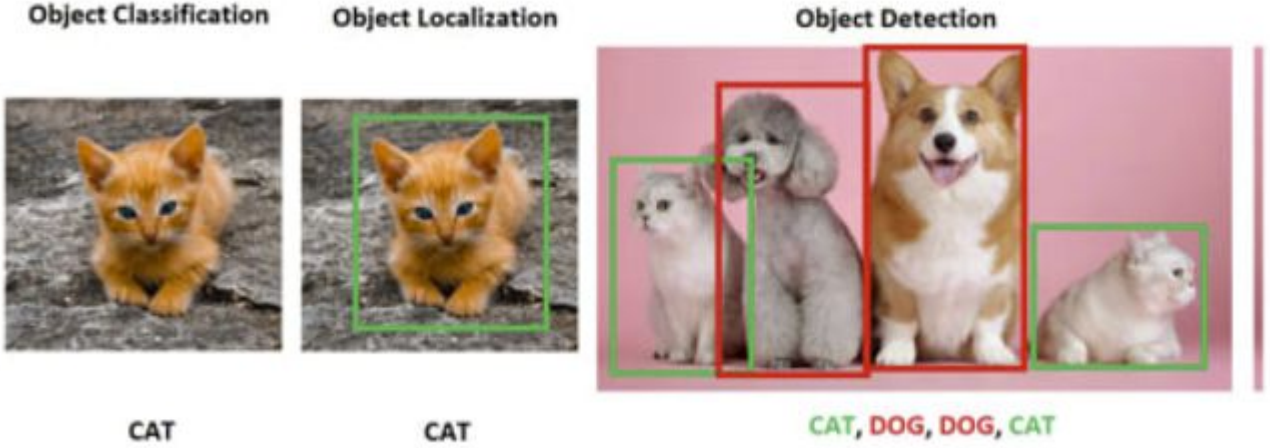
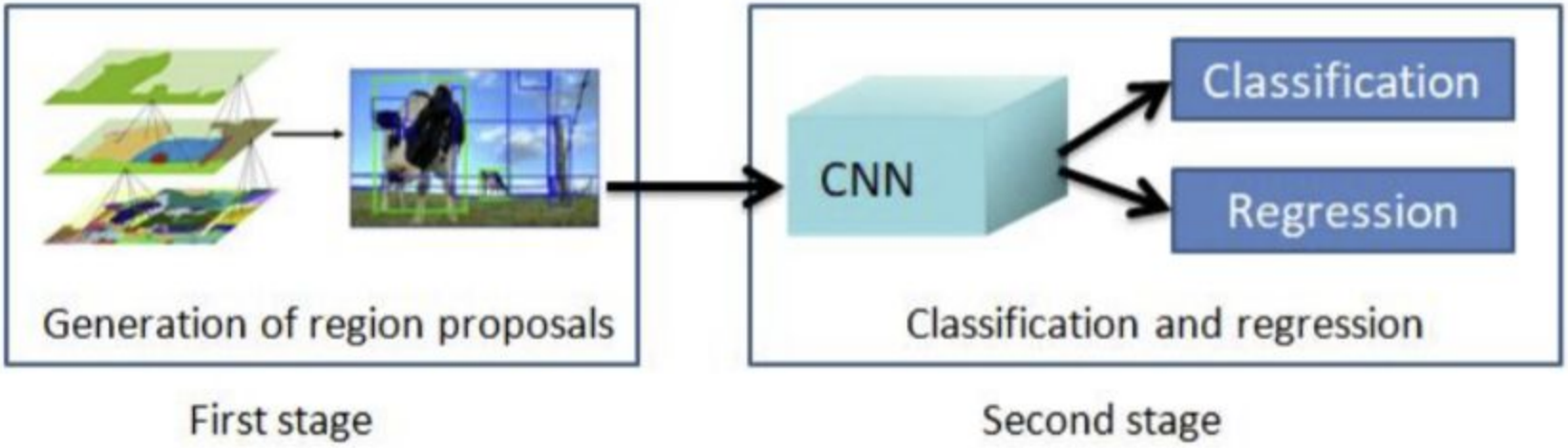
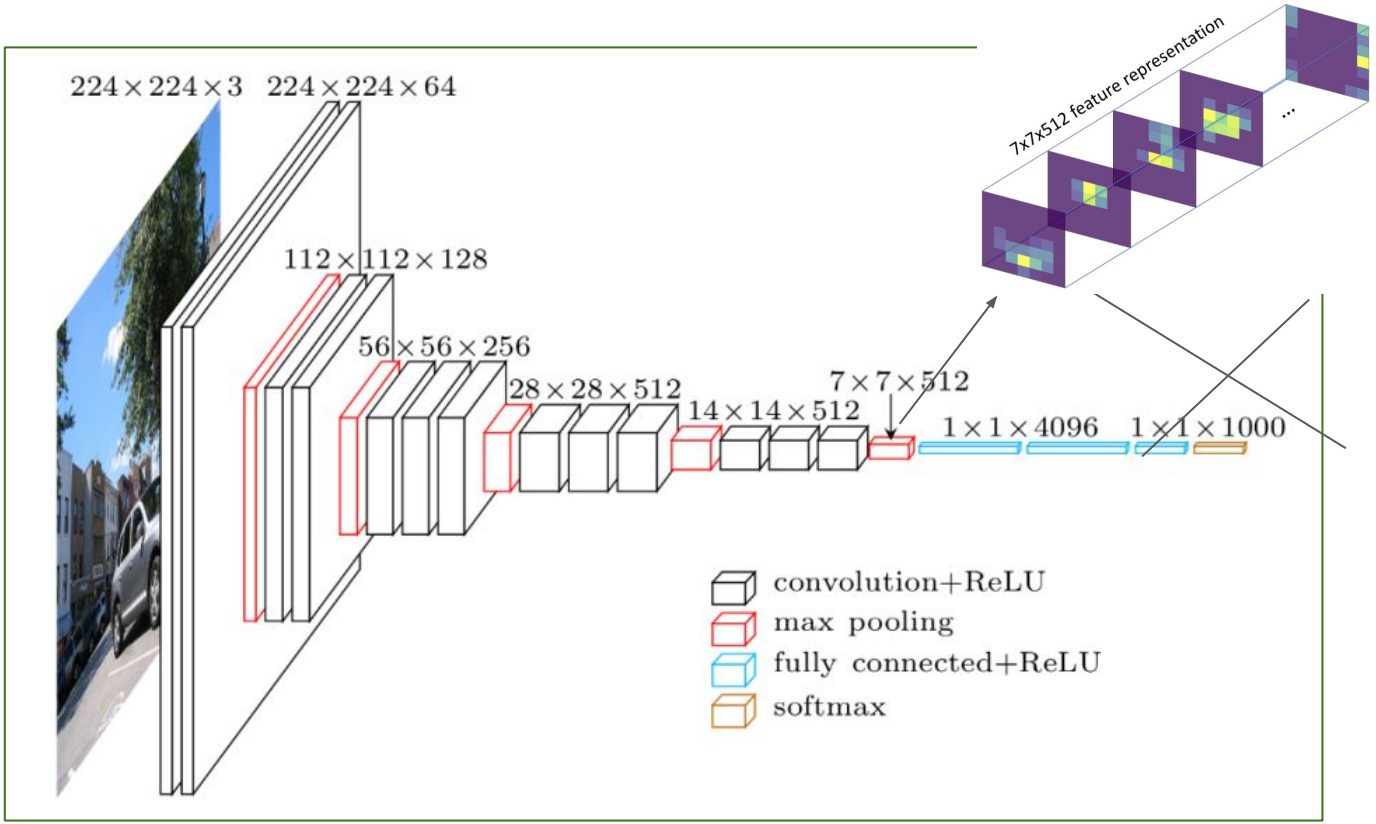
Object detection




CNN
Classification
Regression
Generation of region proposals
Classification and regression
First stage
Second stage

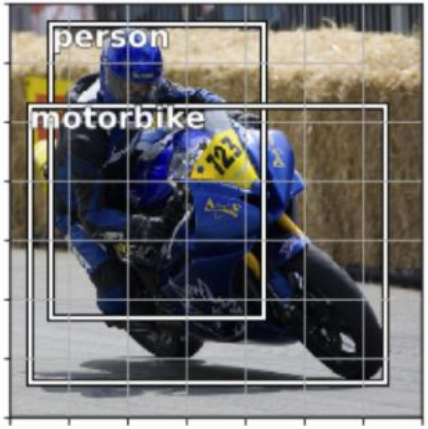
Одно етапні методи

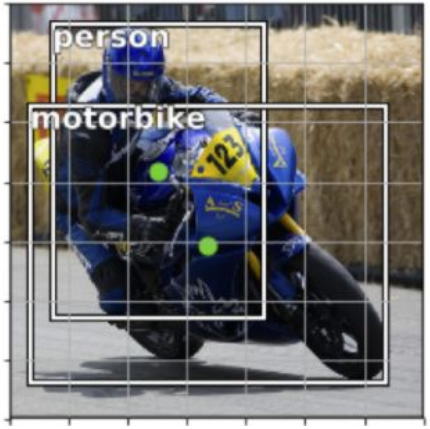


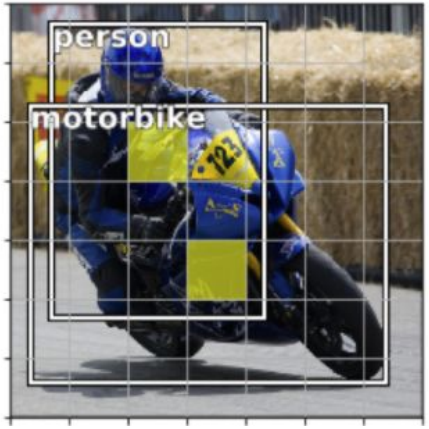

car

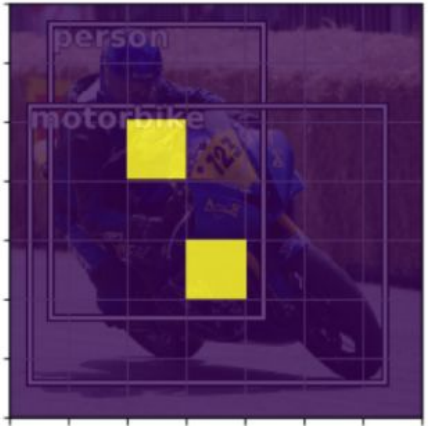

car


car

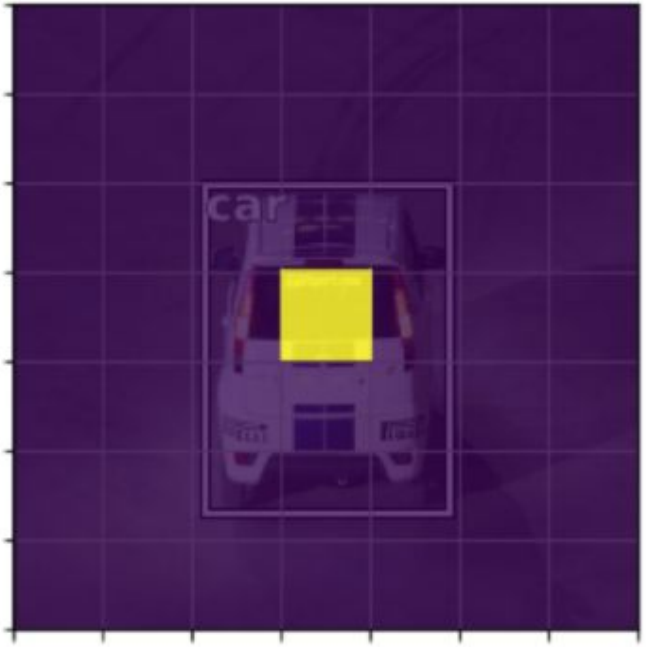

person
motorbike

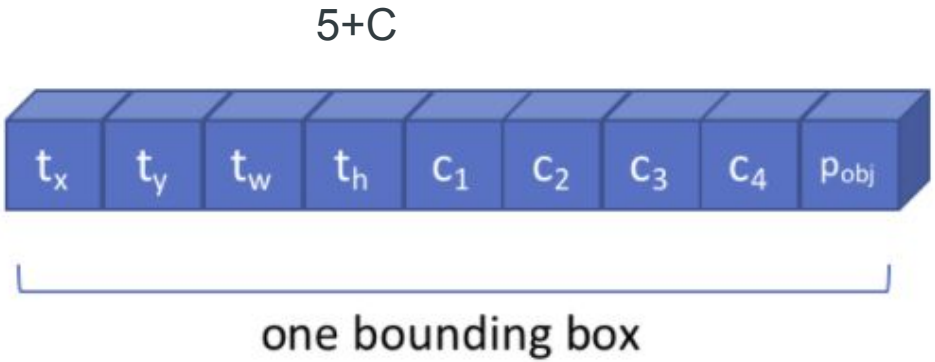

person
motorbike

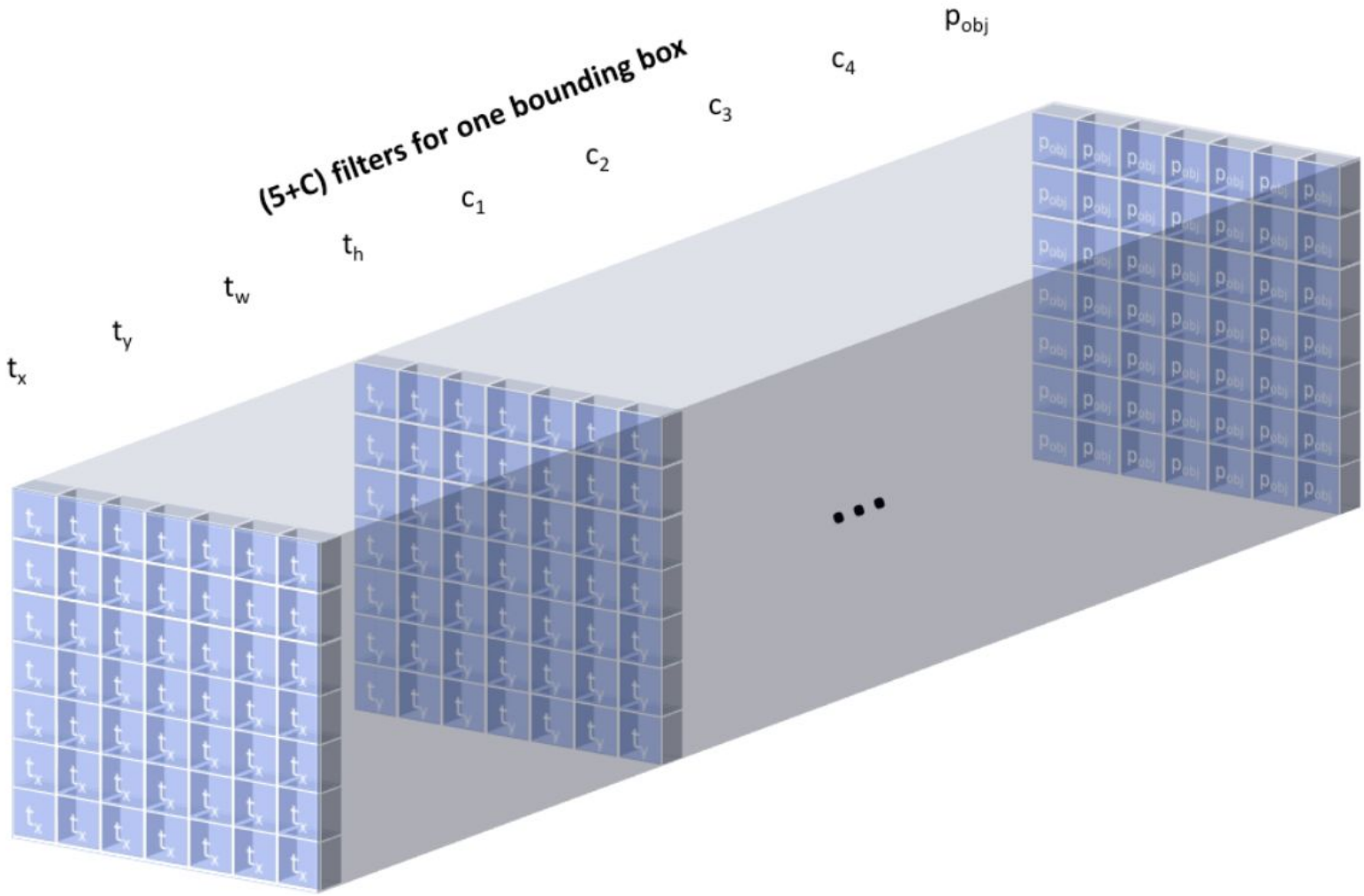

person
motorbike

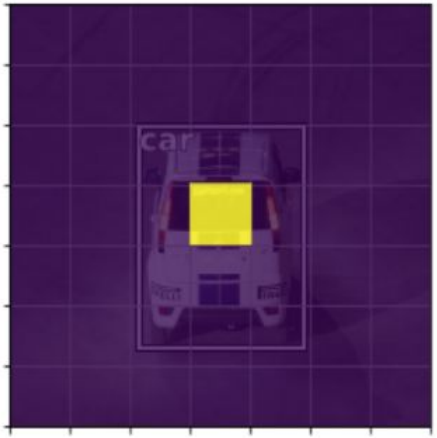

person
motorbike

1. Координати центру рамки
2. Ширина та висота рамки
3. Якому з класів належить об’єкт (Ci)
4. Ймовірність того що в рамці є об’єкт






(5+C) filters for one bounding box
t_x
t_y
t_w
t_h
c_1
c_2
c_3
c_4
p_{obj}

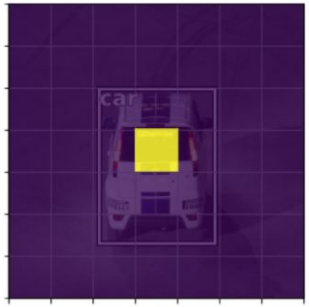


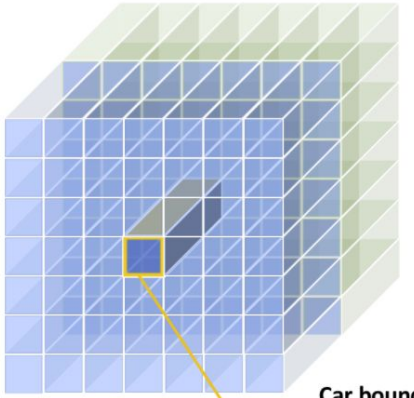
Learning 2 bounding boxes for each grid cell requires 18 channels
(assuming 4 possible classes)

$B(5+C) \quad (B=2)$

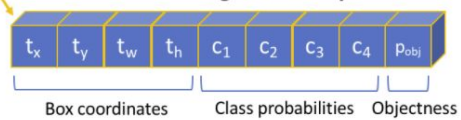
one bounding box

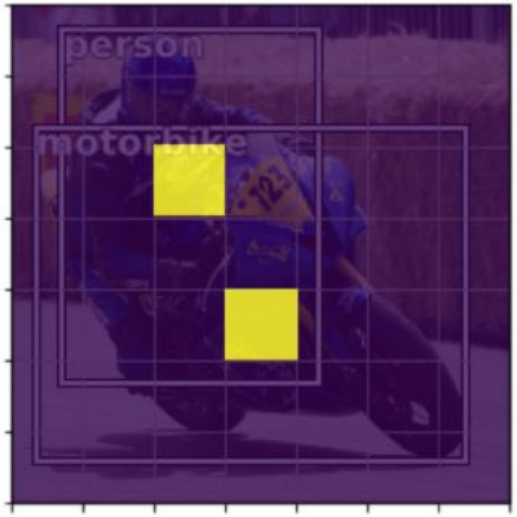
$7 \times 7 \times 2(5+C)$

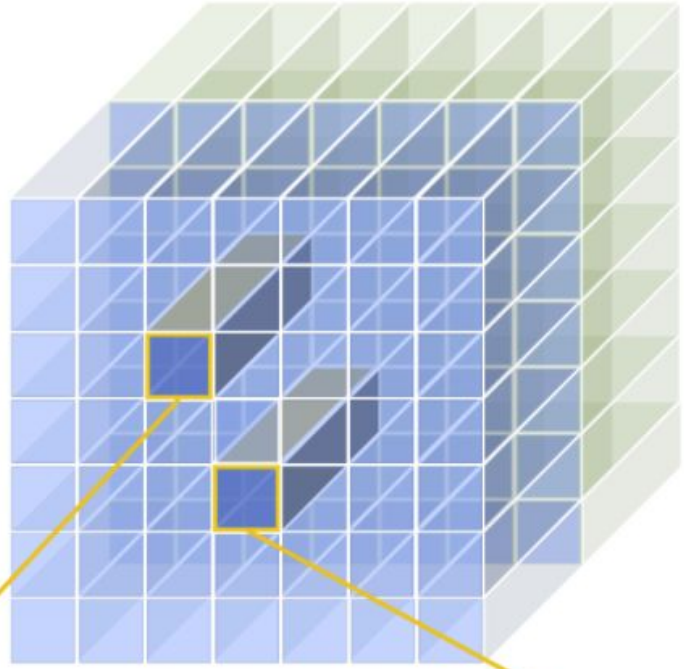




Car bounding box descriptor



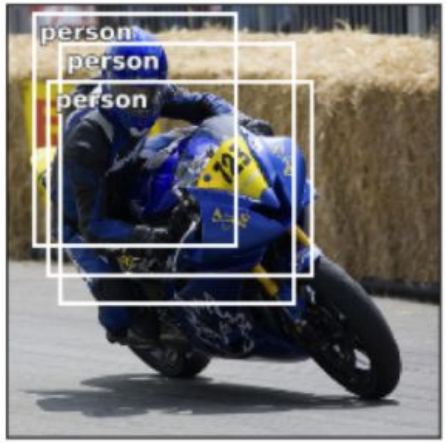


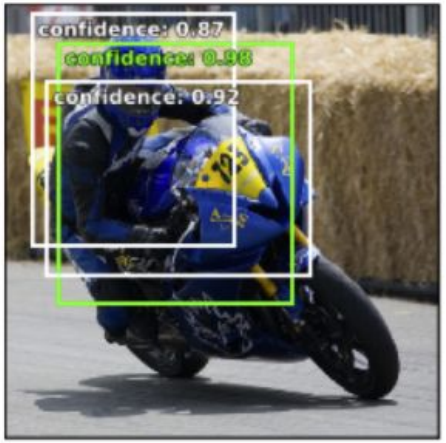


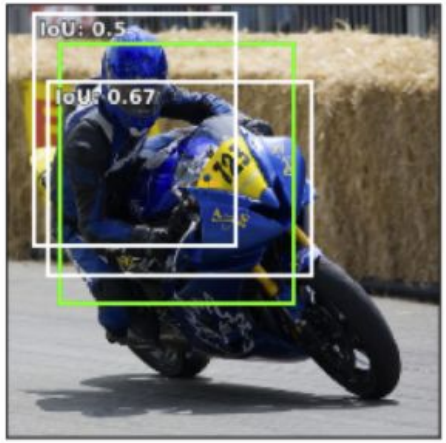
Person bounding box descriptor

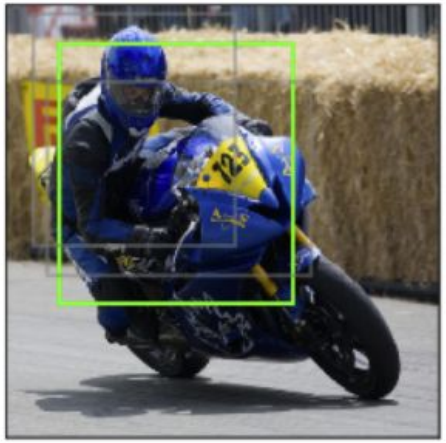
Motorbike bounding box descriptor

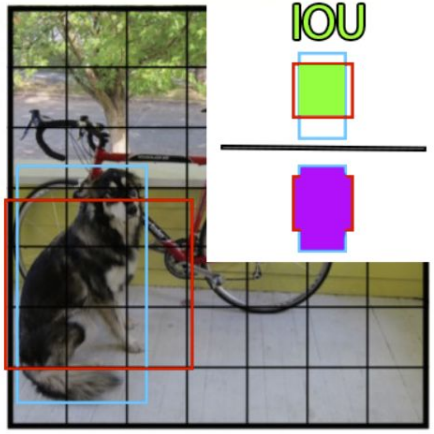
Non-maximum suppression



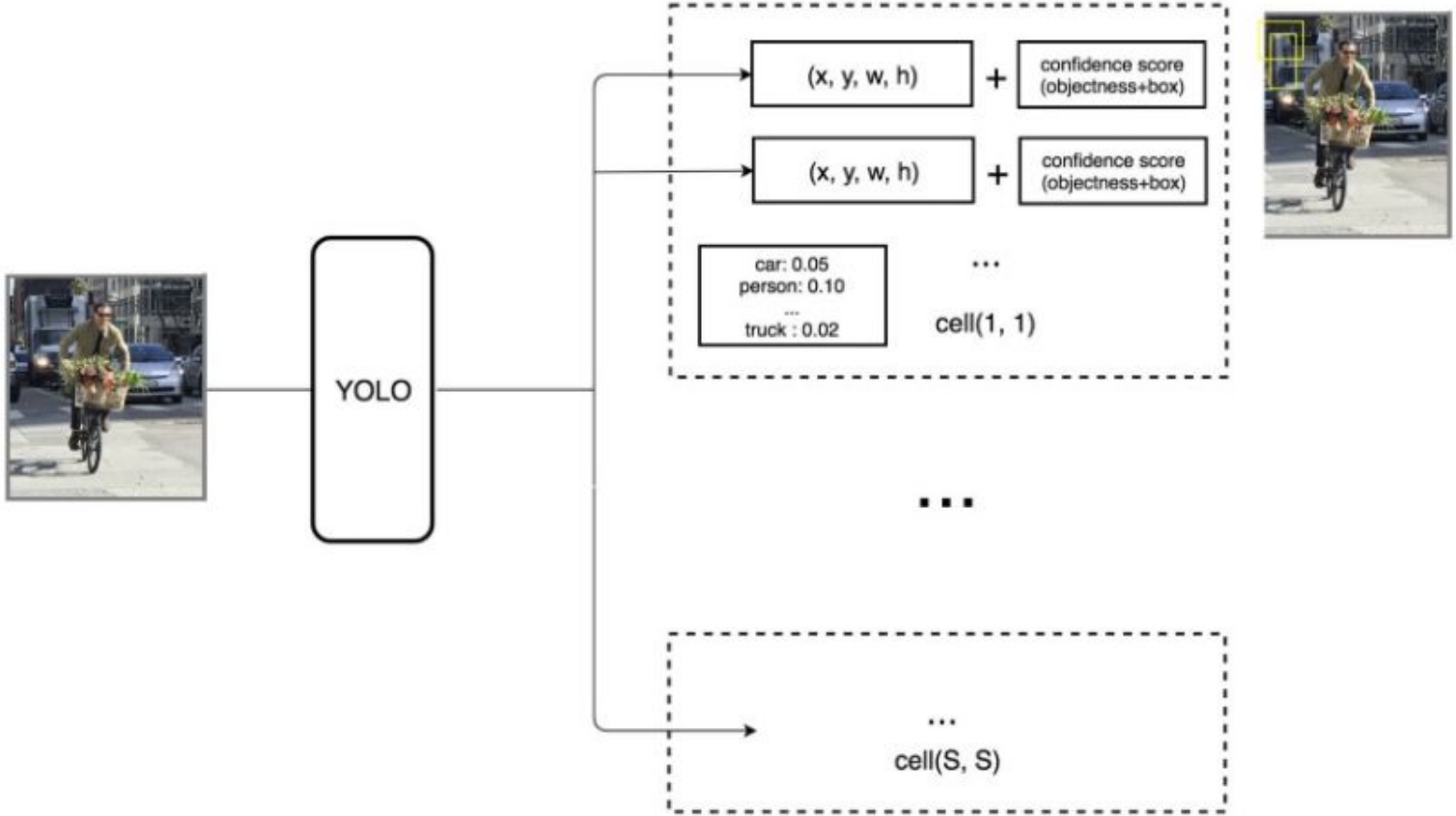






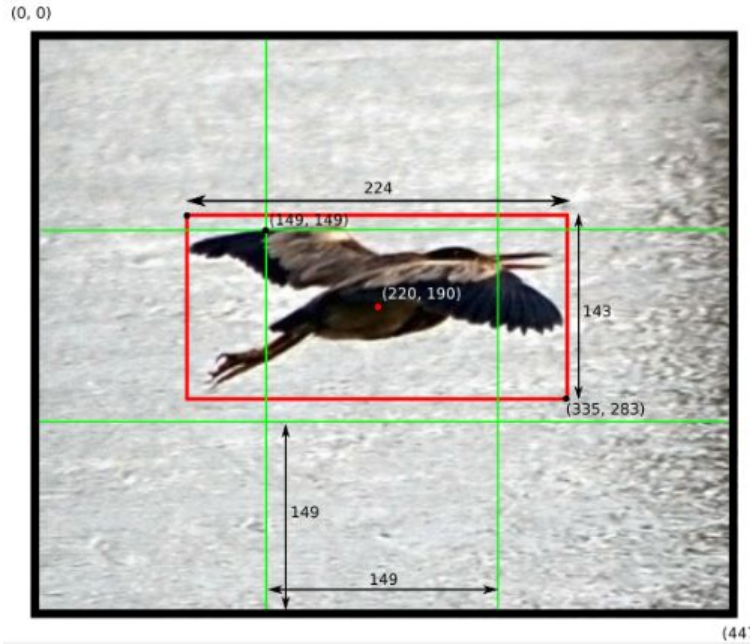


YOLO-You only look once



- *Вихід мережі S x S x (B * 5 + C)*

YOLO-You only look once

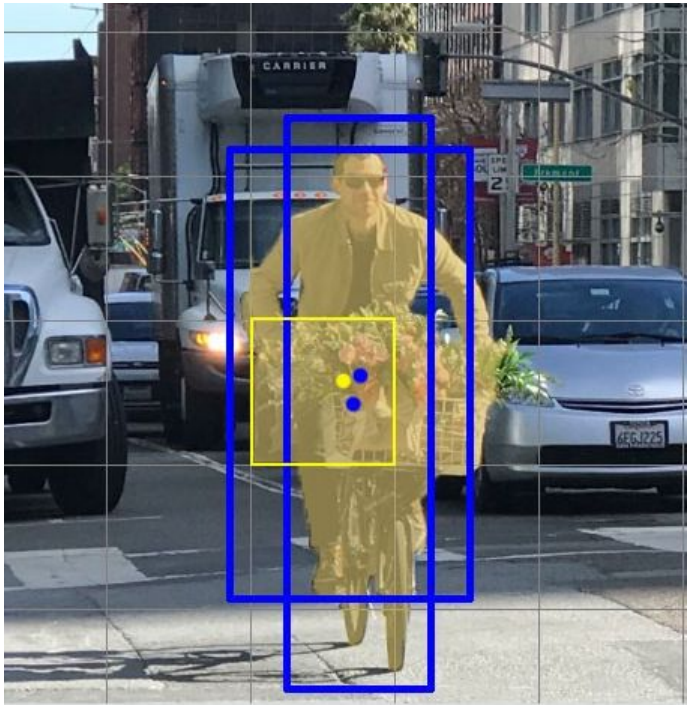


x = (220-149) / 149 = 0.48

y = (190-149) / 149 = 0.28

w = 224 / 448 = 0.50

h = 143 / 448 = 0.32



YOLO v2

t_x, t_y, t_w, t_h are predictions made by YOLO.

c_x, c_y is the top left corner of the grid cell of the anchor.

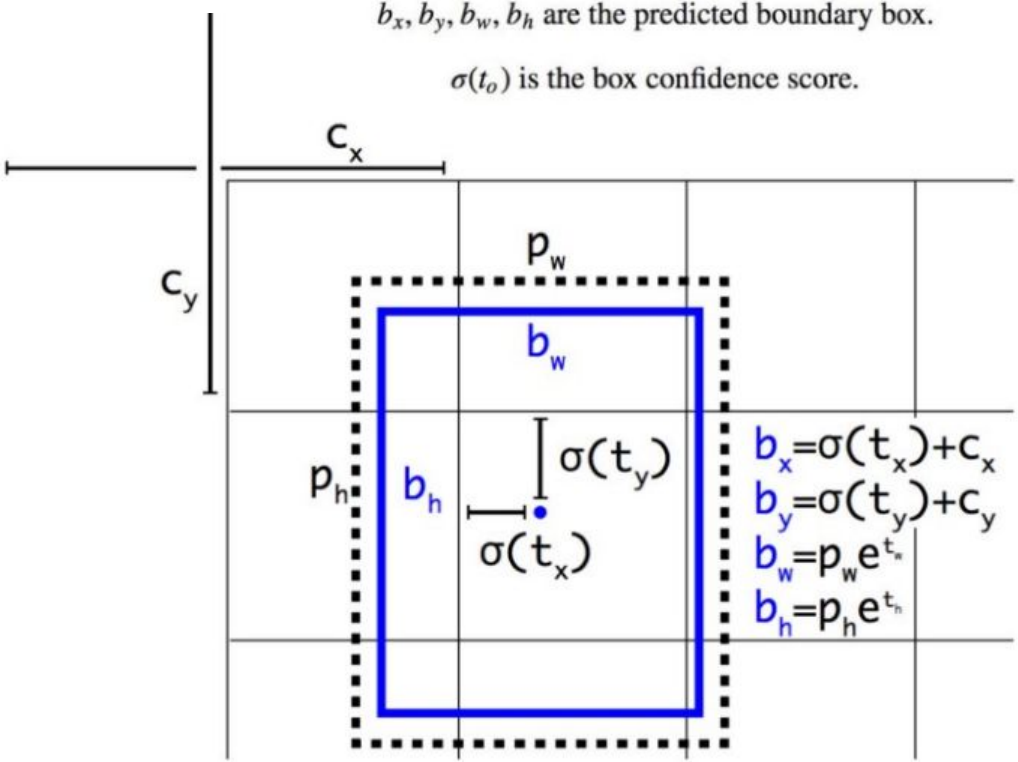
p_w, p_h are the width and height of the anchor.

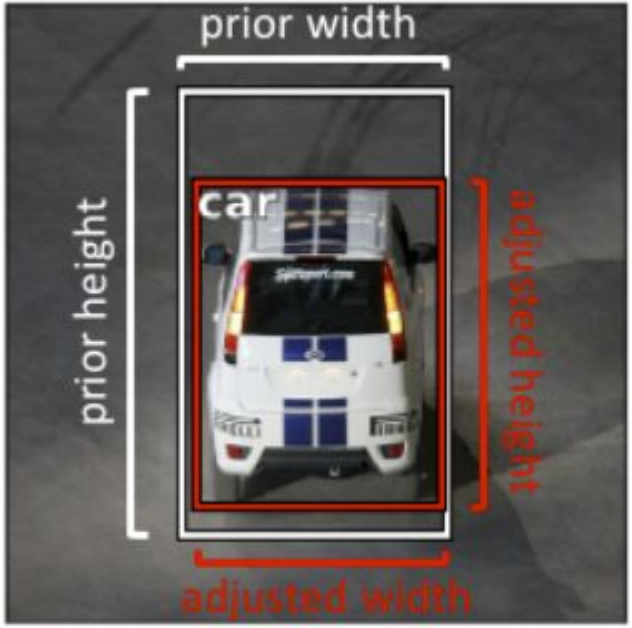
c_x, c_y, p_w, p_h are normalized by the image width and height.

b_x, b_y, b_w, b_h are the predicted boundary box.

$\sigma(t_o)$ is the box confidence score.

$$b_x = \sigma(t_x) + c_x$$
$$b_y = \sigma(t_y) + c_y$$
$$b_w = p_w e^{t_w}$$
$$b_h = p_h e^{t_h}$$
$$Pr(\text{object}) * IOU(b, \text{object}) = \sigma(t_o)$$





Функція втрат

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

$$1_{ij}^{\text{obj}} = \begin{cases} 1 & \text{if the object exists in the i-th cell and j-th box is responsible for detecting it.} \\ 0 & \text{otherwise} \end{cases}$$

- Це рівняння обчислює втрати, пов'язані з передбачуваним положенням обмежувальної комірки ***(x, y)***. Функція обчислює суму по кожному передбаченню обмежувального вікна (***j = 0 .. B***) кожної комірки сітки (***i = 0 .. S ^ 2***). Лише одне передбачення відповідає за об’єкт.

Функція втрат

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

- Це рівняння обчислює втрати, пов'язані з передбачуваним розміром обмежувальної рамки. Квадрат тут через те що малі відхилення в великих рамках мають менше значення ніж маленьких рамках.

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

$$1_{ij}^{\text{noobj}} = \begin{cases} 1 & \text{if there is no object in the i-th cell} \\ 0 & \text{otherwise} \end{cases}$$

- C - впевненість рамки і в комірці j

Функція втрат

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

$\mathbb{1}_i^{obj} = 1$ if an object appears in cell i, otherwise 0.

$\hat{p}_i(c)$ denotes the conditional class probability for class c in cell i.

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

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

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

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

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

Before non-max suppression

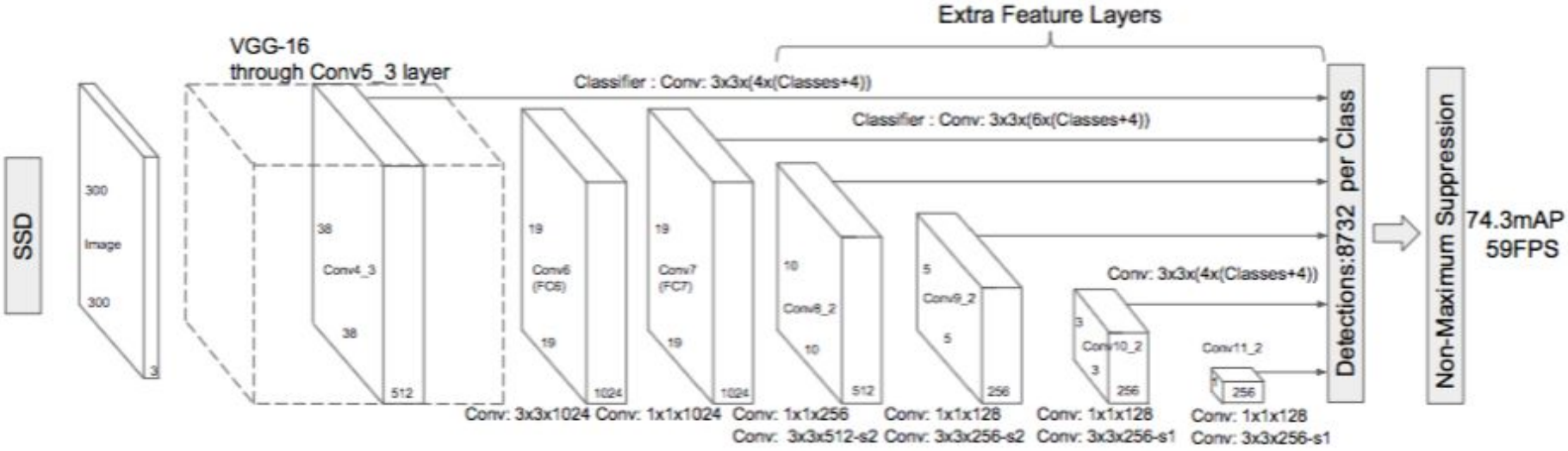
Non-Max Suppression

After non-max suppression

YOLO v3

- Для того щоб в нас були взаємовиключні класи об’єктів використовується softmax (ймовірність 1). В YOLO v3 використовується класифікація з багатьма мітками (>1) (людина і дитина не є взаємовиключними класами), замість softmax незалажні логістичні класифікатори, відповідно замість середьоквадратичної помилки втрат класифікації використовується бінарні крос етропійні втрати.
- YOLO v3 прогнозує впевненість кожної рамки використовуючи логістичну регресію
- Використовує shortcut connections, щоб краще знаходити малі об’єкти

SSD: Single Shot MultiBox Detector



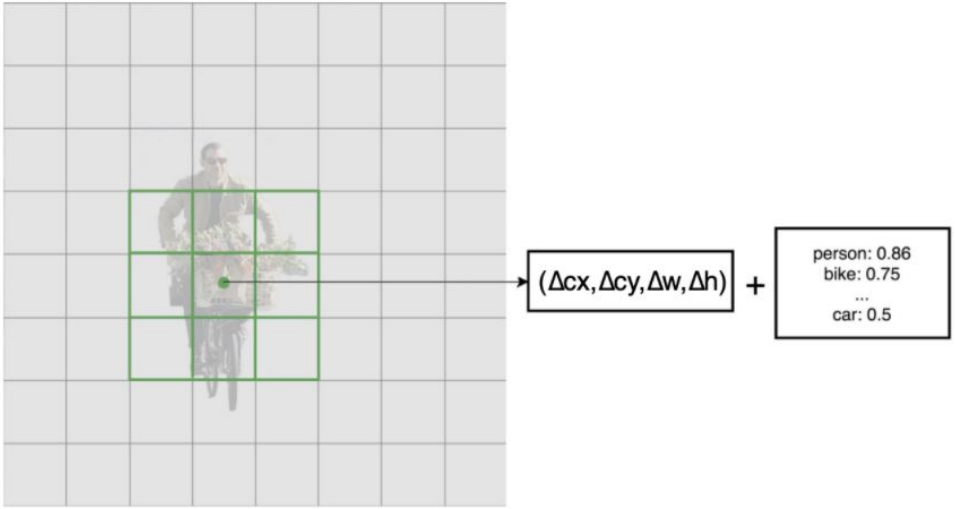
SSD

$(\Delta cx, \Delta cy, \Delta w, \Delta h)$ + person: 0.86, bike: 0.75, ..., car: 0.5

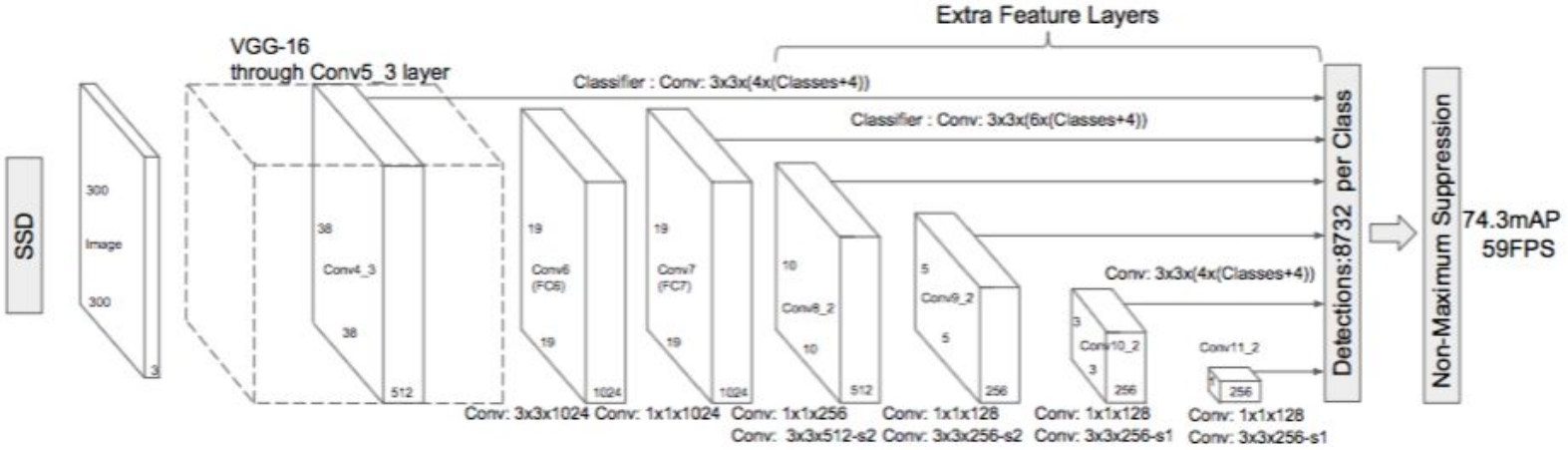
$(\Delta cx, \Delta cy, \Delta w, \Delta h)$ + car: 0.75, bike: 0.25, ..., person: 0.115

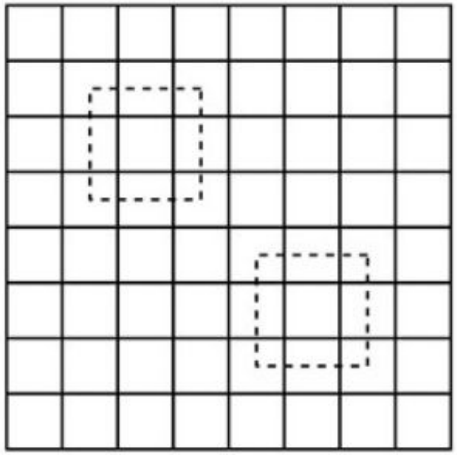
...

$(\Delta cx, \Delta cy, \Delta w, \Delta h)$ + truck: 0.95, car: 0.55, ..., person: 0.01

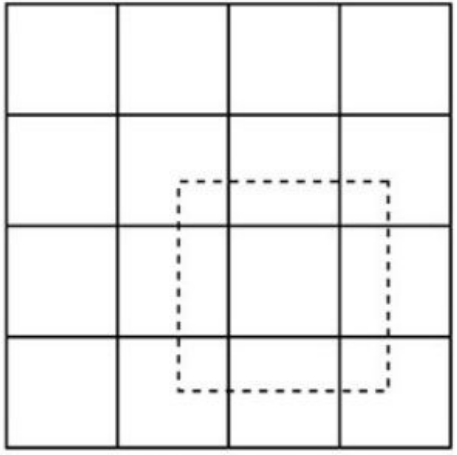


$$(38 \times 38 \times 512) \xrightarrow{(4\times3\times3\times512\times(21+4))} (38 \times 38 \times 4 \times (21+4))$$

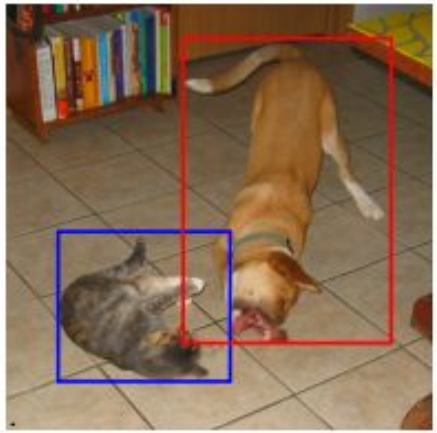


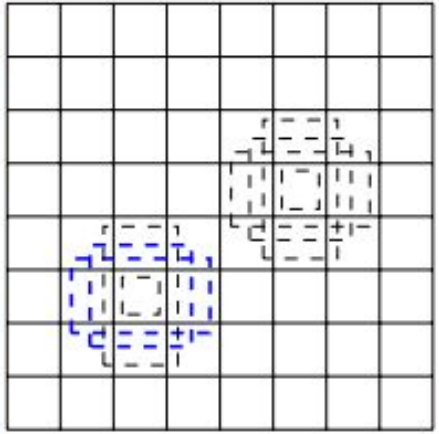


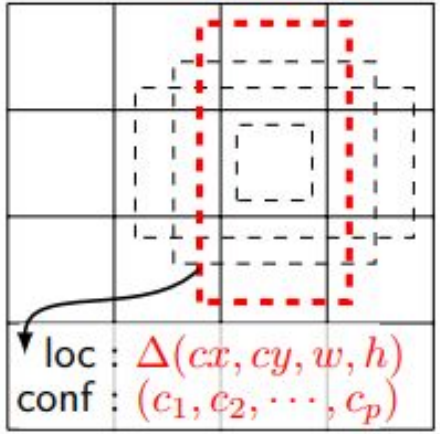
8 × 8 feature map

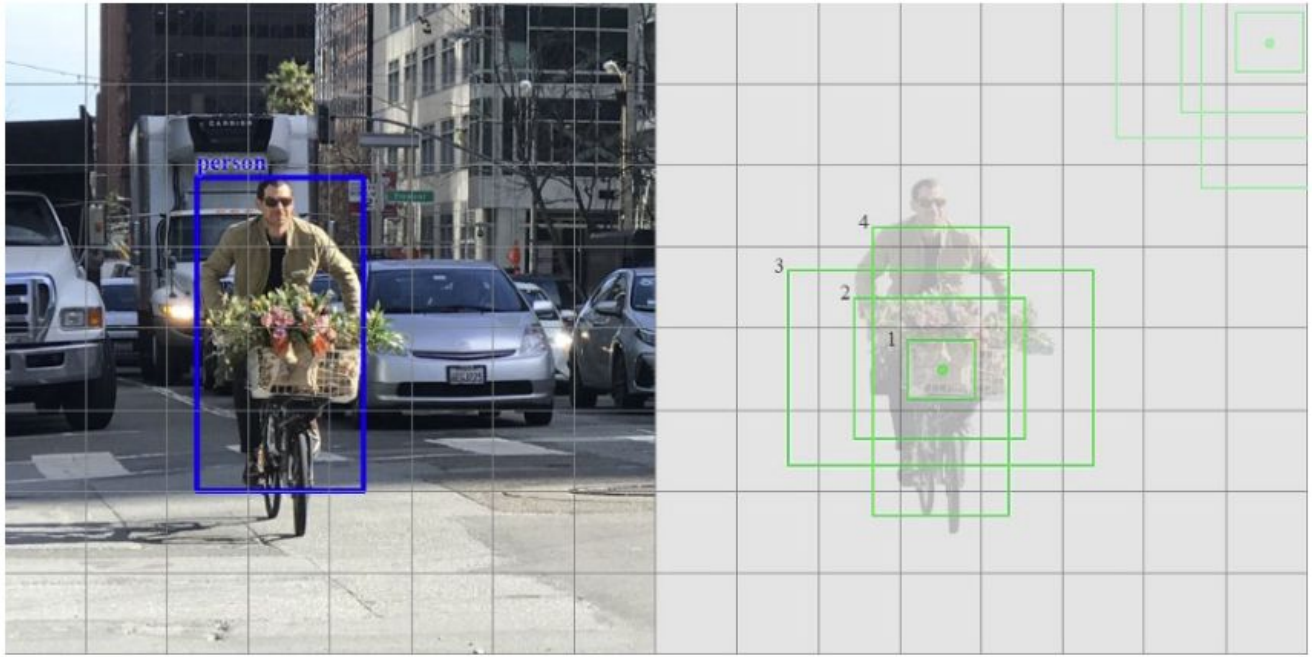


4 × 4 feature map









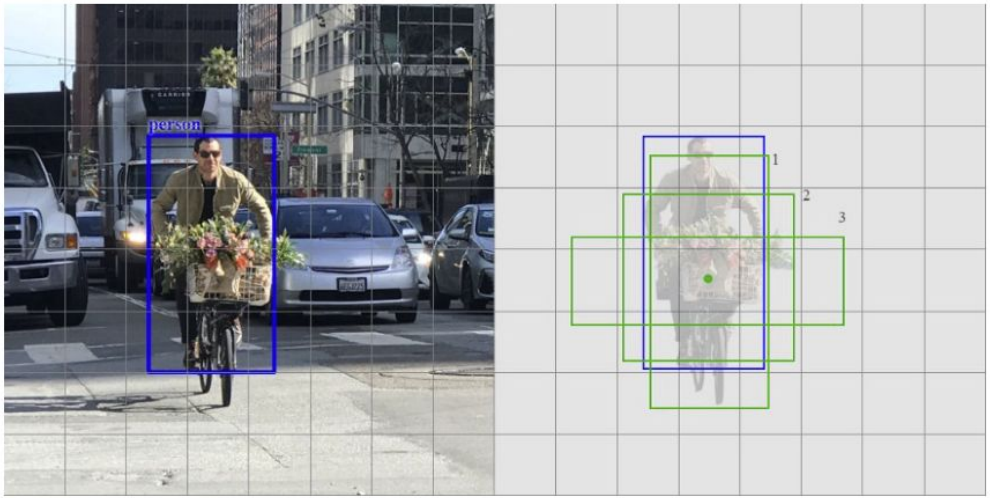
$$w = scale \cdot \sqrt{\text{aspect ratio}}$$

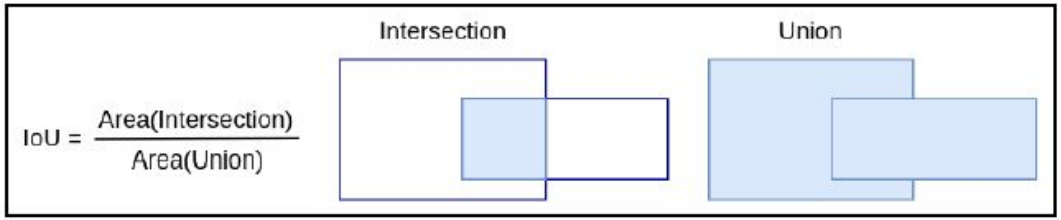
$$h = \frac{scale}{\sqrt{\text{aspect ratio}}}$$

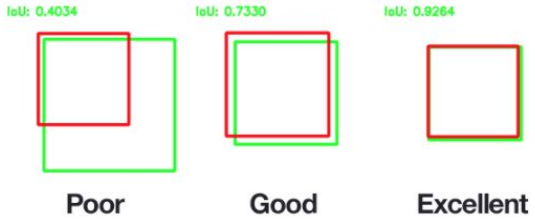
Then SSD adds an extra default box with scale:

$$scale = \sqrt{\text{scale} \cdot \text{scale at next level}}$$

- Кожний елемент карти ознак пов’язаний з набором обмежувальних рамок різних розмірів і пропорцій
- Для кожного об’єкта мають бути задані ground truth boxes







Функція оцінки

$$L(x, c, l, g) = \frac{1}{N}(L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

$$L_{loc}(x, l, g) = \sum_{i \in Pos}^{N} \sum_{m \in \{cx, cy, w, h\}} x_{ij}^{k} \text{smooth}_{\text{L1}}(l_i^m - \hat{g}_j^m)$$

$$\hat{g}_j^{cx} = (g_j^{cx} - d_i^{cx})/d_i^w \qquad \hat{g}_j^{cy} = (g_j^{cy} - d_i^{cy})/d_i^h$$

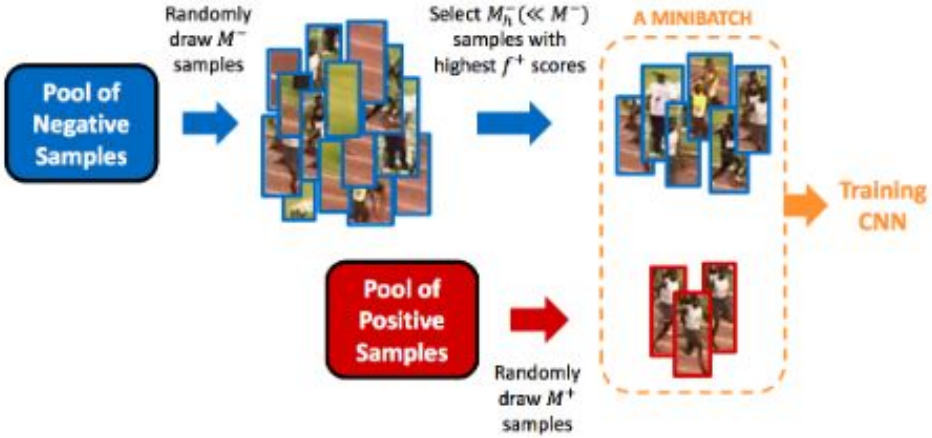
$$\hat{g}_j^w = \log\left(\frac{g_j^w}{d^w}\right) \qquad \hat{g}_j^h = \log\left(\frac{g_j^h}{d^h}\right)$$

$$L_{conf}(x, c) = -\sum_{i \in Pos}^{N} x_{ij}^p log(\hat{c}_i^p) - \sum_{i \in Neg} log(\hat{c}_i^0) \quad \text{where} \quad \hat{c}_i^p = \frac{\exp(c_i^p)}{\sum_p \exp(c_i^p)}$$

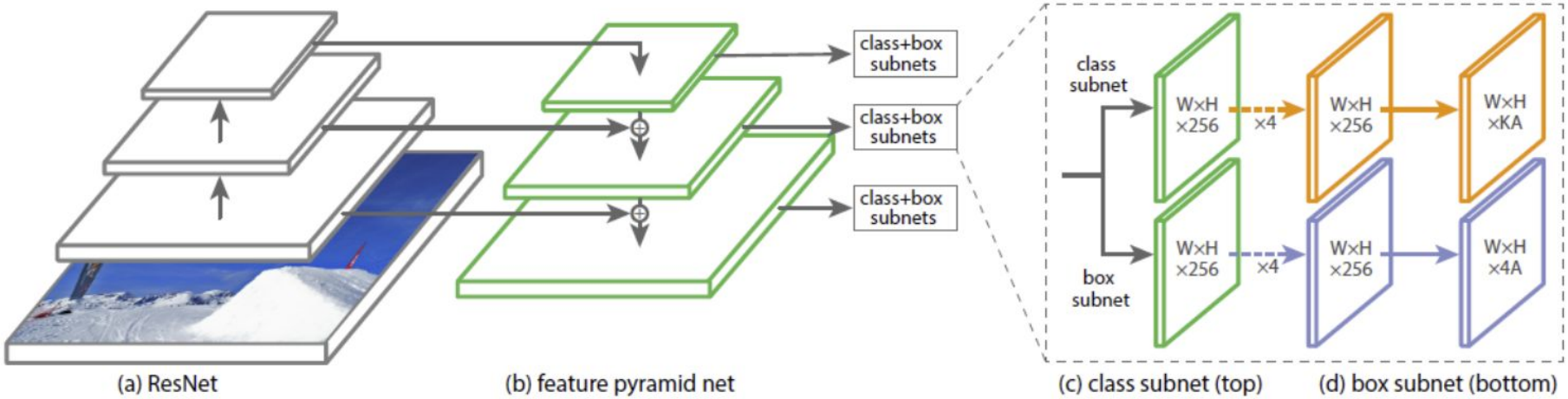
- X_{ij}^{p} індикатор співпадіння і-тої рамки за замовчуванням і j-тої ground truth рамки p-ї категорії, N-кількість рамок що співпали, l- спрогнозована рамка, g- ground truth рамка, d- рамка за замовчуванням.

- Hard Negative Mining
- Аугментація
- Non-Maximum Suppression (NMS)-

Рамки зі значенням впевненості менше 0.01, та IoU менше 0.5



RetinaNet model



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

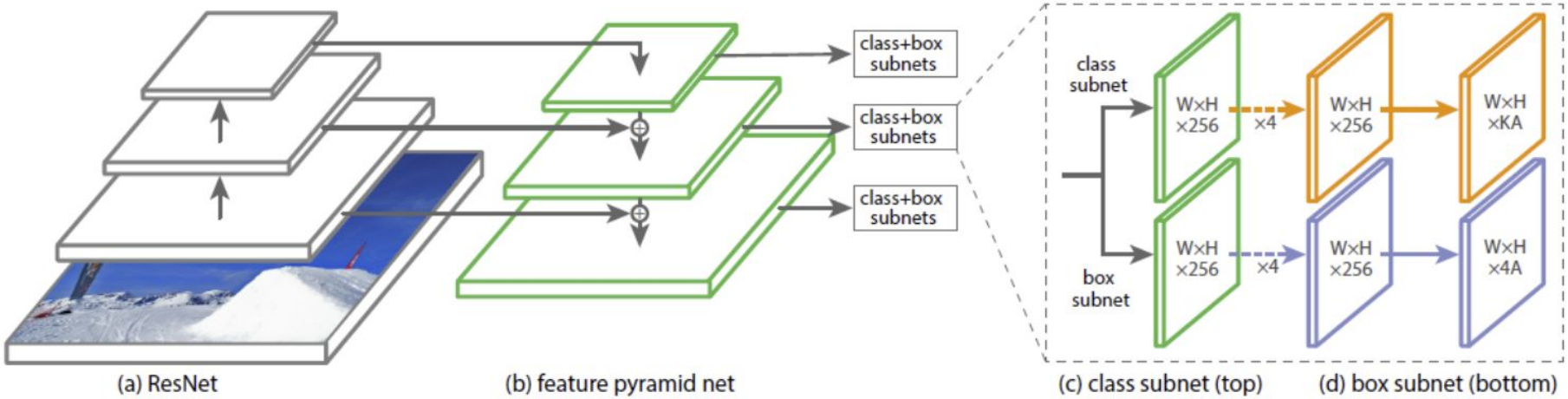
Focal Loss (FL)

$$p_t = \begin{cases} p & \text{if } y = 1 \\ 1 - p & \text{otherwise,} \end{cases}$$

$$\mathrm{CE}(p, y) = \mathrm{CE}(p_t) = -\log(p_t).$$

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

RetinaNet model



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

Object detection

Yolo та SSD

- Використовуються попередньо визначені якірні рамки (kmeans), які потім уточнюються.
- В штрафній функції використовується коефіцієнт щоб впоратися з великою кількістю рамок де немає об’єкта

- Використовуються різні якірні рамки з різним співвідношенням сторін для карт ознак різної роздільної здатності.

level index: $\ell = 1, \ldots, L$

scale of boxes: $s_\ell = s_{\min} + \frac{s_{\max} - s_{\min}}{L-1}(\ell - 1)$

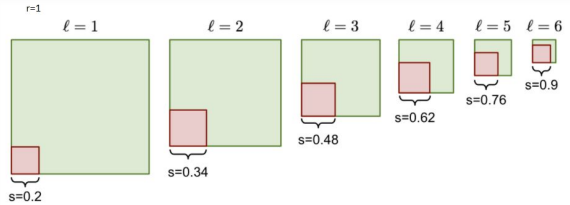
aspect ratio: $r \in \{1, 2, 3, 1/2, 1/3\}$

additional scale: $s'_\ell = \sqrt{s_\ell s_{\ell+1}}$ when $r = 1$thus, 6 boxes in total.

width: $w^r_\ell = s_\ell \sqrt{r}$

height: $h^r_\ell = s_\ell / \sqrt{r}$

center location: $(x^i_\ell, y^j_\ell) = (\frac{i+0.5}{m}, \frac{j+0.5}{n})$



- hard negative mining. SSD штрафує лише неспівпадіння між істинною рамкою і рамкою, що містить об’єкт (позитивною). Оскільки співпадінь рамок що не містять об’єкт буде явно більше виконуємо HNM. Сортуємо негативні рамки за confidence loss і беремо негативи з більшим значенням (3:1)


bus 79.90%
car 95.38%


bus 98.27%
car 84.93%

- Порахувати IOU для всіх bbox
- Налаштування гіперпараметрів