Task: 08

YOLO: YOU ONLY LOOK ONCE -- V1

Architecture:

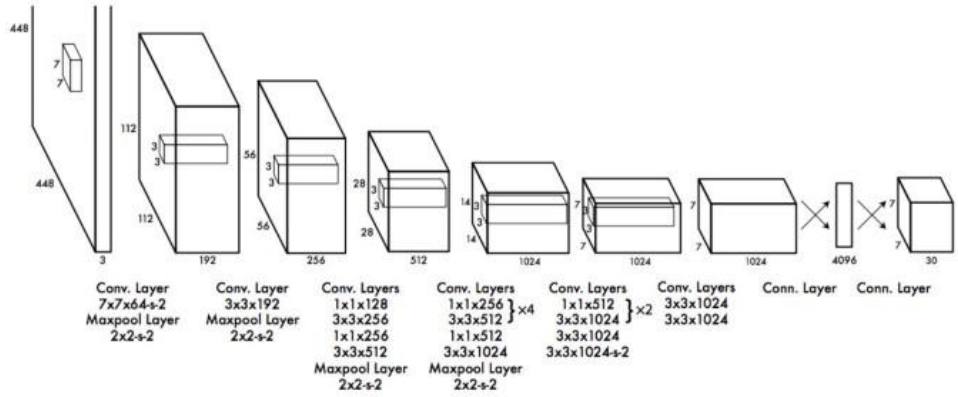

Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

This architecture consists of 24 convolutional layers followed by 2 fully connected layers. The layers are separated by their functionality in the following manner.

1. First 20 convolutional layers followed by an average pooling layer and a fully connected layer are pretrained on the ImageNet 1000 class classification dataset.
2. The pretraining for classification is performed on dataset with esolution 224*224.
3. The layers comprise of 1*1 reduction layers and 3*3 convolutional layers.
4. Last 4 convolutional layers followed by 2 fully connected layers are added to train the network for object detection.
5. Object detection requires more granular detail hence the resolution of the dataset is bumped to 448*448.
6. The final layer predicts the class probabilities and bounding boxes.

The final layer uses a linear activation whereas the other convolutional layers use leaky ReLU activation.

This input is 448*448 image and the output is the class prediction of the object enclosed in the bounding box.

Working:

1. Image splits in s*s grid cells. Each cell is responsible for predicting B bounding boxes. Here s may be anything like (7*7) or (19*19), B is the number of bounding boxes.
2. Each cell's bounding box generates vectors in size (5+ numbers of class). Total output will be (s*s + b*5 +c).
3. Bounding box vectors are predicted with consideration of the center of an object. So , there is a chance for many bounding boxes for one object. This problem is solved by non max suppression(NMS).
4. Simply NMS removes less confidence score bounding boxes with respect to one class. It repeats the operation for every class.
5. Final output has vastor of object has vector of object class and corresponding corresponding coordinates with high confidence score.

Loss function:

Loss is calculated for back propagation of the network to optimize learning parameters. There are three loss is calculated

1. Classification loss
2. Localization loss
3. Confidence loss

Regression loss

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

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

Confidence loss

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

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

Classification loss

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

In above figure i is total grid s*s and j is the number of bounding boxes in each grid cell.

Advantages of yolo V1:

1. Good speed compared to other algorithm
2. Less background mistakes.
3. Unified network architecture

Limitations :

1. Faces difficulties while predicting small objects.
2. More localization error compared to faster R-CNN.
3. If two objects have the same center point, yolo v1 faces difficulties for predicting two classes.