

# OBJECT\_DETECTION REPORT

## 1. Challenges I Faced

- Working on the object detection assignment was both exciting and difficult, especially since I had very limited training data. The model was struggling to generalize, and my initial evaluations kept showing zero precision and recall. It took me quite a bit of time to realize that my predictions were not matching the ground truth at all. I also ran into errors related to OpenCV when loading images or resizing them, mostly because of incorrectly defined paths or trying to load files that didn't exist. Debugging these image loading issues was frustrating, but necessary.
- Another big challenge was evaluating model performance. Writing an `evaluate_model()` function that calculates precision, recall, and visualizes predictions took multiple iterations, especially when trying to convert normalized YOLO-format labels into pixel coordinates correctly.

## 2. How I Used AI Tools

- I used ChatGPT extensively during this project. I asked for help debugging errors (especially OpenCV image read/resize issues), structuring my evaluation function and It was like pair-programming with an assistant.
- For example:
- I used ChatGPT to generate the `evaluate_model()` function structure.
- I got help understanding why `cv2.imread()` was failing (because paths weren't valid).
- I also used it to explain warnings related to `.h5` vs `.keras` model formats.
- However, I didn't just copy-paste. I tried to understand what the code was doing and adjusted it according to my dataset structure.

## 3. What I Learned

- This project helped me understand the **importance of dataset quality and quantity** in computer vision. Even with a well-structured model, poor or small datasets can lead to zero performance. I also learned how YOLO-style bounding boxes are structured and how to convert them for visualization and evaluation.
- I also gained hands-on experience with:

- Model evaluation (IOU, precision, recall)
- Using OpenCV for image handling
- Visual debugging of model predictions

#### 4. What Surprised Me

- I was surprised by how quickly AI tools could generate code that would've taken me hours to write from scratch. But I was also surprised by how easy it was to overlook small things (like a missing image or bad file path) that completely broke the model evaluation.
- Another surprise was that even a “trained” model could output completely wrong predictions if the training data was too small. It made me appreciate how important dataset preparation and augmentation are.

#### 5. Thoughts on Code vs. AI Assistance

- AI tools are extremely helpful for speeding up the process and solving roadblocks, but they shouldn't be a replacement for understanding. I still had to tweak the code a lot, read the error messages, and figure out what worked for my dataset. So I think the best balance is:
- **Use AI for structure, and debugging**
- **Use own logic to adapt and understand**
- I found this balance really empowering. It made me more confident as a developer.

#### 6. Suggestions for Improving the Assignment

- It would help to have **a few sample images and labels** included to avoid setup issues.