EXPLAINABILITY AI(XAI)

Ram Mannuru

Course: Deep Learning

1. Introduction:

In this project, we explore various explainability techniques for deep learning models. Deep neural networks (DNNs) have achieved remarkable success in various tasks such as image classification, object detection, and natural language processing. However, their complex internal mechanisms often make it challenging to interpret their decisions. Explainability techniques aim to provide insights into how DNNs make predictions, thus enhancing trust and understanding of these models.

2. Description of the Dataset:

As this project focuses on explainability techniques rather than training models, we do not utilize a specific dataset. Instead, we leverage pre-trained models and sample images to demonstrate the effectiveness of explainability techniques.

3. Description of the Deep Learning Network and Training Algorithm:

Since we do not train a model from scratch, we utilize pre-trained convolutional neural networks (CNNs) such as ResNet-50. These models have been trained on large-scale datasets like ImageNet and demonstrate strong performance in various visual recognition tasks. We focus on the architecture and internal mechanisms of these models to apply explainability techniques.

4. Explainability Techniques:

4.1 Class Activation Maps (CAMs):

Description: Class Activation Maps (CAMs) are a type of visualization technique used to understand the decision-making process of convolutional neural networks (CNNs). They provide insights into which regions of an input image contribute most to the model's prediction. CAMs highlight the discriminative regions of an image by generating heatmap overlays, indicating where the model focuses its attention during classification.

This technique is particularly useful for understanding which parts of an image are most relevant for a particular class prediction. Workflow:

- **Feature Extraction**: CAMs utilize the activations of the last convolutional layer in the CNN, which capture high-level features of the input image.
- **Global Average Pooling (GAP)**: The feature maps are subjected to global average pooling to obtain a summary of each feature map.
- **Weighted Combination**: The class score is computed by taking a weighted combination of the feature map activations, where the weights are derived from the gradients of the predicted class score with respect to the feature maps.
- **Heatmap Generation**: The weighted combination results in a class activation map, which is then upsampled to the original image size to produce a heatmap overlay.

Mathematical Explanation:

- **Feature Extraction**: Let A^l denote the activation map of the last convolutional layer.
- **Global Average Pooling (GAP)**: The GAP operation computes the class activation map M by taking the weighted average of the activation map A^l for each feature map k, where the weights w_k are derived from the gradients of the predicted class score y_c with respect to the feature maps:

$$M = \sum_k w_k A_k^l$$

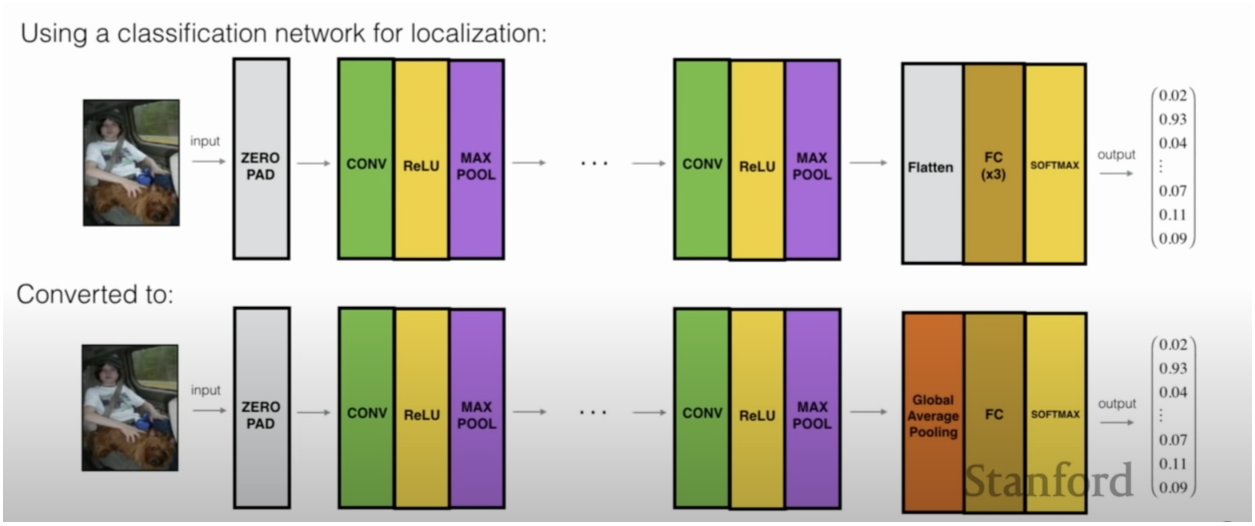
- **Heatmap Generation**: The class activation map M highlights the regions of the input image that contribute the most to the prediction of a specific class c.

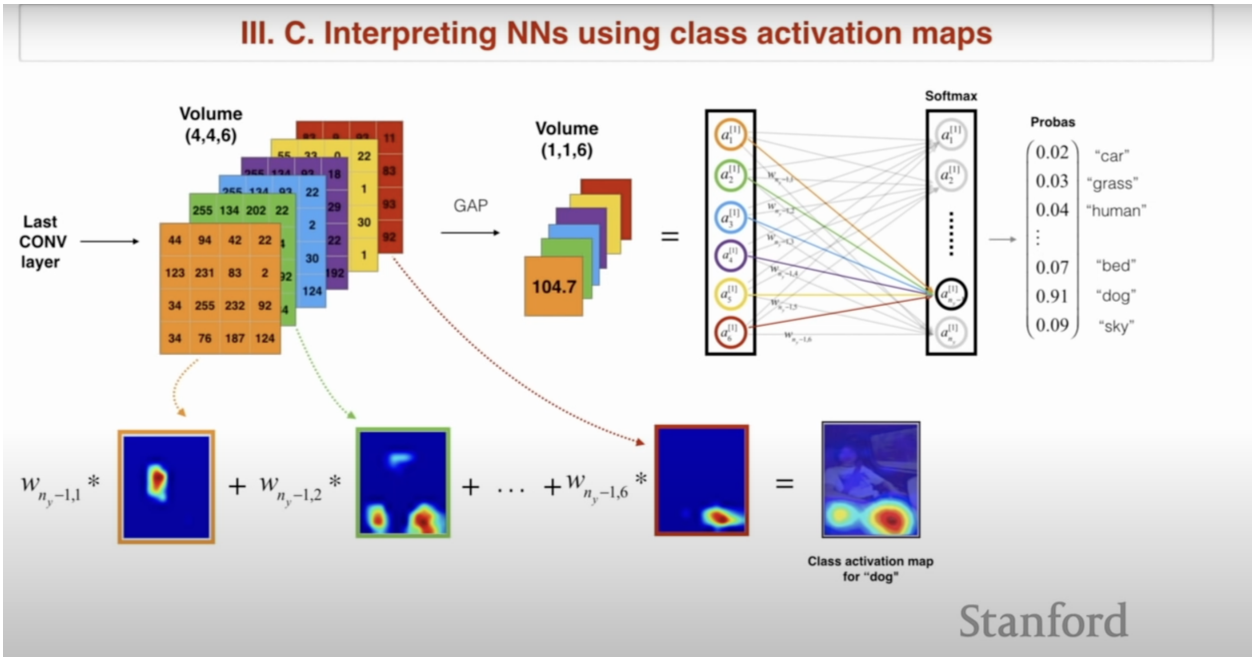
Workflow:

1. Forward pass through the pre-trained CNN to obtain the feature maps.
2. Compute the gradients of the predicted class score with respect to the feature maps.
3. Apply GAP to obtain the class activation map.
4. Generate a heatmap overlaying the activation map on the input image.

CAMS in a view: These two images are taken from CS203, Neural Networks course.

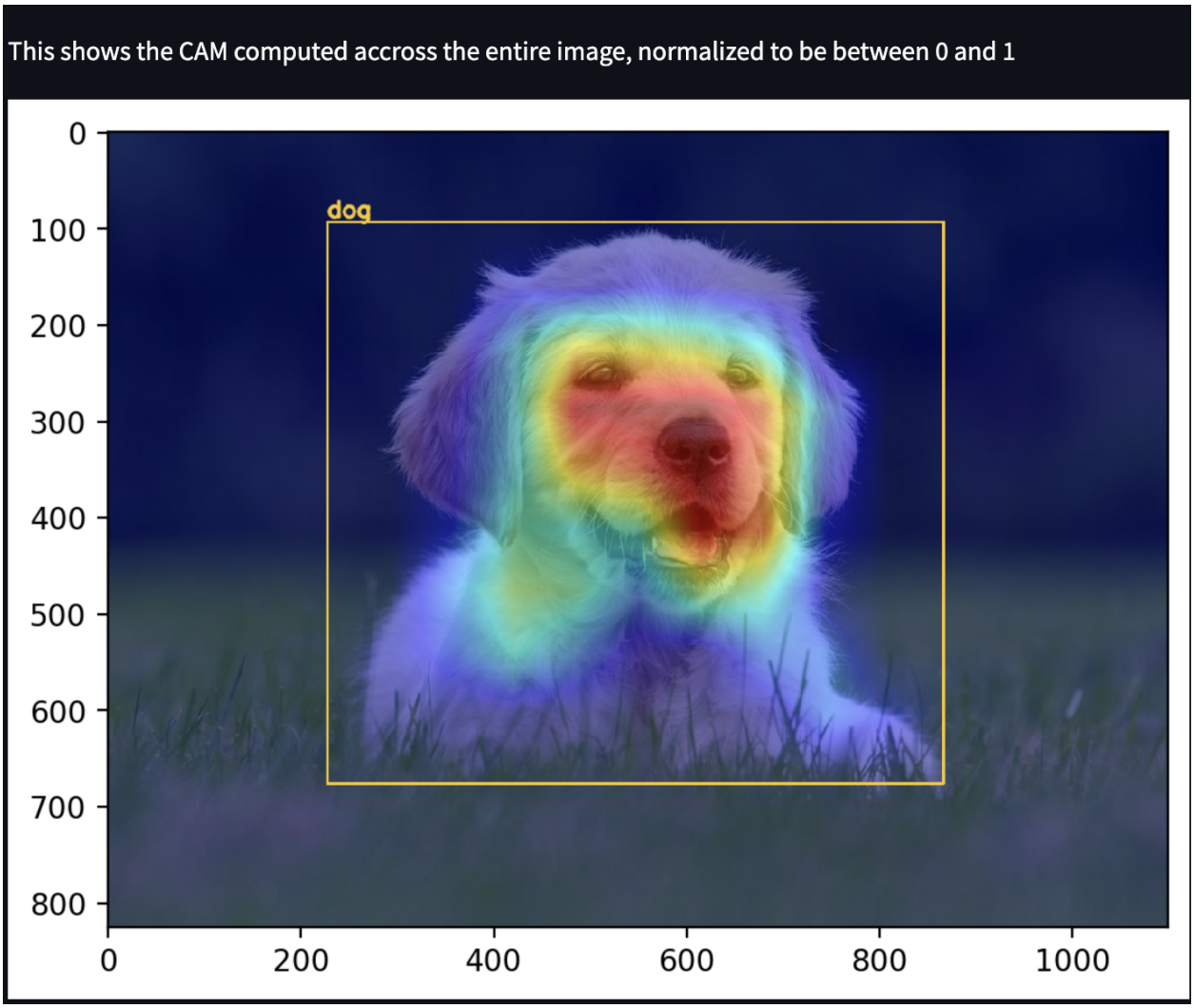
1. The first image represents how the last layer of CNNs are converted to a different network using Global Average Pooling followd by fully connected and a softmax layer.
2. The second image represents how feature maps are averaged to generate a final heatmap that is responsible for predicting the final target class.





Example: Consider an image of a dog classified as a "Labrador Retriever" by the model. CAMs highlight the regions in the image, such as the dog's face and body, that influenced the classification decision.

Image from dashboard: The image showcases a dog with its face prominently highlighted using class activation maps (CAMs). This visualization provides insights into the regions of the image that contribute most to the model's prediction, enhancing our understanding of the model's decision-making process.



4.2 Deep Feature Factorization (DFF):

Description: Deep Feature Factorization (DFF) is an innovative method for creating insightful visualizations about the internal representations of deep neural networks (DNNs). Unlike traditional explainability methods such as Grad-CAM, which focus on highlighting regions of an image that contribute to a specific category prediction, DFF delves deeper into the underlying concepts discovered by the model and how they are classified.

Motivations for Deep Feature Factorization: Traditional explainability methods like Grad-CAM provide valuable insights into where the model looks in an image to make predictions. However, they have limitations:

- Lack of granularity: They often fail to reveal the internal concepts the model identifies, such as individual object parts or features.
- Limited to target categories: They generate heatmaps specific to a single target category, making it difficult to understand the contributions of multiple objects in the image.
- Complexity in visualization: The visualization of multiple heatmaps for different categories can be overwhelming and inefficient for interpretation.

Deep Feature Factorization Workflow:

1. Reshaping Activations: The activations from the last convolutional layer of the pre-trained CNN are reshaped into 1D vectors to prepare them for factorization.
2. Non-negative Matrix Factorization (NMF): NMF is applied to these vectors to decompose them into distinct concepts. This process results in two matrices: W containing the feature representations of the detected concepts, and H containing how the pixels correspond with these concepts.
3. Concept Classification: If using the activations from the last layer, the remaining part of the network (e.g., fully connected layers) can be run on the concepts to classify each concept. This step provides insight into how the model interprets each concept.
4. Concept Visualization: To create a single visualization summarizing all concepts, each concept is assigned a unique color, and the intensity is modulated based on the heatmap. This approach ensures that the most important concept for each pixel is retained in the final visualization.

Mathematical Explanation:

- **Concept Activation Computation**: DFF computes the concept activations C by analyzing the neuron activations A^l in the network:
$$C = A^l \cdot W$$
where W is a weight matrix.
- **Factorization**: The concept activations C are factorized into interpretable concepts X using techniques such as Singular Value Decomposition (SVD) or Non-negative Matrix Factorization (NMF):
$$C = X \cdot Y$$
- **Visualization**: The extracted concepts X are visualized to provide insights into the learned representations of the model.

Connecting Concepts with Model Output: Using activations from the last convolutional layer simplifies connecting concepts with the model's output. For example, in ResNet50, running the fully connected layer on the concepts facilitates classification. For activations from earlier layers, alternative approaches like unpacking concepts to 2D tensors or analyzing concept heatmaps can be explored.

Example of Deep Feature Factorization: Consider an image classification task where DFF is applied to visualize concepts within an image of a dog. DFF may reveal concepts such as "dog face," "dog body," "dog fur," and "paws," providing granular insights into the model's interpretation of the image. Each concept is assigned a distinct color, and the intensity reflects its importance in the final classification.

Image from dashboard: The image displays a dog identified as a Golden Retriever by the ResNet model, with deep feature factorization (DFF) revealing multiple classifications. The model identifies the face region as a Golden Retriever while categorizing other parts as distinct entities such as hare and wood rabbit, showcasing the intricate interpretation of features learned by the neural network.

DFF image from the paper [Deep Feature Factorization]:(https://arxiv.org/pdf/1806.10206) The provided image showcases the effectiveness of Deep Feature Factorization (DFF) in identifying common elements across multiple images. Through DFF, features such as pyramids, animals, people, and monument parts are accurately matched and corresponded across different images, highlighting the robustness and interpretability of the method in understanding similarities within diverse visual datasets.

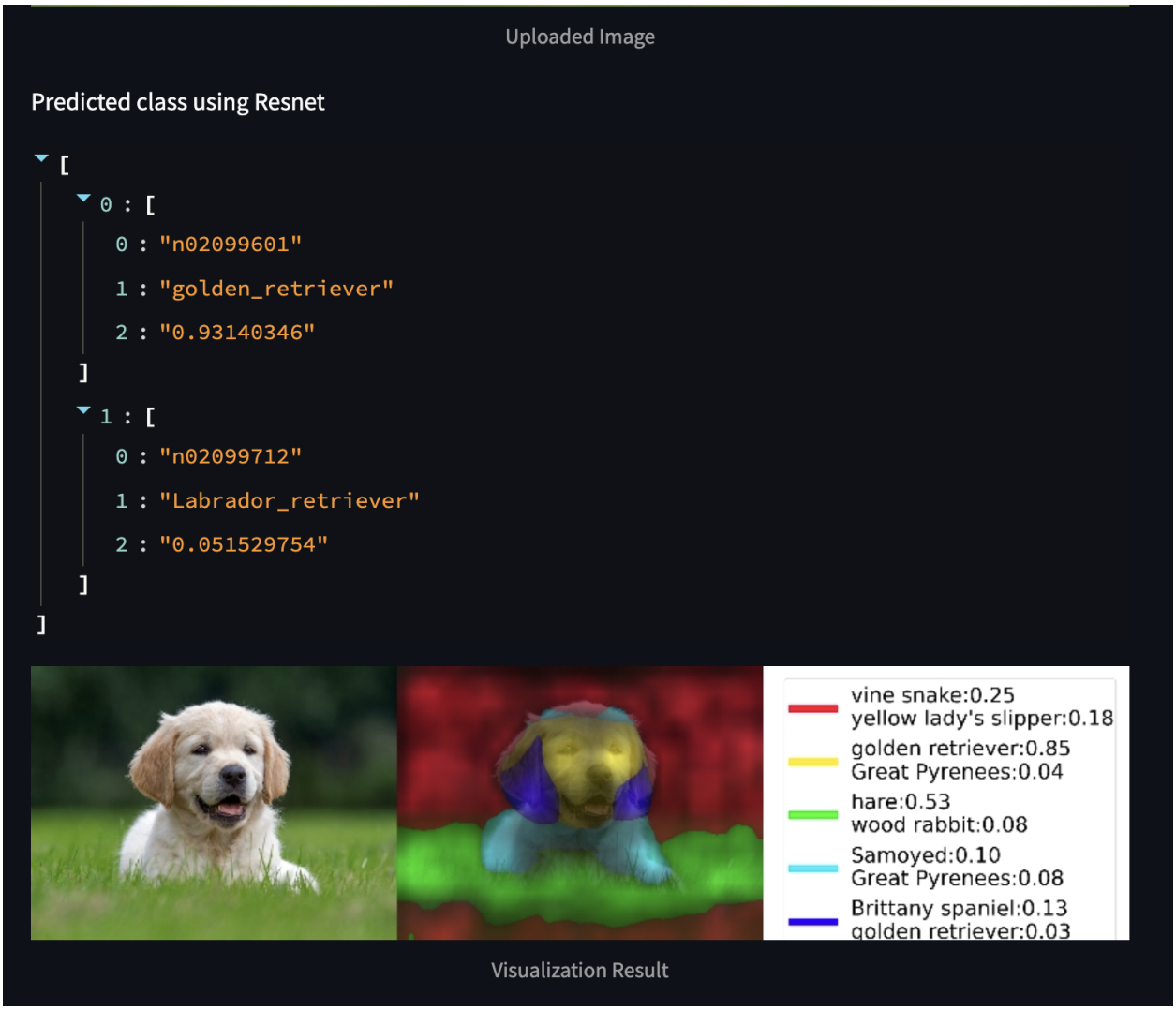


Figure 1: DFF Example1

Fig. 1: *What in this picture is the same as in the other pictures?* Our method, Deep Feature Factorization (DFF), allows us to see how a deep CNN trained for image classification would answer this question. (a) Pyramids, animals and people correspond across images. (b) Monument parts match with each other.

Figure 2: DFF illustration

Conclusion: Deep Feature Factorization offers a more detailed and nuanced understanding of DNNs compared to traditional explainability methods. By uncovering internal concepts and their classifications, DFF enhances transparency and interpretability, leading to more trust in deep learning models. The ability to visualize multiple concepts in a single image streamlines interpretation and facilitates deeper insights into model behavior.

5. Experimental Setup:

As we focus on demonstrating the application of explainability techniques rather than training models, our experimental setup involves loading pre-trained CNN models and processing sample images to generate visualizations.

6. Results:

We present visualizations generated using CAMs and DFF for various sample images, highlighting the regions and concepts learned by the pre-trained CNN models. These visualizations provide insights into the model's decision-making process and help interpret its predictions.

7. Summary and Conclusions:

Explainability techniques such as CAMs and DFF offer valuable insights into the inner workings of deep learning models. By visualizing the regions and concepts that influence model predictions, we enhance our understanding and trust in these complex systems. Moving forward, further research and development of explainability techniques will continue to drive progress in the field of AI transparency and interpretability.

Percentage Of Borrowed Work:

1. Lines of Code from Internet : 300
2. Lines of Code from Internet I modified : 50

3. Lines of Code I wrote : 100
4. Percentage : 62.5 %

References:

1. Deep Feature Factorization For Concept Discovery (https://arxiv.org/pdf/1806.10206)
2. Class Activation maps (http://cnnlocalization.csail.mit.edu/)
3. Youtube Stanford lectures (https://youtu.be/gCJCgQW_LKc?si=wtQbci_TAPL8FsWE)