Defence Project: Identifying Cars from Satellite Images or Drones

Deep Learning–Based Car Detection using YOLOv11-OBB

1. What is the Project?

The project Identifying Cars from Satellite Images or Drones using Deep Learning focuses on building an intelligent system capable of detecting cars automatically from aerial or satellite imagery.

With the increasing use of drones and satellite surveillance for defense and security applications, detecting vehicles accurately from large-scale images has become essential. This project applies deep learning–based object detection using the YOLOv11-OBB (Oriented Bounding Box) model to identify cars from overhead views, even when they appear at different angles, scales, or lighting conditions.

The goal is to train a system that can efficiently process aerial images, detect the presence of cars, and mark their locations, helping defense and surveillance teams with real-time monitoring, intelligence gathering, and operational planning.

2. What are Data Sources?

The dataset used for this project is obtained from open-source repositories such as Kaggle and Open Images Dataset by Google, which provide large-scale collections of labeled images for object detection tasks. A commonly used dataset for this purpose is the Car Detection Dataset (a subset of the Open Images Dataset or custom Kaggle datasets), containing thousands of images of vehicles captured under various conditions.

Key details of the dataset include:

- **Total Images:** Around 5,000–10,000 labeled images (depending on dataset version)
- **Annotations:** Bounding boxes specifying the location of cars in each image
- **Classes:** 1 (Car) or multiple vehicle types (Car, Bus, Truck, Motorcycle)
- **Image Size:** Varies across samples, typically resized to 224×224 or 416×416 pixels for model input

The dataset is preprocessed through image resizing, normalization, and augmentation (rotation, brightness adjustment, flipping, etc.) to improve detection accuracy and make the model robust to different lighting and angle conditions.

3. Useful Libraries

To build, train, and evaluate the car detection model, several Python libraries are used:

Category	**Library**	**Purpose**
Data Handling	os, pathlib, yaml, numpy	File management, dataset handling, and numerical operations
Visualization	matplotlib, opencv, PIL	Visualizing images and drawing oriented bounding boxes
Deep Learning	ultralytics (YOLOv11)	For training and inference of the YOLO object detection model
Evaluation	scikit-learn	For calculating precision, recall, F1-score, and ROC-AUC metrics
Monitoring	tensorboard	For visualizing training metrics and model performance in real time

4. Measurable Goals and Success Metrics

Type of Problem

This is an **Object Detection Problem** using deep learning. The model's task is to detect and localize cars within aerial or satellite images using bounding boxes.

Measurable Goals

1. **Detection Accuracy:**
   Achieve ≥98% detection accuracy on the validation dataset.

2. **Precision and Recall:**

- Precision ≥ 0.99 — ensuring detected objects are truly cars.

- Recall ≥ 0.98 — ensuring the model detects all cars present in an image.

3. **Mean Average Precision (mAP):**

- mAP@50 ≥ 0.98
- mAP@50:95 ≥ 0.96

4. **Speed:**
   Model should process and detect cars in under 1 second per image.

5. **Generalization:**
   The system should perform well across varying lighting conditions, resolutions, and vehicle orientations.

5. What Impact Your Project Will Make

This project has meaningful applications in defense, surveillance, and geospatial analysis. Key impacts include:

- **Defense Surveillance:** Helps in tracking and monitoring vehicles across borders or military zones.
- **Traffic Management:** Enables authorities to assess traffic density from aerial footage.
- **Disaster Response:** Assists in locating vehicles during rescue or evacuation missions.
- **Urban Planning:** Supports government and defense agencies in mapping and resource allocation.
- **Automation:** Reduces manual image analysis, improving efficiency and accuracy.

6. What is the Workflow

The project follows a structured workflow for seamless execution:

1. **Data Preparation:**

- Upload 6 folders (train/val/test images + labels) into Google Colab.
- Auto-generate the YAML configuration file for YOLOv11-OBB.

2. **Data Visualization:**

- Display random sample images with bounding boxes to verify annotations.

3. **Model Selection:**
    - Use pretrained **YOLOv11s-OBB** weights for transfer learning.
4. **Training:**
    - Train the model for 50 epochs with a batch size of 8 and image size of 640×640.
    - Automatically save checkpoints and results in Google Drive.
5. **Evaluation:**
    - Compute metrics such as mAP@50, mAP@50:95, Precision, Recall, and F1score.
6. **Model Saving:**
    - Save the trained model for future inference (car_final_yolov11.pt).
7. **Prediction:**
    - Upload new images for real-time detection and visualize results.
8. **Monitoring:**
    - Use TensorBoard to track training loss, accuracy, and validation metrics.

7. What is the desired outcome?

The desired outcome of this project is a robust, deep learning–based vehicle detection system capable of automatically identifying cars from drone or satellite images with high precision and reliability.

The system should:

- Detect cars regardless of orientation, scale, or illumination.
- Draw accurate oriented bounding boxes around each detected car.
- Deliver near real-time detection results for defense and surveillance use.
- Minimize false detections and missed cars.
- Be easily extendable to other object categories (e.g., trucks, buses, tanks).

Ultimately, this project demonstrates how AI-driven object detection can enhance defense intelligence, support national security operations, and improve the automation of aerial surveillance systems.