Object Detection using RetinaNet

Kimia Afshari, Brian Lee, Punnal Ismail Khan
UC Santa Barbara

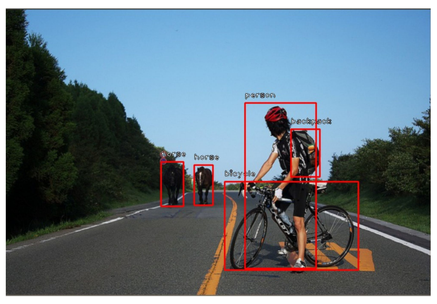
CS 291k Fall 22

1 Problem Definition

Dense Object Detection
Detecting categories and possible locations of small and dense objects in an image using a one-stage approach

Applications
Security and surveillance, automatic navigation and collision avoidance in self-driving cars and face recognition in device unlocking etc.

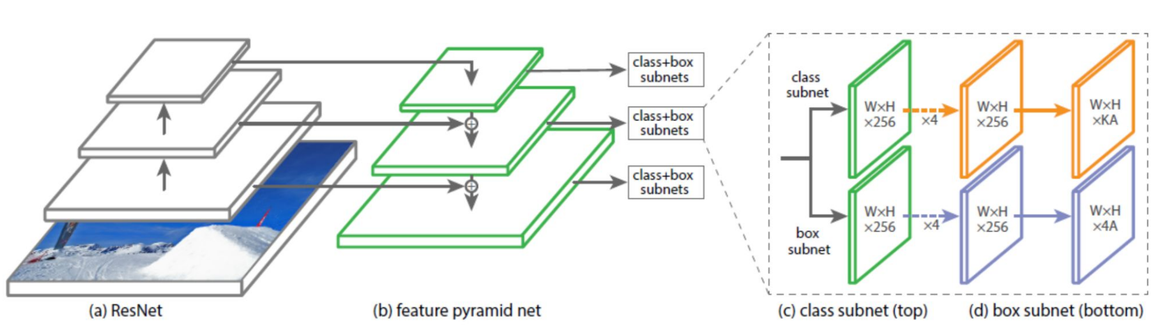
Example
Given an image, it outputs the possible bounding boxes and their corresponding labels for all the objects in the image



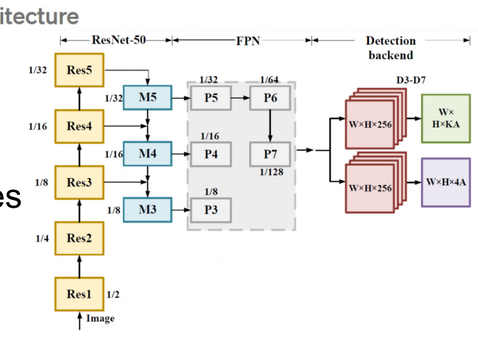
Limitations
- It is hard to detects objects in densely packed scenes.
- There is extreme foreground-background class imbalance problem in one-stage detectors.
- Class imbalance impedes one-stage detectors from achieving high accuracy.

2 RetinaNet

Architecture



RetinaNet Detector Architecture



Backbone
- ResNet-50, ResNet-101
- FPN(Feature Pyramid Net)
  - Detecting objects at different scales
  - Levels of *p3 - p7*
  - *C=256* for all levels

Anchors
- Proposed regions with different areas, scales and aspect ratios
- Each anchor has a vector of K class ids and a vector of 4 positions.

Classification Subnet
- Predicting the probability of presence of object at each position for each *A* anchors and *K* classes
- *4* of *3x3* conv layer with *C=256*

Box Regression Subnet
- Predicting relative offset between each *A* anchors and their groundtruth boxes
- *4* of *3x3* conv layer with *C=256*

Focal Loss
- Addresses class imbalance by focusing on the loss of hard samples.
- This is done by giving more weight to the hard negative examples and down-weighting the easy ones, resulting in the accuracy improvement.

$$FL = -(1 - P_t)^{\gamma} \log(P_t)$$

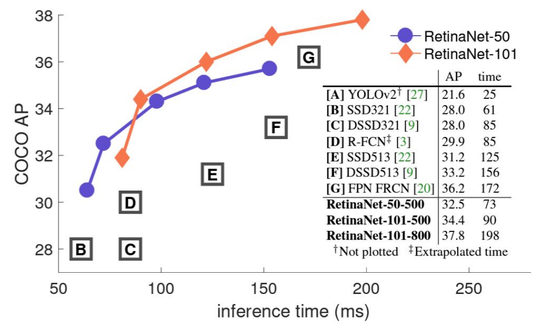
Pt is estimated probability for the class.
γ is a tunable focusing parameter γ ≥ 0.

Evaluation Metric
Based on COCO AP which evaluates detections by different parameters such as catIds, maxDets, area, and IoU.

3 Experiments

COCO 2014 dataset
- The dataset contains over 160k RGB images, which have been annotated by objects categories, locations, outlines etc.
- Dataset has 80 categories.
- We subsampled the dataset down to 4000 and 2000 to train and validate the model respectively.

Speed vs Accuracy
Due to the resources limits, we experimented with pre-trained ResNet-50-FPN although ResNet-101-FPN yields higher accuracy.



	AP	time
[A] YOLOv2† [27]	21.6	25
[B] SSD321 [22]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [22]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [20]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

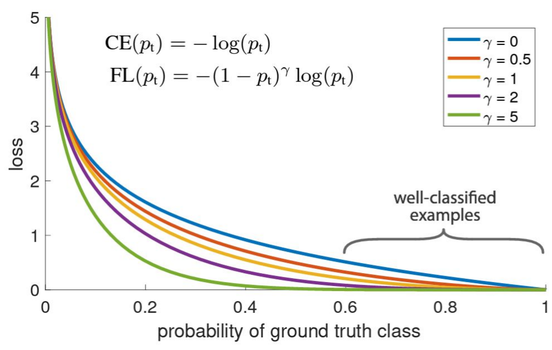
†Not plotted ‡Extrapolated time

Anchor Density
To cover boxes of various scales and aspect ratios, we experimented with multiple anchors of 3 scales (2**k/3, for k<3) and 3 aspect ratios [0.5, 1, 2] and improved the AP by 3.7 points (34.0).

#sc	#ar	AP	AP_{50}	AP_{75}
1	1	30.3	49.0	31.8
2	1	31.9	50.0	34.0
3	1	31.8	49.4	33.7
1	3	32.4	52.3	33.9
2	3	**34.2**	**53.1**	**36.5**
3	3	34.0	52.5	**36.5**

(c) **Varying anchor scales and aspects**

Loss
The approach tried different loss functions to see which one outperforms. Among CE, α-balanced CE, OHEM, and Focal loss, the focal loss with γ = 2 yields a 2.9 AP improvement over the α-balanced CE loss.



γ	α	AP	AP_{50}	AP_{75}
0	.75	31.1	49.4	33.0
0.1	.75	31.4	49.9	33.1
0.2	.75	31.9	50.7	33.4
0.5	.50	32.9	51.7	35.2
1.0	.25	33.7	52.0	36.2
2.0	.25	**34.0**	**52.5**	**36.5**
5.0	.25	32.2	49.6	34.8

(b) **Varying γ for FL** (w. optimal α)

4 Comparison to State of the Art

One-stage methods
RetinaNet achieves 5.9 point AP higher than the closest competitor (39.1 vs. 33.2).

Two-stage methods
This approach achieves 2.3 point above the top-performing Faster R-CNN model.

	backbone	AP	AP_{50}	AP_{75}
Two-stage methods				
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2
One-stage methods				
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**

5 References

Use the QR code below to find more details about the Project.