Car Damage Detection

Computer Vision - Final Project

The Problem

01 Rental agencies, parking services, and insurance companies handle thousands of vehicles daily, but lack the manpower or budget for inspections—leading to missed damages and financial disputes.

02 There is a growing demand for real-time, automated damage detection tools that can quickly and accurately classify vehicle conditions using image inputs, particularly for digital-first insurance and resale platforms.

03 Traditional car damage assessments rely heavily on manual inspection, which is often slow, subjective, and prone to human error—especially in high-traffic environments like insurance claims or rental returns.

The Solution

An Automated, Explainable, and Efficient Solution for Vehicle Assessment

Overview

This program leverages MobileNetV2-based transfer learning to classify car damage types and locations (Front/Rear – Crushed/Breakage/Normal) with high speed and accuracy.

MobileNetV2 is a lightweight, convolutional neural network architecture optimized for embedded vision applications. It improves upon the original MobileNet by introducing inverted residual blocks and linear bottlenecks, resulting in higher accuracy and speed while maintaining low computational costs.

Project Flow at a Lens

Finding the right data

Split dataset into two sets, train (80%) and test (20%)

Installing relevant libraries and extensions

Preprocessing of data, Model building and Data generators

THE BIG PART! Model Training

Model Evaluation and Predictions Visualizations

Finding the Right Data

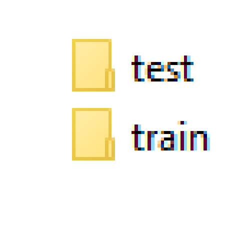
6 classes:

F_Breakage, F_Crushed, F_Normal

R_Breakage, R_Crushed, R_Normal

https://www.kaggle.com/datasets/samwash94/comprehensive-car-damage-detection

Downloading and Splitting the Data



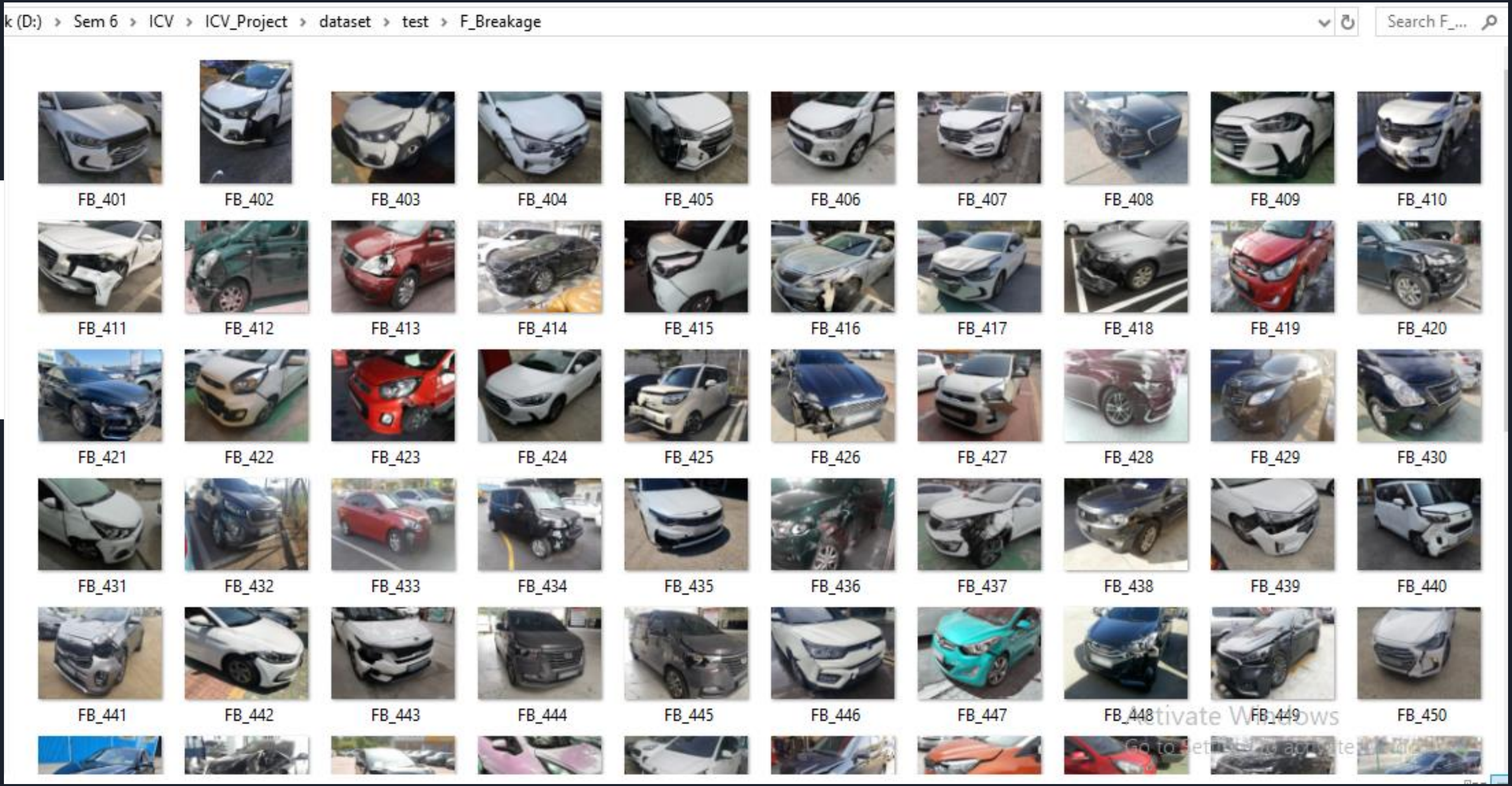

k (D:) › Sem 6 › ICV › ICV_Project › dataset › test › F_Breakage

Image Preprocessing

Histogram equalization - light adjustment and improved contrast

Image resizing to standard dimensions (224x224)

Color space conversion - BGR to RGB for MobileNetv2

Normalization - pixel values from 0-255 to 0-1 for stabilising gradients

Edge Detection and Contour Analysis - read and resize - convert to grayscale - Gaussian blur

Canny edge detection to highlight damages

Contour visualization to identify structural damage

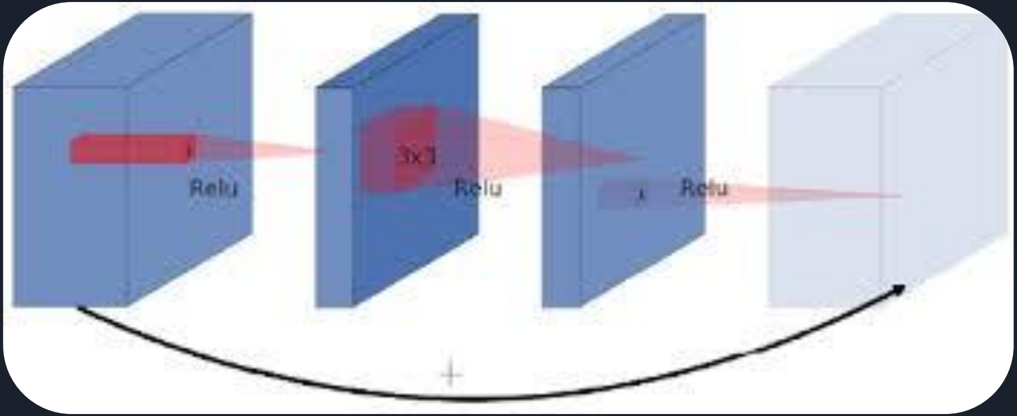
Model Architecture

Transfer learning with MobileNetV2 (faster than VGG)

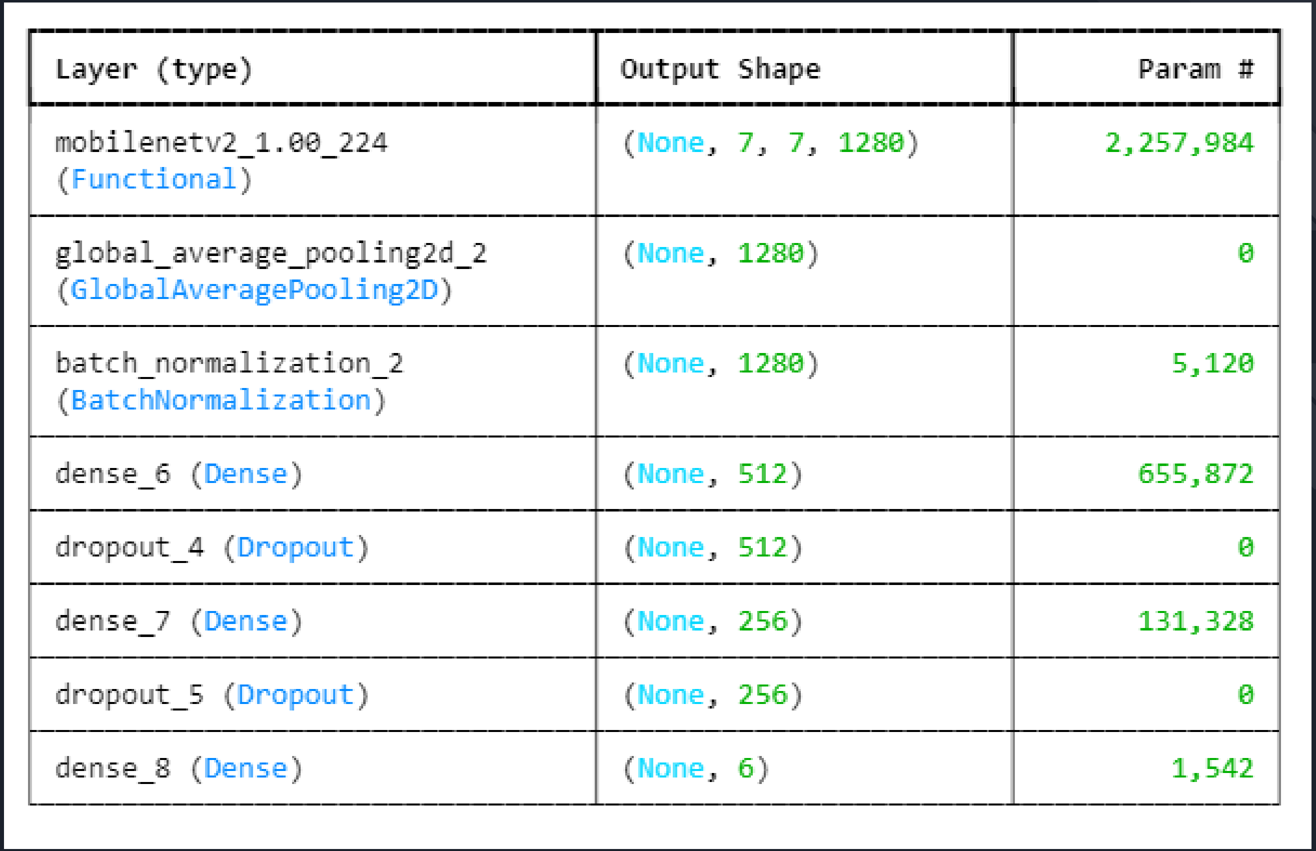
Custom classification head with dropout for regularization

Global Average Pooling

Batch normalization for more stable training

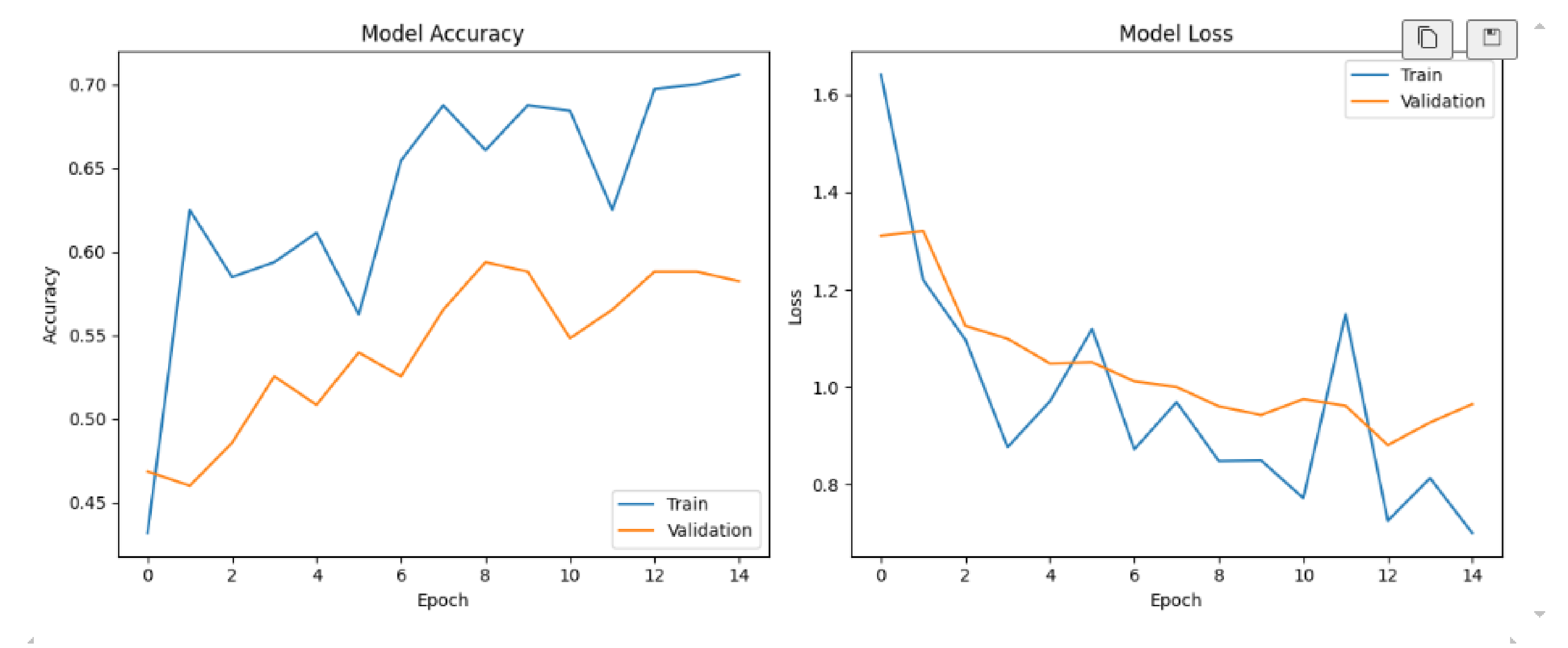


Layers



Layer (type)	Output Shape	Param #
mobilenetv2_1.00_224 (Functional)	(None, 7, 7, 1280)	2,257,984
global_average_pooling2d_2 (GlobalAveragePooling2D)	(None, 1280)	0
batch_normalization_2 (BatchNormalization)	(None, 1280)	5,120
dense_6 (Dense)	(None, 512)	655,872
dropout_4 (Dropout)	(None, 512)	0
dense_7 (Dense)	(None, 256)	131,328
dropout_5 (Dropout)	(None, 256)	0
dense_8 (Dense)	(None, 6)	1,542

Model Training



Model Training

Build CNNs

Load train data, validation data and test data

Prepare Data Generators

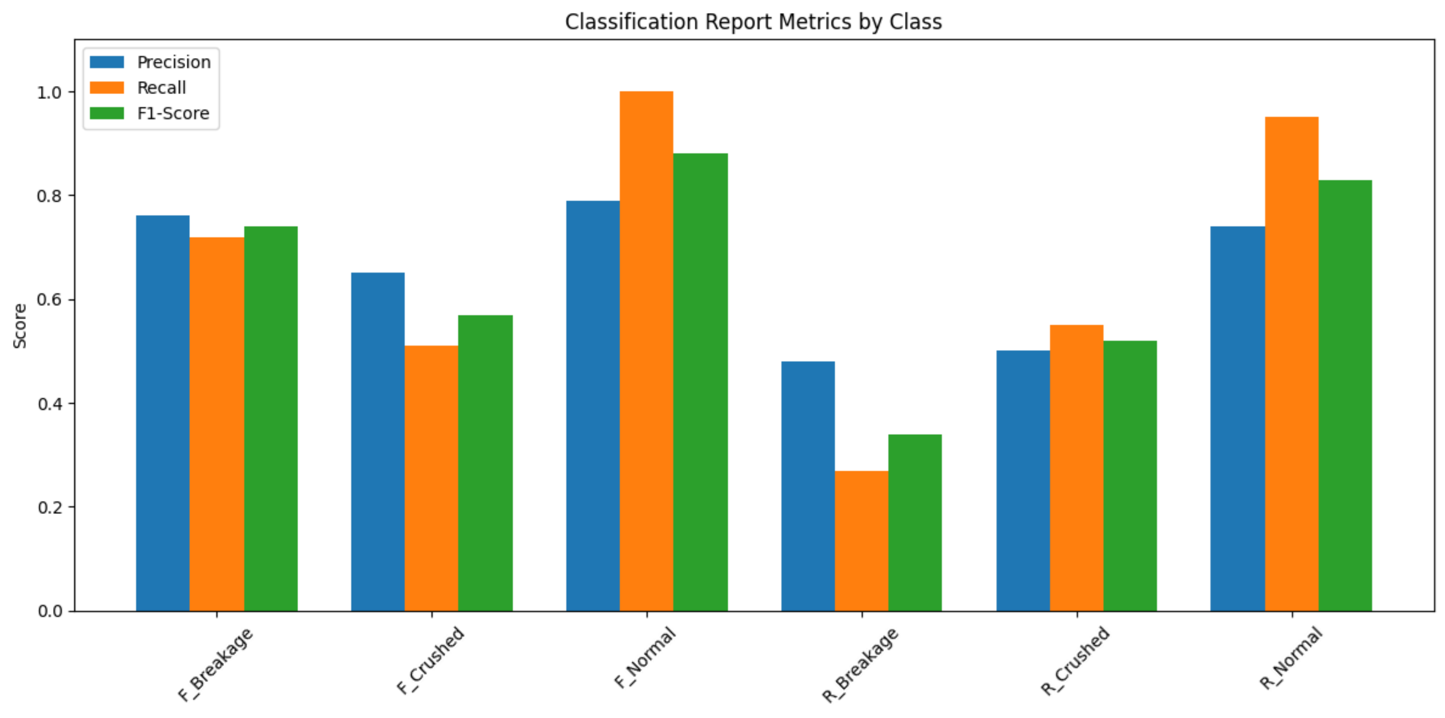
Define training callbacks

Early Stopping - monitor, patience, restore_best_weights

Model Checkpoints

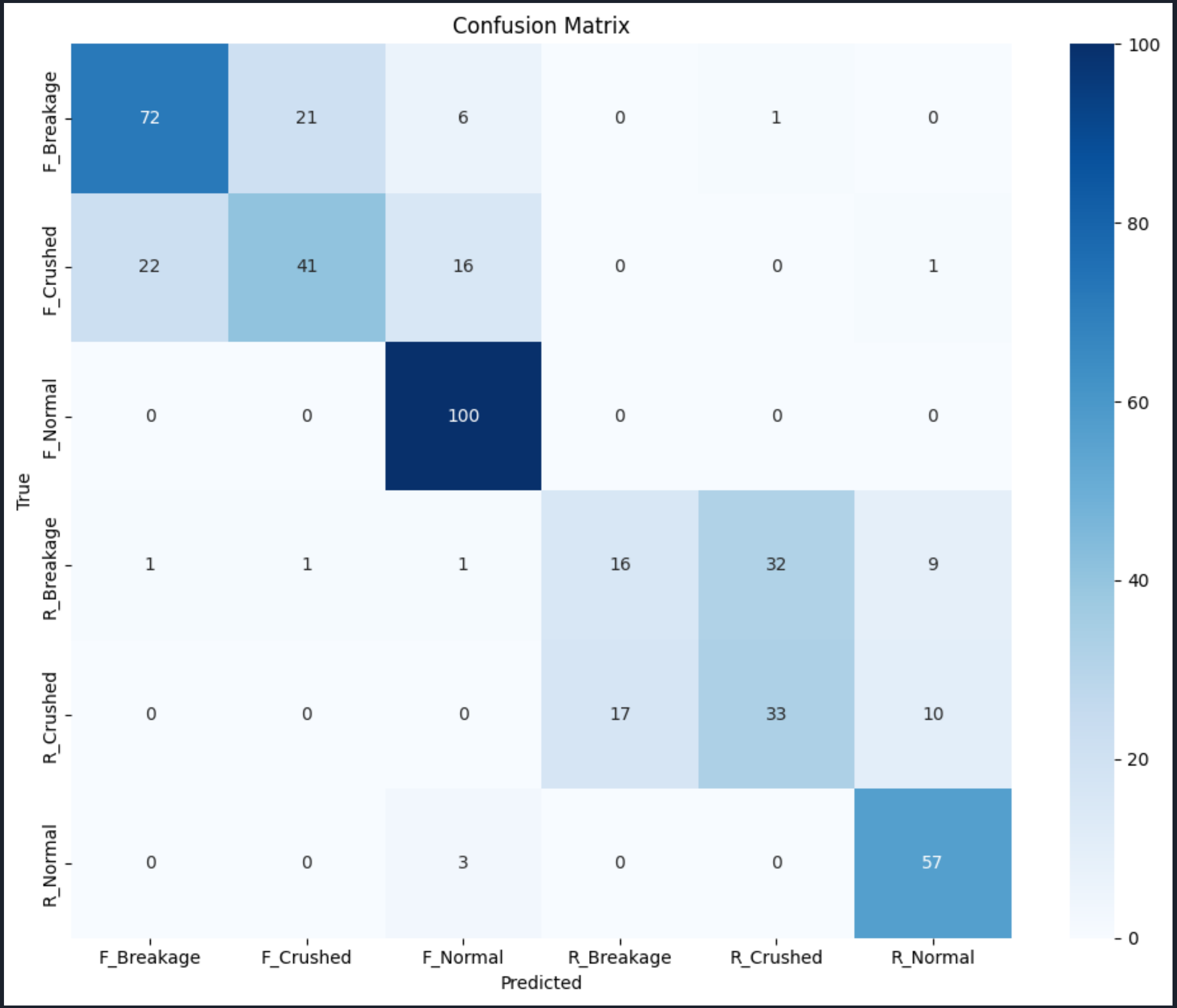
Model Evaluation

Classification Report



Model Evaluation

Confusion Matrix



Confusion Matrix

True \ Predicted	F_Breakage	F_Crushed	F_Normal	R_Breakage	R_Crushed	R_Normal
F_Breakage	72	21	6	0	1	0
F_Crushed	22	41	16	0	0	1
F_Normal	0	0	100	0	0	0
R_Breakage	1	1	1	16	32	9
R_Crushed	0	0	0	17	33	10
R_Normal	0	0	3	0	0	57

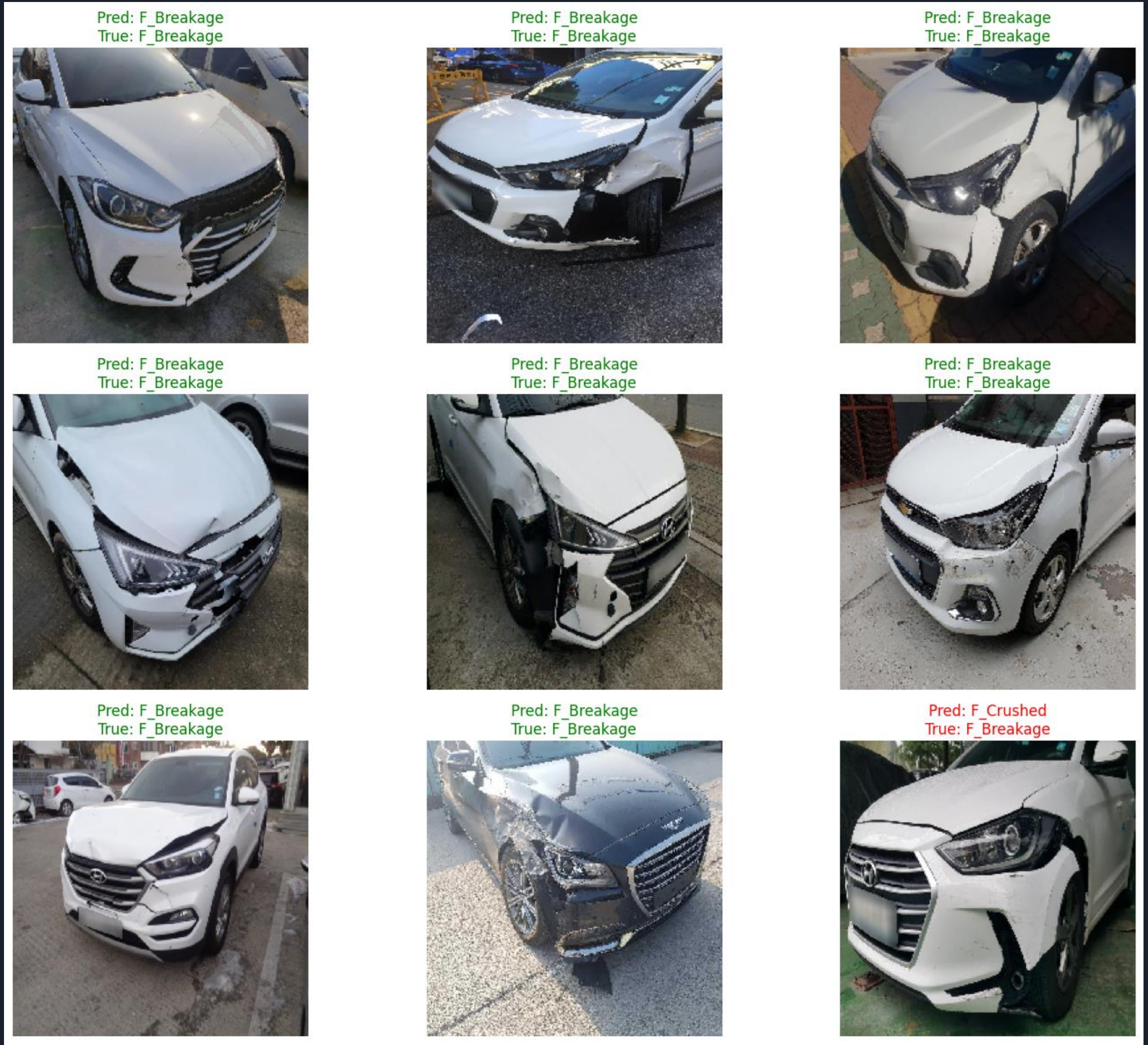
Prediction Visualisations

Prediction Visualization: Visual comparison of predicted vs. actual classes with intuitive red/green color-coding.

Feature Map Viewer: Understand how the model "sees" image patterns in the early layers using intermediate feature maps.

Grad-CAM Heatmaps: Identify which areas of the car influenced the model’s decision using colored attention overlays.

Prediction Visualizations



Pred: F_Breakage
True: F_Breakage

Pred: F_Breakage
True: F_Breakage

Pred: F_Breakage
True: F_Breakage

Pred: F_Breakage
True: F_Breakage

Pred: F_Breakage
True: F_Breakage

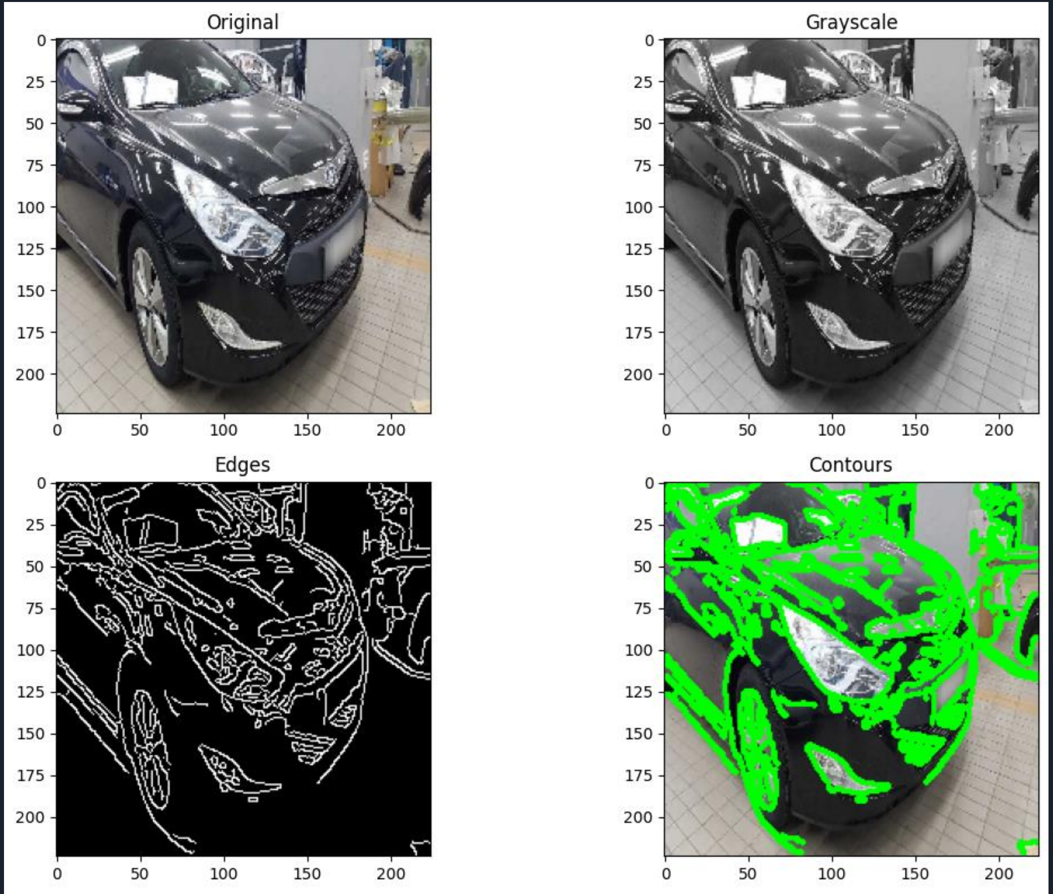
Pred: F_Breakage
True: F_Breakage

Pred: F_Breakage
True: F_Breakage

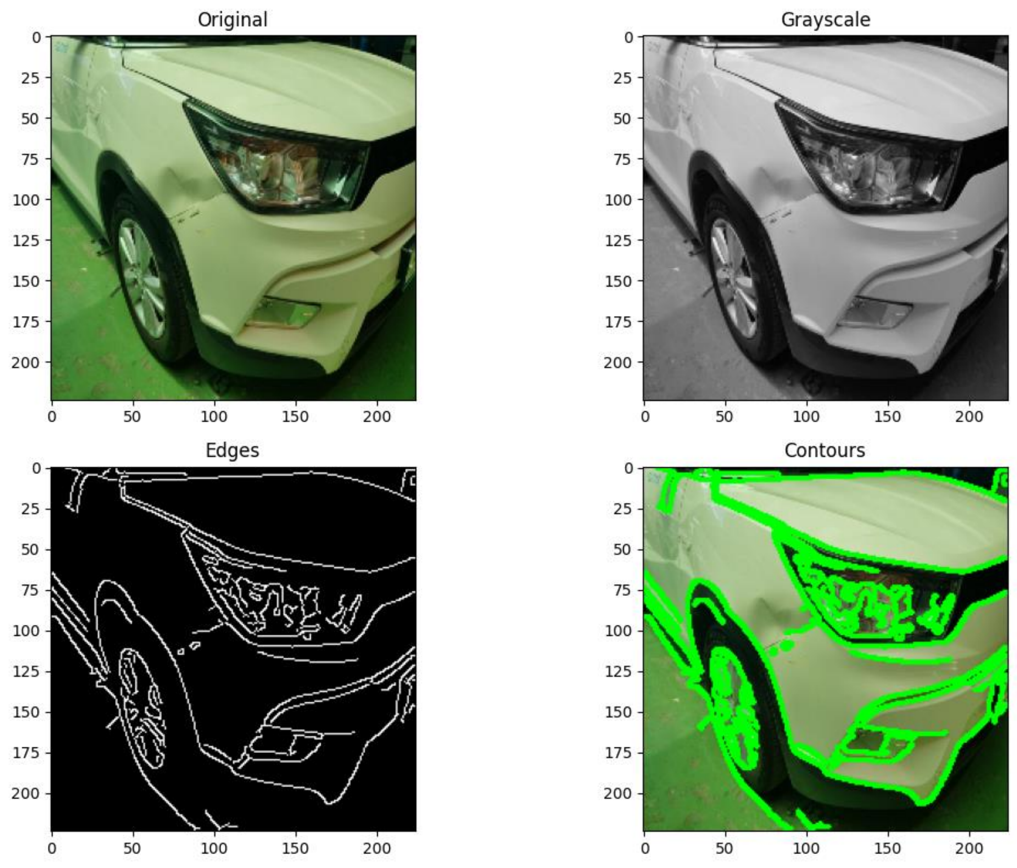
Pred: F_Breakage
True: F_Breakage

Pred: F_Crushed
True: F_Breakage

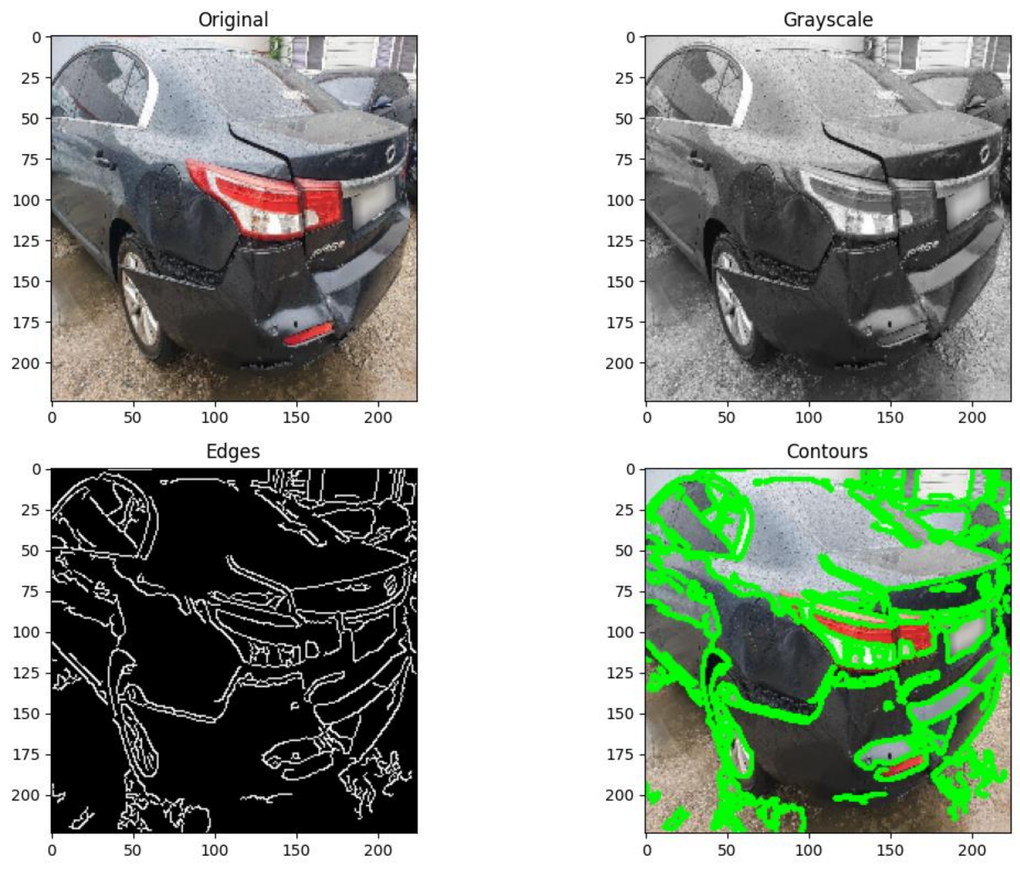
Analyzing a Single Image - Prediction Visualizations



Analyzing a Single Image - Prediction Visualizations



Analyzing a Single Image - Prediction Visualizations



Thank You!

Anusha Randhawa