A Survey of Explainable AI (XAI) Methods for Convolutional Neural Networks

Antonio Fernando Silva e Cruz Filho [1] João Gabriel Andrade de Araujo Josephik[1] Prof. Dr. Nina S. T. Hirata

[1]Institute of Mathematics and Statistics

Introduction

As artificial intelligence (AI) becomes part of critical fields like healthcare, finance, and autonomous vehicles, it's important to understand how these systems make decisions. This is where Explainable AI (XAI) comes in. XAI helps make AI models, which are often complex, easier to understand and interpret. This ensures that AI systems are trusted and used responsibly.

Neural networks are powerful tools for tasks like recognizing images or making predictions. However, they are often seen as "black boxes" because it's hard to explain how they reach their decisions. This lack of clarity can be a problem in areas where understanding the reason behind a decision is as important as the result itself.

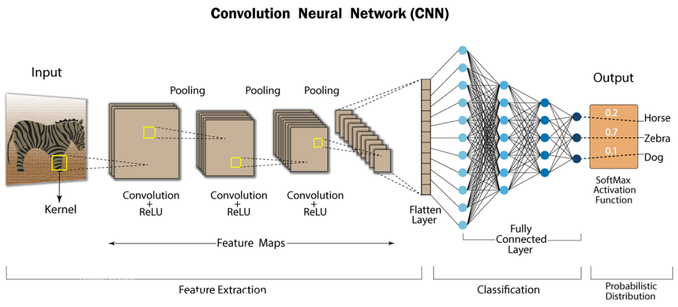

Figure 1. Convolutional Neural Network Architecture

Convolutional Neural Networks (CNNs) are neural networks designed for image and spatial data. They process parts of an image (like edges or textures) and combine these features to make decisions, making them ideal for tasks like face or object recognition. However, CNNs are also complex, and it can be hard to tell what they are focusing on and why. XAI tools can help with this by showing how the CNN works step by step.

Feature Visualization

CNNs can learn abstract features and concepts from images. One can use techniques such as Feature Visualization to visualize the learned features by maximizing a network's neuron (or a set of neurons) value. This technique, called *Activation Maximization*, can be modeled by the formula below, using the *Gradient Ascent* method:

$$x_{t+1} = x_t + \mu \frac{\partial a(\theta, x_t)}{\partial x_t}$$

Where x_t represents an image at iteration t, μ represents a tunable hyperparameter and the function a represents the forward pass of a unit in a Neural Network with parameters θ.

By defining x_0 as a specific image or just random noise, one can create *dreamy-like* [2] images representing that will maximize a certain set of neurons of a Network.

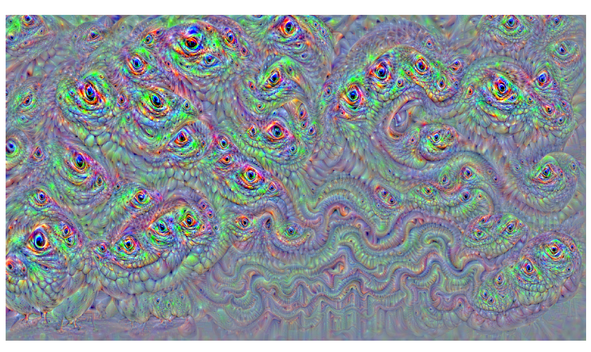

Figure 2. Feature Visualization of CNN VGG16 using a random image

Figure 3. Feature Visualization of CNN VGG16 using a photo of IME-USP

By close inspection in both images, one can notice that certain characteristic features like animal's eyes and dog's faces emerge, showing that the network learned those representations and certain layers are maximized by the presence of those features in images.

GradCAM

At the last convolutional layer of a network, there are multiple channels representing extracted features from the image. Those features may be of multiple natures, such as presence of certain objects, simmetry, contrast, etc. By inspecting those channels, we can get an insight into what information the network is using to make its decision.

The problem is: how can we know which channels are being useful to the decision? GradCAM solves this by averaging the feature maps weighted by the **gradient with regard to the output**. After that, we filter out negative influences by passing the output through the ReLU function. We can then resize the resulting map to the same size and overlay it over the original image to inspect the results.

Figure 4. GradCAM of a polar bear passed though ResNET. Network classification: polar bear, Ursus maritimus.

Figure 5. GradCAM of a airplane passed though ResNET. Network classification: airliner

LIME in Images

LIME (Local interpretable model-agnostic explanations) is a technique to explain a complex model's decisions by creating a simpler explainable model based on the complex model. A LIME model can be defined by the expression bellow:

$$g^*(x) = \arg\min_g L(f, g, \pi_x) + \Omega(g)$$

Where L is a loss function to compare the simpler model performance with the complex model performance in the region close to x, defined by π_x. Ω is a function that returns a *complexity metric* for a simple model g.

LIME can be used in images to create masks that represent areas that were more important for a model's decision.

Figure 6. Original Image

Figure 7. Masked image using LIME

Experiments

We can check a network's robustness empirically by applying different kinds of filters to an image and verifying changes in predictions and changes in focus regions using GradCAM. For example, here, we applied a brightness shift in a Red Fox image to check prediction and GradCAM variation in ResNet18:

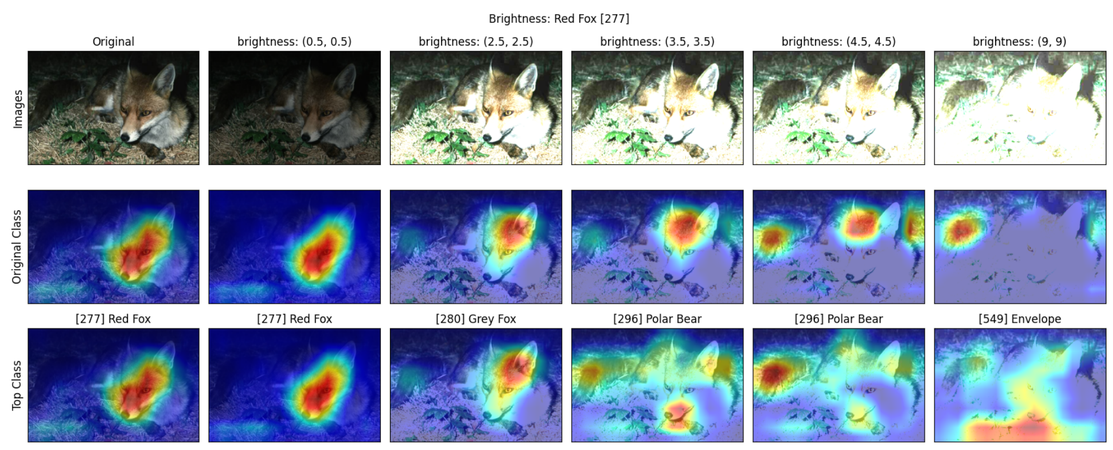

Figure 8. Effect of brightness variation of a Red Fox image in ResNet18 Model

A clear correlation between white objects and a high presence of the color white in the image is reported (predictions change to *Polar Bear* or *Envelope* in really bright images). Also, the GradCAM shows that, when the image is highly occluded by brightness, the fox's face is almost unrecognizable, making the model *"pay attention"* to the animal's tail, since this feature is still recognizable in this high brightness setting.

Conclusion

Explainability tools like Feature Visualization, GradCAM and LIME enhance transparency by revealing how models make predictions. This helps identify biases, debug issues, and build trust, ensuring AI systems are reliable and accountable.

References

[1] C. Molnar. Interpretable machine learning: A guide for making black box models explainable. `https://christophm.github.io/interpretable-ml-book/`. Accessed: 2024-11-26.

[2] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. `https://web.archive.org/web/20150703064823/http://googleresearch.blogspot.co.uk/2015/06/inceptionism-going-deeper-into-neural.html`. Accessed: 2024-11-26.

[3] M. T. Ribeiro, S. Singh, and C. Guestrin. "why should i trust you?": Explaining the predictions of any classifier, 2016. URL `https://arxiv.org/abs/1602.04938`.

[4] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra. Grad-cam: Visual explanations from deep networks via gradient-based localization. *International Journal of Computer Vision*, 128(2):336–359, Oct. 2019. ISSN 1573-1405. doi: 10.1007/s11263-019-01228-7. URL `http://dx.doi.org/10.1007/s11263-019-01228-7`.