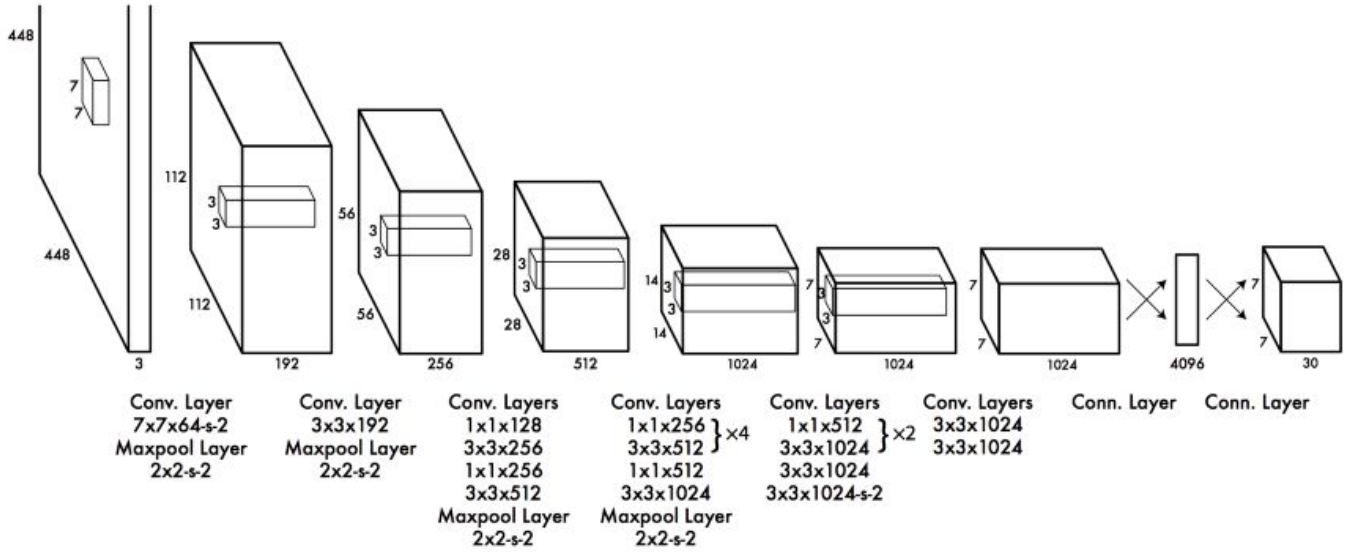
YOLO: A Detection Overview

Samuel Ordonia

Summary

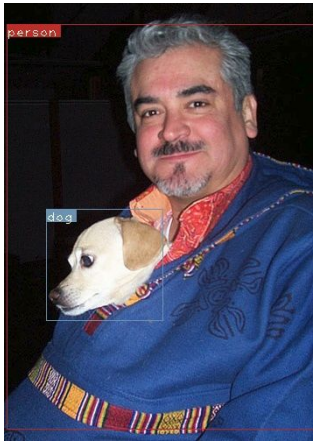
- YOLO = You Look Only Once
  - Real-time object detection using a FCN.
  - Chosen over the other camp of region-based detection (R-CNN) due to much reduced runtime per image.
- Setup
  - Pytorch implementation hacked together from other open source projects on YOLO.
  - Uses the v3 weights and architecture.
  - Runtime ~2 sec/image on an NVIDIA 1070 GPU.
- Detection Accuracy
  - Great at iconic shots.
  - Mostly effective even with non-iconic images, such as those from COCO.
  - Has trouble with high-density objects in an image.

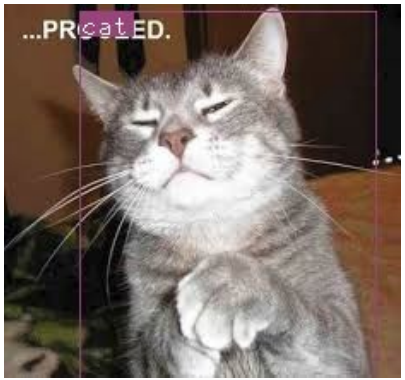
Architecture

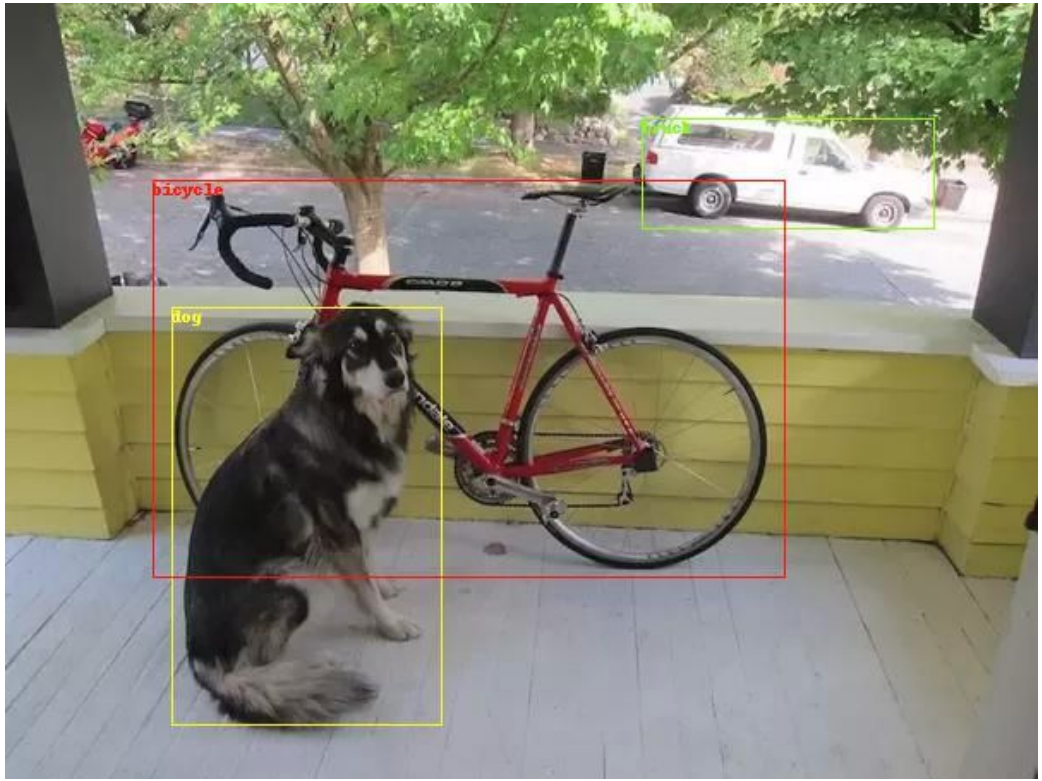


Detection Accuracy

Detection Accuracy

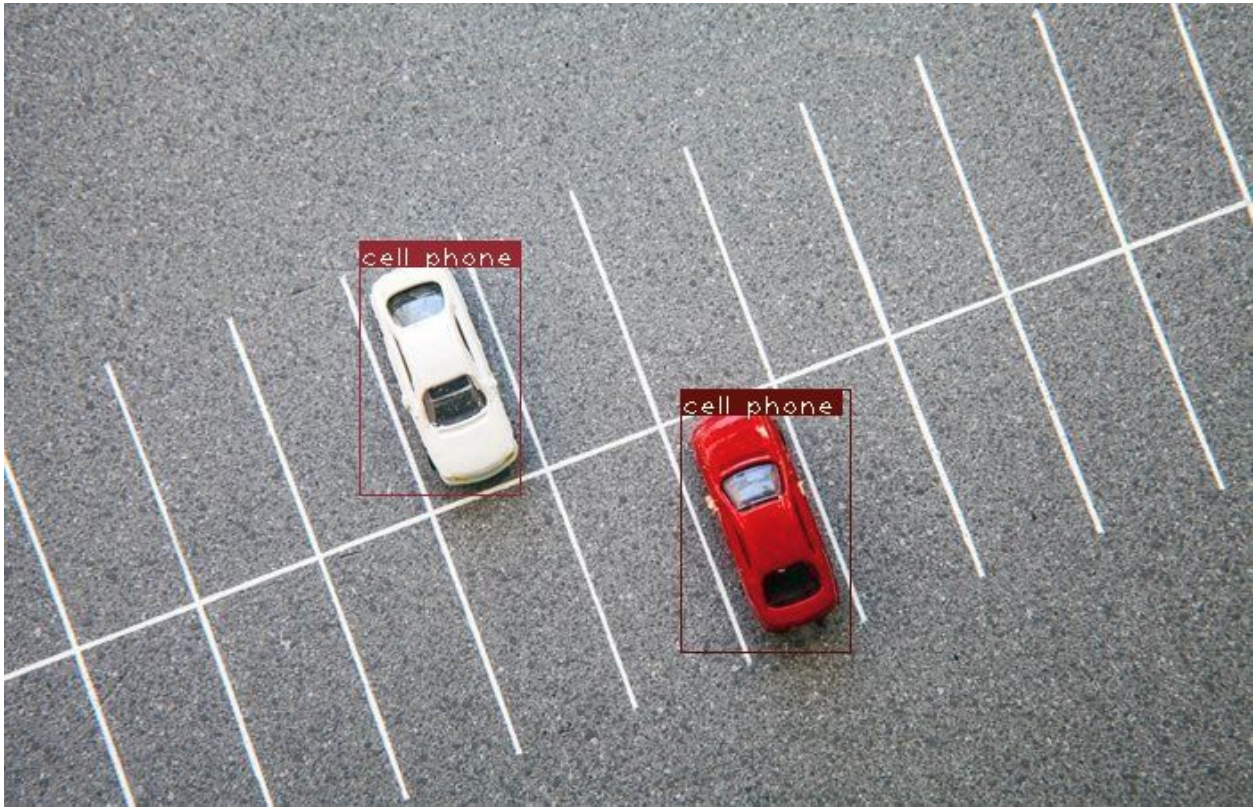






Detection Accuracy

wat



Highly Dense Image

Next Steps

- Docker + Open CV Setup
  - It looks nontrivial to debug and get Docker + open CV working with this implementation.
  - Rather than invest more time into infrastructure, just keep the Docker + open CV code separate. It makes sense to separate image processing from deep learning anyway.
- Training
  - Format data input the way YOLO expects it.
  - Get training working. This means running images + annotations through, followed by an update to the YOLO weights file.
  - Changing up the DNN architecture might mean changing up the weights used. Investigate this.
- Detection Improvements
  - Modify number of bounding boxes
  - Scan different tolerances
  - Understand the YOLO loss function some more