# Papers On Object Detection One-Stage

Gong Qiqi

# Paper List

- 5. You Only Look Once: Unified, Real-Time Object Detection (**YOLO**)
- 6. **SSD**: Single Shot MultiBox Detector
- 7. Focal Loss for Dense Object Detection

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

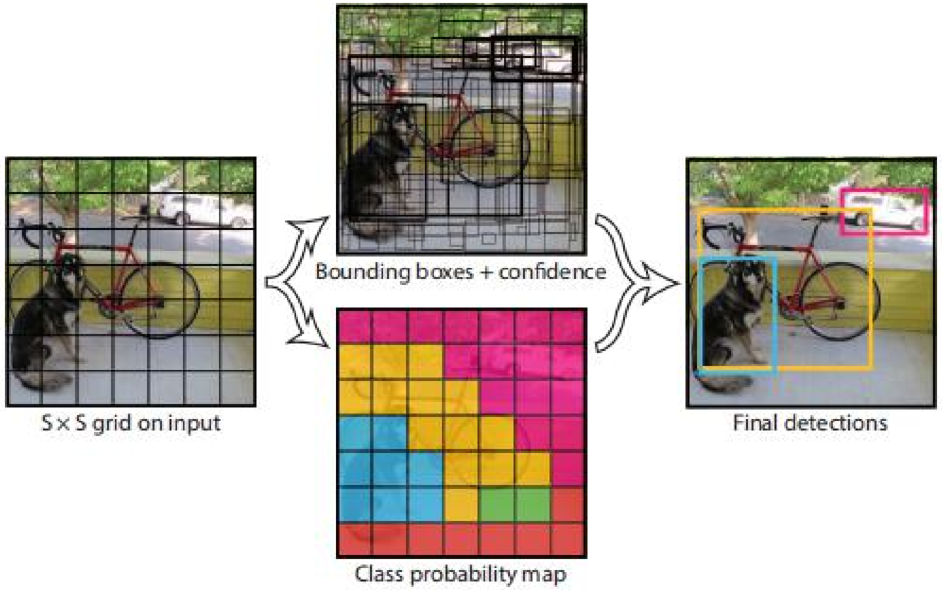
Author: Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi

# YOLO

- Basic Info.:
  - 2016 CVPR
- Motivation:
  - Two-stage  methods are slow and hard to optimize
- Contribution:
  - First to put forward one-stage detection method
  - Deal with detection problem as a **regression problem**
  - Bridge domain gap
- Drawbacks:
  - Localization errors (but less likely to mistake categoery)

# YOLO

- Overview:
  - Divides the input image into an S × S grid
  - Each grid cell predicts B bounding boxes and **confidence scores** for those boxes.
    $\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$.
  - Each bounding box consists of 5 predictions: x, y, w, h, and confidence.
    - x,y: center of the box relative to the bounds of grid cell
    - w,h: width and height relative to the whole image
  - Each grid cell also predicts C conditional class probabilities




These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.
VOC: S=7, B=2, C=20

# YOLO

- Network:

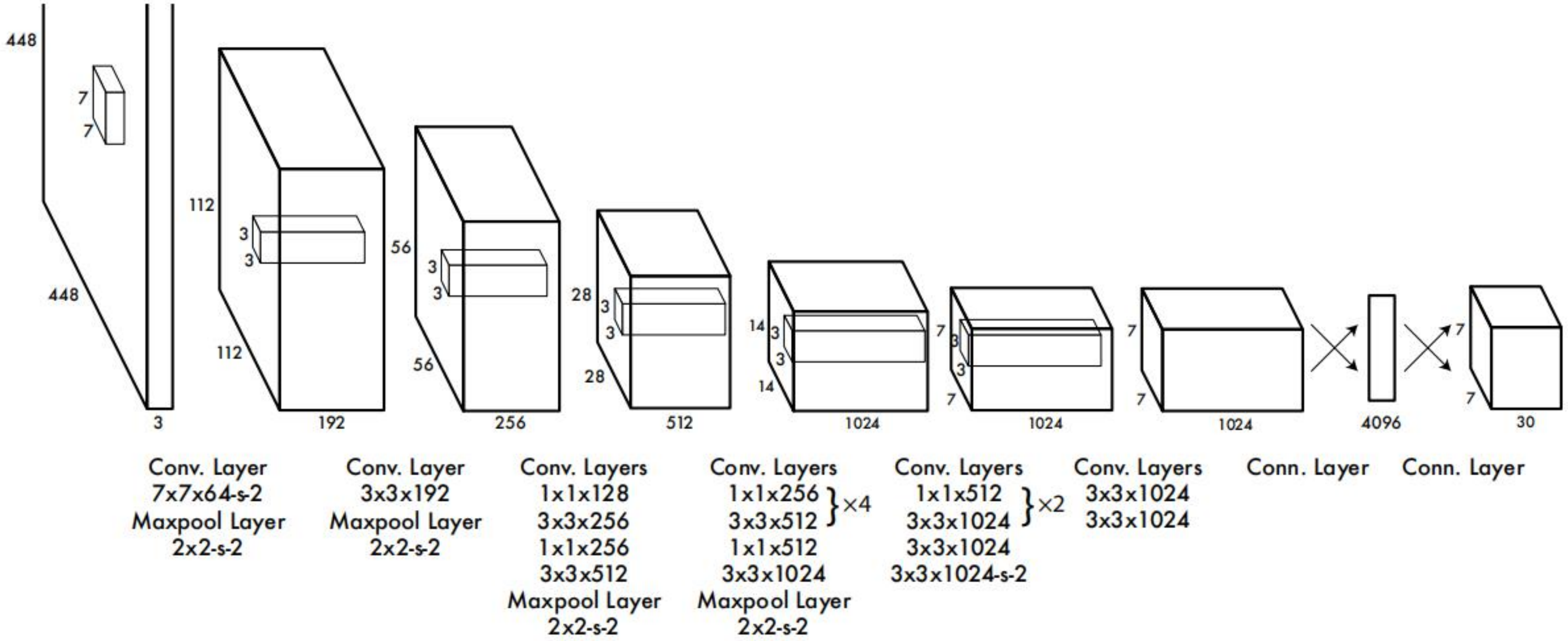



**Figure 3: The Architecture.** Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating $1 \times 1$ convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution ($224 \times 224$ input image) and then double the resolution for detection.

# YOLO

- Training:
  - Pretrain convolutional layers on ImageNet (20conv + avepooling + 2fc)
  - Increase resolution from $224^2$ to $448^2$
  - Final layer predicts class posibilities and b-box position
- Loss:

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

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

cat.

$$
+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} (p_i(c) - \hat{p}_i(c))^2 \tag{3}
$$

where $\mathbb{1}_i^{\text{obj}}$ denotes if object appears in cell $i$ and $\mathbb{1}_{ij}^{\text{obj}}$ denotes that the $j$th bounding box predictor in cell $i$ is "responsible" for that prediction.

# YOLO

- Limitations:
  - Impose strong spatial constraints on bounding box predictions
  - Unsensitive to new aspect ratios
  - Treat small objects the same with big objects

# SSD: Single Shot MultiBox Detector

Author: Wei Liu, Dragomir Anguelov, Dumitru Erhan,
Christian Szegedy, Scott Reed

# SSD

- Basic Info.:
    - 2016 ECCV
- Motivation:
    - Two-stage methods are slow and hard to optimize
    - One-stage methods do not perform well
- Contribution:
    - Faster than Faster R-CNN and more accurate than YOLO

# SSD

- Trainging Method
  - $n$ priors(b-boxes with different aspect ratios) in total
  - Match with gt boxes first and asign lables to the rest of them according to IoU
  - Loss: $L(x,c,l,g) = L_{conf}(x,c) + \alpha L_{loc}(x,l,g),$ (1)

$$L_{loc}(x,l,g) = \frac{1}{2}\sum_{i,j} x_{ij}^p ||l_i - g_j^p||_2^2 \quad (2)$$

$$L_{conf}(x,c) = -\sum_{i,j,p} x_{ij}^p \log(c_i^p) - \sum_{i,p}(1 - \sum_{j,q=p} x_{ij}^q) \log(1 - c_i^p) \quad (3)$$

multi-class logistic loss

# SSD

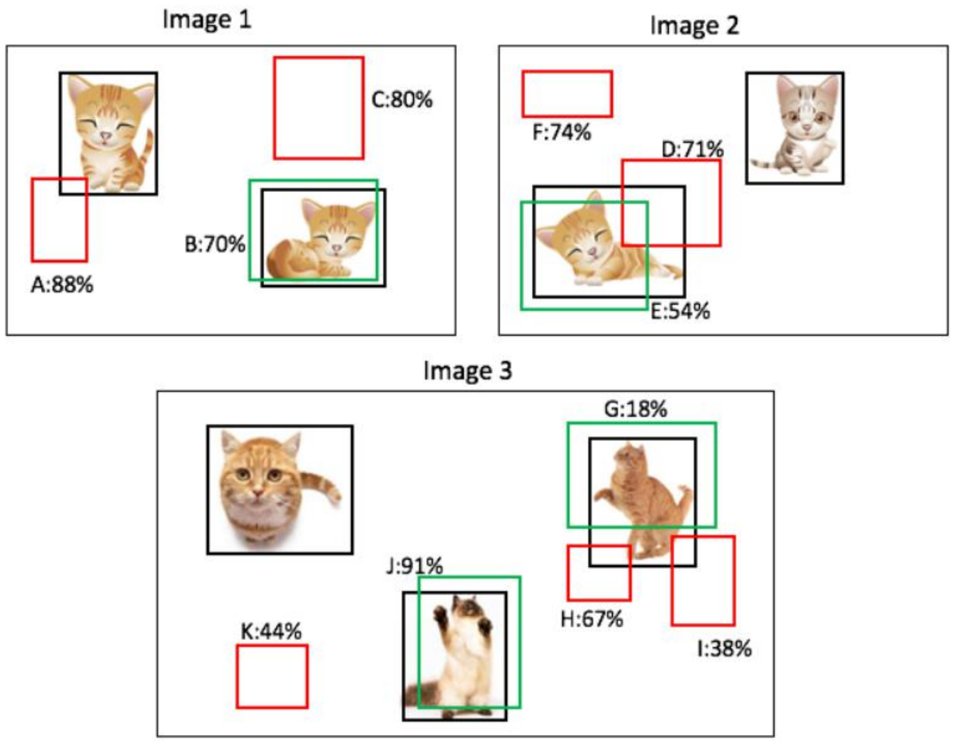
- Trainging Method
  - Loss: $L(x,c,l,g) = L_{conf}(x,c) + \alpha L_{loc}(x,l,g), \quad (1)$
    - Location Loss:

$$L_{loc}(x,l,g) = \frac{1}{2}\sum_{i,j} x_{ij}^{p} ||l_i - g_j^{p}||_2^2 \quad (2)$$

    - Confidence Loss:

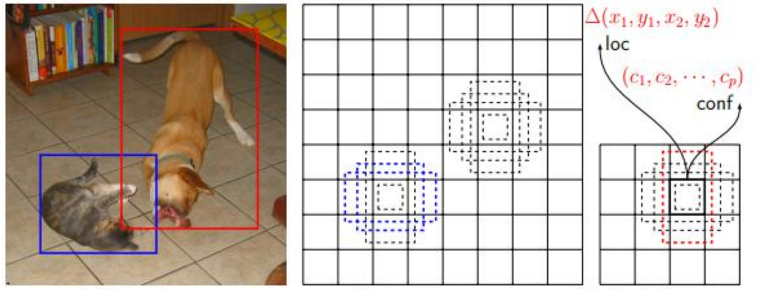
$$L_{conf}(x,c) = -\sum_{i,j,p} x_{ij}^{p} \log(c_i^{p}) - \sum_{i,p}(1 - \sum_{j,q=p} x_{ij}^{q}) \log(1 - c_i^{p}) \quad (3)$$

    - Hyper param: α=0.06



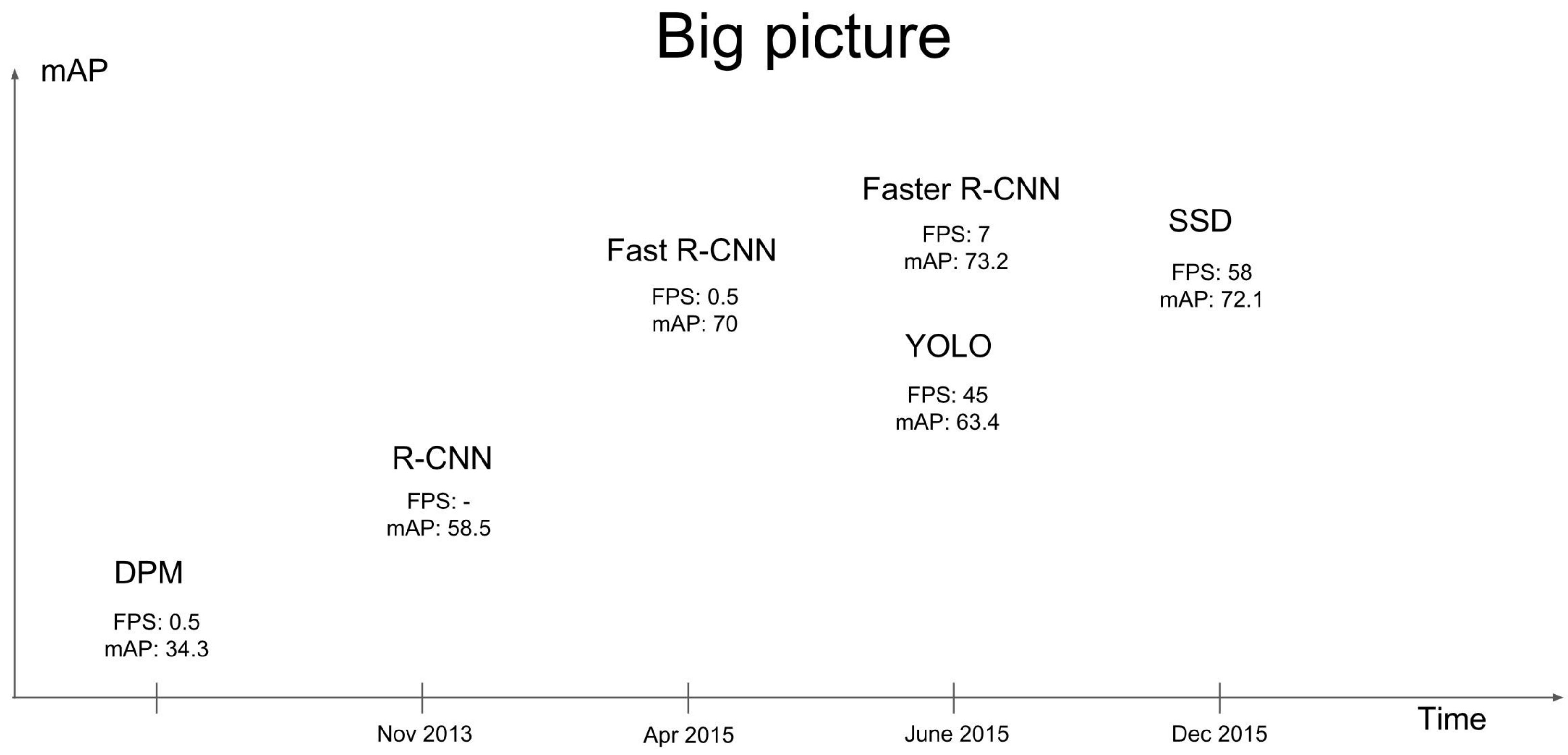
# SSD

- Fully Convolutional priors
  - Similar to RPN, but use 1*1 kernels to predict **offsets** and **confidence**
  - Suppose feat. map: $m*m$, $k$ priors per location
    - $(4+c)km^2$ variables in total
- Combine predictions from multiple feat. maps
  - Suppose $m$ feat. maps in total, $f_k$ is the k-th feat. map
  - scale (relative size) of priors: $s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m-1}(k-1),$
  - aspect ratios: $a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}.$



(a) Input image (b) 8 × 8 feature map (c) 4 × 4 feature map

# SSD


Big picture
mAP
Faster R-CNN
FPS: 7
mAP: 73.2
SSD
FPS: 58
mAP: 72.1
Fast R-CNN
FPS: 0.5
mAP: 70
YOLO
FPS: 45
mAP: 63.4
R-CNN
FPS: -
mAP: 58.5
DPM
FPS: 0.5
mAP: 34.3
Nov 2013
Apr 2015
June 2015
Dec 2015
Time

# Focal Loss for Dense Object Detection

Author: Tsung-Yi Lin， Priya Goyal， Ross Girshick
Kaiming He， Piotr Dollar

# Focal Loss

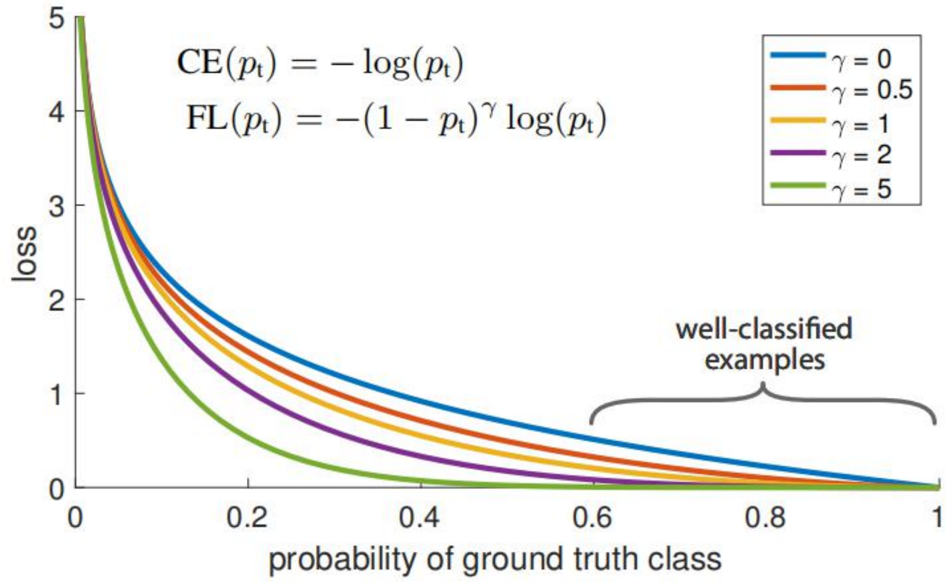
- Basic Info.:
  - 2017 ICCV
  - FAIR
- Motivation:
  - Accuracy-Speed tradoff is due to the imbalance between P&N example
- Contribution:
  - Put forward **Focal Loss**
  - A new detector: RetinaNet

# Focal Loss

- Background:
  - In two-stage method, class imbalance is addressed (by set ratios manually or OHEM)
  - In one-stage method, easy negative examples are overwhelming
- Cross Entropy Loss:
  - Take binary classification as eample

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases} \quad (1)$$

$$p_{\mathrm{t}} = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \quad (2)$$



# Focal Loss

- Balanced CE:

$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t). \qquad (3)$$

- Focal Loss:

focusing param.

$$\mathrm{FL}(p_t) = -\boxed{(1 - p_t)^{\gamma}} \log(p_t). \qquad (4)$$

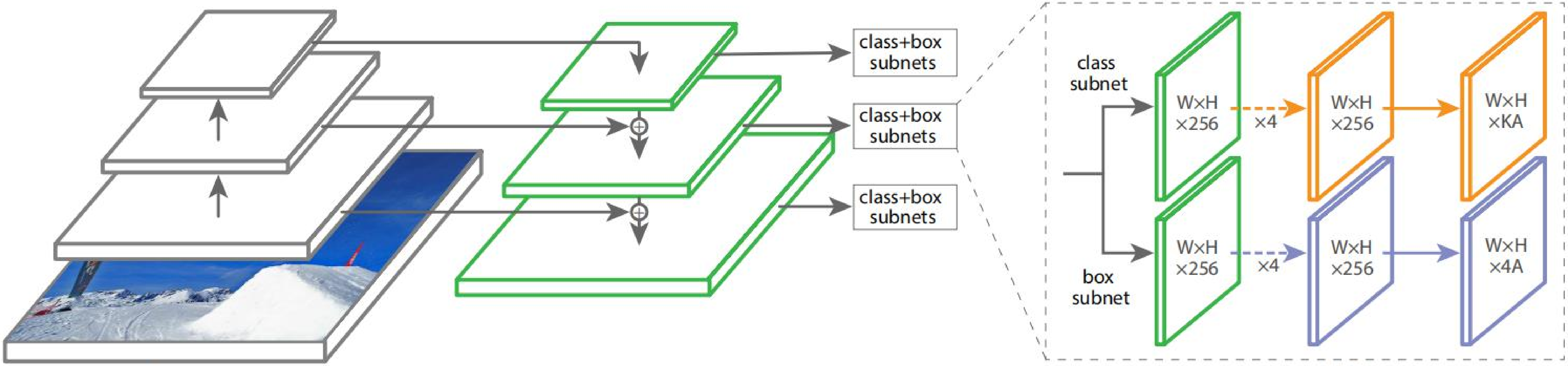
modulating factor

- Combine (3) and (4)

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t). \qquad (5)$$

# Focal Loss

- RetinaNet:



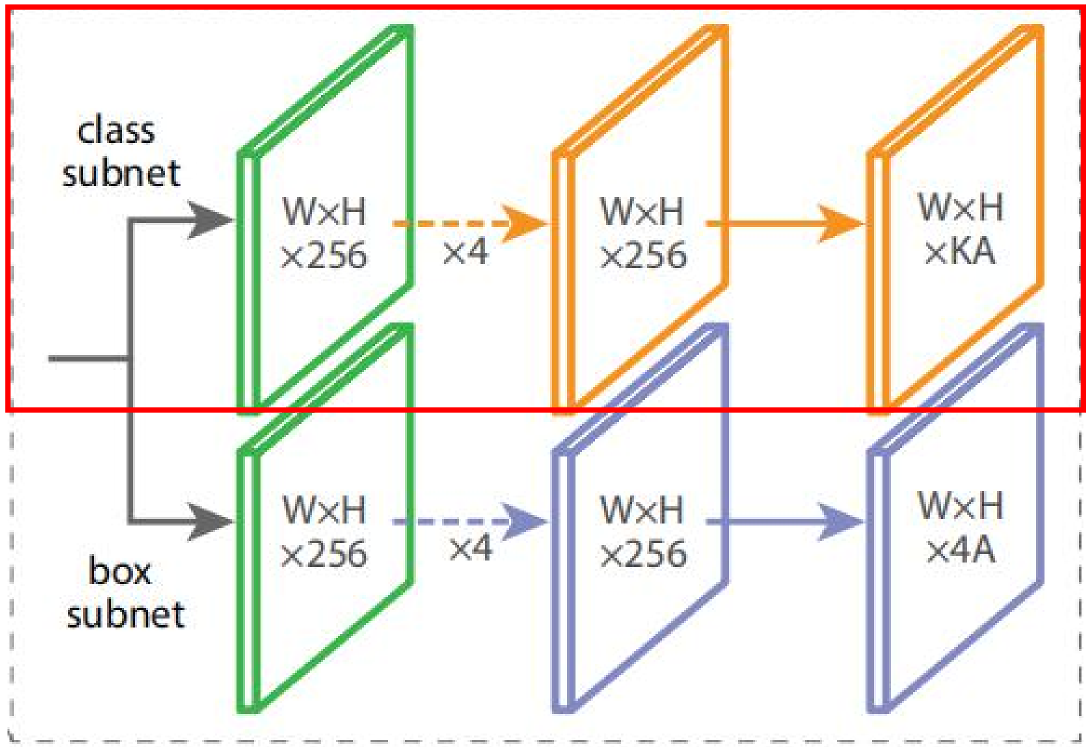
(a) ResNet (b) feature pyramid net (c) class subnet (top) (d) box subnet (bottom)

# Focal Loss

- RetinaNet:
  - Anchors:
    - Aspect ratios:{1:2, 1:1, 2:1}
    - Scale: $\{2^0, 2^{1/3}, 2^{2/3}\}$
    - IoU Threshold: 0.5; IoU BG:[0,0.4); Else: Ignored

# Focal Loss

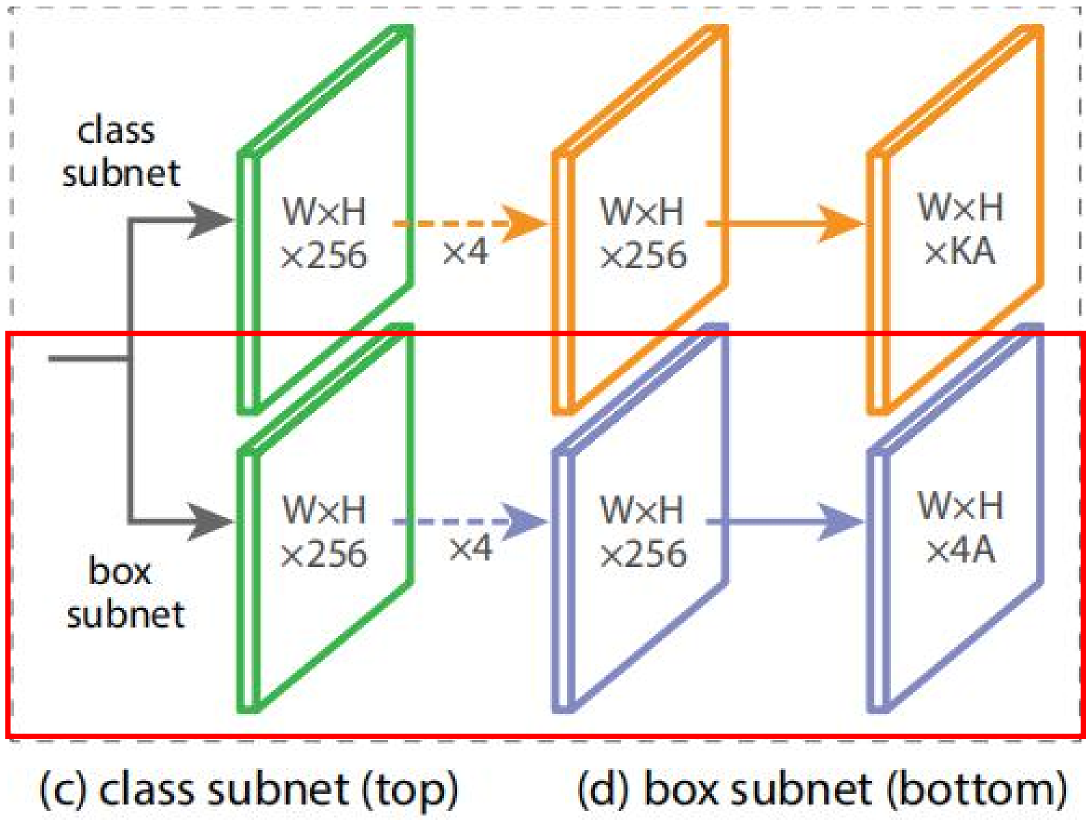
- RetinaNet:
  - Class subnet:
    - Small FCN attached to each pyramid level
    - Share weights
    - Output of FPN level: ***C***-channel; four *C**3*3 conv layers
    - Anchors: *A*



(c) class subnet (top) (d) box subnet (bottom)

# Focal Loss

- RetinaNet:
  - Regression subnet:
    - Regress offects to a nearby GT b-box4
    - Same as design of C-subnet (but end with 4A channels)
    - Class-agnostic* bounding box regressor (fewer params)
    - Separate params. from C-subnet




*class-agnostic: only differentiate BG & FG

# Thanks for Listening~

Gong Qiqi

# Reference List


- 1. *https://zhuanlan.zhihu.com/p/25236464/*, YOLO详解
- 2. *https://zhuanlan.zhihu.com/p/49981234*, Focal loss论文详解