Notes - You Only Look Once (YOLO)

Monday, December 24, 2018 6:24 PM

What is YOLO?
A new approach to object detection. It treats object detection as a regression problem to spatially separated bounding boxes and associated class probabilities.

What are the previous approaches before YOLO?
The approaches used previously have made use of the classifiers to perform detection.

What would the structure of YOLO would look like?
A single neural network predicts bounding boxes and class probabilities directly from full images in one evaluation. This network is optimized end-to-end directly on detection performance. The bigger YOLO can process images at the rate of 45 frames per second. The other network which is Fast YOLO can achieve a speed of 155 frames per second with the mAP more than double times the previous approaches. YOLO reframes the object detection as a single regression problem, straight from image pixels to bounding box coordinates and class probabilities. This algorithm looks at the image only once and predict the objects classes and locations.

What is the analogy between the YOLO and the real life functioning?
Humans glance at an image once and instantly know what objects are in the image, where they are, and how they interact. The YOLO algorithm also tries to implement the same.

What are the significant previous approaches?
1. Deformable Parts Models (DPM)
2. R-CNN

What is Deformable Parts Models (DPM) ?
It is an algorithm that is used before YOLO and it uses the sliding window approach where the classifier is run at evenly spaced locations over the entire image.

What is R-CNN?
They use region proposal methods which first generate the potential bounding boxes in an image and then run a classifier on those proposed boxes. After the classification, post-processing is used to refine the bounding box, eliminate duplicate detections and rescore the box based on other objects in the scene.

What are the advantages of YOLO?
Fast and accurate:
YOLO doesn't have a complex pipeline. With no batch processing, base network runs at 45 frames per second with no batch processing on a Titan X GPU and a fast version runs at more than 150 fps. The video can be processed in real-time with less than 25 milliseconds of latency. YOLO can also achieve more than twice the mean average precision of other real-time systems.

Global inference:
Unlike sliding window and region proposal methods, YOLO reasons globally about the image when making predictions. YOLO sees an entire image during the training and test time so it encodes contextual information about the classes as well their appearance. Fast R-CNN, a top detection method, mistakes background patches in an image for object because it is not able to see the larger context.

Generalizable representations:
After training on natural images, YOLO can also work on the art-work and out performs detection methods like DPM and R-CNN by a wide margin.

How the unified detection works in YOLO?
YOLO combines the detection and classification of objects in an image in a single network and this network can be trained end-to-end. It also reasons globally about the full image and all objects in the image.
YOLO divides the input image into a S x S grid. If the center of an object falls into a particular grid then that grid is responsible for detecting the object.
Each grid cell predicts 'B' bounding boxes and confidence scores for those boxes (to reflect confidence about the box containing an object and also the accuracy of the box dimensions enclosing the object). The confidence can be defined as follows:

$$\Pr(\text{Object}) * \hat{\text{IOU}}_{\text{pred}}^{\text{truth}}$$

If there is no object in that cell, then it has to be zero, else the confidence score has to be equal to intersection over union (IOU) between the predicted box and the ground truth.
Each bounding box has to make 5 predictions: x, y, w, h and confidence. x, y coordinates represent the center of the box relative to the bounds of the grid cell. The width (w) and height (h) are predicted relative to the whole image. The confidence represents the IOU between the predicted box and the ground truth box.
Each grid cell also predicts C Conditional probabilities, $Pr(Class_i|Object)$. One set of class probabilities are predicted regardless of number of boxes B.
During the test time the conditional class probabilities is multiplied with individual box confidence predictions to give class-specific confidence scores for each box. This score gives both the probability of that class appearing in the box and how well the predicted box fits the object.

$$\Pr(\text{Class}_i|\text{Object}) * \Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}} = \Pr(\text{Class}_i) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

The following figure shows the approach for the unified detection in detail.

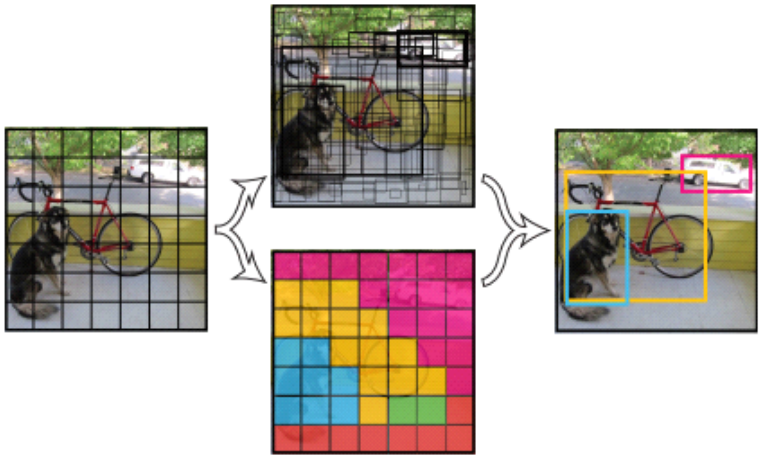

*Figure Description : The Model. Our system models detection as a regression problem. It divides the image into an even grid and simultaneously predicts bounding boxes, confidence in those boxes, and class probabilities. These predictions are encoded as an S × S × (B * 5 + C) tensor.*

What is the design approach for unified detection in YOLO?
The unified detection model mentioned above was implemented using the convolutional neural network and evaluated on the PASCAL VOC detection dataset.
Initial convolutional layers extract features from the image while the fully connected layers predict the output probabilities and coordinates. The network architecture resembles the GoogLeNet model for image classification. The network has 24 convolutional layers followed by two fully connected layers. Instead of the inception modules used by GoogLeNet 1 x 1 reduction layers followed by 3 x 3 convolutional layers are used. The entire network is represented in the diagram below.

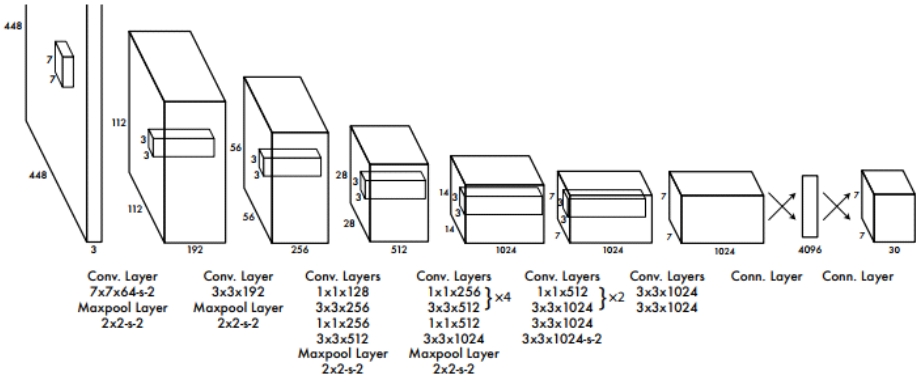

Figure Description: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1 × 1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224 × 224 input image) and then double the resolution for detection.

The paper also proposes a faster version of YOLO in which the number of convolutional layers are 9 instead of 24 and fewer filters in those layers. But all the training and testing parameters are same between both the versions.
For evaluating YOLO on PASCAL VOC Dataset following assumptions are considered for the network dimensions.
S = 7, B = 2, C = 20 (PASCAL VOC has 20 labelled classes). Hence the final prediction layer is a $7 \times 7 \times 30$ tensor.