0.1 YOLO

- *Algorithm:* YOLO
- *Input:* A training set, A full image
- *Complexity:* N/A
- *Data structure compatibility:* N/A
- *Common applications:* Object detection

YOLO

YOLO (You Only Look Once) is an unified, real-time object detection algorithm that does all predictions directly from full images, the bounding boxes, as well as related class probabilities in a single evaluating process.

Description

Introduction

YOLO regards object detection as a single regrssion problem, where input is image pixels, output is bounding box coordinates and class probabilities. YOLO is trained jointly based on the loss function which corresponds to the performance of detection. The overall process of YOLO algorithm can be simplified as

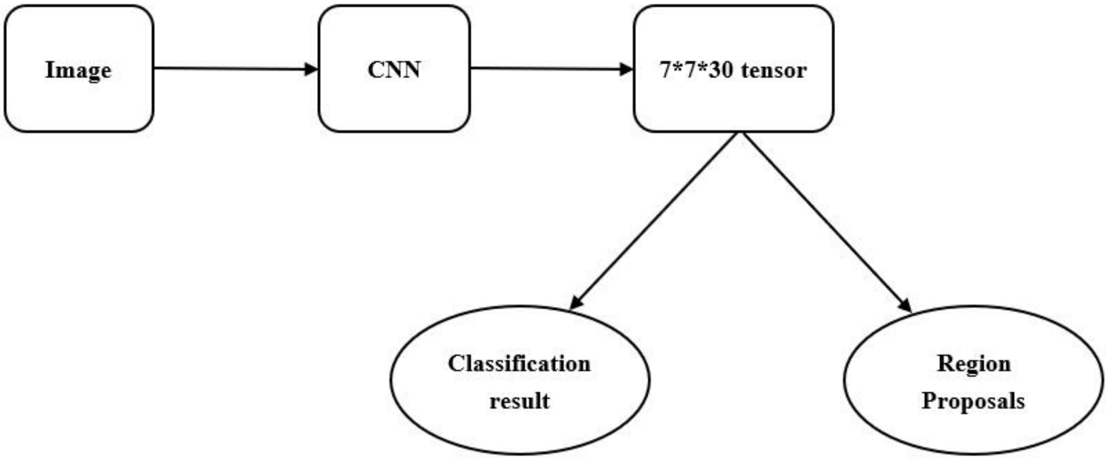

Figure 1: Simplification of YOLO

Unified Detection

YOLO sees the image as an $S \times S$ grid. If the center of an object in the image falls into a cell, the detection will be processed on such cell. Every cell evaluates B bounding boxes and respective confidence scores. The confidence score is defined as $\mathsf{Pr}(\text{ Object }) \cdot \mathsf{IOU}_{\text{pred}}^{\text{truth}}$. If the bounding box contains object,$\mathsf{Pr}(\text{ Object }) = 1$, else $\mathsf{Pr}(\text{ Object }) = 0$

There are 5 values for a bounding box, x, y, w, h and confidence. The (x, y) coordinates stand for the relative location of the bounds of the grid cells to the center of the box. w, h is the width and height of the bounding box.

For YOLO, $S = 7, B = 2, C = 20$, $S \times S \times (B \times 5 + C) = 1470$ dimensions. [1]

Network Design

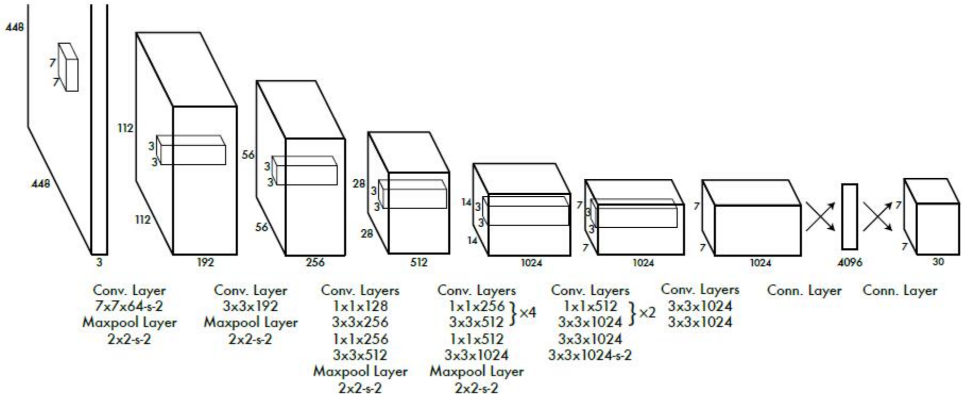

Figure 2: The network architecture for image classification

The detection newtork for classification has 24 convolutional layers, then 2 fully connected layers. The output of the overall network is $7 \times 7 \times 30$ tensor of predictions.

Error and Loss function

YOLO uses sum-squared error as its loss function to optimize the model parameters namely the error of $S \times S \times (B \times 5 + C) = 1470$ dimension vector between the network output and real image. The loss function is defined as

$$\mathsf{loss} = \sum_{i=0}^{S^2} \text{coord Error } + \text{ iou Error } + \text{ class Error}$$

Since corrdinate error and intersection-over-union error contributes differently with class error. $\lambda_{\text{coord}} = 5$ is added to remedy for corrdinate error.

When evaluating intersection-over-union error, cells containing and not containing objects contributes differently to loss function. If they share the same gradient, it pushes the confidence scores of cells that do not contain objects to zero, raising the gradient for cells do contain objects. $\lambda_{\text{noobj}} = 0.5$ is added. The final loss function is [1]

$$
\begin{aligned}
&\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \nVdash_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right] \\
&+\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \nVdash_{ij}^{\text{obj}} \left[(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2 \right] \\
&+\sum_{i=0}^{S^2} \sum_{j=0}^{B} \nVdash_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \\
&+\lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \nVdash_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \\
&+\sum_{i=0}^{S^2} \mathbb{I}_{i}^{\text{obj}} \sum_{c \in \text{ classes}} (p_i(c) - \hat{p}_i(c))^2
\end{aligned}
$$

where $\nVdash_{i}^{\mathsf{obj}}$ denotes if object appears in cell i, $\nVdash_{ij}^{\mathsf{obj}}$ denotes that the jth bounding box predictor in cell i is taken into consideration for that prediction.

Training

The training of YOLO model is seperated into two steps. In the pre-training process, the convolutional layers are trained based on the ImageNet 1000-class competition dataset. The first 20 convolutional layers, an average-pooling layer and a fully connected layer is used. Joseph Redmon and his team did the training for around one week, getting an outcome of 88% accuracy towards the ImageNet 2012 validation set. [1]

In the following step, the resolution of the input image is increased from 224×224 to 448×448 for fine-grained visual information. The last four convolutional layers and two fully connected layers are added and then initialized with random weights.

The linear activation function for all layers is

$$
\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}
$$

Then, optimize for sum-squared error in the output.

Limitations

YOLO model has its limitations as followed when in practical use. [1]

- YOLO does not perform well in detecting small objects that exist in the image in group, like flock of birds.
- YOLO does poorly in generalization to objects that is unusual.
- YOLO has low accuracy in objects localizations.

References.

[1] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. *You Only Look Once: Unified, Real-Time Object Detection*. 2015. arXiv: 1506.02640 [cs.CV] (cit. on pp. 2, 3).