YOLO Introduction

- You only look once, real time object detection deep learning network -

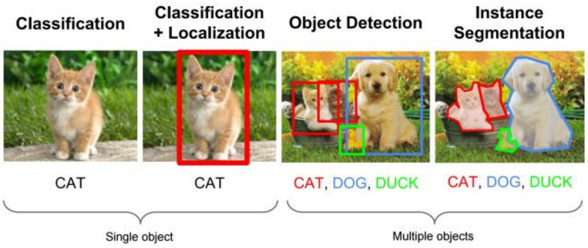
2020

Ando Ki, Ph.D.

adki@future-ds.com

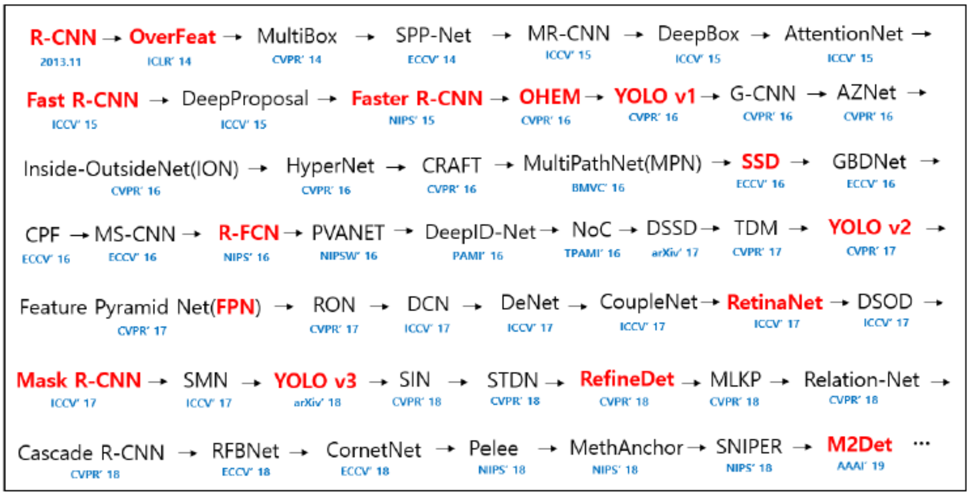
Table of contents

Object recognition / detection



- Image classification
  - to figure out which category is in the picture
- Object localization
  - to figure out where the object locates
  - object localization + classification: for one object
- Object detection
  - to find all the objects in the image and draw bounding boxes
    - dealing with multiple objects in the picture
    - draw bounding box
- Instance segmentation (semantic segmentation)
  - to find exact boundaries of objects

Object detection: state of the art progress



R-CNN → **OverFeat** → MultiBox → SPP-Net → MR-CNN → DeepBox → AttentionNet →
2013.11 ICLR' 14 CVPR' 14 ECCV' 14 ICCV' 15 ICCV' 15 ICCV' 15

Fast R-CNN → DeepProposal → **Faster R-CNN** → **OHEM** → **YOLO v1** → G-CNN → AZNet →
ICCV' 15 ICCV' 15 NIPS' 15 CVPR' 16 CVPR' 16 CVPR' 16 CVPR' 16

Inside-OutsideNet(ION) → HyperNet → CRAFT → MultiPathNet(MPN) → **SSD** → GBDNet →
CVPR' 16 CVPR' 16 CVPR' 16 BMVC' 16 ECCV' 16 ECCV' 16

CPF → MS-CNN → **R-FCN** → PVANET → DeepID-Net → NoC → DSSD → TDM → **YOLO v2** →
ECCV' 16 ECCV' 16 NIPS' 16 NIPSW' 16 PAMI' 16 TPAMI' 16 arXiv' 17 CVPR' 17 CVPR' 17

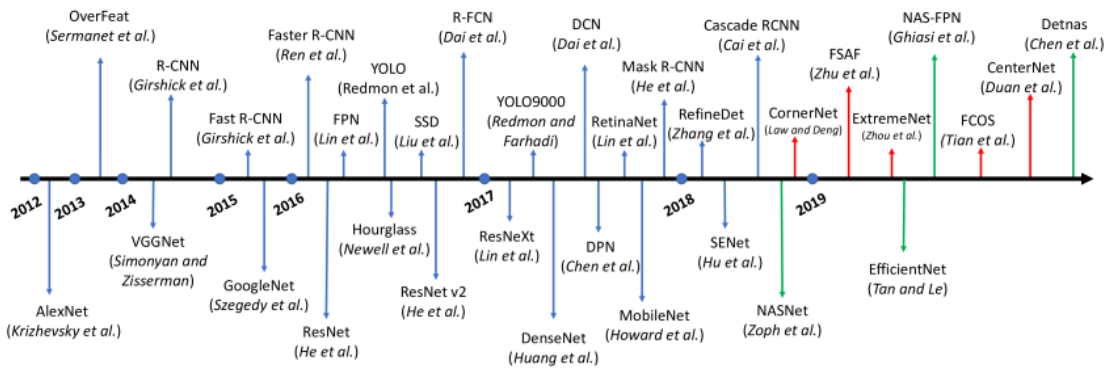
Feature Pyramid Net(**FPN**) → RON → DCN → DeNet → CoupleNet → **RetinaNet** → DSOD →
CVPR' 17 CVPR' 17 ICCV' 17 ICCV' 17 ICCV' 17 ICCV' 17 ICCV' 17

Mask R-CNN → SMN → **YOLO v3** → SIN → STDN → **RefineDet** → MLKP → Relation-Net →
ICCV' 17 ICCV' 17 arXiv' 18 CVPR' 18 CVPR' 18 CVPR' 18 CVPR' 18 CVPR' 18

Cascade R-CNN → RFBNet → CornetNet → Pelee → MethAnchor → SNIPER → **M2Det** …
CVPR' 18 ECCV' 18 ECCV' 18 NIPS' 18 NIPS' 18 NIPS' 18 AAAI' 19

https://deeplearning.mit.edu

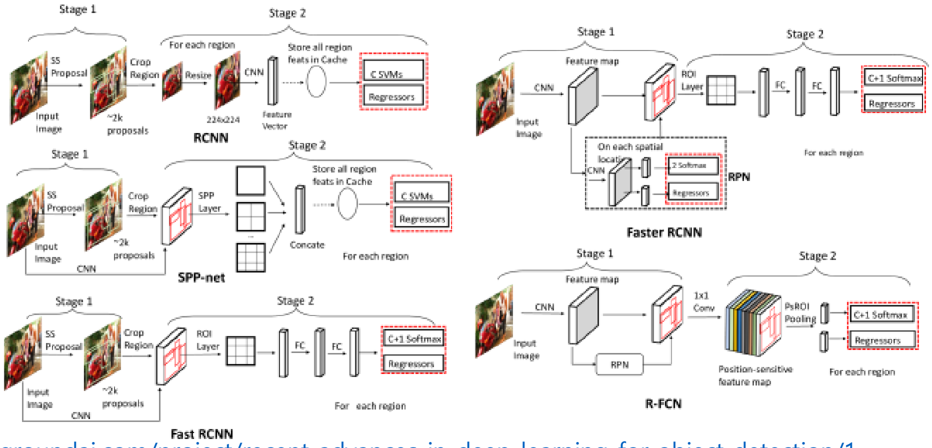
Object detection: state of the art progress



https://www.groundai.com/project/recent-advances-in-deep-learning-for-object-detection/1

Object detections: R-CNN (Region-based CNN)

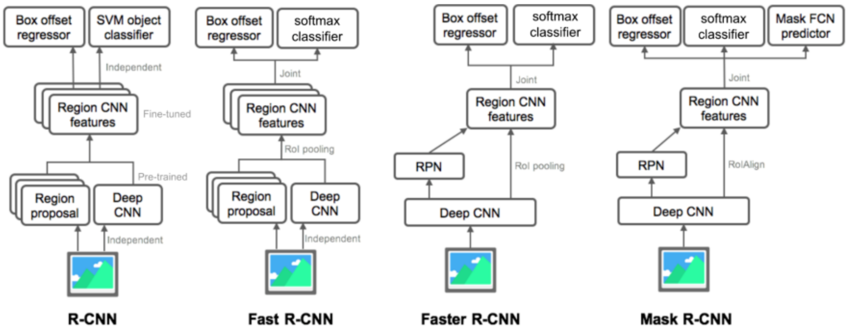
- Two-stage detectors: proposal generation and region classification



https://www.groundai.com/project/recent-advances-in-deep-learning-for-object-detection/1

Object detections: R-CNN (Region-based CNN)

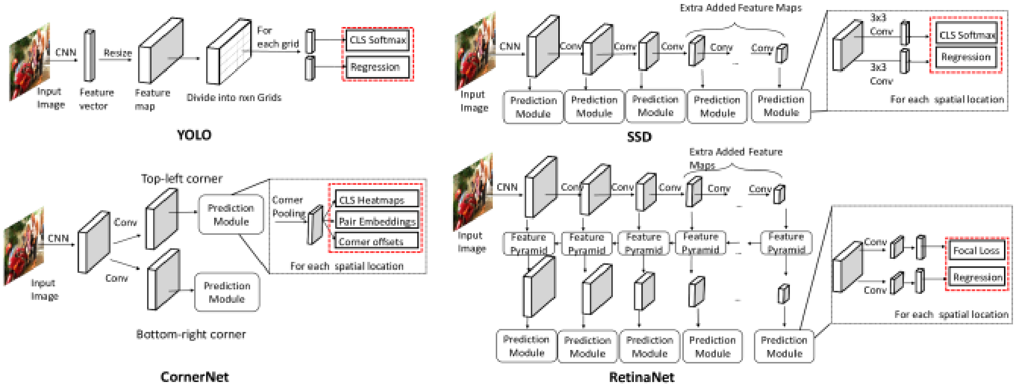
- Two-stage detectors: proposal generation and region classification
  - 1. First, the model proposes a set of regions of interests by select search or regional proposal network.
  - 2. Then a classifier only processes the region candidates



https://lilianweng.github.io/lil-log/2017/12/31/object-recognition-for-dummies-part-3.html

Object detections: YOLO

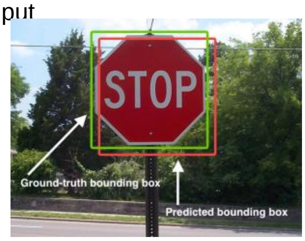
- One-stage detectors (unified detectors)



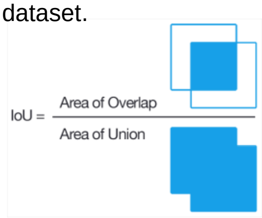
https://www.groundai.com/project/recent-advances-in-deep-learning-for-object-detection/1

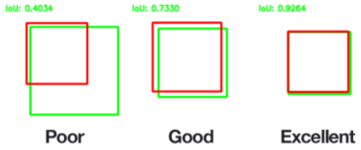
Terminologies

- GT: Ground Truth box (i.e., hand labeled box)
  - ▶ the hand labeled bounding boxes from the training/testing set that specify *where* in the image our object is
  - ▶ represents the desired output (ideal output) of an algorithm on an input
- PB: Predicted box
  - ▶ calculated box



- IoU (Intersection over Union)
  - ▶ an evaluation metric used to measure the **accuracy of an object detector** on a particular dataset.





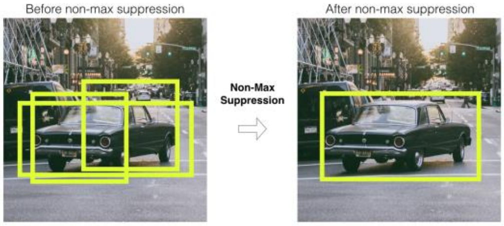
Labeled data가 있으므로 계산이 가능

Terminologies

- Confidence score
  - ▶ how certain it is that the predicted bounding box actually encloses some object.
    - ➲ This score doesn't say anything about what kind of object is in the box, just if the shape of the box is any good.
    - ➲ 0 means no object
  - ▶ E.g., softmax

Confidence Score: Pr(Object)∗IOU(pred, truth)

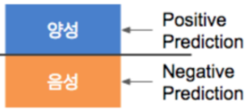
- Non-max suppression
  - ▶ Removes bounding boxes (ROI: region of interest) with low confidence score, since most of bounding boxes will not contain an

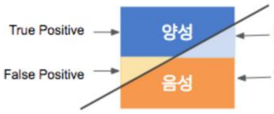


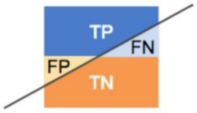
Labeled data가 있으므로 계산이 가능

Terminologies

- Confusion matrix
  - P = TP + FN
  - N = FP + TN



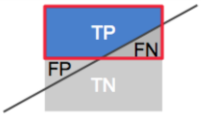




- Accuracy = (TP+TN)/(P+N)
  - 전체 중 제대로 예측한 비 (모델의 정확도)
- Error rate = (FN+FP)/(P+N)
  - 전체 중 잘 못 분류한 비

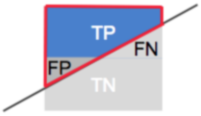
https://bcho.tistory.com/m/1206

- Sensitivity (Recall) = TP/P
  - 민감도
  - 옳다고 예측 한 것 (사선 위) 중 옳은 것(TP)이 전체 옳은 것(P)에 대한 비



Sensitivity (Recall)

- Precision = TP/(TP+FP)
  - 정밀도
  - 맞다고 예측한 것 중 실제로 옳은 것의 비



Precision

Terminologies

- Let define true positive when IoU>0.5

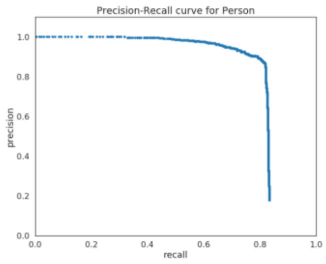
True positive
IoU of predicted BB (yellow) and GT BB (blue) >= 0.5 with correct prediction

True negative

False positive
IoU<0.5
Duplicated BB

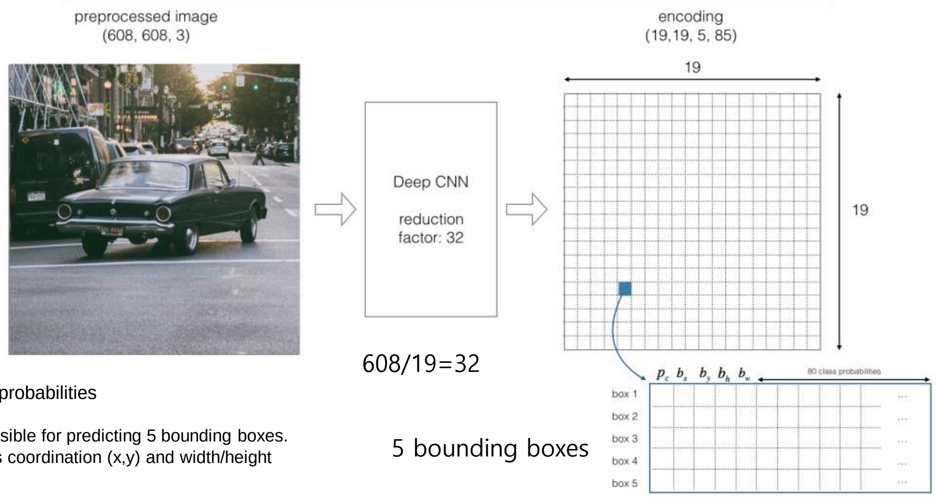
False negative
no detection at all
wrong prediction even IoU>=0.5

- mAP (Mean Average Precision)
  - 한 prediction에서 여러 class에 대한 민감도(recall)와 정밀도(precision)을 2차원으로 표현하고 (precision-recall curve)
  - 이 것의 적분으로 object detection의 정도를 측정 ➔ AP
  - 모든 class에 대해 평균을 낸 것➔ mAP



https://towardsdatascience.com/breaking-down-mean-average-precision-map-ae462f623a52

Terminologies

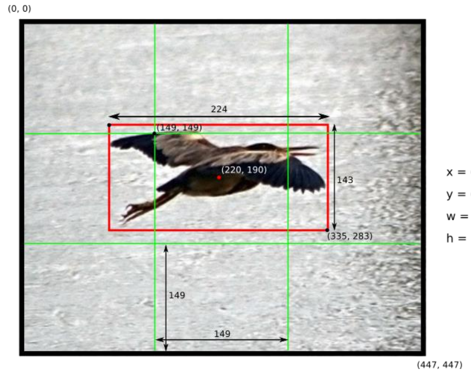


608/19=32

5 bounding boxes

- Grid, bounding box, class probabilities
  - 19x19 grid.
  - Each grid cell is responsible for predicting 5 bounding boxes.
  - Each bounding box has coordination (x,y) and width/height (w,h)
  - 80 classes

Terminologies



x = (220-149) / 149 = 0.48

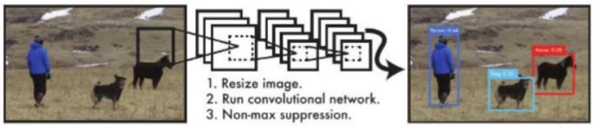
y = (190-149) / 149 = 0.28

w = 224 / 448 = 0.50

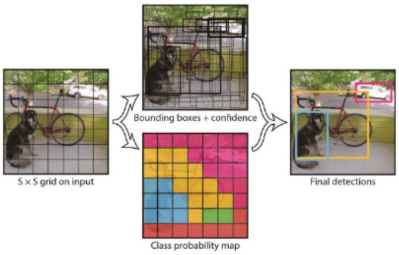
h = 143 / 448 = 0.32

- Grid and bounding box example
  - Example of how to calculate box coordinates in a 448x448 image with S=3.
  - Note how the (x,y) coordinates are calculated relative to the center grid cell.
  - Note how the (w,h) ratio are calculated relative to the size of image.

YOLO (V1) detection system

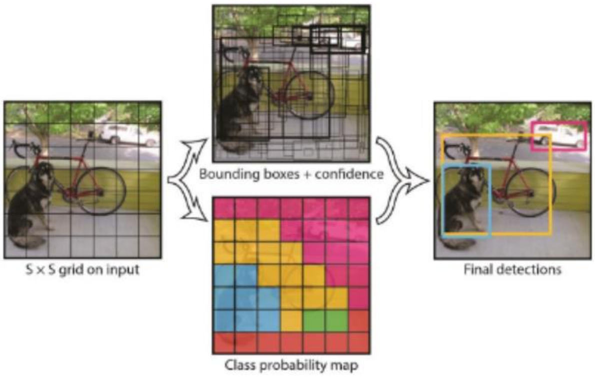


- (1) resize input image to 448x448
- (2) run a single convolution network: a regression
- (3) get result by confidence



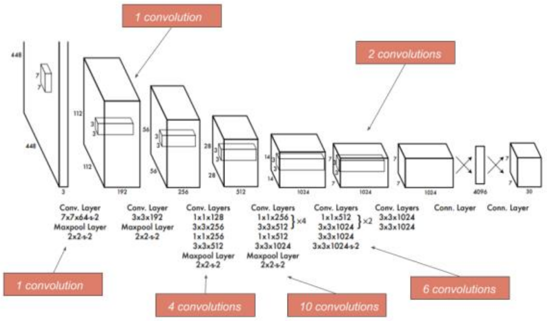
- (1) divides the image into an SxS (7x7) grid
- (2) predicts B (2) bounding boxes for each grid cell
  - only for bounding boxes those center fall in the grid
- (3) Get confidence for the boxes of C class probabilities

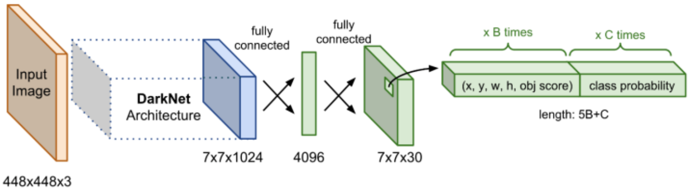
YOLO (V1) detection system



- Divide the input image into an **S × S** grid.
- Each grid cell predicts **B** bounding boxes.
- Each bounding box :
  - **Confidence** = $Pr(oggetto) * IOU_{pred}^{truth}$.
  - $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{w}, \boldsymbol{h}$ = (x, y) bb center, w width, h height
- **C** class probabilities.
- Prediction = **S × S × (B ∗ 5 + C)**

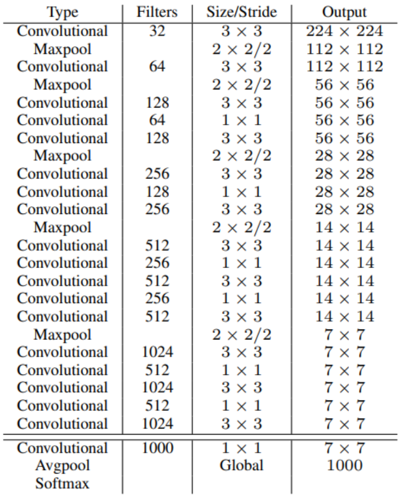
YOLO V1





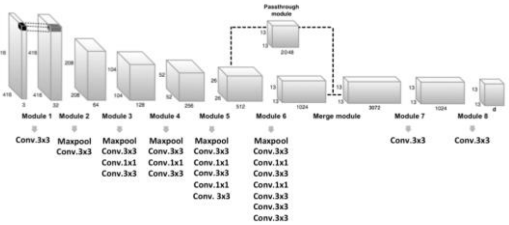
- 24 convolution layers
- 2 fully connected layers

YOLO 2



Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

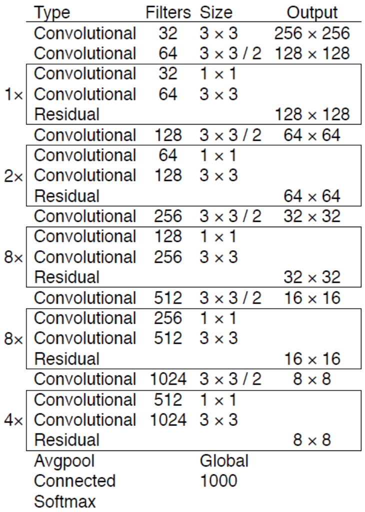
- Use darknet-19 architecture for feature extractor
- 30 layer architecture
- Batch normalization
- Anchor boxes
- High resolution input
  - 224x224 ➔ 448x448
- Fine-grained features
  - 13x13 ➔ 26x26
- No fully connection network at classifier layer



YOLO 9000

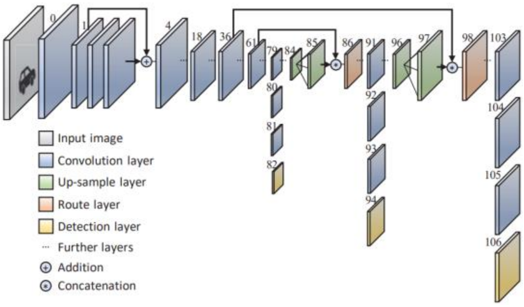
- YOLO9000
  - a real-time system that detects more than 9000 objects categories by combining COCO’s detection dataset (80 classes) with ImageNet’s classification dataset (~22K classes).
  - Use YOLO V2 that trained separately for classification and detection. ➔ Rich dataset training

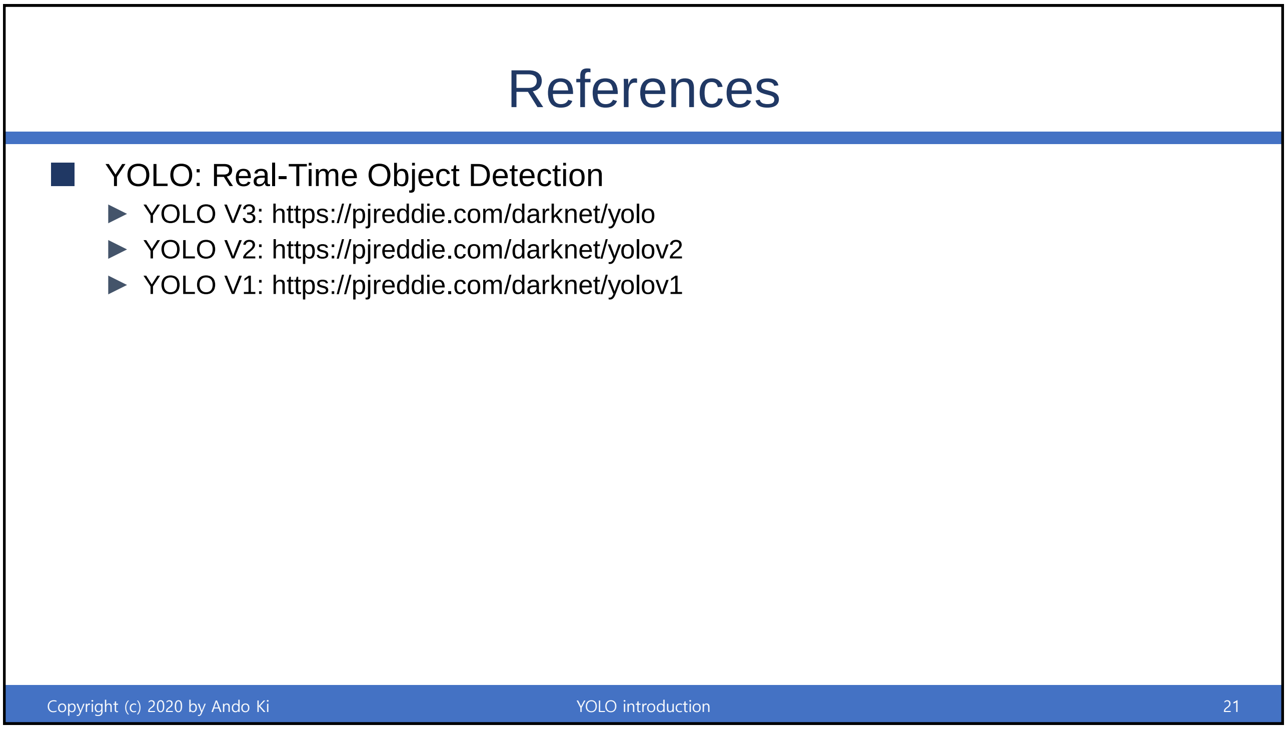
YOLO V3



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

- Use darknet-53 architecture for feature extraction
- 106 layer architecture





References

- YOLO: Real-Time Object Detection
  - YOLO V3: https://pjreddie.com/darknet/yolo
  - YOLO V2: https://pjreddie.com/darknet/yolov2
  - YOLO V1: https://pjreddie.com/darknet/yolov1