Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

TP 2-b - Visualizing Neural Networks

Summary:

In this TP, we explore three different methods that enable us to visualize the information of an image that makes it belong to a certain class. First, saliency maps allows us to visualize the specific pixels of the image that contributed to the classification of an image in a certain class. The adversarial examples method then allowed us to visualize the details of the image that were determinative in its classification in a class by finding the minimal changes of the image that flipped its predicted class. And lastly, class visualization makes use of the score of a class to transform an input image (either noise or an actual image) into a representation of a specified class, which allows us to see the representative characteristics of a class across all images.

We also compared the performance of these techniques between a basic model and VGG16, and saw that the VGG16 depth allowed for sharper and more interpretable results.

Questions:

Section 1

1. We can see that the saliency maps highlight the pixels that have the strongest influence on the predicted class score since it is always the predicted object/animal that is seen in red. The pixels in red correspond to the most discriminative regions of the images, while background regions contribute much less, which indicates that the network relies on meaningful visual cues rather than uniformly using all pixels (Figure 1).

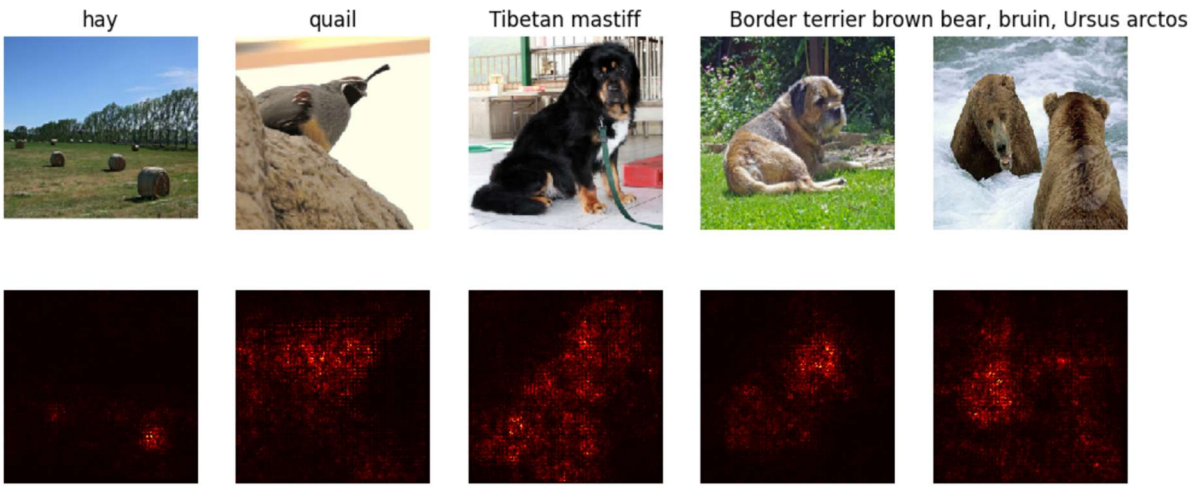

Figure 1 Saliency Maps for 5 ImageNet Images using SqueezeNet

Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

2. A limitation of saliency maps is that they are noisy and difficult to interpret precisely. They also only provide information on the importance of each pixel independently but it provides no information on the interaction of the pixels between each other.

3. Yes, it could be used to detect bias in the dataset. For instance, in the case of the research group that was discriminating dogs from wolves, saliency maps allowed them to see that all their wolf images had snow in the background, which made the model rely on the presence of snow to make the dog/wolf distinction, which is an issue originating from the dataset rather than the model itself.

4. The VGG16 results are way clearer than the previous SqueezeNet results. The objects/animals represented are more delineated than in the previous saliency maps (Figure 2). This can be seen very clearly in the hay image, where the background hays were very weakly represented in the SqueezeNet saliency maps, but are now in bright red in the VGG16 maps. The Border terrier contour is also clearly visible in the VGG16 map while the SqueezeNet one showed a more blurry blob of points where the dog is. This can be explained by the deeper architecture and higher parameter count, which allow VGG16 to learn richer hierarchical representations.

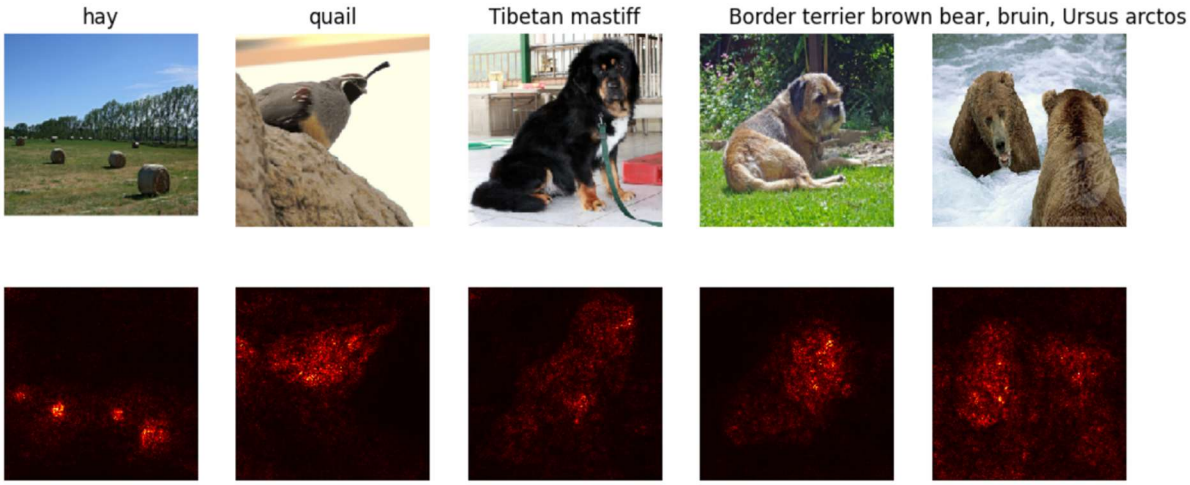

Figure 2 Saliency Maps for 5 ImageNet Images using VGG16

Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

Section 2

5. We can see in the magnified difference between the original image and its corresponding adversarial example that pixels that played a role in switching the predicted class from a dog to a bird are concentrated within the dog pixels (Figure 3). Which shows that the features of the image that were decisive in the 'Tibetan mastiff' classification were specific characteristics of the dog.

Figure 3 Adversarial Examples Results using SqueezeNet on a Tibetan massif-class image to bulbul-class

6. Since the adversarial images appear visually identical to the original images for humans, yet the network confidently misclassifies them, this technique demonstrates that CNNs rely on fragile decision boundaries and can be highly sensitive to small, carefully chosen perturbations. This means that CNNs are very easy to fool, which raises concerns with regards to their real-life applications.

7. This naive approach does not explicitly constrain the perturbation magnitude, which can lead to visually unrealistic adversarial images after many iterations. It is also highly sensitive to hyperparameters such as the learning rate and number of iterations, making it unstable. More robust alternatives include the Fast Gradient Sign Method (FGSM) and Projected Gradient Descent (PGD), which enforce explicit bounds on the perturbation to ensure imperceptible modifications. Additionally, adversarial training improves model robustness by incorporating adversarial examples during training, leading to smoother and more reliable decision boundaries.

Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

Section 3

8. Snail class visualization starting from random noise image

We start from a random noise image and optimize it to maximize the score of the class "snail", and observe that the initially unstructured noise progressively evolves into a pattern containing snail-like low-level features.

After 200 iterations, the image does not resemble a natural photograph of a snail, but instead displays repeated textures, spirals, rounded bodies, and elongated protrusions, which shows that these are the characteristics that the network associates with the snail class (Figure 4). This shows that the network doesn't encode the predicted classes directly, but rather a collection of discriminative visual patterns.

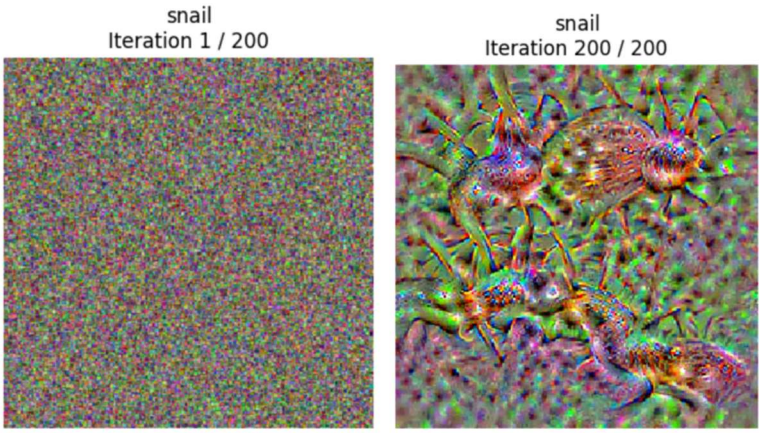

Figure 4 Snail class visualization starting from random noise image using SqueezeNet

9. Testing other hyperparameters for snail class visualization starting from random noise image

Learning rate variation:

The learning rates we used are the following [1, 3, 5, 7, 9], number of iterations is fixed at 200, and L2 regularization term at 0.001. All learning rates result in similar patterns across the class visualization, but lower learning rates show more faded patterns with less intense activations, while higher learning rates show intense activations and patterns that are better delineated (Figure 5). This shows that the learning rate controls how aggressively the optimization exploits the gradient of the target class score.

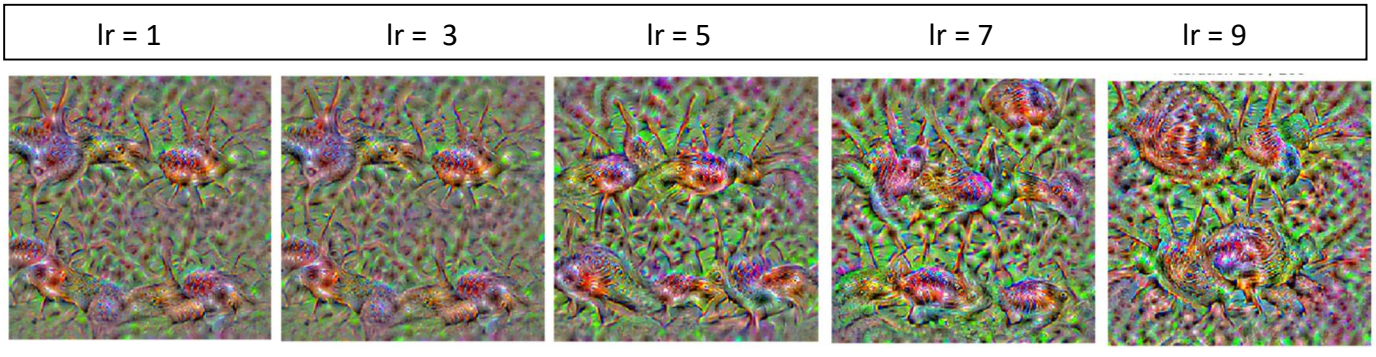

Figure 5 Snail class visualization starting from random noise image using SqueezeNet for 5 different measures of learning rate

Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

Iterations variation:

The number of iterations we used are the following: [100, 200, 300, 400, 500], learning rate is fixed at 5, and L2 regularization term at 0.001. As the number of iterations increases, snail-related features become increasingly pronounced. With few iterations, only vague textures and weak shapes appear. So, with increasing iterations, the patterns become more structured and intense (Figure 6). This illustrates that more optimization pushes the image further into stronger representations of the class, even if these regions are far from the original image.

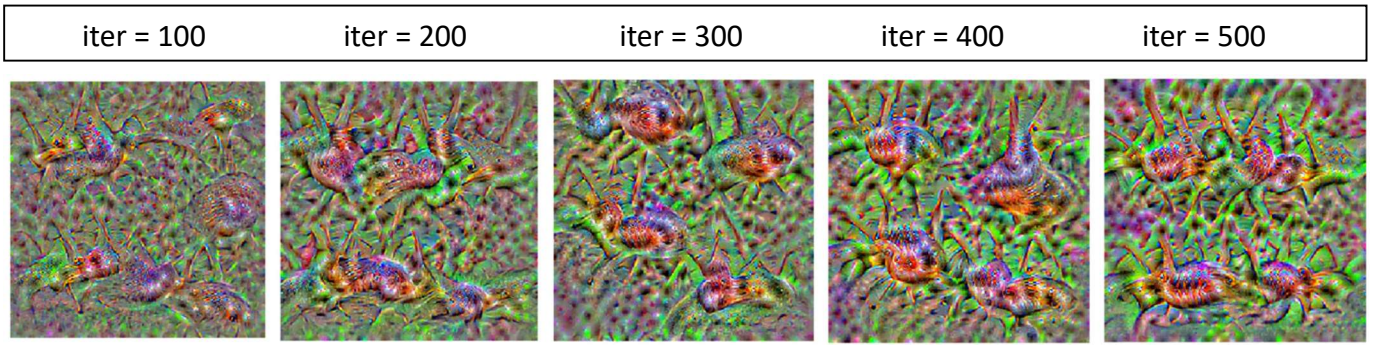

Figure 6 Snail class visualization starting from random noise image using SqueezeNet for 5 different numbers of iterations

L2 regularization variation:

The L2 regularization values we used are the following: [0.0001, 0.001, 0.01, 0.1, 1], learning rate is fixed at 5, and number of iterations at 200. This term plays a crucial role in controlling the original image magnitude. For low values, we can see the class patterns taking over the image, while for higher values, the normalization regulates these activations and we get patterns that are less intense. (Figure 7)

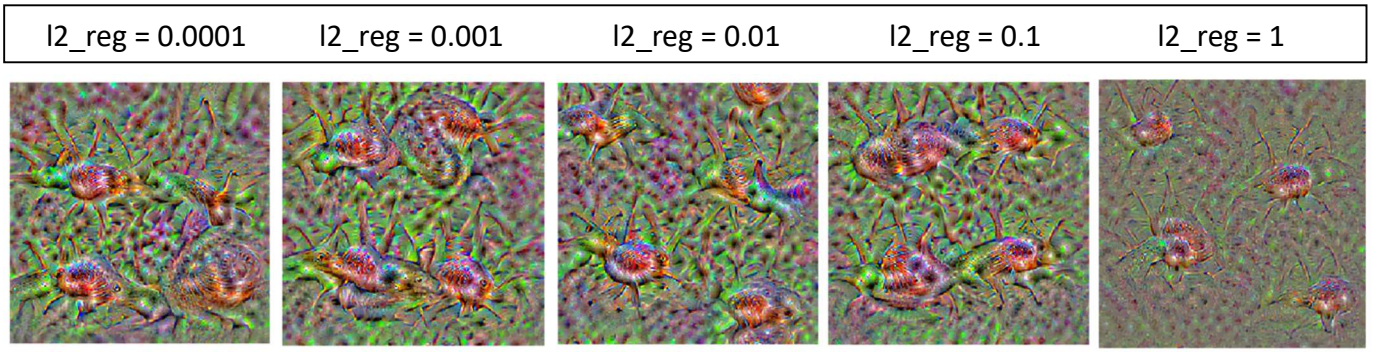

Figure 7 Snail class visualization starting from random noise image using SqueezeNet for 5 different values of L2 regularization

Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

10. Snail class visualization starting from image of hay

When initializing the optimization from a real ImageNet image of a hay field instead of random noise, and using the class snail as the target, the resulting visualization is now grounded in the structure of the original image. Compared to the noise initialization, the optimized image preserves some global structure of the original scene but certain patterns get converted into snail-like patterns (Figure 8). So snail-like features appear superimposed onto meaningful regions of the image rather than uniformly everywhere.

This demonstrates the interest of using a real image as initialization: the optimization is constrained by existing low-level and mid-level features, leading to more interpretable and less chaotic visualizations.

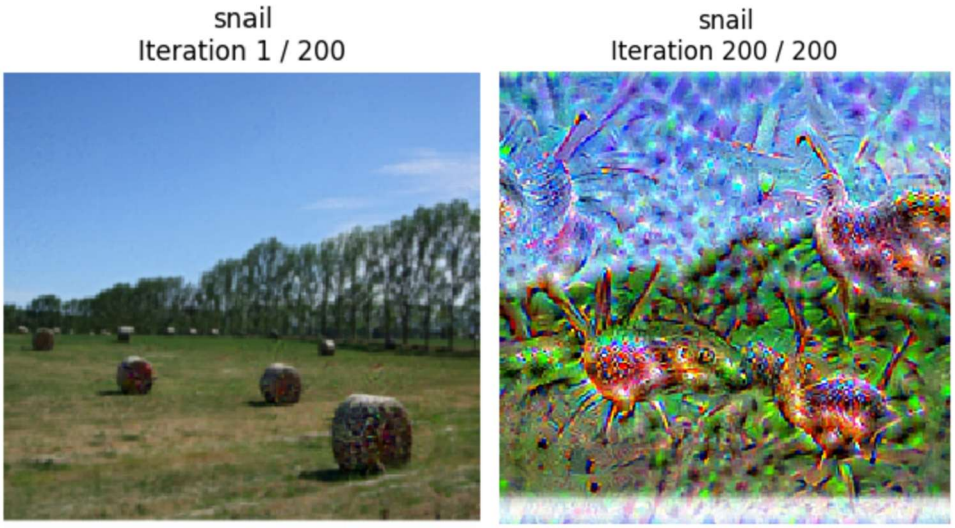

Figure 8 Snail class visualization starting from image of hay using SqueezeNet

11. Class visualization using VGG16

When repeating the experiment using VGG16, the generated images are noticeably more structured and interpretable than with the previous network.
Starting from noise, the VGG16 visualization exhibits clearer spiral shapes, smoother contours, and more coherent color transitions. The patterns also appear less random and more consistently "snail-like." (Figure 9)
Starting from the ImageNet hay image, the class-specific features are more localized and integrated into the scene. The resulting image shows stronger semantic coherence, with snail textures aligning better with object-like regions. (Figure 10)
Similarly to Section 1, this means that VGG16's deeper architecture and higher representational capacity allow it to learn richer and more hierarchical visual features, which translates into more interpretable class visualizations.

Marilyn CHAHINE, 21517286
Jean MACQUET, 3532408

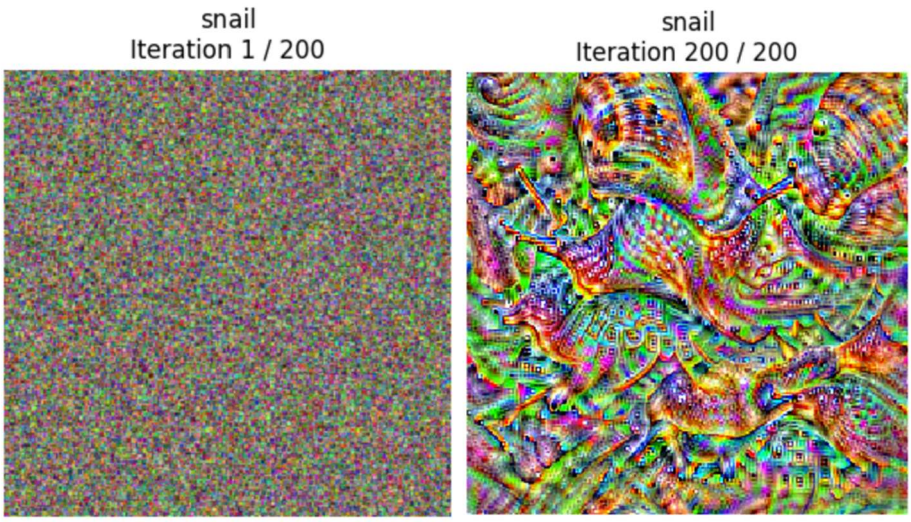

Figure 9 Snail class visualization from random noise image using VGG16

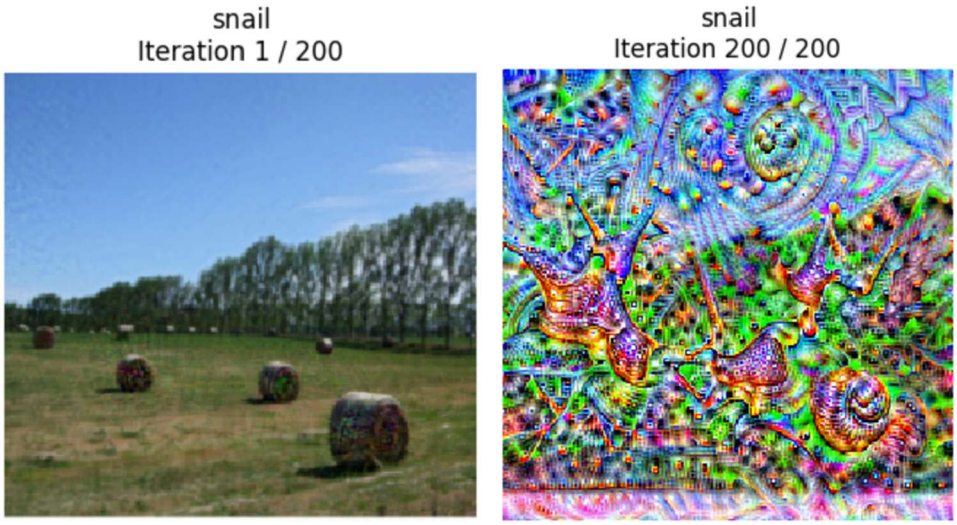

Figure 10 Snail class visualization from hay image using VGG16