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.
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 $D_{Broden}$ 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 $D_{Broden}$:

$$\text{IoU}_{k,c} = \frac{\sum_{x_i \in D_{Broden}} M_k(x_i) \cap L_c(x_i)}{\sum_{x_i \in D_{Broden}} M_k(x_i) \cup L_c(x_i)}.$$  

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 $\eta$ to be 0.04. Note that the binary mask $M_k(x_i)$ are computed via thresholding the spatially scaled activation $S_k(x_i)$, where $S_k(x_i)$ 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 $I_i = E_I(x_i), T_j = E_T(t_j)$. CLIP maximizes the cosine similarity of the $(I_i, T_j)$ 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 \( D_{\text{probe}} \), (c) concept set, denoted as \( S \). 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 \( D_{\text{probe}} \) 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 \( D_{\text{probe}} \) that has been densely labeled with the labels from concept set \( S \). As a results users of Network Dissection are limited to \( D_{\text{probe}} \) and concept set \( S \) pre-defined in Broden unless they are willing to create their own densely labeled dataset.

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 \( S \) and probing dataset \( D_{\text{probe}} \) in our framework are decoupled, we can use any text corpus to form the concept set \( S \) and any image dataset independent of \( S \).

**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 \( S \) and the image embedding \( I_{i} \) of the images \( x_i \) in the probing dataset \( D_{\text{probe}} \), then calculate their inner product. Denote the vector of concept-activation for a concept \( t_m \) as \( p_m = [I_1T_m, \ldots, I_NT_m]^\top \).

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 \( x_i \in D_{\text{probe}} \). 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 \( p_m \) and the activation vector \( q_k \), where \( q_k = [g(A_k(x_1)), \ldots, g(A_k(x_N))]^\top \) and \( d_{mk} = \text{dist}(p_m, q_k) \). The label of neuron \( k \) is defined as \( t_l \), where \( l = \arg\min_m d_{mk} \).
The distance function \( \text{dist} \) 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 (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 \( D_{\text{probe}} \).

**II (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 \( D_{\text{probe}} \) or \( \mathcal{S} \). In addition our method performs better with larger concept set and \( D_{\text{probe}} \). 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 (III) Detecting concepts missing from \( D_{\text{probe}} \).** One surprising ability we found is that our method is able to assign the correct label to a neuron even if \( D_{\text{probe}} \) does not have any examples corresponding to that concept. For example, CLIP-Dissect was able to assign the correct dog breed to 34 out of 100

<table>
<thead>
<tr>
<th>Method</th>
<th>( D_{\text{probe}} )</th>
<th>Concept set</th>
<th>CLIP cos</th>
<th>mpnet cos</th>
</tr>
</thead>
<tbody>
<tr>
<td>Network Dissection (baseline)</td>
<td>Broden</td>
<td>Broden</td>
<td>0.6929</td>
<td>0.2952</td>
</tr>
<tr>
<td>CLIP-Dissect (Ours)</td>
<td>ImageNet val</td>
<td>Broden</td>
<td>0.7358</td>
<td>0.3930</td>
</tr>
<tr>
<td>CLIP-Dissect (Ours)</td>
<td>ImageNet val</td>
<td>3k</td>
<td>0.7353</td>
<td>0.3427</td>
</tr>
<tr>
<td>CLIP-Dissect (Ours)</td>
<td>ImageNet val</td>
<td>10k</td>
<td>0.7578</td>
<td>0.4206</td>
</tr>
<tr>
<td>CLIP-Dissect (Ours)</td>
<td>ImageNet val</td>
<td>20k</td>
<td>0.7832</td>
<td>0.4901</td>
</tr>
<tr>
<td>CLIP-Dissect (Ours)</td>
<td>Broden</td>
<td>20k</td>
<td>0.7476</td>
<td>0.3857</td>
</tr>
<tr>
<td>CLIP-Dissect (Ours)</td>
<td>CIFAR train</td>
<td>20k</td>
<td>0.7227</td>
<td>0.3250</td>
</tr>
<tr>
<td>CLIP-Dissect (Ours)</td>
<td>ImageNet val + Broden</td>
<td>20k</td>
<td><strong>0.7866</strong></td>
<td><strong>0.5040</strong></td>
</tr>
<tr>
<td>CLIP-Dissect (Ours)</td>
<td>ImageNet val</td>
<td>Imagenet</td>
<td>0.9766</td>
<td>0.9458</td>
</tr>
</tbody>
</table>

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).
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.

<table>
<thead>
<tr>
<th>Method</th>
<th>$D_{probe}$</th>
<th>Uses gt labels</th>
<th>Text set</th>
<th>Top1 Acc</th>
<th>CLIP cos</th>
<th>mpnet cos</th>
</tr>
</thead>
<tbody>
<tr>
<td>NetworkDissection (Baseline)</td>
<td>Broden</td>
<td>Yes</td>
<td>Broden</td>
<td>43.82%</td>
<td>0.8828</td>
<td>0.6299</td>
</tr>
<tr>
<td>CLIP-Dissect (ours)</td>
<td>Broden</td>
<td>No</td>
<td>Broden</td>
<td>48.69%</td>
<td>0.8853</td>
<td>0.6493</td>
</tr>
</tbody>
</table>

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


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

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

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).
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).
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.