Explainable AI (INFOMXAI)
2024-2025
Project 1
Explainable AI in a Radiology Scenario

Contents

1 Project Description

Objective. The goal of this project is to explore various techniques of explainable AI and evaluate their suitability for generating explanations in the context of chest X-ray classification. Using the CheXpert dataset (`https://stanfordmlgroup.github.io/competitions/chexpert/`), students will train a classifier and implement explainability methods, analyzing their effectiveness for this specific medical imaging modality. For practical reasons, a smaller downsized version of the dataset, available on Kaggle, will be used (`https://www.kaggle.com/datasets/ashery/chexpert/data`). This project aims to develop the skills of selecting and adapting appropriate explanation methods for different data types and end-users of AI explanations. It directly supports the course objectives by allowing students to gain hands-on experience in implementing and critically evaluating different explanation methods. Students will develop the skills to select and adapt explainability techniques based on the type of data, the nature of the problem, and the needs of end-users.

In this project, you will:

- Select and implement a convolutional neural network (CNN) or transformer-based model of your choice for predicting pathologies (in this case, **pleural effusion**) from chest X-rays.
- Explore and implement multiple explainability techniques for the model's predictions:
  - **Saliency Maps:** Visual explanations that highlight key regions of the image are required. The group of students is responsible for selecting a saliency map method they believe will provide the most effective explanation for predicting pleural effusion.
  - **Example-Based Explanations:** A ranked list of similar cases selected from a predefined catalog of potential images.
- Evaluate the quality and relevance of explanations:
  - **Saliency Maps:** By using Quantus to measure the following explanation properties: Faithfulness, Localisation, Complexity, Randomisation, and Robustness.
  - **Example-Based Explanations:** Using normalized Discounted Cumulative Gain (nDCG) based on a radiologist's similarity rankings.
- Reflect on which explainability techniques are most effective for this application, considering the type of data and the needs of explanation consumers.
- Write a short paper summarizing your methods, findings, and reflections on the evaluation results.
- Present your work through a poster presentation and experience what it's like to be an Explainable AI researcher in practice.

1.1 Project Evaluation

The project will be evaluated based on four components:

- **Quality of the saliency map-based explanations** (15%): by using Quantus (`https://github.com/understandable-machine-intelligence-lab/Quantus`), explanation quality will be assessed based on five key metrics (Faithfulness, Robustness, Randomisation, Localisation, and Complexity).
- **Quality of the example-based explanations** (15%): evaluated using normalized Discounted Cumulative Gain (nDCG) to measure the relevance and ranking effectiveness of retrieved examples.
- **Quality of the short paper** (35%): assessed based on a grading rubric, considering clarity, introduction, methodology, and analysis of results.
- **Quality of the poster presentation** (35%): evaluated using a grading rubric, focusing on structure, communication, and visual clarity.

1.2 Required Software and Libraries

You will write your code in **Python** and use **Google Colab** as the primary environment. Ensure the following libraries are available:

- PyTorch (for implementing models).
- Quantus for assessing the quality of the saliency map-based explanations.

2 Project steps

2.1 Data preparation and model training (February 8 - February 17)

In this step, you will download and preprocess the dataset to prepare it for training your deep learning model. Follow these sub-steps:

1. Download the dataset (February 8 - February 12)
    - Obtain the downsized version of the CheXpert dataset from Kaggle.
    - Explore the dataset to understand its structure and labels.
2. Pre-process the data (February 8 - February 12)
    - Resize all images to 224x224 for compatibility with most pre-trained deep learning architectures.
    - Select only frontal X-ray images and normalize pixel values.
    - Split the training data into separate training and validation sets while preserving the original validation test set as the final test set.
3. Train the classifier (February 13 - February 16)
    - Choose a CNN-based model (e.g., ResNet, DenseNet) or a transformer-based model.
    - Train the model and monitor performance using appropriate metrics.
4. Test and save the model (February 17)
    - Evaluate performance on the test set (original validation set).
    - Save the trained model for later use in generating explanations.

2.2 Generating and Evaluating Explanations (February 17 - February 28)

In this step, you will generate explanations for your trained model using both saliency maps and example-based explanations and evaluate their quality. Follow these steps within the suggested timeline:

1. Receive ground-truth explanations (February 17)
    - A set of ground-truth explanations provided by a board-certified radiologist will be shared.
    - These will serve as a reference to assess the quality of the explanations generated by your model.
2. Generate saliency map-based explanations (February 18 - February 24)
    - Apply a saliency map method or use attention to highlight important regions in the image.
    - Visualize and analyze the saliency maps to ensure they provide meaningful insights.
    - Re-train your model if it focuses excessively on non-relevant regions.
3. Generate example-based explanations (February 18 - February 24)
    - Implement an example-based approach to retrieve and rank similar cases.
    - Choose an appropriate similarity metric.
    - Ensure that retrieved examples are relevant.

4. Evaluate explanations (February 25 - February 28)
    - Assess saliency map explanations using Quantus (Faithfulness, Robustness, Randomisation, Localisation, Complexity).
    - Evaluate example-based explanations using normalized Discounted Cumulative Gain (nDCG) to measure ranking quality.

2.3 Preparing the deliverables (March 1 - March 9)

In this final step, you will compile your results and prepare the required deliverables: the Jupyter Notebook, short paper, and poster presentation. Follow these steps within the suggested timeline:

1. Prepare and present the poster (March 1 to March 5)
2. Write the short paper (March 1 to March 9)
3. Ensure Jupyter Notebook is well-documented (March 1 - March 9)

3 Deliverables

3.1 Poster presentation (scheduled for March 5th, during class time)

Groups will present their work through a poster session, simulating the experience of an Explainable AI researcher. The poster should:

- be in landscape format for readability and will be displayed using the laptop of one of the group members.
- visually summarize the key aspects of the project.
- clearly communicate the methodology, findings, and conclusions.

3.2 Short paper (submission deadline is March 9th at midnight)

A concise research-style paper summarizing the project, structured as follows:

- Introduction: context of problem and motivation for using explainability in chest X-ray classification.
- Materials and methods: Describe the dataset and preprocessing steps, the classifier and training setup, the explanation methods (example-based and saliency maps), and the evaluation metrics.
- Results and Discussion: Findings from the explanation assessments, strengths and limitations of the methods, and critical reflections.
- Conclusion: Key takeaways and potential future directions.

The short paper should follow the MIDL 2025 short paper submission guidelines (`https://2025.midl.io/call-for-papers`), a leading conference in the medical imaging domain. The short paper is then limited to three pages, excluding references. The template to be used can be found here: `https://github.com/MIDL-Conference/MIDLLatexTemplate`.

3.3 Jupyter Notebook (submission deadline is March 9th at midnight)

Students must submit a fully executable Jupyter Notebook containing their code implementation of the project. The students should also submit the weights of their final model. The notebook should:

- Instantiate all package versions in the first cell to ensure reproducibility.
- Include all necessary code for training the classifier, generating the explanations, and evaluating results.
- Clearly document the steps taken, including preprocessing, model training, and explainability analysis.
- Contain inline explanations and comments to enhance clarity.