Project Synopsis

Real-Time Object Detection in Machine Learning

Ishita Kheria

Abstract

The goal is to detect the objects using YOLO algorithm as unlike region based convolution neural networks (R-CNNs) that limits the classifier to a particular region, YOLO looks at the entire image and divides it into grids and then creates probable bounding boxes with a confidence score using convolution networks and finally eliminates the extra bounding boxes on the basis of threshold. It then predicts the class probabilities hence detecting the objects faster.

Introduction

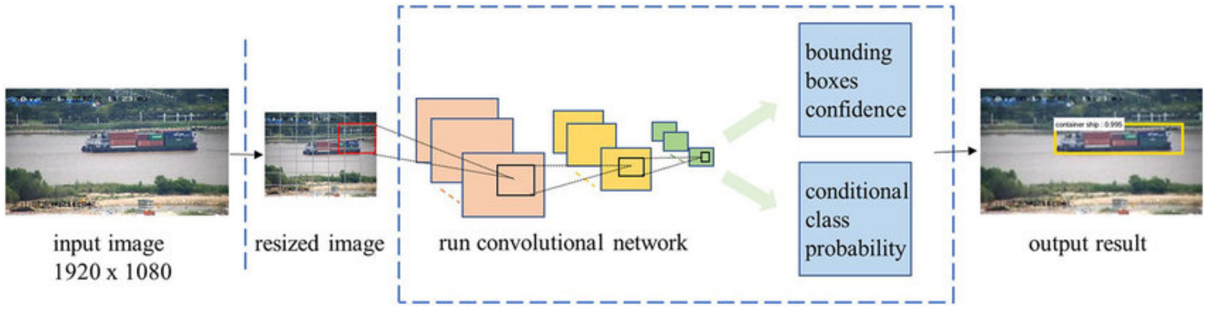
Object Detection is an important but challenging technology in Computer Vision and Image processing. It involves object localisation and image classification. Real world is complex contains many objects so we need to find 1. Difference between main obj from other objects and 2. Relations between the objects. One of it's real time applications is self-driving car. The difficulties faced are variations in orientation, lighting, background and occlusion that can result in completely different images of the very same object. YOLO is seen to make localisation errors but predicts less false positive in the background.

Literature Survey

Real-Time Object Detection with Yolo, by Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam [1]

OBJECT DETECTION AND IDENTIFICATION A Project Report, by Mukkamala Rohith Sri Sai, Sindhusha Rella,Sainagesh Veeravalli[3]

YOLO Architecture


input image
1920 x 1080
resized image
run convolutional network
bounding boxes confidence
conditional class probability
output result

YOLO Architecture

Architecture of YOLO is basically based on Convolutional Networks. The input image is divided into grids based on the complexity of the image then convolution network is applied on each grid to get the bounding boxes with thickness of the box proportional to the bounding box confidence score also called objectness and then the boxes not crossing the threshold set are removed.

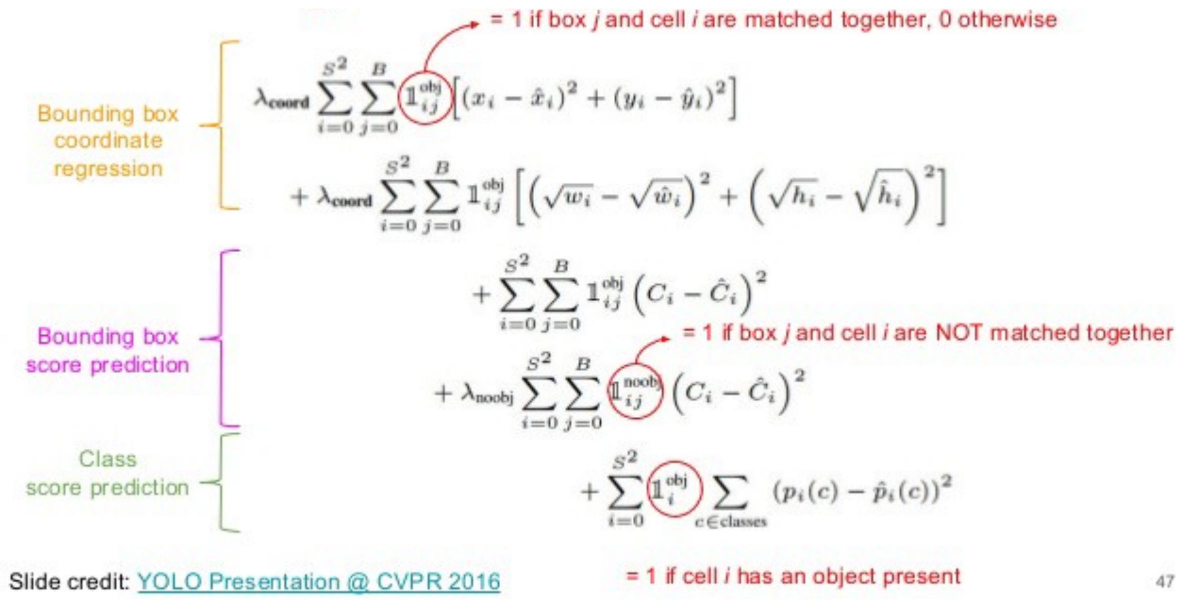
Both image classification and object localization techniques are applied for each grid of the image and each grid is assigned with a label.The labels of the gird without object are marked as zero.

If there is no proper object found in the grid, then the objectness and bounding box value of the grid will be zero or if there found an object in the grid then the objectness will be 1 and the bounding box value will be its corresponding bounding values of the found object.

If two or more grids contain the same object then the center point of the object is found and the grid which has that point is taken. Some of the methods used are Intersection over Union and Non-Max Suppression. In IoU, it will takes the actual and predicted bounding box value and calculates the IoU of two boxes by using the formulae, IoU = Area of Intersection / Area of Union.

If the value of IoU is more than or equal to our threshold value then it's a good prediction. The other method is Non-max suppression, in this, the high probability boxes are taken and the boxes with high IoU are suppressed. Repeat this until a box is selected and consider that as the bounding box for that object.

YOLO Hypothesis



= 1 if box *j* and cell *i* are matched together, 0 otherwise

Bounding box coordinate regression

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

Bounding box score prediction

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

= 1 if box *j* and cell *i* are NOT matched together

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

Class score prediction

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

Slide credit: YOLO Presentation @ CVPR 2016

= 1 if cell *i* has an object present

47

YOLO Hypothesis - loss function

For a single grid cell, the algorithm predicts multiple bounding boxes. To calculate the loss function we use only one bounding box for object responsibility. For selecting one among the bounding boxes we use the high IoU value. The box with high IoU will be responsible for the object. Various loss functions are:

- Classification loss function

-Localization loss function

- Confidence loss function.

Loss from predicting the bounding box center position (x, y), height(h) and width(w), confidence score for each bounding box predictor (C) , Classification Loss p(C) is essentially normal sum-squared error, except for the Iobj term. This term is used so classification error is not penalised when no object is present on the grid cell. The λ parameter weights parts of the loss function. This is necessary to increase model stability.[2]

Localisation loss:- The localization loss is the measure of errors in the predicted boundary box locations and the sizes. The box which is responsible for the object is only counted.Loss from predicting the bounding box center position (x, y). The function computes a sum over each bounding box prediction (j = 0...B) of each grid cell (i = 0...S^2). Iobj is defined as 1 if an object is present in grid cell i and the jth bounding box predictor is 'responsible' for it and 0 otherwise. X and y are the coordinates of the center of the predicted bounding box. Loss from predicting bounding box height and width. Because small deviations in large boxes should matter less than in small boxes, the square root of the height and width is predicted, rather than the height and width directly. Given by Bounding Box Coordinate Regression.

The Confidence loss:- it is given by Bounding box score.

Other Algorithms

HOGs , SIFT, CNN, RCNN, fast- RCNN, faster- RCNN, SSD.

Methods for Python

System Requirements:-

Install Python on your computer system

1. Install ImageAI and its dependencies like tensorflow, Numpy,OpenCV, etc.

Packages used-

Tensorflow, Numpy, SciPy, OpenCV, Pillow, Matplotlib, H5py, Keras, ImageAI.

2. Download the Object Detection model file (eg. Retinanet)[3]

Download Retinanet from:-

https://www.researchgate.net/deref/https%3A%2F%2Fgithub.com%2FOlafenwaMoses%2FImageAI%2Freleases%2Fdownload%2F1.0%2Fresnet50_coco_best_v2.0.1.h5

References:-

1. Real-Time Object Detection with Yolo, by Geethapriya. S, N. Duraimurugan, S.P. Chokkalingam International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-8, Issue-3S, February 2019
2. https://platform.ai/blog/page/7/new-state-of-the-art-in-logo-detection-using-yolov3-and-darknet/
3. https://www.researchgate.net/publication/337464355_OBJECT_DETECTION_AND_IDENTIFICATION_A_Project_Report
4. https://www.kdnuggets.com/2018/05/implement-yolo-v3-object-detector-pytorch-part-1.html