Semantic Network Interpretation

Pei Guo Ryan Farrell
Brigham Young University
{peiguo, farrell}@cs.byu.edu

Abstract

Network interpretation as an effort to reveal the features learned by a network remains largely visualization-based. In this paper, our goal is to tackle semantic network interpretation at both filter and decision level. For filter-level interpretation, we represent the concepts a filter encodes with a probability distribution of visual attributes. The decision-level interpretation is achieved by textual summarization that generates an explanatory sentence containing clues behind a network's decision. A Bayesian inference algorithm is proposed to automatically associate filters and network decisions with visual attributes. Human study confirms that the semantic interpretation is a beneficial alternative or complement to visualization methods. We demonstrate the crucial role that semantic network interpretation can play in understanding a network's failure patterns. More importantly, semantic network interpretation enables a better understanding of the correlation between a model's performance and its distribution metrics like filter selectivity and concept sparseness.

1. Introduction

Network interpretation seeks to illuminate or expose the features that have been learned, and its difficulty lies in the end-to-end learning of the feature extraction and classification sub-networks, which typically contain millions of parameters each. "Debugging" an over-confident network, one which assigns the wrong class label to an image with high confidence, can be extremely difficult, especially when adversarial noise [7] is added to deliberately mislead the network to the wrong conclusion. In that case a meaningful explanation is highly desirable, which contains features responsible for triggering the error, similar to the syntax error highlighting of an intelligent compiler. A thorough understanding of the neural networks is an indispensable part for their continuous success. Network interpretation is also crucial for tasks involving humans due to legal reasons. It is therefore important to distill the knowledge learned by deep models and present it in an easy-to-understand way.

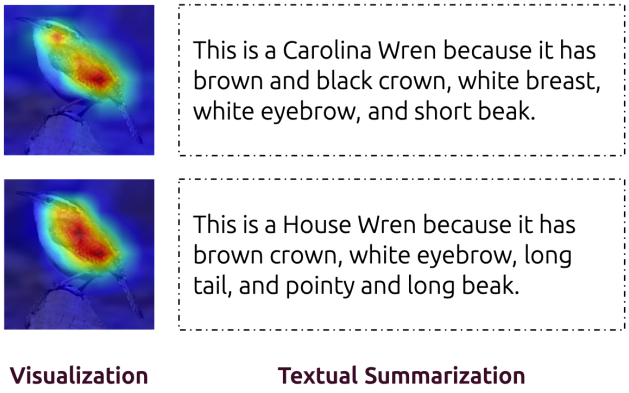


Figure 1: Visualization methods highlight the important image region for different network decisions, but they lack semantic information and finer details compared to semantic interpretation via textual summarization.

Most popular approaches for network interpretation are visualization-based. Filter-level interpretation (understanding the concepts a filter encodes) is often achieved by displaying the maximally activated dataset example [34] (Figure 2) or the optimized input image with image prior regulation [19, 18]. Decision-level interpretation (understanding why the network makes a decision, also called attribution) [38, 22, 27, 25, 24] is often achieved by highlighting a region in the image that's important for the decision-making. Despite their success at providing visual clues, pure visualization is unable to provide semantic explanation and sometimes misses detailed information, as shown in Figure 1. Similarly, Adebayo, *et al*. [1] argues that "reliance, solely, on visual assessment can be misleading."

Humans, on the other hand, can justify their conclusions using natural language. For instance, a knowledgeable person looking at a photograph of a bird might say, "I think this is an Anna's Hummingbird because it has a straight bill, and a red throat and crown. It's not a Broad-tailed Hummingbird because the latter lacks the red crown." This kind of textual description carries rich semantic information and is easily understandable. Semantic information is a logical medium in which to ground the interpretation of deep convolutional

models, serving as a beneficial supplement for the visualization methods.

This paper focuses on semantic network interpretation at both filter and decision level. An intuitive way for semantic filter interpretation is to assign a single concept to each filter, as did in [2]. However, the filter-concept relation is usually not one-to-one: a filter can represent several concepts and a concept can be encoded by multiple filters. This distributed characteristic improves a model's representation efficiency by design [12]. We instead propose to represent a filter with a conditional multinomial probability distribution, called the filter-attribute distribution (see Figure 4 for an example). Intuitively, an attribute t is more likely to represent a filter f if images containing t frequently activate filter f. We further tackle semantic interpretation for network decision using textual summarization. Textual summarization aims to find a list of visual attributes that the network is basing its decision on. A natural sentence is generated with the top attributes as supporting evidence. A direct application of textual summarization is network debugging, generating descriptive error messages when the network's prediction is wrong, and it helps us to identify three major failure patterns for the fine-grained dataset CUB-200-2011 [32] (Section 4.1).

We devise a Bayesian inference algorithm to compute the posterior probability that a filter f is activated by a visual attribute t as $p(t|f)$. The difference between our algorithm and a visual attribute prediction algorithm is that the later usually associates visual attributes to an image in a supervised way, but ours associates visual attributes to filters and decisions in an unsupervised way. The goal of network interpretation is not to predict the target label but to loyally reflect the internal working mechanism of a neural network. The key differences between this work and network dissection [2] are that we use a Bayesian algorithm to represent a filter with an attribute distribution instead of a single concept and we only leverage image-level caption annotations.

The filter-attribute distribution provides a tool to quantitatively understand how concepts are encoded by filters. Specifically, we explored the correlation between a model's performance with the distributed level of its representation. Two metrics of distributed representation are examined, namely filter selectivity and concept sparseness [4]. Filter selectivity is measured by the number of distinctive concepts a filter represents, and concept sparseness refers to the way a single concept is distributed among filters. Understanding the correlation between a network's performance with its distributed characteristics could potentially lead to new optimization functions to train better networks. Section 4.3 provides a thorough evaluation and discussion. Further more, an ablation study shows that deleting less selective filters is likely to cause more damage to a network, a contrast to our intuition. Human study shows that 41.5% of users think textual attributes are a better medium for network interpretation than visualization, and users find 80.1% of the top 5 attributes in the filter-attribute distributions are accurate.

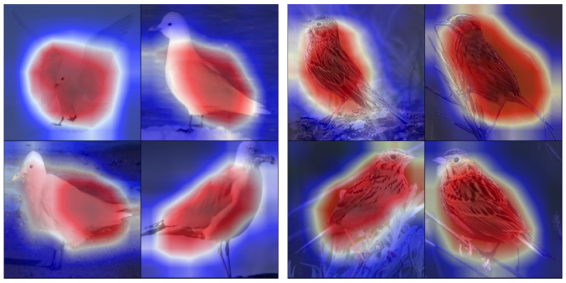


Figure 2: Examples of filter visualization using images with maximal activation, each masked by their corresponding feature maps. One limitation of the visualization-based filter interpretation is the lacking of **diversity**: it is unable to capture the whole space of represented concepts with a limited number of data samples.

2. Related Work

Network interpretation – Two main approaches to network interpretation exist in the literature: filter-level interpretation [5, 28, 16, 19, 8, 18, 17, 2, 37, 33, 26, 34] and decision-level interpretation (or attribution) [23, 38, 22, 39, 13]. The goal of filter-level interpretation is to understand the features that a specific filter (also known as neurons) learns. While it is easy to directly visualize the first convolutional layer filter weights and understand the patterns they detect, it makes little sense to directly visualize deeper layer filter weights because they act as complex composite functions of lower layers' output. Early examples of filter-level interpretation include finding the maximally activated input patches [34] and visualizing the guided back propagation gradients [26]. Some works [19] try to synthesize visually pleasant preferred input image of each filter through back-propagation into the image space. [18] applies a generator network to generate images conditioned on maximally activating certain last-layer neurons. The Plug and Play paper [17] further extends [18] to introduce a generalized adversarial learning framework for filter-guided image generation. Network dissection [2] connects each filter with predefined concepts like object, part, color, etc.

Attempts of decision-level interpretation mainly focus on visualizing important image subregions by re-weighting final convolutional layer feature maps. Examples include [38, 22, 27, 25, 24]. However, the visualization based method only provides coarse-level information, and it remains hard to intuitively know what feature or pattern the

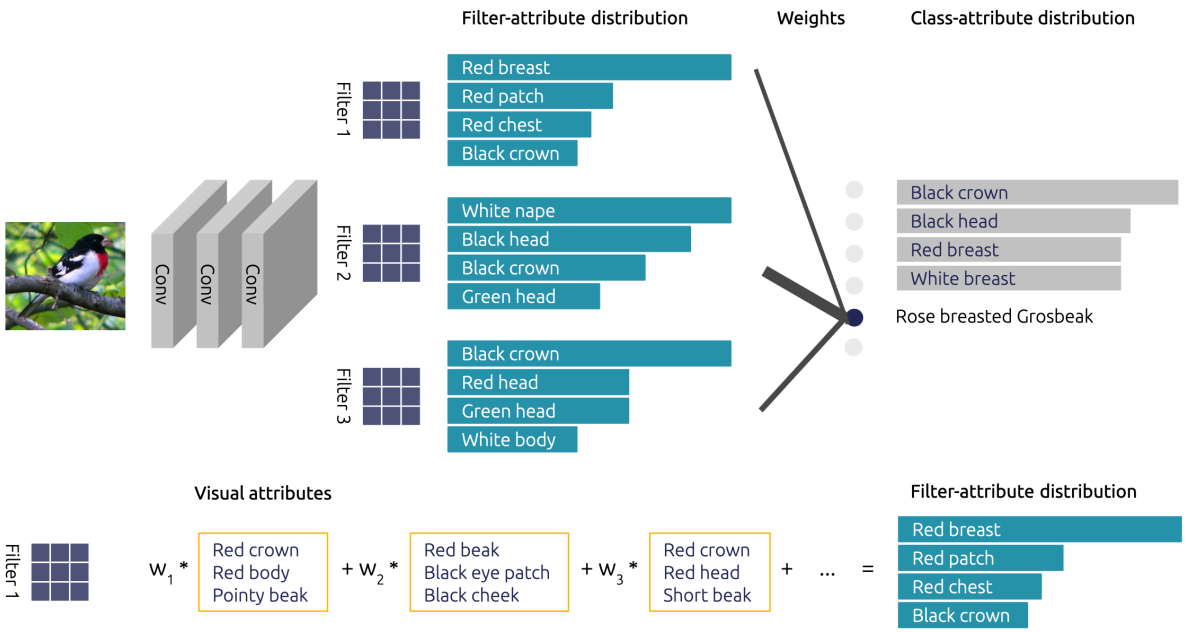


Figure 3: An overview of the algorithms for decision-level and filter-level semantic interpretation using class-attribute distribution (top row) and filter-attribute distribution (bottom). Visual attributes in every image are weighted by the activation strength and its importance factor (TF/IDF) to generate filter-attribute distribution (section 3.1). The filter-attribute distribution are re-weighted by linear layer weight and the activation strength to generate class-attribute distribution (section 3.2).

network has learned to detect. More importantly, the holistic heat map representation is sometimes insufficient to justify why the network favors certain classes over others when the attentional maps for different classes overlap heavily. See Figure 1 for example. [36] proposes represent the image content and structure by knowledge graph.

Visual attribute prediction and image captioning – Visual attribute prediction and image captioning [6] are related but fundamentally different tasks to semantic network interpretation. Visual attribute prediction and image captioning are often supervised with the goal to approximate ground truth labels. Semantic interpretation, on the other hand, aims to loyally reflect the knowledge learned by a model in an unsupervised way. There are no ground truth labels to approximate in semantic interpretation.

We note that [10] defines a task similar to ours, to explain and justify a classification model. Their model is learned in a supervised manner, with explanations generated from an LSTM network which only implicitly depends on the internal feature maps. It is essentially an image captioning task that generates captions with more class-discriminative information. Our method is unsupervised and does not rely on another black-box network to generate descriptions.

Class activation map and network dissection – Class Activation Map (CAM) identifies the most important region in an image by the linear combination of final conv-layer feature maps, whose weight is from the parameter in the fully connected layer that connects the feature map to the class label: $M_c(x,y) = \sum_k w_k^c f_k(x,y)$, where $M_c(x,y)$ measures the importance of spatial location (x,y) for class c. $f_k(x,y)$ is the value at (x,y) on the kth filter's feature map. w_k^c is the weight that connected the kth feature map to the prediction class c.

Network dissection [2] is perhaps the most similar work to ours. In [2], a filter is associated to a concept by measuring the overlap between the thresholded filter feature map and the concept segmentation mask. Intersection over union $\text{IoU}_{k,c}$ is proposed to represent the accuracy of unit k in detecting concept c. The main different between our work and network dissection is that we model the filter attribute relation as conditional multinomial probability distribution and propose a general Bayesian inference algorithm to link a filter to multiple attributes. Our algorithm relies only on image-level caption annotation instead of pixel-level segmentation annotation.

3. Bayesian Inference Framework

For the filter-level interpretation, we seek to represent each network filter with its respective activation patterns in terms of visual attributes. Constructing a paired filter attribute dataset is unrealistic, because the filter (as a composite function) is not a well-defined concept with concrete examples. Instead, we propose leveraging off-the-shelf image caption annotations because they contain rich textual ref-

Figure 4: An example of filter-level semantic interpretation using filter-attribute distribution, which is a probability distribution of visual attributes that best describe the concepts encoded by a filter.

erences to visual concepts. The intuition behind our filter-attribute association is simple: a filter can be represented by the images that strongly activate them and the visual attributes contained in such images should have a high probability of representing the filter. The joint consensus of all images in the dataset can increase the probability of the relevant visual attributes and suppress that of the irrelevant visual attributes.

3.1. Filter-Attribute Distribution

We denote $\mathcal{F} = \{f_i | i = 1, ..., m\}$ as the group of final conv-layer model filters. We denote $\mathcal{X} = \{x_j | j = 1, ..., n\}$ as the set of input images. The filter f's output for input x is written as $f(x)$ (with a slight abuse of notations), which we call a feature map or filter activation. We consider models [9, 14] with a global pooling layer ϕ followed by a single fully-connected layer. The global pooling layer output for x is written as $\phi(f(x))$. The output of the fully-connected layer is the class prediction from $\mathcal{C} = \{c_k | k = 1, ..., o\}$. The weight matrix of the fully-connected layer is $W^{o \times m}$. A list of textual attributes from $\mathcal{T} = \{t_l | l = 1, ..., p\}$ is attached to each image. We loosely denote by $t \in x$ if t is contained in x's attribute list. x^t represents images that contain attribute t.

We're interested to know the representative visual attributes for a filter in the network's final-conv layer. For a given filter f, the probability that an attribute t can represent its activation pattern is:

$$p(t|f) \propto p(f|t)p(t) \tag{1}$$

$p(t)$ is the prior probability for visual attribute t. We consider the relative importance of attributes because attributes carry different amount of information. For example, "small bird" has less information than "orange beak" because the latter appears less in the text corpora and corresponds to a more important image feature. We employ the normalized TF/IDF feature as the attribute prior. The term frequency (TF) of a phrase is its number of occurrences in the same captioning file. The inverse document frequency (IDF) of a phrase is the logarithm of total file number divided by the number of files containing the phrase.

$p(f|t)$ measures the likelihood of attribute t activating filter f. As attributes are not directly involved in the neural network, we introduce the input image as a hidden variable:

$$\begin{aligned} p(f|t) &\propto p(f|\mathcal{X}, t)p(\mathcal{X}, t) \\ &= \prod_j p(f|x_j, t)p(x_j, t) \end{aligned} \tag{2}$$

where $\mathcal{X}$ represents the set of input images. $p(x_j, t)$ measures the probability that x_j contains t:

$$p(x_j, t) = \begin{cases} 1 & \text{if} \quad t \in x_j \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We use x^t to represent images containing t in their attribute list. According to Eqn 2 and Eqn 3, images without attribute t are zeroed out, so we have $p(f|t) \propto \sum_j p(f|x_j^t, t)$. $p(f|x_j^t, t)$ measures the likelihood that the image x_j^t and the attribute t are the reason for filter f's activation. f is conditionally independent of t given x_j^t:

$$\begin{aligned} p(f|x_j^t, t) &= p(f|x_j^t) \\ &\propto \phi(f(x_j^t)) \end{aligned} \tag{4}$$

where $\phi(f(x_j^t))$ is the global pooling layer output for input x_j^t and filter f, which measures how likely an image will activate a filter. To summarize, the posterior probability that attribute t is the reason that filter f activates is given by:

$$p(t|f) \propto \text{TF/IDF}(t) \prod_j \phi(f(x_j^t)) \tag{5}$$

The approximation that $p(f|x_j^t, t) = p(f|x_j^t)$ neglects the fact that when an image activates a filter, the feature map favors certain attributes over others. For example, if $f(x_j)$ highlights the head area of a bird, attributes related to "head", "beak" or "eyes" should be assigned with higher probabilities than attributes related to "wings" and "feet". Although this approximation assigns equal probability to all visual attributes inside an image, it actually works quite well in practice, as the joint consensus of all input images boosts true attributes and suppresses false ones. Note that the proposed method can easily adapt to datasets with other forms of annotations like keypoints or part segmentation. Higher probability can be assigned to the visual attributes associated with a part when the feature activation map overlaps highly with its segmentation mask.

3.2. Textual Summarization

With the help of the filter-attribute distribution, we can find the top attributes that account for the network's classification decision. This task can be formulated as the probability that a visual attribute t is the underlying reason given

A B C D E F

A. This is a **American Pipit** because it has **short beak**, **yellow and white breast**, **brown crown**, and **yellow throat**.
B. This is a **Eared Grebe** because it has **black crown**, **long neck**, **black head**, **red eye**, and **white throat**.
C. This is a **Groove billed Ani** because it has **black head**, **thick and pointy and black beak**, and **black crown**.
D. This is a **Parakeet Auklet** because it has **black and red crown**, **black head**, **white breast**, and **black throat**.
E. This is a **Philadelphia Vireo** because it has **yellow breast**, **grey head**, **yellow throat**, **short beak**, and **white eyebrow**.
F. This is a **Great Grey Shrike** because it has **white breast**, **black crown**, **white throat**, **black beak**, and **black cheek patch**.

Figure 5: Examples of textual summarization that contains the top visual attributes that are accountable for a network's decision-making. Note that the goal of textual summarization is not to accurately predict the image attributes, but to loyally reflect the reasons behind a neural network's classification decisions.

the fact that the network predicts input image x as class c. We introduce final convolutional layer filters $\mathcal{F}$ as hidden variables and by marginalizing over $\mathcal{F}$ we get:

$$\begin{aligned} p(t|x,c) &\propto p(t|\mathcal{F},x,c)p(\mathcal{F}|x,c) \\ &= p(t|\mathcal{F})p(\mathcal{F}|x,c) \\ &= \prod_k p(t|f_i)p(f_i|x,c) \end{aligned} \tag{6}$$

where $p(t|x,c)$ is the probability that t is the reason for the network predicting class c for image x. t is conditionally independent from x and c given $\mathcal{F}$, such that $p(t|\mathcal{F},x,c) = p(t|\mathcal{F})$. $p(t|f_i)$ can be computed from filter f_i's attribute distribution using Eqn 5. $p(f_i|x,c)$ measures the importance of filter f_i in the decision-making process, and it's proportional to the product of the global pooling layer's output of f_i denoted as $\phi(f_i(x))$ and the weight between filter f_i and class c, $w_{i,c}$:

$$p(f_i|x,c) \propto \phi(f_i(x))w_{i,c} \tag{7}$$

We call $p(\mathcal{T}|\mathcal{X},\mathcal{C})$ the class-attribute distribution.

A sentence is generated to describe the network's decision-making process using the class-attribute distribution. Although it's popular to employ a recurrent model for sentence generation, our task is to faithfully reflect the internal features learned by the network and introducing another network could result in additional uncertainty. We instead propose a simple template-based method using the top n attributes, with the following form:

"This is a {class name} because it has {attribute 1}, {attribute 2}, ..., and {attribute n}."

4. Experiments

We evaluate the proposed algorithms on the fine-grained bird dataset of CUB-200-2011 [31] which contains 5997 training images and 5797 testing images. Two ways to obtain image-level attribute annotations for the CUB-200-2011 dataset are explored. The first is to leverage the image caption annotations provided by Zhang, *et al*. [35], which include five captions for every image that describes the visual features the bird in the image has. Visual attributes are extracted from the captions as adjective-noun word phrases. The CUB-200-2011 dataset also provides visual attribute annotation. There are 312 total attributes to be labelled for each image. Examples include: "Has bill length::longer than head", "Has back color::grey" and "Has back color::grey", *etc*. Although the visual attribute annotation can be more accurate, visual attributes from the captions are more diverse and fine-grained. All the visual attributes shown in this paper are generated from image captions.

To extract visual attributes from the image captions, we follow the process of word tokenization, part-of-speech tagging and noun-phrase chunking. A total of 9649 independent attributes are obtained. Examples of the generated filter-attribute distribution are shown in Figure 4. Examples of the generated textual explanations for image classification are shown in Figure 5.

4.1. Network Debugging

In figure 6, we show three major patterns of network failure through textual summarization. In the first example, a Tree Sparrow is incorrectly recognized as a Chipping Sparrow because the network mistakenly thinks "long tail" is

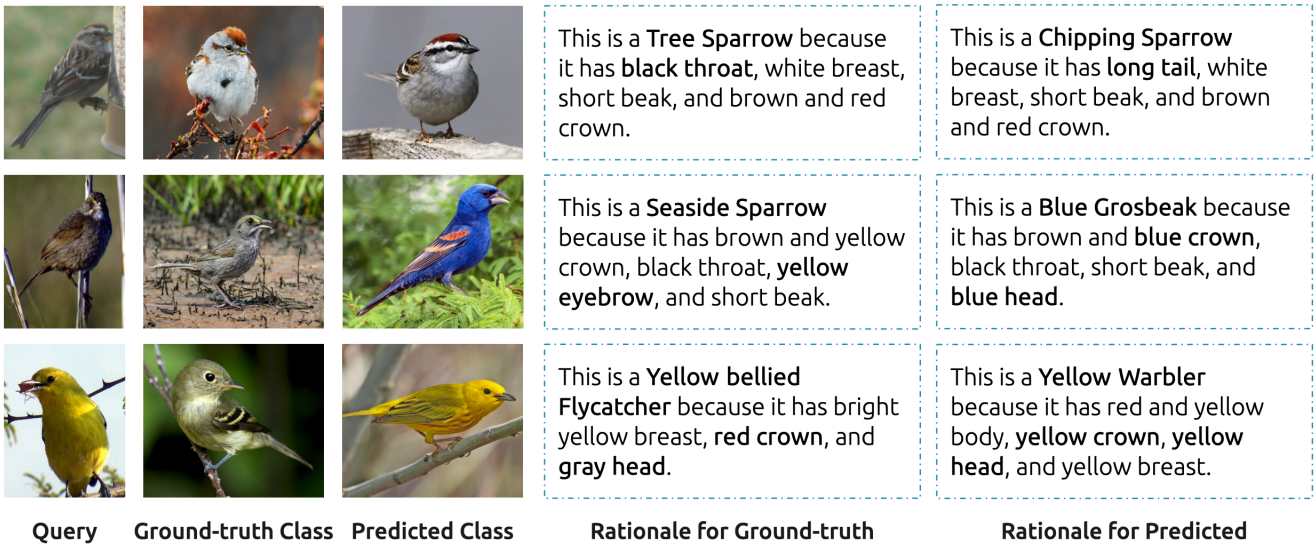


Figure 6: Each row represents a network failure – an incorrectly predicted class label. From left to right, each column shows the query image, canonical images for both the ground-truth and the *incorrectly* predicted classes, and the textual explanations for each of these classes.

a discriminative feature. According to wikipedia, American Tree Sparrows have a rufous stripe through the eye; on Chipping Sparrows it's black. Tree sparrows also have a spot in the middle of the breast and a bicolored bill that Chipping Sparrows don't have. Failing to identify the correct features for discrimination is the most common source of errors across the dataset. In fine-grained classification, the main challenge is to identify discriminative features for visually-similar classes, differences of which are often subtle and localized to small parts.

The second example shows a Seaside Sparrow that has mistakenly been recognized as a Blue Grosbeak. From the textual explanations we ascertain that the low image quality mistakenly activates filters that correspond to blue head and blue crown. The underlying source of this error is complex – the generalization ability of the network is limited such that small perturbations in the image can result in unwanted filter responses. Such failures imply the critical importance of improving network robustness to noisy inputs.

In the third case, the network predicts the image as a Yellow Warbler, however the ground-truth label is Yellow-bellied Flycatcher. According to a bird expert, the network got this correct – *the ground-truth label is incorrect!* The network correctly identifies the yellow crown and yellow head, both obvious features of the Yellow Warbler. Errors like this are not surprising because, according to [30], the class labels on roughly 4% of the CUB dataset are incorrect.

4.2. Human Study

Visualization vs. semantic interpretation – We conduct human study using the Amazon Mechanical Turk platform to evaluate the proposed semantic interpretation. Our first study aims to know users' preference between visualization methods and semantic interpretation using textual summarization. "A picture is worth a thousand words". It's not surprising that normal users prefer the visualization method that have dominated the network attribution field. However, our study shows that 41.5% of users prefer textual explanations and think they provide more helpful information than the visualization methods. This study confirms that semantic interpretation can serve as a helpful alternative or complement to network visualization.

Filter-attribute distribution evaluation – In order to evaluate the filter-attribute distribution, we list the top five attributes along with the top-activated images for each filter. The users are instructed to select the attributes that are present in most, if not all the highlighted regions of these images. Generally 80.1% of the attributes are regarded as accurate to describe the highlighted regions. Specifically, the accuracy for each of the five attributes are 93.9%, 92.9%, 89.8%, 74.3% and 53.2%. This study shows that most of the visual attributes are rated as relevant to reflect the filter's activation pattern. To evaluate our textual summarization algorithm, we asked the turkers to rate the top five attributes for each image. The average accuracy was 75.8%. The per-attribute accuracies were 89.2%, 88.1%, 83.3%, 66.4% and 51.9%. Further study reveals another interesting phenomenon: for images that are correctly predicted by the network, 76.3% of our top attributes were regarded as accurate; for incorrectly predicted images the accuracy was 73.8%. This indicates the probability that a neural network learns better features for correct classifica-

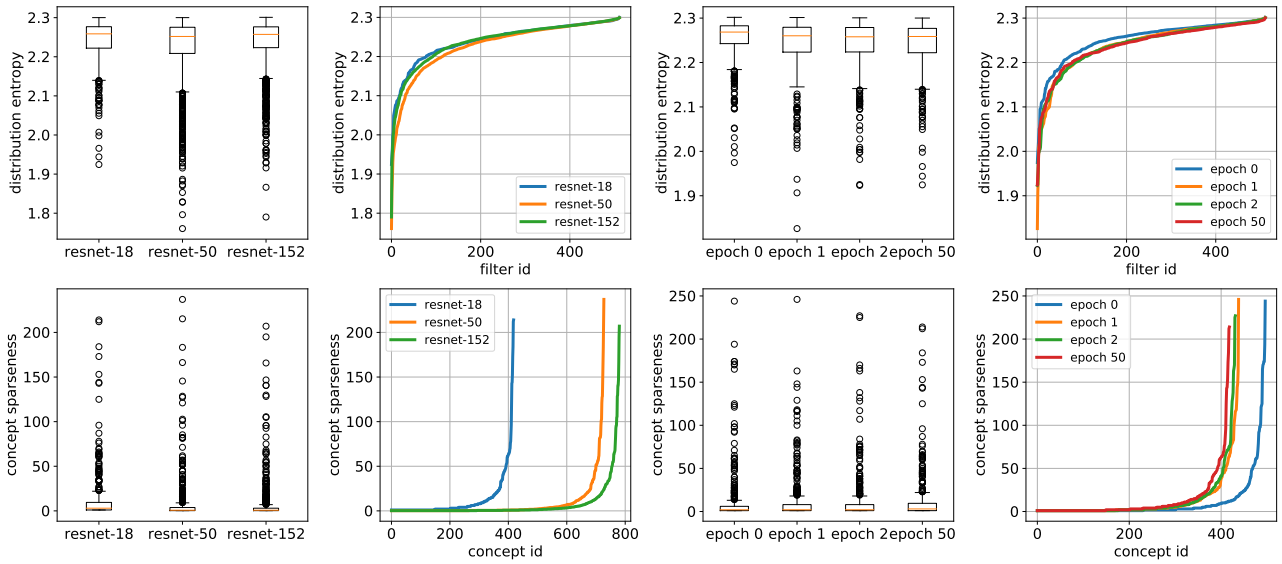


Figure 7: Top Row: first two graphs shows the box plot and the sorted entropy for filter-attribute distribution of ResNet-18, ResNet-50, and ResNet-152 separately; last two graphs shows the filter-attribute distribution entropy of ResNet-18 at different training epochs of 0, 1, 2, 50. Botton Row: first two graphs shows the box plot and the sorted concept sparseness for Resnet-18, ResNet-50, and ResNet-152; last two graphs show the box plot and the sorted concept sparseness during training at epoch 0, 1, 2, 50 separately.

	Mean	No.1	No.2	No.3	No.4	No.5
Filter-attribute distribution	80.1	93.9	92.9	89.8	74.3	53.2
Class-attribution distribution	75.8	89.2	88.1	83.3	66.4	51.9

Table 1: The turkers rated accuracy for filter-attribute distribution and class-attribute distribution.

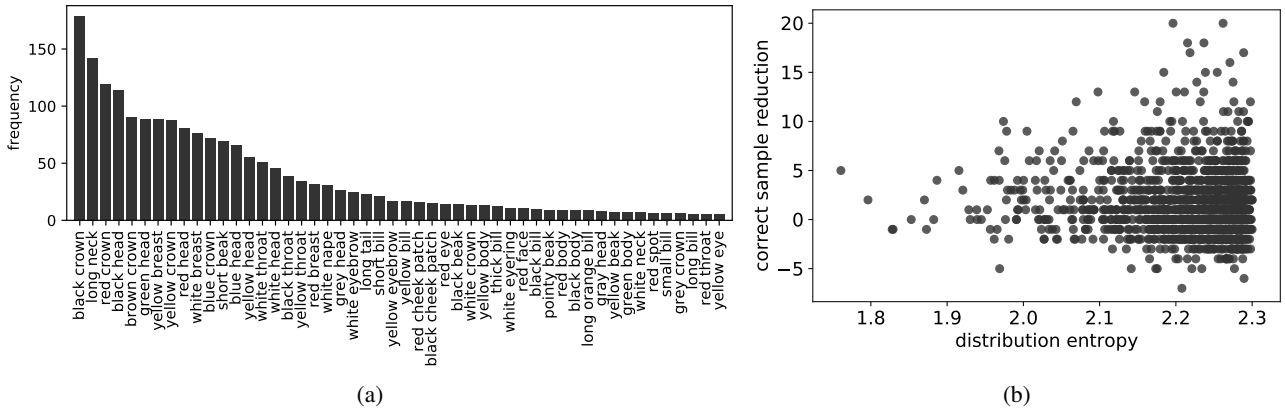
tion than incorrect classification.

4.3. Network Understanding

If network *interpretation* is about knowing what features a network had learned, network *understanding* cares more about what makes a good feature. Defining good features and finding a way to learn them is crucial for the continuous success of deep models. Bengio, *et al*. [3] listed several characteristics a good feature representation should have, *e.g.* disentangling factors of variation, smoothness, abstraction and invariance, and distributed representations. Many efforts [15, 11, 29, 20, 21] have been devoted into understanding each of these properties. In this section, we focus on the distributed representation and its correlation with a model's performance.

Filter Selectivity – A filter's selectivity refers to its representing a small number of concepts. A strongly-selective filter only activates on a narrow set of visual attributes. These filters are more interpretable than those whose activation patterns spread widely across many visual attributes. The entropy of the filter-attribute distribution serves as a good indicator of a filter's selectivity; low entropy means a sparse attribute distribution and strong selectivity. A few questions arise: do models with more interpretable filters perform better? Is the opposite true? Is there a significant correlation between a model's performance with its filters' selectivity?

We compare the filter selectivity of three different models with increasing numbers of parameters: ResNet-18, ResNet-50, and ResNet-152. The classification accuracy on the CUB dataset for these models is 73.2%, 81.6% and 83.4% respectively. The entropy of their filter-attribute distribution are shown in Figure 7. Note that ResNet-18 has 512 filters and ResNet-50, ResNet-152 have 2048 filters. The first two graph on the top row of Figure 7 show the box plot and sorted distribution entropy for the three models. ResNet-18 has the highest entropy and, by definition, lowest selectivity among all three models. ResNet-50 has more strongly-selective filters than ResNet-152 although the later is more accurate. To understand how filter selectivity evolves during training, we take four snapshots of a ResNet-18 network during training with epoch number 0, 1, 2 and 50. The box plot and sorted distribution entropy are shown in last two figures on the top row of Figure 7.



(a) (b)

Figure 8: (a) Visual attributes sorted by the number of filters that encodes them. (b) The scatter plot of filter selectivity (inversely proportional to its attribution distribution entropy) and filter importance (inversely proportional to correct sample reduction after removing it from the model).

Before training, the network has a lower number of selective filter, but the filter selectivity is not strictly increasing during training.

We next study the correlation between a filter's importance and its selectivity. A filter's importance can be measured by the performance drop after deleting it from the model: filters with higher correct sample reduction are of greater importance. We sequentially remove the final convolutional layer filters, one at a time, and record the decrease in correctly predicted samples. Note that the model is not retrained after filter removal. We compare the reduction of correct samples against the filter's selectivity in the scatter plot shown in Figure 8b. Overall, deleting one filter typically has very little impact on the model's performance (± 5), but note that the highest correct sample loss occurs when some of the most weakly-selective filters are deleted. Removing a strongly-selective filter is less likely to result in a performance drop compared to weakly-selective filters. This experiment shows that a filter's importance is surprisingly negatively related to its selectivity. We hypothesis that an important filter encodes some rare concepts and a less important filter encodes some concepts that are highly duplicate. We find that the concepts represented by the most strongly-selective filter are: "yellow crown, black throat, black cheek patch", which are encoded by many filters. Deleting such a filter is less likely to cause a significant dip in a model's performance.

Concept Sparseness – Concept sparseness refers to the fact that a concept is represented by several filters. We represent a concept's sparseness by the number of filters whose top 10 activation pattern contains such a concept. Figure 8a shows the most popular concepts (visual attributes) in descending order. 'black crown', the top concept, is spread across 179 filters, followed by the 'long neck' concept spread across 149 filters. Note that 'Black crown' is also the most frequent attribute in the caption annotation file. The bottom row of Figure 7 shows how concepts are encoded in different models and how they changed during the process of training. ResNet-18 has less concepts encoded than ResNet-50, which is then followed by ResNet-152. During the training phrase for each model, the total concepts reduced but the number of filters that encode a concept increases. Generally speaking, better model encodes more concepts and the concepts become increasingly more distributed in the filters during training.

5. Conclusion

In this paper, we focus on the task of semantic network interpretation at both filter and decision level. We represent the concepts a filter learns as a conditional multinomial probability distribution on visual attributes. A Bayesian inference algorithm is proposed to compute the attribute distribution for both filers and network decision. We study the correlation between a model's performance with its distributed representation. Two metrics (filter selectivity and concept sparseness) are examined. Generally, better models have higher filter selectivity and encode more concepts. During training, the filter selectivity increases and the concepts become increasingly more distributed in the filters. For decision-level semantic interpretation, textual summarization is generated to justify a network's classification results and can be used to uncover the common failure patterns on fine-grained recognition. Human studies are conducted to evaluate the accuracy of the proposed algorithms and validate the importance of semantic network interpretation.

References

[1] Julius Adebayo, Justin Gilmer, Michael Muelly, Ian Goodfellow, Moritz Hardt, and Been Kim. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

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

[3] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.

[4] Jeffrey S. Bowers. What is a grandmother cell? and how would you know if you found one? *Connection Science*, 23(2):91–95, 2011.

[5] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. Technical Report 1341, University of Montreal, June 2009.

[6] Ali Farhadi, Mohsen Hejrati, Mohammad Amin Sadeghi, Peter Young, Cyrus Rashtchian, Julia Hockenmaier, and David Forsyth. Every picture tells a story: Generating sentences from images. In *ECCV*, 2010.

[7] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and Harnessing Adversarial Examples. *ArXiv e-prints*, Dec. 2014.

[8] Google. Inceptionism: Going deeper into neural networks. `https://ai.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html`.

[9] Kaiming He, Georgia Gkioxari, Piotr Dollar, and Ross Girshick. Mask r-cnn. In *ICCV*, Oct 2017.

[10] Lisa Anne Hendricks, Zeynep Akata, Marcus Rohrbach, Jeff Donahue, Bernt Schiele, and Trevor Darrell. Generating visual explanations. In *ECCV*, 2016.

[11] Irina Higgins, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner. Towards a definition of disentangled representations. *arXiv preprint arXiv:1812.02230*, 2018.

[12] Geoffrey E Hinton. Distributed representations. 1984.

[13] Andreas Holzinger, Bernd Malle, Anna Saranti, and Bastian Pfeifer. Towards multi-modal causability with graph neural networks enabling information fusion for explainable ai. *Information Fusion*, 71:28–37, 2021.

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

[15] Francesco Locatello, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem. Challenging common assumptions in the unsupervised learning of disentangled representations. *arXiv preprint arXiv:1811.12359*, 2018.

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

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

[18] Anh Nguyen, Alexey Dosovitskiy, Jason Yosinski, Thomas Brox, and Jeff Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. In *NIPS*. 2016.

[19] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *CVPR*, 2015.

[20] Anh Nguyen, Jason Yosinski, and Jeff Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 427–436, 2015.

[21] Avraham Ruderman, Neil C Rabinowitz, Ari S Morcos, and Daniel Zoran. Pooling is neither necessary nor sufficient for appropriate deformation stability in cnns. *arXiv preprint arXiv:1804.04438*, 2018.

[22] 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.

[23] K. Simonyan, A. Vedaldi, and A. Zisserman. Deep Inside Convolutional Networks: Visualising Image Classification Models and Saliency Maps. *ArXiv e-prints*, Dec. 2013.

[24] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.

[25] Jost Tobias Springenberg, Alexey Dosovitskiy, Thomas Brox, and Martin Riedmiller. Striving for simplicity: The all convolutional net. *arXiv preprint arXiv:1412.6806*, 2014.

[26] J. T. Springenberg, A. Dosovitskiy, T. Brox, and M. Riedmiller. Striving for Simplicity: The All Convolutional Net. *ArXiv e-prints*, Dec. 2014.

[27] Mukund Sundararajan, Ankur Taly, and Qiqi Yan. Axiomatic attribution for deep networks. *arXiv preprint arXiv:1703.01365*, 2017.

[28] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *ArXiv e-prints*, Dec. 2013.

[29] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[30] Grant Van Horn, Steve Branson, Ryan Farrell, Scott Haber, Jessie Barry, Panos Ipeirotis, Pietro Perona, and Serge Belongie. Building a bird recognition app and large scale dataset with citizen scientists: The fine print in fine-grained dataset collection. In *CVPR*, 2015.

[31] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The caltech-ucsd birds-200-2011 dataset. Technical Report CNS-TR-2011-001, 2011.

[32] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, California Institute of Technology, 2011.

[33] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson. Understanding Neural Networks Through Deep Visualization. *ArXiv e-prints*, June 2015.

[34] M. D Zeiler and R. Fergus. Visualizing and Understanding Convolutional Networks. *ECCV*, 2013.

[35] Han Zhang, Tao Xu, Hongsheng Li, Shaoting Zhang, Xiaolei Huang, Xiaogang Wang, and Dimitris Metaxas. Stackgan: Text to photo-realistic image synthesis with stacked generative adversarial networks. *arXiv preprint*, 2017.

[36] Q. Zhang, R. Cao, F. Shi, Y. Nian Wu, and S.-C. Zhu. Interpreting CNN Knowledge via an Explanatory Graph. *AAAI*, Aug. 2017.

[37] B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba. Object Detectors Emerge in Deep Scene CNNs. *ICLR*, Dec. 2014.

[38] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, 2016.

[39] Bolei Zhou, Yiyou Sun, David Bau, and Antonio Torralba. Interpretable basis decomposition for visual explanation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 119–134, 2018.