Reflective Journal- Lab 09: Object Detection using Transfer Learning and Pascal VOC 2007 Dataset

1. **Conceptual Understanding**:

* *What is the main difference between image classification and object detection*? Image classification assigns a single label to an entire image while object detection attempts to identify the objects of interest in an image while localizing (or defining where objects are within an image) each identified object by drawing a bounding box around it.
* *How is this difference evident in the output of this exercise*? Though I suspect I ran into errors due to my lack of proficiency with Python, the additional steps involved in object detection including identifying objects, labeling them, and the formation of bounding boxes dramatically and negatively affected the model’s performance, where none of the predictions were accurate. Our work with image classification in earlier labs resulted in much higher performance metrics, perhaps due to its simpler nature.
* *Explain why we chose the SSD MobileNet V2 model for this task. What are its advantages and limitations, especially in the context of limited computational resources*? We chose this model because it is an SSD (Single Shot Detector), which in particular, is small and efficient, speeding up image processing and requiring less computational power. SSD MobileNet V2’s limitations are that it is less accurate compared to larger and more robust models, like those in the R-CNN family.

1. **Code Interpretation**:

* *Describe the role of the find\_images\_with\_classes function. Why is it useful when working with a large dataset like COCO*? This function finds images containing our target classes and can be especially helpful when working with large datasets that have many classes. This allows us to return only the target class and to filter out the irrelevant classes in large datasets.
* *In the plot\_detections function, how does the threshold value (threshold=0.5) impact the number of objects displayed*? My main struggle in this lab was that both run\_detector and plot\_detections are never defined in the notebook, which led to problems later on. I was unsure of how to define the functions myself and spent a lot of time trying to figure this problem out, which ultimately did not result in a successful outcome. If plot\_detections was defined and ran as intended, it would aid in visualizing the detected objects by drawing bounding boxes and labels on the image. The threshold value of > 0.5 means that only detections that are higher than 50% will be displayed.
* *Explain how the heatmap visualization helps you understand the model's confidence in its detections*. If the code ran as intended and I was successful in displaying a heatmap, it would’ve helped to depict the confidence scores using the map’s built-in color gradient, where the darker the color the higher the confidence score. It would also help to visualize low confidence areas and possibly depict which target classes the model performed better on compared to others.

3. **Observing Results and Limitations**:

* Run the exercise multiple times. Which types of objects does the model tend to detect more accurately? Which ones are more challenging? Can you explain why? My model did not perform as intended and I can only speak to this question in theory. Models tend to detect more accurately on objects that are large and unique because they take up more of the image and reduce the chance of extra noise. Backgrounds settings play a role as well, those with plain settings would perform better than those with busy, cluttered backgrounds. Objects that are smaller or less distinct are not detected as accurately.
* *Observe the bounding boxes. Are there any instances where the boxes are inaccurate or miss the object entirely? What factors in the images might be contributing to these errors*? Yes, I did have several instances of the predicted bounding boxes focusing on a smaller portion of a greater object. For example, one box is drawn around a car, but within that box, a smaller box has focused on the window, identifying it as a “chair”. This could be due to the environment or the atypical positioning of the cars, both have their hoods open which might have contributed to the model’s poor detection, especially if most other images containing cars within the dataset do not have their hoods open.
* *How would you expect the accuracy of the model to change if we had used the entire Pascal VOC 2007 dataset instead of a small subset? Why*? If we had worked with the entire dataset, I would expect the accuracy of the model to improve given the model would be trained on all 20 classes, greatly increasing diversity/ reducing bias and helping the model detect objects from a greater variety of images.

4. **Critical Thinking**:

* *How could you modify the code to detect a specific set of objects, like only animals or only vehicles*? First being aware of all 20 classes using the class\_names function, then to only detect animals for example, we could change the target\_class\_ids function to keep just those containing animals, in this case :bird, cat, cow, dog, horse, or sheep.
* *If you wanted to train your own object detection model, what steps would you need to take? What are some challenges you might encounter*? Steps would include data preparation, like importing a previously labeled (including bounding boxes) dataset from well-known sources like Kaggle or GitHub, utilizing train\_test\_split to divide the dataset into training, testing, and validation sets. Then choosing a model to train based on computational resources or time. Preprocessing data, using techniques like scaling, resizing, and normalization. Then training the model on the processed training set data and evaluating the model’s performance before tweaking and finetuning parameters.
* *Given the limitations of this model, in what real-world scenarios might it still be useful for object detection*? Given that this model is not as accurate as larger ones, scenarios that do not pose a risk to public health would be most suitable. Autonomous vehicles or medical imaging that utilizes CV for example would not be a desirable choice for this model. Lower risk object detection like for surveillance or industrial quality control are better options.