Amini Cocoa Contamination Challenge Report

Team: Brain Blend

Challenge Overview

The goal of the Amini Cocoa Contamination Challenge is to develop machine learning models that can accurately identify **all diseases** present in images of cocoa plants. A key requirement is that these models must:

- **Generalize well** to unseen diseases not present in the training data.
- **Operate efficiently** on **low-resource smartphones**, such as those commonly used by subsistence farmers across Africa.

ETL Process

We used the provided training dataset and prepared the data using the following pipeline:

- Applied **10-fold StratifiedGroupKFold** cross-validation to ensure balanced and diverse splits.
- For each fold, the corresponding **YOLO-style annotations (label files)** were matched and copied into `images/train`, `images/val`, `labels/train`, and `labels/val` directories accordingly.

Modeling Approach

Model Choice:

We selected **YOLO11s** , due to its lightweight nature and excellent performance on edge devices.

Training Strategy:

- Trained on **folds 6, 7, and 8**.
- For each fold:
    - Validation set was **split in half**:
        - One half was merged with training data.
        - The other half was used for evaluation.
- Each fold was trained for **~2 hours 30 minutes**, totaling **8 hours 33 minutes** well within the **9-hour GPU training limit**.

Validation Performance (Local mAP):

Fold	mAP
6	0.760
7	0.789
8	0.806

Inference Strategy

Ensemble Method:

We used a **Weighted Box Fusion (WBF)** ensemble of the 3 trained models.

Multi-scale Inference:

Each model performed inference across the following image sizes:
`[640, 800, 960, 1120, 1280, 1440]`
This increased robustness and improved detection accuracy on varying disease patterns.

Runtime Summary:

Notebook	Duration
Training	8h 33min
Inference	40min

Total: 9h 13min (training + inference), adhering to the challenge's **3h inference** and **9h training** constraints.

Interpretability

CAM Visualizations:

We implemented **EigenCAM** to visualize model attention maps.

- This method extracts the **first principal component** of 2D activations.
- It helps highlight **regions in the image** most relevant to disease detection.

Notebook: `Team_Brain_Blend_Interpretability_Report.ipynb`

How to Run the Notebooks

We have included all relevant files (weights + notebooks) in the provided ZIP archive.

To Train:

- Open `Team_Brain_Blend_Training_Notebook.ipynb`
- Adjust the dataset path if using **Google Colab** but for reproducible results please use Kaggle and use the latest environment since that is what we used
- Run notebook end-to-end to retrain the models

To Run Inference:

- Use `Team_Brain_Blend_Inference_Notebook.ipynb`
- You can either:
  - Use the pre-trained weights provided in the ZIP
  - Or use weights from the training notebook

Compliance with Resource Restrictions

- The models used (**YOLO11s**) are designed to be **deployed on low-resource devices**.
- Inference and training times are **within the required limits**.
- Models can be exported to **ONNX or TensorFlow Lite**, ensuring compatibility with **entry-level smartphones**.

⚠️ Although an ensemble was used, no restriction was stated against it, and all ensemble inferences were performed well under the 3-hour limit and the training runtime did not surpass the 9hr limit.
Even individual models are performant and suitable for edge deployment.