Real Time Object Detection With YOLO

Page No.
Date

Object detection is one of the classical problems in Computer Vision.

Recognize what the objects are inside a given image & also where they are in the image.

Detection is more complex problem than classification which can also recognize objects but doesn't tell you exactly where the object ~~dete~~ located in img.

Yolo is clever Neural Network for doing object detection in real-time.

How YOLO works?

You can take a classifier like VGGNet or Inception & turn it into an object detector by sliding a small window across the image. At each step run classifier to get prediction of what sort of object is inside the current window. Using sliding window gives several hundred or thousand predictions for that image.

This approach works but very slow.

Next approach is to first predict which parts of Image contain intresting information, → Region Proposals & run classifier only on this region, classifier do less work than sliding window, but still gets run many times over.

YOLO take completely diff. approach. It's not a traditional classifier that is repurposed to be object detector YOLO actually looks at the image just once hence Name- YOU ~~look~~ ONLY LOOK ONCE but in cleaver way.

Page No.
Date

Yolo divides up the image into a grid of 13 x 13 cells. Each of these cells is responsible for predicting 5 bounding boxes. A bounding box describes the rectangle that encloses an object.

YOLO also outputs a confidence score that tells us how certain it is that the predicted bounding box actually encloses same object.

for each bounding box, the cell also predict a class. The confidence score for the bounding box & class predictio are combined into one final score that tells us probability that this box contains a specific type of object.

since there are 13 x 13 = 169 grid cells & each cells predict 5 bounding boxes, 169 x 5 = 845 bounding boxes in total, among that pick best results.

Even through there were 845 seperate predictions, they where all made at the same time. - The Neural Network just ran once. & that's why YOLO is so powerful & fast.

The Architecture of YOLO is simple, its just a Convolution Neural Network.

This Neural Network only uses standard layer types: Convolution with a 3x3 kernel & max-Pooling with 2x2 kernel. There is No fully-connected layer in YOLOv2

Tiny version of Yolo that using has only 9 convolution layers & 6 pooling layers. - full YoloV2 model uses 3 times as many layers & has a slightly more complex shape.

Yolo runs 45 FPS (frame per seconds) - 22ms/ing.

Pascal voc dataset has 20 different classes
Microsoft coco has 80 object categories.

YOLO is written in Darknet, a custom deep learning framework from YOLO's author.

YOLO uses a regularization technique called batch Normalization after its convolutional layers.

The idea behind "batch Norm" is that Neural Network layer works best when the data is clean.

Batch Normalization does a similar kind of feature scaling for the data in beton layers. This technique helps NN to perform better because it stops the data from becoming progressively worse.

Batch Normalization usually happens after the convolution layer but before the activation function get applied. (also called Leaky - Relu in case of YOLO)

since both conv. & batch norm perform a linear transformation of the data, we can combine the batch Normalization layer's parameters with the weights for the convolutions.

convolution layer calculates:

If x - is the pixels in the input image

w - is the weights of the layer.

$$out[j] = x[i] * w[0] + x[i+1] * w[1] + x[i+2] * w[2] + \dots + x[i+k] * w[k] + b$$

dot product of input pixels with weights of conv. kernel + bias value b.

calculation performed by batch Normalization to output of that conv.

$$bn[j] = \frac{gamma * (out[j] - mean)}{sqrt(Variance)} + beta$$

It substract the mean from the output pixel, divides by variance, multiply by scaling factor

gamma & add beta.

This 4 parameters mean, variance, gamma, beta are what batch Normalization layer learns as the Nlw trained.

The convolution layers in Yolo don't actually use bias ie. $b = 0$

Biggest Advantages

- speed (45 frames/second)
- open source
- to train the Nlw in real world image & predictions on artwork was still acurate

Our system model detection as a regression problem. It divides the image into 5x5 grid & for each grid cell predicts B bounding boxes, confidence for those boxes, & C class probabilities. These predictions are enceded as $5 \times 5 \times (B * 5 + C)$ tensor.

Batch Normalization also helps regularize the model. with batch normalization we can remove dropout from the model without overfitting.

what is in this picture ? → object detection
what is it & where it is ? → object localization.

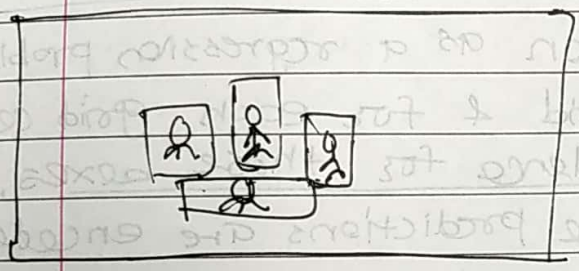
In object detection task we are interested in finding all object in the images & drawing so-called 'Bounding Boxes' find exact boundaries of our objects in the process called Instance Segmentation

DarkNet

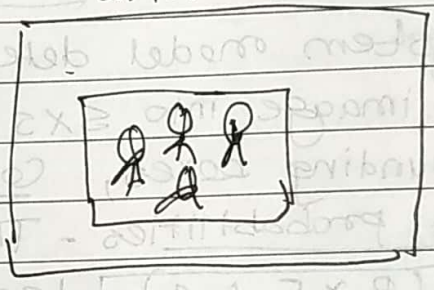
There are a few different implementation of YOLO algorithm on web, open saurce Neural Ntw framework called Darknet
Darknet was written in C, & Cudg technology

we removing boxes with low object probability & bounding boxes with the highest shared area in the process called non-max suppression

Before non-max supporession



after non_max_Supression



Yolo V1

May 2016

Yolo is a Network Inspired by GoogleNet It has 24 convolution layers working as feature Extractors & 2 dense layers for doing predictions. The architecture it works upon is called Darknet

YOLO V2 (Yolo 9000)

December 2016

- Improve the fact that ~~it~~ is not very good at detecting objects that are very near & tends to do a few mistakes in localisation.
- yolo V2 introduce New things called — anchor boxes (pre-determined set of boxes such that the Ntw moves from predicting the bounding boxes to predicting the offset from those)

smaller objects can be predicted better.

- Yolo 9000 a high speed, real time detection algorithm that can detect an over 9000 object categories

Page No.
Date

Yolo V3

- April 2018
- add small improvements, included the fact that bounding boxes get predicted at different scales. Darknet, is expanded in this version to have 53 convolutional layers.

Darknet

- framework to train NN., framework for training YOLO

Yolo 9000

- detect objects over 9000 classes using hierarchical classification with 9418 node wordTree. It combines samples from COCO & top 9000 classes from ImageNet.

> Softmax function to convert scores into probabilities that sum up to one.

Yolo V3 uses multi-label classification. Yolo V3 replace the softmax function with independent logistic classifiers to calculate the likeliness of the input belongs to specific labels.

Instead of using mean square error in calculating the classifier loss, Yolo V3 uses binary cross-entropy loss for each labels. This also reduces the computation complexity by avoiding the softmax function

DarkFlow

It is a NLW builder adapted from Darknet, it allows building TensorFlow networks from cfg file & loading pre trained weights.

- Darknet translation to run over Tensorflow

Data Annotation

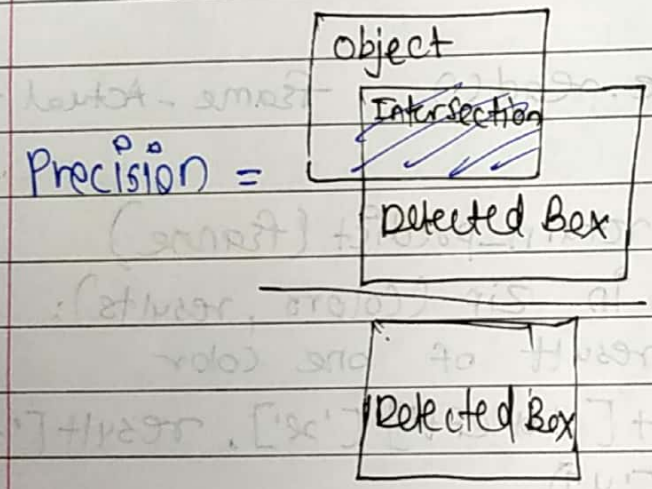
(create .txt file for each .jpg image →

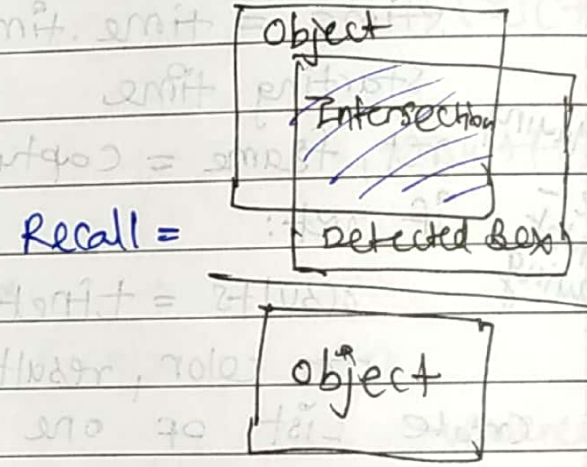
<object-class> <x> <y> <width> <height>

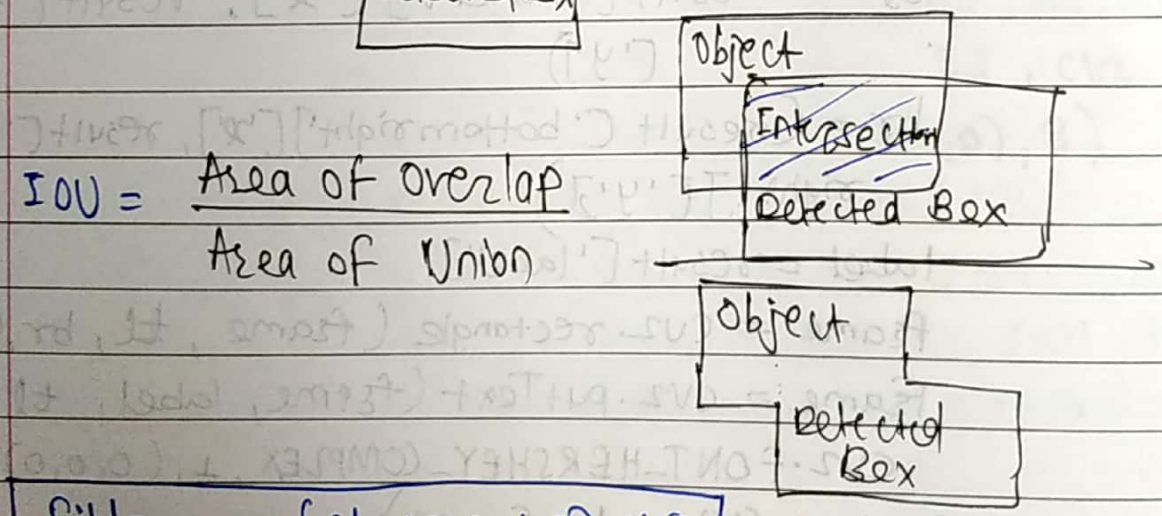
Int No → <x> = <absolute_x> / <image_width> or <height> / <absolute_h

<x> <y> - center of rectangle

<class-No> (<absolute_x> / <image_width>) (<absolute_y> / <image_height>) (<absolute_width> / <image_width>) (<absolute_height> / <image_height>)







Filters = (classes + 5) * 5

Intersection over Union (IOU):

~~for each rounding Box~~ anchor Box, calculate which object's bounding box has the highest overlap divided by non overlap.

If the highest IOU is greater than 50%, tell the anchor box that it should detect the object that gave the highest IOU.

If the highest IOU is greater than 40%, tell the NN that true detection is ambiguous & Not to learn from that example.

If highest IOU is less than 40%, anchor box should predict that there is no object.

(height)

Anchor Boxes are a set of predefined bounding boxes of a certain height & width.

yolo V3 Vs V2

V3 :- Better, not faster, stronger

V2 :- custom deep architecture darknet-19, Supplemented with 11 more layers of object detection (30-layer Architecture)
problem for small object detection

V3 - Originally has 53 layer NW trained on ImageNet & 53 more layers added = 106.

V3 make prediction at 3 scales, downsampling the diamension of ip image by 32, 16 & 8.

- Better for detecting small objects
- If uses 9 anchor Boxes, 3 for each scale

V2 - Squared Error

V3 - cross-entropy error terms.

objectConfidence & class predictions in yolo V3 are now predicted through logistic regression.

Page No.
Date

Yolo V1

- It uses Darknet framework which is trained on ImageNet - 1000 dataset, but has many limitations,
  It could not find small objects if they are appeared as clusters. (closeness of object)
- Architecture found difficulty in generalisation of objects If img is of other diamensions diff from trained img.
- major issue localization in the Input img

Yolo V2 (Yolo 9000)

- Better, faster & advanced to meet the faster R-CNN (object detection Algo.) which uses region proptional N/w).
- Changes of Yolo V2 from V1

Batch Normalization - it normalize the input layer by adjusting slightly, improve the stability of the NN.
By adding Batch Normalization to CNN layer MAP (Mean Average precision) has been improved by 2%.
helped the model for overfitting problem.

High Resolution - Input size increases from 224*224 to 448*448 also increase MAP by 4%.

Anchor Boxes - Yolo V2 does classification & predictions in a single framework. this anchor boxes responsible for predicting Bounding boxes

Smaller object - Yolo V2 divides the img into 13x13 grid cells which was smaller compared to V1, this enables V2 to Identify & localize smaller object

Multi-Scale Training - It is trained with random size images not on V1.

Page No.
Date

Darknet 19 - V2 uses darknet 19 with 19 conv & 5 max pooling & a softmax layer

Yolo V3

increamental improvement, quickly object detected.

Bounding Box predictions -
give score for objects for each bounding boxes. It uses logistic regression to predict objectiveness score.

Class prediction -
It uses logistic classifier for every class instead of softmax (Yolo V2), so have multi-label classification.

Feature peramid Nlw (FPN) -
3 predictions are made for every location the i/p img

Darknet - 53
Yolo V3 uses Darknet-53 Nlw for feature extractor, which has 53 conv layers.