II. One Stage Detection

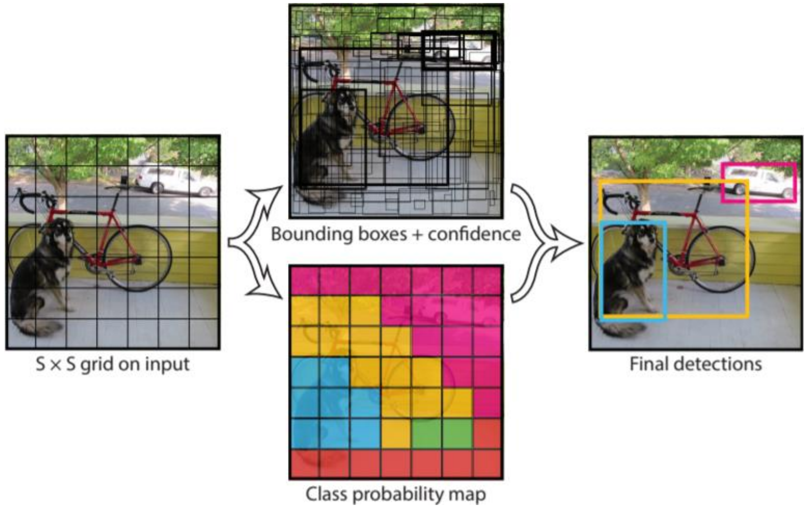
II. One Stage Detection

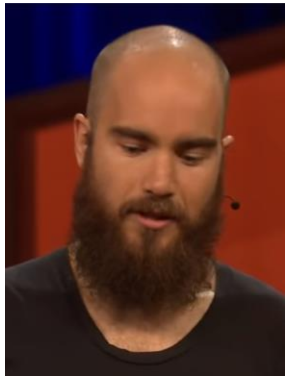
Why we have to train proposal first?

II. One Stage Detection

E. Yolo V1 [2015, Joseph]

E0. You Only Look Once!



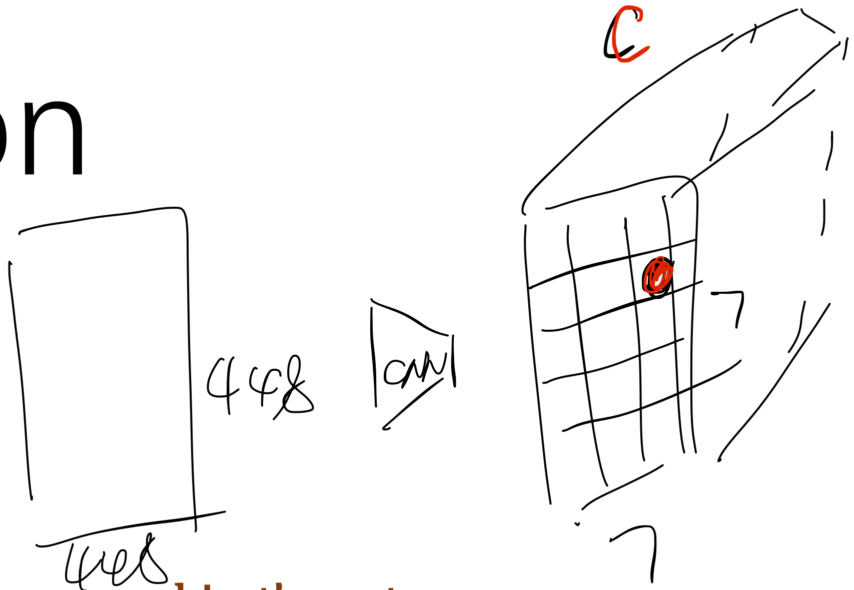


- Really fast (18 faster rcnn [ZF] vs 45 yolo)
- Becoming prevalent (62.1 mAP vs 63.4)

II. One Stage Detection

E. Yolo V1 [2015, Joseph]

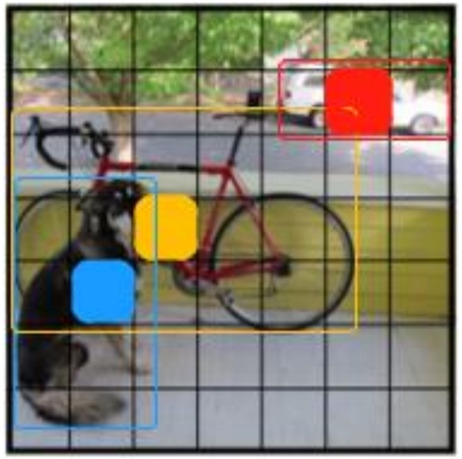
E1. Procedure



- Grid an image into S x S cells [448 x 448 -> 7 x 7]

  One cell will be responsible for predicting an object as long as an object's center locating in that cell.

S × S grid on input



S × S grid on input

II. One Stage Detection

E. Yolo V1 [2015, Joseph]

E1. Procedure

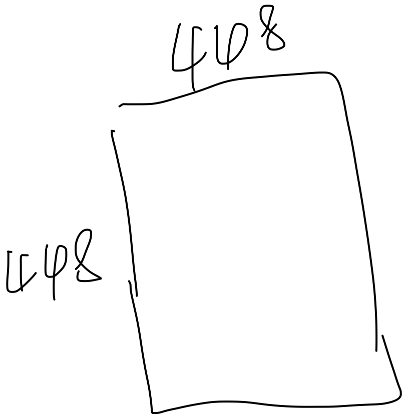
- Each cell predicts **B bounding box** with a **confidence**

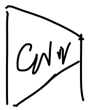
Bounding Box: x, y, w, h (center)

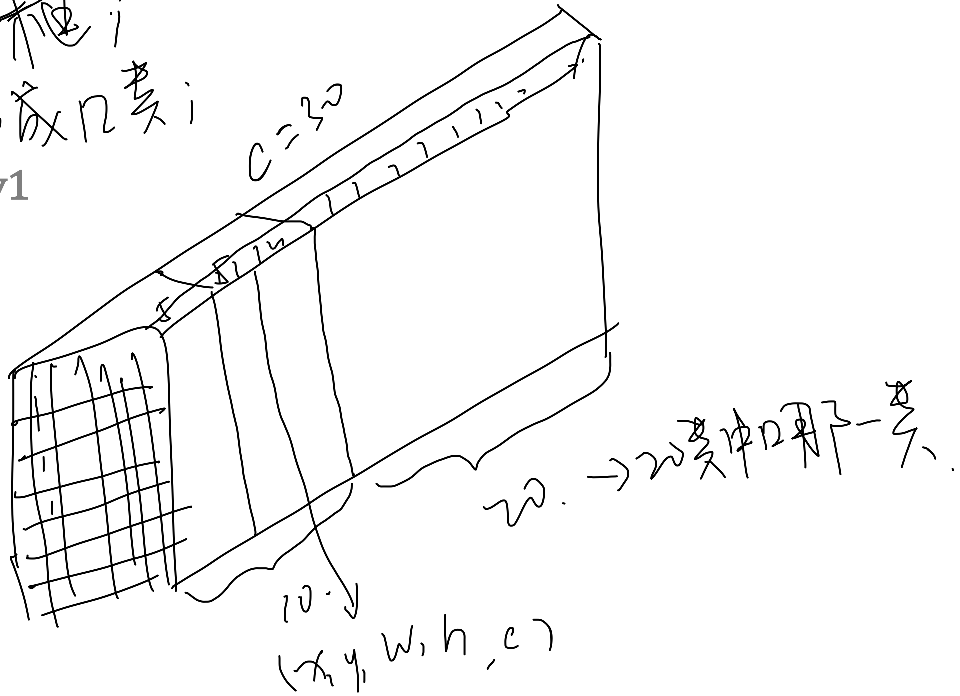
Confidence: $P_r(object) \cdot IoU^{truth}_{pred}$ → 多少个框;

Final output tensor: S x S x (5 * B + C) → 分成几类;

[7 x 7 x (5 * 2 + 20)] -> v1







II. One Stage Detection

E. Yolo V1 [2015, Joseph]

E2. Loss Function

正样本

2.1

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

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

负样本

2.2

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2 \quad (3)$$

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2 \quad (4)$$

正样本类别

2.3

$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \quad (5)$$

$i: 0 \sim (S^2 - 1)$ [iterate each grid cell (0~48)]

$j: 0 \sim (B - 1)$ [iterate each bbox (0~2)]

$\mathbb{1}_{ij}^{\text{obj}}$ & $\mathbb{1}_{ij}^{\text{noobj}}$:

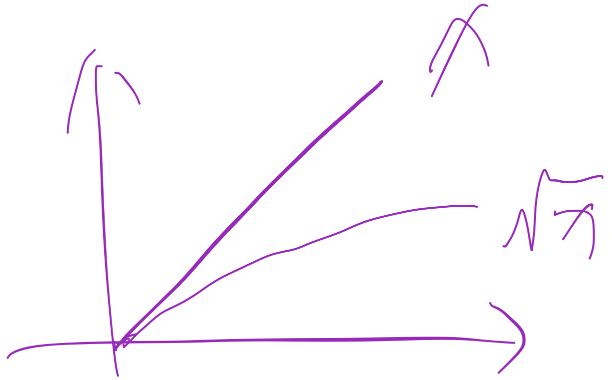
0	0	0	0	0	0	0
0	0	0	0	0	1	0
0	0	0	0	0	0	0
0	0	1	0	0	0	0
0	1	0	0	0	0	0
0	0	0	0	0	0	0
0	0	0	0	0	0	0

1	1	1	1	1	1	1
1	1	1	1	1	0	1
1	1	1	1	1	1	1
1	1	0	1	1	1	1
1	0	1	1	1	1	0
1	1	1	1	1	1	1
1	1	1	1	1	1	1

For $\mathbb{1}_{ij}^{\text{obj}}$, we have B predictions in each cell, only the one with largest IoU shall be labeled as 1.

II. One Stage Detection

E. Yolo V1 [2015, Joseph]



E2. Loss Function

E2.1: Coordinate loss

加权 5.

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

2.1

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

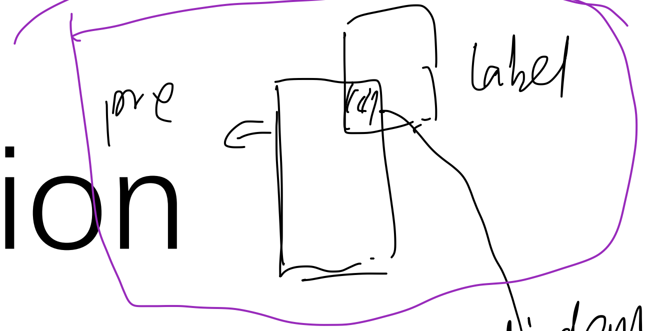
减缓大物体的效果

x, y: predicated bbox center,
w, h: predicated bbox width & height

$\hat{x}, \hat{y}$: labeled bbox center,
$\hat{w}, \hat{h}$: labeled bbox width & height

$\sqrt{w}, \sqrt{h}$: Suppress the effect for larger bbox

λ_{coord}: 5, because there's only 8 dimensions. Too less comparing to other losses
Weighted loss essentially.



II. One Stage Detection

E. Yolo V1 [2015, Joseph]

E2. Loss Function

E2.2: Confidence loss

前景/背景 confidence.

2.2

$$+\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{\text{obj}}\left(C_i-\hat{C}_i\right)^2 \quad (3)$$

$$+\lambda_{\text{noobj}}\sum_{i=0}^{S^2}\sum_{j=0}^{B}\mathbb{1}_{ij}^{\text{noobj}}\left(C_i-\hat{C}_i\right)^2 \quad (4)$$

0.5

$\widehat{C}_i$: confidence score [IoU] of predicated and labeled bbox

C_i : predicated confidence score [IoU] generated from networks

Note: $\widehat{C}_i$ in (4) is 0

λ_{noobj}=0.5, because there's so many non-object bboxes

Train: $confidence = P_r(object) \cdot IoU_{pred}^{truth}$

$\mathbb{1}_{ij}^{\text{obj}}$ $\widehat{C}_i$

classification

Test: Individual box confidence prediction:

$confidence = P_r(cls_i/obj)P_r(obj) \cdot IoU_{pred}^{truth} = \underline{P_r(cls_i)} \cdot IoU_{pred}^{truth}$

II. One Stage Detection

E. Yolo V1 [2015, Joseph]

E2. Loss Function

E2.3: Classification Loss

Each sell will only predict 1 object, which is decided by the bbox with largest IoU

$$2.3 \quad + \sum_{i=0}^{S^2} \mathbb{1}_i^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c)\right)^2 \quad (5)$$

* Don't forget to do NMS after generating bboxes

II. One Stage Detection

E. Yolo V1 [2015, Joseph]

E3. Pros & Cons

E2.3: Classification Loss

Pros:

- One stage, really fast 速度快.

Cons:

- Bad for crowed objects [1 cell 1 obj] 多个物体落在一个格子里.
- Bad for small objects 对小物体不友好.
- Bad for objects with new width-height ratio
- No BN

II. One Stage Detection

F. Yolo V2 [2016, Joseph]

F1. Improvements comparing to V1

F1.1: Add BN

F1.2: High Resolution Classifier [Focusing on backbone]

a. Train on ImageNet (224 x 224) // Model trained on small images may not be good
b. Resize & Finetune on ImageNet (448 x448) // So we finetune the model on larger images
c. Finetune on dataset // To let the model be used to larger images
d. We get 13 x 13 feature maps finally

F1.3: We Use Anchors [We'll talk it later]

II. One Stage Detection

F. Yolo V2 [2016, Joseph]

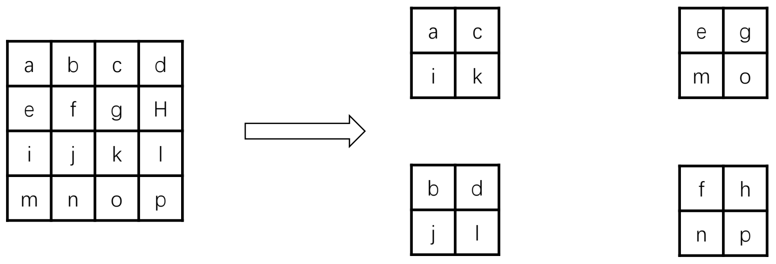
F1. Improvements comparing to V1

F1.4: Fine-Grained Features

a. Lower features are concatenated directly to higher features

b. A new layer is added for that purpose: reorg

拆成4小图.



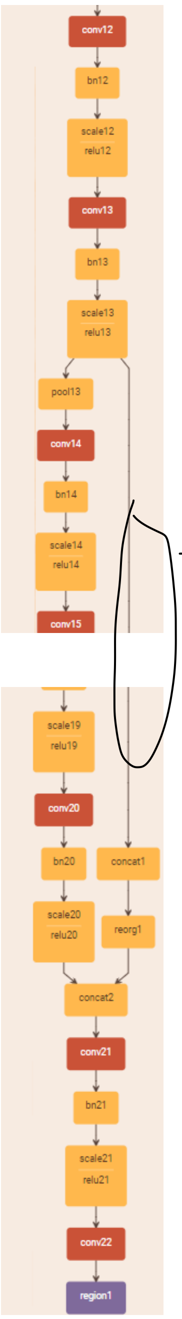
F1.5: Multi-Scale Training

- Remove FC layers: Can accept any size of inputs, enhance model robustness.
- Size across 320, 352, ..., 608. Change per 10 epochs
  [border % 32 = 0, decided by down sampling]

localization
偏
物理

semantic
info.
(语义)

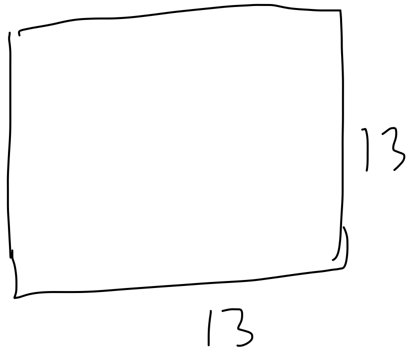
short
cut.



II. One Stage Detection

F. Yolo V2 [2016, Joseph]

F2. Anchor in Yolo V2



F2.1: Anchor size and number

a. Faster RCNN: 9 by hands → 数学推导.

b. Yolo: 5 by K-Means [dist: 1 – iou(bbox, cluster)] → 聚类

F2.2: Anchors, Truth BBoxes & Predicted BBoxes

- Anchors: 0.57273, 0.677385, ..., 9.77052, 9.16828 [10 numbers]

  a_{w_0}, a_{h_0}, ..., a_{w_4}, a_{h_4}

  $$\text{anchors}[0] = a_{w_i} = \frac{a_{w_{ori}}}{W} * 13$$ [not strict, just aiming to say how we get those numbers]

- Truth Bboxes:

II. One Stage Detection

F. Yolo V2 [2016, Joseph]

F2. Anchor in Yolo V2

F2.2: Anchors, Truth BBoxes & Predicted BBoxes

- **Anchors:** anchors[0] $= a_{w_i} = \frac{a_{w_{ori}}}{W} * 13$ [not strict, just aiming to say how we get those numbers]
- **Truth Bboxes:**

original bbox: $[x_o, y_o, w_o, h_o] \epsilon [0, W|H]$

normalize in 0~1

normalized original bbox: $[x_r, y_r, w_r, h_r] \epsilon [0,1]$

$[x_r, y_r, w_r, h_r] = [x_o/W, y_o/H, w_o/W, h_o/H]$

transfer to feature map size: 13 x 13

box: $[x, y, w, h] \epsilon (0,13]$

$[x, y, w, h] = [x_r, y_r, w_r, h_r] * (13|13)$

save this for calculating

transfer to 0~1 corresponding to each grid cell

final box: $[x_f, y_f, w_f, h_f] \epsilon [0,1]$

save this for calculating

transfer to 0~1 corresponding to each grid cell

final box: $[x_f, y_f, w_f, h_f] \epsilon [0,1]$

$$\begin{cases} x_f = x - i \\ y_f = y - j \\ w_f = \log(w/anchors[0]) \\ h_f = \log(h/anchors[1]) \end{cases}$$

$10.6 - 10 = 0.6$

II. One Stage Detection

F. Yolo V2 [2016, Joseph]

F2. Anchor in Yolo V2

F2.2: Anchors, Truth BBoxes & Predicted BBoxes

- Output of Yolo V2: features[0:25]

$$
\begin{cases}
0,1: x, y \\
2,3: w, h \\
4: Confidence\ [IoU] \\
5\sim := \Pr(class/Obj)
\end{cases}
$$

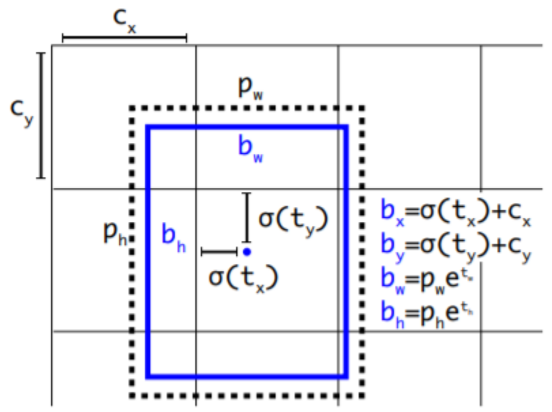


Figure 3: Bounding boxes with dimension priors and location prediction. We predict the width and height of the box as offsets from cluster centroids. We predict the center coordinates of the box relative to the location of filter application using a sigmoid function.

- Code:

```
box_xy = K.sigmoid(feats[..., :2])  → 存log
box_wh = K.exp(feats[..., 2:4])
box_confidence = K.sigmoid(feats[..., 4:5])  → [0, 1]
box_class_probs = K.softmax(feats[..., 5:])  → [0, 1]

# Adjust preditions to each spatial grid point and anchor size.
# Note: YOLO iterates over height index before width index.
box_xy = (box_xy + conv_index) / conv_dims
box_wh = box_wh * anchors_tensor / conv_dims
```

II. One Stage Detection

F. Yolo V2 [2016, Joseph]

F3. Problems

➢ Better for small & crowded objects.

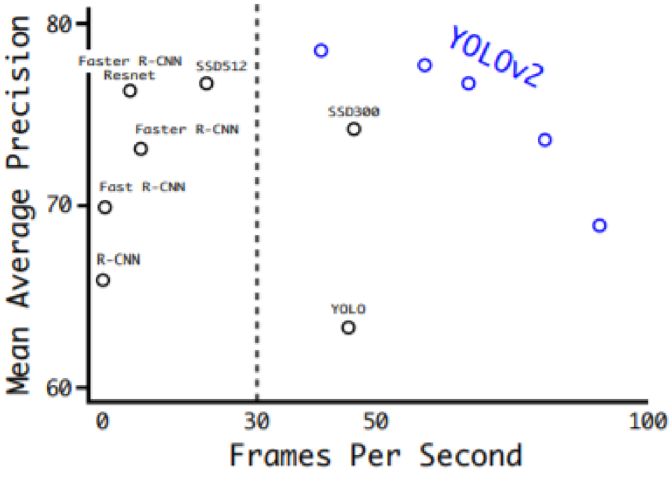


Figure 4: Accuracy and speed on VOC 2007.

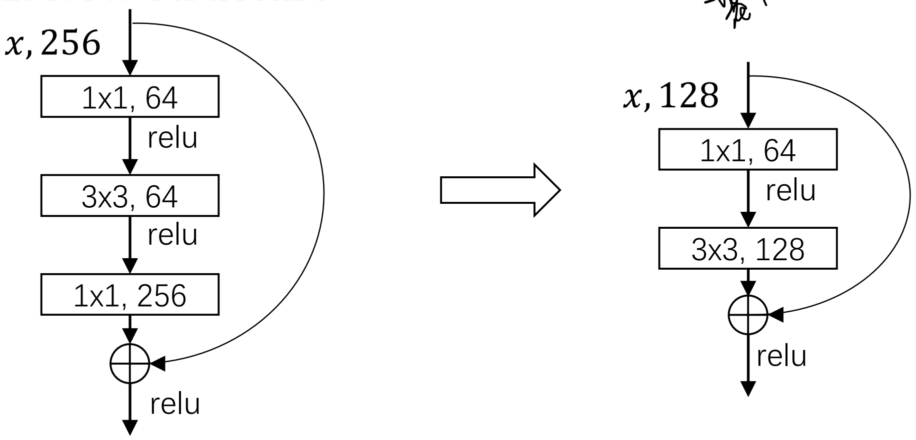
II. One Stage Detection

G. Yolo V3 [2018, Joseph]

G1. Improvements

G1.1: New structure

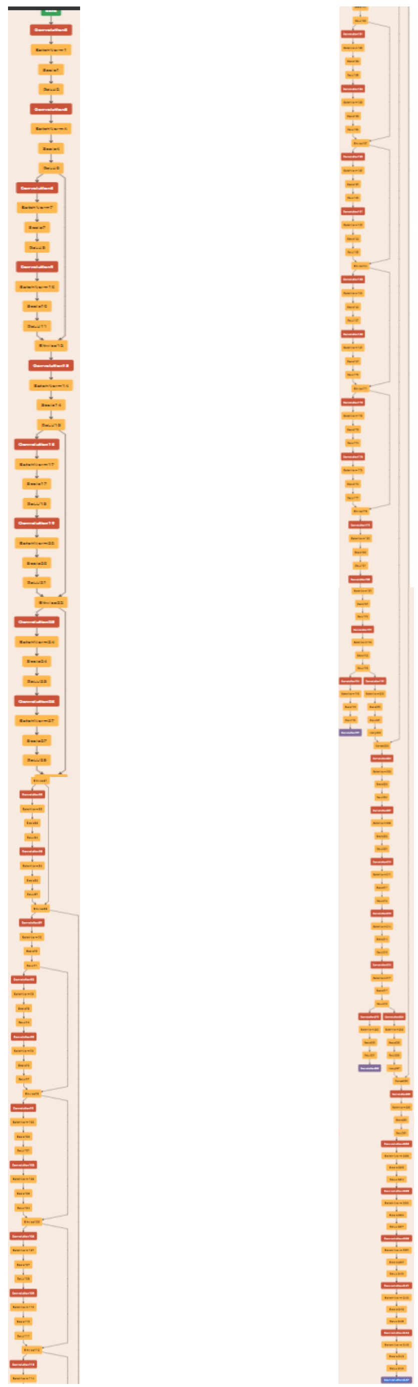
整个网络更深了.



G1.2: Multiscale Structure

- 3 scales; 3 anchors per scale per grid
- /32, small scale (13 x 13) —> large anchor
- /16, mid scale (26 x 26) —> medium anchor
- /8, large scale (52 x 52) —> small anchor

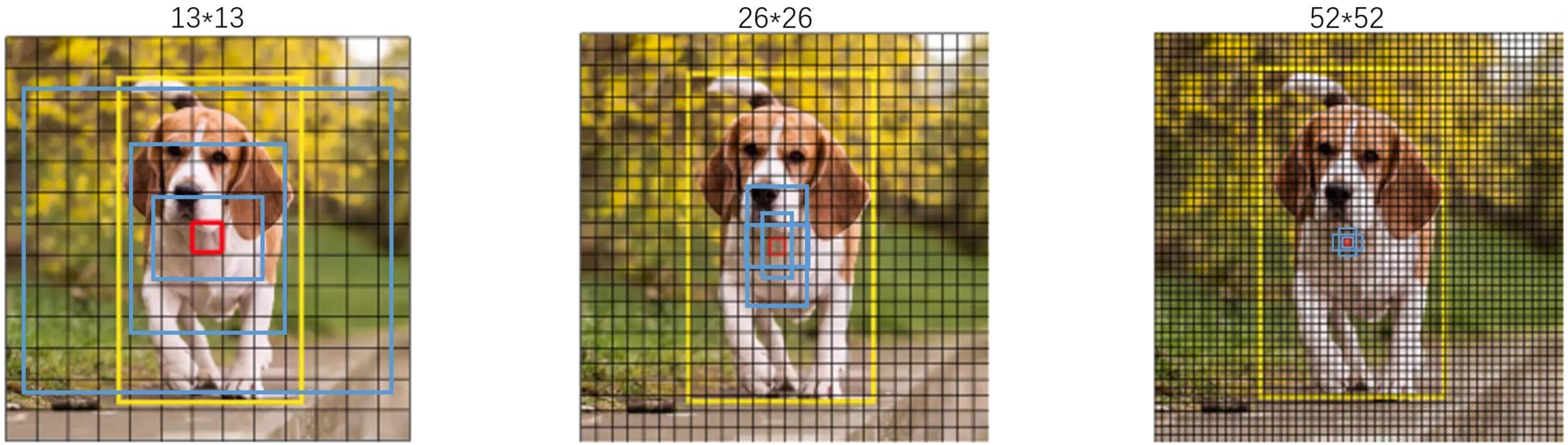
3×3=9个



II. One Stage Detection

G. Yolo V3 [2018, Joseph]

G1. Improvements



G1.3: Change Classification :

- 80 classes, from softmax —> logistic

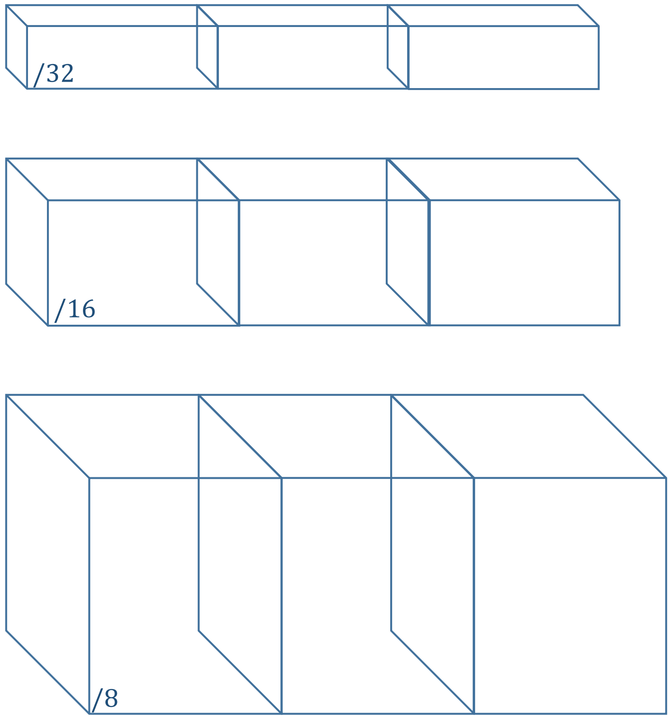
一个物体可能满足多个类别，
都是二分类，对每个标签都可以区分为是或否.

II. One Stage Detection

G. Yolo V3 [2018, Joseph]

G2. Summary

G2.1: Output



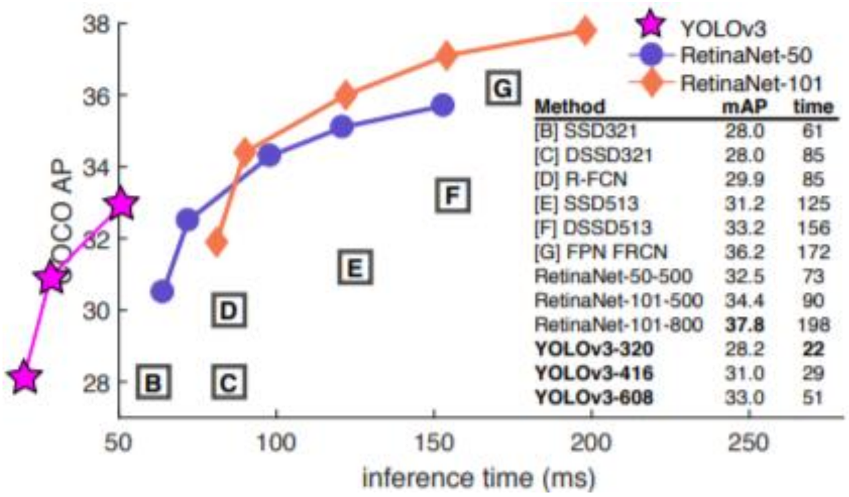
3个 scale 框

$13*13*3$

$26*26*3$ $\quad *(4+1+80)$

$52*52*3$

F2.2: Performance

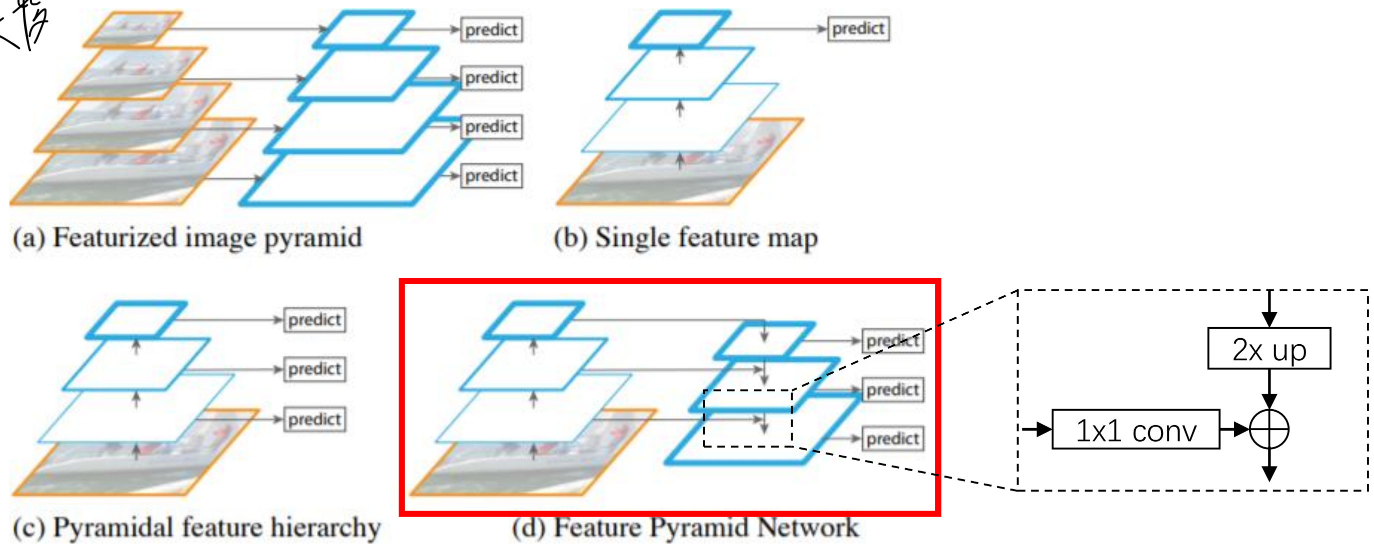


II. One Stage Detection

G. Yolo V3 [2018, Joseph]

G3. FPN Net [2017, Lin]

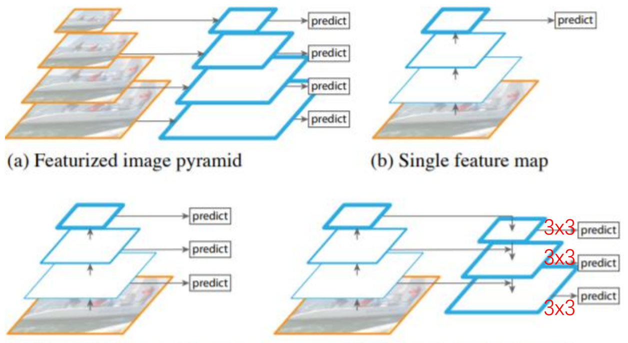
输入多尺度



II. One Stage Detection

G. Yolo V3 [2018, Joseph]

G3. FPN Net [2017, Lin]



(a) Featurized image pyramid (b) Single feature map

(c) Pyramidal feature hierarchy (d) Feature Pyramid Network

➢ **Pros:**

1. Lower layers have accurate localization info; 融合高层,低层
   Higher layers have ample semantic info
   FPN combine them together. Like U-Net

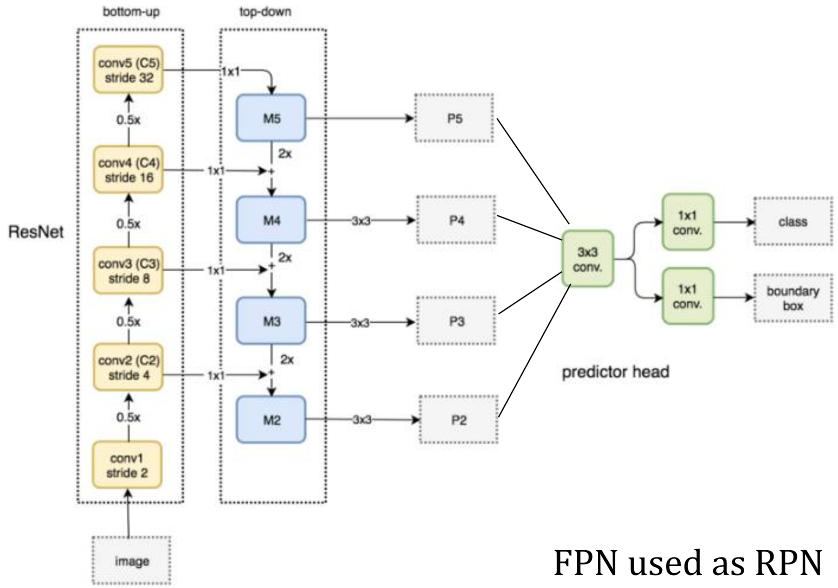
2. Feature detector: worked as a part in a whole bigger network.
   模块化.

➢ **Details:**

1. 3x3convolution is appended on each merged map to generate the final feature map to reduce the aliasing effect

2. Feature dimension at output is 256.

3. When used in FPN. P2-P6. Anchor areas: $32^2, \sim, 512^2$ per scale, with ratio {1:2,1:1,2:1}, 15 in total. And, predictor head is shared.



FPN used as RPN

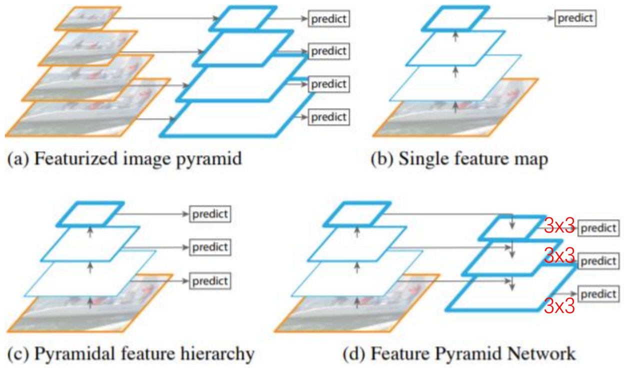
II. One Stage Detection

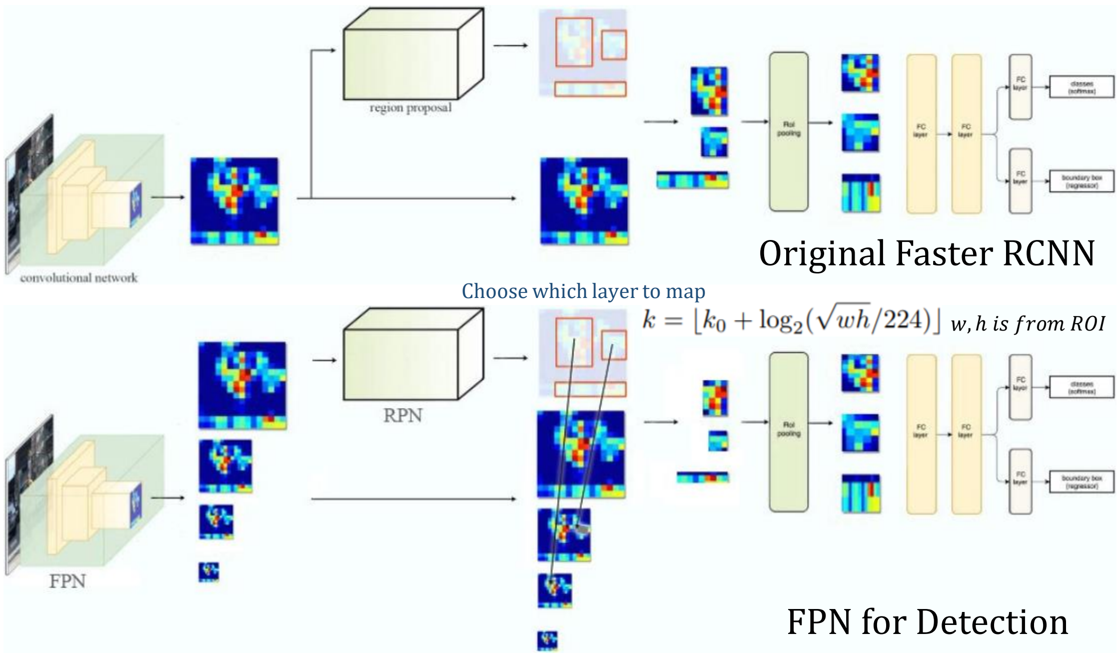
G. Yolo V3 [2018, Joseph]

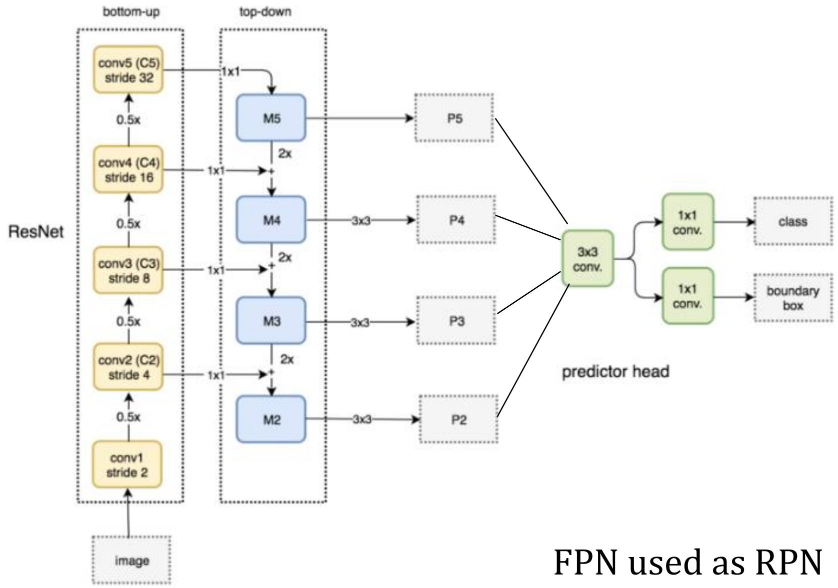
G3. FPN Net [2017, Lin]

- **Details:**

  4. When used in faster rcnn:







II. One Stage Detection

H. RetinaNet [2018, Lin]

$$FL = \begin{cases} -\alpha(1-\hat{y})^{\gamma}\log\hat{y} & \text{当 } y=1 \\ -(1-\alpha)\hat{y}^{\gamma}\log(1-\hat{y}) & y=0 \end{cases}$$

$$\Downarrow$$

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

➢ **Structure:**

Resnet + FPN + FCN :

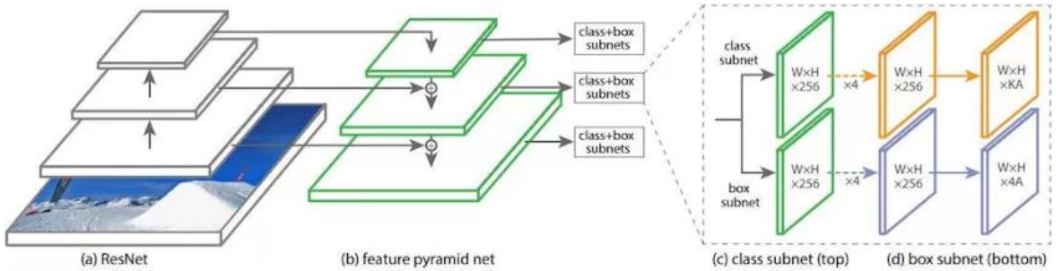
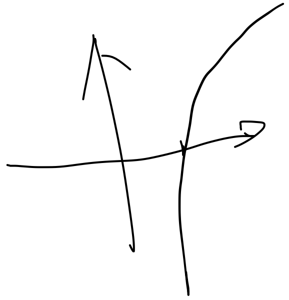


Figure 3. The one-stage **RetinaNet** network architecture uses a Feature Pyramid Network (FPN) [20] backbone on top of a feedforward ResNet architecture [16] (a) to generate a rich, multi-scale convolutional feature pyramid (b). To this backbone RetinaNet attaches two subnetworks, one for classifying anchor boxes (c) and one for regressing from anchor boxes to ground-truth object boxes (d). The network design is intentionally simple, which enables this work to focus on a novel focal loss function that eliminates the accuracy gap between our one-stage detector and state-of-the-art two-stage detectors like Faster R-CNN with FPN [20] while running at faster speeds.



➢ **Focal Loss:**

Q. Why one-stage performs worse than two stage?

A. 1. Because neg/pos samples are extremely unbalanced ✓

2. Gradient is dominated by easy samples. ✓

S. We can use FOCAL LOSS to solve it.

多分类：

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

II. One Stage Detection

I. Some Resources

- I1: [Yolo V1]
- I2: [Yolo V2]
- I3: [Yolo V3] / [From scratch]

How to implement a YOLO (v3) object detector from scratch in PyTorch: Part 1

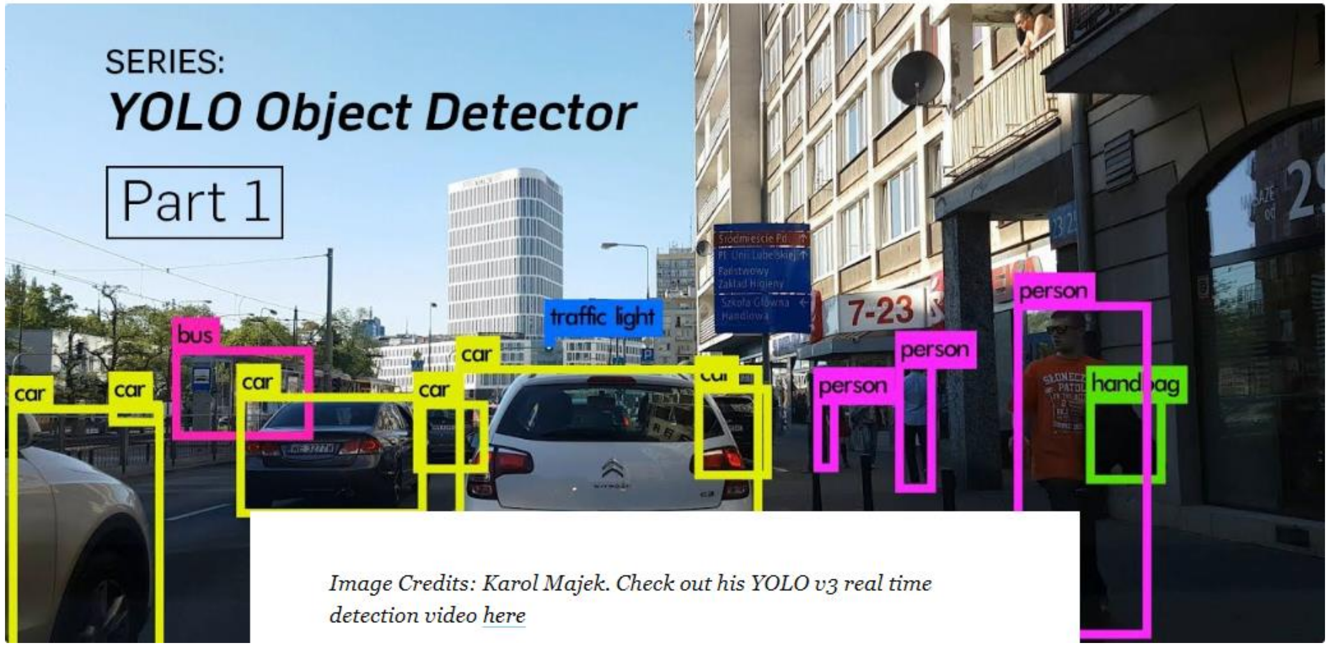


Image Credits: Karol Majek. Check out his YOLO v3 real time detection video here

III. Other Methods

J. Other Method

➢ SSD Series: (2015, Liu, another one-stage, also popular!)

- **Methods based on SSD**
- **FaceBoxes / Github [Contents we'll cover in Face Detection. Better to pre-study]**