Name: Mohammad Sufyaan
Course: ITAI 1378
Instructor: Patricia McManus
Assignment: Lab 09

Reflective Journal: Object Detection using TensorFlow and Pascal VOC

This week, I explored *object detection* concepts through a hands-on exercise using TensorFlow and the Pascal VOC dataset. Diving into object detection really helped me understand how it differs from image classification. While classification assigns a single label to an image based on its primary content, object detection goes a step further by identifying and localizing multiple objects within the same image. This concept is essential in applications like self-driving cars and surveillance, where knowing *what* and *where* objects are located is crucial.

Learning about SSD MobileNet V2

For this task, we used the *SSD MobileNet V2* model, which I learned is particularly effective for environments with limited computational resources, like my Colab setup. SSD MobileNet V2 has a lightweight architecture that makes it suitable for real-time applications, balancing efficiency with reasonable accuracy. I can see why it's often chosen for applications where speed is key. However, I also noticed some limitations: it struggled with detecting smaller or less defined objects, especially compared to larger, more complex models. This was eye-opening because it showed me how choosing a model requires balancing speed and accuracy, especially when computational resources are limited.

Understanding the Key Code Functions

Working with functions like `find_images_with_classes` highlighted the importance of filtering when dealing with large datasets like COCO. This function was particularly helpful as it let me focus on specific object categories, making the dataset smaller and more manageable. In real-world tasks, I imagine this function could save time and computational power by enabling more efficient processing.

The `plot_detections` function allowed me to set a threshold, which I found directly affects the number of detections displayed. A higher threshold (e.g., 0.5) shows only those objects the model is most confident about, while a lower threshold includes more objects, even those with low confidence. Experimenting with this threshold taught me how it acts as a control to manage false positives and how I could adapt this based on whether I want more sensitivity or precision.

Visualizing Confidence with Heatmaps

Using heatmap visualizations was a new and exciting experience for me. Seeing the model's confidence levels overlaid on the original image made it easy to interpret the areas the model "believed in" most. Warmer colors like red showed strong confidence in its detections, while cooler colors like blue indicated less certainty. This tool felt essential because it provided an intuitive way to understand the model's decision-making process and gave me ideas on how heatmaps could be used to improve model accuracy in future applications.

Observations on Model Performance

Through repeated trials, I noticed the model could identify clear and large objects, such as people or cars, quite accurately. However, when it came to objects that blended into their surroundings—like a toucan bird with camouflaged colors—it struggled to detect them reliably. This made me realize how much an object's appearance and background can impact detection accuracy. For instance, the model often missed drawing a bounding box around the toucan, which I suspect was due to its irregular patterns and color blending with the background. Observing these results helped me appreciate the limitations of SSD MobileNet V2 in complex visual scenes.

The exercise also made me wonder how using the full *Pascal VOC 2007* dataset might impact the model's accuracy. With more data, I believe the model would generalize better and become more robust, but this would require higher computational power, which is a limitation in my current setup. This trade-off was important for me to understand as I realized how resource limitations shape decisions in model training.

Customizing Object Detection

As part of the exercise, I considered ways to customize detection for specific types of objects, such as animals or vehicles. One approach would be to train or fine-tune a model like YOLOv8 on a specialized dataset, or alternatively, filter the model's outputs to show only relevant classes. Adjusting the confidence threshold was another option, as increasing it could help focus on the most relevant objects by reducing false positives. I found it interesting how such small tweaks could help tailor a model for specific detection needs.

Insights on Training a Detection Model

Thinking about training an object detection model I highlighted the various steps and potential challenges involved. First, I would need to prepare the data, which involves collecting, annotating, and splitting it into training and test sets. Choosing the right model would be essential, balancing between accuracy and resource needs. Training and evaluating the model would follow, with an emphasis on fine-tuning for improved results. I anticipate some key challenges, especially around data quality and ensuring I have enough labeled examples. Another limitation is computational power, especially when trying to achieve high accuracy on a budget. Additionally, real-world variations in images and data imbalance could complicate training. These insights reinforced for me the complexity of training an object detection model and the careful planning that goes into it.

Name: Mohammad Sufyaan
Course: ITAI 1378
Instructor: Patricia McManus
Assignment: Lab 09

Real-World Applications of Limited Object Detection Models

Learning about SSD MobileNet V2 and its strengths and weaknesses helped me imagine where it might be useful despite its limitations. For example, in manufacturing, a model like this could assist with quality control and anomaly detection. In retail, it could manage inventory by detecting missing items on shelves. Autonomous vehicles use similar models for obstacle detection, while in the medical field, these models assist with diagnostic imaging tasks like tumor detection. Seeing the potential for real-world impact, even with limited resources, emphasized the value of lightweight object detection models and how they contribute to automation in diverse industries.