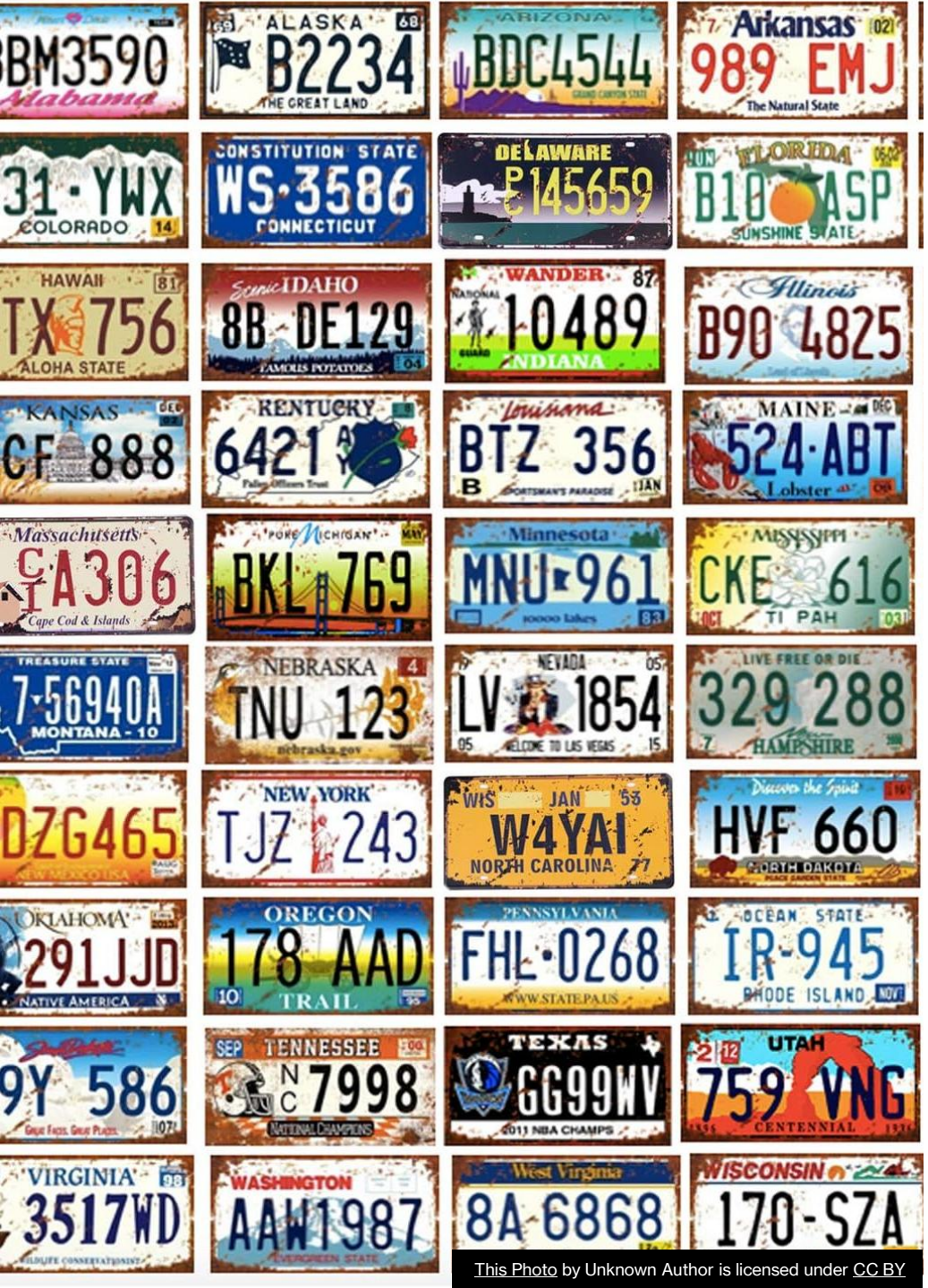


Project: License Plate Detector

Tier: 1 (Beginner – Core CV Project)

Student: SM Tawhid

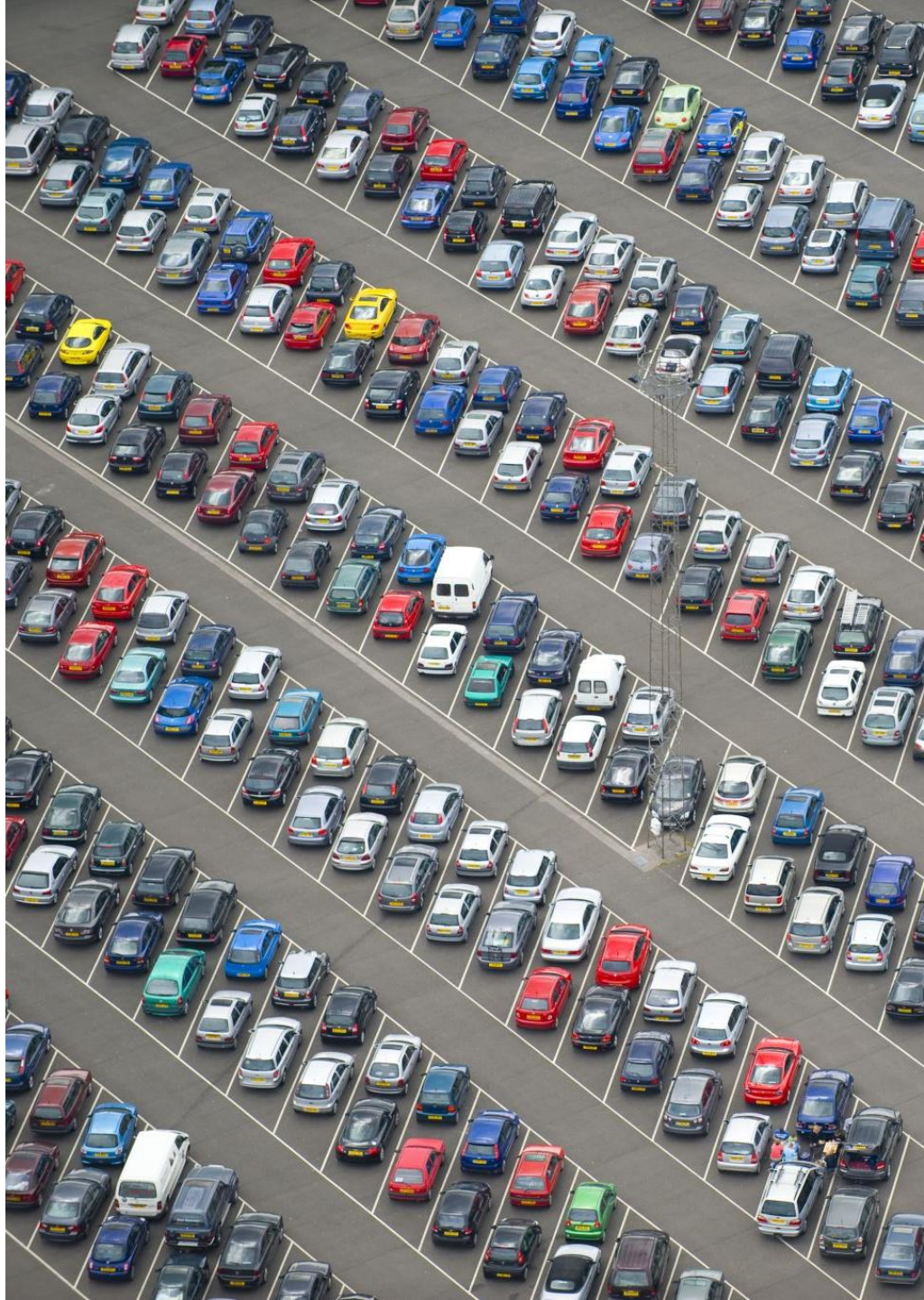
Email: w211161582@student.hccs.edu

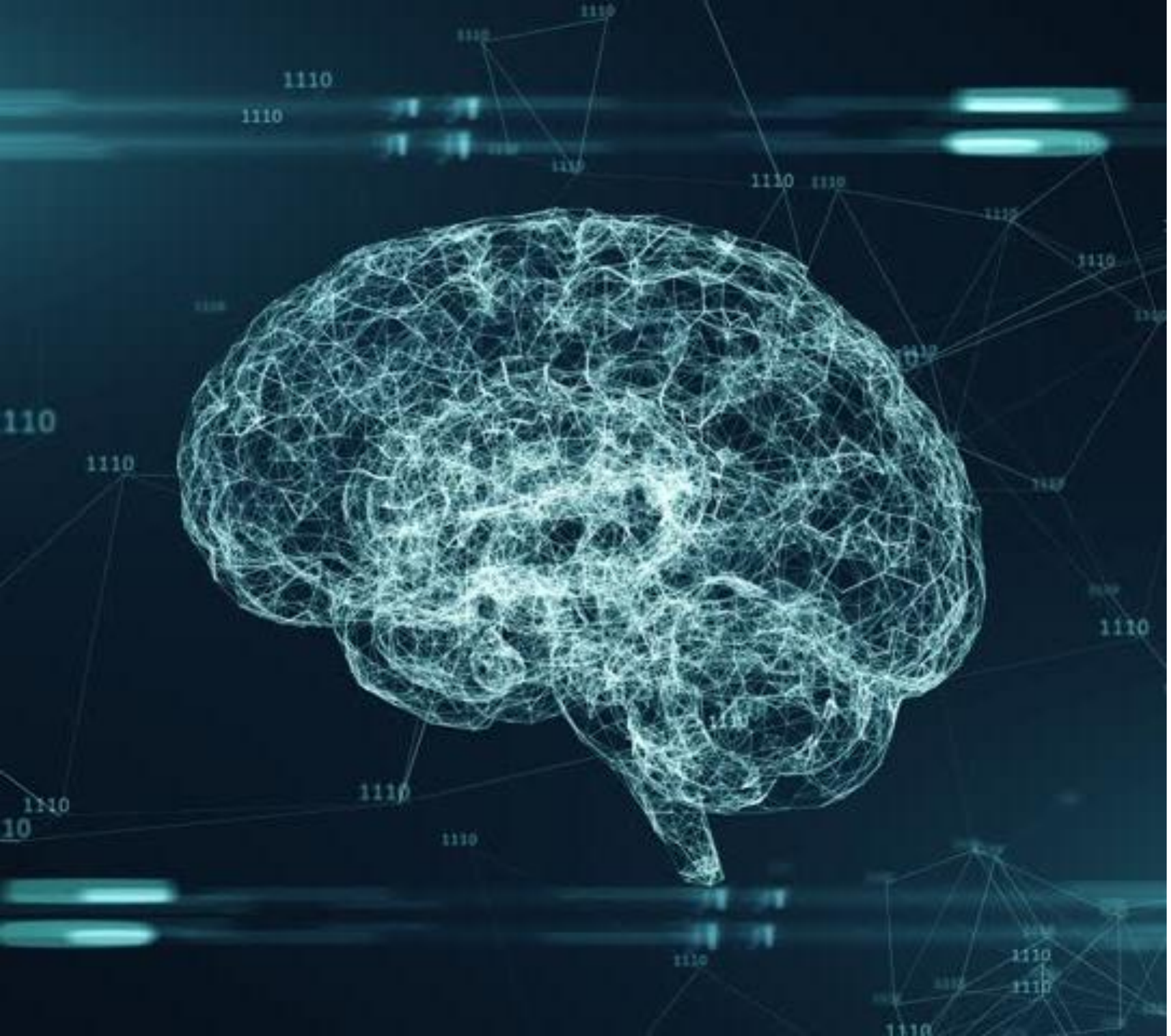
Budget: $0

Platform: Google Colab

The Problem

- Identifying license plates manually from vehicle images is slow and unreliable.
- Automated plate detection helps parking management, traffic monitoring, and public safety by speeding up analysis.





Proposed Solution

- Build an AI system that detects and localizes vehicle license plates automatically.
- Use a pre-trained YOLOv8 model to draw bounding boxes around plates in photos or frames.

Technical Approach

Framework: YOLOv8 (Python / PyTorch)

Task: Single-class object detection (“license plate”)

Steps: Load YOLOv8n → Fine-tune on dataset → Evaluate with mAP metrics

Tools: OpenCV, NumPy, Matplotlib on Google Colab

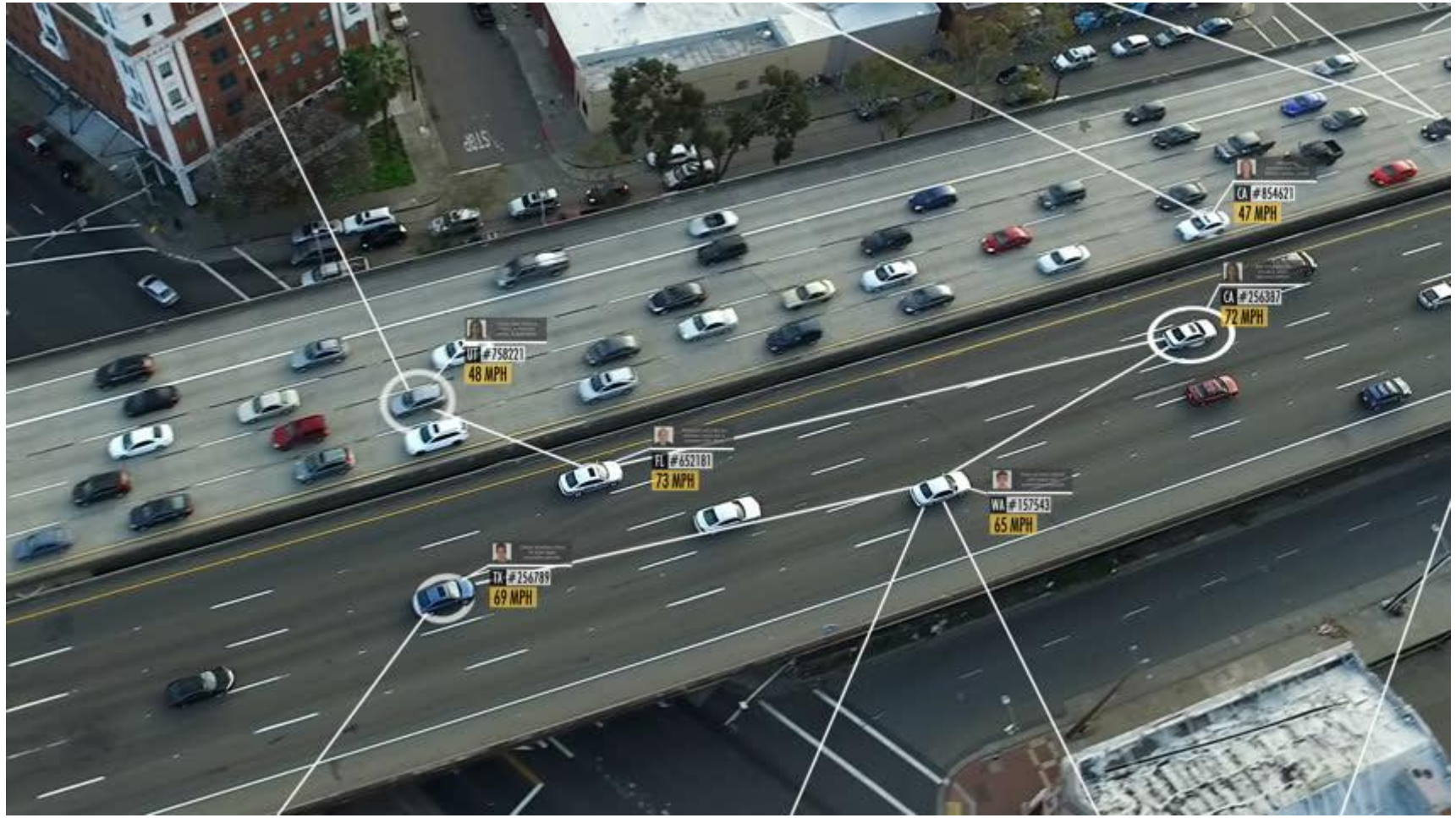
Data Plan

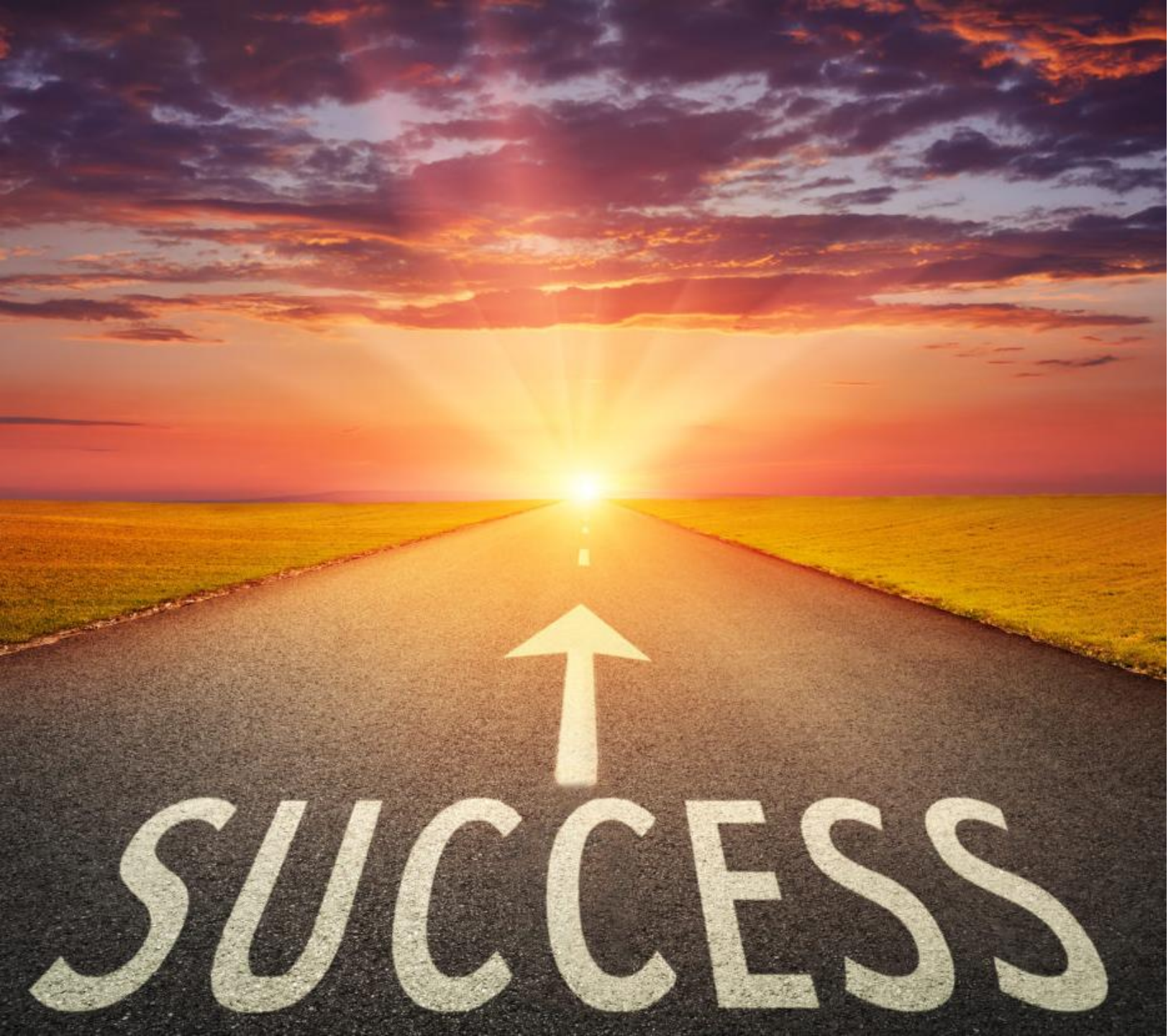
Dataset: *License Plate Recognition Dataset* by Roboflow Universe Projects

- 10,125 images, YOLO-ready format
- Single class label: “License_Plate”
- Split: 70 % train / 20 % val / 10 % test
- Link: https://universe.roboflow.com/roboflow-universe-projects/license-plate-recognition-rxg4e

System Diagram

Input: Vehicle Image → **YOLOv8 Model** → **Bounding Box Output** → **Detected Plate Region**





Success Metrics

Primary: mAP50 ≥ 0.85

Secondary: Precision > 0.90, Recall > 0.85

Goal: Accurate plate detection in > 90 % of test images

Week-by-Week Plan

Week 10: Proposal + Slides

Week 11: Setup Colab + Dataset

Week 12: Fine-tune YOLOv8 model

Week 13: Evaluate & Visualize Results

Week 14: Build Demo Notebook + Record Video

Week 15: Finalize README + Present

Challenges & Backup Plan

Challenges: Limited GPU runtime on Colab; dataset imbalance

Backup Plan: Use smaller subset or pre-trained YOLOv8n without fine-tuning

Resources Needed

Google Colab Free GPU (Tesla T4)

Pre-trained YOLOv8 weights

Libraries: torch, torchvision, ultralytics, opencv-python, numpy, matplotlib, pandas, Pillow

Dataset from Roboflow