Pedestrian Detctor

By Jack Sheridan

GMIT
INSTITIÚID TEICNEOLAÍOCHTA NA GAILLIMHE-MAIGH EO
GALWAY-MAYO INSTITUTE OF TECHNOLOGY

What is a pedestrian detector?

A pedestrian detector is an object detector designed specifically to look for Pedestrians on roadways.

Why a pedestrian detector?

The goal with this project was to gain a greater knowledge of how machine learning models work and how to develop my own. I chose the subject of pedestrians as there is an abundance of images on the internet which I could use in my dataset e.g. PennFudan and PennPed datasets, image from which can be seen below

Image from the PennFudan dataset

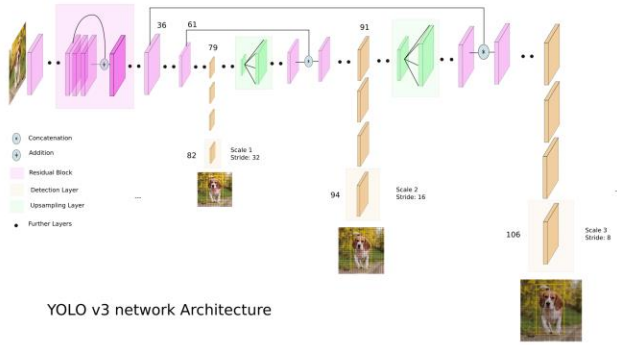
Image from the PennPed dataset

Description

The model is built using the Yolov3/ Darknet framework. This means the model is 106 layers in total (53 from base Darkenet plus additional 53 from Yolov3).

The model makes its predictions at 3 layers (82, 94, 106). These predictions happen at different scales (13x13, 26x26, 52x52). This is so the model can accuratly etect ddifferent sized objects.

Small objects are detected at layer 106 at a scale of 52x52.
medium objects are detected at layer 94 at a scale of 26x26.
Small objects are detected at layer 82 at a scale of 13x13.



YOLO v3 network Architecture

Here is what a model prediction looks like!

Technologies used

Google colab
It allows you to write and execute python code as well as text cells through the browser.

LableImg, used to label the images in our dataset.

YOLOv3, our object detecion algorith

Darknet, the framework YOLO is based on

YOLO (You Only Look Once)

YOLO is an object detection algorithm/ framework which uses classes and creates bounding boxes around individual objects for each object in the image, in one run of the algorithm. Hence the name.

Darknet

Darknet is an open-source neural network framework.
The main difference with Darknet and other models is that other models are applied at multiple locations and scales. Darknet applies the model once over the whole image. This network divides the image into regions and predicts bounding boxes and probabilities for each region. These bounding boxes are weighted by the predicted probabilities

Model Output

When training the model we are looking for a few things which would indicate that the model is performing well.

The first is "avg loss". We want this value to be as low as possible. Generally anything less than 5 is considered good.

The other thing to look for is IoU or Intersection over Union, which tells us how accurately the bounding boxes are being placed relative to the "ground truth" bounding box (the bounding box we placed ourselves in LabelIng). The higher the value the better.

Here is an example: