

Multimodal Neurons in CLIP

Pantelis Emmanouil

03119018

*School of Electrical and Computer Engineering
National Technical University of Athens
el19018@mail.ntua.gr*

Achilleas Theocharopoulos

03119602

*School of Electrical and Computer Engineering
National Technical University of Athens
el19602@mail.ntua.gr*

Chrysoula Panigyraiki

03119010

*School of Electrical and Computer Engineering
National Technical University of Athens
el19010@mail.ntua.gr*

Aliki Zachou

03119215

*School of Electrical and Computer Engineering
National Technical University of Athens
el19215@mail.ntua.gr*

Athanasios Boufeas

03119175

*School of Electrical and Computer Engineering
National Technical University of Athens
el19175@mail.ntua.gr*

Abstract—Our objective is to investigate the presence of multimodal neurons in CLIP through a series of computational experiments. Our inspiration stems from the established correlation between pictures and text in certain neurons of the human brain, which tend to respond in an invariant manner to an entity and its textual description, written or spoken. Similarly, the neurons of an artificial neural network named CLIP convey the same functionality, recognising higher level concepts in various fields, including different people, regions, religions, brands, colours etc. In this paper we examine the multimodal nature of CLIP through feature visualizations, text generation and dataset examples that activate a targeted neuron in a remarkable way. Finally, we explore potential vulnerabilities of CLIP as well as the modality gap phenomenon.

Index Terms—CLIP, Multimodal, Feature visualization

I. INTRODUCTION

The motivation for analyzing and categorizing the neurons of an artificial neural network was the findings from publications that point to the existence of specialized multimodal neurons in parts of the human brain [12]. Several brain studies using depth electrodes have shown that single neurons in the medial temporal lobe (MTL) fire in an invariant manner to different pictures of a person and its spoken and written name. Apart from that, a single neuron can respond to presentations of different people with common attributes (a group of researchers), even if they are not previously known to the subject. The associations between different individuals and their placement into categories by the MTL neuron can occur in a short time span. In a similar manner, it has been shown that neurons on multimodal models such as CLIP operate in the same way. The neurons it comprises are specialized in varied topics and categories, like certain people, emotions, countries etc. Each neuron yields higher activations when given images that correspond to the topic they are specialized

in, much like the neurons in the human brain. In this study, we experiment with individual neurons, aiming to ascertain and visualize their area of interest. Furthermore, we explore the ways CLIP composes text from images and the effect of typographic attacks. We run our experiments on the model RN50-x4, in which the image encoder consists of a CNN network with a feed-forward layer at the end, but in general a vision transformer can also be used. In this study, the neurons we study and evaluate are the filters in the last layer of the convolutional network, which are expected to be specialized in specific subsets of the training dataset in multiple modalities. All of our code is available at <https://github.com/ntua-el19010/clip-multimodal/tree/main>.

II. THEORETICAL BACKGROUND

CLIP (Contrastive Language-Image Pre-training), is a multimodal neural network developed by OpenAI [13], designed to understand, represent and relate images and text in a unified manner. CLIP’s versatility is derived from its ability to perform a wide range of tasks without task-specific training. By encoding both images and text into a shared embedding space, CLIP can be applied to tasks such as image classification, object detection, image captioning, and zero-shot learning. In contrast to other natural language supervision models, the model understands text as a whole instead of individual words. CLIP is pre-trained on a dataset consisting of 400 million images and their natural language captions or descriptions, which are publicly available on the web. During the pretraining, the model learns to encode both images and texts into a shared embedding space, where similar images and texts are mapped closer together, while dissimilar ones are farther apart.

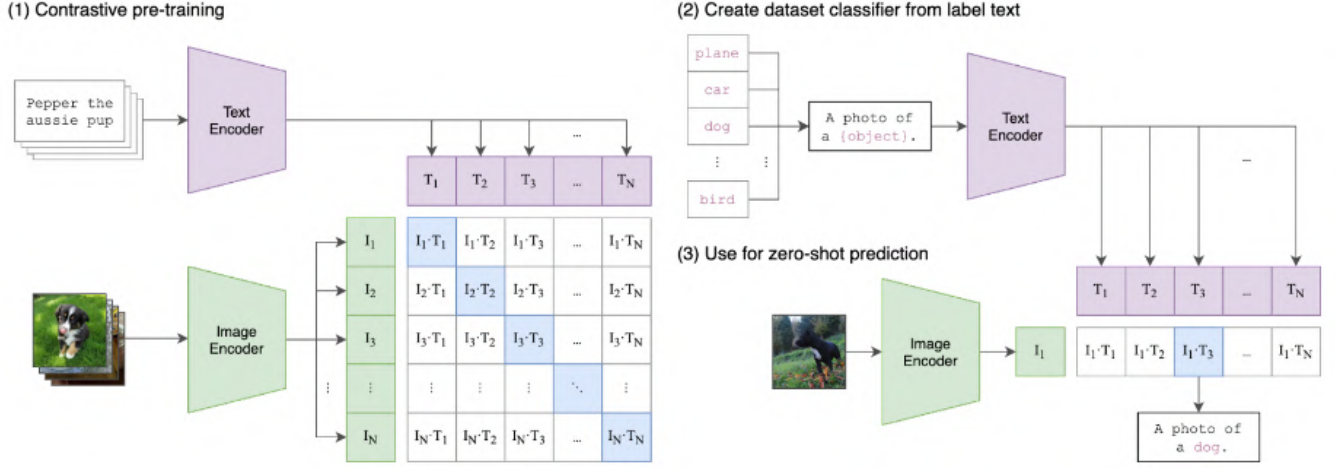


Figure 1. CLIP architecture

CLIP is trained to detect the actual pairs of images and text among the $N \times N$ potential (image, text) combinations in a batch of N image and text samples. Both texts and images are mapped through corresponding encoders to an embedding space, and then compared. The goal of the training is to get the image vectors and the corresponding text vectors as close as possible, with cosine similarity being used as the similarity metric. It calculates the cosine similarity scores between all possible image and text combinations and as such, it distinguishes between positive pairs (images and texts that belong together) and negative pairs (images and texts that do not belong together) [2]. One way to utilize CLIP for classification inference is to tokenize/encode a large set of potential text captions resulting in a corresponding vector. All captions should have the same format, for example “A photo of ...” followed by a potential class label. Subsequently, this vector is compared to the encoding of a single image in order to pick the caption that best describes it, based on the maximum cosine similarity.

III. METHODS

A. Neuron Visualization

1) *Maximal activation while capturing a natural pre-image:* Our target here is to derive a realistic image I that maximizes the activation of a selected neuron, which, in that case, is a convolutional filter at the final layer of the visual encoder. The process of synthesizing I from random noise is formulated as an optimization algorithm and is decomposed in the following steps: First, the weights of the given model (ResNet50-4x) are being fixed and the pixels of the input image are randomly initialized and being set as trainable parameters. At each optimization step, the pixels are being updated through adaptive-momentum gradient descent, which is applied on the following objective function: $J(x) = R_a(x) + R_{TV^\beta}(x) + CE_\tau[l(\Phi(jitter(x; \tau)), \Phi_0)]$

$[jitter(x; \tau)](v, u) = x(v + \tau_2, u + \tau_1)$: The jitter operator is applied on an image tensor x , resulting in a shift across

both height and width dimensions. $\tau = (\tau_1, \tau_2)$ is a random vector and τ_1, τ_2 are independent discrete variables, uniformly distributed in the $0, \dots, T-1$, where T (maximum shift) is a parameter to be set. According to the paper originally proposing the jitter technique [9], random jittering counterbalances the downsampling effects in deep CNNs, interpolating between pixels in the backpropagation and thus producing “crisper” pre-images.

$l(\Phi(x), \Phi_0)$: Loss function between a reference tensor $\Phi(\cdot)$ (here is the weight tensor of a selected convolutional filter) and a transformed image tensor $\Phi(x)$ (here $\Phi(\cdot)$ is the function performed by the intermediate layers of the visual encoder).

$CE_\tau[l(\Phi(jitter(x; \tau)), \Phi_0)]$: Expected value of the loss function $l(\cdot)$, applied on the random tensor $\Phi(jitter(x; \tau))$ and Φ_0 . C is a hyperparameter to be set.

$R_a(x)$: “Realistic image regularizer” intending to limit pixel intensity to a bounded range.

$$R_a(x) = \begin{cases} N_a(x) & \forall u, v : \sqrt{\sum_k x(u, v, k)^2} \leq B_+ \\ +\infty & \text{otherwise} \end{cases}$$

where B_+ is the upper bound on intensity and

$$N_a(x) = \frac{1}{HWB^\alpha} \sum_{v=1}^H \sum_{u=1}^W \left(\sum_{k=1}^D x(v, u, k)^2 \right)^{\frac{\alpha}{2}}$$

$N_a(x)$ can be interpreted as a modified L_2 regularizer.

$R_{TV^\beta(x)}$: Additional realistic image regularizer, intending to encourage smooth transition in the intensity of neighboring pixel patches. Its expression is:

$$R_{TV^\beta(x)} = \frac{1}{HWB^\beta} \sum_{vuk} ((x(v, u+1, k) - x(v, u, k))^2 + (x(v+1, u, k) - x(v, u, k))^2)^{\frac{\beta}{2}}$$

The optimization process is executed until convergence and the output image $x^* \in \mathbb{R}^{H \times W \times C}$ is expected to portray certain

concepts to which the targeted neuron is specialized, in a human - understandable form.

In our work, a variation of that method has also been employed : There, the $R_{TV\beta(x)}$ regularizer is modified to enforce similar R,G and B values in every pixel. Practical results conveyed that this regularization technique, although it leads to gray scale images, tends to allow the formation of sharper and more distinct edges and shapes. Moreover, it is more computationally efficient, since the learning process tends to converge in approximately one fifth of the steps required for the initial version of the method. $R_{TV\beta(x)}$ has now the following expression :

$$R_{TV\beta}(x) = \frac{1}{HWB^\alpha} \sum_{vuk} ((x(v, u, k + 1) - x(v, u, k))^\beta$$

Hyperparameter tuning for both the original method and its variation has been based on the general directions given at [9] . However, it has to be pointed out that different neurons may call for slight modifications.

2) *Deep Dream Visualization*: The "Deep Dream" ([10]) is an alternative optimization-based algorithm intending to visualize the features captured by a neuron. Similarly to the previous method, our purpose is to design a sequential learning process such that, starting from a random noise image, to converge to an image that depicts human - interpretable patterns that maximally activate a neuron of interest. The distinguishing feature of Deep Dream is rooted in traditional computer vision and involves the processing of the input image in multiple resolutions : At each of these resolutions, a sequence of Gaussian smoothing gradient ascent steps is executed, aiming to increase the mean square error between the activation map of the neuron and the zero tensor. Mathematically, that process can be described as following:

H : Height of optimal image I^* . Here, $H = 288$

W : Width of optimal image I^* . Here, $W = 288$

$L(I)$: Loss function with respect to a given input image

I_0 : Input image of resolution (H_0, W_0) , randomly initialized

$\phi(I)$: Neuron activation map w.r.t. a given input image

pyramid size: Number of different resolutions

pyramid scale: Scaling factor between different resolutions

shift: Random variable, uniformly distributed in $[-(shift\ bound - 1), shift\ bound]$

$'*$: Symbol for depthwise convolution

η : Learning rate

$G(\sigma_i, \mu)$: Gaussian kernel of standard deviation σ_i and mean μ

σ_i : Standard deviation of the i_{th} Gaussian kernel.

Here, optimizing the input image in multiple resolutions is expected to lead to a more comprehensive visualization, as it can possibly enable the capture of neuron's features in different layers of abstraction. Moreover, the Gaussian smoothing and normalization of the gradient tensor is expected to increase the correlation between the gradients of neighboring pixels, thus leading to a more stable and coordinate trajectory in the pixel space. As in the previous case, our goal is synthesize an

Algorithm 1 Deep Dream Visualization

```

1:  $H_0 = H / (pyramid\ size)$ 
2:  $W_0 = W / (pyramid\ size)$ 
3:  $I_0 \in \mathbb{R}^{3 \times H_0 \times W_0}$ ,  $I_0(x, y, z) \sim \text{Uniform}(0, 1)$ 
4:  $L(I) = \text{MSE}(\phi(I), 0)$ 
5: for  $k = 1$  to pyramid size do
6:    $H_k = H_0 \cdot (pyramid\ scale)^{k-1}$ 
7:    $W_k = W_0 \cdot (pyramid\ scale)^{k-1}$ 
8:    $I_k = \text{bilinear interpolation}(I_{k-1}, H_k, W_k)$ 
9:   for  $j = 1$  to total iterations do
10:     $I_k(x, y, z)^{(j)} = I_k(x, y - shift, z - shift)^{(j-1)}$ 
11:    reflective padding( $\nabla_{I_k^{(j)}} L$ ,  $\frac{1}{2}$  kernel size)
12:    for  $i = 1$  to 3 do
13:       $grad_i^{(j)} = \nabla_{I_k^{(j)}} L * G(\sigma_i, \mu)$ 
14:    end for
15:     $\nabla L_s^{(j)} = \frac{1}{3} \sum_i grad_i^{(j)}$ 
16:     $\nabla L_{s,norm}^{(j)} = (\nabla L_s^{(j)} - \text{mean}(\nabla L_s^{(j)})) / \text{std}(\nabla L_s^{(j)})$ 
17:     $I_k^{(j)} = I_k^{(j)} + \eta \nabla L_{s,norm}^{(j)}$ 
18:     $I_k(x, y, z)^{(j)} = I_k(x, y + shift, z + shift)^{(j)}$ 
19:  end for
20:   $I_k = I_k^{(total\ iterations)}$ 
21: end for
```

image that will depict certain interpretable features to which the neuron is specialized.

Most of the parameters above (except for H and W which are fixed) are hyperparameters to be set. Their tuning is, again, based on the Colab Notebook that is originally implementing the Deep Dream method.

B. Neuron Activations

From the neuron visualizations we can infer that specific neurons respond to particular patterns, concepts and images. In order to label these neurons more confidently, we examine and compare their responses to different stimuli.

First, we need to use a sufficiently large and representative dataset, for this reason we chose the Tiny ImageNet Dataset [6], which includes 100,000 training images divided up into 200 classes. We isolate the final convolutional layer of the model and feed images to the model in batches of 50. We rank the samples based on the activation of the neuron of interest and consequently limit the initial dataset to a set of images that cause it to fire remarkably. Using this technique and by choosing an adequately representative dataset containing samples from all classes of interest, we can assign neurons to object classes. In our research we experimented with two distinct methods to gather the activations, which are described in more detail below:

- **Modifying CLIP by removing the outer layers**: We create a new version of CLIP, based on a Modified Residual Network architecture, which is similar to the version of CLIP used for our other experiments, although it is missing the final layers after the last convolutional layer. Therefore the activations of this last layer coincide with the model's output.

- Utilizing hooks to collect neuron’s activation [1]: We load CLIP RN50-x4 and attach a forward hook to the last convolutional layer of CLIP. Hooks are functions called every time a single layer is used, that allow us to inspect the output of the selected layer.

Furthermore, to confirm and expand on these results, we propose an additional experiment, in which we construct a dataset using images from Flickr that we label ourselves. We then arrange the images into categories, i.e. food, art, music. As such, we can visualize the different levels of activation for individual neurons by category instead of examining single images, thus being able to generalize instead of only providing the images generating the top activations. The activations are grouped by the category in which the corresponding image belongs, and thus we get a better idea about how each topic activates and represents the neuron, verifying the result of the previous experiment.

C. Composition of Maximal Text

Our objective is to create a text that exhibits maximum correlation with a given input image. The CLIP model is illustrated in the photo below. The aim is to minimize the distance between the embedded vectors of the image and text. During training, vectors are normalized before comparison using their inner product, and we choose their cosine distance as our optimization loss function.

Initially, we calculate the embedded image vector. Optimizing with respect to the input text proves challenging, as the gradient descent algorithm necessitates real-valued parameters to compute gradients. Consequently, we cannot optimize either the text input or the integer tokens directly; instead, we optimize the token embeddings. This allows us to achieve a text vector close to the image goal vector (cosine similarity near 1). However, the obtained token embeddings lack practical value as they do not correspond to any actual tokens. Even if we identify the closest tokens for our embeddings, their distance is often substantial, and the resulting decoded text no longer aligns with an encoded vector close to our goal.

To address this issue, we choose to optimize the embeddings over a small number of steps (referred to as an epoch), identify the nearest tokens and corresponding text, and repeat this process multiple times. The idea is to ensure that the embedding vector never strays too far from the feasible embedding space, gradually improving results.

For training, we employ the SGD optimizer with a high learning rate of 2 and 50-100 iterations per epoch. The elevated learning rate is intended to cause the optimization vector to deviate significantly from its initial value, allowing it to match a different token vector (i.e., a different text). The number of epochs is arbitrary, and generally, a higher number leads to improved results. Since this optimization process is not strictly “increasing,” we store all intermediary texts generated, rank them based on their cosine similarity, and retain the best ones.

Additionally, we limit the text size to 5 tokens (approximately 5 words) to prevent the generation of large texts during later epochs (up to more than 100 tokens). To achieve this, we

explicitly set the vectors corresponding to all tokens with an index greater than 6 to their initial value (“<end>”).

While this method may not generate syntactically correct text, we anticipate it to output words related to the input image. Our goal is not to generate coherent sentences but to produce text sequences that maximally match the image encoding in the model.

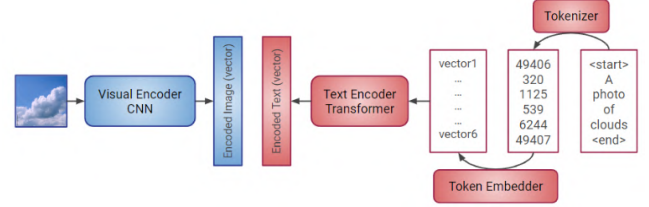


Figure 2. Encoders for the composition of maximal text

D. Exploring Vulnerabilities in CLIP’s Multimodal Neurons

1) *Executing Typographic Attacks:* Given that CLIP contains multimodal neurons, which respond to images of a specific class as well as a set of words that often relate to the class, one may exploit this feature to guide the model into making inaccurate classifications or predictions. Generally, attacks of this nature are described as “adversarial attacks” and are based on minor perturbations of input samples that are initially categorized correctly, aiming to alter the predicted label given by the model and misclassify those samples [5]. In this case, instead of making imperceptible changes to the images we opt to add an “adversarial patch” to them, an image-independent element that is noteworthy for the neural network [3]. Since the patch is a word or short text, the attacks may also be referred to as “typographic attacks”. In order to execute these attacks we place the attack text on the same eight random coordinates of each input image, as shown in the before and after example below.



Figure 3. An example of typographic attacks

The texts we used were mostly selected from the ImageNet class names, aiming for short words that could potentially result in effective attacks, although a few other words and symbols that relate to specific neurons were also tested. We inserted the different texts on 1000 images from the ImageNet validation set and performed classification using the zero-shot method supported by CLIP. The prompt used to format the input for CLIP’s text encoder is “This is a photo of a

{label}}". As far as performance is concerned, we measured and considered the following parameters to assess each attack:

- Pixel coverage: The percentage of the initial image's pixels that were altered by adding the attack text, averaged over the 1000 samples.
- Success rate: The percentage of images for which the attack class is found to be the most likely at the end, but is not the same as the initial classification of the image.

2) *Analyzing response to Stroop Effect*: Another experiment we can implement to assess CLIP's multimodal neurons is imitating the Stroop Effect [8], an established phenomenon in cognitive science. It is based on a task of determining the color in which a word is written, while ignoring the actual word. For humans, this task is easy if the word has no correlation to the color, but gets increasingly difficult when the word is also another color, for example the word "blue" printed in red ink. In that case our brains require more time to process the contradictory stimuli in order to respond correctly. However, CLIP does not have the ability to delay inference when faced with more challenging tasks, therefore we are interested in measuring its performance on the classification of mislabeled colors. We produced images with a white background and the name of a color written in various other shades and performed classification using the zero-shot method supported by CLIP. The prompt used to format the input for CLIP's text encoder is "My favorite word, written in the color {color}".

E. Modality Gap

In this section, we explore the positional relation between embeddings of different modalities, observed in [7]. Multimodal models often project data from different modalities in a common latent space. In this context, it is observed that the encoded vectors of different modalities occupy different cone regions of the high dimension latent space. This phenomenon is not caused by the distinct modalities, but is attributed to the different neural networks for each modality themselves. Due to the ReLU and other non-linearities present in a network, its output is limited in a small sub-cone of the euclidean space. Different networks lead to distinct sub-cones, resulting in a gap between their outputs, which in this context we call modality gap. Thus, this gap is generated first and foremost by the random initialization of the networks.

The training objective used in CLIP models is the contrastive loss: $L = -\frac{1}{N} \sum \log \frac{e^{x_i y_i / \tau}}{\sum e^{x_j y_j / \tau}}$. This objective is shown to increase the modality gap for small temperature τ values, like the $\tau = \frac{1}{100}$ used in CLIP. It is also observed that closing the modality gap does not necessarily lead to better performance in downstream tasks.

IV. RESULTS

A. Neuron Visualization

We provide some visualization examples using both colored/gray scale maximal activation and Deep Dream techniques. The selected examples are considered illustrative of

our methods' visualization capabilities and capture a particular type of behavior that manifested in a wide range of neurons. Based on the results presented in [4] and its supporting website [11], neuron 129 is activated more by images consisting of spirals and circular objects (such as doll faces and eyes) while 275 is activated more by images portraying dog faces. In Figure 4, we examine neuron 129, and the effect of longer training on the visualization quality. We notice that in each case, some head-like shapes such as large mouths and smaller eyes can be observed. Thus, in a wider sense, we may claim an interpretable relationship between the portrayed artificial patterns and the neuron's theme. The convergence of the training process to increasingly better results is definitely a desired property and confirms the stability of the method. In Figure 5, we examine neurons 129 and 275 using our gray scale visualization technique. While a dog figure can be clearly observed in the neuron 275 visualization, the image generated for neuron 129 contains mostly cave painting-like figures and faces, therefore demonstrating just a slight accordance to the neuron's dominant features.

In Figure 6, we examine three different neurons (31, 32 and 39) and their visualizations using the Deep Dream techniques. The conclusion derived is that, in all three cases, the algorithm fails, to an extent or completely, to capture the essential features of the neurons (baseball caps and scripted time signs, Israel/Gaza and political symbols respectively). Neuron 31 may be considered an exception, as a bowl-shaped object resembling to a cap can be distinguished. By testing that method with numerous neurons, we have noticed that, on one hand, it generally tends to portray a variety of patterns and especially words, which were scarce in the visualizations produced by our first method. In the case of neurons 32 and 39 for example, we notice the appearance of word-like patterns like "KAKO", "RAK", "ΠΑΟΚ", "ΠΟΠΟ". On the other hand, Deep Dream presents a significant drawback and that is the lack of interpretability: Although it produces interesting patterns, they generally cannot be interpreted in a meaningful sense and have little or almost no relevance to the concept of the corresponding neuron.

B. Neuron Activations

We select a few indicative neurons to explore the activations. Majority of the results are shown in the Appendix along with representative labels chosen by us. It is evident that certain neurons respond to images of very specific categories, such as Superman, Jesus, Catholicism etc. Some neurons display counting capabilities and can distinguish objects or people by count, while others detect the presence of objects in a specific color. Furthermore, we highlight the fact that the birthday neuron responds to the word "birthday" as well as birthday cakes and people blowing candles. We also discovered two neurons that respond to the Apple brand and products, however one seems more targeted towards the iPod, while the other fires for the Apple logo and pictures of apples.

There are no relevant images in Tiny ImageNet, therefore we decided to test this method on some smaller datasets we

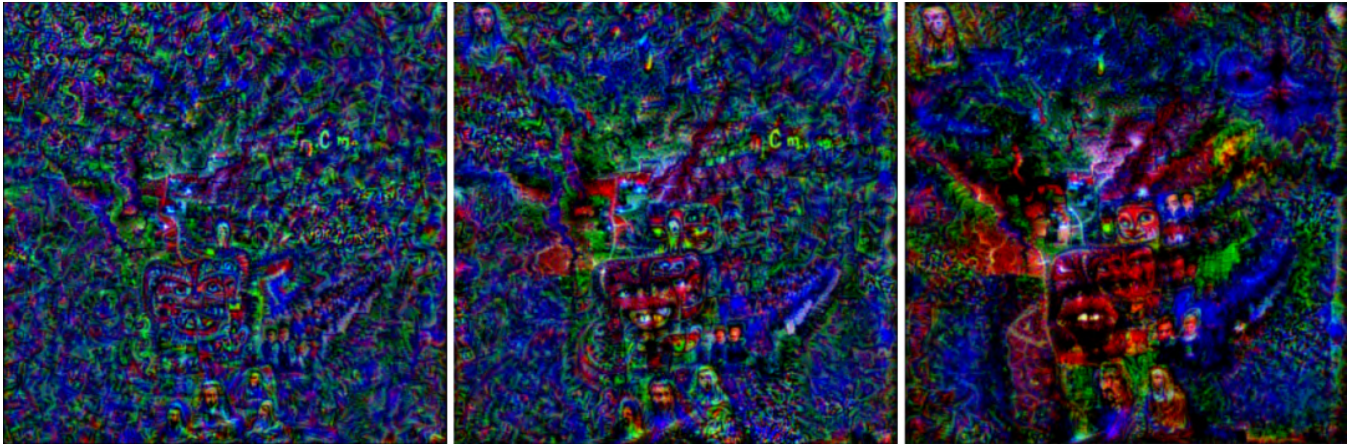


Figure 4. Neuron 129 visualized using our proposed maximal activation technique, running for 5.000, 20.000 and 100.000 iterations respectively

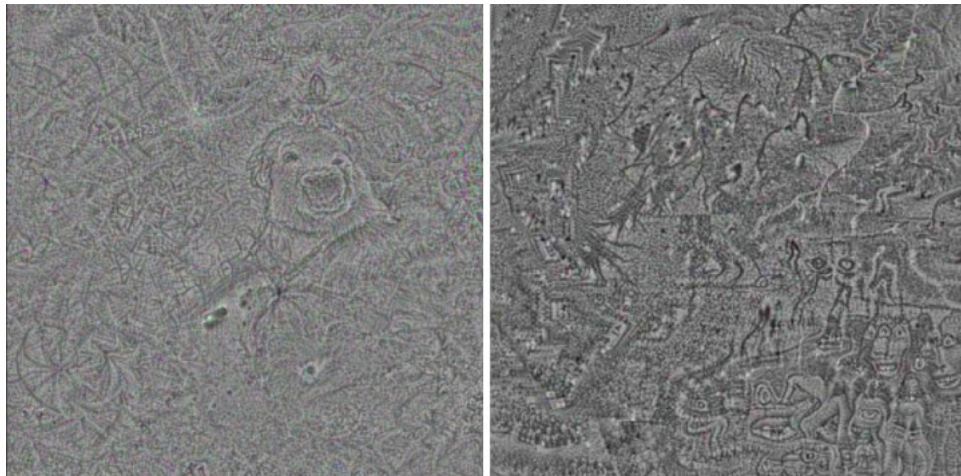


Figure 5. Neurons 275 and 129 visualized using our black and white maximal activation technique

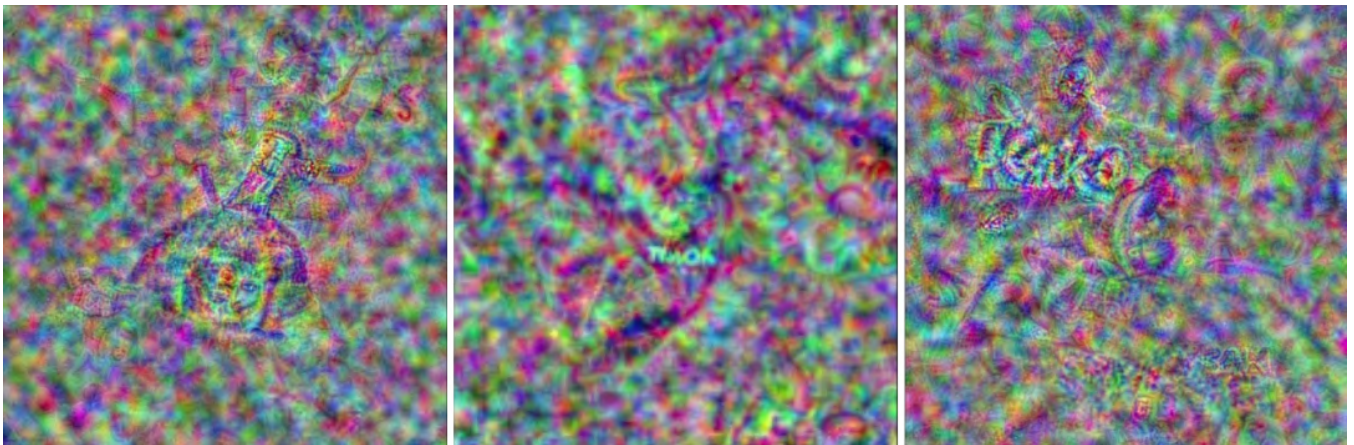


Figure 6. Neurons 31, 32, 39 visualized using the Deep Dream technique

created that are described in more detail below.

1) *Neuron 89*: With our proposed methods, we evaluate neuron 89, which turns out to be most stimulated by the images shown in figure 8, which return the highest activation scores.

As such, we can infer that neuron 89 is strongly activated by Donald Trump. It becomes apparent that this neuron is not only stimulated by photographs of this person, additionally it responds to drawings and photographs that contain his name.

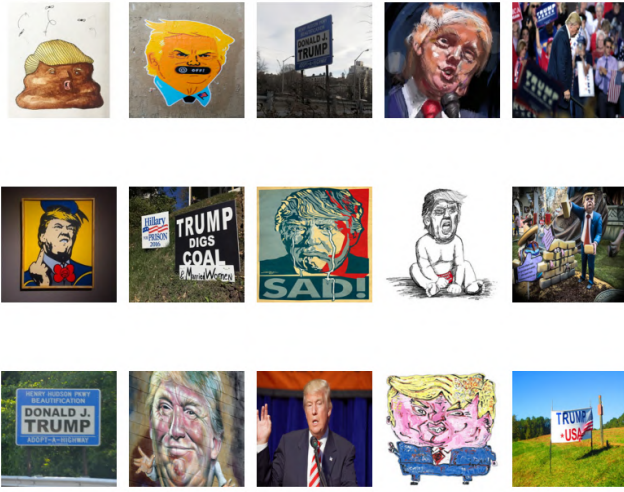


Figure 8. Top 15 images activating neuron 89

To visualize the different levels of activation of this specific neuron we created a dataset with images representing the following categories:

- Photographs of Donald Trump's face up close, which resemble a profile picture
- Paintings, cartoons and street art depicting him or inspired by him
- Text containing his full name or signature
- Partial photographs of him, taken from further away or at an angle
- Images with political content, including the typical phrase "Make America Great Again" and photographs of other relevant politicians (Mike Pence, Hillary Clinton)
- Non political images, for instance pictures of food, animals and houses
- Symbols of the LGBTQ+ community and photographs of activist Martin Luther King Jr., that represent black and gay rights
- Images related to musicians and video games like Fort-night

We then calculate the activation of this neuron when presented with images from each category, and the visualized results are shown in Figure 7. As seen on the figure, pictures with his profile, his name and art pieces in varying mediums depicting him cause the neuron to fire most, with the activation being comparably the largest. In contrast, pictures containing non-political, musical and gaming imagery, as well as photos concerning LGBTQ+ rights, yield minimal activation values, as expected. Although, we did not notice negative activations for any image sample, the values for images that may be considered unrelated to Donald Trump are close to zero. The neuron's reactions align to a great degree with the results from [4] and our own perceptions about Donald Trump, his political symbols and associations.

2) *Neuron 2191*: We conduct the same experiment for neuron 2191, which supposedly (according to [4]) fires the most for images related to mental illness.

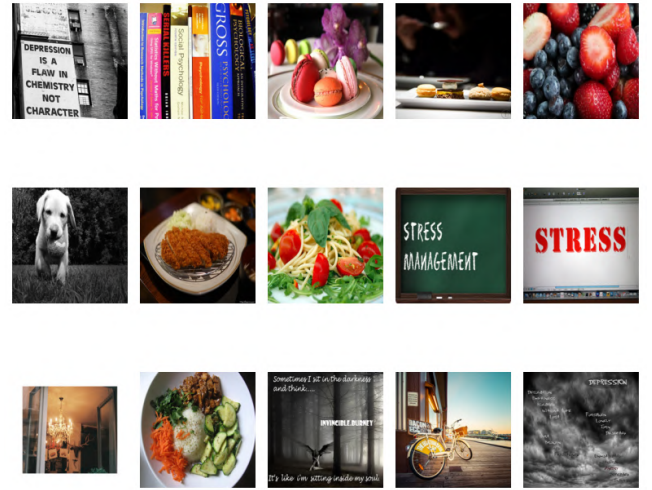


Figure 9. Top 15 images activating neuron 2191

The results in this case, as shown in figure 9 contain a few images associated with depression and anxiety, specifically ones that include the words "depression" and "stress" written

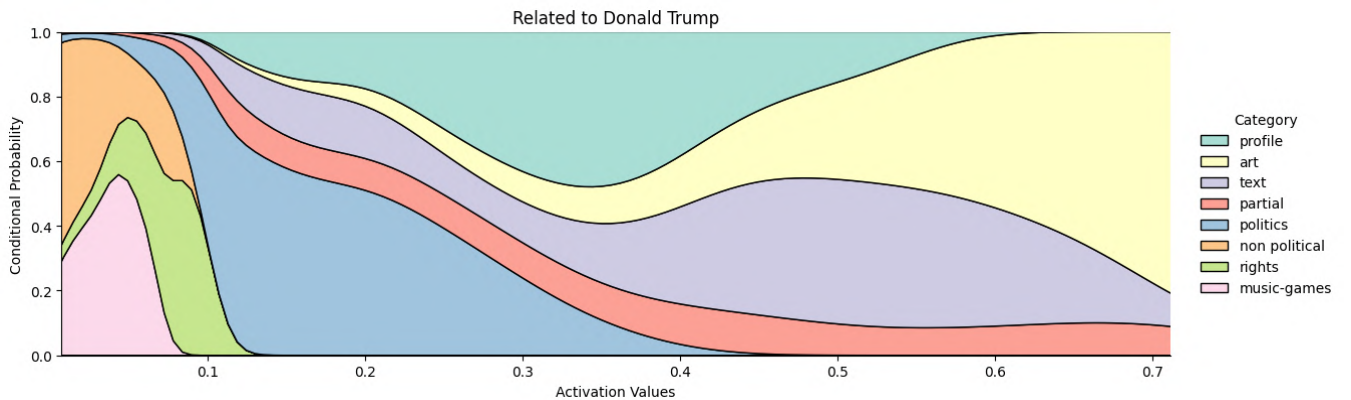


Figure 7. Activations Related to Donald Trump

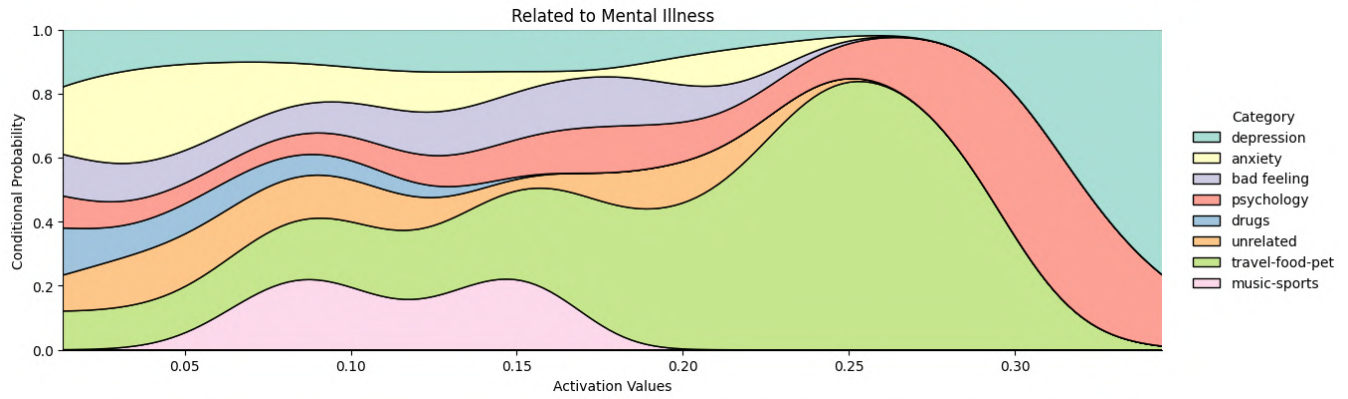


Figure 10. Activations Related to mental illness

on them.

To verify this hypothesis we create a dataset with images representing the following categories:

- Pictures of people in a depressive mood and text containing the word "depression"
- Images of people feeling anxious and text containing the words "anxiety" or "stress"
- Scenes that cause a bad feeling to the viewer and text containing the words "anger" or "loneliness"
- Pictures about psychology and therapy, as well as relevant book covers
- Different kinds of drugs, pills and medicine
- Pictures unrelated to mental illness (art, nature etc)
- Images about travel, food and pets
- Images about music and sports

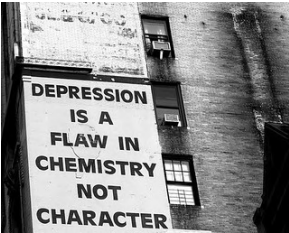
We then calculate the average activation of this neuron, and the visualized results are shown in Figure. 10.

In contrast to the previous figure, these results presented an unexpected pattern. The neuron exhibited heightened activity in response to images depicting various locations and sights, food and pets, categories typically unrelated to mental illness and anticipated to yield negative activation values. Conversely, images depicting psychology and depression elicited the anticipated response. However, pictures related to anxiety and depression evoked low activation values. It's evident that the model struggled to perform adequately on this dataset. This performance could be caused by the complexity of mental states such as depression and anxiety, which are considered to be multifaceted, thus much more difficult for the model to recognise them effectively compared to the case of Donald Trump. Apart from that, the pictures were all chosen by hand, which makes the whole process less accurate. We mention that the results improved as soon as we enriched the dataset with pictures including the text about each category.

C. Composition of Maximal Text

In Table. I, we show two input images based on the categories we studied in the above section (Donald Trump and Mental Illness) and the respective text generated by our method, ranked by cosine similarity.

Table I
MAXIMAL TEXT

Image	Maximal Text Ranked by Cosine Similarity
	<ol style="list-style-type: none"> 1) factfriday depression 2) prohibition psychology depression 3) depression are 4) dayoignorance depression 5) depression gibbons 6) depressing former cpd 7) scott intrin 8) prohibition equation pre-vail 9) advocacy symptoms 10) paralleled symptoms
	<ol style="list-style-type: none"> 1) treats trump accomodnavis 2) candidacy otc potus 3) ics trump avoi 4) trump unsuccessful shee-han 5) realdonaldtrump sabcnews marcorubio 6) realdonaldtrump uk trade-mark 7) caregiver explodes ddled 8) trumpalternatives monochrome 9) trump ancho 10) manafort endorses tableau feathers

The resulting maximal texts match our expectations, since the majority include essential words and phrases that can be used to describe the respective topics. In the case of mental illnesses, the word "depression" appears in almost every case. "Symptoms" also appears twice, which is a word often associated with mental illnesses. In the case of Donald Trump the combination of words "realdonaldtrump" appears multiple times, which is reasonable given that it is his Twitter/X username and was possibly included in image caption that

were used when training CLIP. The results also include the names of other public figures and politicians at the time, that appeared in news headlines alongside Donald Trump, such as Paul Manafort and Marco Rubio.

D. Exploring Vulnerabilities in CLIP’s Multimodal Neurons

1) *Executing Typographic Attacks:* Results of the Typographic Attacks we executed are presented in Table. II. The most successful attacks achieve 88.43% success rate with altering only 6.75% of the original image’s pixels on average. Multiple attack texts lead to highly effective attacks, although none are comparable with the results found in [4]. These discrepancies may be attributed to the utilization of different classification methods, in our research we use zero-shot prediction while the other experiments were conducted using linear probes. Nonetheless, a success rate in the range of 60-80% is considered satisfactory, taking into account the notably small percentage of pixels changed. Typographic Attacks are proven to be effective against CLIP, transforming one of the model’s greatest strengths in recognizing text to a flaw.

2) *Analyzing response to Stroop Effect:* In Figure. 11, we show a sample of CLIP’s attempts at recognizing mislabeled colors.

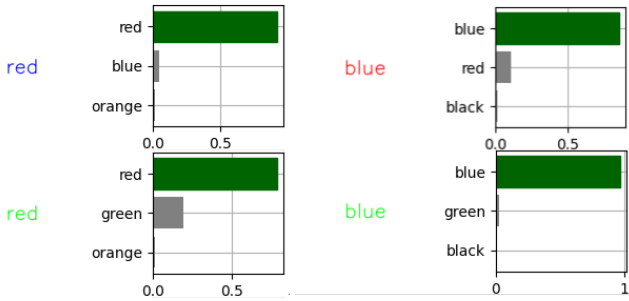


Figure 11. Sample response to Stroop Effect

In this experiment the correct answer is the color in which the word is written, however the model’s ability to recognize the word itself interferes and leads to inaccurate responses for the majority of color-name combinations. A more comprehensive set of predictions is included in the Appendix. The model’s response to a random word printed in multiple different colors is also included for comparison, which shows

that in that case CLIP identifies the color correctly. As was foreseeable the model cannot slow down to manage the harder task, contrary to the way the human brain adapts, resulting in a high error rate.

E. Modality Gap

We observe that the text and image embeddings of a few select image - text pairs occupy different subspaces in the embedding space. This is illustrated in Figure 12, where the 640-dimension embedding vectors have been projected in a 2-dimension space using the Umap dimensionality reduction technique.

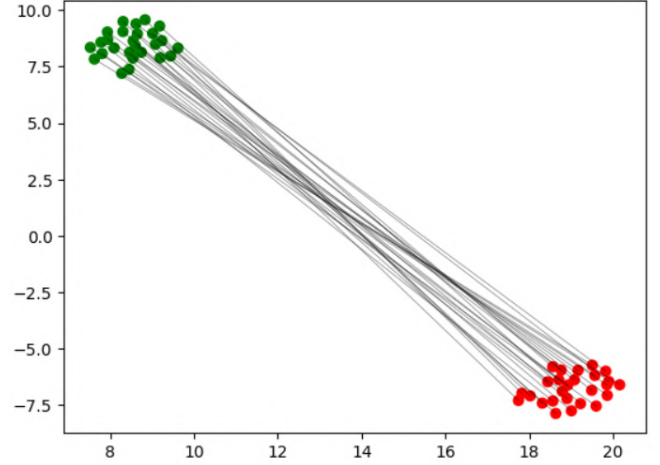


Figure 12. Modality Gap Illustration, image embeddings in red and text embeddings in green

V. DISCUSSION AND CONCLUSION

The primary target of the majority of our methods was to outline the multimodal nature of CLIP’s processing units. Within that framework, neuron visualization techniques could prove very informative, especially if the concepts portrayed in the corresponding visualizations turn out to be human interpretable and of multiple modalities (e.g. text and visual concepts). The algorithms we employed, however, did not lead to particularly satisfying results : Although interesting text and visual patterns did appear, they generally lacked relevance to the neuron’s main concept and they were overshadowed by the high frequency patterns that dominated the visualizations.

Table II
TYPOGRAPHIC ATTACKS

Target Class	Attack Text	Pixel Coverage (%)	Success Rate (%)	Target Class	Attack Text	Pixel Coverage (%)	Success Rate (%)
iPod	iPod	6.75	88.43	library	library	9.55	57.73
jeep	jeep	6.77	85.40	tick	tick	5.85	51.80
radio	radio	7.65	75.04	pizza	pizza	8.13	39.57
rifle	rifle	6.08	71.21	pug	pug	6.09	36.05
ski	ski	4.57	66.77	great white shark	shark	8.13	8.97
tram	tram	6.65	66.70	waste container	trash	7.53	3.04
oboe	oboe	6.79	59.10	Siamese cat	meow	7.63	0.03

The fact that hyperparameter tuning was purely empirical and many of the techniques we adopted (such as jittering, gradient smoothing etc.) were not accompanied by a comprehensive mathematical justification of why they work made it difficult for us to design our visualization techniques in a structured way. Furthermore, both of our visualization methods were originally tested in shallower and purely visual neural networks such as VGG-16. It is possible that these methods cannot match the complexity of CLIP, which roots both in its depth and its multimodal nature. Despite the weaknesses exposed above, we claim that visualization could be used as an assistive tool to a series of techniques with more interpretable results, such as neuron activations.

For our selected neurons in our experiments involving neuron activations, we compare our results with those provided in OpenAI’s Microscope [11], which provides many of the images that activate most each particular neuron for different versions of the CLIP model. In OpenAI’s Microscope the provided image samples have been sourced from ImageNet, which spans 1,000 object classes and contains 1,281,167 training images, as well as Yahoo Flickr Creative Commons Dataset, which includes around 99.2 million photographs. We created and tested with different and much smaller datasets, since we couldn’t use a large one due to very limited Google Colab resources. It is evident that the dataset we were able to process is minuscule in comparison, which may explain the lack of representative images for some neurons. Another notable difference is that we show entire images which give the largest activation values, whereas in the Microscope environment only smaller parts of the pictures are included, meaning that the results are more specific and transparent, allowing for more precise labeling of each neuron.

Concerning the composition of maximal text, as a first experiment we tried this method by providing to the model the visualizations we created, as seen here. Since the images do not provide clear visuals, the resulting text was not very representative of the neuron, based on the fact that the entries had no apparent theme. As such, we opted to experiment with real images, which would give us more conclusive results. Examples of the text composed by our own visuals are provided in the appendix. It is noted that in those examples, in some cases the produced text contained emojis which we opted not to include.

In [4] the images used for the respective experiment were the results of specific neuron’s visualization. Nevertheless, since our results in the visualizations are less than ideal in the sense that there are too little details, we opted for images we found to greatly activate previously studied neurons.

Through our experiments, we managed to investigate and prove the multimodality of CLIP in an effective and satisfactory way, considering our limited resources and the small scale of our datasets. These explorations merely skim the surface when it comes to comprehending CLIP’s behavior. For future research, we could experiment with additional models, such as CLAP, and compare the results. An interesting approach would be to research the differences in the modality gap of

both models and embedding spaces.

REFERENCES

- [1] Nandita Bhaskhar. “Intermediate Activations — the forward hook”. In: *Blog: Roots of my Equation (web.stanford.edu/~nanbhas/blog/)* (2020). URL: <https://web.stanford.edu/~nanbhas/blog/forward-hooks-pytorch/>.
- [2] Jeffrey Boschman. *Clip Paper explained easily in 3 levels of detail*. July 2023. URL: <https://medium.com/one-minute-machine-learning/clip-paper-explained-easily-in-3-levels-of-detail-61959814ad13#:~:text=CLIP%2C%20which%20stands%20for%20Contrastive>.
- [3] Tom B. Brown et al. *Adversarial Patch*. 2018. arXiv: 1712.09665 [cs.CV].
- [4] Gabriel Goh et al. “Multimodal Neurons in Artificial Neural Networks”. In: *Distill* (2021). <https://distill.pub/2021/multimodal-neurons>. DOI: 10.23915/distill.00030.
- [5] Taro Kiritani and Koji Ono. *Recurrent Attention Model with Log-Polar Mapping is Robust against Adversarial Attacks*. 2020. arXiv: 2002.05388 [cs.CV].
- [6] Ya Le and Xuan S. Yang. “Tiny ImageNet Visual Recognition Challenge”. In: 2015.
- [7] Weixin Liang et al. *Mind the Gap: Understanding the Modality Gap in Multi-modal Contrastive Representation Learning*. 2022. arXiv: 2203.02053 [cs.CV].
- [8] Colin M. MacLeod. “The Stroop Effect”. In: *Encyclopedia of Color Science and Technology*. Ed. by Ronnier Luo. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1–6. ISBN: 978-3-642-27851-8. DOI: 10.1007/978-3-642-27851-8_67-1. URL: https://doi.org/10.1007/978-3-642-27851-8_67-1.
- [9] Aravindh Mahendran and Andrea Vedaldi. “Visualizing deep convolutional neural networks using natural pre-images”. In: *International Journal of Computer Vision* 120.3 (May 2016), pp. 233–255. DOI: 10.1007/s11263-016-0911-8.
- [10] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. “Inceptionism: Going deeper into neural networks”. In: (2015).
- [11] *OpenAI Microscope*. <https://microscope.openai.com/models>.
- [12] Rodrigo Quiñan Quiroga et al. “Explicit encoding of multimodal percepts by single neurons in the human brain”. In: *Current Biology* 19.15 (Aug. 2009), pp. 1308–1313. DOI: 10.1016/j.cub.2009.06.060.
- [13] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV].

APPENDIX NEURON ACTIVATIONS

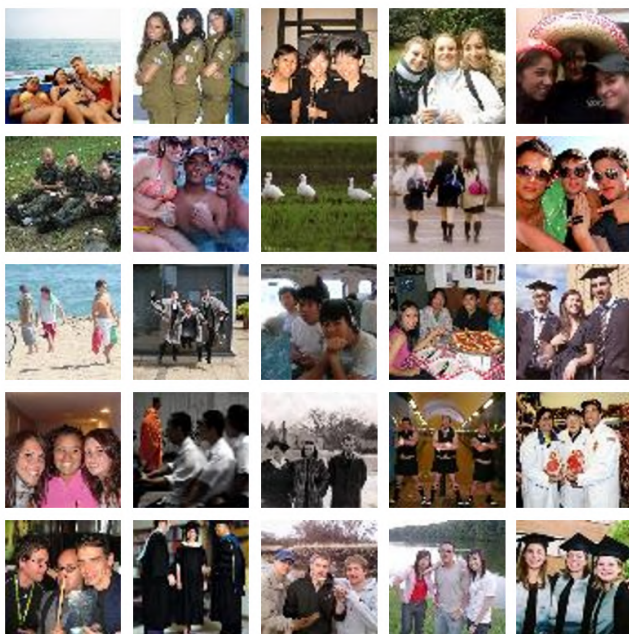


Figure 13. Top 25 images activating neuron 17 - Trios

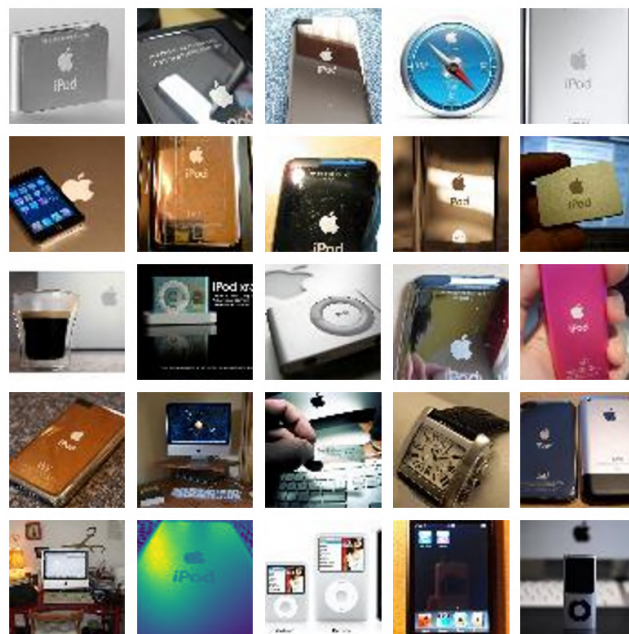


Figure 14. Top 25 images activating neuron 46 - Apple, iPod



Figure 15. Top 25 images activating neuron 293 - Catholicism

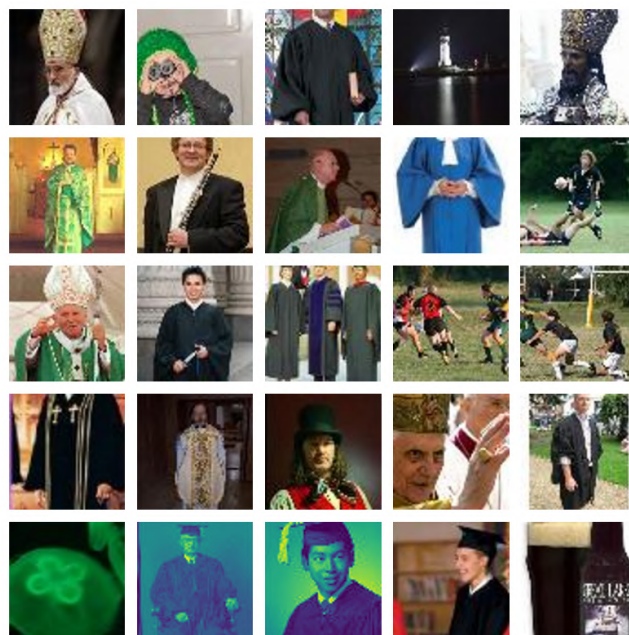


Figure 16. Top 25 images activating neuron 542 - Green



Figure 17. Top 25 images activating neuron 776 - Birthday



Figure 18. Top 25 images activating neuron 1326 - Christmas

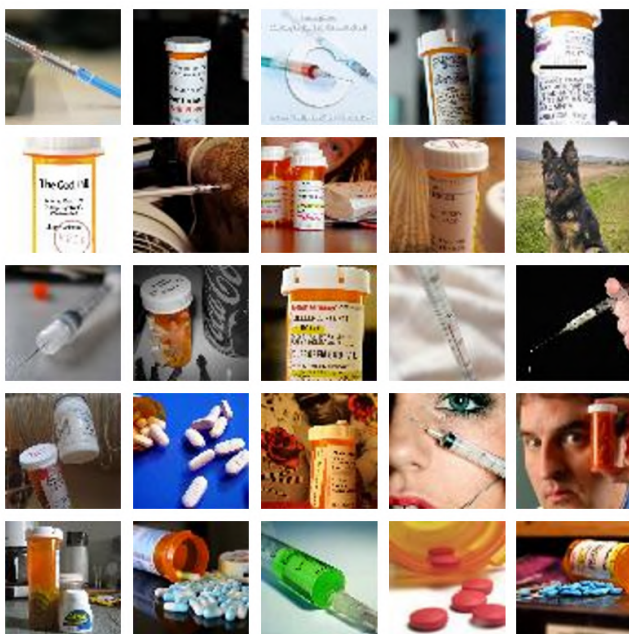


Figure 19. Top 25 images activating neuron 1411 - Needles, pills

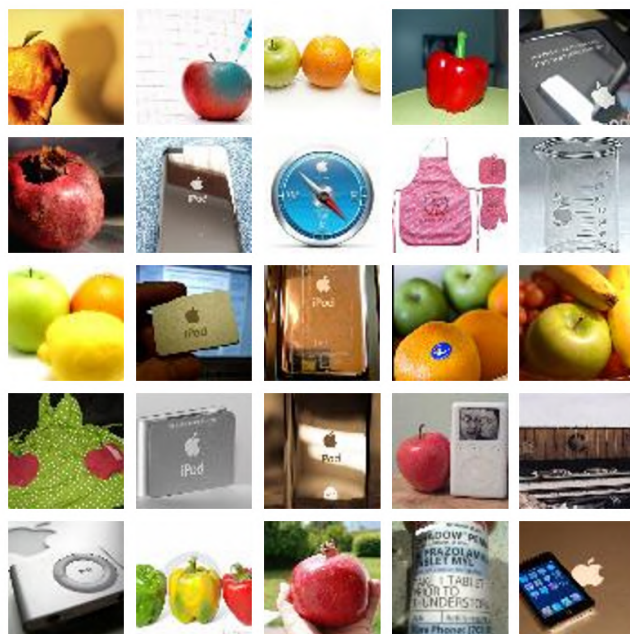


Figure 20. Top 25 images activating neuron 1450 - Apple, apples

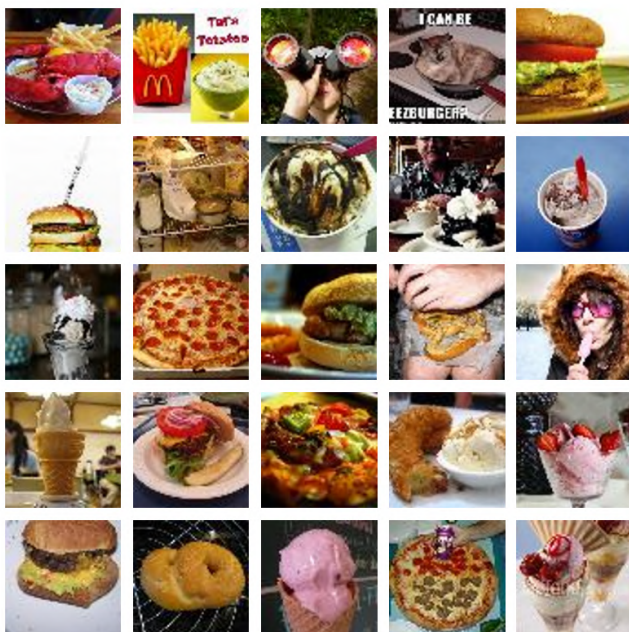


Figure 21. Top 25 images activating neuron 1459 - McDonald's, fast food

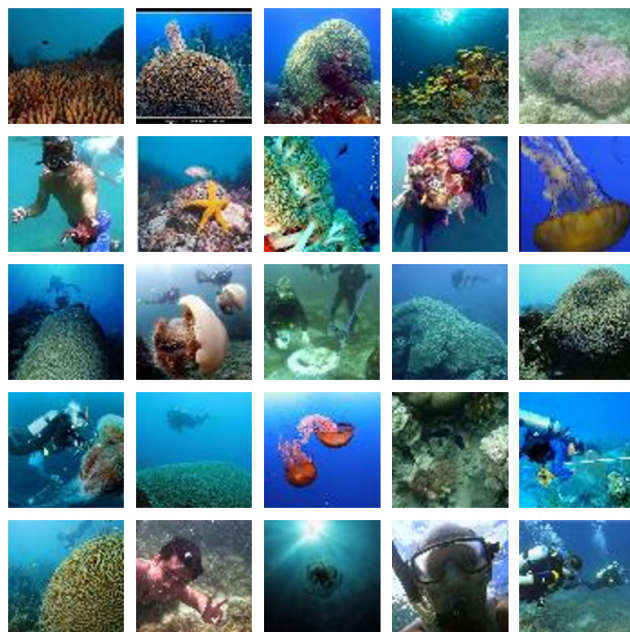


Figure 22. Top 25 images activating neuron 1634 - Underwater



Figure 23. Top 25 images activating neuron 1777 - Jesus

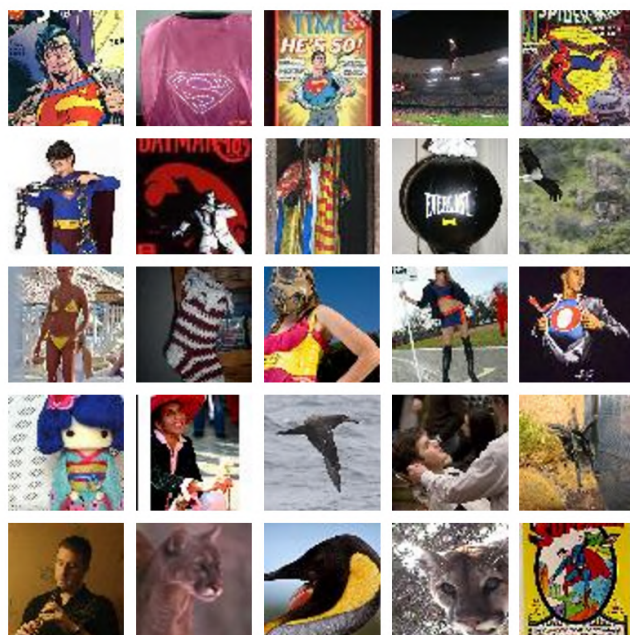
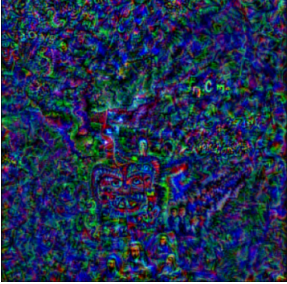
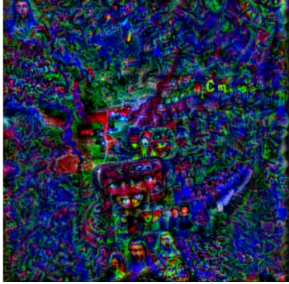
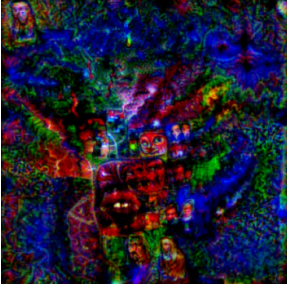
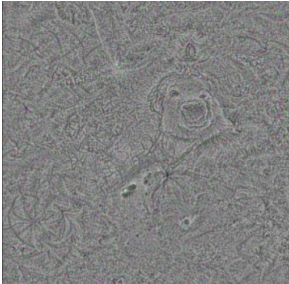
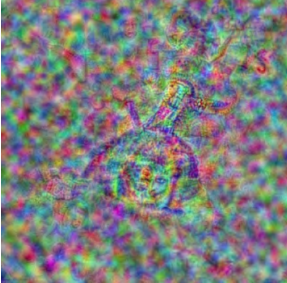
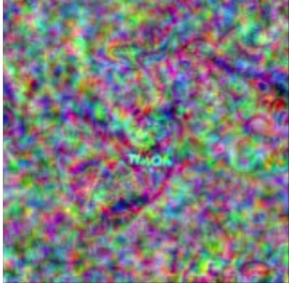
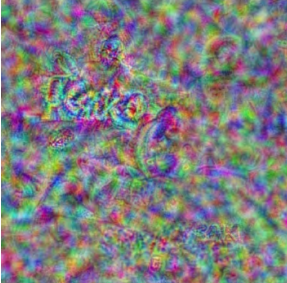
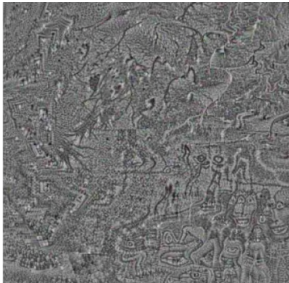


Figure 24. Top 25 images activating neuron 2065 - Superman

APPENDIX
COMPOSITION OF MAXIMAL TEXT

Table III
MAXIMAL TEXT (CONT.)

Image	Maximal Text	Image	Maximal Text
	<ol style="list-style-type: none"> 1) corbyn nepalsatanic infrared 2) yellowstone gaia frequencies grind 3) sarawak minecraft forest goddess 4) aboriginal poisonous museumweek 5) cern geometric dalailama 6) aboriginal electricscope 7) haunts warhammer fractal 8) kilometres angtertcircle 9) frequency ahmadiyya karen beethoven 10) indigenous volunteatlantic 		<ol style="list-style-type: none"> 1) i cern earth chevron 2) masky methane mathematics 3) organic psychedelic marketing 4) northernlights mandala womenof 5) aboriginal yegscotch thepersonalnetwork 6) hindu possessed northern lights 7) mashup many astronomy characteristics 8) botany psychsea religion 9) bolivia gmo metallic light 10) armies surrounding
	<ol style="list-style-type: none"> 1) psychedreal shaw ourable 2) machine learning koturbulent smile 3) hallucinatitute ecoun 4) database migraine sanctioned buda 5) leary rael powerful 6) pai transcaboriginal 7) prose distortion echelensences 8) jaredleto macdonald mural 9) abstraction various instalwith 10) anthropology dogeaster webpage 		<ol style="list-style-type: none"> 1) beingsalmankhan psychedelic pegasus 2) bacteria gif strains gar 3) marbled bacteria slur 4) bacteria ata - erotica 5) bacteria - selfish 6) lama snowflake agon 7) ultrasound monkey agi 8) bacteria geometric selfie 9) tricstencil particles 10) bacteria guerrero
	<ol style="list-style-type: none"> 1) artillery oof satanic 2) seizures pressing notorious 3) itz ches thepersonalnetwork 4) tracnightmares brady thepersonalnetwork 5) doses bonfire fragments imperialism 6) drowning skeletons holy 7) mauritius skeletons holy 8) ecstatic exfolipicasso 9) incense pigs 10) arsenal kas illusion 		<ol style="list-style-type: none"> 1) lenovo koo rubbing weak 2) nikon loofingerprint 3) yokotransatlike lix 4) objecmilkyway laos 5) radiohead fooled ssw ohana 6) yearbook - sotho ultrasound 7) loy pale koh 8) turo bse stash electro 9) kgmak pox 10) lonely mak kobo
	<ol style="list-style-type: none"> 1) frank cuz faze peshwar 2) fik hei felix 3) electromax 4) thanku ako freda 5) ero paprika 6) felixfoley phil 7) enco omile fest 8) frank cze 9) frank tyga collective 10) flipoku 		<ol style="list-style-type: none"> 1) moly yae sightings 2) brightest biomarbolivia 3) consequences underwater javascript madagascar 4) tini tanzania bacteria 5) deep wolf demar 6) travelgram wgnsatisfisu 7) primnationaldogday arkcmeras 8) grids bus bolivia 9) freda tanzania ramatta 10) flowprost khan rejuven

APPENDIX RESPONSE TO STROOP EFFECT

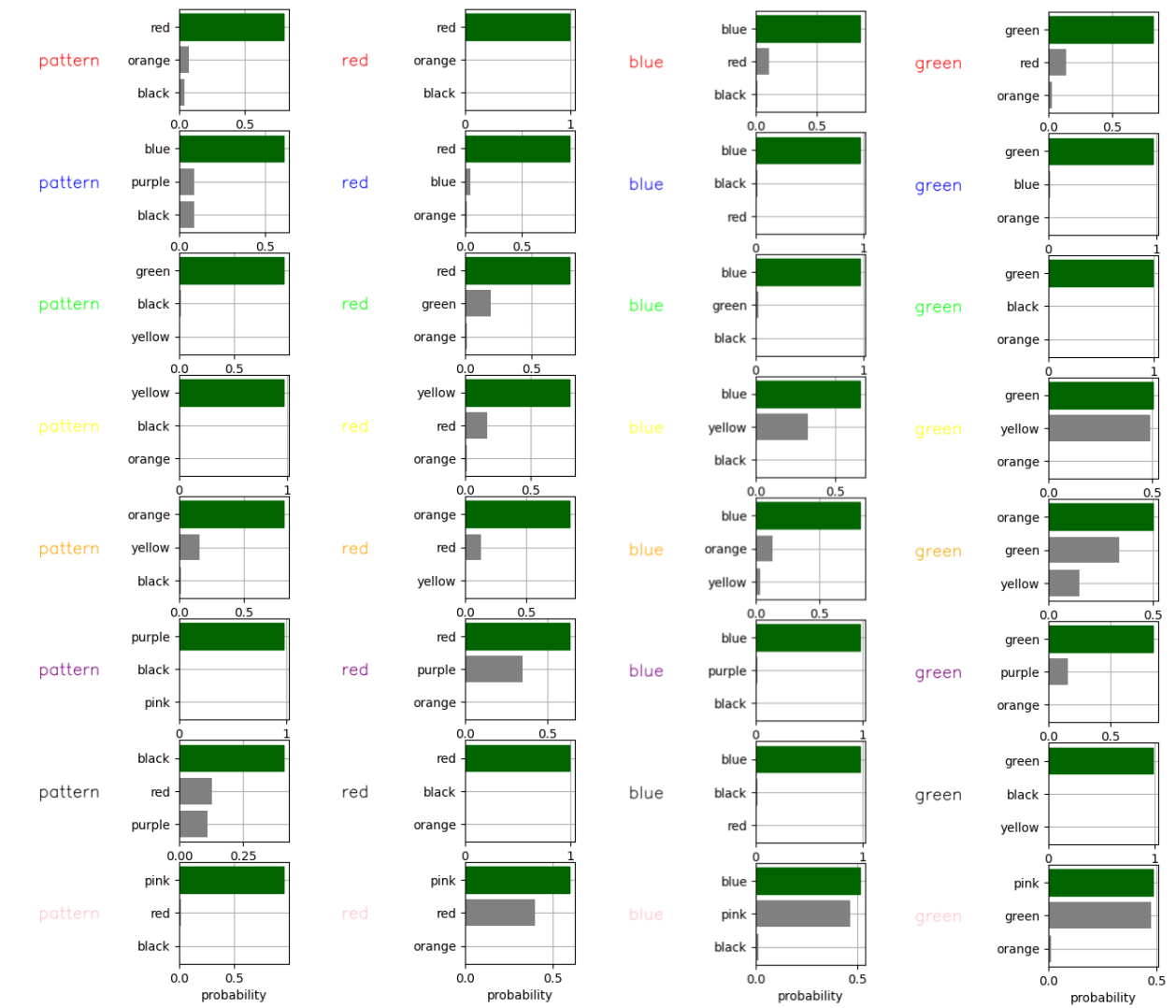


Figure 25. Analyzing response to Stroop Effect



Figure 26. Analyzing response to Stroop Effect