# 

# Object Detection using Transfer learning

Adapting an image classification task to an object detection task involves a fundamental shift in approach. Image classification assigns a single label to an entire image, whereas object detection identifies multiple objects and their locations. This distinction was evident in this exercise, as the model generated bounding boxes around detected objects instead of a single classification label.

The SSD MobileNet V2 model was chosen for its balance between efficiency and accuracy, making it well-suited for resource-limited environments. It offers fast inference times but struggles with detecting small or overlapping objects compared to models like Faster R-CNN. Given our computational constraints, this trade-off was acceptable for the scope of this exercise.

Working with a large dataset such as COCO requires effective filtering of relevant images, which is where the find\_images\_with\_classes function proved useful. By selecting images containing specific object classes, we optimized processing time and avoided unnecessary computation. Additionally, in the plot\_detections function, setting a threshold of 0.5 ensured that only detections with sufficient confidence were displayed, balancing false positives and missed detections.

The heatmap visualization was particularly helpful in interpreting model confidence, as it highlighted regions where the model was more certain about object presence. Brighter areas corresponded to higher confidence, allowing for an intuitive understanding of detection reliability.

Running the experiment multiple times revealed that the model performed well on larger, well-defined objects like cars and people, while struggling with small or occluded objects. Bounding box inaccuracies were often due to poor lighting, background clutter, or occlusion, all of which hindered detection accuracy. Using the full Pascal VOC 2007 dataset instead of a subset would likely improve performance, as the model would have access to more diverse object instances, leading to better generalization.

Modifying the code to detect only specific objects, such as animals or vehicles, would involve filtering the dataset and fine-tuning the model on those particular classes. If I were to train a custom object detection model, essential steps would include dataset collection, annotation, preprocessing, model selection, training, and evaluation. Challenges in this process would range from acquiring high-quality labeled data to managing computational costs and preventing overfitting.

Despite its limitations, SSD MobileNet V2 remains valuable for real-world applications that prioritize speed over absolute accuracy, such as surveillance and mobile-based object detection. Future improvements could include experimenting with more advanced models like YOLO and Faster R-CNN, as well as implementing techniques such as focal loss and anchor box optimization to enhance detection accuracy.

This lab reinforced the importance of transfer learning and demonstrated how pre-trained models can be efficiently adapted to new tasks. The experience highlighted both the challenges and the iterative process involved in optimizing an object detection model. Moving forward, continued experimentation with different architectures and fine-tuning strategies will be key to further improving performance in real-world scenarios.