Milestone-3

Streamlit-Based User Interface for PCB Defect Classification

Duration: Week 5 – Week 6

---

1. Milestone Title & Duration

Milestone-3 focuses on designing and implementing an interactive Streamlit web interface that enables end-users to upload multiple PCB images, process them using the trained YOLO model, visualize predictions, and download annotated outputs individually or as a complete batch in ZIP format.

This milestone transitions the project from backend model development to a user-friendly application layer, making the system accessible to non-technical users.

---

2. Objectives

The objectives of this milestone were:

- To build a simple, responsive, and user-friendly interface using Streamlit.
- To allow the user to upload multiple PCB images at once.
- To run the trained YOLO model on each uploaded image and generate:
    - Annotated output image (with bounding boxes and defect labels)
    - Classification results for each ROI
- To allow users to download individual annotated results.
- To generate a ZIP file containing all annotated outputs and prediction summaries for batch download.

This milestone ensures that the trained model can be used practically and efficiently by end-users.

---

3. Tasks Completed

During Milestone-3, the following tasks were completed:

- Designed a Streamlit layout with sections for file upload, processing, and result visualization.
- Integrated the YOLO model (trained in Milestone-2) into the Streamlit backend.

- Implemented support for multi-image uploading using Streamlit's `file_uploader` with `accept_multiple_files=True`.
- Added backend processing logic to run inference on each image and generate annotated outputs.
- Added individual download buttons for per-image results.
- Implemented a mechanism to compress outputs into a ZIP archive and make it downloadable.
- Added user-friendly status indicators such as progress bars, success messages, and processing logs.
- Ensured compatibility across different systems using Python virtual environments.

---

4. Technologies Used

Category	Tools / Frameworks
Frontend UI	Streamlit
Backend Model	YOLOv8-s (Ultralytics, PyTorch)
Image Processing	OpenCV
Inference Support	Python, Numpy
File Management	Zipfile library for batch ZIP downloads
Runtime	Local execution / Google Colab tunneling

The technologies were chosen to ensure ease of deployment, fast prototyping, and real-time interaction.

---

5. System Workflow & Implementation Summary

The user can select the model to be used either it can be best.pt or last.pt for prediction.

Below is a summary of how the Streamlit interface processes user-uploaded images:

Step 1: User Uploads PCB Images

- The interface accepts multiple images in JPG/PNG format.
- Uploaded files are temporarily stored in memory for processing.

Step 2: Inference Pipeline Execution

- For each uploaded image:
    - The model performs **defect localization and classification**.
    - Annotated images with bounding boxes and class labels are generated.
    - Classification summary is extracted.

Step 3: Results Display

- Each processed image is shown on the interface with:
    - Defect bounding boxes
    - Class names
    - Confidence scores
- The user can **scroll, inspect, and interact** with each result.
- Streamlit's column layout was used for cleaner visualization.

Step 4: Individual File Download

- A "Download Annotated Image" button appears below each processed output.
- This enables selective downloads.

Step 5: Batch ZIP Download

- All processed outputs are collected into a folder structure same YOLO follows.
- A ZIP archive is generated programmatically and provided for download.
- ZIP includes for each image:
    - All annotated images
    - An annotations txt file

---

6. Model Outputs (Displayed Through Streamlit)

Screenshots of the Streamlit interface, including upload section, processing logs, and output previews, will be inserted here.

The interface allows:

- Visual inspection of each detected defect
- Quick comparison across multiple images
- Seamless downloading of clean annotated results

This creates a smooth downstream workflow after Milestone-2 model training.

---

7. Challenges & How I Solved Them

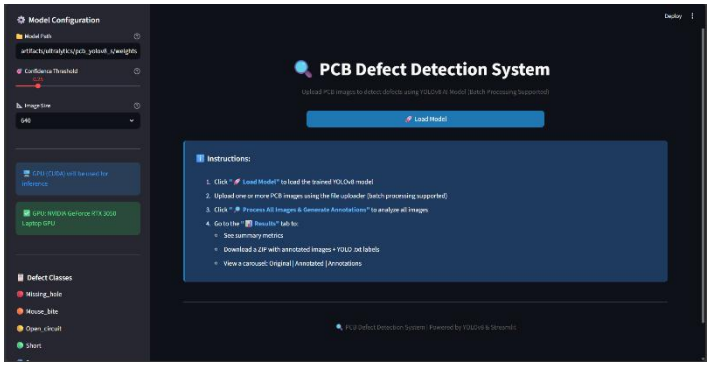
Several implementation-level challenges were encountered while building the interface:

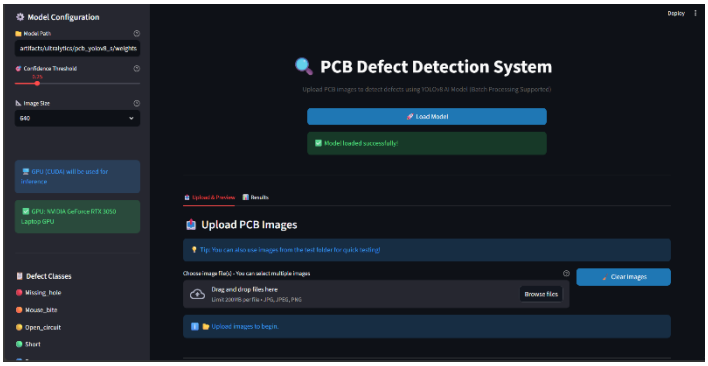
Challenge	Observation	Solution
Handling multiple images	Some images exceeded memory limits	Processed images sequentially + cleared buffers after each iteration
ZIP file creation issues	Temporary files not saving consistently	Used Python's `tempfile` + `zipfile` libraries for safer file handling

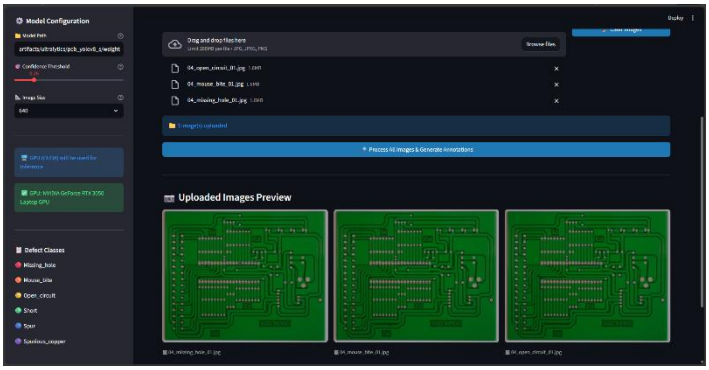
Challenge	Observation	Solution
Slow inference during batch processing	High number of uploaded images increased runtime	Added progress bars + optimized model loading (loaded once instead of per-image)
Large annotated images failing to download	Streamlit download button limit	Compressed images before packaging to ensure smooth download

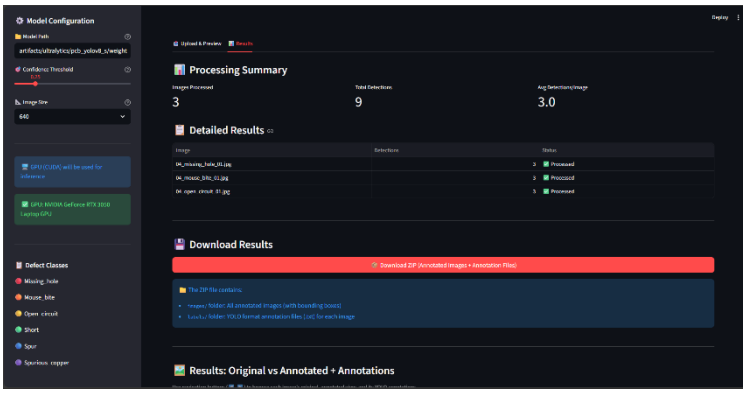
These solutions improved performance, user experience, and system reliability.

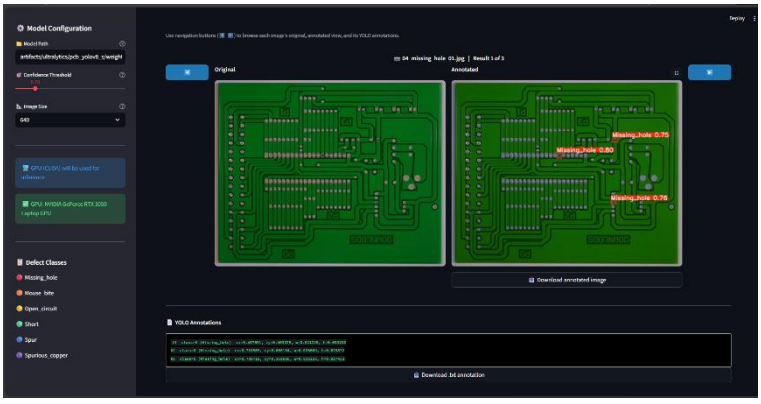
Results:











Summary of Milestone-3

Milestone-3 successfully transformed the PCB defect classification model into a fully interactive, user-ready application. With the Streamlit interface, users can now:

- Upload multiple PCB images in one step
- Receive annotated results instantly
- Download individual outputs or an entire batch

This milestone bridges the gap between deep learning model development and practical real-world usability. The interface can be extended further for industrial deployment, API integration, or real-time factory-floor inspection systems.

The completion of Milestone-3 marks the final stage of the core project cycle:

Dataset Preparation → Model Training → User-accessible Application.