TwinCar Project

Project Name: TwinCar
Date: 2025-03-11
Prepared for: Data Science Group 18
Prepared by: Sols & Vladimir Ilievski (Instructor)

1. Introduction & Project Overview

TwinCar is pioneering an automated vehicle inspection platform, leveraging drones and ground robots to capture vehicle images for damage assessment and identification. In the first stage, your team will focus on developing computer vision models to recognize vehicle brands and models from these captured images.

2. Business Context & Requirements

TwinCar's broader mission is to deploy an autonomous AI agent on drones or robots for real-time vehicle inspection. This agent will utilize various tools, with a crucial component being deep learning models for visual identification of vehicle make and model, and optionally the year of production. Given multiple images of the same vehicle, your model should be able to output:

- **Vehicle Make** (e.g., Toyota, Ford, BMW)
- **Vehicle Model** (e.g., Camry, F-150, X5)
- **Year of Production** (optional, e.g., "2012")

For example, “Audi S4 Sedan 2012”. Correct identification of make and model is mandatory; year is optional.

3. Technical Requirements

This task is a **supervised multi-class, multi-label classification** problem:

- **Input:** Single image of a vehicle
- **Output:** Predicted vehicle make, model, and optionally year

Key Expectation:
In addition to model development, the final deliverable should be a robust, production-ready coded solution, following industry standards for reusability and clarity.

---

4. Datasets

4.1. Training Data

Official training data is not provided; you must select and curate your own dataset. Potential public datasets include:

1. [Stanford Cars Dataset]
   - 16,185 images, 196 classes (Make, Model, Year)
2. [CompCars Dataset]
   - ~136,727 web-nature images, 1,600+ models; ~36,000 surveillance-nature images
   - Includes make, model, year, type, bounding box, viewpoint

Combining multiple datasets is encouraged for broader coverage. Data augmentation is recommended to ensure model robustness.

4.2. Inference Data (Production Inputs)

The real-world data presented to your model after deployment may differ from public datasets. Consider the following during preparation and augmentation:

- **Sources:** Images from prototype drones and ground robots, primarily in parking lots and vehicle inspection environments
- **Format:** JPG (preferred) or PNG; RGB (3 channels)
- **Resolution:** Varies, typically 1920x1080 up to 4K, must be preprocessed consistently
- **Image Composition:**
  - Clear foreground vehicle, background may include other cars or infrastructure
  - Occlusion: minimal (≤ 10%) to moderate (up to 40%)
  - Orientation: Includes drone (top-down, oblique), robot (front, rear, side) perspectives
  - Scale: Vehicle occupies 10–80% of frame
  - High image quality, minimal blur

- **Visual Conditions:** Natural daylight, clear/overcast, varied shadows, surface reflections as found in standard vehicle photography

---

5. Deliverables

- **GitHub Repository:**
  - Well-structured repo in Python, with modular, clearly commented code.
  - At least one Python script for batch prediction on image lists.
  - Complete README with setup, usage instructions, and evaluation results (metrics such as precision, recall, F1-score, loss curves, etc).
  - Include saved model files (e.g., `.h5`, `.pt`, `.onnx`).
  - Hugging Face model card and (optionally) hosting the model on Hugging Face Hub.
- **Final Presentation:**
  Slides summarizing your approach, experiments, and results.
- **Demo/Prototype (Optional):**
  Web-based model demo (e.g., Hugging Face Spaces).
- **Explainability (Optional):**
  Implement explainability techniques to interpret model predictions.

---

6. Technology & Resources

6.1. Preferred Technologies

- **Programming Language:** Python (recommended)
- **ML Frameworks/Libraries:**
  - Scikit-learn, PyTorch, TensorFlow, Keras, Hugging Face Transformers

6.2. Hardware Resources

- Coordinator Sols can provide limited GPU access and Amazon Cloud credits for cloud-based experiments and training.

---

7. Evaluation Criteria

Your solutions will be evaluated on:

- **Prediction Performance:** Accuracy on a private benchmarking set reflecting production data format
- **Code Quality:** Organization, readability, and reusability of code
- **Presentation:** Clarity and professionalism in presenting your results
- **Bonus:** Optional deliverables, such as a web demo or explainability analysis

---

> **For further information or dataset exploration, consult the original dataset links**:
> Stanford Cars Dataset (Hugging Face)
> CompCars Dataset (CUHK)

Good luck! We look forward to seeing your innovative approach to solving this real-world machine learning challenge.

Source:

- Stanford Cars Dataset – Hugging Face
- CompCars Dataset – CUHK
- Hugging Face Model Card Documentation