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

Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

2015, CVPR2016

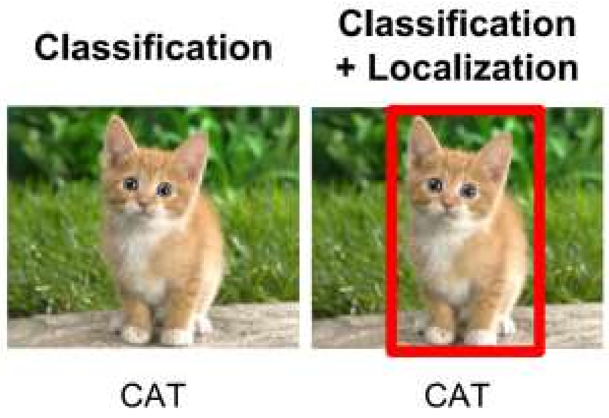
https://arxiv.org/abs/1506.02640v5

# Contents

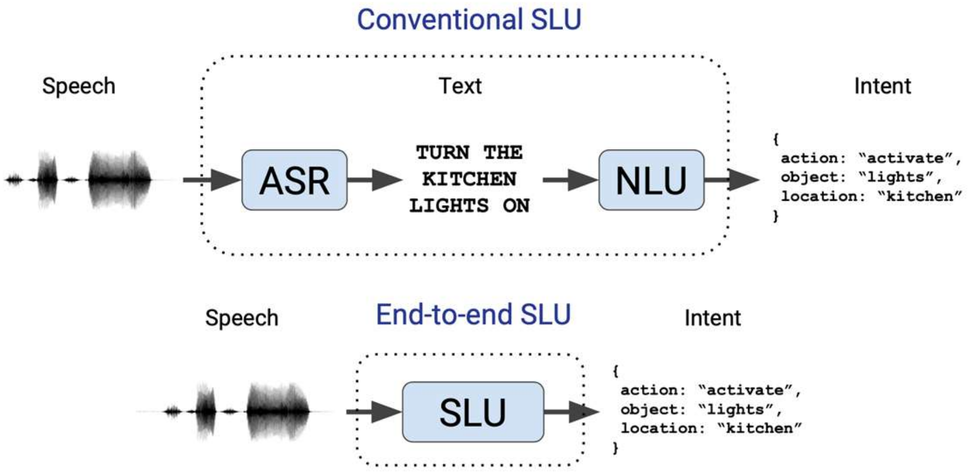
- Task and Contribution
- Object Detection
- 2 Stage Detector : R-CNN
- 1 Stage Detector : YOLO
    - Train : Bbox & Confidence & Class Prob.
    - Eval. : IOU Non maximum Suppression
- Limitation
- Results

# Task and Contribution

- Object Detection
  - Localization + Classification
    - Bounding Box + Class Probability



- Unified (End to End) -> Real-time
  - Applying gradient-based learning to the system as a whole*

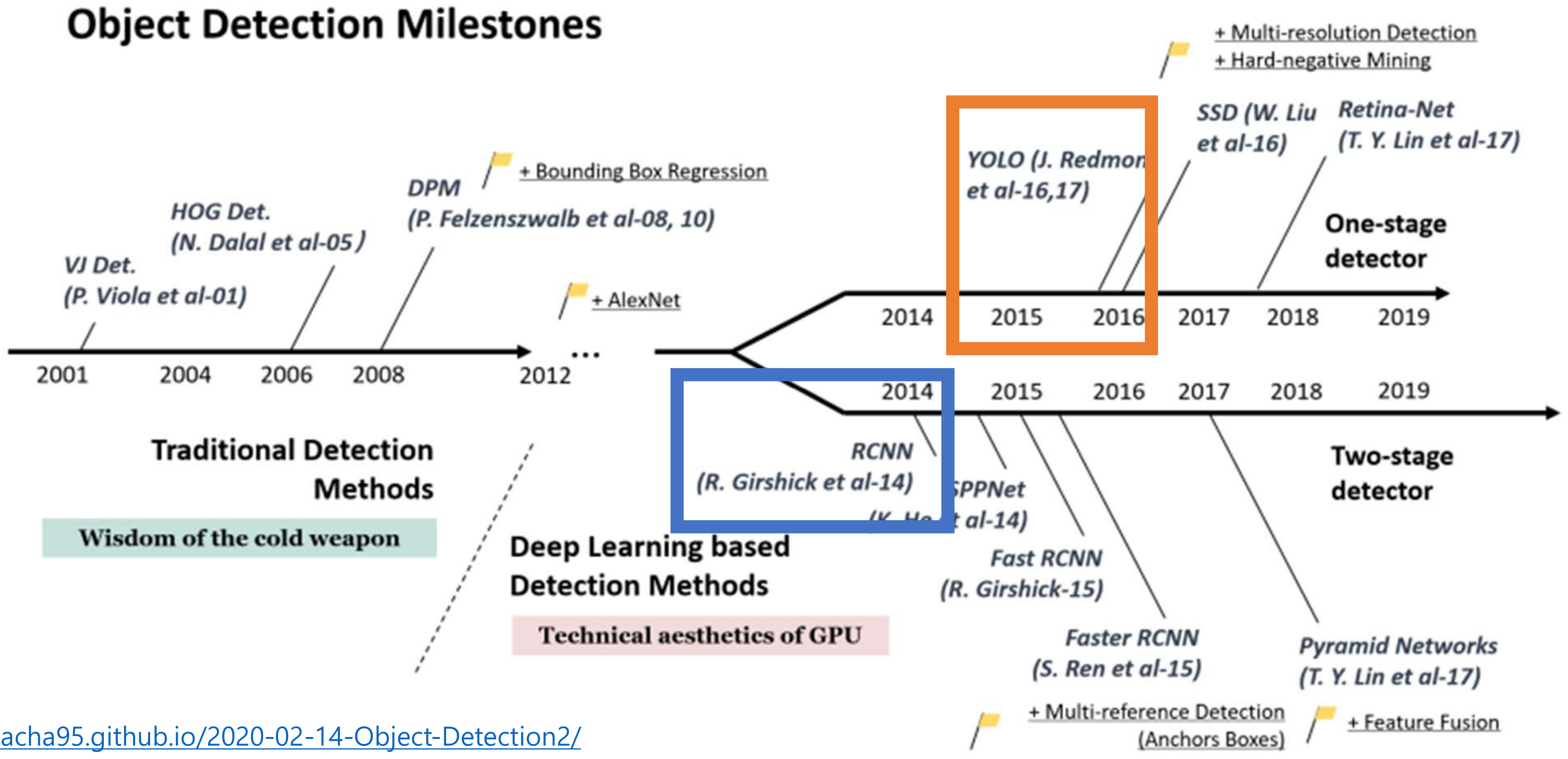

Conventional SLU
Speech
Text
Intent
ASR
TURN THE KITCHEN LIGHTS ON
NLU
{ action: "activate", object: "lights", location: "kitchen" }
End-to-end SLU
Speech
Intent
SLU
{ action: "activate", object: "lights", location: "kitchen" }

https://medium.com/analytics-vidhya/object-localization-using-keras-d78d6810d0be
https://ratsgo.github.io/speechbook/docs/neuralam/e2eslu

*http://proceedings.mlr.press/v77/glasmachers17a/glasmachers17a.pdf

# Object Detection

- Before YOLO…

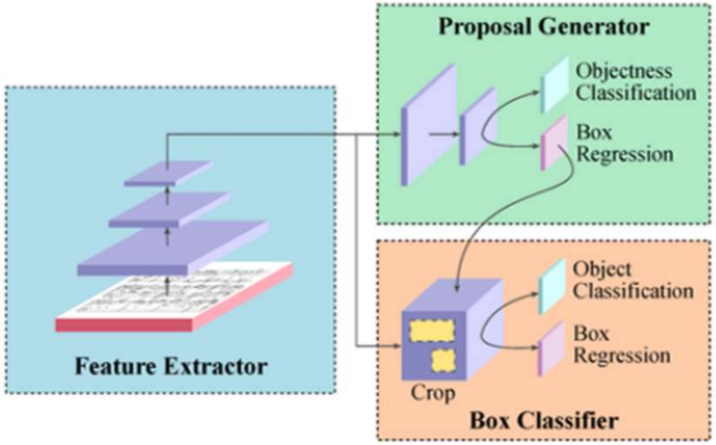


Object Detection Milestones
+ Multi-resolution Detection
+ Hard-negative Mining
SSD (W. Liu et al-16)
Retina-Net (T. Y. Lin et al-17)
YOLO (J. Redmon et al-16,17)
+ Bounding Box Regression
DPM (P. Felzenszwalb et al-08, 10)
HOG Det. (N. Dalal et al-05)
VJ Det. (P. Viola et al-01)
One-stage detector
+ AlexNet
2014 2015 2016 2017 2018 2019
2001 2004 2006 2008 2012 ...
2014 2015 2016 2017 2018 2019
RCNN (R. Girshick et al-14)
SPPNet (K. He et al-14)
Two-stage detector
Traditional Detection Methods
Wisdom of the cold weapon
Deep Learning based Detection Methods
Technical aesthetics of GPU
Fast RCNN (R. Girshick-15)
Faster RCNN (S. Ren et al-15)
Pyramid Networks (T. Y. Lin et al-17)
+ Multi-reference Detection (Anchors Boxes)
+ Feature Fusion

https://chacha95.github.io/2020-02-14-Object-Detection2/

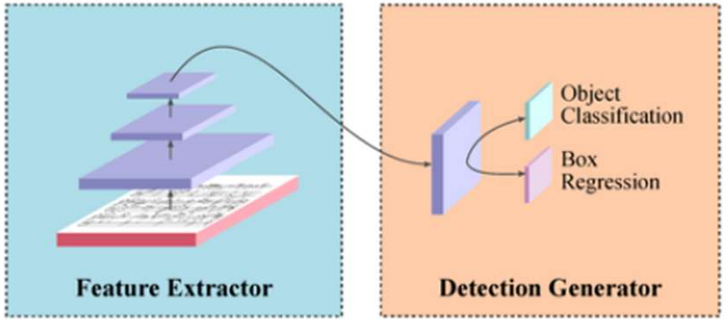
# Object Detection

- 2 Stage
  - 2 Output = Localization + Classification
  - E.g. R-CNN



(**b**) Basic architecture of a two-stage detector.

- 1 Stage
  - 1 Output = Localization & Classification
  - E.g. YOLO

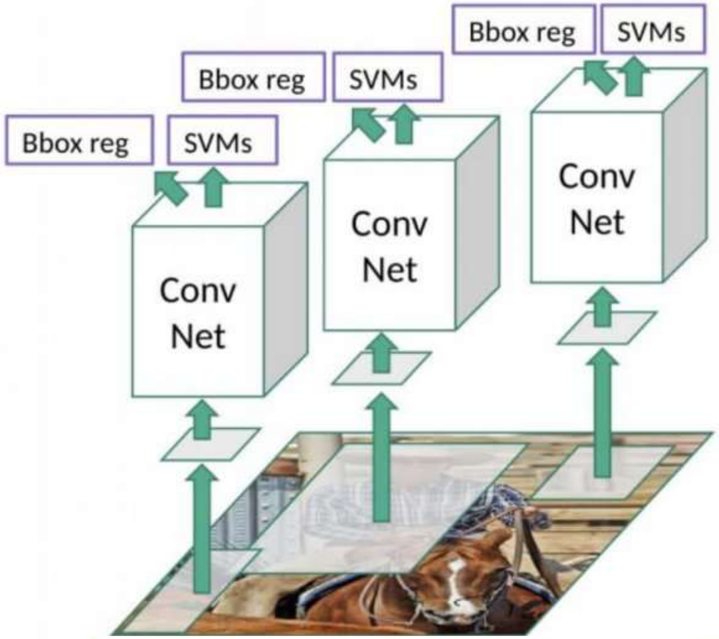


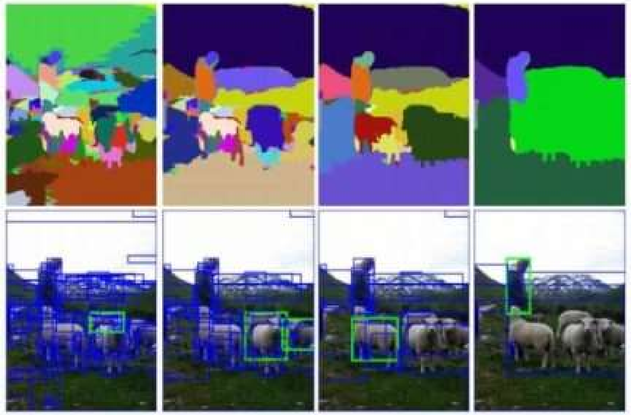
(**a**) Basic architecture of a one-stage detector.

https://gaussian37.github.io/vision-detection-table/

# 2 Stage Detector : R-CNN

- R-CNN
  1. Region Proposal : Candidate Region (2K)
     - Hierarchy Clustering : Can not be trained
  2. Feature Extraction
     - CNN
  3. Classification
     - SVM
  4. Bounding box (Bbox) Regression
     - Linear Regression



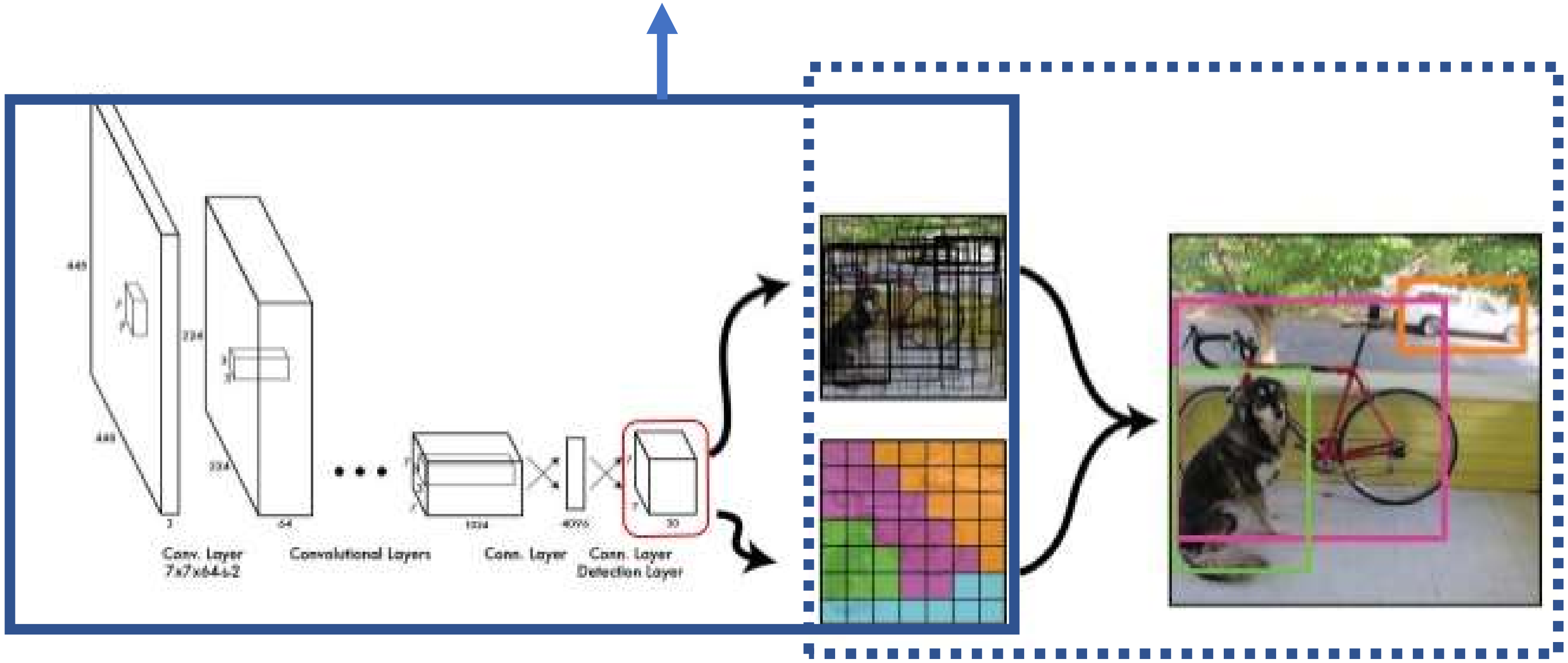


- Slow and Complicated
  - To solve : YOLO vs Fast R-CNN
- Hard to Optimization
  - Different and Separated models

https://neurohive.io/en/popular-networks/r-cnn/

# 1 Stage Detector : YOLO

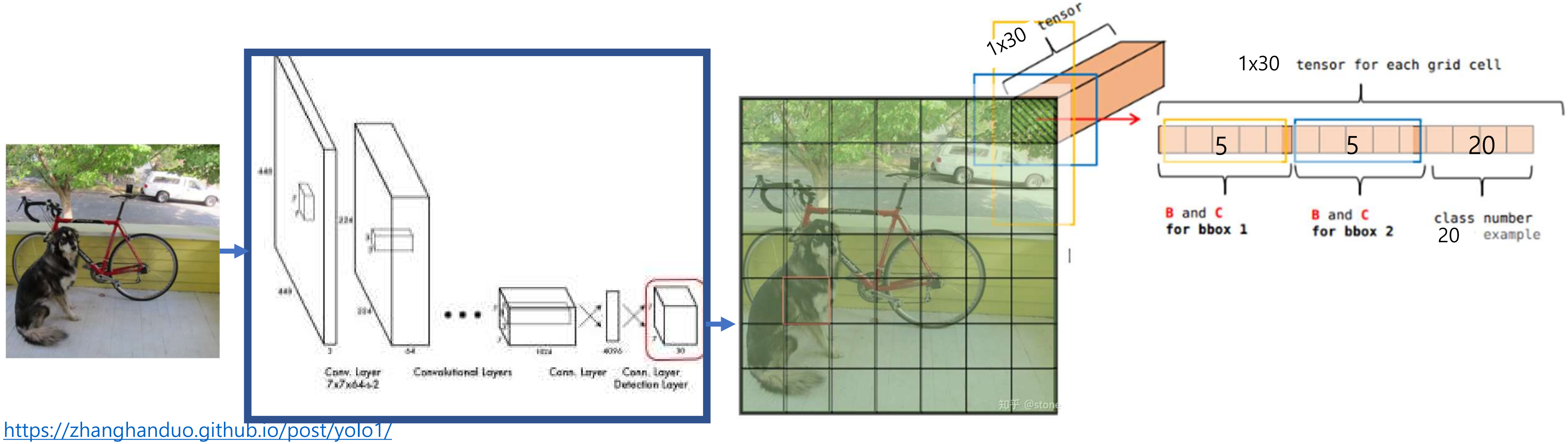
- YOLO - Train
  1. SxS Gridded Bbox Location & Confidence & Class Probability



https://stackoverflow.com/questions/49707542/yolo-v1-bounding-boxes-during-training-step

# 1 Stage Detector : YOLO

- YOLO - Train
  1. SxS Gridded Bbox Location & Confidence & Class Probability
     - Output designed to divide grid on Image and contain the feature for detection from each grid

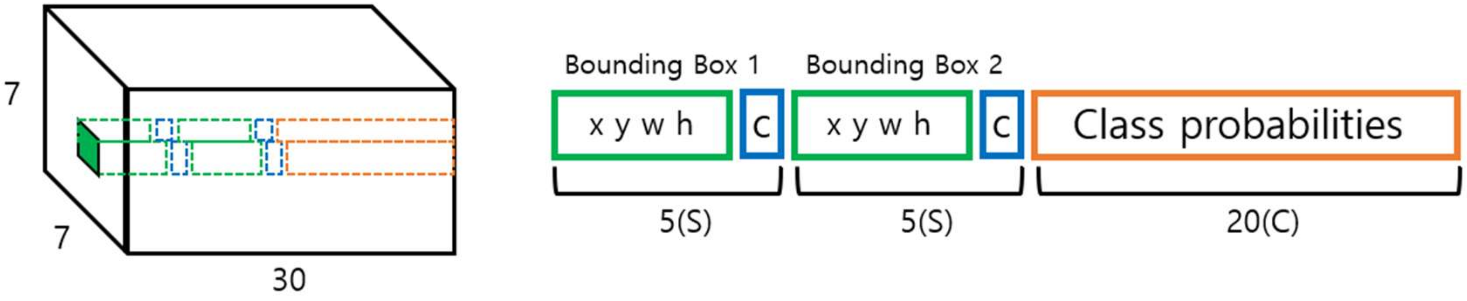


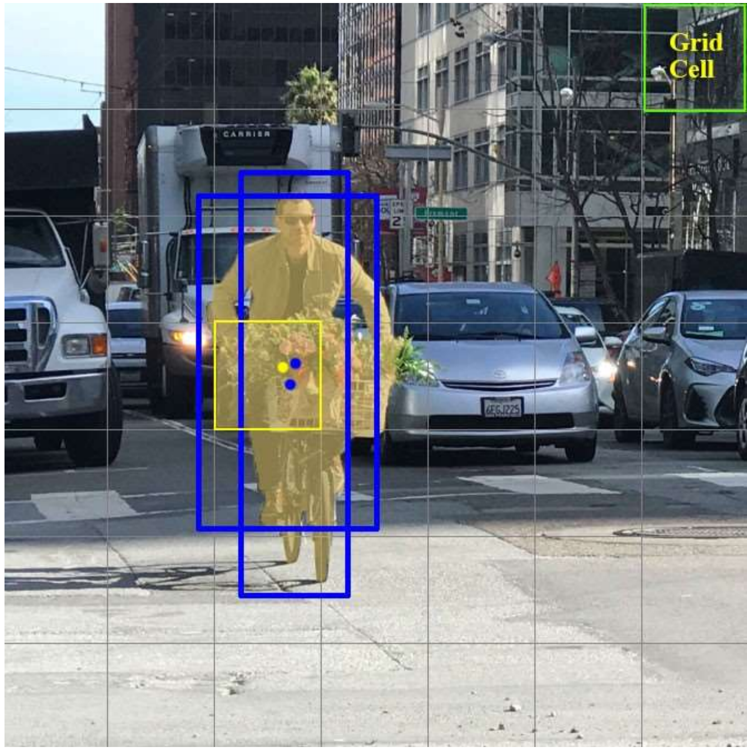
https://zhanghanduo.github.io/post/yolo1/

https://stackoverflow.com/questions/49707542/yolo-v1-bounding-boxes-during-training-step

# 1 Stage Detector : YOLO

- YOLO - Train
  1. SxS Gridded Bbox Location & Confidence & Class Probability
     - 1 Grid = 1 Object*
       - 2 Bbox
       - 2 Confidence : Object Probability
       - 1 Class**
       - All Value 0~1 Normalized





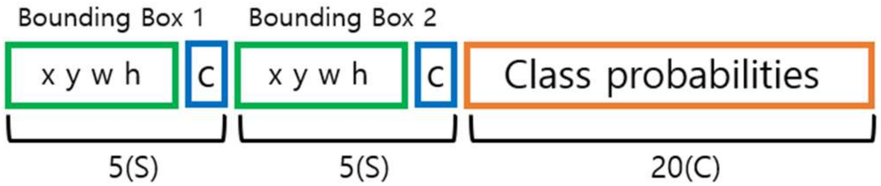
*https://amrokamal-47691.medium.com/yolo-yolov2-and-yolov3-all-you-want-to-know-7e3e92dc4899

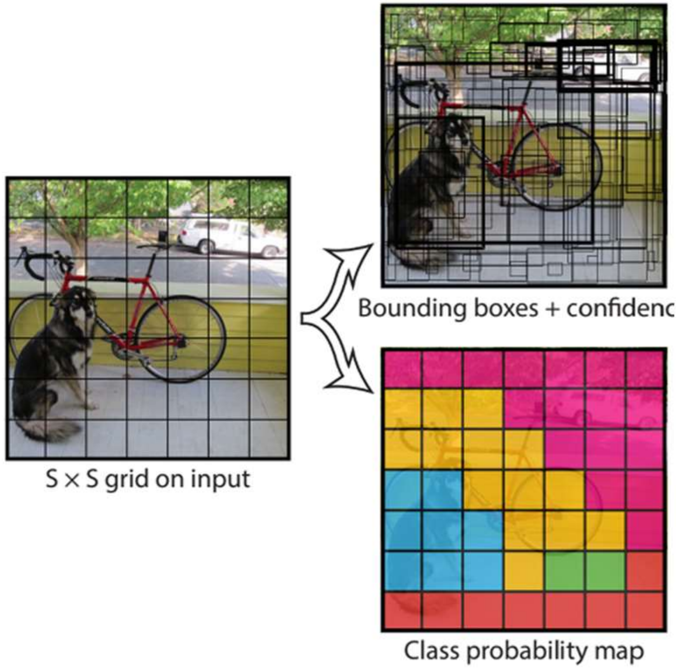
**https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

https://deepbaksuvision.github.io/Modu_ObjectDetection/posts/04_02_Model.html

# 1 Stage Detector : YOLO

- YOLO - Train
  1. SxS Gridded Bbox Location & Confidence & Class Probability

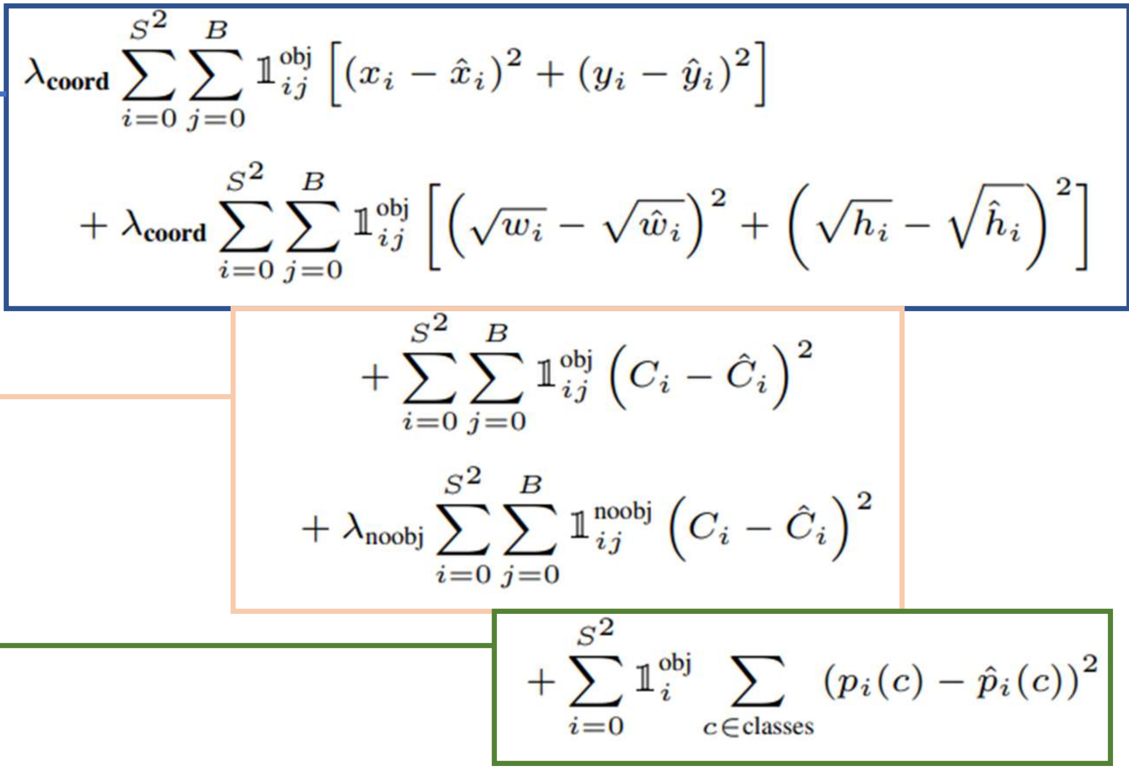




https://curaai00.tistory.com/8
https://deepbaksuvision.github.io/Modu_ObjectDetection/posts/04_02_Model.html

# 1 Stage Detector : YOLO

- YOLO - Train
  1. SxS Gridded Bbox Location & Confidence & Class Probability



- Bbox Loss
  - Bbox Position Loss
  - Bbox Scale Loss

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

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

- Confidence Loss
  - Positive Confidence Loss
  - Negative Confidence Loss

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

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

- Class Probability Loss
  - Positive Class Probability Loss

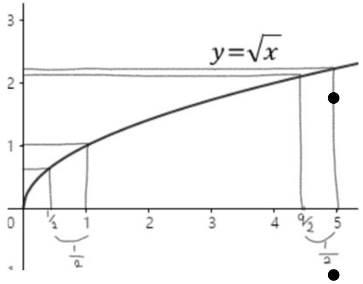
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2$$

# 1 Stage Detector : YOLO

- YOLO - Train

1. SxS Gridded Bbox Location & Confidence & Class Probability

- Coordinate Parameter(=5)
  - Localization > Classification
- Object in Bbox == Positive
  - Calculate only Positive Bbox
- Relative Scale Loss
  - Difference is more lethal for small boxes.
- No obj Parameter (=0.5)
  - Most of Grid has no object
- Class Probability Loss
  - Calculate only Negative Bbox



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

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

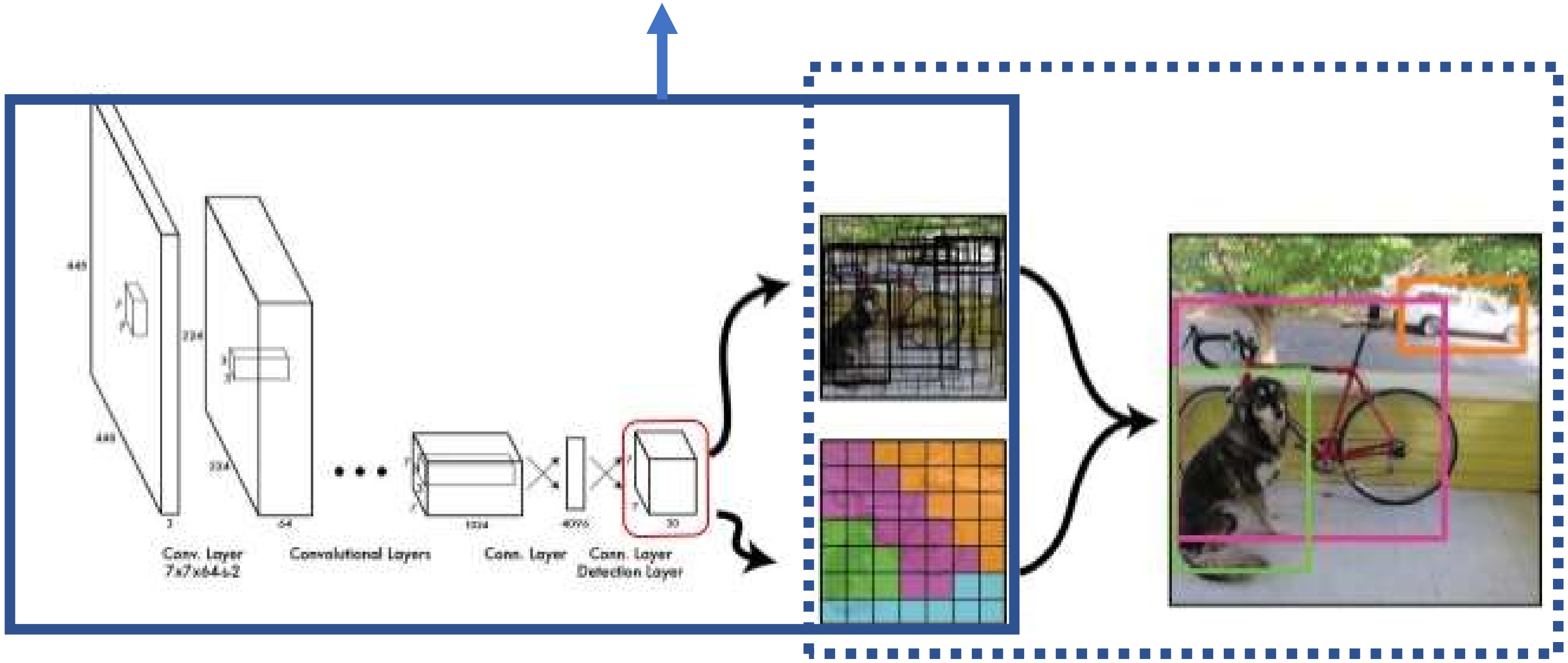
$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left( C_i - \hat{C}_i \right)^2$$

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

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

# 1 Stage Detector : YOLO

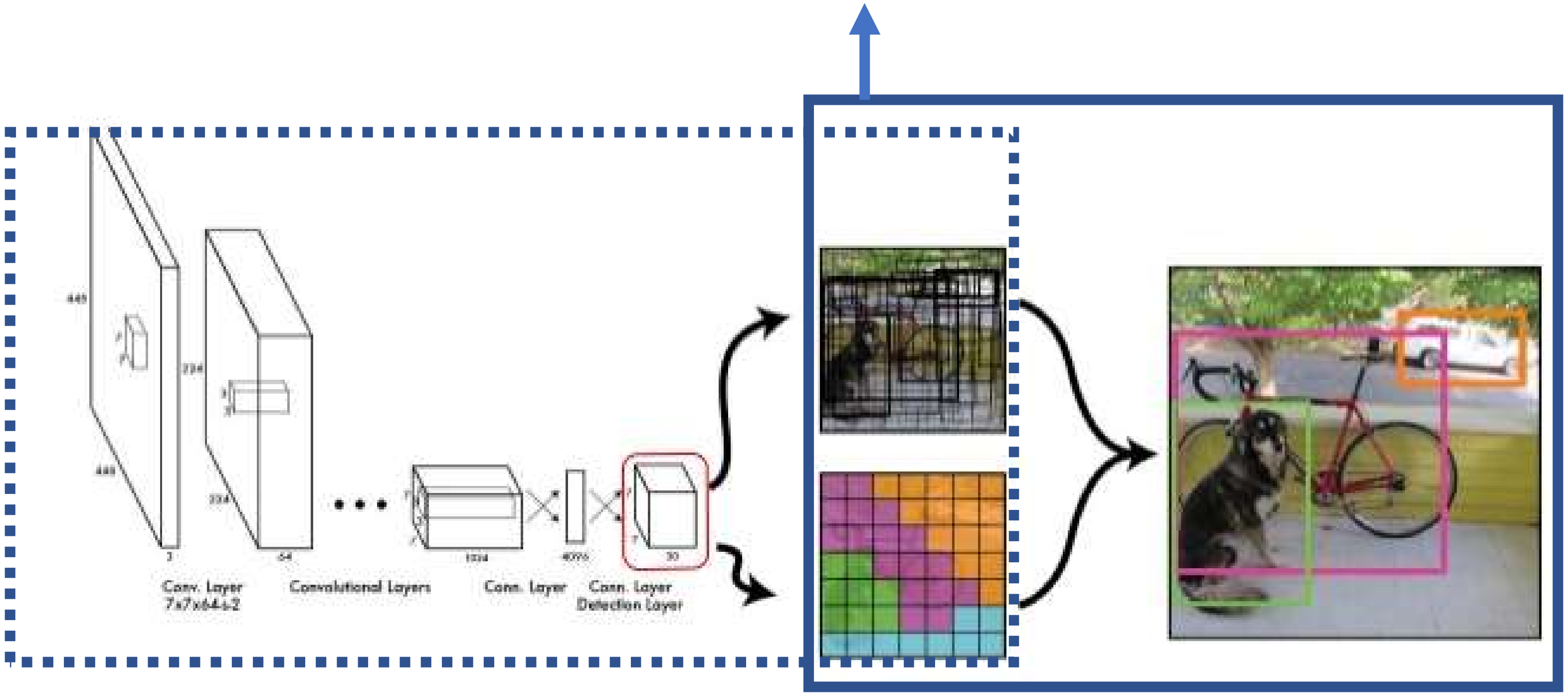
- YOLO - Train
    1. SxS Gridded Bbox Location & Confidence & Class Probability



https://stackoverflow.com/questions/49707542/yolo-v1-bounding-boxes-during-training-step

# 1 Stage Detector : YOLO

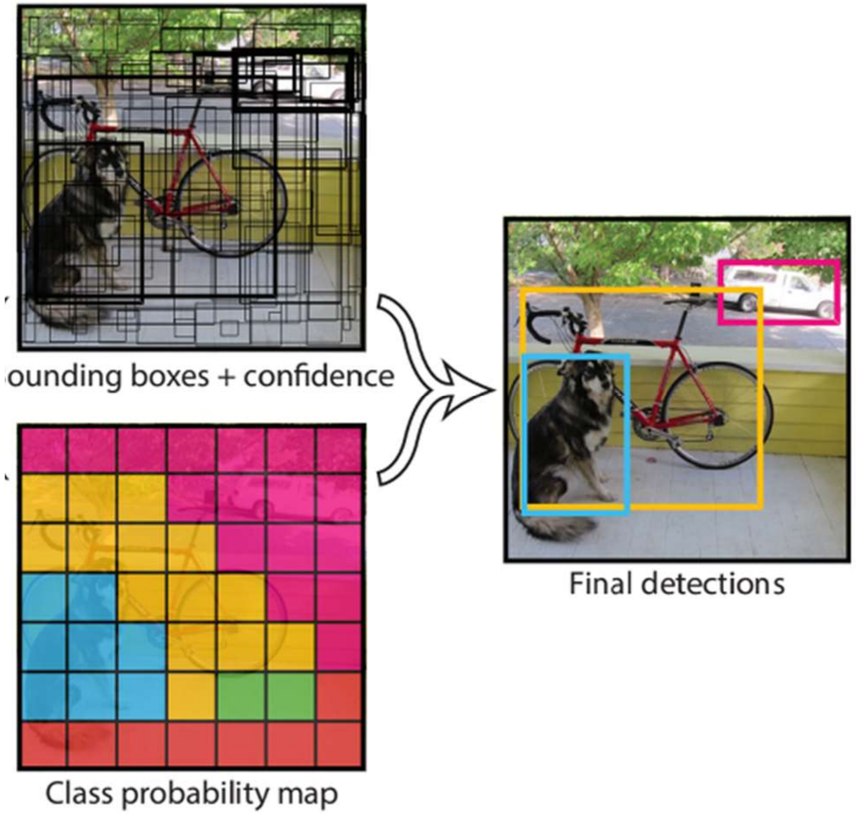
- YOLO – Eval.
    2. IOU Non-maximum Suppression



https://stackoverflow.com/questions/49707542/yolo-v1-bounding-boxes-during-training-step

# 1 Stage Detector : YOLO

- YOLO – Eval.
  2. IOU Non-maximum Suppression
     - Need to Select Bbox -> mAP +2~3%
       - 1 Grid, 2 Bounding Box, 1 Object
         - By Confidence
       - N Grid, N Bounding Box 1 Object
         - By Confidence &
           IOU Non-maximum Suppression

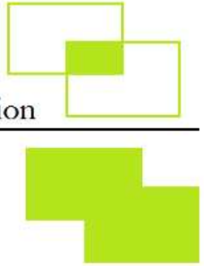


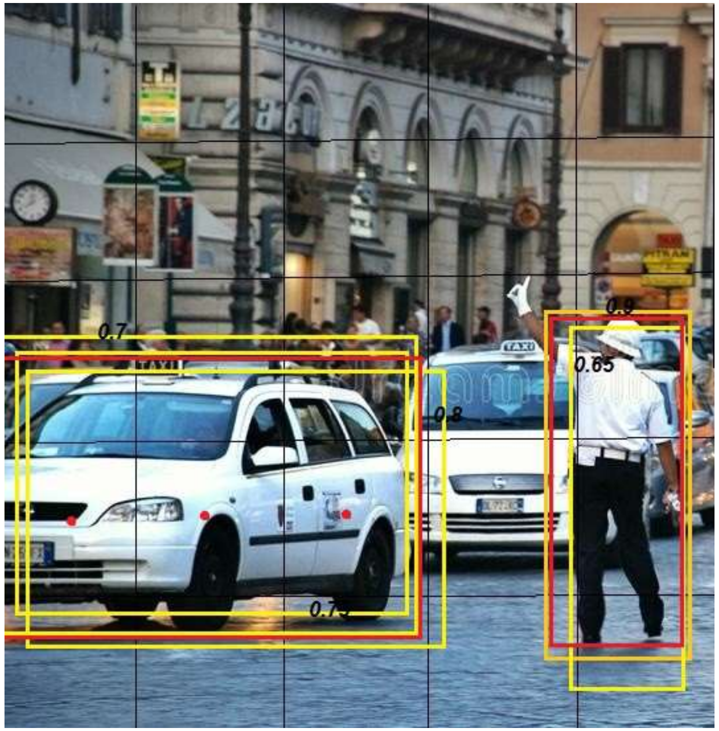
https://curaai00.tistory.com/8

# 1 Stage Detector : YOLO

- YOLO – Eval.
  2. IOU Non-maximum Suppression

    - $\text{IOU} = \dfrac{\text{Area of Intersection}}{\text{Area of Union}}$



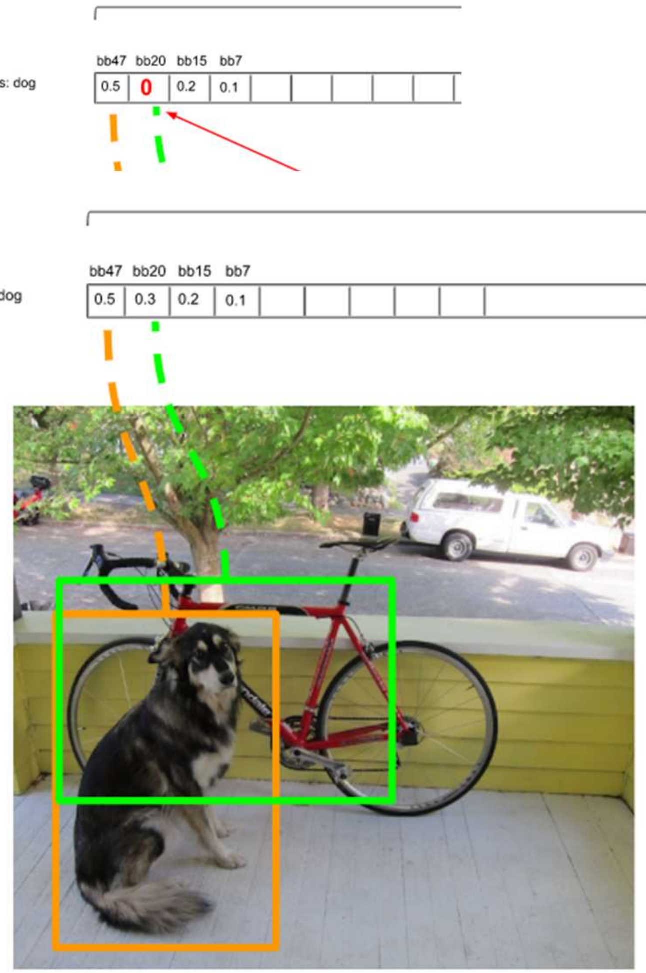


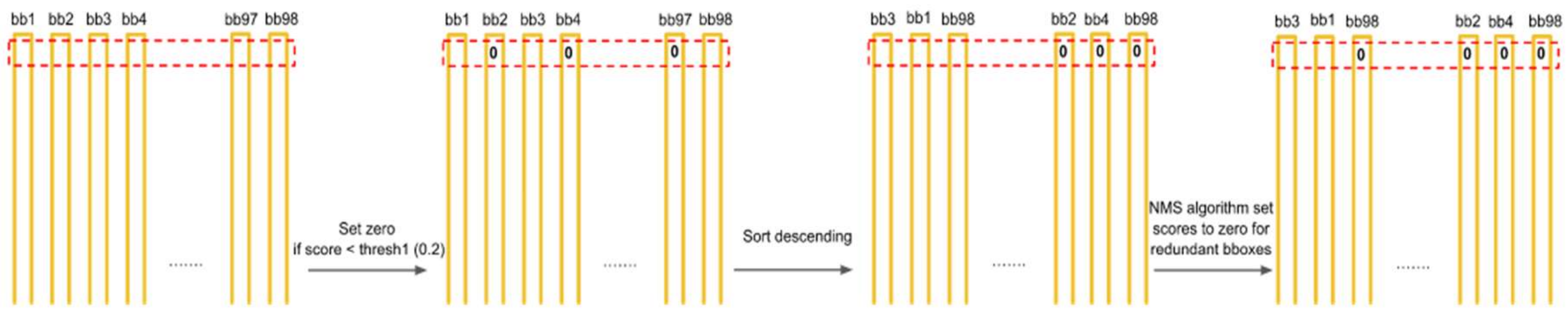
    - Non-maximum Suppression
      - Leaves only the maximum value among those high IOU.

https://amrokamal-47691.medium.com/yolo-yolov2-and-yolov3-all-you-want-to-know-7e3e92dc4899

# 1 Stage Detector : YOLO

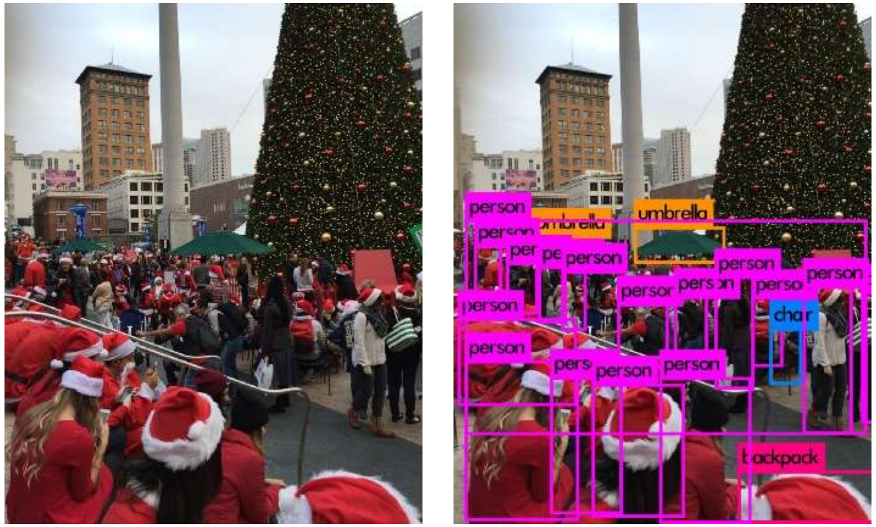
- YOLO – Eval.
  2. IOU Non-maximum Suppression
     - Sorted by confidence
     - Calculate IOU




bb1 bb2 bb3 bb4 bb97 bb98
.......
Set zero
if score < thresh1 (0.2)
bb1 bb2 bb3 bb4 bb97 bb98
0 0 0
.......
Sort descending
bb3 bb1 bb98 bb2 bb4 bb98
0 0 0
.......
NMS algorithm set scores to zero for redundant bboxes
bb3 bb1 bb98 bb2 bb4 bb98
0 0 0 0
.......

https://www.youtube.com/watch?v=L0tzmv--CGY

# Limitation

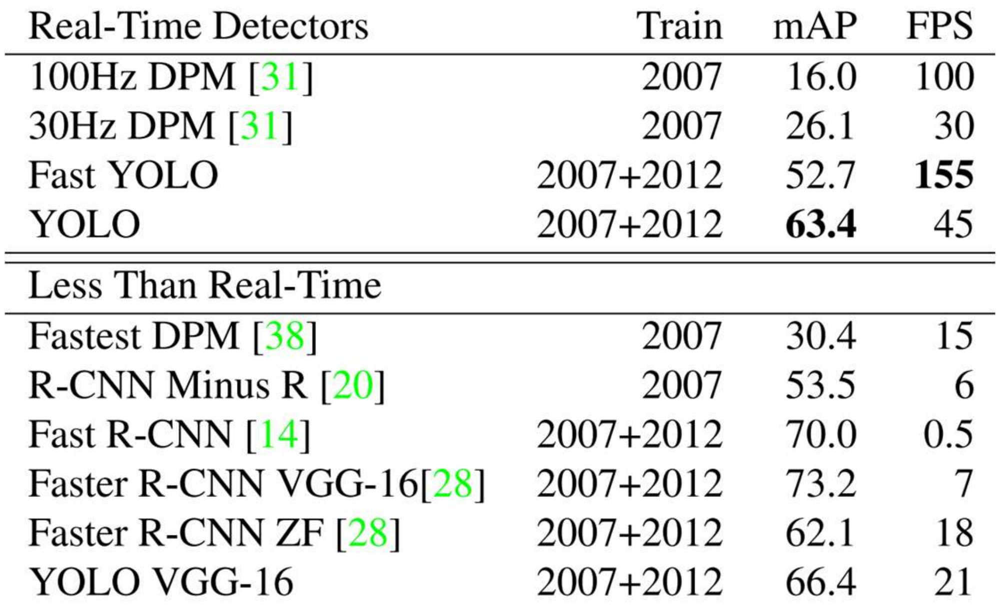


- Grid
  - Maximum Detections = S * S
  - Problem of nearby object detection
    - Small objects that appear in groups
- Data
  - Struggle to generalize to object with unusual ratio
- Loss
  - Use same loss for small Bbox and large Bbox -> Localizing Error

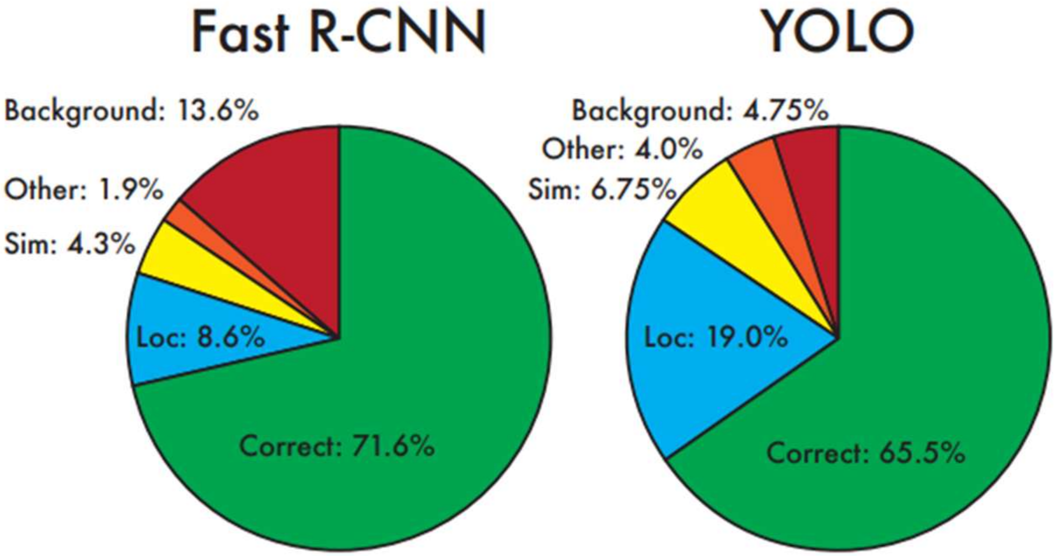
https://jonathan-hui.medium.com/real-time-object-detection-with-yolo-yolov2-28b1b93e2088

# Results

- Real Time
- Background Loss



| Real-Time Detectors | Train | mAP | FPS |
|---|---|---|---|
| 100Hz DPM [31] | 2007 | 16.0 | 100 |
| 30Hz DPM [31] | 2007 | 26.1 | 30 |
| Fast YOLO | 2007+2012 | 52.7 | **155** |
| YOLO | 2007+2012 | **63.4** | 45 |
| **Less Than Real-Time** | | | |
| Fastest DPM [38] | 2007 | 30.4 | 15 |
| R-CNN Minus R [20] | 2007 | 53.5 | 6 |
| Fast R-CNN [14] | 2007+2012 | 70.0 | 0.5 |
| Faster R-CNN VGG-16[28] | 2007+2012 | 73.2 | 7 |
| Faster R-CNN ZF [28] | 2007+2012 | 62.1 | 18 |
| YOLO VGG-16 | 2007+2012 | 66.4 | 21 |


Fast R-CNN
Background: 13.6%
Other: 1.9%
Sim: 4.3%
Loc: 8.6%
Correct: 71.6%
YOLO
Background: 4.75%
Other: 4.0%
Sim: 6.75%
Loc: 19.0%
Correct: 65.5%