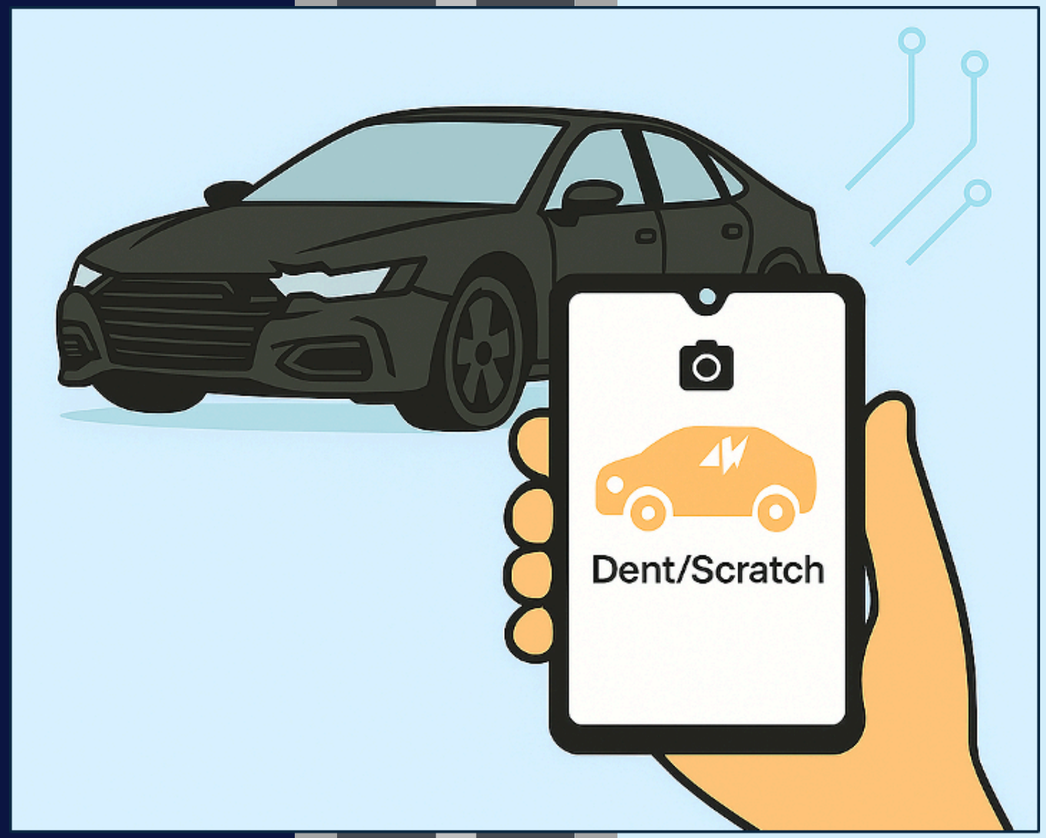
Car Damage Classification



Prepared by:
Ahlam Alqahtani
Walaa Saif Aleslam
Rama Khalid Alomair
Danyh Ghazi Alotaibi

Supervised by:
Dr. Luluah Alhusain

INTRODUCTION:

Riyadh is one of the fastest-growing cities in the world, facing heavy traffic congestion and frequent accidents that lead to significant economic losses and delays. The process of manual vehicle damage assessment requires expert inspectors, making it slow, costly, and prone to human error.

Our project aims to develop an AI-based computer vision model that automatically classifies vehicle damage from images.

- The input to the model is an RGB image of a damaged car.
- output is the predicted damage category, such as door scratch, bumper dent, or glass shatter.

This task is highly significant as it accelerates the assessment process, reduces operational costs, and enhances accuracy for insurance and repair services. Moreover, it supports Saudi Vision 2030 by fostering digital transformation and intelligent transportation systems.

RELATED WORK:

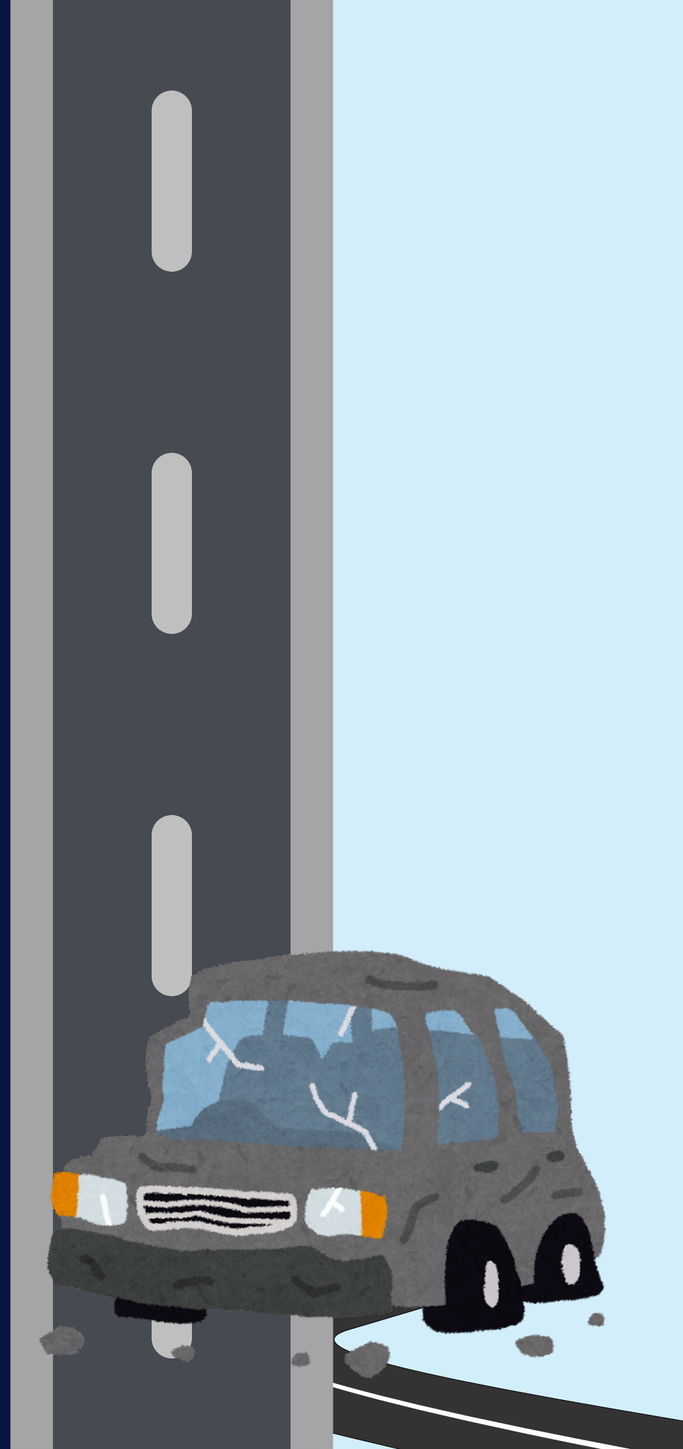
Study	Dataset	Task	Methods	Key Results
Sharma et al. (2022)	Custom shared mobility dataset (12 classes)	Damage detection & classification for shared mobility	Faster R-CNN, RetinaNet, EfficientDet + Transfer Learning	Fine-tuning outperformed feature extraction by 8-10%
Parslov et al. (2024)	CrashCar101 synthetic (~100K images, 15 classes)	Damage detection with synthetic-to-real transfer	ResNet-50, Mask R-CNN + Procedural generation	Synthetic pre-training improved accuracy by 12-15%
Kumar & Khan (2023)	Review: 74.5% private, 25.5% public (CarDD, Kaggle)	Systematic review of damage detection methods	CNN, YOLO, Mask R-CNN analysis	80%+ accuracy, challenges with minor damage
Hasan et al. (2025)	Review of 55 papers (CarDD, Kaggle datasets)	Systematic review of AI-based vehicle damage detection (2018–2024)	SLR methodology: database search, filtering 178 papers, selecting 55 for analysis; comparison of models, datasets & evaluation techniques	85-95% accuracy, need for diverse datasets

DATASET:

The dataset was obtained from Kaggle “ Car Damage Classification Dataset”.

The images are divided into 8 classes: door_scratch,bumper_scratch,door_dent,bumper_dent, glass_shatter, head_lamp, tail_lamp, and unknown.

IT contains 1,594 RGB images of vehicles showing different types of external damage.



DATASET:

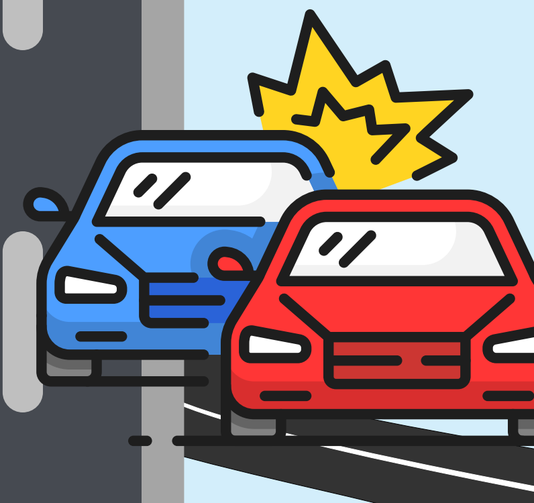
Data Preparation Steps:

- Verified all images and removed missing or corrupted files.
- Removed the “Unknown” class, which contained 549 images with unclear or unidentifiable damage types.
- After removal, the dataset contained 1,045 labeled images across the remaining classes
- Split dataset into Train (70%) ”731 samples” , Validation (15%) “157 samples” , and Test (15%) ” 157 samples” .
- Resized all images to 224×224 pixels for model consistency.
- Normalized pixel values to the range [0, 1].
- Used Data Augmentation: rotation (±20°), shifting (±10%), zoom (20%), horizontal flip, and brightness (0.8–1.2).

door_dent

door_scratch

glass_shatter



OVERVIEW OF MACHINE LEARNING METHODS

1. CUSTOM CNN

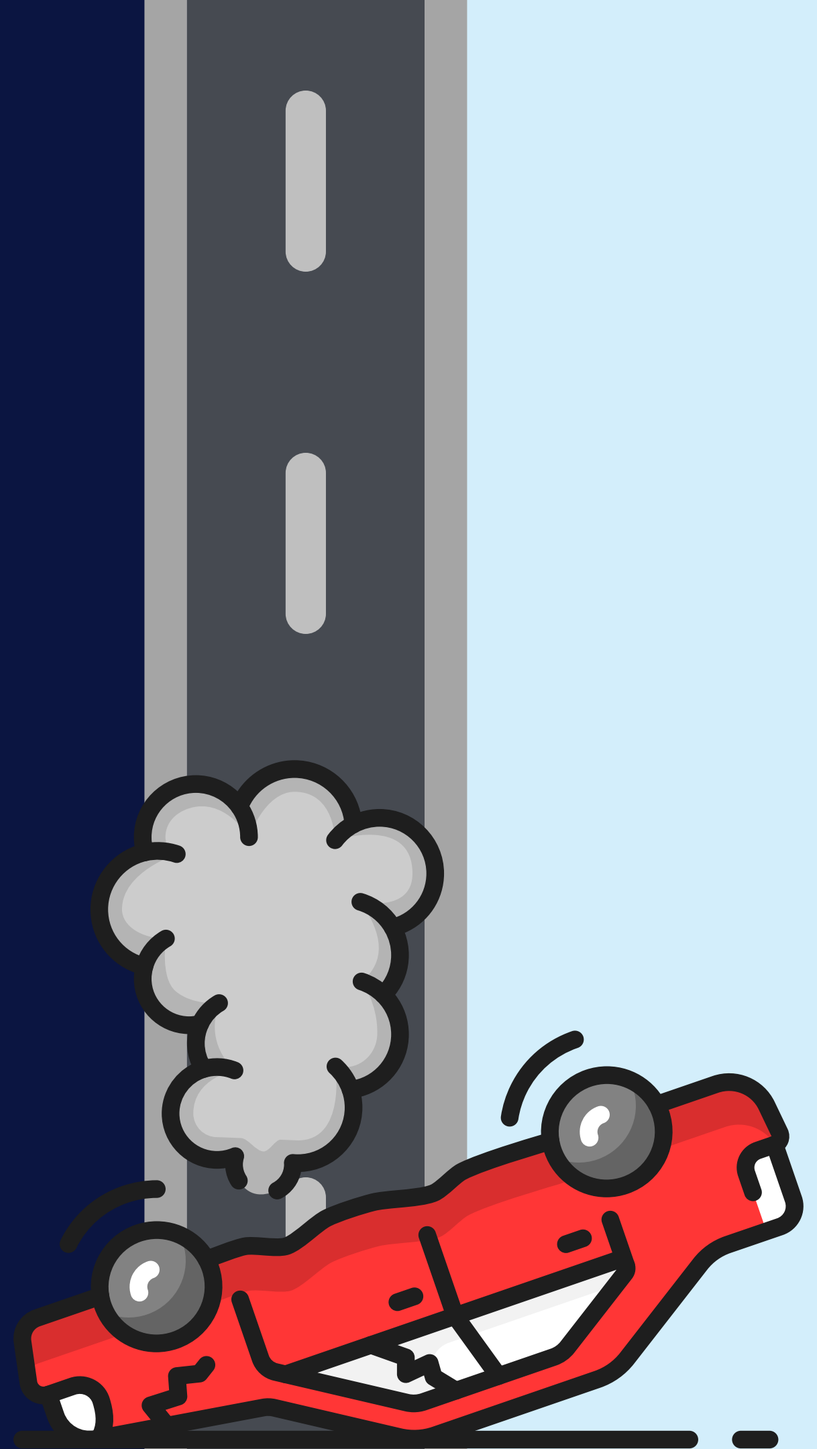
- Convolutional Neural Network built from scratch
- Baseline approach for comparison

2. EFFICIENTNETB0 + CLASSICAL ML CLASSIFIERS

- Use pre-trained EfficientNetB0 to extract 1280-D feature embeddings
- Train multiple classifiers: Logistic Regression, Random Forest, SVM, and KNN

3. TRANSFER LEARNING - DENSENET201

- Pre-trained on ImageNet with 20M parameters
- Fine-tuned using two-stage strategy: freeze base layers then gradually unfreeze



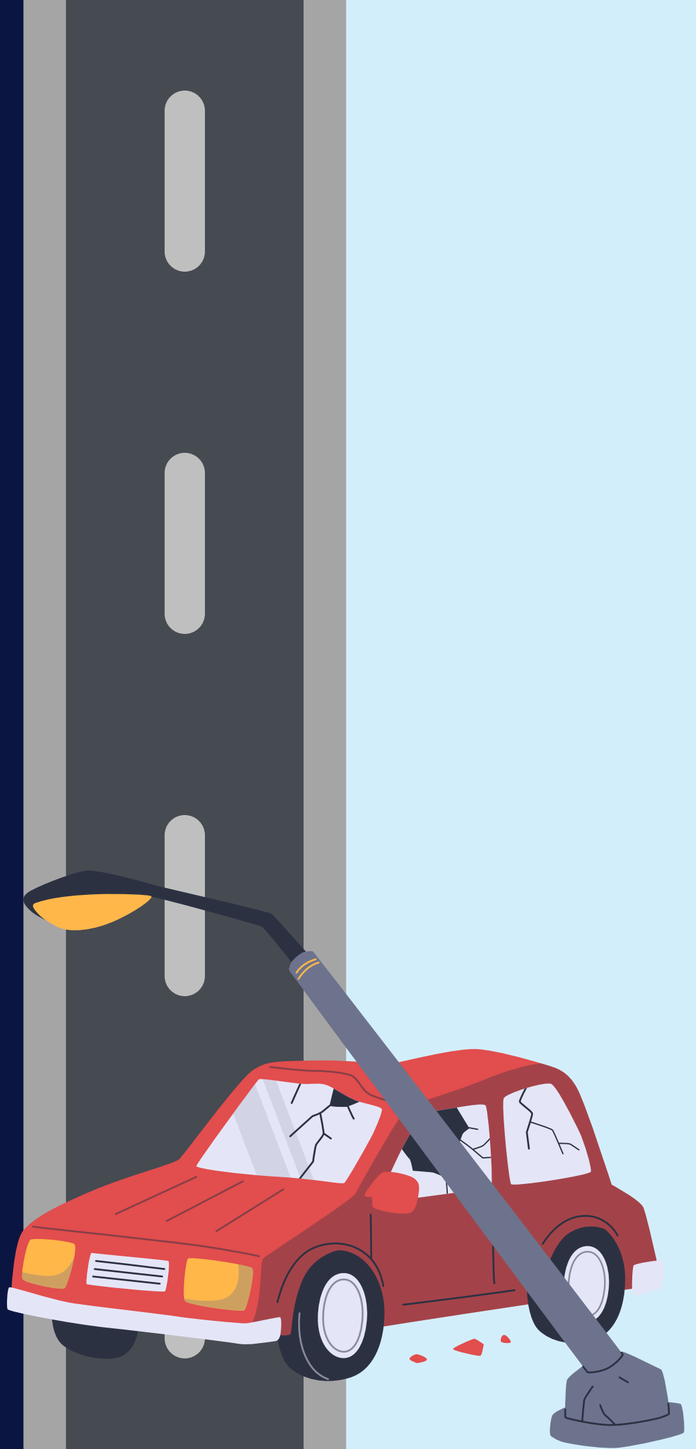
EXPERIMENTS - CNN

Hyperparameter Tuning:

- Strategy: Keras Tuner (Hyperband), 30 trials over ~8.5 hours
- Search Space: filters, dropout (0.2–0.5), L2 (1e-5–1e-3), LR (1e-5–1e-3), optimizers (Adam/RMSprop)
- Best Config: Filters 32→64→128→512, Dropout 0.20/0.30/0.30/0.30, L2=0.000290, Optimizer=RMSprop, LR=1.1e-5
- Result: 20.38% validation accuracy (baseline before full training)

Training Process Summary

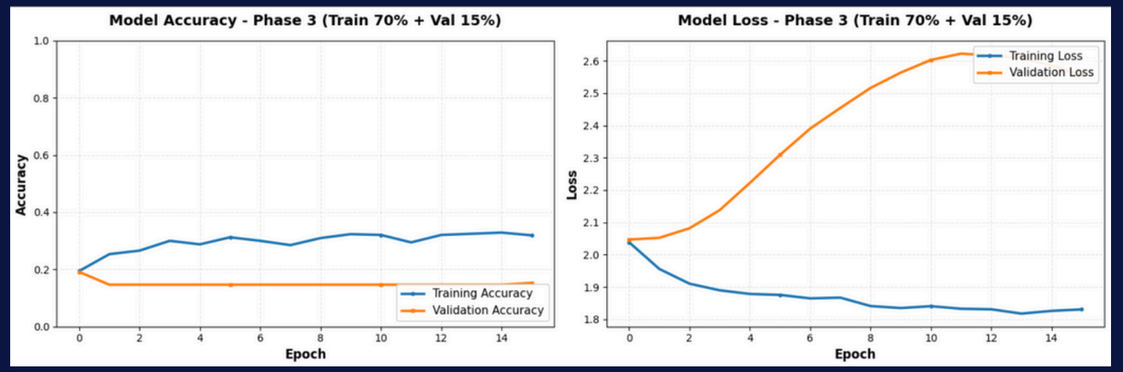
- Phase 1 (70% Train + 15% Val):
  - 730 augmented training samples, 157 validation samples
  - Batch size 32, up to 100 epochs with early stopping
  - Callbacks: ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
- Phase 2 (85% Combined Data):
  - 887 total images, lighter augmentation
  - Learning rate reduced to 5.73×10^{-6}
  - Up to 30 epochs, stopped at epoch 26 due to plateau



PRELIMINARY RESULTS - CNN

Model performance Analysis:

- Training accuracy stuck near 30%, validation stayed low at ~20%
- Training loss dropped slightly, but validation loss rose steadily
- Clear overfitting and failure to generaliz

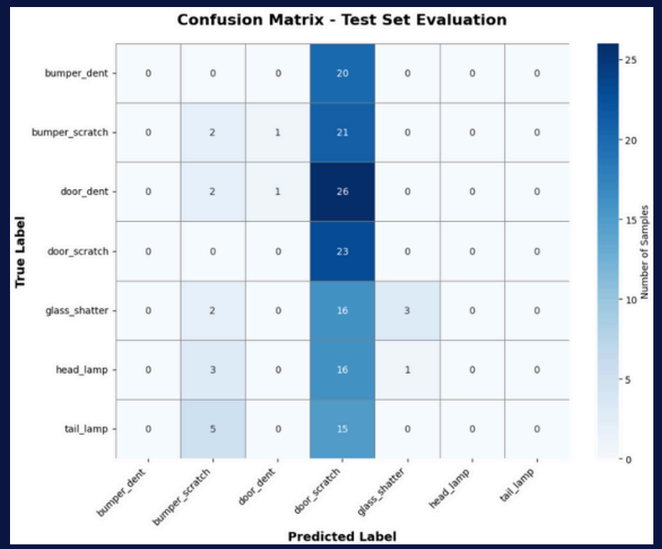


Test Accuracy:

- Test Accuracy: 18.47%
- Test Loss: 2.1861

Confusion Matrix:

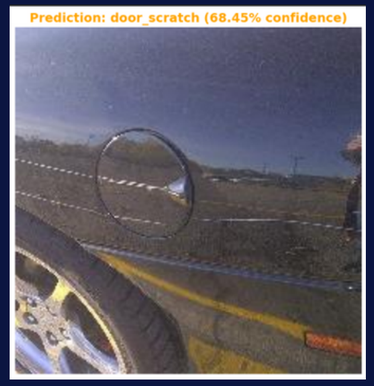
- Most predictions defaulted to door_scratch
- 3 classes had 0 correct predictions
- Model failed to separate visually similar classes



TESTING ON UNKNOWN IMAGES - CNN

Predict unknown images:

- Randomly selected 5 unseen images from the class unknown dataset.
- Preprocessed each image (resize to 224×224, normalization)
- Predicted the damage category using the trained model.



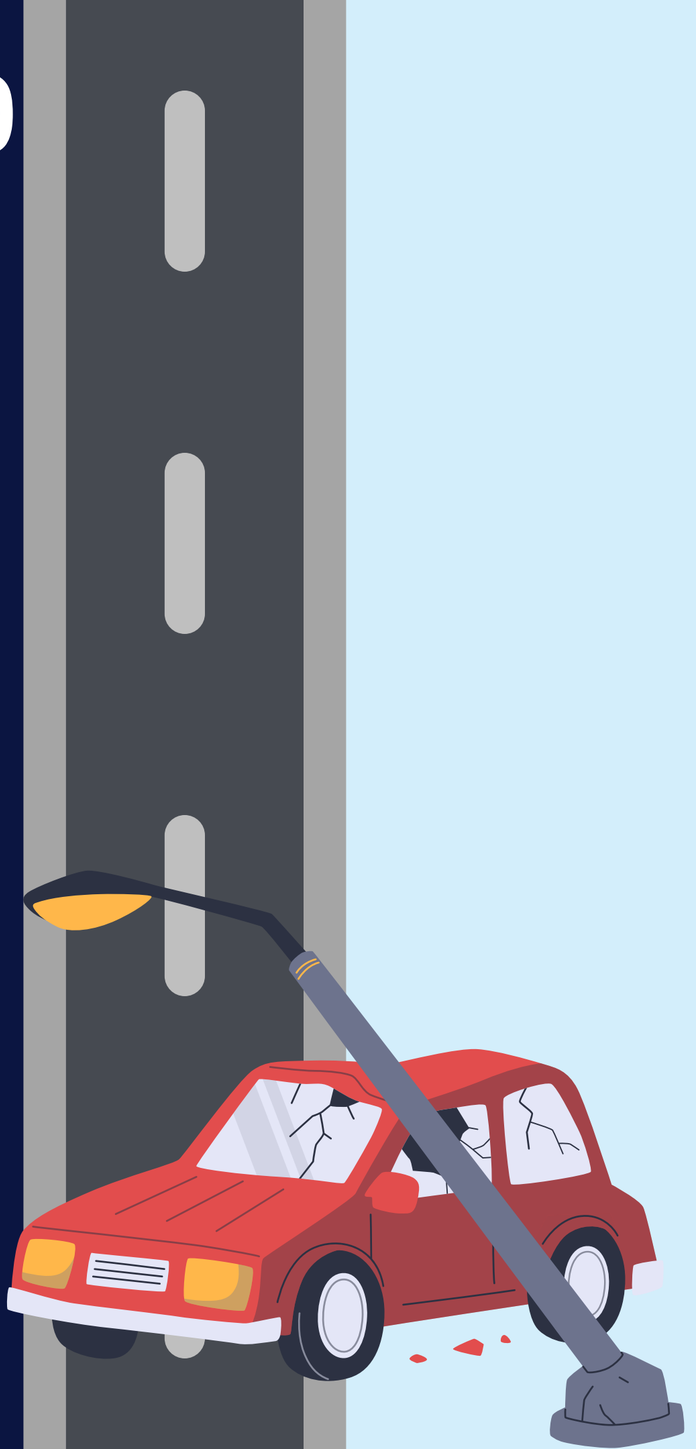
EXPERIMENTS - EFFICIENTNETB0

Hyperparameter Tuning:

- **Strategy: Manual search (no automated tuner)**
- **Search Space:**
- **Learning rate: 0.001 → 0.0001**
- **Batch size: 16, 32, 64**
- **Dropout: 0.2–0.4**
- **Selected Configuration: LR=1e-4, Batch=32, Dropout=0.3, Optimizer=Adam, Loss=categorical crossentropy**
- **Backbone Status: EfficientNetB0 fully frozen during feature extraction**

Training Process Summary

- Mode: Single-stage feature extraction training (train head only)
- Duration: 20 epochs with EarlyStopping (patience=5)
- Callbacks Used: ModelCheckpoint, EarlyStopping, ReduceLROnPlateau
- Data Augmentation: Heavy augmentation applied to improve generalization
- Goal: Train classification head on frozen EfficientNetB0 features for stable performance



PRELIMINARY RESULTS - EFFICIENTNETB0

Model performance Analysis:

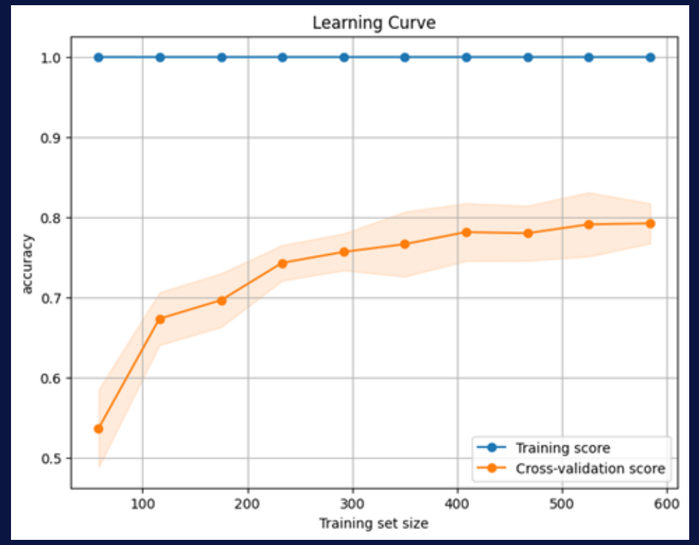
- Models reached very high training accuracy (~100%) but much lower validation/test accuracy → clear overfitting.
- Logistic Regression performed best (85%), followed by SVM and Random Forest (82–80%), while KNN was weakest (68%).

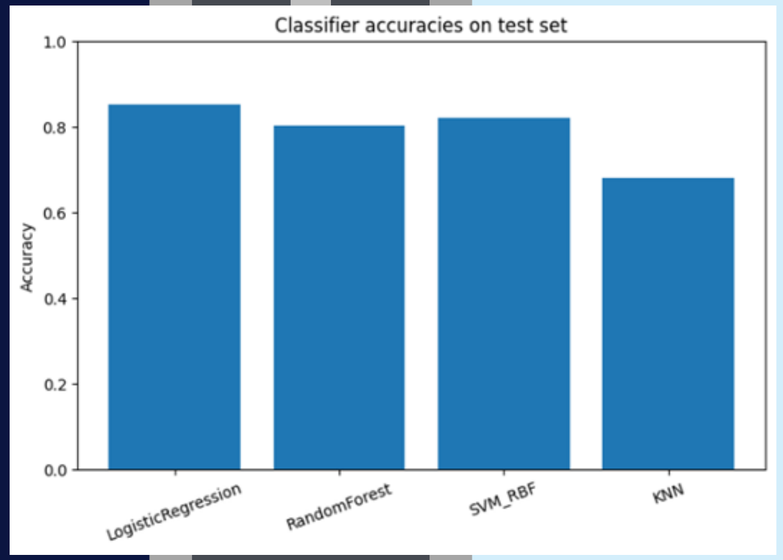
Test Accuracy:

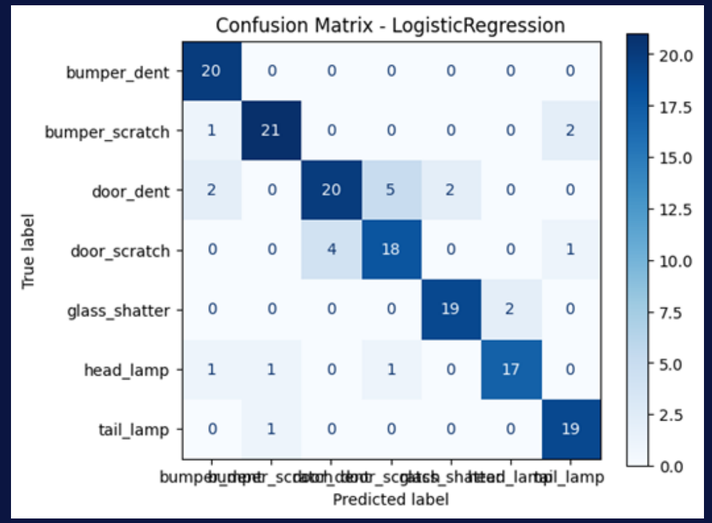
- Logistic Regression: 85.35% (best)

Confusion Matrix:

- Most classes were classified correctly, especially glass_shatter and tail_lamp.
- Largest confusion occurred between door_dent and door_scratch due to strong visual similarity.



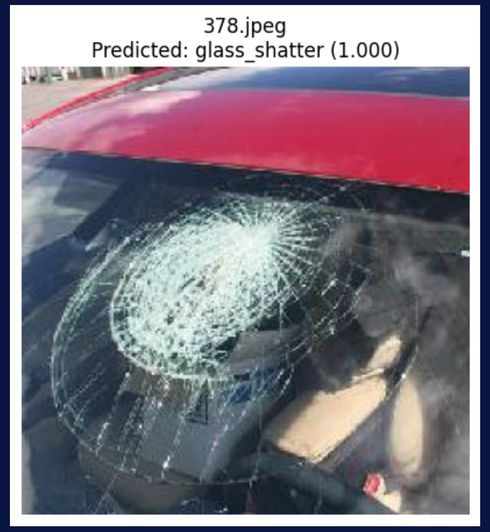


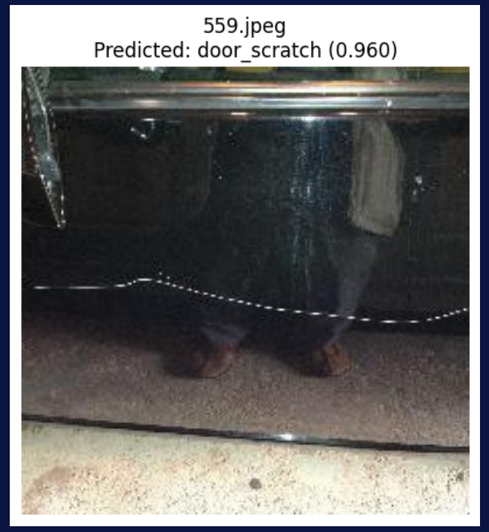


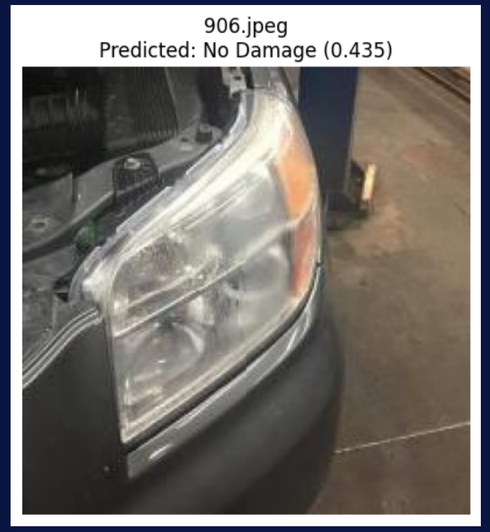
TESTING ON UNKNOWN IMAGES - EFFICIENTNETB0

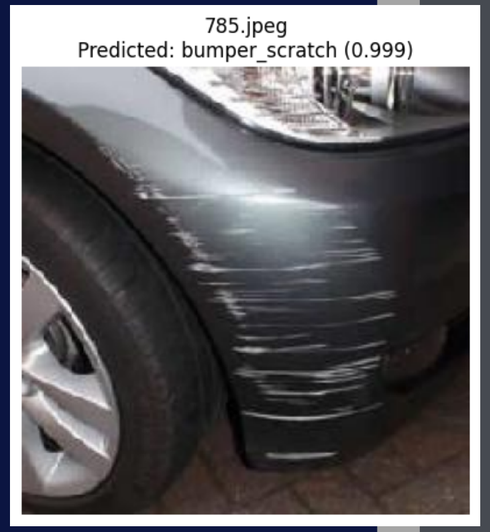
Predict unknown images:

- Randomly selected 5 unseen images from the class unknown dataset.
- Preprocessed each image (resize to 224×224, normalization)
- Applied a confidence threshold (60%) – if prediction confidence < 0.6 → labeled as “No Damage”
- Predicted the damage category using the trained model.









EXPERIMENTS - DENSENET201

Hyperparameter Tuning:

- Random Search strategy with 3 trials.
- Tested combinations:
- Learning rate: 1e-3, 1e-4, 1e-5
- Dropout rate: 0.3, 0.4, 0.5
- Frozen layers: 40, 50, 60
- Short training runs performed to select best hyperparameters.

Selected configuration: LR=0.00001, Dropout=0.4, 50 frozen layers, Batch=32

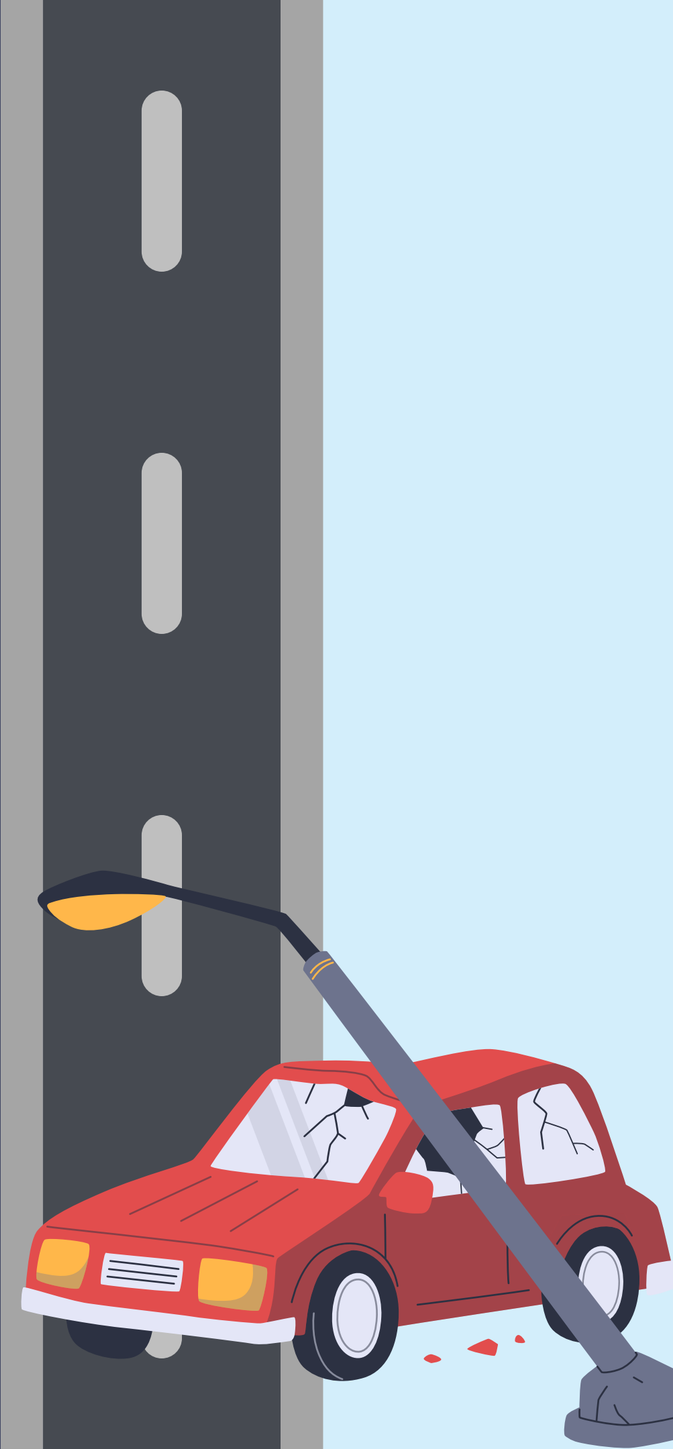
Two-Stage Training Strategy

Stage 1: Classification Head Training (10 epochs)

- Freeze all 201 DenseNet layers
- Train only custom classification head

Stage 2: Fine-Tuning (25 epochs)

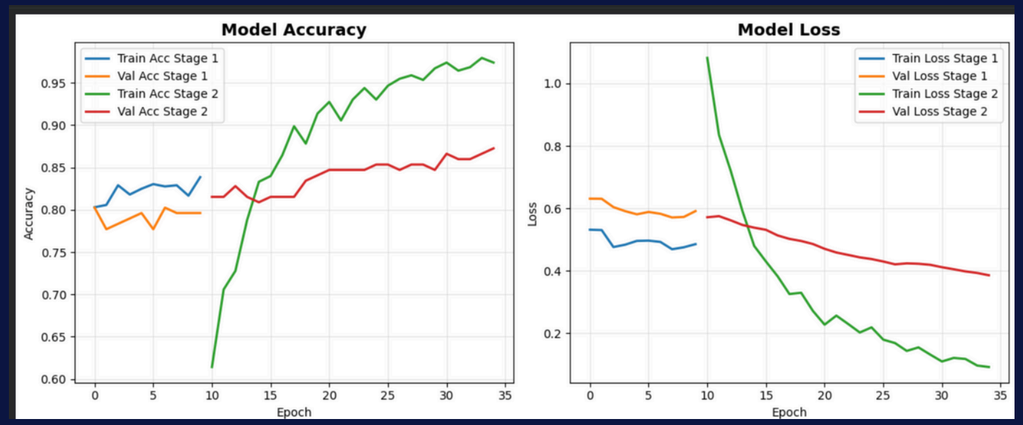
- Unfreeze last 151 layers, keep first 50 frozen
- Early stopping: patience=5



PRELIMINARY RESULTS - DENSENET201

Model performance Analysis:

- Stage 1 (0-10): Baseline ~80%
- Stage 2 (10-35): Improved to ~88%

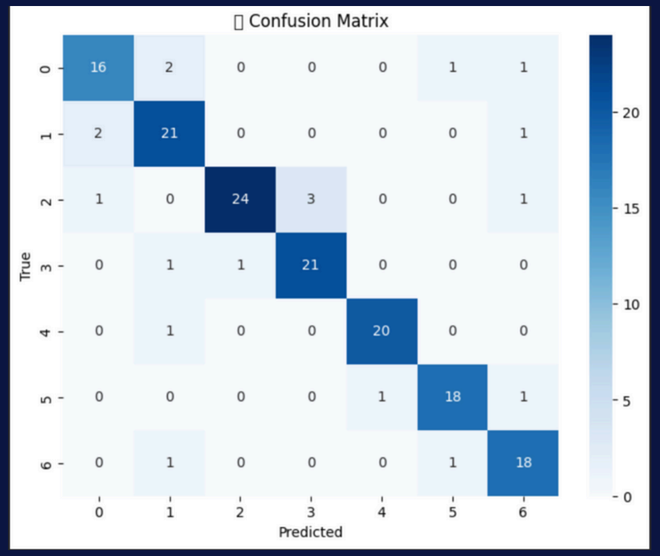


Test Accuracy:

```
Test Accuracy: 87.90%
Test Loss: 0.3319
```

Confusion Matrix:

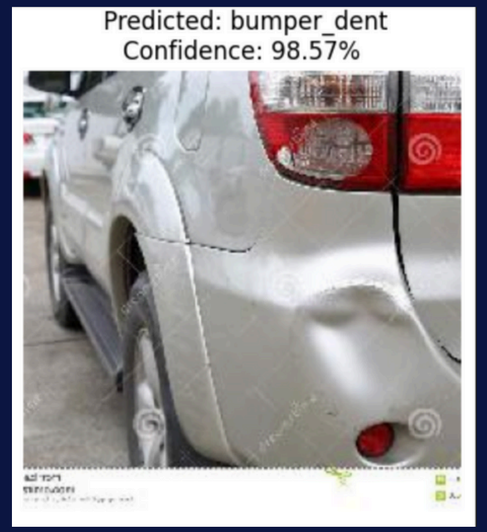
- Strong diagonal (high true positives)
- Main confusions: door_scratch ↔ bumper_scratch, door_dent ↔ bumper_dent

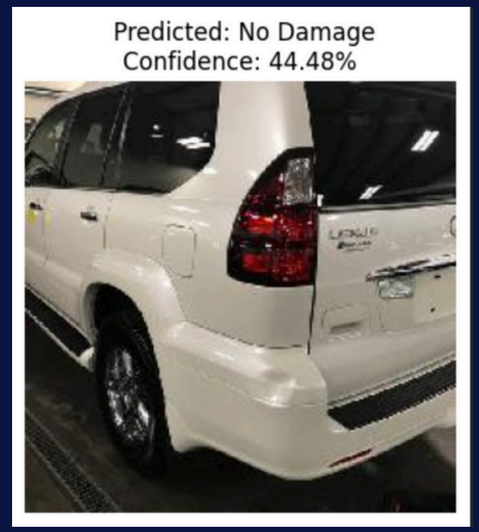


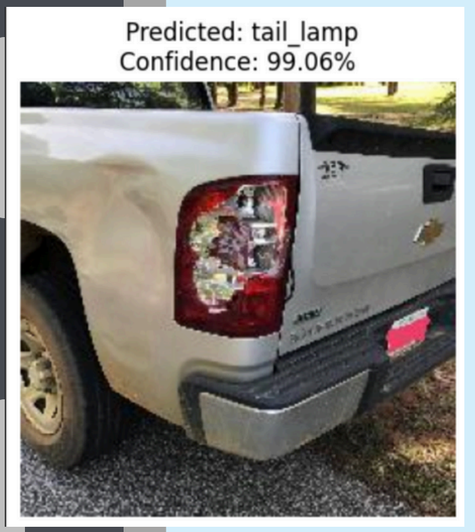
TESTING ON UNKNOWN IMAGES - DENSENET201

Predict unknown images:

- Randomly selected 5 unseen images from the class unknown dataset.
- Preprocessed each image (resize to 224×224, normalization)
- Applied a confidence threshold (60%) – if prediction confidence < 0.6 → labeled as “No Damage”
- Predicted the damage category using the trained model.

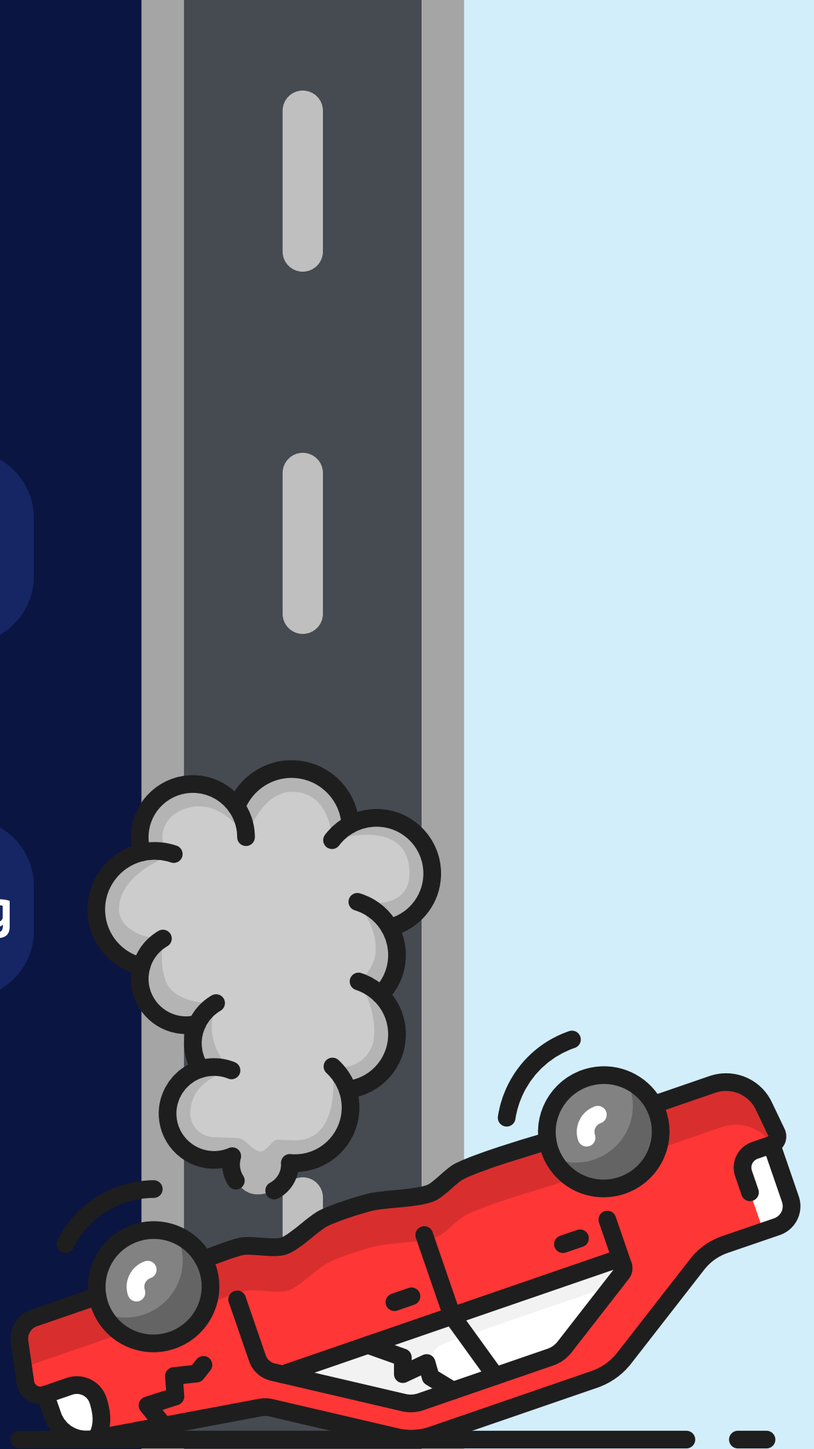






Main challenges

- Small dataset
- Variable image quality
- Imbalanced classes
- Time constraints during training

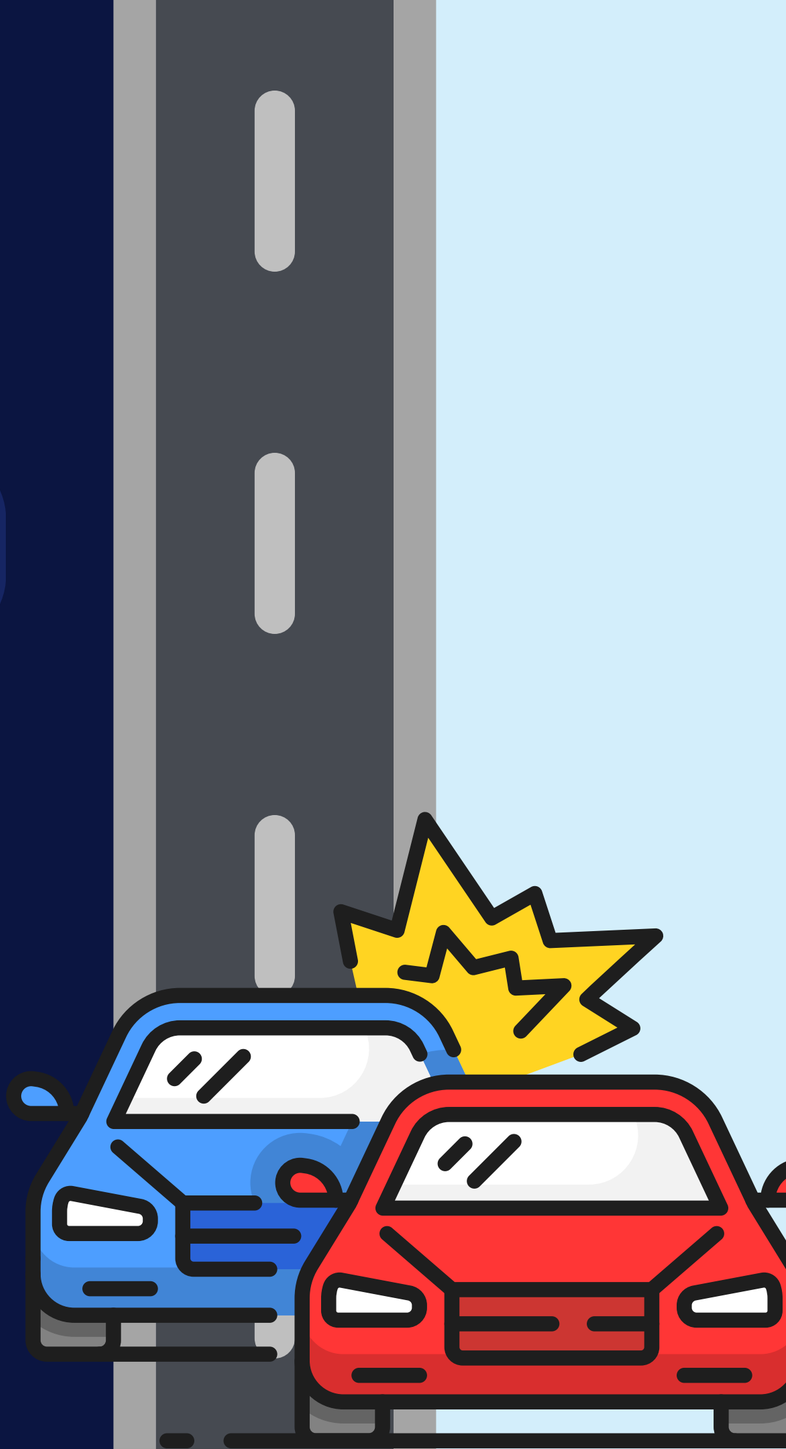


Future Work

Collect more diverse and higher-quality real-world data

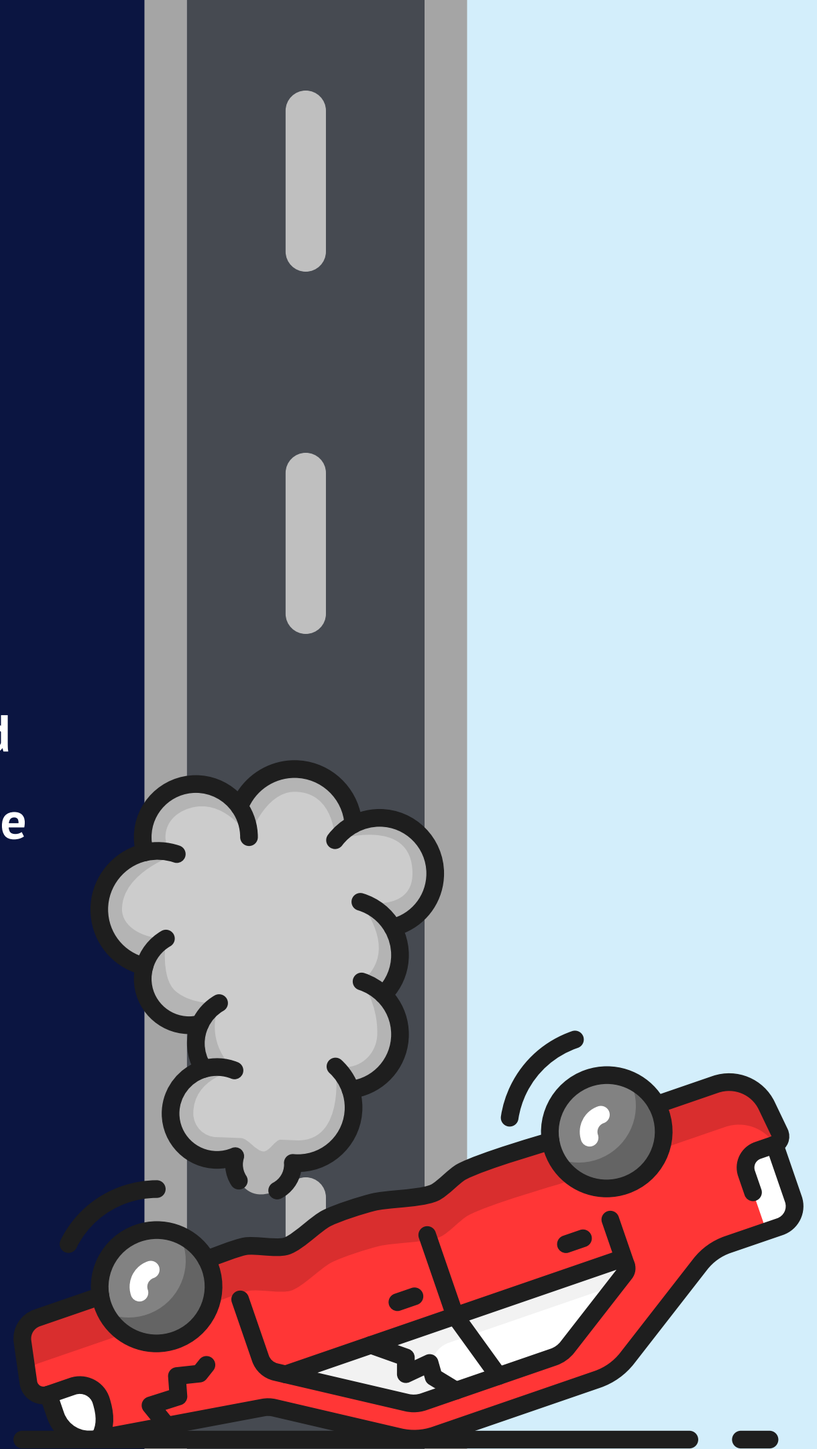
Stronger models

Mobile deployment testing

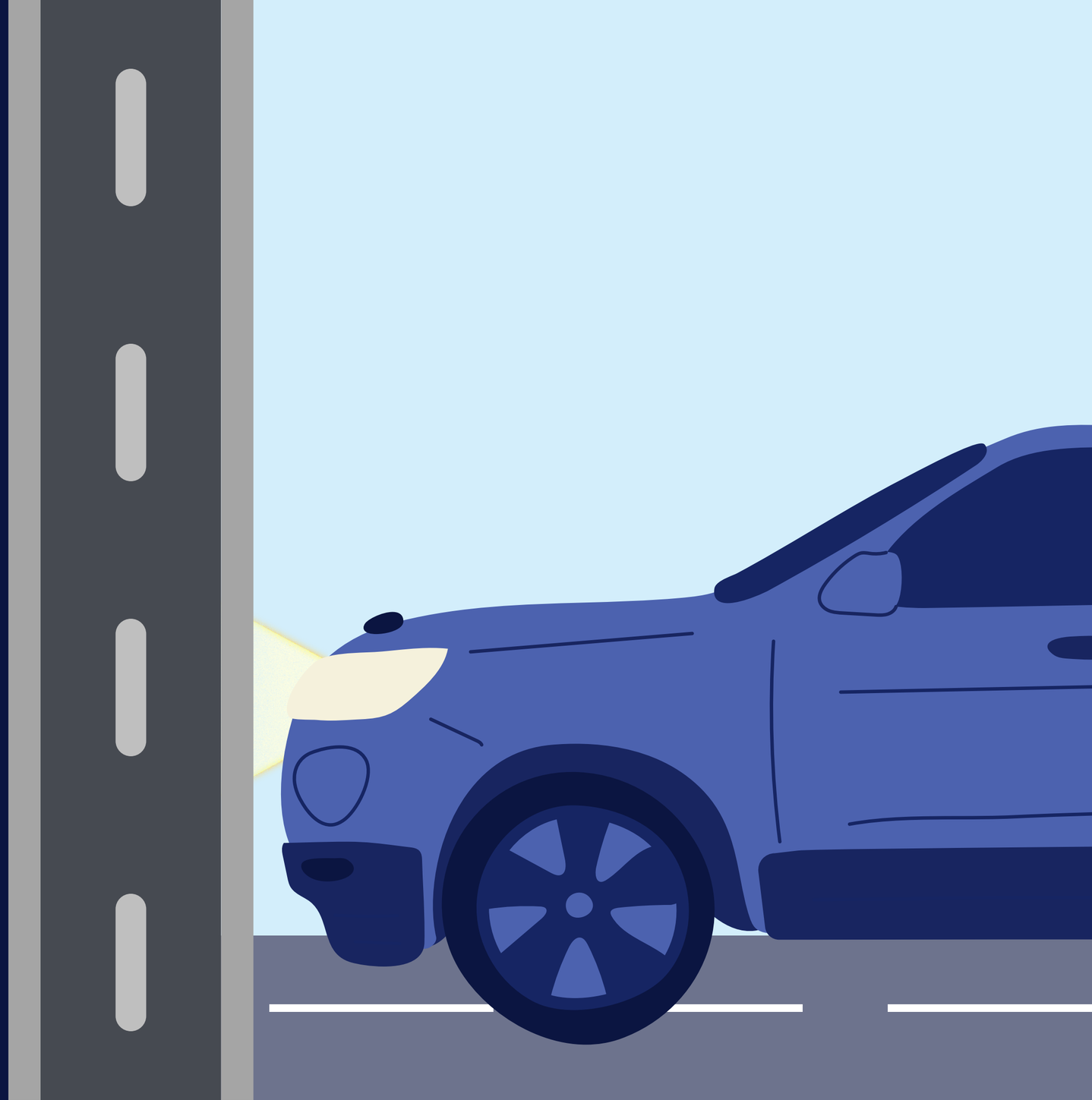


CONCLUSION

This project showed that transfer learning especially DenseNet201 provides an effective foundation for automated vehicle damage assessment, supporting Saudi Arabia’s digital transformation efforts and enabling smarter, more efficient services in the automotive and insurance .sectors



LIVE TESTINNG



THANK YOU !

Any Questions?