nodeflux

Extending Vision Beyond Imagination

Week 3 - YOLOv3

Martin Dominikus
Muhammad Rizky M

DEFINITION

- Computer detects instances of semantic objects of a certain class, looking for higher level understanding
- CV is not Image Processing
- Input: digital image, video

METHODS

- Generative models
- Discriminative models
- Computational methods

METHODS (CONTINUED)

Machine Learning approaches:

- Viola-Jones (Haar features) → face detection
- SIFT
- HOG features

Deep Learning approaches:

- Region based detection (R-CNN families)
- SSD
- YOLO

APPLICATIONS

- Robotics
- Medical image analysis
- Surveillance
- Human computer interaction
- Biometric
- Image retrieval
- Face detection
- Pedestrian detection
- Tracking movement

nodeflux

Extending Vision Beyond Imagination

Section 2 –
YOU ONLY LOOK ONCE
[The algorithm]

OVERVIEW

- YOLO & YOLOv3: Intro
- Training
  - ➢ Network Design
  - ➢ Loss Function
- Inference
  - ➢ Anchor
  - ➢ Non-maximum suppression
- References

Joseph Redmon

Ross Girshick

Santosh Divvala

Ali Farhadi

YOLO

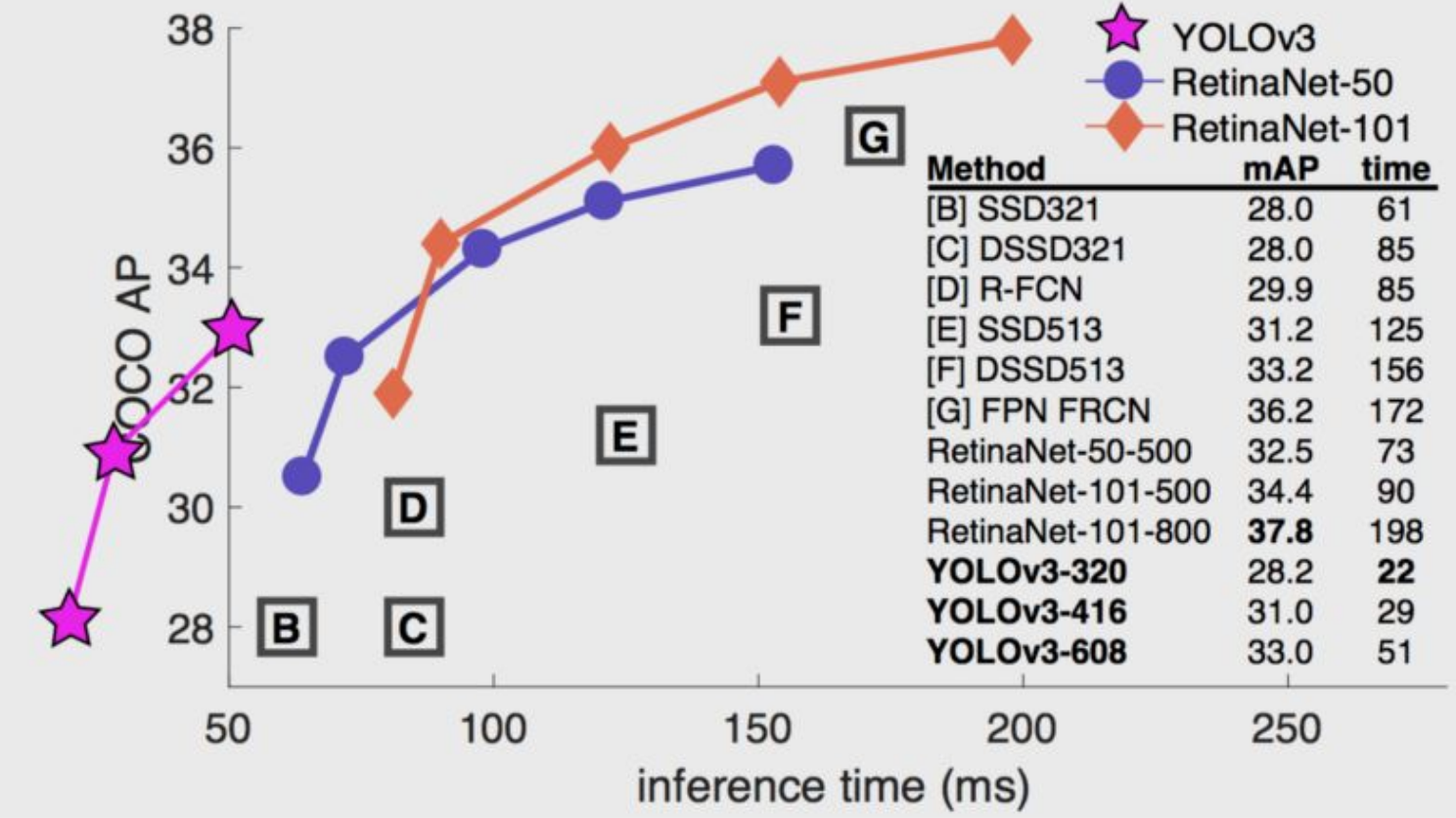
- Object detection algorithm
- A regression problem to spatially separated bounding boxes and associated class probabilities
- Single network, one evaluation, optimized end-to-end

YOLOv3

- Class Prediction
  - Softmax function → logistic classifiers
- Bounding box prediction & cost function calculation
- Feature Pyramid Networks (FPN) like Feature Pyramid

Why use YOLOv3?

- Very fast (~45 fps on Titan X) without sacrifice too much accuracy
- BFLOP
- Reasons globally



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

nodeflux

Extending Vision Beyond Imagination

Section 3 – YOLO: Training Network Design

Darknet-53

Feature Extractor

	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

YOLO

(x, y, w, h) + confidence score (objectness+box)

(x, y, w, h) + confidence score (objectness+box)

car: 0.05
person: 0.10
...
truck : 0.02

...

cell(1, 1)

...

...

cell(S, S)

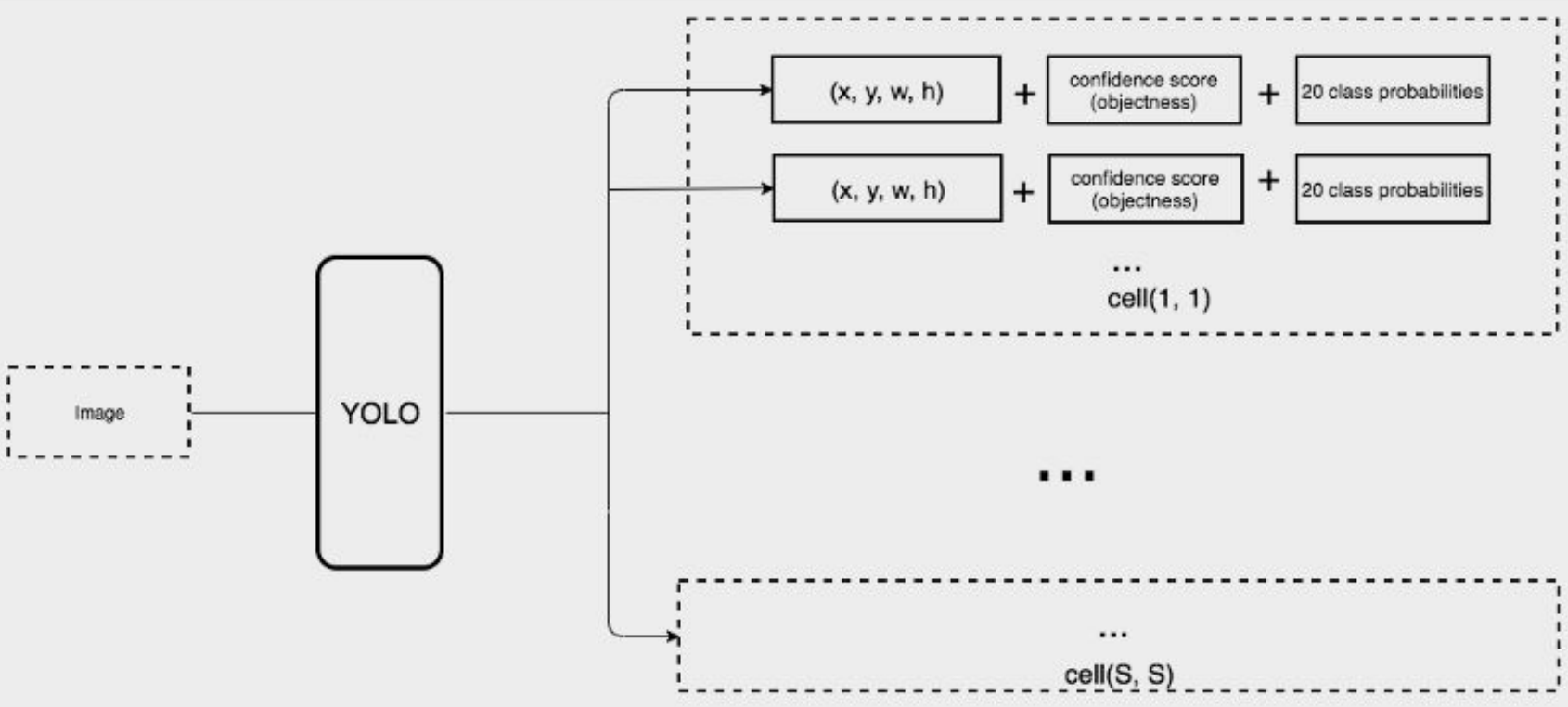
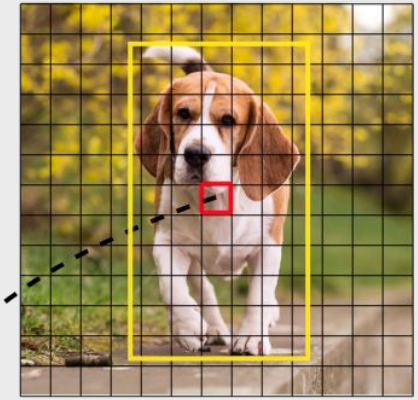
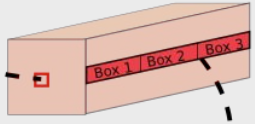


Image
YOLO
(x, y, w, h)
+
confidence score (objectness)
+
20 class probabilities
(x, y, w, h)
+
confidence score (objectness)
+
20 class probabilities
...
cell(1, 1)
...
...
cell(S, S)

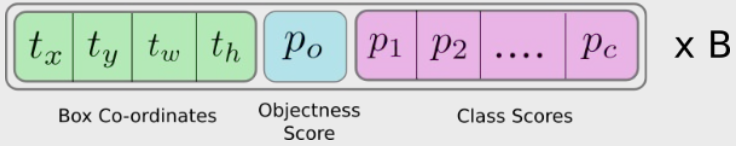
Image Grid. The Red Grid is responsible for detecting the dog



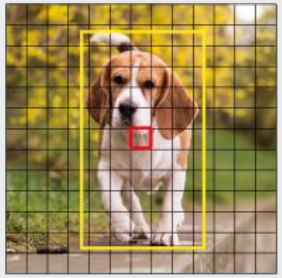
Prediction Feature Map



Attributes of a bounding box



Prediction Feature Maps at different Scales



13 x 13

26 x 26

52 x 52

Training: Loss Function

YOLOv2

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

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

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

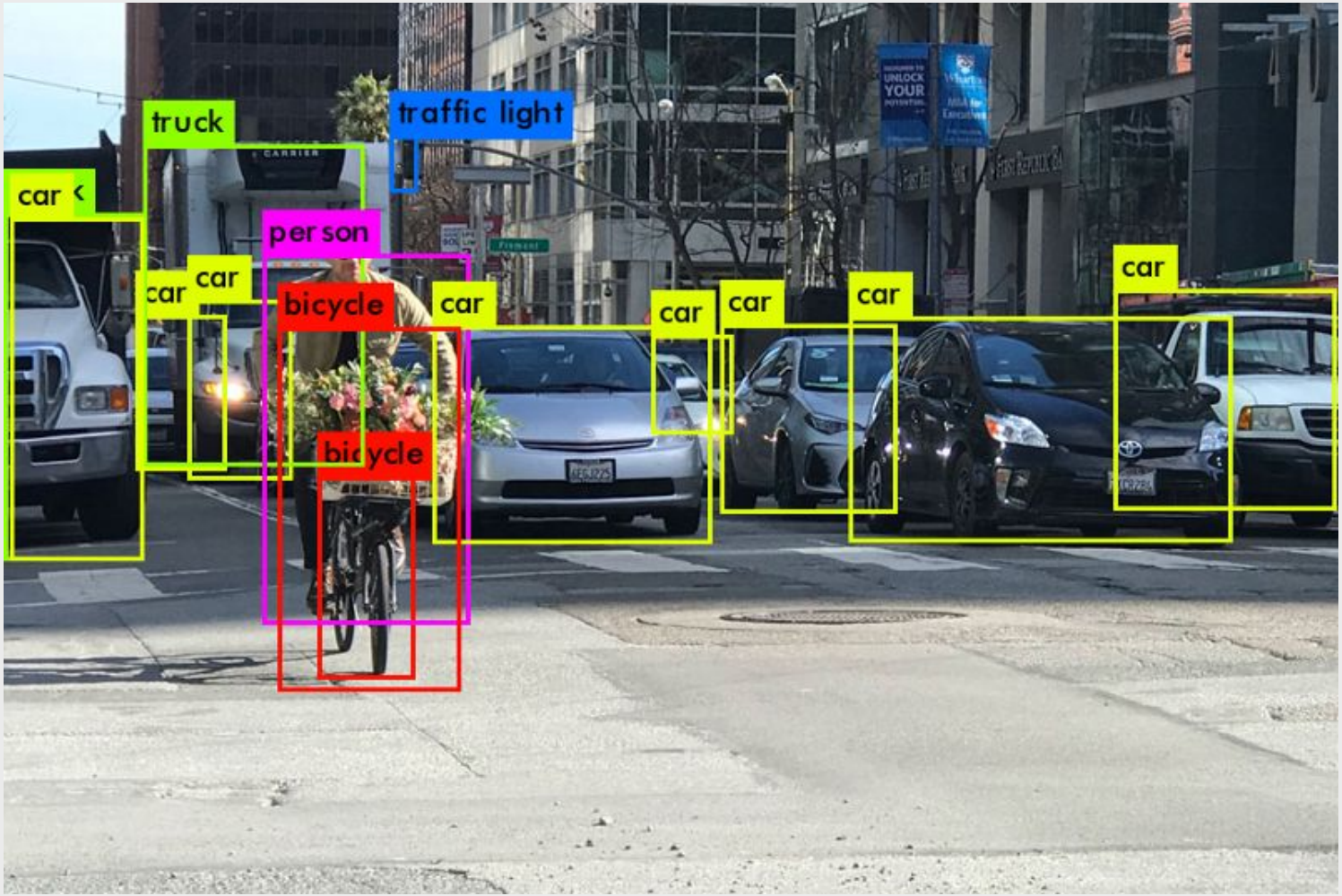
How about YOLOv3? Cross-binary entropy

nodeflux

Extending Vision Beyond Imagination

Section 4 –
YOLO: Inferencing


CARRIER
UNLOCK
YOUR

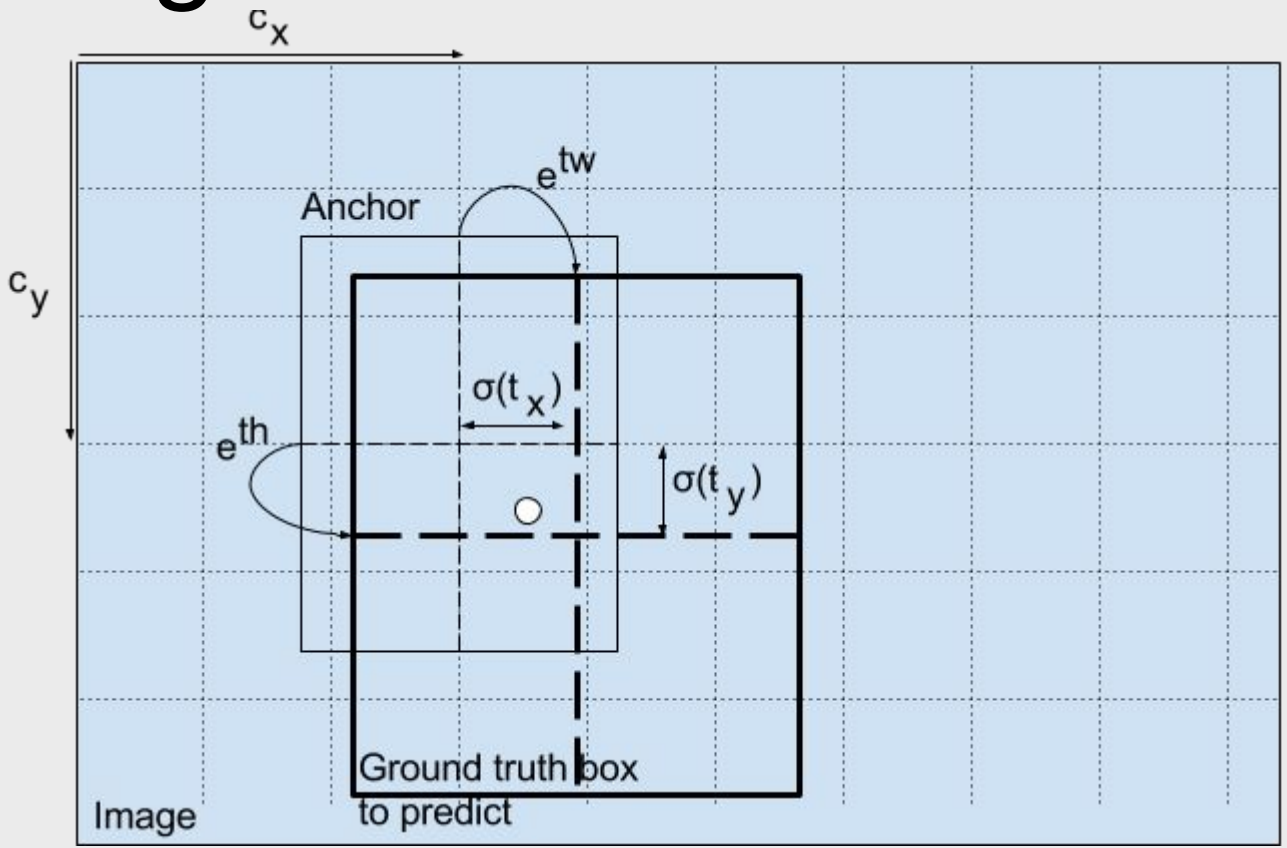

car
truck
traffic light
person
car
car
bicycle
car
car
car
car
car
bicycle

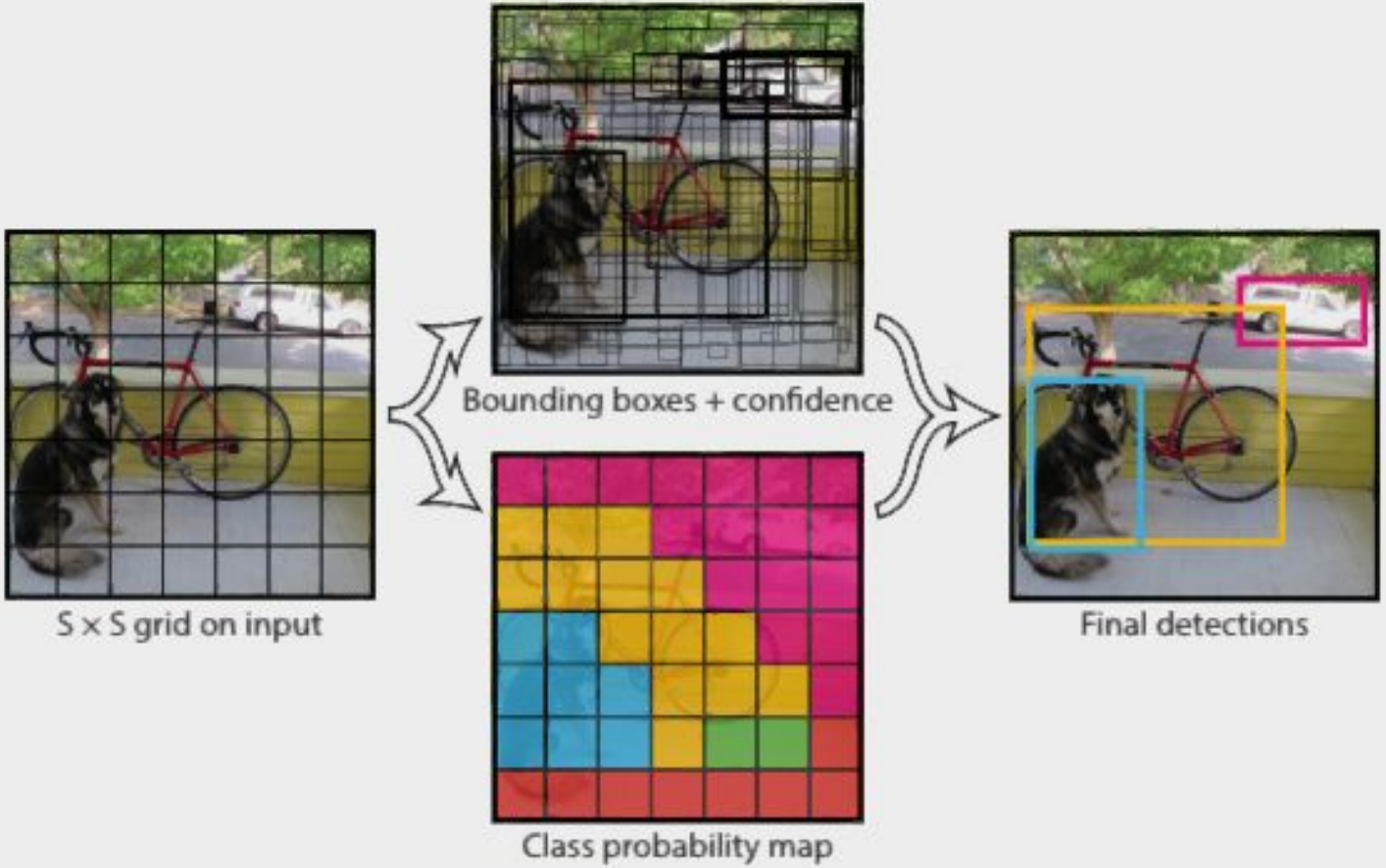
How to choose an anchor?

- Objectness score threshold
- Non-maximum suppression

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$

Training: Anchor




Bounding boxes + confidence
S x S grid on input
Class probability map
Final detections

References

[1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," arXiv:1506.02640v5 [cs.CV], May 2016.
[2] J. Redmon and A. Farhadi, "YOLO9000: Better, Faster, Stronger", arXiv:1612.08242v1 [cs.CV], Dec. 2016.
[3] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement", arXiv:1804.02767v1 [cs.CV], Apr. 2016.
[4] Kathuria, Ayoosh. (2018, April 16). How to implement a YOLO (v3) object detector from scratch in PyTorch. Retrieved Mar 1, 2019, from https://blog.paperspace.com/how-to-implement-a-yolo-object-detector-in-pytorch/
[5] Luo, Ray. (2018, November 10). Implementing YOLO-V3 Using PyTorch. Retrieved Mar 1, 2019, from http://leiluoray.com/2018/11/10/Implementing-YOLOV3-Using-PyTorch
[6] Hui, Jonathan. (2018, March 18). Real-time Object Detection with YOLO, YOLOv2 and now YOLOv3. Retrieved Mar 1, 2019, from https://medium.com/@jonathan_hui/real-time-object-detection-with-yolo-yolov2-28b1b93e2088
[7] Wikipedia contributors. (2019, February 10). Computer vision. In Wikipedia, The Free Encyclopedia. Retrieved March 1, 2019, from https://en.wikipedia.org/w/index.php?title=Computer_vision&oldid=882705228
[8] https://github.com/pjreddie/darknet
[9] https://github.com/AlexeyAB/darknet
[10] https:/www.pytorch.org