Assignment: Detecting Swimming Pools from Aerial Images using YOLOv11

Introduction

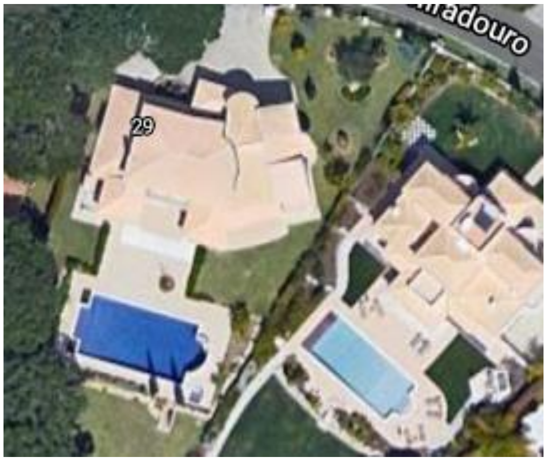
Object detection from aerial imagery is a valuable computer vision task with real-world applications ranging from urban planning to commercial analytics. In this assignment, you'll build a swimming pool detector using the latest YOLOv11 model. Your first task will be to annotate an unlabeled dataset using **GroundingDINO**, followed by training a YOLOv11 model using transfer learning. As an optional extension, you'll implement **oriented bounding box detection** for more precise localization.

Rationale

Swimming pool detection can power applications like:

- **Commercial Targeting**: Identify pool owners to offer cleaning, renovation, or smart pool equipment.
- **Insurance and Tax Assessment**: Support aerial property audits.
- **Urban Monitoring**: Track land use, water usage, and recreational infrastructure.

This project brings together AI for social and commercial impact.



Objectives

- 📍 **Image Annotation**: To speed up the annotation task start by an initial automated annotation using **GroundingDINO**. You then must review and correct the generated annotations to make sure it is correctly done. You can use Roboflow annotation tools to perform that. (**This is the only task roboflow use is allowed**)

- **Model Training**: Train a **YOLOv11** object detection model using transfer learning to detect swimming pools.
- *(Optional: for extra points)*: Use YOLO with oriented bounding boxes to enhance spatial precision in object detection.

Materials Provided

- **Image Set**: Aerial images with unlabelled swimming pools. https://drive.google.com/drive/folders/1aIao-EjuOLvppD_hP592TV1XZfO0jm0l?usp=sharing

- 📚 **Example Notebooks**:
  - GroundingDINO annotation guide:
    Notebook attached in same location as the assignment text
  - YOLOv11 training on custom dataset: *(Assuming YOLOv11 is backward compatible with YOLOv10 notebooks, adapt as needed)* https://colab.research.google.com/github/roboflow-ai/notebooks/blob/main/notebooks/train-yolo11-object-detection-on-custom-dataset.ipynb?ref=blog.roboflow.com

---

Task Instructions

Step 1: Annotate Images Using GroundingDINO

1. **Set Up Environment**
   - Use Google Colab or your local machine.
   - Install dependencies for GroundingDINO.
2. **Automatic Annotation**
   - Load the image set.

    - Run GroundingDINO with a prompt like "swimming pool" to generate bounding boxes.
    - Export the annotations in YOLO format.

Step 2: Train the YOLOv11 Model

1. **Transfer Learning**
    - Load a pre-trained YOLOv11 model: Yolov11 comes in different sizes (n,s,m,l,x), at least one is needed. Experimenting with more is preferred.
    - Adapt it for single-class detection (swimming pool).
2. **Training**
    - Split the annotated dataset into training (70%) and validation (30%).
    - Configure training parameters.
    - Train the model and monitor performance metrics.
3. **Evaluation**
    - Evaluate model performance (mAP, precision, recall).
    - *(Optional)*: Test the model on new aerial images from online sources.

Optional Challenge: Oriented Bounding Box Detection

- Instead of standard rectangular bounding boxes, use rotated boxes to more accurately detect oblique pools. You can privilege Yolo11-obb model (or earlier versions if you prefer)
- Reference on YOLOv8-Oriented: https://blog.roboflow.com/train-yolov8-obb-model/
-

Evaluation Criteria

- **Pipeline Quality**: Logical and well-structured approach.
- **Model Performance**: Detection accuracy and robustness.

- **Code Quality**: Clean, documented, and reproducible code.
- **Innovation**: Bonus for implementing oriented bounding boxes or creative uses of the model.

Submission Guidelines

- **Submit a ZIP file** containing:
  - Your annotated dataset.
  - Jupyter/Colab notebooks with documentation.
  - Screenshots or results of your model inference.
- **Deadline**: Submit by **Sunday, June 1st, 23:59** on Blackboard.

By completing this assignment, you'll gain hands-on experience in working with aerial imagery, image annotation tools, state-of-the-art object detection models, and their real-world applications.