**Project Documentation**

**1. Overview**

This document provides an overview of the project, including dataset preparation, YOLOv8 model training, and backend API development using FastAPI.

**2. Dataset Preparation**

**2.1 Dataset Conversion**

**Script**: xml\_to\_txt.py

* **Description**: Converts XML annotations from the Kaggle PCB dataset to YOLO format (TXT files).
* **Source**: Kaggle PCB Defects Dataset [Link](https://www.kaggle.com/datasets/akhatova/pcb-defects/data)

**2.2 Dataset Splitting**

**Script**: data.py

* **Description**: Splits the dataset into training, validation, and test sets in YOLO format.

**2.3 Directory Structure**

**File**: dataset.yaml

* **Description**: Defines the directory structure for YOLO.

**3. YOLOv8 Model Training**

**3.1 Training Command**

* **Command**:

bash

Copy code

yolo train model=yolov8n.yaml data=dataset.yaml epochs=5 batch=32 imgsz=640

* **Description**: Trains the YOLOv8 model using the specified configuration file and dataset.

**4. FastAPI Backend Development**

**4.1 FastAPI Script**

**Script**: FastAPI.py

* **Description**: Implements the backend API with the following endpoints:
  + /predict (POST): Accepts image data and a confidence limit, and returns predictions (bounding boxes and labels) as structured data.
  + /visualize (POST): Accepts image data and a confidence limit, and returns the image with drawn bounding boxes and label annotations.

**4.2 Running FastAPI**

1. **Install Dependencies**:
   * **Create requirements.txt**:

plaintext

Copy code

fastapi

uvicorn

pillow

ultralytics

* + **Install**:

bash

Copy code

pip install -r requirements.txt

1. **Start the FastAPI Server**:

bash

Copy code

uvicorn FastAPI:app --reload

**4.3 Testing the Endpoints**

* **Predict Endpoint**:
  + **Request**:

bash

Copy code

curl -X POST "http://localhost:8000/predict" -F "file=@/path/to/image.jpg" -F "confidence=0.5"

* + **Response**: Returns predictions (bounding boxes and labels) as structured data.
* **Visualize Endpoint**:
  + **Request**:

bash

Copy code

curl -X POST "http://localhost:8000/visualize" -F "file=@/path/to/image.jpg" -F "confidence=0.5"

* + **Response**: Returns the image with drawn bounding boxes and label annotations.

**5. Additional Notes**

* Ensure that all required Python packages are installed in your environment.
* Update file paths and model paths in the scripts as needed.
* Test the endpoints using tools like curl or Postman to ensure they are functioning as expected.