AI in Medicine I
WiSe 2022/23

Technical University Munich
Department of Informatics

I31 - AI in Medicine and Healthcare
I32 - Computational Imaging in AI and Medicine

Practical exercise 9 **21. Dec. 2023**

Trustworthy AI: Interpretability, Explainability and Uncertainty

Submission deadline: 12. Jan. 2024, 11.59 p.m.

Please submit your solutions (via Moodle).
The corresponding tutorial session is

21. Dec. 2023, 4-6 p.m. in lecture hall 5901.EG.051

For questions regarding this exercise sheet, please contact: `du216@ic.ac.uk`, `ivan.ezhov@tum.de`
For general questions, please contact: `course.aim-lab@med.tum.de`

1. (70% points) Interpretability and Explainability
   (a) (70% points) **Using CAM-based methods** In this task you are provided with a general template for the use of CAM-based methods for common imaging datasets. The task requires you to replicate these methods and analyse your results on the MedMNIST dataset (you are free to select any task from this collection of datasets, but we recommend you start with **classification**). You do not get evaluated on the completeness of your implementation (i.e. how efficient/pretty your code is), but rather on the quality of your analysis. Here is the to the tutorial: CAM-Explanations

   You can use the CAM implementation provided in PyTorch-CAM or use one of the alternatives such as Captum. Experiment with different CAM-based methods on the given datasets in order to identify the best one. Here the definition of the "best one" is subjective: You are expected to use between 3 and 5 methods, record the images produced along with the metrics associated with the model (e.g. class-specific accuracy). Your results should be recorded in a table which describes how you concluded that a specific method outperforms the rest (e.g. performance, similarity to how you would explain the image etc.) Be creative with your choice of metrics.

   The expected deliverables are: Short (under 1 page) report on which datasets and models you considered; a table with ranking of explanation methods; figures demonstrating ROIs of a few exemplary images.

   N.B. Addition of more datasets is highly encouraged and can contribute towards a more objective evaluation of the explanation method (and could earn extra marks).

   (b) (30% points) **Analysing the feature selection**

   As training progresses, your model is (hopefully) learning more representative features that can explain its predictions. Analyse the predictions made by your model at the start of training and compare to the ones made later in the training process. Do you notice a substantial difference between the features that are selected for explanations? At what point do you think you can stop the training because the features look "good enough"?

For this part we expect a short comment (1-2 paragraphs is enough) on the relative change in explanation quality associated with your selected learning task. You are allowed to re-use your code from part (a) to simplify the implementation.

2. (30% points) **Using MC Dropout to estimate uncertainty**

   This is a part of the practical exercise on uncertainty. Your task is to go through the tutorial and fill in the missing code and answer the questions.

   Here is the LINK to the tutorial.

3. (30% points) **(EXTRA) Applying explanations to multi-label medical tasks**

   The marks from this task get added to the overall score of this specific tutorial (e.g. you score 85% in Tasks 1 and 2: Task 3 can get you up to full marks should you attempt to do it). If you total would be more than 100%, then it is capped at 100%.

   One of the more challenging tasks is multi-label classification, where the same image contains features associated with many classes at once. Explanations for these tasks can result in much more fine-grained features, useful for the practitioners. The library that we recommend you use can be easily extended to support multi-label explanations.

   In this task you are required to download CheXpert dataset and perform explanations with respect to **each individual class**. Do you notice how the model changes which features it looks at during different stages of training? Report your results in a table.

   Note: we are not requiring you to train the model on the **whole** dataset, you are free to pick a specific subset of CheXpert and report your findings (explaining why you made that data split decision).

   This (whole) dataset is very challenging to train a typical model on. Using the knowledge you gained from previous AIMED lectures explain a) why might this be the case, b) how to address this and c) what implications might this have on your results.

   Hint: If you choose to work with the library we recommend, here is the sample code that allows you to use the library in a multi-label setting:

```
batch_cams_per_category = []
for i in range(num_classes):
    batch_targets=[ClassifierOutputTarget(i) for _ in range(len(input_tensor))]
    category_cams = cam(input_tensor=selected_images, targets=batch_targets)
    batch_cams_per_category.append(category_cams)
```