**Step 1: Install VS Code and Essential Extensions**

1. **Download and install** [**VS Code**](https://code.visualstudio.com/Download)
2. Open VS Code and install the following extensions:
   * **Python** (Microsoft) - For running Python scripts and Jupyter notebooks.
   * **Jupyter** (Microsoft) - For running Jupyter Notebooks in VS Code.
   * **Pylance** (Microsoft) - For Python language support and code suggestions.

**Step 2: Set Up a Python Virtual Environment**

1. Open the command terminal (you can use VS Code's terminal).
2. Create a virtual environment:

python -m venv myenv

1. Activate the virtual environment:
   * **Windows**: myenv\Scripts\activate
   * **Mac/Linux**: source myenv/bin/activate
2. In VS Code, select the interpreter associated with your virtual environment:
   * Go to **View** > **Command Palette** > **Python: Select Interpreter**.
   * Choose the interpreter located in the myenv folder.

**Step 3: Install Required Packages**

1. With your virtual environment activated, install the necessary libraries by running:

pip install pandas numpy scikit-learn opencv-python tensorflow flask

1. Optionally, you can create a requirements.txt file to track dependencies. In your terminal:

pip freeze > requirements.txt

1. To install from requirements.txt in the future:

pip install -r requirements.txt

**Step 4: Organize the Project Folder Structure**

Create a folder structure for your project:

predictive\_maintenance\_quality\_control/

│

├── data/

│ ├── sensor\_data.csv

│ └── product\_images/

│ ├── defective/

│ └── non\_defective/

│

├── models/

│ ├── predictive\_model.pkl

│ └── cnn\_model.h5

│-\_\_ predictive\_maintenance.py

|-\_\_ train\_cnn.py

├── app.py

└── requirements.txt

**5. Final Steps: Running the Project**

1. **Train Models**: Run it

* python -m venv myenv
* myenv\Scripts\activate
* python predictive\_maintenance.py
* python train\_cnn.py
* python app.py
* curl -X POST http://localhost:5000/predict -H "Content-Type: application/json" -d "{\"features\": [80.0, 0.20, 200, 80.0, 0.25, 160], \"image\": [[[150, 150, 150], [150, 150, 150], [150, 150, 150], [150, 150, 150]], [[150, 150, 150], [150, 150, 150], [150, 150, 150], [150, 150, 150]], [[150, 150, 150], [150, 150, 150], [150, 150, 150], [150, 150, 150]], [[150, 150, 150], [150, 150, 150], [150, 150, 150], [150, 150, 150]]]}"
* curl -X POST http://localhost:5000/predict -H "Content-Type: application/json" -d "{\"features\": [85.0, 0.18, 145, 75.0, 0.22, 155], \"image\": [[[255, 0, 0], [255, 0, 0], [255, 0, 0], [255, 0, 0]], [[255, 0, 0], [255, 0, 0], [255, 0, 0], [255, 0, 0]], [[255, 0, 0], [255, 0, 0], [255, 0, 0], [255, 0, 0]], [[255, 0, 0], [255, 0, 0], [255, 0, 0], [255, 0, 0]]]}"
* curl -X POST http://localhost:5000/predict -H "Content-Type: application/json" -d "{\"features\": [500.0, 1.5, 1000, 500.0, 2.0, 500], \"image\": [[[0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]], [[0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]], [[0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]], [[0, 0, 0], [0, 0, 0], [0, 0, 0], [0, 0, 0]]]}"
* curl -X POST http://localhost:5000/predict -H "Content-Type: application/json" -d "{\"features\": [150.0, 0.5, 250, 100.0, 0.5, 250], \"image\": [[[0, 255, 0], [0, 255, 0], [0, 255, 0], [0, 255, 0]], [[0, 255, 0], [0, 255, 0], [0, 255, 0], [0, 255, 0]], [[0, 255, 0], [0, 255, 0], [0, 255, 0], [0, 255, 0]], [[0, 255, 0], [0, 255, 0], [0, 255, 0], [0, 255, 0]]]}"