OBJECT DETECTION

Object detection is a branch of Computer Vision and Image Processing that pertains to detecting instances of various classes of objects in a digitally captured Image or Video.

To the naked eye, one single glance is enough to understand and realize the various objects and items in an image. Our capacity of visual perception is fast and precise, allowing us to perform complex tasks like driving with minimal effort. Super-fast algorithms that allows computers to detect objects would highly assist products like driving cars, without the need for special sensors and other assistive tools to fetch real-time data to humans.

Elementary object-detection algorithms were not as systematic as we want them to be today. To detect an object, the methodology involved implementing a classifier for the particular object, and estimate its closeness at several locations of the image. Many of the said algorithms used a sliding-window style to run the classifier at uniformly spaced regions over the entire image matrix. More recent trends include the use of R-CNNs that employ the use of region proposal methods to initially generate probable bounding boxes. The said classifier was limited to running over these boxes for recognition, rather than the entire image. Post-processing techniques for filtering and increasing the accuracy of the boxes, as well as the removal of duplicate boxes were included.

Most of the popular object-detection algorithms have one main drawback - Speed for real-time object detection. YOLO re-defines object detection as an uncomplicated regression model. The naming ‘You Only Look Once’ is administered literally, as the system only looks once at the image to predict the objects. The consolidated model has multiple advantages over earlier methods and is specifically optimized for detection performance. The decreased processing time can be attributed to the fact that object detection is defined as a regression problem, which negates the need for a complex pipeline.

OVERVIEW

First, a single CNN concurrently predicts various bounding boxes and the probability or confidence level of each object being present in the described box.

Secondly, YOLO aims at the image globally, rather than region-restricted techniques mentioned earlier. The entire image is understood during training and testing to encode the correspondent data of the objects, as well as their other visual attributes.

Thirdly, it generates generic rendition of objects and their boxes. This step also involves the usage of mathematical methods such as Non-Maximal suppression and Intersection-over-union to excise duplicate boxes.

YOLOv3

The third version of the YOLO algorithm was published on **[INSERT DATE].** Several tweaks were made from the previous versions for improvement in efficiency and accuracy, including a new classifier neural network.

YOLOv3 calculates the ‘score’ of each box using Logistic Regression, which ultimately denotes the probability that a particular object that the model aims to detect, exists inside the said box.

The new network design used by YOLOv3 was an improved version of the Darknet CNN used in previous versions of YOLO. The improved architecture uses consecutive 3x3 and 1x1 convolutional layers and proportionately larger number of filters.

**[INSERT ARCHITECTURE DIAGRAM OF DARKET-53]**

Training

Pre-trained weights are publicly available by the developers of YOLO. The network was trained on the COCO-object dataset..**[INSERT DETAILS OF TRAINING]**

**\*\*\*\*\*\*\*\*\*\*\*\*PULUTHIFY FROM HERE\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

To apprehend the YOLO algorithm, first we need to understand what is actually being predicted. Ultimately, our goal is to predict a category or class of an object and a bounding box denoting the location of the target object. Each bounding box can be described using four parameters:

Center of the box

Width

Height

Integer corresponding to the Class of the object

Along with that we predict a real number P, which denotes the possibility that there is an object in the bounding box. (confidence level)

After predicting the probabilities, the subsequent step is Non-max suppression. It enables the algorithm to delete the unnecessary boxes.

To resolve this issue Non-max suppression removes the boxes that are very close by preforming the IoU (Intersection over Union) with the one having the best confidence level among them.

It calculates the IoU value for all the bounding boxes respective to the one having the best confidence level. It then eliminates the bounding boxes whose IoU value is greater than a threshold. This means that that those two bounding boxes were overlaying the same object but the other one has a low confidence level for the same.

Once completed, algorithm finds the bounding box with subsequent maximum confidence level and repeats the same process. This is continued until we obtain unique bounding boxes.

**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***