Assignment 3 Writeup

DO NOT TAG

Name:
GT Email:

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:

Because the gradients of a sum is the sum of the separate gradients. Therefore, taking the sum of all the elements of the tensor and do a single backpropagating with the resulting scalar is equivalent to backpropagating through each element separately and then summing their gradients.

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

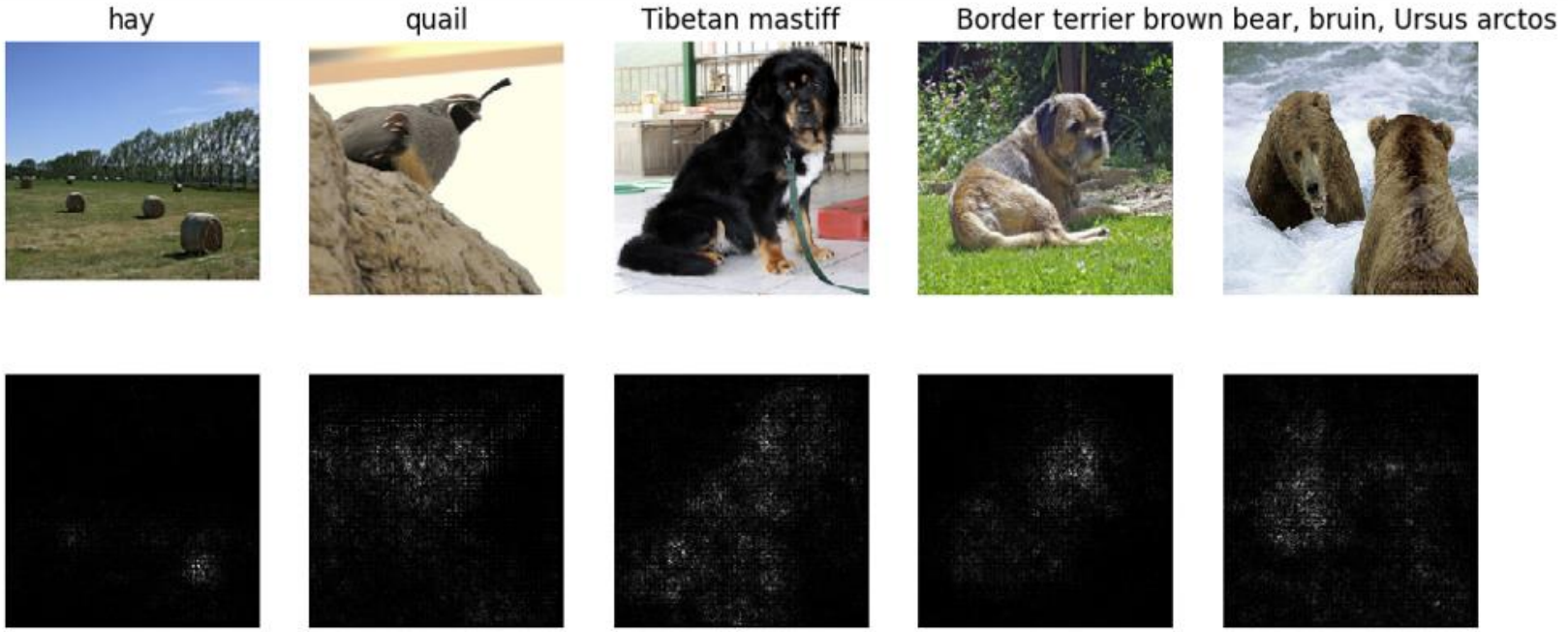
Answer (A, B or C): B

Now briefly justify why the other two options are not optimal for tasks on hand.

For option A: Because cross-entropy loss quantifies the difference between the predicted probability distribution and the true probability distribution. It couldn’t reflect how the input changes will impact the raw output.
For option C: Class probabilities are obtained by passing in raw scores to SoftMax operation, which introduces non-linearity and normalization. However, when performing the visualization tasks, we would like to understand how input changes impact the raw model output directly. Gradients w.r.t probabilities may not provide a clear view of the direct impact of input changes on the raw model output.

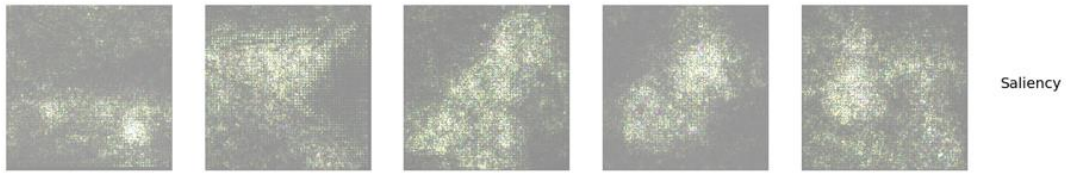
Saliency Map

- Include your saliency map here



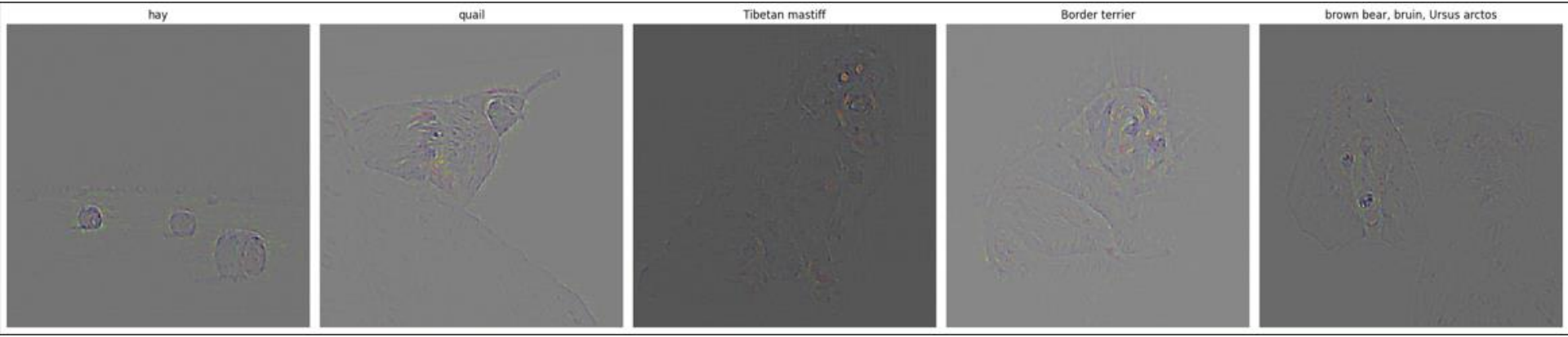
Saliency Map

- Include your saliency map from Captum here



GradCam

- Include your visualization of Guided Backprop here

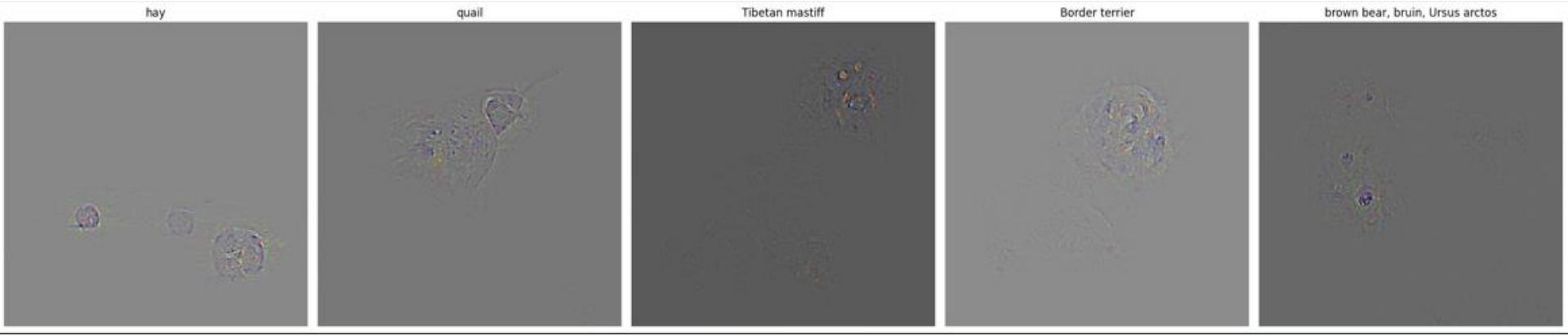


GradCam

- Include your visualization of GradCam here

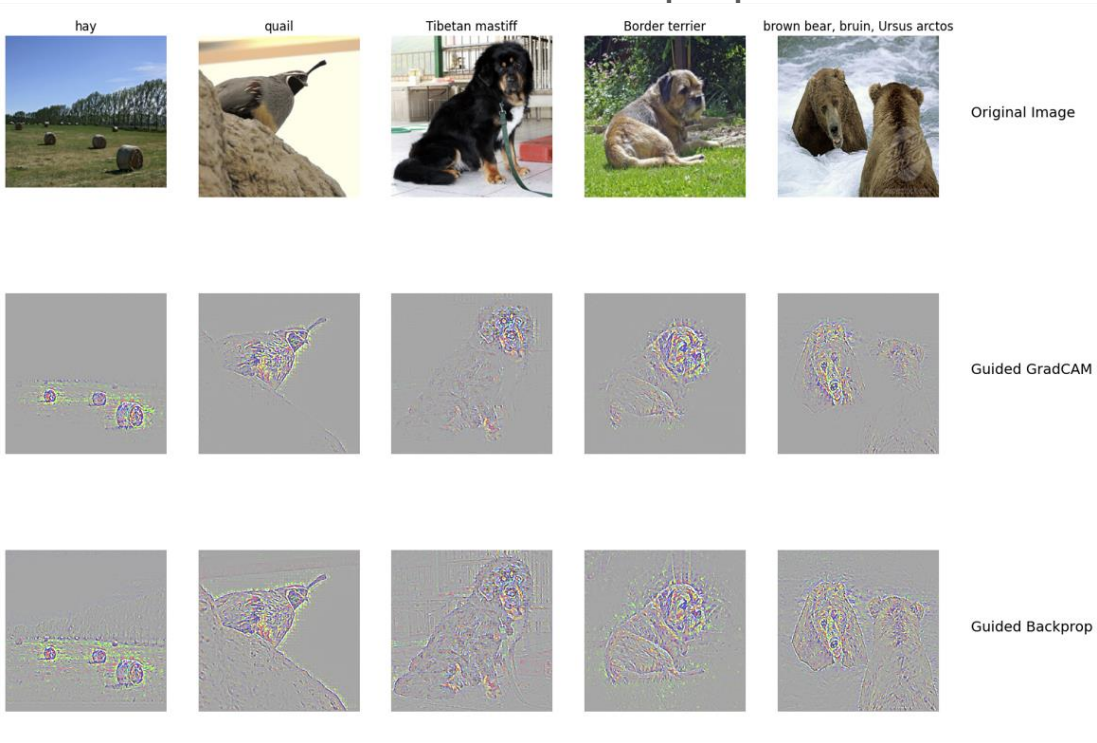
GradCam

- Include your visualization of Guided GradCam here



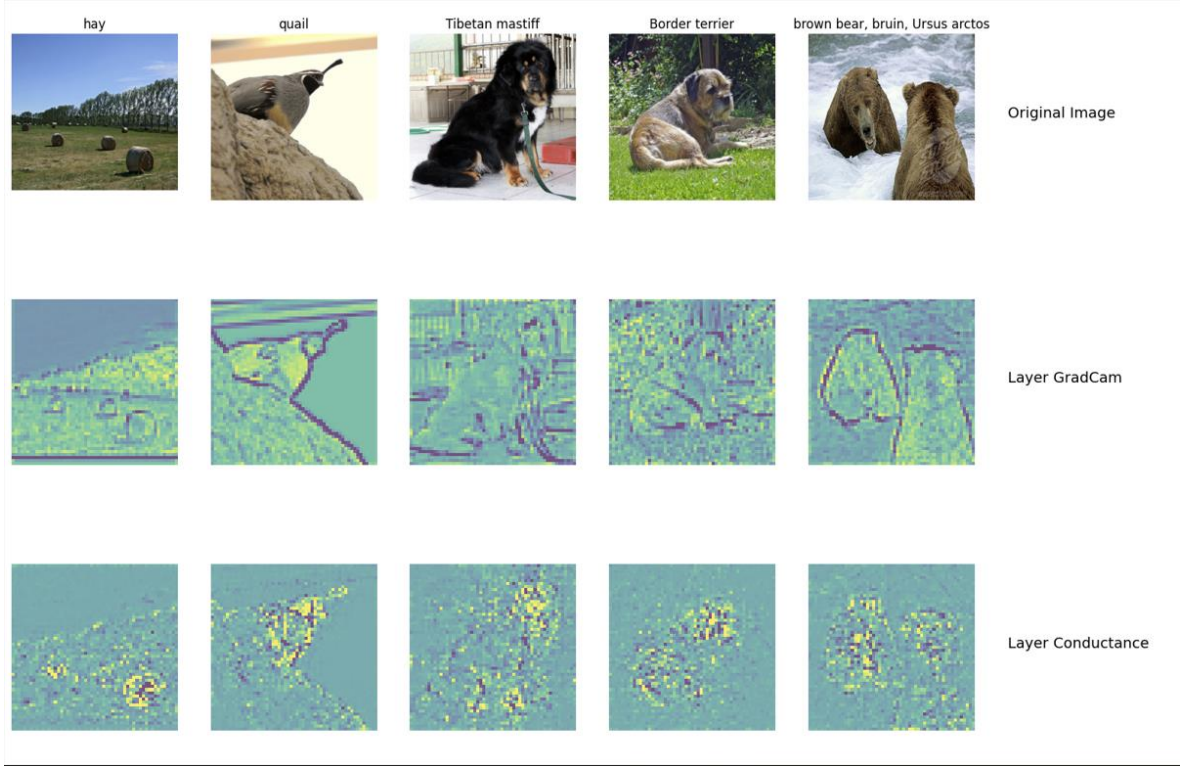
GradCam

- Include your visualization of Guided Backprop and Guided Gradcam from Captum h



GradCam

- Visualization of layers and neurons using Captum here:



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

Answer:

The saliency map highlists the regions in an input image that contribute to a model's predictions. Grad-cam is an extension of the saliency map, it is class-specific and localizes features that are important for a particular class. Grad-cam integrates information from both gradients and the class activation maps.

When we're interested in the feature importance across all classes, the saliency map is more suitable. However, when we're interested in the important regions for a specific class, the Grad-cam is better.

Fooling Image

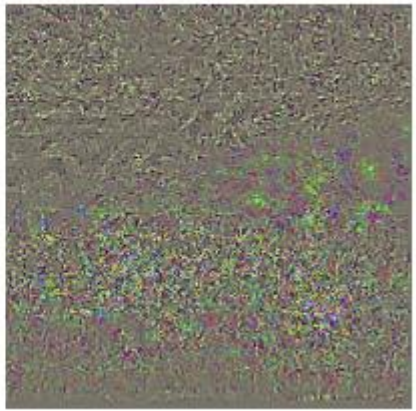
Include the fooling image here:

hay

stingray

Difference

Magnified difference (10x)



Fooling Image

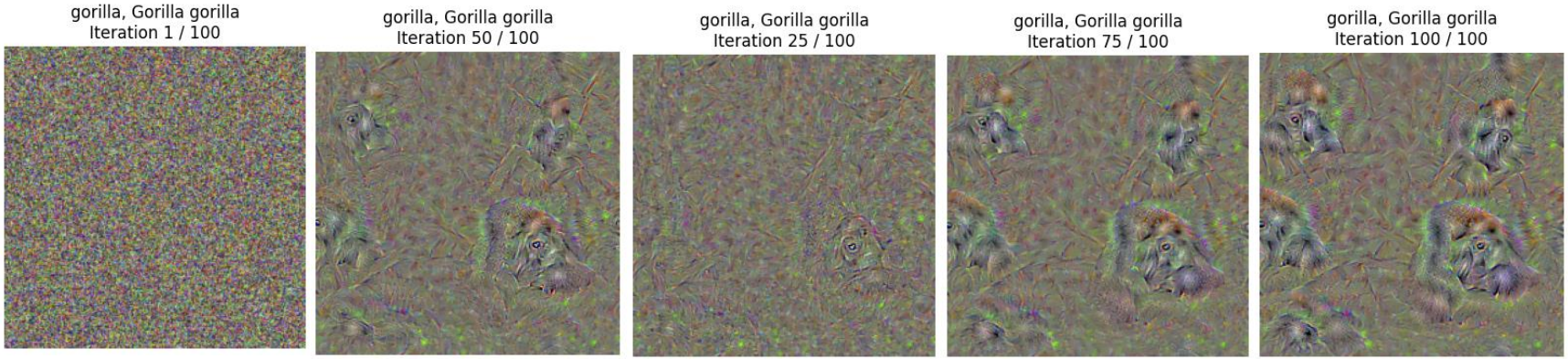
What insights do you get from fooling images:

Answer:

Fooling image can be generated by implementing small, carefully crafted perturbations to the input image. It demonstrates that models are sentitive to specific features in the input. There is no obvious pattern in the difference between the fooling image and original image. So, it reveals that the model may rely on subtle and non-intuitive features to make predictions. In addition, we can improve geenralization strategies and indentify the corresponding ares to enhance the robustness of the model.

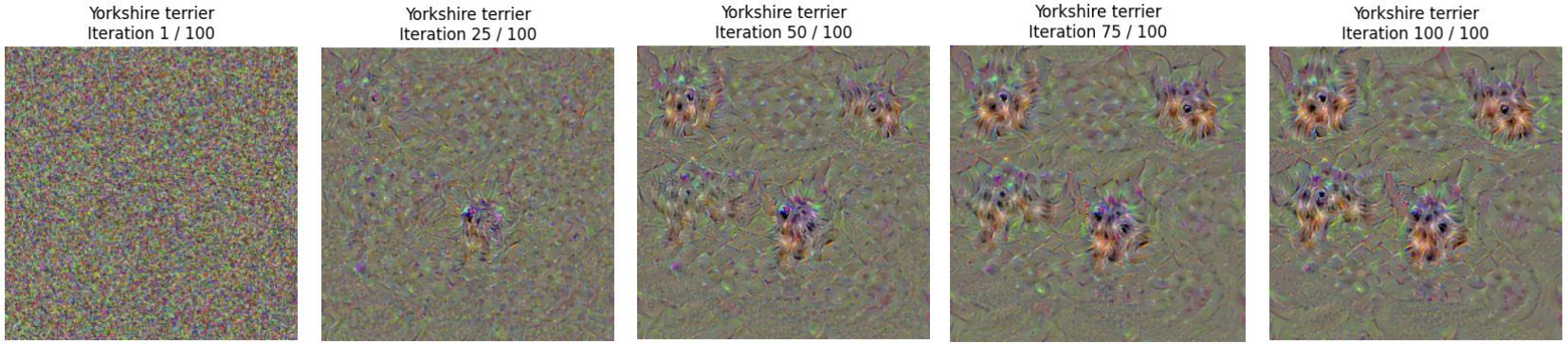
Class Visualization

Include class visualization of Gorilla (target_y = 366) here:



Class Visualization

Include class visualization of Yorkshire Terrier (target_y = 187) here:



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:

1. A saliency map only contains absolute values. However, the sign of the gradient is crucial for gradient ascent. The sign determines whether the pixel values should be increase or decrease to maximize the correct score.

2. The saliency map is the absolute values of the gradient of the unnormalized score w.r.t the pixel values without being scaled correctly. This may lead to inefficient optimization or divergence.

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, which discourages the optimization process from creating high unrealistic pixels values and generating unrealistic and erratic visual patterns, thus promoting smoother distributed pixel values. Blurring applies a filter to the generated image, which reduces high-frequency noise and sharp transitions between pixels. The smoothing effect makes the generated image more visually coherent.

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

Content Source Img.

Style Source Img.

Style Transfer – Unleash Your Creativity

Include your two original images (content and style images)

Content Source Img.

Style Source Img.

Style Transfer – Unleash Your Creativity

Include your final stylized image

Paper Review

The paper “Axiomatic Attribution for Deep Networks” introduced a method called integrated gradients that attributes the prediction of a deep network to its inputs. It satisfied key axioms of Sensitivity and Implementation invariance, ensuring that the attributions are both sensitive to the input features and invariant under different implementations of the same function. Unlike other attribution methods, Integrated Gradients needs no instrumentation of the network. It can be implemented using a few calls to the gradients operator and be applied to a variety of networks. However, this paper has not addressed the interactions between the input features or the logic employed by the network. My observation is that this paper found a new approach to clarify desirable features of an attribution method using axiomatic framework, making the methodology theoretically sound. In addition, Integrated Gradients not only aids in interpreting model predictions but also offers insights into the model's functioning. This represents a notable advancement in making complex models more transparent and easier to interpret and is a significant contribution to explainable AI.