papergrid
Date: / /

Object detection.

(R-CNN, Fast RCNN, Fast RCNN)
Two shot

(Yolo & SSD)
Single shot

* R-CNN

i/p image
↓ selective search
extract region proposals (~2k)
↓ (~~warp~~ warped image regions)
Compute CNN feature
↓ (forward each region to CNN)
classify regions.
(classify regions with SVM)

* How it was (Linear Regr of bbox)

→ Multi task loss
→ Gradient descent
→ wt fn to the loss fn
→ other performance metric as hyper param is tricky to choose
→ weighted sum of two losses.

→ Disadvantage
Separate CNN for each box (slow)

Fast R-CNN

I/p image

↓

Forward whole image ~~to~~ through ~~conv~~ ConvNet

↓

Conv5 Feature map of image

Reagion of Interest from a proposal method → ROI pooling layer

selective ~~search~~ again

↓

FC

Linear + softmax classifier

Linear

Log loss + Smooth L1 loss

papergrid

Date: / /

Faster R-CNN

I/P Image → CNN

I/P Image → Anchor boxes (aspect ratio)

CNN → Region proposal layer

Anchor boxes → Region proposal layer

Region proposal layer
↓
proposed Region
↓
NMS → Spatial pooling

CNN → Spatial pooling

Spatial pooling → Regression (fitted bounding box)

Spatial pooling → classifier

→ Insert : Region proposal N/w to predict from features

Jointly train with 4 losses

→ RPN classify object / not object
→ " regress box co-ordinates
→ Final classification score
→ Final box co-ordinates

papergrid

Date: / /

Detection without Proposals

Yolo / SSD

→ Yolo

It divides the image into S×S grid and for each grid cell predict B bounding boxes, confidence for those boxes and C class probabilities.

These predictions are encoded as an S×S×(B∗5 +C) tensor.

→ You Only Look Once

→ Not traditional ~~cl~~ classifier that is repurposed to be an object ~~det~~ detector

→ Actually looks at the image just once but in clever way

→ Divide the image into a grid of say 13×13 cells

→ Each of these cells is responsible for predicting 5 bounding boxes

→ A bounding box describes the rectangle that encloses an object.

→ YOLO outputs a confidence ~~score~~ score that tells how good the shape of the box is.

papergrid

Date: / /

→ For each bounding box, the cell also predicts a class.

→ Yolo was trained on PASCAL VOC dataset of 20 different classes

→ The confidence score of bounding box and class prediction are combined into final score, probability that this bounding box contains a specific object.

* 13×13 = 169 grid cell
169 × 5 = 845 bounding boxes
most of them have low confidence score
Threshold of 30% or more ⇒ 3
i/p image 416×416 resized
13×13×125 tensor describing the bounding boxes for grid cells

*

papergrid

Date: / /

Yolo Bounding boxes

→ The i/p image is divided into S×S grid (S=7). If the center of an object falls into a grid cell, that grid cell is responsible for detecting that object

→ Each grid cell predict B bounding boxes (B=2) and confidence scores for those boxes.

These confidence scores reflect how confident the model is that box contains an object i.e.

any objects in the box, P(objects)

→ Each bounding box consists of 5 predictions x, y, w, h, and confidence

→ The (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 bounds of the grid cell

→ The width w & height h are predicted relative to the whole image

→ The confidence represents the IOU b/w the predicted box and the any ground truth box.

Single Shot Detection

→ By using SSD, we only need to take one single ~~sto~~ shot to detect multiple objects within the image.

→ Regional proposal network (RPN) based approaches such as R-CNN, Fast R-CNN series needs two shots, one for generating region proposals, one for detecting the object of each proposal

→ SSD is much faster

→ ~~2 loss fundn~~ loss function has 2 ter.
Lconf, Lloc

* A feature layer of size mxn (# of locations) with p channels

* for each location, we got k bounding boxes

* For each of the bounding box, we will compute c class ~~scor~~ scores and 4 offset relative to the original default bounding box shape.

* Thus, we got (c+4) kmn o/p