

Unified Explainable AI Interface

This project aims to **integrate two existing Explainable AI (XAI) systems** into a single interactive platform capable of processing both **audio and image data**.

- **Deepfake Audio Detection (Repo 1):** Detects real vs. fake audio using neural networks such as **VGG16, MobileNet, ResNet and custom CNNs** for audio synthesis on the [Fake-or-Real \(FoR\)](#) dataset. XAI techniques including **LIME, Grad-CAM, and SHAP** provide insights into the model's predictions.
 - source: [Deepfake Audio Detector with XAI](#)
- **Lung Cancer Detection (Repo 2):** Detects malignant tumors in chest X-rays using **AlexNet and DenseNet**, leveraging the [CheXpert](#) dataset. Grad-CAM visualizations explain model decisions to support radiologists in early and accurate diagnosis.
 - source: [Lung Cancer Detection](#)

The goal is to **refactor and combine these two systems** into a single interface where users can:

- Upload or drag-and-drop **audio or image**.
- Select a **compatible classification model**.
- Apply one or multiple **XAI techniques** to visualize model decisions.
- **Compare results** across different explainability methods, with incompatible techniques automatically filtered out (e.g., image-only methods when audio is selected).

This unified platform enables exploration of **multi-modal classification, interpretability and comparative analysis of XAI techniques**.

Project Objectives

Students are expected to:

1. **Refactor and integrate** the two existing repositories into a unified, coherent interface. You may use **any libraries or frameworks** you are comfortable with for the graphical user interface (GUI) such as **Flask, Node.js, Streamlit, Gradio, React** or others of your choice.
2. Support **audio and image classification** using multiple pretrained models. **Audio (.wav) and image inputs are mandatory**, while support for additional formats such as CSV or others is considered a bonus.
3. Implement **XAI** for all applicable methods from both repositories (**LIME, Grad-CAM and SHAP are required**). Adding support for additional explainability techniques is encouraged and will be considered a plus.
4. Create a **comparison tab** allowing side-by-side evaluation of different XAI techniques on the same dataset.

5. Automate **compatibility checks** to ensure that only relevant XAI methods can be applied to each input type. For example, if a technique is **image-specific** (like Grad-CAM for chest X-rays) it should be automatically **disabled or hidden** when the user uploads an audio file, preventing selection. Similarly, audio-only explainability methods should not be available for image inputs. This ensures that the interface only presents **applicable XAI options** based on the dataset type, avoiding errors and improving usability.
6. Build a **user-friendly interface** that clearly presents classification results and explanations. The **interface design is flexible and up to your creativity**. You are **not required to follow the layout or style** like shown in this example image.



You can choose any GUI framework or workflow that best fits your implementation.

Mandatory Requirements

- Unified interface supporting **multi-modal inputs** (audio .wav files and chest X-ray images).
- **Integration of all original XAI techniques** from both repositories. **LIME, Grad-CAM and SHAP are required.**
- **Automatic filtering** of XAI methods based on input type.
- Start GUI with the **basic functionality**: allow the user to **select a dataset, choose one classification model, and apply one XAI method**, then display the result with the corresponding explanation.
- Extend this with a **Visualization and comparison tab** for multiple explainability outputs (e.g. a separate page or tab). Allows users to view and analyze multiple explainability outputs side by side. The tab should:

- Display the **classification results** alongside the corresponding XAI visualizations for each model.
- Support **multiple XAI techniques** for the same input (e.g., LIME, SHAP, Grad-CAM) in a **side-by-side or stacked view**.
- Clearly indicate **which XAI method was used** for each visualization.
- Automatically **filter out incompatible methods** based on input type (e.g., image-only XAI not shown for audio inputs).
- Well-documented **code, interface, and setup instructions**.

Team Requirements

- Teams of **1–5 students**. All members must be from the same TD group e.g. All team members must belong to the same TD group (e.g., all from CDOF1). Students from different TD groups (e.g., CDOF1 and CDOF3) are not allowed to form a team.
- Collaboration must be **clearly evident**.
- Solo projects are allowed.

Deliverables

Each team must submit **one complete submission** on DVL by the **final TD session** including:

1. **Source Code:** Fully refactored, unified interface for audio and image classification with XAI.
2. **Documentation:**
 - README.md including:
 - Group member names and TD group number
 - Project overview
 - Setup and installation instructions
 - Instructions to run the interface and demo
 - Short technical report describing:
 - Design and integration decisions
 - Selected models and XAI methods
 - Improvements made over the original repositories
3. **Live Demo:** Conducted during the **final TD session**, showcasing:
 - Upload and classification of audio and image datasets
 - Application of selected XAI techniques
 - Comparison tab with multiple explainability outputs

Optional Enhancements (Bonus)

- Implement **additional XAI techniques**.
- Provide **interactive elements**, such as zooming, toggling layers or overlaying explanations on the input.
- Add **interactive visualizations** (e.g. [BertViz](#)).

Evaluation Criteria

- **Code Quality:** Structure, readability, and modularity.
- **Functionality:** Fully working multi-modal classification, XAI visualization and comparison workflow.
- **Improvements:** Enhancements beyond original repositories.
- **Documentation:** Clear setup, usage instructions and technical explanations.
- **Teamwork & Demo:** Smooth collaboration and presentation during the final TD session.

Learning Outcomes

This project will allow you to **start from existing codebases** and gain hands-on experience with **Explainable AI (XAI) for both audio and image data**. It introduces **multi-modal analysis** providing a foundation to later extend the system to **support additional input types** beyond audio and images.

Generative AI Usage & Penalties

- The use of **Generative AI tools (LLMs)** is **allowed** for this project.
- **Any use of Generative AI must be explicitly declared** in the project documentation.
- Teams must include a **“Generative AI Usage Statement”** specifying:
 - Which tools/models were used
 - For what purpose (e.g., code refactoring, documentation writing, debugging, design, etc.)
- This declaration must be included in the **project documentation** (e.g., in the README.md or technical report).

Penalty Policy

- **Failure to declare the use of Generative AI**, or attempting to hide such usage, will result in a **severe penalty**.
- If Generative AI usage is detected and **not disclosed**, the project may receive a **significantly reduced grade or be considered non-compliant**.

*Transparency is mandatory. Using Generative AI responsibly and declaring its use will **not** negatively impact your evaluation.*