Explainable AI - Understanding and Interpreting Computer Vision Models

Gunti Lahari
IIT Hyderabad
ai22btech11008@iith.ac.in

J Hima Chandh
IIT Hyderabad
ai22btech11009@iith.ac.in

S Divija
IIT Hyderabad
ai22btech11026@iith.ac.in

K Anuraga Chandan
IIT Hyderabad
ai22btech11011@iith.ac.in

***Abstract*—Understanding how AI models make decisions is crucial in computer vision. In this project, we compared four explainable AI methods—Grad-CAM, Grad-CAM++, FullGrad-CAM, and LIME—across different tasks. Grad-CAM was used for multi-label classification, Grad-CAM++ and FullGrad-CAM for image classification, and LIME for object detection. Using datasets like ImageNet, COCO, and BDD-100K with models such as ResNet-50 and Xception, we evaluated each method based on faithfulness and plausibility. Our study offers insights into the strengths and limitations of each approach in making AI more interpretable.**

.

Keywords—XAI(Explainable AI),Grad-CAMGrad-CAM++, Human Attention Deep Learning , saliency map Computer vision,Black-box testingObject detection,Hierarchical masking

I. Literature Review

In this paper the review is continuing from paper[1]

A. Evaluation Metrics of FullGradCAM

Continuing from the preliminary project report, we now discuss the evaluation metrics used for FullGradCAM and FullGradCAM++.

1) Faithfulness Evaluation Methods: Faithfulness was measured using deletion and insertion methods:

- **Deletion:** Salient areas were gradually removed and replaced with random colors.
- **Insertion:** Salient areas were progressively added to a black background.

Both methods were performed in 100 steps, modifying 1% of the total area per step. The maximum modified area was constrained to the sum of detected bounding boxes for object detection models. The area under the insertion (i-AUC) and deletion (d-AUC) curves served as faithfulness metrics.

2) Plausibility Evaluation Methods: Plausibility was evaluated using human attention as a benchmark, measuring the similarity between XAI saliency maps and human attention maps. Two similarity metrics were used:

1) **Pearson Correlation Coefficient (PCC)** – A relative similarity measure that assesses the correlation between vectorized saliency maps.

$$PCC(u_1, u_2) = \frac{\bar{u_1}^T \bar{u_2}}{\sqrt{\bar{u_1}^T \bar{u_1} \cdot \bar{u_2}^T \bar{u_2}}}$$

2) **Root Mean Square Error (RMSE)** – An absolute similarity measure that calculates the error between saliency maps and human attention maps.

$$RMSE = \frac{1}{HW} \|u_1 - u_2\|_2$$

These metrics quantify how well the explanations align with human perception.

3) Their Conclusions: For image classification, Grad-CAM saliency maps showed higher faithfulness than human attention maps, as models and humans focus on different features. However, human attention during explanation tasks improved plausibility. This suggests integrating human attention into XAI enhances plausibility but not faithfulness.

For object detection, FullGrad-CAM and FullGrad-CAM++ outperformed Grad-CAM variants in plausibility, with FullGrad-CAM++ achieving the highest scores. Surprisingly, human attention maps demonstrated higher faithfulness than existing XAI methods, suggesting that models may use similar information extraction strategies as humans. Operations on gradients and smoothing techniques significantly impacted faithfulness and plausibility, motivating the development of human-guided XAI to enhance explanations for object detection models.

B. Human Attention-Guided XAI

This study introduces HAG-XAI, which leverages human attention to optimize activation and gradient maps for improved plausibility and faithfulness in XAI. It reweights feature maps, applies smoothing kernels, and uses learnable activation functions to align explanations with human perception.

1) Learnable activations and smoothing kernels:: They had considered adaptive piecewise linear activation function with two learnable parameters:

$$\varphi_{\alpha^-}^{\alpha^+}(\theta) = \alpha^+ \max(\theta, 0) + \alpha^- \min(\theta, 0),$$

where θ is the input activation map. The two learnable parameters α^+ and α^- allow for different scalings (or complete truncation) of the positive and negative parts of the activation, for ReLU $\alpha^+ = 1$ and $\alpha^- = 0$

During training, the learnable activation function selects and weighs key components by initializing activation map parameters (α^+ and α^-) to 1 (linear function) and gradient map parameters (β^+ and β^-) to 1 and 0 (ReLU function), respectively. Smoothing kernels enhance plausibility by refining

gradients and saliency maps, using a learnable 2D-Gaussian kernel (21×21 for object detection, 9×9 for classification) with adaptive variance and amplitude. The learnable Gaussian smoothing kernel is defined as:

$$G_A^v(x,y) = A\exp\left(\frac{-(x-x_c)^2+(y-y_c)^2}{2|v|+\varepsilon}\right),$$

Here, (x, y) represents spatial coordinates, $(x_c, y_c)$is the mean at half the kernel size,v is the learnable variance (initialized to 3 for object detection, 1 for classification), A is the learnable amplitude (initialized to 1), and ε prevents division by zero.

2) HAG-XAI::

$$S_{HI} = G_{A_\gamma}^{v_\gamma} * \sum_{m=1}^{N_{obj}} \bar{\mu}\Bigg(\text{ReLU}\left(\sum_{k=1}^{N_{ch}}\left(G_{A_\alpha}^{v_\alpha} * \varphi_{\alpha^-}^{\alpha^+}\left(\frac{\partial y^m}{\partial A^k}\right)\right) \odot \varphi_{\beta^-}^{\beta^+}\left(A^k\right)\right)\Bigg),$$

Note that the same kernel is applied to each channel of gradient and activation tensors.

Trainable Parameters: Only eight parameters are optimized.

- Loss Function: Minimizes dissimilarity between AI saliency maps and human attention maps using PCC & RMSE.
- Training:
  - Image Classification: Five-fold cross-validation on ImageNet subset.
  - Object Detection: Trained on BDD-100K, validated on MS-COCO.
  - Optimization: Adam optimizer with early stopping for object detection.
- Hyperparameters:
  - Learning rate: 0.05 (decayed to 0.005 in 120 epochs).
  - Mini-batch size: 30 (object detection), 144 (image classification).
  - Training epochs: 120 (object detection), 200 (image classification).
- Generalization: Tested on MS-COCO for object detection.

C. Discussion

This study examines whether embedding human attention knowledge into saliency-based XAI methods improves faithfulness and plausibility in image classification and object detection models. The findings reveal that while human attention enhances plausibility in classification models, it reduces faithfulness, as AI models focus on different features than humans. However, in object detection, human attention maps show higher faithfulness than existing XAI methods, enhancing both plausibility and faithfulness when integrated.

Further analysis of learned HAG-XAI parameters shows model-specific adaptations. For Yolo-v5s, a larger smoothing kernel was needed for gradients, while for Faster-RCNN, it was necessary to smooth grid-like patterns from ROI pooling. Additionally, both models benefited from incorporating negative activations and gradients, indicating counterfactual information's importance in explanations. These findings suggest that human attention can guide XAI to generate more meaningful and interpretable saliency maps, particularly for object detection tasks.

II. Implementations

We implemented FullGrad-CAM, Grad-CAM, and Grad-CAM++ for classification, as well as LIME and PODEM for object detection.

A. Data Preparation for XAI on Classification Task

1) **Downloading ImageNet Class Labels**: We retrieve ImageNet class labels from an online source to map numerical IDs to human-readable labels.
2) **Loading the ImageNet-Mini Dataset**: A subset of ImageNet is used to evaluate model performance and visualize class activation maps.
3) **Model Selection**: We employ ResNet-50 (from torchvision.models) and Xception (from the timm library), both pre-trained on ImageNet.
4) **COCO Dataset for multi label classification XAI**

B. FullGrad-CAM Implementation

FullGrad-CAM is an advanced visualization technique that enhances Grad-CAM by propagating element-wise gradients throughout the network. This leads to more precise and interpretable heatmaps, improving explainability in deep learning models. Our implementation consists of the following steps:

Algorithm 1 FullGrad-CAM Implementation

Require: Pretrained model M, input image I, target layer L_t, target class C_t

Ensure: Class Activation Map (CAM)

Step 1: Hook onto Target Layer

Define a forward hook to capture activations: $A = L_t(I)$

Define a backward hook to capture gradients: $G = \frac{\partial \text{Loss}}{\partial L_t}$

Step 2: Forward Pass

Compute model output:$O = M(I)$

Select target class C_t if not specified: $C_t = \arg\max(O)$

Step 3: Backward Pass

Compute loss for the target class: Loss $= O[C_t]$

Perform backpropagation to compute gradients G

Step 4: Generate FullGrad-CAM

Compute weighted activation map:

$$\text{CAM} = \sum G \cdot A$$

Apply ReLU to retain positive values:

Normalize CAM to [0,1] and resize to match I

Step 5: Visualization

return Class Activation Map (CAM)

By implementing FullGrad-CAM on standard image classification networks, we obtain more detailed feature visualizations that enhance model interpretability. The next sections discuss its comparison with other explainability methods.

C. GradCAM++ Implementation

GradCAM++ is an enhanced visualization technique that extends Grad-CAM by refining the weight computation of activation maps using higher-order derivatives. This method assigns different importance to different spatial locations, leading to sharper and more detailed heatmaps. GradCAM++ is particularly effective in highlighting multiple regions contributing to a prediction, improving model interpretability.

Algorithm 2 GradCAM++ Implementation

Require: Pretrained model M, input image I, target layer L_t, target class C_t
Ensure: Class Activation Map (CAM)
Step 1: Hook onto Target Layer
Define a forward hook to capture activations: $A = L_t(I)$
Define a backward hook to capture gradients:

$$G = \frac{\partial \text{Loss}}{\partial L_t}$$

Step 2: Forward Pass
Compute model output: $O = M(I)$
Select target class C_t if not specified:

$$C_t = \arg\max(O)$$

Step 3: Backward Pass
Compute loss for the target class: Loss $= O[C_t]$
Perform backpropagation to compute gradients G
Step 4: Compute GradCAM++ Weights
Compute sum of gradients across spatial dimensions:

$$\alpha = \sum G$$

Normalize gradients:

$$\alpha = \frac{G}{2 \cdot \alpha + \epsilon}$$

Apply ReLU to compute final weights:

$$W = \max(0, \alpha \cdot G)$$

Step 5: Generate GradCAM++ Map
Compute weighted activation map:

$$\text{CAM} = \sum W \cdot A$$

Apply ReLU to retain positive values:

$$\text{CAM} = \max(0, \text{CAM})$$

Normalize CAM to $[0, 1]$ and resize to match I
Step 6: Visualization
Overlay CAM onto the input image for interpretation
return Class Activation Map (CAM)

D. Grad-CAM Implementation

Grad-CAM is a visualization technique that enhances interpretability in deep learning models by highlighting important regions in an image that influence the model's prediction.
It works by computing gradients of a target class with respect to the feature maps of the last convolutional layer. These gradients are then used to generate a heatmap, which is overlaid on the original image to visualize the most critical areas.

Algorithm 3 Grad-CAM Implementation

Require: Pretrained model M, input image I, target convolutional layer L_t, target class C_t
Ensure: Class Activation Map (CAM)
Step 1: Preprocessing
Resize and normalize input image I
Convert to tensor and add batch dimension
Step 2: Hook onto Target Layer
Define a forward hook to capture activations: $A = L_t(I)$
Define a backward hook to capture gradients:

$$G = \frac{\partial \text{Loss}}{\partial L_t}$$

Step 3: Forward Pass
Compute model output:$O = M(I)$
Select target class C_t if not specified: $C_t = \arg\max(O)$
Step 4: Backward Pass
Compute loss for the target class: Loss $= O[C_t]$
Perform backpropagation to compute gradients G
Step 5: Generate Grad-CAM Heatmap
Compute global average pooled weights:

$$w_k = \frac{1}{Z} \sum_{i,j} G_{i,j}^k$$

Compute weighted sum of activations:

$$\text{CAM} = \sum_k w_k A^k$$

Apply ReLU to retain positive values:

$$\text{CAM} = \max(0, \text{CAM})$$

Normalize CAM to $[0, 1]$ and resize to match I
Step 6: Visualization
Convert CAM to a heatmap using a colormap (e.g., JET)
Overlay the heatmap onto the original image
Display the final Grad-CAM visualization
return Class Activation Map (CAM)

III. Review on Evaluation Metrics of XAI Methods

A. Evaluating Model Interpretations / Explanations

1) Evaluating the Meaningfulness or Correctness of Explanations
2) Evaluating the Interpretability of Explanations

B. Evaluating the Meaningfulness or Correctness of Explanations

There are various ways to evaluate the meaningfulness or correctness of explanations depending on the type of model interpretation.

C. Evaluating the Interpretability of Explanations

Evaluation Types: The following are the main evaluation types for interpretability:

1) **Application-grounded Evaluation** – Real humans, real tasks.
2) **Human-grounded Evaluation** – Real humans, simple tasks.
3) **Functionally-grounded Evaluation** – No real humans, proxy tasks.

The more specific and costly the evaluation, the higher it ranks.

D. Functionally-grounded Evaluation

Quantifiable metrics – Example: Number of rules, prototypes (lower is better).

E. Human-grounded Evaluation

1) **Binary Forced Choice** – Users choose between two explanations.
2) **Forward Simulation / Prediction** – Given input and explanation, users predict the output.
3) **Counterfactual Simulation** – Given input and explanation, users determine feature changes required to impact the prediction.

F. Application-grounded Evaluation

Domain Expert Involvement – Experts perform exact, similar, or partial tasks.

G. Evaluating Post hoc Explanations

1) **Evaluating the faithfulness (or correctness) of post hoc explanations.**
2) **Evaluating the stability of post hoc explanations.** (If we perturb the input, how much does the explanation change?)
3) **Evaluating the fairness of post hoc explanations.** (The accuracy of explanations should be roughly similar for majority and minority groups.)
4) **Evaluating the interpretability of post hoc explanations.**

H. Evaluating Faithfulness

The accuracy of explanations should be roughly similar for majority and minority groups.)

1) **Ground truth (associated with the model, not data) is available.**
   (Example: top K features that the model is using when making these kinds of predictions.)
   1. Feature agreement
   2. Rank agreement
   3. Sign agreement
   4. Signed Rank Agreement
   5. Rank correlation
   6. Pairwise rank agreement

   (1)
2) **Explanations as Models**
   a) If the explanation is itself a model (e.g., linear model fit by LIME), we can compute the fraction of instances for which the labels assigned by the explanation model match those assigned by the underlying model.

   What if we do not have any ground truth? What if explanations cannot be considered as models that output predictions?
3) **How Important are Selected Features?**
   Deletion: Remove important features (as designated by explanation) and see what happens. (How does prediction probability drop?)
   - In image datasets, removal can involve replacing regions with background means.
   - If background deletion does not occur naturally, perturbing pixel values can be an alternative.

   Insertion: Add important features and observe the effect.

I. Evaluating Stability of Post hoc Explanations

Are post hoc explanations unstable with small input perturbations?

- Local Lipschitz constant of explanation
- Maximum relative change of prediction (explanation method) w.r.t. input change

What if the underlying model itself is unstable?

1) **Relative output stability**:
   - Denominator accounts for changes in prediction probabilities.
2) **Relative representation stability**:
   - Denominator accounts for changes in the intermediate representations of the underlying model.

J. Evaluating Fairness of Post hoc Explanations

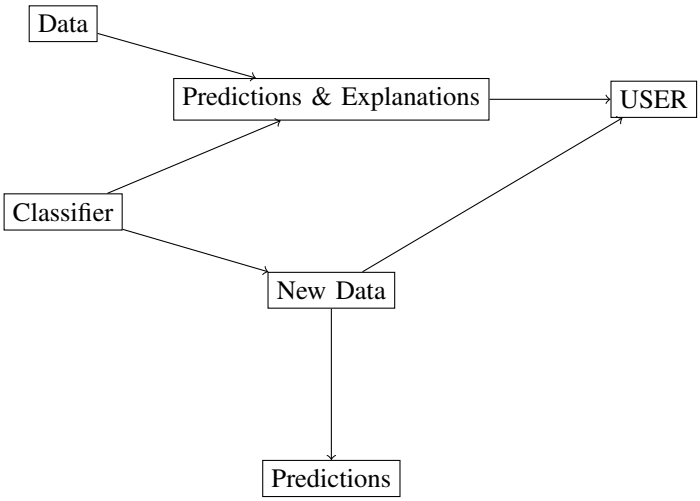
1) Compute mean faithfulness/stability metrics for instances from majority and minority groups. (e.g., race A vs. race B, male vs. female)
2) If the difference between the two means is statistically significant, then there is unfairness in the post hoc explanation.

Why/when can such unfairness occur?

1) When the model is linear in some parts, non-linear in other parts, but the explanation method tries to fit a linear model at all points.
2) Data insufficient

K. Evaluating Interpretability

Predicting Behavior ("Simulation")



L. Challenges of Evaluating Interpretable Models / Post hoc Explanation Methods

1) **Evaluating interpretations/explanations** is an ongoing endeavor.
2) **Parameter settings** heavily influence the resulting interpretations/explanations.
3) **Diversity of explanation/interpretation methods** leads to diverse metrics.
4) **User studies are not consistent**, affected by:
   - Choice of UI
   - Phrasing
   - Visualization
   - Population
   - Incentives
5) **All the above lead to conflicting findings.**

M. Empirically/Theoretically Analyzing Interpretational Explanations

Limitations

1) Faithfulness:

- Even when parameters at different layers are randomized, explanations remain the same (gradient-based).
- Explanations are biased independent of model behavior.
- Randomizing class labels of instances also didn't impact explanations.

2) Stability:

- Are post-hoc explanations unstable with small non-adversarial input perturbations?
- Perturbation approaches like LIME can be unstable.
- Running LIME on the same instance repeatedly gives non-converging explanations (different explanations every time).

N. FullGrad-CAM and GradCAM++ on Resnet-50

To analyze the interpretability of deep learning models, we compare FullGrad-CAM and GradCAM++ visualizations applied to ResNet-50.

IV. Results

A. FullGrad-CAM and GradCAM++ on Resnet-50

To analyze the interpretability of deep learning models, we compare FullGrad-CAM and GradCAM++ visualizations applied to ResNet-50. Figure 1 presents heatmaps generated using these two methods.

The results demonstrate that FullGrad-CAM produces more focused heatmaps, whereas GradCAM++ offers better coverage of multiple salient regions.

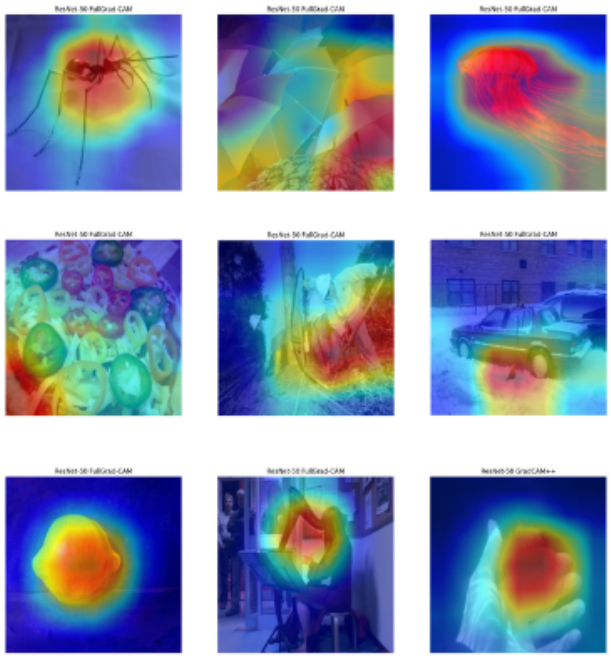


Fig. 1. heat maps by FullGrad-CAM (first 8 images) and GradCAM++ (9th image) on ResNet-50 The visualizations highlight the most important regions in the input images that contribute to the model's predictions. FullGrad-CAM propagates element-wise gradients throughout the network, leading to more precise and interpretable heatmaps. GradCAM++, on the other hand, refines the weighting of activation maps to better capture class-discriminative regions. These techniques enhance the explainability of deep learning models by providing insights into their decision-making process.

B. FullGrad-CAM and GradCAM++ on Xception

To further evaluate the explainability of deep learning models, we applied FullGrad-CAM and GradCAM++ to the Xception architecture. The generated visualizations are presented in Figure 2.

Overall, these findings suggest that FullGrad-CAM's performance is architecture-dependent, and additional tuning may be required to optimize its results for models like Xception.

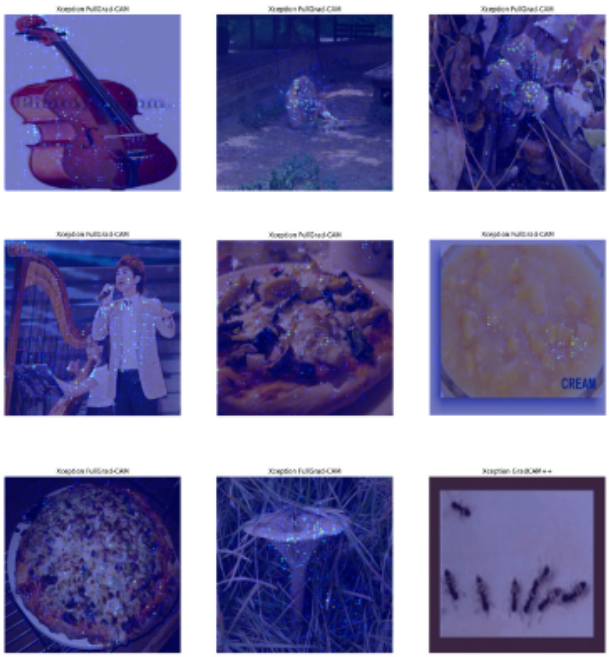


Fig. 2. heat maps by FullGrad-CAM (first 8 images) and GradCAM++ (9th image) on Xception The first 8 images represent FullGrad-CAM results, while the last image illustrates GradCAM++ results. Compared to ResNet-50, FullGrad-CAM on Xception produces lower contrast activation maps, whereas GradCAM++ maintains sharper and more interpretable attention regions.

C. Comparision between FullGradCAM and GradCAM++

Below Figure 7. is the comparision on Resnet-50

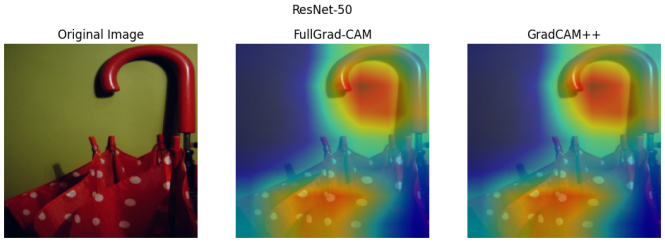


Fig. 3. FullGradCAM vs GradCAM++ on Resnet-50

Below Figure 6. is the comparision on Xception

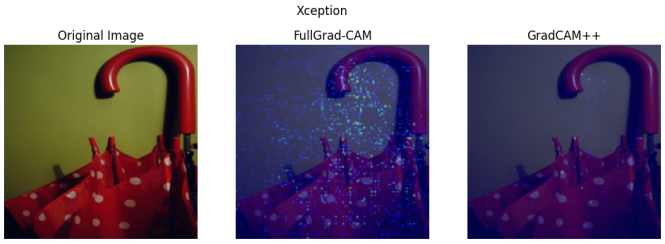


Fig. 4. FullGradCAM vs GradCAM++ on Xception

D. LIME

To analyze the interpretability of the object detection model, we applied the Local Interpretable Model-agnostic Explanations (LIME) framework to a YOLO model trained on the COCO dataset. LIME provides insights into which regions of an image influence the model's predictions by perturbing the input and evaluating changes in output confidence.

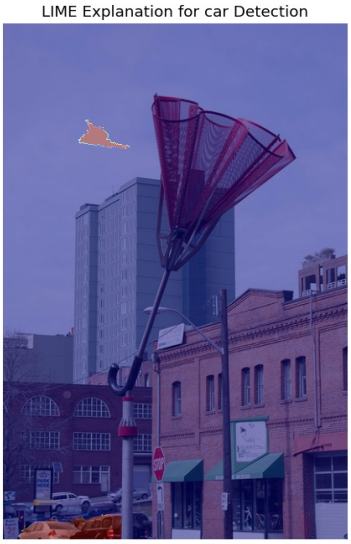


Fig. 5. LIME-based explanation for a YOLO object detection model, highlighting the regions contributing to the detection of a car. Some unrelated areas are also highlighted, indicating possible model misinterpretation.

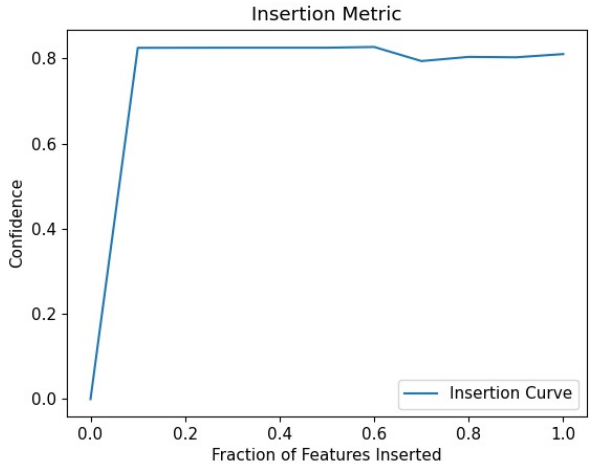


Fig. 6. Insertion metric curve showing the model's confidence as important features are gradually inserted. The steep rise indicates that a small subset of features significantly impacts the decision.

To further evaluate the explanation's reliability, we computed the insertion score, which measures the model's confidence as relevant features are progressively added. A sharp rise in confidence at the start suggests that a few key features strongly influence the detection.
These findings highlight the importance of explainability in object detection, especially in applications where model transparency is critical. Future work could explore methods like Grad-CAM or SHAP to compare and improve the interpretability of YOLO's predictions.

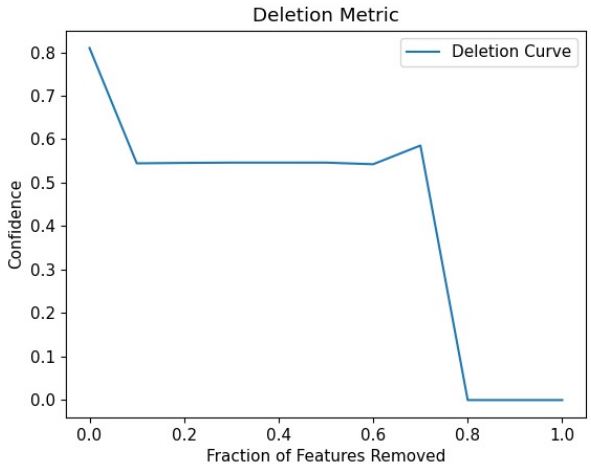


Fig. 7. "Deletion metric curve showing the confidence of the object detection model as important features are progressively removed. A sharp decline in confidence, especially towards the end, indicates that the removed features were crucial for the model's decision-making."

- ***Deletion AUC: 0.426***
- ***Insertion AUC: 0.775***

E. GradCAM

Fig. 8. GradCAM for multilabel Classification

In this we can see that when label is table the blue region is all over the table and when it is bottle it is a part on the table where it is present

REFERENCES

[1] Guoyang Liu, Jindi Zhang , Antoni B. Chan,Janet H. Hsiao "Human Attention-Guided Explainable Artificial Intelligence for Computer Vision Models" arXiv:2305.03601v1 [cs.CV] 5 May 2023

[2] Milad Moradi, Ke Yan,David Colwell,Matthias Samwald,Rhona Asgari "Model-agnostic explainable artificial intelligence for object detection in image data"Tricentis GmbH, Leonard-Bernstein-Straße 10, 1220 Vienna, Austria

[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," in Proceedings of the IEEE international conference on computer vision, Conference Proceedings, pp. 618–626.

[4] https://www.sciencedirect.com/org/science/article/pii/S1526149224003084

[5] https://github.com/jacobgil/pytorch-grad-cam?tab=readme-ov-file

[6] https://github.com/GitVirTer/HAG-XAI

[7] https://open-xai.github.io/