SSD

임수현

2019.08.06

Single Shot MultiBox Detection

SSD 란?

SSD :
Single Shot MultiBox Detection

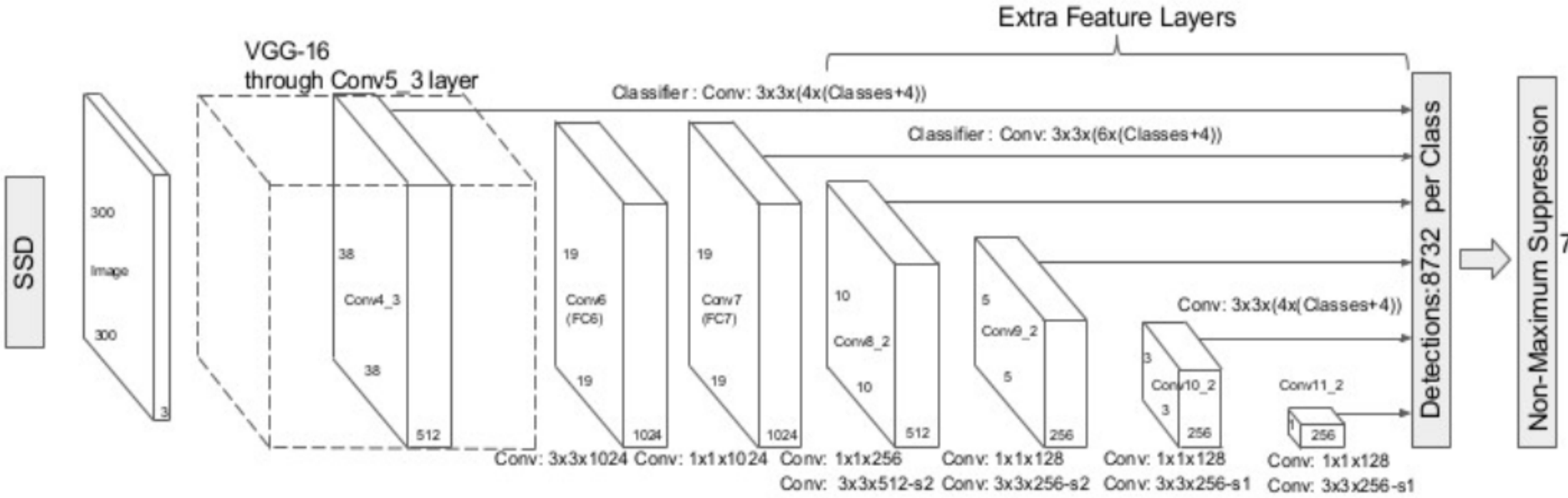
SSD의 가장 큰 특징

여러 크기의 feature map 에서 detection

참고)
YOLO 는 마지막 feature map에서만
detection 정보를 가짐

Single Shot MultiBox Detection

SSD 네트워크

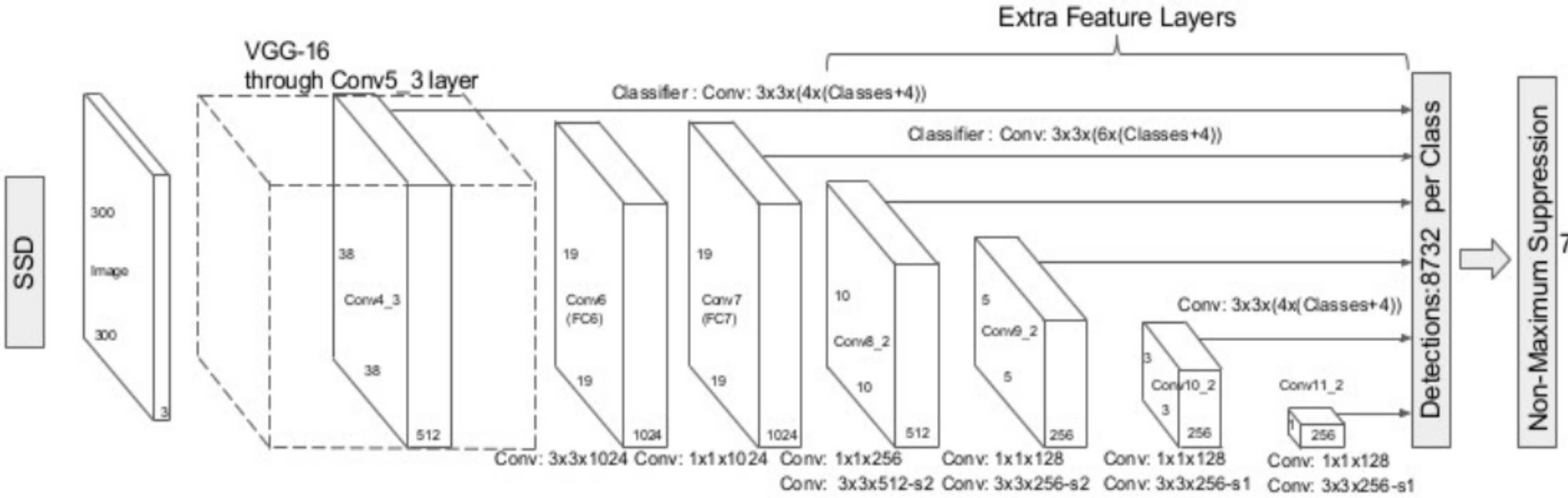


1. 수정된 VGG-16을 이용해 이미지 특징 추출한다.

2. Convolution 과정 중, 6개의 Layer가 경계박스와 Class정보를 담는다.

Single Shot MultiBox Detection

SSD 네트워크



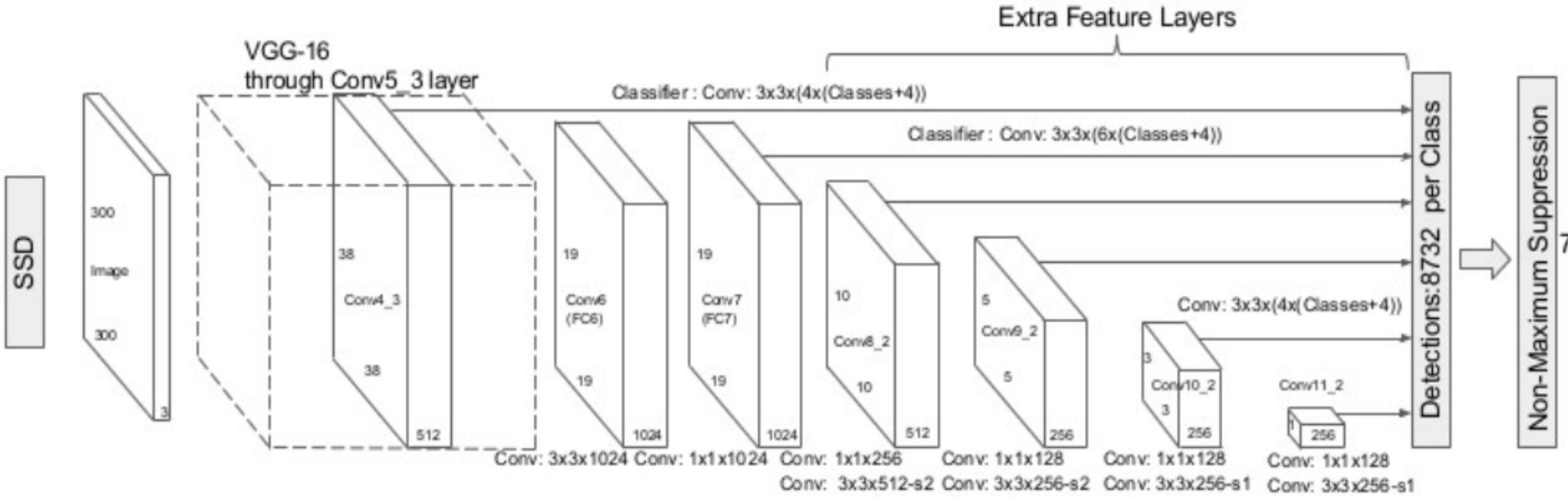
1. VGG-16

- ImageNet 데이터베이스의 1백만개가 넘는 이미지에 대해 훈련된 컨볼루션 신경망
- 이를 약간 변경하여 이미지 특징 추출에 사용

  ex) conv7은 VGG-16에서 원래 FC이었으나, SSD에서 conv로 변형

Single Shot MultiBox Detection

SSD 네트워크



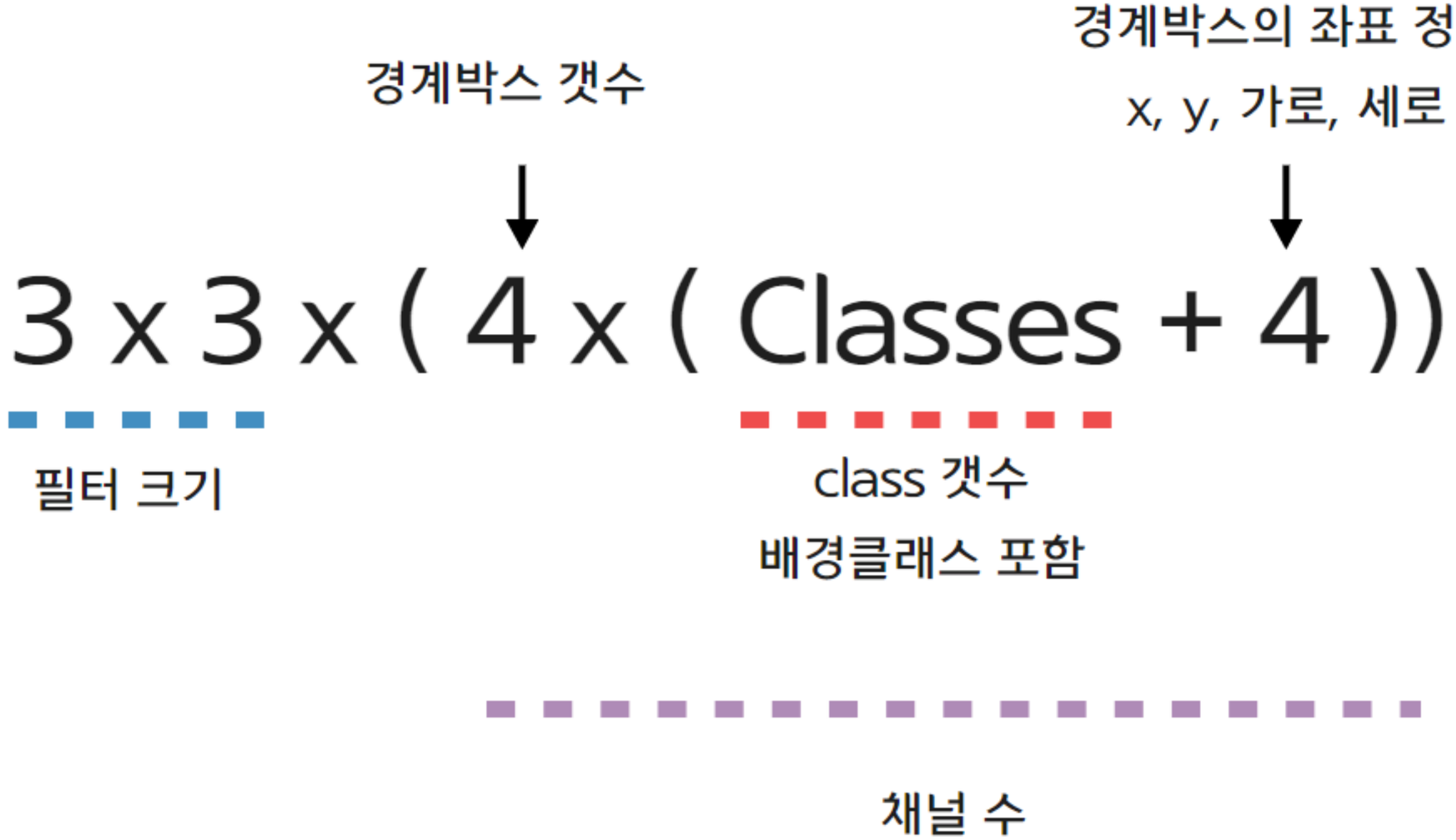
2. 6개의 Layer가 경계박스와 Class 정보를 포함

conv4_3 (38x38) , conv7 (19x19) , conv8_2(10x10),

conv9_2(5x5), conv10_2(3x3), conv11_2(1x1)

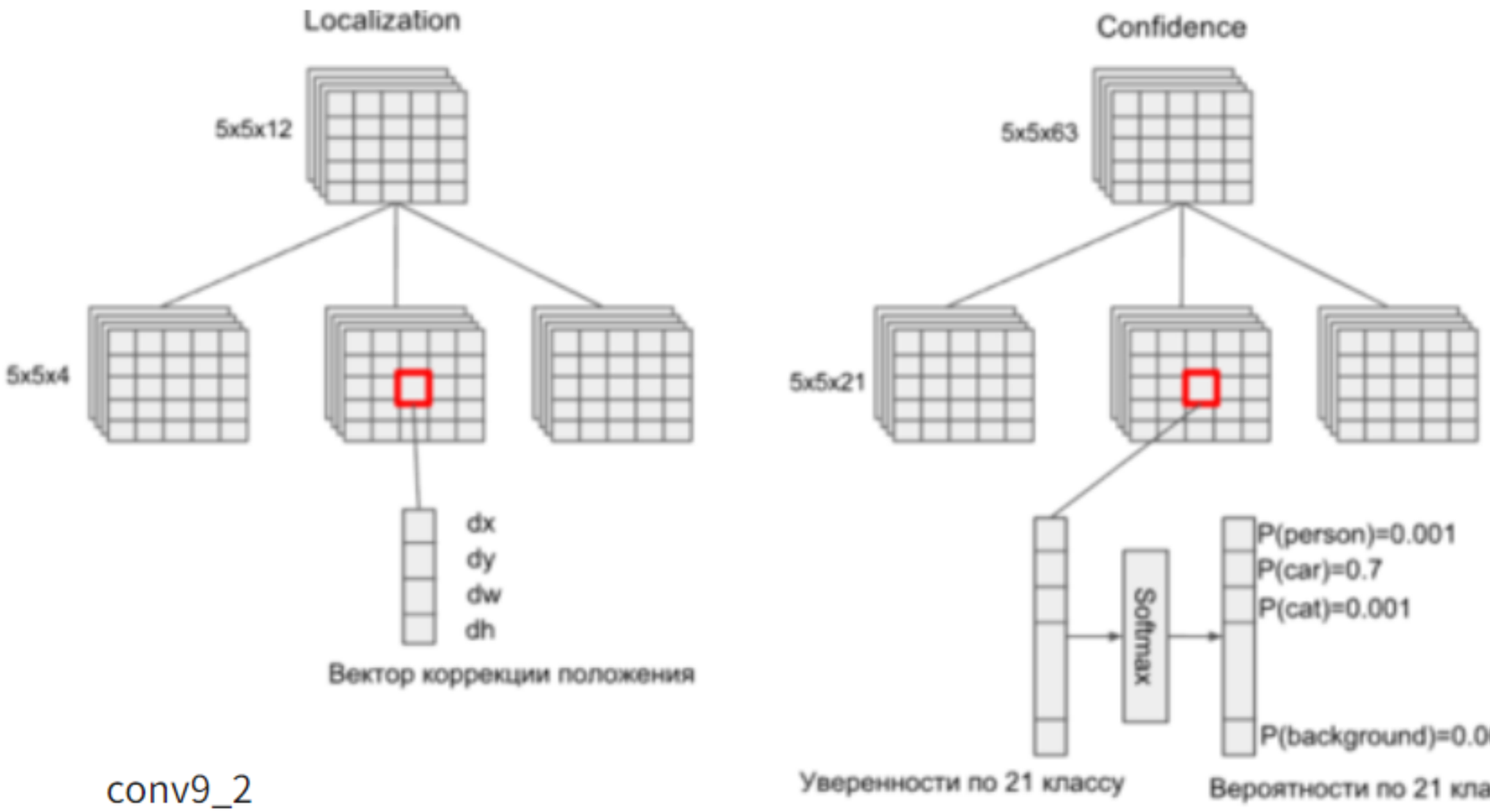
SSD 네트워크

conv4_3 (38x38x512) 을 3x3x(4x(Classes+4))의 가중치로 컨볼루션
= 38x38x(4x(Classes+4))의 특징맵



Single Shot MultiBox Detection

SSD 네트워크



conv9_2

$$5 \times 5 \times (3 \times (21 + 4))$$

5 x 5 그리드의 각 셀에서 3개의 경계박스 좌표 정보가 21개의 클래스 중 어떤클래스에 속하는지를 나타낸다.

SSD 네트워크

conv4_3 로 부터 38x38x(4x(Classes+4)) = **5776x(Classes+4)**

conv7 로 부터 19x19x(6x(Classes+4)) = **2166x(Classes+4)**

conv8_2로 부터 10x10x(6x(Classes+4)) = **600x(Classes+4)**

conv9_2로 부터 5x5x(6x(Classes+4)) = **150x(Classes+4)**

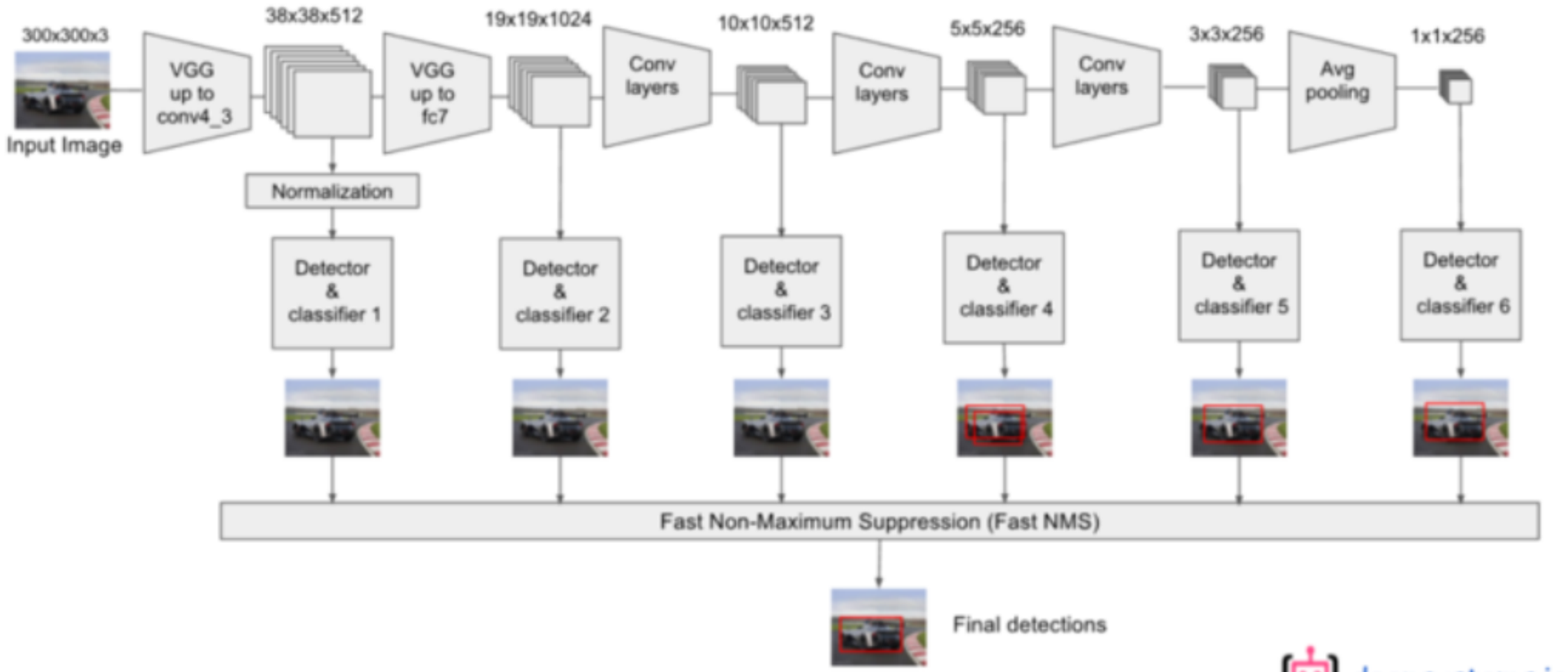
conv10_2로 부터 3x3x(4x(Classes+4)) = **36x(Classes+4)**

conv11_2 로 부터 1x1x(4x(Classes+4)) = **4x(Classes+4)**

총 **8732 x (Classes+4)** 로 클래스당 8,732개의 경계박스를 예측한다.

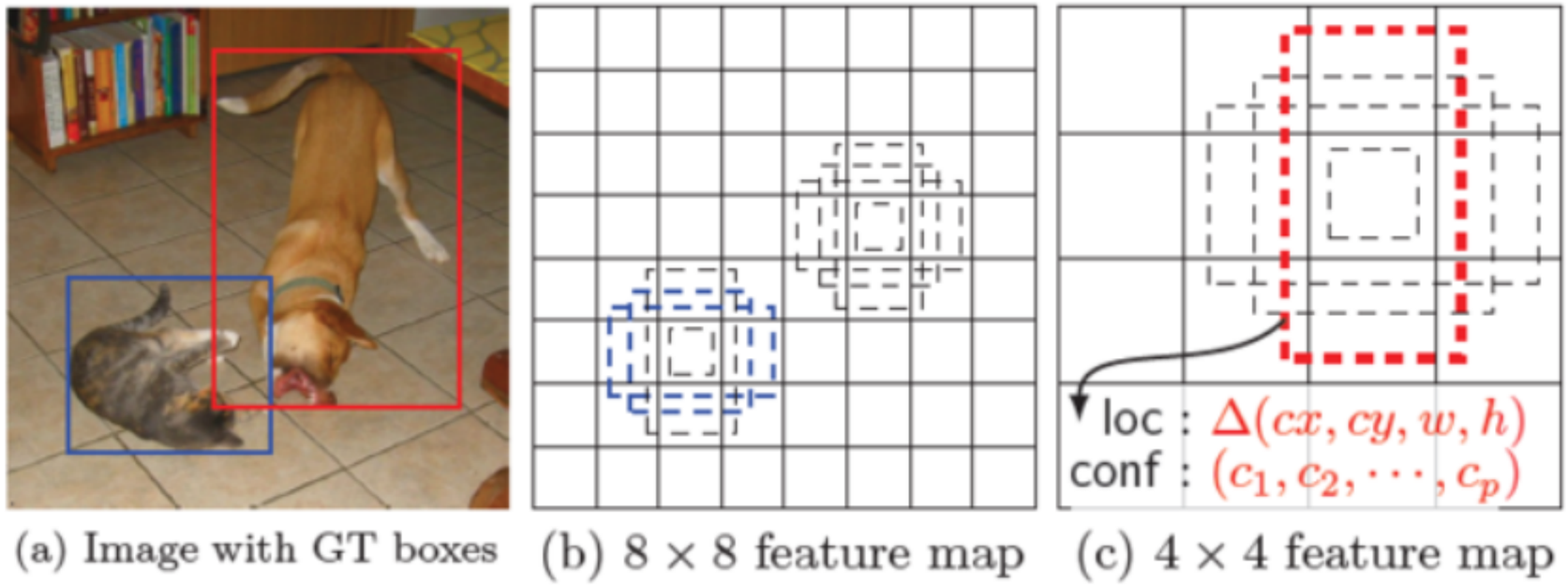
Single Shot MultiBox Detection

SSD 네트워크



deepsystems.io

SSD 경계박스 (default bounding box)



(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

여러 크기의 feature map 에서 detection

- 그리드 셀에따라 경계박스의 크기가 결정
- 6개의 특징맵은 각각 다른 크기의 오브젝트를 검출 책임
- 작은 그리드 셀에서는 작은 오브젝트
- 큰 그리드 셀에서는 큰 오브젝트

SSD 경계박스 (default bounding box)

어떻게 그리지?

어떤 경계박스를 선택하지?

SSD 경계박스 (default bounding box)

어떻게 그리지?

aspect ratio = 종횡비 = 가로 / 세로

4x : 1, (작은)1, 2, 1/2

6x : 1, (작은)1,2, 3, 1/2, 1/3

SSD 경계박스 (default bounding box)

어떻게 그리지?

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \quad k \in [1, m]$$

S : scale

Smin: 가장작은 스케일 0.2

Smax : 가장 큰 스케일 0.9

m : feature map 갯수

k : feature map 번호

Single Shot MultiBox Detection

SSD 경계박스 (default bounding box)

어떻게 그리지?

k = 1일때
최소 스케일 보장

k가 커질수록
step증가

$$s_k = s_{\min} + \frac{s_{\max} - s_{\min}}{m - 1}(k - 1), \quad k \in [1, m]$$

k 번째 경계박스의 스케일

step

$$a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$$

$$w_k^a = s_k\sqrt{a_r}$$

$$h_k^a = s_k/\sqrt{a_r}$$

width and height of default bbox

a종횡비를 갖는
k번째 경계박스의
가로, 세로

Single Shot MultiBox Detection

SSD 경계박스 (default bounding box)

어떻게 그리지?

$$a_r \in \{1, 2, 3, \frac{1}{2}, \frac{1}{3}\}$$

$$(w_k^a = s_k\sqrt{a_r})$$

$$(h_k^a = s_k/\sqrt{a_r})$$

width and height of default bbox

종횡비가 1인 경계박스와 크기를 같게 하면서 주어진 종횡비를 유지하기 위해 루트를 사용

ar = 1

넓이 : Sk^2

종횡비 :1

ar = 2

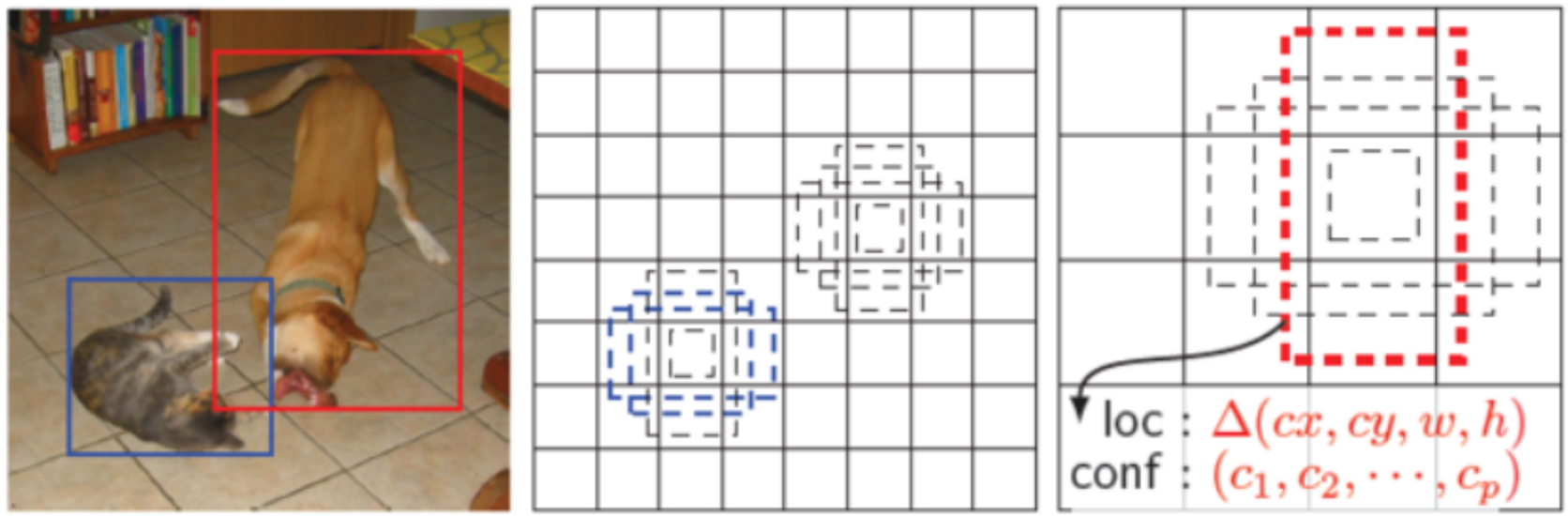
넓이 : Sk^2

종횡비 : 2^2

SSD 경계박스 (default bounding box)

어떤 경계박스를 선택하지?

경계박스와 Ground Truth 박스가 임계값(0.5)이상 overlap되는 모든 경계박스들이 학습할 책임을 갖는다.

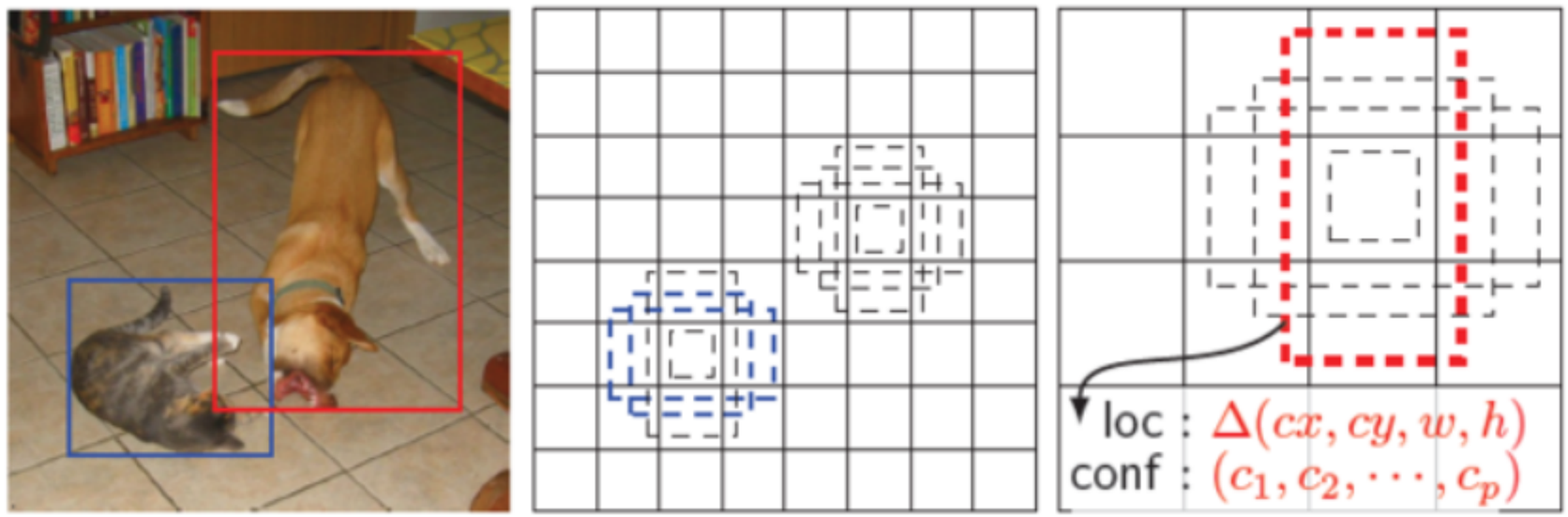


(a) Image with GT boxes (b) 8 × 8 feature map (c) 4 × 4 feature map

만약 negatve 경계박스가 너무 많다면 적절한 학습진행 x

SSD 경계박스 (default bounding box)

어떤 경계박스를 선택하지?



(a) Image with GT boxes (b) 8 × 8 feature map (c) 4 × 4 feature map

학습과정에서 모든 negative 샘플 사용 x

positive : negative = 1 : 3 이 되도록

confidence loss가 높은 것만 선별

SSD 장점

여러 크기의 경계박스

물체의 크기 별로 검출 책임을 맡는 feature map이 다르다.

경계박스의 선택적 학습

학습에 필요한 요소들만 적절하게 학습하게 된다.

보다 빠르고 정확한 네트워크 성능!

SSD Loss

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

N : 매칭된 경계 박스 수 (N=0 이면, loss=0)

Lconf : confidence loss (클래스 예측)

Lloc : localization loss (경계박스 정보)

a : 교차 검증 값 1

총 Loss =

클래스 예측 loss + 경계박스 정보 loss

SSD Loss

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

l : predicted box

g : ground truth box

d: bounding box

cx. cy : center of bounding box

w, h : weight and height of bounding box

x : i번째 bounding box와 p물체의 j번째 GT간의 IoU가 0.5 이상이면 1 아니면 0

SSD Loss

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

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

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

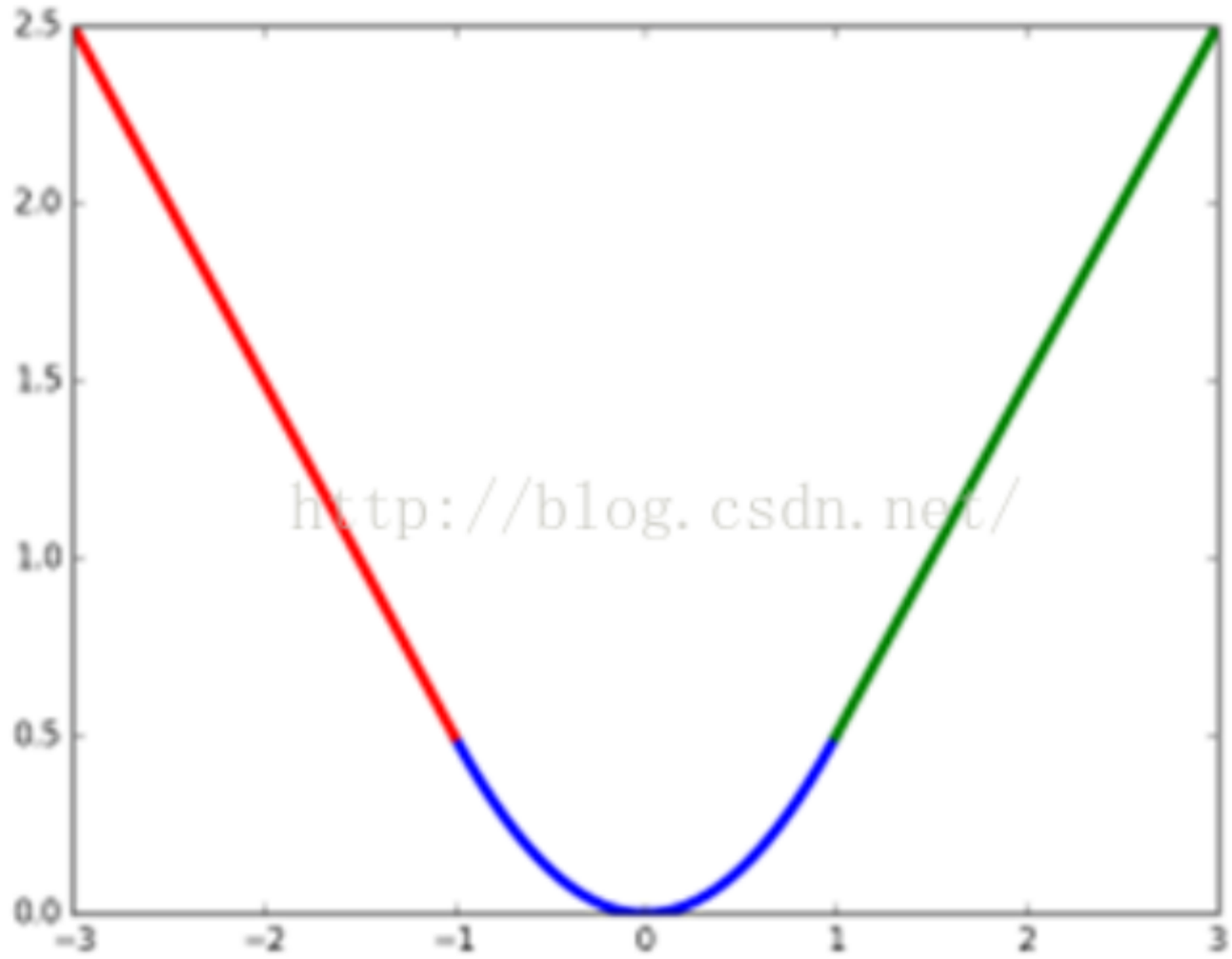
localization loss =

예측 값 파라미터와 GT 값 파라미터를 뺀 값을 SmoothL1

g햇 : gt 박스와 경계박스 사이의 조정 값

Single Shot MultiBox Detection

SSD Loss



$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

SSD Loss

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

Soft max

– 레이블과 매칭된 상자는 특정 클래스에대해 높은 값

: 전체 클래스 확률 / 특정 클래스 확률

– 매칭되지 않은 상자는 배경 클래스에 대해 높은 값

: 전체 클래스 확률 / 배경 클래스 확률

SSD 실험

Prediction source layers from:						mAP use boundary boxes?		# Boxes
conv4_3	conv7	conv8_2	conv9_2	conv10_2	conv11_2	Yes	No	
✔	✔	✔	✔	✔	✔	74.3	63.4	8732
✔	✔	✔	✔	✔		**74.6**	63.1	8764
✔	✔	✔	✔			73.8	68.4	8942
✔	✔	✔				70.7	69.2	9864
✔	✔					64.2	64.4	9025
	✔					62.4	64.0	8664

Table 3: **Effects of using multiple output layers.**

conv11_2 : 1 x 1 x 128

conv11_2 로 부터 1x1x(4x(Classes+4)) = 4x(Classes+4)

Single Shot MultiBox Detection

SSD 실험

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	**74.8**	80.4	70.4
Faster [2]	73.2	76.5	79.0	70.9	**65.5**	**52.1**	83.1	84.7	86.4	52.0	**81.9**	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300	72.1	75.2	**79.8**	70.5	62.5	41.3	81.1	80.8	86.4	51.5	74.3	**72.3**	83.5	84.6	80.6	74.5	46.0	71.4	73.8	83.0	69.1
SSD500	**75.1**	**79.8**	79.5	**74.5**	63.4	51.9	**84.9**	**85.6**	**87.2**	**56.6**	80.1	70.0	**85.4**	**84.9**	**80.9**	**78.2**	**49.0**	**78.4**	72.4	**84.6**	**75.5**

Table 1: **PASCAL VOC2007 `test` detection results.** Both Fast and Faster R-CNN use input images whose minimum dimension is 600. The two SSD models have exactly the same settings except that they have different input sizes (300×300 vs. 500×500). It is obvious that larger input size leads to better results.

Single Shot MultiBox Detection

SSD 실험

Method	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	70.0	77.0	78.1	69.3	59.4	38.3	81.6	78.6	86.7	42.8	78.8	68.9	84.7	82.0	76.6	69.9	31.8	70.1	**74.8**	80.4	70.4
Faster [2]	73.2	76.5	79.0	70.9	**65.5**	**52.1**	83.1	84.7	86.4	52.0	**81.9**	65.7	84.8	84.6	77.5	76.7	38.8	73.6	73.9	83.0	72.6
SSD300	72.1	75.2	**79.8**	70.5	62.5	41.3	81.1	80.8	86.4	51.5	74.3	**72.3**	83.5	84.6	80.6	74.5	46.0	71.4	73.8	83.0	69.1
SSD500	**75.1**	**79.8**	79.5	**74.5**	63.4	51.9	**84.9**	**85.6**	**87.2**	**56.6**	80.1	70.0	**85.4**	**84.9**	**80.9**	**78.2**	**49.0**	**78.4**	72.4	**84.6**	**75.5**

Table 1: **PASCAL VOC2007 `test` detection results.** Both Fast and Faster R-CNN use input images whose minimum dimension is 600. The two SSD models have exactly the same settings except that they have different input sizes (300×300 vs. 500×500). It is obvious that larger input size leads to better results.