# Object Detection Technologies

## A simplified explanation of YOLO class of algorithms

Edited by **Dr. DIBAKAR RAY**

**It is of the highest importance in the art of detection to be able to recognise out of a number of facts which are incidental and which are vital**

**- Arthur Conan Doyle**


**Dr. A.K. Hota**
Dy. Director General
& SIO
ak.hota@nic.in

**A.K. Somasekhar**
Sr. Technical Director
som@nic.in

**Shom C. Abraham**
Scientific Assistant - A
shom.abraham@nic.in


Object detection is the task of localizing as well as classifying objects of interest from an image or video. It covers a wide range of techniques, including image processing, pattern recognition, artificial intelligence, and machine learning. Object detection has a variety of uses, some of which are surveillance and security, traffic monitoring, video communication, image annotation, activity detection, face recognition, robot vision and animation.

### Classes of Algorithms

The three prominently used techniques in Object Detection are

- R-CNN and its variations like Fast R-CNN, Faster R-CNN, Mask R-CNN etc.
- Single Shot Detectors
- YOLO

### R-CNN

Girshik et al. first proposed R-CNN in 2013 wherein the system would make region proposals and then these regions would be passed to the CNN for classification and outputting bounding box. The problem with this approach is that it is painstakingly slow. Another version by the name Fast-RCNN was published by Girshik et al. in 2015 which used implementation of sliding windows convolution to identify all the proposed regions. However, it was still slow. It wasn't until the third paper came out by the name Faster R-CNN that this technique was used in practical applications. It replaced the use of an external algorithm like Selective Search with CNN to propose regions.

### YOLO

YOLO is the acronym for "You Only Look Once", whose first version appeared in 2016 by Redmon et al. Unlike previous approaches, the image is passed only once to the network rather than using a pipeline for region proposals, classification etc. and it simultaneously predicts the co-ordinates of the bounding box and the class of object. This increased the task's performance. Subsequently, here has been many versions of it namely YOLO v2, YOLO v3, YOLO v4 and YOLO v5 with the most recent one being YOLO v5 published in 2021. The concepts of YOLO v3 forms the basis for all subsequent works.

### YOLO v3

YOLO v3 uses only convolutional layers as the pooling layers are also simulated by convolutional layers. The training network's input is of the form (n, X,X,3), with n denoting the number of images , X denoting the width, height and 3 denoting the number of channels. The number X is chosen such that it is divisible by 32. YOLO v3 has 106 layers, with 53 CNN layers (Darknet-53) stacked on top of each other. The predictions are done at three different layers corresponding to strides 32, 16 and 8. For each cell of the image, we predict 3 bounding boxes at every scale. The bounding boxes are predicted as offsets to the prior boxes also known as anchors.

Common Objects in Context (COCO) is the dataset containing 80 classes of commonly occurring real life objects and is the standard dataset to test object detection algorithms. For the COCO dataset, YOLO v3 produces a tensor of the shape 3* (4+ 1+ 80), where 3 is for the number of the bounding boxes, 4 is for the offset location of bounding box, 1 is for the objectness score and 80 is for confidence probabilities of the number of classes. The offsets are given by $t_x$, $t_y$, $t_w$ and $t_n$ where $t_x$ and $t_y$ are the center co-ordinates and $t_w$, $t_n$ represents the width and height. The objectness score represents the IOU between the predicted box and any ground truth box.

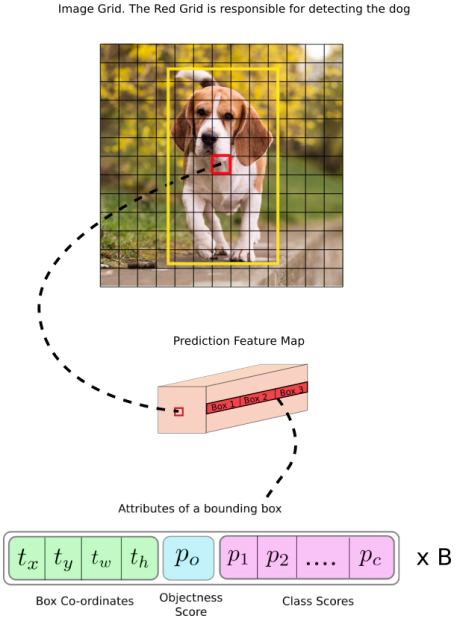



Image courtesy - http://medium.com

Each grid cell also predicts 80 conditional class probabilities, $Pr(Class_i | Object)$. These probabilities are conditioned on the grid cell to containing an object. At test time we multiply the conditional class probabilities and the individual box confidence predictions.

Given the anchor has width $p_w$, height $p_h$ and $(c_x, c_y)$ represents the coordinates of the center cell measured with respect to the top left corner of the image, the bounding box co-ordinates are given by:

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w * \exp(t_w)$$
$$b_h = p_h * \exp(t_h)$$

where σ represents the sigmoid function.

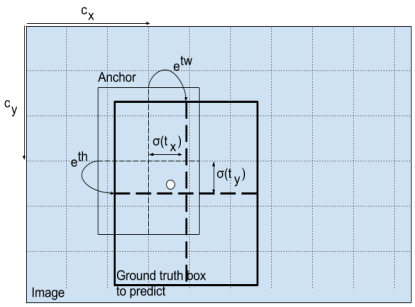



Image courtesy - http://christopher5106.github.io

The concept of IOU (Intersection over Union) is important here to do the post-processing tasks. IOU is simply a ratio.

$$IoU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

In the numerator we compute the area of overlap between the predicted bounding box and the ground-truth bounding box. The denominator is the area of union, or more simply, the area encompassed by both the predicted bounding box and the ground-truth bounding box.

The network may output several bounding boxes. Hence, we need to do some post processing tasks to identify the right bounding box. We apply the Non-max Suppression (NMS) algorithm which is discussed below:

**Step 1:** Select the box with the highest objectness score.

**Step 2:** Then, compare the overlap (intersection over union) of this box with other boxes.

**Step 3:** Remove the bounding boxes with overlap (intersection over union) >50%.

**Step 4:** Then, move to the next highest objectness score.

**Step 5:** Finally, repeat steps 2-4

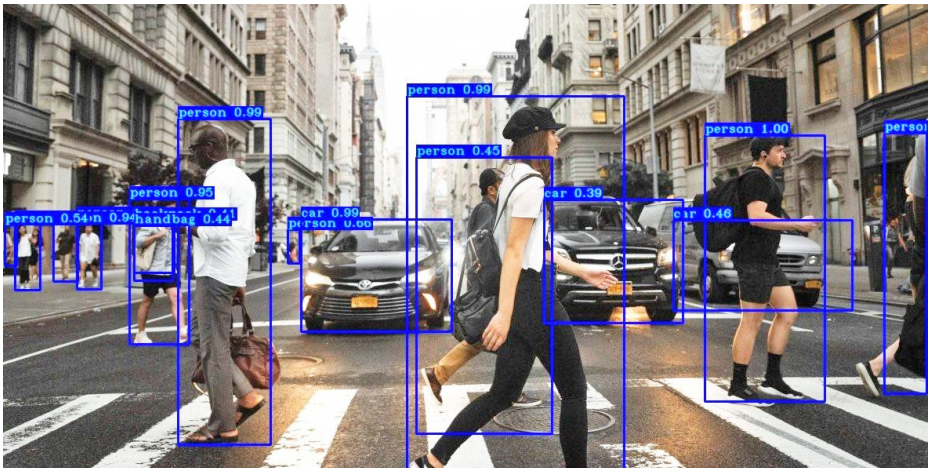
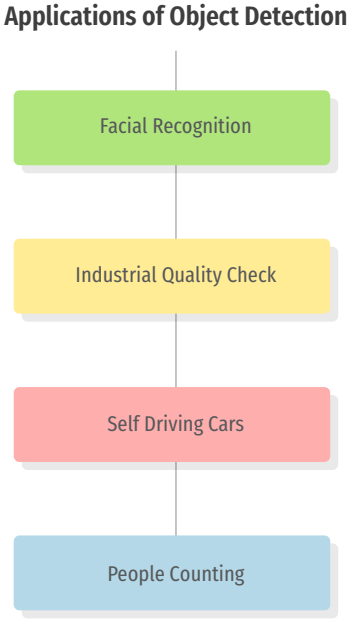



Image courtesy - http://medium.com




## How to use YOLO in projects

The official implementation of YOLO is available through Darknet (Neural Network implementation in C ) at https://pjreddie.com/darknet/yolov2/. Subsequently, the Python version of YOLO has been widely available using both TensorFlow and PyTorch Deep Neural Network libraries. For detecting objects from the COCO dataset, we can directly use the pre-trained weights and do prediction by passing the video frames to the model. However, to perform custom object detection we do training with the dataset of annotated images on YOLO model.

## Advantages of YOLO

- It's incredibly fast at the rate of 45 fps to 150 fps.
- Predictions are made from single network
- It outperforms other methods when generalizing from natural images

## Disadvantages of YOLO

- Struggles to detect small objects
- Comparatively low recall and more localization error
- YOLO imposes strong spatial constraints on bounding box predictions since each grid cell only predicts limited number boxes
- It struggles to generalize to objects in new or unusual aspect ratios or configurations

"A single neural network predicts bounding Boxes and class probabilities directly from full images in one evaluation. Since the whole detection pipeline is a single network, it can be optimized end-to-end directly on detection performance. "

— You Only Look Once: Unified, Real-Time Object Detection, 2015

## Examples of Applications of Object Detection in Government

- Counting Trucks at paddy storage centres
- To power real-time availability of parking information
- In Traffic surveillance for detecting over-speeding vehicles
- Teacher taking class attendance by taking a frame of group of students
- To identify objects from images acquired via satellites
- In surveillance cameras to detect suspicious events or gather intelligence
- Security of Government buildings can be strengthened by detecting intrusions

For further information, please contact:

**A. K. Somasekhar**
Sr. Technical Director
AD-2/14, Admin Block, Mahanadi Bhavan
Nava Raipur Atal Nagar
Chhattisgarh - 492002
Email: som@nic.in, Phone: 0771-2221238