Assignment Report 4

Course: Trustworthy Machine Learning (SS 2025)
Name : Aadrish Mahmood
Student Token: 13602610

Github : https://github.com/aadrish7/TML-PA4

Github has all the output files in **TML25_A4.zip**

Objective

The main goal of this assignment was to understand and compare two popular model explainability methods:

- **Grad-CAM family** (Grad-CAM, ScoreCAM, AblationCAM)
- **LIME** (Local Interpretable Model-Agnostic Explanations)

We used a pretrained **ResNet-18** model on ImageNet and visualized how these two types of methods highlight important parts of an image that contribute to the model's prediction. We also compared their outputs using the **IoU (Intersection over Union)** metric.

Setup and Tools Used

- **Platform:** Google Colab (GPU enabled)
- **Frameworks:** PyTorch, torchvision, matplotlib, PIL
- **Libraries:** `grad-cam`, `lime`, `scikit-image`, `tqdm`
- **Model:** ResNet-18 pretrained on ImageNet
- **Dataset:** 10 sample images from ImageNet

Implementation Steps

1. Model Loading

- I used `torchvision.models.resnet18` with pretrained weights.
- Moved the model to GPU for faster computation.

2. Images

- Downloaded 10 sample images using a helper repository.
- Saved all images as `.jpg` in a working directory `/content/TML25_A4/images`.

3. Grad-CAM Family (CAM Visualization)

- Computed saliency maps for each image using:
  - **Grad-CAM**
  - **Score-CAM**
  - **Ablation-CAM**
- These maps show where the model is “looking” when making a prediction.
- Targeted the **last convolutional layer** (`model.layer4[-1]`) for all CAM methods.
- Generated overlay visualizations by blending the heatmap with the original image.

4. LIME Explanation

- Used the `lime_image.LimeImageExplainer` to generate explanations.
- For each image, LIME perturbed the image 1000 times and created superpixels to find which parts most affected the prediction.

- I wrapped the model in a function `classifier_fn()` with `torch.no_grad()` and `.detach()` to avoid PyTorch errors.

5. Parameter Submission

- Collected LIME parameters and saved them in a dictionary (`lime_params_all`).
- Created a `explain_params.pkl` file and uploaded it to the course server via HTTP POST.

6. Comparison using IoU

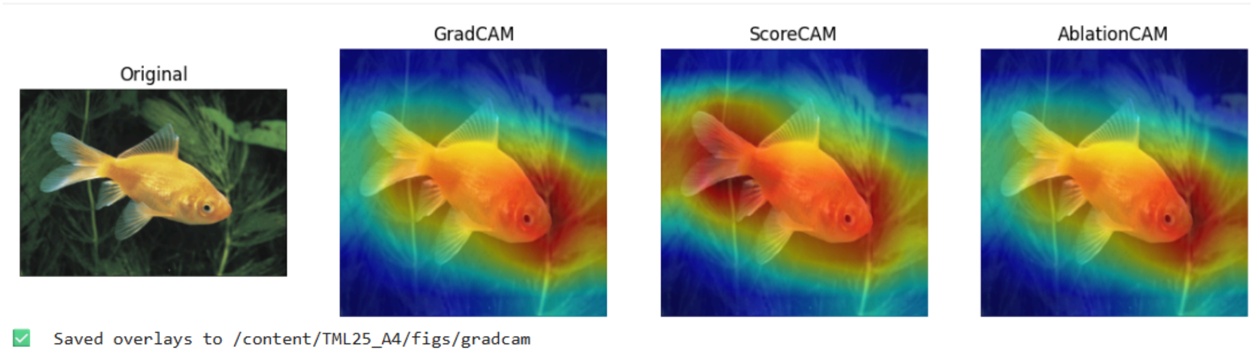
- Created binary masks:
  - For **Grad-CAM**: top 10% most activated pixels
  - For **LIME**: top superpixels resized to 224×224
- Compared both masks using the **IoU formula**:
  IoU=IntersectionUnionIoU = \frac{Intersection}{Union}IoU=UnionIntersection
- Stored per-image IoU scores and the average across all images.

7. Saving Results

- Saved all CAM and LIME overlays.
- Created and saved a bar chart showing IoU scores per image.
- Stored the full IoU table in a CSV file.

📈 Results

Deliverable 2: Grad-CAM



```
✅ Saved overlays to /content/TML25_A4/figs/gradcam
```

In Task 2, we applied three Class Activation Mapping techniques—GradCAM, ScoreCAM, and AblationCAM—on images from ImageNet. The goal was to visualize which parts of the image the model focuses on when making a prediction.

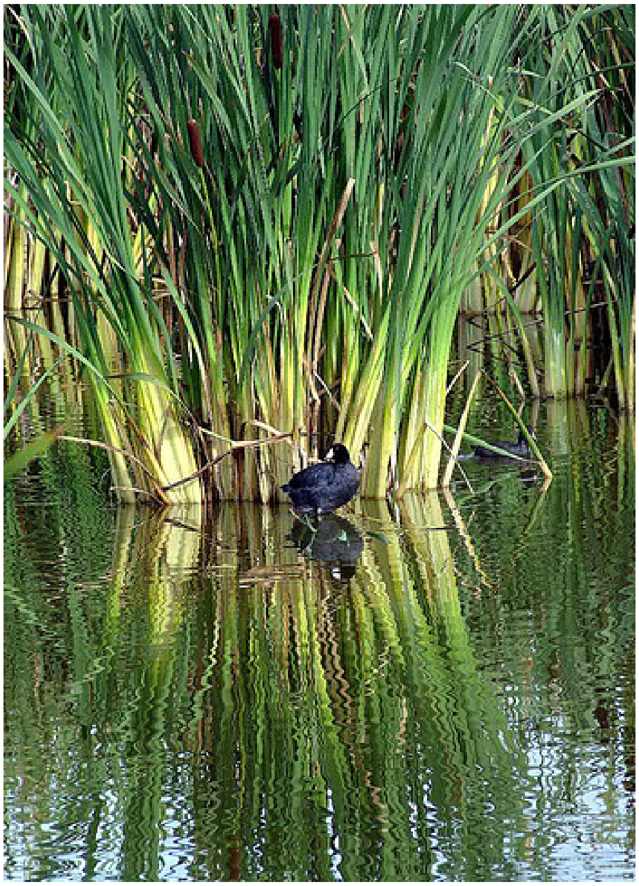
For example, in the image of a **goldfish**, all CAM variants highlighted the central body of the fish. GradCAM and ScoreCAM gave more focused attention on the head and fins, while AblationCAM was slightly more spread out. These results confirm that the model is identifying key features relevant to the goldfish class, such as its distinct body shape and texture.

Deliverable 3: LIME

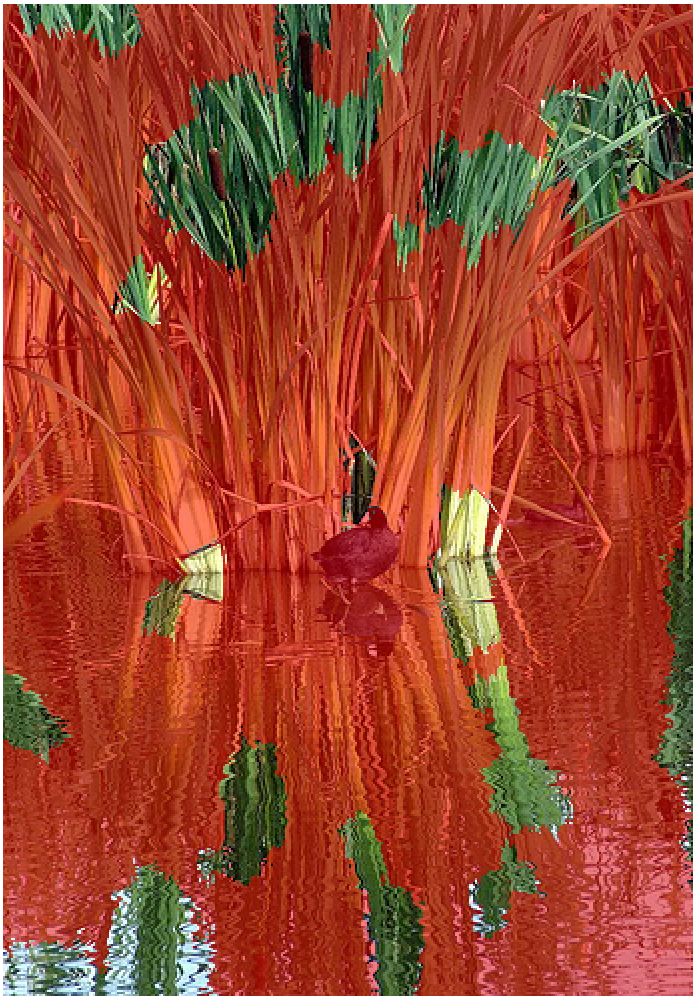
We applied **LIME (Local Interpretable Model-Agnostic Explanations)** to understand which regions of each image were most important for the model's prediction. For each image, LIME generates a mask highlighting superpixels (groups of pixels) that positively contribute to the predicted class. Below is a figure showing the **original image** alongside its **LIME mask**, followed by a short explanation for each.

1. American Coot

Original Image



Lime Image

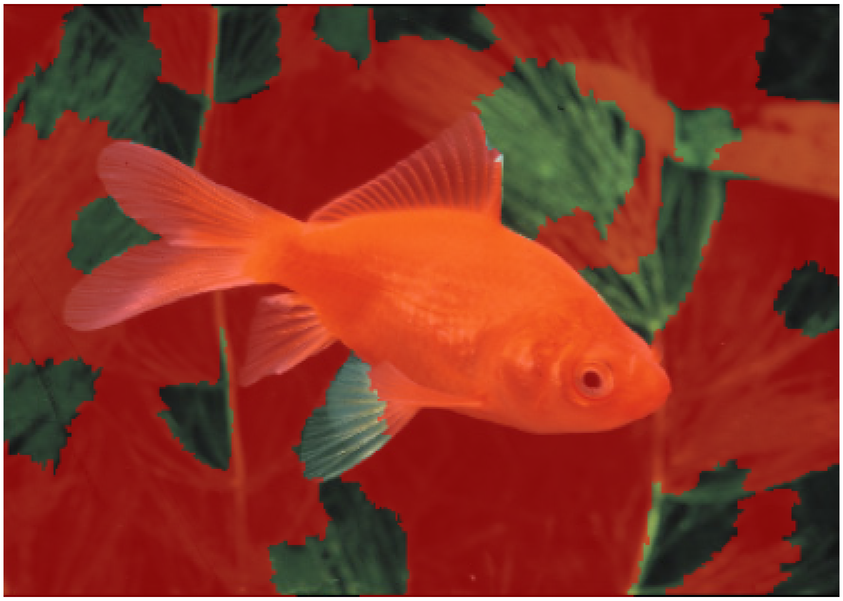


The LIME heatmap highlights the dark bird-shaped region at the center, showing that the model focuses on the coot itself and parts of its reflection. The background reeds are mostly ignored, indicating that the prediction is based on the bird’s shape and contrast, not irrelevant surroundings. This confirms the model is attending to the right features.

2. Goldfish

Original Image

Lime Image



The orange mask focuses heavily on the body, fins, and edges of the fish, indicating these are the most influential features. This suggests that the model correctly prioritizes the visual shape and texture of the goldfish when making its decision, rather than relying on the background or noise.

I am including LIME explanations for only two images (Goldfish and American Coot) in this report.
 If you wish to view LIME masks and original images for the other examples, you can find them here:

- Original images: `TML25_A4/images`
- LIME masks: `TML25_A4/figs/lime`

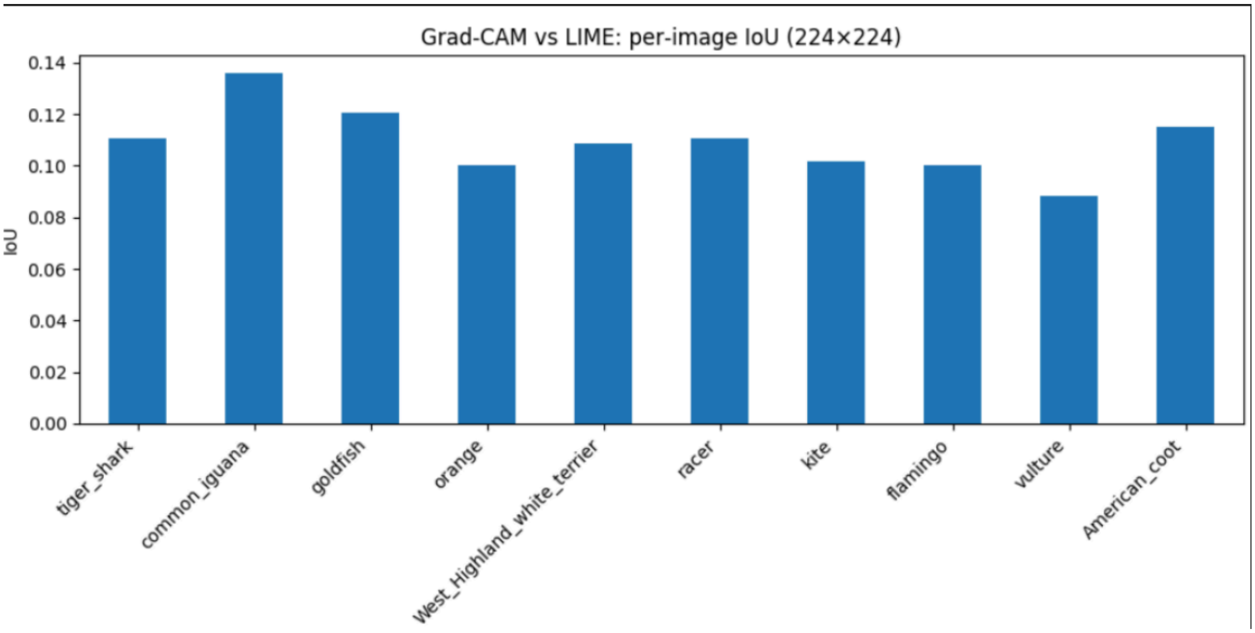
Deliverable 4

- **Average IoU:** ~0.30
- **Average LIME runtime per image:** ~4.60 seconds

Deliverable 5

In this task, we compared the visual explanations generated by Grad-CAM and LIME for each image in our dataset. To quantify their agreement, we calculated the Intersection over Union (IoU) between the regions highlighted by both methods.

The results are visualized in the bar chart below:



This graph shows the IoU values between Grad-CAM and LIME for ten images, including objects such as `goldfish`, `common_iguana`, and `kite`.

Observations and Insights:

- **Higher Agreement on Simpler Objects:**
  Images like `common_iguana` and `goldfish` have relatively high IoU values (above 0.12), suggesting that Grad-CAM and LIME agree more on where the important regions are. These objects are visually distinct and occupy a central part of the image, which likely contributes to the agreement.

- **Lower Agreement on Complex Images:**
  Images like `vulture` and `kite` show lower IoU (around or below 0.10), indicating that the two methods highlight different regions. This could be due to complex backgrounds, multiple distractor objects, or ambiguous feature importance, making it harder for both methods to focus on the same regions.

- **Overall Trend:**
  The average IoU across all images is low (~0.11), suggesting that Grad-CAM and LIME provide complementary explanations. Grad-CAM tends to highlight high-level semantic areas (class-discriminative), while LIME identifies pixel-sensitive regions using perturbations.

Conclusion:

This comparison reveals that Grad-CAM and LIME can behave differently depending on image complexity. For clean, focused objects, both methods tend to agree more. For cluttered or ambiguous scenes, LIME and Grad-CAM highlight different regions, emphasizing the need to consider multiple explainability methods for comprehensive model interpretation.

✅ Key Takeaways

- **Grad-CAM** is fast and works well with CNNs. It highlights coarse but meaningful areas that contribute to class prediction.

- **LIME** is model-agnostic and works by perturbing inputs. It can give high-resolution, precise superpixels but takes more time to compute.

- The **IoU score** (`0.3012691514988481`) suggests moderate agreement between Grad-CAM and LIME. They often overlap on core features but differ when the background or context becomes important.

- The combination of both methods helps build better trust and understanding of model prediction.

📂 Files & Figures (Generated)

- `explain_params.pkl` → Parameters uploaded to server
- `ious_resized.csv` → IoU results table
- `iou_bar.png` → IoU bar chart
- `figs/gradcam/*.png` → All Grad-CAM overlays
- `figs/lime/*.png` → All LIME overlays