Explainable Artificial Intelligence (XAI)

Introduction and Interpretability

Gianni Franchi

MVA course

Image Saliency in Traditional Computer Vision

What Is Saliency in an Image?

- **Saliency** refers to regions of an image that naturally attract human attention
- Salient regions are:
  - Visually distinctive
  - Informative
  - Perceptually important
- Saliency answers the question:

"Where do we look first in an image?"

Biological Motivation

- Human vision is selective and resource-limited
- The visual system prioritizes:
  - High contrast
  - Rare patterns
  - Strong edges
  - Movement or orientation changes
- Saliency models aim to computationally reproduce this behavior

Key Principles of Saliency

Saliency relies on three fundamental principles:

- **Contrast:** Salient regions differ from surroundings
- **Rarity:** Rare patterns are more salient
- **Spatial coherence:** Salient regions are structured

Center–Surround Contrast

- Inspired by receptive fields in the human visual system
- Compare a pixel (center) to its neighborhood (surround)

$$S(x) = |I(x) - \text{mean}_{\mathcal{N}(x)} I|$$

- High difference $\rightarrow$ high saliency
- Used in many early saliency models

Itti–Koch–Niebur Model (1998)

One of the first computational saliency models

- Extracts multi-scale feature maps:
  - Intensity
  - Color
  - Orientation
- Uses center–surround differences
- Combines feature maps into a global saliency map

Itti Model: Processing Pipeline

1. Build Gaussian pyramids
2. Compute center–surround contrasts
3. Normalize feature maps
4. Combine into conspicuity maps
5. Fuse into final saliency map

Frequency-Based Saliency

- Saliency relates to irregularities in frequency domain
- Natural images have predictable spectra
- Deviations from this regularity are salient

Fourier Decomposition of an Image

Given an image $I(x, y)$, its Fourier transform is:

$$F(u, v) = \mathcal{F}\{I(x, y)\}$$

We decompose it as:

$$F(u, v) = A(u, v)\, e^{i\Phi(u,v)}$$

- $A(u, v) = |F(u, v)|$: **Amplitude spectrum**
- $\Phi(u, v)$: **Phase spectrum**

Important observation:

- Amplitude captures global frequency statistics
- Phase preserves spatial structure

Why Frequency Domain for Saliency?

- Natural images have strong regularities in frequency space
- Their amplitude spectrum typically follows a smooth decay:

$$|F(u,v)| \approx \frac{1}{\sqrt{u^2+v^2}}$$

- This corresponds to:
  - Many low frequencies (smooth regions)
  - Few high frequencies (edges, details)
- **Salient regions correspond to deviations from this regular pattern**

What Is the Spectral Residual?

Step 1: Take logarithm of amplitude

$$L(u, v) = \log A(u, v)$$

Step 2: Compute a local average (smooth version)

$$\bar{L}(u, v) = h * L(u, v)$$

Step 3: Define the spectral residual

$$R(u, v) = L(u, v) - \bar{L}(u, v)$$

Interpretation:

- $\bar{L}(u, v)$: expected frequency content
- $R(u, v)$: unexpected / surprising frequencies
- Saliency = surprise

From Spectral Residual to Saliency Map (Hou & Zhang, 2007)

Reconstruction:

$$\hat{F}(u,v) = e^{R(u,v)+i\Phi(u,v)}$$

Inverse Fourier Transform:

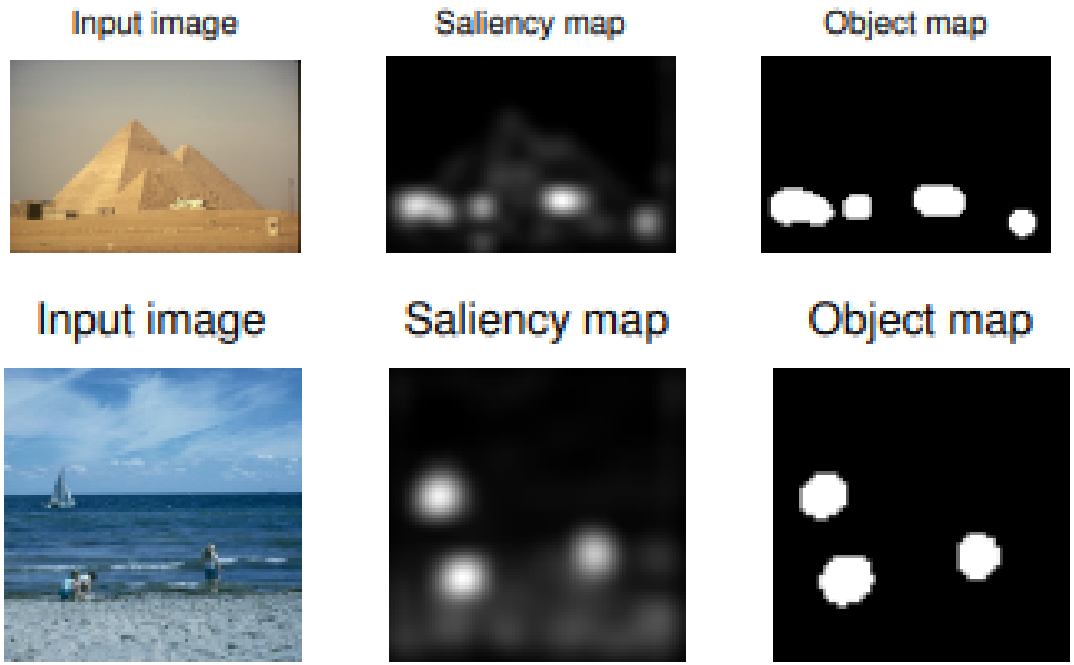
$$S(x,y) = \left|\mathcal{F}^{-1}(\hat{F}(u,v))\right|^2$$

Why this works:

- Phase restores spatial localization
- Residual highlights rare structures
- Squaring emphasizes strong responses

Final result: A pixel-wise saliency map

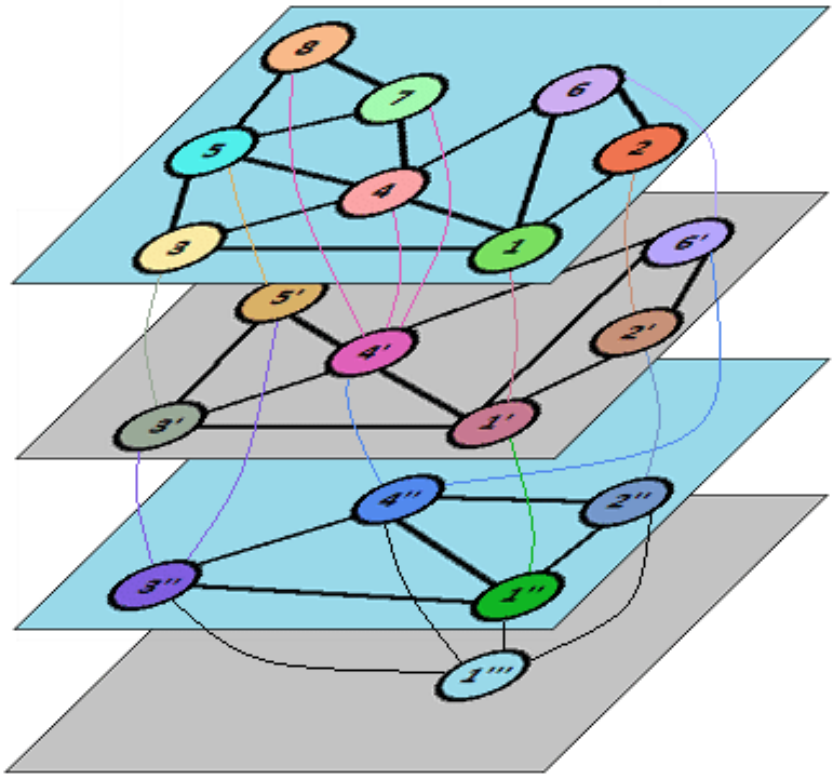
Spectral Residual Saliency (Hou & Zhang, 2007)



Graph-Based Visual Saliency

There are a lot of different techniques. One of them is based on mathematical morphology.

- Represent image as a graph
- Nodes = pixels or regions
- Edges = similarity relationships
- Saliency computed via minimum spanning tree



Region-Based Saliency

- Segment image into regions (superpixels)
- Compute contrast between regions

$$S(R_i) = \sum_j w_{ij} \|\mu_i - \mu_j\|$$

- Encourages spatial coherence
- Reduces noise

Local Entropy: Intuition

- Entropy measures information content
- High entropy = unpredictable, complex region
- Visually complex regions attract attention

Local Entropy: Definition

Given a local neighborhood $\mathcal{N}(x)$:

$$H(x) = -\sum_i p_i \log p_i$$

- p_i: histogram probability of intensity/color
- Computed locally for each pixel or patch

Local Entropy as Saliency Measure

- Flat regions $\rightarrow$ low entropy
- Textured or complex regions $\rightarrow$ high entropy
- Saliency map = entropy map

Used in:

- Texture analysis
- Medical imaging
- Anomaly detection

Image Saliency in XAI

From Saliency to XAI

- Saliency maps inspired:
  - Gradient-based explanations
  - CAM / Grad-CAM
  - Feature attribution methods

Layer-Wise Relevance Propagation

Goal: Pixel-wise Decomposition [2,9]

We want to explain a classifier prediction at the pixel level.

Given:

$$f : \mathbb{R}^V \to \mathbb{R}$$

- Input: image $x = (x_1, \ldots, x_V)$
- Output: real-valued score $f(x)$
- $f(x) > 0$: presence of a learned structure

Objective:

$$f(x) \approx \sum_{d=1}^{V} R_d$$

Each pixel receives a relevance score R_d.

Interpretation of Relevance [2,9]

- $R_d > 0$: pixel supports the prediction
- $R_d < 0$: pixel contradicts the prediction
- Magnitude: strength of contribution

This produces a **signed heatmap** over pixels.

Key constraint:

$$\sum_d R_d = f(x)$$

Relevance must fully explain the prediction.

From Pixels to Layers [2,9]

Modern classifiers are layered:

$$x \to z^{(1)} \to z^{(2)} \to \cdots \to f(x)$$

Let:

$$z^{(l)} = (z_1^{(l)}, \ldots, z_{V(l)}^{(l)})$$

- Input layer: pixels
- Output layer: prediction score

Layer-Wise Relevance Propagation (LRP)[2,9]

LRP propagates relevance **backward** through layers.

For each layer l, define relevance:

$$R_d^{(l)} \quad \text{for each neuron } z_d^{(l)}$$

Conservation principle:

$$\sum_{d \in l+1} R_d^{(l+1)} = \sum_{d \in l} R_d^{(l)}$$

Ultimately:

$$f(x) = \sum_d R_d^{(1)}$$

Relevance as a Conservation Law [9]

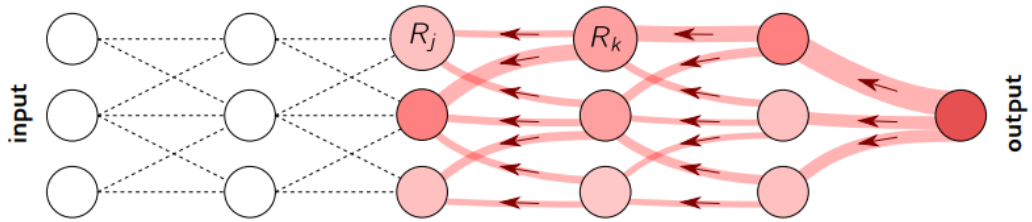
Equation:

$$f(x) = \sum_d R_d^{(L)} = \cdots = \sum_d R_d^{(1)}$$

Interpretation:

- Relevance is neither created nor destroyed
- It is redistributed across neurons

Analogy: conservation of energy or mass.



Why Conservation Alone Is Not Enough[9]

Conservation does **not** guarantee meaningful explanations.

Many decompositions satisfy:

$$\sum_d R_d = f(x)$$

But:

- Signs may be wrong
- All pixels may look equally important
- No relation to model structure

Counterexample: Linear Model[9]

Consider:

$$f(x) = b + \sum_d w_d \phi_d(x_d)$$

A bad relevance definition:

$$R_d = f(x) \frac{|w_d \phi_d(x_d)|}{\sum_d |w_d \phi_d(x_d)|}$$

Problem:

- All R_d have the same sign
- No negative evidence
- No contrast between features

Why This Decomposition Is Wrong[9]

Even pixels that **oppose** the prediction receive positive relevance.

This violates intuition:

- Background pixels should not support the object
- Evidence for and against should coexist

Thus: conservation alone is insufficient.

A More Meaningful Decomposition[9]

Instead define:

$$R_d = \frac{b}{V} + w_d \phi_d(x_d)$$

Now:

- Sign of relevance reflects sign of contribution
- Positive and negative evidence coexist

This motivates **local contribution-based propagation**. **Problem:**

- Many classifiers rely on feature interactions
- This decomposition cannot express synergy

LRP Philosophy[9]

LRP assigns relevance based on:

- Actual signal flow in the network
- Local contributions to activations
- Layer-wise redistribution

Relevance is:

> "how much this neuron contributed to the prediction"

Deep Neural Networks with ReLU[9]

Neuron model:

$$a_k = \max\left(0, \sum_j a_j w_{jk}\right)$$

- a_j: lower-layer activations
- w_{jk}: weights
- Bias implemented via $a_0 = 1$

LRP-0: Basic Rule[9]

Relevance redistribution:

$$R_j = \sum_k \frac{a_j w_{jk}}{\sum_{j'} a_{j'} w_{j'k}} R_k$$

Interpretation:

- Proportional to contribution
- Mirrors forward computation

Properties of LRP-0[9]

- If $a_j = 0$, then $R_j = 0$
- If $w_{jk} = 0$, no relevance flows
- Conserves total relevance

But:

- Equivalent to Gradient $\times$ Input
- Often noisy and unstable

LRP-ϵ: Stabilized Rule

To improve stability:

$$R_j = \sum_k \frac{a_j w_{jk}}{\epsilon + \sum_{j'} a_{j'} w_{j'k}} R_k$$

$\epsilon > 0$ prevents division by small numbers.

Effect:

- Suppresses weak evidence
- Produces sparser explanations

LRP-γ: Favor Positive Evidence[9]

Modified weights:

$$w_{jk} \rightarrow w_{jk} + \gamma w_{jk}^{+}$$

$$R_j = \sum_k \frac{a_j(w_{jk} + \gamma w_{jk}^{+})}{\epsilon + \sum_{j'} a_{j'}(w_{j'k} + \gamma w_{j'k}^{+})} R_k$$

$\gamma > 0$ amplifies positive contributions.

Effect of LRP-γ[9]

- Negative contributions suppressed
- Positive evidence emphasized
- More interpretable heatmaps

Useful for:

- Object localization
- Concept discovery

Connection to Taylor Decomposition[2,9]

LRP can be derived from a first-order Taylor expansion.

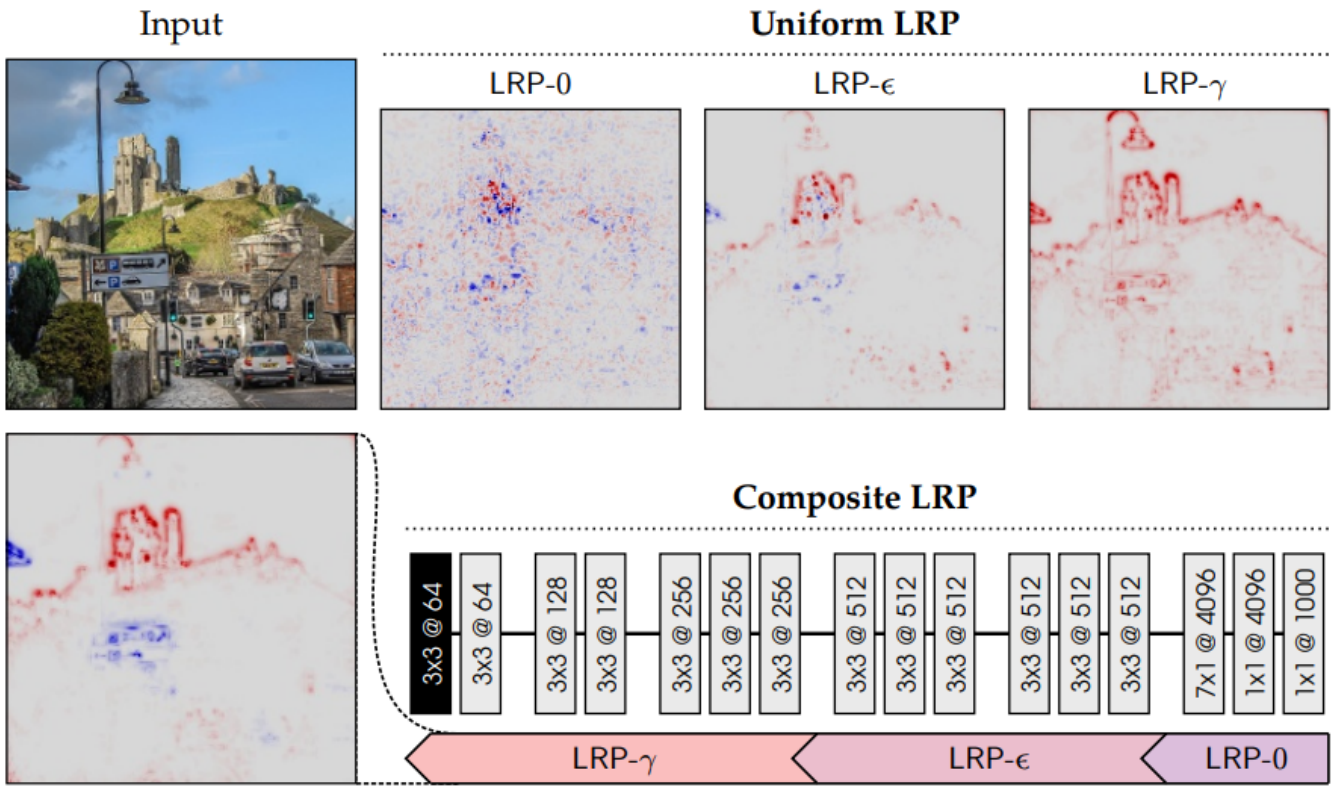
For a neuron:

$$f(x) \approx f(x_0) + \sum_d \frac{\partial f}{\partial x_d}(x_0)(x_d - x_{0,d})$$

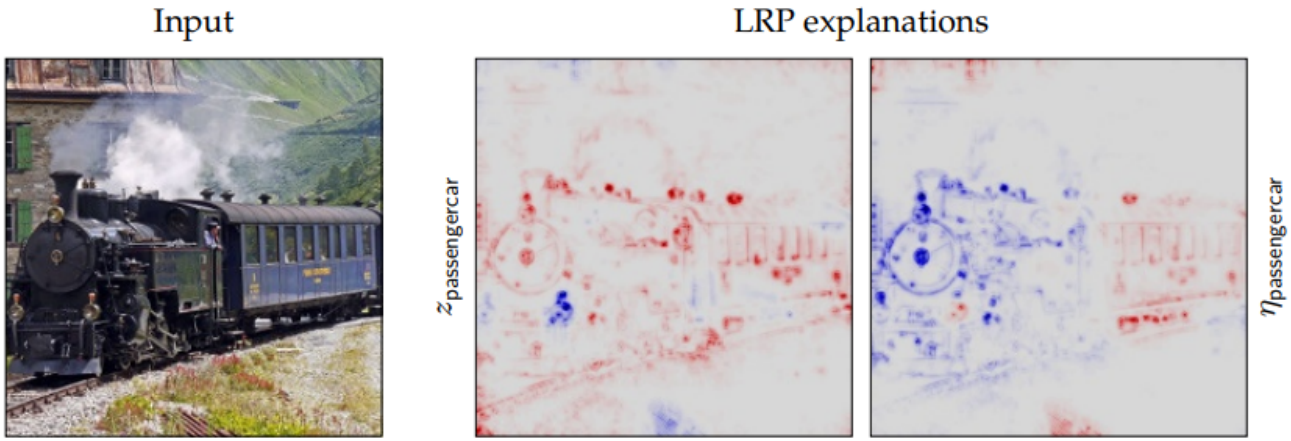
Interpretation:

- Local linear approximation
- Relevance as sensitivity-weighted deviation

LRP in practice[9]



LRP Results[9]



CAM, Grad-CAM, Grad-CAM++, and EigenGrad-CAM

Why Visual Explanations for CNNs?

- Convolutional Neural Networks (CNNs) achieve high predictive accuracy
- However, their decision process is often opaque
- **A solution:** Extract inside the DNN the saliency information to produce a class-specific spatial explanation highlighting important image regions.

CNN Notation

- Input image: $x \in \mathbb{R}^{H \times W \times C}$
- Feature maps at convolutional layer l: $A^k \in \mathbb{R}^{H_l \times W_l}, \quad k = 1, \ldots, K$
- Class score (pre-softmax output): y^c

We seek a spatial map showing how A^k contributes to y^c.

CNN Notation

- Input image:
$$x \in \mathbb{R}^{H \times W \times C}$$
- Feature maps at convolutional layer l:
$$A^{k,l} \in \mathbb{R}^{H_l \times W_l}, \quad k = 1, \ldots, K$$
- Class score (pre-softmax output):
$$y^c$$

We seek a saliancy map Sal showing the contributions of y^c. Most of the techniques
$$\text{Sal}[i,j] = \sum_{k,l} w_{k,l} \tilde{A}^{k,l}_{ij}$$
. $\tilde{A} \in \mathbb{R}^{\tilde{H} \times \tilde{W}}$, are a feature map of different layer l at a given size.

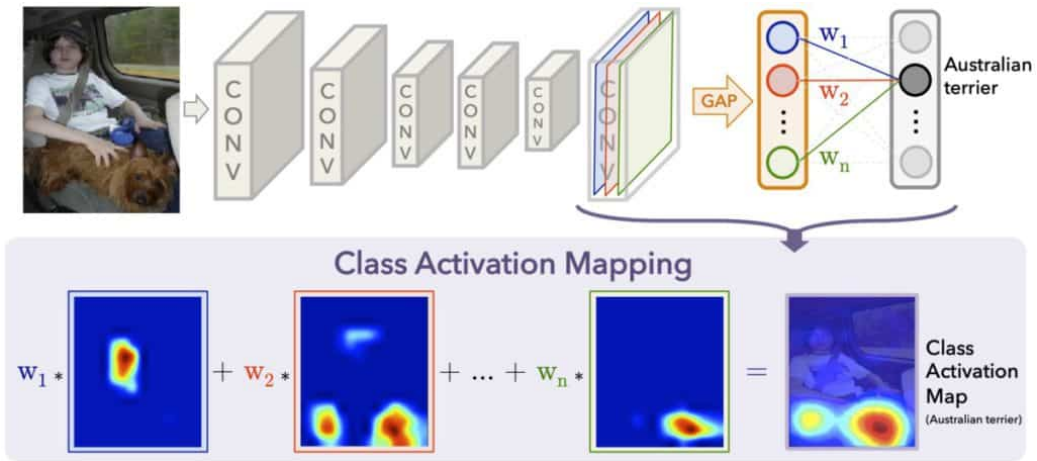
Architectural Constraint of CAM

CAM requires a specific CNN architecture:

- Last convolutional layer
- Global Average Pooling (GAP)
- Linear classifier (no fully connected layers)

$$y^c = \sum_k w_k^c \cdot \text{GAP}(A^k)$$

This constraint limits CAM's applicability.



CAM Mathematical Derivation[4]

Global Average Pooling:

$$\text{GAP}(A^k) = \frac{1}{Z} \sum_{i,j} A^k_{ij}$$

Class score:

$$y^c = \sum_k w^c_k \frac{1}{Z} \sum_{i,j} A^k_{ij}$$

Rewriting:

$$y^c = \frac{1}{Z} \sum_{i,j} \sum_k w^c_k A^k_{ij}$$

Define the class activation map:

$$\text{Sal}^c_{\text{CAM}}[i,j] = \sum_k w^c_k A^k_{ij}$$

- High values indicate important spatial regions
- Map is upsampled to input image resolution

CAM: Strengths and Limitations[4]

Strengths

- Simple and intuitive
- Class-discriminative
- No gradients required

Limitations

- Requires architectural modification
- Cannot be applied to arbitrary pretrained CNNs

Original Image | Grad-CAM 'Cat' | Grad-CAM 'Dog'

Why Grad-CAM?

CAM is limited by architectural constraints.

Grad-CAM addresses this by:

- Using gradients instead of learned weights
- Working with **any CNN architecture**

Grad-CAM: Key Intuition[5]

- Gradients indicate sensitivity of the class score
- Measure how much each feature map influences y^c

$$\frac{\partial y^c}{\partial A_{ij}^k}$$

Larger gradients indicate greater importance.

Grad-CAM Channel Weights[5]

Channel importance is computed as:

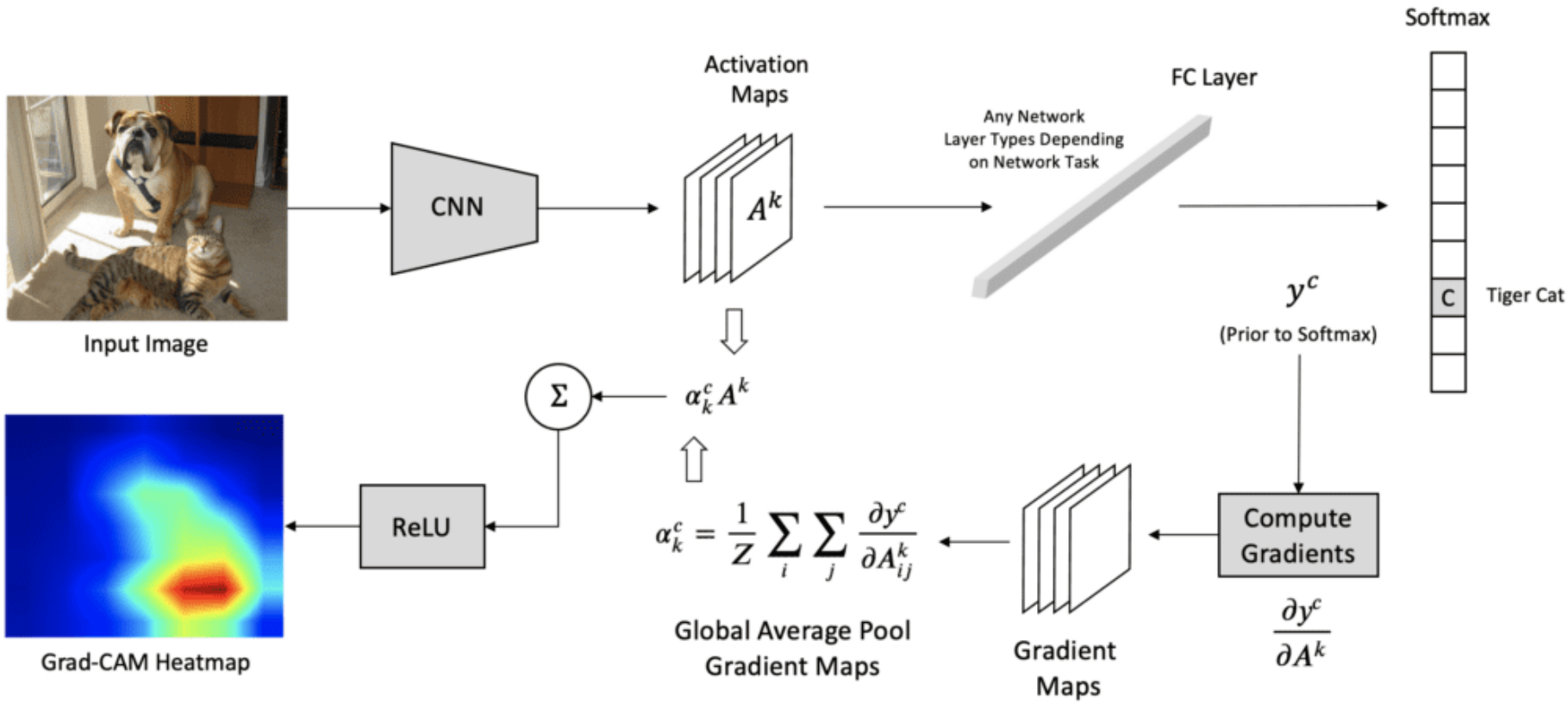
$$\alpha_k^c = \frac{1}{Z} \sum_{i,j} \frac{\partial y^c}{\partial A_{ij}^k}$$

- Global average pooling of gradients
- Measures overall influence of feature map k

$$\text{Sal}^c_{\text{Grad-CAM}}[i,j] = \text{ReLU}\left(\sum_k \alpha_k^c A^k \right)$$

- ReLU keeps only positive contributions
- Focuses on regions supporting the class

Grad-CAM Heatmap[5]



Motivation for Grad-CAM++

Grad-CAM limitations:

- Struggles with multiple object instances
- Poor localization for small objects

Grad-CAM++ refines the weighting scheme.

- Not all spatial locations contribute equally
- Introduce pixel-level weighting
- Use higher-order derivatives

Grad-CAM++ Weight Formula[6]

Pixel-wise weights:

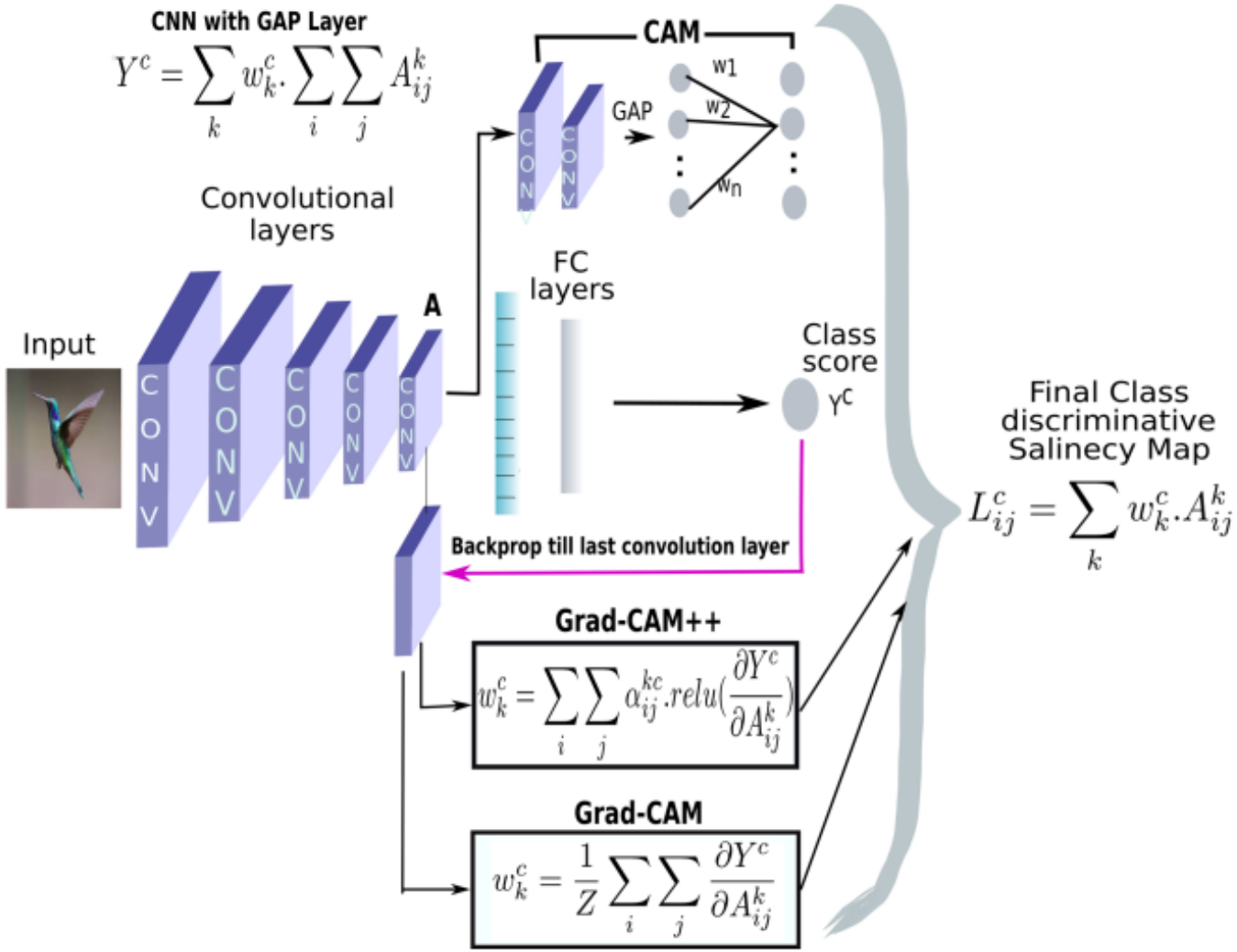
$$\alpha_{ij}^{kc} = \frac{\frac{\partial^2 y^c}{(\partial A_{ij}^k)^2}}{2\frac{\partial^2 y^c}{(\partial A_{ij}^k)^2} + \sum_{a,b} A_{ab}^k \frac{\partial^3 y^c}{(\partial A_{ij}^k)^3}}$$

Allows fine-grained attribution.

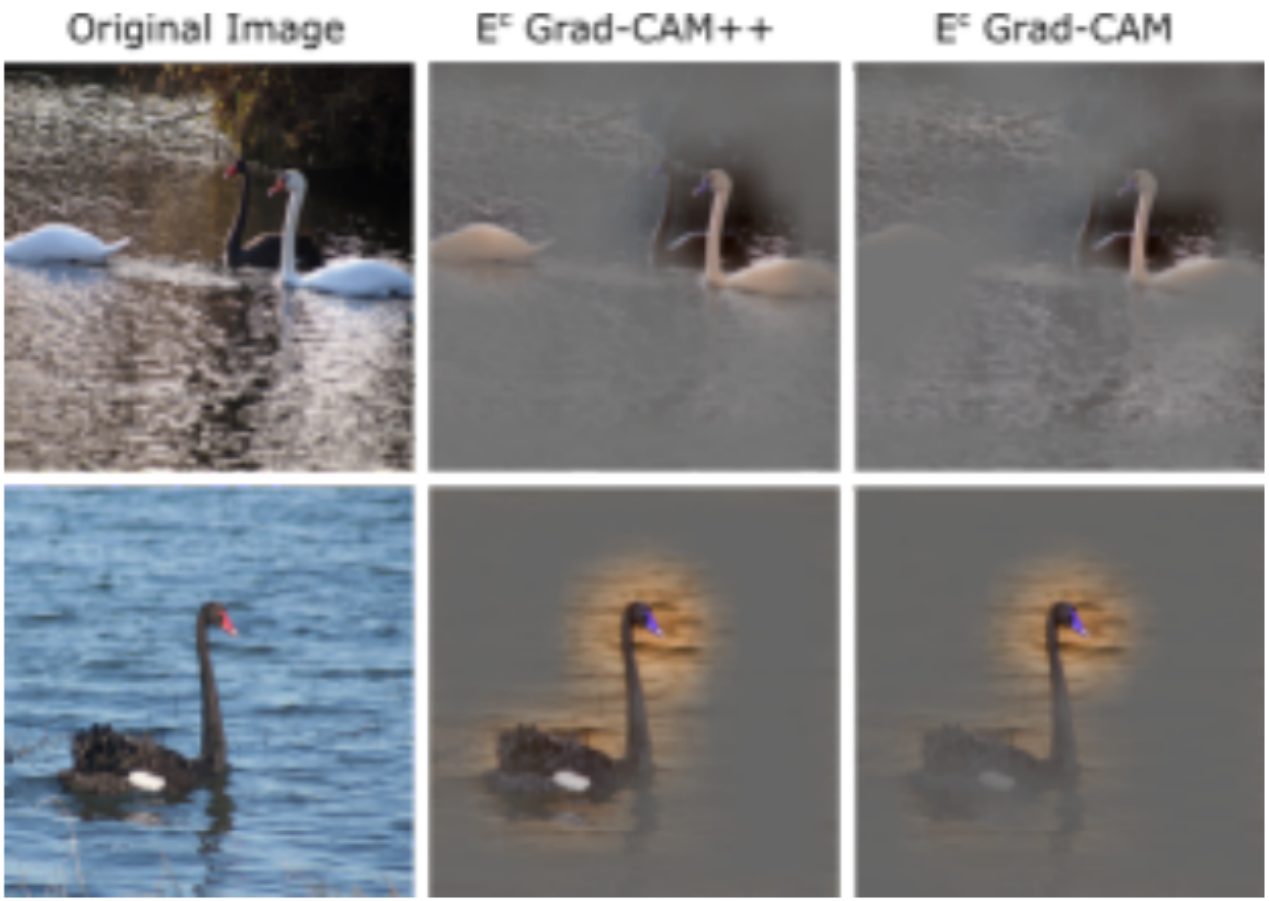
$$L_{\text{Grad-CAM++}}^c = \sum_k \alpha_k^c A^k$$

- Improved localization
- Handles multiple instances effectively

Grad-CAM++ Heatmap[6]



Grad-CAM++ Heatmap[6]



CLIP Similarity Score[7]

CLIP computes a similarity between an image I and a text T as:

$$S(I, T) = \langle M_{\text{img}}(I),\, M_{\text{text}}(T) \rangle$$

- $M_{\text{img}}(I) \in \mathbb{R}^d$: image embedding
- $M_{\text{text}}(T) \in \mathbb{R}^d$: text embedding
- $\langle \cdot, \cdot \rangle$: Euclidean inner product

Objective: explain which image regions contribute to $S(I, T)$.

Vision Transformer Structure[7]

The image encoder M_{img} is a Vision Transformer (ViT):

- Image split into N patches
- One additional **class token** (index $i = 0$)
- L transformer layers
- H attention heads per layer

Indices:

$$l \in \{1, \ldots, L\}, \quad h \in \{1, \ldots, H\}, \quad i \in \{0, \ldots, N\}$$

Residual Stream and Final Embedding

Let Z^l denote the residual stream at layer l. The final image embedding is obtained from the class token:

$$M_{\text{img}}(I) = P\,[Z^L]_{\text{cls}}$$

- P: learned linear projection
- $[Z^L]_{\text{cls}}$: class token at final layer

Unrolling the Residual Connections[7]

Each transformer layer updates the residual stream as:

$$Z^l = Z^{l-1} + \mathrm{MSA}^l(Z^{l-1}) + \mathrm{MLP}^l(\hat{Z}^l)$$

Unrolling across layers gives:

$$M_{\text{img}}(I) = P[Z^0]_{\text{cls}} + \sum_{l=1}^{L} P[\mathrm{MSA}^l(Z^{l-1})]_{\text{cls}} + \sum_{l=1}^{L} P[\mathrm{MLP}^l(\hat{Z}^l)]_{\text{cls}}$$

Decomposition into Attention Heads

The multi-head self-attention output can be written as:

$$P[\mathrm{MSA}^l(Z^{l-1})]_{\mathrm{cls}} = \sum_{h=1}^{H} \sum_{i=0}^{N} PW_O^{l,h} \left(\alpha_{\mathrm{cls},i}^{l,h} \, v_i^{l,h} \right)$$

- $\alpha_{\mathrm{cls},i}^{l,h}$: attention weight
- $v_i^{l,h}$: value vector
- $W_O^{l,h}$: output projection

Token–Head Contribution Vectors [10]

Define the contribution vector:

$$m_{i,l,h} = PW_O^{l,h} \left(\alpha_{\text{cls},i}^{l,h} \, v_i^{l,h} \right)$$

- $m_{i,l,h} \in \mathbb{R}^d$
- Represents how token i contributes through head h at layer l

TEXTSPAN Score Decomposition [10]

Define the scalar alignment:

$$A_{i,l,h}(T) = \langle m_{i,l,h}, M_{\text{text}}(T) \rangle$$

Then the CLIP score decomposes as:

$$S(I, T) = \sum_{l=1}^{L} \sum_{h=1}^{H} \sum_{i=0}^{N} A_{i,l,h}(T) + \epsilon$$

- ϵ: class-token initialization and MLP terms

TEXTSPAN Saliency Map [10]

Group contributions per head and layer:

$$A_{l,h} = \left[A_{i,l,h}\right]_{i=0}^{N}$$

Summing over heads and layers:

$$\mathrm{Sal}_{\mathrm{TEXTSPAN}}[i] = \sum_{l=1}^{L}\sum_{h=1}^{H} A_{i,l,h}$$

(Here we have just ('i') because token do not really have a 'i' and 'j')
CAM analogy:

$$\mathrm{Sal}_{\mathrm{CAM}}[i,j] = \sum_{k} w_k A_{ij}^{k}$$

Object vs Context Confusion

TEXTSPAN highlights:

- object evidence
- contextual cues that *suggest* the concept

Example:

- Snow $\Rightarrow$ polar bear
- Water $\Rightarrow$ boat

This leads to **concept hallucination**.

CHILI: Core Idea

Goal:

- Separate object-related activations
- From context-related activations

For each head and layer:

$$A_{l,h} = A_{l,h}^{\text{Object}} + A_{l,h}^{\text{Context}}$$

Removing Pseudo-Register Artifacts [7]

Vision Transformers contain high-norm global tokens.

Remove them via a median filter:

$$A_{l,h}^{\text{P.reg}} = A_{l,h} - f_m(A_{l,h})$$

Remaining activations are spatially meaningful.
Each head–layer pair receives a weight:

$$w_{l,h} \in [0, 1]$$

Computed using a probing dataset:

$$w_{l,h} = \mathbb{E}\left[1 - e^{-\alpha\,\mathrm{IoU}(h_m(A_{l,h}), G)}\right]$$

Final Decomposition [7]

$$A_{l,h}^{\text{Object}} = w_{l,h}\, f_m(A_{l,h})$$

$$A_{l,h}^{\text{Context}} = (1 - w_{l,h})\, f_m(A_{l,h})$$

Score decomposition:

$$S(I, T) = S^{\text{Object}} + S^{\text{Context}} + \epsilon$$

$$S_{\text{CHILI}} = S^{\text{Object}}$$

$$\text{Sal}_{\text{CHILI}}[i] = \sum_{l=1}^{L} \sum_{h=1}^{H} w_{l,h}\, f_m(A_{i,l,h})$$

Feature Formation in Vision Transformers [8]

An image is split into n patches and represented as tokens:

$$Z^0 = \{z_0^0, z_1^0, \ldots, z_n^0\} \in \mathbb{R}^{(n+1)\times d}$$

- z_0^0: class token ([CLS])
- z_i^0: patch tokens

Each layer updates tokens via:

$$\hat{Z}^l = \mathrm{MSA}^l(Z^{l-1}) + Z^{l-1}, \quad Z^l = \mathrm{MLP}^l(\hat{Z}^l) + \hat{Z}^l$$

Features are refined progressively across layers.

Layer-wise Predictions [8]

Let $Z^l = \{z_0^l, \ldots, z_n^l\}$ be the tokens at layer l.
Define the layer-wise pooled representation:

$$\bar{z}^l = \frac{1}{n+1} \sum_{i=0}^{n} z_i^l$$

A mapping model (classifier or text embedding):

$$\mathcal{C} \in \mathbb{R}^{d \times C}$$

Layer-wise prediction:

$$\bar{y}^l = \bar{z}^l \cdot \mathcal{C} \in \mathbb{R}^C$$

LeGrad explains how *patch interactions* influence $\bar{y}^l$.
Key insight:

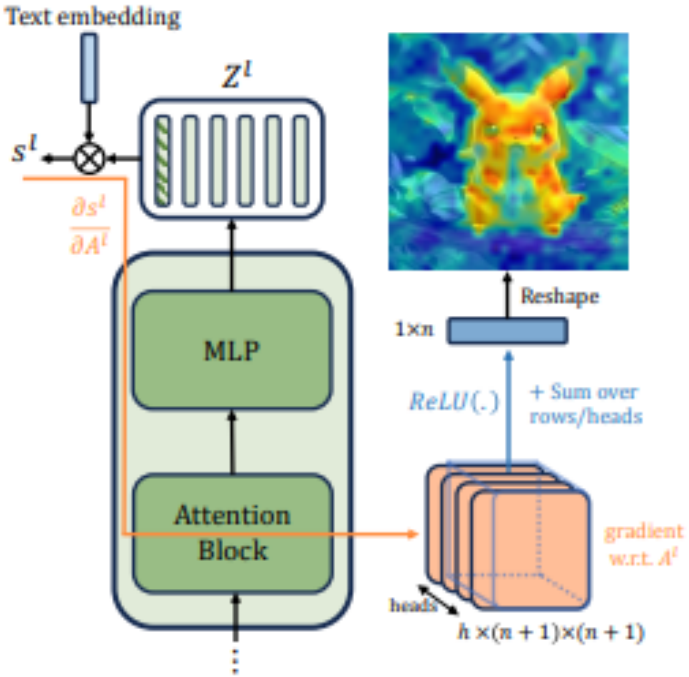
- Information flows between tokens via *self-attention*
- Attention maps determine how patches influence each other

LeGrad: Single-Layer Intuition [8]

For a target class or prompt $\hat{c}$:

$$s^l = \bar{y}^l_{[\hat{c}]}$$

LeGrad measures: *How sensitive is s^l to each attention connection?*



Gradients with Respect to Attention

Let:

$$\mathbf{A}^l \in \mathbb{R}^{h \times (n+1) \times (n+1)}$$

be the self-attention map at layer l.
Compute gradients:

$$\nabla \mathbf{A}^l = \frac{\partial s^l}{\partial \mathbf{A}^l}$$

- h: number of attention heads
- Measures importance of attention links

Negative gradients are discarded:

$$(\nabla \mathbf{A}^l)^+ = \max(\nabla \mathbf{A}^l, 0)$$

Layer-wise LeGrad Saliency[8]

Aggregate gradients across heads and query tokens:

$$\hat{E}^l = \frac{1}{h(n+1)} \sum_h \sum_i \left(\nabla \mathbf{A}^l_{h,i,:} \right)^+ \in \mathbb{R}^{n+1}$$

- One score per token
- Remove [CLS] token
- Reshape to image grid

Final layer-wise map:

$$E^l \in \mathbb{R}^{W \times H}$$

LeGrad Across Multiple Layers[8]

LeGrad computes a saliency map at *each layer*.
Average across layers:

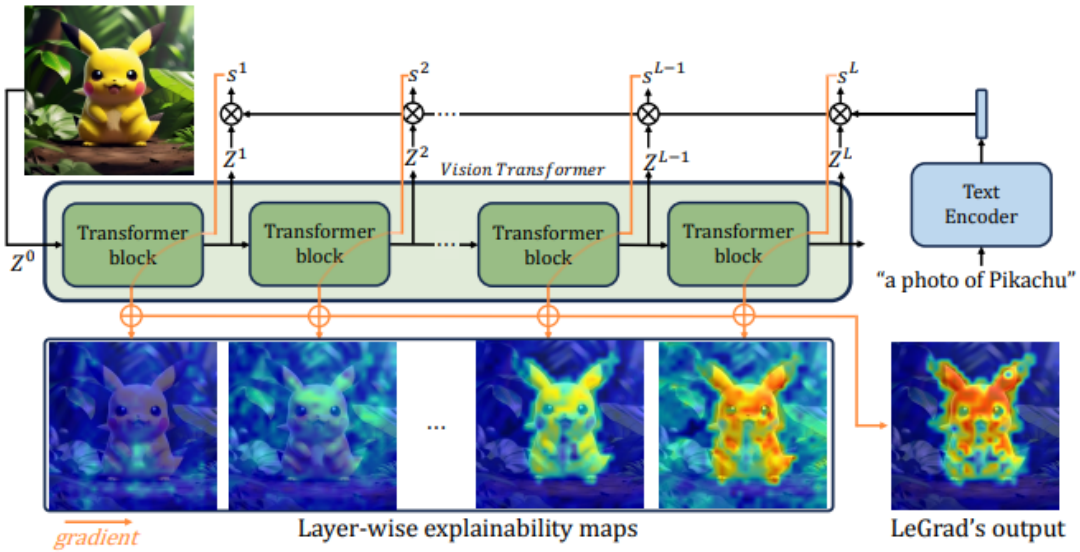
$$\bar{E} = \frac{1}{L} \sum_{l=1}^{L} \hat{E}^{l}_{1:}$$

Final saliency map:

$$\mathrm{Sal}_{\mathrm{LeGrad}}[j] = \frac{1}{h(n+1)L} \sum_{h} \sum_{i} \sum_{l=1}^{L} \left(\nabla \mathbf{A}^{l}_{h,i,:} \right)^{+} \in \mathbb{R}^{n+1}$$

This captures contributions accumulated throughout the transformer.

LeGrad's Principle[8]



Limits of Saliency-Based Explanations

What Saliency Methods Claim to Explain

- Saliency methods (Grad-CAM, Grad-CAM++, gradients) aim to answer:
  Which parts of the input influenced the model's prediction?
- They rely on gradients or gradient-like quantities:

$$\text{Saliency}(x) \propto \frac{\partial f(x)}{\partial x}$$

- Intuition:
  "If changing a pixel changes the output a lot, then that pixel must be important."

Observational vs Counterfactual Questions

There are two fundamentally different questions we can ask about a model:

Observational question

- "What happens to the output when the input changes slightly?"
- Measures *sensitivity*
- Typical tool: gradients

Counterfactual (causal) question

- "Would the model's prediction change if this feature were not used?"
- Measures *causal importance*
- Requires interventions, not just derivatives

What Gradients Actually Measure

Gradients compute:

$$\frac{\partial f(x)}{\partial x_i}$$

Interpretation:

- Local slope of the function at the current input
- How the output would change under an *infinitesimal perturbation*

Important:

- This is a purely **observational quantity**
- No feature is removed
- No causal intervention is performed

The Counterfactual vs Observational Confusion

Saliency methods implicitly assume:

> *"If the output is sensitive to a pixel, then the model must be using it."*

This assumption is **false**.

Why?

- Sensitivity does not imply causality
- A feature can be:
  - Causally important but have low gradient
  - Causally irrelevant but have high gradient

Example: Saturated Neurons

Consider a ReLU or sigmoid neuron:

- The neuron is fully activated (saturated)
- Small input changes do not change the output

Gradient-based explanation:

- Gradient ≈ 0
- Feature appears *unimportant*

Counterfactual reality:

- Removing the feature would completely change the prediction

Conclusion:

Low gradient $\neq$ low importance

Example: Spurious Sensitivity

- A background pixel slightly correlates with the class
- The model is sensitive to it locally
- Gradient is high

Saliency interpretation:

- Pixel appears important

Counterfactual test:

- Remove that pixel entirely
- Prediction barely changes

Conclusion:

High gradient $\neq$ *causal relevance*

Why Humans Are Misled by Saliency Maps

- Saliency maps look smooth and object-aligned
- They match human visual intuition
- But visual plausibility is not correctness

Empirical result:

- Randomized networks can produce similar saliency maps
- (Adebayo et al., 2018 — Sanity Checks)

Bibliography:

1. Itti, Laurent, Christof Koch, and Ernst Niebur. "A model of saliency-based visual attention for rapid scene analysis." IEEE Transactions on pattern analysis and machine intelligence 20.11 (2002): 1254-1259.
2. Bach, S., Binder, A., Montavon, G., Klauschen, F., Müller, K. R., & Samek, W. (2015). On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. PloS one, 10(7), e0130140.
3. Ha, M. L., Franchi, G., Moller, M., Kolb, A., & Blanz, V. (2018, March). Segmentation and shape extraction from convolutional neural networks. In 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) (pp. 1509-1518). IEEE.
4. Zhou, B., Khosla, A., Lapedriza, A., Oliva, A., & Torralba, A. (2016). Learning deep features for discriminative localization. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 2921-2929).
5. Selvaraju, Ramprasaath R., et al. "Grad-cam: Visual explanations from deep networks via gradient-based localization." Proceedings of the IEEE international conference on computer vision. 2017.

Bibliography:

6 Chattopadhay, Aditya, et al. "Grad-cam++: Generalized gradient-based visual explanations for deep convolutional networks." 2018 IEEE winter conference on applications of computer vision (WACV). IEEE, 2018.

7 Kazmierczak, Rémi, Steve Azzolin, Eloïse Berthier, Goran Frehse, and Gianni Franchi. "Enhancing Concept Localization in CLIP-based Concept Bottleneck Models." arXiv preprint arXiv:2510.07115 (2025).

8 Bousselham, Walid, et al. "Legrad: An explainability method for vision transformers via feature formation sensitivity." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2025.

9 Montavon, Grégoire, et al. "Layer-wise relevance propagation: an overview." Explainable AI: interpreting, explaining and visualizing deep learning (2019): 193-209.

Bibliography:

10 Gandelsman, Yossi, Alexei A. Efros, and Jacob Steinhardt. "Interpreting clip's image representation via text-based decomposition." arXiv preprint arXiv:2310.05916 (2023).