SSD : Single Shot multibox Detector

2021210088 허지혜

Contents

Abstract

- A single Deep Neural Network로 Object Detection을 구현한 모형입니다.
- 우리의 approach인 SSD는 다양한 size와 ratio를 가진 default boxes로 각 feature map에서 bounding box를 뽑아낸다.
- predict를 할 때, Network는 각 default box가 각각의 object categories에 속하는 score와 object shape에 잘 맞는 box를 만들어낸다.
- 다양한 feature map을 결합하여 predict에 상ㅇ한다.
- 쉽게, 지금까지의 논문과 다른 점은 object proposal 과정을 없애고 single network에서 convolutional network를 채운다.
- 결과로는, PASCAL VOC 2007에서 74.3%의 정확도(mAP)와 59FPS가 나왔다.

Introduction

- We introduce SSD, a single-shot detector for multiple categories that is faster than the previous state-of-the-art for single shot detectors (YOLO), and significantly more accurate, in fact as accurate as slower techniques that perform explicit region proposals and pooling (including Faster R-CNN).

  -> YOLO보다 빠르고, Faster R-CNN 등 보다 정확하다.

- The core of SSD is predicting category scores and box offsets for a fixed set of default bounding boxes using small convolutional filters applied to feature maps.

  -> SSD의 핵심은 category score(범주 점수)와 small convolutional filter를 적용하여 default bounding boxes(상자) offset을 예측하는 것이다.

- To achieve high detection accuracy we produce predictions of different scales from feature maps of different scales, and explicitly separate predictions by aspect ratio.

  -> 높은 detection 정확도를 달성하기 위해, multi-scale에서의 예측을 생성하고 aspect ratio로 예측을 명시적으로 분리한다.

Introduction

- These design features lead to simple end-to-end training and high accuracy, evenon low resolution input images, further improving the speed vs accuracy trade-off.

  -> 이러한 디자인은 정확도와 스피드의 trade-off 관계를 개선한다.

- Experiments include timing and accuracy analysis on models with varying input size evaluated on PASCAL VOC, COCO, and ILSVRC and are compared to a range of recent state-of-the-art approaches.

  -> PASCAL VOC, COCO, ILSVRC dataset에서 평가된 다양한 input size를 가진 최신 state-of-the-art approaches와 비교된다.

  -> Faster R-CNN : 7FPS with mAP 73.22%

  -> YOLO : 45 FPS with mAP 63.4%

  -> SSD : 59 FPS with mAP 74.3% in VOC 2007 test

2. The Single Shot Detector (SSD)

2.1 Method

- The **SSD** approach is based on **a feed-forward convolutional network** that produces a fixed-size collection of bounding boxes and scores for the presence of object class instances in those boxes, followed by a non-maximum suppression step to produce the final detections.

  -> SSD는 bounding box와 bounding box의 score를 반환하는 feed-forward convolutional network 기반으로 접근한다.

- The early network layers are based on a standard architecture used for high quality image classification (truncated before any classification layers), which we will call the base network.

  -> 초기 Network layer = base Network인 고품질 이미지 분류에 사용되는 표준 Architecture를 기반으로 한다.

  -> 논문에서는 VGG-16을 사용하였다.

- We then add auxiliary structure to the network to produce detections with the following key features:

  -> 그 이후 auxiliary structure를 추가하여 detetions를 생성한다.

2. The Single Shot Detector (SSD)

2.1 Method

- Multi-scale feature maps for detection
  - ✓ base network의 끝에 convoultional feature layers를 추가한다.
  - ✓ 이 layers는 점진적으로 size를 줄이고 multi-scale에서 detection을 예측한다.
  - ✓ detection predicting을 위한 convolutional model은 각 feature layer마다 다르다. [Overfeat and YOLO 참고]

- Convolutional predictors for detection
  - ✓ 추가된 각 feature layer는 a set of convolutional filters를 사용하여 a set of detection prediction를 만들 수 있다. 이는 SSD Network 상단에 표시된다.(Fig2)
  - ✓ (m,n,p) size의 feature layer의 경우 kernel은 (3,3,p)의 convoltional detector를 사용한다. 이는 category score와 default box 좌표들의 상대적인 offset을 출력한다.

3. Experimental Results

- Base network
  - ✓ Base network는 ILSVRC CLC-LOC dataset으로 pretrained된 VGG-16이다.
  - ✓ DeepLab-LargeFOV와 유사하게, 우리는 fc6과 fc7을 convolutional layers로 변경하고 fc6과 fc7로부터 subsample parameters
  - ✓ pool5에서 2x2-s2에서 3x3-s1을 변경하였다.
  - ✓ "holes"(구멍)를 채우기 위해 a trous algorithm을 사용하였다.
    (DeepLab-LargeFOV 일부)
  - ✓ dropout + fc8 layer를 삭제하였다.

  [Fine-tuning]
  - ✓ SGD with initial learning rate , 0.9 momentum, 0.0005 weight decay
  - ✓ batch-size 32
  - ✓ 이때 learning rate decay는 dataset마다 약간 다르다.
  - ✓ 코드는 Caffe로 구현하였다.

2. The Single Shot Detector (SSD)

2.1 Method

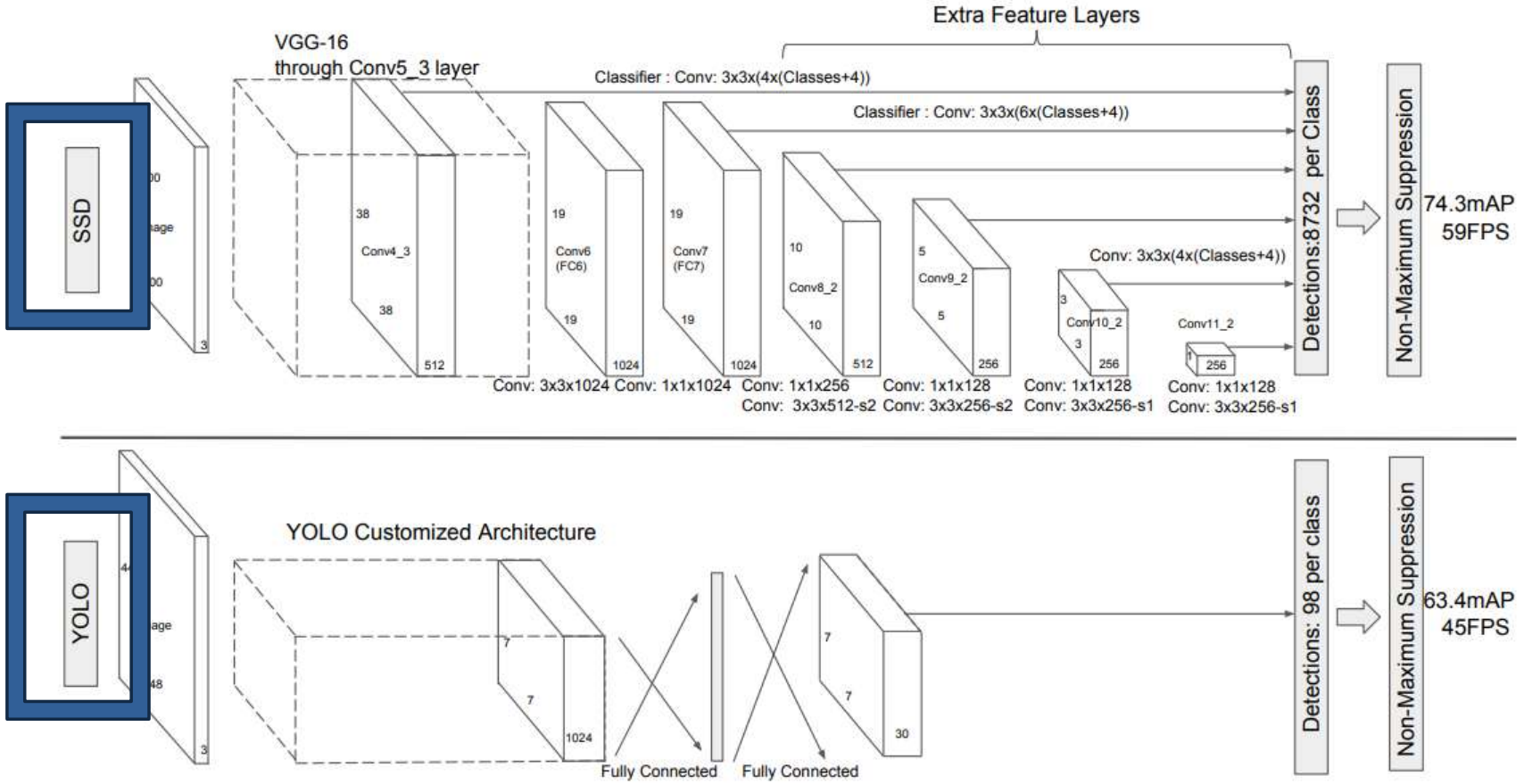
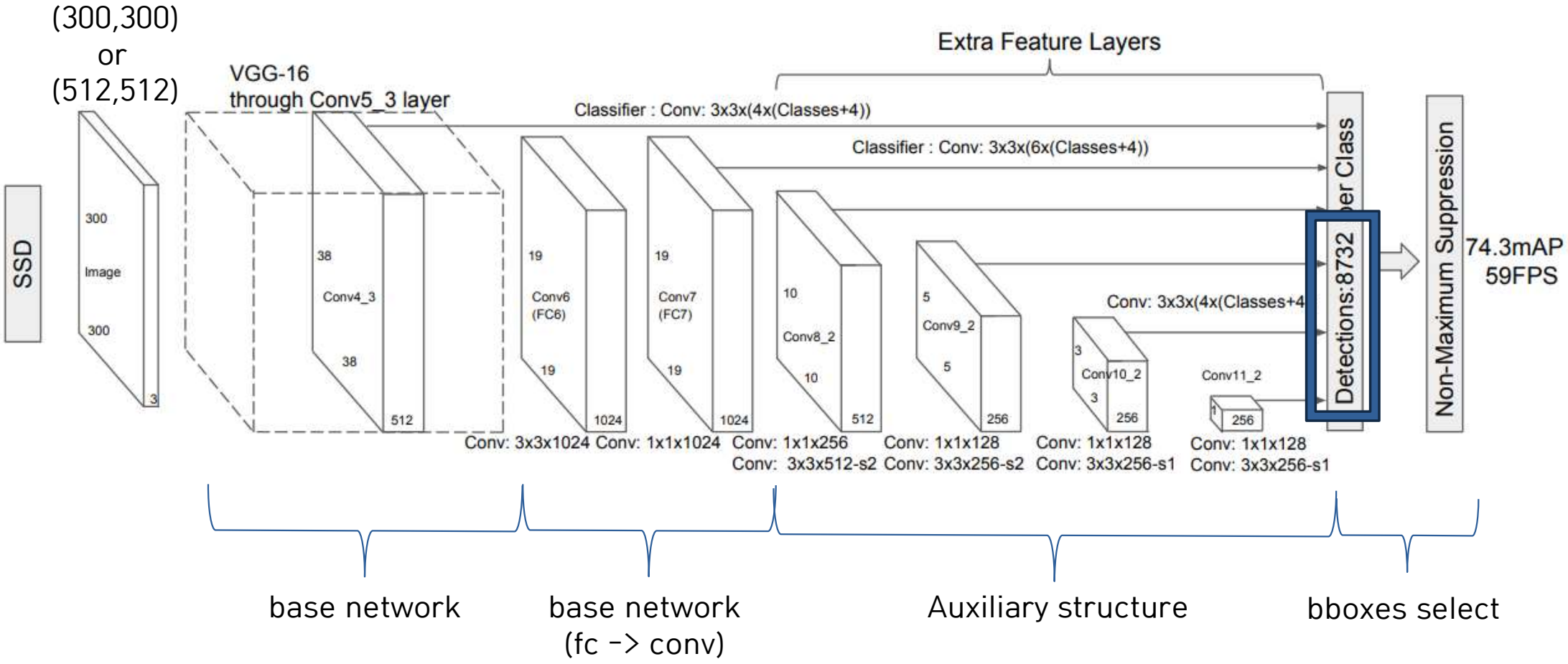


Fig 2

2. The Single Shot Detector (SSD)

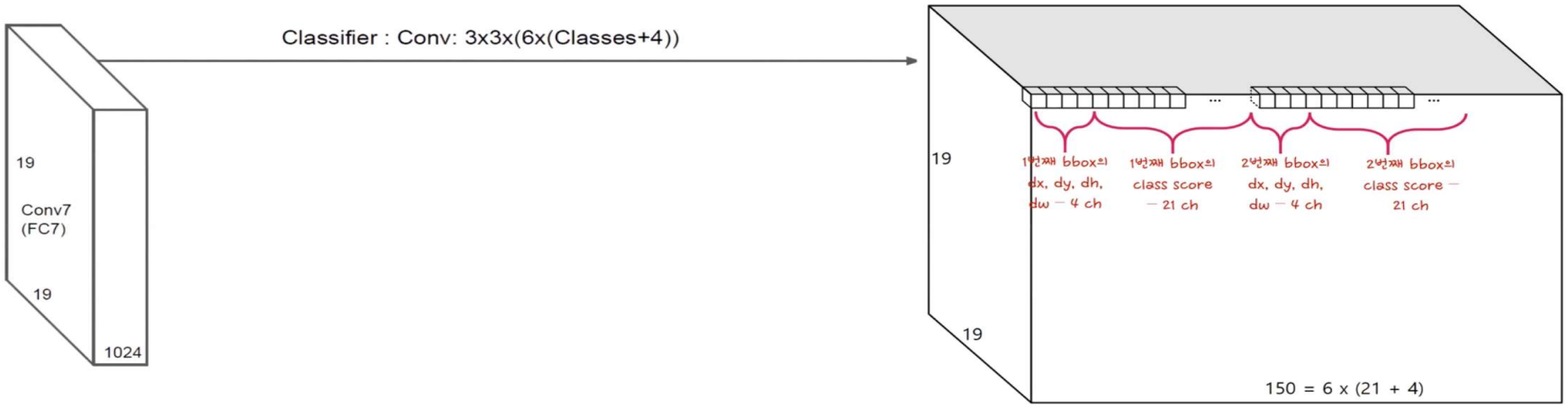
2.1 Method



2. The Single Shot Detector (SSD)

2.1 Method

- Default boxes and aspect ratios(종횡비 가로세로비율)
  - ✓ Feature map의 각 cell마다 a set of default bounding boxes가 만들어진다.
  - ✓ default box와 Matching되는 자리에서 predict되는 box의 offset과 per-class scores(box 안 object 존재 유무)를 예측한다.
  - ✓ k개의 cell 위치가 있고, c개의 class와 4개의 offset 정보를 계산해야 한다면 각 셀마다 k(c+4)개의 filter를 가지게 되어 m*n 크기의 feature map은 m*n*k*(c+4)개의 output을 가지게 된다.



2. The Single Shot Detector (SSD)

2.2 Training

- Matching strategy
  - ✓ 학습하는 동안 어떤 default box가 grounf truth box에 해당하는지 결정하고 그에 따라 network를 학습해야 한다.
  - ✓ 이를 위해 default box와 ground truth box를 대응시키는데, ground truth box와의 **IoU**가 0.5 이상인 default box를 positive sample로 설정한다.
  - ✓ IoU가 가장 높은 box만 positive sample로 사용하는 것보다 0.5 이상인 box를 다 사용할 때가 학습 문제를 단순화 시켜 더 높은 성능의 예측을 수행한다.

2. The Single Shot Detector (SSD)

2.2 Training

- Training objective
  - ✓ MultiBox의 목표에서 파생된 SSD 훈련 목표는 multiple object categories를 처리하도록 확장되었다.

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

  - ✓ 전체 loss 함수는 다음과 같다.
  - ✓ N : ground truth와 maching된 default box의 개수
  - ✓ N = 0, loss = 0
  - ✓ alpha = 1로 논문에서 사용한다.
  - ✓ Faster R-CNN과 비슷한 Loss 함수이다.

2. The Single Shot Detector (SSD)

2.2 Training

- Training objective
  - ✓ Bounding box 예측에 사용된 loss는 다음과 같다.

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

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

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

  - ✓ smooth L1 loss이다. $smooth_{L1}(x) = \begin{cases} 0.5x^2 & if\ |x| < 1 \\ |x| - 0.5 & overwise \end{cases}$
  - ✓ l : predicted box, g : ground box, d : default box
  - ✓ default box를 나눠 bounding box를 정규화 시키고 log를 이용하여 w,h도 정규화 시킨다.

2. The Single Shot Detector (SSD)

2.2 Training

- Training objective
  - ✓ 분류에 사용된 loss 함수는 다음과 같다.

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

  - ✓ softmax를 사용한 cross entropy loss 함수이다.
  - ✓ c : multiple classses confidences
  - ✓x_{ij}^{p} 에서 특정 grid의 i번째 default box가 p class의 j번재 ground truth box와 maching(IoU > 0.5) 된다. => 1

2. The Single Shot Detector (SSD)

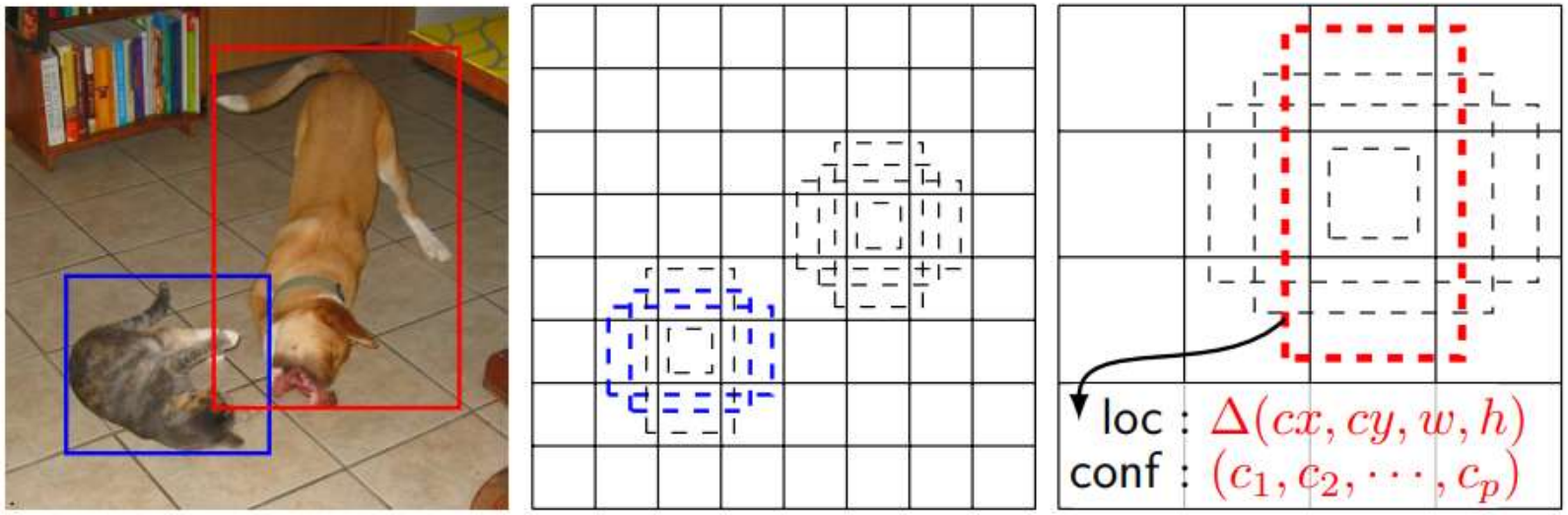
2.2 Training

- Choosing scales and aspect ratios for default boxes
  - ✓ SSD는 각각 다른 muiti-scale의 feature maps를 통해 예측을 수행한다. 즉, 38x38, 19x19, 10x10, 5x5, 3x3, 1x1 6개의 scale의 feature map의 각 cell마다 default box를 생성한다.
  - ✓ 각각 detection을 수행하는 feature map에서 default box의 scale을 수식으로 정의한다.

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

  - ✓ 이렇게 나온 는 원본 이미지에 대한 비율을 나타낸다.
  - ✓ 각 feature map의 cell의 중앙이 default box의 중앙으로 향한다.
  - ✓ 다양한 scale과 aspect ratio를 통해 생성된 많은 default boxes를 예측에 사용하여 input image에 속한 디양한 object의 size와 shape를 포함하는 예측을 수행한다.

2. The Single Shot Detector (SSD)



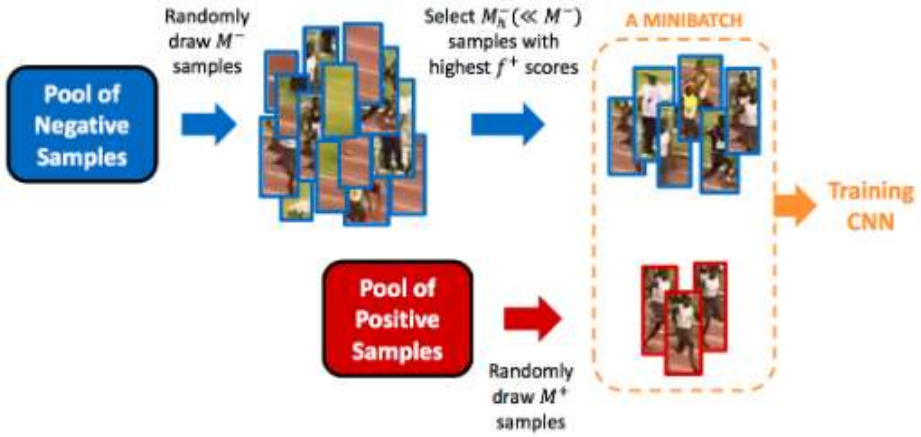
(a) Image with GT boxes (b) 8×8 feature map (c) 4×4 feature map

- SSD framework
- (a) Imagae with Ground Truth boxes ex) 100x100
- (b) 8x8 feature map -> small object detection
- (c) 4x4 feature map -> big object detection
- (b), (c)에 있는 점선 = default box

2. The Single Shot Detector (SSD)

2.2 Training

- Hard negative mining
  - ✓ Matching step 후, 대부분의 default boxes는 negatives이다.
  - ✓ 이는 training examples에서 positive와 negative 사이의 **불균형**을 의미한다.
  - ✓ 이를 해결하기 위해 hard negative mining을 수행한다.
  - ✓ Hard negative mining은 모형이 예측에 실패하는 어려운(hard) samples를 모으는 기법으로 이를 통해 수집된 데이터를 활용해 모형을 더 강하게 training하는 것이 가능해진다.
  - ✓ positive/negative samples의 비율 = 1:3 이 되도록 한다.



2. The Single Shot Detector (SSD)

2.2 Training

- Data augmentation
  - ✓ 다양한 input object sizes 및 shape에 대하여 모형을 보다 강력하게 만들기 위해 각 training image는 다음 옵션 중 하나로 무작위로 샘플링된다.
    - 전체 original input image를 사용
    - objects와의 overlap이 0.1, 0.3, 0.5, 0.7, 0.9로 patch를 샘플링한다.
    - 무작위로 patch를 샘플링한다.
  - ✓ 샘플링된 patch의 크기는 [0.1,1]이며 aspect ratio는 [1,2]
  - ✓ 중심이 맞으면 ground truth box의 overlap 부분을 유지한다.
  - ✓ 0.5의 확률로 horization 한다.

3. Experimental Results

3.1 PASCAL VOC2007

✓ 기존에 높은 성능을 보이는 R-CNN 계열보다 더 높은 성능을 보이는 것을 확인할 수 있다.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast [6]	07	66.9	74.5	78.3	69.2	53.2	36.6	77.3	78.2	82.0	40.7	72.7	67.9	79.6	79.2	73.0	69.0	30.1	65.4	70.2	75.8	65.8
Fast [6]	07+12	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]	07	69.9	70.0	80.6	70.1	57.3	49.9	78.2	80.4	82.0	52.2	75.3	67.2	80.3	79.8	75.0	76.3	39.1	68.3	67.3	81.1	67.6
Faster [2]	07+12	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
Faster [2]	07+12+COCO	78.8	84.3	82.0	77.7	68.9	65.7	88.1	88.4	88.9	63.6	86.3	70.8	85.9	87.6	80.1	82.3	53.6	80.4	75.8	86.6	78.9
SSD300	07	68.0	73.4	77.5	64.1	59.0	38.9	75.2	80.8	78.5	46.0	67.8	69.2	76.6	82.1	77.0	72.5	41.2	64.2	69.1	78.0	68.5
SSD300	07+12	74.3	75.5	80.2	72.3	66.3	47.6	83.0	84.2	86.1	54.7	78.3	73.9	84.5	85.3	82.6	76.2	48.6	73.9	76.0	83.4	74.0
SSD300	07+12+COCO	79.6	80.9	86.3	79.0	**76.2**	57.6	87.3	88.2	88.6	60.5	85.4	**76.7**	**87.5**	**89.2**	84.5	81.4	55.0	81.9	**81.5**	85.9	78.9
SSD512	07	71.6	75.1	81.4	69.8	60.8	46.3	82.6	84.7	84.1	48.5	75.0	67.4	82.3	83.9	79.4	76.6	44.9	69.9	69.1	78.1	71.8
SSD512	07+12	76.8	82.4	84.7	78.4	73.8	53.2	86.2	87.5	86.0	57.8	83.1	70.2	84.9	85.2	83.9	79.7	50.3	77.9	73.9	82.5	75.3
SSD512	07+12+COCO	**81.6**	**86.6**	**88.3**	**82.4**	76.0	**66.3**	**88.6**	**88.9**	**89.1**	**65.1**	**88.4**	73.6	86.5	88.9	**85.3**	**84.6**	**59.1**	**85.0**	80.4	**87.4**	**81.2**

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. 512×512). It is obvious that larger input size leads to better results, and more data always helps. Data: "07": VOC2007 `trainval`, "07+12": union of VOC2007 and VOC2012 `trainval`. "07+12+COCO": first train on COCO `trainval35k` then fine-tune on 07+12.

3. Experimental Results

3.2 Model analysis

- ✓ SSD를 더 잘 이해하기 위해 통제된 실험을 수행하였다.
- ✓ 같은 setting을 하고 input size를 (300,300)으로 통일한 상태로 진행하였다.

	SSD300				
more data augmentation?		✔	✔	✔	✔
include $\{\frac{1}{2}, 2\}$ box?	✔		✔	✔	✔
include $\{\frac{1}{3}, 3\}$ box?	✔			✔	✔
use atrous?	✔	✔	✔		✔
VOC2007 `test` mAP	65.5	71.6	73.7	74.2	**74.3**

Table 2: **Effects of various design choices and components on SSD performance.**

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.**

- ✓ Data Augmentation 중요? -> 성능 8.8% 향상
- ✓ 많은 default box? -> box 예측이 더 쉬워짐
- ✓ 여러 feature map을 사용? -> ㄴㄴ 5개만 했을 때 성능이 가장 좋았다.

3. Experimental Results

3.2 Model analysis

- ✓ small object에서 accuracy가 떨어지는 단점이 있다. 따라서 data augmentation 기법을 사용하여 해결하였다.

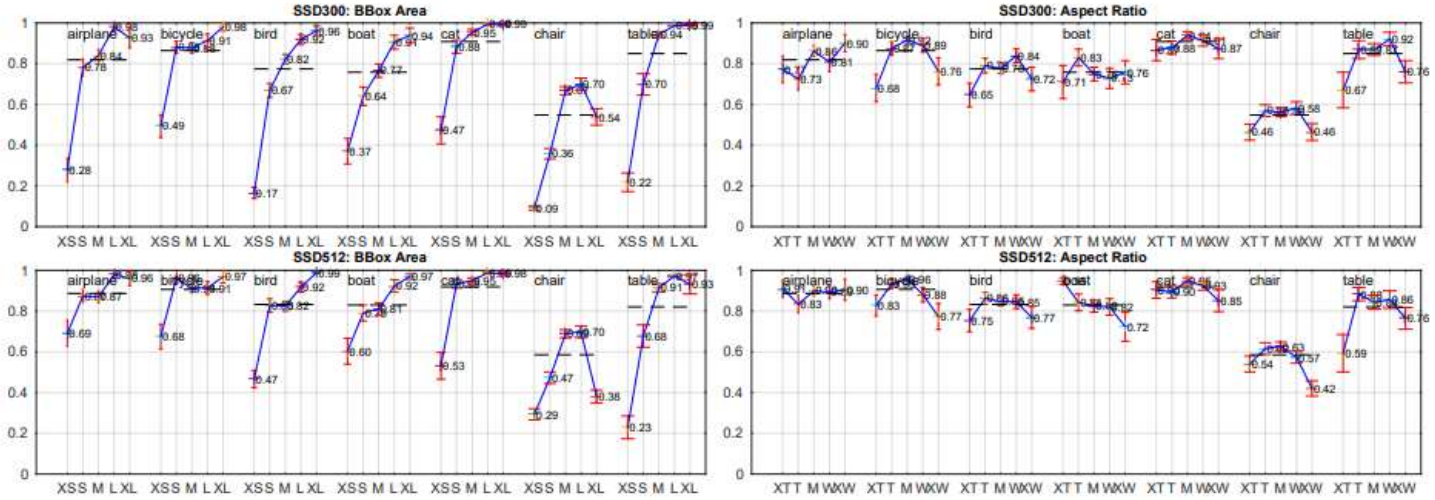


Fig. 4: **Sensitivity and impact of different object characteristics on VOC2007 `test` set using [21].** The plot on the left shows the effects of BBox Area per category, and the right plot shows the effect of Aspect Ratio. Key: BBox Area: XS=extra-small; S=small; M=medium; L=large; XL =extra-large. Aspect Ratio: XT=extra-tall/narrow; T=tall; M=medium; W=wide; XW =extra-wide.

3. Experimental Results

3.3 PASCAL VOC2012

- Multiple output layers at different resolutions is better.
  - ✓ Training data가 늘어나면 accuracy가 늘어난다.
  - ✓ input의 해상도를 높일수록 accuracy가 늘어난다.

Method	data	mAP	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv
Fast[6]	07++12	68.4	82.3	78.4	70.8	52.3	38.7	77.8	71.6	89.3	44.2	73.0	55.0	87.5	80.5	80.8	72.0	35.1	68.3	65.7	80.4	64.2
Faster[2]	07++12	70.4	84.9	79.8	74.3	53.9	49.8	77.5	75.9	88.5	45.6	77.1	55.3	86.9	81.7	80.9	79.6	40.1	72.6	60.9	81.2	61.5
Faster[2]	07++12+COCO	75.9	87.4	83.6	76.8	62.9	59.6	81.9	82.0	91.3	54.9	82.6	59.0	89.0	85.5	84.7	84.1	52.2	78.9	65.5	85.4	70.2
YOLO[5]	07++12	57.9	77.0	67.2	57.7	38.3	22.7	68.3	55.9	81.4	36.2	60.8	48.5	77.2	72.3	71.3	63.5	28.9	52.2	54.8	73.9	50.8
SSD300	07++12	72.4	85.6	80.1	70.5	57.6	46.2	79.4	76.1	89.2	53.0	77.0	60.8	87.0	83.1	82.3	79.4	45.9	75.9	69.5	81.9	67.5
SSD300	07++12+COCO	77.5	90.2	83.3	76.3	63.0	53.6	83.8	82.8	92.0	59.7	82.7	63.5	89.3	87.6	85.9	84.3	52.6	82.5	**74.1**	**88.4**	74.2
SSD512	07++12	74.9	87.4	82.3	75.8	59.0	52.6	81.7	81.5	90.0	55.4	79.0	59.8	88.4	84.3	84.7	83.3	50.2	78.0	66.3	86.3	72.0
SSD512	07++12+COCO	**80.0**	**90.7**	**86.8**	**80.5**	**67.8**	**60.8**	**86.3**	**85.5**	**93.5**	**63.2**	**85.7**	**64.4**	**90.9**	**89.0**	**88.9**	**86.8**	**57.2**	**85.1**	72.8	**88.4**	**75.9**

Table 4: **PASCAL VOC2012 `test` detection results.** Fast and Faster R-CNN use images with minimum dimension 600, while the image size for YOLO is 448 × 448. data: "07++12": union of VOC2007 `trainval` and `test` and VOC2012 `trainval`. "07++12+COCO": first train on COCO `trainval35k` then fine-tune on 07++12.

3. Experimental Results

3.6 Data Augmentation for Small Object Accuracy

- ✓ 기존 image의 16배 되는 빈 image를 만들고 원본을 넣은 뒤 원본 image의 R,G,B 평균값으로 빈 부분을 채우고 다시 resize하여 accuracy를 높였다.
- ✓ * 부분이 data augmentation을 적용한 부분이다.

Method	VOC2007 test		VOC2012 test		COCO test-dev2015		
	07+12	07+12+COCO	07++12	07++12+COCO	trainval35k		
	0.5	0.5	0.5	0.5	0.5:0.95	0.5	0.75
SSD300	74.3	79.6	72.4	77.5	23.2	41.2	23.4
SSD512	76.8	81.6	74.9	80.0	26.8	46.5	27.8
SSD300*	77.2	81.2	75.8	79.3	25.1	43.1	25.8
SSD512*	**79.8**	**83.2**	**78.5**	**82.2**	**28.8**	**48.5**	**30.3**

Table 6: **Results on multiple datasets when we add the image expansion data augmentation trick.** SSD300* and SSD512* are the models that are trained with the new data augmentation.

3. Experimental Results

- Conclusions
  - ✓ 이 논문에서는 multiple categories에 대한 fast single-shot object detector인 SSD를 소개한다. SSD의 핵심 아이디어는 multi-scale convolutional bounding box outputs를 사용하는 것이다.
  - ✓ SSD512 모형은 PASCAL VOC 및 COCO dataset에 대해 정확도 측면에서 Faster R-CNN을 크게 능가하면서도 3배 빠른 결과를 보였다.
  - ✓ 마찬가지로, SSD300모형은 YOLO보다도 현저히 우수한 detection 정확도를 생성하면서도 빠른 53FPS 결과를 보였다.
  - ✓ SSD는 1-stage detector로 2-stage detector 수준의 높은 detection 성능과 빠른 속도를 보여준 모형이다.
  - ✓ 앞으로 연구 방향에서, 비디오에서 동시에 object를 detection하고 tracking하기 위해 RNN을 사용하여 탐색하는 곳에도 잘 사용될 것이다.

Reference

1. SSD 논문 : https://arxiv.org/pdf/1512.02325.pdf
2. SSD 논문 리뷰 :

https://velog.io/@skhim520/SSD-%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0

https://yeomko.tistory.com/20

https://www.youtube.com/watch?v=Gc233mo6r9c

https://www.youtube.com/watch?v=ej1ISEoAK5g&t=949s

끝