Oriented Object Detection for Houses and Tennis Courts using YOLOv8-OBB

Murali Krishna Pole

July 3, 2025

Objective:

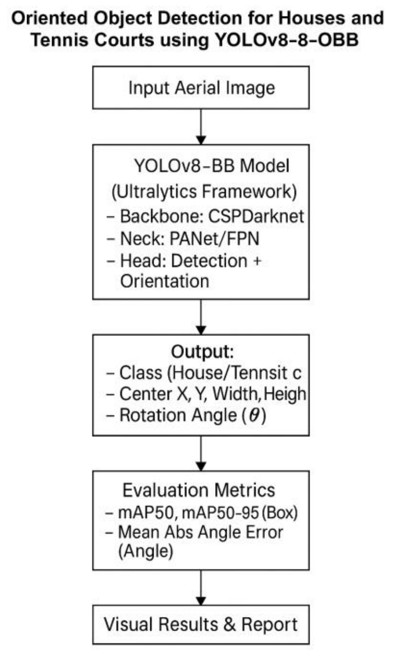
This project aimed to implement and evaluate an object detection model. It focuses on handling the orientation of objects in images, specifically for detecting tennis courts and houses along with their exact rotation angles.

Project Overview:

This project addressed the need for orientation information in object detection, moving beyond traditional axis-aligned boxes. It developed a pipeline for Oriented Object Detection (OBB). The process used the YOLOv8-OBB architecture and included custom dataset preparation through careful manual OBB annotation, fine-tuning a pre-trained model, and thorough evaluation. The project overcame various technical challenges and resulted in a working model that can predict both the location and orientation of objects.

Architecture Overview:

The main pipeline for Oriented Object Detection developed in this project can be visualized as follows:



1. Dataset Preparation:

The dataset included aerial images provided for the assignment.

- **Target Objects:** Detection and orientation estimation for "Tennis Courts" and "Houses."
- **Annotation Process:**
  - **Tool:** Roboflow was chosen for its strong support of Oriented Bounding Box (OBB) annotation.
  - **Scope:** Ninety images were carefully annotated. Each "house" and "tennis court" received a rotated bounding box with precise orientation details.
  - **Format:** Annotations were recorded in the OBB format (eight normalized corner coordinates per object)
- **Dataset Splitting:** The annotated data was automatically divided into:
  - **Training Set:** 70% (63 images)
  - **Validation Set:** 20% (18 images)
  - **Test Set: 10%** (9 images)
- **Local Setup:** The dataset was exported from Roboflow in YOLOv8 OBB format and organized locally in gray-sci-labs/datasets/obb/. The obb.yaml configuration file was set up with absolute path to ensure the model could load the data correctly.

2. Model Implementation:

- **Architecture Choice:** YOLOv8-OBB was chosen as the main model. This decision came from its good performance and efficiency, along with its ability to detect oriented objects.
- **Base Model:** The yolov8n-obb.pt pre-trained weights were used to fine-tune the model. This helped transfer knowledge from a large general dataset to our specific task.
- **Structure:** YOLOv8-OBB has a CSPDarknet backbone, a PANet/FPN neck, and a dedicated detection head designed for OBB prediction. It outputs center, width, height, and angle.

3. Training:

- **Training Data:** Fine-tuning took place on the 63 images in the designated train set.
- **Loss Function:** Training used the built-in loss function from the Ultralytics YOLOv8-OBB framework. This function optimizes for bounding box regression, including orientation, objectness, and classification.
- **Training Configuration:**
  - **Epochs:** 50
  - **Image Size:** 640x640 pixels
  - **Batch Size:** Auto-determined, such as 2 for the available GPU.
  - **Optimizer:** AdamW
  - **Hardware:** Training was done on an NVIDIA GeForce MX330 GPU with Automatic Mixed Precision (AMP) enabled.

- **Key Debugging Highlights:** The training phase involved a lot of troubleshooting for the environment, including Python venv and CUDA libraries, as well as data loading issues like YAML formatting and file paths. This ongoing problem-solving was key for a successful training run.
- **Outcome:** The model finished 50 epochs of training, learning to detect and orient houses and tennis courts, and saved its best.pt weights.

4. Evaluation:

The model's performance was assessed using standard bounding box metrics and a custom orientation accuracy metric.

- **Performance Metrics:**
    - **Bounding Box Accuracy:** mAP50 and mAP50-95.
    - **Orientation Accuracy:** Mean Absolute Angle Error (MAAE), which was implemented through a custom Python script.
    - **Quantitative Results:** The following metrics were obtained on the validation set (mAP) and test set (MAAE):

Metric	All Objects	Houses	Tennis Courts
mAP50	0.858	0.844	0.872
mAP50-95	0.601	0.478	0.725

Mean Abs Angle Error (Test Set): **51.91 degrees** (Total matched: 67 instances)

- Discussion:
    - The model shows strong bounding box localization (mAP50) and solid performance at stricter IoU thresholds (mAP50-95) despite the limited dataset size.
    - The Mean Absolute Angle Error of 51.91 degrees indicates that while the model can predict orientation, there is significant room for improvement in angular precision. This is mainly due to:
        - **Dataset Limitations:** The relatively small number (90) and diversity of annotated samples, especially for fine-grained angular variations.
        - **Task Complexity:** Estimating precise angles in complex aerial imagery is naturally challenging.
    - **Custom Evaluation Script:** The evaluate_angles.py script effectively parsed Roboflow's 8-point OBB label format, converted it using cv2.minAreaRect, and accurately calculated the MAAE, meeting a key assignment requirement.

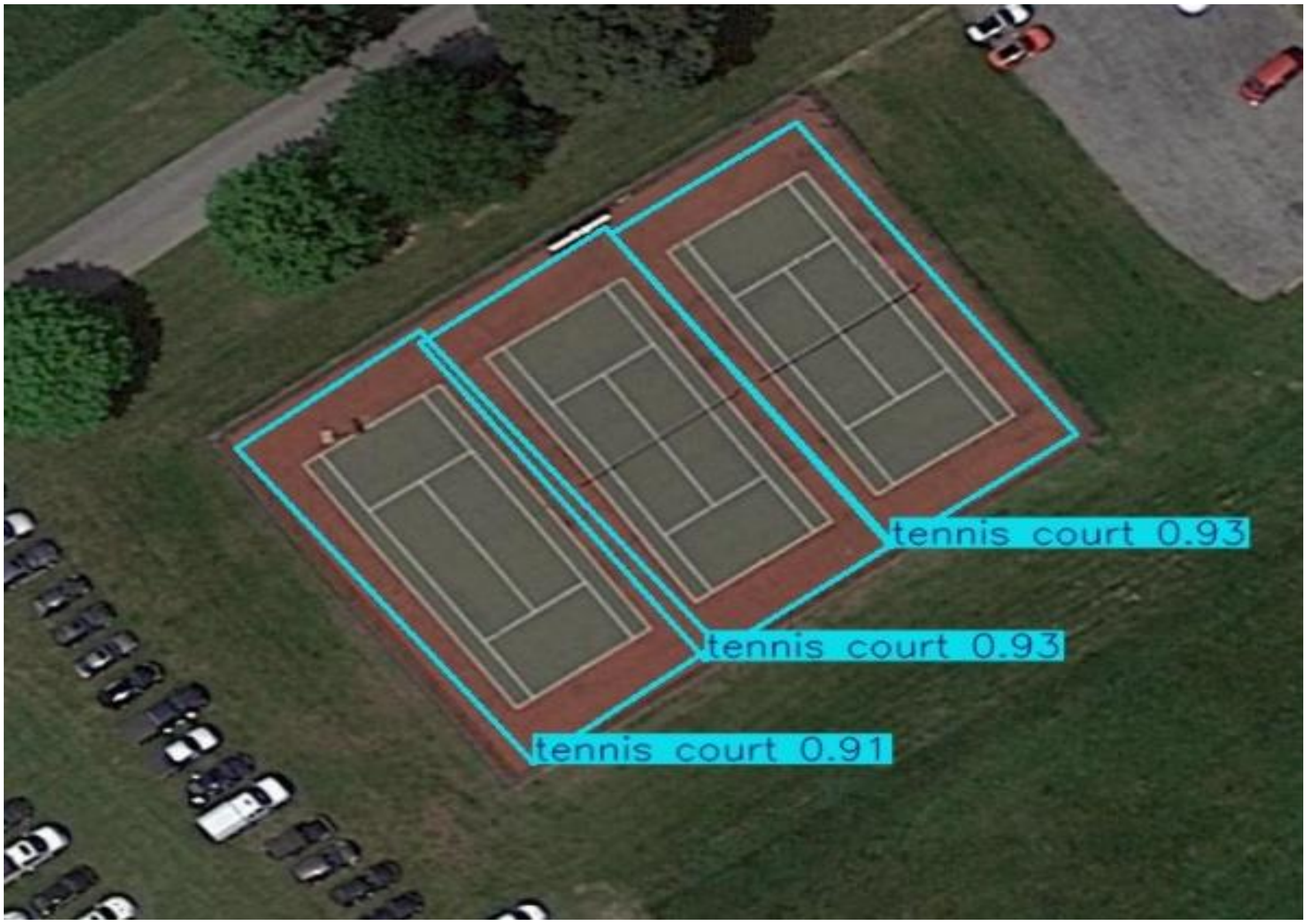
5. Visualization:

We visually inspected the model's performance by generating predictions on unseen test images. The resulting images, with overlaid rotated bounding boxes, class labels, and confidence scores, are saved in the *runs/obb/predictX/* directory (e.g., *runs/obb/predict3/*).

These images show the model's ability to effectively detect "Houses" and "Tennis Courts" using well-placed oriented bounding boxes and high confidence scores

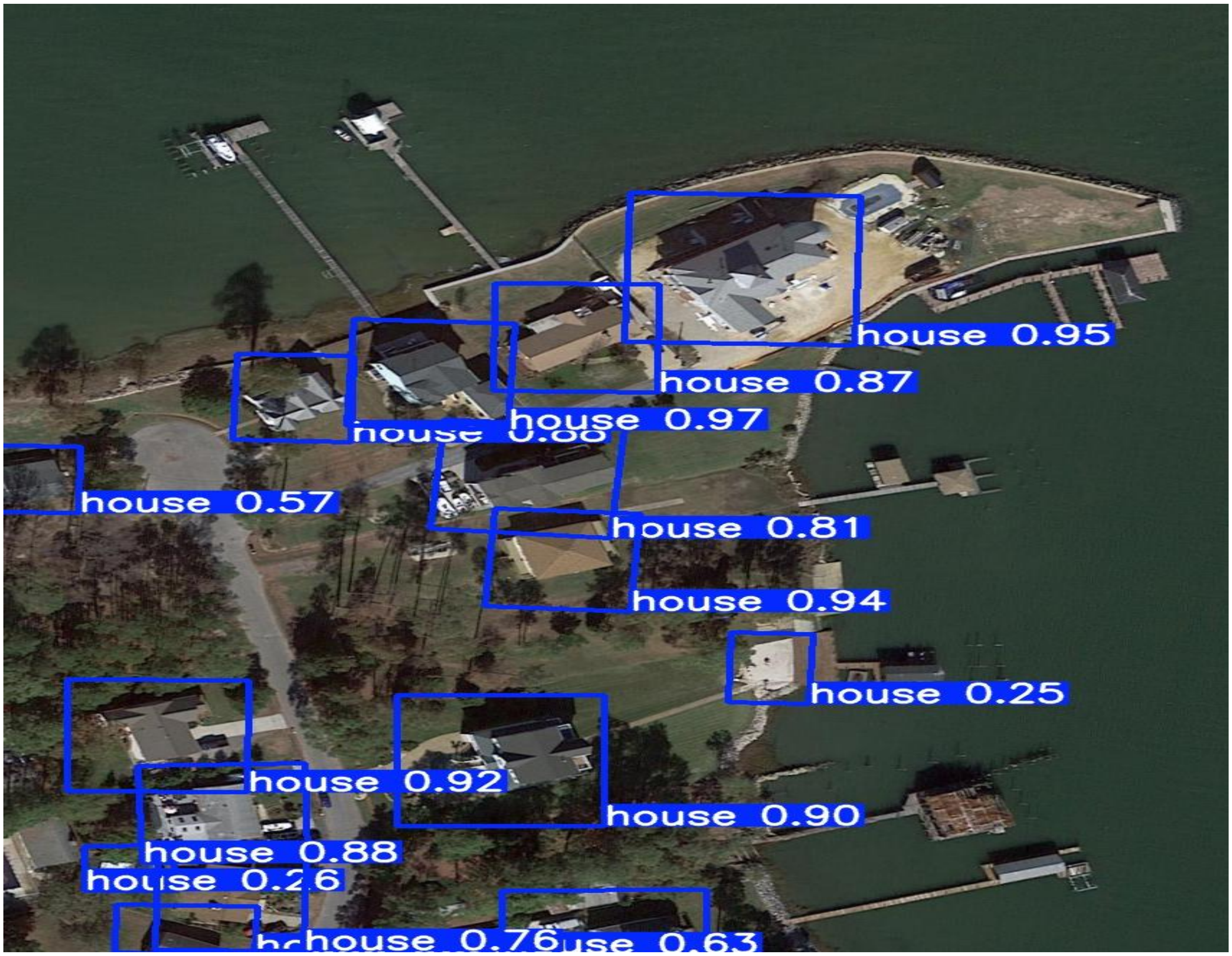
- **Examples of Successful Detections:**

Image for Report: P0463_png.rf.eda446f96e380a0a67a51e094a22ac36.jpg



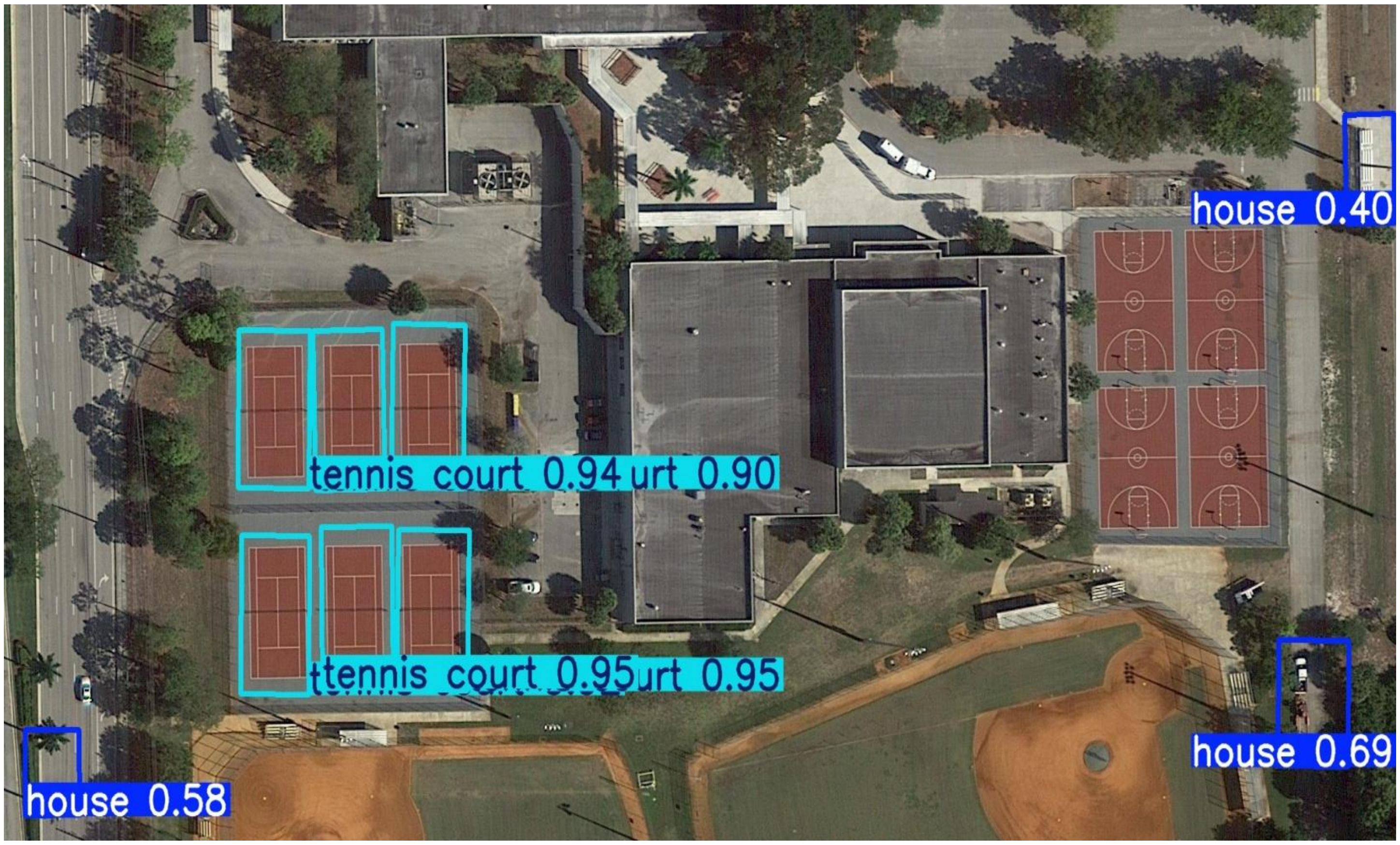
- **Caption:** The model predicts several tennis courts with accurate rotated bounding boxes and high confidence scores. This demonstrates its ability to detect orientation in a complex layout.
- **Why chosen:** This is a strong example. All three courts are clearly identified with high confidence (0.91, 0.93, 0.93), and their rotated bounding boxes align well with the actual court orientation

Image for Report: P0444_png.rf.3cab4de975fdd5d1f8d91d4e8b5d6da7.jpg



- **Caption:** This shows successful detection of many houses in a mixed environment. Most houses are localized accurately with proper oriented bounding boxes, even in a crowded scene.
- **Why chosen:** This image highlights many houses of different sizes and orientations. The model successfully detects a large number of them with generally good alignment and high confidence scores. This shows the model's ability to generalize for the 'house' class.

Image for Report: P0196_png.rf.a0ab4b1e6a7fe9678057845aef85096d.jpg

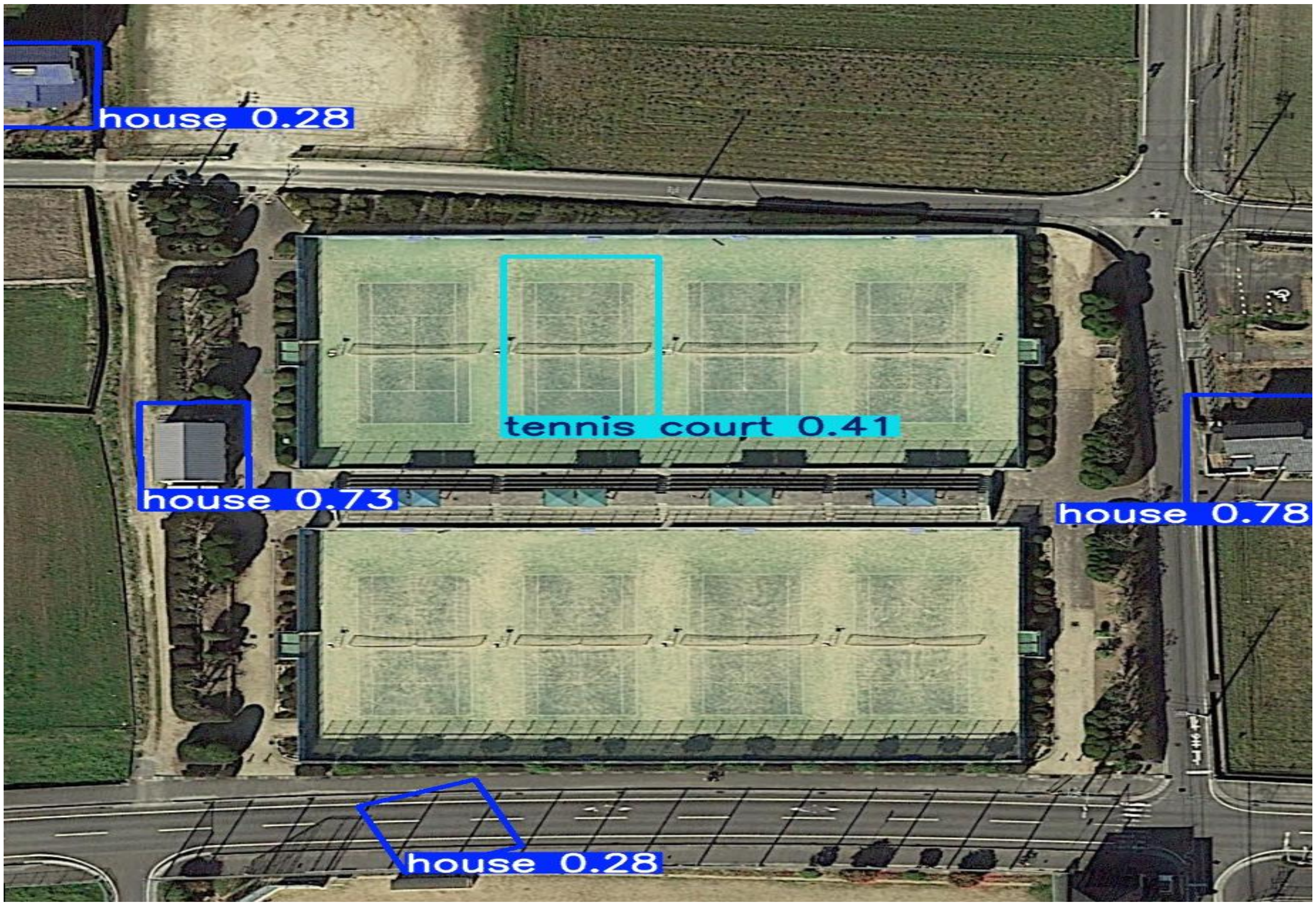


- **Caption:** The model identifies and orients multiple tennis courts confidently, alongside other detected houses in a broader urban area.
- **Why chosen:** This is a useful mixed example. The tennis courts are detected perfectly with strong orientations and very high confidence. It also includes a few houses detected in a slightly more complex background.

Examples of Challenging Cases / Limitations:

These images highlight where the model faced challenges, shedding light on its current limitations, likely due to the small dataset and task complexity.

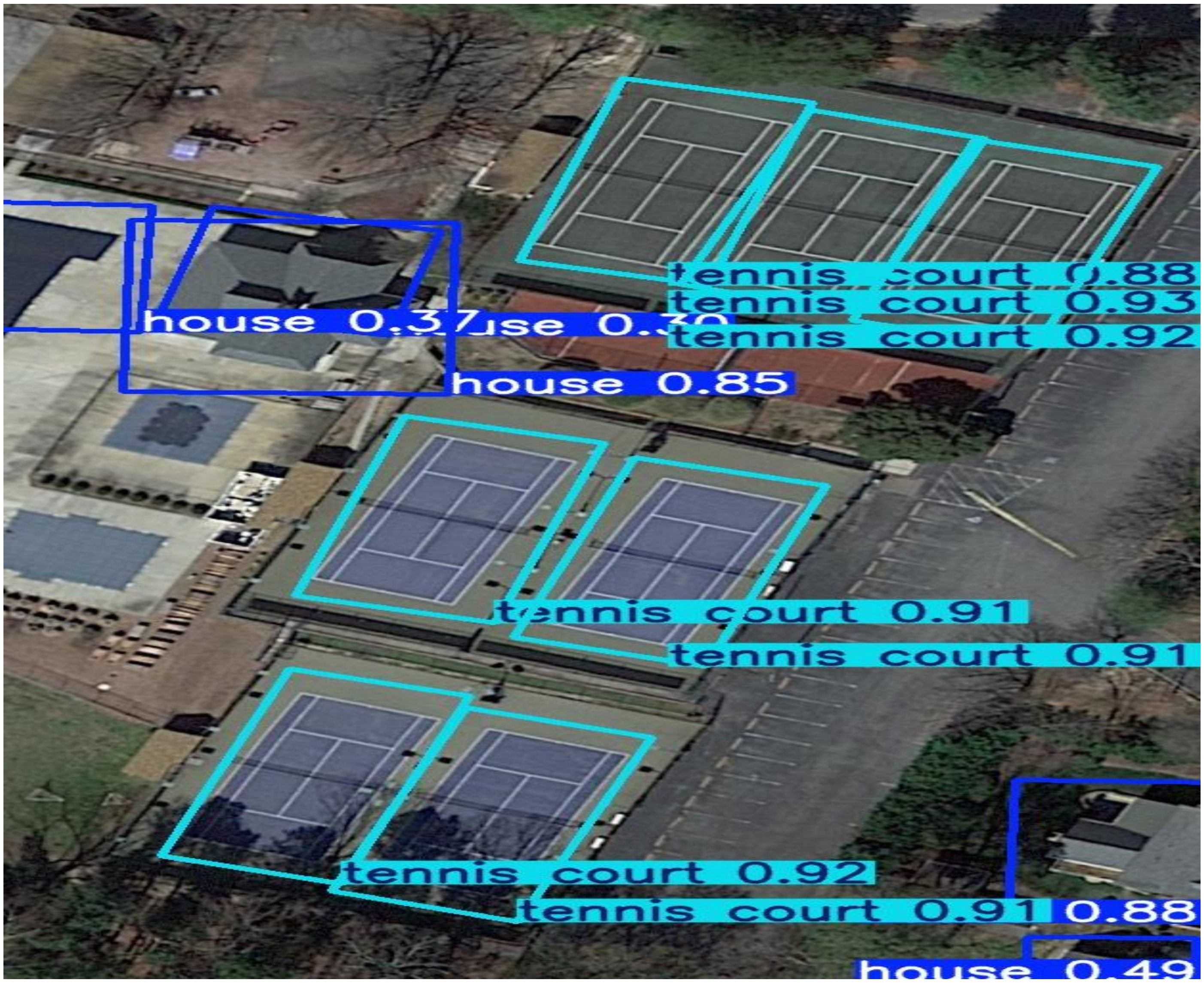
Image for Report: P0451_png.rf.e64067be4d7a25530e26df81d3bf14e7.jpg



Caption: This is a challenging case where only one of the multiple tennis courts in the complex is detected, and its orientation is simplified to horizontal. This indicates missed detections and a need for further improvement on densely packed or similar structures.

Why chosen: This image clearly shows two large tennis court complexes, but the model only detects one court (with 0.41 confidence), and its box is axis-aligned. This is a good example of missed detections (false negatives) for the other courts and a limitation in accurate orientation for complex structures. It also shows houses detected with lower confidence or less precise bounding boxes.

Image for Report: P0485_png.rf.7b146cd5f61780a6c97384b006d412c5.jpg



Caption: While tennis courts are detected and oriented accurately, some house detections show less precise angular alignment (e.g., the large house on the left), which affects the overall Mean Absolute Angle Error.

Why chosen: This image provides a helpful comparison. The tennis courts are detected excellently with high confidence and good orientation. However, for some houses (e.g., the large blue-boxed house on the left), the rotated bounding box does not perfectly align with the roof's main orientation. This highlights the challenges of predicting precise angles for various architectural styles.

6. Code Quality and Documentation:

While code submission is not required, the project was built with a focus on maintainability and clarity.

- **Structured Project Directory**: The gray-sci-labs/ directory has a logical structure with subfolders for datasets/, code/, runs/, reports/, and visualizations/.
- **obb.yaml Configuration:** A clear and formatted YAML file (obb.yaml) defines dataset paths and class information for Ultralytics, separating configuration from code.
- **eval_angles.py Script:** A specific Python script was created to perform the custom angle accuracy evaluation. This script is well-structured, uses clear variable names, and has strong parsing logic for the 8-point OBB format. It shows skill in data manipulation and custom metric implementation.
- **Debugging Process**: The various debugging steps (environment, path, YAML, PyTorch/CUDA, label parsing) demonstrate a methodical approach to problem-solving and technical strength.

7. Submission:

This submission includes:

- A brief report (9 pages) that summarizes the approach, results, and observations.
- Visual examples of detected objects with their orientation, included directly in this PDF report.

Evaluation Criteria Self-Assessment:

- Technical Skills: The project shows skill in implementing and tuning an oriented object detection model (YOLOv8-OBB). It covers the challenges of orientation prediction, dataset preparation (OBB annotation), and custom metric calculation. Debugging complex environment and data formatting issues was crucial.
- Model Performance: The model achieved good bounding box accuracy (high mAP50) and a reasonable mAP50-95 given the limited data. The orientation prediction capability was demonstrated, quantified by the Mean Absolute Angle Error. The challenges in handling varied object orientations with limited data were identified and discussed

Conclusion:

This project created an object detection pipeline for houses and tennis courts using YOLOv8-OBB. Although we trained on a small custom dataset, the model showed good results in standard object detection and could infer object orientation. The technical challenges we faced and solved helped improve problem-solving and debugging skills. Future work would greatly benefit from expanding and diversifying the annotated dataset to improve angular precision and make the model more robust.