Report Assignment Sheet #4: Explainability

Nima DindarSafa – 7072844
Samira Abedini – 7072848

Task-1: Network Dissection

- Summary of the method

Network dissection method has been proposed to evaluate every individual convolutional unit in a CNN. For every input x, the activation map $A_K(x)$ of every internal convolutional unit k is collected. Then, for each unit k, the top quantile level T_k is determined. Finally, the activation map is scaled up to the mask resolution using bilinear interpolation. [1]

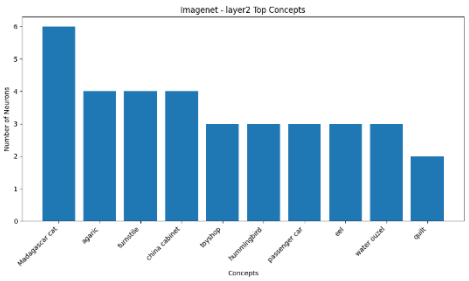
- Task

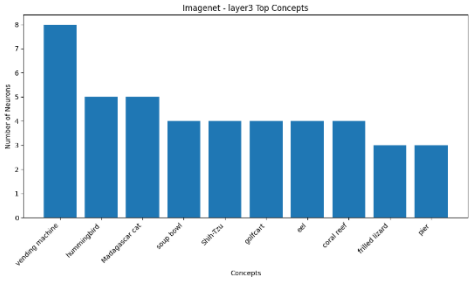
In this task we analyze which neurons are responsible for learning a specific class in a neural network. To do so, we label all the neurons from the last 3 layers of ResNet18 trained on ImageNet and Places365. Finally, we will analyze the labeled neurons.

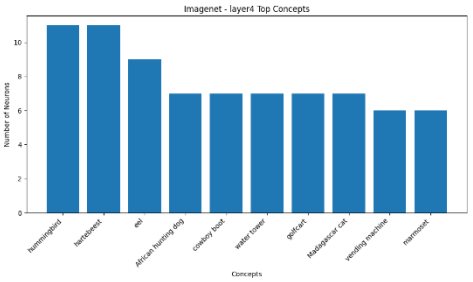
- Analysis

By analyzing the last 3 convolution layers of ResNet18 trained on ImageNet and Places365 datasets we have the following histograms of learned concepts by each neuron in each of mentioned layers:

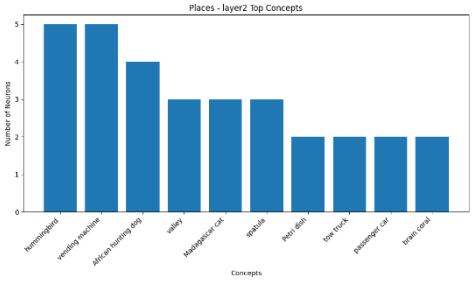
Concepts Learned by ResNet18 (ImageNet)

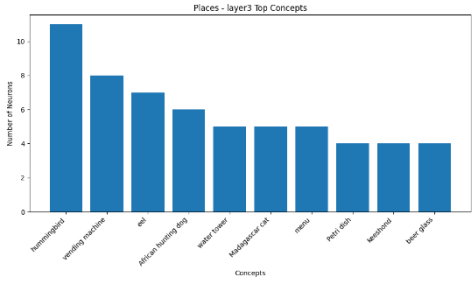


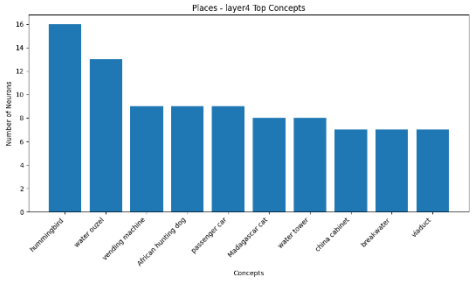




Concepts Learned by ResNet18 (Places365)







According to the histograms for model trained on ImageNet dataset, the neurons of layer 3 have learned the following top3 concepts: “Madagascar cat” (6 neurons), “agaric” (4 neurons), “turnstile” (4 neurons). These concepts are “vending machine” (8 neurons), “hummingbird” (5 neurons) and “Madagascar cat” (5 neurons) for layer 3 and “hummingbird” (11 neurons), “hartebeest” (11 neurons) and “eel” (9 neurons) in layer 4.

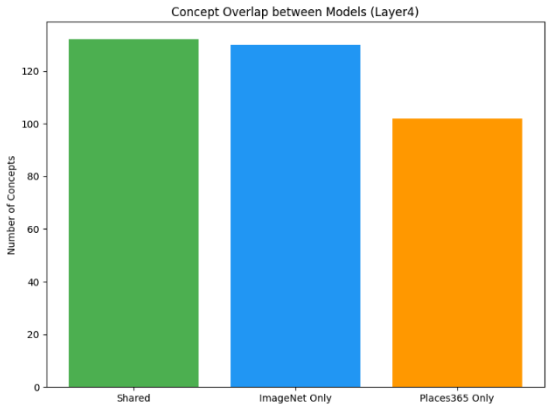
The same results for the model trained on Places365 dataset are ordered as follows:

- Layer 2: “hummingbird” (5 neurons), “vending machine” (5 neurons), “African hunting dog” (4 neurons)
- Layer 3: “hummingbird” (11 neurons), “vending machine” (8 neurons), “eel“ (7 neurons)
- Layer 4: “hummingbird” (16 neurons), “water ouzel” (13 neurons), “vending machine” (8 neurons)

According to these results “hummingbird” is the most learned concept by the neurons, and it has been learned by 19 neurons in the ResNet18 model trained on ImageNet dataset and by 32 neurons of the model trained on Places365. There are also concepts like vending machine and eel learned by many neurons.

The most obvious comparison between these two networks is that they are trained on datasets that include semantically different concepts. In other words, Places365 includes data point for scene classification and ImageNet has 1000 classes that include both scene and non-scene classes. So, we expect some shared concepts that are related to scene concepts in both of these models and there should be some exclusive concepts for each of these models.

The following histogram confirms our hypothesis:



According to this histogram, there are over 120 shared concepts among these 2 networks. However, over 120 concepts are only for ImageNet and around 100 concepts only for Places365. Considering that ImageNet is a more extensive dataset than Places365, it is reasonable to have more exclusive concepts in the model trained on ImageNet dataset. The model trained on ImageNet dataset has learned around 255 concepts in the last 3 layers and the model trained on Places365 has learned around 225 concepts in the mentioned layers.

As additional findings according to these results, we can infer that not all neurons are unique according to the concept they have learned. There are many duplicates among neurons and many of them share the concepts they have learned. The overall observation is that if we extend the training time, then we will have fewer duplicate neurons.

In the following table you can find the concept learned by each neuron (unit 1-3) for each layer (layers 2, 3, 4) for models trained on ImageNet and Places365:

	ImageNet			**Places365**		
	Layer 2	**Layer 3**	**Layer 4**	**Layer2**	**Layer 3**	**Layer 4**
Unit 0	agaric	menu	Shih-Tzu	African hunting dog	obelisk	China cabinet
Unit 1	quilt	water ouzel	grocery store	Petri dish	lens cap	Toy shop
Unit 2	turnstile	hummingbird	bookshop	altar	worm fence	rock crab

The RestNet18 model has 128 units in layer 2, 256 units in layer 3 and 512 units in layer 4 of convolution part. Creating a table containing the concepts of all neurons is infeasible. However, you can find the full description of each neuron in the following path:

“./results/resnet18_imagenet_25_07_17_15_28/descriptions.csv” (ImageNet)
“./results/resnet18_places_25_07_17_15_37/descriptions.csv” (Places365)

Task-2: Grad-CAM

- Summary of the method

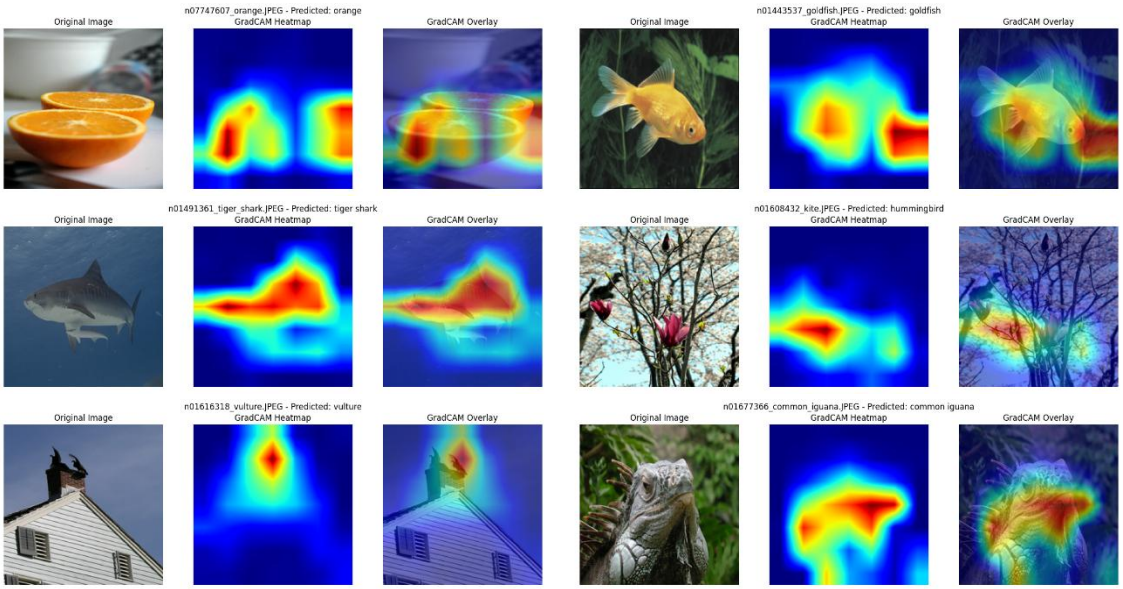
Grad-CAM method utilizes the gradient information flowing into the last convolutional layer of the CNN to assign importance values to each neuron for a particular decision of interest. In order to obtain localization map of width u and height v for any class c, we first compute the gradient of the score for class c with respect to feature map activations of convolutional layer. These gradients flowing back are global average pooled over the width and height dimensions to obtain the neuron importance weights. In this method, ReLU is applied to the linear combination of maps because we are only interested in the features that have a positive influence on the class of interest. [2]

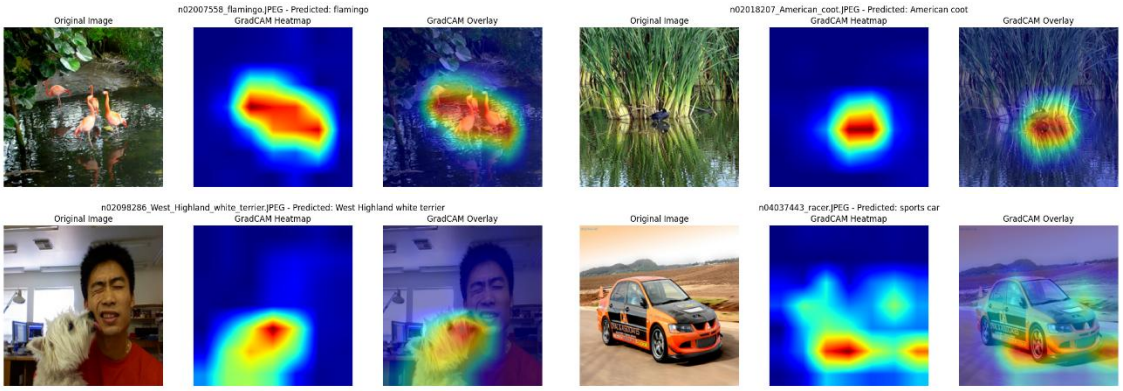
- Task

In this task we utilize Grad-CAM on each of 10 provided ImageNet pictures. To do so, we compute the gradient of the output with respect to the last convolutional layer and visualize the parts of the input image that are responsible for the main prediction by the model.

- Analysis

The Grad-CAM heatmaps achieved using 10 specified images are listed below:

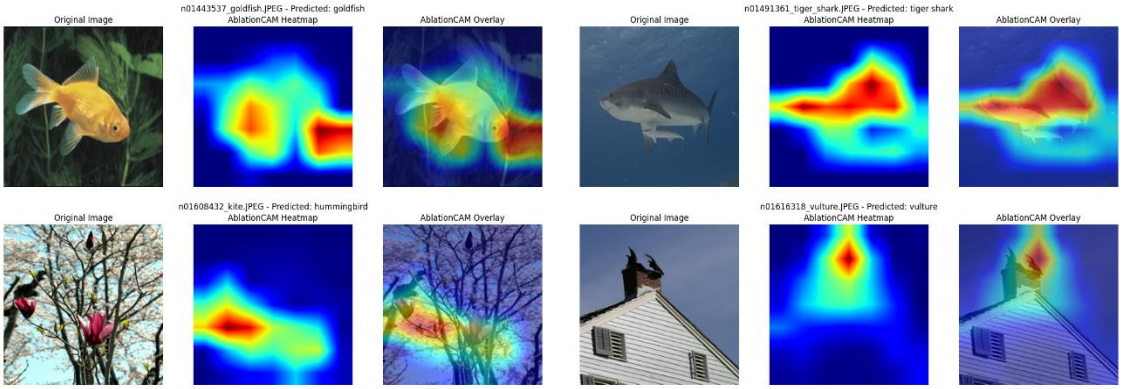


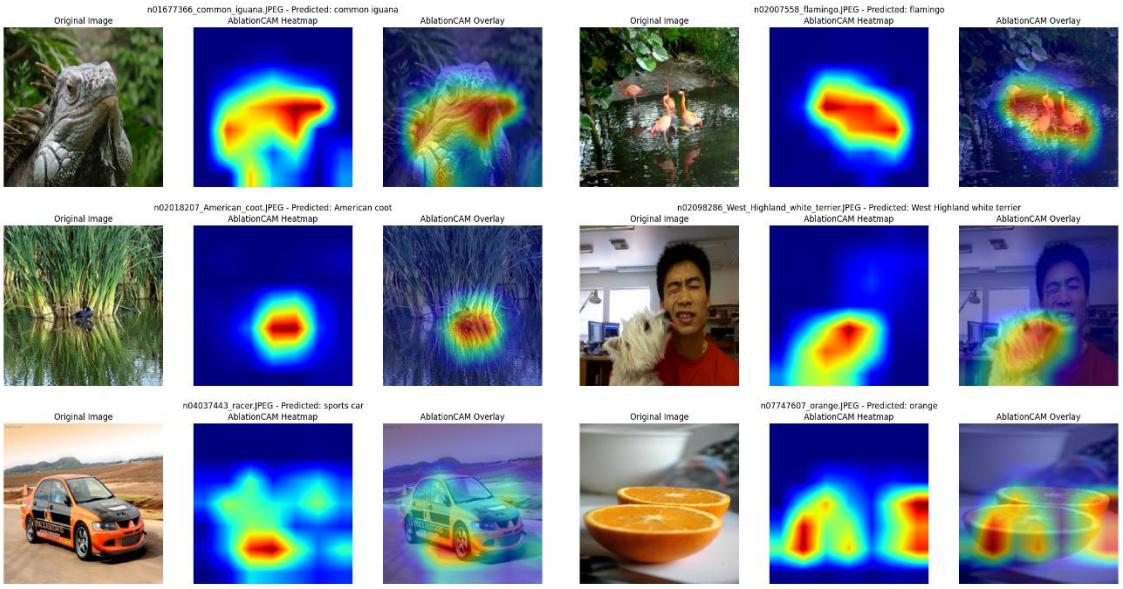


The results indicate that the overall performance of Grad-CAM method for obtaining the most important parts of the input image is acceptable. For instance, in the image predicted as “West Highland white terrier” the model has successfully utilized the related concepts to the given class to predict. The heatmap fully covers the related semantic information. However, in images like “sports car” and “goldfish” the model is not primarily focused on the object of the class, and it uses the information of its surrounding (e.g. road for sports car and sea vegetables for goldfish). This might eventually harm the model’s predictions if these objects are presented in another surrounding. For instance, if goldfish are presented in a bowl or a car is parked in a parking lot the model’s performance for these instances might drop.

Finally, the heatmap of the “orange” class does not fully cover the given semantics in the image. The heatmap includes parts of orange and parts of background that are not related to the concept of orange. This might have a reason like the one we explained above. In many instances of orange there might have been many images that had related background that model utilized to learn that class and in this specific instance this is not the case and model utilizes information that is not related to the concept of orange.

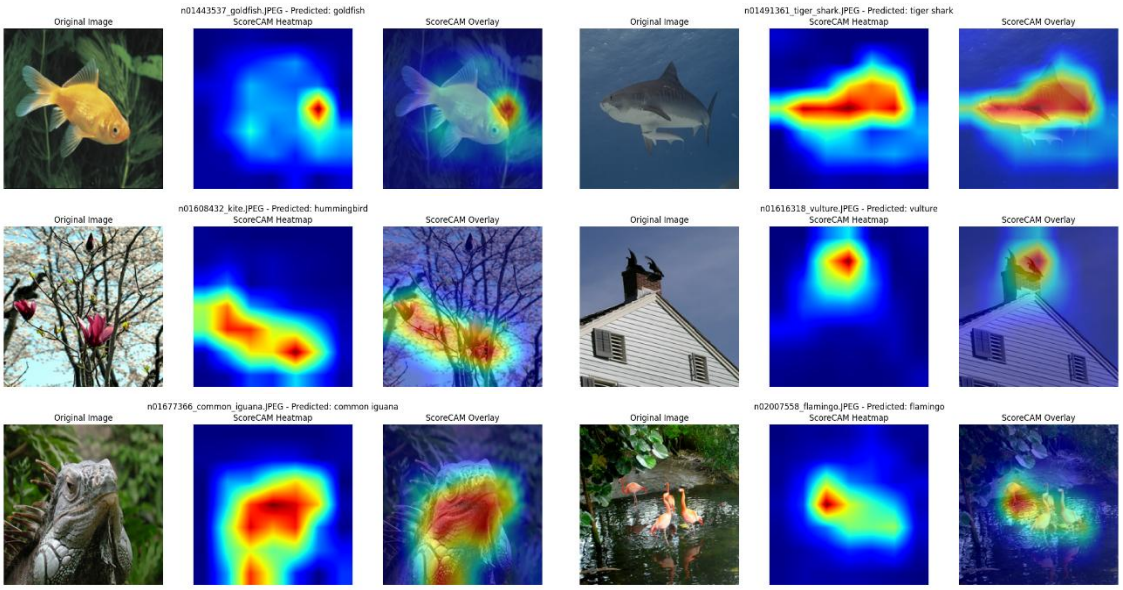
The Ablation-CAM heatmaps achieved using 10 specified images are listed below:

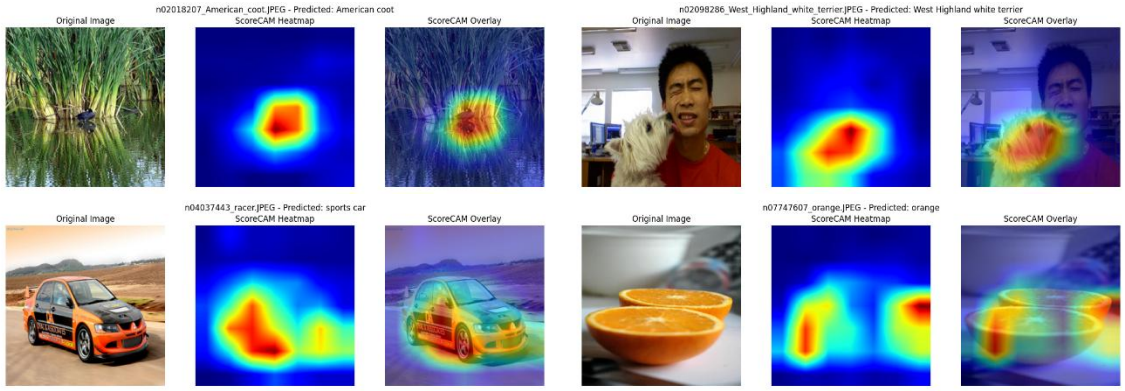




The results achieved using Ablation-CAM are almost identical to results of Grad-CAM. However, there are differences. For instance, in the explanation created by Ablation-CAM for “sports car” the identified important part of image is the car’s tire which is different from the one for Grad-CAM and there is a lesser focus on the road.

The Score-CAM heatmaps achieved using 10 specified images are listed below:





There are some differences between Score-CAM and the other 2 methods. For instance, in the “sports car” this method’s focus is more on the body of the car instead of the road. This method has successfully detected the important semantics for classification of “sports car”.

In another instance, the class predicted as “tiger shark”, unlike the other 2 methods, Score-CAM uses the whole body of shark to create its explanation. Also, for predicting “flamingo”, this method’s focus is on the one flamingo behind. Score-CAM has a less focus on the whole body of “goldfish” and uses semantically non-relevant information. For “common iguana”, unlike the other 2 methods, there are a less focus on the back of the iguana.

Task-3: LIME

- Summary of the method

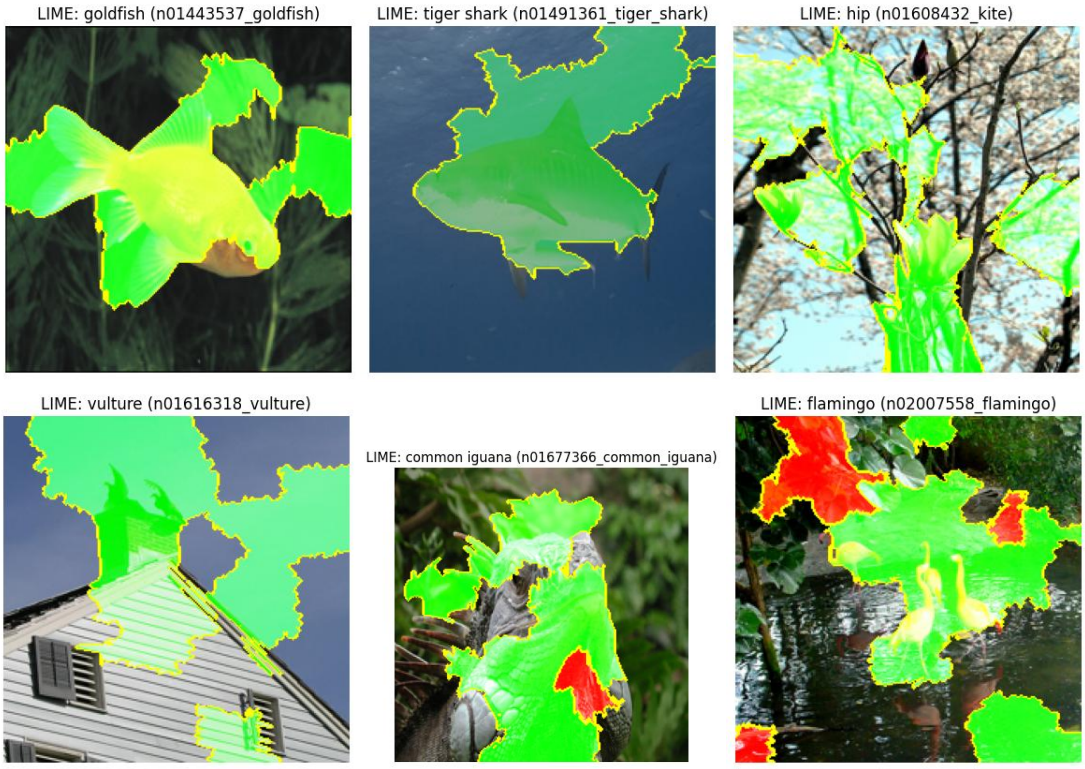
The overall goal of LIME (Local Interpretable Model-agnostic Explanations) is to identify an interpretable model over the interpretable representation that is locally faithful to the classifier. LIME method assumes that the given neural network model is black box. Then, this approach perturbs each original data point and passes the perturbed data points through the black box network to achieve their corresponding output. LIME also weighs the data points as a function of proximity to the original data points. Finally, this method fits a surrogate model g to the newly achieved data points. Each original data point can be explained by the newly trained explanation model. [3]

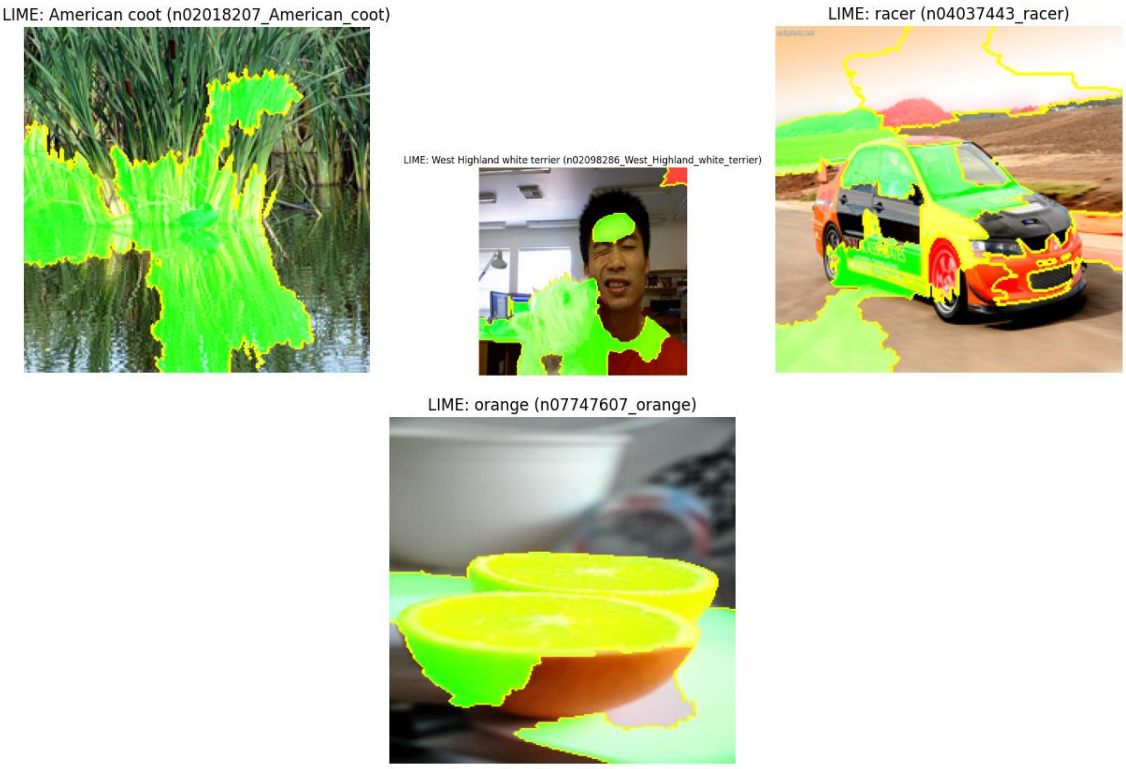
- Task

In this task we will visualize the explanation created by the LIME method for the same 10 images of ImageNet dataset from the previous task.

- Analysis

The LIME explanations achieved using 10 specified images are listed below:





As discussed earlier, LIME method creates an interpretable dataset that is visually understandable for humans. To do this, LIME utilizes an image segmentation algorithm to segment input image and the perturbed instances are the samples of the given image where the other segments are masked out. This segmentation might result in including the parts of the image in the explanation that are not semantically relevant.

For instance, in explanation of “vulture” is considered positive contribution to the final prediction of this class. However, in methods like Grad-CAM the explanation was specific to the regions of the image that “vulture” was present. In another example, a part of “common iguana” has a negative contribution to the prediction of this class, though it is semantically relevant to that concept. In the “sports car” example, the explanation has a good performance using parts of the car’s body in addition to its wind shield to infer “sports car”. However, parts of sky also contribute positively to the class prediction.

In use of explain_instance method of LimeImageExplainer we used the following arguments which are submitted to the score board:

```
params = {
    "labels": None,
    "top_labels": 1,
    "hide_color": 0,
    "num_features": 100000,
    "num_samples": 1000,
    "batch_size": 10,
    "segmentation_fn": None,
    "distance_metric": "cosine",
    "model_regressor": None,
    "random_seed": None,
}
```

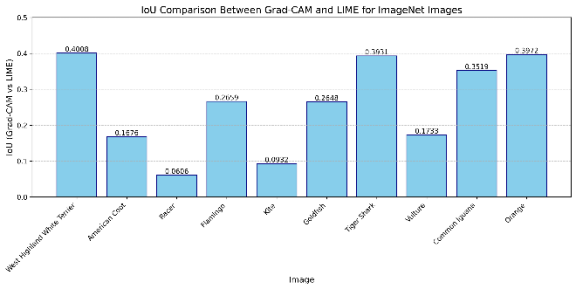
By submitting our arguments to the score board, we achieved the following results:

```
{'avg_iou': 0.3081, 'avg_time': 4.6519}
```

Task-4: Compare results of Grad-CAM and LIME

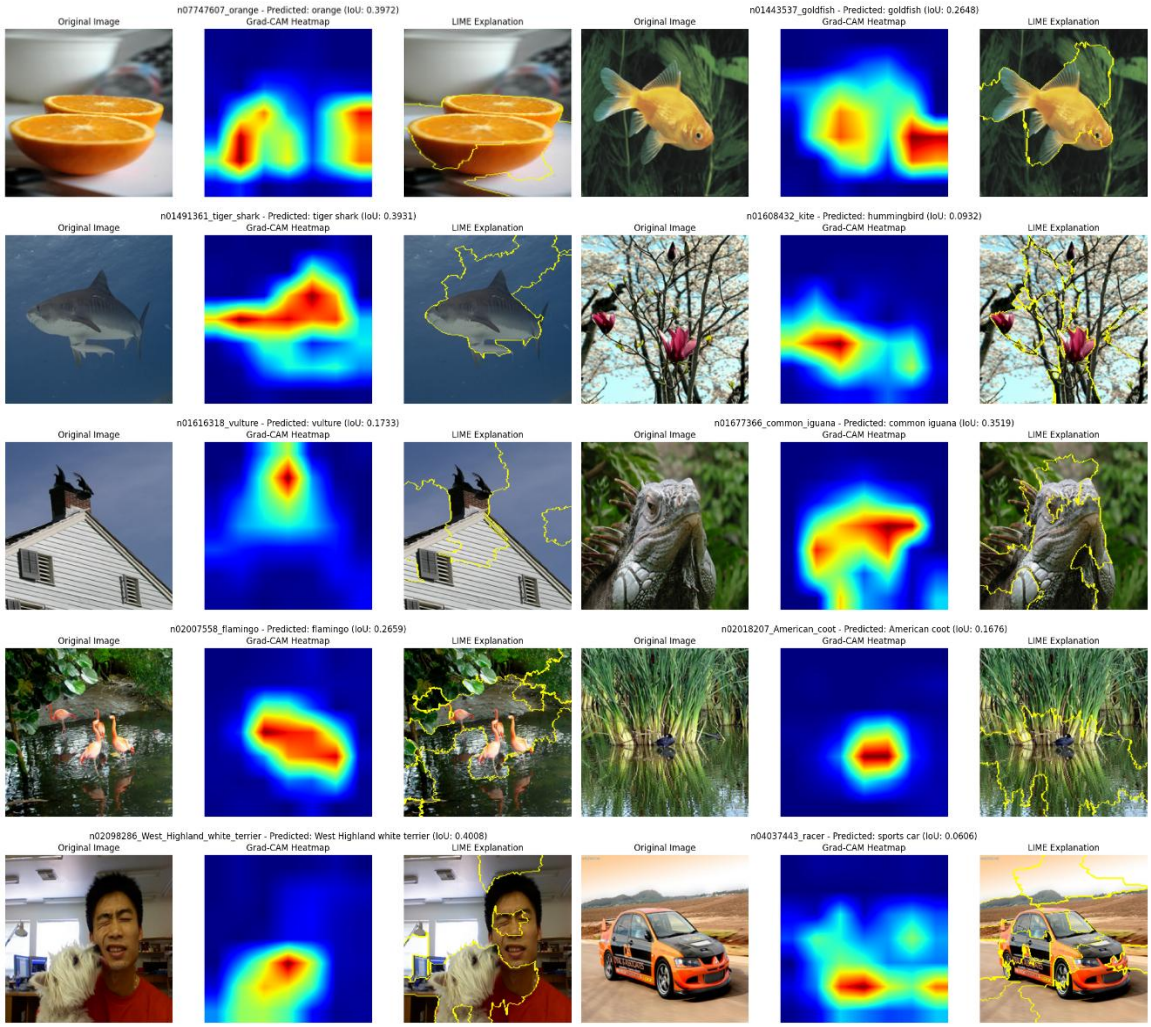
- Analysis

In the following you can view the comparison of IoU for Grad-CAM and Lime:



As we can see class “West Highland White Terrier” has the highest intersection over union between the 2 methods achieving the IoU of 0.4008. The classes “Orange” (0.3972) and “Tiger Shark” (0.3931) achieve the second and third highest IoU respectively. These three images are relatively simpler with respect to the semantic details in the image, leading to a higher IoU between the two methods.

On the other hand, “Racer” (0.0606), “Kite” (0.0932) and “American Coot” (0.1676) the least 3 IoU between Lime and Grad-CAM. As we can see in the images of these classes, they have relatively complex features contributing to a lower IoU. The overall trend is the higher the complexity of the image, the lower the agreement (i.e. IoU score) between the two methods.
Here we present a detailed comparison between Lime and Grad-CAM in the following figures.



As is evident from the figures above, most of the explanation generated by Grad-CAM lies within the segmentation generated by Lime leading to a high IoU. This is also the case for “Orange” and “Tiger shark”. For tiger shark for instance, the segmentation (explanation) created by Lime includes a part of the background (water) whereas the Grad-CAM is highly focused on the shark itself. Yet, there is a relatively high agreement between these two methods.
The lower IoU for complex instances like “Sports car” roots from this explanation that Grad-CAM is highly focused on a part of road and the car’s wheel, but Lime uses wind shield and car’s body as its explanation.

- References

[1] Bau, David, et al. *Network Dissection: Quantifying Interpretability of Deep Visual Representations*. arXiv:1704.05796, 2017. arXiv, https://arxiv.org/abs/1704.05796.

[2] Selvaraju, Ramprasaath R., et al. *Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*. arXiv:1610.02391, 2016. arXiv, https://arxiv.org/abs/1610.02391.

[3] Ribeiro, Marco Tulio, Sameer Singh, and Carlos Guestrin. *"Why Should I Trust You?": Explaining the Predictions of Any Classifier*. arXiv:1602.04938, 2016. arXiv, https://arxiv.org/abs/1602.04938.