Defect Detection in Photovoltaic (PV) Modules: Approach Document

1. Introduction

This document outlines the approach used for defect detection in photovoltaic (PV) modules using deep learning techniques. The problem is framed as a combination of image classification and object detection, where defects such as hotspots and diode failures are identified and localized within images. The methodology leverages the YOLO (You Only Look Once) object detection framework to achieve robust and efficient detection.

2. Data Understanding & Preprocessing

2.1 Dataset Overview

The dataset comprises images of PV modules along with corresponding annotations provided in a CSV file. The annotations contain bounding box coordinates for detected defects and their respective class labels.

2.2 Exploratory Data Analysis (EDA)

Several analyses were performed to understand the dataset:

- **Class Distribution Analysis**: The number of instances per defect type was analyzed to identify any class imbalance.
- **Image Size Distribution**: Histogram plots were generated to study the distribution of image dimensions.
- **Visual Inspection:** Sample images were plotted with bounding boxes to ensure annotation correctness.

3. Data Preprocessing

3.1 Conversion to YOLO Format

To train a YOLO-based model, annotations were converted from CSV format to YOLO-compatible text files. The following transformations were applied:

- Bounding box coordinates were normalized based on image dimensions.
- Class labels were mapped to numerical indices.
- Image files were copied to a structured directory format required by YOLO.

4. Model Training

4.1 YOLOv8 Selection

The YOLOv8 model was chosen due to its efficiency in object detection tasks. A pre-trained YOLOv8 model was fine-tuned on the prepared dataset.

4.2 Training Process

- The training pipeline was executed using the Ultralytics YOLOv8 framework.
- Training parameters:
  - Number of epochs: 50
  - Batch size: 8
  - Image size: 416x416
  - Loss function: Standard YOLO loss
- The trained model weights were saved for inference.

5. Model Inference & Evaluation

5.1 Object Detection

The trained YOLO model was used to predict bounding boxes on the test dataset. Predictions were extracted, and defect labels were assigned to each detected region.

5.2 Submission File Generation

The final predictions were compiled into a CSV file in the required format, including filename, defect class, and bounding box coordinates.

6. Conclusion

This approach effectively utilizes deep learning for defect detection in PV modules. The YOLO-based pipeline ensures high-speed inference while maintaining detection accuracy. Further improvements can be explored by experimenting with data augmentation, hyperparameter tuning, and ensembling techniques.