**Assignment Experience Report**

**1. Challenges I faced during implementation**

* One of the main difficulties was dealing with images in the COCO dataset that were not annotated, which caused training to crash at the beginning.
* Debugging errors on data formatting and tensor shapes was not trivial to comprehend.
* It was tricky to make them produce output in the correct format for COCO evaluation and fix the JSON serialization errors I was getting with numpy data types.
* How to balance the dataset size with training time to get meaningful results when you have limited resources.

**2. How I used AI tools to help with coding**

* I generated initial code snippets for loading the dataset, training loops, and evaluation pipelines with AI assistance.
* AI enabled me to debug problems like missing targets and serialization errors by providing recommendations to filter datasets and change data types.
* And I asked AI to explain model architecture elements such as SSD and MobileNetV3.
* AI accelerated coding, but I always review and adjust the code to make sure I understand it completely..

**3. What I learned from the project**

* I got experience building and training an object detection model with a pretrained backbone and a SSD detection head was quite good.
* Finally, I have learnt how to carrying out data preprocessing, especially filtering out invalid samples when dealing with the real-world data.
* COCO evaluation metrics (mAP, precision, recall) can help me understand how detection models are evaluated.
* Seeing predictions was useful in interpreting the model’s strengths and limitations.

**4. What surprised me about the process**

* It was interesting to see how many COCO images have no corresponding bounding box annotations, which I did not realize.
* The pretrained SSD-lite MobileNetV3 could still have relatively good performance on detection with few training data taking only 50% of the dataset.
* The serialization of numpy floats to JSON failed.
* I also liked that COCO dataset was quite large and complex: over 200K images, 80 object categories, hence it was a challenging dataset to load in colab and Kaggle

**5. How I feel about the balance between writing code myself vs. using AI assistance**

* I love AI for development tooling that groks repetitive boilerplate and debugging hints.
* But I realize that the maintaining full understanding of how the code and models works does take some level of engagement beyond AI suggestions.
* The most effective workflow for me was to have AI draft and debug then to manually scan through, play with and learn from the outputs..

**6. Suggestions for improving this assignment**

* More obvious guidance or tools for filtering datasets to avoid common mistakes.
* Add example evaluation scripts that produce richer and more accessible metrics.
* Allow smaller dataset for quick experiment.
* Promote the use of visualization tools in interpreting results.

**Results Showcase**

**1. Summary of Key COCO Metrics**

| **Metric** | **Value** |
| --- | --- |
| mAP @[IoU=0.5:0.95] (All sizes) | 0.143 |
| mAP @[IoU=0.5] | 0.214 |
| mAP @[IoU=0.75] | 0.159 |
| mAP (Small objects) | 0.008 |
| mAP (Medium objects) | 0.140 |
| mAP (Large objects) | 0.311 |
| Average Recall (AR) @[maxDets=100] | 0.124 |
| AR (Small objects) | 0.172 |
| AR (Medium objects) | 0.184 |
| AR (Large objects) | 0.025 |
| Approximate F1 Score | 0.133 |

**2. Visualizations**





