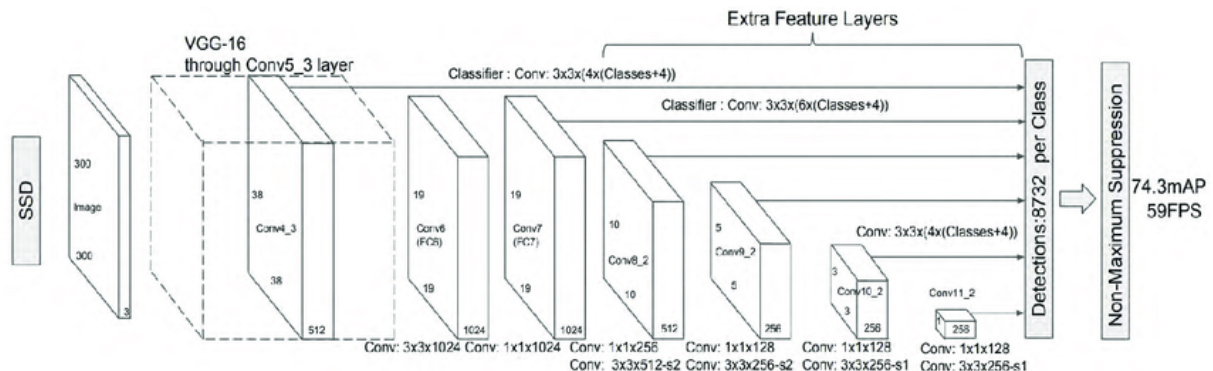


5. SSD(Single Shot MultiBox Detector)



SSD 모델은 VGG16을 Base Network로 사용하고 보조 Network(Auxiliary Network)를 추가한 구조를 가짐

두 Network를 연결하는 과정에서 fc layer를 Conv Layer로 대체하면서 Detection 속도가 향상됨

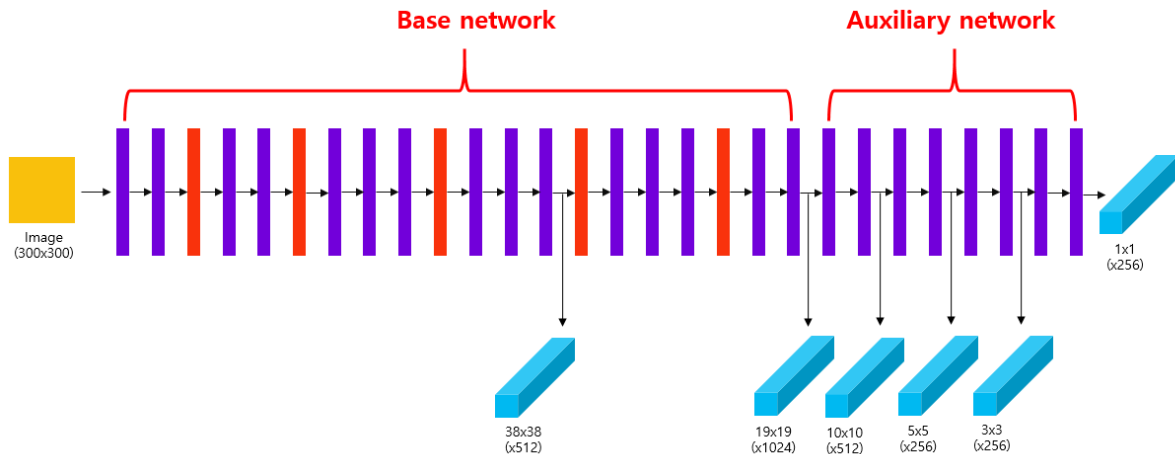
SSD 모델은 Convolutional Network 중간의 Conv Layer에서 얻은 Feature Map을 포함시켜 총 6개의 서로 다른 Scale의 Feature Map을 예측에 사용함

Feature Map의 각 Cell마다 서로 다른 Scale과 Aspect Ratio를 가진 Bounding Box인 Default Box를 사용하여 객체의 위치를 추정

Main Ideas

- Multiscale Feature Maps
 - SSD모델은 하나의 통합된 네트워크로 Detection을 수행하는 1 - Stage Detector
 - VGG16 Pretrained base Network 사용 후 Auxiliary Network(보조 네트워크)를 추가한 구조
 - 보조 Network는 일반적인 Conv Layer로 구성
 - Base Network는 후반부에 등장하는 Fc Layer를 Conv Layer로 바꿔주어 보조 네트워크와 연결
 - Fc Layer가 제거되면서 Detection 속도가 향상됨

- SSD 모델의 핵심적인 아이디어는 다양한 scale의 feature map을 사용한다는 점
- 기존의 Detection Model들은 Convolution Network를 거쳐 단일한 Scale을 가진 Feature Map을 사용
- YOLOv1 모델의 경우 7*7(*30) 크기의 feature map만을 사용함
 - 이처럼 단일한 scale의 feature map을 사용할 경우, 다양한 크기의 객체를 포착하는 것이 어렵다는 것이 단점



- Feature map을 추출하는 Conv Layer와 Feature Map을 파악
 - Input : 300 × 300
 - Base_Network Conv4_3 Layer에서 38x38x(512) 크기의 Feature Map 추출
 - Base_Network Conv7에서 19x19x(1024) 크기의 feature Map 추출
 - Auxiliary Network의 Conv8_2, conv9_2, conv10_2, conv11_2 layer에서 각각 10x10x(512), 5x5x(256), 3x3x(256), 1x1x(256) 크기의 feature map을 추출
 - 총 6개의 scale을 가진 feature map을 얻을 수 있어 다양한 scale의 featur Map을 사용 가능

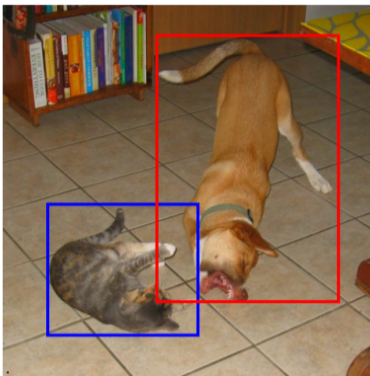
Default Boxes

- 원본 이미지에서 보다 다양한 크기의 객체를 탐지하기 위해 feature map의 각 Cell마다 서로 다른 Scale과 Aspect Ratio(가로세로비)를 가진 Default Box를 생성
 - Default Box는 Faster R-CNN모델에서 사용하는 anchor box와 개념적으로 유사하지만 서로 다른 크기의 feature map에 적용한다는 점에서 차이가 있음

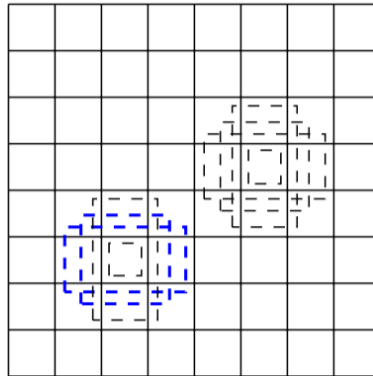
- SSD 모델은 38x38, 19x19, 10x10, 5x5, 3x3, 1x1 총 6개의 scale의 feature map의 각 cell마다 default box를 생성
- Default Box의 Scale = S_k 여기서 S_k 는 원본 이미지에 대한 비율을 의미
 - 원본 이미지의 크기가 300 x 300, s = 0.1 aspect ratio가 1:1일때, Default Box의 크기는
30 x 30 (=300x0.1 x 300x0.1)
 - 각 Feature Map 별로 적용할 Default Box의 Scale을 구하는 공식

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

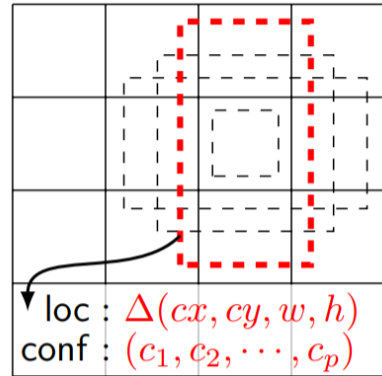
- $s_{min} = 0.2$
 - $s_{max} = 0.9$
 - m : 예측에 사용할 feature map의 수, SSD 모델의 경우 m = 6
- Aspect Ratio이 $a_r \in [1, 2, 3, 1/2, 1/3]$ 이며, default box의 너비 $ww_k^a = s_k \sqrt{a_r}$ 이며, 높이는 $h_k^a = s_k / \sqrt{a_r}$
 - aspect ratio가 1:1인 경우 scale이 $s'_k = \sqrt{s_k s_k + 1}$ 인 default box를 추가적으로 사



(a) Image with GT boxes



(b) 8 x 8 feature map

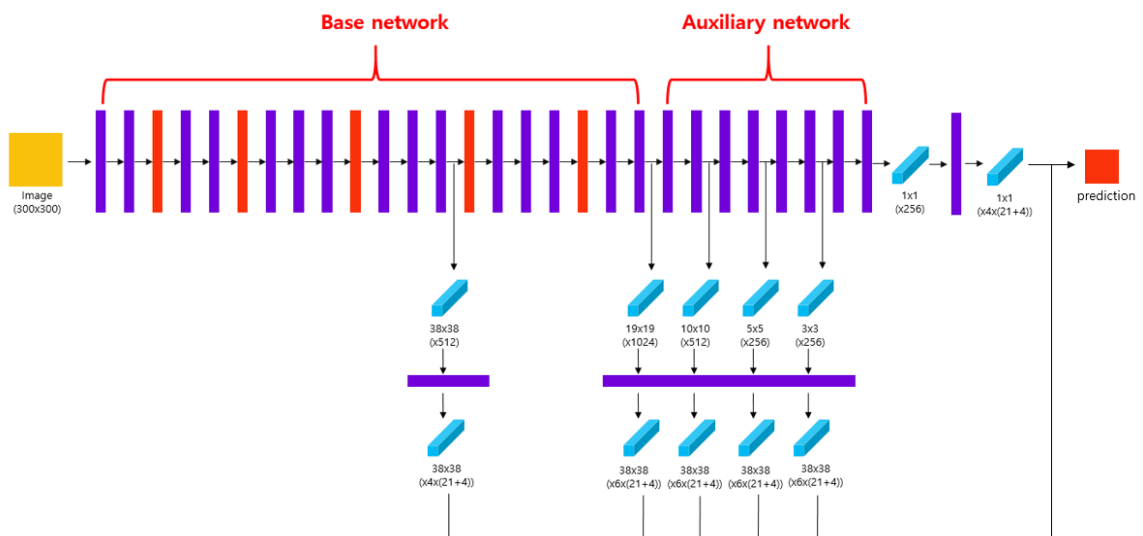


(c) 4 x 4 feature map

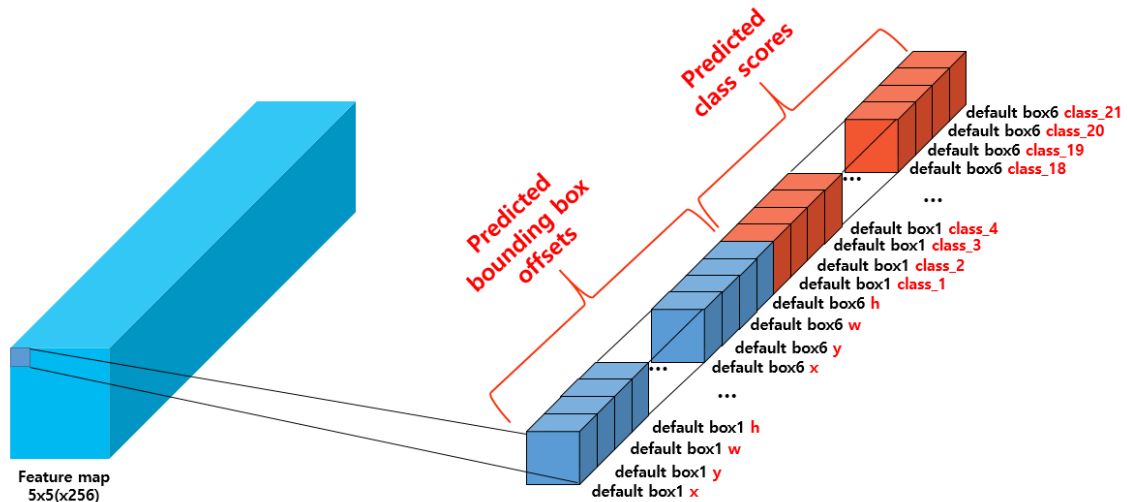
- 첫 번째 feature map(크기가 38x38)의 $s_k = 0.2$, 두 번째 feature map k = 2 이기 때문에 $s_k = 0.34$
마지막 feature map(크기가 1x1)의 경우
 $s_k = 0.9$ 즉, feature map의 scale이 작아질수록 default box의 scale은 커짐

Predictions

- 각각의 feature map은 서로 다른 수의 default box를 적용
 - 첫 번째(38x38)와 마지막(1x1) feature map은 aspect ratio가 1:1, 1:2, 1:1/2인 box와 aspect ratio가 1:1일 때 추가적으로 사용하는 box까지 총 4개의 default box를 적용
 - feature map의 각 Cell마다 4개의 Default Box가 생성됨을 의미
 - 나머지 4개의 Feature Map은 6개의 Default Box를 모두 적용
 - 총 서로 다른 Scale의 Feature Map에 맞는 Default Box를 적용한 경우 총 Default Box의 수는 8732
 $(=38 \times 38 \times 4 + 19 \times 19 \times 6 + 10 \times 10 \times 6 + 5 \times 5 \times 6 + 3 \times 3 \times 6 + 1 \times 1 \times 4)$



- 최종 예측을 위해 서로 다른 scale의 feature map을 추출한 후 3x3(stride=1, padding=1) conv 연산을 적용
 - default box의 수를 k , 예측하려는 class의 수를 c 라고 할 때, **output feature map의 channel 수는 $k(4+c)k(4+c)$** 가 되도록 설계
 - 이는 각 feature map의 cell이 k 개의 default box를 생성하고 각 box마다 4개의 offset과 class score를 예측한다는 것을 의미



- SSD 모델은 예측하려는 class의 수가 20개인 PASCAL VOC 데이터셋을 사용해 학습을 진행
 - 따라서 class의 수는 배경을 포함하여 $c = 21$
 - 예를들어 $5 \times 5 \times 256$ 크기의 feature map을 추출할 경우 $k = 6$, $c = 21$ 이므로 conv연산을 적용한 output feature map의 크기는 $5 \times 5 \times 6 \times (4 + 21)$

Matching Strategy

- default box의 학습 대상을 지정하기 위해 어떤 default box가 어떤 ground truth box와 대응하는지 결정해야함
 - 이를 위해 **default box와 ground truth box를 매칭하는 작업**이 필요
 - 먼저 ground truth box와 가장 큰 **jaccard overlap(IOUS)**를 가지는 box와 **jaccard overlap이 0.5 이상인 box는 모두 positive로 label**
 - 반대로 ground truth box와 0.5 미만의 jaccard overlap을 가지는 box는 negative로 label
 - 일반적으로 이미지 내에서 배경(background)에 해당하는 box가 많기 때문에 negative sample의 수가 positive sample의 수보다 훨씬 많음
 - 이로 인해 클래스 불균형(class imbalance) 문제가 발생
 - 이를 해결하기 위해 높은 confidence loss를 가진 sample을 추가하는 hard negative mining을 수행, 이 때 positive/negative sample의 비율은 1:3이 되도록 함

Loss function

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

- SSD모델의 loss function은 confidence loss인 L_{conf} 와 localization loss인 L_{loc} 의 합으로 구성

α 는 두 loss 사이의 가중치를 조절하는 **balancing parameter**로 디폴트값으로 $\alpha = 1$ 을 사용

N 은 ground truth box와 매칭된 Default Box의 수. 만약 $N = 0$ 이라면 $Loss = 0$

- localization loss는 Faster R-CNN 모델과 마찬가지로 Default Box의 중심 좌표(c_x, c_y)와 너비와 높이 (w, h)를 사용하여 smooth L1 loss를 통해 구함
 - l 은 예측한 box의 파라미터(좌표), g 는 ground truth box의 파라미터(좌표)
 - x_{ij}^k 는 i 번째 Default Box와 Class가 K 인 j 번째 ground truth box와의 매칭 여부를 알려주는 indicator parameter로, 매칭될 경우 1, 그렇지 않을 경우 0
- confidence loss는 모든 class에 대한 loss를 softmax loss를 통해 계산

Training SSD

1. 전체 Network 구성(Base Network + Auxiliary Network)

- a. 학습을 위해 base network와 auxiliary network를 합쳐 전체 네트워크를 구성
- b. pre-trained된 VGG16 모델을 불러와 마지막 2개의 fc layer를 conv layer로 대체
- c. 최종 output feature map의 크기가 1x1이 되도록 auxiliary network를 설계

2. 이미지 입력 및 서로 다른 scale의 feature map 얻기

- a. SSD network에 300x300 크기의 이미지를 입력
- b. 이후 전체 network 구간 중 conv4_3, conv7, conv8_2, conv9_2, conv10_2, conv11_2 layer에서 각각 feature map을 추출
 - **Input** : 300×300 sized image
 - **Process** : feature map extraction
 - **Output**
 - 38×38(x512) sized feature map
 - 19×19(x1024) sized feature map
 - 10×10(x512) sized feature map

- $5 \times 5(x256)$ sized feature map
- $3 \times 3(x256)$ sized feature map
- $1 \times 1(x256)$ sized feature map

3. 서로 다른 scale의 feature map에 conv 연산 적용

- 앞의 과정에서 얻은 서로 다른 scale의 feature map에 3×3 conv(stride=1, padding=1) 연산을 적용
- 이 때 각 feature map마다 서로 다른 수의 default box를 사용함에 주의
 - **Input** : 6 feature maps
 - **Process** : 3×3 conv(stride=1, padding=1)
 - **Output**
 - $38 \times 38(x4 \times (21+4))$ sized feature map
 - $19 \times 19(x6 \times (21+4))$ sized feature map
 - $10 \times 10(x6 \times (21+4))$ sized feature map
 - $5 \times 5(x6 \times (21+4))$ sized feature map
 - $3 \times 3(x6 \times (21+4))$ sized feature map
 - $1 \times 1(x4 \times (21+4))$ sized feature map

4. 전체 feature map 병합

- 3)번 과정에서 얻은 모든 feature map을 $8732 \times (21+4)$ 크기로 병합
- 이를 통해 default box별로 bounding box offset 값과 class score를 파악 가

5. loss function을 통해 SSD network 학습

- 위에서 얻은 feature map과 ground truth 정보를 활용하여 localization loss를 계산
- 이후 negative sample에 대하여 Cross entropy loss를 구한 후 loss에 따라 내림차순으로 정렬
- 그 다음 negative sample에서 loss가 높은 순으로 positive sample의 3배만큼의 수를 추출
- 이러한 **hard negative mining** 과정을 통해 얻은 hard negative sample과 positive sample을 사용하여 confidence loss를 계산

- e. 앞서 얻은 localization loss와 confidence loss를 더해 최종 loss를 구한 후 backward pass를 수행하여 network를 학습


Detection

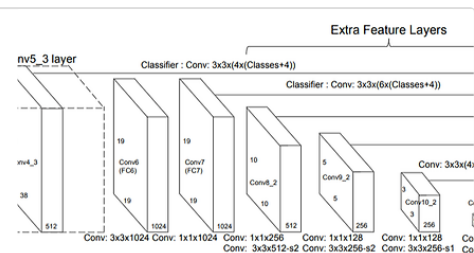
- SSD 모델은 detection 시, 마지막 예측에 대하여 **Non maximum suppression**을 수행
- 이를 통해 겹치는 default box를 적절하게 제거하여 정확도 향상

<https://herbwood.tistory.com/15>

SSD 논문(SSD: Single Shot MultiBox Detector) 리뷰

이번 포스팅에서는 SSD 논문(SSD: Single Shot MultiBox Detector)을 읽고 정리해봤습니다. RCNN 계열의 2-stage detector는 region proposals와 같은 다양한 view를 모델에 제공하

 <https://herbwood.tistory.com/15>



해당 논문은 위 사이트의 정리글을 보고 공부했습니다.