Visualizing Intermediate Neurons of Convolutional Neural Networks via CLIP-Dissect

Evan Luo, Atri Pandya, Alex Battikha
REHS program, San Diego Supercomputer Center, UC San Diego, CA, USA
{eluo2015, atri.pandya1, alex.battikha} @gmail.com
Correspondence to: lweng@ucsd.edu

Abstract

In this summer project, we perform an analysis of the intermediate layer neurons primarily through the use of CLIP-dissect, a powerful model to describe neurons, and its Soft-WPMI similarity function. Specifically, we first validate the relative accuracy of CLIP-dissect when analyzing intermediate layer neurons using ground-truth labeling, before using the large pretrained model and the Soft-WPMI similarities to perform analysis of convolutional neural networks (CNNs) such as ResNet-50. We then create an interactive GUI with the visualizations of model, layer, and neuron-level analysis to allow anyone to access the intermediate neurons of these CNNs using output from CLIP-dissect and images from Broden. We allow users to directly search for specific concepts within any network so that they can better understand the inner workings of the models that increasingly define our daily life. Our website with an interactive version of this project can be found at https://dr4nx.github.io/clip-search/index.html.

1 Introduction

In this project, we evaluate, describe, and visualize intermediate layers of deep neural networks such as ResNet-50 (He et al., 2015) using CLIP-dissect (Oikarinen & Weng, 2022). It aims to provide an improved understanding of the inner workings of the ResNet-50 model by allowing users to observe its behavior and investigate how it makes choices. ResNet-50 is a powerful deep convolutional neural network that was trained on large-scale datasets to perform image classification. It is well known for its innovative use of skip connections. We primarily utilize CLIP-Dissect for Neuron Visualization in our analysis, using it to understand the neurons within the ResNet-50 model to observe what images they respond to. This helps us in determining whether patterns or concepts are significant to individual neurons, and if so, which ones.

Our primary goal for this project was first, to verify the accuracy of CLIP-dissect's results, and second, to use the mentioned results to create visualizations of ResNet-50 to understand the complexity and concepts used in each of the different layers. In this report, we will explain our methods of doing so show the accuracy of CLIP-dissect, and then use its results to determine neuron interpretability and level of complexity.

2 Background and Definitions

2.1 Background

CLIP: CLIP stands for Contrastive Language-Image Pre-training is an efficient method of learning visual representations from natural language supervision. CLIP is built to learn from the practically unrestricted amount of image and text pairs by training an image encoder E_I and text encoder E_T simultaneously, then given a batch of N image x_i and text t_i training example pairs denoted as $(x_i, t_i)_{i\in[N]}$ with $[N]$ defined as the set $1, 2, \cdots, N$, CLIP aims to increase the similarity of the (x_i, t_i) pair. It does this by encoding the x_i and t_i with E_I and E_T to create I_i and T_i, then maximizes cosine similarity of the (I_i, T_i) in the batch of N pairs while minimizing the cosine similarity of (I_i, T_j), $j \neq i$ 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 a_1, we can feed in the natural language names for a set of M labels. The predicted label of a_1 is the label t_k that has the largest cosine similarity among the embedding pairs (I_1, T_k). CLIP plays a key part in both CLIP-dissect and our own analysis in CLIP-avg (described in section 4.3).

CLIP-dissect: CLIP-dissect is a technique used to automatically describe the function of individual hidden neurons inside of vision networks. It is extremely flexible, adapting to any number of concepts and images. It works in three steps. First, it uses image and text encoders E_I and E_T from a CLIP model to find the text embedding T_i of the concepts t_i in the concept set S and image embedding I_i of the images x_i in the probing dataset D_{probe}. It then computes concept-activation matrix $P \in \mathbb{R}^{N\times M}$ where element (i, j) is $I_i \cdot T_j$. After computing this concept matrix, CLIP-dissect then calculates the activation map $A_k(x_i)$ for target neuron k for every image x_i in D_{probe}, then defines a summary function g to create an activation vector q_k for neuron k where $q_k = [g(A_k(x_1)), g(A_k(x_2)), \cdots, g(A_k(x_N))]^\top \in \mathbb{R}^N$. Then, using a similarity function sim, with Soft-WPMI similarity as defined in section 2.2.1 used at all points in our paper, defined by $\text{sim}(t_m, q_k; P)$ where the label of neuron k is the label that gives the highest similarity or, l where $l = \arg\max_m \text{sim}(t_m, q_k; P)$. For this paper, we will reference all CLIP-dissect similarities using $\text{sim}(t_m, q_k; P)$.

SAM: SAM is an image segmentation model that can generate segmentation masks for a wide range of input prompts and has zero-shot transfer abilities across a wide range of tasks and datasets. It was trained on 11 million photos and over 1 billion masks. It has three main modules: an image encoder, a prompt encoder, and a mask decoder. The image encoder generates a single image embedding, whilst separate prompt encoding modules are specially intended for the effective encoding of various prompt types. A lightweight decoder may then build segmentation masks with amazing speed and quality by combining image embedding with quick encodings.

Grounding DINO: The goal of the Grounding DINO model is to provide a robust framework for unspecific object recognition using natural language inputs, often known as open-set object detection. The DINO model is based on a single-decoder-dual-encoder architecture. It consists of an image backbone that extracts image features, a text backbone that extracts text features, a feature enhancer that combines image and text features, a language-guided module for query selection, and a cross-modality decoder that refines boxes.

Broden: The Broden dataset is a compilation of many heavily labeled image data sets, including ADE, Open-Surfaces, Pascal-Context, Pascal-Part, and the Describable Textures Dataset. Each image in the collection comprises a visual idea that is labeled using a pixel-by-pixel binary segmentation map. It includes 63,305 images and 1197 visual concepts. Concepts are classified into six categories: textures, colors, materials, parts, objects, and scenes.

2.2 Definitions

In our report, we mention two different types of similarity functions; Cosine similarity as defined by CLIP, and Soft-WPMI similarity as defined in CLIP-dissect. We will briefly explain the definitions and our usage here.

2.2.1 Soft-WPMI Similarity (CLIP-dissect)

Soft-WPMI similarity is the default similarity function used by CLIP-dissect. Soft-WPMI is a more flexible version of WPMI (Weighted Pointwise Mutual Information) (Wang et al., 2020). The equation for this similarity is defined as:

$$\text{sim}(t_m, q_k; P) \triangleq \text{soft\_wpmi}(t_m, q_k) = \log\mathbb{E}[p(t_m|B_k)] - \lambda\log p(t_m)$$

where $p(t_m|B_k)$ represents the probability that every image in the image set B_k has the specific concept t_m, and $p(t_m)$ represents the probability that a fully random set of images B will be described by the same concept. To elaborate, $\mathbb{E}[p(t_m|B_k)] = \prod_{x \in D_{probe}}[1 + p(x \in B_k)(p(t_m|x) - 1)$. This effectively checks every image in D_{probe} to see if it is a part of the specified image set (which in CLIP-dissect is the top-100 activating images), and if so it checks the probability that the image matches the particular concept based on the concept matrix. λ is a hyperparameter. More details about this similarity function and how it is directly applied can be found in the CLIP-dissect paper (Oikarinen & Weng, 2022).

We use the output of this similarity function in later sections as a metric of CLIP-dissect's "confidence" in its labeling. In particular, we analyze the individual and layer-wide implications of this metric, and its potential usage to determine whether neurons are interpretable or not.

2.2.2 Cosine Similarity (CLIP)

Cosine similarity is defined as the cosine of the angle between two vectors and is used to calculate the distance between two points in the plane. The cosine similarity metric is based purely on cosine principles, and as distance increases, the similarity of data points decreases. It's a value with a constrained range of 0 to 1. The cosine of the angle between the two non-zero vectors is used to calculate similarity. A cosine similarity is a number between 0 and 1. The closer the value is to 0, the more orthogonal or perpendicular the two vectors are to one another. When the value is closer to one, the angle is less and the images are more similar. During the Contrastive Pre-Training phase of CLIP, a batch of 32,768 combinations of image and text are simultaneously sent through the text and image encoders to create vector representations of the text and the related image, respectively. The training is carried out by looking for the closest text representation for every image over the full batch, which corresponds to maximizing cosine similarity between the actual N pairings that are the most similar. It also distances the real images from the other texts by minimizing their cosine similarity. The cosine formula is defined as follows:

$$\cos(\theta) = \frac{I_i \cdot T_j}{||I_i|| \cdot ||T_j||}$$

where I_i and T_j are the vector representations (created by their respective encoders) of image x_i and text concept t_j.

3 Ground-truth Label Derivation

In order to test all of our neuron labeling methods, we began by manually labeling 3840 intermediate neurons in ResNet-50, found through Layers 1-4 as defined by *torchvision* (Paszke et al., 2017) trained on ImageNet (Deng et al., 2009) that were available to us through the use of CLIP-dissect.

To perform this ground-truth labeling, we used the Broden dataset (Bau et al., 2020) as D_{probe} and the top twenty-thousand words[1] in the English language as the Concept Set S, and used these as parameters for a CLIP-dissect model. A full set of our input parameters for CLIP-dissect can be found in our Github Repository linked in the abstract.

Below, we outline the complete set of steps we used to derive our ground-truth labels:

[1]Source: https://github.com/first20hours/google-10000-english/blob/master/20k.txt

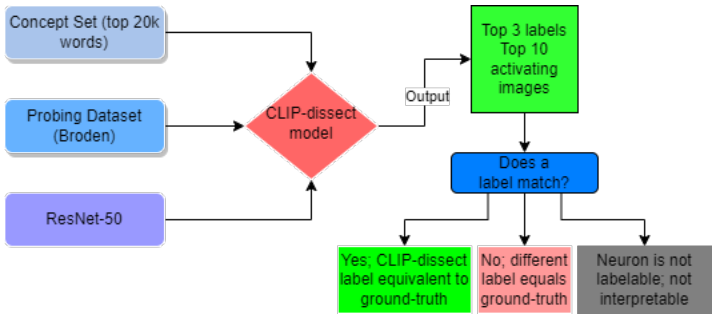

Figure 1: Overview of Ground-Truth Derivation process for all layers of ResNet-50

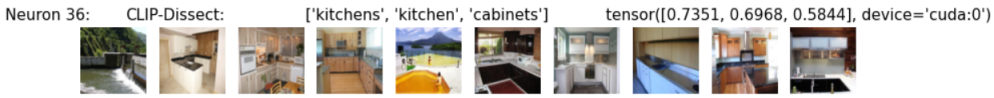

Figure 2: Example of a neuron where CLIP-dissect accurately describes the top-10 activating images

1. Generate a CLIP-dissect model of layers 1-4 of ResNet 50, using the Broden dataset as D_{probe} and the top 20000 English words as the concept set S. We use a Vit-B/16 CLIP model.
2. For each neuron in each layer, we have CLIP-dissect output the top-10 highest activating images, the top-3 labels by Soft-WPMI similarity, and the corresponding value.
3. If we find the majority of activating images sufficiently match a CLIP-dissect label, we wrote that label to be a ground truth.
4. Otherwise, if the majority of images matched a description not provided by CLIP-dissect, we would write that description in as long as it was contained in the top 20000 English words.
5. Finally, if there was no common pattern among the majority of images, we would simply label the neuron as non-interpretable. The number of these neurons may be found in Table 1.

You can find a simplified illustration of this derivation in Figure 1.

The results and code used for this derivation can be found in the Github Repository in the abstract.

An example for each scenario from layer 4 as described in the steps above is shown:

1. Figure 2 illustrates an example where CLIP-dissect is accurate
2. Figure 3 illustrates an example where CLIP-dissect is wrong, but a different concept is correct.
3. Figure 4 illustrates a neuron with no derivable concept.

4 Computing Neuron Labels

4.1 SAM-label

The Segmentation Anything Model (SAM) (Kirillov et al., 2023) alongside Meta AI's Grounding Self-Distillation with No Labels (DINO) (Liu et al., 2023), facilitates unsupervised

Figure 3: Example of a neuron where CLIP-dissect does not accurately describe the top-10 activating images. In this image, it is most likely detecting the concept stripes.

Figure 4: Example of a neuron where CLIP-dissect is incorrect and no concept can be derived

object identification and segmentation. By extracting the top 10 highly activated images from the ResNet-50 intermediate layer neurons, Grounding DINO identifies objects within these images based on a series of class concepts provided. The model then incorporates this information into the segmentation process, accurately segmenting the objects and assigning corresponding labels based on Grounding DINO's predictions. Although this approach is highly efficient with common concepts, after thorough experimentation, we determined that Grounding DINO's computational complexity provides non-interpretable labeling data for large class lists.

We have provided a brief overview of our pipeline for SAM and Grounding DINO pipeline in Figure 5.

4.2 Broden-label

The Broden (Broadly and Densely Labeled) dataset is the primary dataset we have been using for D_{probe}. It was assembled for the purposes of training and validating the Net-Dissect model Bau et al. (2020).

We use this model not only as D_{probe} for most of our experiments, but we also use it to extract potential neuron labels. In order to do so, we first derived the top-10 highest activating images from ResNet-50 intermediate layer neurons. Since all of these images naturally come from the Broden dataset, as it is our D_{probe}, we then extract the file paths and corresponding densely and broadly annotated labels for each highly activating image. When this is complete, we compile the labels of all ten images into one full dataset and normalize the results.

There were a few issues with this approach. One that was difficult to resolve is that the concept set of the Broden dataset does not match the concept set of our ground-truth labeling, which consisted of words from the top 20000 most common words in the English language, while the Broden dataset uses 1197 specified labels.

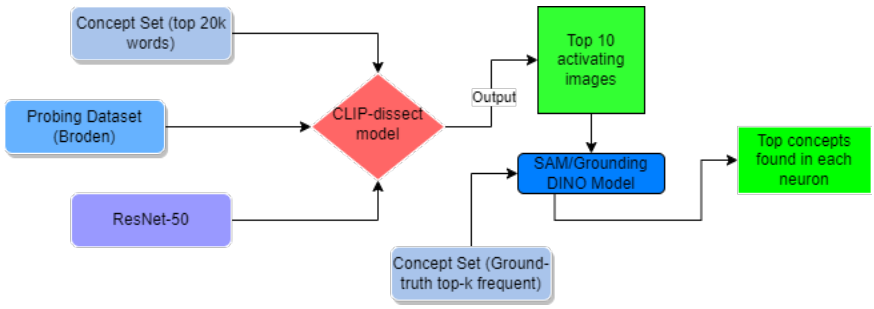

Figure 5: Overview of SAM/Grounding-DINO process for all layers of ResNet-50

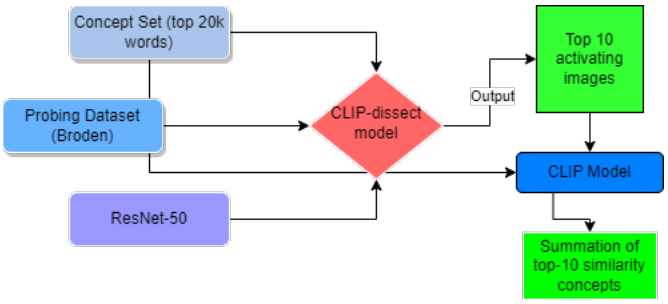

Figure 6: Overview of CLIP process for all layers of ResNet-50

Further, the Broden dataset consists of both broad (per-image) and dense (per-pixel) labels, each of which has different properties. Therefore, while it is possible to calculate the average within each of the different categories, it is difficult to measure the impact of each label.

Indeed, while we did extract the top 3 labels for all 3840 intermediate neurons, it was not possible to fairly compare our results quantitatively from this extraction to either our ground-truth labeling or CLIP-dissect output. Thus, we mostly use Broden for a qualitative analysis of our results.

4.3 CLIP-avg-label

In order to derive its labels, CLIP-dissect uses a CLIP model to create a concept-activation matrix. From this concept-activation matrix, it then calculates the cosine similarity between the activation vector and concepts (note this is different from the Soft-WPMI similarity used in CLIP-dissect). For our setup, we decided to try a simpler method.

Essentially, we took the top-k activating images (which are separate from CLIP, as shown in section 2.1) and ran CLIP on each of them to get the specific top-10 labels it would provide. Then, similar to what we did for the Broden set, we would add the cosine similarities for each particular label together to come up with a compilation result of added cosine similarities.

We used a more powerful CLIP model (ViT-B/32). We will call our method CLIP-avg.

As you will see in Section 5.2, we noted that CLIP-avg did not match our ground-truth results as well as CLIP-dissect. This suggests that the CLIP-dissect Soft-WPMI similarity function is more accurate than our method for deriving labels.

A brief overview of our process for labeling neurons using CLIP can be found in Figure 6.

5 Results

In this section, we provide both qualitative and quantitative results of our experimentation with CLIP-dissect, CLIP-avg, Broden, and SAM/Grounding-DINO.

In Section 5.1 we will introduce observed results from all three methods described in Section 4, and their specific successes and failures. We find that SAM and Broden provide either incorrect or non-useful labels for our purposes, while CLIP and CLIP-dissect provide relatable information that we use in quantitative analysis.

In Section 5.2 we will be discussing the accuracy of the CLIP-dissect model, and the CLIP neuron label computation described in Section 4.3. In that section, we find that for interpretable neurons specifically, CLIP-dissect is relatively accurate.

In Section 5.3, we will discuss the interpretability of neurons and results derived from ground-truth labeling, CLIP-dissect, and our CLIP method. Each of these methods paints a different picture of how interpretable neurons are, which we will be discussing the implications of.

Table 1: Quantitative Accuracy

Type	Layer 1	Layer 2	Layer 3	Layer 4
CLIP-dissect top-1 correct (neurons)	131	225	430	1050
CLIP-dissect top-3 correct (neurons)	151	269	495	1483
CLIP-dissect top-5 correct (neurons)	156	277	516	1509
CLIP-dissect incorrect (neurons)	66	173	271	180
CLIP-avg top-1 correct (neurons)	30	57	140	259
CLIP-avg top-3 correct (neurons)	113	183	290	500
CLIP-avg top-5 correct (neurons)	142	234	359	653
CLIP-avg incorrect (neurons)	80	216	428	1036
Non-interpretable (neurons)	34	62	237	359
Total (neurons)	256	512	1024	2048
CLIP-dissect top-5 accuracy/interpretable (%)	70.27%	61.56%	65.57%	90.34%
CLIP-avg top-5 accuracy/interpretable (%)	63.96%	52.00%	45.62%	38.66%

5.1 Qualitative Analysis

In general, we noted that the quality of SAM labels was not useful. When we presented the 20k dataset to SAM, we noticed it was unable to come up with any meaningful labels. We further reduced this amount of classes to 3k, then the number of classes that existed within our ground-truth labeling, neither of which improved performance. Therefore, we were unable to use SAM labeling to derive results.

Further, we observed an inability to directly compare CLIP-dissect results with Broden labels. As mentioned in Section 4.2, Broden labels come from a different set than our concept set S. Further study can be conducted on the results when these Broden labels are set to be S.

Our best comparison to our CLIP-dissect result came from our CLIP-avg model. These two methods are quantitatively compared in Section 5.2, however, we also noted that our CLIP result often did not match our CLIP-dissect result as it trended towards higher-level concepts. For lower-level concepts, CLIP-avg would often provide similar concepts as CLIP-dissect. This result is proven quantitatively as shown in Table 1.

5.2 Quantitative Accuracy

In order to determine the quantitative accuracy of CLIP-dissect and our CLIP-avg model, we first came up with ground-truth labels as described in Section 3. Unfortunately, in this scenario, it is not possible to avoid human evaluation, and thus it was used. Then, we ran CLIP-dissect using Soft-WPMI similarity $\text{sim}(t_m, q_k; P)$, using Broden as D_{probe}, and the top 20000 English words as concept set S and compared the top-1, 3, and 5 labels based on $\text{sim}(t_m, q_k; P)$ to our ground-truth.

Since a large portion of our ground-truth labels came from the top-3 labels provided by CLIP-dissect, it should be noted that there is a very small increase between top-3 and top-5 accuracy (that is, the correctness of at least one of the top-3 labels vs. top-5 labels), while there is a larger gap between top-1 and top-3. It, therefore, can be suggested that CLIP-dissect either very accurately matches concepts, or completely misses the common concept in the images.

Unlike analysis for the fully connected layer, where each neuron corresponds to a specific subject, there is no specific "ground truth" for every neuron in the intermediate neuron.

Therefore, we used CLIP-dissect labels to motivate our ground-truth labeling since we believe that if CLIP-dissect accurately describes the set of top-10 highly activating images, it is reasonable to say that its label matches a perceived ground-truth. Full results are displayed

Table 2: Soft-WPMI Similarity vs. Correctness

	max $\text{sim}(t_m, q_k; P)$ Average			
Correctness	Layer 1	Layer 2	Layer 3	Layer 4
Top-1 correct (avg)	0.2119	0.1992	0.2572	0.2969
Top-3 correct (avg)	0.2161	0.2014	0.2623	0.2965
Top-5 correct (avg)	0.2092	0.1956	0.2516	0.2914
Incorrect or non-interpretable (avg)	0.1411	0.1397	0.1656	0.1914
Total Average	0.1826	0.1700	0.2089	0.2651

in Table 1, where top-5 accuracy/interpretable is defined as the percentage of neurons that are ground-truth labeled as interpretable where CLIP-dissect is able to acquire the correct label in at least one of the top 5 labels as defined by $\text{sim}(t_m, q_k; P)$.

Interestingly, we note our CLIP-avg model has a comparable result to CLIP-dissect for Layers 1 and 2, but falls off towards Layers 3 and 4. Noticeably, while CLIP-dissect experiences a noticeable spike in accuracy on Layer 4, CLIP-avg's performance suffers in comparison. We believe this to be a result of CLIP-avg's inability to notice a pattern of specific high-level concepts since each image is labeled separately in our method while the activations our combined in CLIP-dissect.

In Table 1, we observed that CLIP-dissect has a top-5 accuracy averaging around 65% for layers 1-3, with the accuracy spiking for layer 4. This suggests that CLIP-dissect is best able to recognize higher-level concepts that activate more often in layer 4 (i.e. dogs and kitchens) better than the lower-level concepts that activate in the first few layers (i.e. stripes and checkers). This corresponds to a similar spike in average similarity $\text{sim}(t_m, q_k; P)$ as described in section 5.3.

5.3 Interpretability

One important facet of our study is the analysis of the human interpretability of these intermediate-layer neurons. As mentioned in Section 3, we labeled neurons we could not derive a pattern for as non-interpretable. We wanted to see if the Soft-WPMI similarity ($\text{sim}(t_m, q_k; P)$) we acquire from CLIP-dissect labels correlated to its accuracy. To do this, we took the average of the Soft-WPMI similarities of all neurons in each layer, and further took the average of the Soft-WPMI similarities for neurons that were predicted correctly, with top-3 accuracy, top-5 accuracy, and those that were either incorrectly predicted or were not interpretable. The results of this experiment can be found in Table 2.

We can note that the neurons that CLIP-dissect predicts correctly have a higher similarity average than those that it predicts incorrectly as well as the total average. Thus, we can conclude that the CLIP-dissect $\text{sim}(t_m, q_k; P)$ metric is an effective tool for measuring the potential accuracy of a particular neuron label. We also note a spike in average similarity for Layer 4 compared to the other 3 layers, which corresponds to our result in Section 5.2.

In order to better gauge the spread of the interpretability of the various neurons using the Soft-WPMI similarities, we create Table 3 and Figures 8, 9, 10, and 11 to illustrate the relative interpretability of a layer as a whole.

From these results, we note that similarities are skewed right, suggesting the majority of neurons have low interpretability; indeed, this matches our observations during ground-truth labeling. While many neurons indeed had a specific concept in the majority of highly activating images, it should be noted there was no distinction made as to how many images had a particular concept. In several ground-truth labels, there were seemingly random images that did not match the overall pattern, as seen in Figure 7.

Table 3: Interpretability as Result of Soft-WPMI Similarity

Interpretability	Layer 1	Layer 2	Layer 3	Layer 4
High Interpretability (0.3-0.99 similarity)(Neurons)	32	38	157	628
Medium Interpretability (0.2-0.3 similarity)(Neurons)	44	100	320	728
Low Interpretability (0.1-0.2 similarity)(Neurons)	143	308	481	652
Non-interpretable (0-0.1 similarity)(Neurons)	37	66	66	40
Total	256	512	1024	2048

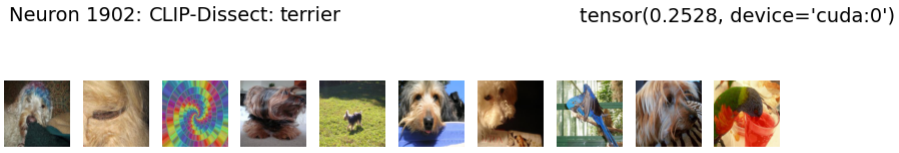

Figure 7: Displays neuron 1902 of layer 4. The broad concept is a terrier, but there are random images (i.e. spiral and drink) sprinkled in.

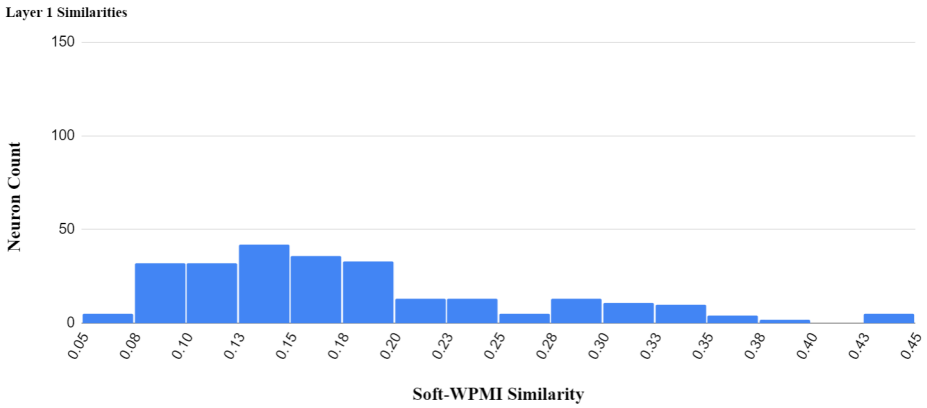

Figure 8: Comparison of Soft-WPMI Similarity and Neuron Counts from Layer 1.

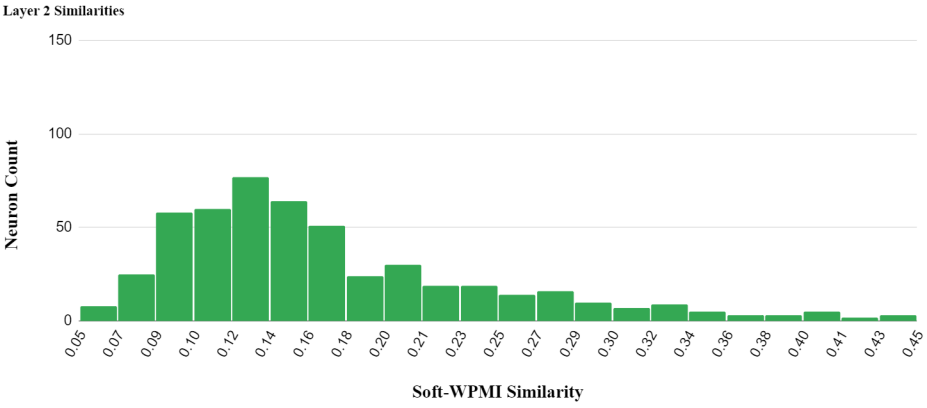

Figure 9: Comparison of Soft-WPMI Similarity and Neuron Counts from Layer 2.

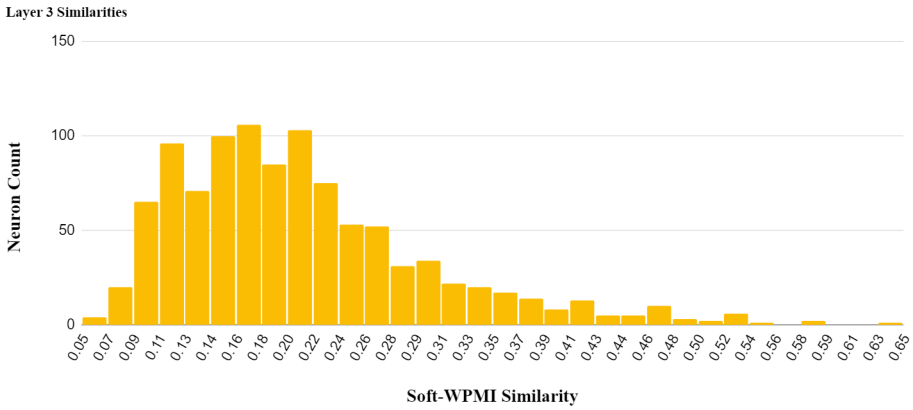

Figure 10: Comparison of Soft-WPMI Similarity and Neuron Counts from Layer 3.

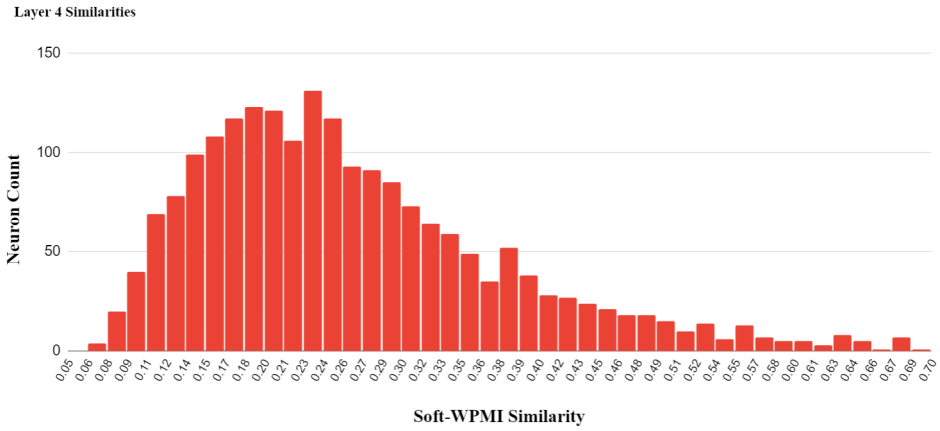

Figure 11: Comparison of Soft-WPMI Similarity and Neuron Counts from Layer 4.

6 Visualizations

6.1 Overall Concepts per Layer

Based on our quantitative analysis, we determined that CLIP-dissect activates most highly on neurons associated with abstract concepts. By comparing the top 26 neuron concepts and reference frequency, as shown in Figures 12, 13, 14, and 15 for ResNet-50's intermediate layers, CLIP-dissect activated most highly on "stripes", "dotted", "spiral", "lattice" and "grid".

We note through this analysis that while layers 1 through 3 as defined by *torchvision* are stacked heavily toward certain concepts, layer 4 is more spread out between concepts. Again, this can be attributed to the progressive increase in concept complexity as images pass through the different layers. For example, layer 4 neurons activate on animals (i.e. dogs) and more complex forms such as certain types of rooms and objects, while the other three layers have most of their activations on more abstract concepts such as stripes and checkers.

This therefore provides us with a broad overview of how these models work; earlier neurons are able to identify basic concepts (i.e. stripes, colors, certain patterns), while later layers combine this information to identify more complex concepts.

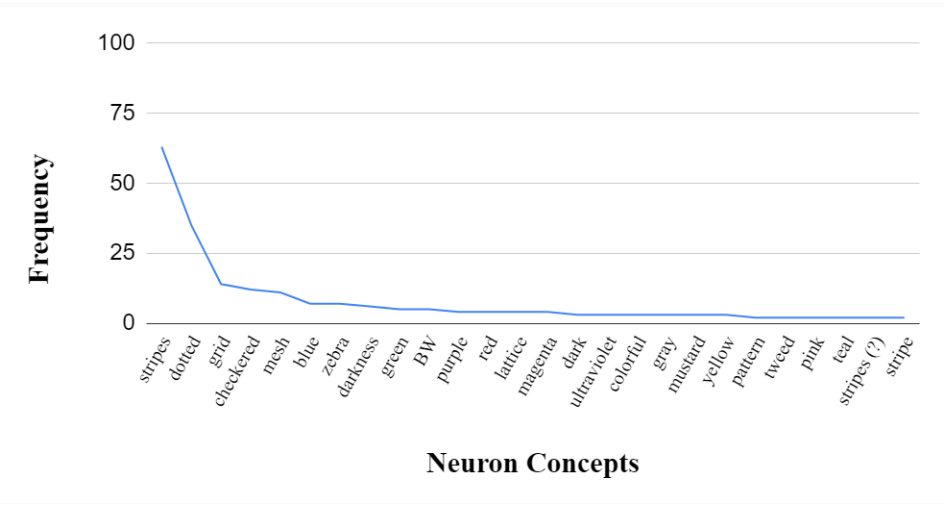

Figure 12: Displays a specific neuron concept and its class frequency within layer 1.

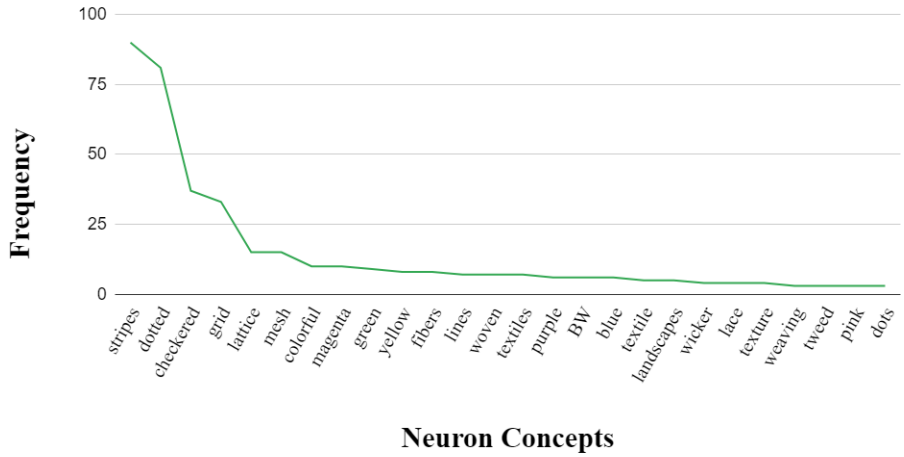

Figure 13: Displays a specific neuron concept and its class frequency within layer 2.

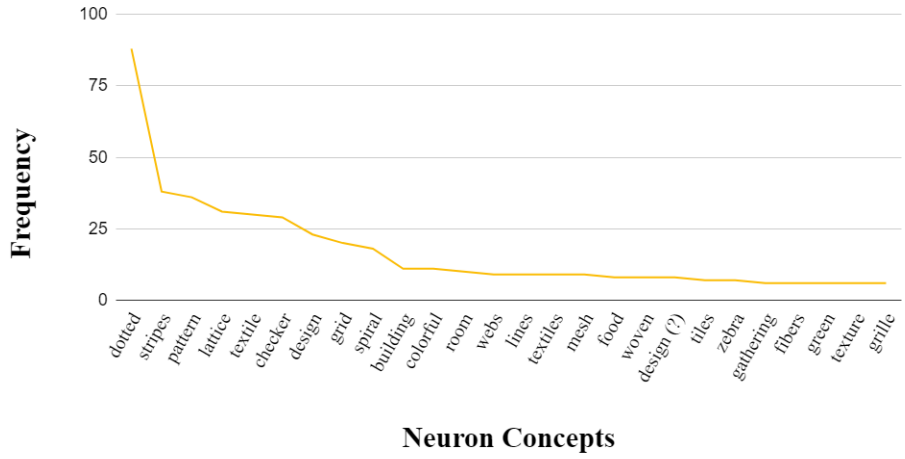

Figure 14: Displays a specific neuron concept and its class frequency within layer 3.

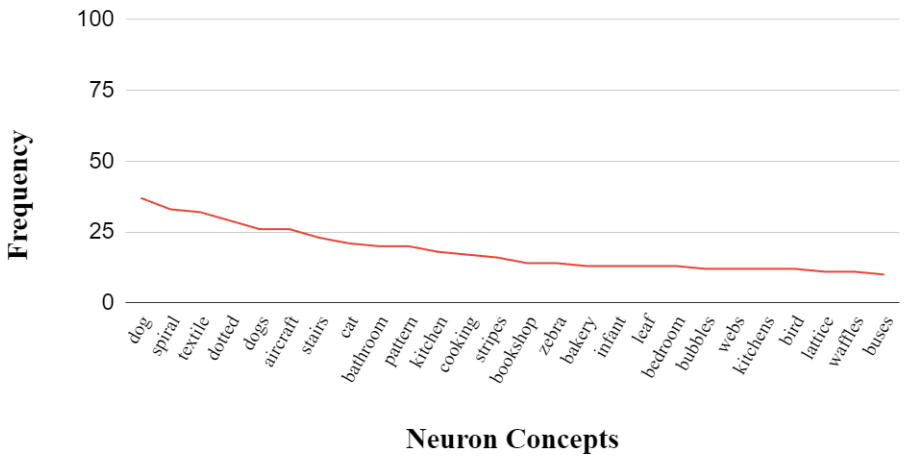

Figure 15: Displays a specific neuron concept and its class frequency within layer 4.

```
Current Layer Number: layer4
Concept Searching For: stripe

Found concept 'stripes' in Neuron 1330 and Index 2 with Similarity Value: 0.432708740234375
Found concept 'stripes' in Neuron 761 and Index 2 with Similarity Value: 0.403717041015625
Found concept 'stripes' in Neuron 769 and Index 2 with Similarity Value: 0.396942138671875
Found concept 'stripes' in Neuron 610 and Index 2 with Similarity Value: 0.391510009765625
Found concept 'stripes' in Neuron 1069 and Index 2 with Similarity Value: 0.3839111328125
Found concept 'stripes' in Neuron 779 and Index 1 with Similarity Value: 0.383758544921875
Found concept 'stripes' in Neuron 1173 and Index 1 with Similarity Value: 0.371429443359375
Found concept 'striped' in Neuron 853 and Index 1 with Similarity Value: 0.371368408203125
Found concept 'stripes' in Neuron 1989 and Index 1 with Similarity Value: 0.332183837890625
Found concept 'stripes' in Neuron 1872 and Index 2 with Similarity Value: 0.331085205078125
Found concept 'striped' in Neuron 582 and Index 3 with Similarity Value: 0.328216552734375
```

Figure 16: Example, searching for concept "stripes" in layer 4 of ResNet-50

6.2 Searching for Specific Concepts

Utilizing CLIP-dissect's ground-truth neuron analysis, we developed a function able to effectively search for specific concepts within the top 10 most highly activating labels and top 20 most highly activated images obtained from the ResNet-50 intermediate layer neurons. An example of this is shown in Figure 16. Similarly, we developed a layer and index search function for the CLIP-dissect analysis. This function allows us to determine the concept assigned by CLIP-dissect to each neuron and measure its similarity with other neurons that may relate to a similar concept. Through these search functions, we are able to measure the similarity between different neurons based on the model's interpretation of specific labels, allowing us to gain a deeper understanding of how the model perceives and represents different visual concepts.

Specifically, we are able to visualize both which layers and in which neurons concepts may appear, and how often concepts appear. This index allows users to more greatly appreciate the complexity of convolutional networks such as ResNet-50, and provides a foundation for which more complex visualizations can be built.

One important idea is that this visualization can easily be created for other convolutional neural networks as well. While in this work we only focus on ResNet-50, this analysis can also be conducted on other ResNet models, as well as visual transformers.

One such visualization that can be presented is determining how neurons activate each other. This is one limit to CLIP-dissect; while we can see the function of each individual neuron, we are unable to see the connections between the different neurons and layers. However,

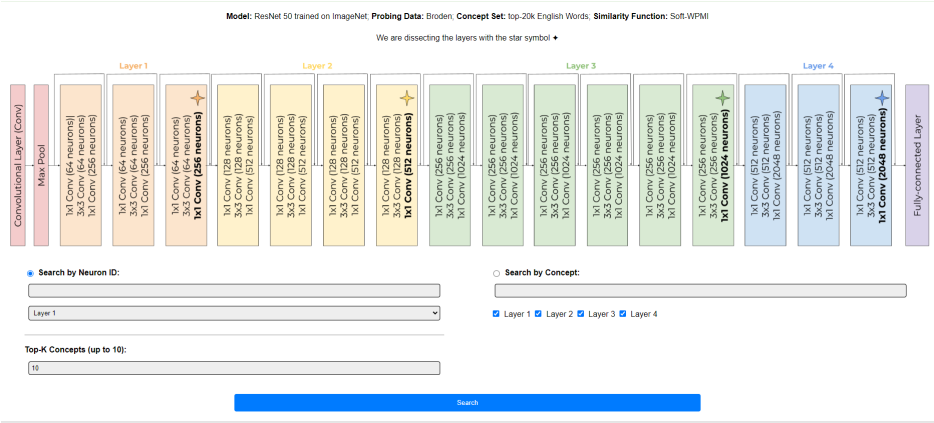

Figure 17: Homepage of the Interactive Site

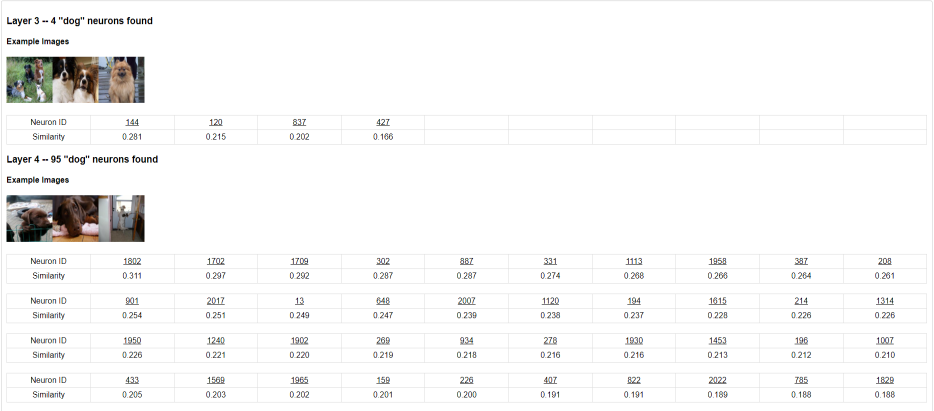

Figure 18: Searching for the concept "dog"

since an activation map is provided by CLIP-dissect, it may be possible to integrate this to identify more complex relationships between neurons.

6.3 Interactive Website

To allow anyone to visualize the selected neurons of the ResNet model with CLIP-dissect labels, we have created an interactive website which you can find at https://dr4nx.github.io/clip-search/index.html.

Through the site, users can interact with select layers of the ResNet-50 network. They can both search for specific neurons as shown in 19 or search the network for a specific topic within the network as shown in 18. This allows users to seamlessly navigate through the network and quickly flip through neurons as desired.

For neuron-specific searches, users can also see the top activating images CLIP-dissect output is based on, as well as the top 10 labels by similarity that are outputted by CLIP-dissect. This is useful as sometimes the label with the highest similarity is not necessarily the most accurate to the images. Users can also see the actual similarity value; this is helpful as we have shown in 5.3 that the similarity function provides a reasonable indication of whether a neuron is human interpretable.

Within the website, there are also hyperlinks to help users find neurons that have similar concepts to each other, as well as specific neurons that might match a specific target label.

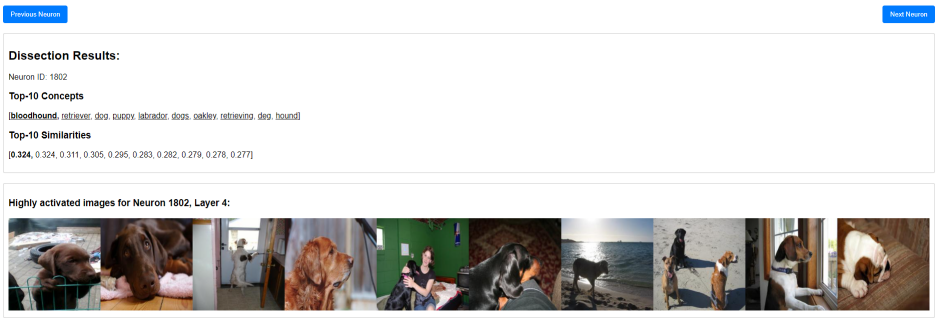

Figure 19: Information for Neuron 1802 of Layer 4

7 Conclusion

In this summer project, we have constructed an analysis of the intermediate neurons of layers of ResNet-50 using CLIP-dissect. We have validated the accuracy and ability of CLIP-dissect for labeling intermediate neurons, and following this validation, we have successfully used the Soft-WPMI similarity constructed by CLIP-dissect to analyze the different concepts and interpretability of neurons in intermediate layers.

Acknowledgements

The authors would like to thank mentor Tuomas Oikarinen and supervisor Professor Lily Weng for mentoring them throughout the project via the outreach program (REHS, Research Experience for High School Student) in UCSD. The authors also thank the UCSD HDSI Department and the UCSD Supercomputer Center for providing the computing resources necessary to run this work.

References

David Bau, Jun-Yan Zhu, Hendrik Strobelt, Agata Lapedriza, Bolei Zhou, and Antonio Torralba. Understanding the role of individual units in a deep neural network. *Proceedings of the National Academy of Sciences*, 2020. ISSN 0027-8424. doi: 10.1073/pnas.1907375117.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, K. Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, 2009.

Kaiming He, X. Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2015.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross B. Girshick. Segment anything. *ArXiv*, abs/2304.02643, 2023.

Siyi Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Chun yue Li, Jianwei Yang, Hang Su, Jun-Juan Zhu, and Lei Zhang. Grounding dino: Marrying dino with grounded pre-training for open-set object detection. *ArXiv*, abs/2303.05499, 2023.

Tuomas P. Oikarinen and Tsui-Wei Weng. Clip-dissect: Automatic description of neuron representations in deep vision networks. *ArXiv*, abs/2204.10965, 2022.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch. 2017.

Zeyu Wang, Berthy Feng, Karthik Narasimhan, and Olga Russakovsky. Towards unique and informative captioning of images. *ArXiv*, abs/2009.03949, 2020.