Jonathan Smith
ITAI 1378
Chapter 9 Reflective Journal

In this lab, we explored object detection using TensorFlow library. Essentially, we used our categorization skills to instruct the model to detect objects by pre-training it from the dataset. I had to use GPU because of the computational power needed for this task so I used Google Colab. I used the Pascal VOC 2007 dataset which is used for visual object recognition and detection in machine learning models.

First I imported the TensorFlow libraries and then loaded the Pascal VOC 2007 dataset into the model for training. The TensorFlow library creates and trains machine learning models. TensorFlow Hub helps set up the model without setting it up scratch. TensorFlow datasets has a popular machine learning datasets that are readily available. Numpy performs calculations with numbers when the user is work with lists. Matplotlib and patches creates graphs, images in different formats. Requests allows the machine to use images from the Web and also can grab pictures from a link. Bytes10 allows the machine to handle the image data so it can be used with Pillow or OpenCV. These libraries set up the toolkit to work with images and building machine learning models. The Pascal VOC 2007 datasets uses bounding boxes and class labels for objects belonging to 20 different categories. This dataset trains ML models (“An Open Source Image Annotation Tool.”).

Next, I trained the model by loading the dataset into 2 small sample portions the training and validation. Then I extracted the names of the object classes in the dataset. This step is known as preprocessing which is necessary to locate and identify objects in images on what it learned from the data. Next, the dataset is downloaded and then the files were extracted from the files to access the images. TensorFlow datasets organizes the files to a structure that can be loaded to the program. This is needed to make the data readily available and useful for training (“SSD MobileNet V2 Object Detection Model.”). Next, I displayed the sample images from Pascal VOC

2007 dataset and included the bounding boxes that marked each object with a green bounding box that. The bounding boxed must match the image category to ensure the computer is comprehending.

The main difference between image classification and object detection is the type of information provided by the model. Image classification identifies what is in the image. In object detection, both what objects are and where they are located are both identified. The reason we used the SSMobileNetV2Model is because it can be used when a task requires fast image detection but has limited computational resources. This is usually used to run on smaller items that don't require a lot of computational power like cell phones ("SSD MobileNet V2 Object Detection Model."). Due to its cost efficiency, it works well with the Pascal VOC datasets because the machine can recognize a variety of objects with reasonable accuracy. Although it works well on less resource intensive models, they still are not as efficient or accurate as something like RetinaNet.

I used the find_images_classes to filter and retrieve images containing specific classes from the COCO dataset. I used plot_detections function to access a confidence threshold to filter the objects in the image. The threshold is the min confidence number required for the detection to be valid. The heat-map overlays colors on the image to display where the models are most confident. In this lab I made the threshold 0.5 so only the instances with a confidence score of 50% or higher were displayed. The heat-map visualization displays colors on the images to show areas where the model is most confident about its prediction. This is useful for evaluating the model's performance ("Facebook AI Research's Next Generation Library for Object Detection and Segmentation." ). The smaller objects in lower-resolution images can be harder to detect because they have less pixels. As far as the bounding boxes, they seemed to be

accurate in finding the images but they did not have the correct labels on the images nor were they necessarily classified correctly either. In my model, I did not see where the bounding boxes missed objects completely as the boxes did encompass each object in the image. The only smaller objects that were not captured by the bounding boxes but training the model with the entire dataset and/or continuously running the machine through epoch loops can cause the model to either learn, or overfit ("Object Detection Tutorial.").

The code could be modified to detect certain objects like animals and vehicles by finding out which class IDS in the VOc207 dataset correspond to animals and which and vehicles. Then you add the additional check to ensure only the bounding boxes for the selected items and target classes are displayed by modifying the loop. When you run the code, the list will include indexes for both new animals and vehicles. The loop then checks if class_id belongs to target_class_ids before the bounding boxes are printed.

As far as training my own model, I would have fed it the entire dataset to test its predictions. First, I would collect the data from the dataset and preprocess it. I would then choose a model architecture and then train the model. Next, the model's validity will be evaluated before fine tuning; and finally I would deploy the model and see how well it runs. Some of the challenges in object detection models include data security, cpu resources, bias, and overfitting. Overfitting is probably the most common challenge I would likely face if I was to continuously train the model on the same dataset.

As far as real life applications, this technology can be essential in many ways. Businesses can use this technology to thwart theft and monitor customers as they shop. The automobile industry uses this technology to program the autonomous vehicles to

detect humans and and objects. Retail can also uses this technology to manage inventory through object recognition techniques that would actually calculate the weight of an an item on the shelf as its picked up by the customer.

In conclusion I was able to gain an understanding of object detection and also understand its importance in the field of computer vision. This technology has been around for decades and can only continuously improve by the day. I found it very interesting that this is the same technology that Tesla uses for their electric vehicles to assist with drivers.

Works Cited

"SSD MobileNet V2 Object Detection Model." TensorFlow Hub, https://tfhub.dev/tensorflow/ssd_mobilenet_v2/2. Accessed 10 Nov. 2024.

"An Open Source Image Annotation Tool." Tzutalin, https://github.com/tzutalin/labelImg. Accessed 6 Nov. 2024.

"Object Detection Tutorial." PyTorch Tutorials, https://pytorch.org/tutorials/intermediate/torchvision_tutorial.html. Accessed 6 Nov. 2024.

"Common Objects in Context." COCO Consortium, https://cocodataset.org/. Accessed 6 Nov. 2024.

"Facebook AI Research's Next Generation Library for Object Detection and Segmentation." Facebook AI, https://github.com/facebookresearch/detectron2. Accessed 6 Nov. 2024.