# Assignment-4

Shivam Balwani Barno Kaharova

## I. Task-1: Clip Dissect

In this task we analysed the internal representation of the neurons in the last 3 layers i.e. fc. layer4, layer3 of ResNet18 model which is trained on two seperate dataset Imagenet and Places365. We used Network Dissect using Clip Dissect library to label neurons with the concept they have learned. The primary objective of this was to identify and compare the concepts learned on both models, how learning concepts looks like on different layer.

### A. Data and Methodology

By using the `describe_neurons.py` [8] script from the CLIP-dissect repository [1], we labeled each neuron in the specified layers of the models. The script provides labels based on a predefined set of concepts, assigning the most relevant label to each neuron based on its activation patterns.

After we had the description of what concepts and similarity related to each concept on each layer and neurons learns, we used these to plot multiple plots to get a better understanding. We used the reference implementation for obtaining the representation [2].

### B. Visualization

In this report we are providing two visualization we created to understand the concept learned and draw some conclusion and get the better understanding how neurons and layer learn

- We plotted bar graphs showing top 20 frequent neuron concept labels learned by each layer in both models
- We plotted scatter plots depicting the similarity scores of neuron activations across the top 20 concepts for each model.

Here are the above two mentioned plots for FC layer for both model rest plots could be found here [9].

Figure-1&2 correspond to the plots for ResNet(ImageNet) and Figure 3&4 correspond to ResNet(Places365).

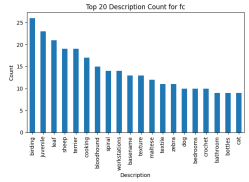


Fig. 1: Concept Learned Plot ResNet18(ImageNet)

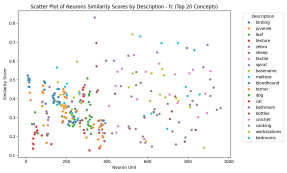


Fig. 2: Similarity Plot ResNet18(ImageNet)

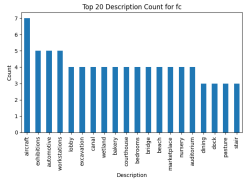


Fig. 3: Concept Learned Plot ResNet18(ImageNet)

### C. Findings and Conclusion

- *ImageNet Model:* The fc showed a high frequency of neurons associated with specific object categories. Lower layers(layer3, layer4) contained neurons responding to more abstract features. From the similarity plot we can see that at lower layer3 and layer4 the similarity is not that much but as at fc layer the similarity is really high . This shows as that the model learns the specific object categories.
- *Places365 Model:* The fc layer learned scene categories. Lower layers similarly contained abstract and structural concepts.From the similarity plots we can see at the layer 3 and layer 4 i.e the lower layers show more similarity values for abstract and structural concepts and as we move towards fc we see that the similarity score increase for scene categories.

From the above observation and plots we can say that, the two models have different representations of concepts and these are a consequence of the characteristics in data. Unlike the ImageNet model, where its primary focus is on object identification and differentiation in an image or vision

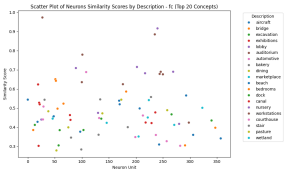


Fig. 4: Similarity Plot ResNet18(Places365)

(as opposed to sound recognition), Places365 classifies more with scene classification. In this we can see the similarity and difference in internal feature representations of neural networks due to dataset characteristic.

## II. TASK2: LIME

LIME(Local Interpretable Model-agnostic Explanations), It generate interpretations of any type for a wide range of models by explaining the model in subgroups around nearest neighbor points. Heavily on local interpretation: It acts as a simple model (like linear) locally approximating to the black box, showing which features of input were important for generating the prediction . In this task we have used LIME on 10 ImageNet images to see an example of how we can look into the regions responsible visual predictions.

### A. Methodology

We used the LIME code from the following references and made some changes as per requirement [3], [4], we applied the technique to each of the ten images. For each image, we generated explanations by using a simple model like linear locally approximating to the model, this shows which features are are important in the prediction.

### B. Results [10]

Here are few of the images generated by LIME. These result are generated with top 10 positive features from obtained LIME methods. Yellow boundary defines the features that were significant in making the prediction. You cab find the rest of the resultant image on this link.Results

Fig. 5: Gold Fish

Fig. 6: Kite

Fig. 7: Tiger Shark

Fig. 8: Vulture

### C. Observations

Here is the observation for various image results.

- **Gold Fish**: The code highlighted the body and fins of the fish, showing that these features were key for the prediction. Fins are most important feature for the model to predict due to bright color and distinctive shape. This could be seen in figure 5 too.
- **Tiger Shark**: As we can see in figure 5 that the some part of whales body and its fins are most important features for model to make the prediction. Fins and head are prime regions that play prime role for the model to make the prediction.
- **Kite**:For the kite image as seen figure 6, LIME highlighted the bird's wings and tail, essential features for identifying this bird species. The long, narrow wings were important region used by model to make predictions.
- **Vulture**: In figure 8 we can see that vulture's head and neck area are few important features that help model to make prediction.
- **Iguana**: In the iguana image, the head, scales, rough texture and unique coloration patterns were highlighted areas which means these features are important and crucial for the model's prediction.
- **Flamingo**:In this image, LIME emphasized the color, shape of its legs and neck, which are unique features for this image by the model.
- **American Coot**: The features in the image of coot are head and body, especially the plumage and frontal shield, were also highlighted. The above features are important for model's prediction.
- **West Highland White Terrier and Human**: As in the result image the faces of both the human and the dog are highlighted, also focusing on the eyes and their interaction. These features are important for the model's prediction. The model also note the presence of human as highlighted in results.
- **Racing Car**: The analysis of the racing car image highlighted the front and sides of the car. The car's logo and aerodynamic features were key feature that help the model to predict it as a racing car.
- **Orange**: The LIME output for the orange image focused on the texture and color of the fruit's. This is one of the simplest image and most clearly recognising the image most of the important features.

### D. Conclusion

The LIME most important features of each image that helps a model to learn and predict.LIME helps us understand and trust the model's predictions better. This explainability provided by LIME helps us to learn how model learns to predict and the resoning behind it.

## III. TASK3: GRAD-CAM

In this task, our aim to visualize the regions of input images that contribute most to the predictions of a neural

network model using different Gradient-weighted Class Activation Mapping (Grad-CAM) techniques. Grad-CAM helps interpret the model's decision by highlighting important regions in the input image. We will also compare Grad-CAM with other methods like LIME and extend the analysis using AblationCAM and ScoreCAM.

### A. Methodology

- **Grad-CAM:** Grad-CAM computes the gradient of the target class score with respect to the feature maps of the last convolutional layer. These gradients are then globally averaged to obtain the importance weights. The weighted combination of these feature maps is passed through a ReLU to obtain the heatmap.
- **AblationCAM:** This method systematically ablates (removes) parts of the network (e.g., neurons, layers) and observes the change in the output to determine the importance of the ablated part.
- **ScoreCAM:** ScoreCAM generates a heatmap without using gradients. Instead, it perturbs the input image and measures the change in the target score, attributing higher importance to parts of the image that significantly impact the score.

### B. GradCam v/s LIME

As we compare the results of GradCam to LIME as we can see in Figure 5 that LIME predicts are larger area and also focuses on more features in the image than just the object where as GradCam from figure 9 we can say that it generated a more compact region using the gradients and activation map. Most of the significant regions in model prediction are inside the object and not its surrounding thus providing us a better understanding of the part of the features that help the model to predict. In task 4 below we will be comparing in the above 2 methods in details.

We used a already implemented code for obtaining the GradCAM, AblationCAM and ScoreCam results. [5], [6]

### C. Results

Here are the image heatmap generated by GradCam, AblationCam and ScoreCam. This is a simple image of a gold fish showing how activation heatmaps by highlighted are.

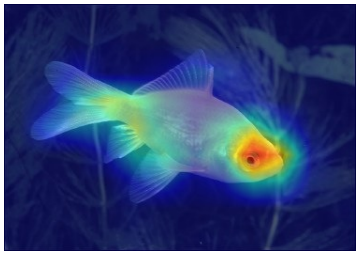

Fig. 9: GradCam

From the above images we can say that:

- **Grad-CAM**: As seen in figure 9 head and fins of the fish as the main contributing regions. This approach emphasise on the emphasizes on these features are essential for the model's decision, as the fins are distinct in shape and its texture
- **AblationCAM**: From figure 10 we can see that the regions near the head and upper body of the fish are highlighted which shows that these are most influential features used by model to predict.
- **ScoreCAM**: Figure 11 presents a smoother heatmap, focusing on the head and fin of the fish we can see like the other two the heat map is not that significant on the body but other two had activation on the body also .In this method we can say that it highlights the same regions as other two but smoother transition and more integrated view of important feature as the head and fins are highlighted more than the body.

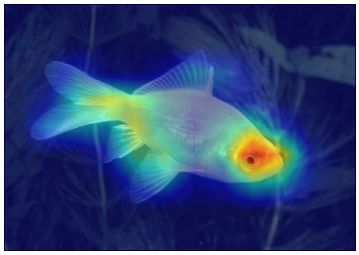

Fig. 10: AblationCam

Fig. 11: ScoreCam

We generated results for 9 more images which could be found on this link. [11]**Results**.

The above observation stay same over all the images we analysed that Scorecam provides a smoother heatmap and highlight lesser features. AblationCam provide a how different ablated parts of image helps the model to learn and make prediction and GradCam uses gradient from the output and highlight region that was more significant in the model predictions.

### D. Conclusion

The CAM methods (Grad-CAM, AblationCAM, and ScoreCAM) effectively highlight the critical regions of the input image influencing the model's predictions. This analysis shows the utility of CAM methods in understanding how models learn and make prediction. We also saw that how GradCam provide better insights than LIME by marking a more compact and useful area which is used by model to make prediction, hence providing us a better understanding.

## IV. Task4: Compare results from LIME and Grad-CAM

In this report we compare results gotten from Local Interpretable Model-agnostic Explanations (LIME) and Gradient-weighted Class Activation Mapping with the same set of images. LIME 1 will just use masks of top 10 features with only positive imapct, the LIME2 includes both posiitve and negative impacts. This means these methods are evaluated for IoU of the highlighted regions. This will tell us in which shape these two methods are agreed, measuring an IoU as the overlap between 2 regions.

### A. Data Preparation

- Processed the Images and Created Explanatory Masks
- The LIME 1 and LIME 2 masks are different criteria:
  - **LIME 1**: Top 10 features with positive impact only.
  - **LIME 2**: Top 10 features with both positive and negative impact.

### B. IoU Calculation [7]

- Masks from both were resized to a same shape.
- The IoU was computed as the ratio of the intersection to the union of the two masks.

Here is the snippet from implementation:

```
def calculate_iou(mask1, mask2):
    target_shape = max(mask1.shape, mask2.shape, key
    =lambda x: x[0] * x[1])

    resized_mask1 = resize_mask(mask1, target_shape)
    resized_mask2 = resize_mask(mask2, target_shape)

    intersection = np.logical_and(resized_mask1,
    resized_mask2).sum()
    union = np.logical_or(resized_mask1,
    resized_mask2).sum()
    iou = intersection / union if union != 0 else 0

    return iou
```

Here you can find the code implimentation: Code

### C. Results

The IoU values for the comparisons between IOU1(LIME 1 and GradCam), and IOU2(LIME 2 and Grad-CAM) for various images given to us are summarized in Table below I.

| Image | IOU1 | IoU2 |
|---|---|---|
| Kite | 0.3177 | 0.3177 |
| Orange | 0.3878 | 0.3457 |
| Vulture | 0.3900 | 0.3570 |
| West_Highland_white_terrier | 0.4457 | 0.3845 |
| Tiger_shark | 0.3664 | 0.3664 |
| American_coot | 0.4087 | 0.3853 |
| Flamingo.JPEG | 0.2799 | 0.5343 |
| Common_Iguana | 0.2894 | 0.2894 |
| Goldfish | 0.2828 | 0.2828 |
| Racer | 0.2306 | 0.2282 |

TABLE I: IoU1 and IoU2 across various images

### D. Analysis

- **Simpler Images**:It is probably apparent that both for simpler images like the goldfish IoU are small, which indicates LIME and Grad-CAM place importance on different areas even in simple scenarios.
- **Complex Images**: IoU is low for the kite and other complex images, showing that key region identification methods greatly disagree with each on this category of important target regions.identifying important regions.
- **LIME 1 vs LIME 2**: For some images, LIME 1 has a slightly better intersection over union with Grad-CAM (better agreement when only considering the positive effects). For some images, LIME2 provides a higher IoU by including both positive and negative features, which might occasionally be more similar to Grad-CAM positions while some shows equal scores for both meaning that the LIME1 and LIME2 are highlighting few images i.e.: For less complex image they learn same top 10 features.

### E. Insights

- **Method Agreement**: The agreement between LIME and Grad-CAM is generally low, reflecting different underlying mechanisms of how each method identifies important regions. LIME, being model-agnostic and focusing on local perturbations, highlights features differently compared to Grad-CAM, which leverages gradient information specific to convolutional neural networks.
- **Effect of Feature Selection**:Including both positive and negative features in LIME (LIME 2) does not consistently lead to better or worse alignment with Grad-CAM. The specific impact seems to vary with the complexity and nature of the image.
- **Image Complexity**: Both simpler and more complex images show variability in IoU, indicating that the complexity of the image alone does not determine the level of agreement between the methods.

### F. Conclusion

It is demonstrated that there are considerable differences in the contrast exhibition provided by Grad-CAM and LIME on crucial locations in photos. This is a little less clear-cut, but still: the characteristics that LIME employs (positive only vs. both pos and neg) also affect the agreement with Grad-CAM, albeit this isn't the case for all photos. These findings emphasize the necessity of using a wide range of explanation techniques to get a comprehensive knowledge of model behavior.

## References


[1] Clip Dissect https://github.com/Trustworthy-ML-Lab/CLIP-dissect
[2] Task-1 (Implementation) https://github.com/balwanishivam/tml-assignment/blob/Assignment-4/Assignment-4/task-1/interpretation.ipynb
[3] LIME https://github.com/marcotcr/lime/blob/master/doc/notebooks/Tutorial%20-%20images%20-%20Pytorch.ipynb
[4] Task-2 (Implementation) https://github.com/balwanishivam/tml-assignment/tree/Assignment-4/Assignment-4/task-2

[5] Grad Cam https://github.com/jacobgil/pytorch-grad-cam?tab=readme-ov-file#using-from-code-as-a-library
[6] Task-3 (Implementation) https://github.com/balwanishivam/tml-assignment/tree/Assignment-4/Assignment-4/task-3
[7] Task-4 (Implementation) https://github.com/balwanishivam/tml-assignment/blob/Assignment-4/Assignment-4/task-4/iou.ipynb
[8] Clip Dissect Library https://github.com/Trustworthy-ML-Lab/CLIP-dissect/blob/main/describe_neurons.py
[9] Plots for Task-1 https://github.com/balwanishivam/tml-assignment/tree/Assignment-4/Assignment-4/task-1/results
[10] Result Images for Task-2(LIME https://github.com/balwanishivam/tml-assignment/tree/Assignment-4/Assignment-4/task-2/results/boundary_1/images
[11] Result Images for Task-3(GradCam https://github.com/balwanishivam/tml-assignment/tree/Assignment-4/Assignment-4/task-3/results