You Only Look Once: Unified, Real-Time Object Detection

Presentation By: Aritra Mukhopadhyay

What is Object Detection?

Object detection is a computer vision task that involves both classification and prediction of bounding boxes (localization) for a given object in an image.

Real-Time?

Real-time object detection is the task of doing object detection in real-time. This means that the object detection algorithm must be able to detect objects in an image in a very short amount of time (usually 30 frames per second).

In the next slide we will compare the speed and efficiency (measured in MAP) of YOLOv1 with other object detection algorithms.

Real-Time Algorithms

Algorithms	MAP	FPS
100Hz DPM	16.0	100
30Hz DPM	26.1	30
Fast YOLO	52.7	**155**
YOLO (YOLOv1)	**63.4**	45

Less than Real-Time Algorithms

Algorithms	MAP	FPS
Fastest DPM	30.4	15
R-CNN Minus R	53.5	6
Fast R-CNN	70.0	0.5
Faster R-CNN VGG-16	**73.2**	7
Faster R-CNN ZF	62.1	18
YOLO VGG-16	66.4	21

Algorithm

- Divide the image into a grid of SxS cells. (here S=7)
- The grid cell containing the center of the object is responsible for detecting that object.
- Each grid cell predicts B bounding boxes and confidence scores for those boxes.
- One grid cell can predict only one type of object.

Network Pipeline

- Preprocessed 448x448 coloured image is fed in.
- Starts with an array of CNN, Batch Norm, Activation and Pooling Layers.
- Finally ends with a couple of fully connected layers. And gives out a $S \times S \times (C + 5 * B)$ length vector for each image.

Defining the Constants

- $S \times S$ is the number of grid cells. (here S=7)
- B is the number of bounding boxes each grid cell predicts. (here B=2)
- C is the number of classes in the dataset. (here C=20)

Intuition

- Each grid predicts a $(20 + 5 * *2) = 30$ length vector.
- The first 20 elements are the one-hot encoded vector which represent the 20 classes in the dataset.
- The next B sets of 5 elements are the (x, y, w, h, confidence) values for each bounding box.
- $0 \leq x, y \leq 1$ and $0 \leq w, h$.

Loss Function

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

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

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

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

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

Limitations of YOLOv1

- Predicts only one bounding box per grid cell so struggles with objects that are close together.
- YOLOv1 struggles with small objects.
- Can't understand objects in unusual aspect ratios.
- Recognises coarse features only due to a lot of downsampling.

Thank You!

Aritra Mukhopadhyay