

# Road Damage Detection and Classification Using YOLOv8 and MobileNetV2

Nasim Abdirahman Ismail

# PROBLEM STATEMENT

- Poor road conditions cause accidents
- Manual inspection is slow and expensive
- Need for automated AI solutions

# PROJECT OBJECTIVES

## Objectives:

- Detect road damage locations
- Classify damage types
- Build an end-to-end AI pipeline
- Deploy a user-friendly interface

# DATASET DESCRIPTION

**Dataset: Road Damage Dataset 2022 (RD2022)**

- Source: Kaggle
- Total images: 26,870 images
- Countries: 6 countries
- Damage classes: 5 classes
- Annotation format: YOLO bounding boxes

# ROAD DAMAGE CATEGORIES & ANNOTATIONS

**Total damage classes: 5**

- Classes defined by road condition standards
- D00: Longitudinal cracks
- D10: Transverse cracks
- D20: Alligator/fatigue cracks
- D40: Potholes
- Other: Non-critical or miscellaneous damage
- Each damage is annotated using bounding boxes



# END-TO-END DETECTION & CLASSIFICATION WORKFLOW

## 1. Input Image

Upload road image via Streamlit

## 2. YOLOv8 Detection

Detect damage regions (bounding boxes)

## 3. ROI Extraction

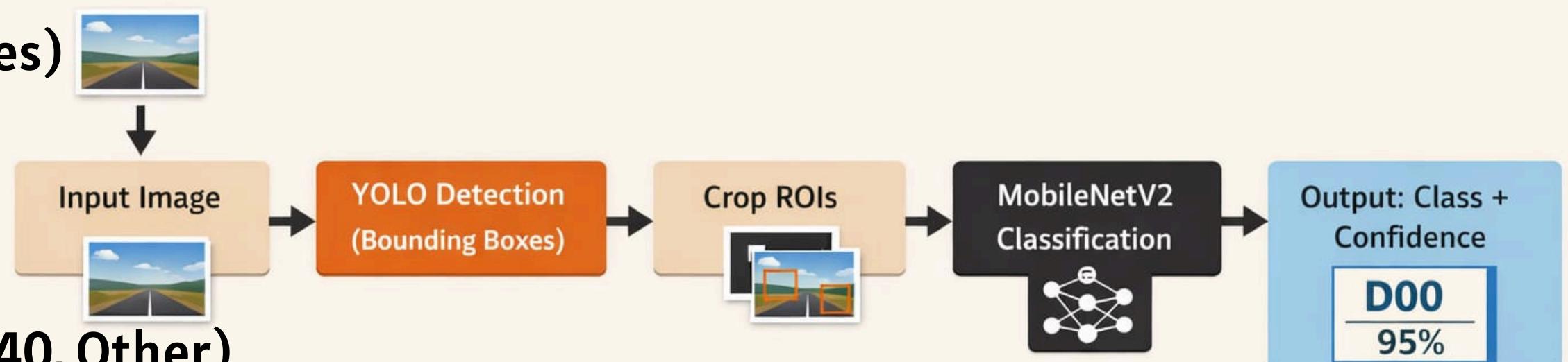
Crop detected damage areas

## 4. MobileNetV2 Classification

Classify damage type (D00, DI0, D20, D40, Other)

## 5. Final Output

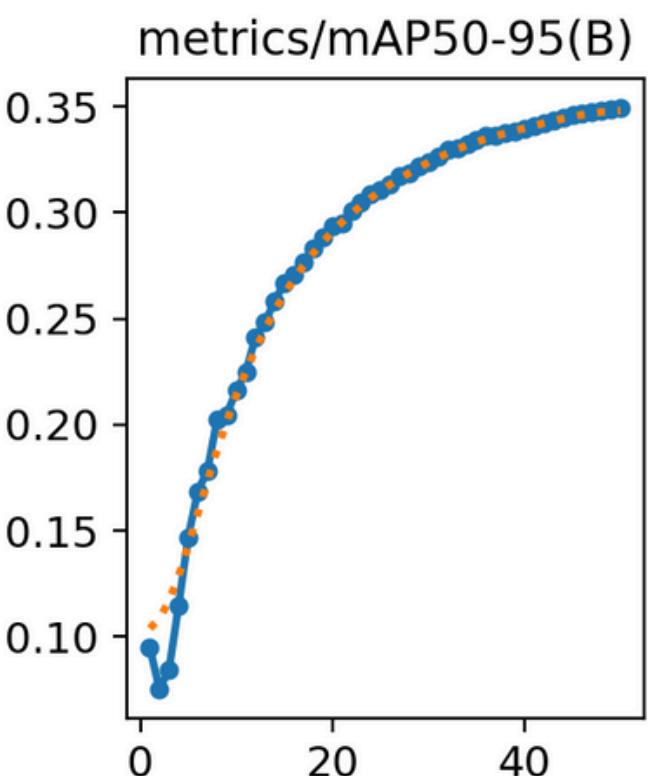
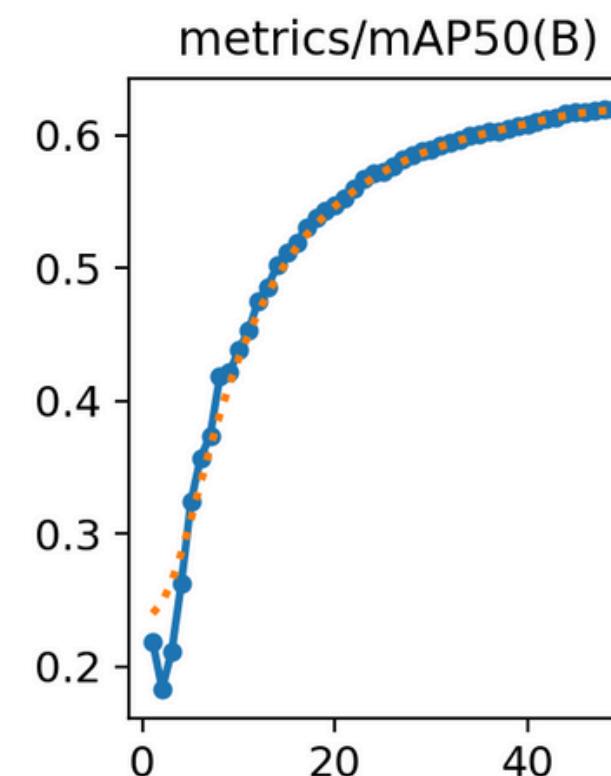
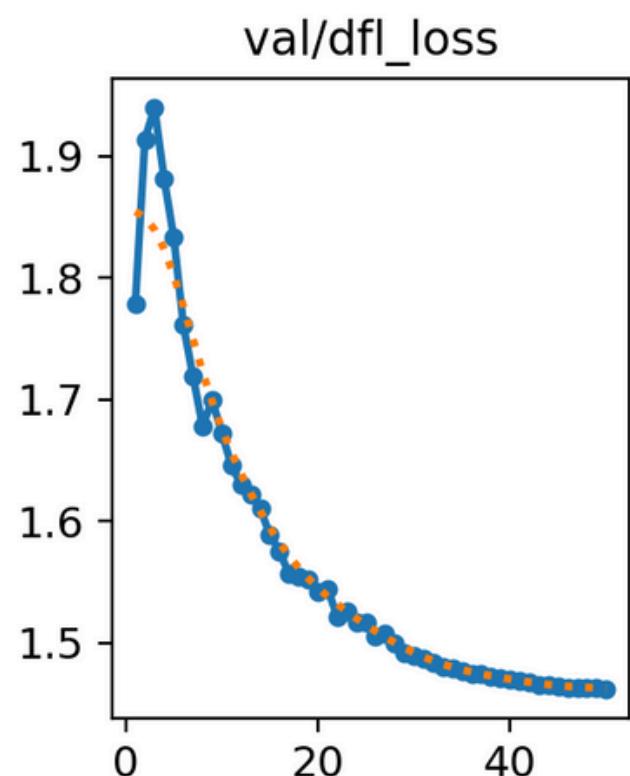
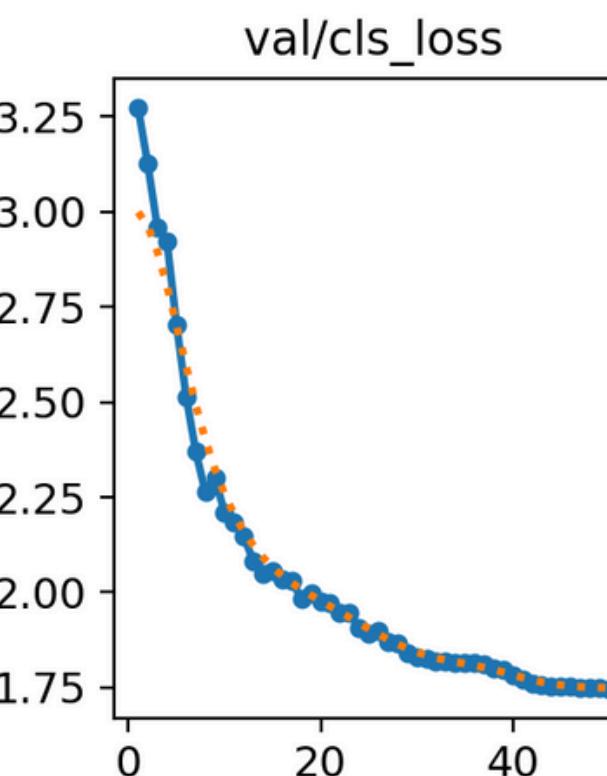
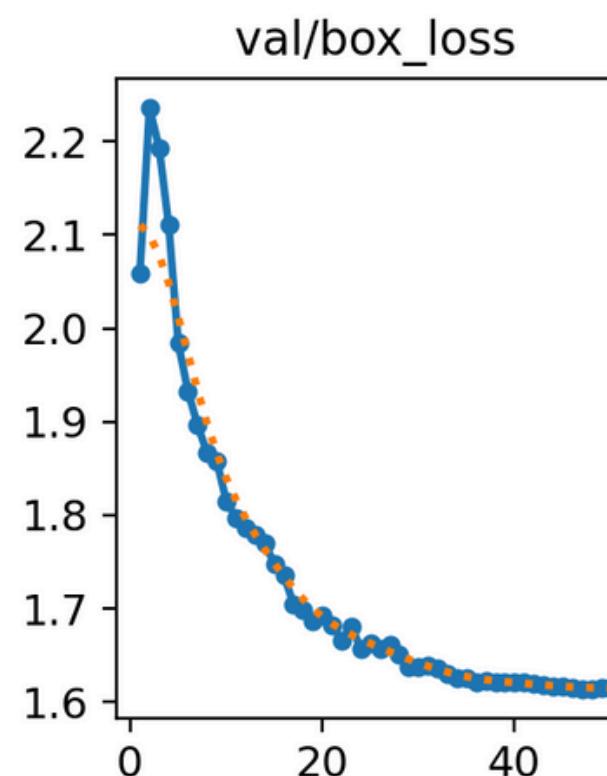
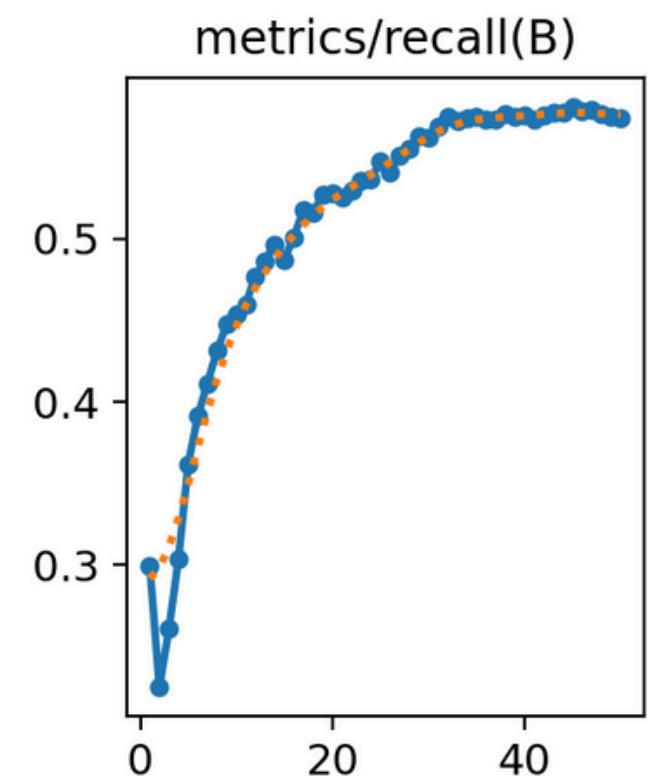
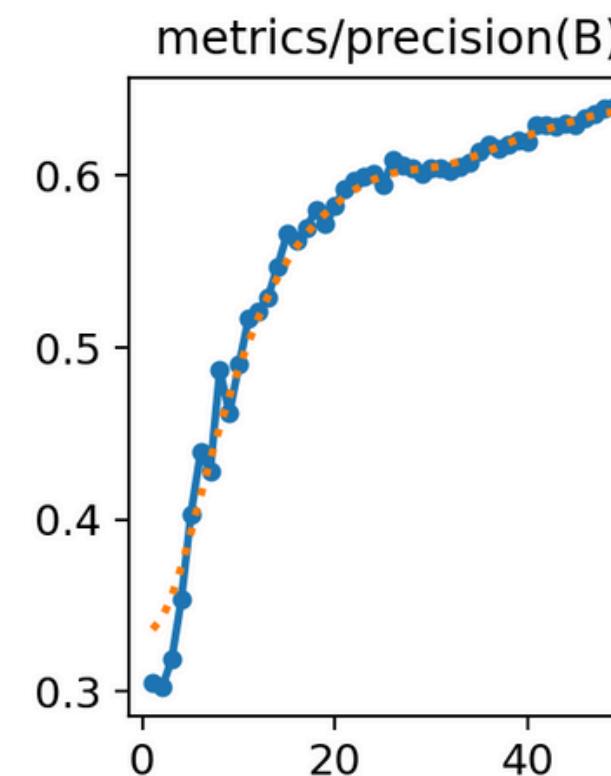
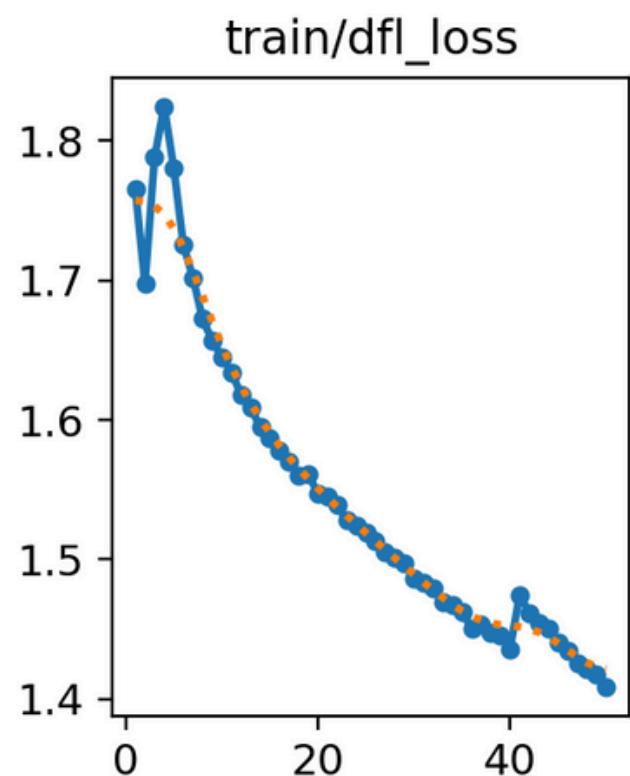
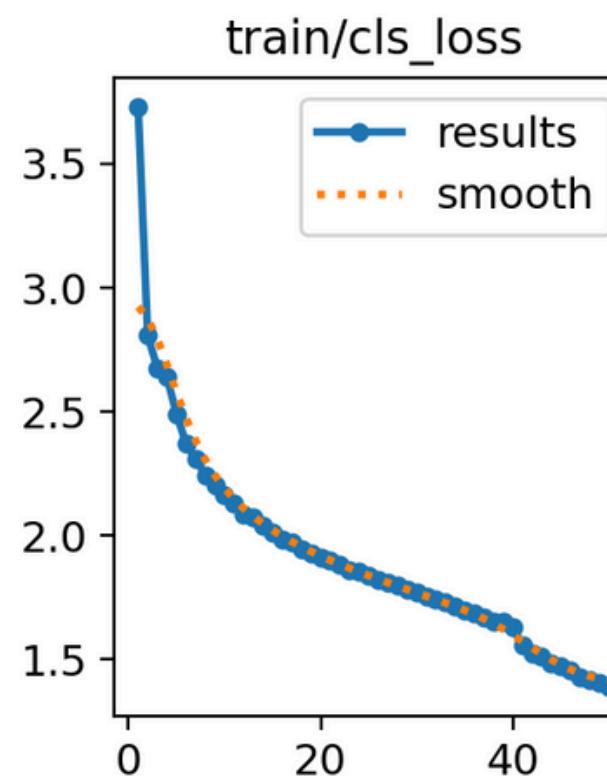
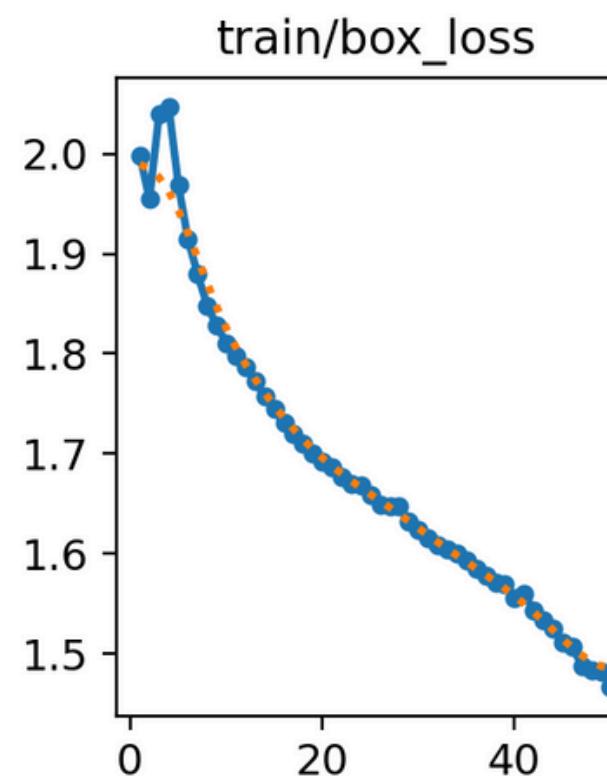
Detection + class label + confidence

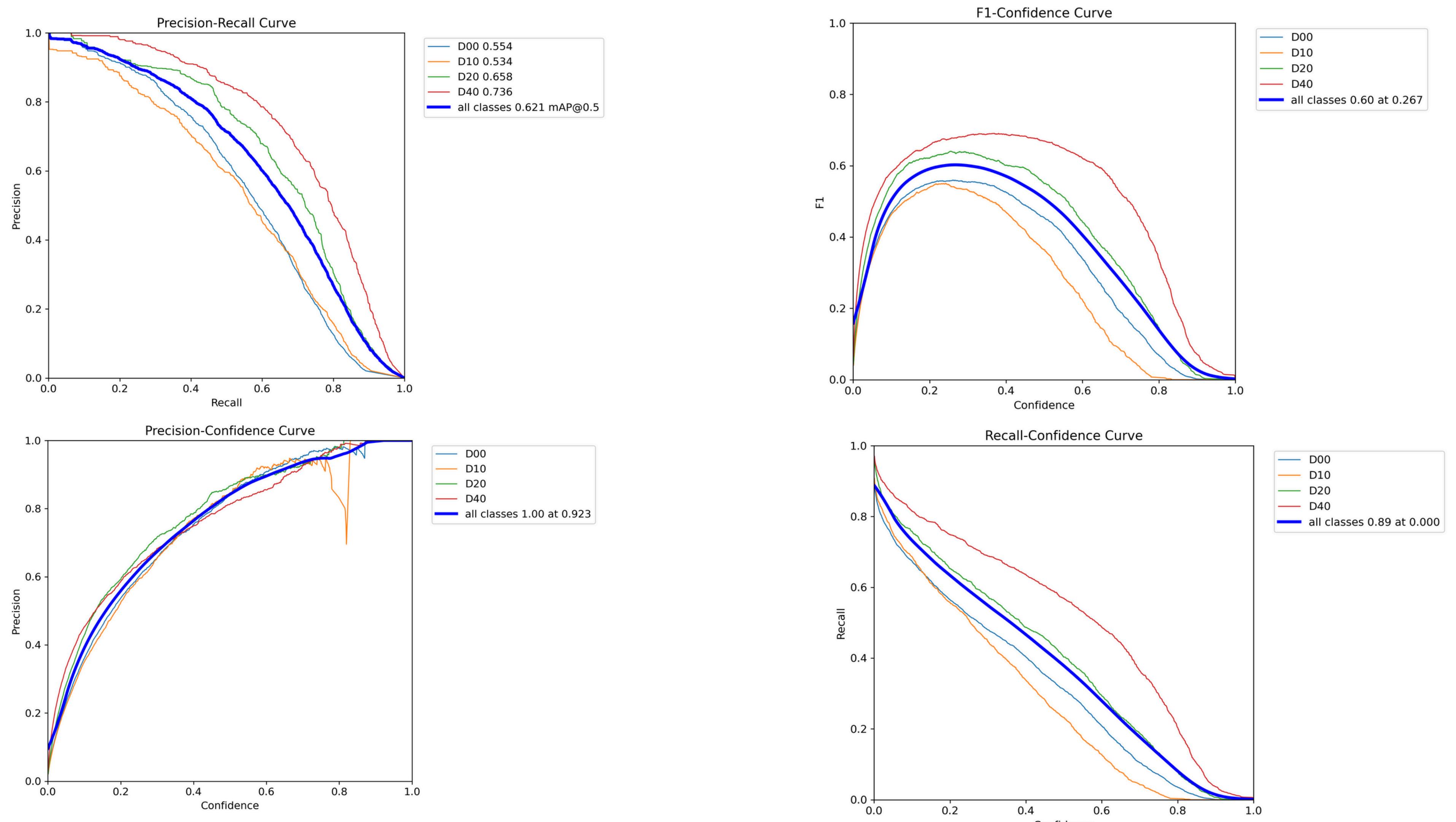


# YOLOV8 DETECTION RESULTS

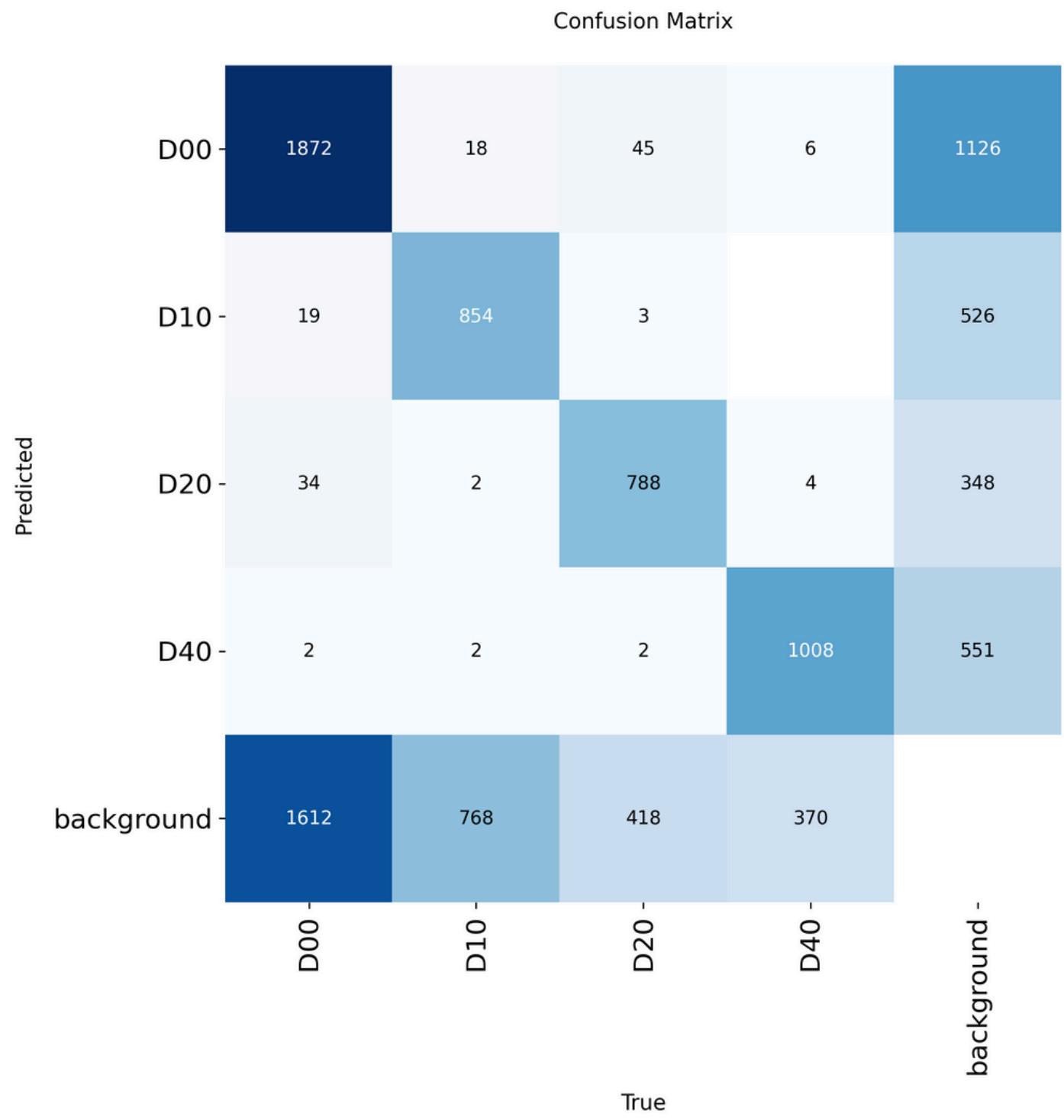
- YOLOv8 detects multiple road damage regions in a single image
- Bonding boxes indicate the location of detected damage
- Each box is associated with a predicted damage class
- Confidence scores reflect detection reliability
- Results are visualized directly on the input image

# YOLOV8 DETECTION RESULTS AND TRAINING CURVES



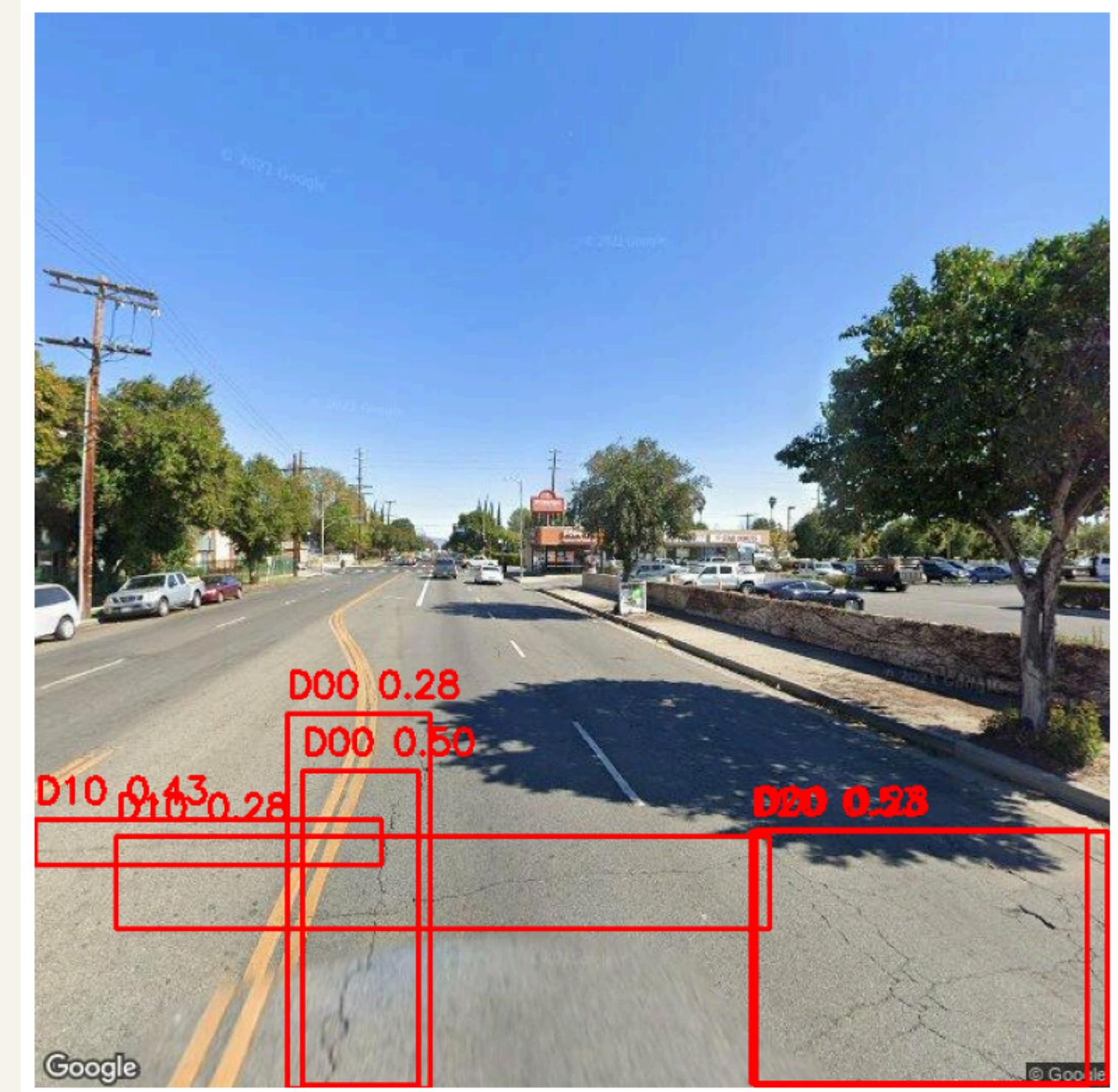


# YOLOV8 CONFUSION MATRIX ANALYSIS



# OPENCV PROCESSING

- OpenCV was used for image preprocessing:
- Image resizing
- Color space conversion
- Preparing detected regions for MobileNetV2
- OpenCV is used to crop and process the YOLO-detected regions before Mobilenet classification

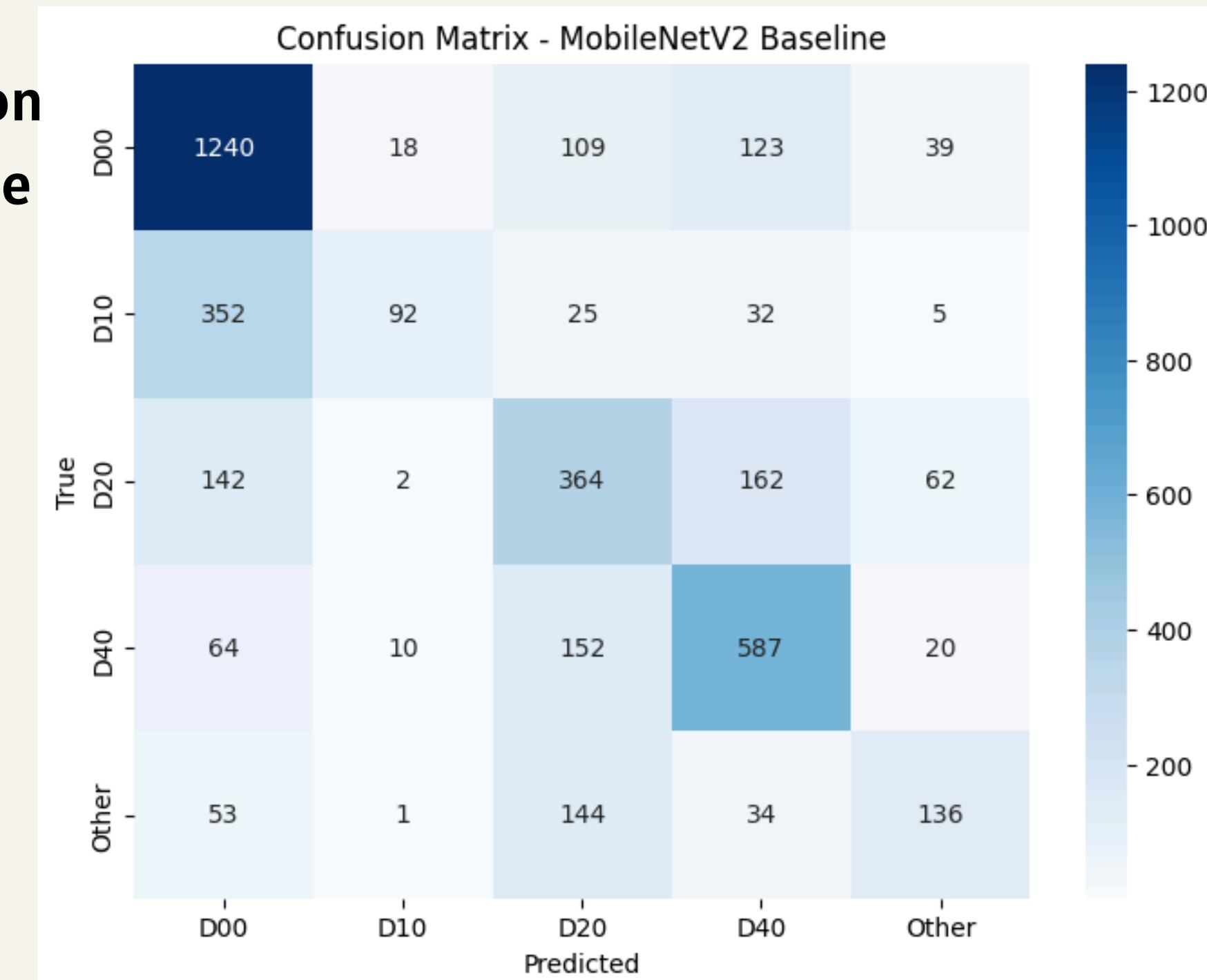


# MOBILENETV2 CLASSIFICATION RESULTS

11

- Cropped damage regions are classified using MobileNetV2
- Each detected region is assigned a specific damage type
- Supported classes: D00, D10, D20, D40, and Other
- The model outputs class probabilities for each region
- The final decision is based on the highest confidence score

- The confusion matrix summarizes the classification performance across all detected region



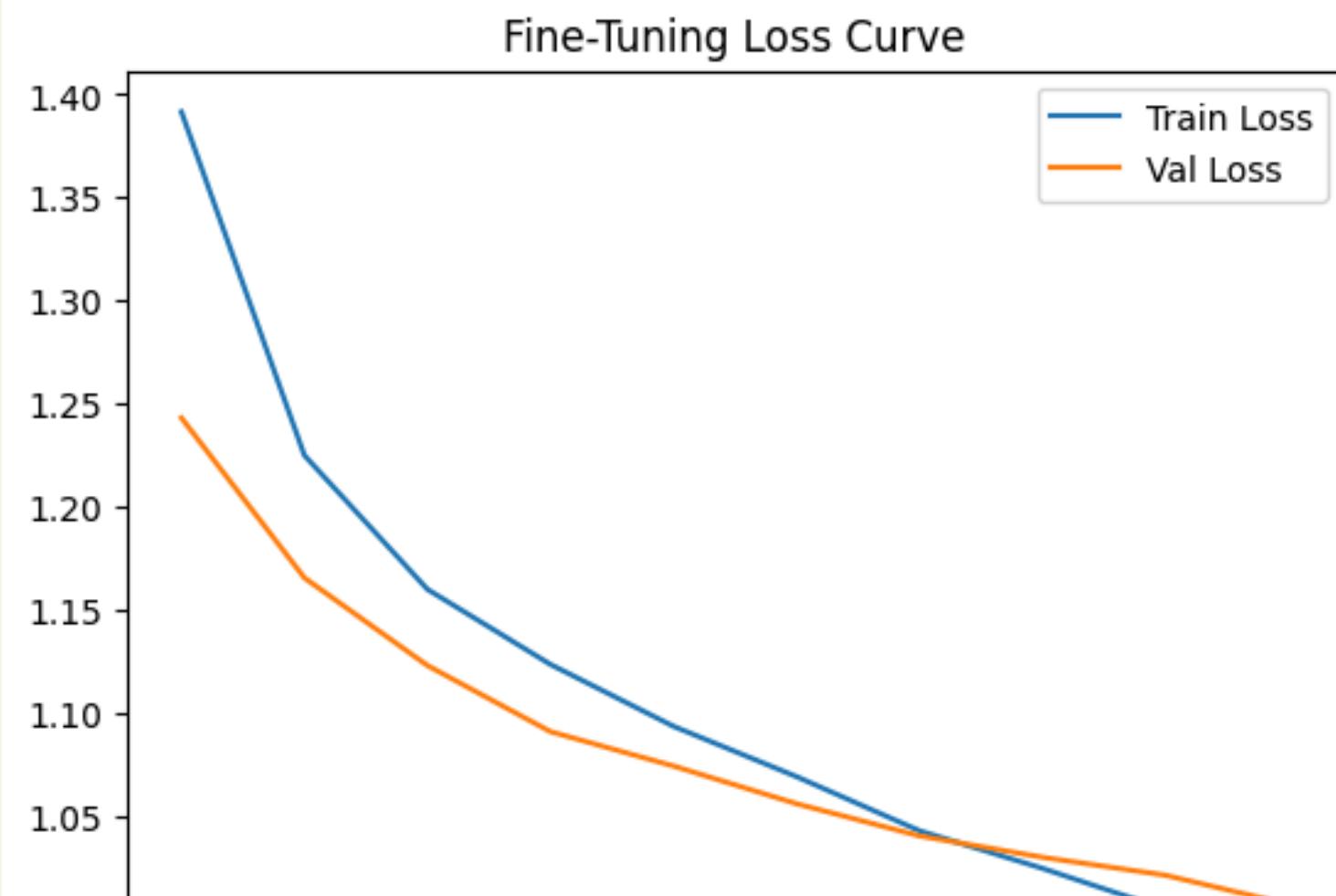
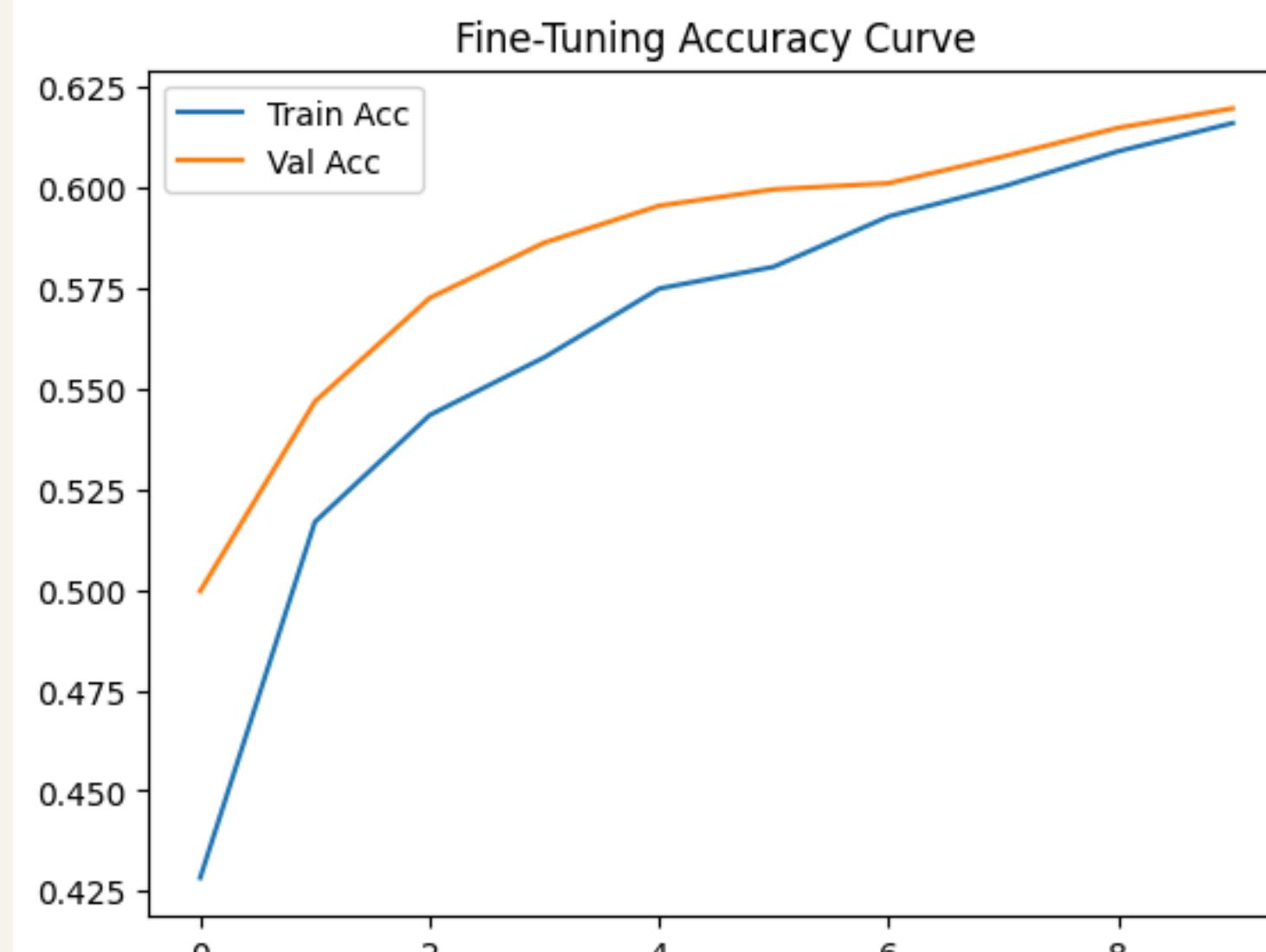
# MOBILENETV2 FINE-TUNING ACCURACY COMPARISON

## Training Setup:

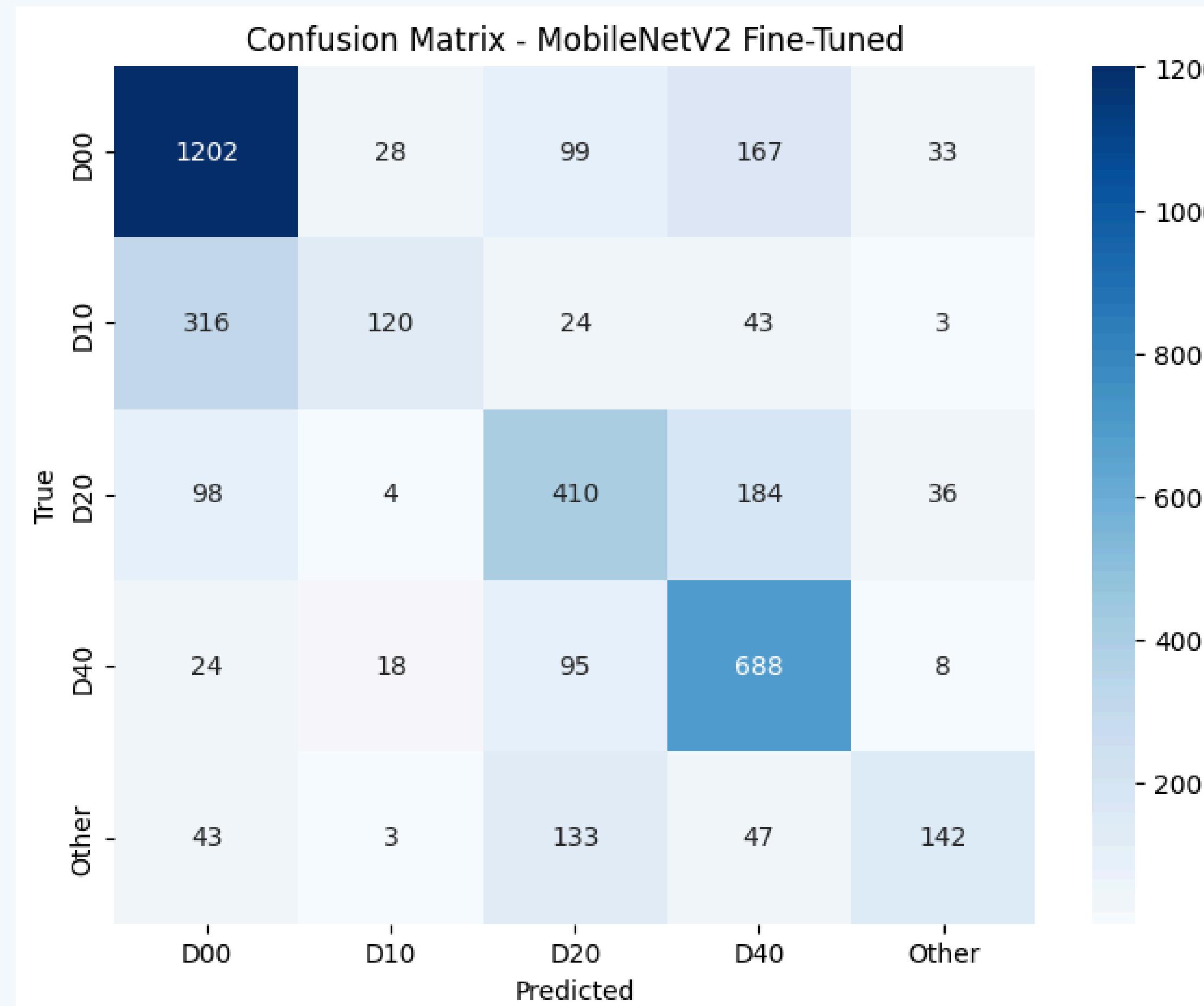
- MobileNetV2 pre-trained on ImageNet
- Transfer learning applied
- Fine-tuning by unfreezing last layers
- Low learning rate for stable training

## Accuracy Results:

- Before Fine-Tuning: ~59% validation accuracy
- After Fine-Tuning: ~62–63% validation accuracy



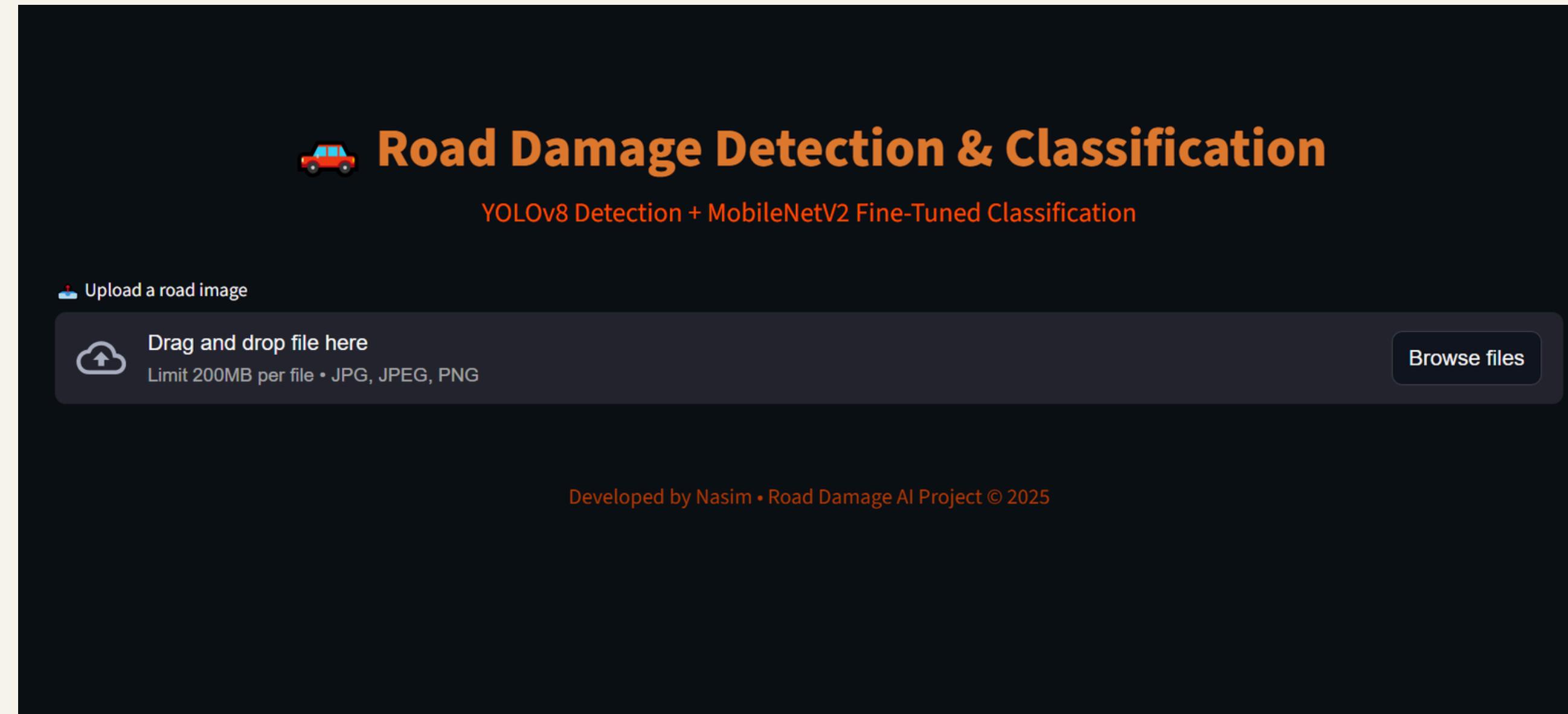
# CLASSIFICATION PERFORMANCE AFTER FINE-TUNING



# STREAMLIT DEPLOYMENT SLIDE

14

- user uploads road images
- Model runs detection and classification
- visual results displayed
- Easy to use interface



# CONCOLUSION

15

- YOLOv8 successfully detects road damage regions
- MobileNetV2 improves classification accuracy
- Fine-tuning enhances model performance
- System can support road maintenance applications

# **THANK YOU**

**Presented By: Nasim Abdirahman Ismail**