“Interpreting CNNs via Decision Trees”

Quanshi Zhang, Yu Yang, Haotian Ma, and Ying Nian Wu

Shanghai Jiao Tong University, UCLA, South China University of Technology

Presented by: Jason T. Smith

Motivation

- An atte
  the "bla
  trained
  open w
  of Decis
  - Auth
    to "e
    know

1) Explaining features of a CNN's middle layers at a semantic level. **Acquire less abstract meaning, such as object parts, to help understanding**.
2) How to quantitatively analyze the rationale of each CNN prediction. "… ***which*** filters/parts pass their information through the CNN" at time of inference?
   - The authors propose obtaining the numerical contribution ("**attribution score**") of each filter during individual inference.

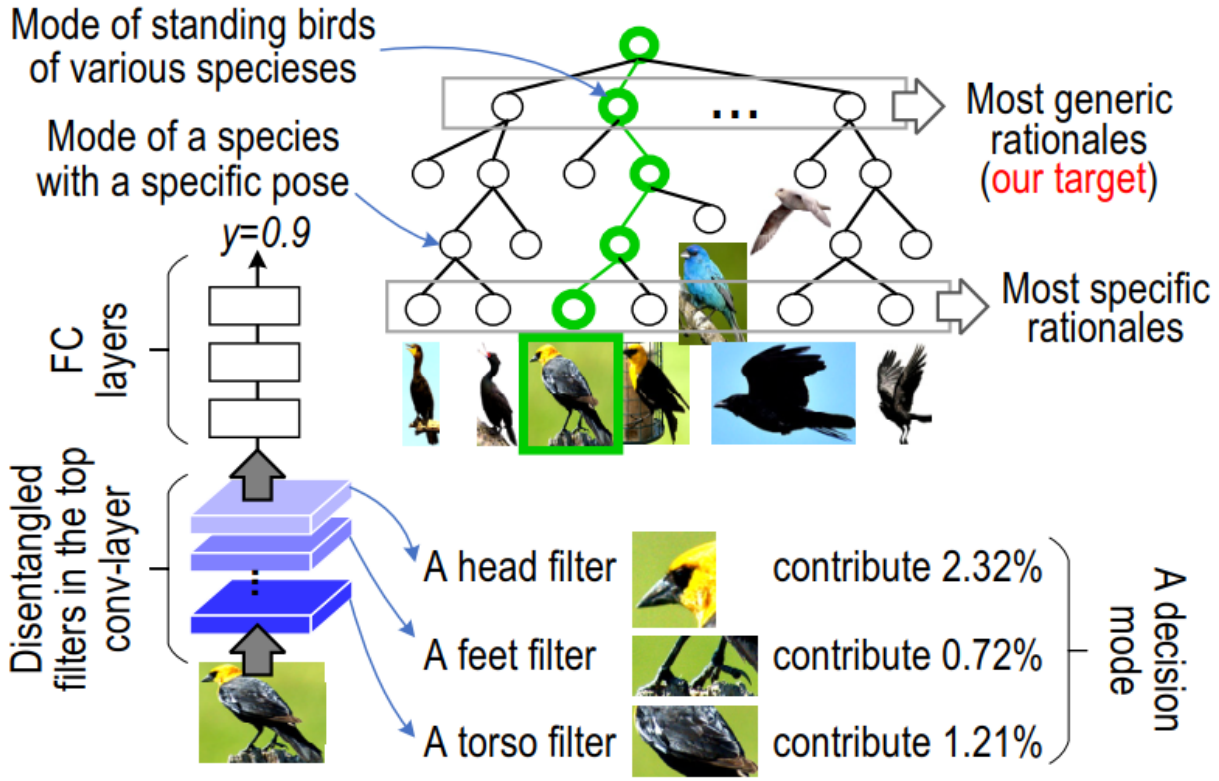
ers

ibution of
butions of
ent filters

ure maps
hape filter

Crude Methodology Overview

- Encoding all potential decision modes of a CNN in a coarse-to-fine manner.
  - **Top conv-layers** represent an **object** part.
  - A parse tree is inferred for each input image to obtain **scores** for each **filter**.
    - Which filters contribute & how much so?



Major Problem Statements

- For each input image, “the first issue is to learn more interpretable feature representations inside a CNN” whilst associating each convolutional operation output with a semantic concept.
  - Forcing feature representations in middle convolutional layers to be well disentangled during learning.
  - Association of each disentangled filter with a corresponding **semantic representation**.
    - “how many parts are memorized in the CNN?”
    - “how are they organized?”

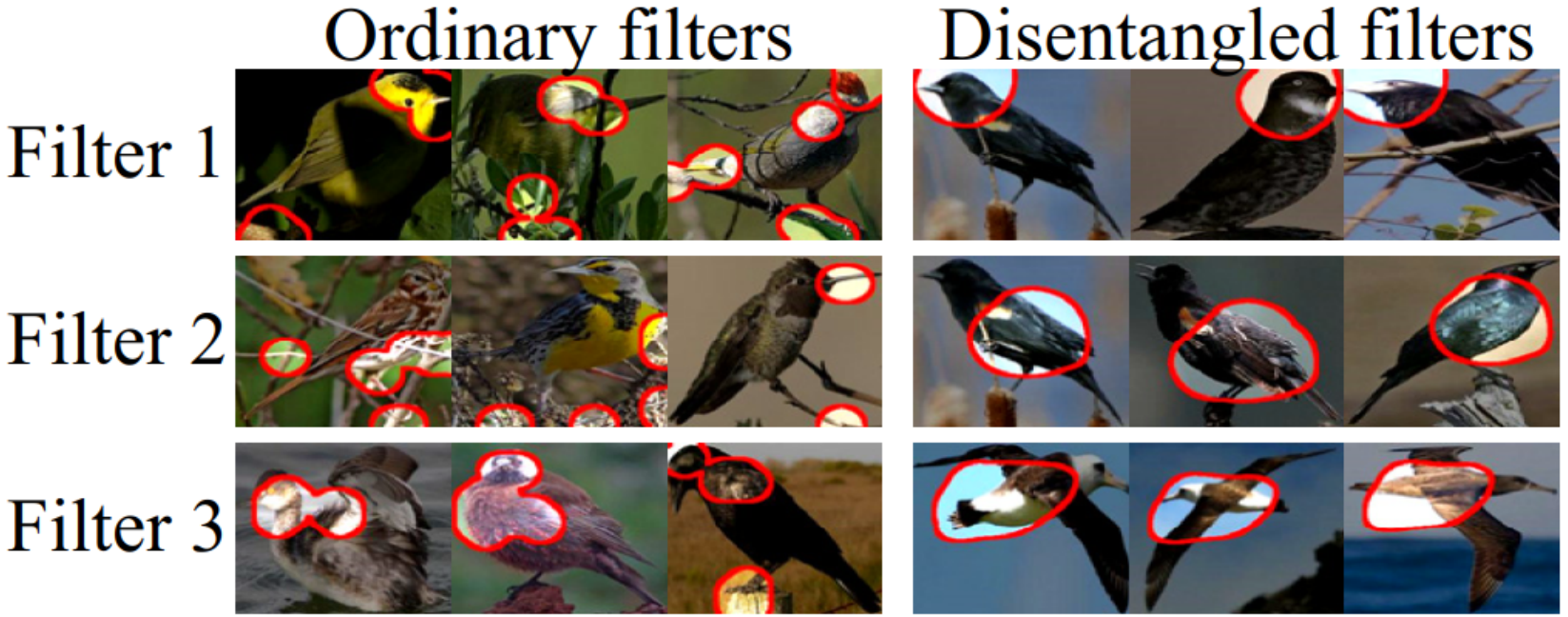
Framework in Brief

- The authors summarize every rationale behind CNN inference into different ***decision modes***.
  - Each tree node represents a mode.
  - Each node describes common prediction rationales shared by multiple images.
    - For these images, the CNN activates similar filters (object parts) and each part illustrates similar “attribution” scores.
- The decision tree represents all potential decision modes of a CNN (“coarse-to-fine”).
  - Nodes closest to the start correspond to common rationales whilst leaf nodes correspond to more fine-grained prediction rationale (minority images).

Building the Tree

- The authors learn filters to represent object parts (without further supervision).
- Each filter is subsequently assigned a part name.
- Decision nodes are thoroughly analyzed to obtain the underlying semantic representation and construct the tree.

Learning Disentangled Features



In brief, the authors “revise a benchmark DNN, in order to make each filter in the top conv-layer represent a specific object part”.

- Filter loss is applied to each filter in the top conv-layers to achieve the above.
- Each filter is learned to be activated by the same object part across images.

Filter loss

$$\mathbf{Loss}_f = \sum_{\mu} Loss_f(x_f) = -MI(\mathbf{X}_f; \mathbf{P})$$

- "… ensures that given an input image, x_f should match ***only one*** of all $L^2 + 1$ location candidates."
- The authors assume that repetitive shapes of various image regions are more likely to describe low-level textures than high.
- When parts appears in the image, x_f should have a single activation peak at the image location. Otherwise, x_f should stay inactive.

$p(x_f, \mu)$ = Joint probability "compatibility between x_f & μ"

(… continued)

- Qualitative analysis as to how the FC layer makes predictions based on the top conv layers

$$x^{(d)} = \frac{1}{s_d} \sum_{h,w} x^{(h,w,d)}$$

$$g^{(d)} = \frac{L^2}{s_d} \sum_{h,w} \frac{\partial y}{\partial x^{(h,w,d)}}$$

$$\therefore y \approx g^T x + b$$

$g^{(h,w,d)} \times x^{(h,w,d)}$ = Quantitative contribution of $x^{(h,w,d)}$ on inference.

y corresponds to scalar classification score

Building the Tree

- The authors learn filters to represent object parts (without further supervision).
- Each filter is subsequently assigned a part name. (manually, Slide #8)
- Decision nodes are thoroughly analyzed to obtain the underlying semantic representation and construct the tree.

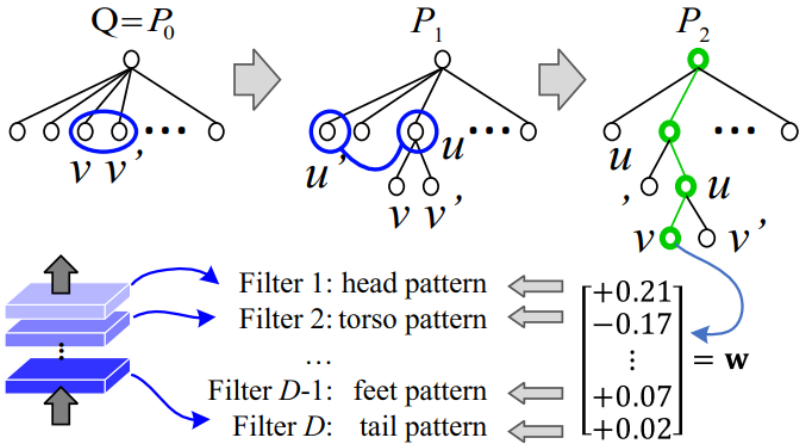


Figure 4. Learning a decision tree. Green lines in P_3 indicate a parse tree to explain the rationale of the prediction on an image.

(… continued)

$$Q = P_0 \rightarrow P_1 \rightarrow P_2 \rightarrow \cdots \rightarrow P_T = \hat{P} \tag{7}$$

We formulate the objective for learning as follows.

$$\max_P E, \qquad E = \underbrace{\frac{\prod_{i\in\mathbf{\Omega}^+} P(\mathbf{x}_i)}{\prod_{i\in\mathbf{\Omega}^+} Q(\mathbf{x}_i)}}_{\text{Discrimination power}} \cdot \underbrace{e^{-\beta\|V\|}}_{\substack{\text{Sparsity of} \\ \text{decision modes}}} \tag{8}$$

where $P(\mathbf{x}_i)$ denotes the likelihood of $\mathbf{x}_i$ being positive that is estimated by the tree P. β is a scaling parameter[5]. This objective penalizes the decrease of the discriminative power and forces the system to summarize a few generic decision modes for explanation. We compute the likelihood of $\mathbf{x}_i$ being positive as

$$P(\mathbf{x}_i) = e^{\gamma \hat{h}(\mathbf{x}_i)} / \sum_{j\in\mathbf{\Omega}} e^{\gamma \hat{h}(\mathbf{x}_j)} \tag{9}$$

where $\hat{h}(\mathbf{x}_i) = h_{\hat{v}}(\mathbf{x}_i)$ denotes the prediction on $\mathbf{x}_i$ based on best child $\hat{v} \in V$ in the second tree layer. γ is a constant scaling parameter[5].

Algorithm 1 Learning a decision tree for a category

Input: 1. A CNN with disentangled filters, 2. training images $\mathbf{\Omega} = \mathbf{\Omega}^+ \cup \mathbf{\Omega}^-$.
Output: A decision tree.
Initialize a tree $Q = P_0$ and set $t = 0$
for each image I_i, $i \in \mathbf{\Omega}^+$ **do**
  Initialize a child of the root of the initial tree Q by setting $\bar{\mathbf{g}} = \mathbf{g}_i$ based on Equation (3) and $\boldsymbol{\alpha} = \mathbf{1}$.
end for
for $t = t + 1$ **until** $\Delta \log E \leq 0$ **do**
  1. Choose (v, v') in the second tree layer of P_{t-1} that maximize $\Delta \log E$ based on Equation (8)
  2. Merge (v, v') to generate a new node u based on Equations (5) and (6), and obtain the tree P_t.
end for
Assign filters with semantic object parts to obtain $\mathbf{A}$.

v and v' corresponding to children of the root node

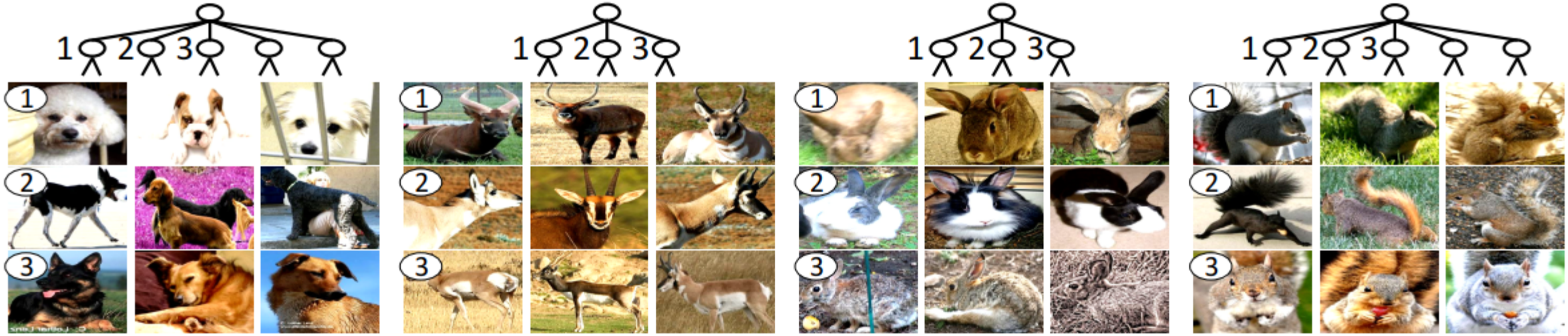


Figure 5. Visualization of decision modes corresponding to nodes in the 2nd tree layer. We show typical images of each decision mode.

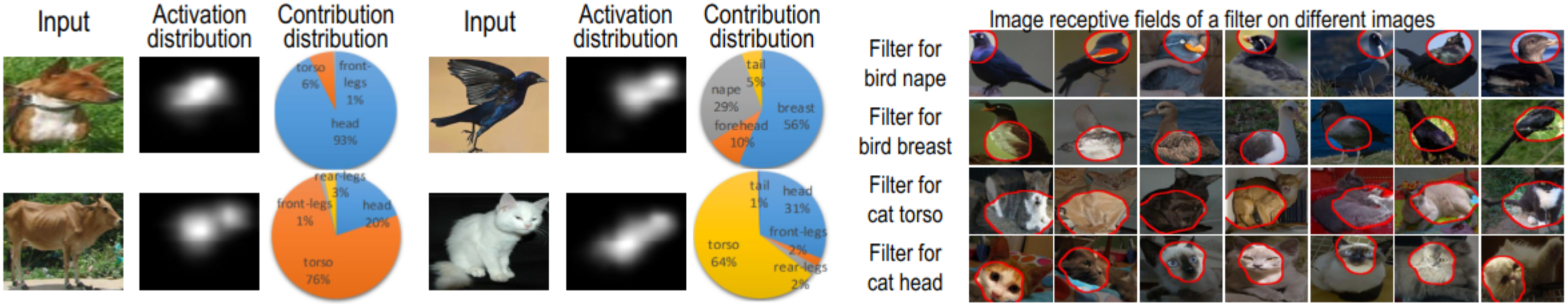


Figure 6. Object-part contributions for CNN prediction. Pie charts show contribution proportions of different parts, which are estimated using nodes in the second tree layer. Heat maps indicate spatial distributions of neural activations in the top conv-layer (note that the heat maps do not represent distributions of "contributions," because neural activations are not weighted by $\mathbf{g}_i$). Right figures show image receptive fields of different filters estimated by [44]. Based on these receptive filters, we assign the filters with different object parts to compute the distribution of object-part contributions.

Evaluation for nodes in different layers

Evaluation of nodes in the *n*-th layer was performed by constructing a new try comprising only the information from the n-th layer and down.

All quantitative results were obtained using just **six classes**, i.e., **six animals** (*bird, cat, cow, dog, horse and sheep*)

Dataset	2nd	5th	10th	50th	100th
ILSVRC Animal-Part	4.8	31.6	69.1	236.5	402.1
VOC Part	3.8	25.7	59.0	219.5	361.5
CUB200-2011	5.0	32.0	64.0	230.0	430.0

Table 1. Average number of nodes in the 2nd, 5th, 10th, 50th, and 100th layer of decision trees learned for VGG-16 nets.

	breast	forehead	nape	tail	average
2nd layer	0.028	0.004	0.013	0.005	0.013
5th layer	0.024	0.004	0.010	0.006	0.011
10th layer	0.022	0.004	0.010	0.005	0.010
50th layer	0.018	0.003	0.008	0.005	0.009
100th layer	0.019	0.003	0.008	0.005	0.009

Table 2. Errors of object-part contributions that were estimated using nodes in the 2nd/5th/10th/50th/100th layer of the decision tree. The CNN was learned using the CUB200 dataset.

Further Evaluation

	Dataset	2nd	5th	10th	50th	100th	leaves
VGG-16	ILSVRC Animal-Part	0.23	0.30	0.36	0.52	0.65	1.00
	VOC Part	0.22	0.30	0.36	0.53	0.67	1.00
	CUB200-2011	0.21	0.26	0.28	0.33	0.37	1.00
VGG-M	VOC Part	0.35	0.38	0.46	0.63	0.78	1.00
	CUB200-2011	0.44	0.44	0.46	0.59	0.63	1.00
VGG-S	VOC Part	0.33	0.35	0.41	0.63	0.80	1.00
	CUB200-2011	0.40	0.40	0.43	0.48	0.52	1.00
AlexNet	VOC Part	0.37	0.38	0.47	0.66	0.82	1.00
	CUB200-2011	0.47	0.47	0.47	0.58	0.66	1.00

Table 3. Average fitness of contribution distributions based on nodes in the 2nd/5th/10th/50th/100th layer and leaf nodes, which reflects the accuracy of the estimated rationale of a prediction.

		Dataset	CNN	2nd	5th	10th	50th	100th	leaves
Classification accuracy	VGG-16	ILSVRC Animal-Part	96.7	94.4	89.0	88.7	88.6	88.7	87.8
		VOC Part	95.4	94.2	91.0	90.1	89.8	89.4	88.2
		CUB200-2011	96.5	91.5	92.2	88.3	88.6	88.9	85.3
	VGG-M	VOC Part	94.2	95.7	94.2	93.1	93.0	92.6	90.8
		CUB200-2011	96.0	97.2	96.8	96.0	95.2	94.9	93.5
	VGG-S	VOC Part	95.5	92.7	92.6	91.3	90.2	88.8	86.1
		CUB200-2011	95.8	95.4	94.9	93.1	93.4	93.6	88.8
	AlexNet	VOC Part	93.9	90.7	88.6	88.6	87.9	86.2	84.1
		CUB200-2011	95.4	94.9	94.2	94.3	92.8	92.0	90.0
Prediction error	VGG-16	ILSVRC Animal-Part	–	0.052	0.064	0.063	0.049	0.034	0.00
		VOC Part	–	0.052	0.066	0.070	0.051	0.035	0.00
		CUB200-2011	–	0.075	0.099	0.101	0.087	0.083	0.00
	VGG-M	VOC Part	–	0.053	0.051	0.051	0.034	0.019	0.00
		CUB200-2011	–	0.036	0.037	0.038	0.035	0.030	0.00
	VGG-S	VOC Part	–	0.047	0.047	0.045	0.035	0.019	0.00
		CUB200-2011	–	0.045	0.046	0.050	0.051	0.038	0.00
	AlexNet	VOC Part	–	0.055	0.058	0.055	0.038	0.020	0.00
		CUB200-2011	–	0.044	0.044	0.045	0.039	0.033	0.00

Table 4. Average classification accuracy and average prediction error based on nodes in the 2nd/5th/10th/50th/100th layer and leaf nodes of the tree. The classification accuracy and the prediction error reflect the CNN knowledge not encoded by the decision tree.

Authors Explanation

- Claim that “decision modes objectively reflected the knowledge hidden inside a CNN”.
  - Supported by results illustrated in Figure 5 & 6.
- “… because fine-grained decision modes are close to the image-specific rationale” decision modes higher in the tree yielded lower error prediction rates.
- This said, the overall classification accuracy was lower using fine-grained modes “because our method is designed to mine common decision modes for objects of a certain category”.

Conclusion and Discussion

- Thus, the decision tree presented can only provide an approximate explanation for CNN predictions.
- The authors provide two reasons for this:
    1. “without accurate object-part annotations to supervise the learning of CNNs, the filter loss can only roughly make each filter represent an object part.” The filter may be activated by unrelated visual concepts in a few exceptional images.
    2. “The decision mode in each node ignores insignificant object-part filters to ensure a sparse representation.”
- Only data from six different animals was used during the group’s study.