CMP_SC 8690: Computer Vision

Homework 3B: Object Detection Using Pre-Trained Deep Learning Networks

By: Mikey Joyce

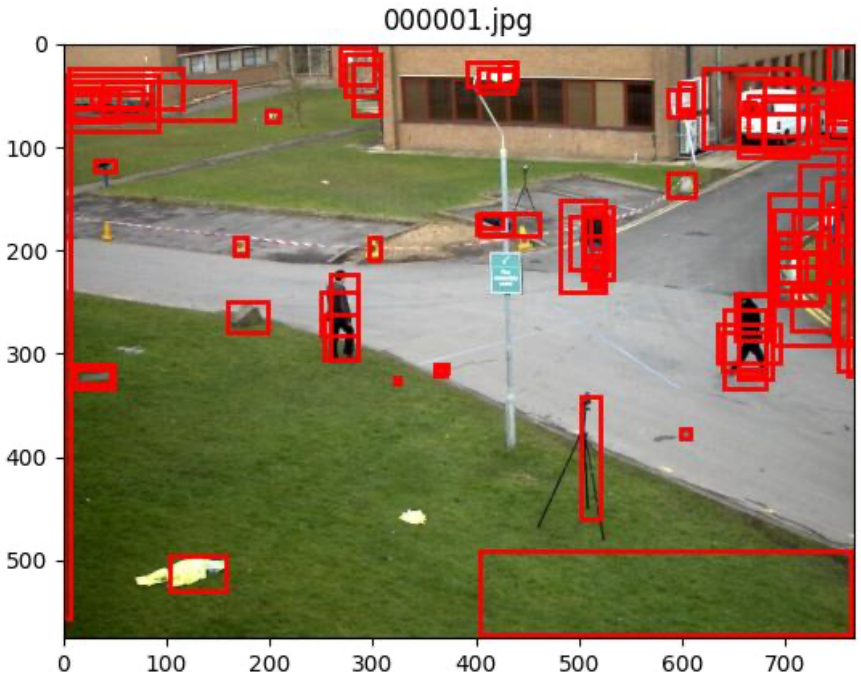
Due: 3/19/2024

Abstract:

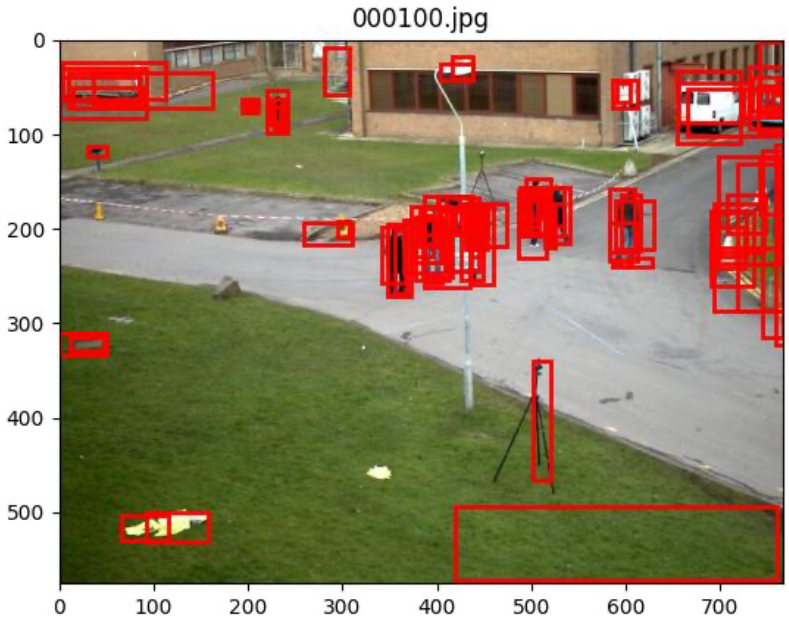
The goal of this assignment was to continue building familiarity with PyTorch. We were asked to load in the Faster R-CNN model that PyTorch provides while also utilizing the COCO_V1 weights from the FasterRCNN_ResNet50. With this we take the model and do direct inference upon the four defined frames 1, 100, 200, and 400. After this, Matplotlib will be utilized to visualize the results from the inference.

Experiments and Results:

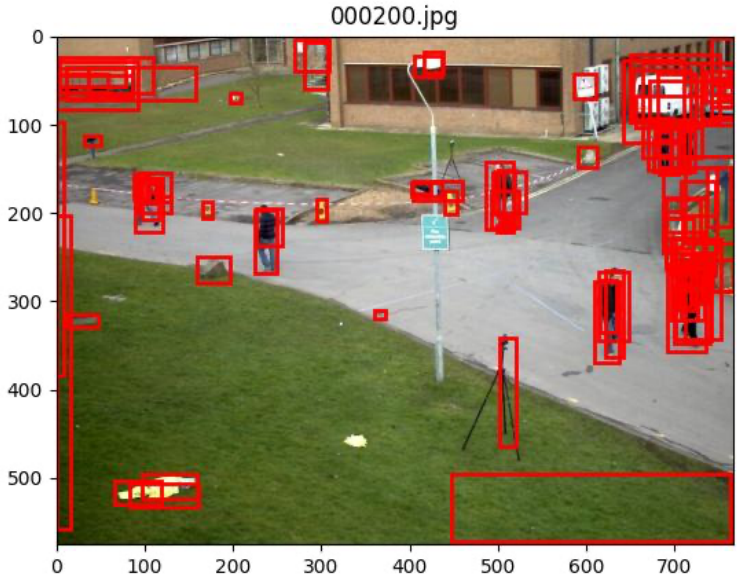
Frame 1 Inference Result



Frame 100 Inference Result



Frame 200 Inference Result



Frame 400 Inference Result

Conclusion:

To obtain this inference we utilized the Faster R-CNN with the COCO_V1 weights to attempt to separate pedestrians from the background. The only information that the model we tried to utilize had on our problem set was the fact that we were trying to separate 2 classes from each other (pedestrians vs the background). As we can see the results give lots of detections that are more than just the pedestrians. This is most likely because we are using a generalized model that was trained on a general dataset, the model we are using does not inherently know that we are trying to detect only the pedestrians. Because of this, if we ran the program I provided multiple times, each time it is run there are different detection results, with most of the detection results not being within the same realm (very different predictions). One way we could combat this is by using the COCO_V1 weights as a jump start in the training procedure. We can utilize the COCO_V1 weights as a starting point and train further on a specific pedestrian detection dataset. This would require less compute to get a successful pedestrian detector because we already started off with good weights. In the end, if we did the proposed strategy, we would probably have a much better pedestrian detector than what was tested above.

References:

- Libraries and tools: PyCharm, OpenCV, Matplotlib, PyTorch, TorchVision, Preview.
- CV2024_HW3B_DeepLearningObjectDetection.pdf