Object Detection using YOLO

Shreeram

Aug 10 · 7 min read

This is one of the areas of computer vision that is in rise and working much better than previous years.

Before getting into Object Detection,we need to know about “Object Localisation”.What does this mean?

We already know what object classification is.Given an image, the algorithm will be able to classify it as a car,a person,a cycle etc.,This is called as classification.

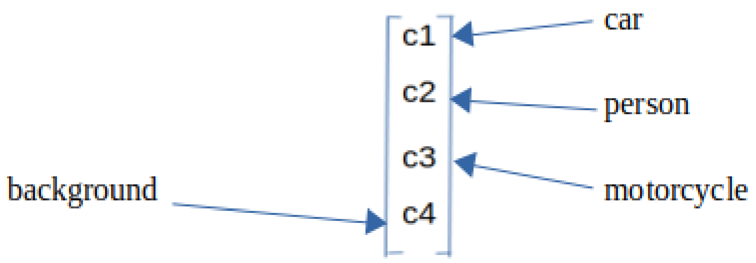
Classification with Localisation means,the algorithm not only classifies it as a car/cycle but also places a bounding box(red rectangular box around the position of the object present in the image)around it.

Here Localisation refers to,where in the picture is the object that the algorithm detected.

Now comes the Detection problem where there are multiple objects in the image.The algorithm must be able to detect them all and localise them all.It not only has to detect cars,cycles or any other vehicle,it also has to detect the pedestrians on the road.

Let us start with Classification with Localisation:

I hope you are already familiar with the image classification problem where you need to send an image to CONVNET with multiple layers which results in a vector of features.The classes means the possibilities of objects present in the image.Car,person,motorcycle,background etc., if there is nothing in the image then it is classified as “background”.



f c1 is 1 then it is a car,if c2 is 1 then it is a person,if there is nothing in the image then c4 will be equal to 0.

How can we localise the object in the image?

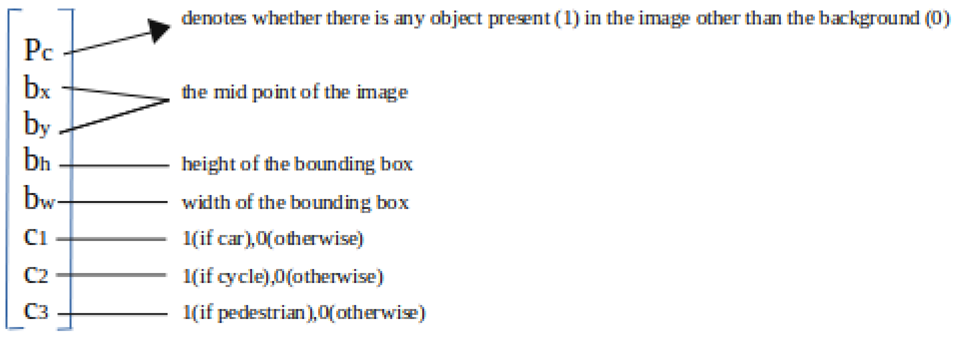
To do that we need to have few more features in the output vector of CONVNET(Convolutional Neural Network)that holds the values of a bounding box(b[x,by,bh,bw) of the detected object.

Let us use the notational convention that the upper left of the image is (0,0) and the lower right is (1,1) .

(b[x,by) is mid point of the image, bh is the height of the bounding box and bw is the width of the box.

Now the training set not only contains the object class labels to be classified but also the 4 additional numbers related to the bounding box.Now we can use supervised learning to make the algorithm output not just the class labels but also the four parameters to tell you where the bounding box of the object detected.

The target label (output) will be as follows:


Pc
bx
by
bh
bw
C1
C2
C3
denotes whether there is any object present (1) in the image other than the background (0)
the mid point of the image
height of the bounding box
width of the bounding box
1(if car),0(otherwise)
1(if cycle),0(otherwise)
1(if pedestrian),0(otherwise)

If there is no object present in the image, then the first variable itself turns up to be 0.When Pc is 0,then all the other variables are don’t cares.

Let us take an example and see what the output vector y looks like:

We have the classes:

1 — car

2 — pedestrian

3 — motorcycle

4 — background

If X is the input image

then the output will be:

$$y = \begin{bmatrix} 1 \\ b_x \\ b_y \\ b_h \\ b_w \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

pc = 1 indicate that there is an object present in the given image,the four parameters gives the information about the bounding box,and c1 is 1 because it is a car.

If there is no object in the given image,then the output will be

$$y = \begin{bmatrix} 0 \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \\ ? \end{bmatrix}$$

This (x,y) will be your training data.

Let us describe the Loss Function you use to train the Neural Network:

if we are using a squared loss function then:

$$L(\hat{y},y) = (\hat{y}_1 - y_1)^2 + (\hat{y}_2 - y_2)^2 + \ldots\ldots + (\hat{y}_8 - y_8)^2$$

- if Pc value is 1

$$L(\hat{y},y) = (\hat{y}_1 - y_1)^2$$

- if Pc value is 0

here y1 is the Pc in the y vector (since there are 8 values it goes till y8)

We can use Log Likelihood loss on the c1,c2,c3 and squared error on the bounding box coordinates and for Pc we can use Logistic Regression loss.

This is how we train the Neural Network to not just classify the object but also to localise it.

Bounding Box detection:

There are many methods to detect a bounding box but the most common one used is the sliding window.In this method we take the locations that are pre setted and run the classifier through it.

The boxes slides through all the positions from (0,0) corner to (1,1) corner as mentioned above and finds the place where most part of the object is sitting.

This method is not accurate as the window may not contain the image as a whole and also , since we have pre-setted the grid to be a square, it does not work for objects that are rectangular in shape.

The best way to increase the accuracy of this method is to use YOLO algorithm.

YOLO stands for You Only Look Once.

Let us take an image of 100 x 100 size. We place a grid on the image.This grid could be of any order.Here (3x3) is used for explanation purpose,but in practical we use much finer grids say (21x21) or even more.

What we have to do is to run the image classification and localisation algorithm that we have seen above and apply it on each of the 9 grids.

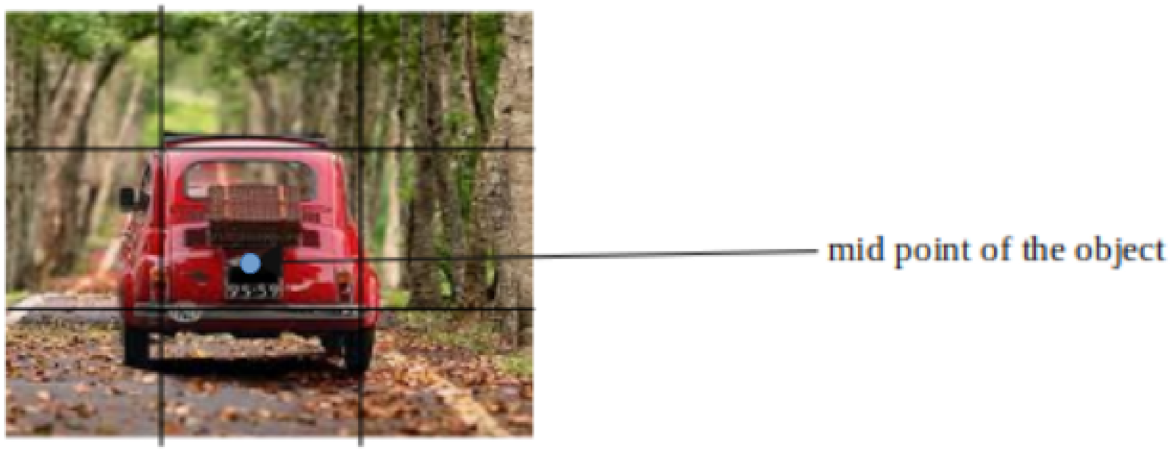
By applying the image classification and localisation algorithm on each grid, we get the output

$$y= \begin{bmatrix} P_c \\ b_x \\ b_y \\ b_h \\ b_w \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \qquad \text{for each grid cell}$$

Since we have divided the image into 9 grid cells we will have 9 such y outputs.

These outputs will be helpful in finding out where in the image is object/objects present.

As we have seen above that the mid point of the object is found by using the image classification and localisation algorithm, the grid cell containing the mid point is taken as the grid cell containing the object.



Even if a grid cell consists of small part of the object in it.It will not be taken into consideration.

Since we have divided the image into 3x3 grids,the output volume from this algorithm will be

3x3x8. Since each grid cell gives the output vector consisting of 8 values in it.

Therefore if we train the model with such input images x and the target values y ,we use this to train the neural network.The advantage of this algorithm is that the neural network will be able to find the accurate values of the bounding boxes which is what we want.

What if there are multiple objects(multiple mid points) in the same grid cell?

As already mentioned above,we always can increase the number of grids we use may be 19x19.

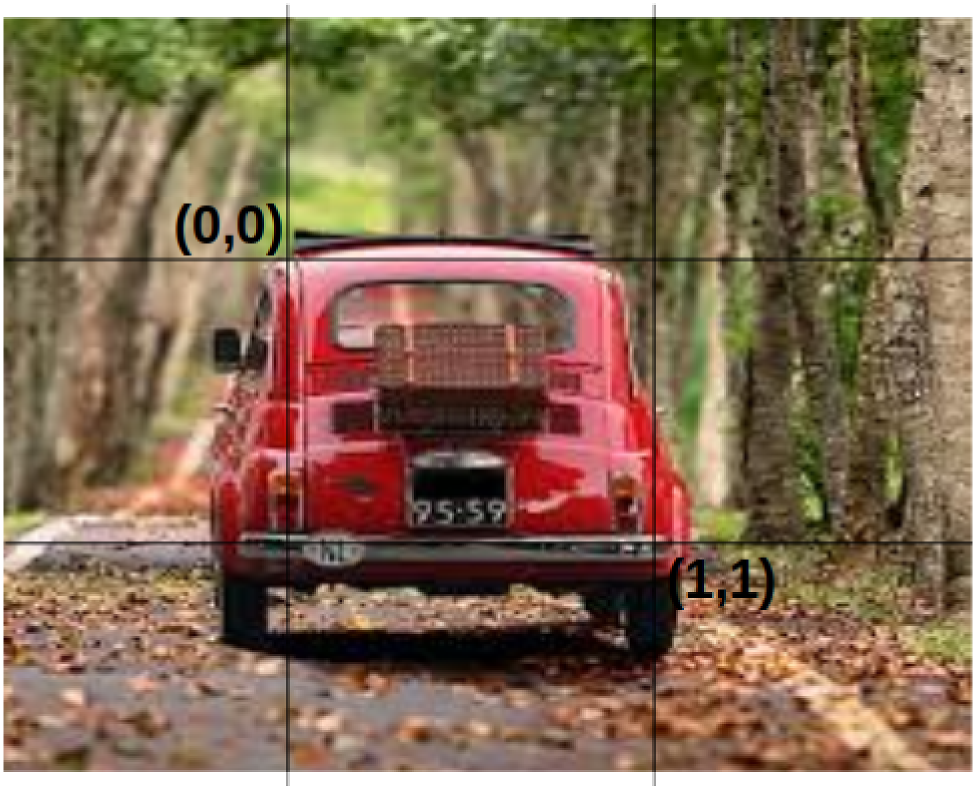
Therefore we end up with the output volume being 19x19x8 which also makes the grid much finer and reduces the chances of multiple objects getting assigned to same grid cell.

Here we are not running the algorithm 9 times since we have 9 grid cells (3x3) .We run the algorithm once and get the desired output.So this is a efficient algorithm which can be used for real time detection.

How does the values of the bounding box (bx,by,bh,bw) look like?

We have to know that each grid cell has (0,0) on its top left corner and (1,1) on the lower right corner.

The value of the mid point will always be between 0 and 1 But the values of the bounding boxes can be more than 1,because if the object is present in two grid cells the bounding box values can be more than 1.



Let us move into ***"Landmark Detection"***:

This is the basic building block for emotion detection.Suppose that the image is a face and we want to find some landmarks.We train the model by giving the input that denotes the different landmark points on the face with their respective x and y coordinates.

To see the uses of this landmark detection,if you have used the snapchat, you would wonder how the application is able to place a crown on your head,moustache on your face exactly in the position where it has to be!

Landmark detection is the answer.The model is trained to detect the landmarks of your fore head and places the crown using those coordinates.

If the model gets the coordinates of mouth of the person in the image,it could say if the person is smiling or frowning.

It is also helpful in pose detection,emotion detection and so on.Therefore Landmark Detection is the building block for any detection.

Machine Learning Object Detection Yolo Landmark Detection Object Localisation