Judith Barrios

07/22/24

ITAI 1378 Computer Vision

Patricia McManus

Journal on Transfer Learning Model

The difference between image classification and object detection are that image classification focus mainly on assigning a single label to an entire image based on its content. It answers the question of what is in the image. Meanwhile object detection focuses way more specifically on identifying and locating multiple objects within the image, each having their specific class label. This difference between them is evident in the output, where image classification usually provides a single label for the whole image, meanwhile object detection outputs many bounding boxes with each having class labels, showing the presence and location of individual objects within images.

We decided to use the SSD MobileNet V2 model for this task, because it has the balance between speed, accuracy, and efficiency, which are important factors given limited computational resources. SSD provides real-time object detection capabilities by leveraging MobileNet V2’s lightweight architecture, making it a good choice for applications where fast inference is required without having to sacrifice much in terms of accuracy. MoblieNet V2 is designed to be efficient, employing depth-wise separable convolutions to reduce computational cost, while maintaining the performance, this allows the model to run smoothly on devices with constrained computational power like mobile phones. However, a limitation is that it’s lightweight nature and efficiency may come at the cost of slightly reducing the accuracy compared to heavier and more complicated models. It can struggle with being able to detect very tiny objects or capturing fine -grained details in specific situations where higher resolution or deeper networks might perform better. Overall SSD MoblieNet V2 excels in efficiency and speed, though we still need to take in consideration what is needed to balance these advantages with the specific requirements and constraints of the task at the moment.

The ‘find\_images\_with\_classes’ function plays crucial roles in data preprocessing and analysis, especially when dealing with COCO, it efficiently identifies and filters images that have specific classes or categories of objects. For example, in COCO it encompasses a diverse array of object categories across multiple images, so the function helps developers pinpoint subsets of data related to their tasks. It can be used to take out images that have particular objects of interest, making it easier for targeted training models, or focused analysis of specific object interactions within places. It optimizes computational resources and accelerates the development and validation of computer vision applications.

In the plot\_detections function, the threshold value (threshold=0.5) determines the minimum confidence score required for an object detection to be displayed. Objects that have detection confidence scores below this threshold will not be showing, meaning adjusting the threshold affects the number of objects displayed by filtering out detections that have lower confidence scores. The higher the threshold, the fewer objects with higher confidence would be displayed, meanwhile lower threshold would display more objects, and even those with lower confidence scores. This is crucial for controlling trade-off between showing more potential objects or showing only specific detections that are considered more reliable.

A heat-map visualization helps understand the models confidence in its detection by showing the intensity or probability scores assigned by the model across spatial or feature dimensions. This indicates where the model is most confident in its predictions, and overlaying the heat-map on the input data like an image or sequence can visually identify areas that have powerful influence on the model’s decisions. Areas of high and low confidence become visually apparent based on the intensity of colors or values. This is helpful for understanding where the model is certain about its predictions, and also areas where it many be unsure or where further investigation is required.

When running the exercise multiple times, it can be seen that the model often tends to detect objects that are well represented in the training data more accurately. Meaning common objects such as cars, and specific types of animals are more often detected with higher accurate results, because they have diverse appearances and are rich in datasets used for training. In contrast, less frequently encountered objects, and those with variable shapes, textures, or orientations can be more difficult to identify. For example, small things or distant signs might be harder due to their size, variability appearance, lighting conditions, or partial obstruction. These factors can significantly affect the models result and the ability to generalize from the training data, causing inaccuracies.

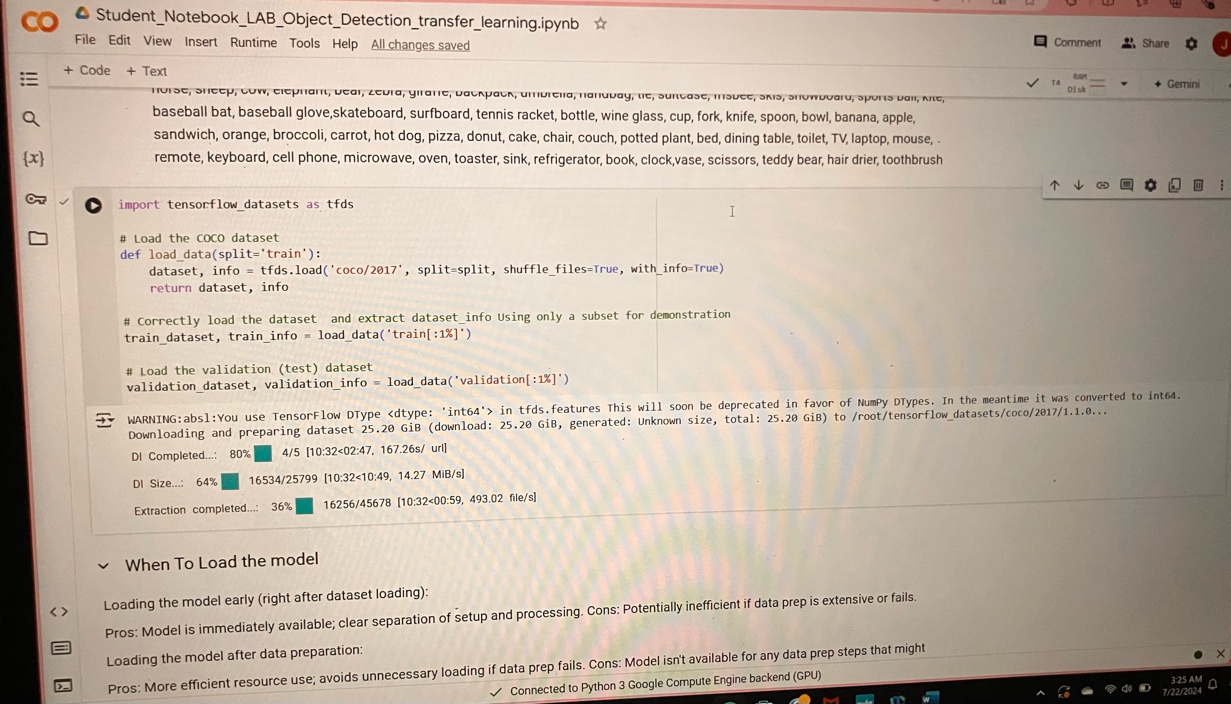
In analyzing bounding boxes, inaccuracies or misses often occur from several factors within an image. Some factors can include occlusions where parts of objects are concealed, many object scales affecting detection algorithms, and complicated backgrounds that can make the model confuse on the main goal. Lighting situations and object orientation can also be a major issue for instance, shadow or reflections can make the object appear different by making it harder for the model to correctly identify and box it. Also, where objects blend into their surroundings or have unusual shapes can cause problems with accuracy bounding box placement. All these factors are troublesome, and it shows the complexity of object detection and the necessity of robust algorithms capable of handling these problems.

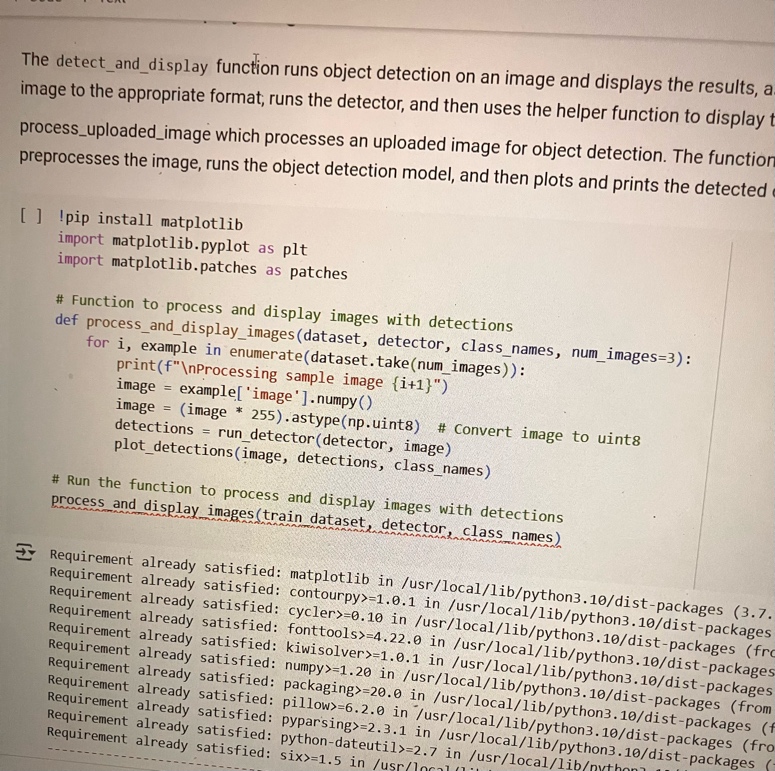
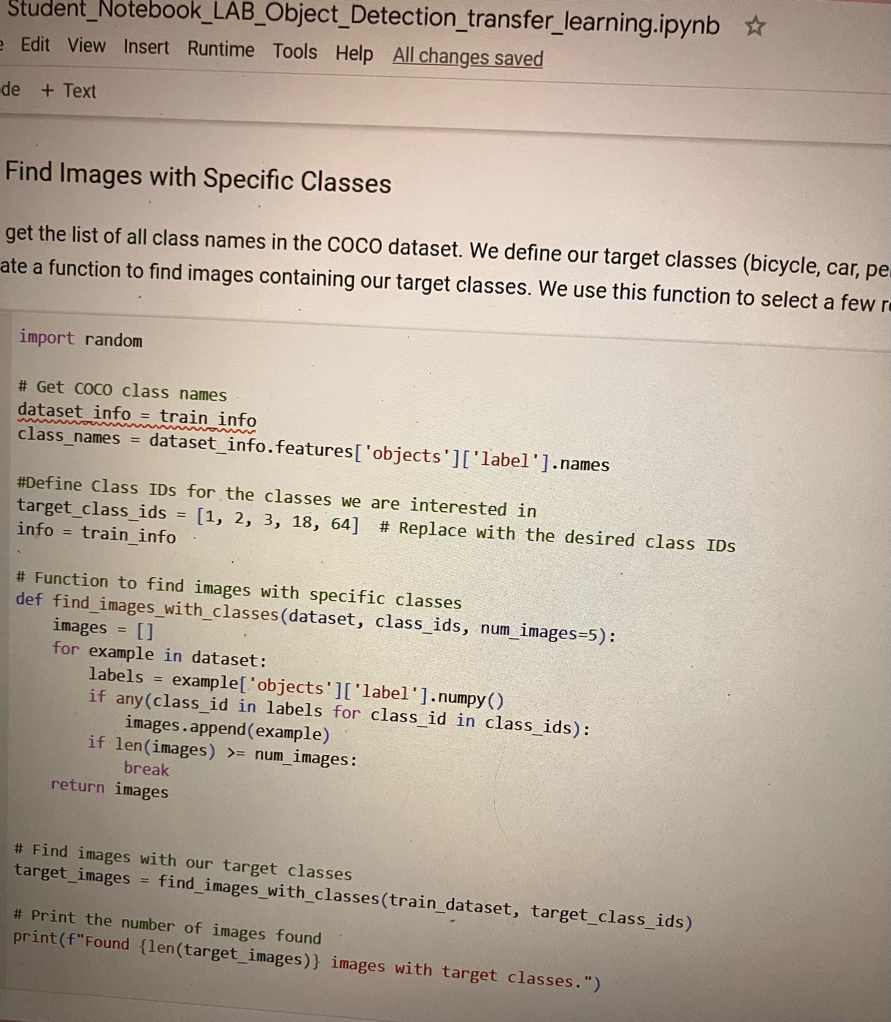
Using the whole COCO dataset instead of a small subset would most certainly improve the accuracy of the model by a lot. The COCO dataset is extensive and diverse, it contains a wide variety of objects, backgrounds, scales, and occlusion scenarios. By training the full dataset, it would increase the range of visual contexts, helping it learn more robust features and patterns. Also, the bigger dataset offers more examples for the model to learn from, which would reduce overfitting and improve generalization to new unknown data. Overall, the exposure of the larger COCO dataset would enhance the ability to accurately detect objects within different conditions, and improve its performance metrics.

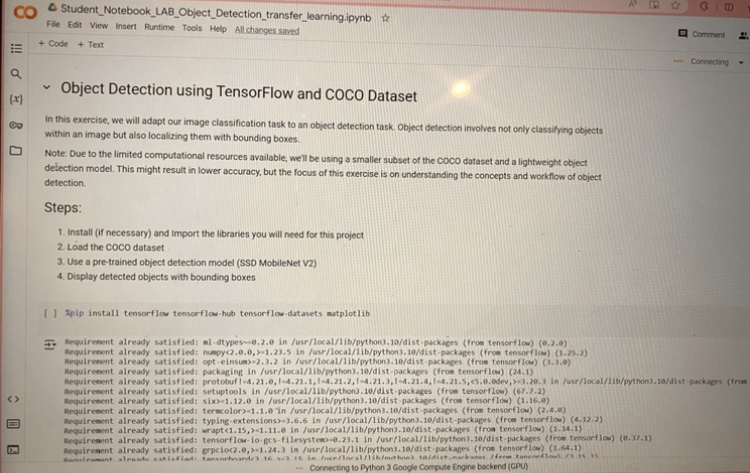
For the model to be able to detect a specific set of objects like only animals or vehicles, there would need to be several key components adjusted. First gather the dataset that contains images annotated with bounding boxes around the object wanted and ensure each annotation includes a class label to represent what it is. Then choose the right pre-trained object detection that handles diverse shapes and sizes well, and modify the data loading section to load the images and annotations for the classes of interest. Also, adapt the model architecture to match the number of classes wanted to detect, which would involve adjusting the final output layer to predict just the ones related to the application. Ensure the loss function and metrics are suitable for detecting correctly, using COCO with weights, and fine-tuning them. After, evaluate the model on a validation set to assess its performance, using precision, recall, etc. Leveraging transfer learning techniques is important to enhance the performance and ensure the model can effectively learn to detect and classify whatever objects we want.

To train my own object detection model, it would require various steps. First of all I would have to define clearly what objects I want the machine to detect, annotate my data accordingly, and collect enough labeled data that contain the objects of interest. Second I would have to clean and preprocess my dataset like resizing images, normalizing pixel values, etc. Argument my data to increase variability like rotating, scaling, flipping, adding noise, and improving generalization. Third I would select a suitable detection architecture like YOLO, SSD, or faster R-CNN, and consider factors like speed, accuracy, and ease of implementation based on the specific requirements. I can also initialize my model with weights from a pre-trained network, which helps leverage feature learning from genetic object recognition tasks. Then I would need to train my model on the annotated dataset by feeding it the prepared data and optimize its weights based on a chosen loss function through methods like gradient descent. Evaluate the trained model to assess its performance metrics with precision, recall, and mean Average Precision. Fine-tune hyper-parameters, which include batch size, learning rate, and more, plus architecture choices based on performance evaluations using regularization techniques for example. Once I feel confident and satisfied with my model I can deploy it for inference on new images or video streams. Some challenges that can occur during the process are insufficient data, meaning collecting and annotating a large complex dataset can be time consuming and expensive. Models can be complex and require substantial computational resources for training, overfitting can occur, it can get difficult to find the right set of hyper-parameters, which have a huge impact on the model’s performance, and the quality and consistency of annotations impact heavily on the model’s ability to learn effectively. All these problems can be prevented with enough research and right methods used for the model.

Despite the limitations of the model, it can still be very useful in real-world scenarios. For example, in fields like retail or inventory management where specific objects need to be detected and tracked down within controlled environments, the transfer learning model can prove to be a cost effective solution. It can be deployed in surveillance systems also to detect predefined objects like cars or people, because real-time performance and adaptability matter more than precision. Plus, in applications where the computational resources are limited, the model can create a balance of performance and resource efficiency. This type of model can still be very useful in many cases where efficiency and applicability is priority instead of cutting edge performance.







Work Cited:

“Transfer Learning and Fine-Tuning  :  Tensorflow Core.” *TensorFlow*, www.tensorflow.org/tutorials/images/transfer\_learning. Accessed 22 July 2024.

“Coco Dataset: All You Need to Know to Get Started.” *V7*, www.v7labs.com/blog/coco-dataset-guide. Accessed 22 July 2024.

“5 Significant Object Detection Challenges and Solutions.” *Plugger.Ai - Creative AI Assistant to Make Your Photo, Video, and Sounds Tasks.*, www.plugger.ai/blog/5-significant-object-detection-challenges-and-solutions. Accessed 22 July 2024.

Mehta, Vidish. “Object Detection Using SSD Mobilenet V2.” *Medium*, Medium, 22 May 2021, vidishmehta204.medium.com/object-detection-using-ssd-mobilenet-v2-7ff3543d738d.

Kniazieva, Yuliia. “The Difference between Image Classification & Object Detection.” *High Quality Data Annotation for Machine Learning*, Label Your Data, 7 Sept. 2023, labelyourdata.com/articles/object-detection-vs-image-classification.

“Heatmaps in Data Visualization: A Comprehensive Introduction .” *Inforiver*, 26 July 2023, inforiver.com/insights/heatmaps-in-data-visualization-a-comprehensive-introduction/.