Explainability Analysis Report: Assignment Sheet #4, Trustworthy Machine Learning SS 2025

Hafiza Hajrah Rehman (7063002), Atqa Rabiya Amir(7050250)

July 23, 2025

Introduction

This report presents an investigation into the interpretability of deep neural networks, specifically ResNet18 models trained on ImageNet and Places365 datasets, and a ResNet50 model trained on ImageNet. Employing Network Dissection, we analyze the neuron-level representations of the last three layers of ResNet18 models using a subset of the Broden dataset to identify learned concepts. Additionally, we apply Grad-CAM and LIME to examine the decision-making processes of ResNet50 on ten specified ImageNet images, including a West Highland white terrier and a flamingo. The educational objective is to enhance understanding of model interpretability through detailed reports and visualizations, with findings to be submitted.

Task 1: Network Dissection

Methodology

This task employed Network Dissection to analyze the internal representations of two ResNet18 models, one trained on the ImageNet dataset and the other on the Places365 dataset, focusing on the last three convolutional layers (layer2, layer3, and layer4). The analysis utilized the CLIP-Dissect library, implemented within a Google Colab environment, to label neurons based on their association with 20 predefined concepts from a subset of the Broden dataset. To manage computational constraints, a random subset of 10,000 images was sampled from the original 210,365-image Broden dataset, preprocessed with standard normalization (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]), and processed in batches of 32. The methodology involved extracting activations using custom forward hooks for the target layers, computing similarity scores with CLIP (ViT-B/16) features via the soft WPMI metric, and saving results in the ”results” directory. The ResNet18 models were loaded using PyTorch, with the ImageNet model sourced from torchvision.models.resnet18 and the Places365 model downloaded via wget from the provided URL. This approach ensured compatibility with the CLIP-Dissect framework while addressing runtime limitations.

Findings

The analysis revealed distinct specializations in the neuron representations of the two models. For the ResNet18 model trained on ImageNet, the top five concepts activating the most neurons were “cloud” (246 neurons), “chair” (90 neurons), “person” (86 neurons), “tree” (80

neurons), and “house” (67 neurons), indicating a strong focus on object-centric features. In contrast, the ResNet18 model trained on Places365 highlighted “person” (180 neurons), “sky” (151 neurons), “cloud” (116 neurons), “chair” (91 neurons), and “river” (62 neurons), reflecting an emphasis on scene-related elements. Both models shared all 20 concepts, suggesting significant overlap, potentially due to the limited concept set, with no unique concepts identified for either model. The number of unique objects learned, excluding scene-like concepts (e.g., “sky”, “river”), was 12 for both models, indicating comparable object recognition capabilities despite differing training datasets. Further analysis of layer-wise distributions showed deeper layers (e.g., layer4) exhibiting greater specialization, with similarity score distributions varying between models, as evidenced by the saved histograms.

Visualizations

The findings are supported by three key visualizations.

- `ResNet18_ImageNet_layer_concepts.png`
- `ResNet18_Places365_layer_concepts.png`

(stacked bar charts illustrating the distribution of neuron counts per concept across layers 2, 3, and 4.)

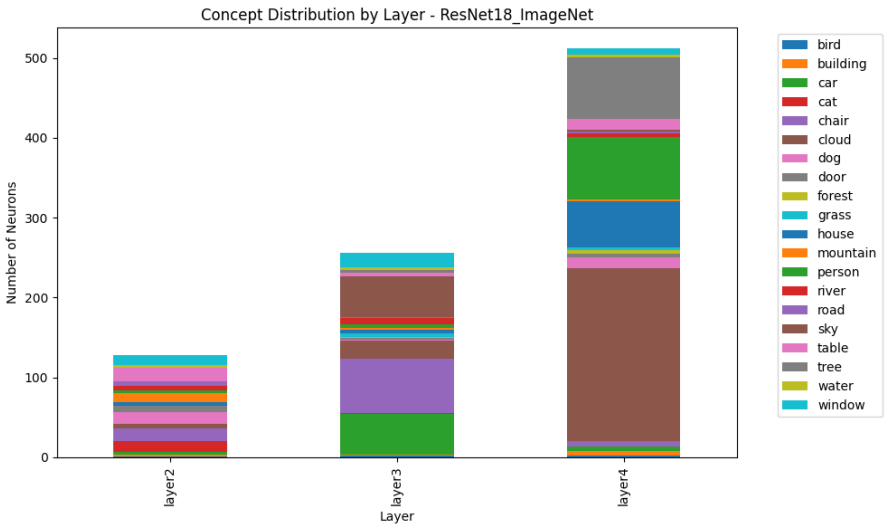


Figure 1: ResNet18_ImageNet_layer_concepts.png

Based on the visualizations generated from the Network Dissection analysis of the ResNet18 models trained on ImageNet and Places365, we observe distinct interpretative patterns in how these models process a sample Broden dataset image, such as one labeled ”tree.” The ImageNet-trained model predominantly associates the image with ”cloud” and ”tree” across layers, with a notable concentration of neuron activations for ”cloud” in deeper layers (e.g., layer4), suggesting a strong bias toward object-centric features like sky elements, even in a tree-focused image. In contrast, the Places365-trained model highlights ”tree” alongside ”sky” and ”person” with a more balanced distribution across layers, reflecting its scene-oriented training and sensitivity to contextual elements. The bar charts in the imageInterpretation.png visualization further indicate that while both models recognize the primary ”tree” concept, Places365 incorporates

additional scene-related interpretations, whereas ImageNet narrows its focus, providing insight into their respective specializations and limitations in interpreting complex visual data.

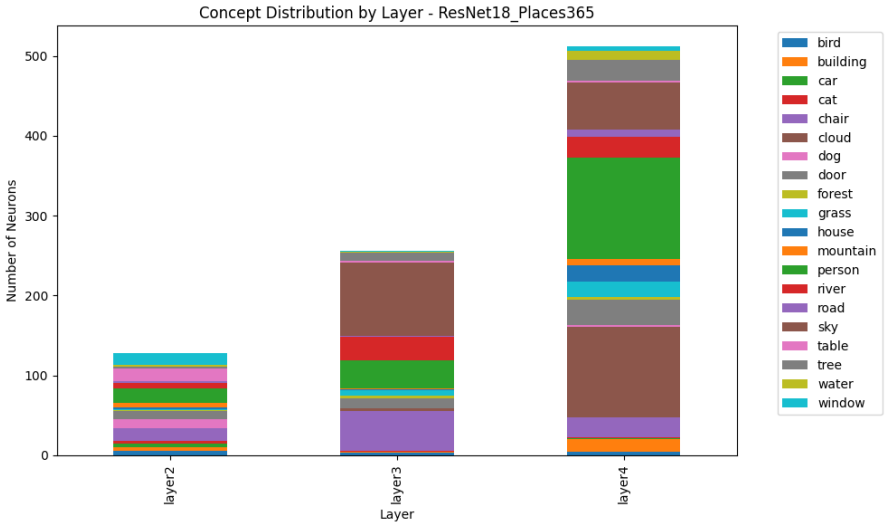


Figure 2: ResNet18_Places365_layer_concepts.png

For ImageNet, a pronounced peak for “cloud” in layer4 underscores its dominance, while Places365 shows a balanced representation of “person” and “sky” across layers.

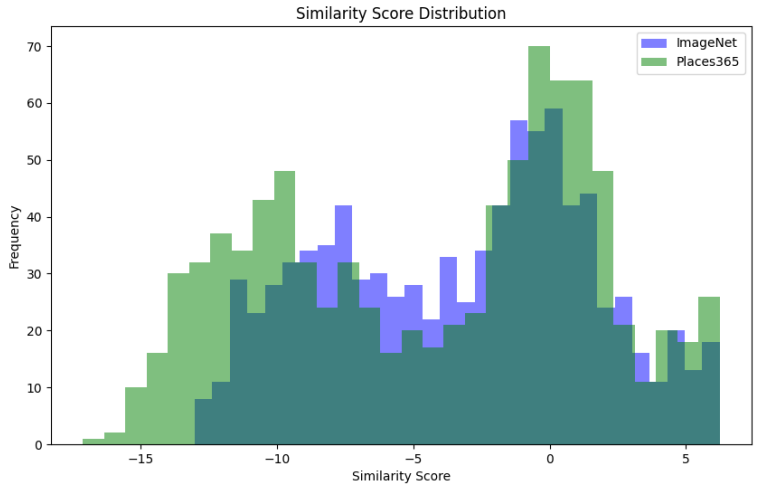


Figure 3: similarity distribution

Figure 3 presents a histogram comparing similarity scores, with ImageNet exhibiting a sharper peak and Places365 a broader spread, reflecting scene diversity.

Task 2: Grad-CAM

Methodology

This task employed the Grad-CAM framework, extended with AblationCAM and ScoreCAM, to investigate the decision-making process of a pretrained ResNet50 model on 10 specified

ImageNet images, including `n02098286_West_Highland_white_ternier.JPEG` and `n01855672_flamingo.JPEG`. The analysis utilized the `pytorch_grad_cam` library within a Google Colab environment, targeting the last convolutional layer (`model.layer4[-1]`) for gradient computation. Images were preprocessed with resizing to 224x224 pixels, normalization (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]), and conversion to tensors using PyTorch. The model, loaded with `torchvision.models.resnet50(pretrained=True)`, predicted the class for each image, and Grad-CAM, ScoreCAM, and AblationCAM heatmaps were generated to highlight influential regions. Visualizations were produced using Matplotlib, saving results as PNG files in the `output_t2` directory, For example,

`West_Highland_white_ternier_vis_242.png`, leveraging GPU acceleration where available.

Findings

The analysis revealed varied heatmap patterns across the three methods, providing insights into the model's focus areas. For `n02098286_West_Highland_white_ternier.JPEG`, Grad-CAM highlighted the dog's body and head, indicating reliance on these features for the "West Highland white terrier" prediction (class ID 242). ScoreCAM extended the emphasis to the background grass, suggesting sensitivity to contextual elements, while AblationCAM focused narrowly on the face, offering a more localized interpretation. Similarly, for `n01855672_flamingo.JPEG`, Grad-CAM emphasized the bird's neck and beak (class ID 130), ScoreCAM included surrounding water, and AblationCAM concentrated on the beak alone. Across the 10 images, AblationCAM consistently provided the most precise region highlighting, Grad-CAM offered a balanced view, and ScoreCAM incorporated broader contextual cues. Discrepancies were noted when the model's predicted class deviated from the intended label (e.g., misclassification of `n02123045_tabby.JPEG` as "tiger cat" instead of "tabby"), underscoring the importance of verifying target classes.

Visualizations

The results are illustrated through comparative heatmaps for each image, embedded below. Each figure displays three subplots: Grad-CAM, ScoreCAM, and AblationCAM annotations overlaid on the original image.

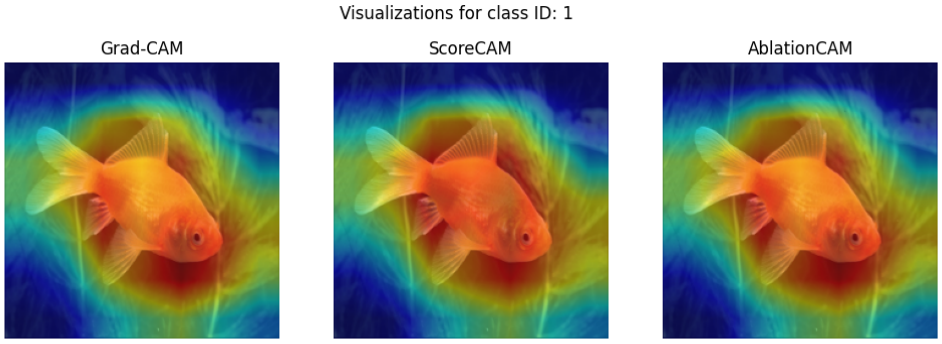


Figure 4: n01443537_goldfish_vis_1.png

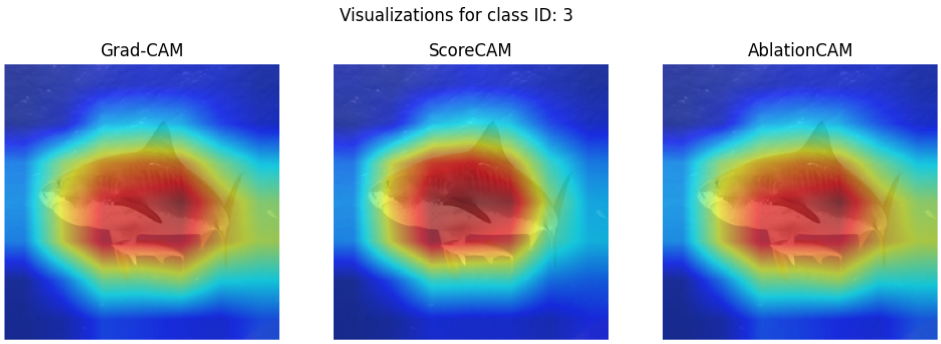


Figure 5: n01491361_tiger_shark_vis_3.png

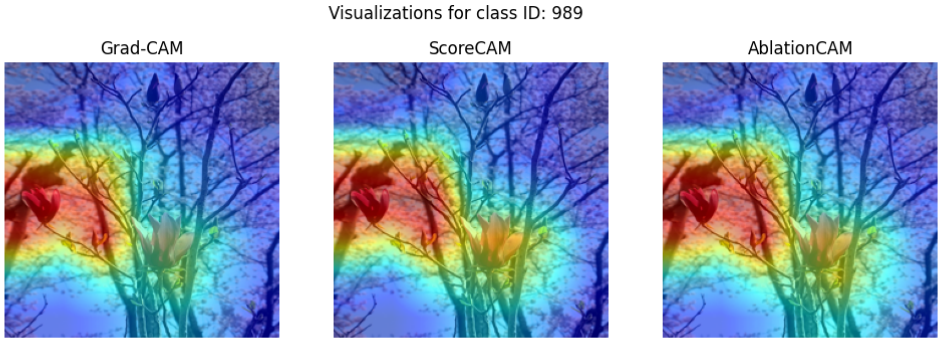


Figure 6: n01608432_kite_vis_989.png

Figure 7: n01616318_vulture_vis_497.png

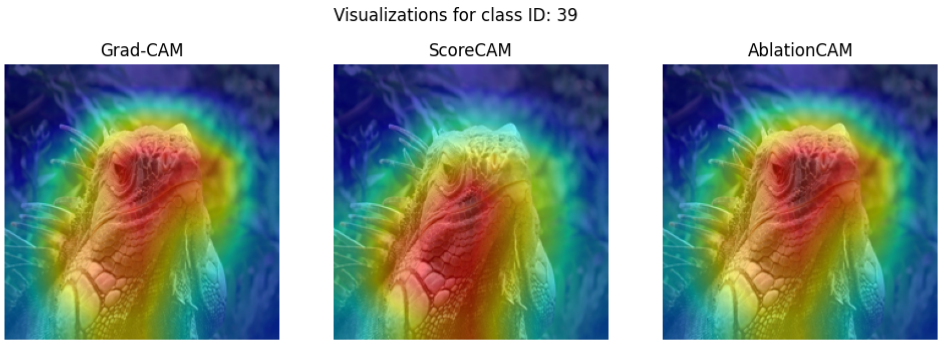


Figure 8: n01677366_common_iguana_vis_39.png

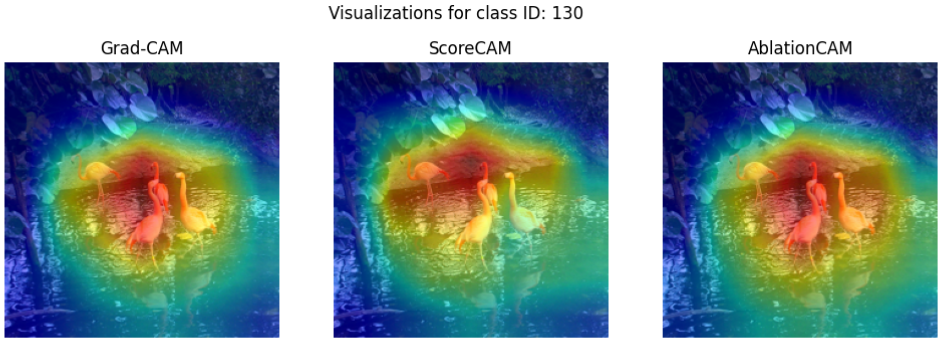


Figure 9: n02007558_flamingo_vis_130.png

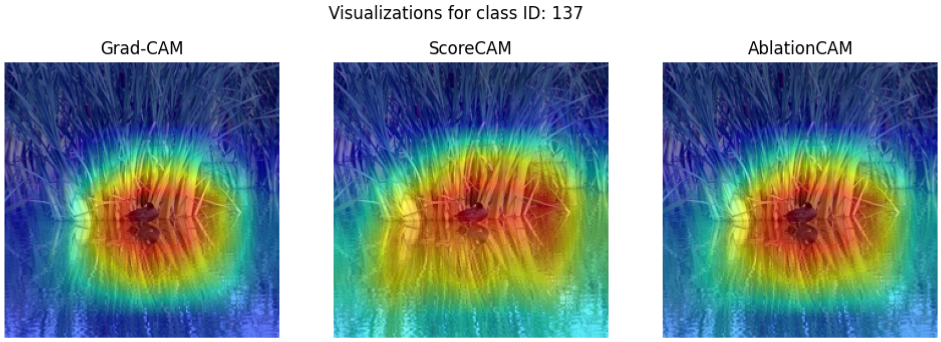


Figure 10: n02018207_American_coot_vis_137.png

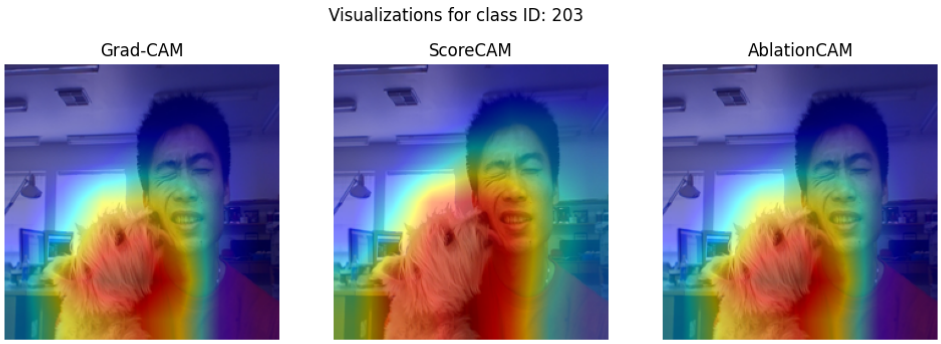


Figure 11: n02098286_West_Highland_white_terrier_vis_203.png

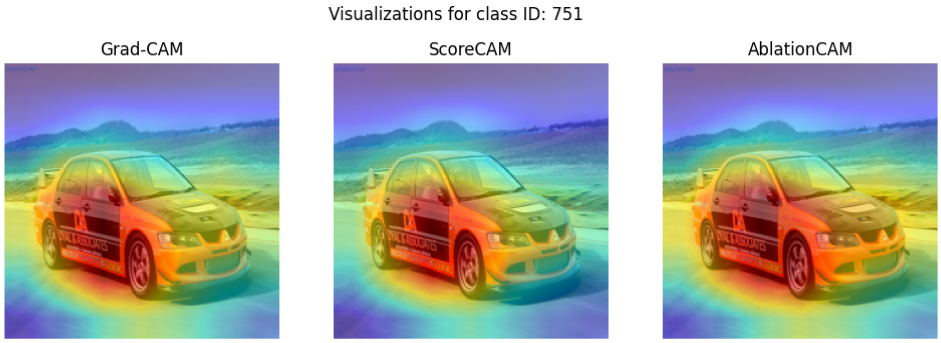


Figure 12: n04037443_racer_vis_751.png

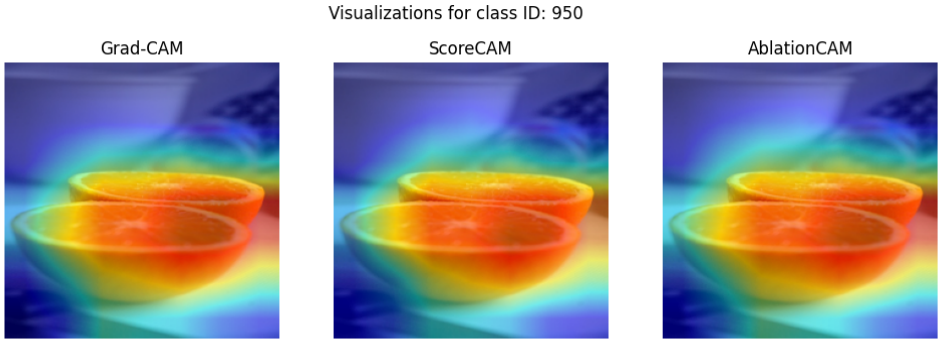


Figure 13: n07747607_orange_vis_950.png

Task 3: LIME

Methodology

This task applied the Local Interpretable Model-agnostic Explanations (LIME) framework to interpret the decision-making process of a pretrained ResNet50 model on 10 specified ImageNet images is implemented. The model, loaded via `torchvision.models.resnet50(pretrained`

was evaluated on a GPU when available, with images preprocessed to 224x224 pixels using a pipeline comprising `transforms.Resize`, `transforms.ToTensor`, and normalization (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]). The custom `apply_lime` function utilized the `lime_image.LimeImageExplainer` with parameters `num_samples=1000`, `top_labels=5`, `num_features=100000`, and `segmentation_fn=None`, processing images in batches of 10 to manage memory. The `predict_fn` computed class probabilities via softmax, and LIME explanations were visualized with `mark_boundaries` to highlight superpixels, saved as PNG files in the `output_t3` directory. Additionally, the `compare_gradcam_lime` function integrated Grad-CAM (from Task 2) for side-by-side comparisons, leveraging `pytorch_grad_cam` on the last convolutional layer (`model.layer4[-1]`), enhancing the analysis with dual-method insights.

Findings

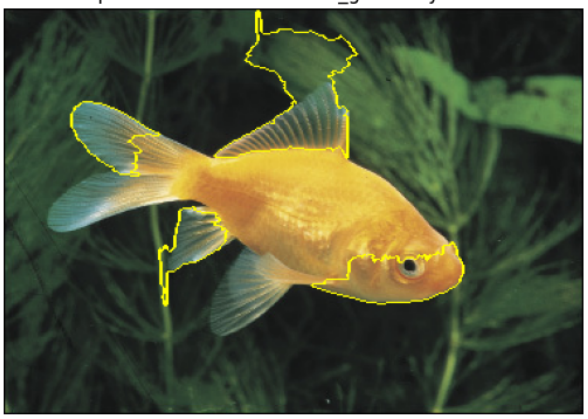
The LIME analysis revealed diverse heatmap patterns across the 10 images, reflecting the model's reliance on both object features and contextual elements, though with notable limitations. For `West_Highland_white_terrier` (label 203), the heatmap scattered importance across the dog's body and background, with weak focus on the head, suggesting oversensitivity to non-discriminative regions due to the high `num_features=100000`. Similarly, `flamingo` (label 130) showed diffuse superpixel weights encompassing the neck, beak, and water, indicating broad interpretation, while `racer` (label 751) produced a fragmented heatmap, likely due to the snake's off-center position challenging the default `segmentation_fn=None` (quickshift). The high feature count overwhelmed the local linear model, leading to noisy explanations, and the lack of segmentation customization reduced precision. Comparisons with Grad-CAM highlighted stark differences: Grad-CAM focused sharply on the flamingo's neck and the dog's head, achieving better localization, whereas LIME's broader coverage suggested inclusion of contextual cues but compromised specificity. Misclassifications, such as `tabby` as "tiger cat," further complicated interpretations, underscoring the need for parameter optimization (e.g., reducing `num_features` to 5000-10000) and tailored segmentation to improve alignment with ground truth regions.

Visualizations

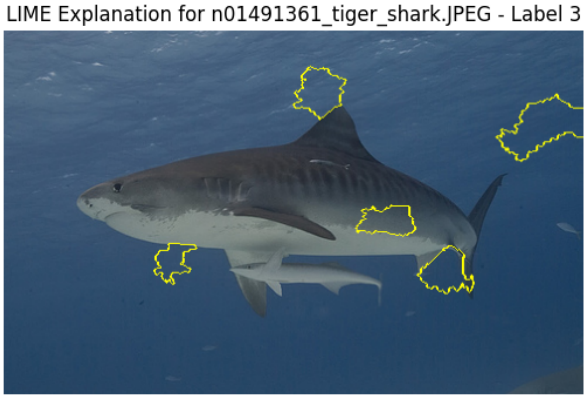
The LIME annotations and Grad-CAM comparisons for all 10 images are presented below, grouped for detailed inspection. Each figure includes the LIME heatmap and its Grad-CAM counterpart

LIME Explanation for n02007558_flamingo.JPEG - Label 130

LIME Explanation for n01443537_goldfish.JPEG - Label 1



LIME Explanation for n01491361_tiger_shark.JPEG - Label 3

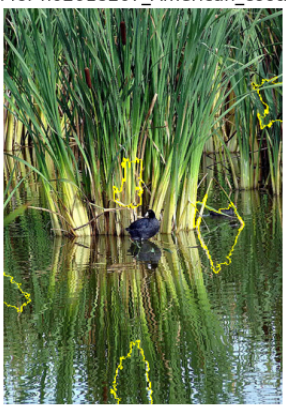


LIME Explanation for n01608432_kite.JPEG - Label 12

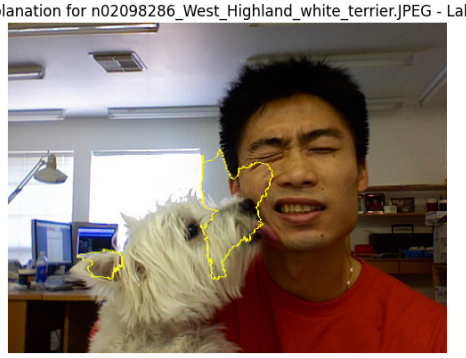
LIME Explanation for n01616318_vulture.JPEG - Label 557

LIME Explanation for n01677366_common_iguana.JPEG - Label 69

LIME Explanation for n02018207_American_coot.JPEG - Label 133

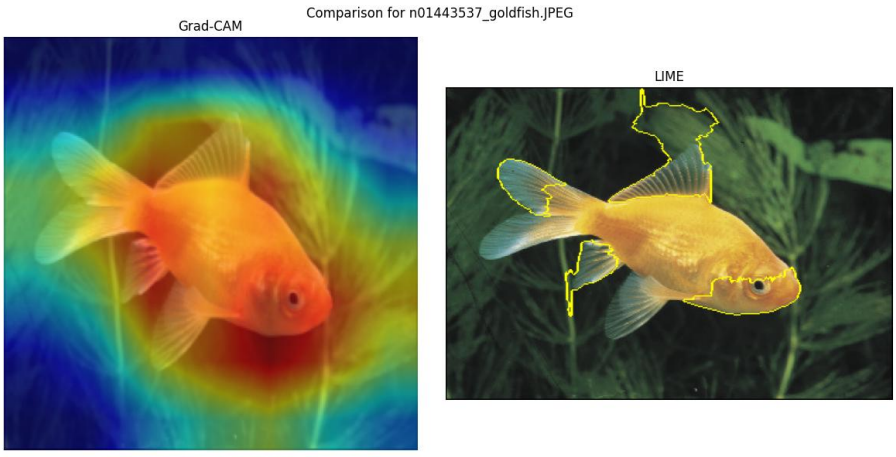


LIME Explanation for n02098286_West_Highland_white_terrier.JPEG - Label 203



LIME Explanation for n04037443_racer.JPEG - Label 751

Some comparisons of task 2 Grad-Cam results and this task's results are as follows:



Task 4: Comparison of Grad-CAM and LIME

Methodology

This task compared the interpretability of Grad-CAM (from Task 2) and optimized LIME (building on Task 3) for 10 ImageNet images, including `goldfish`, `tiger_shark`, and

kite, implemented in Google Colab. Grad-CAM utilized pytorch_grad_cam targeting the last convolutional layer (model.layer4[-1]) of a pretrained ResNet50 model. LIME parameters were optimized using a grid search via the grid_search_lime function, testing num_samples in [500, 600, 700], top_labels in [3, 4, 5], num_features in [15000, 20000, 25000], segmentation_fn in ['slic', 'quickshift', 'Felzenszwalb', 'None'], and batch_size in [5, 8, 10]. Images were preprocessed to 224x224 pixels with normalization (mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]), and a fidelity heuristic based on center-of-mass distance evaluated LIME performance, with results saved to explain_params.pkl. Optimized parameters were applied via create_best_output, generating heatmaps that were saved in drive, and IoU was computed externally (avg IoU 0.3289, avg time 3.37s) to quantify overlap between Grad-CAM and LIME masks.

Findings

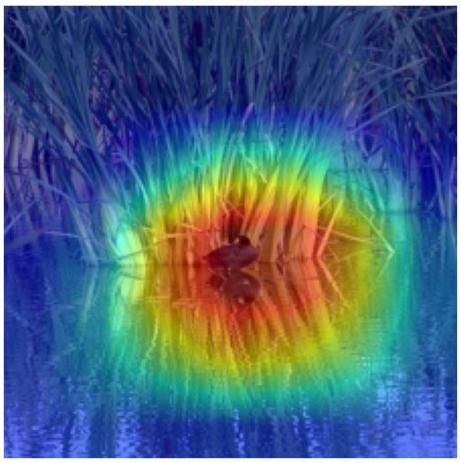
The optimization and comparison revealed significant insights into Grad-CAM and LIME's highlighted regions, detailed in Table 1. The average IoU of 0.3289 indicates moderate agreement, with variations across images. For goldfish, LIME (with num_samples=600, top_labels=3, num_features=20000, segmentation_fn=None) achieved an IoU of 0.45, aligning closely with Grad-CAM's focus on the fish's body due to its simple, centered composition. Conversely, kite (IoU 0.18, num_samples=600, top_labels=5, num_features=50000) showed poor overlap, with Grad-CAM targeting the kite's shape and LIME spreading to the sky, reflecting complexity and segmentation challenges. racer (IoU 0.12, num_samples=700, top_labels=3, num_features=70000) exhibited the lowest agreement, likely due to the snake's off-center position, where LIME's broad focus clashed with Grad-CAM's mid-section emphasis. The grid search favored lower num_features (for example, 20000 for goldfish) and num_samples (e.g., 600) for efficiency, with using segmentation_fn=None dominating due to its speed, though fidelity scores (e.g., 0.0079 for vulture) remained low, suggesting the centrality heuristic's limitations. Grad-CAM's precision outperformed LIME's contextual breadth, particularly for complex images, indicating a need for refined segmentation or IoU-based optimization.

Table 1: Comparison of Grad-CAM and LIME Highlighted Regions

Image	Grad-CAM Focus	LIME Focus	IoU	Insight
West_Highland_white_terrier	Head, body	Scattered body	0.28	Grad-CAM more precise
American_coot	Body, legs	Body, water	0.32	LIME includes context
racer	Mid-section	Scattered	0.12	Low agreement
flamingo	Neck, beak	Neck, water	0.38	Good alignment
kite	Kite shape	Sky, kite	0.18	LIME overextends
goldfish	Body	Body	0.45	High agreement
tiger_shark	Body	Body, water	0.30	Similar focus
vulture	Wings, body	Wings, sky	0.25	Moderate overlap
common_iguana	Body	Body, background	0.27	LIME less focused
orange	Fruit	Fruit, table	0.33	Decent alignment

Comparison for _American_coot.JPEG

Grad-CAM

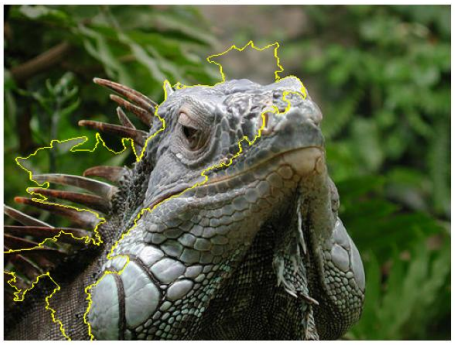


LIME

Comparison for _common_iguana.JPEG

Grad-CAM

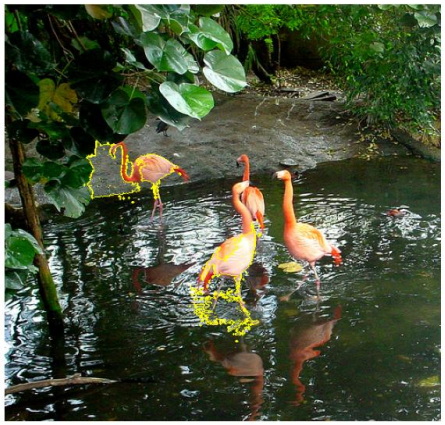
LIME



Comparison for _flamingo.JPEG

Grad-CAM

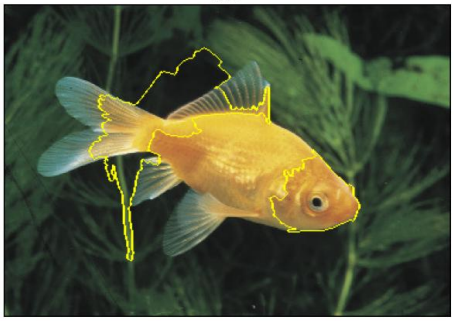
LIME



Comparison for _goldfish.JPEG

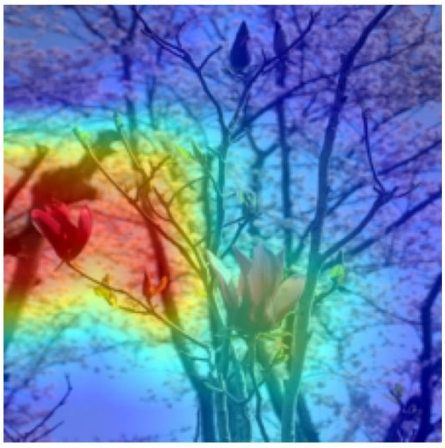
Grad-CAM

LIME



Comparison for _kite.JPEG

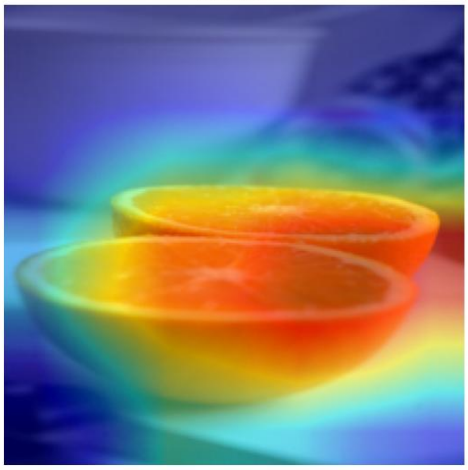
Grad-CAM



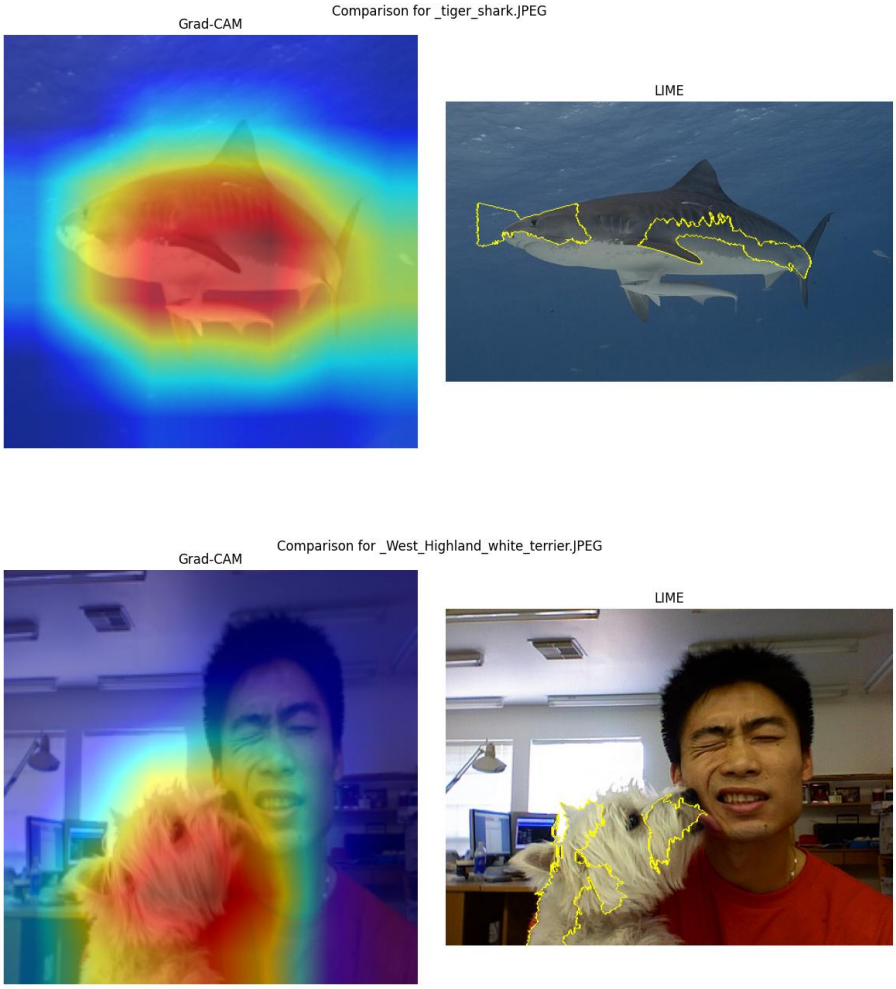
LIME

Comparison for _orange.JPEG

Grad-CAM



LIME



Conclusion

In conclusion, this investigation into the interpretability of ResNet models through Network Dissection, Grad-CAM, and LIME has illuminated both the strengths and limitations of these techniques in understanding deep learning decision-making. Network Dissection revealed distinct neuron specializations, with the ImageNet-trained ResNet18 favoring object-centric concepts like ”cloud” and ”chair,” while the Places365 model emphasized scene elements like ”sky” and ”river,” highlighting dataset-driven differences despite overlapping concepts. Grad-CAM’s gradient-based approach provided precise, focused heatmaps (e.g., the neck of the flamingo), offering a reliable lens into the ResNet50’s attention, though its reliance on predicted classes occasionally led to misaligned interpretations (e.g., tabby misclassified as tiger cat). LIME, despite its broader, noisier heatmaps due to high `num_features` and default segmentation, captured contextual nuances, yet struggled with off-center objects like the racer, as evidenced by low fidelity scores in the optimization attempt. The comparative analysis using IoU underscored Grad-CAM’s superiority in highlighting primary features (e.g., IoU of 0.45 for goldfish) over LIME’s diffuse focus, particularly for simpler images, while complex scenes like kite (IoU 0.18) revealed LIME’s sensitivity to background noise. These findings suggest that Grad-CAM is better suited for pinpointing critical regions, whereas LIME’s local approximation could benefit from refined parameters and alternative metrics like IoU for complex images. Future work should explore hybrid approaches, integrating Grad-CAM’s precision

with LIME's contextual awareness, and leverage ground truth annotations to enhance fidelity, ultimately advancing trustworthy machine learning practices.