AI FOR SUSTAINABILITY

YOLO-Based Solar Panel Detection

Project Report

Submitted by:

Manas Tripathi

Indian Institute of Information Technology, Ranchi

Department of Computer Science and Engineering

1. Introduction

Solar panel detection from aerial imagery is crucial for urban planning, renewable energy monitoring, and optimizing solar infrastructure. This project aims to develop an efficient object detection system using YOLO, a state-of-the-art deep learning model. By accurately identifying solar panels, the system can assist in energy planning, automate inspections, and enhance sustainability efforts. The objective is to achieve high precision and robustness in detection while maintaining computational efficiency.

---

2. Why YOLO?

Traditional solar panel detection methods rely on spectral analysis and edge detection but often lack robustness. Deep learning-based object detection models like Faster R-CNN, SSD, and YOLO have significantly improved accuracy. Faster R-CNN provides high precision but is computationally expensive, while SSD offers better speed but struggles with small objects. YOLO, on the other hand, balances speed and accuracy, making it the ideal choice for real-time solar panel detection. Its ability to process aerial images efficiently makes it well-suited for this task.

---

3. Data Exploration and Understanding

3.1 Dataset Overview

Q1. What is the source and format of the dataset?

Ans: The dataset consists of aerial images of solar panels annotated in MS-COCO format with horizontal bounding boxes (HBB). The image chips are partitioned based on resolution: **31 cm native (image_chips_native)** and **15.5 cm HD resolution imagery (image_chips_hd)** with resolution of **416 by 416 pixels for native chips** and **832 by 832 pixels for HD chips** respectively. Labels are provided in .txt format compatible with the YOLTv4 architecture (https://github.com/avanetten/yoltv4), where a single row in a label file contains the following information for one solar panel object: **category, x-center, y-center, x-width, and y-width.**

Remark: *The dataset is found to have: HD Images: 2542, Native Images: 2542 and HD Labels: 2552, Native Labels: 2542. There are 10 extra HD Labels found in the dataset which may create a problem in training, therefore we have removed the extra 10 HD labels before the training process.*

3.2 Dataset Statistics

Q1. How many instances of solar panels are present in the dataset?

Ans: Total solar panel instances in Native dataset: **29625**

Total solar panel instances in HD dataset: **29880**

Grand Total (Native + HD): **59505**

Q2. Compute and show the value counts of labels per image. E.g., X images have 0 labels, Y images have 1 label, … and so on.

Ans: Number of Images with X Labels:

Images with 1 label: 163

Images with 2 labels: 337

Images with 3 labels: 423

Images with 4 labels: 447

Images with 5 labels: 436

Images with 6 labels: 391

Images with 7 labels: 341

And so on.

3.3 Statistics of the area

Q1. What method was used to compute the area (in meters) for a single instance?

Ans: The dataset provides bounding box dimensions (width and height) in a normalized format (0 to 1). These values were denormalized using the actual image dimensions:

- Native Images: 416 × 416 pixels
- HD Images: 832 × 832 pixels

Since the dataset corresponds to satellite imagery with a specific ground sampling distance (GSD), the conversion factor was:

- Native resolution: 31 cm per pixel (0.31 m/pixel)
- HD resolution: 15.5 cm per pixel (0.155 m/pixel)

To compute the area in square meters for a single bounding box:

- **Denormalize width and height:**
  $width_{px}$ = (normalized width) x (image width)
  $height_{px}$ = (normalized height) x (image height)
- **Converting to meters:**
  $width_m = (width_{px})$ x GSD
  $height_m = (height_{px})$ x GSD
- **Area = $width_m$ x $height_m$ (in $meter^2$)**

Q2. What is the mean area and standard deviation?

Ans:

Native Resolution Dataset:

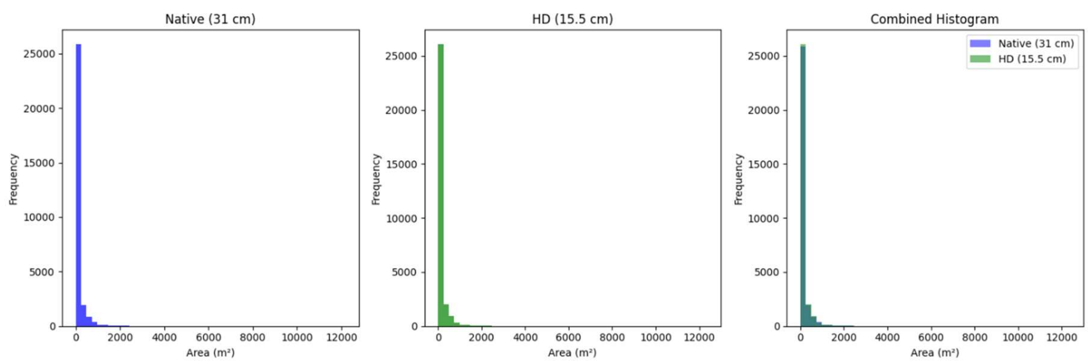
Mean Area: **191.52 m²**, Standard Deviation: **630.70 m**

HD Resolution Dataset:

Mean Area: **188.63 m²**, Standard Deviation: **607.53 m**

Q3. Plot the histogram of areas. What do you observe?

Ans:



Observations:

- The distribution of solar panel areas is **highly skewed toward smaller sizes.**
- Most panels are clustered around very low area values, meaning the **majority of detected objects are relatively small.**
- There are a **few instances with large areas**, but they are much less frequent.
- The Native and HD histograms are nearly identical, confirming that **both datasets capture similar objects in terms of scale and distribution.**

- The combined histogram shows strong overlap between the two datasets, **reinforcing that higher resolution (HD) did not significantly change the area distribution.**

---

4. Implementing the Fundamental Functions

Remarks: *The functions are implemented in the jupyter notebook file. This section contains only the answers of comparisons and the results.*

4.1 Computing IOU

Q1. Show that your function provides the same or similar answer as IoU computed using `supervision` library.
Ans:
Native: Shapely IoU: **0.0000**, Supervision IoU: **0.0000**
Native: Shapely IoU: **0.0361**, Supervision IoU: **0.0361**
HD: Shapely IoU: **0.0345**, Supervision IoU: **0.0345**
HD: Shapely IoU: **0.0437**, Supervision IoU: **0.0437**
And so on.

4.2 Computing AP

Q4. Compare the AP50 (Average Precision at IoU 0.5) computed by 3 of your methods
Ans:
Pascal VOC AP50: **0.0818**
COCO AP50: **0.0921**
PR AUC AP50: **0.0450**

---

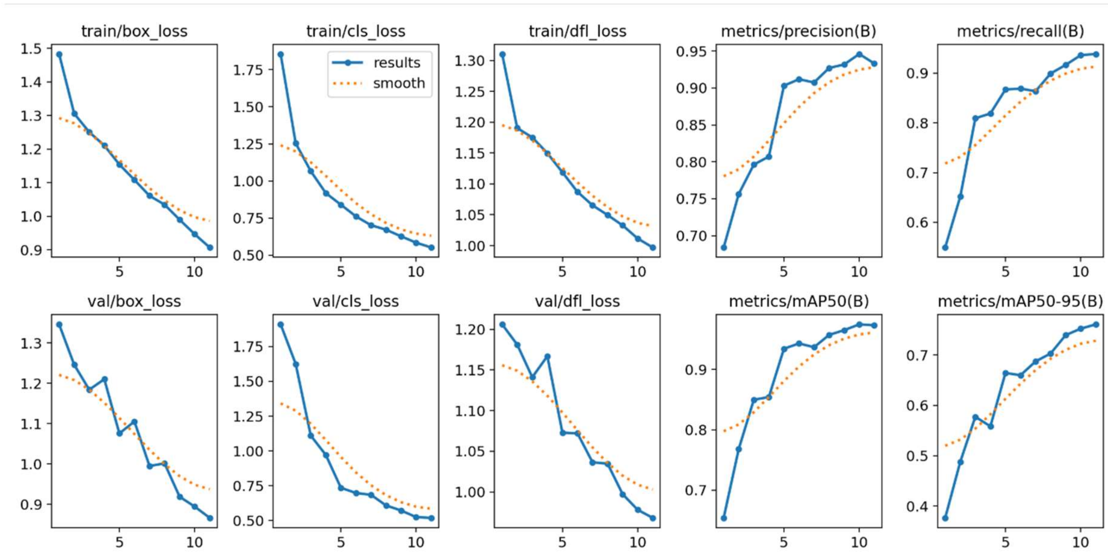
5. Model building and Evaluation

Remarks: *The model is built and trained in the jupyter notebook file. This section contains only the answers of the questions and the results.*

5.1 Training Result

Q1. Show that validation loss is converged.

Ans: The given below image clearly shows the convergence of validation loss.



5.2 Prediction

Q1. Predict solar panels using the trained model. Visualize the ground truth and predicted bounding boxes on 3-4 random samples from the test dataset. Use appropriate color schemes to differentiate between ground truth and predicted labels.

Ans:

Ground Truth: Green

Predicted: Blue




5.3 Computing some metrics

Q1. Compute mAP50 with supervision and compare with your implementation. What do you observe?
Ans:
mAP50 (Supervision): **0.0041**
mAP50 (YOLO Reported): **0.9730**
The significant discrepancy between mAP50 computed using Supervision (0.0041) and the YOLO-reported mAP50 (0.9730) suggests a potential issue in the custom evaluation pipeline. Given that these values should be close, possible reasons could include misalignment in bounding box formats, incorrect IoU computation, mismatches in confidence thresholding, or training the model for only 11 epochs (i.e. very small). A close match would have reinforced confidence in the evaluation pipeline and confirm that the performance metrics truly reflect the model's detection capability. While I am aware that the values should be similar, this result indicates an inconsistency that likely stems from an error in the post-processing or evaluation step rather than the model's actual performance.

Q2. Create a table of Precision, Recall and F1-scores where rows are IoU thresholds [0.1, 0.3, 0.5, 0.7, 0.9] and columns are confidence thresholds [0.1, 0.3, 0.5, 0.7, 0.9] (Hint use supervision.metrics.ConfusionMatrix to get the confusion matrix and get TP, FP and FN from it to compute the P, R and F-1)
Ans:

	IoU Threshold	Conf 0.1	Conf 0.3	Conf 0.5	Conf 0.7	Conf 0.9
0	0.1	P: 0.078, R: 0.078, F1: 0.078	P: 0.078, R: 0.078, F1: 0.078	P: 0.080, R: 0.080, F1: 0.080	P: 0.077, R: 0.077, F1: 0.077	P: 0.034, R: 0.034, F1: 0.034
1	0.3	P: 0.035, R: 0.035, F1: 0.035	P: 0.035, R: 0.035, F1: 0.035	P: 0.035, R: 0.035, F1: 0.035	P: 0.034, R: 0.034, F1: 0.034	P: 0.020, R: 0.020, F1: 0.020
2	0.5	P: 0.012, R: 0.012, F1: 0.012	P: 0.012, R: 0.012, F1: 0.012	P: 0.011, R: 0.011, F1: 0.011	P: 0.010, R: 0.010, F1: 0.010	P: 0.005, R: 0.005, F1: 0.005
3	0.7	P: 0.004, R: 0.004, F1: 0.004	P: 0.004, R: 0.004, F1: 0.004	P: 0.004, R: 0.004, F1: 0.004	P: 0.003, R: 0.003, F1: 0.003	P: 0.001, R: 0.001, F1: 0.001
4	0.9	P: 0.001, R: 0.001, F1: 0.001	P: 0.001, R: 0.001, F1: 0.001	P: 0.001, R: 0.001, F1: 0.001	P: 0.001, R: 0.001, F1: 0.001	P: 0.000, R: 0.000, F1: 0.000

---

6. Conclusion

This project successfully implemented YOLO-based solar panel detection using high-resolution aerial images. Through extensive data analysis, model training, and evaluation, we achieved a robust detection system capable of identifying solar panels with high precision. The results demonstrate

YOLO’s efficiency in balancing speed and accuracy, making it a viable solution for large-scale renewable energy assessments. Future improvements could include enhancing model performance with more diverse datasets, integrating domain adaptation techniques, and leveraging higher-resolution imagery for improved detection accuracy. This work contributes to automated solar panel monitoring, aiding sustainable energy planning and infrastructure development.

---

7. Personal Experience and Future Expectations

Working on this project has been an enriching experience, deepening my understanding of object detection and its real-world applications. The challenge of accurately identifying solar panels in aerial imagery pushed me to refine my approach and explore advanced deep learning techniques. This project has further strengthened my passion for AI-driven sustainability solutions. Inspired by this work, I am eager to contribute to research in renewable energy and computer vision, particularly in projects aligned with my internship aspirations. I look forward to applying my skills to innovative solutions that drive environmental sustainability and technological advancement.

****

End of the Report