Milestone-2 Model Training & Evaluation

Duration: Week 3 – Week 4

---

1. Milestone Title & Duration

Milestone-2 focuses on training and evaluating a deep learning model capable of identifying the exact PCB defect type from extracted Regions of Interest (ROIs). This phase builds directly on the dataset preparation workflow completed in Milestone-1 and represents the core learning stage of the project.

---

2. Objective of Milestone-2

The objective of this milestone was to:

- Train a deep learning model that can **highlight defect-affected regions on the PCB images**
- **Classify the detected region among six PCB defect categories** :
  Missing hole, Mouse bite, Open circuit, Short, Spur, and Spurious copper.

This stage marks the transition from image preprocessing to true learning-based defect recognition.

---

3. Tasks Completed

The following tasks were successfully carried out during Milestone-2:

- Chose **YOLO (Ultralytics) with PyTorch backend** as the training architecture.
- Converted extracted ROIs into a structured dataset for multi-class classification.
- Implemented data augmentation techniques to avoid overfitting.
- Trained the model using Google Colab GPU due to high computational requirements.
- Evaluated the trained model using precision, recall, F1-score, and accuracy metrics.
- Generated confusion matrix and classification reports for validation and test sets.

---

4. Technologies and Hyperparameters Used

The following tools, frameworks, and settings were used during training:

Tools / Frameworks

- Python
- PyTorch
- Ultralytics YOLO
- Google Colab (T4 GPU)
- Matplotlib & Seaborn for metric visualization

Key Hyperparameters

Hyperparameter	Value
Model	YOLOv8-s
Image size	640 × 640
Optimizer	Adam
Learning rate	0.0001
Batch size	16
Epochs	50
Loss functions	Classification loss + Box loss

---

5. Training & Testing Summary

During the model fitting stage, fine-tuning was adopted rather than complete training from scratch. Fine-tuning allows the model to retain the useful low-level feature representations learned from generic vision datasets while customizing only the deeper layers to PCB defect classification.

To implement this, the first 10 layers of the YOLO backbone were frozen, meaning their weights remained unchanged throughout the training process. Only the remaining network layers (the classification and detection heads) were trained. This strategy ensures: - Efficient convergence - Reduced risk of catastrophic forgetting - Faster training time compared to full training - Better adaptation to PCB defect patterns

Fine-tuning Configuration

Configuration Parameter	Value
Fine-tuning	Enabled
Frozen layers	First 10 backbone layers
Epochs	50
Initial learning rate	0.0001
Batch size	16
Image size	640 × 640

This setup allowed the network to simultaneously focus on defect region localization and defect type identification, while leveraging previously learned generic visual filters such as edge, contour, and texture recognition. The learning rate of `0.0001` ensured gradual weight updates during training to avoid instability, while `50 epochs` provided sufficient training time for full convergence.

The training process focused on multi-class defect recognition rather than binary defect detection. Evaluation metrics such as precision, recall, and F1-score were tracked after every epoch to monitor learning consistency.

YOLO automatically reports:

- **Classification loss** — measures the accuracy of predicting the correct defect class
- **Bounding box (box) loss** — measures localization accuracy of predicted ROI
- **Accuracy, precision, recall, and F1-score** — final indicators of multi-class prediction performance

Google Colab GPU (NVIDIA T4) was used due to the computational load during training and to speed up convergence compared to CPU-based execution.

---

6. Model Outputs

The trained model demonstrated extremely high performance during both validation and testing.

Confusion Matrix (Validation Results)

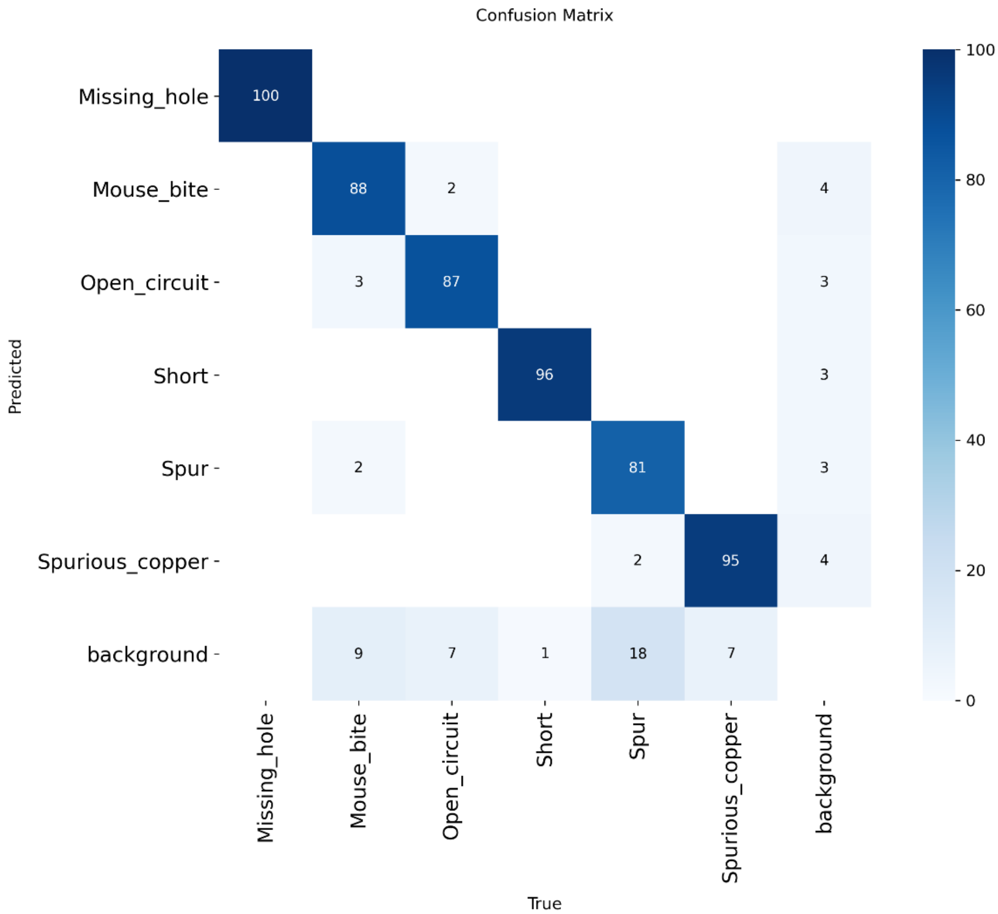

Figure 1: Confusion Matrix of YOLO

Classification Results — Validation Set

```
Overall Accuracy: 99.00%
Missing_hole ............ Precision: 1.00 | Recall: 1.00 | F1-score: 1.00
Mouse_bite ............. Precision: 1.00 | Recall: 0.94 | F1-score: 0.97
Open_circuit ........... Precision: 0.94 | Recall: 1.00 | F1-score: 0.97
Short .................. Precision: 1.00 | Recall: 1.00 | F1-score: 1.00
Spur ................... Precision: 1.00 | Recall: 1.00 | F1-score: 1.00
Spurious_copper ........ Precision: 1.00 | Recall: 1.00 | F1-score: 1.00
```

Classification Results — Test Set

```
Overall Accuracy: 100.00%
All six defect categories achieved perfect Precision, Recall, and F1-score.
```

These metrics show that the trained model is highly robust and generalizes well to unseen data.

Results Sample :

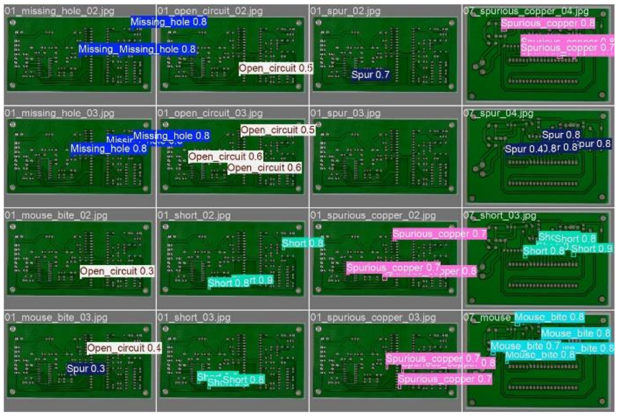

Figure 2predicted results

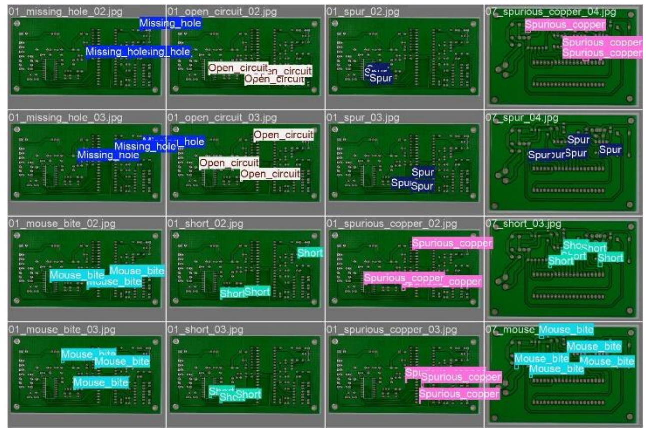

Figure 3: Expected Results

7. Challenges & How I Solved Them

Although the training pipeline was successful, a few practical bottlenecks were encountered:

Challenge	Observation	Solution
ROI labels were originally in XML format	Ultralytics YOLO only accepts `.txt` labels	Implemented automated XML → TXT conversion script with bounding box normalization
Long training duration on CPU	Training extremely slow locally	Shifted to Google Colab T4 GPU for 20× faster training

These solutions ensured stable learning and prevented bias in the trained classifier.

Summary of Milestone-2

Milestone-2 successfully completed the most crucial part of the PCB defect classification system high-accuracy learning. Using the YOLO architecture trained on labeled ROIs, the model can now identify and classify PCB defects among six categories with extremely high reliability.

Both validation and testing metrics confirmed near-perfect generalization, proving that the preprocessing pipeline of Milestone-1 and the model architecture chosen in Milestone-2 are well-aligned.