Assignment 3 Writeup

DO NOT TAG

Name: Jing Yu
GT Email: jyu497@gatech.edu

Visualization

DO NOT TAG

Implementation Question 1

DO NOT TAG

In your coding homework, you were given the following hint:
“There are two approaches to performing backprop using the PyTorch command tensor.backward()… Alternatively, one can take the sum of all the elements of the tensor and do a single backprop with the resulting scalar. This second approach is simpler and preferable as it lends itself vectorization.”

Question: Referring to the coding task completed by you, why is the suggested alternative approach mathematically sound? Please provide a brief but succinct answer on the next slide.

Answer for Implementation Question 1

Answer:

If the tensor is created with requires_grad=True, torch.autograd records operations on it for automatic differentiation. When you sum all the elements of the tensor and call .backward() with the resulting scalar, all the gradients will be computed automatically and accumulated into .grad attribute. By vectorization, it’s more computational efficient.

Implementation Question 2

DO NOT TAG

In your network visualization tasks, you need to compute gradients for which one of the following three quantities:

A. Cross entropy loss
B. Unnormalized score corresponding to the correct class
C. Class probabilities

Please answer on the next slide.

Now briefly justify why the other two options are not optimal.

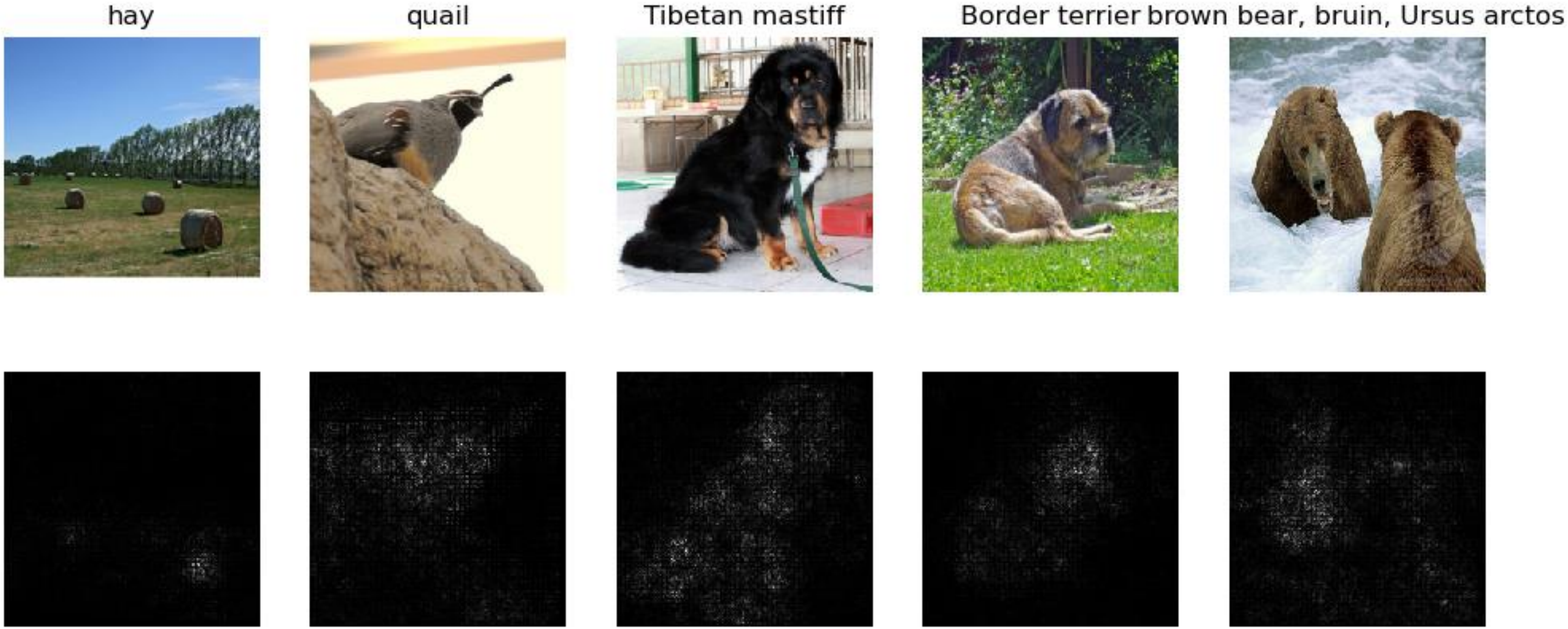
Answer for Implementation Question 2

Answer (A, B or C):

B

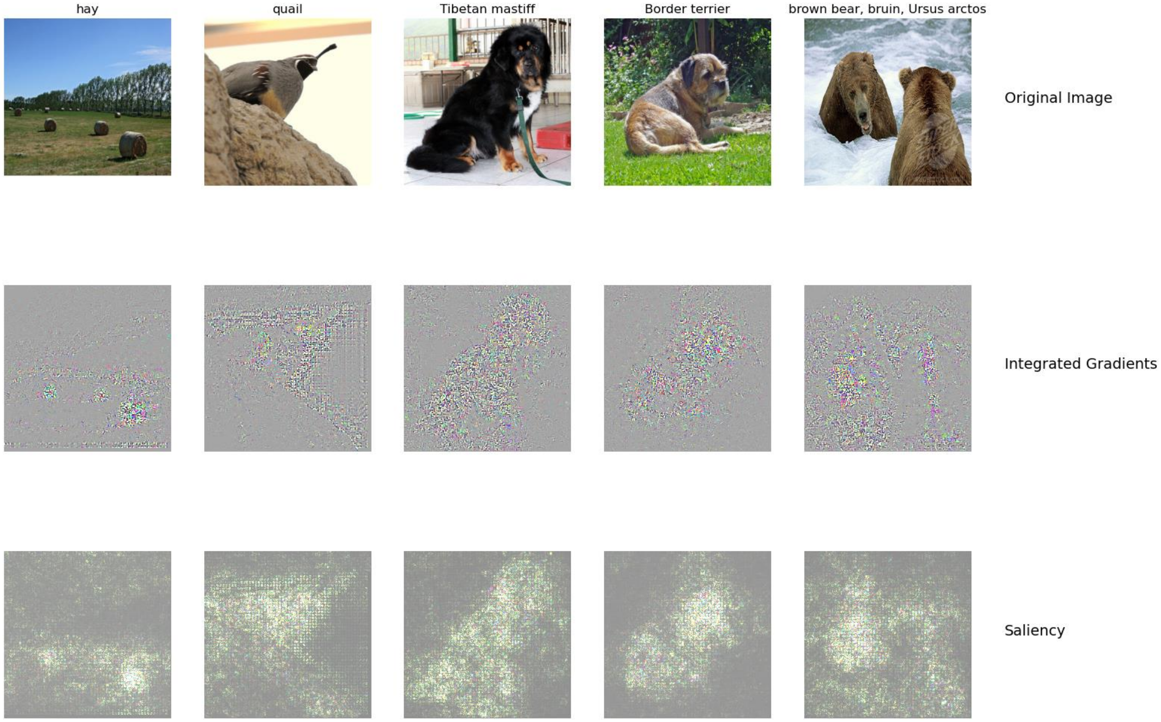
We care about the degree to which each pixel in the image affects the classification score for that image. The maximization of the class probabilities can be achieved by minimizing the scores of other classes. We should ensure that the optimization concentrates only on the specific class. Therefore the CE loss and class probabilities are not optimal.

Saliency Map



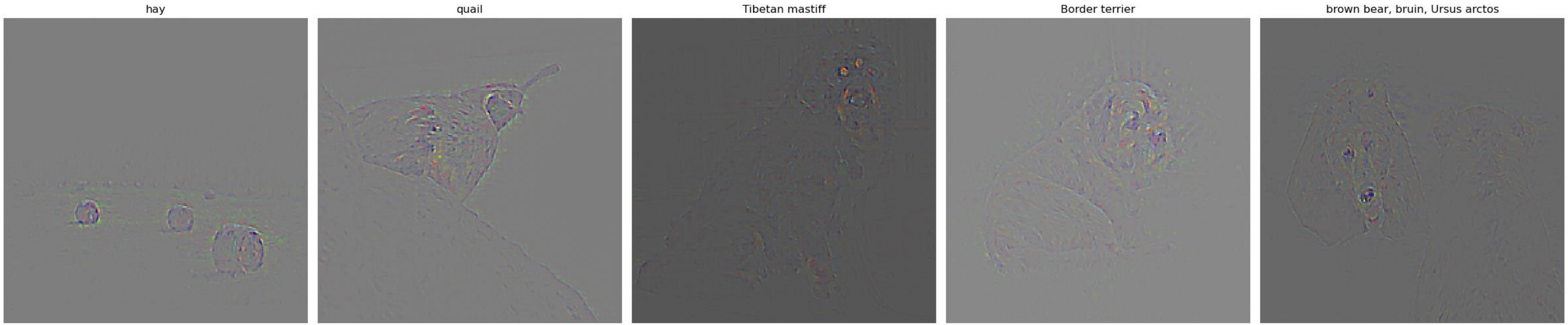
Saliency Map

- Saliency map from Captum



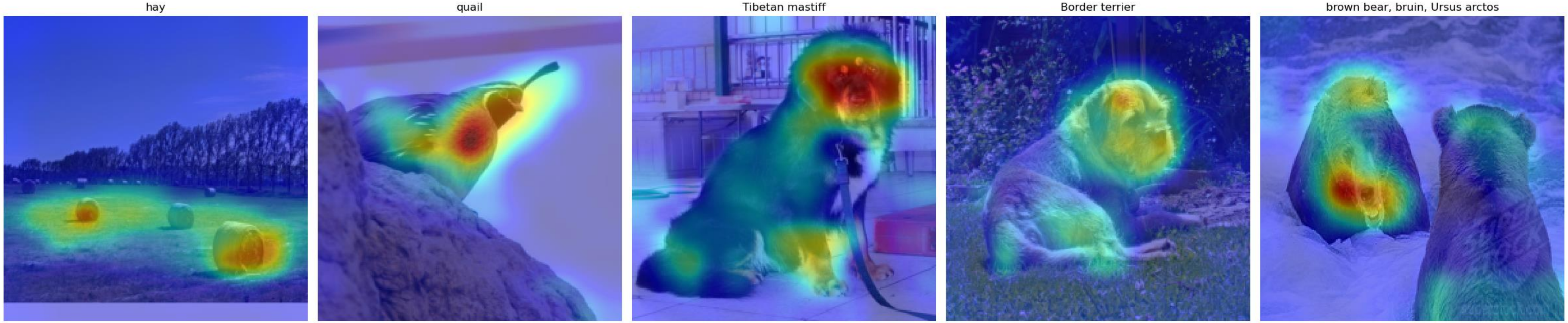
GradCam

- Visualization of Guided Backprop



GradCam

- Visualization of GradCam



hay

quail

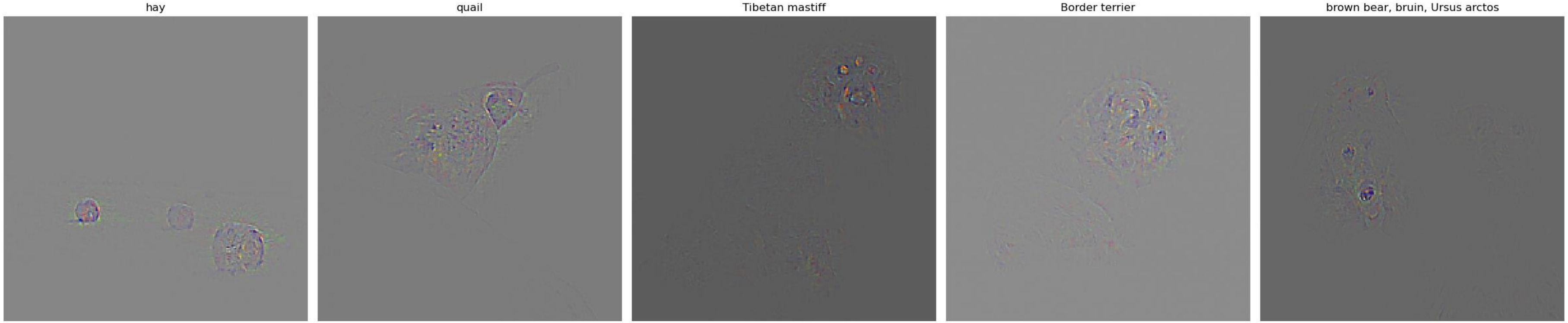
Tibetan mastiff

Border terrier

brown bear, bruin, Ursus arctos

GradCam

- Visualization of Guided GradCam



hay

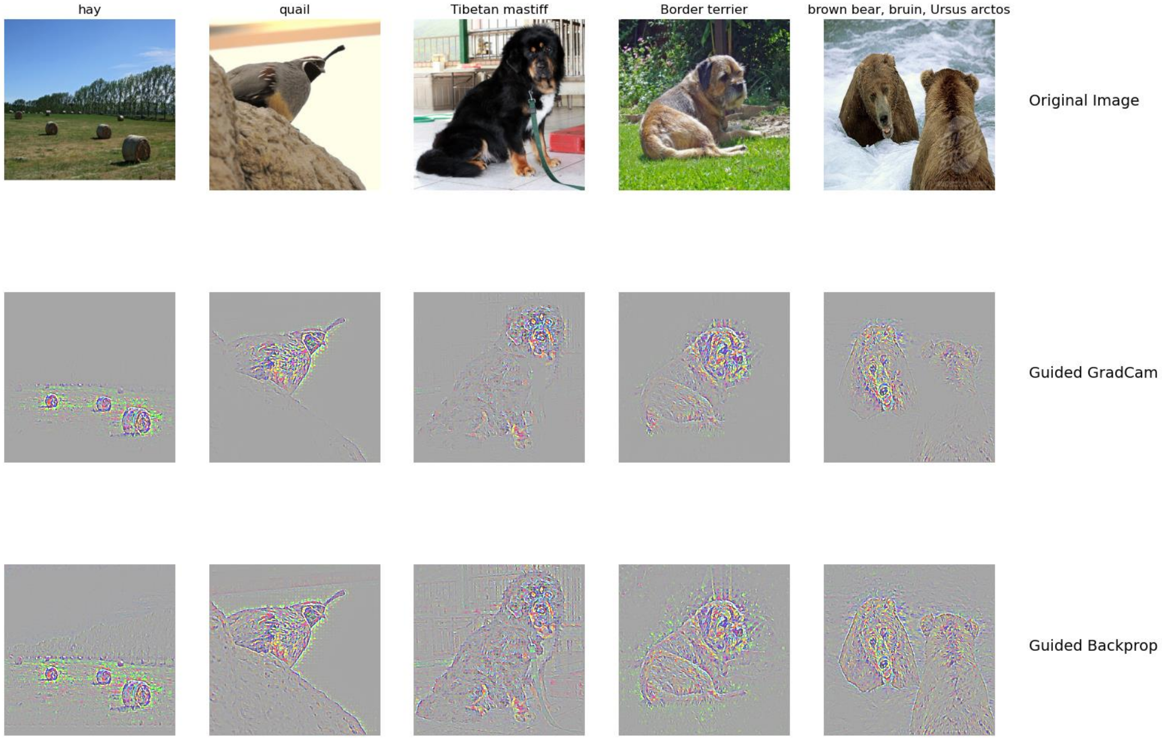
quail

Tibetan mastiff

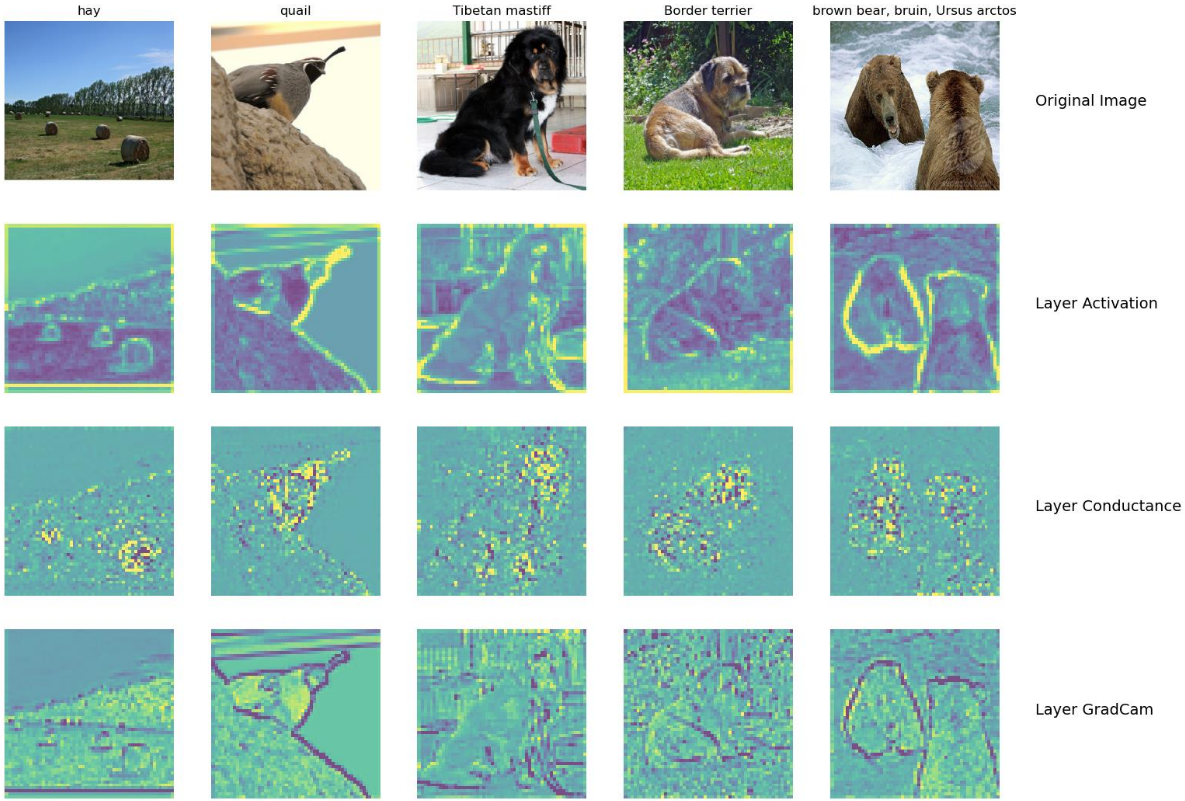
Border terrier

brown bear, bruin, Ursus arctos

GradCam


hay
quail
Tibetan mastiff
Border terrier
brown bear, bruin, Ursus arctos
Original Image
Guided GradCam
Guided Backprop

GradCam


hay
quail
Tibetan mastiff
Border terrier
brown bear, bruin, Ursus arctos
Original Image
Layer Activation
Layer Conductance
Layer GradCam

What do saliency map and Gradcam tell you? How are they different? Is one better than the other?

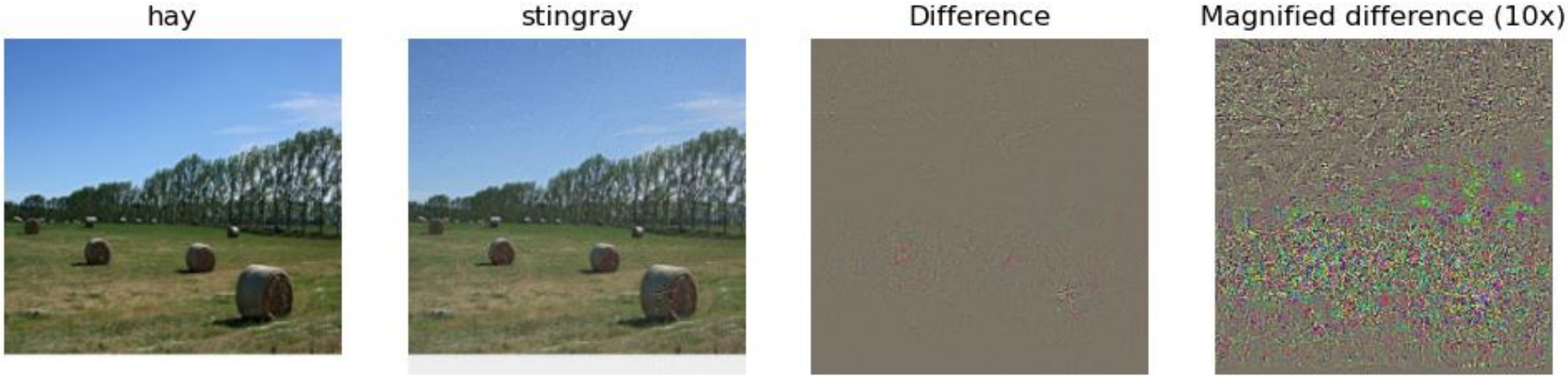
Answer:

Saliency map tells us the degree to which each pixel in the image affects the classification score for that image.

Gradcam uses the gradients of any target concept flowing into the final convolutional layer to produce a coarse localization map highlighting the important regions in the image for predicting the concept.

Which approach is better depends on your purpose.

Fooling Image



Fooling Image

What insights do you get from fooling images:

Answer:

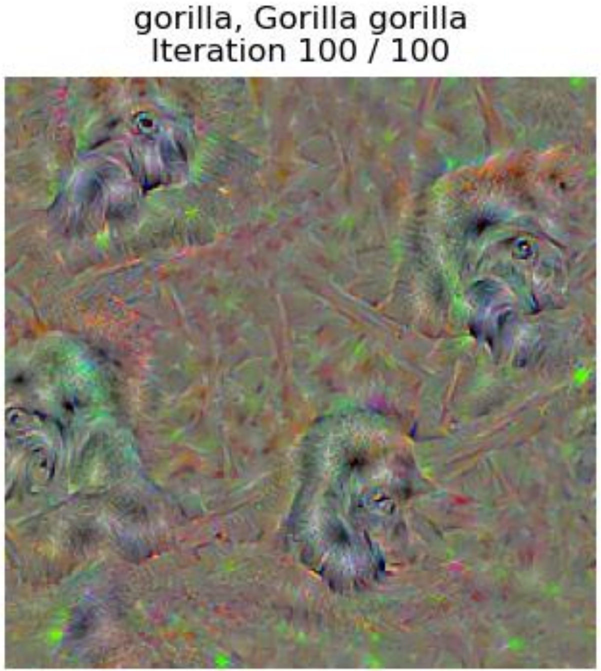
We can perform gradient ascent over the image to maximize the target (incorrect) class.

Small pixel changes can lead to wrong classification.

By adding some adversarial noise, can easily make the network fail => adversarial attacks

Class Visualization

- Class visualization of Gorilla



Question: Class Visualization – Use saliency?

In order to find an image that maximizes the correct score, Jane performs gradient ascent on the input image, but instead of the gradient she uses the saliency map in each step to update the image. List and briefly explain two reasons why this is an incorrect approach. (Hint: refer to Section 1.1 of the assignment pdf)

Answer:

To compute the saliency map

1. Call .backward() on the total loss other than score of the target class.
2. Take the absolute value of this gradient, then take the maximum value over the 3 input channels.

Question: Class Visualization – Regularization

DO NOT TAG

When generating an image that the network will recognize as the target class, the quality of the generated image is improved by regularization. In your work, you applied L2-regularization and blurring for this purpose. What is the effect of these on the optimization process (that is, what is it that these techniques are discouraging)?

Please answer on the next slide.

Answer for Class Visualization – Regularization

Answer

L2 regularization:

Penalizes large values and tends to prevent a small number of extreme pixel values from dominating the example image. When applied too strongly, causes the optimization to fail or the images to be less interpretable

Blurring:

Producing images via gradient ascent tends to produce examples with high frequency information, causing high activations. Blurring penalizes high frequency information. While convolving with a blur kernel is more computationally expensive than the other regularization methods.

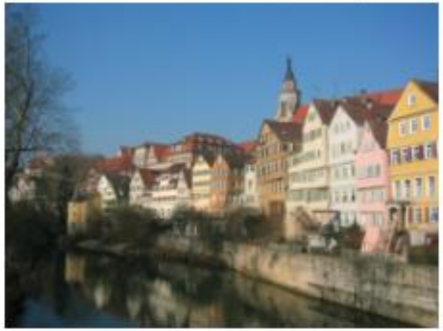
Style Transfer

DO NOT TAG

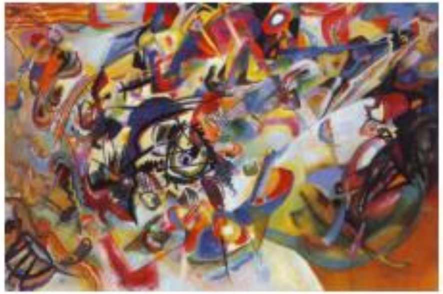
Composition VII + Tubingen

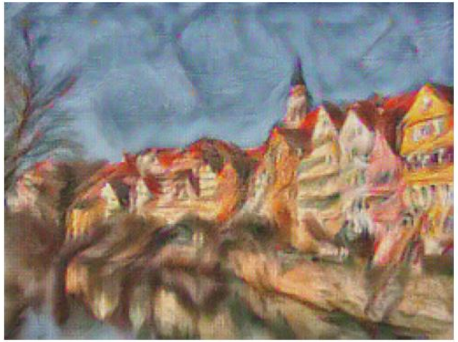
- Include both original images and the transferred image

Content Source Img.



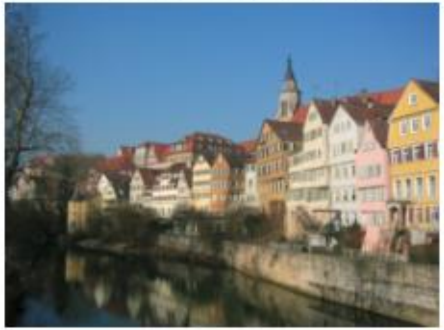
Style Source Img.



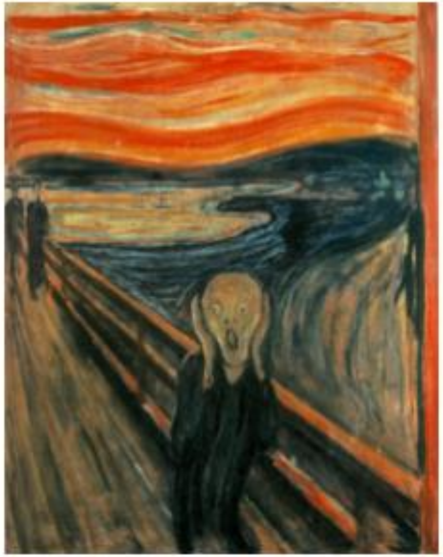


Scream + Tubingen

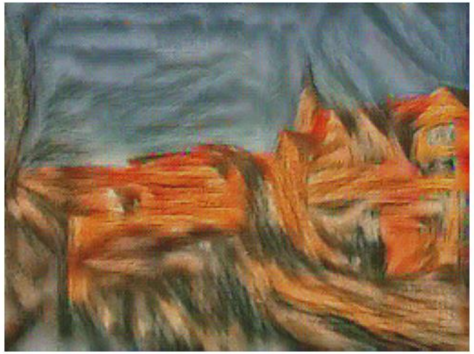
- Include both original images and the transferred image



Content Source Img.

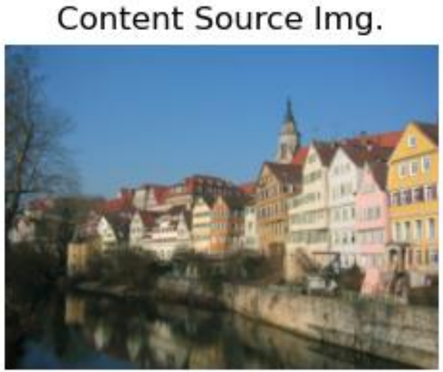


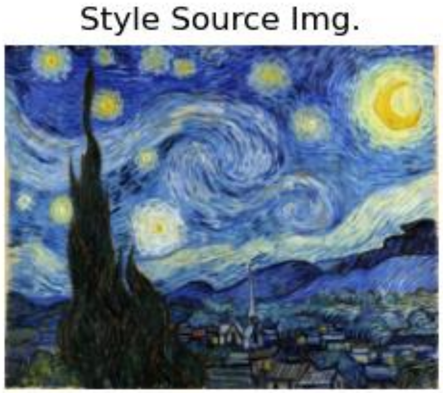
Style Source Img.

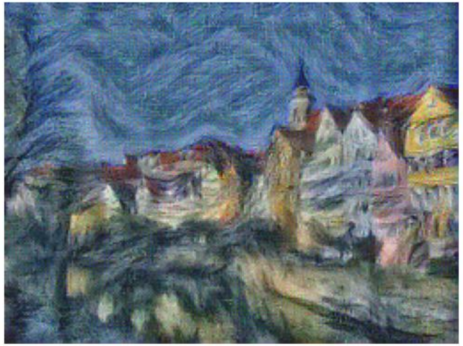


Starry Night + Tubingen

- Include both original images and the transferred image

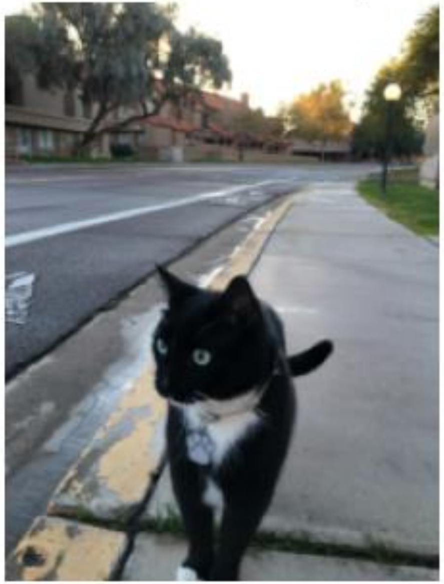






Style Transfer – Unleash Your Creativity

Content Source Img.



Style Source Img.

Style Transfer – Unleash Your Creativity