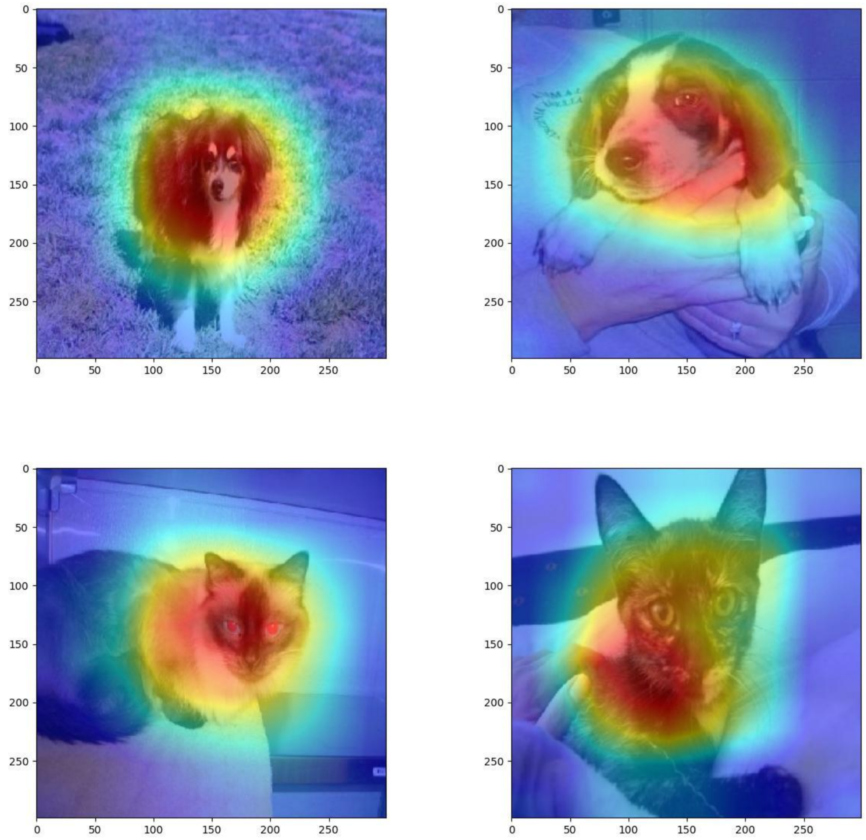
GradCAM

Rationale, Strengths, and Weaknesses

What is Grad-CAM?

Grad-CAM (Gradient-weighted Class Activation Mapping) is a **visual explanation method** that highlights regions in an input image that were important for a CNN’s prediction. It does so by **computing the gradients of a target class score with respect to the feature maps** of a convolutional layer, and then using these gradients to weigh the feature maps, producing a **class-specific heatmap**.



Strengths of Grad-CAM

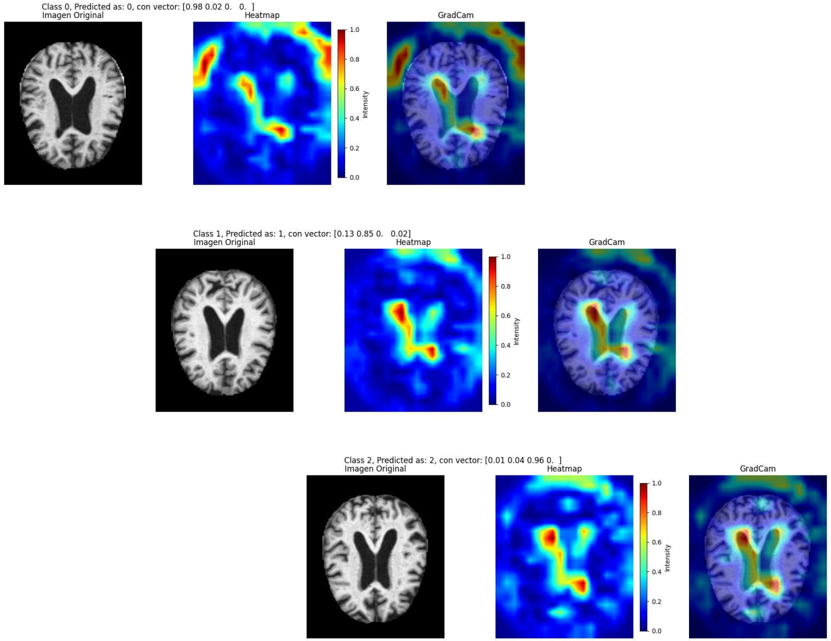
- **Model-agnostic for CNN-based architectures**: Works with various CNN models without architectural changes.

- **Class-specific explanations**: Shows *why* the model predicted a particular class.

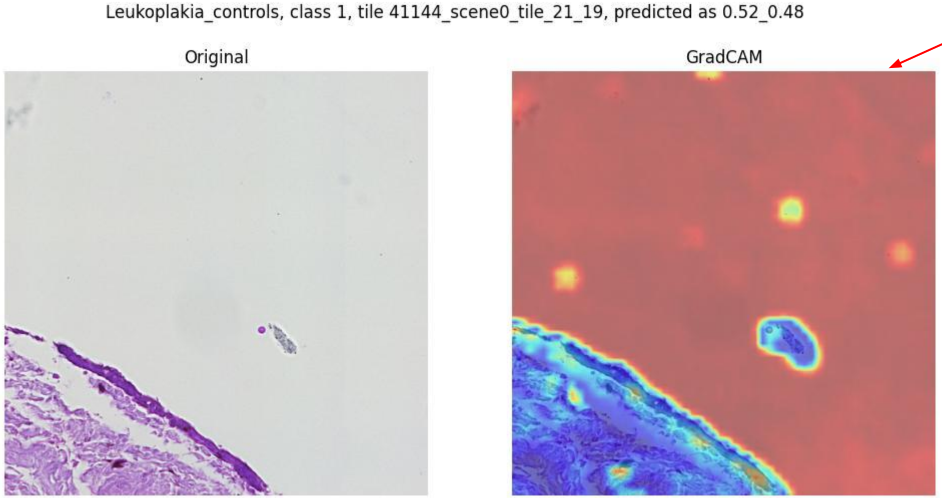
- **Applicable to a wide range of tasks**: Classification, image captioning, object detection, etc.

- **Interpretable**: Produces human-understandable heatmaps that highlight salient regions.

- **Preserves spatial information**: Unlike saliency maps that may be noisy and hard to interpret.



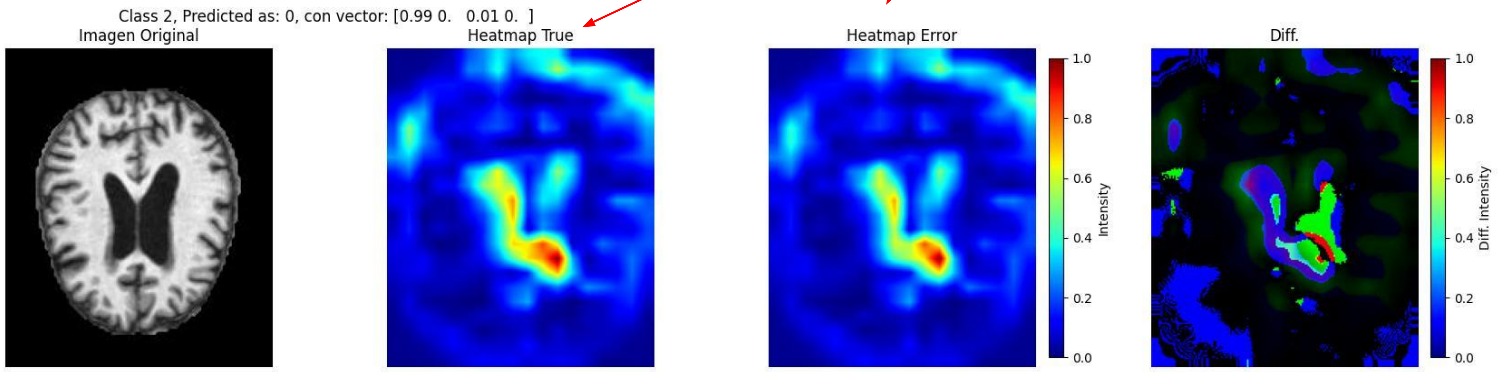
Weaknesses of Grad-CAM



- **Not pixel-precise**: The heatmap is based on coarse, low-resolution convolutional layers, so it lacks fine-grained details.
- **Biased towards high-activation regions**: Can overemphasize areas that are bright or textured, even if not semantically important.
- **Class dependence**: Heatmaps are generated for a specific class, so additional computation is needed for multi-class explanations.
- **Only positive influence**: Due to ReLU, it ignores negative contributions (which could be important for understanding why a class is not chosen).

Weaknesses of Grad-CAM

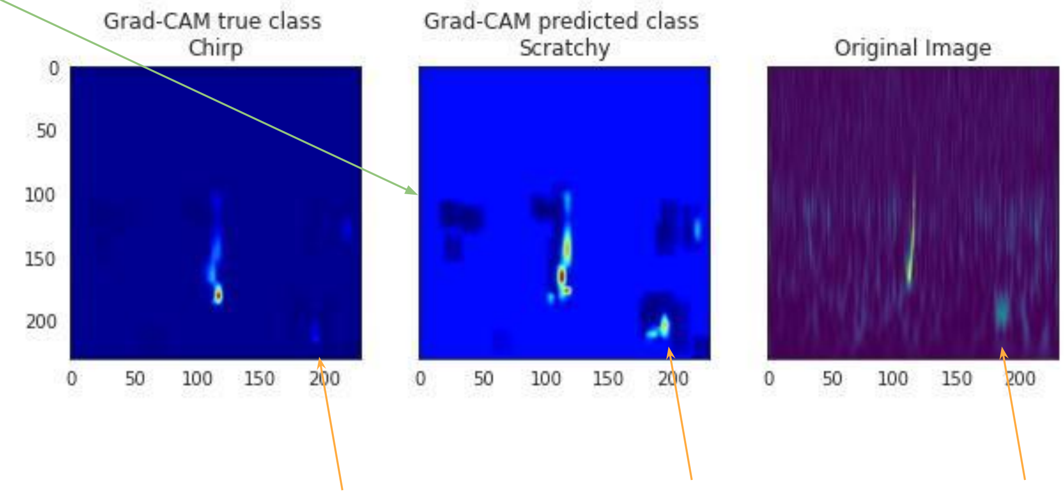
- **Class dependence**: Heatmaps are generated for a specific class, so additional computation is needed for multi-class explanations.



Weaknesses of Grad-CAM

- **Class dependence**: Heatmaps are generated for a specific class, so additional computation is needed for multi-class explanations.

- **Class dependence**: However this allows understand why the image is misclassified, why the feature map for the predicted class is stronger than for the true one.

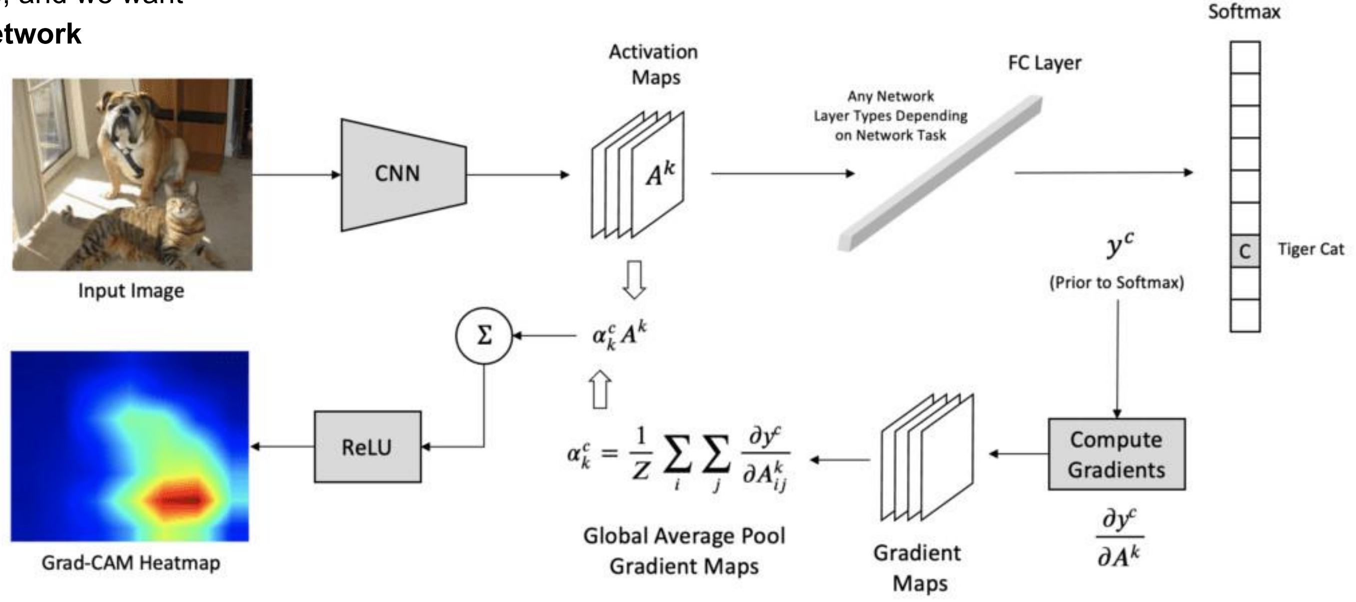


Comparison: Grad-CAM vs CAM (Class Activation Mapping)

Feature	**CAM**	**Grad-CAM**
Requires architecture modification	Yes (requires GAP layer and linear output layer)	No
Works with pre-trained models	No	Yes
Generalizable to different architectures	Limited	Yes
Explanation precision	Moderate	Moderate
Class-specific heatmaps	Yes	Yes
How weights are computed	From trained linear classifier	From gradient of class score w.r.t. feature maps
Flexibility	Low	High

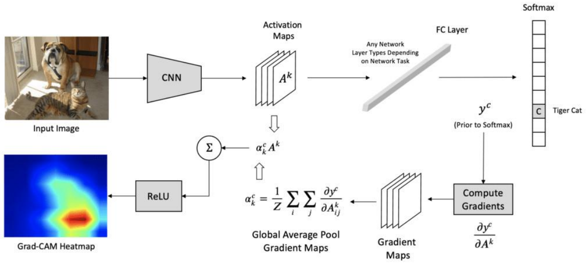
How Grad-CAM Works (Step-by-Step)

Let's suppose we have a trained CNN model and an input image, and we want to understand **why the network predicted class *c***.



How Grad-CAM Works (Step-by-Step)

Let's suppose we have a trained CNN model and an input image, and we want to understand **why the network predicted class *c***.



1. Forward Pass:

- Pass the input image through the CNN.
- Extract the **feature maps** *A* from the last convolutional layer. These maps capture high-level spatial information:

$$A \in \mathbb{R}^{H \times W \times K}$$

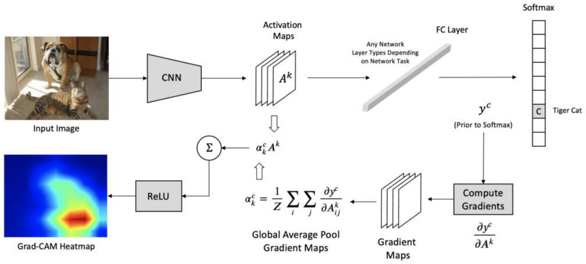
2. Compute Gradients:

- Compute the gradient of the **score for class c** (before the softmax) with respect to each feature map:

$$\frac{\partial y^c}{\partial A^k}$$

How Grad-CAM Works (Step-by-Step)

Let's suppose we have a trained CNN model and an input image, and we want to understand **why the network predicted class *c***.



3. Global Average Pooling of Gradients:

- For each feature map, average the gradient values over all spatial locations to obtain the **importance weight**:

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

where Z is the total number of pixels in the feature map.

4. Compute the Weighted Sum:

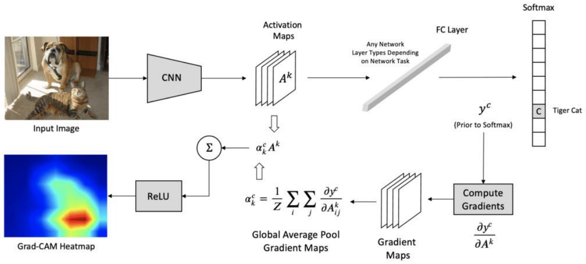
- Compute a **weighted sum** of the feature maps using these importance weights:

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\left(\sum_k \alpha_k^c A^k\right)$$

- The **ReLU** ensures that only features that have a **positive influence** on the class score are retained, focusing on supportive features.

How Grad-CAM Works (Step-by-Step)

Let's suppose we have a trained CNN model and an input image, and we want to understand **why the network predicted class *c***.



5. Upsample the Heatmap:

- Resize the resulting heatmap to the same dimensions as the input image using bilinear interpolation.

6. Overlay:

- Overlay the heatmap on the original image to visualize the regions most relevant to the class prediction.

Summary

GradCAM

Gradient-weighted Class Activation Mapping

Core

Compute the gradients of a target class score with respect to the feature maps of a convolutional layer, and then using these gradients to weigh the feature maps, producing a class-specific heatmap.

Advantages

Model-agnostic for CNN-based architectures, per class-specific explanations, applicable to a wide range of tasks, interpretable, preserves spatial information.

Limitations

Not pixel-precise, biased towards high-activation regions, class dependence, only positive influence.