computation of a filter's location instability (Zhang et al., 2018b)

References to color refer to the online version of this figure

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,b) 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 by Zhang et al. (2016), Zhang et al. (2017a) developed a method to use active question-answering to semanticize neural patterns in conv-layers of a pre-trained CNN and built a model for hierarchical object understanding.

As shown in Fig. 13, CNN is pre-trained for object classification. The method aims to extract a four-layer interpretable 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 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 in the second layer (AND node) 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.

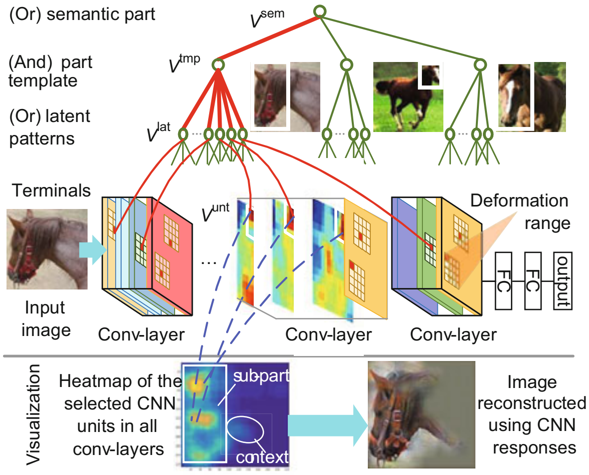

Fig. 13 And-Or graph (AOG) grown on a pre-trained convolutional neural network (CNN) as a semantic branch (Zhang et al., 2017a)

AOG associates specific CNN units with certain image regions. Red lines indicate the parse graph. References to color refer to the online version of this figure

To learn an AOG, Zhang et al. (2017a) allowed 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 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 a 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 performance compared with fast-RCNN methods.

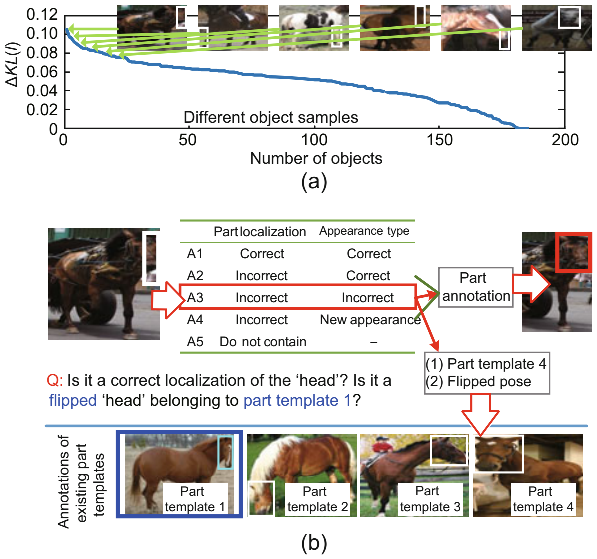

Fig. 14 Illustration of the question-answering (QA) process (Zhang et al., 2017a): (a) method of sorting and selecting unexplained objects; (b) questions for each target object

In (a), ΔKL indicates the predicted information gain of the And-Or graph (AOG) model obtained from asking about different objects, and the horizontal axis indicates different objects sorted w.r.t. the predicted information gain

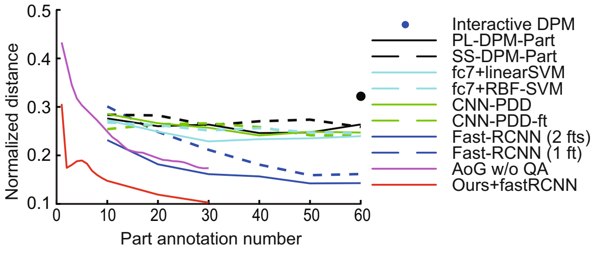

Fig. 15 Part localization performance on the Pascal VOC Part dataset (Zhang et al., 2017a)

References to color refer to the online version of this figure

7.2 Interactive manipulations of convolutional neural network patterns

Let a CNN be pre-trained using annotations of object bounding boxes for object classification. Zhang et al. (2017b) explored an interactive method to diagnose knowledge representations of a CNN, 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 CNN in the scenario of one/multi-shot learning.

Specifically, the method uses part annotations on a few (e.g., three) object images for supervision. Given a bounding-box annotation of a part, the proposed method first uses the method proposed by Zhang et al. (2016) to mine latent patterns, which are related to the annotated part, from conv-layers of 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 AOG. Fig. 16 visualizes initially mined patterns and the remaining patterns after human interaction. With guidance of human interactions, Zhang et al. (2017b) exhibited a superior performance of part localization.

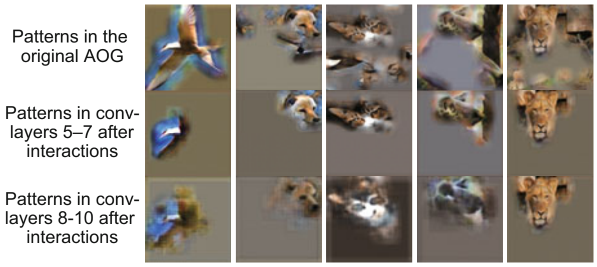
Patterns in the original AOG
Patterns in conv-layers 5–7 after interactions
Patterns in conv-layers 8-10 after interactions

Fig. 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 have 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 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 was proposed. It exhibited a significant efficiency in knowledge transfer and weakly-supervised learning.

End-to-end learning of 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 was proposed to speed up the learning process. Compared with traditional end-to-end learning, middle-to-end learning allows human interactions to guide the learning process and can be applied with a few annotations for supervision.

In the future, we believe that 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.

References

Aubry M, Russell BC, 2015. Understanding deep features with computer-generated imagery. IEEE Int Conf on Computer Vision, p.2875-2883. https://doi.org/10.1109/ICCV.2015.329

Aubry M, Maturana D, Efros A, et al., 2014. Seeing 3D chairs: exemplar part-based 2D–3D alignment using a large dataset of CAD models. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.3762-3769.

Bau D, Zhou B, Khosla A, et al., 2017. Network dissection: quantifying interpretability of deep visual representations. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1063-6919. https://doi.org/10.1109/CVPR.2017.354

Chen X, Duan Y, Houthooft R, et al., 2016. Infogan: interpretable representation learning by information maximizing generative adversarial nets. NIPS, p.2172-2180.

Dosovitskiy A, Brox T, 2016. Inverting visual representations with convolutional networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4829-4837.

Fong RC, Vedaldi A, 2017. Interpretable explanations of black boxes by meaningful perturbation. IEEE Int Conf on Computer Vision, p.3429-3437. https://doi.org/10.1109/ICCV.2017.371

Goyal Y, Mohapatra A, Parikh D, et al., 2016. Towards transparent AI systems: interpreting visual question answering models. https://arxiv.org/abs/1608.08974

He K, Zhang X, Ren S, et al., 2016. Deep residual learning for image recognition. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.770-778. https://doi.org/10.1109/CVPR.2016.90

Hu Z, Ma X, Liu Z, et al., 2016. Harnessing deep neural networks with logic rules. http://arxiv.org/abs/1603.06318

Huang G, Liu Z, Weinberger KQ, et al., 2017. Densely connected convolutional networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.4700-4708.

Kindermans PJ, Schütt KT, Alber M, et al., 2017. Learning how to explain neural networks: patternnet and patternattribution. http://arxiv.org/abs/1705.05598

Koh P, Liang P, 2017. Understanding black-box predictions via influence functions. Proc 34th Int Conf on Machine Learning, p.1885-1894.

Krizhevsky A, Sutskever I, Hinton GE, 2012. Imagenet classification with deep convolutional neural networks. NIPS, p.1097-1105.

Kumar D, Wong A, Taylor GW, 2017. Explaining the unexplained: a class-enhanced attentive response (clear) approach to understanding deep neural networks. IEEE Conf on Computer Vision and Pattern Recognition Workshops, p.1686-1694. https://doi.org/10.1109/CVPRW.2017.215

Lakkaraju H, Kamar E, Caruana R, et al., 2017. Identifying unknown unknowns in the open world: representations and policies for guided exploration. Proc 31st AAAI Conf on Artificial Intelligence, p.2124-2132.

LeCun Y, Bottou L, Bengio Y, et al., 1998a. Gradient-based learning applied to document recognition. *Proc IEEE*, 86(11):2278-2324. https://doi.org/10.1109/5.726791

LeCun Y, Cortes C, Burges CJ, 1998b. The MNIST Database of Handwritten Digits. http://yann.lecun.com/exdb/mnist/ [Accessed on June, 2017]

Liu Z, Luo P, Wang X, et al., 2015. Deep learning face attributes in the wild. IEEE Int Conf on Computer Vision, p.3730-3738. https://doi.org/10.1109/ICCV.2015.425

Lu Y, 2015. Unsupervised learning on neural network outputs (v9). http://arxiv.org/abs/1506.00990

Mahendran A, Vedaldi A, 2015. Understanding deep image representations by inverting them. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.5188-5196. https://doi.org/10.1109/CVPR.2015.7299155

Netzer Y, Wang T, Coates A, et al., 2011. Reading digits in natural images with unsupervised feature learning. NIPS, p.1-9.

Nguyen A, Clune J, Bengio Y, et al., 2017. Plug & play generative networks: conditional iterative generation of images in latent space. IEEE Conf on Computer Vision and Pattern Recognition, p.3510-3520. https://doi.org/10.1109/CVPR.2017.374

Olah C, Mordvintsev A, Schubert L, 2017. Feature visualization. Distill. https://doi.org/10.23915/distill.00007

Paysan P, Knothe R, Amberg B, et al., 2009. A 3D face model for pose and illumination invariant face recognition. 6th IEEE Int Conf on Advanced Video and Signal Based Surveillance, p.296-301. https://doi.org/10.1109/AVSS.2009.58

Ribeiro MT, Singh S, Guestrin C, 2016. "Why should I trust you?" explaining the predictions of any classifier. Proc 22nd ACM SIGKDD Int Conf on Knowledge Discovery and Data Mining, p.1135-1144 . https://doi.org/10.1145/2939672.2939778

Sabour S, Frosst N, Hinton GE, 2017. Dynamic routing between capsules. NIPS, p.3859-3869.

Selvaraju RR, Cogswell M, Das A, et al., 2017. Grad-CAM: visual explanations from deep networks via gradient-based localization. IEEE Int Conf on Computer Vision, p.618-626. https://doi.org/10.1109/ICCV.2017.74

Simonyan K, Vedaldi A, Zisserman A, 2013. Deep inside convolutional networks: visualising image classification models and saliency maps. http://arxiv.org/abs/1312.6034

Springenberg JT, Dosovitskiy A, Brox T, et al., 2015. Striving for simplicity: the all convolutional net. Inte Conf on Learning Representations, p.1-14.

Su J, Vargas DV, Kouichi S, 2017. One pixel attack for fooling deep neural networks. http://arxiv.org/abs/1710.08864

Szegedy C, Zaremba W, Sutskever I, et al., 2014. Intriguing properties of neural networks. http://arxiv.org/abs/1312.6199

Wang P, Wu Q, Shen C, et al., 2017. The VQA-machine: learning how to use existing vision algorithms to answer new questions. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1173-1182. https://doi.org/10.1109/CVPR.2017.416

Wu TF, Zhu SC, 2011. A numerical study of the bottom-up and top-down inference processes in And-Or graphs. *Int J Comput Vis*, 93(2):226-252.

Wu TF, Xia GS, Zhu SC, 2007. Compositional boosting for computing hierarchical image structures. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.1-8. https://doi.org/10.1109/CVPR.2007.383034

Wu TF, Li X, Song X, et al., 2017. Interpretable R-CNN. http://arxiv.org/abs/1711.05226

Yang X, Wu TF, Zhu SC, 2009. Evaluating information contributions of bottom-up and top-down processes. IEEE 12th Int Conf on Computer Vision, p.1042-1049. https://doi.org/10.1109/ICCV.2009.5459386

Yosinski J, Clune J, Bengio Y, et al., 2014. How transferable are features in deep neural networks? NIPS, p.1173-1182.

Zeiler MD, Fergus R, 2014. Visualizing and understanding convolutional networks. European Conf on Computer Vision, p.818-833. https://doi.org/10.1007/978-3-319-10590-1_53

Zhang Q, Cao R, Wu YN, et al., 2016. Growing interpretable part graphs on convnets via multi-shot learning. Proc 30th AAAI Conf on Artificial Intelligence, p.2898-2906.

Zhang Q, Cao R, Wu YN, et al., 2017a. Mining object parts from CNNs via active question-answering. Proc IEEE Conf on Computer Vision and Pattern Recognition, p.346-355. https://doi.org/10.1109/CVPR.2017.414

Zhang Q, Cao R, Zhang S, et al., 2017b. Interactively transferring CNN patterns for part localization. http://arxiv.org/abs/1708.01783

Zhang Q, Wang W, Zhu SC, 2018a. Examining CNN representations with respect to dataset bias. Proc 32nd AAAI Conf on Artificial Intelligence, in press.

Zhang Q, Cao R, Shi F, et al., 2018b. Interpreting CNN knowledge via an explanatory graph. Proc 32nd AAAI Conf on Artificial Intelligence, p.2124-2132.

Zhang Q, Yang Y, Wu YN, et al., 2018c. Interpreting CNNs via decision trees. http://arxiv.org/abs/1802.00121

Zhang Q, Wu YN, Zhu SC, 2018d. Interpretable convolutional neural networks. Proc IEEE Conf on Computer Vision and Pattern Recognition, in press.

Zhou B, Khosla A, Lapedriza A, et al., 2015. Object detectors emerge in deep scene CNNs. http://arxiv.org/abs/1412.6856

Zintgraf LM, Adel TSCT, Welling M, 2017. Visualizing deep neural network decisions: prediction difference analysis. http://arxiv.org/abs/1702.04595