CLIP-Dissect: Automatic description of neuron representations in deep vision networks

Tuomas Oikarinen
UC San Diego CSE
toikarinen@ucsd.edu

Tsui-Wei Weng
UC San Diego HDSI
lweng@ucsd.edu

Abstract

In this paper, we propose CLIP-Dissect, a new technique to automatically describe the function of individual hidden neurons inside vision networks. CLIP-Dissect leverages recent advances in multimodal vision/language models to label internal neurons with open-ended concepts without the need for any labeled data or human examples, which are required for existing tools to succeed. We show that CLIP-Dissect provides more accurate descriptions than existing methods for neurons where the ground-truth is available as well as qualitatively good descriptions for hidden layer neurons. In addition, our method is very flexible: it is model agnostic, can easily handle new concepts and can be extended to take advantage of better multimodal models in the future. Finally CLIP-Dissect is computationally efficient and labels all neurons of a layer in a large vision model in tens of minutes.

1 Introduction

Deep neural networks (DNNs) have demonstrated unprecedented performance in various machine learning tasks spanning computer vision, natural language processing and application domains like healthcare and autonomous driving. However, due to their complex structure, it has been challenging to understand why and how DNNs achieve such great success across numerous tasks and domains. Understanding how the trained DNNs operate is essential to trust their deployment in safety-critical tasks like healthcare, and can help reveal important failure cases or biases of a given model.

One way to achieve these goals is inspecting the functionality of individual neurons in the DNNs, which is the focus of our work. This includes methods based on manual inspection (Zhou et al., 2015; Olah et al., 2020; Goh et al., 2021), which can provide high quality explanations and understanding of the network but require large amounts of manual effort. To address this, researchers have developed automated methods to evaluate the functionality of individual neurons, such as Network Dissection (Bau et al., 2017). In (Bau et al., 2017), the authors first created a new dataset named *Broden* with dense labels associated with a pre-determined set of concepts, and then use *Broden* to find neurons whose activation pattern matches with that of a pre-defined concept. In (Mu & Andreas, 2020), the authors further extend Network Dissection to detect more complex concepts that are logical compositions of the concepts in *Broden*. These methods based on Network Dissection can provide accurate labels in some cases but suffer from a few limitations: (1) they require a densely annotated dataset, which is expensive and often time-consuming to collect and may not cover relevant images for all networks; (2) they can only detect concepts from their fixed concept set that is difficult to expand, as new (densely labelled) data is required for each new concept.

To address the above limitations, we propose CLIP-Dissect, a novel method to automatically dissect DNNs with unrestricted concepts *without* the need of any labeled data. Our method is training-free and based on the publicly available Contrastive Language-Image Pre-training (CLIP) model (Radford et al., 2021) to identify the functionality of individual neuron units. We note that a contemporary work (Hernandez et al., 2022) also aims to address these issues and can achieve impressive results in some settings. However their approach is technically very different from ours since they frame the problem as learning to caption the set of most highly activating images for a given neuron, and train a network to do this using imitation learning from human examples, while our method is training-free. This has some advantages and disadvantages over our method which are discussed in A.1.

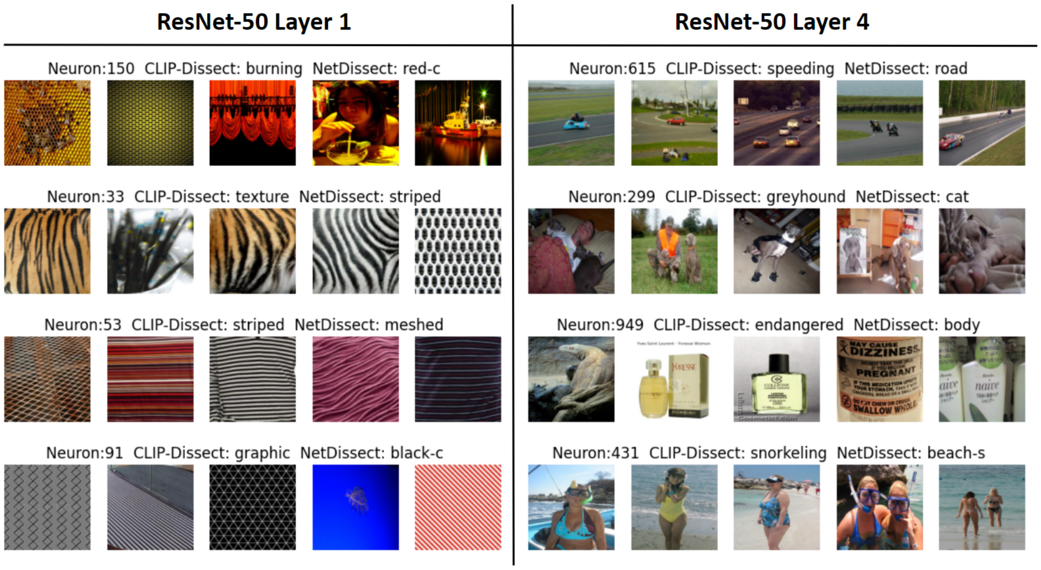
ResNet-50 Layer 1
ResNet-50 Layer 4
Neuron:150 CLIP-Dissect: burning NetDissect: red-c
Neuron:615 CLIP-Dissect: speeding NetDissect: road
Neuron:33 CLIP-Dissect: texture NetDissect: striped
Neuron:299 CLIP-Dissect: greyhound NetDissect: cat
Neuron:53 CLIP-Dissect: striped NetDissect: meshed
Neuron:949 CLIP-Dissect: endangered NetDissect: body
Neuron:91 CLIP-Dissect: graphic NetDissect: black-c
Neuron:431 CLIP-Dissect: snorkeling NetDissect: beach-s

Figure 1: Labels generated by our method and Network Dissection for random neurons of ResNet-50 trained on ImageNet. Displayed together with 5 most highly activating images for that neuron. Following *torchvision* (Marcel & Rodriguez, 2010) naming scheme where layer4 is the second to last layer.

2 Background

Network dissection. Network dissection (Bau et al., 2017) is the first work on automatically understanding DNNs by inspecting the functionality (described as *concepts*) of each individual neuron. At the core of this approach is to reformulate the problem of identifying concepts of intermediate neurons (we follow the convention of describing channels of CNNs as a single ”neuron”) as a task of matching the pattern of neuron activations to the pattern of a pre-defined label mask. They build an auxiliary crowd-labeled dataset $\mathcal{D}_{\text{Broden}}$ with a set of pre-determined concepts c labeled on each pixel of x_i, which provides a ground-truth binary mask $L_c(x_i)$ associated with the concept c, and use such information to compute the intersection over union score (IoU) between a binarized mask M_k from the activations of the concerned neuron unit k over all the images x_i in $\mathcal{D}_{\text{Broden}}$: $\text{IoU}_{k,c} = \frac{\sum_{i\in\mathcal{D}_{\text{Broden}}} M_k(x_i)\cap L_c(x_i)}{\sum_{i\in\mathcal{D}_{\text{Broden}}} M_k(x_i)\cup L_c(x_i)}$. If $\text{IoU}_{k,c} > \eta$, then the neuron k is identified to be detecting concept c. In (Bau et al., 2017), the authors set the threshold η to be 0.04. Note that the binary mask $M_k(x_i)$ are computed via thresholding the spatially scaled activation $S_k(x_i) > \xi$, where ξ is the top 0.5% largest activations for the neuron k and $S_k(x_i)$ has the same resolution as the input image x_i by interpolating the original neuron activations $A_k(x_i)$.

CLIP. CLIP stands for Contrastive Language-Image Pre-training (Radford et al., 2021), an efficient method of learning deep visual representations from natural language supervision. CLIP is designed to address the limitation of static softmax classifiers with a new mechanism to handle *dynamic* output classes. The core idea of CLIP is enable learning from practically unlimited amounts of raw texts and training an image feature extractor (encoder) E_I with a text encoder E_T simultaneously. Given a batch of N (image, text) training examples denoted as $(x_i, t_i)_{i\in[N]}$ pair with $[N]$ defined as the set $\{1, 2, \ldots, N\}$, CLIP aims to increase the similarity of the (x_i, t_i) pair in the embedding space. Let $I_i = E_I(x_i), T_i = E_T(t_i)$, CLIP maximizes the cosine similarity of the (I_i, T_i) in the batch of N pairs while minimizing the cosine similarity of $(I_i, T_j), j \neq i$ using a multi-class N-pair loss (Sohn, 2016; Radford et al., 2021). Once the image encoder E_I and the text encoder E_T are trained, CLIP can perform zero-shot classification for any set of labels: given a test image x_1 we can feed in the natural language names for the set of M labels $\{t_j\}_{j\in[M]}$. The predicted label of x_1 is the label t_k that has the largest cosine similarity among the embedding pairs: (I_1, T_k).

3 Main Method

In this section, we describe CLIP-Dissect, a novel method for automatic, flexible and generalizable label generation for vision networks. An overview of CLIP-Dissect is illustrated in Figure 2.

Inputs & Outputs. There are 3 inputs of the CLIP-Dissect algorithm: (a) DNN to be dissected/probed, denoted as $f(x)$, (b) dataset of DNN inputs for dissecting the DNN, denoted as $\mathcal{D}_{\text{probe}}$, (c) concept set, denoted as $\mathcal{S}$. The output of CLIP-Dissect is the neuron labels, which identify the concept associated with each individual neuron. Compared with Network Dissection (Bau et al., 2017), our goals are the same – we both want to inspect and find concepts associated with each neuron. The input (a) is also the same, we both want to dissect the DNN $f(x)$; however, the inputs (b) and (c) have stark differences. Specifically, our $\mathcal{D}_{\text{probe}}$ does not require any concept labels and thus can be any publicly available dataset such as CIFAR-100, ImageNet, a combination of datasets or unlabeled images collected from the internet. On the other hand. Network Dissection can only use a $\mathcal{D}_{\text{probe}}$ that has been densely labeled with the labels from concept set $\mathcal{S}$. As a results users of Network Dissection are limited to D_{probe} and concept set $\mathcal{S}$ pre-defined in Broden unless they are willing to create their own densely labeled dataset.

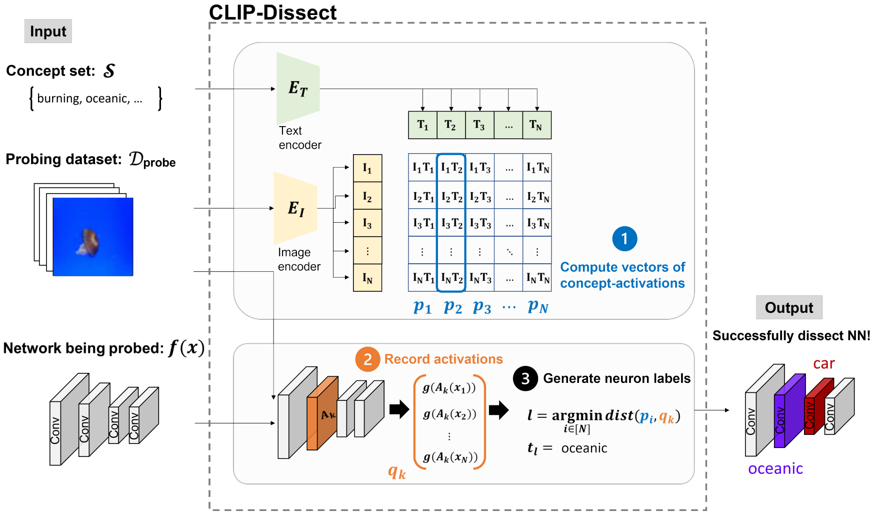
Input
Concept set: $\mathcal{S}$
{ burning, oceanic, ... }
Probing dataset: $\mathcal{D}_{\text{probe}}$
Network being probed: $f(x)$
Conv
Conv
Conv
Conv
CLIP-Dissect
E_T
Text encoder
T_1 T_2 T_3 ... T_N
E_I
Image encoder
I_1 I_2 I_3 ⋮ I_N
I_1T_1 I_1T_2 I_1T_3 ... I_1T_N
I_2T_1 I_2T_2 I_2T_3 ... I_2T_N
I_3T_1 I_3T_2 I_3T_3 ... I_3T_N
I_NT_1 I_NT_2 I_NT_3 ... I_NT_N
p_1 p_2 p_3 ⋯ p_N
1 Compute vectors of concept-activations
2 Record activations
A_k
$g(A_k(x_1))$
$g(A_k(x_2))$
⋮
$g(A_k(x_N))$
q_k
3 Generate neuron labels
$l = \operatorname{argmin}_{i\in[N]} dist(p_i, q_k)$
$t_l =$ oceanic
Output
Successfully dissect NN!
car
Conv
Conv
Conv
Conv
oceanic

Figure 2: Overview of CLIP-Dissect: a 3-step algorithm to dissect neural network of interest.

This is a major limitation of Network Dissection and its follow-up works (Mu & Andreas, 2020). In contrast, the concept set $\mathcal{S}$ and probing dataset $\mathcal{D}_{\text{probe}}$ in our framework are *decoupled*, we can use any text corpus to form the concept set $\mathcal{S}$ and any image dataset independent of $\mathcal{S}$.

CLIP-Dissect Algorithm

1. *Compute the vector of concept-activations.* Using the image encoder E_I and text encoder E_T of a CLIP model, we compute the text embedding T_i of the concepts t_i in the concept set $\mathcal{S}$ and the image embedding I_i of the images x_i in the probing dataset $\mathcal{D}_{\text{probe}}$, then calculate their inner product. Denote the vector of concept-activation for a concept t_m as $\mathbf{p}_m = [I_1T_m, \ldots, I_NT_m]^\top$.
2. *Record activations of target neurons.* Given the neuron unit k, compute the activation $A_k(x_i)$ of the k-th neuron for every image $x_i \in \mathcal{D}_{\text{probe}}$. Define a summary function g, which takes the activation map $A_k(x_i)$ as input and return a real number. Here we let g be the mean function that computes the mean of the activation map over spatial dimensions. We record $g(A_k(x_i))$, for all i, k.
3. *Generate the neuron labels.* Given a concept t_m, compute the distance d_{mk} between the concept-image vector $\mathbf{p}_m$ and the activation vector $\mathbf{q}_k$, where $\mathbf{q}_k = [g(A_k(x_1)), \ldots, g(A_k(x_N))]^\top$ and $d_{mk} = \textit{dist}(\mathbf{p}_m, \mathbf{q}_k)$. The label of neuron k is defined as t_l, where $l = \arg\min_m d_{mk}$.

The distance function *dist* can be any function that compares two vectors, such as cosine similarity or l_p norm. For our results we use a scaled measure of similarity between the rankings of the images, which we found to perform best on labeling final layer neurons of ResNet-50.

4 EXPERIMENTS

In this section we evaluate our method in various ways through analyzing two pre-trained networks: ResNet-50 (He et al., 2016) trained on ImageNet (Deng et al., 2009), and ResNet-18 trained on Places-365 (Zhou et al., 2017). Unless otherwise mentioned we use 20,000 most common English words as the concept set $\mathcal{S}$.

(I) Qualitative results. Figure 1 shows examples of neuron labels generated by CLIP-Dissect for randomly chosen hidden neurons in different layers compared against the label assigned to those same neurons by Network Dissection (Bau et al., 2017). We can see that not every neuron corresponds to a clear concept, but our method can detect low-level concepts on early layers and provide more descriptive labels than Network Dissection in later layers, such as the 'snorkeling' and 'greyhound' neurons. These results use the union of ImageNet validation set and Broden as D_{probe}.

(II) Quantitative results. We also quantitatively compare our methods performance against Network Dissection. We do not compare against Compositional Explanations (Mu & Andreas, 2020) as it is much more computationally expensive, and it is complementary to our approach as their composition could also be applied to our explanations. We also do not compare against MILAN (Hernandez et al., 2022) due to the newness of their method and lack of released code.

The key idea of this experiment is to generate labels for neurons where we have access to ground truth descriptions, i.e. neurons in the final layer of a network, where the ground truth concept is the name of the class that neuron is detecting. This avoids the need for human evaluation and uses real function of the target neurons while human evaluations are usually limited to describing a few most highly activating images, ignoring all other images. In table 1 we can see that the labels generated by our method are closer to ground truth in a sentence embedding space than those of Network Dissection regardless of our choice of D_{probe} or $\mathcal{S}$. In addition our method performs better with larger concept set and D_{probe}. For embeddings we use the CLIP ViT-B/16 text encoder as well as the all-mpnet-base-v2 sentence encoder and measure cosine similarity. We also measure the accuracy of methods by evaluating if they can exactly match the ground truth description in cases where ground-truth is part of the concept set. Table 2 shows that our method outperforms Network Dissection even on a task that is favorable to their method as the Places365 dataset has large overlaps with Broden. We want to highlight that we can reach higher accuracy even though Network Dissection relies on ground truth labels of Broden while ours doesn't use any label information.

(III) Detecting concepts missing from D_{probe}. One surprising ability we found is that our method is able to assign the correct label to a neuron even if D_{probe} does not have any examples corresponding to that concept. For example, CLIP-Dissect was able to assign the correct dog breed to 34 out

Method	D_{probe}	Concept set	CLIP cos	mpnet cos
Network Dissection (baseline)	Broden	Broden	0.6929	0.2952
CLIP-Dissect (Ours)	ImageNet val	Broden	0.7358	0.3930
CLIP-Dissect (Ours)	ImageNet val	3k	0.7353	0.3427
CLIP-Dissect (Ours)	ImageNet val	10k	0.7578	0.4206
CLIP-Dissect (Ours)	ImageNet val	20k	0.7832	0.4901
CLIP-Dissect (Ours)	Broden	20k	0.7476	0.3857
CLIP-Dissect (Ours)	CIFAR train	20k	0.7227	0.3250
CLIP-Dissect(Ours)	ImageNet val + Broden	20k	**0.7866**	**0.5040**
CLIP-Dissect (Ours)	ImageNet val	Imagenet	0.9766	0.9458

Table 1: The cosine similarity of predicted labels compared to ground truth labels on final layer neurons of ResNet-50 trained on ImageNet. The best performing setting is bolded (higher is better).

Method	D_{probe}	Uses gt labels	Text set	Top1 Acc	CLIP cos	mpnet cos
NetworkDissection (Baseline)	Broden	Yes	Broden	43.82%	0.8828	0.6299
CLIP-Dissect (ours)	Broden	No	Broden	**48.69%**	**0.8853**	**0.6493**

Table 2: Performance when labeling final layer neurons of a ResNet18 trained on Places365. Accuracy measured on 267/365 neurons whose label is a directly included in Broden labels.

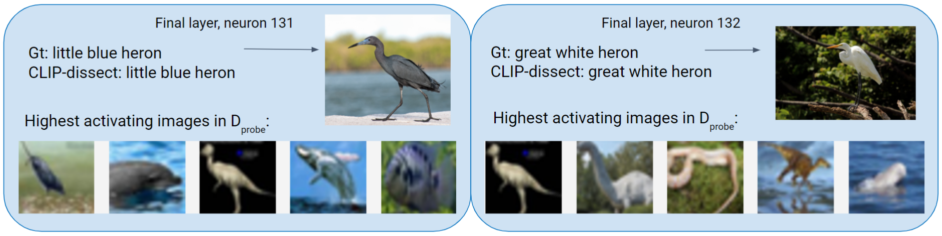
Final layer, neuron 131
Gt: little blue heron
CLIP-dissect: little blue heron
Highest activating images in D_{probe}:
Final layer, neuron 132
Gt: great white heron
CLIP-dissect: great white heron
Highest activating images in D_{probe}:

Figure 3: Example of CLIP-Dissect correctly labeling neurons that detect the little blue heron and the great white heron based on pictures of dolphins and dinosaurs.

of 118 neurons detecting dog breeds, and correct bird species to 11 out of 59 final layer neurons of ResNet-50 trained on ImageNet, using CIFAR-100 training set as D_{probe}, which doesn't include any images of dogs or birds. This is fundamentally impossible for any label based methods (Bau et al., 2017; Mu & Andreas, 2020) (as IoU will be 0 for any concept not in D_{probe}) or methods based on captioning highly activated images (Hernandez et al., 2022) (as humans won't assign a captions missing from images). Example labels and highest activating probe images can be seen in Figure 3.

(IV) Compositional Concepts. So far our method has focused on choosing the most fitting concept from the pre-defined concept set. While changing the concept set in CLIP-Dissect is as easy as editing a text file, we show it can also detect more complex compositional concepts. We experimented with generating explanations by searching over concatenations of two concepts on our concept space. To reduce computational constraints, we only looked at combinations of 25 most accurate single word labels for each neuron. Example results are shown in Fig 4. While the initial results are promising, some challenges remain to make these compositional explanations more computationally efficient and consistent, which is an important direction for future work. In addition to textual composition, our method is compatible with logical composition of (Mu & Andreas, 2020).

5 Conclusions

In this work, we have developed CLIP-Dissect, a novel, flexible and computationally efficient framework for generating automated labels for hidden layer neurons. We also proposed new methods to quantitatively compare neuron labeling methods based on labeling final layer neurons. Importantly, we have shown CLIP-Dissect can outperform previous automated labeling methods both qualitatively and quantitatively and can even detect concepts missing from the probing dataset.

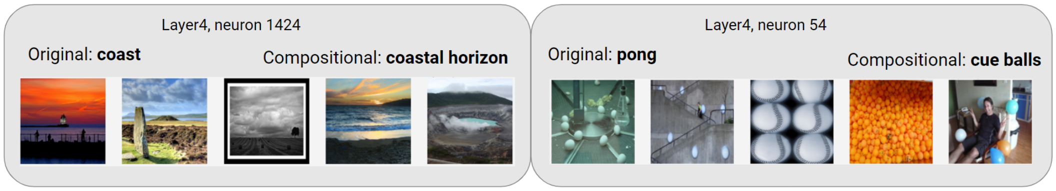

Figure 4: An example of compositional explanations generated by our method for two neurons of ResNet50 trained on ImageNet.

References

David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Computer Vision and Pattern Recognition*, 2017.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Gabriel Goh, Nick Cammarata †, Chelsea Voss †, Shan Carter, Michael Petrov, Ludwig Schubert, Alec Radford, and Chris Olah. Multimodal neurons in artificial neural networks. *Distill*, 2021. doi: 10.23915/distill.00030. https://distill.pub/2021/multimodal-neurons.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Evan Hernandez, Sarah Schwettmann, David Bau, Teona Bagashvili, Antonio Torralba, and Jacob Andreas. Natural language descriptions of deep visual features. *arXiv preprint arXiv:2201.11114*, 2022.

Sébastien Marcel and Yann Rodriguez. Torchvision the machine-vision package of torch. In *Proceedings of the 18th ACM International Conference on Multimedia*, MM '10, pp. 1485–1488, New York, NY, USA, 2010. Association for Computing Machinery. ISBN 9781605589336. doi: 10.1145/1873951.1874254.

Jesse Mu and Jacob Andreas. Compositional explanations of neurons. *Advances in Neural Information Processing Systems*, 33:17153–17163, 2020.

Chris Olah, Nick Cammarata, Ludwig Schubert, Gabriel Goh, Michael Petrov, and Shan Carter. Zoom in: An introduction to circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. https://distill.pub/2020/circuits/zoom-in.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision, 2021.

Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2016.

Bolei Zhou, Aditya Khosla, Àgata Lapedriza, Aude Oliva, and Antonio Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

Bolei Zhou, Agata Lapedriza, Aditya Khosla, Aude Oliva, and Antonio Torralba. Places: A 10 million image database for scene recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

Property / Method	Computationally Efficient	Unrestricted concepts	Does not binarize activations	Describes more than top few images	Uses spatial information of neuron activations	Generative natural language descriptions	Generalizes to different vision tasks
Network Dissection (Bau et al.)	**Yes**	No	No	**Yes**	**Yes**	No	No
Compositional Explanations (Mu & Andreas)	No	No	No	**Yes**	**Yes**	No	No
MILAN (Hernandez et al.)	?	**Yes**	No	No	**Yes**	**Yes**	?
CLIP-Dissect (Ours)	**Yes**	**Yes**	**Yes**	**Yes**	No	No	**Yes**

Table 3: Comparison of existing automated neuron labeling methods and whether they have certain desirable properties.

A Appendix

A.1 Comparing features of automatic neuron labeling methods

Table 3 compares the strengths of different automated neuron labeling methods. The main limitation of our method compared to previous work is that it's not taking advantage of the spatial information of neuron activations. Our results suggest this limitation is not too restrictive, especially on later layers but it likely reduces our performance on earlier layers. We believe this is a reasonable tradeoff to achieve the generalizability and computational efficiency of our method. On the other hand, ours is the only method that deals with scalar values of activations and as such has access to more information than the other methods that lose information by binarizing activations to 0 or 1.

MILAN (Hernandez et al., 2022) has some question marks, as their code has not been released yet and we have no information about its computational efficiency. It is also unclear how well it will generalize to networks performing different vision tasks. (Hernandez et al., 2022) show some evidence that it can generalize from Places to ImageNet, but they also show it struggles to explain concepts not seen in training set. Since their method was only trained on 20k neurons from two tasks, it is unlikely to generalize broadly. On the other hand, our method relies on CLIP which was trained on a broad dataset of 400M images and has been demonstrated to perform well on a very large variety of tasks (Radford et al., 2021).

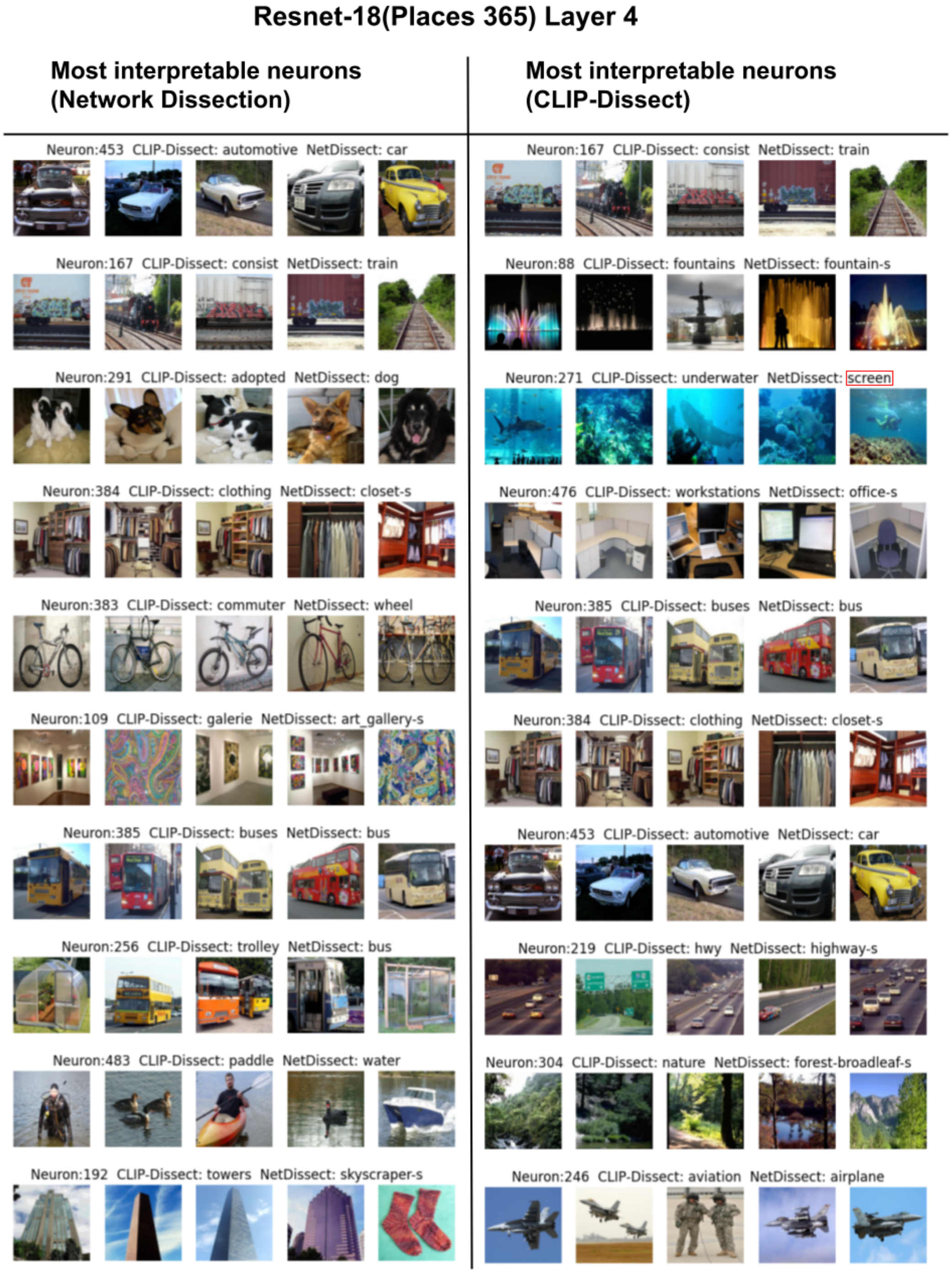
Resnet-18(Places 365) Layer 4
Most interpretable neurons (Network Dissection)
Most interpretable neurons (CLIP-Dissect)
Neuron:453 CLIP-Dissect: automotive NetDissect: car
Neuron:167 CLIP-Dissect: consist NetDissect: train
Neuron:291 CLIP-Dissect: adopted NetDissect: dog
Neuron:384 CLIP-Dissect: clothing NetDissect: closet-s
Neuron:383 CLIP-Dissect: commuter NetDissect: wheel
Neuron:109 CLIP-Dissect: galerie NetDissect: art_gallery-s
Neuron:385 CLIP-Dissect: buses NetDissect: bus
Neuron:256 CLIP-Dissect: trolley NetDissect: bus
Neuron:483 CLIP-Dissect: paddle NetDissect: water
Neuron:192 CLIP-Dissect: towers NetDissect: skyscraper-s
Neuron:167 CLIP-Dissect: consist NetDissect: train
Neuron:88 CLIP-Dissect: fountains NetDissect: fountain-s
Neuron:271 CLIP-Dissect: underwater NetDissect: screen
Neuron:476 CLIP-Dissect: workstations NetDissect: office-s
Neuron:385 CLIP-Dissect: buses NetDissect: bus
Neuron:384 CLIP-Dissect: clothing NetDissect: closet-s
Neuron:453 CLIP-Dissect: automotive NetDissect: car
Neuron:219 CLIP-Dissect: hwy NetDissect: highway-s
Neuron:304 CLIP-Dissect: nature NetDissect: forest-broadleaf-s
Neuron:246 CLIP-Dissect: aviation NetDissect: airplane

Figure 5: Explanations of most interpretable neurons according to both methods in the second to last layer of ResNet-18. Note the noun consist is another word for train. Both methods do a good job on the interpretable neurons except for neuron 271 which Network Dissection mistakes for a screen (highlighted in red).

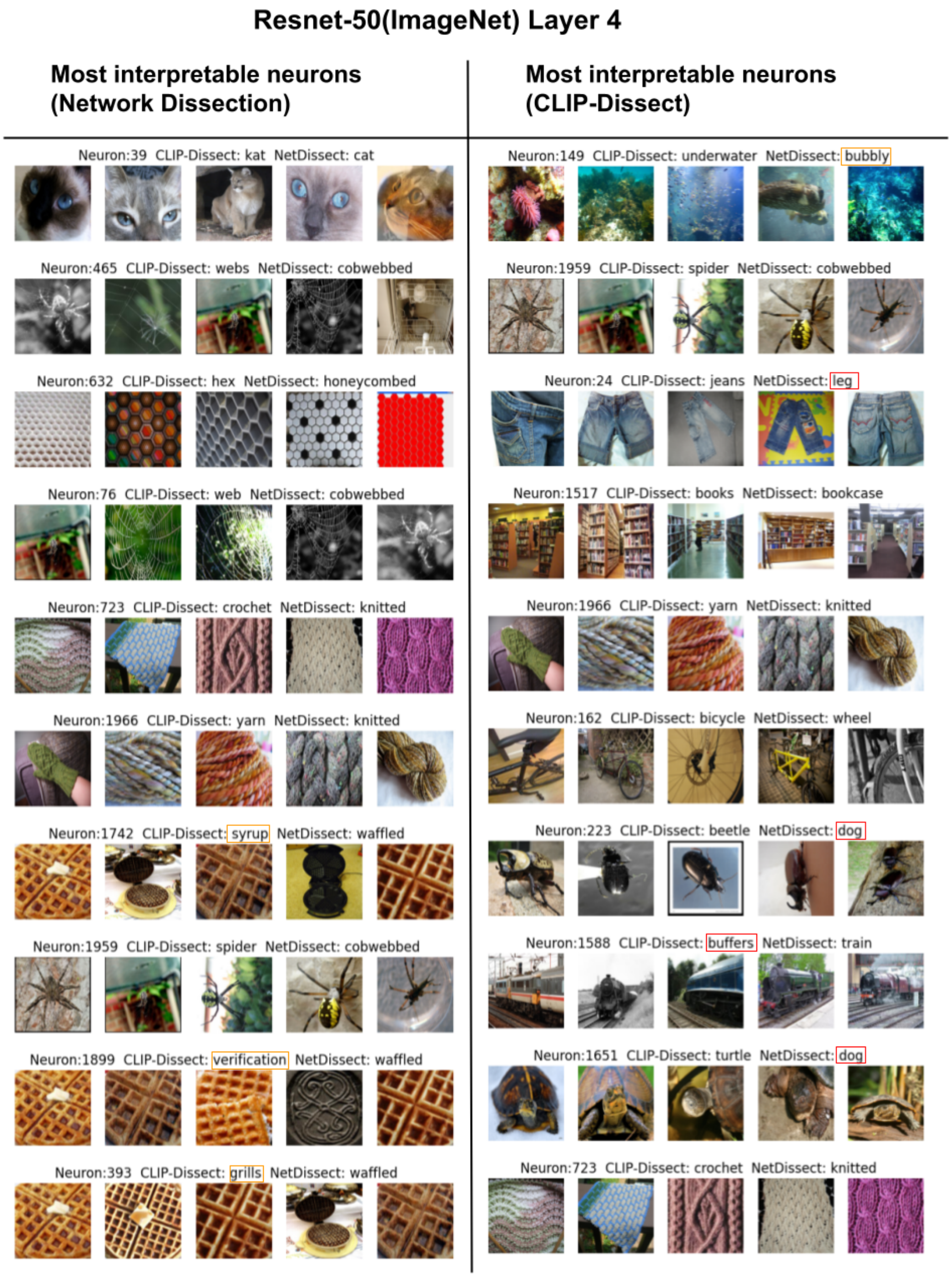
Resnet-50(ImageNet) Layer 4
Most interpretable neurons (Network Dissection)
Most interpretable neurons (CLIP-Dissect)
Neuron:39 CLIP-Dissect: kat NetDissect: cat
Neuron:465 CLIP-Dissect: webs NetDissect: cobwebbed
Neuron:632 CLIP-Dissect: hex NetDissect: honeycombed
Neuron:76 CLIP-Dissect: web NetDissect: cobwebbed
Neuron:723 CLIP-Dissect: crochet NetDissect: knitted
Neuron:1966 CLIP-Dissect: yarn NetDissect: knitted
Neuron:1742 CLIP-Dissect: syrup NetDissect: waffled
Neuron:1959 CLIP-Dissect: spider NetDissect: cobwebbed
Neuron:1899 CLIP-Dissect: verification NetDissect: waffled
Neuron:393 CLIP-Dissect: grills NetDissect: waffled
Neuron:149 CLIP-Dissect: underwater NetDissect: bubbly
Neuron:1959 CLIP-Dissect: spider NetDissect: cobwebbed
Neuron:24 CLIP-Dissect: jeans NetDissect: leg
Neuron:1517 CLIP-Dissect: books NetDissect: bookcase
Neuron:1966 CLIP-Dissect: yarn NetDissect: knitted
Neuron:162 CLIP-Dissect: bicycle NetDissect: wheel
Neuron:223 CLIP-Dissect: beetle NetDissect: dog
Neuron:1588 CLIP-Dissect: buffers NetDissect: train
Neuron:1651 CLIP-Dissect: turtle NetDissect: dog
Neuron:723 CLIP-Dissect: crochet NetDissect: knitted

Figure 6: Explanations of most interpretable neurons according to both methods in the second to last layer of ResNet-50. Both methods do a pretty well again, but ours seems to be more precise. We have highlighted descriptions that seem slightly erroneous in orange and clear errors in red.