Explainability in Deep Models

By

Ramprasad TC

Vinod Kumar

Sharad Kumar Patesaria

Rajesh Dey

Sailaja Velpula

Report

Submitted in partial fulfillment of the requirements for the

DLFA Program

Centre for Continuing Education

Indian Institute of Science

Bangalore – 560 012 India

Abstract

Explainable AI (XAI) is the recent advancement in bringing transparency to complex AI Deep Model which helps define model accuracy, transparency and clarity in decision-making process. XAI frameworks are established for entrusting while implementation of Deep models.

The primary objective of this study is to examine the visual explanation property of interpretability methods adopted by XAI frameworks by making an attempt to understand how Convolutional neural networks can be explained. The primary motive is to understand and infer precisely how a model predicts the output. Our experimentation is targeted for XAI interpretability of image detection and classification by CNN's. We have used widely adopted XAI frameworks like LIME, Grad-CAM and Grad-CAM++ with CNN models and image datasets for our experimentation from a theoretical perspective and to observe the outcomes of these XAI techniques can infer model reasoning and generalization.

We also studied few impacts and drawbacks when these methods are applied with respect to choice of Deep models, datasets based on model's complexity, distribution on input data (images used for this experiment) and how application of these models vary depending on the factors mentioned.

Since these XAI frameworks / techniques are being used in multiple domain areas, the choice of the appropriate XAI frameworks is also crucial in interpreting and enhancing the model's capabilities.

Keywords: Explainability, XAI, CNN, Grad-CAM, Grad-CAM++, LIME

Table of Contents

Chapter 1: Introduction

Problem Statement

Despite widespread adoption of machine learning and deep neural models which illustrated state-of-art results and elicited performances, they remain mostly black boxes for end users, hence, inherently unexplainable. Understanding the reasons behind predictions is, however, quite important in assessing trust, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model.

Especially in the scope of Deep Learning, the complexity of Deep Neural Networks leads to pertinent questions about accuracy of model prediction, over estimation of model performance, data contamination and leakage.

Hence, a new research direction of XAI (Explainable Artificial Intelligence) was envisaged to provide solutions for understanding the intricacies of Deep neural networks as well as ML models [1] [2].

This area of Explainable AI (XAI) allows stakeholders to understand and entrust the model's decision-making process by demystifying the model's complex algorithms [2].

Purpose of Study

To understand how exactly AI Models take decisions based on different features and variables and take necessary action to correct them based on the results.

Overview

AI creates a strong foundation, but we need additional frameworks to help the user understand how the model makes decisions. Explainable AI techniques provide the means to try to unravel the mysteries of AI decision-making, helping end users easily understand and interpret model predictions [3].

For instance, suppose an AI model flags a warning that an email is fraudulent. Before the user makes any decision whether he has to keep or discard the email, the user may want to know, why did the model take this decision, what are the features/variables that triggered this conclusion? Explainable AI (XAI) can help answer these questions.

XAI is essential for:

- Ensuring transparency
- Trust and accountability in AI, especially in high-stakes applications like healthcare, finance, and law.
- XAI enables responsible and human-centered AI deployment.
- XAI addresses concerns that arise from the "black-box" nature of many AI models, especially complex models like deep neural networks, by providing a means to understand and interpret how these models work.

Applications of Explainable AI: Healthcare, Finance, Autonomous Vehicles, Criminal Justice.

Current Benchmark

There are several Explainable AI (XAI) frameworks available depending on different techniques and the underlying models they are applied on. Some of them are mentioned below:

- **LIME (Local Interpretable Model-Agnostic Explanations) –** It is based on the popular approximation in the perturbation-based methods - the local linear approximation.
- **SHAP (SHapley Additive exPlanations) -** It is an interpretability framework for machine learning models, based on game theory concepts.
- **Grad-CAM (Gradient-weighted Class Activation Mapping) & Grad-CAM++** - It is a technique for visually interpreting decisions made by convolutional neural networks (CNNs), particularly in image classification and object detection tasks.
- **DAX (distillation aided explainability)** - is a framework that combines model distillation with explainability techniques to make complex, often opaque models (like deep neural networks) more interpretable. It is a gradient free framework.

Assumptions and Limitations

We are limiting our experimentation to understand the explainability using the 3 existing XAI frameworks which are applied on various Deep models for interpretability and explanation for better understanding of interpretability by XAI frameworks, visualize some observations and make inferences based on the outcome of multiple parameters.

Chapter 2: Method/Experimentation

Introduction

Since, Deep models are core to many advances in various fields, the trust on model behavior and its prediction remains crucial when used for real time data and for critical engagements related to healthcare, justice, autonomous driving etc. Hence, evaluation of model is an important aspect before even deployed on real world data which is often significantly different then the validation dataset. Mostly, models are evaluated using accuracy metrics on validation dataset which is mostly post prediction outcome.

However, models need to be interpreted during the validation which gives us more meaningful insights on model's prediction accuracies.

As a part of this experimentation, our plan is to show usage of different XAI techniques to inspect the individual predictions and with explanations which helps in understanding models behavior and outcome. We will implement couple of Explainable AI framework like LIME, Grad-CAM, Grad-CAM++ on top of a Deep models for providing explanations for individual predictions as a solution to "trusting a prediction/trusting a model" and explain them why XAI is useful. We will also try to understand model weights and how it allows us to interpret features and inputs.

Choice of dataset

Since, the criteria for XAI as a part of this paper is for object detection We have used dataset as follows:

a. **ImageNet** – ImageNet is a large-scale, widely used database of labeled images, specifically designed to aid in visual object recognition research.

ImageNet contains millions of images organized by the WordNet hierarchy, covering over 20,000 categories.

b. **Pascal VOC** – The Pascal Visual Object Classes (VOC) dataset is another influential dataset in computer vision research, particularly for tasks like object detection, image classification, and segmentation. The dataset contains images of 20 primary object categories, such as person, bicycle, car, dog, bird, and more, which were carefully chosen to represent common objects in various scenes.

Choice of Model

For following experimentations below pretrained models were chosen

I. ResNet50: A pre-trained ResNet-50 model is used. This model is designed for image classification tasks with 1,000 classes from the ImageNet dataset.

II. DeepLavv3: This is used with the Pascal VOC dataset for object detection.

III. For a model selection study some pretrained models were used e.g. AlexNet, VGG16, Resnet101, Densenet161, Squeezenet 1.1

Interpretability and Explanation

1. Using LIME with ResNet50 & IMAGENET dataset:

This experiment demonstrates how LIME can be applied to a pre-trained ResNet-50 model for image classification tasks.

Purpose of LIME: Provides local interpretability for any black-box model (including for Object detection) by approximating it with a simpler interpretable model in the vicinity of a specific instance

How LIME Interprets the model (How LIME works):

I. Perturbation

LIME generates a set of perturbed samples around the original input, while preserving the local structure of the data

I. **Model Evaluation**
LIME then evaluates the black-box model on these perturbed samples to understand how the model's predictions change.

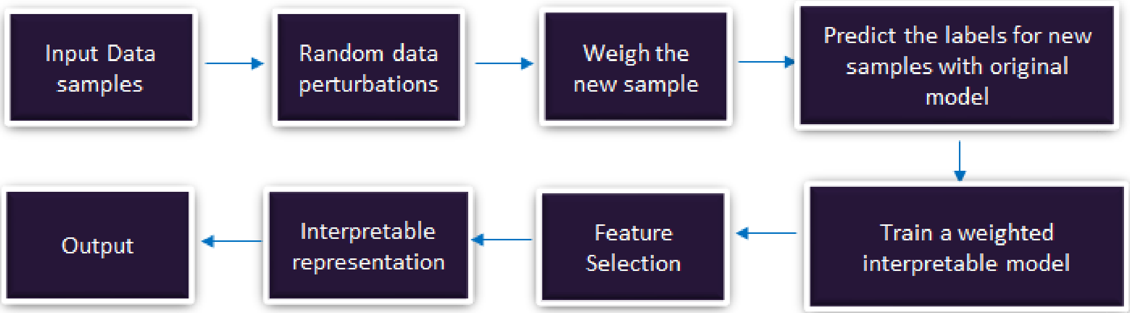
II. **Interpretation**
Finally, LIME trains a simple, interpretable model to approximate the black-box model's behavior in the local region.

III. **Model Selection**: A pre-trained ResNet-50 model from PyTorch's torch-vision library was loaded. This model is designed for image classification tasks with 1,000 classes from the ImageNet dataset.

IV. **Image Preprocessing**: The input image was preprocessed to match the input requirements of the ResNet model. This involved resizing the image to 224x224 pixels, converting it to a tensor, and normalizing it using the mean and standard deviation specific to ImageNet.

V. **Prediction Function**: A prediction function was defined to convert images into the appropriate format, perform inference using the ResNet model, and return the softmax probabilities of the predicted classes.

VI. **LIME Explanation**: An instance of LimeImageExplainer was created, and the explain_instance method was used to generate explanations for the selected image. The method specified the top labels to consider, the number of samples for perturbation, and the color to hide.

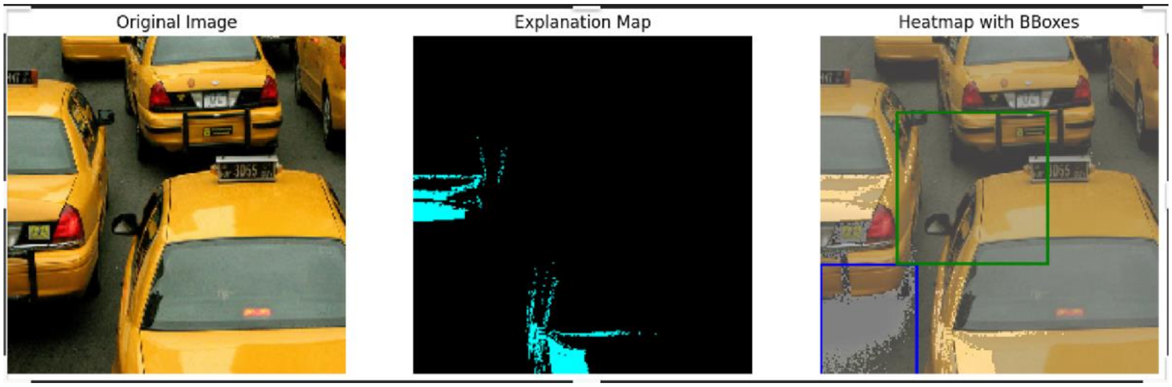

Input Data samples
Random data perturbations
Weigh the new sample
Predict the labels for new samples with original model
Train a weighted interpretable model
Feature Selection
Interpretable representation
Output

Fig; LIME workflow (How LIME Works)

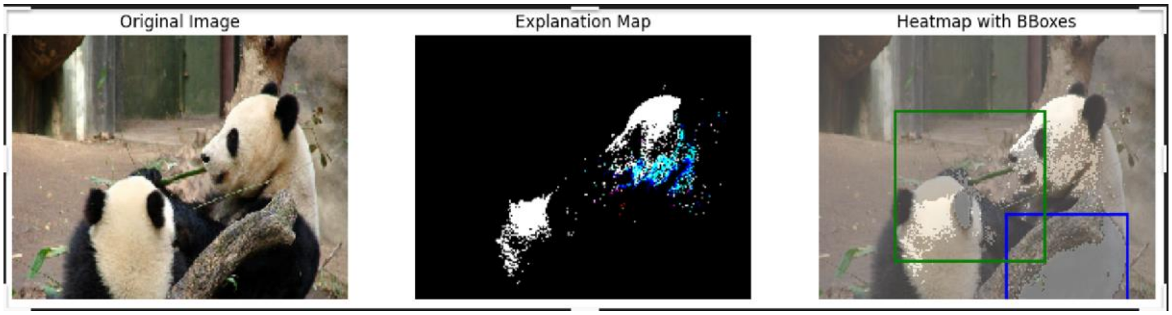
VII. **Observations / Results**: The LIME experiment produced visual explanations for the top predicted labels of the input image. Each explanation included an overlaid mask indicating the regions of the image that contributed most to the model's decision.

The results demonstrated that LIME effectively highlighted critical areas of the image relevant to the classification, providing insights into the model's behavior and decision-making process.

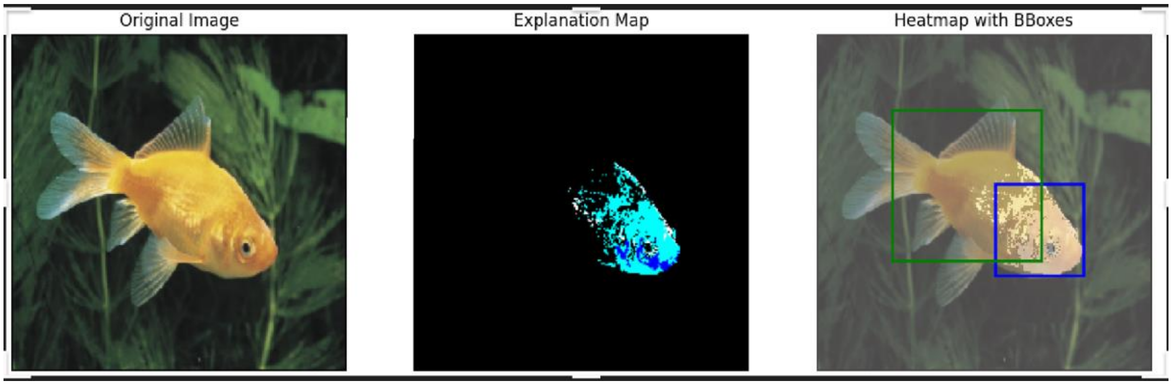
a. **Multiple Objects**



b. **Two Objects**



c. **Single Object**



2. **Using Grad-CAM with ResNet50 & IMAGENET dataset:**

This experiment demonstrates how Grad-CAM can be applied to a pre-trained ResNet-50 model for image classification tasks.

Purpose of Grad-CAM: Visualizes important regions in an image that influence the prediction of a convolutional neural network (CNN). Grad-CAM (Gradient-weighted Class Activation Mapping) is a visualization technique used in deep learning to help interpret and understand how Convolutional Neural Networks (CNNs) make the decisions. It highlights the important regions in an input image that contribute to the network's predictions

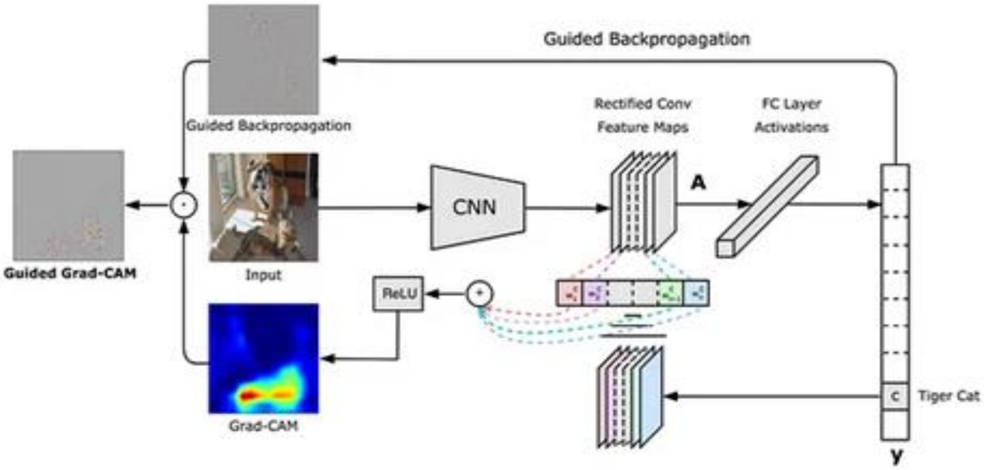


Fig: Grad-CAM++ (An overview of Grad-CAM computational flow)

How Grad-CAM Interprets the model (How Grad-CAM works):

I. **Gradient Calculation:** Grad-CAM calculates the gradients of the final classification score with respect to the activations of the last convolutional layer.

II. **Weighting**: The gradients are averaged over the spatial dimensions, resulting in weights for each feature map in the last convolutional layer.

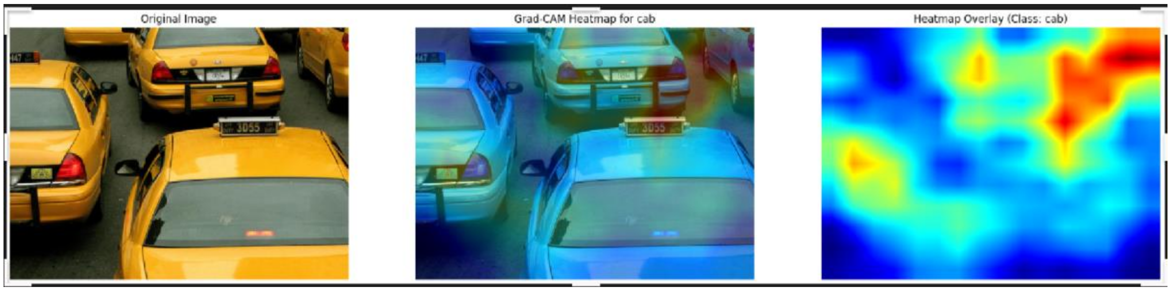
III. **Heatmap Generation:** The weights are multiplied with the corresponding feature maps, and the results are summed to create a heatmap. This heatmap highlights the regions in the image that are most important for the model's decision.

IV. **Grad-CAM Explainability:** Grad-CAM is used to explain different epochs and to understand which images are easier to identify different stages in prediction model development.

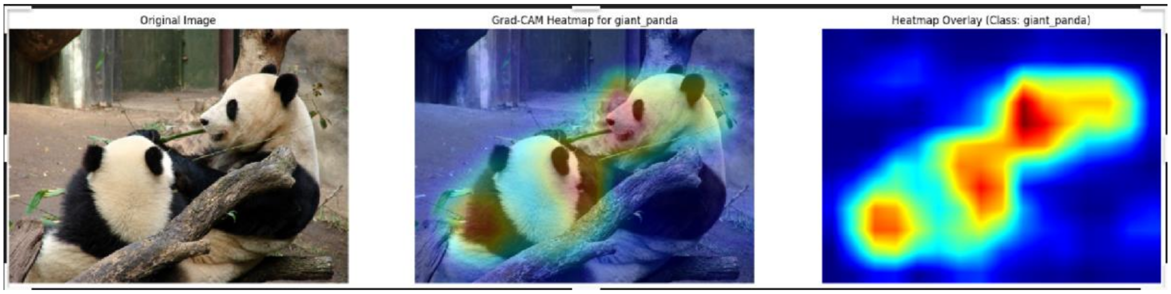
V. **Observations:** this includes certain issues and benefits of the model using XAI.

   a. By flowing the gradient information into the last convolutional layer of CNNs, Gradient-weighted Class Activation Mapping (Grad-CAM) computes a feature-importance map (i.e., a coarse localization) highlighting regions in the image corresponding to a certain concept
   b. The gradient methods utilize the gradients of the loss with respect to each input token such as an image region or a word in the question to provide an explanation for visual reasoning.
   a. Gradient-based methods generate post-hoc attribution maps by utilizing gradients or backpropagation to identify the important parts of the input image for the prediction.
   b. Gradient-weighted class activation mapping (Grad-Cam) visualizing attention weights, and visualization of embedded feature spaces were three major approaches to cope with model explainability

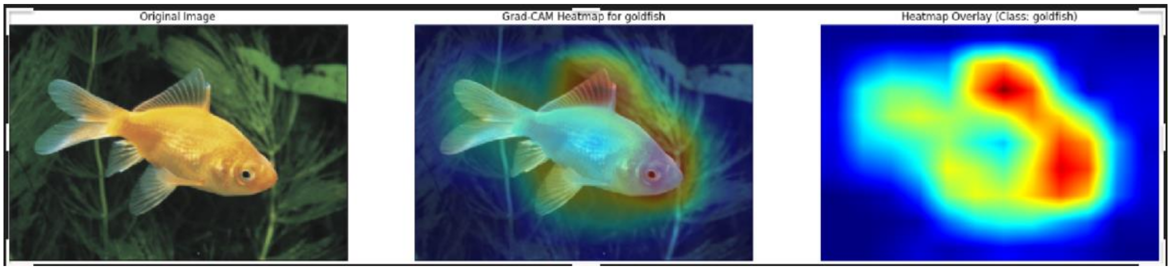
a. Multiple Objects



b. Two Objects


Original Image
Grad-CAM Heatmap for giant_panda
Heatmap Overlay (Class: giant_panda)

c. **Single Object**


Original Image
Grad-CAM Heatmap for goldfish
Heatmap Overlay (Class: goldfish)

3. Using Grad-CAM++ with ResNet50 & IMAGENET dataset:

This experiment demonstrates how Grad-CAM++ can be applied to a pre-trained ResNet-50 model for image classification tasks.

Purpose of Grad-CAM++: An extension of Grad-CAM, offering improved localization of important regions in images that influence the prediction of a convolutional neural network (CNN).

Thus, Grad-CAM++, as its name suggests, can be (loosely) considered a generalized formulation of Grad-CAM.

Grad-CAM++ is an advanced visualization technique that builds upon the original Grad-CAM method. It provides more accurate and detailed heatmaps by utilizing higher-order gradients.

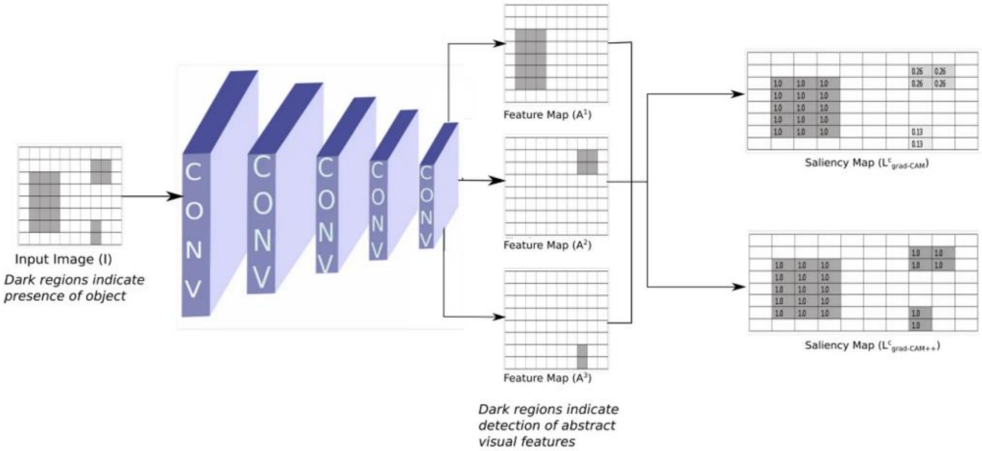

Input Image (I)
Dark regions indicate presence of object
CONV
CONV
CONV
CONV
CONV
Feature Map (A^1)
Feature Map (A^2)
Feature Map (A^3)
Dark regions indicate detection of abstract visual features
Saliency Map ($L^c_{grad\text{-}CAM}$)
Saliency Map ($L^c_{grad\text{-}CAM++}$)

Fig: Grad-CAM++ (An overview of Grad-CAM computational flow)

I. **Model Selection:** A pre-trained ResNet-50 model was loaded from PyTorch's torch-vision library. This model is designed for image classification tasks with 1,000 classes from the ImageNet dataset.

II. **Image Preprocessing**: The input image was loaded and preprocessed to match the input requirements of the ResNet model. This involved resizing the image to 224x224 pixels and converting it to a tensor.

III. **Class Prediction**: A function was defined to predict the class of the input image using the model. The predicted class index was obtained from the model's output.

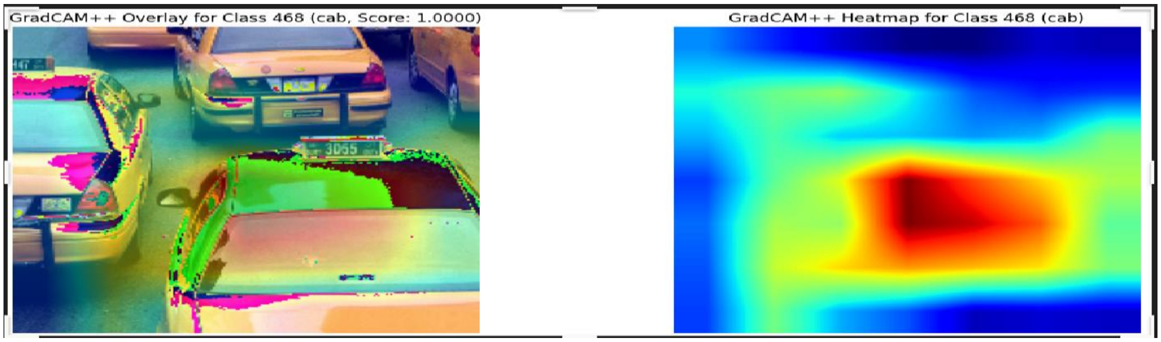
IV. **Backward Pass for Gradients**: A backward pass was performed to compute the gradients for the predicted class score. This step is essential for generating the Grad-CAM++ heatmap.

V. **Heatmap Generation**: The Grad-CAM++ heatmap was generated using the captured gradients and activations. The heatmap was normalized to the range [0, 1] for visualization.

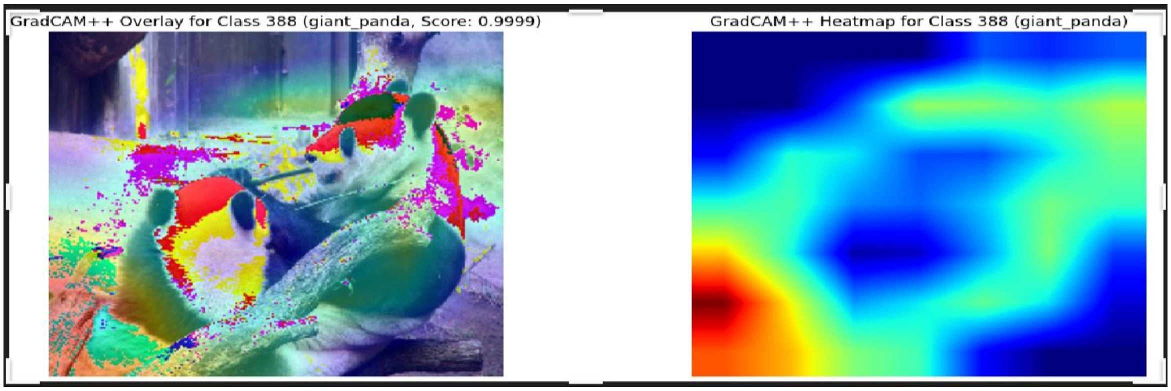
VI. **Observations / Results**: The Grad-CAM++ experiment produced a detailed heatmap overlay on the original image, highlighting the regions that contributed most to the model's prediction for the identified class. The results illustrated how the model focused on specific areas within the image, providing deeper insights into the decision-making process of the ResNet-50 model.

   a. The pixel wise weighting adopted by Grad-CAM++ in generating the visual explanations are more model-appropriate and consistent with the model’s prediction.
   b. Grad-CAM++ tends to highlight the context of the video (similar to images) as less bright and most discriminative parts as brighter regions in the video explanations.

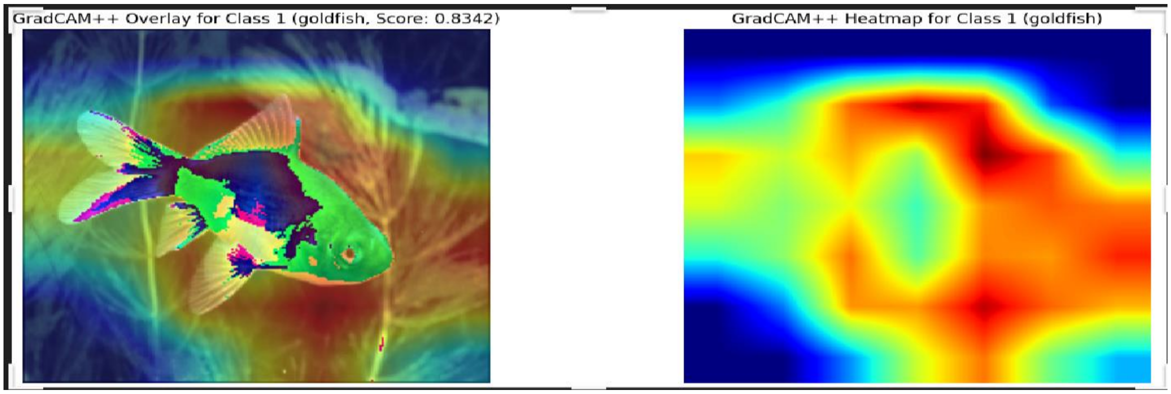
a. Multiple Objects


GradCAM++ Overlay for Class 468 (cab, Score: 1.0000)
GradCAM++ Heatmap for Class 468 (cab)

b. Two Objects


GradCAM++ Overlay for Class 388 (giant_panda, Score: 0.9999)
GradCAM++ Heatmap for Class 388 (giant_panda)

c. Single Object


GradCAM++ Overlay for Class 1 (goldfish, Score: 0.8342)
GradCAM++ Heatmap for Class 1 (goldfish)

Comparative use cases of XAI Frameworks used for experimentation

Model	Model-Agnostic	Scope	Computation cost	Visualization	Typical Domain
LIME	Yes	Local	High	Perturbation-based explanations	Tabular, Image, Text
GradCAM	No	Local	Moderate	Heatmap (Class Discriminative)	Vision (CNN)
GradCAM++	No	Local	Moderate	Improved Heatmap	Vision (CNN)

Computational Cost* for same inference model (ResNet50)

LIME	GradCAM	GradCAM++
177.18 Sec	2.19 Sec	3.44 Sec

Visualization of Learning during training

XAI can be applied as part of the learning process to visualize the process of training. As part of the study, A CNN model was chosen to learn the mask of an image, which was then compared with the actual segmentation (ground truth) to validate the training progression. The Following are two examples from different epoch in the stage of learning.

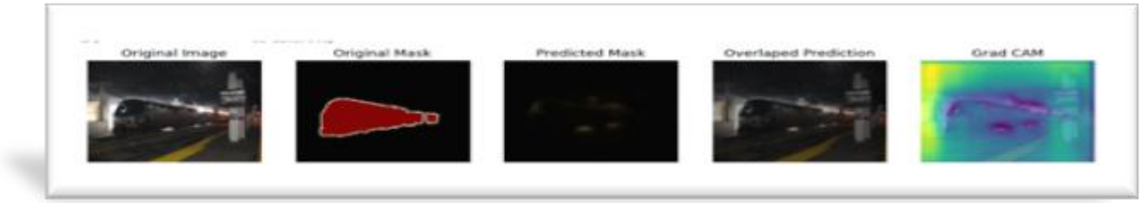


Fig: Showing the learning of model after 15th epoch

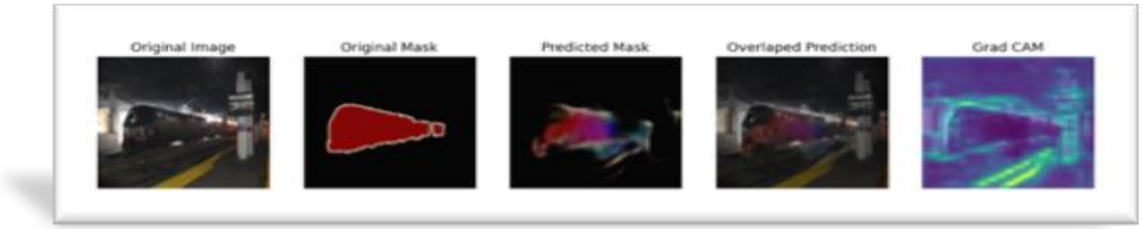
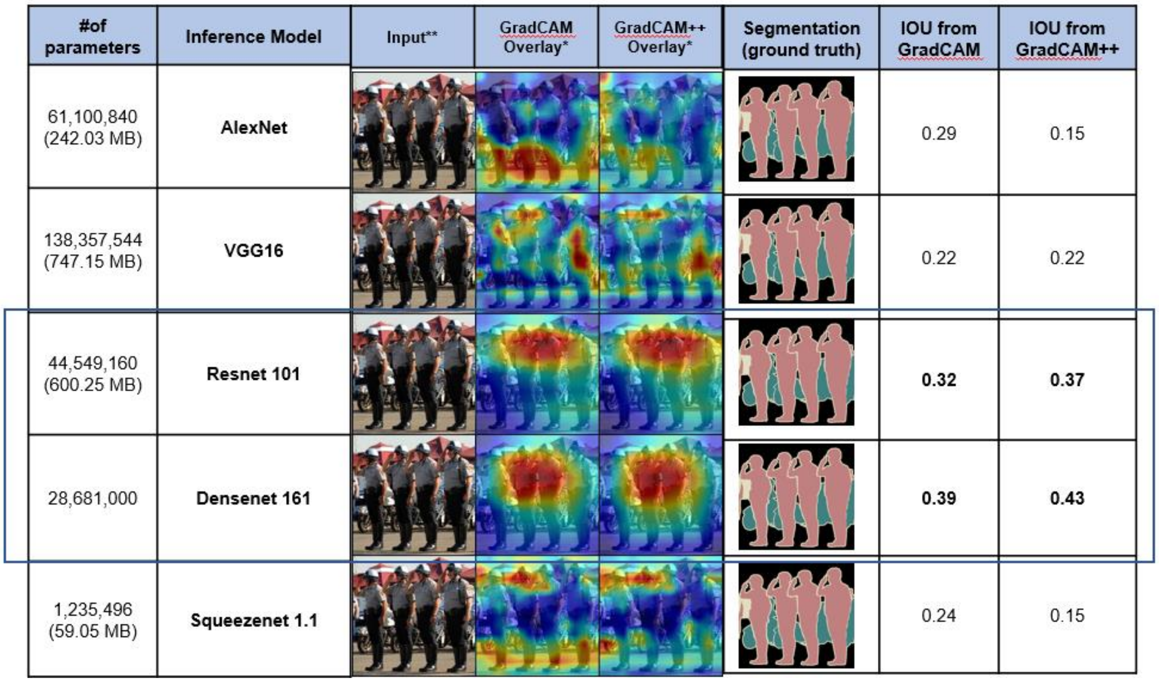


Fig: Showing the learning of model after 45th epoch

From the above study, it is evident that the XAI model (Grad-CAM) can help the user to conclude or understand whether learning progresses in the right direction or not. This again increases the transparency of the CNN model in question.

Using XAI for selection of classification/detection model

Often in actual application situations arises, when the user needs to select a single model out of multiple available model architecture. XAI along with IOU metric can also be used to help understand the performance of the models by visualising the heatmap. In the following study, multiple pre-trained CNN models were chosen and outcome was visualized using heatmap and IOU (Intersection Over Union) metric. From the following table it is quite evident that this kind of comparison can help the end user to select the model based on the requirements.



#of parameters	Inference Model	Input**	GradCAM Overlay*	GradCAM++ Overlay*	Segmentation (ground truth)	IOU from GradCAM	IOU from GradCAM++
61,100,840 (242.03 MB)	**AlexNet**					0.29	0.15
138,357,544 (747.15 MB)	**VGG16**					0.22	0.22
44,549,160 (600.25 MB)	**Resnet 101**					**0.32**	**0.37**
28,681,000	**Densenet 161**					**0.39**	**0.43**
1,235,496 (59.05 MB)	**Squeezenet 1.1**					0.24	0.15

Fig: Above table showing the performance comparison of the pre-trained model on an object detection task. The dataset used here is from Pascal VOC

From the above results, it is evident, even though the object detected on all models is correct, the reliability of the Resnet and Densenet model is higher because the contribution of the pixel group that contributes to the detection/classification is from the actual pixels where the object is present. If we compare the heatmap from AlexNet/VGG16/Squeezenet, even though they have classified/detected the images correctly, the main pixels that are contributing to the decision are not according to the human way of classification so less reliable.

Summary

With advent of various Explainability models, the explanation and hence transparency of various complex Deep models and Neural networks has been achieved through visual outcomes and hence, made the understanding of interpretation easier. Explainability will give us an understanding of which all elements lead to a prediction. This new area is now the focus interest with multiple experimentation to design and develop better XAI techniques for more precision, performance and transparency which can extract inferences about the complex Deep neural networks. There is also a lot of experimentation ongoing for the usability of these XAI techniques with multiple models and datasets for more precise outputs.

Chapter 3: Conclusion & Recommendations

Conclusion

We chose to take up “Explainability AI in Deep Models” to understand the popular XAI frameworks and their capability to make the “black box” transparent.

Key Observations:

- **XAI Framework outcomes:** Different visual explanations were generated by XAI frameworks – Heatmap, binary segmentation etc.

- **XAI Framework Interpretation:** Interpretation of multiple XAI frameworks for the decision-making process of the models

- **Data selection & Model Selection:** Combining the chosen XAI models with Deep models and image datasets, led to the creation of guided visualizations offering the superior interpretability and fidelity to the original models

Key Takeaways:

- XAI techniques are valuable tools for enhancing the transparency of Deep models which help take better decisions for reliability and for future monitoring and improvement
- How XAI and other metrics (IOU) can help us choose best model for us
- Also, XAI can help us visualize very important questions – “When to stop the training?”

XAI model help us prevent *“correct prediction for wrong reason”* – specially in sensitive domain/industries

Recommendations

With the increasing usability of Deep models and advent of the XAI framework to support the deeper understanding of the decision-making process, we studied a few recommendations to use the XAI framework to its full potential. Overall recommendation can be based on multiple combinations of factors.

i. Computational efficiency: Choosing XAI framework for Deep models is a tradeoff for better computational efficiency

ii. Usability: Choosing the appropriate XAI framework for model interpretability is based on the intrinsic XAI algorithm to extract the inferences

iii. Time factor: Time is also one of the significant aspects for the choice of XAI framework to be used. With multiple analysis and experimentation, we can choose the XAI framework for better results considering the time factor for a Deep Model.

iv. Precision: The choice of XAI model is also based on precision in explainability. This aspect can be inferred while experimenting on multiple XAI frameworks with multiple models and on variation of datasets.

References

[1] *https://arxiv.org/pdf/2409.11123*

[2] *https://arxiv.org/pdf/1602.04938*

[3] *https://christophm.github.io/interpretable-ml-book/*

[4] *https://github.com/marcotcr/lime*

[5] https://arxiv.org/pdf/2205.10838

[6] *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization, Selvaraju et al, ICCV, 2017*

[7] *Grad-CAM++: Generalized Gradient-based Visual Explanations for Deep Convolutional Networks, Chattopadhyay et al, WACV, 2018*