Real Time Object Detection

Using Deep Learning

INTRODUCTION

Humans can detect and identify objects present in an image. The human visual system is fast and accurate and can also perform complex tasks like identifying multiple objects and detect obstacles with little conscious thought. Now, we can easily train computers to detect and classify multiple objects within an image with high accuracy using object detection.

Efficient and accurate object detection has been an important topic in the advancement of computer vision systems. With the increase of machine learning and deep learning techniques, the accuracy for object detection has increased drastically.

Classification of images and estimation of the class and location of objects contained within the images is known as object detection. Object detection majorly involves image classification and localization.

OBJECTIVES

The goal of object detection is to achieve high accuracy with a real-time performance. It aims to identify what all objects are present in the picture stream and where they are located and to filter out the object of attention i.e. to give selective attention to the objects. It has the three basic objectives:

- Object detection, segmentation, location, and recognition
- Object tracking
- Different perspectives on the same scene or object-based image retrieval

APPROACH

Object detection can be done by machine learning and deep learning. But in this project, we will focus on deep learning.

You can use a variety of techniques to perform object detection. Popular deep learning–based approaches using convolutional neural networks (CNNs), such as R-CNN and YOLO v3, which automatically learn to detect objects within images.

1. You can select any one from two key approaches to get started with object detection using deep learning:

- **Create and train a custom object detector:** To train a custom object detector from scratch, you need to design a network architecture for learning the features for the objects of interest. You also need to compile a very large set of labelled data to train the CNN. The results of a custom object detector can be remarkable. You need to manually set up the layers and weights in the CNN, which requires a lot of time and training data.
- **Use a pretrained object detector:** Many object detection workflows using deep learning, an approach that enables you to start with a pretrained network and then modifies it for your application. This method can provide faster results because the object detectors have already been trained on thousands and millions of images.

  Whether you create a custom object detector or use a pretrained one, you will need to decide what type of object detection network you want to use: a two-stage network or a single-stage network.

- **Single Stage Networks:** In single-stage networks, such as YOLOv3, the CNN produces network predictions for regions across the entire image using anchor boxes, and the predictions are decoded to generate the final bounding boxes for the objects. Single-stage networks can be much faster than two-stage networks, but they may not reach the same level of accuracy, especially for scenes containing small objects.

- **Two Stage Networks:** The initial stage of two-stage networks, such as RCNN, identifies region proposals, or subsets of the image that might contain an object. The second stage classifies the objects within the region proposals. Two-stage networks can achieve very accurate object detection results; however, they are typically slower than single-stage networks.

METHODS

Faster-R-CNN:

Our object detection system, called Faster R-CNN, is composed of two modules. The first module is a deep fully convolutional network that proposes regions (RPN), and the second module is the Fast R-CNN detector that uses the proposed regions. The entire system is a single, unified network for object detection and the Fast R-CNN module uses selective search to generate region proposals.

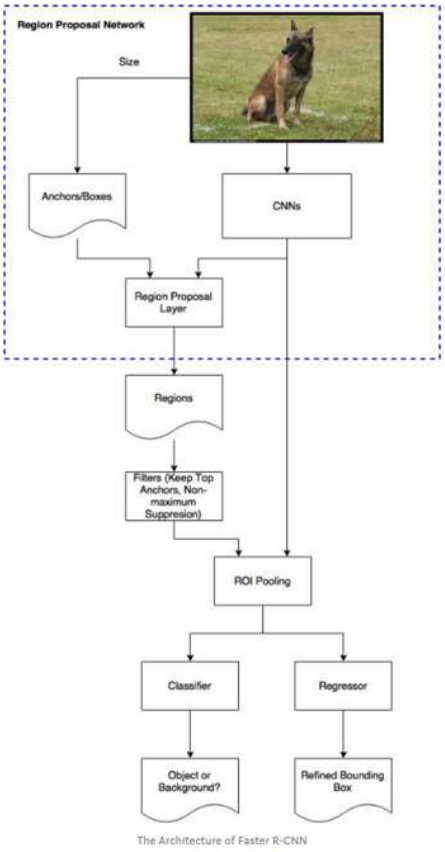
Anchors play an important role in Faster R-CNN. An anchor is a box. In the default configuration of Faster R-CNN, there are 9 anchors at a position of an image. A neat set of anchors may increase the speed as well as the accuracy.

The output of a region proposal network (RPN) is a bunch of boxes/proposals that will be examined by a classifier and regressor to eventually check the occurrence of objects.

The first step of training a classifier is make a training dataset. The training data is the anchors we get from the above process and the ground-truth boxes. The basic idea is that we want to label the anchors having the higher overlaps with ground-truth boxes as foreground, the ones with lower overlaps as background and use receptive field for reusing a trained network as the CNN's in the process.

If the process of labelling anchors is followed, you can also pick out the anchors based on the similar criteria for the regressor to refine.

Region of Interest (RoI) Pooling can simplify the problem by reducing the feature maps into the same size. Unlike Max-Pooling which has a fix size, ROI Pooling splits the input feature map into a fixed number (let's say k) of roughly equal regions, and then apply Max-Pooling on every region. Therefore, the output of ROI Pooling is always constant regardless the size of input.



The Architecture of Faster R-CNN

YOLOv3 :

YOLO v3 uses a variant of Darknet, which originally has 53 layers network trained on Imagenet. For the task of detection, 53 more layers are stacked onto it, giving us a 106 layer fully convolutional underlying architecture for YOLO v3. This is the reason behind the slowness of YOLO v3 compared to YOLO v2.

The most salient feature of v3 is that it makes detections at three different scales. The detection is done by applying 1 x 1 detection kernels on feature maps of three different sizes at three different places in the network.

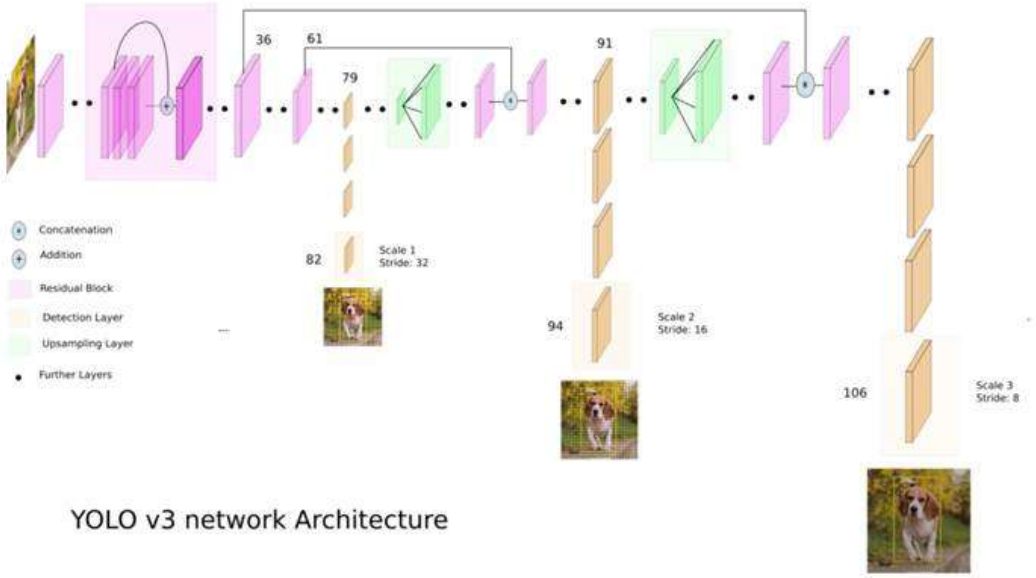
YOLO v3 makes prediction at three scales, which are precisely given by down sampling the dimensions of the input image by 32, 16 and 8 respectively.

YOLO v3, in total uses 9 anchor boxes. Three for each scale. If you're training YOLO on your own dataset, you should go about using K-Means clustering to generate 9 anchors. Then, arrange the anchors is descending order of a dimension. Assign the three biggest anchors for the first scale, the next three for the second scale, and the last three for the third.

YOLO v3 predicts 10x the number of boxes predicted by YOLO v2 i.e. YOLOv3 predicts more bounding boxes than YOLOv2.

Object confidence and class predictions in YOLO v3 are now predicted through logistic regression for changes in loss function.

YOLO v3 performs multilabel classification for objects detected in images instead of softmaxing the classes.



YOLO v3 network Architecture

APPLICATIONS

A well-known application of object detection is face detection, that is used in almost all the mobile cameras.

A more generalized application can be used in autonomous driving where a variety of objects need to be detected.

It has an important role to play in surveillance systems.

It can be used for tracking objects and thus can be used in robotics and medical applications.

Sports broadcasting will be utilizing this technology in instances such as detecting when a football team is about to make a touchdown and notifying fans via their mobile phone or at-home virtual reality setup in a highly creative way.