Object Detection using TensorFlow and Pascal VOC 2007 Dataset

1. What is the main difference between image classification and object detection? How is this difference evident in the output of this exercise?

Image classification - It entails assigning a label to an entire image based on its content. For example, in the previous assignment we did an image classification for muffin and chihuahua.

Object Detection - It involves finding and pinpointing objects in an image, marking them with bounding boxes, and labeling each one according to its type. For example, when we make a video, we see lot of different objects in the video, like people, animals, vehicles, birds, instruments, etc. It can also use for object inside of images.

In this exercise, we are mainly using object detection to find different objects within images. Part of the output of this exercise was multiple bounding boxes within the image so that the model could determine what all objects were present in the images.

1. Explain why we chose the SSD MobileNet V2 model for this task. What are its advantages and limitations, especially in the context of limited computational resources?

The SSD detector is ideal for identifying a large number of objects of various sizes and types because it uses multiple feature maps at different scales. MobileNet-v2 network models have struck a good balance between fast computation and accurate object recognition. MobileNetV2 is a light and efficient convolutional neural network (CNN) made for mobile and embedded vision tasks. Google researchers designed it to be better than the original MobileNet model. It's great in the context of limited computational resources because it balances size and accuracy well, making it perfect for gadgets with limited power and storage like mobile phones. These features, like depth wise separable convolution, inverted residuals, bottleneck design, linear bottlenecks, and squeeze-and-excitation (SE) blocks, make the model quicker and more efficient. Although it is efficient, it is not great for tasks requiring high accuracy and with complex backgrounds.

A diagram of a network

Description automatically generated

1. Describe the role of the find\_images\_with\_classes function. Why is it useful when working with a large dataset like COCO?

The `find\_images\_with\_classes` function is very handy when you're dealing with massive datasets like COCO (Common Objects in Context Dataset). It filters and retrieves images containing specific classes. It picks out images that have the specific thing we are interested in, making it way easier to manage our data. This means we don't waste time and computer power on irrelevant images, which is a big deal when training machine learning models. Plus, it helps us focus on the right images for our research or projects, letting us create custom datasets that fit our needs perfectly. Overall, it just makes working with big image datasets a lot more efficient and straightforward. This function is useful because of several regions. It handles data efficiently. COCO has a large dataset containing various objects and images and annotations. This function filters all the data we need and retrieve all the images and classes. It focuses and analyze specific patterns, characteristics, or behaviors related to those classes. While training the machine learning model, this function helps in creating subsets of data that include only the relevant classes, especially in object detection and image classification models. Using this function also allows for targeted data augmentation and preprocessing, improving model performance by focusing on relevant images. It optimizes computational resources by reducing the dataset size, making it more manageable. This function is especially useful for researchers and developers who need to create custom datasets for specific applications, streamlining specialized research and development.

1. In the plot detections function, how does the threshold value (threshold=0.5) impact the number of objects displayed?

When the threshold is high (like 0.7), you'll see fewer objects because only really confident detections are shown, which cuts down on false positives but might miss some true ones. With a low threshold (like 0.3), more objects pop up, including less confident ones, so you get more detections but also more false positives. Here in our assignment, we are using a middle-ground threshold value (0.5), which shows a balanced number of objects and is usually the default since it nicely balances catching true positives and avoiding false ones.

1. Explain how the heatmap visualization helps you understand the model's confidence in its detections.

A heatmap using as an example “plot\_detections\_with\_heatmap(image\_np, detections, class\_names)” is a visual representation that highlights parts of an images based on the model’s confidence. Warmer colors, such as red and yellow, indicate better confidence levels, while cooler colors such as blue and green indicate lower confidence. The areas that are highlighted in the image represent the areas that the model is focusing on. This helps us better understand what the model is focusing on and makes it easier for us to understand how to improve it. Heatmaps serve as a debugging tool by telling us which areas it is focusing on incorrectly, so we can find ways to improve the training process.

1. Run the exercise multiple times. Which types of objects does the model tend to detect more accurately? Which ones are more challenging? Can you explain why?

After running the exercise multiple times, the big, easy-to-spot objects like cars, buildings, or large animals are usually picked up by detection models more accurately because they stand out with their clear, distinctive features. Smaller or hidden objects, like birds in trees or people in crowds, are harder to detect because there's less detail and they often blend in with or overlap other objects. Also, objects that are hidden behind something else are also not being detected accurately. This is because a blocked image is not correctly matching what the model is being trained on, so it is harder for the model to pick it up.

1. Observe the bounding boxes. Are there any instances where the boxes are inaccurate or miss the object entirely? What factors in the images might be contributing to these errors?

After careful observation, I found that most of the bounding boxes were inaccurate or missed the object entirely. Inaccurate or missed bounding boxes often occur because of poor image resolution, occlusion, or complex backgrounds. These issues make it hard for the model to clearly isolate and identify objects. Also, the model is being trained on very little data, so the model is not as successful at identifying the objects and bounding boxes.

A car with the hood open

Description automatically generatedA screenshot of a screenshot of a horse jumping

Description automatically generated

1. How would you expect the accuracy of the model to change if we had used the entire Pascal VOC 2007 dataset instead of a small subset? Why?

Using the entire dataset would likely improve accuracy because a more extensive and diverse set of training examples allows the model to learn more robust features and generalize better to new data. This comprehensive training helps the model handle a wider variety of scenarios and object appearances, reducing overfitting and enhancing its performance on unseen images.

1. How could you modify the code to detect a specific set of objects, like only animals or only vehicles?

We can make some changes in the code to detect the specific set of objects. For example, we can define a target vehicle class and create a list for the classes we want to detect (e.g., vehicles – car, bicycle etc). The next one is we can get the corresponding class IDs and extract the class IDs for the target vehicle classes from the class names. And then we can modify the detection loop and filter the detections based on the target class IDs.

1. If you wanted to train your own object detection model, what steps would you need to take? What are some challenges you might encounter?

First, we need to gather many images and label all the objects we want the model to detect. Then, we pick a model—either a pre-trained one or we design our own custom model. Next, we train the model with our labeled data, teaching it to recognize and locate the objects. After training, we check how well it performs using test data to make sure it's accurate enough. Once we are happy with the results, we export the trained model so we can use it in our application. Finally, we might need to tweak and fine-tune the model based on how it performs in the real world to keep it working well.

We can encounter some challenges while training our own data. It can be time consuming and costly. if we are annotating images, it can be prone to human error. Picking the right model architecture and hyperparameters can be difficult and often feels like a guessing game. We have to try out different setups and tweak things here and there to see what works best. There is a lot of trial and error. It also needs a lot of computational resources. We likely have to work with GPU’s or it could take a longer time to complete the task. It is necessary to make sure the new data won’t overfit on the training set, which is a common challenge.

1. Given the limitations of this model, in what real-world scenarios might it still be useful for object detection?

SSD MobileNet V2 maybe not very good choice where it requires high accuracy. It is great for real-world scenarios where speed and efficiency are key. It is perfect for real time applications like video analysis (traffic monitoring). In robotics, it can be used for obstacle avoidance, and interaction with objects. It can also be used for defect detection in manufacturing process, inventory tracking in retail environment, assistive technologies like in cars to help with navigation.

Reference –

Google colab

Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., & Zisserman, A. (2010). The PASCAL Visual Object Classes (VOC) Challenge. International Journal of Computer Vision, 88(2), 303-338. doi:10.1007/s11263-009-0275-4.

[ITAI 1378 2024 Mod 09 Advanced Architectures.pptx](https://eagleonline.hccs.edu/courses/256864/modules/items/17085987" \o "ITAI 1378 2024 Mod 09 Advanced Architectures.pptx)

Younis, Ayesha, Li Shixin, Shelembi Jn, and Zhang Hai. "Real-Time Object Detection Using Pre-Trained Deep Learning Models MobileNet-SSD." ACM, 2020, pp. 44-48. <https://doi.org/10.1145/3379247.3379264>