In this lab I explored object detection with GPU with a model. The difference between image classification and object detection is how the model is able to recognize certain images it is looking for. In image classification, the model has a number of categories to identify the images given to it and place them where it best fits giving an accuracy of what right from what we know. In object detection, the model is identifying the images given within the objects inside each image with bounding boxes. The model then identifies the object inside each bounding box and decides if it is the correct object in classification. The goal is to identify the objects of interest inside an image. We chose SSD MobileNet V2 model for this task because the bounding boxes and class probabilities are predicted through the first pass of the model. SSD is generally faster because it is a one stage detector which allows multiple feature maps at different scales to improve the detections of objects at different sizes depending on the object. It is limited to the accuracy of the model after the single pass because it only gives off the output based on the predictions and probabilities from the one loop of information. With limited computational resources and a one pass across the model there is only so much SSD can return as the output because it works with what is given to it the first time.

The role of the find\_images\_with\_classes function allows it to find images containing our target classes or category. Basically, what we are looking for in return with the output. This function is useful with a large dataset like COCO because it simplifies the object detection process by proposal region and object classification steps into one stage. A large dataset means there are a lot of images being fed into the model expecting an efficient output that does not consume so much time. In the plot\_detection function, the threshold value 0.5 impacts the number of objects displayed by only detecting the objects of interest with bounding boxes and labels with an accuracy of 83%. The higher the threshold, the stricter the evaluation is which could lower the recall and precision. The heatmap visualization helps me understand the model’s confidence in its detections because it gives me a visual of what the model is thinking when detecting an object in an image. I am able to see if the model focuses on the correct objects in the image, if it is on track or not.

In addition, I can see that the types of objects the model tends to detect more accurately and are able to predict the objects in the image that are complete rather than the objects behind hidden and cut off. Like the horses that are complete and the car. There are instances when the bounding boxes are inaccurate due to other objects in the way covering the objects. This can be due to the single pass SSD which does not allow the model to be as accurate as the two stage detectors. If we had used the entire Pascal VOC 2007 dataset instead of a smaller subset the accuracy would be slightly more accurate because there would be more images to compare and contrast whereas the smaller set gives a limited comparison.

In conclusion, the code could be modified to detect specific sets of objects by being more specific in the input of images given to the model. Also choosing specific objects to be detected in the detection function. If I wanted to train my own object detection model I would choose a wide variety of photos to input into the model and specify a limited classification for the detector to find in each image. I would also prefer the two-stage detector instead of the one-stage detector because I would have a more accurate outcome even if more time is consumed. Time would be an issue because I would not be able to get an outcome right away and I would not be able to predict how long it would take based on the dataset I give the model. This model might still be useful for object detection for surveillance cameras because it is a consistent view for the most part. The model would be able to identify without much failure what is in the image.

Works Cited

Liu, Wei, et al. “SSD: Single Shot MultiBox Detector.” *Computer Vision – ECCV 2016*, vol. 9905, 2016, pp. 21–37, arxiv.org/abs/1512.02325, https://doi.org/10.1007/978-3-319-46448-0\_2.

‌

[ITAI 1378 2024 Mod 09 Advanced Architectures.pptx](file:///C:\Users\Neida\Downloads\ITAI%201378%202024%20Mod%2009%20Advanced%20Architectures.pptx)