Trustworthy Machine Learning, SS 2025
Prof. Dr. Adam Dziedzic and Prof. Dr. Franziska Boenisch
Assignment Sheet #4: Explainability

CISPA HELMHOLTZ CENTER FOR INFORMATION SECURITY
UNIVERSITÄT DES SAARLANDES

Teaching Assistant: Jyotika Mahapatra

Deadline: 23rd of July, 2025

Objective: The main aim of this assignment is to delve deeper into the topic of **explainability**. The assignment requires you to provide explanations to predictions provided by deep neural networks.

In this assignment you need to use the main three methods described in the lecture on explainability: **Network Dissection**, **LIME**, and **Grad-CAM.** Refer to the following **lecture** and **notes** for better understanding of the concepts. The assignment is divided into **3** main tasks and each task comes with its own deliverables.

Task Artifacts:
Datasets:

- For the **Network Dissection** task (Task-1), **ImageNet** is used.
  The dataset can be downloaded from various sources. We provide to you the following thread for help: Link
  You can use examples of images from ImageNet from the following repository too: GitHub Repo

- For the **LIME** and **Grad-CAM**, the following specific images from the ImageNet dataset need to be used:
  a. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n02098286_West_Highland_white_terrier.JPEG
  b. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n02018207_American_coot.JPEG
  c. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n04037443_racer.JPEG
  d. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n02007558_flamingo.JPEG
  e. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01608432_kite.JPEG
  f. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01443537_goldfish.JPEG
  g. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01491361_tiger_shark.JPEG
  h. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01616318_vulture.JPEG

i. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n01677366_common_iguana.JPEG
j. https://github.com/EliSchwartz/imagenet-sample-images/blob/master/n07747607_orange.JPEG

The images are provided at the end of the document as well for your reference.

Models:
The first two models mentioned below (ResNet 18 trained on places 365 and ResNet18 trained on ImageNet) are for the Network Dissection task and the third model is for the LIME and Grad-CAM task.

- **ResNet18 trained on places 365**
  To download the model, you can run the following command:
  wget --progress=bar
  http://places2.csail.mit.edu/models_places365/resnet18_places365.pth.tar

- **ResNet18 trained on ImagNet**
  To use this model, you can simply use the following:
  from torchvision.models import resnet18, ResNet18_Weights.IMAGENET1K_V1
  resnet18(weights=ResNet18_Weights.IMAGENET1K_V1)

- **ResNet50 trained on ImageNet**
  To use this model, you can simply use the following:
  from torchvision.models import resnet50, ResNet50_Weights
  resnet50(weights=ResNet50_Weights.IMAGENET1K_V2)

Task-1: Network Dissection

Network Dissection identifies which neurons are responsible to learn a specific class in a neural network. In this task you need to analyze the internals of models with Network Dissection.
To do so, you need to label all the neurons from the last 3 layers of ResNet18 trained on ImageNet and ResNet18 trained on places 365. You then need to analyze the labeled neurons.

For this task, use the latest library. We recommend leveraging the code from: **CLIP-dissect**
Specifically, you can run the following script: **code**

Your analysis should contain information such as:

- Which concepts are learned by most neurons?
- Comparison of the concepts learned by ResNet18 trained on ImageNet vs ResNet18 trained on places 365.
- How many different objects are learned by the two models?
- What additional analyses can you run and what additional findings can you make?

Deliverable 1: A short report regarding your analysis (recommended 1 page) with some visualizations of your findings, e.g., histograms of the counts of neurons that learned different concepts, images that correspond to the neurons, etc. Please describe your findings well.

Task-2: Grad-CAM

Grad-CAM helps understand how an input affects the prediction of a class. It highlights regions of an input image that are important for a model's decision about a particular class. In this task you need to use Grad-CAM on each of the 10 images of ImageNet provided to you, to visualize the parts of the input image that are responsible for the main prediction by the model. For this task, compute the gradient of the output with respect to the *last convolutional layer*.

Use the following library for Grad-CAM: **Grad-CAM library** to analyze the images.

Further, you need to extend the analysis using other types of Grad-CAM, namely: AblationCAM and ScoreCAM.

Deliverable 2: For the 10 given ImageNet images, obtain and plot the annotations by Grad-CAM and analyze your results in a short report.

Task-3: LIME

LIME provides Local Interpretable Model-agnostic Explanations which also helps understand how an input affects the prediction of a class. The main intuition behind LIME is that it learns an interpretable model locally around the prediction. In this task you need to visualize for each of the 10 given ImageNet images which parts of them are responsible for the main prediction using LIME. Compare the results with the ones obtained using Grad-CAM.

You can follow the tutorial on LIME from: **LIME Tutorial**

Deliverable 3: For the 10 given ImageNet images, obtain and plot the annotations by LIME and analyze your results in a short report.

Deliverable 4: For this deliverable you need to send us the parameters you use for the explain_instance for every image. Below is the detailed explanation:

- The explain_instance that needs to be used for the generation of the explanation is defined as follows:

```
def explain_instance(self, image, classifier_fn, labels=(1,),
                     hide_color=None,
                     top_labels=5, num_features=100000, num_samples=1000,
                     batch_size=10,
                     segmentation_fn=None,
                     distance_metric='cosine',
                     model_regressor=None,
                     random_seed=None,
                     progress_bar=True):
```

- We need the parameters that you pass into this function, **except** the image and the classifier_fn.
- **Format of sending the parameters**:
    a. You first need to create a dictionary whose keys are the elements of this list : **['West_Highland_white_terrier', 'American_coot', 'racer', 'flamingo', 'kite', 'goldfish', 'tiger_shark', 'vulture', 'common_iguana', 'orange']**
    Thus one for each image. Make sure the naming is correct!
    b. Now the value of each key(image) needs to be a dictionary of the necessary parameters as mentioned previously. Make sure to **not include** the image and classifier_fn parameters.
    c. This **example code** will show how you can create the dictionary and the format it is expected to be in.
    d. Once the dictionary is made, turn it into a pickle file and submit it to the server (**http://34.122.51.94:9091/lime**) along with your token number. Follow the example code for the same. Please note the change in the submission server!

Important Note: Make sure the format and the keys of the dictionary are exactly the same as mentioned above (refer **example code**). Otherwise it might hamper the evaluation. Do not include the image or the classifier_fn to the parameter to the submitted pickle file.

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

In this task, based on taks 2 and task 3, identify the differences in the results between Grad-CAM and LIME.
Your analysis should contain information about:

- Comparison between the highlighted regions, for example, using Intersection over Union (IoU) metric (Higher IoU indicates greater agreeability).
- Further questions such as: is it the case that for simpler images, like goldfish, the IoU is higher than for more complex images such as kite, which would indicate that both methods agree more.

Deliverable 5: A short report on the differences you observed and the insights you gained on the differences in the two methods (recommended length, including any visualizations, if applicable, 1 page).

Things that can go wrong on your side:

- Incorrect format/structure of the pickle file you submit. Remember to have the exact same keys as mentioned previously (case sensitive as well). Use the **example code** to understand what we expect from you.
- Sending the dictionary with a number of keys not equal to 10. There are 10 images, we expect you to send parameters for each and every image.

Evaluating your submission:

We evaluate your tasks using the deliverables provided to us by you.

1. We go through your reports presented in Deliverable 1, 2, 3 and 5 and provide you scores according to the depth of the content and explanations.
2. We use the Deliverable 4 for the scoreboard.
    a. We first generate the explanation for each image using the parameters provided by you.
    b. We then use the explanation to compare it with the ground truth explanation on our side, using the IoU metric.
    c. We also keep a track of the time of execution of your method, as a measure of complexity for themodel_regressor. This is because in essence, LIME gives a trade-off between fidelity and complexity.
    d. Therefore you receive points for two criterias:
        i. Higher IoU
        ii. Lower complexity/execution time

Note: We also have a cut-off time for the generator which must not be exceeded, else we do not continue the evaluation of this particular deliverable.

Scoreboard

- You can access the scoreboard for this task here http://34.122.51.94:9091/score_board in the LIME IoU, LIME Time. This will help you to compare your solutions with other teams and see where you stand.

Important Note: Your results will be overwritten with each submission. This is because there exists a trade-off between fidelity and complexity of the model_regressor, and that needs to be measured.

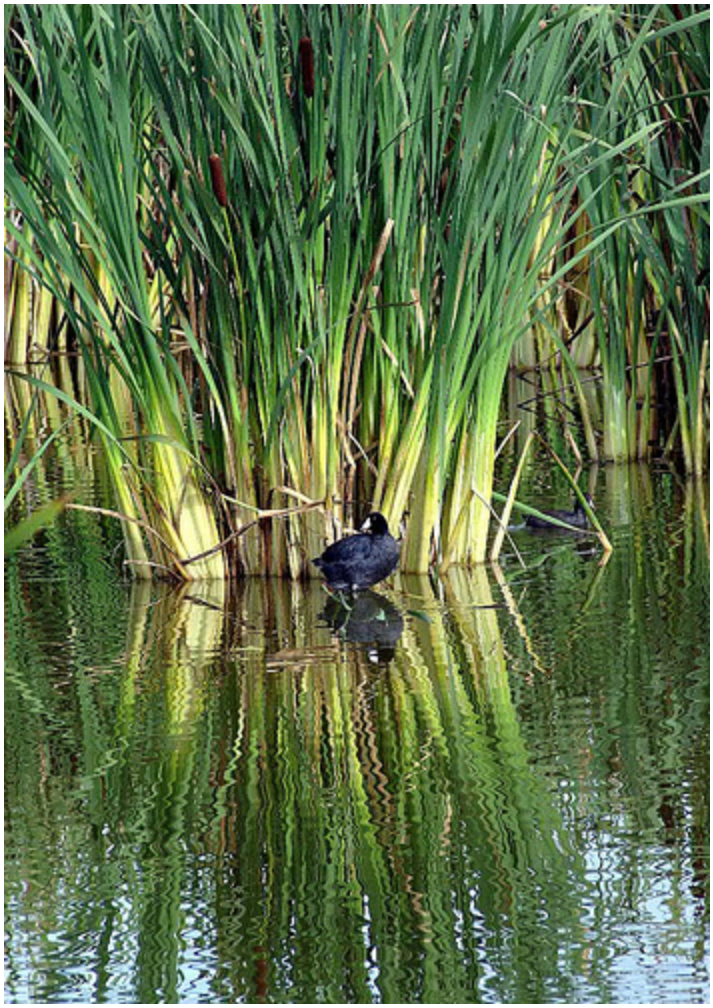
Remember : Both the submission and the scoreboard links have changed. Be careful while submitting.

Read the following instructions carefully

- Create a GitHub repository TML25_A4_YourTeamNumber. For example, if you are in Team 13, your repository should be named TML25_A4_13. ***The team number here should be the one assigned to you via email and not the one you see in CMS***.

- Upload your code files to the repository.
- The documentation for the assignment is split into a **README and a report, and both should be submitted**. The final grade will heavily depend on the documentation.
- Add a README.md to your GitHub repository that points us to the most important files and pieces of code.
- Add a report (PDF format) to your GitHub repository that describes your solution.
- **Submit a zip file (from your GitHub repository) with your readme, report, and the whole repository (only the important files, please avoid including the large models and artifacts in your repository).**
- You are only supposed to make *one submission* per group.

The images from the ImageNet dataset for your reference:




epicphoto.net
DYAL & ASSOCIATES
accountants + business advisors
MOTOR SPORT LOGIC