Practicum in Neuroimage Analysis

BIOENG 2195
Spring 2025

Unveiling the Black Box: Explainable AI in Medical Imaging

Ganesh Puthiaraju

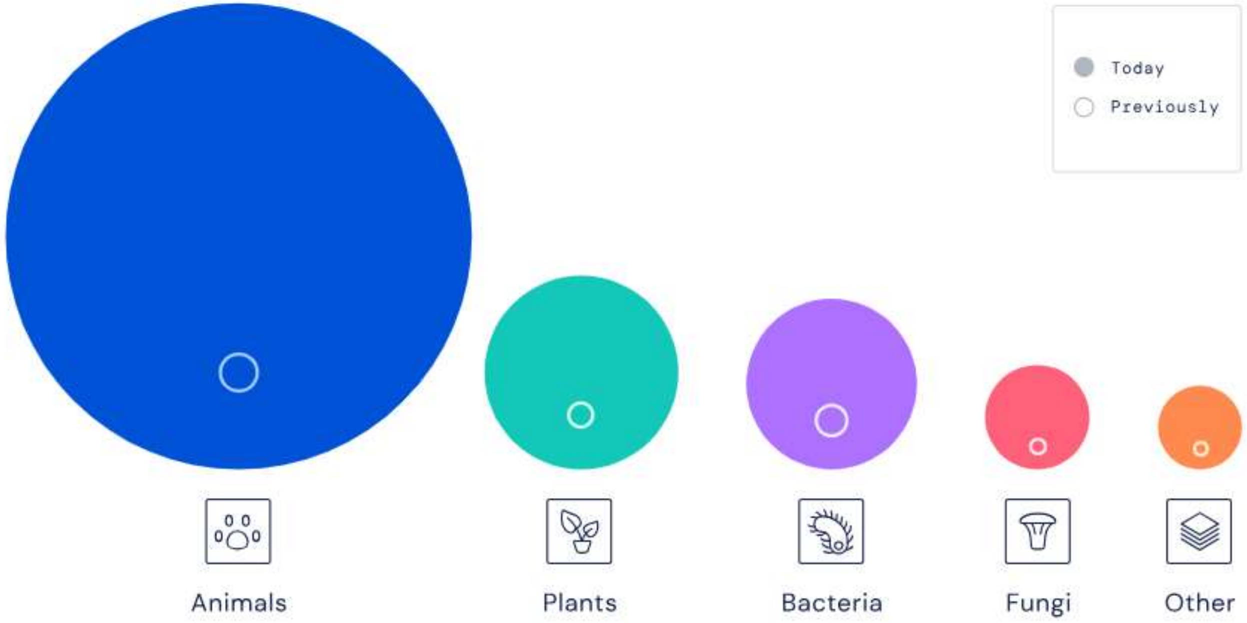
A Glimpse of AI Achievements in Recent Years

Protein Structure Prediction

- ***AlphaFold2 (2020)*** predicted protein structures with atomic-level accuracy, solving a 50-year grand challenge

Number of species represented in AlphaFold DB

Total increase from ~10K to ~1M



[Reference: Jumper, J., et al. (2021). Highly accurate protein structure prediction with AlphaFold. *Nature*, *596*(7873), 583-589.]

A Glimpse of AI Achievements in Recent Years

Conversational and Generative AI

Large Language Models (ChatGPT o1, Claude Sonnet 3.7, Google Gemini Pro 2.0): Demonstrating advanced reasoning and conversational abilities.

Generative AI (DALL-E 2, Stable Diffusion): Creating realistic images from text descriptions.

- Sora

[Reference: Ramesh, A., et al. (2022). Hierarchical text-conditional image generation with CLIP latents. arXiv preprint arXiv:2204.06125.]

A Glimpse of AI Achievements in Recent Years

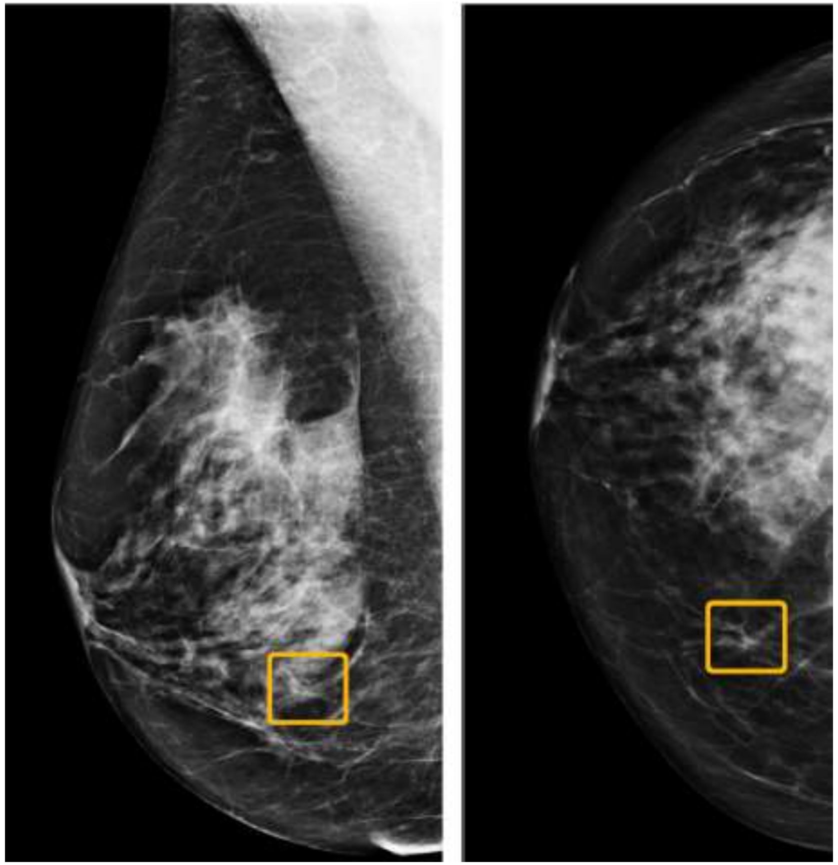
Autonomous Systems

Self-driving cars (Waymo, Tesla prototypes) navigate safely in test environments due to advances in perception and planning.

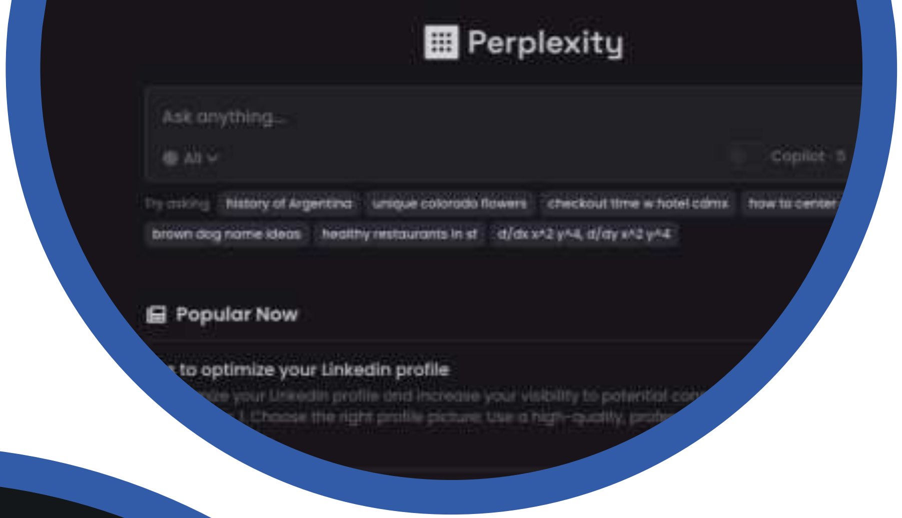
[Reference: End-to-End Deep Learning for Self-Driving Cars by Bojarski et al. from NVIDIA Research (2016)]

A Glimpse of AI Achievements in Medical Imaging

- **Expert-Level Diagnosis:** CNNs and Transformers detect conditions like diabetic retinopathy and lung nodules with radiologist-level accuracy
- **Image Segmentation:** Automated delineation of organs and tumors improving surgical planning and treatment
- **Computer-Aided Detection:** AI screening systems for mammograms and X-rays reduce missed abnormalities
- **Prognostic Models:** Networks combining imaging with clinical data predict disease progression trajectories
- **Workflow Optimization:** AI triage prioritizes urgent cases, accelerating treatment for critical patients



[Reference: McKinney SM, Sieniek M, Godbole V, et al. International evaluation of an AI system for breast cancer screening. Nature. 2020;577(7788):89-94.]



Try asking history of Argentina unique colorado flowers checkout time w hotel cdmx how to center brown dog name ideas healthy restaurants in sf d/dx x^2 y^4, d/dy x^2 y^4

Popular Now

to optimize your Linkedin profile

ize your Linkedin profile and increase your visibility to potential con

1. Choose the right profile picture: Use a high-quality, profe

using all the same too

ChatGPT: Optimizing Language Models for Dialogue

trained a model called ChatGPT which in a conversational way. The dialogue it possible for ChatGPT to uestions, admit its rect premises, and

Find videos of how to quickly get grape juice out of a wool rug

How long will it take to walk from Times Square to Central Park?

incorporating plant-based options in diet

AI works ...

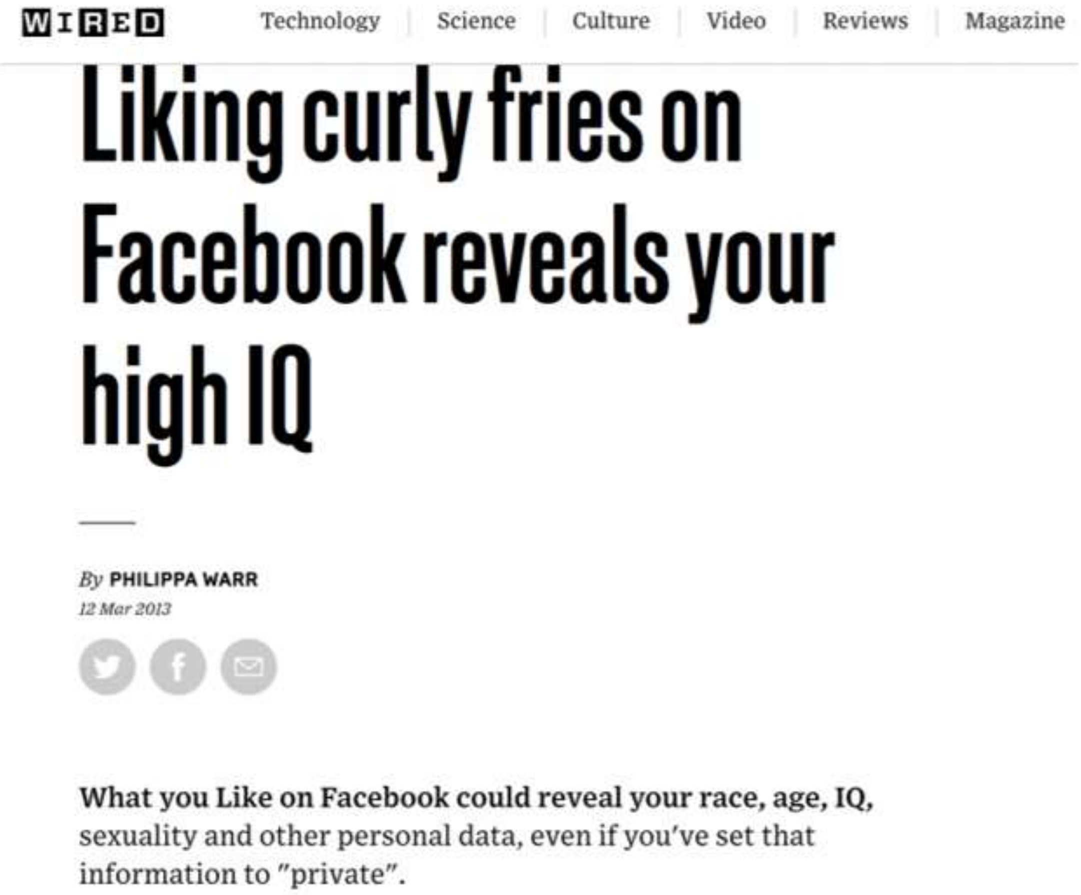
nini
Good morning
Find videos of how to quickly get grape juice out of a wool rug
How long will it take to walk from Times Square to Central Park?
Chats

AI works. But is it reliable, interpretable, and robust ?

openai.com
OpenAI
API
RESEARCH
BLOG
ABOUT
ChatGPT: Optimizing Language Models for Dialogue
We've trained a model called ChatGPT which interacts in a conversational way. The dialogue format makes it possible for ChatGPT to answer followup questions, admit its mistakes, challenge incorrect premises, and

XAI: Explainability in AI

We do not want models to fail when they get in the real world.



Liking curly fries on Facebook reveals your high IQ

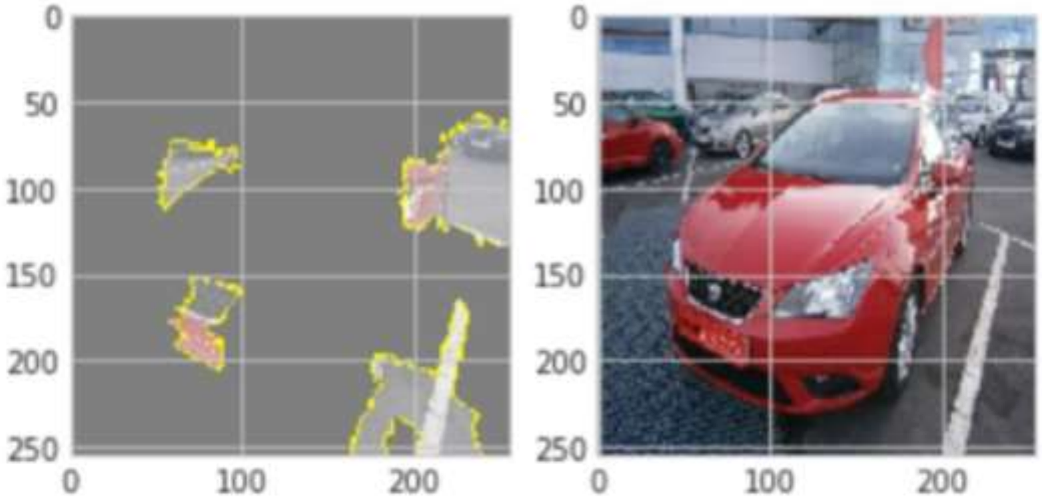
By PHILIPPA WARR

12 Mar 2013

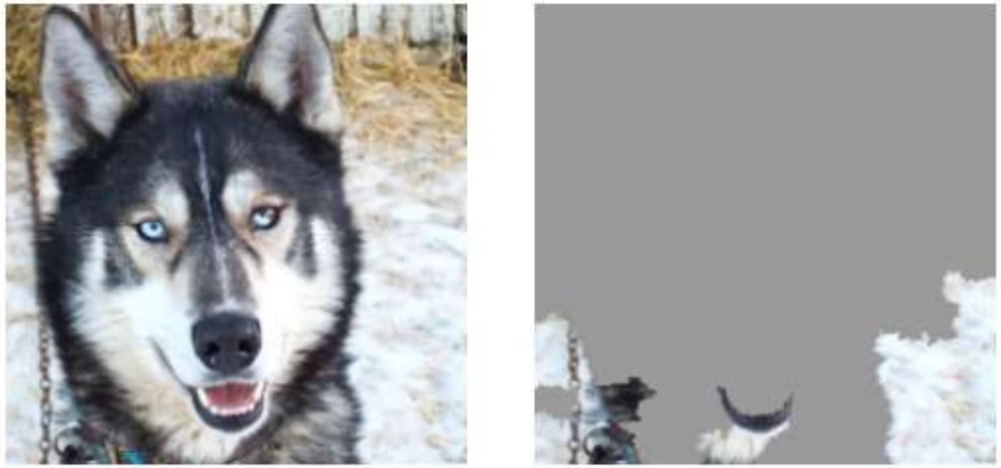
What you Like on Facebook could reveal your race, age, IQ, sexuality and other personal data, even if you've set that information to "private".

XAI: Explainability in AI

Without explainability, we might deploy models that make decisions for the wrong reasons

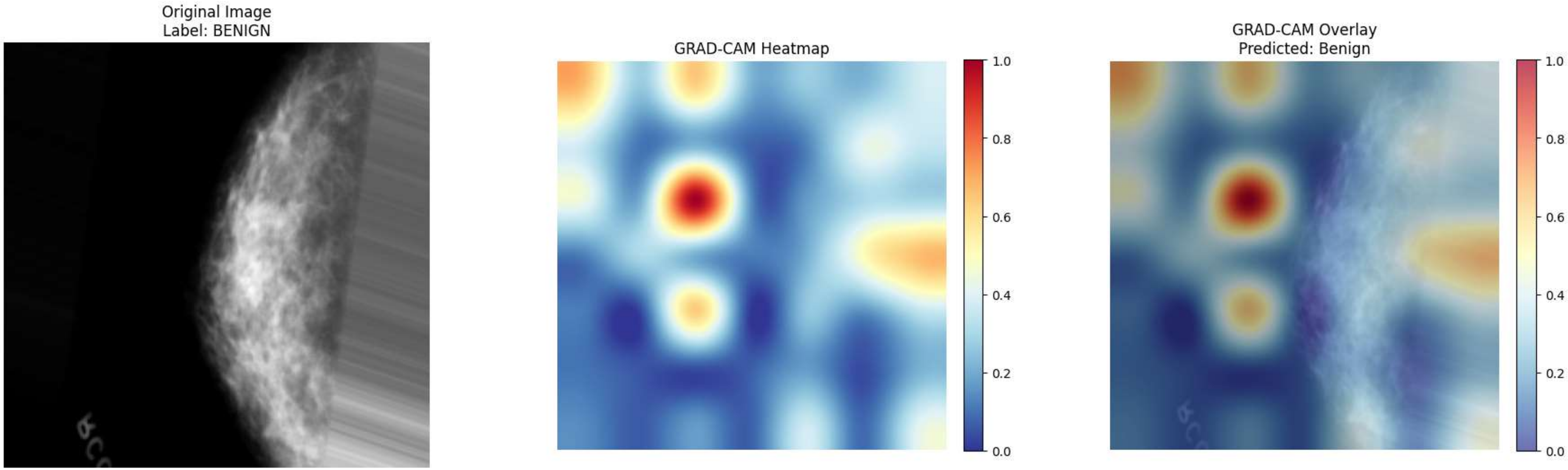


Explanation Failure: Model classifies vehicle based on background context rather than car features



Misleading Correlation: Husky misclassified as wolf based on snowy background rather than animal features

XAI: Explainability in AI



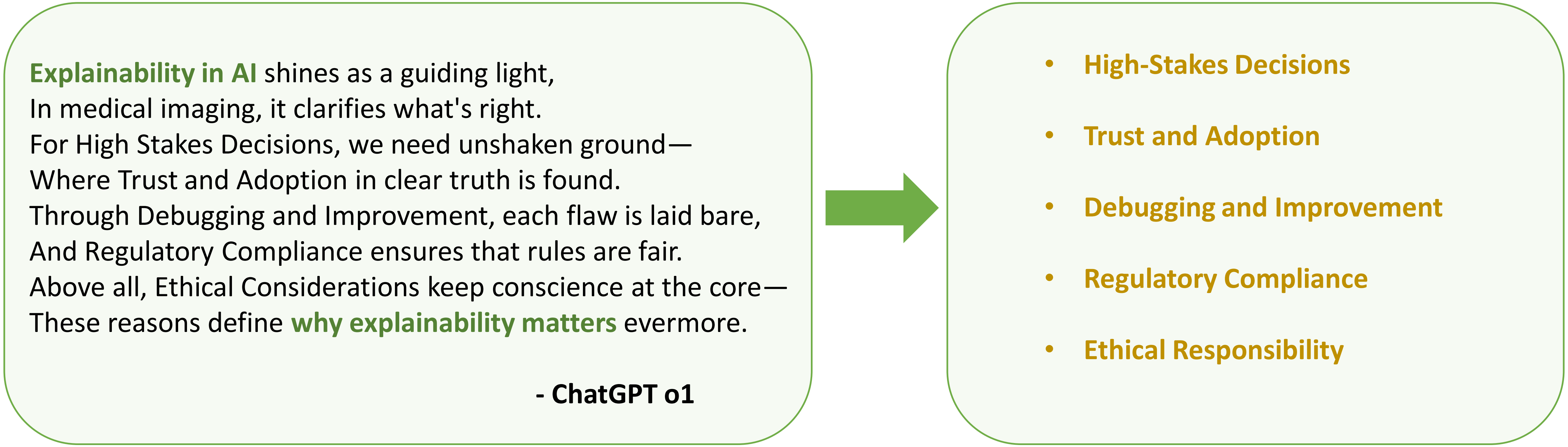
Model classifies mammogram as benign by focusing on image background rather than breast tissue features

Why Explainability Matters?

Explainability in AI shines as a guiding light,
In medical imaging, it clarifies what's right.
For High Stakes Decisions, we need unshaken ground—
Where Trust and Adoption in clear truth is found.
Through Debugging and Improvement, each flaw is laid bare,
And Regulatory Compliance ensures that rules are fair.
Above all, Ethical Considerations keep conscience at the core—
These reasons define **why explainability matters** evermore.

- **High-Stakes Decisions**
- **Trust and Adoption**
- **Debugging and Improvement**
- **Regulatory Compliance**
- **Ethical Responsibility**

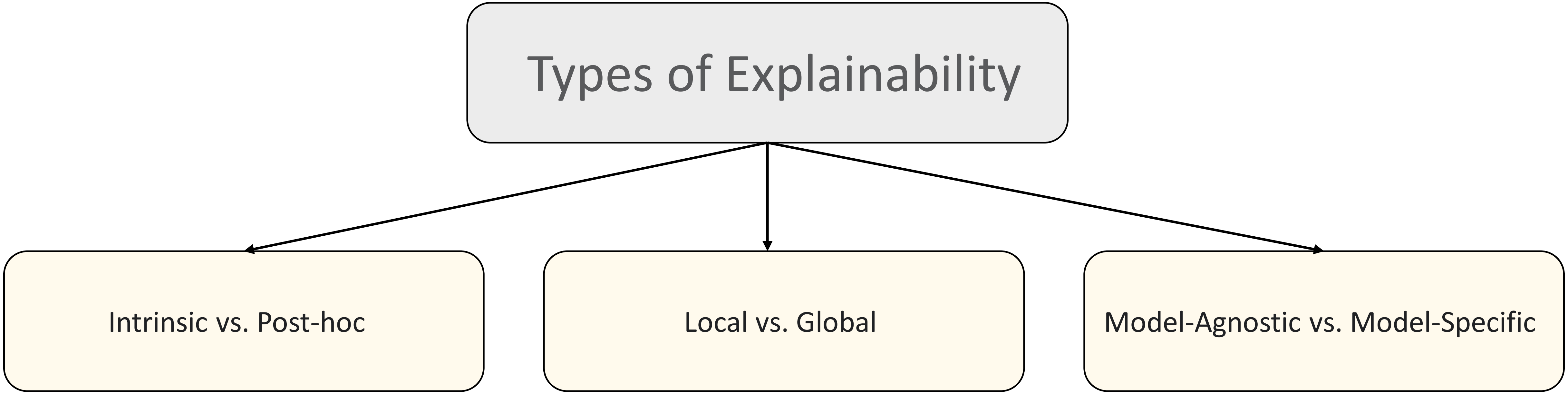
Why Explainability Matters?



Explainability in AI shines as a guiding light,
In medical imaging, it clarifies what's right.
For High Stakes Decisions, we need unshaken ground—
Where Trust and Adoption in clear truth is found.
Through Debugging and Improvement, each flaw is laid bare,
And Regulatory Compliance ensures that rules are fair.
Above all, Ethical Considerations keep conscience at the core—
These reasons define **why explainability matters** evermore.

- ChatGPT o1

- **High-Stakes Decisions**
- **Trust and Adoption**
- **Debugging and Improvement**
- **Regulatory Compliance**
- **Ethical Responsibility**



Types of Explainability

Intrinsic vs. Post-hoc

Local vs. Global

Model-Agnostic vs. Model-Specific

Intrinsic vs. Post-hoc

- **Intrinsic:** Models designed to be inherently interpretable (e.g., decision trees, linear models).
- **Post-hoc:** Techniques applied *after* a model is trained to explain its predictions.

Local vs. Global

- **Local:** Explaining individual predictions (e.g., "Why did the model classify this image as malignant?").

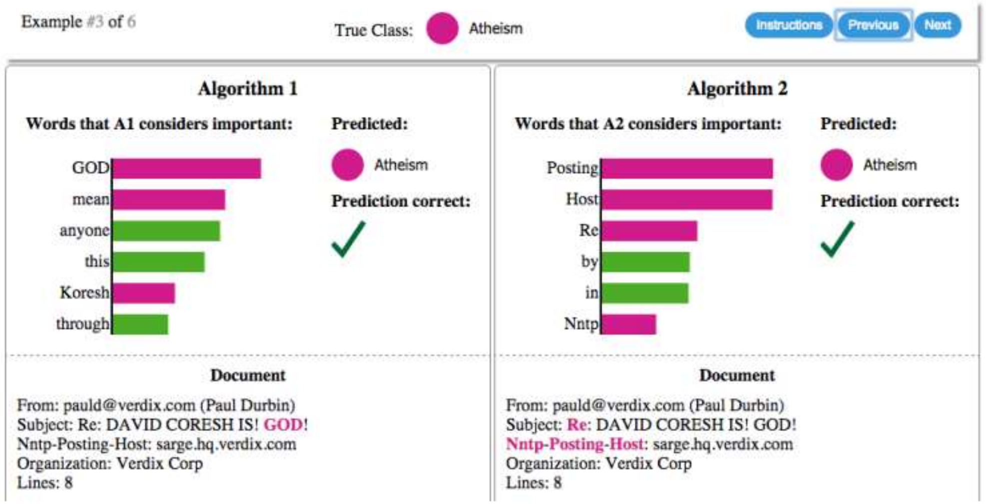
- **Global:** Understanding the overall behavior of the model (e.g., "What features are most important overall?").

Model-Agnostic vs. Model-Specific

- **Model-Agnostic:** Methods that can be applied to any model (e.g., LIME, SHAP).
- **Model-Specific:** Methods designed for specific model architectures (e.g., Grad-CAM for CNNs).

Explaining Predictions with LIME

Local Interpretable Model-Agnostic Explanations – is a technique introduced by to explain individual predictions of any classifier. The core idea is to **approximate the model locally** with a simpler interpretable model. Think of it as "**peeking inside the black box, one prediction at a time**" by asking "what happens if we tweak this input slightly?"



Example #3 of 6

True Class: ● Atheism

Algorithm 1

Words that A1 considers important:

Predicted: ● Atheism

Prediction correct: ✓

Document

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8

Algorithm 2

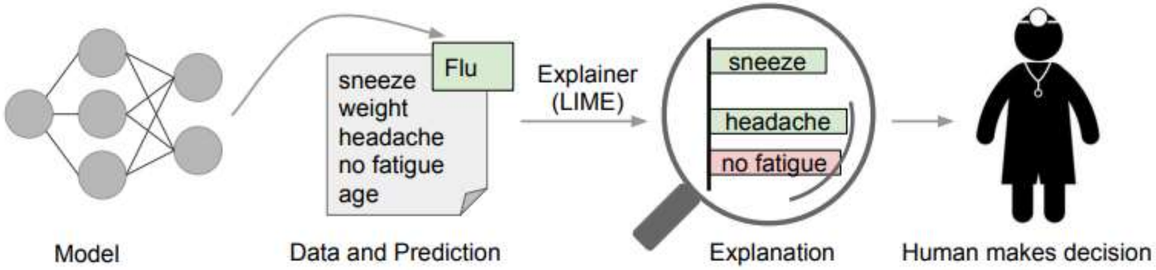
Words that A2 considers important:

Predicted: ● Atheism

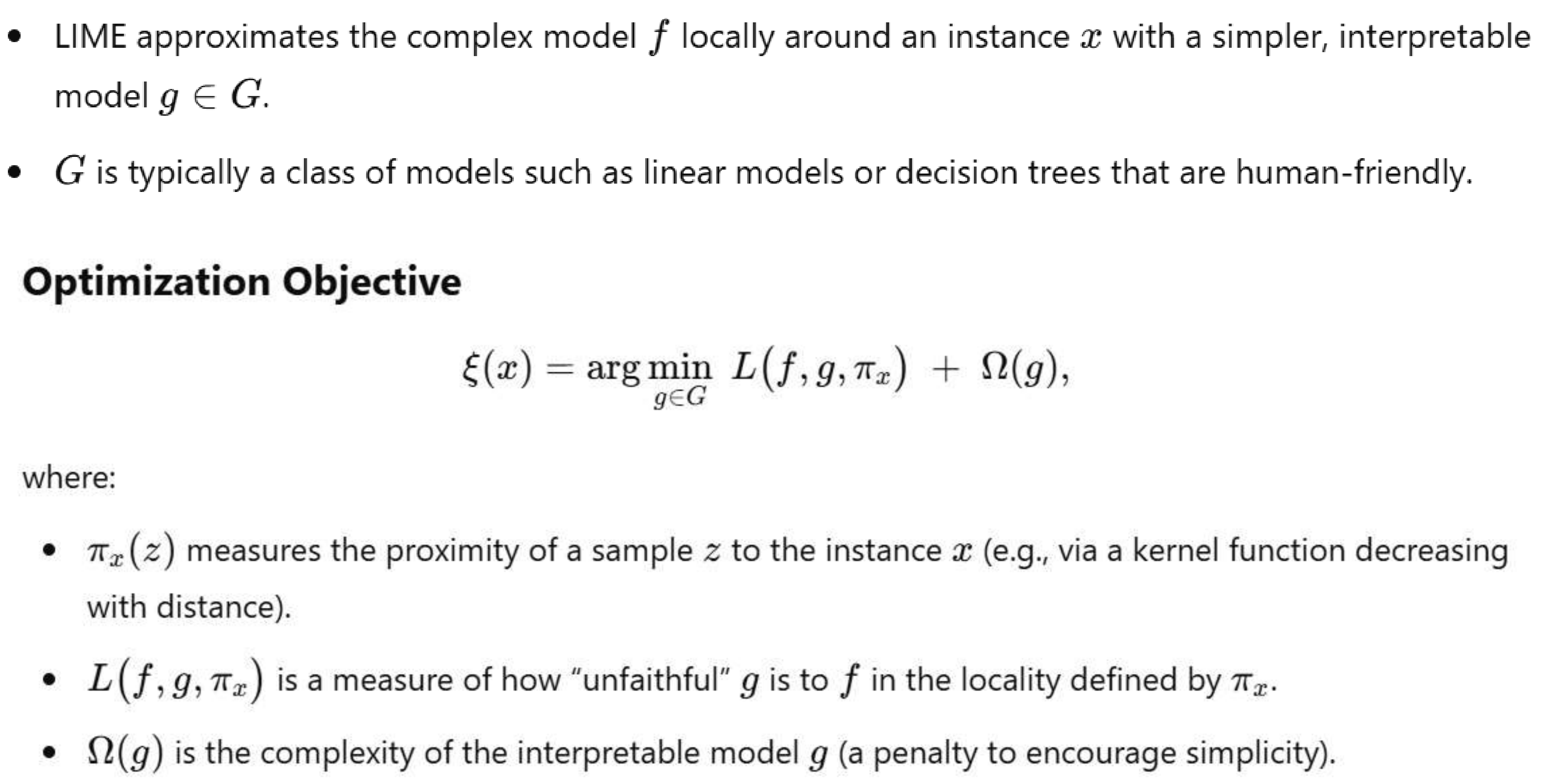
Prediction correct: ✓

Document

From: pauld@verdix.com (Paul Durbin)
Subject: Re: DAVID CORESH IS! GOD!
Nntp-Posting-Host: sarge.hq.verdix.com
Organization: Verdix Corp
Lines: 8



Explaining Predictions with LIME



- LIME approximates the complex model f locally around an instance x with a simpler, interpretable model $g \in G$.
- G is typically a class of models such as linear models or decision trees that are human-friendly.

Optimization Objective

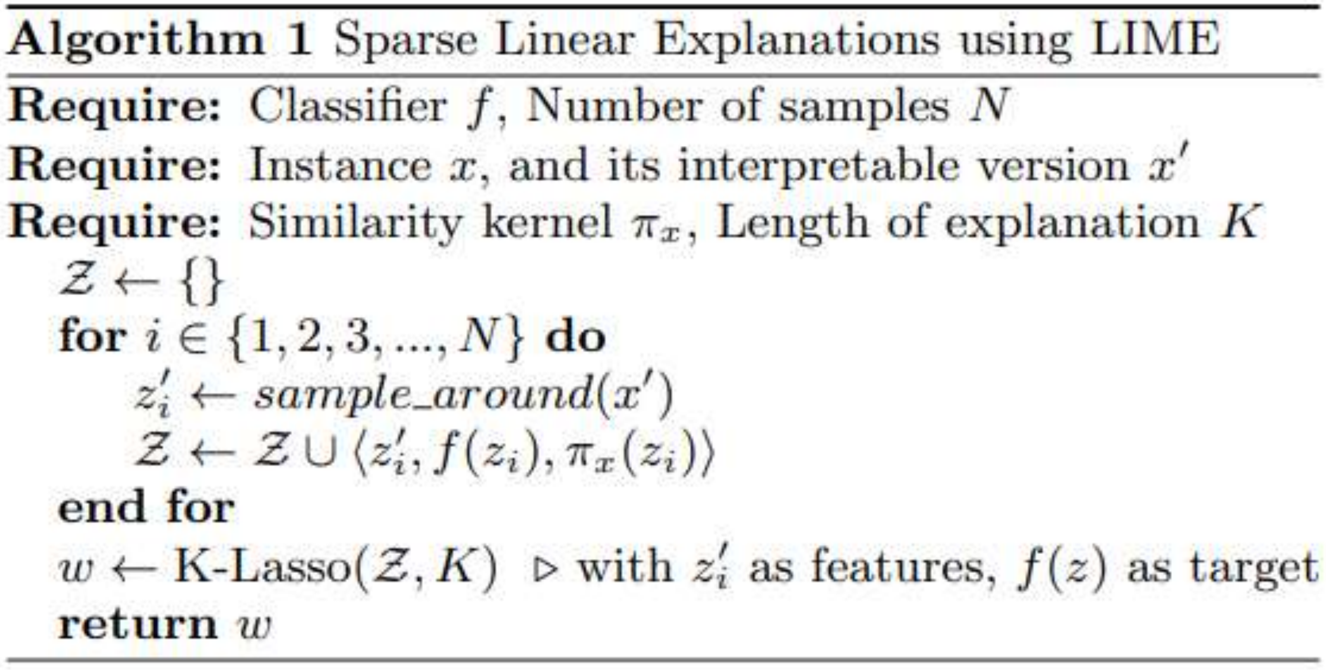
$$\xi(x) = \arg\min_{g \in G} \; L(f, g, \pi_x) \; + \; \Omega(g),$$

where:

- $\pi_x(z)$ measures the proximity of a sample z to the instance x (e.g., via a kernel function decreasing with distance).
- $L(f, g, \pi_x)$ is a measure of how "unfaithful" g is to f in the locality defined by π_x.
- $\Omega(g)$ is the complexity of the interpretable model g (a penalty to encourage simplicity).

LIME: Explaining Predictions with Local Linear Approximations

- **Perturb**
- **Predict**
- **Weight**
- **Fit**
- **Explain**

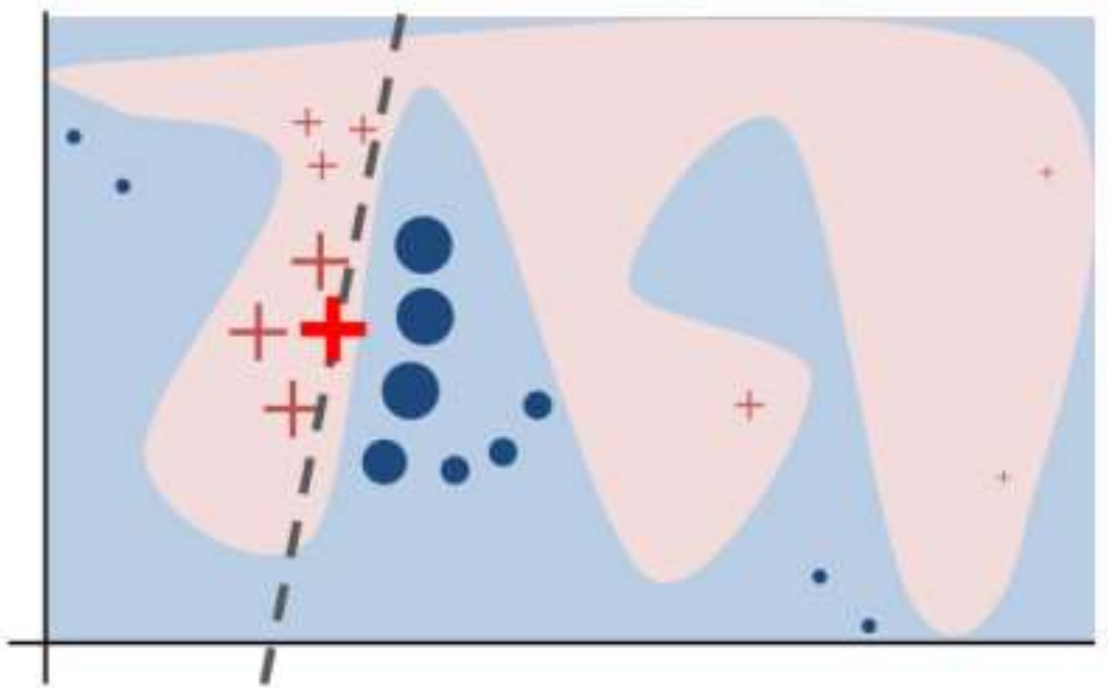


```
**Algorithm 1** Sparse Linear Explanations using LIME
**Require:** Classifier $f$, Number of samples $N$
**Require:** Instance $x$, and its interpretable version $x'$
**Require:** Similarity kernel $\pi_x$, Length of explanation $K$
  $\mathcal{Z} \leftarrow \{\}$
  **for** $i \in \{1, 2, 3, ..., N\}$ **do**
      $z'_i \leftarrow sample\_around(x')$
      $\mathcal{Z} \leftarrow \mathcal{Z} \cup \langle z'_i, f(z_i), \pi_x(z_i) \rangle$
  **end for**
  $w \leftarrow$ K-Lasso($\mathcal{Z}, K$) ▷ with $z'_i$ as features, $f(z)$ as target
  **return** $w$
```

[Reference: Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 1135-1144.]

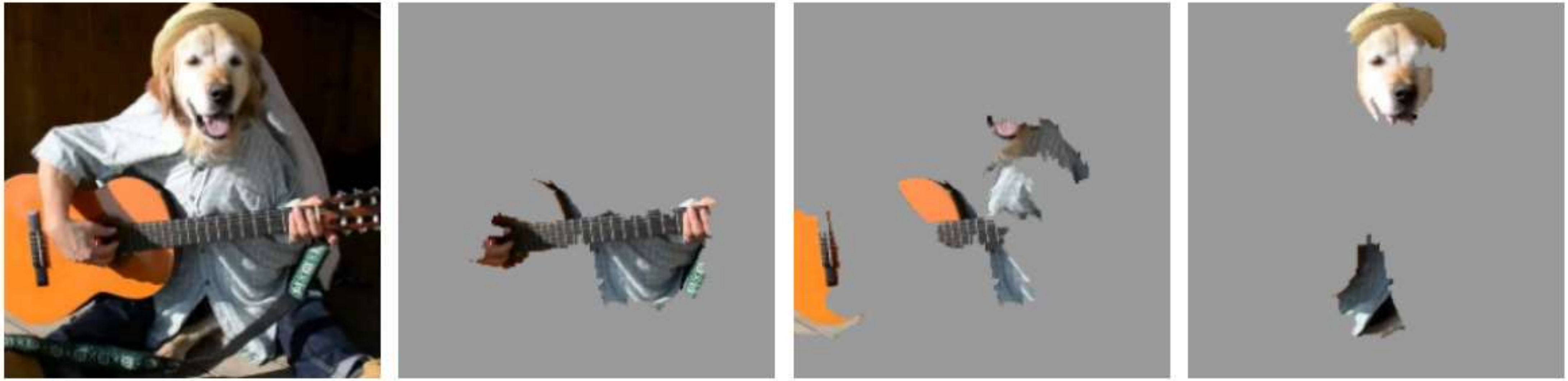
LIME: Explaining Predictions with Local Linear Approximations

- **Perturb**
- **Predict**
- **Weight**
- **Fit**
- **Explain**



Explaining Predictions with LIME

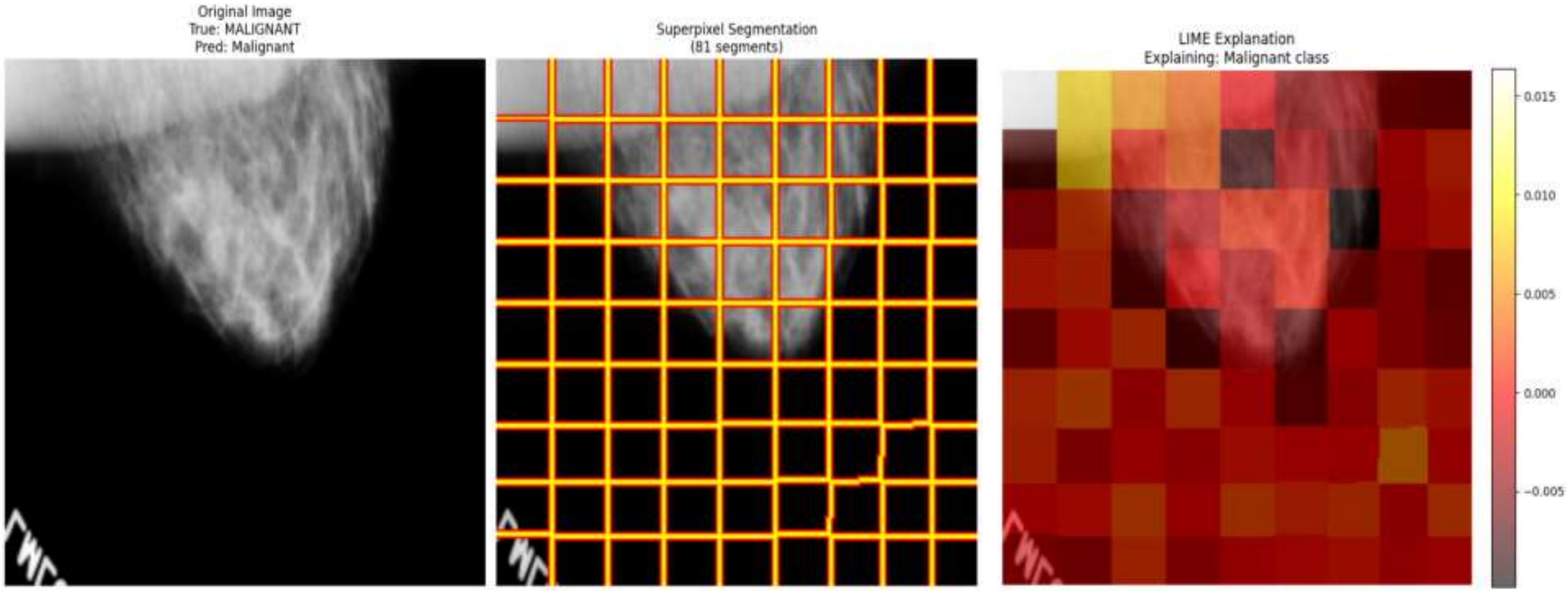
Superpixels: Groups of similar, adjacent pixels forming coherent image regions (e.g., tissue areas in medical scans)



(a) Original Image (b) Explaining *Electric guitar* (c) Explaining *Acoustic guitar* (d) Explaining *Labrador*

Explaining Predictions with LIME

Superpixels: Groups of similar, adjacent pixels forming coherent image regions (e.g., tissue areas in medical scans)



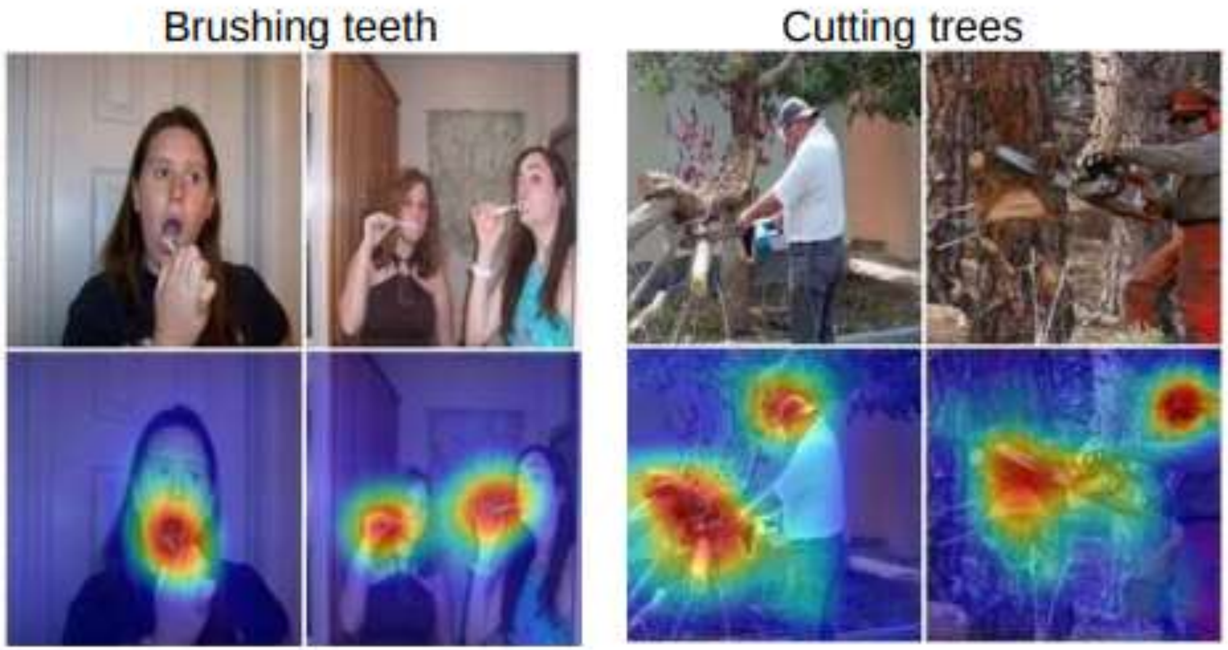
LIME: Pros and Cons

- **Advantages:**
  - **Model-agnostic:** Works with any black-box model.
  - **Local explanations:** Provides insights into individual predictions.
  - **Relatively easy** to understand and implement.

- **Limitations:**
  - **Instability:** Explanations can vary depending on the perturbation strategy and the choice of the local model.
  - **Linearity Assumption:** May not accurately capture complex, non-linear relationships.
  - **Defining "Locality":** Choosing the right neighborhood size can be challenging.
  - **Computational Cost:** Can be computationally expensive for high-dimensional data.

Class Activation Map (CAM)

A Class Activation Map for a category highlights *discriminative image regions* used by a CNN to identify that category.



[Reference: B. Zhou, A. Khosla, A. Lapedriza, A. Oliva, and A. Torralba, Learning Deep Features for Discriminative Localization, CVPR 2016.]

Class Activation Map (CAM)

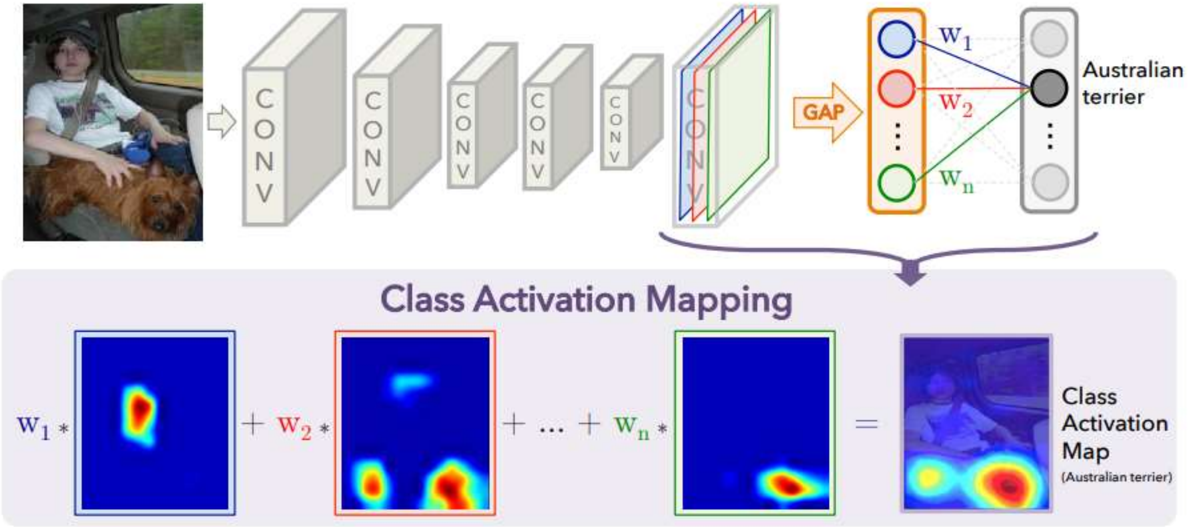
Notation

- Let $f_k(x, y)$ be the activation of feature map k at spatial location (x, y) in the last convolutional layer.
- Perform **global average pooling** (GAP) to obtain

$$F^k = \sum_{x,y} f_k(x, y).$$

- For class c, the input to the softmax is

$$S_c = \sum_k w_k^c F^k,$$



where w_k^c are the weights from the final (fully connected) layer for class c.

Class Activation Map (CAM)

Forming the Class Activation Map

- Substitute $F^k = \sum_{x,y} f_k(x,y)$ into S_c:

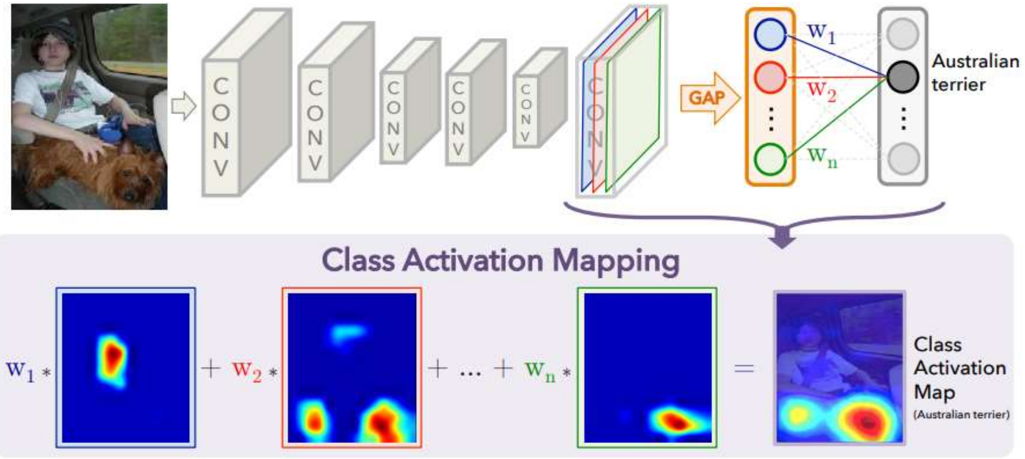
$$S_c = \sum_k w_k^c \sum_{x,y} f_k(x,y) = \sum_{x,y} \Big(\sum_k w_k^c \, f_k(x,y) \Big).$$

- Define the CAM for class c at each spatial location (x,y) as

$$M_c(x,y) = \sum_k w_k^c \, f_k(x,y).$$

- Then the class score S_c is simply

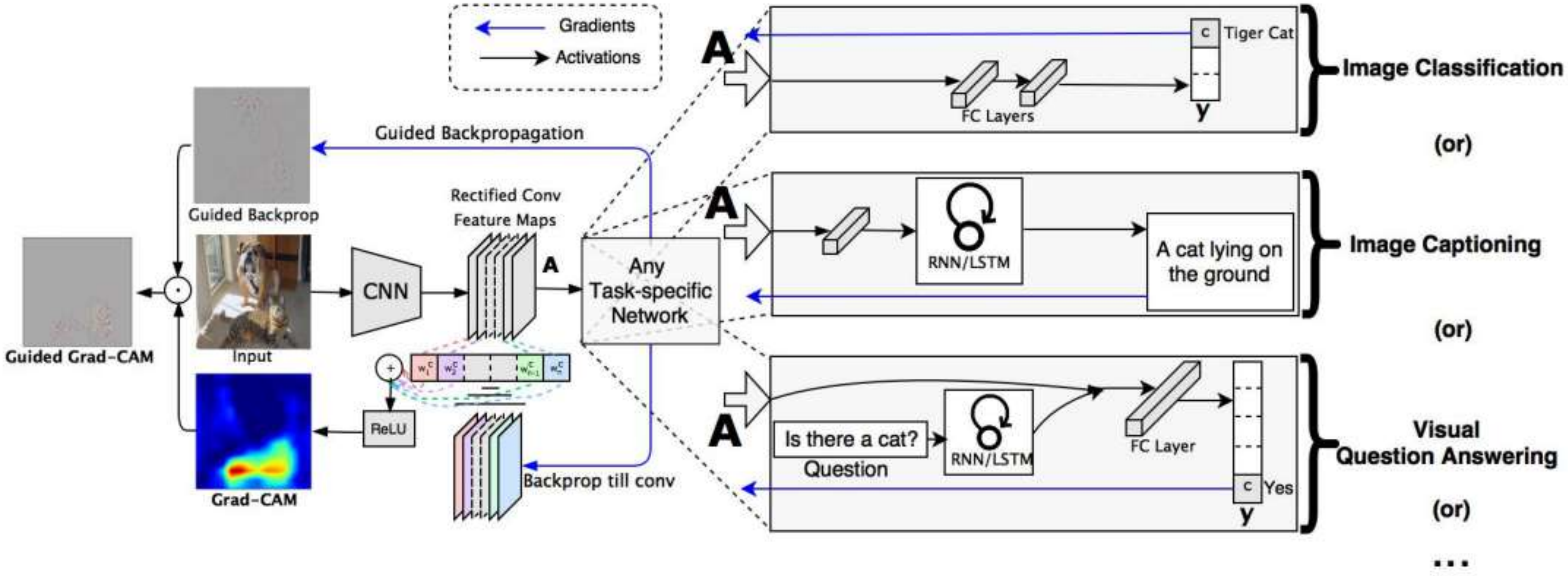
$$S_c = \sum_{x,y} M_c(x,y).$$



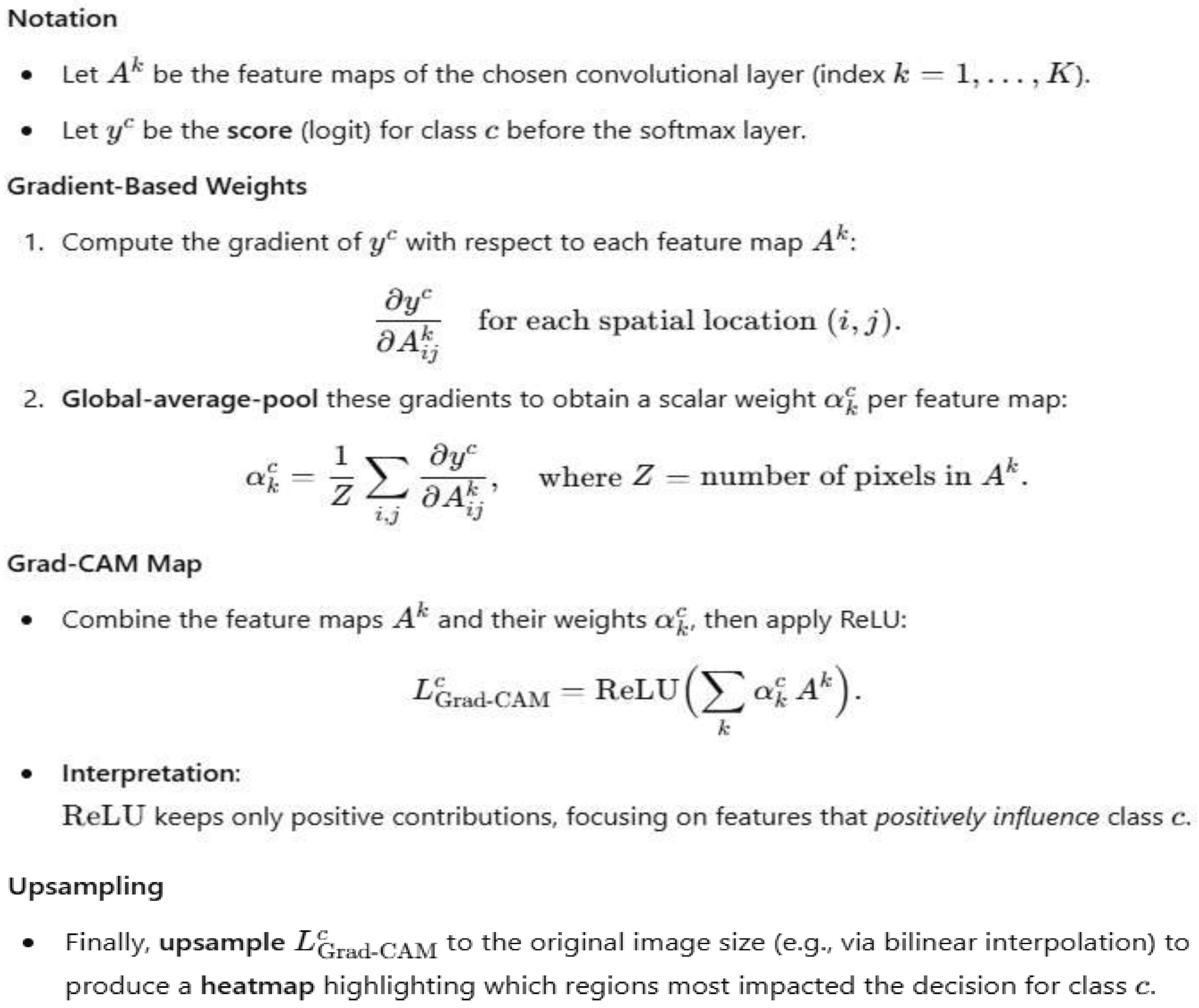
- $M_c(x,y)$ directly indicates *how important* the activation at (x,y) is for classifying the image into class c.

Grad-CAM: Visual Explanations for CNNs

Gradient-weighted Class Activation Mapping (Grad-CAM) produces a **heatmap** over the input image, highlighting which regions were most influential for a given class prediction



Grad-CAM: Mathematical Formulation



Notation

- Let A^k be the feature maps of the chosen convolutional layer (index $k = 1, \dots, K$).
- Let y^c be the **score** (logit) for class c before the softmax layer.

Gradient-Based Weights

1. Compute the gradient of y^c with respect to each feature map A^k:

$$\frac{\partial y^c}{\partial A^k_{ij}} \quad \text{for each spatial location } (i, j).$$

2. **Global-average-pool** these gradients to obtain a scalar weight α^c_k per feature map:

$$\alpha^c_k = \frac{1}{Z} \sum_{i,j} \frac{\partial y^c}{\partial A^k_{ij}}, \quad \text{where } Z = \text{number of pixels in } A^k.$$

Grad-CAM Map

- Combine the feature maps A^k and their weights α^c_k, then apply ReLU:

$$L^c_{\text{Grad-CAM}} = \mathrm{ReLU}\Big(\sum_k \alpha^c_k \, A^k\Big).$$

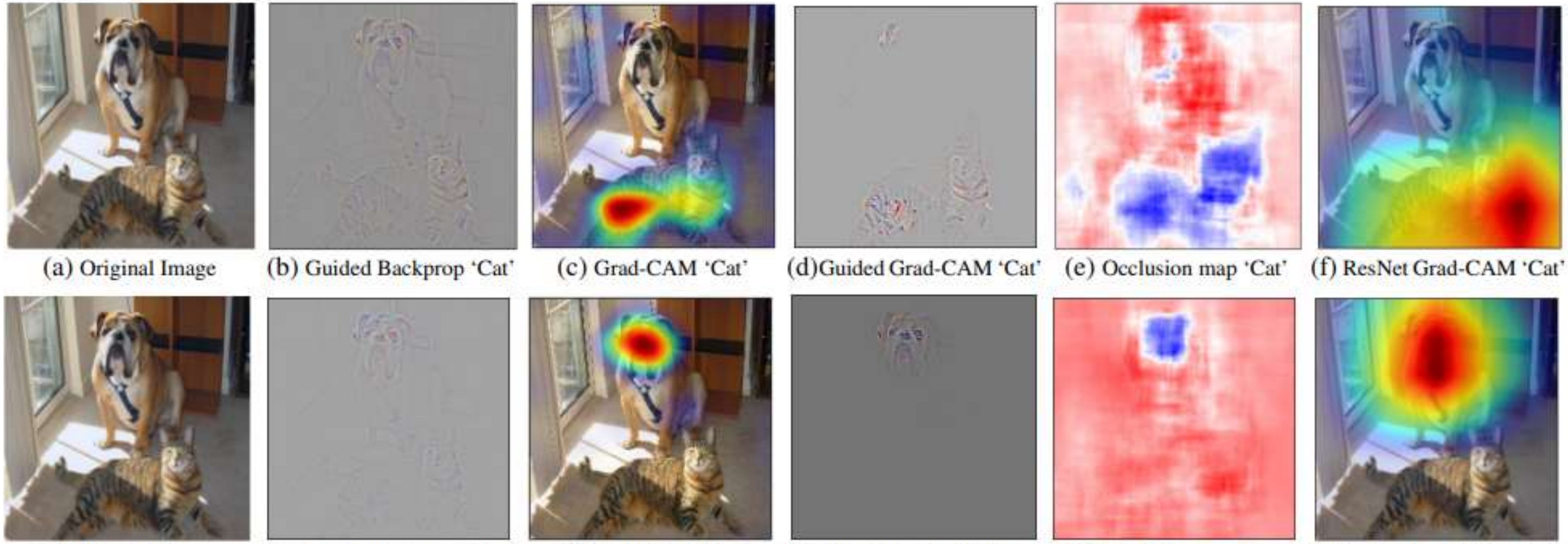
- **Interpretation:**
  ReLU keeps only positive contributions, focusing on features that *positively influence* class c.

Upsampling

- Finally, **upsample** $L^c_{\text{Grad-CAM}}$ to the original image size (e.g., via bilinear interpolation) to produce a **heatmap** highlighting which regions most impacted the decision for class c.

Grad-CAM: Visual Explanations for CNNs

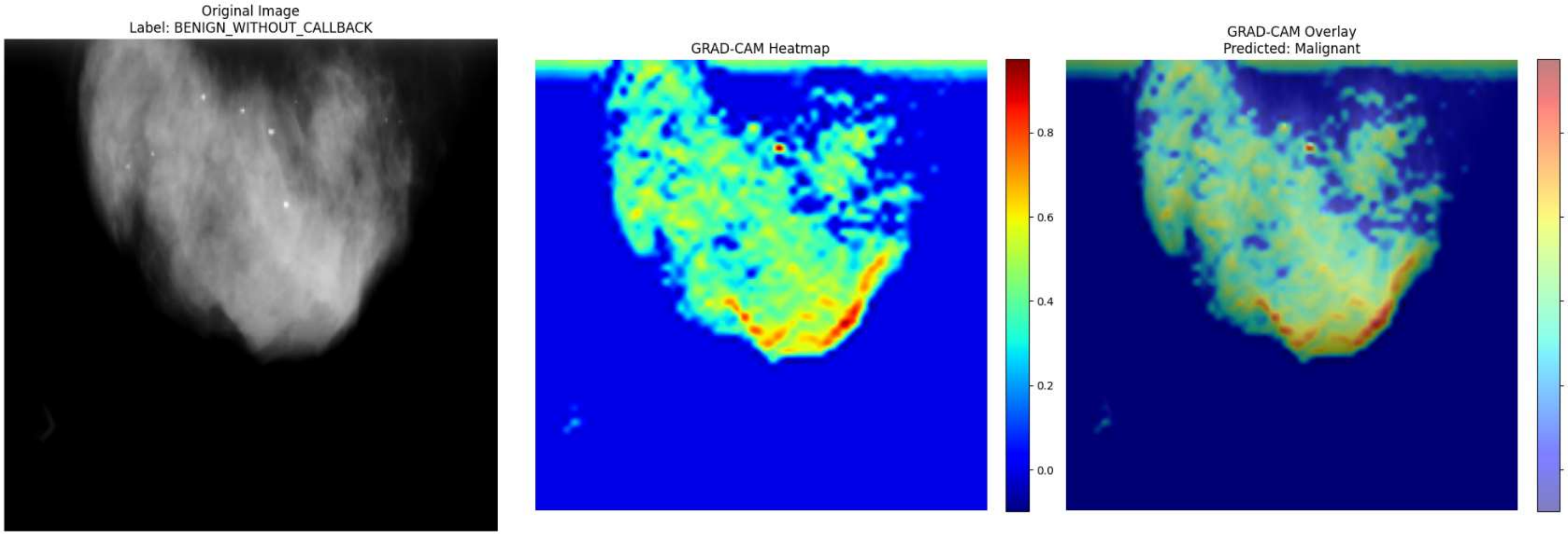
When dealing with convolutional neural networks (CNNs) on images (like most medical image models), **Grad-CAM** is a go-to XAI method.



(a) Original Image (b) Guided Backprop 'Cat' (c) Grad-CAM 'Cat' (d)Guided Grad-CAM 'Cat' (e) Occlusion map 'Cat' (f) ResNet Grad-CAM 'Cat'

(g) Original Image (h) Guided Backprop 'Dog' (i) Grad-CAM 'Dog' (j)Guided Grad-CAM 'Dog' (k) Occlusion map 'Dog' (l)ResNet Grad-CAM 'Dog'

Grad-CAM: Visual Explanations for CNNs



Grad-CAM

- **Forward Pass:** Pass the input image through the CNN.
- **Compute Gradients:** Calculate the gradients of the class score (e.g., “Benign") with respect to the feature maps of a chosen convolutional layer (usually the last one).
- **Global Average Pooling:** Average the gradients across each feature map to obtain "neuron importance weights."
- **Weighted Combination:** Multiply each feature map by its corresponding weight and sum them up.
- **ReLU:** Apply a ReLU activation to keep only positive contributions (features that positively influence the class).
- **Upsample:** Upsample the resulting heatmap to the size of the input image.

[Reference: Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-CAM: Visual explanations from deep networks via gradient-based localization. Proceedings of the IEEE international conference on computer vision, 618-626.]

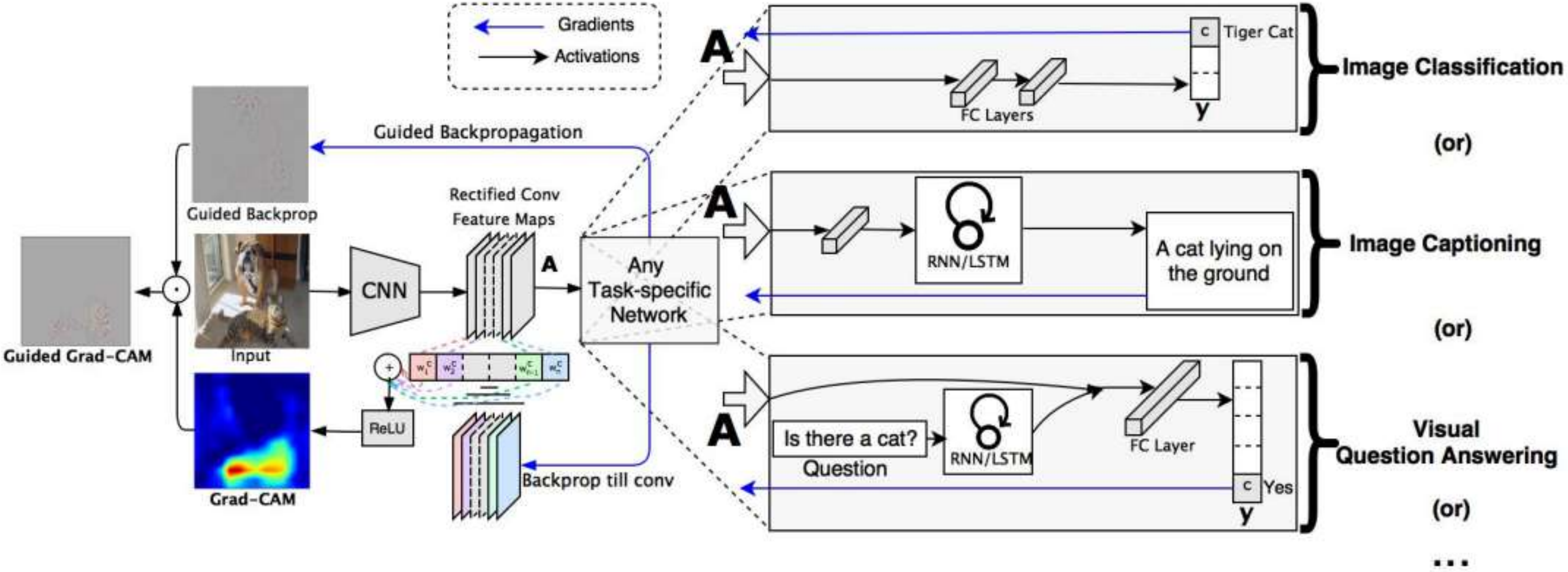
Grad-CAM

- **Advantages:**
  - **Visual Explanations:** Provides intuitive heatmaps that highlight important regions.
  - **Model-Specific (CNNs):** Well-suited for convolutional neural networks, leveraging their architecture.
  - **Relatively Simple:** Easy to implement and computationally efficient.

- **Limitations:**
  - **Coarse Localization:** Heatmaps can be low-resolution and may not precisely pinpoint the exact boundaries of relevant regions.
  - **Layer Choice:** The choice of the convolutional layer can affect the results.
  - **Gradient Saturation:** Gradients can saturate, leading to less informative heatmaps.

Guided Grad-CAM = Guided Backpropagation + Grad-CAM

- Combines **Grad-CAM** with **Guided Backpropagation** to produce higher-resolution, more detailed visualizations.



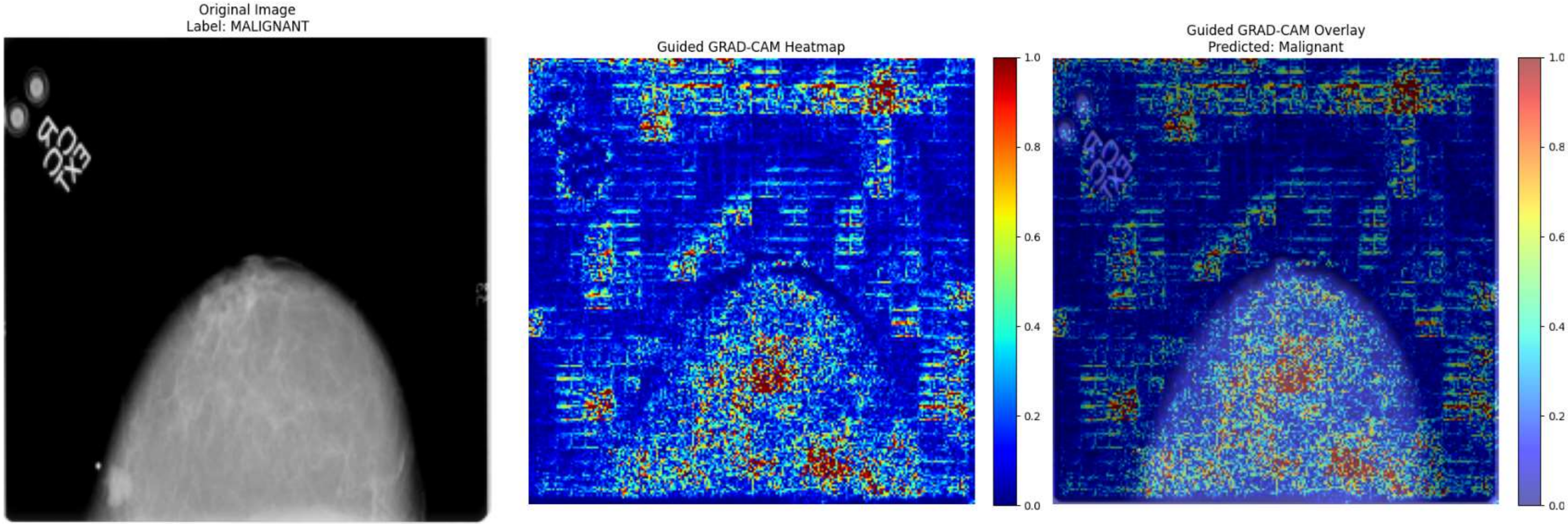
Guided Grad-CAM = Guided Backpropagation + Grad-CAM

- Combines **Grad-CAM** with **Guided Backpropagation** to produce higher-resolution, more detailed visualizations.

- **Guided Backpropagation:** A modified backpropagation algorithm that only propagates positive gradients, highlighting features that positively contribute to the activation of a neuron.

- **Combination:** Element-wise multiplication of the Grad-CAM heatmap and the Guided Backpropagation output.

Guided Grad-CAM = Guided Backpropagation + Grad-CAM



SmoothGrad

Problem: Gradient-based methods (like Grad-CAM) can be sensitive to noise in the input image, leading to noisy explanations.

Solution: SmoothGrad averages the explanations obtained from multiple noisy versions of the input image.

Note: SmoothGrad requires multiple forward-backward passes, so it’s a bit more computational work for a cleaner visualization.

SmoothGrad

Steps:

- Add small amounts of Gaussian noise to the input image multiple times.
- Compute the explanation (e.g., Grad-CAM, Guided Backpropagation) for each noisy image.
- Average the explanations to obtain a smoother, more robust explanation.

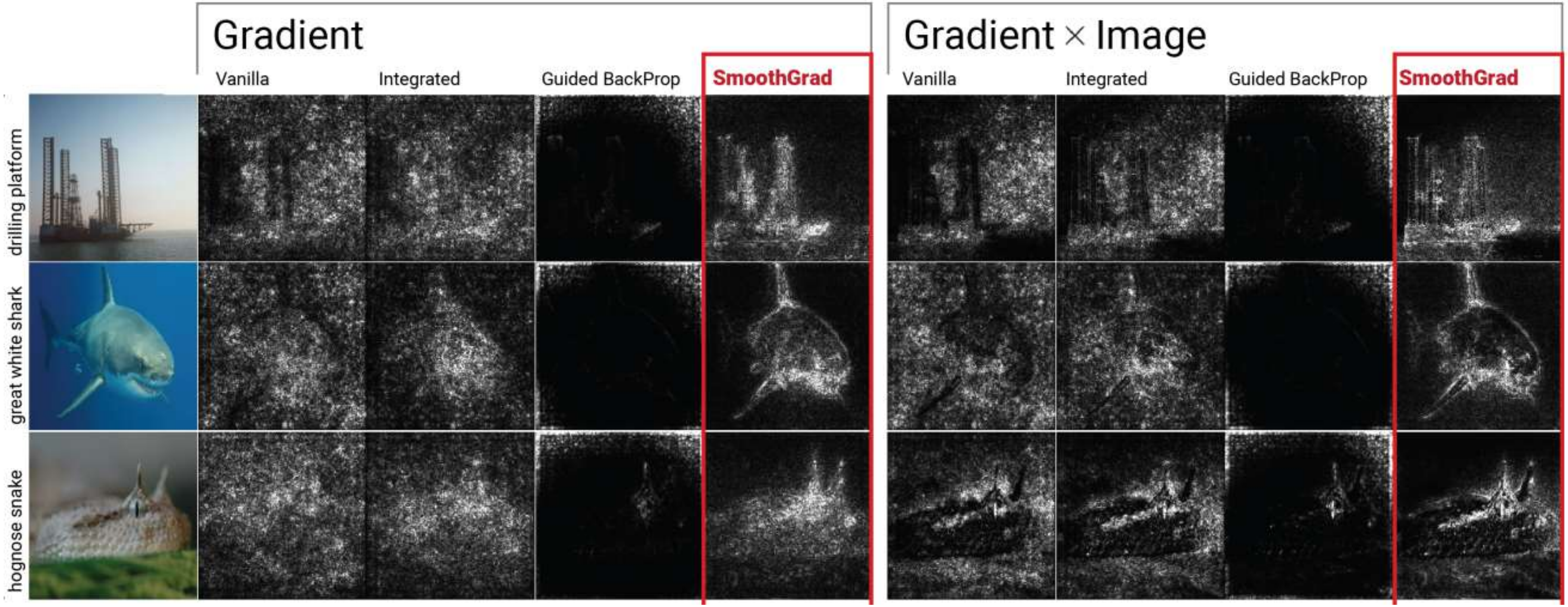
SmoothGrad

Steps:

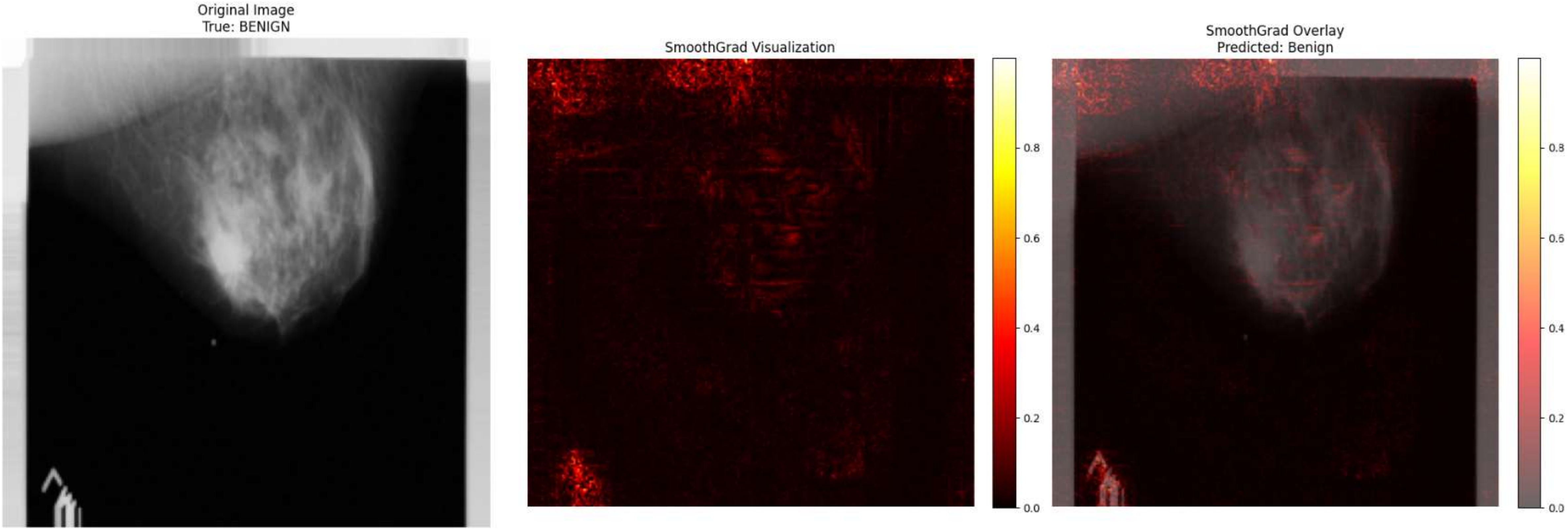
- Add small amounts of Gaussian noise to the input image multiple times.
- Compute the explanation (e.g., Grad-CAM, Guided Backpropagation) for each noisy image.
- Average the explanations to obtain a smoother, more robust explanation.

[Reference: Smilkov, D., Thorat, N., Kim, B., Viégas, F., & Wattenberg, M. (2017). Smoothgrad: removing noise by adding noise. arXiv preprint arXiv:1706.03825.]

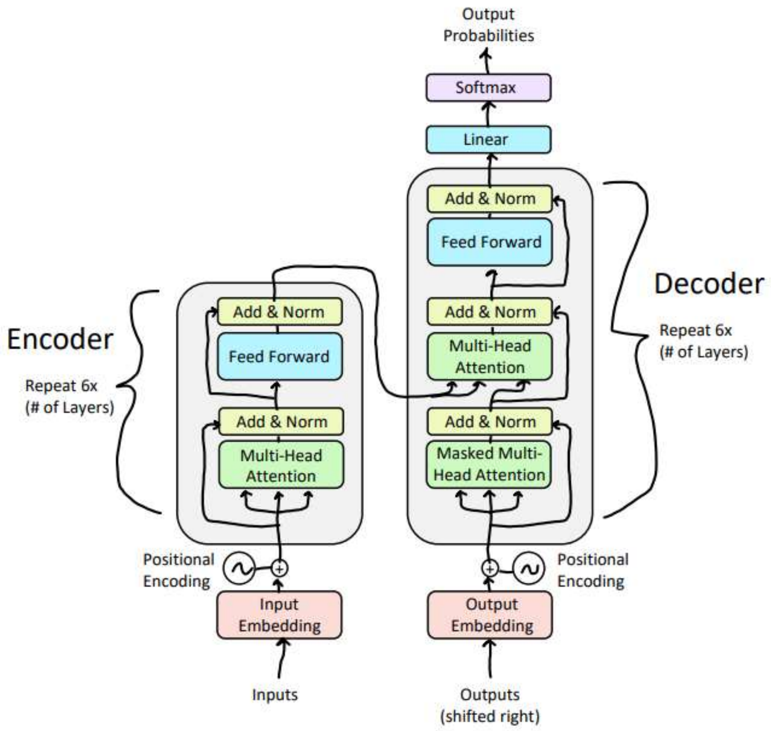
SmoothGrad


Gradient
Vanilla
Integrated
Guided BackProp
SmoothGrad
Gradient × Image
Vanilla
Integrated
Guided BackProp
SmoothGrad
drilling platform
great white shark
hognose snake

SmoothGrad


Original Image
True: BENIGN
SmoothGrad Visualization
0.8
0.6
0.4
0.2
0.0
SmoothGrad Overlay
Predicted: Benign
0.8
0.6
0.4
0.2
0.0

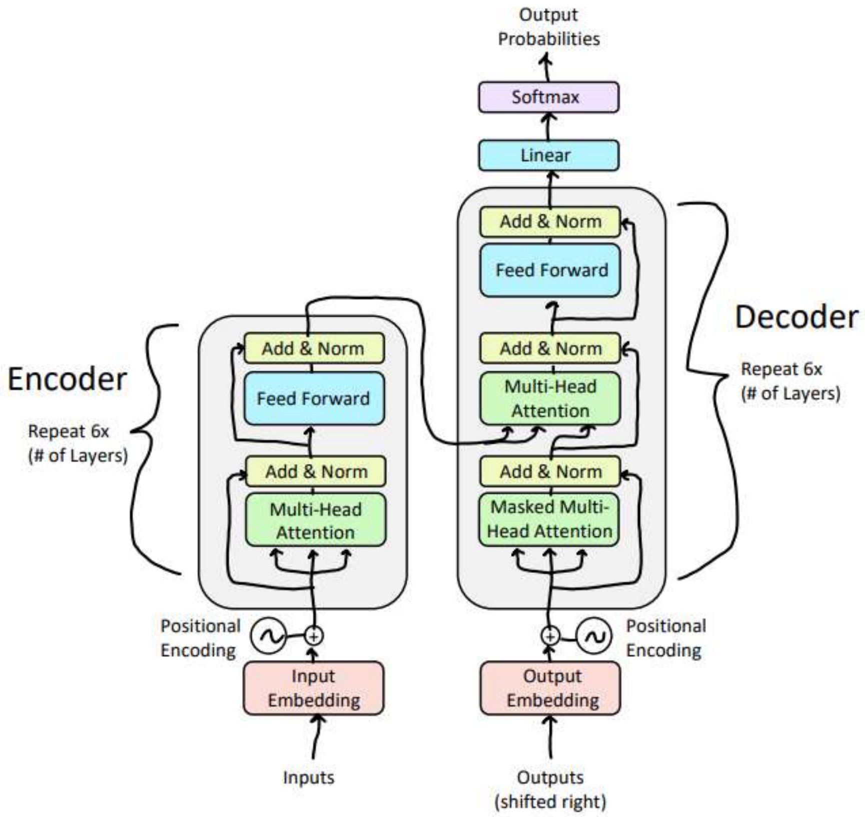
Introduction to Transformers



Courtesy of Paramount Pictures

Introduction to Transformers

- **Relative Positional Embeddings**: Encode sequence order in a more context-specific manner, often improving performance over fixed sine/cosine positional embeddings.
- **Attention Mechanism**: Focuses on the most relevant parts of the input; reduces reliance on strictly local (convolutional) or sequential (recurrent) structures.
- **Parallelization**: Processes the entire sequence in parallel, allowing more efficient training on modern hardware (e.g., GPUs, TPUs).
- **Computational Efficiency**: Self-attention can scale better than RNNs for very long sequences (though attention can be O(n^2) in sequence length, many variants/improvements exist).
  - ❖ **Multi-Purpose Architecture:**
  - ❖ **Large Language Models (LLMs)**: GPT, BERT, etc.
  - ❖ **Protein Structure Prediction**: AlphaFold, RosettaFold.
  - ❖ **Computer Vision**: Vision Transformers (ViT) for image classification, segmentation, etc.



Mathematical View of Attention in Mammogram Patch Analysis

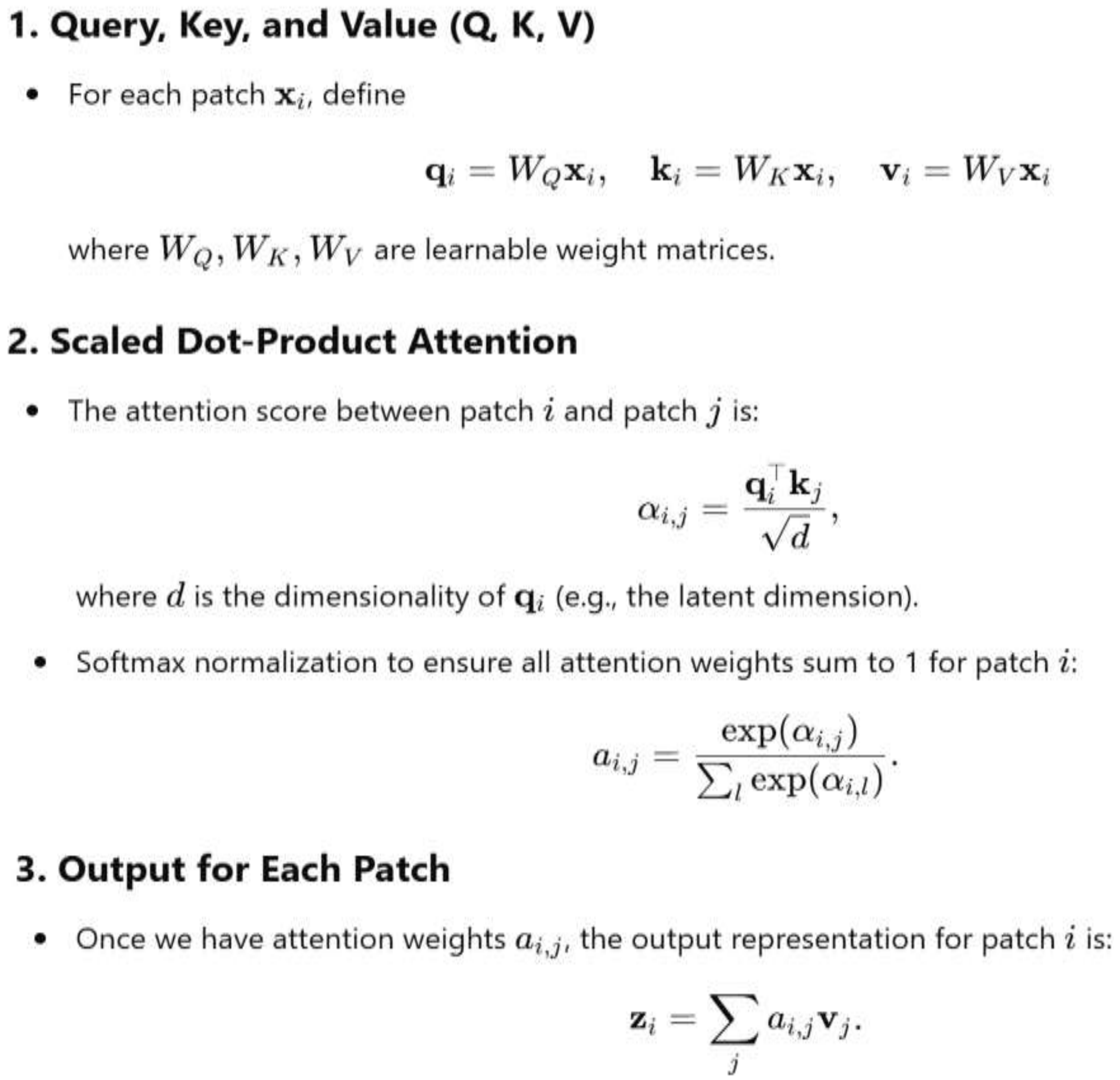
Context

- We have mammogram images subdivided into patches (e.g., 16×16 or 32×32 areas).

- Each patch is represented as a vector $\mathbf{x}_i$

- Goal is to compute attention weights that highlight suspicious or diagnostically relevant patches.

Mathematical View of Attention in Mammogram Patch Analysis



1. Query, Key, and Value (Q, K, V)

- For each patch $\mathbf{x}_i$, define

$$\mathbf{q}_i = W_Q \mathbf{x}_i, \quad \mathbf{k}_i = W_K \mathbf{x}_i, \quad \mathbf{v}_i = W_V \mathbf{x}_i$$

where W_Q, W_K, W_V are learnable weight matrices.

2. Scaled Dot-Product Attention

- The attention score between patch i and patch j is:

$$\alpha_{i,j} = \frac{\mathbf{q}_i^\top \mathbf{k}_j}{\sqrt{d}},$$

where d is the dimensionality of $\mathbf{q}_i$ (e.g., the latent dimension).

- Softmax normalization to ensure all attention weights sum to 1 for patch i:

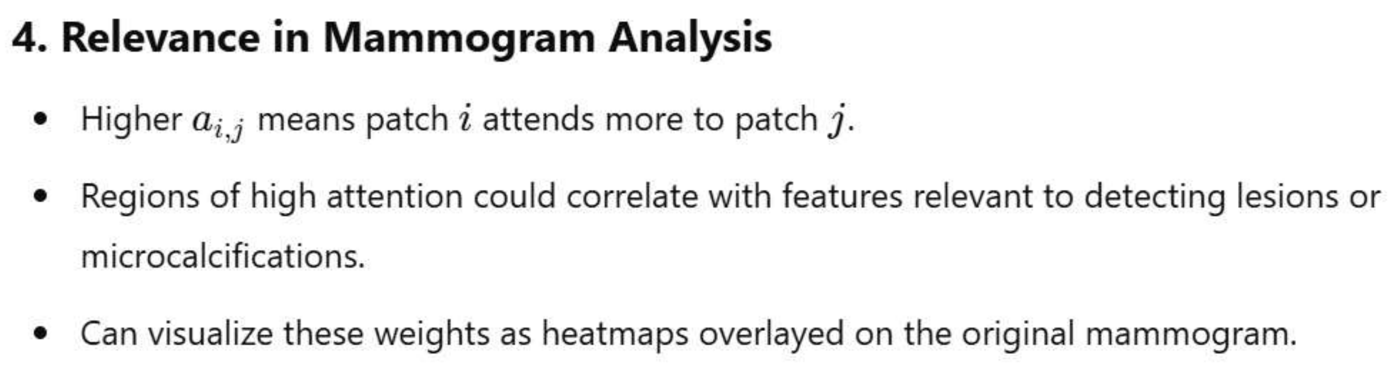
$$a_{i,j} = \frac{\exp(\alpha_{i,j})}{\sum_l \exp(\alpha_{i,l})}.$$

3. Output for Each Patch

- Once we have attention weights $a_{i,j}$, the output representation for patch i is:

$$\mathbf{z}_i = \sum_j a_{i,j} \mathbf{v}_j.$$

Mathematical View of Attention in Mammogram Patch Analysis



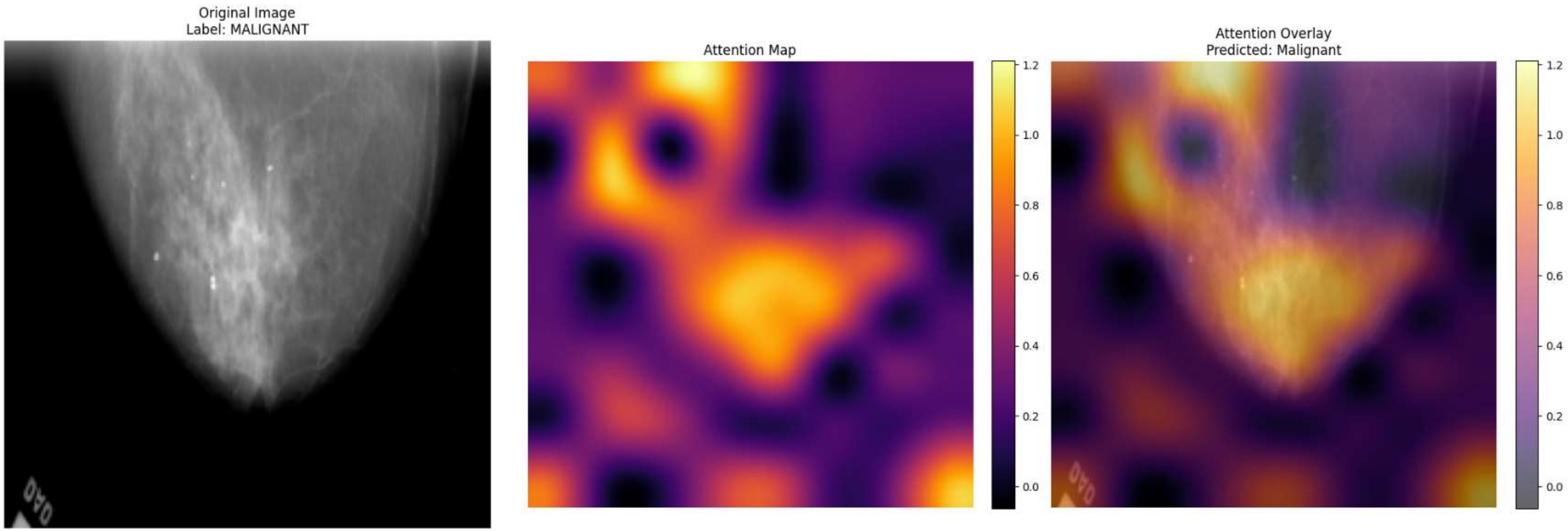
4. Relevance in Mammogram Analysis

- Higher $a_{i,j}$ means patch i attends more to patch j.
- Regions of high attention could correlate with features relevant to detecting lesions or microcalcifications.
- Can visualize these weights as heatmaps overlayed on the original mammogram.

Instead of treating the whole image uniformly, the model computes *attention weights* that indicate which regions (or patches) are more important for the current inference. It’s analogous to how a radiologist might scan an X-ray but pay extra attention to a suspicious spot – the model learns to do this focusing by itself.

[Reference: Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., ... & Polosukhin, I. (2017). Attention is all you need. Advances in neural information processing systems, 30.]

Attention Visualization



Original Image
Label: MALIGNANT

Attention Map

Attention Overlay
Predicted: Malignant

Practical Applications of Explainability

Informed Model Selection

- Facilitates choosing models or designing architectures that align best with clinical needs.
- Helps clinicians identify if a model relies on confounding artifacts or genuine pathology cues.

Robust Training and Fine-Tuning

- Explanation methods can pinpoint weaknesses in the model's learned representations.
- Guides data augmentation or hyperparameter tuning to address misclassified or underrepresented cases.

Performance Under Noise or Adversarial Attacks

- Identifies brittle spots and guides adversarial training or regularization strategies.
- Enhances trust by demonstrating robust predictions even when data is imperfect or attacked.

Standardization Across Clinical Diversity

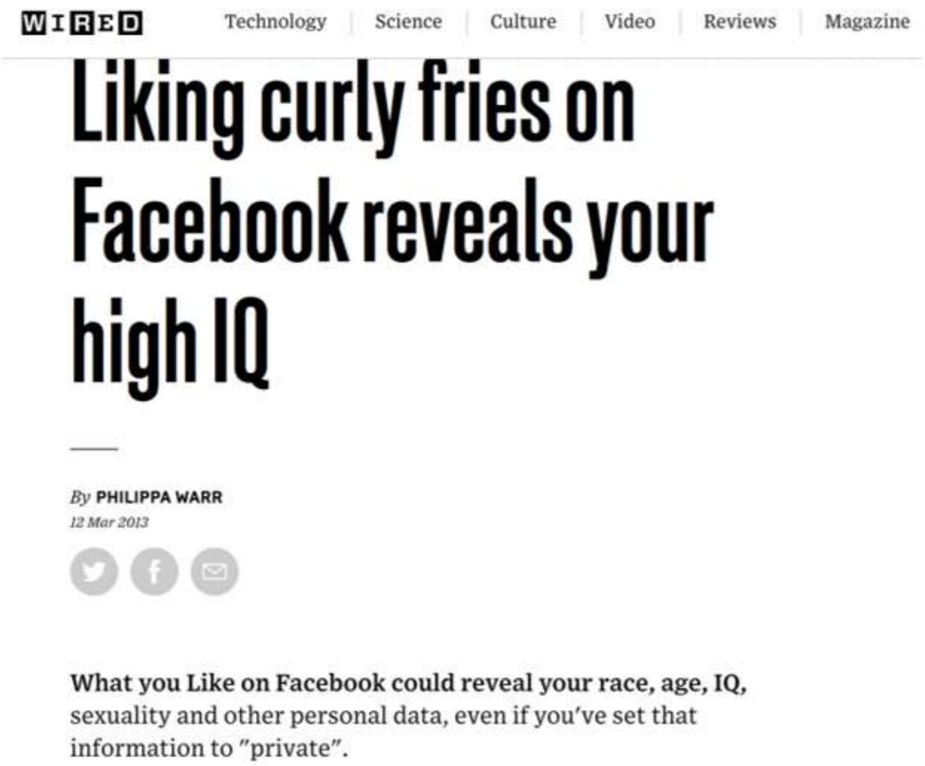
- Explaining model decisions reveals if the model is overfitting to specific devices, protocols, or demographics.
- Supports model "fairness" by uncovering biases or disparities in performance.

Future Directions

- **Robustness and Reliability:** Improving the stability and consistency of explanations.
- **User-Centered Design:** Developing XAI methods that are tailored to the needs of clinicians.
- **Quantitative Evaluation:** Developing objective metrics for evaluating the quality of explanations.
- **Beyond Image Classification:** Extending XAI techniques to other medical imaging tasks (e.g., segmentation, object detection, image registration).
- **Integration with Clinical Workflow:** Seamlessly integrating XAI into clinical decision support systems.
- **Multimodal Explanations:** Combining explanations from different modalities (e.g., imaging, clinical notes, lab results).

Adversarial examples

- **Adversarial examples** are subtly perturbed inputs designed to fool AI models. These changes are often imperceptible to humans.
- This vulnerability is a serious concern in healthcare, where incorrect diagnoses can have life-or-death consequences.

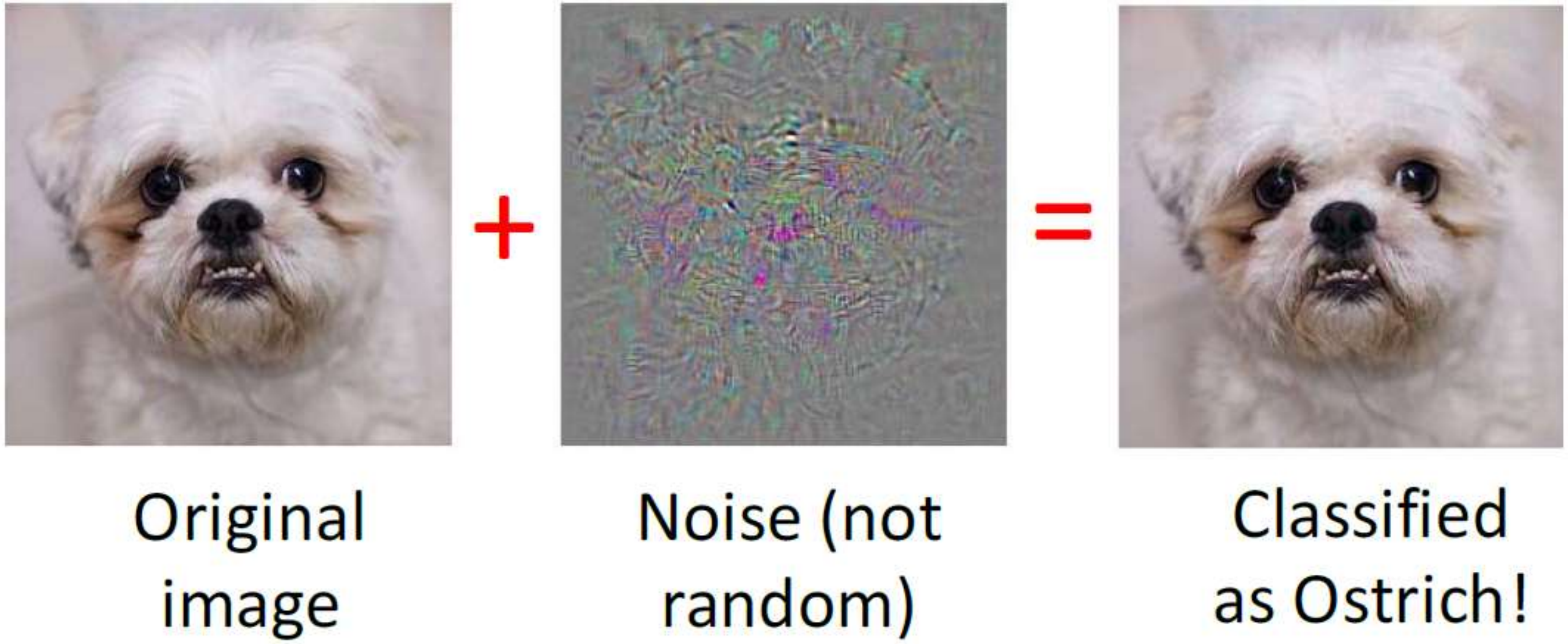


Liking curly fries on Facebook reveals your high IQ

By PHILIPPA WARR
12 Mar 2013

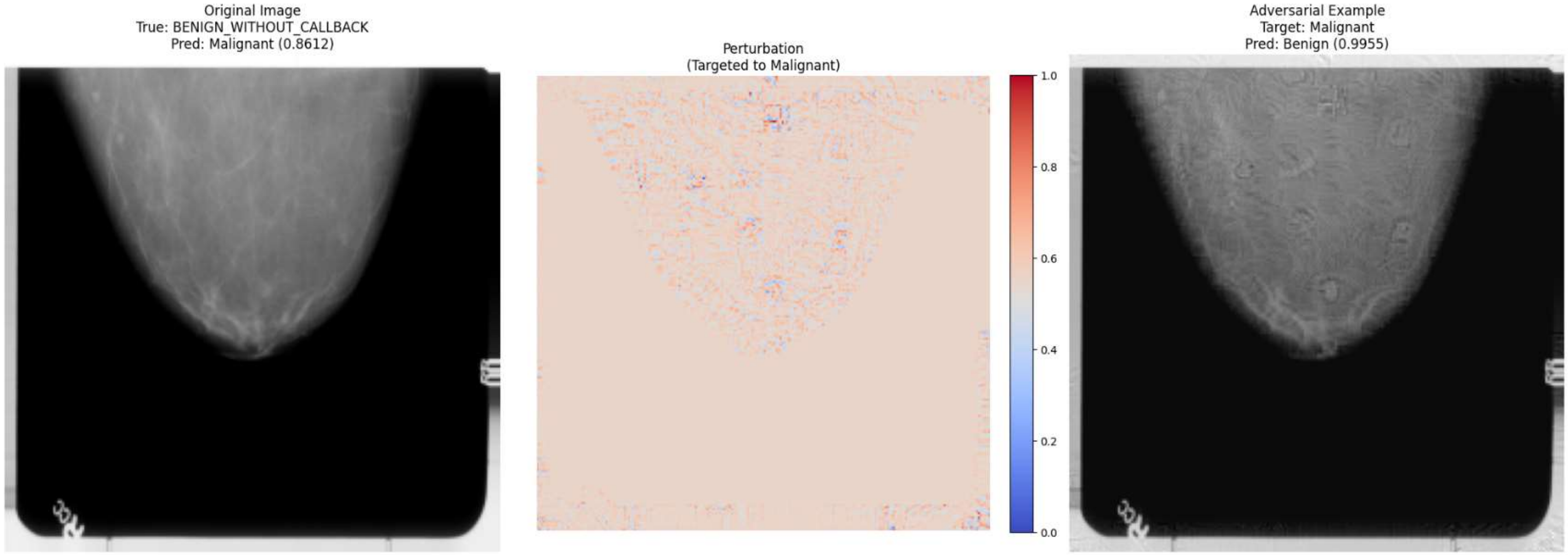
What you Like on Facebook could reveal your race, age, IQ, sexuality and other personal data, even if you've set that information to "private".

Adversarial examples



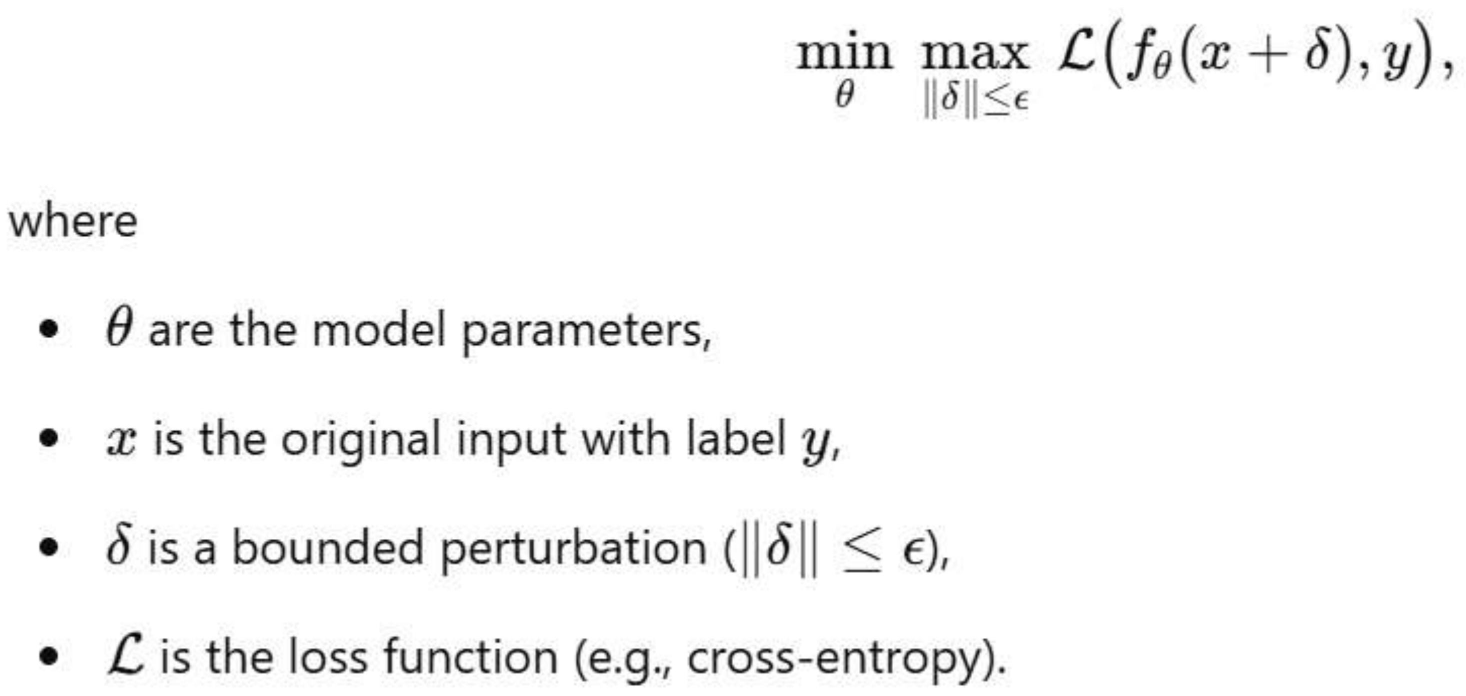
[Reference: Goodfellow, I. J., Shlens, J., & Szegedy, C. (2014). Explaining and harnessing adversarial examples. arXiv preprint arXiv:1412.6572.]

Adversarial examples



Adversarial training

Optimization Objective



$$\min_{\theta} \max_{\|\delta\| \leq \epsilon} \mathcal{L}\big(f_\theta(x+\delta), y\big),$$

where

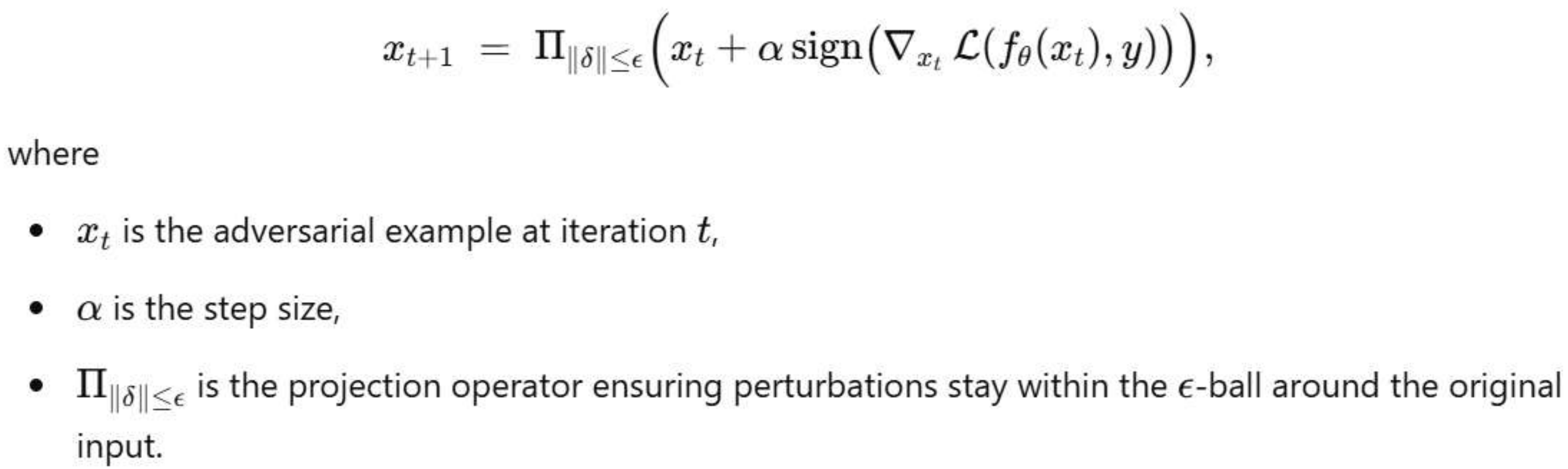
- θ are the model parameters,
- x is the original input with label y,
- δ is a bounded perturbation ($\|\delta\| \leq \epsilon$),
- $\mathcal{L}$ is the loss function (e.g., cross-entropy).

Adversarial Example Generation

- **Fast Gradient Sign Method (FGSM)**: Single-step gradient-based attack.

- **Projected Gradient Descent (PGD)**: Iterative version of FGSM, repeatedly refining the perturbation.

Both the gradient based attack methods rely on,



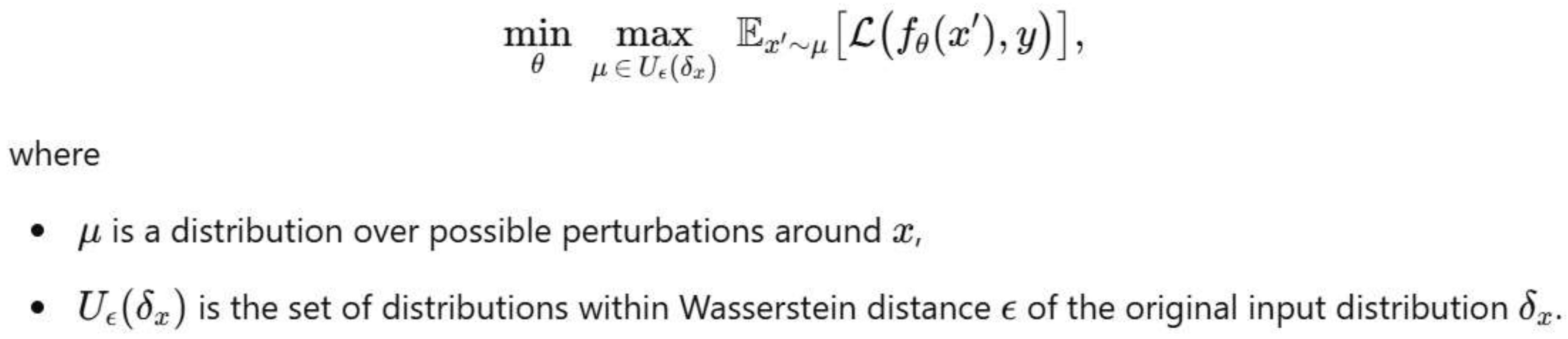
$$x_{t+1} = \Pi_{\|\delta\|\le\epsilon}\Big(x_t + \alpha\,\mathrm{sign}\big(\nabla_{x_t}\mathcal{L}(f_\theta(x_t),y)\big)\Big),$$

where

- x_t is the adversarial example at iteration t,
- α is the step size,
- $\Pi_{\|\delta\|\le\epsilon}$ is the projection operator ensuring perturbations stay within the ϵ-ball around the original input.

Adversarial Example Generation

- Wasserstein attack focuses on perturbations measured by the **Wasserstein distance** (distributional robustness).
- Seek adversarial distribution shifts rather than just pointwise perturbations.
- Ensures robustness under broader transformations (e.g., geometric changes).



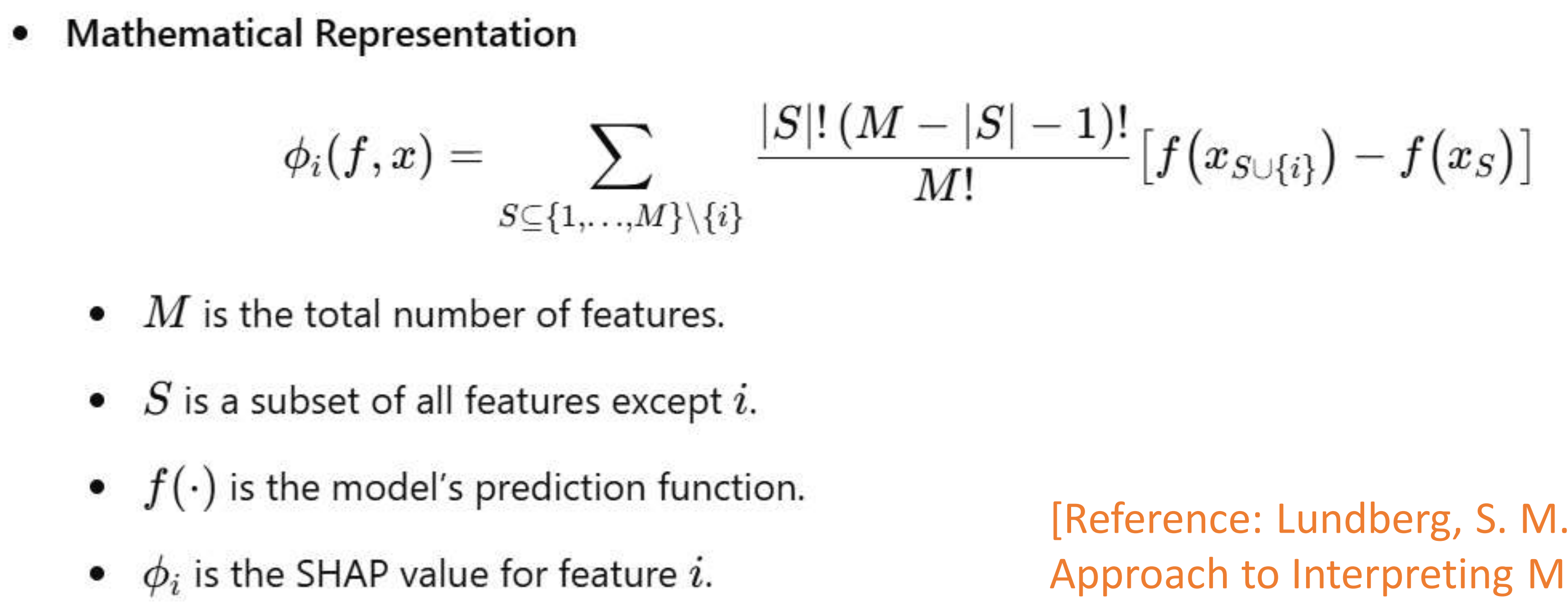
$$\min_{\theta} \max_{\mu \in U_\epsilon(\delta_x)} \mathbb{E}_{x' \sim \mu}\left[\mathcal{L}\left(f_\theta(x'), y\right)\right],$$

where

- μ is a distribution over possible perturbations around x,
- $U_\epsilon(\delta_x)$ is the set of distributions within Wasserstein distance ϵ of the original input distribution δ_x.

SHAP

- **SHAP (SHapley Additive exPlanations)** is a game-theoretic approach to explain individual predictions.
- Assigns each feature a *Shapley value* that quantifies its contribution to the prediction, averaging over all possible subsets of features.



- Mathematical Representation

$$\phi_i(f,x) = \sum_{S \subseteq \{1,\ldots,M\} \setminus \{i\}} \frac{|S|!\,(M-|S|-1)!}{M!}\left[f\left(x_{S\cup\{i\}}\right) - f\left(x_S\right)\right]$$

  - M is the total number of features.
  - S is a subset of all features except i.
  - $f(\cdot)$ is the model's prediction function.
  - ϕ_i is the SHAP value for feature i.

[Reference: Lundberg, S. M., & Lee, S.-I. (2017). A Unified Approach to Interpreting Model Predictions. NeurIPS.]

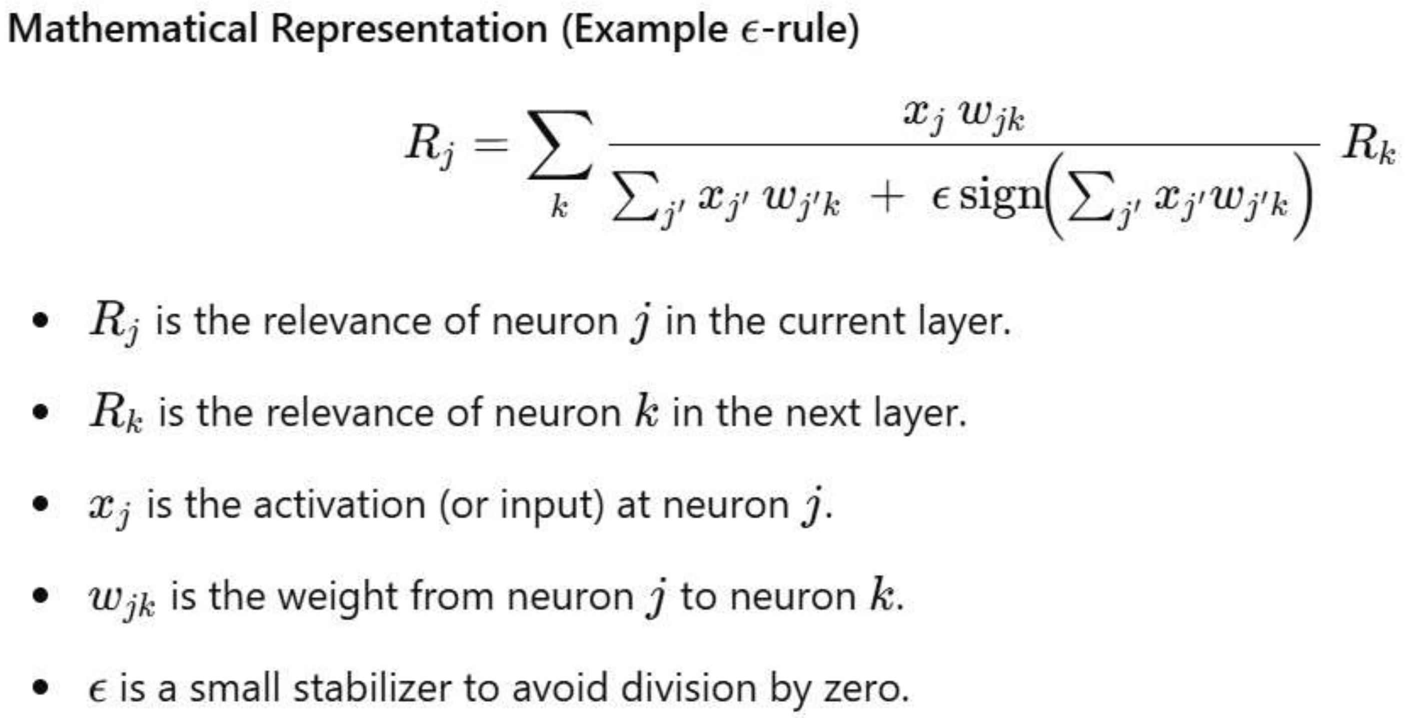
Layerwise Relevance Propagation (LRP)

- **LRP** backpropagates a “relevance score” from the output layer down to the input to show which input features are most responsible for a given prediction.

Steps

- **Forward Pass**: Obtain the model’s output for a given input.
- **Relevance Backpropagation**: Starting from the output neuron’s relevance, redistribute this relevance backwards through each layer.
- **Layer Rules**: Specific propagation rules define how relevance flows through the network.

Layerwise Relevance Propagation (LRP)



Mathematical Representation (Example ϵ-rule)

$$R_j = \sum_k \frac{x_j \, w_{jk}}{\sum_{j'} x_{j'} \, w_{j'k} \; + \; \epsilon \, \mathrm{sign}\left(\sum_{j'} x_{j'} w_{j'k}\right)} R_k$$

- R_j is the relevance of neuron j in the current layer.
- R_k is the relevance of neuron k in the next layer.
- x_j is the activation (or input) at neuron j.
- w_{jk} is the weight from neuron j to neuron k.
- ϵ is a small stabilizer to avoid division by zero.

[Reference: Bach, S., et al. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS One, 10(7): e0130140.]