Interpretability of MLP Layers in Vision Transformers

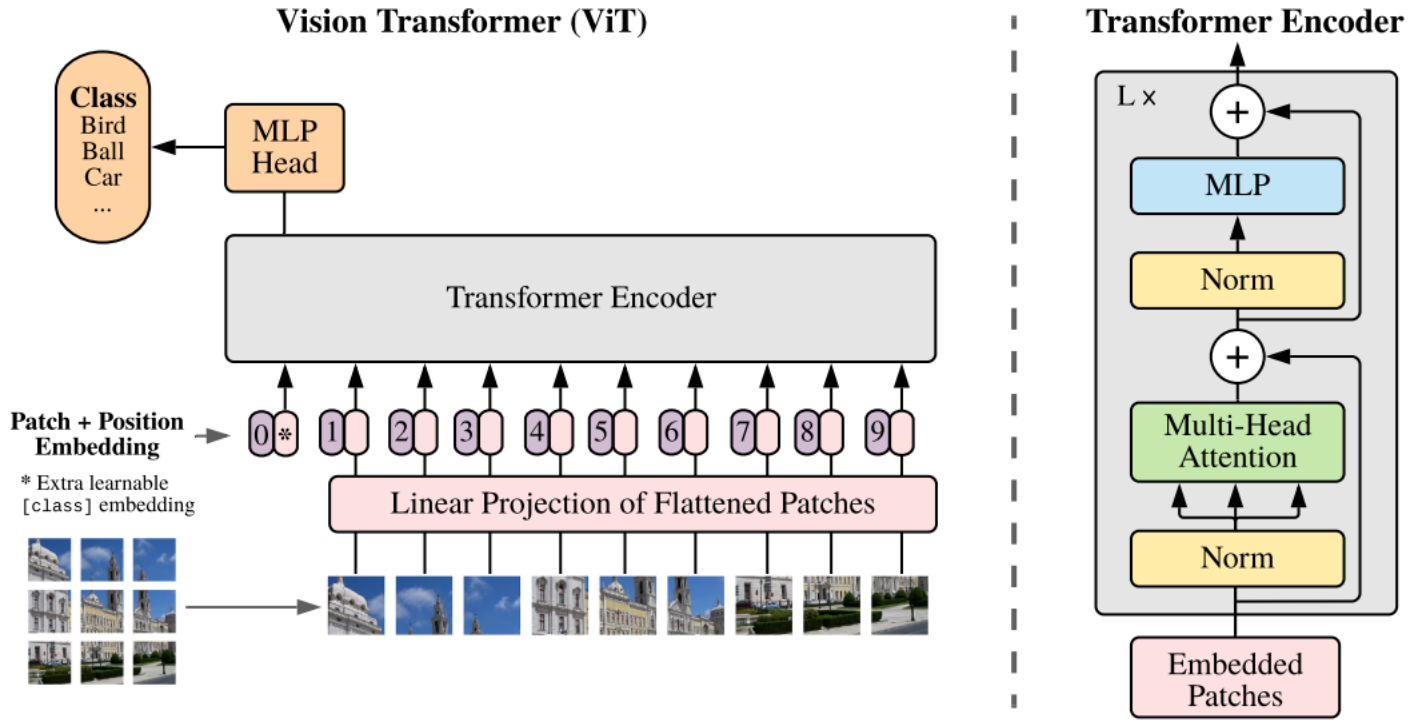
Mika Allert, Ngoc Ahn Trung Pham, Ramazan Özdemir, Moritz Schwerdt

Overview

1. Introduction
2. Experimental Foundation
3. Heatmap Visualization
4. Network Dissection
5. MILAN
6. Stimulative Image Generation
7. Conclusion
8. Future Work

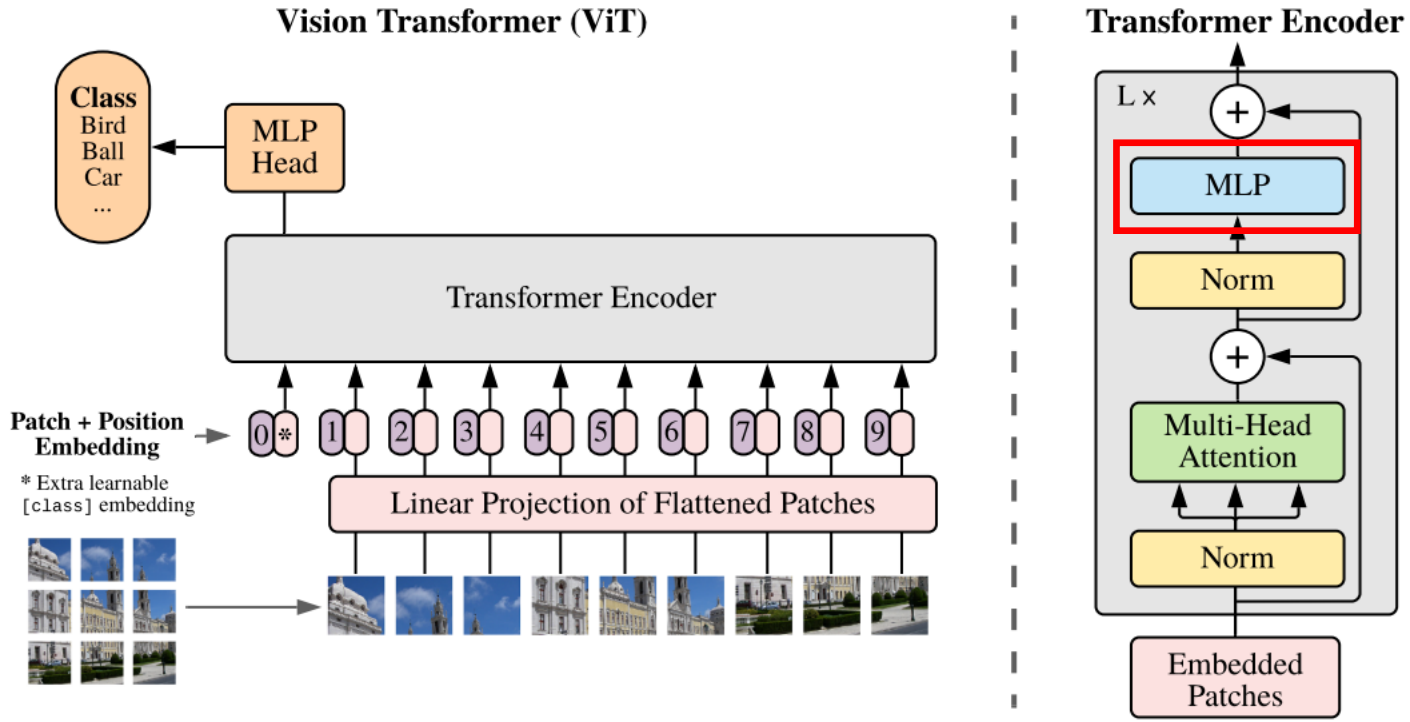
1. Introduction

- Interpretability of a Vision Transformer



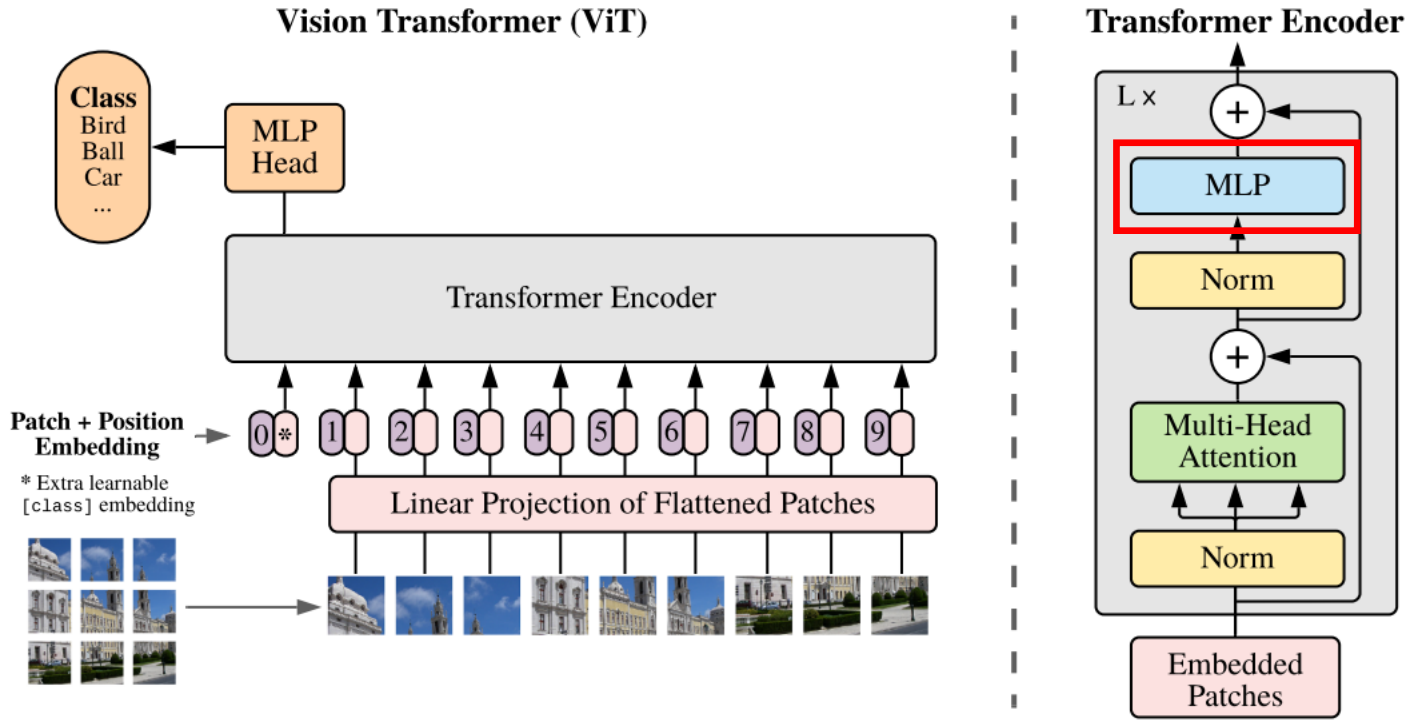
1. Introduction

- Interpretability of a Vision Transformer
- More specifically the MLP Layer in each Block



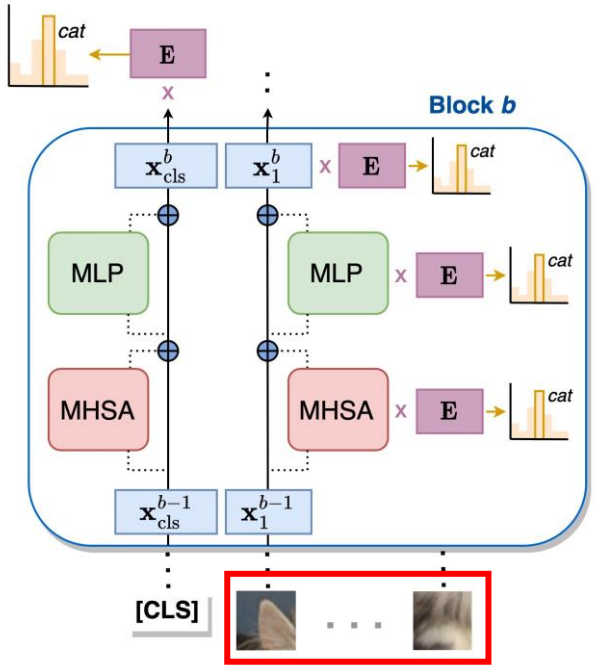
1. Introduction

- Interpretability of a Vision Transformer
- More specifically the MLP Layer in each Block
- Basic Terminology



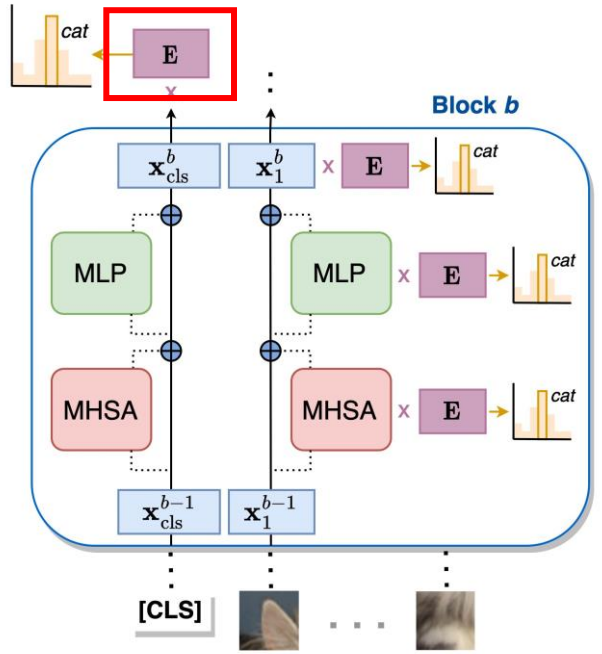
1. Introduction

- Interpretability of a Vision Transformer
- More specifically the MLP Layer in each Block
- Basic Terminology
  - Image Patches



1. Introduction

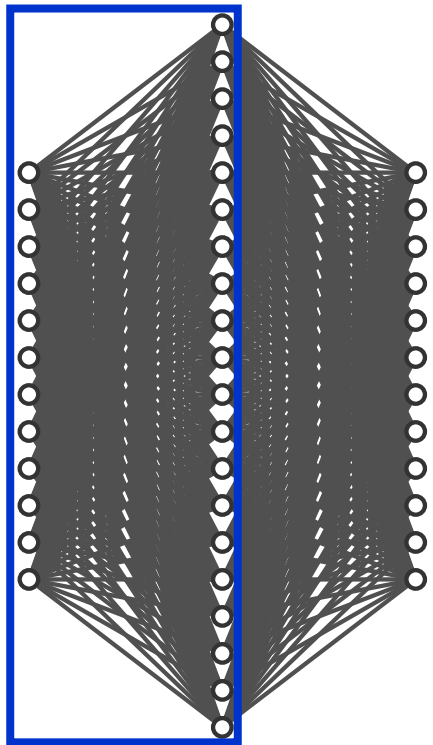
- Interpretability of a Vision Transformer
- More specifically the MLP Layer in each Block
- Basic Terminology
  - Image Patches
  - Embedding Matrix



1. Introduction

- Interpretability of a Vision Transformer
- More specifically the MLP Layer in each Block
- Basic Terminology
  - Image Patches
  - Embedding Matrix
  - Key Vectors W_{inp}

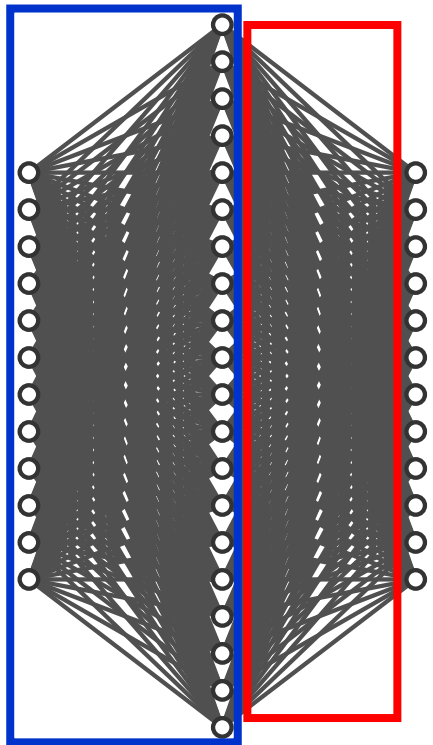
$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$



1. Introduction

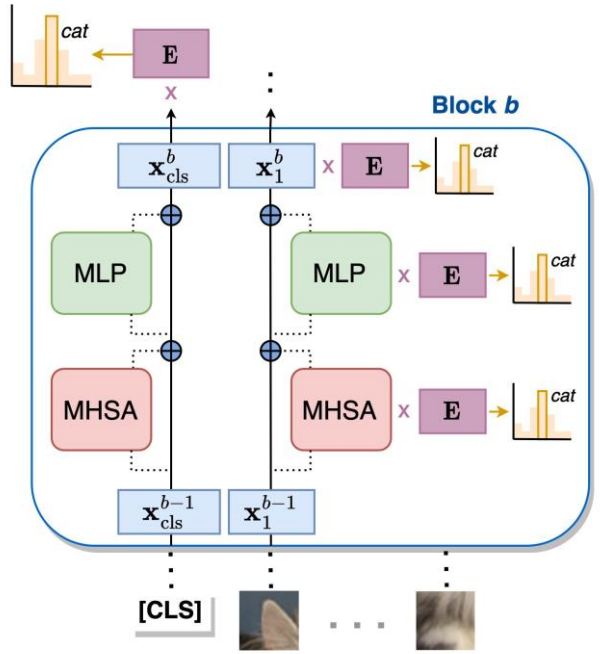
- Interpretability of a Vision Transformer
- More specifically the MLP Layer in each Block
- Basic Terminology
  - Image Patches
  - Embedding Matrix
  - Key Vectors W_{inp}
  - Value Vectors W_{out}

$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$



2. Experimental Foundation (Martina)

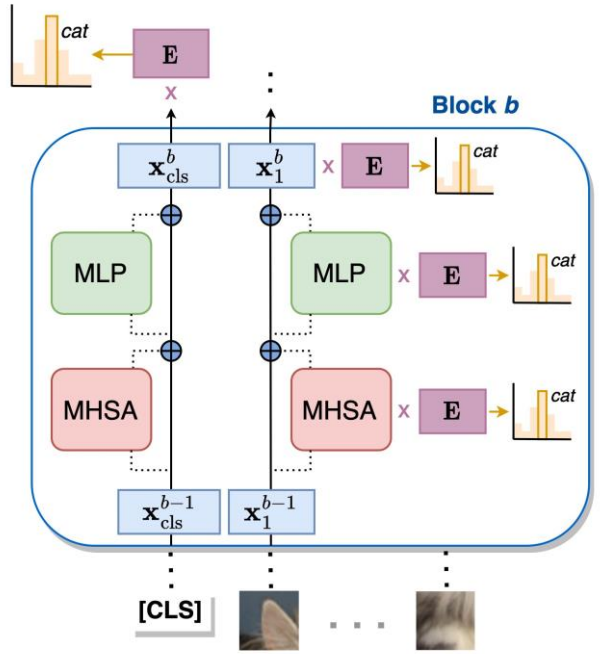
$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$



2. Experimental Foundation (Martina)

- Projection of the value vectors into the class embedding space $E \cdot W_{out}$

$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$

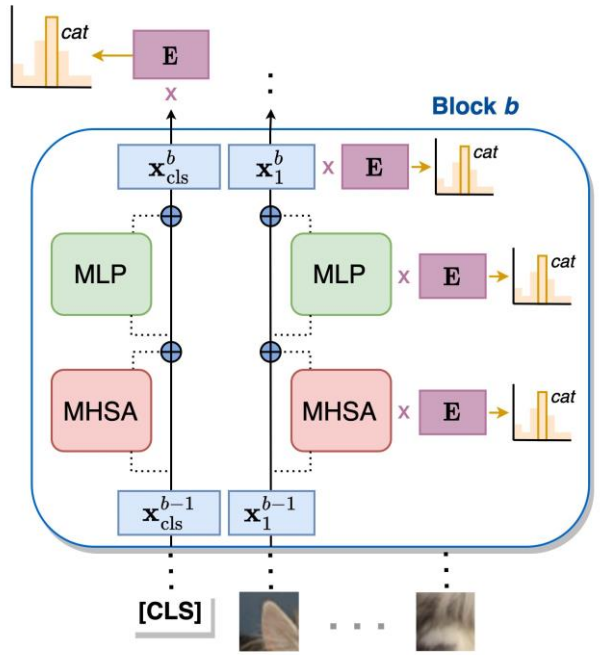


2. Experimental Foundation (Martina)

- Projection of the value vectors into the class embedding space $E \cdot W_{out}$

- For each class, search for the block of W_{out} and row in W_{out} where the class has the maximum probability

$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$



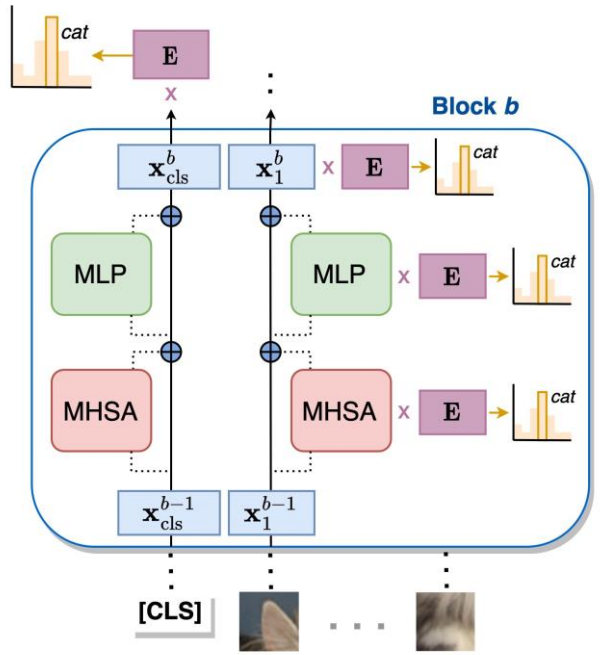
2. Experimental Foundation (Martina)

- Projection of the value vectors into the class embedding space $E \cdot W_{out}$

- For each class, search for the block of W_{out} and row in W_{out} where the class has the maximum probability

- We investigate these and the corresponding key neurons (columns in W_{inp})

$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$



Heatmap visualization - Concepts:

Step 1: Find top key vector for a class

Step 2: Visualize how this key vector reacts to different image patches

Find top key vector for each class

key vectors

$$\mathrm{MLP}(\mathbf{X}) = \mathbf{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$

value vectors

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

$$\mathbf{W}_{\mathrm{inp}} \in \mathbb{R}^{d \times |M|}$$

$$\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{|M| \times d}$$

Geva et al. (2020)

Find top key vector for each class

key vectors

$$\mathrm{MLP}(\mathbf{X}) = \mathbf{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$

value vectors

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

$$\mathbf{W}_{\mathrm{inp}} \in \mathbb{R}^{d \times |M|}$$

$$\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{|M| \times d}$$

n: number of tokens/patches

Geva et al. (2020)

Find top key vector for each class

key vectors

$$\mathrm{MLP}(\mathbf{X}) = \mathbf{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$

value vectors

$\mathbf{X} \in \mathbb{R}^{n \times d}$

$\mathbf{W}_{\mathrm{inp}} \in \mathbb{R}^{d \times |M|}$

$\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{|M| \times d}$

n: number of tokens/patches

d: embedding dimension

Geva et al. (2020)

Find top key vector for each class

key vectors

$$\mathrm{MLP}(\mathbf{X}) = \mathbf{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$

value vectors

$\mathbf{X} \in \mathbb{R}^{n \times d}$ n: number of tokens/patches

$\mathbf{W}_{\mathrm{inp}} \in \mathbb{R}^{d \times |M|}$ d: embedding dimension

$\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{|M| \times d}$ |M|: number of key/value vectors

Geva et al. (2020)

Find top key vector for each class

1. Projection head / class embedding matrix

$$\mathbf{E} \in \mathbb{R}^{d \times |C|}$$

key vectors

$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$

value vectors

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

$$\mathbf{W}_{\mathrm{inp}} \in \mathbb{R}^{d \times |M|}$$

$$\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{|M| \times d}$$

Geva et al. (2020)

Find top key vector for each class

1. Projection head / class embedding matrix

$$\mathbf{E} \in \mathbb{R}^{d \times |C|}$$

2. Multiply:

$$(\mathbf{W}_{out} \cdot \mathbf{E}) \in \mathbb{R}^{|M| \times |C|}$$

key vectors

$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$

value vectors

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

$$\mathbf{W}_{\mathrm{inp}} \in \mathbb{R}^{d \times |M|}$$

$$\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{|M| \times d}$$

Geva et al. (2020)

Find top key vector for each class

1. Projection head / class embedding matrix

$$\mathbf{E} \in \mathbb{R}^{d \times |C|}$$

2. Multiply:

$$(\mathbf{W}_{out} \cdot \mathbf{E}) \in \mathbb{R}^{|M| \times |C|}$$

	cat	dog	fish
val_vec 1	3.4	13.2	1.9
val_vec 2	6.5	2.3	2.1

Geva et al. (2020)

Find top key vector for each class

1. Projection head / class embedding matrix

$$\mathbf{E} \in \mathbb{R}^{d \times |C|}$$

2. Multiply:

$$(\mathbf{W}_{out} \cdot \mathbf{E}) \in \mathbb{R}^{|M| \times |C|}$$

	cat	dog	fish
val_vec 1	3.4	13.2	1.9
val_vec 2	6.5	2.3	2.1

3. For each class choose top k value vectors with highest score

Geva et al. (2020)

Find top key vector for each class

$$(\mathbf{W}_{out} \cdot \mathbf{E}) \in \mathbb{R}^{|M| \times |C|}$$

key vectors

$$\mathrm{MLP}(\mathbf{X}) = \mathrm{GELU}(\mathbf{X}\mathbf{W}_{\mathrm{inp}})\mathbf{W}_{\mathrm{out}}$$

value vectors

Each value vector corresponds to **exactly one** key vector

-> **Top k key vector**

$$\mathbf{X} \in \mathbb{R}^{n \times d}$$

$$\mathbf{W}_{\mathrm{inp}} \in \mathbb{R}^{d \times |M|}$$

$$\mathbf{W}_{\mathrm{out}} \in \mathbb{R}^{|M| \times d}$$

Geva et al. (2020)

Visualize what top key vector looks for in image

Step 1: Take an image from a class

Visualize what top key vector looks for in image

Step 1: Take an image from a class

Step 2: Create feature extractor (use a pretrained ViT)

Visualize what top key vector looks for in image

Step 1: Take an image from a class

Step 2: Create feature extractor (use a pretrained ViT)

Step 3: Use this to extract the activations of the neuron on 196 patches (CLS token omitted)

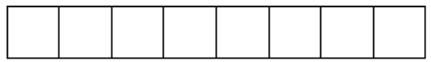
Visualize what top key vector looks for in image

Step 1: Take an image from a class

Step 2: Create feature extractor (use a pretrained ViT)

Step 3: Use this to extract the activations of the neuron on 196 patches (CLS token omitted)

196

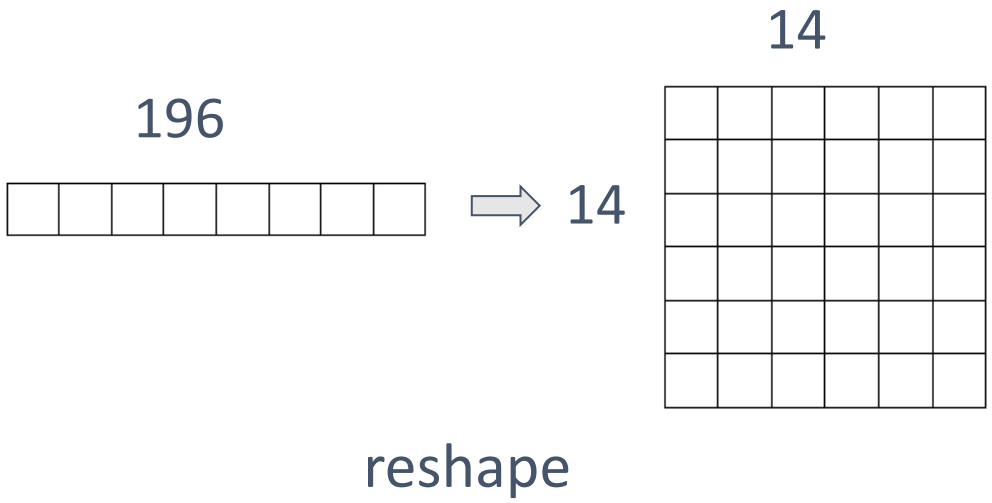


Visualize what top key vector looks for in image

Step 1: Take an image from a class

Step 2: Create feature extractor (use a pretrained ViT)

Step 3: Use this to extract the activations of the neuron on 196 patches (CLS token omitted)

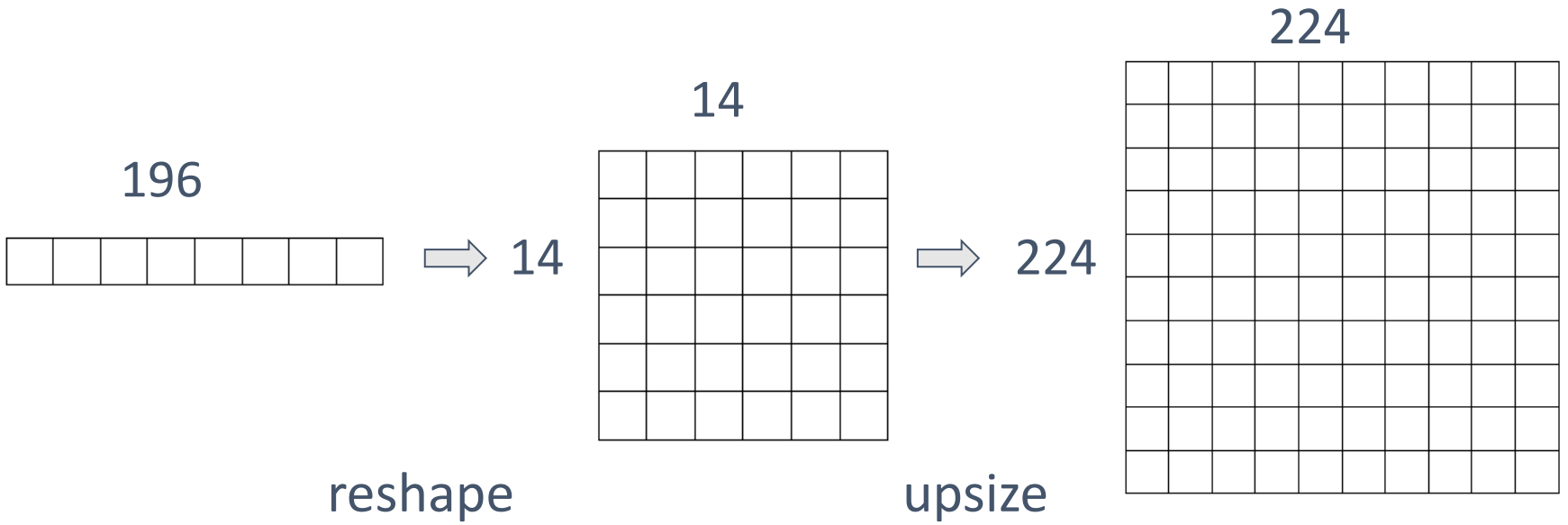


Visualize what top key vector looks for in image

Step 1: Take an image from a class

Step 2: Create feature extractor (use a pretrained ViT)

Step 3: Use this to extract the activations of the neuron on 196 patches (CLS token omitted)

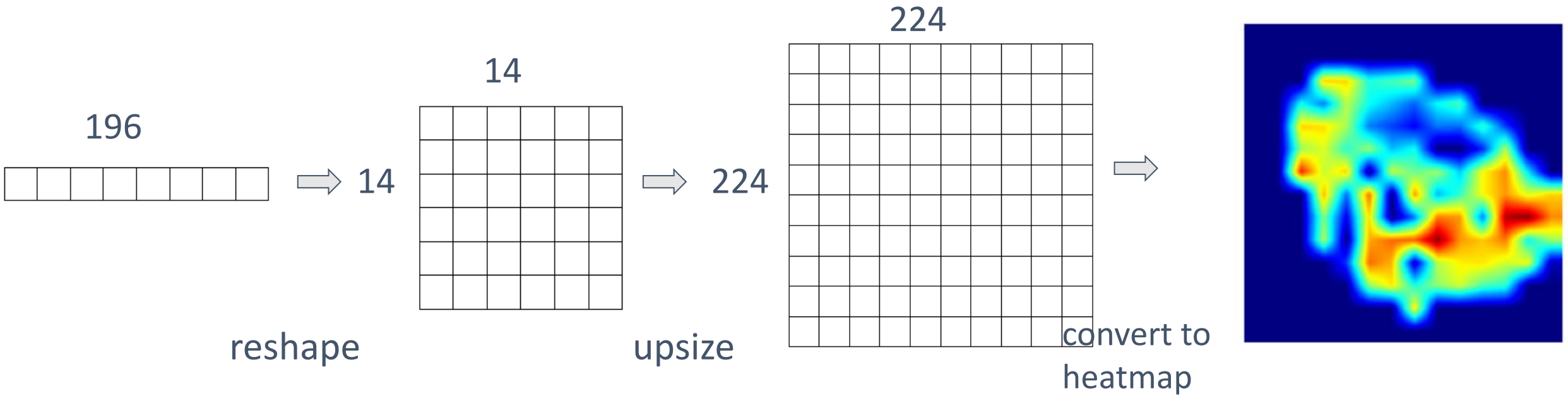


Visualize what top key vector looks for in image

Step 1: Take an image from a class

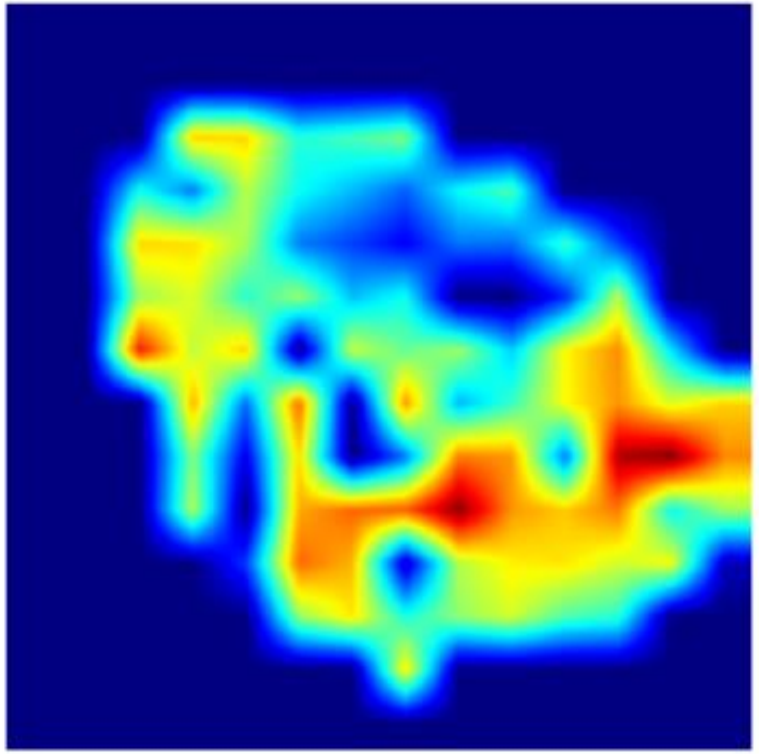
Step 2: Create feature extractor (use a pretrained ViT)

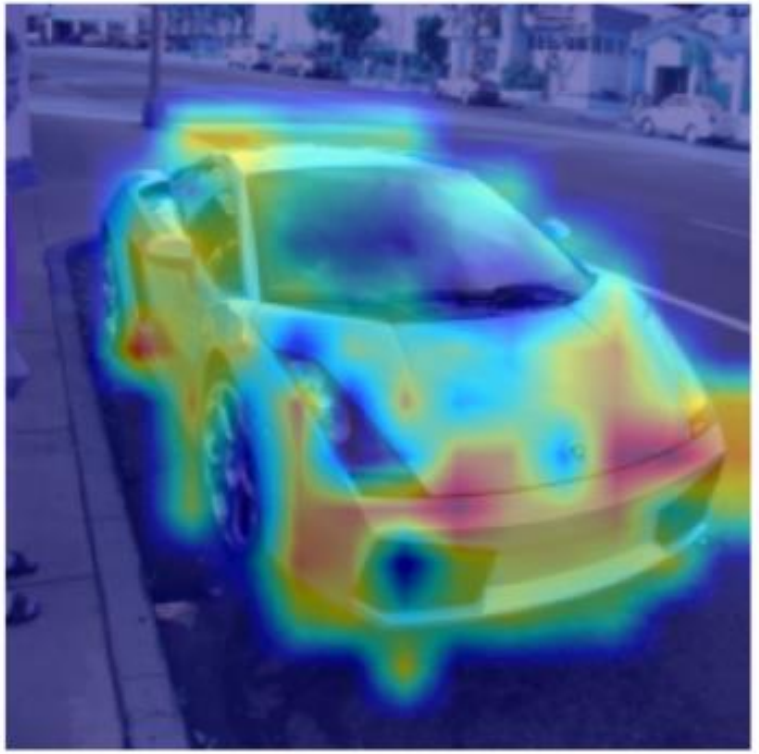
Step 3: Use this to extract the activations of the neuron on 196 patches (CLS token omitted)



Visualize what top key vector looks for in image

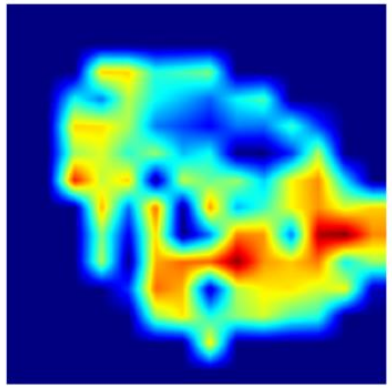
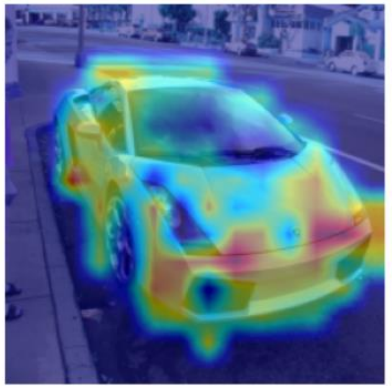
Apply the heat map on the image





Observation and Interpretation

- The key vector (neuron) focus on the highlighted region more than the other.

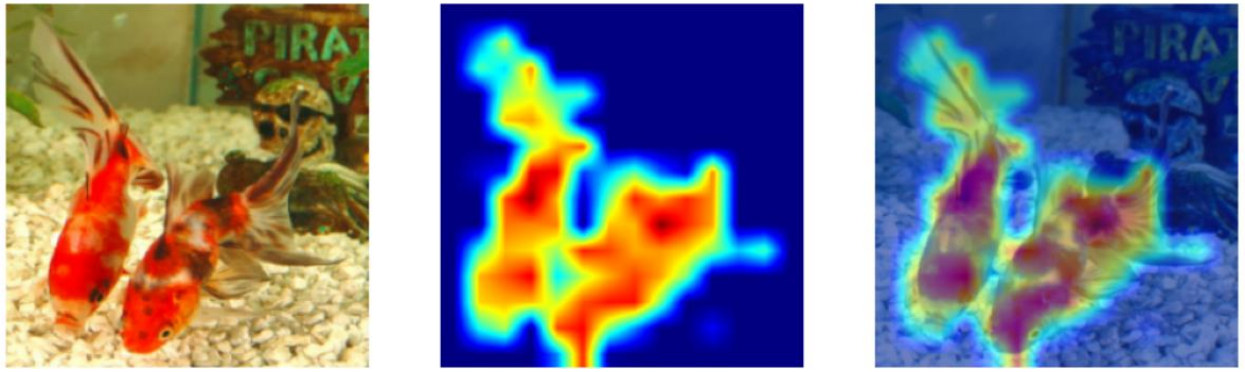




- Most of the object of interest is covered by the highlighted region.

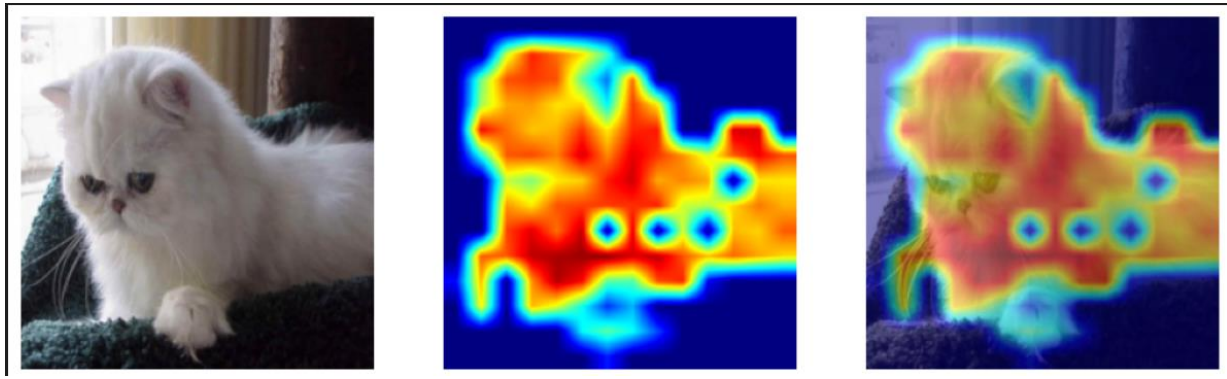
-> The top key vector for a class always look for the object in the image.

Results

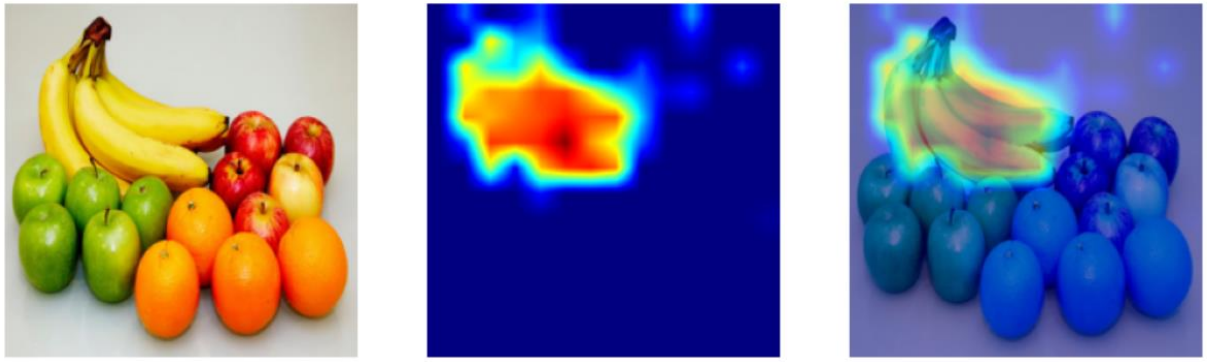
gold fish



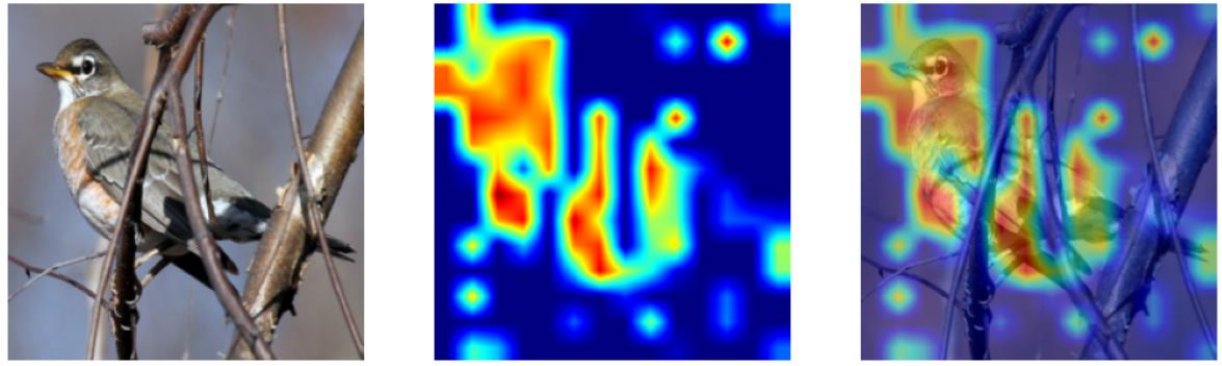
cat



banana



bird



Conclusion and Future work

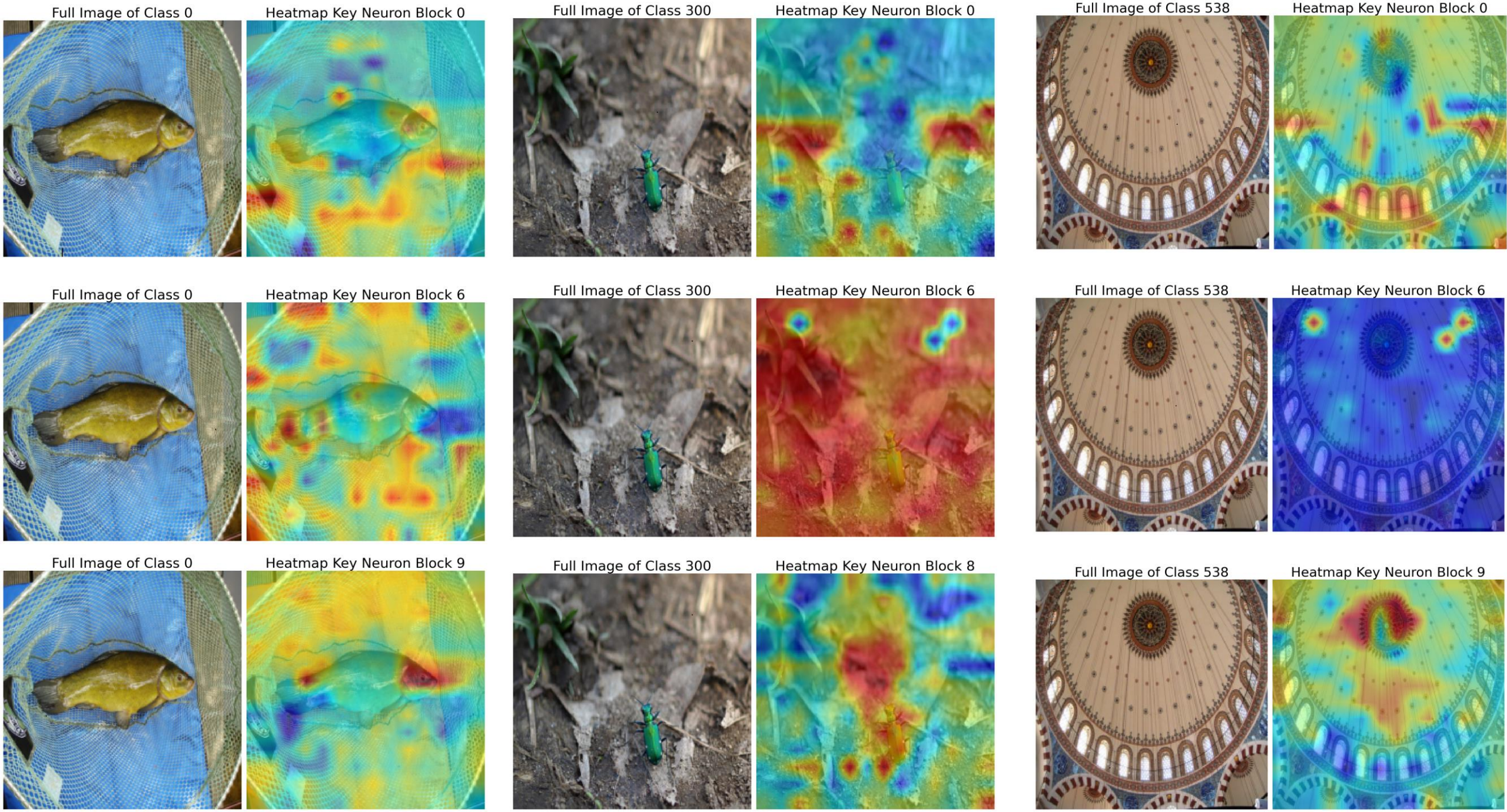
- Conclusion:
  - Heatmap visualization of what a class's top key vector looking for in an image is human-interpretable.
  - An approach based on neural activation for highlighting the object of interest.

- Future work:
  - Try a gradient-based approach, which highlight the pixels belonging to the object of interest.

4. Network Dissection (Concept)

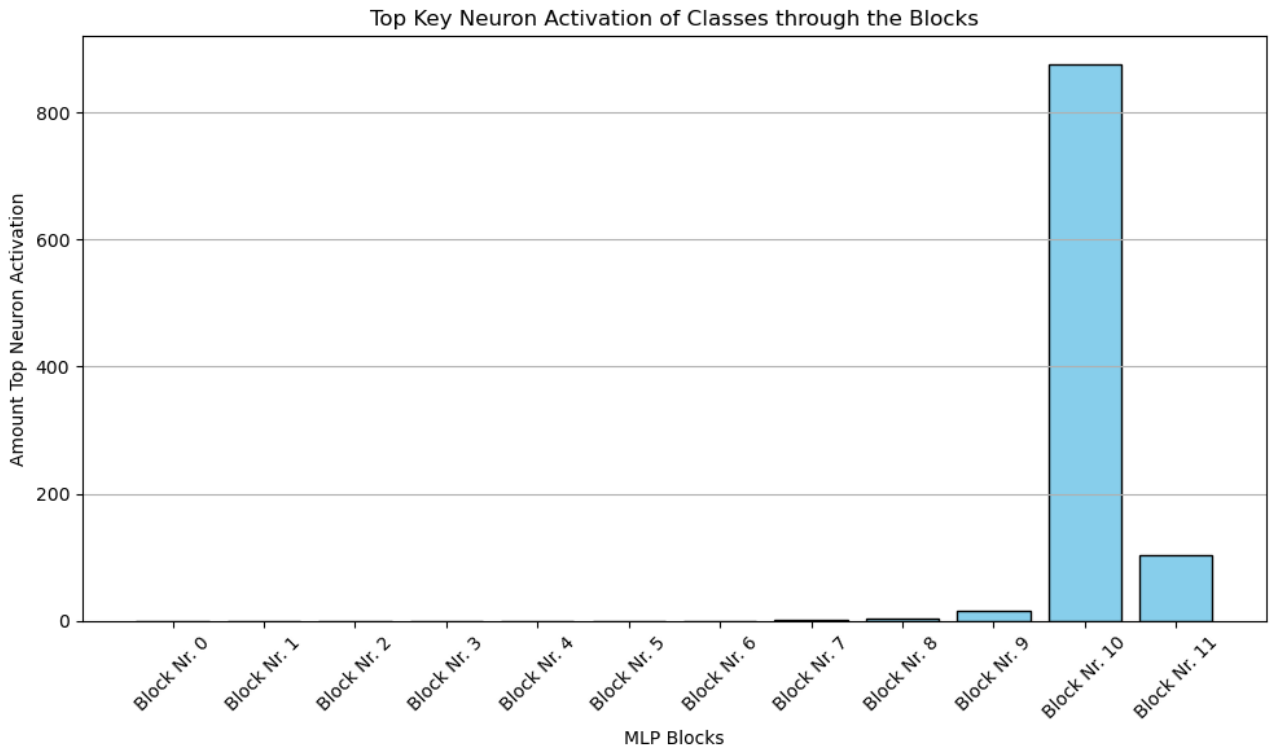
We want to find most activating parts of Images for Class Representation and project them back to Image Space

Source "Network Dissection: Quantifying Interpretability of Deep Visual Representations" Bau et. Al. CVPR 2017

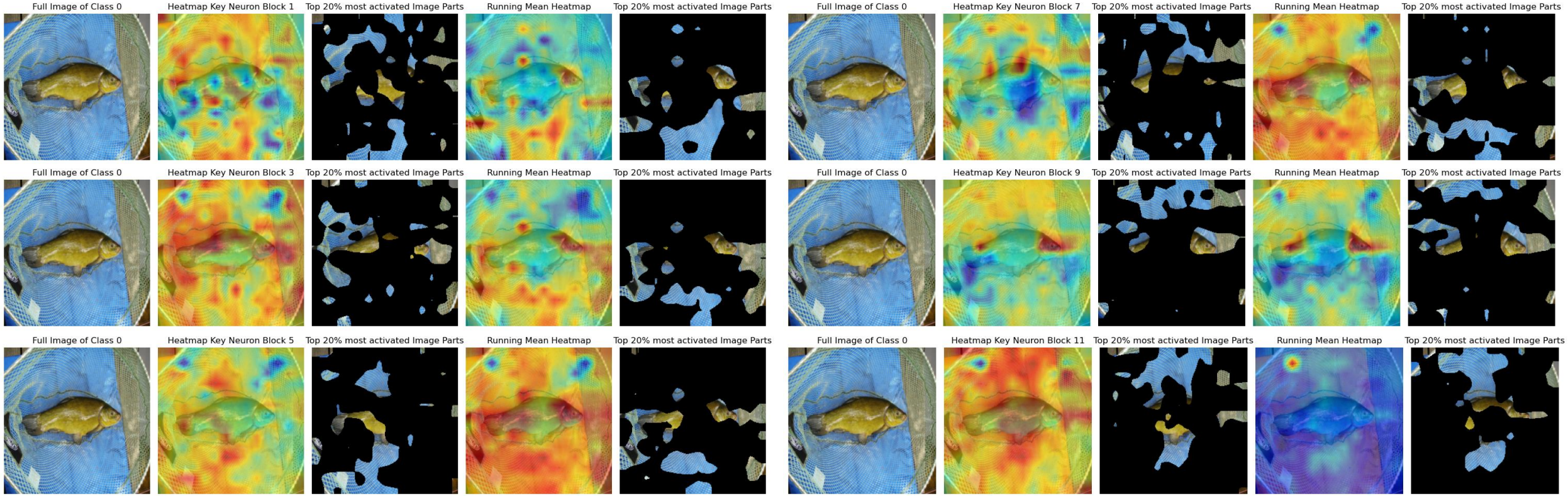

Full Image of Class 0
Heatmap Key Neuron Block 0
Full Image of Class 300
Heatmap Key Neuron Block 0
Full Image of Class 538
Heatmap Key Neuron Block 0
Full Image of Class 0
Heatmap Key Neuron Block 6
Full Image of Class 300
Heatmap Key Neuron Block 6
Full Image of Class 538
Heatmap Key Neuron Block 6
Full Image of Class 0
Heatmap Key Neuron Block 9
Full Image of Class 300
Heatmap Key Neuron Block 8
Full Image of Class 538
Heatmap Key Neuron Block 9

4. Network Dissection (Experiments)

In which Blocks are the Top Key Neuron Activation?

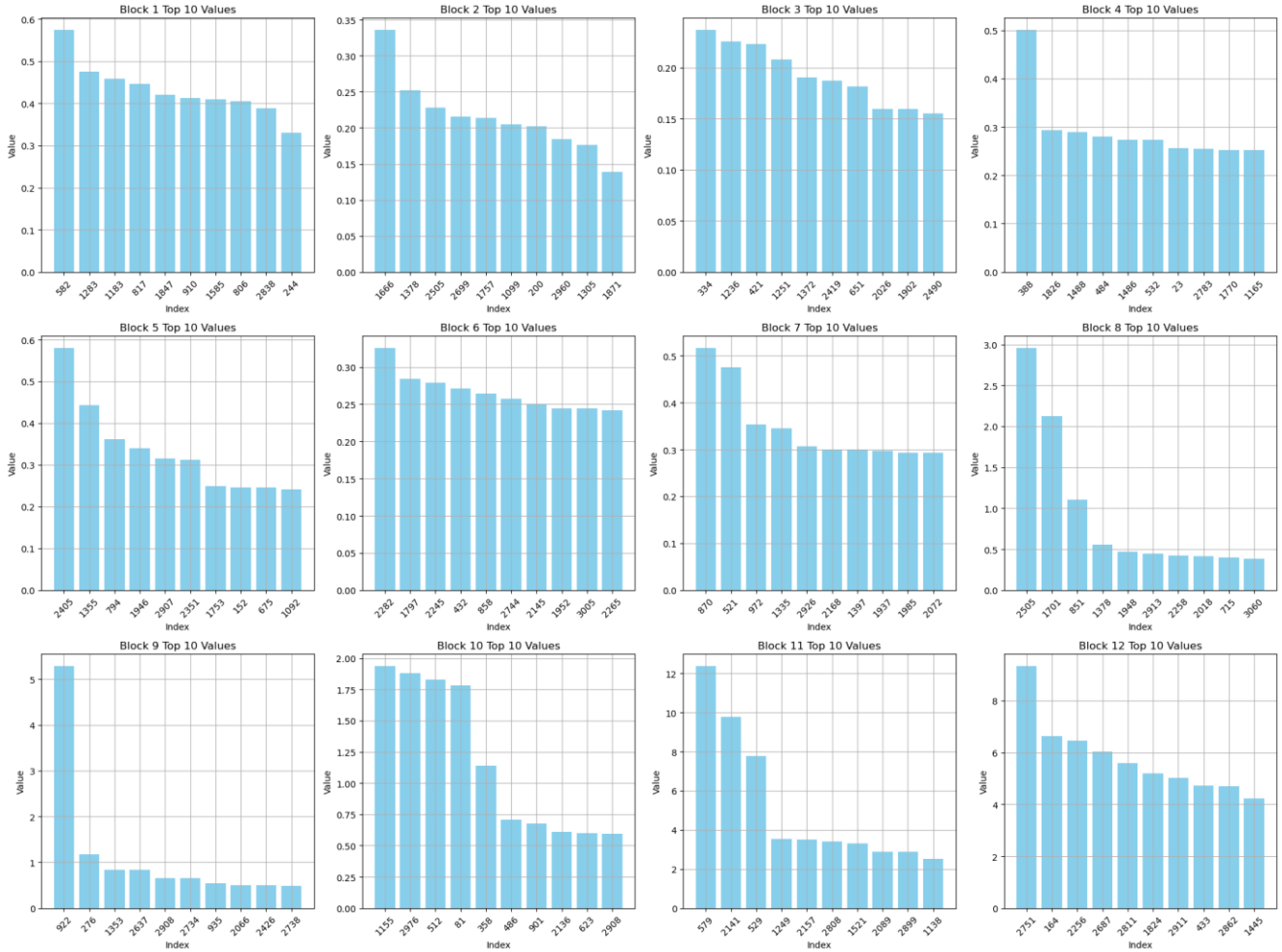


4. Network Dissection (Experiments)



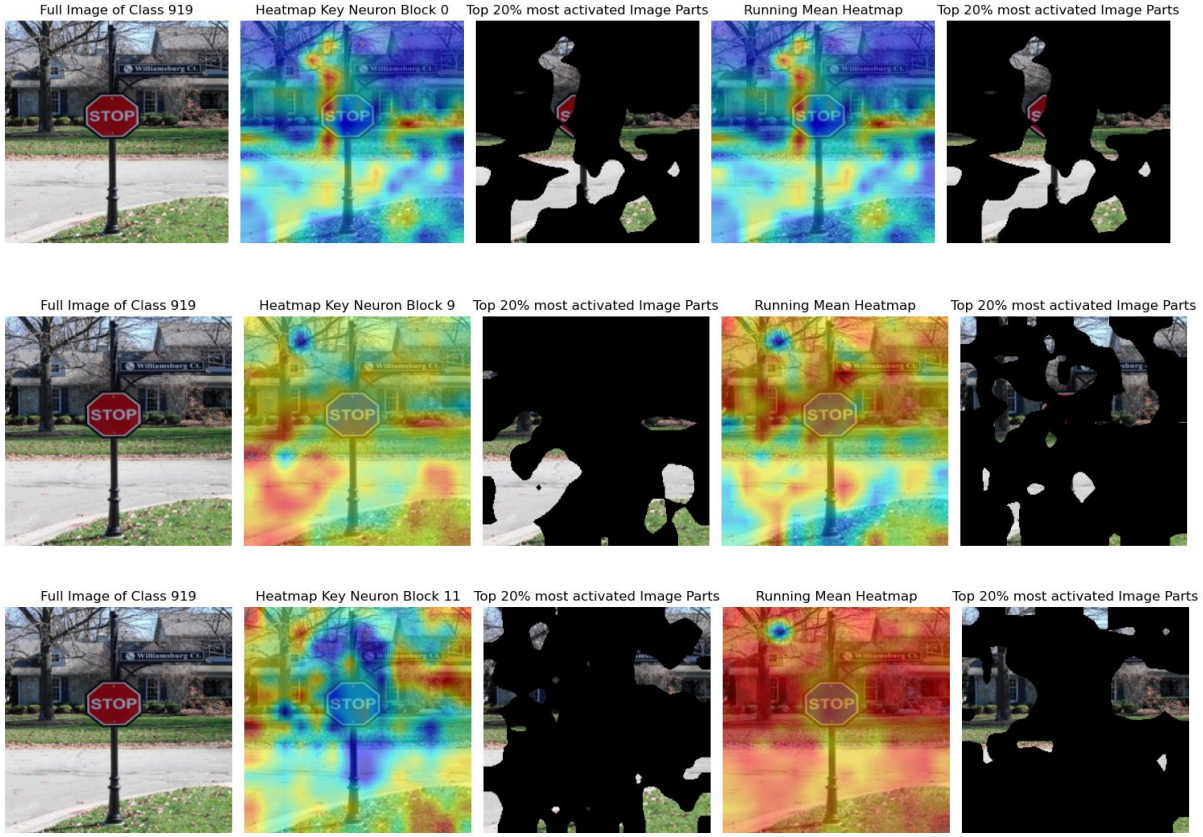
4. Network Dissection (Experiments)

Are Classes represented by only 1 Top Key Neuron?

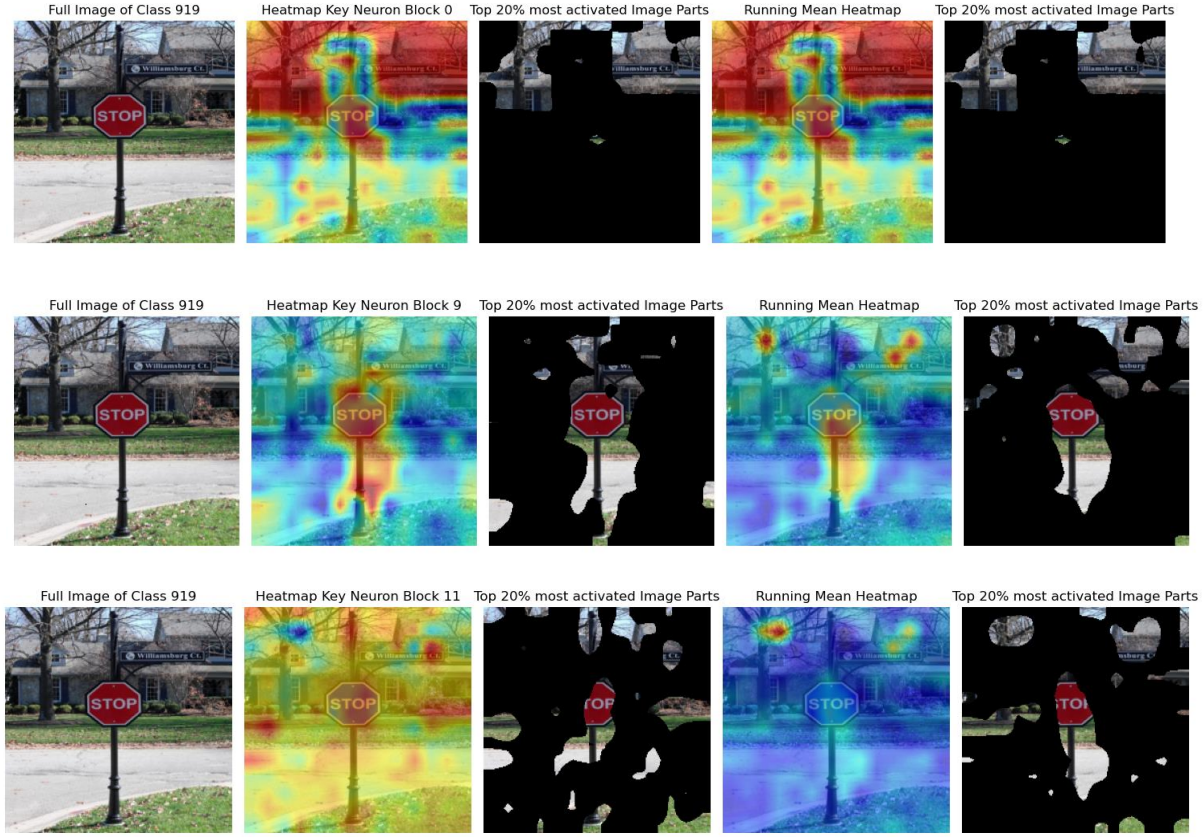


Class 229
(English Sheepdog)

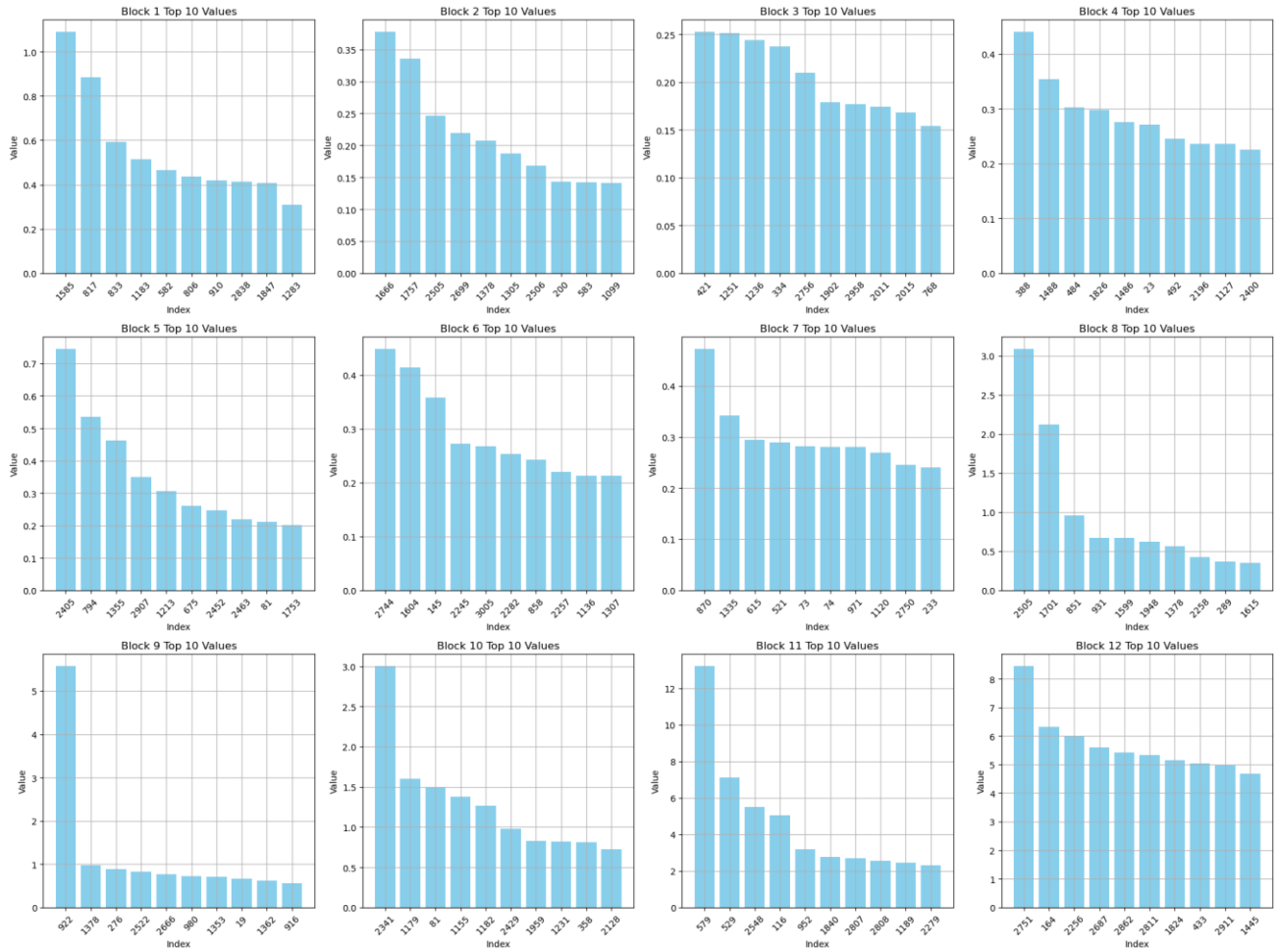
Top1 Key Neuron



Top3 Key Neurons



Top 10 Key Neuron Activation per Block for Class 919



4. Network Dissection Summary

- Key Neuron Activation varies throughout MLP Blocks
- Most Classes are represented by few Key Neurons
- Similar Classes lead to similar Neuron Activations
- Hard to extract inherent meaning for Key Neurons, especially in earlier Blocks

5. MILAN (Mutual-Information-guided Linguistic Annotation of Neurons)

- Procedure to annotate neurons with NLP descriptions
- Represents neurons via a set called Exemplars
  - Exemplars consist of the top n images that cause the greatest activation in the neuron and their respective activation masks indicating the regions in which the images fired the most
- Constructs descriptions for each neuron based on exemplar sets
- How MILAN was trained:
  - Different models neurons were used to obtain set of image regions
  - Human annotations of these obtained image regions are used to train the models

5. Annotating Neurons

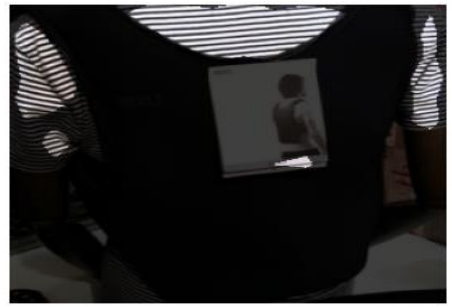
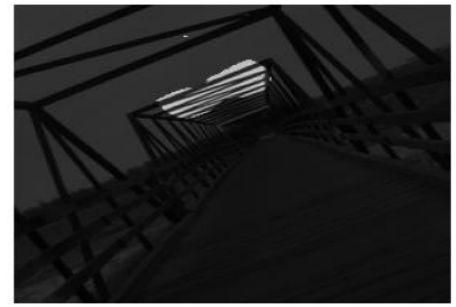
- Model used to get the top activating images is the vit_base_patch16_224 model with pre-trained weights from the timm library
  - Vision Transformer pre-trained on ImageNet-21k at resolution 224x224 and fine-tuned on ImageNet-1k
- In our experiment we used the ImageNet-1k Validation Set (50k Images)
- To give descriptions to the extracted top images we used the base model for MILAN which is trained on all available data
  - All combinations of {alexnet, resnet152, biggan} x {imagenet, places365}

blocks.0.mlp.fc2/unit_0
Description: Stripes











blocks.3.mlp.fc2/unit_6
Description: Spots of the color white



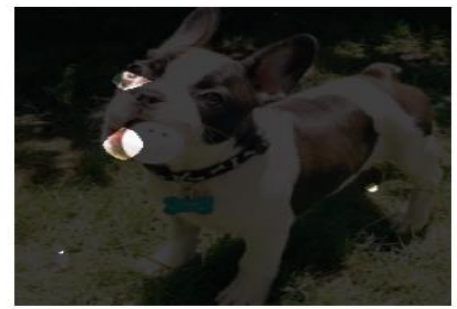




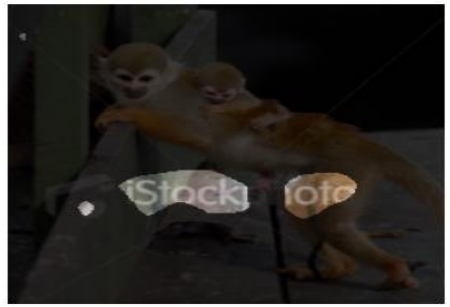
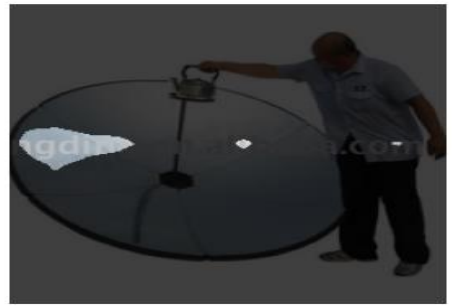




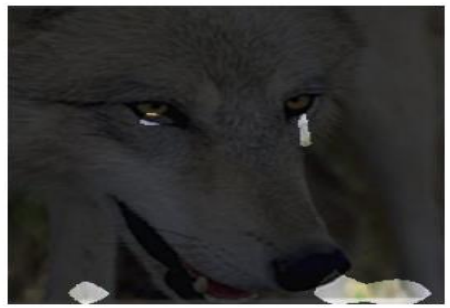


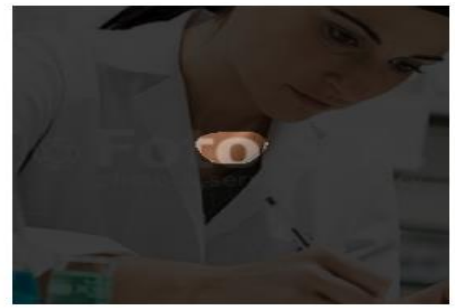



blocks.5.mlp.fc2/unit_24
Description: Bras

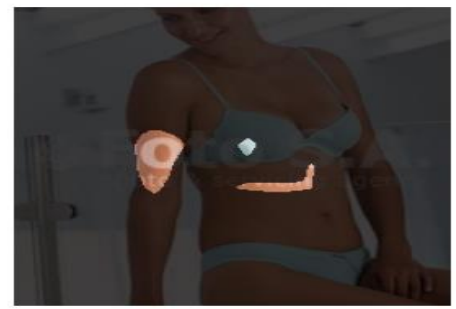






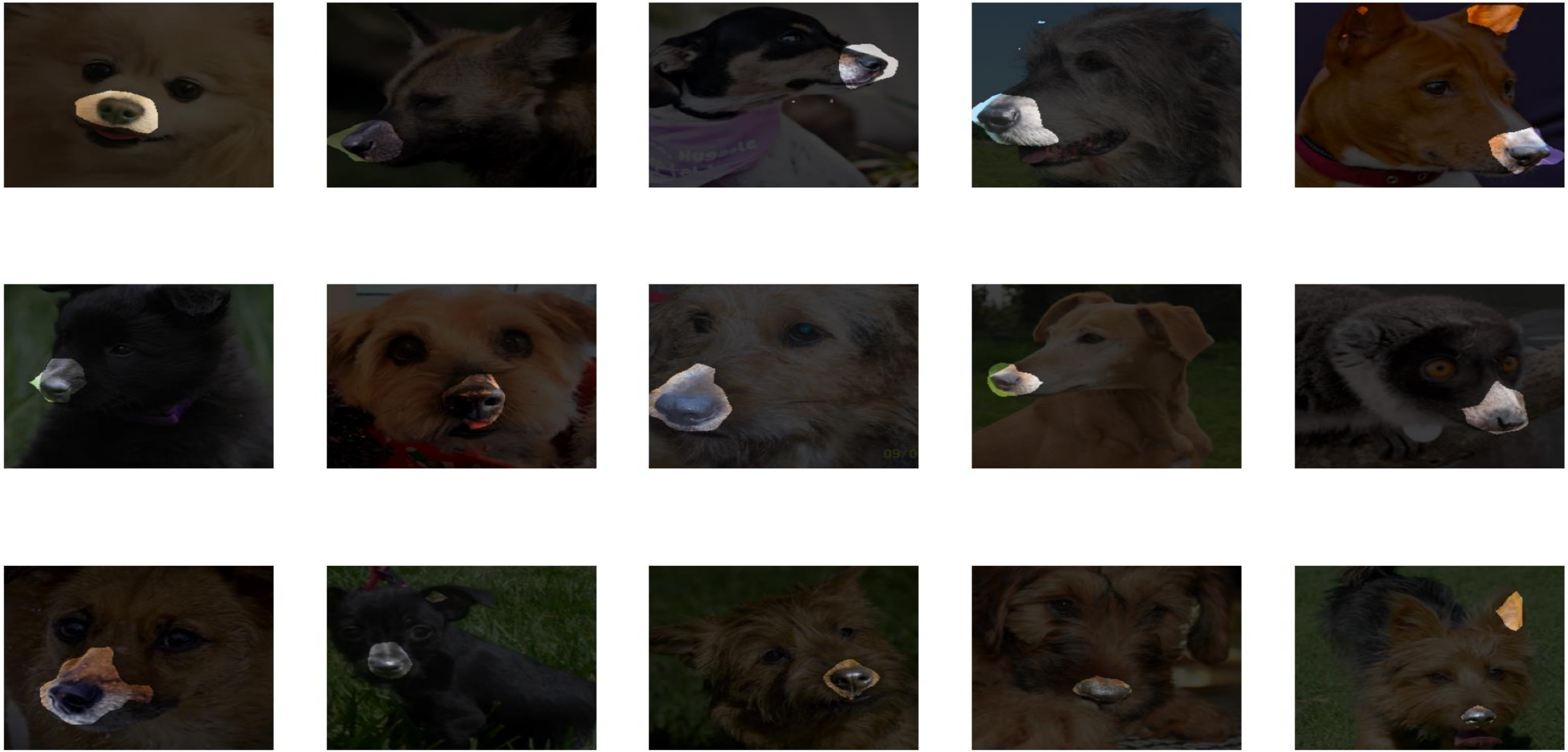


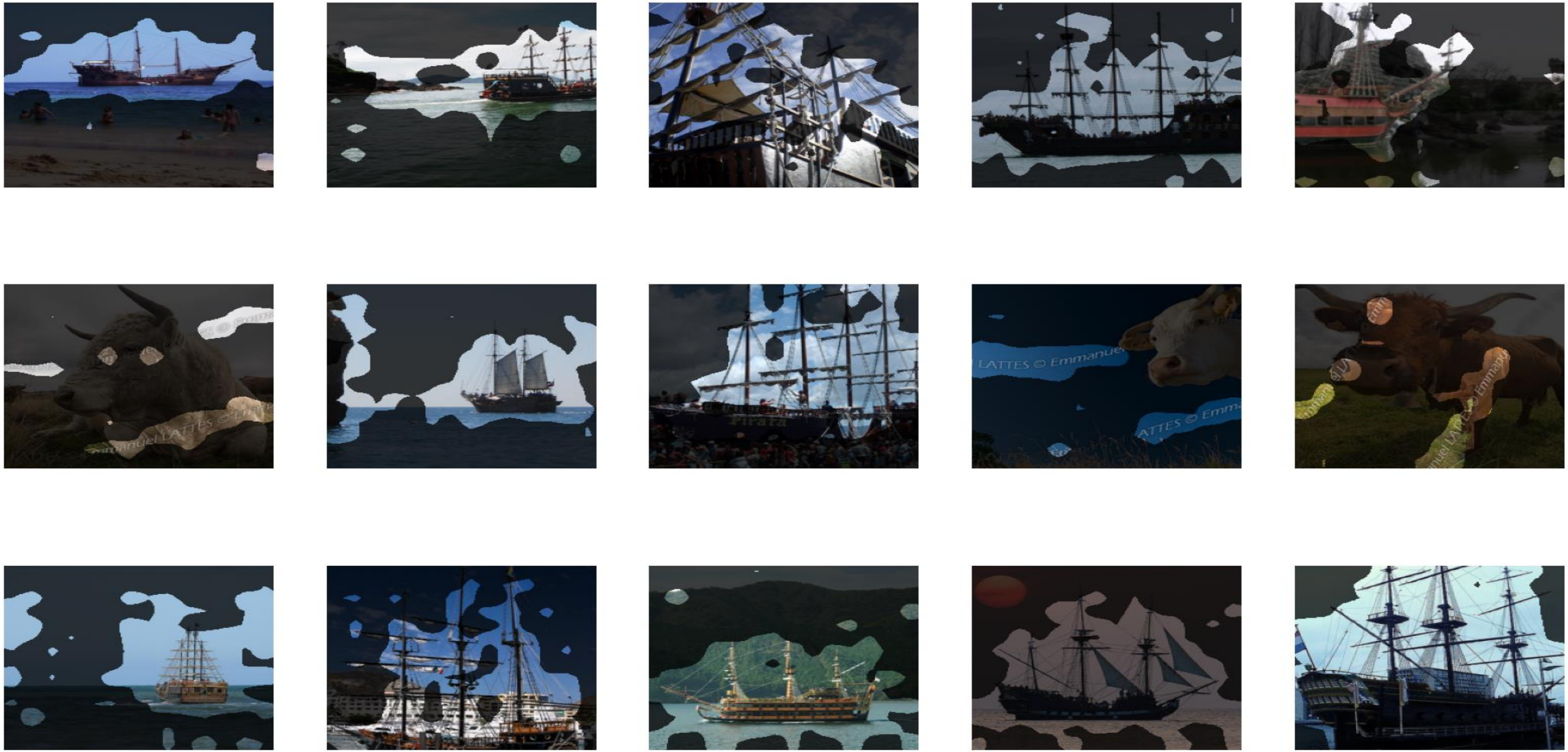




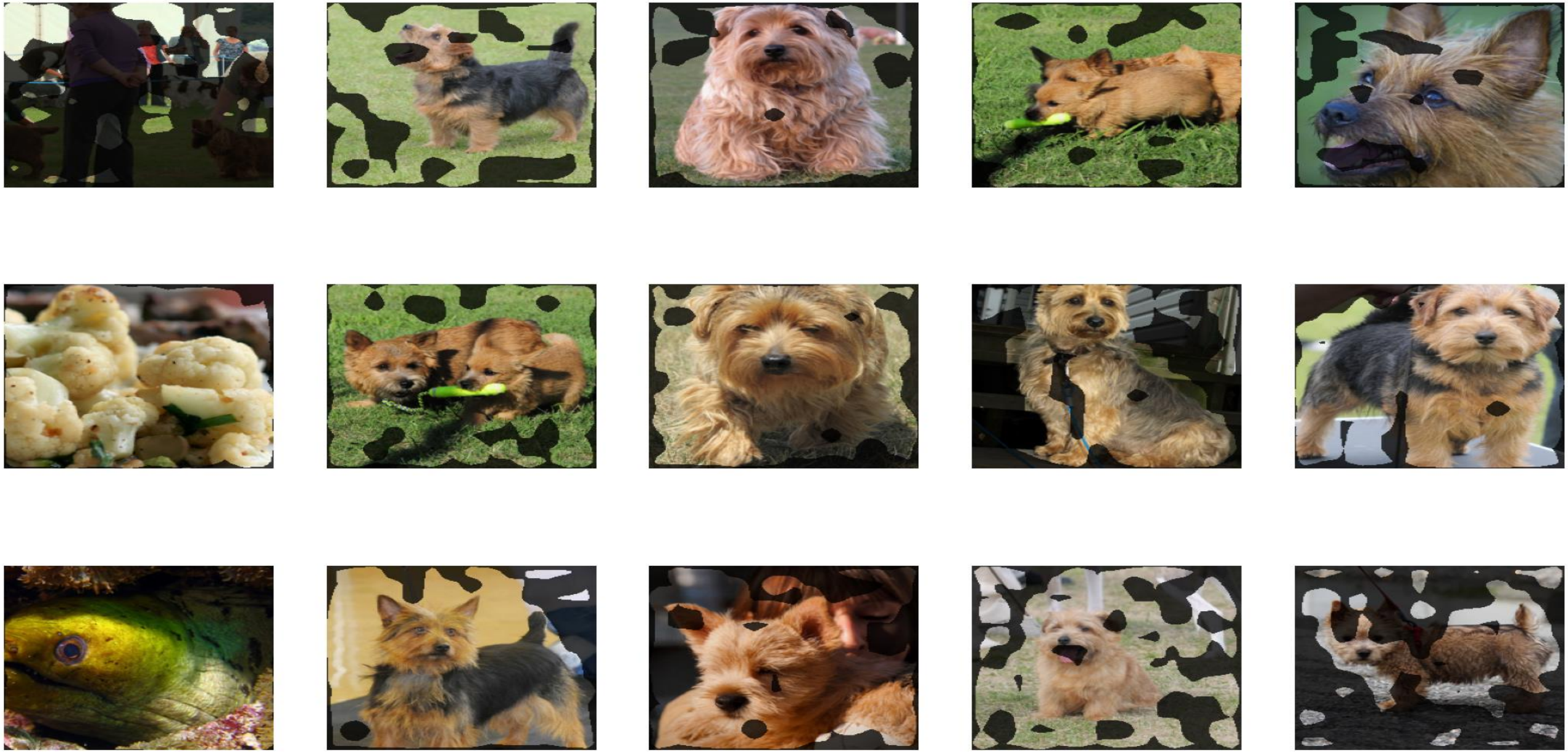

blocks.7.mlp.fc2/unit_39
Description: Dog noses



blocks.10.mlp.fc2/unit_5
Caption: "Ships, ships, and boats"



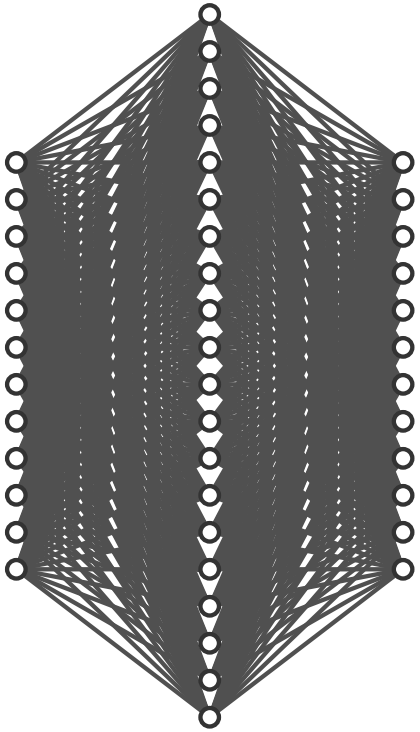
blocks.11.mlp.fc2/unit_2
Description: Animals



5. MILAN Conclusion

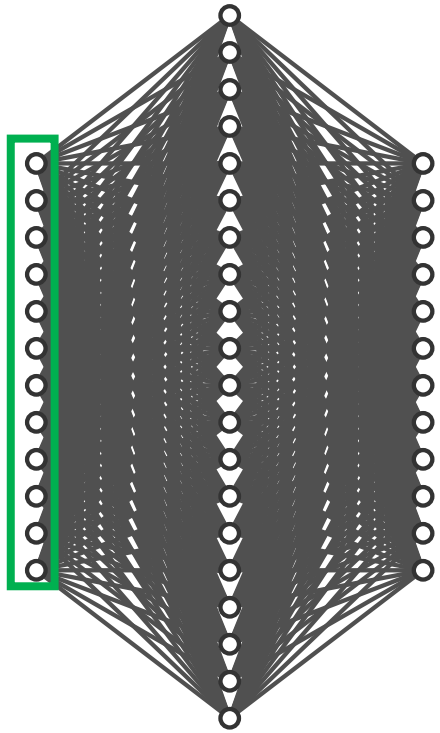
- MILAN gives a lot of general descriptions (Lines, Stripes, Dots, …) especially in the earlier blocks
- Most of the time descriptions don't match the image region where the neuron has the highest activity
- Has a few good descriptions that also fit the image region that was observed in later blocks
- Give us small insight into what neurons might be looking at in images

6. Stimulative Image Generation (Concept)



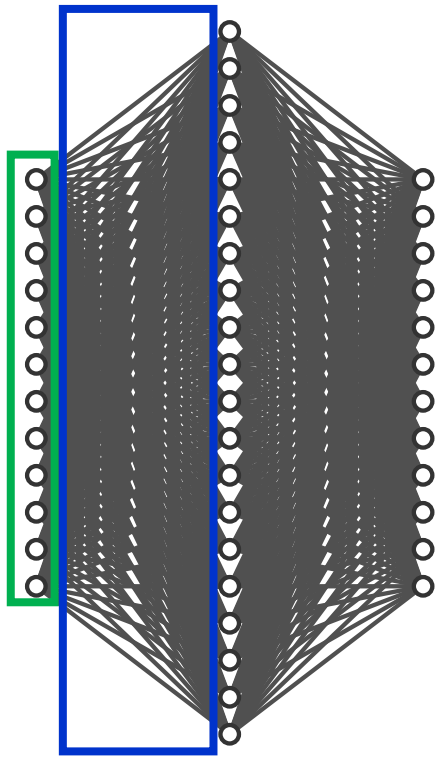
6. Stimulative Image Generation (Concept)

- MLP inputs are patches $\{p_i\}_{i=1}^{N}, p_i \in \mathbb{R}^d$



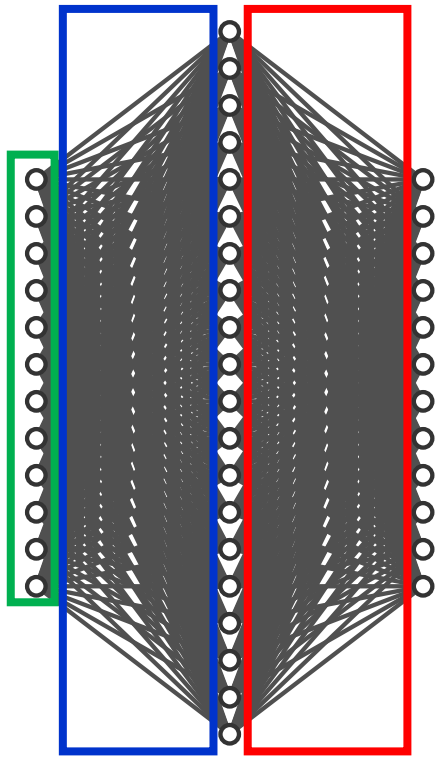
6. Stimulative Image Generation (Concept)

- MLP inputs are patches $\{p_i\}_{i=1}^{N}, p_i \in \mathbb{R}^d$
- Patches multiplied with key neurons (columns of W_{inp}) $h_i = p_i W_{inp}$



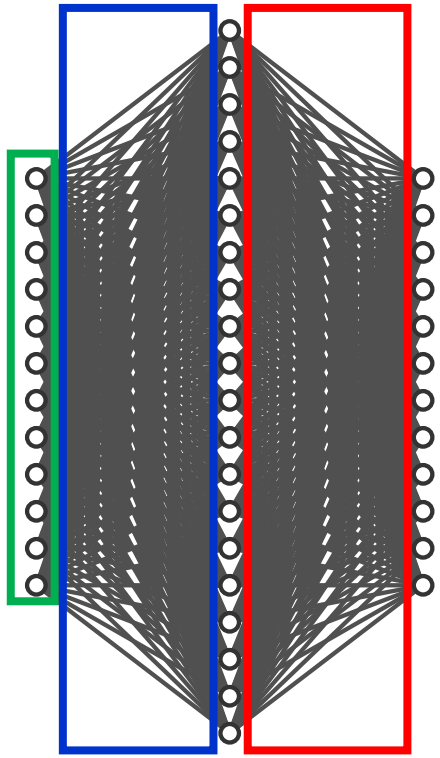
6. Stimulative Image Generation (Concept)

- MLP inputs are patches $\{p_i\}_{i=1}^{N}, p_i \in \mathbb{R}^d$
- Patches multiplied with key neurons (columns of W_{inp}) $h_i = p_i W_{inp}$
- Every key neuron corresponds to a value neuron (row of W_{out})

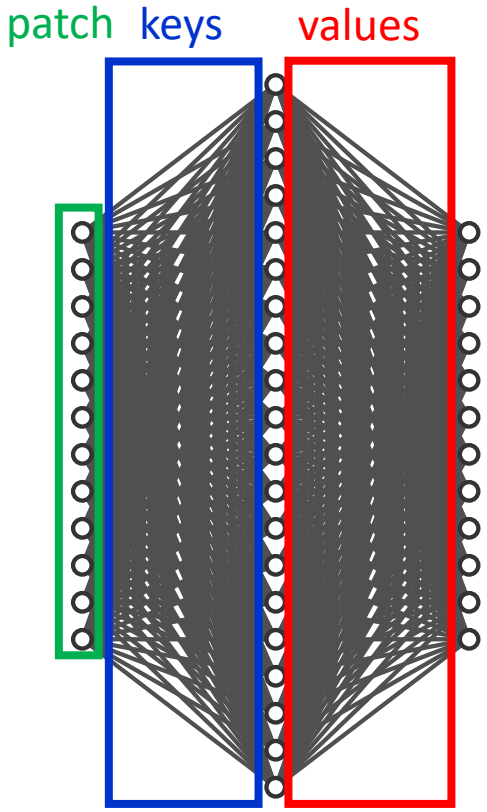


6. Stimulative Image Generation (Concept)

- MLP inputs are patches $\{p_i\}_{i=1}^{N}, p_i \in \mathbb{R}^d$
- Patches multiplied with key neurons (columns of W_{inp}) $h_i = p_i W_{inp}$
- Every key neuron corresponds to a value neuron (row of W_{out})
- For each patch and a singular key neuron we get a scalar value h_{ij}

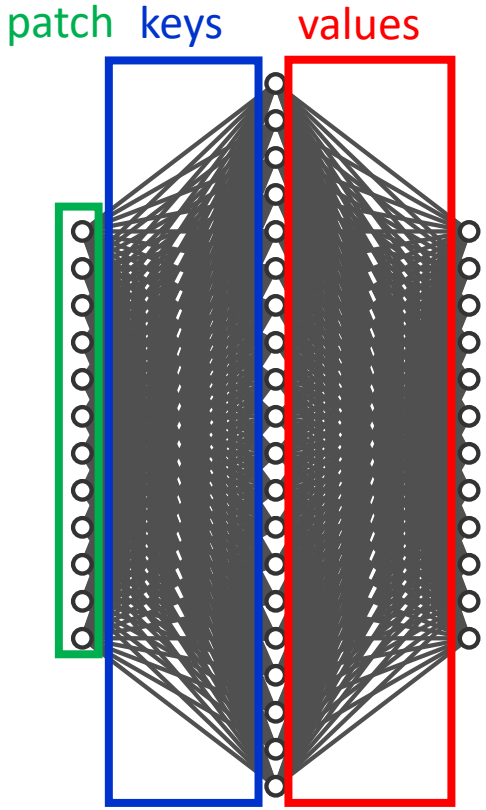


6. Stimulative Image Generation (Concept)



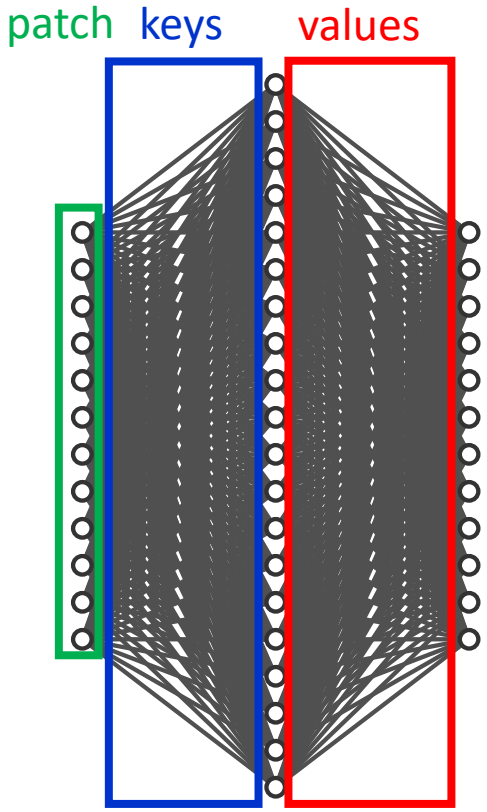
6. Stimulative Image Generation (Concept)

- Collect these scalars over all patches as a vector $a_j = [h_{1,j}, \ldots, h_{N,j}]$



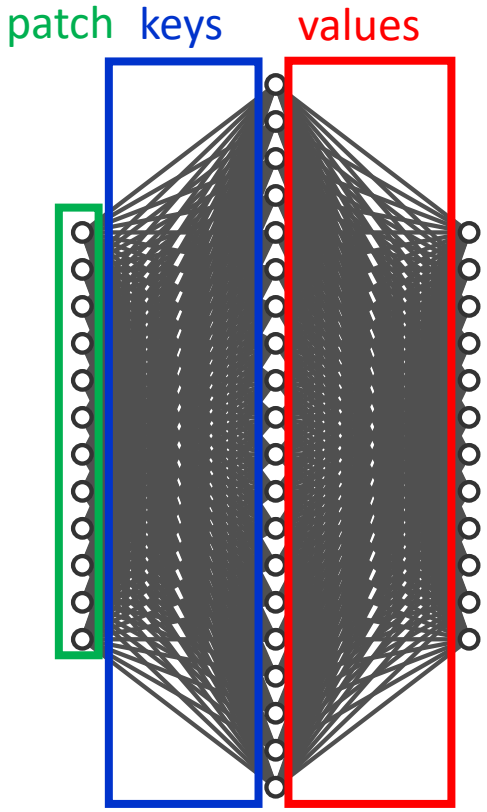
6. Stimulative Image Generation (Concept)

- Collect these scalars over all patches as a vector $a_j = [h_{1,j}, \ldots, h_{N,j}]$
- Let k be row of value vector most predicting a class c



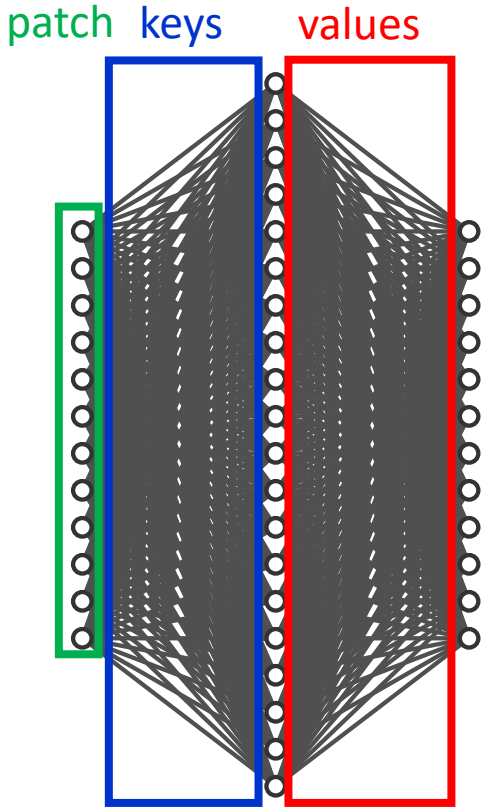
6. Stimulative Image Generation (Concept)

- Collect these scalars over all patches as a vector $a_j = [h_{1,j}, \ldots, h_{N,j}]$
- Let k be row of value vector most predicting a class c
- Optimize an image with respect to $-a_k$ mean $\mathcal{L} = -\frac{1}{h}\sum_{j=1}^{h} a_{k,j}$



6. Stimulative Image Generation (Concept)

- Collect these scalars over all patches as a vector $a_j = [h_{1,j}, \ldots, h_{N,j}]$
- Let k be row of value vector most predicting a class c
- Optimize an image with respect to $-a_k$ mean $\mathcal{L} = -\frac{1}{h}\sum_{j=1}^{h} a_{k,j}$
- During backpropagation propagate loss to input patch pixels and adjust them via gradient descend (leave model parameters static)



6. Stimulative Image Generation (Experiments)

6. Stimulative Image Generation (Experiments)

- Input optimization for different number of iterations

6. Stimulative Image Generation (Experiments)

- Input optimization for different number of iterations

250 Iterations

500 Iterations

750 Iterations

6. Stimulative Image Generation (Experiments)

- Input optimization for different number of iterations

250 Iterations

500 Iterations

750 Iterations

250 Iterations

500 Iterations

750 Iterations

6. Stimulative Image Generation (Experiments)

- Input optimization for different number of iterations

250 Iterations

500 Iterations

750 Iterations

250 Iterations

500 Iterations

750 Iterations

250 Iterations

500 Iterations

750 Iterations

6. Stimulative Image Generation (Experiments)

- Input optimization for different number of iterations
- Diversity objective to increase distance between image batch

250 Iterations

500 Iterations

750 Iterations

250 Iterations

500 Iterations

750 Iterations

250 Iterations

500 Iterations

750 Iterations

6. Stimulative Image Generation (Experiments)

- Input optimization for different number of iterations
- Diversity objective to increase distance between image batch

250 Iterations

500 Iterations

750 Iterations

750 Iterations Div

750 Iterations Div

750 Iterations Div

250 Iterations

500 Iterations

750 Iterations

250 Iterations

500 Iterations

750 Iterations

6. Stimulative Image Generation (Experiments)

- Input optimization for different number of iterations
- Diversity objective to increase distance between image batch

250 Iterations

500 Iterations

750 Iterations

750 Iterations Div

750 Iterations Div

750 Iterations Div

250 Iterations

500 Iterations

750 Iterations

750 Iterations Div

750 Iterations Div

750 Iterations Div

250 Iterations

500 Iterations

750 Iterations

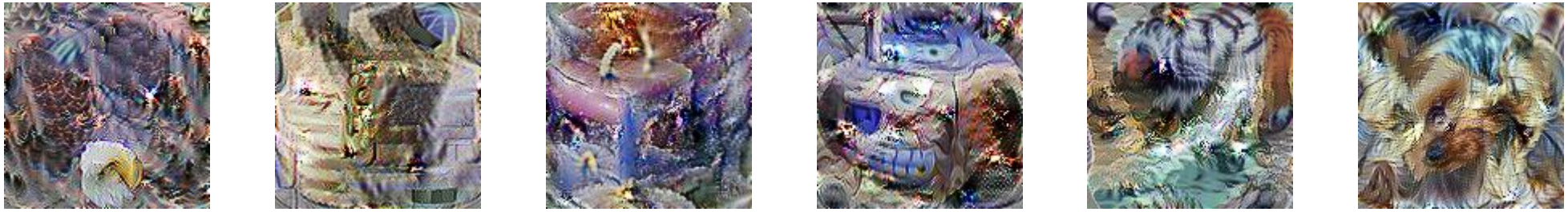
6. Stimulative Image Generation (Experiments)

- Input optimization for different number of iterations
- Diversity objective to increase distance between image batch

6. Stimulative Image Generation (Conclusion)

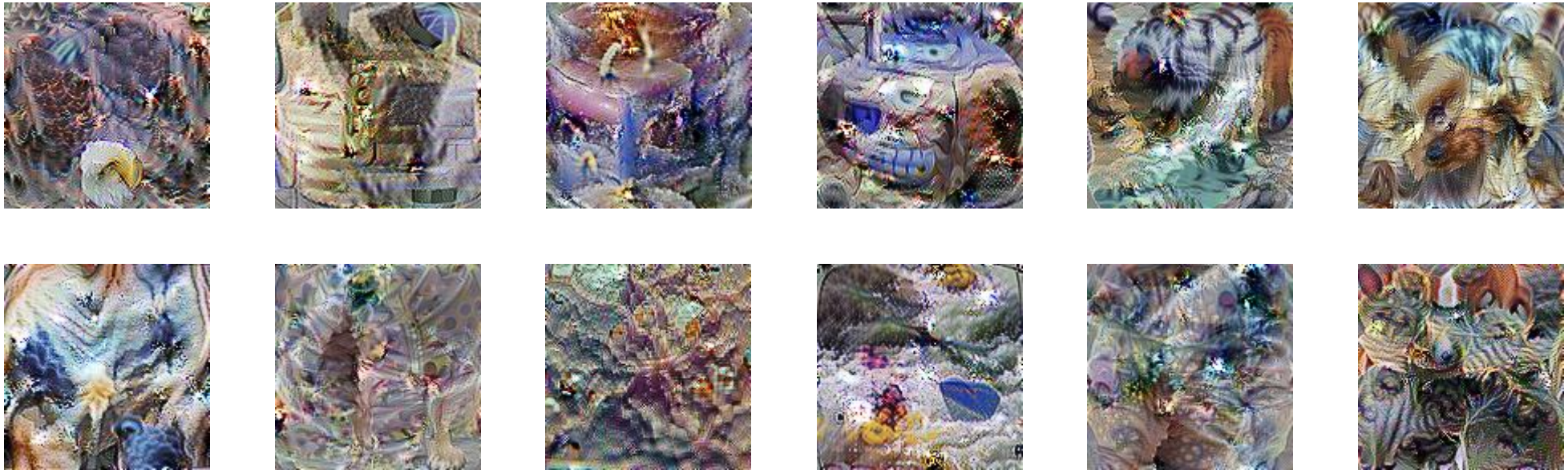
6. Stimulative Image Generation (Conclusion)

- For some classes, value neurons provide great interpretability
  - classes that are unique/distinct in the ImageNet-1k dataset
  - classes that contain very distinct patterns or shapes



6. Stimulative Image Generation (Conclusion)

- For some classes, value neurons provide great interpretability
  - classes that are unique/distinct in the ImageNet-1k dataset
  - classes that contain very distinct patterns or shapes
- Other classes have a shared most activated value vector
  - usually bad interpretability; imbalance in Imagenet-1k (120 dog breeds)



6. Stimulative Image Generation (Conclusion)

- For some classes, value neurons provide great interpretability
  - classes that are unique/distinct in the ImageNet-1k dataset
  - classes that contain very distinct patterns or shapes
- Other classes have a shared most activated value vector
  - usually bad interpretability; imbalance in Imagenet-1k (120 dog breeds)
- Future Work
  - Evaluate results for class based on contrastive learning and evaluation images

6. Stimulative Image Generation (Conclusion)

- For some classes, value neurons provide great interpretability
  - classes that are unique/distinct in the ImageNet-1k dataset
  - classes that contain very distinct patterns or shapes
- Other classes have a shared most activated value vector
  - usually bad interpretability; imbalance in Imagenet-1k (120 dog breeds)
- Future Work
  - Evaluate results for class based on contrastive learning and evaluation images
  - Try to generate images optimizing contrastive loss to evaluation images and most stimulated neurons

6. Stimulative Image Generation (Conclusion)

- For some classes, value neurons provide great interpretability
  - classes that are unique/distinct in the ImageNet-1k dataset
  - classes that contain very distinct patterns or shapes
- Other classes have a shared most activated value vector
  - usually bad interpretability; imbalance in Imagenet-1k (120 dog breeds)
- Future Work
  - Evaluate results for class based on contrastive learning and evaluation images
  - Try to generate images optimizing contrastive loss to evaluation images and most stimulated neurons
  - Evaluate on more datasets and models
    - Optimizing nonlinear value significantly degraded results

7. Conclusion

7. Conclusion

- Certain Value Neurons predict a class disproportionally well

7. Conclusion

- Certain Value Neurons predict a class disproportionally well
- The corresponding Key Neurons
  - respond highly to the subject
  - respond little to the background
  - are highly interpretable

7. Conclusion

- Certain Value Neurons predict a class disproportionally well
- The corresponding Key Neurons
  - respond highly to the subject
  - respond little to the background
  - are highly interpretable
- Later blocks, especially the block of the most predictive value neuron is responsible for the correct detection

7. Conclusion

- Certain Value Neurons predict a class disproportionally well
- The corresponding Key Neurons
  - respond highly to the subject
  - respond little to the background
  - are highly interpretable
- Later blocks, especially the block of the most predictive value neuron is responsible for the correct detection
- Earlier blocks are not very interpretable with respect to whole objects and have a noisy focus

8. Future Work

8. Future Work

- Focus where the context of an image is analyzed

8. Future Work

- Focus where the context of an image is analyzed
- Quantitative Analysis of generated images and heatmaps

8. Future Work

- Focus where the context of an image is analyzed
- Quantitative Analysis of generated images and heatmaps
- Understand Influence of Key Neurons on Subject / Background

8. Future Work

- Focus where the context of an image is analyzed
- Quantitative Analysis of generated images and heatmaps
- Understand Influence of Key Neurons on Subject / Background
- Investigate most heavily weighted value vectors for multiple classes

8. Future Work

- Focus where the context of an image is analyzed
- Quantitative Analysis of generated images and heatmaps
- Understand Influence of Key Neurons on Subject / Background
- Investigate most heavily weighted value vectors for multiple classes
- Investigate the predictiveness of the sum of two or more value vectors
  - Find composed classes (classes that are made up of multiple subclasses)

8. Future Work

- Focus where the context of an image is analyzed
- Quantitative Analysis of generated images and heatmaps
- Understand Influence of Key Neurons on Subject / Background
- Investigate most heavily weighted value vectors for multiple classes
- Investigate the predictiveness of the sum of two or more value vectors
  - Find composed classes (classes that are made up of multiple subclasses)
- Quantitative analysis of activated value vectors over similar classes

Sources

- Dosovitskiy, Alexey, et al. "An image is worth 16x16 words: Transformers for image recognition at scale." arXiv preprint arXiv:2010.11929 (2020)
- Analyzing Vision Transformers for Image Classification in Class Embedding Space, Vilas et al. (2023)
- https://alexlenail.me/NN-SVG/index.html
- https://github.com/evandez/neuron-descriptions
- ImageNet Dataset