X-CV Primary attribution methods for Image Classification

Abhishek Kumar Singh
220982845
ec22395@qmul.ac.uk

***Abstract*—Explainable AI has been a field in growing demand because it characterizes model accuracy, fairness, transparency, and outcomes in AI-powered decision making. In this work we aim to give an overview of Explainability in Computer Vision and it's methods for Image classification application using relevant metric for their evaluation. We observe visually and quantitatively how each of these attribution methods in tandem perform for this task.**

I. Introduction

Explainability for AI[1] typically refers to post hoc analysis and techniques used to understand a previously trained model or predictions. Explainability seeks to augment the training process, the learned representations and the decision with human-interpretable explainations. Explainability in Computer Vision helps in gaining insight on the blackbox model and tells us which image features are contributing towards the final decision. There are many methods and techniques which can be used to interpret CV models, in this report we will discuss some of these and compare them for the image classification problem.

II. Problem Definition

The complexity of a state-of-the-art model for image classification hinders the ability to explain and understand the model predictions in an interpretable way. In this report we aim to contrast and compare how different explainability methods for Computer vision can be used to explain how different features of an image are contributing to the classification process.

The need for explainability is so important because the insights leads to more trust in the model, better compliance with laws and regulations, ameliorates the trust barrier for many companies and investors. Few examples where explainability plays a crucial role would be medical image diagnosis, damage assessment in insurance and self-driving cars.

III. Key Works

The interpretation methods for CV can be classified into two broad categories, Primary attribution and Layer attribution. The Primary attribution methods determine the contribution of each input features to a model's output. Some major works in this category are Integrated Gradients,Occlusion , Guided Grad-CAM, LIME [2] etc. Layer attribution methods evaluate the contribution of each neuron in a particular layer to the model's output. These are widely used in the tasks involving Convolutions for eg. Grad-CAM [3].

IV. Evaluation Criteria

For this experimentation I used the pretrained ResNet-18 model trained on the ImageNet dataset. The image used for experimentation is of my dog "Happy". The experimentation is done using Captum and pytorch. The evaluation criterion used by me is time taken and Captum's metric Senstivity Max which measures the extent of explaination change when the input is slightly perturbed.

V. Discussion

This sections shows the results observed from our experimentation on different baselines and the observations we incur from the comparisons based on sensitivity metric.

A. Abbreviations and Acronyms

Explainable-Computer-Vision(X-CV),Convolutional Neural Network(CNN), Integrated Gradients(IG), Guided gradient-weighted class activation mapping(Guided Grad-CAM), Local interpretable model-agnostic explanations(LIME). Smooth Grad on IG(IG-SG).

B. Figures and tables

The work involved on this report was done using lime package in python and Captum[4]. Captum is an open-source extensible library for model interpretability built on PyTorch. For the classification process we used the pretrained ResNet-18 model, which is a deep-CNN that has 18 layers trained on more than a million images from ImageNet databases. The model can classify images into 1000 different classes. The network takes input images of the size 224-by-224. So we had to first center crop our image and then normalize them using mean and standard deviation. After the image has been preprocessed it was passed through the trained model which gave us the prediction of it being a dog with 72% accuracy on ResNet-18. The same image was used with VGG-19 and ResNet-34 but are not covered in this report due to the computational complexities of these models when calculating attributions.
After a class has been successfully identified, we input the preprocessed image through various attribution methods as can be seen from Fig 1.
First the results from Occlusion are computed, which is a perturbation based approach for computing attributions, here we replace each contiguous region with a given baseline and compute the difference in output.We run a sliding window of size 15x15 with a stride of 8 along both dimensions and we occlude the image with baseline value of 0 which corresponds to gray patch We observe although in our case occlusion did not perform exceptionally well but it still identified the parts of the image which contribute to class "Golden Retriever". The parts that contribute the most towards the decision are the face and the lower limbs.
Next, we use the Integrated Gradient and Integrated Gradients with Smooth grad methods. It computes the integral of gradients of the output of the model w.r.t to predicted class and the input image pixels along the path from a black image to our input image. We observe that the output of this method although computes the contributing parts for classification but suffers from a lot of noise Fig1b). This can be reduced by using Smooth Grad on the Integrated Gradient. This smoothens across multiple images by

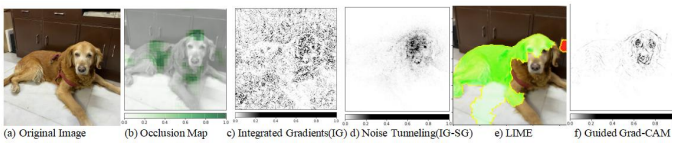

Fig 1. a) Original Image of my dog, b-f) Support for the dog category according to various methods for ResNet-18. b) Occlusion Map highlighting contributing features, c) Integrated Gradients, d) Noise Tunneling after c) to show features clearly, e) LIME model, f) Guided Grad-CAM gives high-resolution class-discriminative visualizations

generating a noise tunnel and it successfully smoothens the attributions across the given noise samples.We observe that visually the contributing part for the classification is the dog's face.

We then use Guided-GradCAM to compute attributions, in which guided backpropagation attributions compute element-wise product of guided GradCAM attributions with upsampled GradCAM attributions. Attribution computation is done for a given layer and upsampled to fit the input size. The last layer of our model is the layer on which we generally apply Guided GradCAM. We observe that the Guided-GradCAM gives really good result and the parts contributing the most towards the classification process here as can be seen from Fig 1 f) is the dog's face.
Lastly, We implemented Lime using lime module in pytorch. Lime is a method which trains an interpretable surrogate model by sampling data points around an input and using model evaluations at these points to train a linear model which is self explanatory. It performs a local sensitivity analysis for each input instance that we pass to the model and their respective predictions.

In our case, Lime and Guided-GradCAM gave the best results out of all the attribution methods we implemented.
This can also be observed from TABLE-I in which we compare the different methods on the basis of the time taken to execute them and their max sensitivity values. Sensitivity gives us the degree of explanation change to tiny input perturbations. The models with a high sensitivity are more prone to adversial attacks. Lime gives us impeccable results but at the cost of taking a lot of time to compute which can be attributed to it's training the local surrogate. Guided-GradCAM not only has a low sensitivity but an extremely .

TABLE I.

Metrics	Attribution methods				
	Occlusion	IG	IG-SG	Guided Grad-CAM	LIME
Max Sensitivity	0.1300	0.4889	0.1600	0.1255	0.0400
Time taken(Secs)	74.29	41.49	84.55	0.299	85.97

[a] Sample of a Table footnote. (*Table footnote*)

VI. Conclusion

We empirically assess the primary attribution methods for image classification applications to find that the few methods like guided-GradCAM and Lime work exceptionally well for our case and both are visually and quantitatively proven in this work. Both of these methods work well for CNNs. The one drawback is that we only tested this on few sample images, the results could dramatically differ for certain samples. The metric for evaluating these methods are also not perfect as they take hours to compute, also there are certain hyperparameters in these methods which when tuned give better results but at high computation cost.

References

[1] W. Samek, G. Montavon, S. Lapuschkin, C. J. Anders and K. -R. Müller, "Explaining Deep Neural Networks and Beyond: A Review of Methods and Applications," in Proceedings of the IEEE, vol. 109, no. 3, pp. 247-278, March 2021, doi: 10.1109/JPROC.2021.3060483.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why Should I Trust You?": Explaining the Predictions of Any Classifier. arXiv. https://doi.org/10.48550/arXiv.1602.04938R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[3] 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," 2017 IEEE International Conference on Computer Vision (ICCV), 2017, pp. 618-626, doi: 10.1109/ICCV.2017.74.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[4] Kokhlikyan, N., Miglani, V., Martin, M., Wang, E., Alsallakh, B., Reynolds, J., Melnikov, A., Kliushkina, N., Araya, C., & Yan, S. (2020). Captum: A unified and generic model interpretability library for PyTorch. arXiv. https://doi.org/10.48550/arXiv.2009.07896.