CAM and Grad-CAM

Sojung Yun
soysoj1810@gmail.com

Abstract

In recent years, deep learning models have achieved remarkable success across various domains, including computer vision, natural language processing, and speech recognition. Despite their impressive performance, interpretability remains a significant challenge. This paper focuses on two popular methods for generating visual explanations: Class Activation Mapping (CAM) and Gradient-weighted Class Activation Mapping (Grad-CAM). We conduct a thorough comparison of these methods using the Stanford Dogs dataset and a pre-trained ResNet-50 model to evaluate their strengths and weaknesses. CAM and Grad-CAM heatmaps were generated and compared visually, and bounding boxes were created to highlight the location of dogs in the images. The bounding boxes were evaluated using the Intersection over Union (IoU) metric. Our results indicate that both methods effectively localize regions of interest, with Grad-CAM achieving a slightly higher IoU score (0.868396) compared to CAM (0.866587). The detailed visualizations provided by Grad-CAM offer better insights into the model's decision-making process, highlighting its superiority in terms of explainability. This study underscores the importance of visual explanations in understanding deep learning models and enhancing their transparency and trustworthiness.

1 Introduction

In recent years, deep learning models have achieved remarkable success across various domains, including computer vision, natural language processing, and speech recognition. Despite their impressive performance, a significant challenge remains: the interpretability of these models. Understanding how a model arrives at a particular decision is crucial for several reasons. First, it builds trust with users by providing transparency, which is essential in critical applications such as medical diagnosis, autonomous driving, and financial forecasting. Second, it aids in the debugging and improvement of models by identifying potential biases and errors. Finally, interpretability facilitates compliance with regulatory requirements, which increasingly demand explainable AI solutions.

Visual explanations are particularly valuable for interpreting deep learning models, especially in the field of computer vision. These explanations help in understanding which parts of an input image are most influential in the model's decision-making process. By highlighting these regions, visual explanations can provide insights into the model's behavior, making it easier to diagnose and rectify errors or biases. Moreover, visual explanations can enhance user interaction and trust by making the model's operations more transparent and understandable.

This research focuses on two popular methods for generating visual explanations: Class Activation Mapping (CAM) and Gradient-weighted Class Activation Mapping (Grad-CAM). CAM generates visual explanations by using the weights of the final convolutional layer, making it limited to specific types of architectures. Grad-CAM, on the other hand, extends this approach by utilizing the gradients of any target concept, making it applicable to a wider range of models. While both methods aim to provide insights into the model's decision-making process, their effectiveness and applicability can vary under different conditions. The primary objective of this study is to conduct a thorough comparison between CAM and Grad-CAM to evaluate their strengths and weaknesses.

We hypothesize that Grad-CAM will demonstrate superior explainability due to its broader applicability and the use of gradient information, which allows for more precise localization of important features.

2 Related Work

The interpretability of deep learning models has been a significant area of research, particularly due to the complexity and often opaque nature of these models. Early efforts in model interpretability focused on simpler methods such as feature importance rankings and decision trees. However, with the advent of deep learning, more sophisticated techniques became necessary to understand the intricate patterns learned by these models.

2.1 Early Methods of Model Interpretability

1. **Feature Importance and Sensitivity Analysis**: Traditional machine learning models often utilized feature importance scores to determine which features most influenced the model's predictions. Sensitivity analysis examined how changes in input features affected the output, providing insights into the model's behavior. However, these methods were limited in their ability to handle the complexity of deep learning models.

2. **Saliency Maps**:Introduced by Simonyan et al. (2013), saliency maps were one of the first attempts to visualize the areas of an input image that most influence the output of a convolutional neural network (CNN). By calculating the gradient of the output with respect to the input image, saliency maps highlight the pixels that significantly affect the model's decision. However, saliency maps have limitations such as gradient saturation, sensitivity to input perturbations, and difficulty in interpretation.

3. **Deconvolution and Guided Backpropagation**:Zeiler and Fergus (2014) introduced deconvolutional networks to visualize the learned features in CNNs by mapping the activations back to the input space. Springenberg et al. (2015) extended this approach with guided backpropagation, which combines deconvolution and standard backpropagation to create more interpretable visualizations [3]. However, both methods can produce misleading visualizations due to the inclusion of irrelevant gradients.

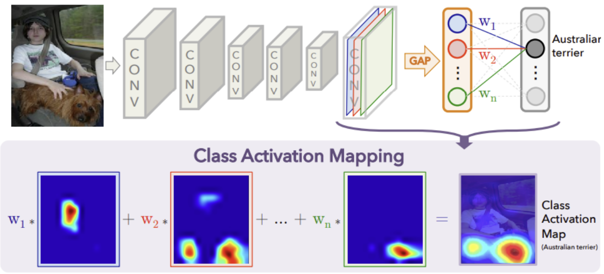


Figure 1: The Architecture of CAM from Learning Deep Features for Discriminative Localization

2.2 Why CAM and Grad-CAM?

CAM and Grad-CAM have been chosen for this study due to their widespread use and proven effectiveness in providing visual explanations for CNNs. CAM offers simplicity and ease of implementation, making it suitable for models with global average pooling layers. Grad-CAM, with its ability to utilize gradient information, offers greater flexibility and precision, making it applicable to a broader range of CNN architectures. By comparing these two methods, we aim to understand their relative strengths and weaknesses in different contexts, providing deeper insights into their applicability and effectiveness in generating visual explanations.

3 Class Activation Mapping (CAM)

3.1 What is CAM?

Class Activation Mapping (CAM) is a technique that provides visual explanations for the decisions made by convolutional neural networks (CNNs). Introduced by Zhou et al. (2015), CAM highlights regions in an input image that are important for the prediction of a specific class. This technique is particularly useful for interpreting the inner workings of deep learning models, thereby enhancing their transparency and trustworthiness.

3.2 Implementation

We applied CAM by modifying the final layers of ResNet-50 to include a global average pooling layer followed by a dense layer for classification.

1. **Forward Pass**: Input the image into the CNN to obtain the feature maps from the last convolutional layer.

2. **Global Average Pooling**: Apply global average pooling to the feature maps to obtain a feature vector.

3. **Fully Connected (FC) layer**: Use the weights of the final fully connected layer corresponding to the target class to weight the pooled features.

3.3 Equation used in CAM

Class Activation Mapping (CAM) is a technique used to visualize the important regions of an image that influence a Convolutional Neural Network's (CNN) predictions for a specific class. The Class Activation Map (CAM) for class c at spatial location (x, y) is computed as

$$M_c(x, y) = \sum_k w_k^c f_k(x, y) \tag{1}$$

, where w_k^c represents the weight for the k-th channel relative to class c, and $f_k(x, y)$ denotes the activation of the k-th channel at location (x, y)
To obtain the overall class score S_c, the sum of weighted activations across all channels and spatial locations is computed:

$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x, y) = \sum_{x,y} \sum_k w_k^c f_k(x, y) \tag{2}$$

This score reflects the total importance of the class c in the image, highlighting how the CNN utilizes different regions of the image for classification.

4 Gradient-weighted Class Activation Mapping (Grad-CAM)

4.1 What is Grad-CAM?

Grad-CAM (Gradient-weighted Class Activation Mapping), introduced by Selvaraju et al. (2019), is an advanced technique for visualizing the regions of an input image that are most influential in the decision-making process of a convolutional neural network. Building upon the earlier Class Activation Mapping (CAM) approach, Grad-CAM leverages the gradients of the target class with respect to the feature maps of the final convolutional layer. By computing the gradients and pooling them, Grad-CAM generates a localization map that highlights the areas of the image that have the highest impact on the model's prediction. This method provides a more detailed and interpretable visualization of the model's focus, allowing for better understanding and validation of its decision-making process, especially in complex or high-stakes applications.

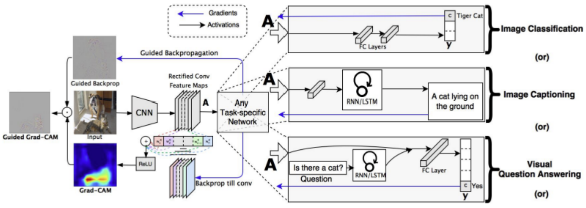


Figure 2: The Architecture of Grad-CAM from Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization

4.2 Implementation

We applied Grad-CAM by computing the gradients of the target class with respect to the feature maps from the last convolutional layer and generating the class activation map.

1. **Forward Pass**: Initially, the image is fed into the convolutional neural network (CNN), which processes it through multiple layers to yield feature maps from the last convolutional layer and the predicted class label.

2. **Cradient Compuation**: Following this, the gradients of the predicted class score are computed with respect to the feature maps of this final convolutional layer. These gradients indicate how changes in the feature maps would affect the class score.

3. **Weight Calculation**: The gradients are then aggregated by performing global average pooling to compute a weight for each feature map, reflecting its importance in the prediction.

4. **Class Activation Map**: Subsequently, these weights are used to scale the feature maps, and the weighted feature maps are summed to create a coarse class activation map. This map is then upsampled to match the original image dimensions, resulting in a visualization that highlights the critical regions of the image that influenced the model's decision.

4.3 Equation used in Grad-CAM

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k} \tag{3}$$

The weight calculation shown in the weight image involves using the gradient of the model's output y^c with respect to the k-th channel of the activation map A^k to obtain a weight score. Here, y represents the model's output, and A represents the activation map. Indices i and j correspond to the x and y axes, respectively, while Z represents the total size of the map. For i=1,2,...,u and j=1,2,...,v, Z equals u×v, which serves as the denominator for the global average pooling.

Using the equation in the weight image, we can determine the weight scores for each channel in the activation map. These scores indicate how significant each channel is for activating a particular class. Notably, there is no need to introduce separate weight parameters as in CAM.

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha^c_k A^k\right) \tag{4}$$

To visualize the class-specific activation within the activation map, we combine this information by multiplying the k-th activation map with its corresponding weight and summing the results. This sum is then passed through a ReLU activation function to produce the class-specific Grad-CAM, as shown in the Grad-CAM image's equation.

5 Experiments

We present the experiments conducted using both Class Activation Mapping (CAM) and Gradient-weighted Class Activation Mapping (Grad-CAM) to evaluate their performance and visual explainability in the context of identifying the location of dogs in images from the Stanford Dogs dataset. We used the ResNet-50 model pre-trained on ImageNet for our experiments.

5.1 Data and Model

For our experiments, we utilized the Stanford Dogs dataset available in TensorFlow Datasets, which consists of 20,580 images of 120 breeds of dogs. We employed the ResNet-50 model to generate the CAM and Grad-CAM visualizations. The ResNet-50 model is known for its robust feature extraction capabilities due to its deep architecture with residual connections.

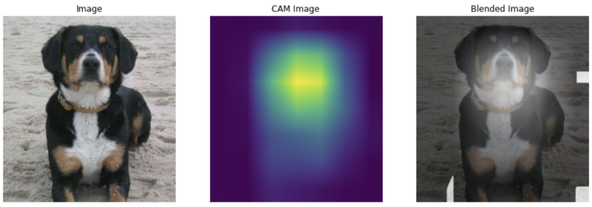


Figure 3: original, cam heatmap, and blended image

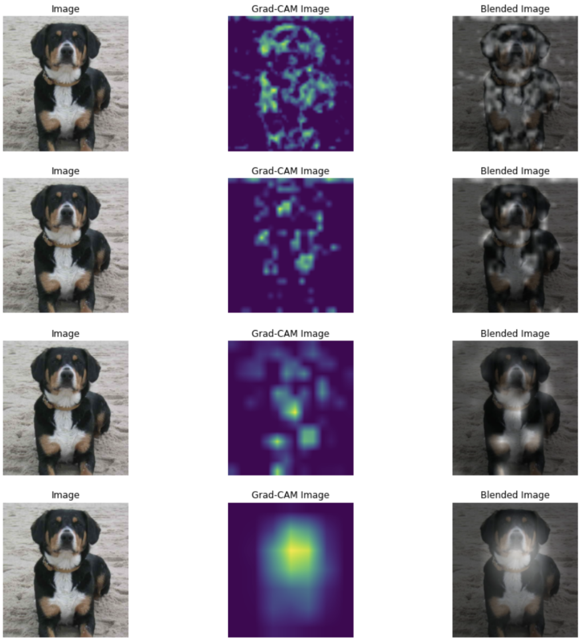


Figure 4: original, grad-cam heatmap, and blended image

5.2 Visualization of CAM and Grad-CAM

- **Fig 3**:
  - **Original Image**: The input image from the Standford Dogs dataset.
  - **CAM heatmap**: The heatmap generated using CAM, highlighting the regions of the image that the model considers important for classifying the image as a specific breed of dog.
  - **Blended Image**: A composite image created by overlaying the CAM heatmap on the original image.

- **Fig 4**:
  - **Original Image**: The sam input image used for CAM visualizations.
  - **Grad-CAM Heatmaps**: Heatmaps obtained from the last layer of each

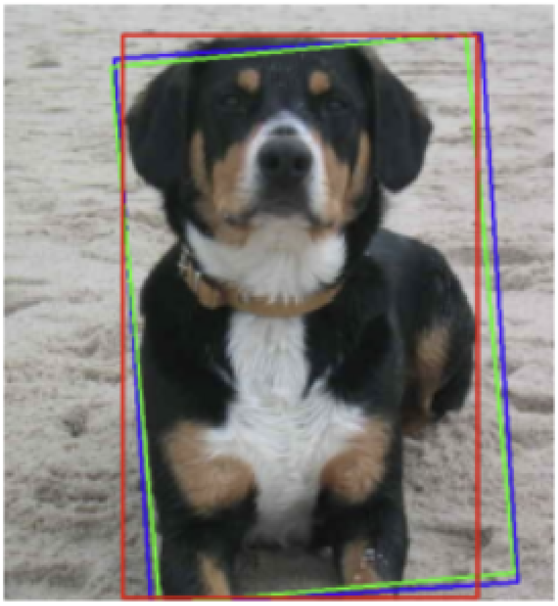


Figure 5: CAM, Grad-CAM, Original Bounding Box

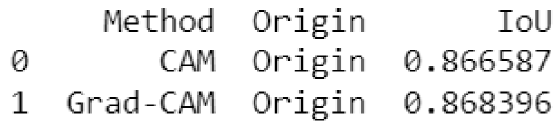
block in the ResNet-50 model. These heatmaps provide insights into how the model preocesses and extracts features at different levels of the network.

- **Blended Images**: Composite images created by overlapping each Grad-CAM heatmap on the original image.

Upon visual inspection, the heatmaps generated by CAM and the final layer of Grad-CAM appeared similar, indicating that both methods effectively highlight important regions in the image. However, Grad-CAM provided a more detailed visualization of the feature extraction proocess through heatmaps from intermediate layers, offering a better understanding of how the model constructs its feature maps.

5.3 Bounding Box Generation and Evaluation

Using the heatmaps generated by CAM and Grad-CAM, we created bouding boxes to highlight the location of dogs in the images. The bounding boxes were generated based on the regions with the highest activation values in the heatmaps. These bounding boxes were then compared to the ground truth bouding boxes provided in the Stanfor Dogs dataset.



```
     Method  Origin       IoU
0       CAM  Origin  0.866587
1  Grad-CAM  Origin  0.868396
```

Figure 6: IoU of CAM and Original Box and IoU of Grad-CAM and Original Box

In the Fig 5, we will illustrate a smaple image with the following bouding boxes:

- **Blue Bounding Box**: Generated using the CAM heatmap.
- **Green Bounding Box**: Generated using the Grad-CAM heatmap
- **Red Boudning Box**: The original ground truth bouding box from the dataset.

To evaluate the accuracy of the bouding boxes, we used the Intersection over Union (IoU) metric. IoU measures teh overlap between the predicted bouding box and the ground truth bouding box, providing a quantitative assessment of localization accuracy.

According to Fig 6,

- **IoU for CAM**:The IoU score for the bounding boxes generated using CAM was 0.866587.
- **IoU for Grad-CAM**: The IoU score for the bounding boxes generated using Grad-CAM was 0.868396.

These results indicate that while both CAM and Grad-CAM effectively localize the regions of interest, Grad-CAM achieved a slightly higher IoU score, suggesting a marginally better performance in terms of bounding box accuracy.

6 Conclusion

Our experiments demonstrate that both CAM and Grad-CAM are effective tools for visualizing and understanding the decision-making process of convolutional neural networks. Grad-CAM, in particular, provides additional insights through intermediate layer visualizations, making it a valuable method for detailed feature extraction analysis. The IoU evaluation further supports the efficacy of Grad-CAM, showing a slight improvement over CAM in accurately localizing the regions of interest.

References

Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., and Batra, D. 2019, December 3 *Grad-cam: Visual explanations from deep networks via gradient-based localization*, arXiv.org.

Simonyan, K., Vedaldi, A.,and Zisserman, A. 2014, April 19 *Deep inside convolutional networks: Visualising Image Classification Models and saliency maps*, arXiv.org.

Springenberg, J.T., Dosovitskiy,A., Brox, T., and Riedmiller, M. 2015 April 13 *Striving for Simplicity: The All Convolutional Net*, arXiv.org.

Zeiler, M. D.and Fergus, R. 2013, November 28 *Visualizing and Understanding Convolutional Networks*, arXiv.org.

Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., and Torralba, A. 2015 December 14 *Learning Deep Features for Discriminative Localization*, arXiv.org.