L09 Object Detection using Transfer learning and Pascal VOC 207 Dataset

Martin Demel

Department of Science, Technology, Engineering & Math, Houston Community College

ITAI-1378: Computer Vision Artificial Intelligence

Patricia McManus

November 2nd, 2024

**Introduction**

During this laboratory work, I focused on learning key concepts in object detection by applying bounding boxes to locate the objects within the image. I have learned the technique by using TensorFlow and the Pascal VOC dataset. Furthermore, we utilized the SSD MobileNet V2 model, which was well-balanced between accuracy and computational efficiency.

**Understanding the key concepts**

A key difference I have learned during this laboratory work is that object detection differs significantly from image classification. At first, I assumed it would be very similar, but after completing the laboratory work, I can see a significant difference. **While image classification gives one label per picture, with object detection we identify multiple objects within one image and localize them by using bounding boxes.**

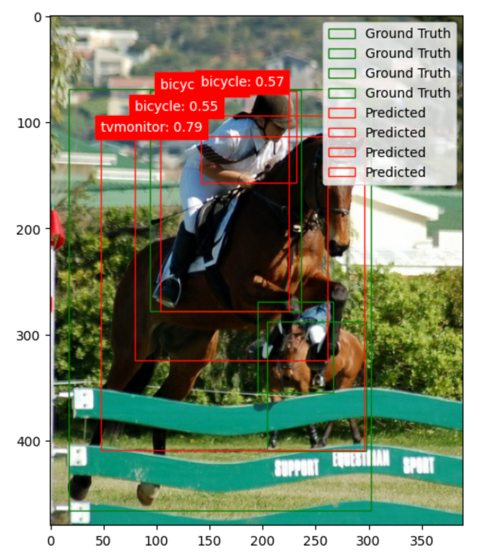
The SSD MobileNet V2 model was something new I had not heard about. I have learned that it’s a preferred choice when we have limited computational options. On the contrary, the model can underperform with complex images or small objects.

**Laboratory work**

During the laboratory exercise, I performed tasks that deepened my understanding of object detection. I also learned new code functions, such as find\_images\_with\_classes and plot\_detections.

**Preparation and dataset loading**

After downloading the file provided from Canvas, I uploaded it to my Amazon SageMaker environment. At first, I had a problem starting the project with the GPU. I received an error message saying there was no time available right now, even though I had 4 hours left to use. Eventually, the system let me start the project with CPU only. After ensuring that I had the right version of TensorFlow, we loaded the subset of the Pascal VOC 2007 dataset. Here I learned about the find\_images\_with\_classes function. This function allowed me to filter images based on specific object classes. This was very beneficial for the project overall because it managed memory and processing time well. That way, I could focus more on the actual work.

**This approach is very beneficial when I will be working with large datasets where efficient handling is essential.**

**Model selection**

SSD MobileNet V2 for the speed and low resource requirements. Even though we ran the project in a cloud environment, we always wanted to ensure we used the right tools with the right setup adequate for the laboratory work.

**A person on a horse jumping over a fence

Description automatically generated**As noted earlier, the model provides a balance between speed and accuracy, but it will significantly underperform in complex or dense scenarios.

I have adjusted the confidence threshold from 0.5 to 1.0. By doing so, the model only displayed detections about which it is 100% confident. This excluded any lower-confidence detections.

This resulted in not showing any predictions.

**Detection results across multiple runs**

Larger objects, such as people or cars, were detected accurately as they occupy more space in the image and have clear shapes.

Smaller objects like bottles and animals are more challenging to detect. The model's simplified architecture makes it struggle to identify smaller objects. This is by design, to my understanding, after reading more about the model’s design information.

**Bounding Box Accuracy**

Regarding bounding boxes, it performs well with large, separated objects. But when the objects overlap, bounding boxes miss or inaccurately outline objects. The design prioritizes speed and can sacrifice accuracy in scenes requiring detailed localization.

**Code modification and Real-World applications**

To detect specific types, for example, only animals, I would have to define a list of target classes and filter the detections accordingly. This would ensure that only relevant categories are displayed. This can be very beneficial, for example, for camera detection in wildlife. By doing so, we would exclude all other unnecessary objects.

Another example with the SSD MobileNet V2 model application can be traffic lights and car detection. Since the model can detect large, distinct objects, it will perform accurately.

**Other Object Detection models (Optional)**

Additionally, I have explored other object detection models on TensorFlow Hub and compared them with SSD MobileNet V2.

**Faster R-CNN** offers higher accuracy but is slower and has a higher resource demand. Ultimately, this makes the model less suitable for real-time applications unless resources are available.

**YOLO** provides a better balance of speed and accuracy than the SSD MobileNet V2 model, but it requires more computational power.

Working with a more powerful model showed improved accuracy with smaller objects but at the cost of increased processing time. It is very important to select the right model for the use case scenario, as it might significantly impact speed, accuracy, and demand for computational power.

A screenshot of a screen shot of a room

Description automatically generated**Personal Images**

Since I am using Amazon SageMaker, I had to rewrite the code so that I could run it. After a few tweaks to the code, I could upload the desired images.

A car parked in a driveway

Description automatically generatedTo my surprise, as you can see from the images below, the model had a problem detecting the car, and it mistakenly predicted it as a chair. Regarding the second picture, it is understandable that the model is trained on common objects like cars, dogs, horses, etc. But still, in the second image, it is very close; it identified the prediction as sheep, which is closer to a unicorn. ☺

**Conclusion**

This laboratory exercise offered me a deep insight into the difference between image classification and object detection. It was very interesting to see how different settings for confidence threshold can influence the detection and predictions.

The SSD MobileNet V2 showed the perfect fit for this type of laboratory work. However, model limitations become evident in more complex and dense scenes.

I am excited to continue further to explore the potential of the other models that I have mentioned in the optional section. By doing so, I will further develop and deepen my knowledge of how the models work and also for what type of use cases which models fit the most. As mentioned on several occasions, it is very important to ensure the right model selection for real-world scenarios.

I have learned a new skill today that I hope I will be able to practice in real life.

**Resources:**

GeeksforGeeks. (2024, June 12). *What is Object Detection in Computer Vision?* GeeksforGeeks. <https://www.geeksforgeeks.org/what-is-object-detection-in-computer-vision/>

Mehta, V. (2022, January 6). Object Detection using SSD Mobilenet V2 - Vidish Mehta - Medium. *Medium*. <https://vidishmehta204.medium.com/object-detection-using-ssd-mobilenet-v2-7ff3543d738d>

GeeksforGeeks. (2022, December 6). *YOLO v2 Object Detection*. GeeksforGeeks. <https://www.geeksforgeeks.org/yolo-v2-object-detection/>

GeeksforGeeks. (2023, August 23). *Faster RCNN | ML*. GeeksforGeeks. <https://www.geeksforgeeks.org/faster-r-cnn-ml/>