AIML Capstone Project: Car Detection and Classification - Interim Report

Jannet Akanksha Ekka, Adari Pooja

1. Summary of problem statement, data and findings

Domain: Automotive Surveillance
Problem Statement: Design a deep learning-based car identification model that can detect and classify cars based on their make, model, and year.

1.1 Dataset Overview:

- Cars dataset containing 16,185 images across 196 classes
- Split into 8,144 training images and 8,041 testing images
- Classes defined at Make, Model, Year level (e.g., "Volkswagen Golf Hatchback 2012")
- Each image includes annotations with bounding box coordinates and corresponding class label
- ~50-50 split per class between training and testing sets

1.2 Initial Data Findings:

- Successfully loaded and processed all images and annotations
- Verified data integrity and bounding box accuracy
- Images vary in size but maintain good quality for classification
- Balanced class distribution with 24-68 images per class
- Bounding box annotations are accurate and well-fitted to the cars

Sample Image
Class: Volkswagen Golf Hatchback 2012

2. Summary of the Approach to EDA and Pre-processing

2.1 Data Organization:

- Created structured mappings between:
  - Images and their respective classes
  - Images and their bounding box annotations
  - Class names and numerical indices

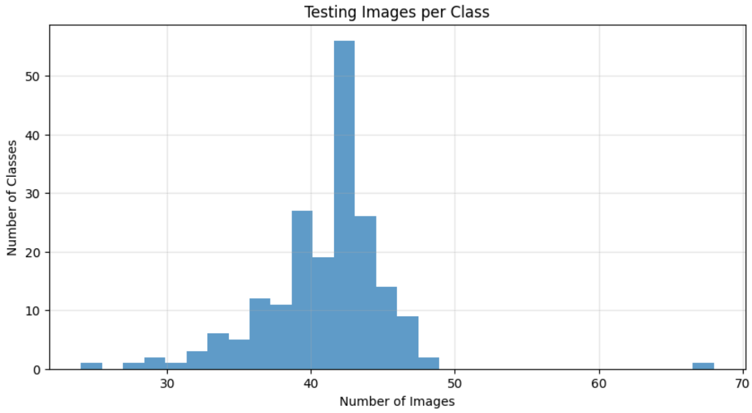
2.2 Data Distribution Analysis

Class Distribution Analysis for:

Train:

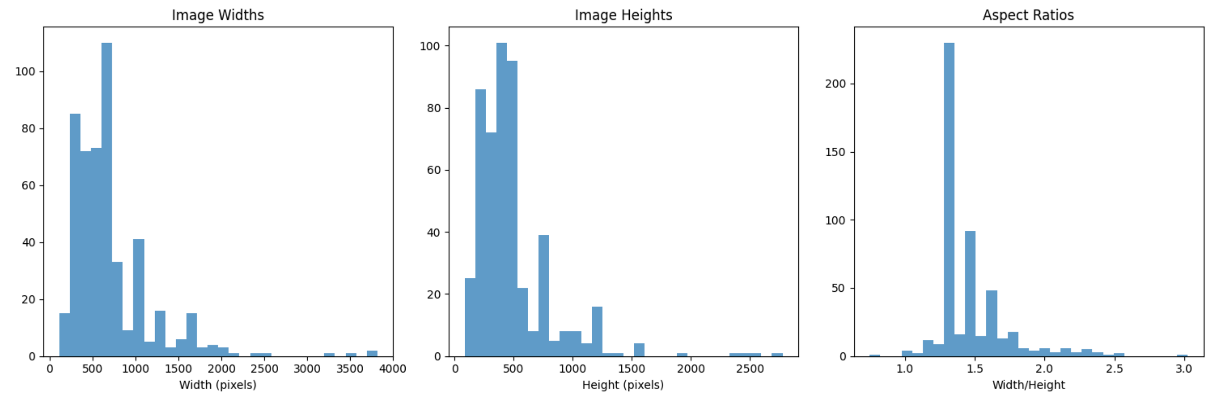
- Min images per class: 24
- Max images per class: 68
- Mean images per class: 41.6

Test:



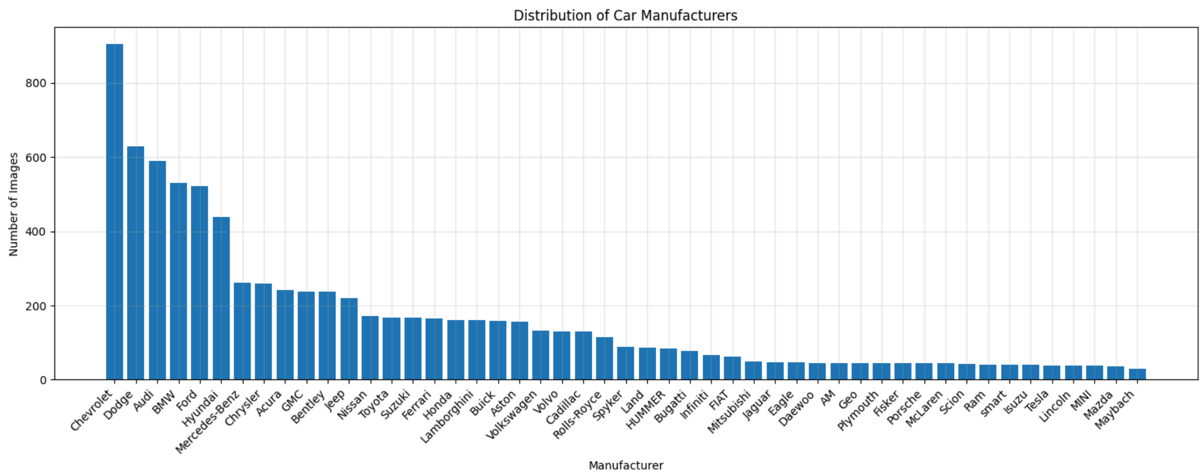
- Min images per class: 24
- Max images per class: 68
- Mean images per class: 41.0

2.3 Image Dimension Analysis using 500 random images:

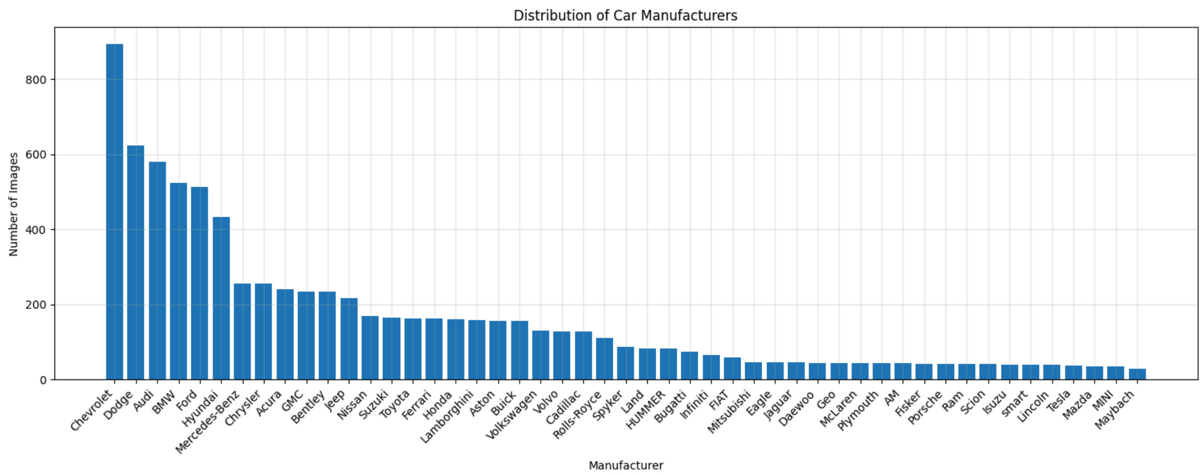


- Average dimensions: 698.6x482.0 pixels
- Average aspect ratio: 1.48

2.4 Manufacturer Analysis



Top 10 manufacturers by number of train images:
Chevrolet: 905 images
Dodge: 630 images
Audi: 589 images
BMW: 531 images
Ford: 521 images
Hyundai: 438 images
Mercedes-Benz: 261 images
Chrysler: 260 images
Acura: 242 images
GMC: 238 images

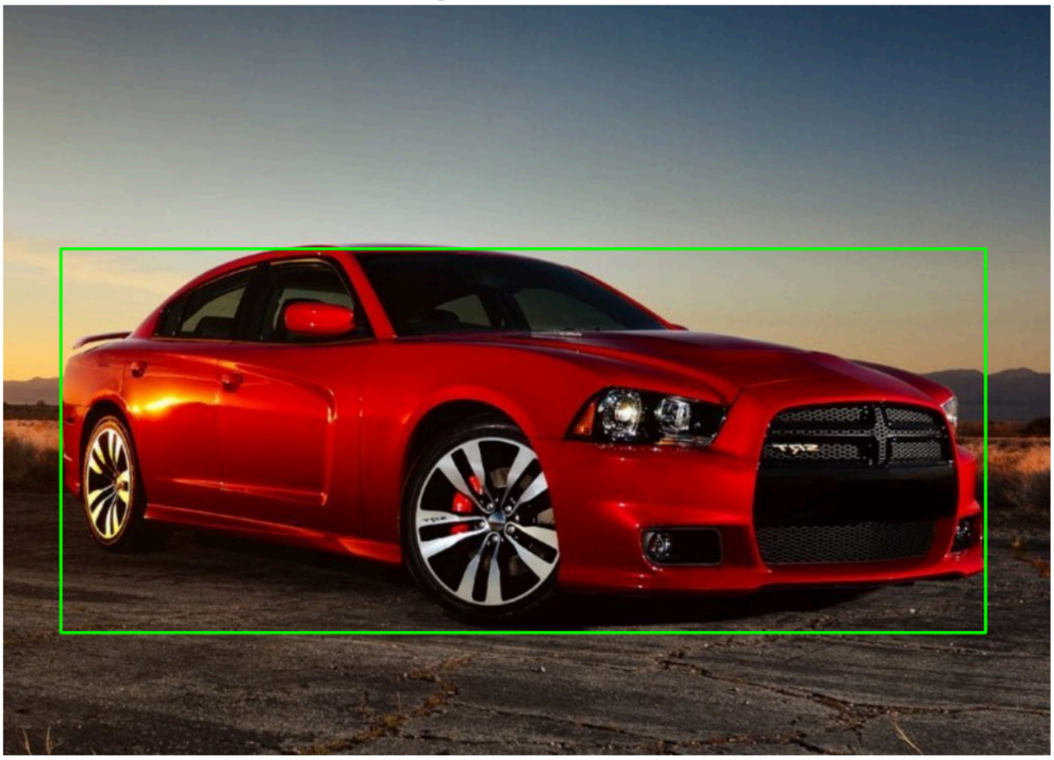


Top 10 manufacturers by number of test images:
Chevrolet: 894 images
Dodge: 623 images
Audi: 580 images
BMW: 524 images
Ford: 514 images
Hyundai: 433 images
Mercedes-Benz: 257 images
Chrysler: 256 images
Acura: 240 images
GMC: 235 images

2.5 Pre-processing Steps:

- Image Standardization:
  - Re-sized all images to 224x224 pixels
  - Normalized pixel values to range [0,1]
  - Maintained aspect ratio during resizing
- Data Organization:
  - Implemented chunk-based processing to handle memory constraints
  - Created TensorFlow datasets for efficient training
  - Batch size: 32 images
  - Added shuffling for training data
  - Created mappings of the images with classes
- Annotation Processing:
  - Verified bounding box coordinates
  - Ensured all coordinates are within image dimensions
  - Created mappings for easy access during training
- Visualizations:
  - Displayed sample images with bounding boxes
  - Verified correct class labeling
  - Confirmed bounding box accuracy
  - Checked image quality and resolution distribution

Class: Dodge Charger Sedan 2012
Bounding Box: [56, 237, 960, 611]



Class: Chevrolet Sonic Sedan 2012
Bounding Box: [30, 116, 963, 523]

3. Deciding Models and Model Building

3.1 Approach to Model Development

For our car classification task, we adopted a two-pronged strategy to establish a comprehensive understanding of model performance:

1. Basic CNN Implementation
   - Develop a CNN model from scratch to understand fundamental deep learning concepts
   - Establish baseline performance metrics for comparison
   - Gain insights into dataset challenges and model behavior
   - Identify limitations when learning features without pre-training
2. Transfer Learning Implementation
   - Leverage pre-trained models that have proven success on ImageNet
   - Utilize established architectures of varying complexities
   - Take advantage of features learned from millions of images
   - Address limited dataset challenges through knowledge transfer

3. Resource-Constrained Implementation:

   - Working with data chunks due to computational limitations
   - Used 16 classes with ~29 samples per class
   - Total working dataset: 650 training and 642 testing samples
   - Classes selected based on sample availability and balance.

This dual approach enables us to:

- Compare effectiveness of learned vs pre-trained features
- Understand the impact of the model complexity on our specific dataset
- Evaluate the resources' requirements and their performance trade-offs
- Follow the best practices for car classification tasks within computational constraints

3.2 Planned Model Architectures

1. Basic CNN
2. ResNet50
3. VGG16
4. MobileNetV2

Basic CNN (Baseline)
- **Architecture Overview:** 4 conv blocks with batch normalization and dropout
- **Performance Characteristics**: Focuses on fundamental CNN capabilities
- **Resource Usage:** Lightweight computing requirements
- **Key Strengths/Weaknesses**: Simple architecture, baseline performance benchmark

ResNet50
- **Architecture Overview:** Deep residual learning with skip connections
- **Performance Characteristics**: Deep feature extraction capability
- **Resource Usage**: Moderate GPU requirements
- **Key Strengths/Weaknesses**: Proven architecture on ImageNet, transfer learning potential

VGG16
- **Architecture Overview:** Deep residual learning with skip connections
- **Performance Characteristics**: Deep feature extraction capability
- **Resource Usage**: Moderate GPU requirements
- **Key Strengths/Weaknesses**: Proven architecture on ImageNet, transfer learning potential

MobileNetV2
- **Architecture Overview:** Deep residual learning with skip connections
- **Performance Characteristics**: Deep feature extraction capability
- **Resource Usage**: Moderate GPU requirements
- **Key Strengths/Weaknesses**: Proven architecture on ImageNet, transfer learning potential

3.3 Current Results

Model Version	Parameters	Training Time (min)	Best Train Acc (%)	Best Val Acc (%)	Epochs to Converge
Basic CNN	532,176	~25	26.50	7.59	14
VGG16	14,987,600	~60	98.74	63.62	20
ResNet50	24,647,056	~45	99.09	69.20	20
MobileNetV2	2,924,112	~27	99.82	65.17	20

3.4 Class Accuracies

Model	Class Accuracy
Basic CNN	1: 0.00% (0/44 samples) 2: 0.00% (0/32 samples) 3: 0.00% (0/43 samples) 4: 0.00% (0/42 samples) 5: 0.00% (0/40 samples) 6: 0.00% (0/44 samples) 7: 0.00% (0/39 samples) 8: 0.00% (0/45 samples) 9: 17.07% (7/41 samples) 10: 0.00% (0/33 samples) 11: 0.00% (0/38 samples) 12: 0.00% (0/36 samples) 13: 0.00% (0/41 samples) 14: 100.00% (42/42 samples) 15: 0.00% (0/43 samples) 16: 0.00% (0/43 samples)
VGG16	1: 100.00% (44/44 samples) 2: 40.62% (13/32 samples) 3: 60.47% (26/43 samples) 4: 52.38% (22/42 samples) 5: 57.50% (23/40 samples) 6: 70.45% (31/44 samples) 7: 76.92% (30/39 samples) 8: 42.22% (19/45 samples) 9: 46.34% (19/41 samples) 10: 39.39% (13/33 samples) 11: 81.58% (31/38 samples) 12: 75.00% (27/36 samples) 13: 75.61% (31/41 samples) 14: 40.48% (17/42 samples) 15: 69.77% (30/43 samples) 16: 81.40% (35/43 samples)

ResNet50	1: 100.00% (44/44 samples) 2: 43.75% (14/32 samples) 3: 55.81% (24/43 samples) 4: 59.52% (25/42 samples) 5: 55.00% (22/40 samples) 6: 79.55% (35/44 samples) 7: 69.23% (27/39 samples) 8: 71.11% (32/45 samples) 9: 58.54% (24/41 samples) 10: 51.52% (17/33 samples) 11: 86.84% (33/38 samples) 12: 66.67% (24/36 samples) 13: 68.29% (28/41 samples) 14: 59.52% (25/42 samples) 15: 76.74% (33/43 samples) 16: 93.02% (40/43 samples)
MobileNetV2	1: 100.00% (44/44 samples) 2: 43.75% (14/32 samples) 3: 62.79% (27/43 samples) 4: 59.52% (25/42 samples) 5: 52.50% (21/40 samples) 6: 65.91% (29/44 samples) 7: 64.10% (25/39 samples) 8: 68.89% (31/45 samples) 9: 41.46% (17/41 samples) 10: 33.33% (11/33 samples) 11: 84.21% (32/38 samples) 12: 75.00% (27/36 samples) 13: 75.61% (31/41 samples) 14: 42.86% (18/42 samples) 15: 74.42% (32/43 samples) 16: 86.05% (37/43 samples)

3.5 Insights

- Class level accuracies
- Per-model analysis with respect to the layers, parameters, trainable parameters etc
- Comparative strengths/weaknesses
- Key challenges identified

3.6 Sample Predictions of Basic CNN

- Training Dynamics:
  - Early stopping at epoch 6
  - Learning rate reduction at epoch 4
  - Training accuracy: 22.50%
  - Validation accuracy: 6.25%
  - Test accuracy: 7.17%

- Class-wise Performance Analysis:
    - Strong Performers:
        - Class_176: 65.79% (25/38 samples)
        - Class_158: 44.83% (13/29 samples)
        - Class_175: 20.00% (6/30 samples)
    - Moderate Performers:
        - Class_182: 4.35% (2/46 samples)
    - Poor Performers:
        - 12 classes showing 0% accuracy (details in sample predictions)
- Sample Predictions:
    - Visual Examples:
        - Image 1: Predicted Class_158 vs True: Class_186
        - Image 2: Predicted Class_176 vs True: Class_186
        - Image 3: Predicted Class_182 vs True: Class_186
        - Image 4: Predicted Class_158 vs True: Class_186
- Prediction Analysis:
    - Model shows bias towards certain classes (158, 176)
    - Consistent misclassification patterns observed
    - Poor generalization across most classes
    - Better performance on visually distinctive vehicles

Pred: Class_158
True: Class_186

Pred: Class_176
True: Class_186

Pred: Class_182
True: Class_186

Pred: Class_158
True: Class_186

3.7 Future work for Milestone 2

- Model-specific improvements for better accuracy
    - Overfitting issues
    - Entire dataset usage (currently only ~30 samples per class used)
    - Use transfer learning techniques
    - Data augmentation
    - Further data refinements if necessary like Random seeding etc
- Fine-tuning strategies
- RCNN implementations