Midterm Steps

Albritton

April 12, 2022

Objective: Implement a simple solution to run Detectron Mask R-CNN algorithm for object detection and instance segmentation with webcam.

1. Installed Detectron

Graphical user interface

Description automatically generated with medium confidence

1. Load COCO Dataset

This was an extremely cumbersome task. I attempted multiple methods of merely accessing the COCO dataset, including APIs, Google Cloud Platform, and directly downloading samples from the internet. I’ve never worked with a dataset this large nor have I worked with image data prior to this course, so it took some trial and error to figure out how to load the image files to my local workspace in Google Colab.

Graphical user interface, text, application, email

Description automatically generated

In addition to accessing the image files in the COCO dataset, I had to figure out how to convert a json file containing the bounding box and image file information into a dataframe. To do this, I used the importer package from pylabel.

Graphical user interface, text, application

Description automatically generated

1. Train model with COCO Dataset
   1. Given the size of the dataset, I had to cut down on the number of images substantially. I initially attempted to run the model on the full dataset, but the performance was poor (< .60) and it took over six hours to train the model. This was too cumbersome to deal with when troubleshooting/debugging my code.
   2. To cut down on the size, I randomly sampled from the 2017 dataset and I only retained images with labels belonging to eight classes ('person', 'backpack', 'handbag', 'tie', 'suitcase', 'cell phone', 'laptop', 'tv').

Subsetting classes -

Graphical user interface, text, application, email

Description automatically generated

Randomly selecting a sample of 500 pictures –

Logo, company name

Description automatically generated with medium confidence

Training model (hyperparameters)-

Graphical user interface, text, application, email

Description automatically generated

1. Get camera using OpenCV using the VideoCapture()
   1. Instead of using VideoCapture(), I opted to capture a single image first, and hoped to eventually move on to video. Unfortunately, I ran out of time to do so. My model was also still underperforming so it would’ve been a fruitless effort to move to video, when I wasn’t even labeling stable images. Ultimately, I opted to submit a notebook with code that ran, rather than my other failed attempts at training with large datasets, VideoCapture(), and visualizations.

Graphical user interface, text, application

Description automatically generated