What is Object Detection?

Object detection is a computer technology related to a collection of computer vision tasks that contribute towards identification of semantic objects of a certain class in digital images and videos.

Computer Vision tasks related to Object Detection

Object detection mechanism evolved around three different computer vision tasks:

1) **Image Classification**: Predict the type or class of an object in an image.
   - Input: An image with a single object, such as a photograph.
   - Output: A class label denoting the object type.

2) **Object Localization**: Locate the presence of objects in an image and indicate their location with a bounding box.
   - Input: An image with one or more objects, such as a photograph.
   - Output: One or more bounding boxes (e.g. defined by a point, width, and height).

3) **Object Detection**: Locate the presence of objects with a bounding box and types or classes of the located objects in an image.
   - Input: An image with one or more objects, such as a photograph.
   - Output: One or more bounding boxes (e.g. defined by a point, width, and height), and a class label for each bounding box.

What is YOLO model?

"You Only Look Once" (YOLO) is a state-of-the-art model because it achieves high accuracy while also being able to run in real-time. This algorithm “only looks once” at the image in the sense that it requires only one forward propagation pass through the network to make predictions. The approach involves a single neural network trained end to end that takes a photograph as input and predicts bounding boxes and class labels for each bounding box directly. The technique offers lower predictive accuracy (e.g. more localization errors), although operates at 45 frames per second and up to 155 frames per second for a speed-optimized version of the model.
A guide to the object detection exercise using YOLO model
Soumava Dey, Actuarial IT Analyst and Data Science Graduate Student
American International Group, Inc., Houston, Texas
University of Illinois at Urbana-Champaign, Champaign, IL. United States

YOLO Convolutional Network Diagram

In figure 3, it demonstrates the YOLO detection network which comprised of 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. The pertained convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

Inputs and Outputs of YOLO model

- The input is a batch of images, and each image has the shape (m, 608, 608, 3)
- The output is a list of bounding boxes along with the recognized classes. Each bounding box is represented by 6 numbers \((p_c, b_x, b_y, b_h, b_w, c)\) and if we expand c into an 80-dimensional vector, each bounding box is then represented by 85 numbers. The variable c represents the number of classes model will use for detection purpose (e.g. car, truck, traffic light etc.)

Since the model use 5 anchor boxes, each of the 19 x19 cells thus encodes information about 5 boxes. Anchor boxes are defined only by their width and height. See figure 4 for more details.

The YOLO architecture for my model:
**IMAGE** (m, 608, 608, 3) -> **DEEP CNN** -> **ENCODING** (m, 19, 19, 5, 85).
Probability Extraction

Class score: After creation of anchor boxes (of each cell) it is imperative to compute the following element-wise product and extract a probability that the box contains a certain class. See figure 5 for details.

The class score is \( \text{score}_c = p_c \times c_i \) i.e. the probability that there is an object \( p_c \) times object \( c_i \) times the probability that the object is a certain class \( c_i \).

Non-max suppression and IoU

After successful class prediction Each cell gives you 5 boxes. In total, the model predicts: \( 19 \times 19 \times 5 = 1805 \) boxes just by looking once at the image (one forward pass through the network) and it plotted boxes with high probability as shown in figure 6.

However, due to the large number of anchor boxes YOLO technique applies non-max suppression logic to perform following steps:

- Get rid of boxes with a low score (meaning, the box is not very confident about detecting a class; either due to the low probability of any object, or low probability of this particular class).
- Select only one box when several boxes overlap with each other and detect the same object.

In order to remove those overlapping boxes model will use a technique called Intersection over Union(IoU).This technique can be achieved using an evaluation metric called IoU which is used to measure the accuracy of an object detector on a particular dataset.

\( \text{IoU} \): This validation metric is useful to determine the ratio of intersection and union area between ground-truth box and prediction box.

---

Fig. 5 class score [Image source]

Fig. 6 [Image source]

Fig. 7 IoU formula [Image source]
Defining classes, anchors and image shape:
In order to detect class and anchor box information, the details of 80 classes and 5 boxes have been gathered in two files "coco_classes.txt" and "yolo_anchors.txt". The car detection dataset has 720x1280 images which have been pre-processed into 608x608 images.

For this exercise, a pre-trained Keras YOLO model has been used to perform object prediction.