

Cascade R-CNN: Delving into High Quality Object Detection

📎 파일

비어 있음

🔗 URL

비어 있음

☰ 분야

Object Detection

☰ 이해도

상

발행년도

2017

+ 속성 추가

இ 댓글 추가

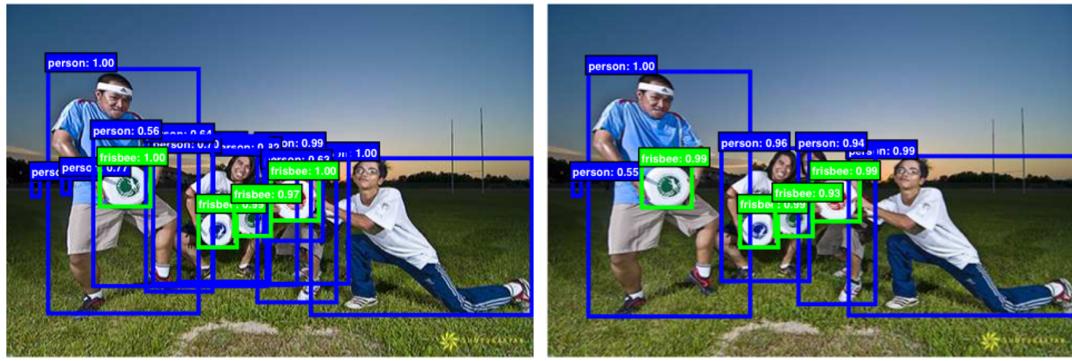
1. 논문이 풀고자 하는 문제

일반적으로 $IOU=0.5$ 로 두고 훈련 -> noisy한 bounding box가 잘 생김

반면 $IOU = 0.7$ 을 하면 -> positive sample이 너무 적어져 overfitting

(만약 같은 classifier를 iterative하게 사용한다면? 2번 이상 사용하는건 의미 없음)

따라서 본 논문은 IOU 를 점차 증가시킨 detector를 여러개 쌓아 'close false positives'를 잘 걸러내는 것이 목표

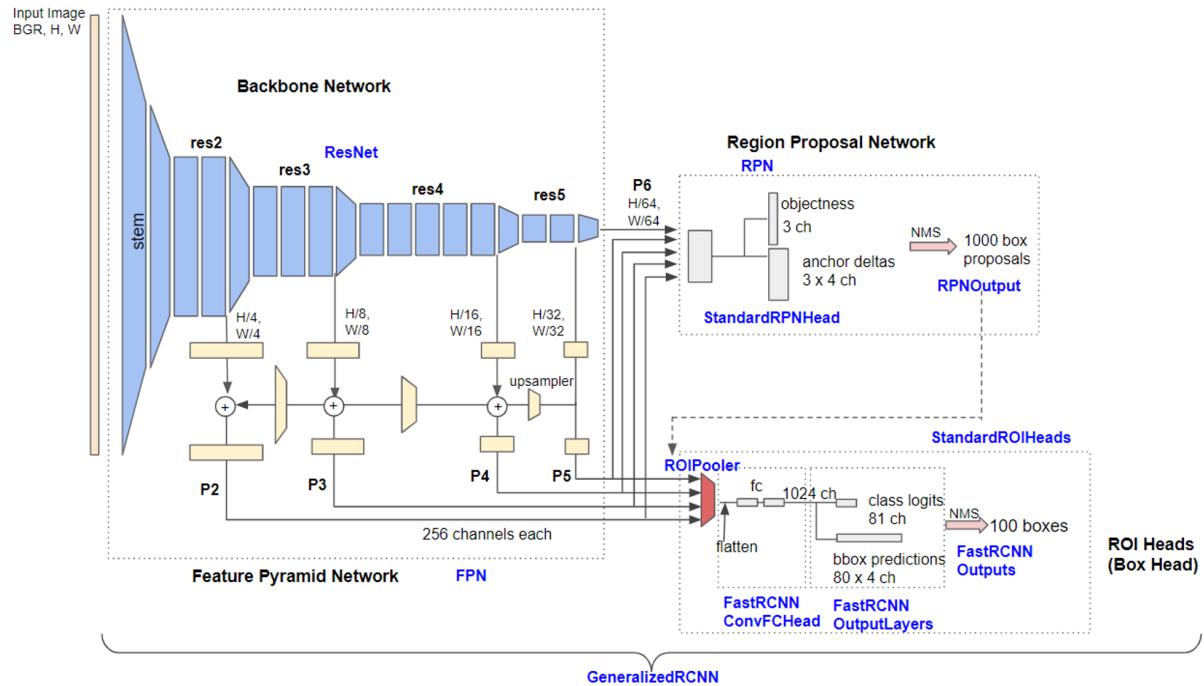


(a) Detection of $u = 0.5$

(b) Detection of $u = 0.7$

2. 기존에 문제를 풀었던 방법

R-CNN => Fast R-CNN => Faster R-CNN => Cascade R-CNN

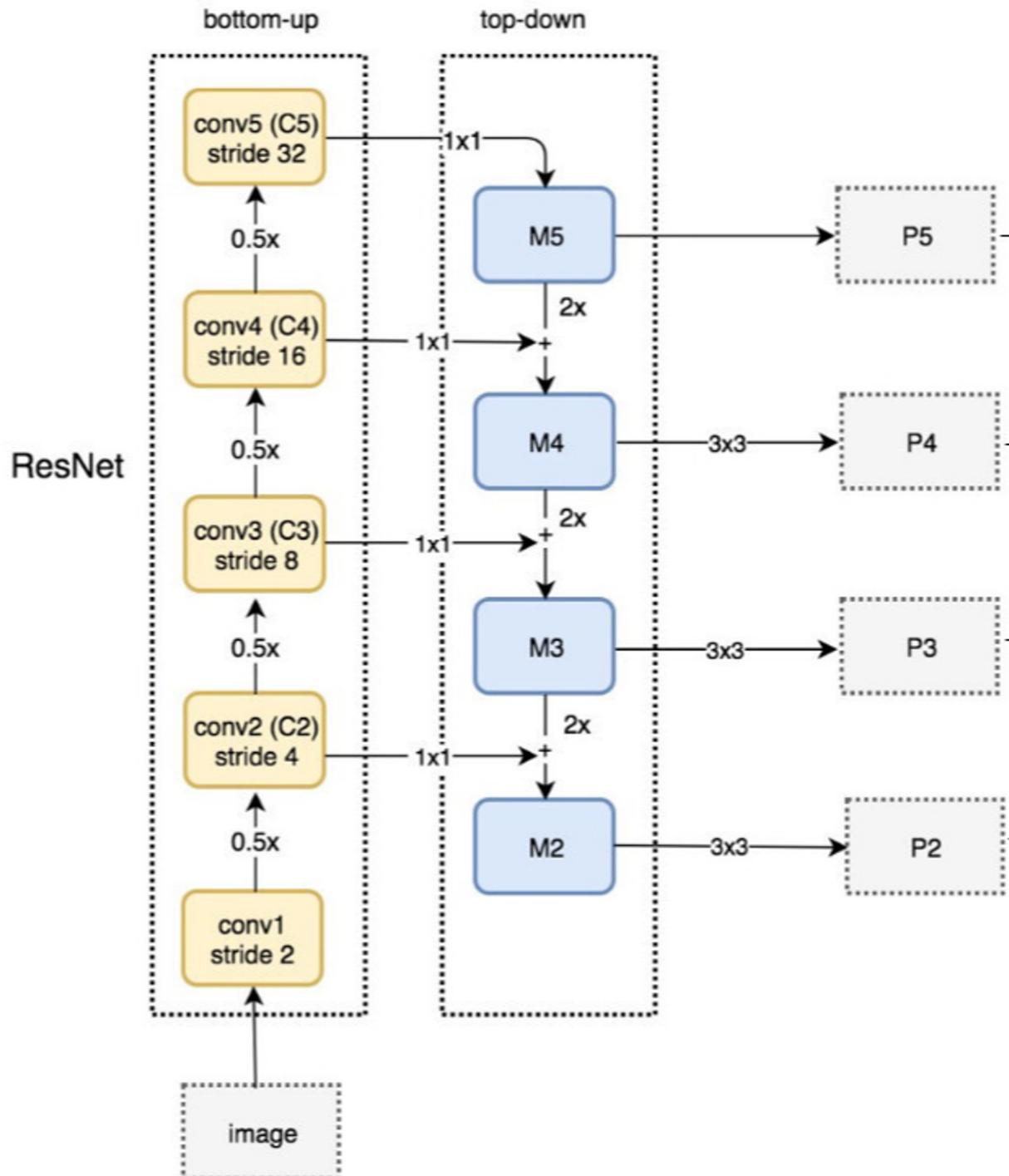


전처리

- rescale
- horizontal random flip
- padding (가장 큰 이미지를 기준으로 zero padding)

Backbone model

- Resnet + FPN(feature pyramid network) 구조



5번의 conv가 이루어지고 stride는 2씩 증가하므로 feature map size는 2배로 작아진다.

1x1 conv로 filter 개수를 맞춰주고 size를 2배로 증가하여 이전 feature map과 더해준다.

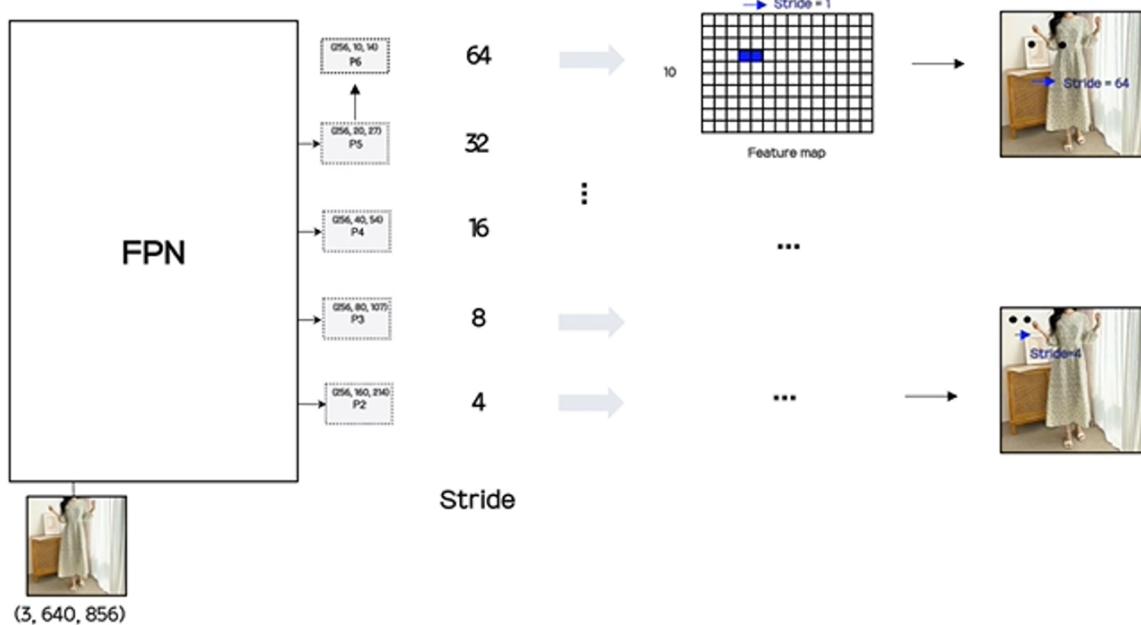
이렇게 나온 multi scale feature map P2, P3, P4, P5는 RPN head로 들어간다

Proposal

- RPN
- Anchor box

▪ Anchor box

- Receptive field를 활용한 grid cell 정의



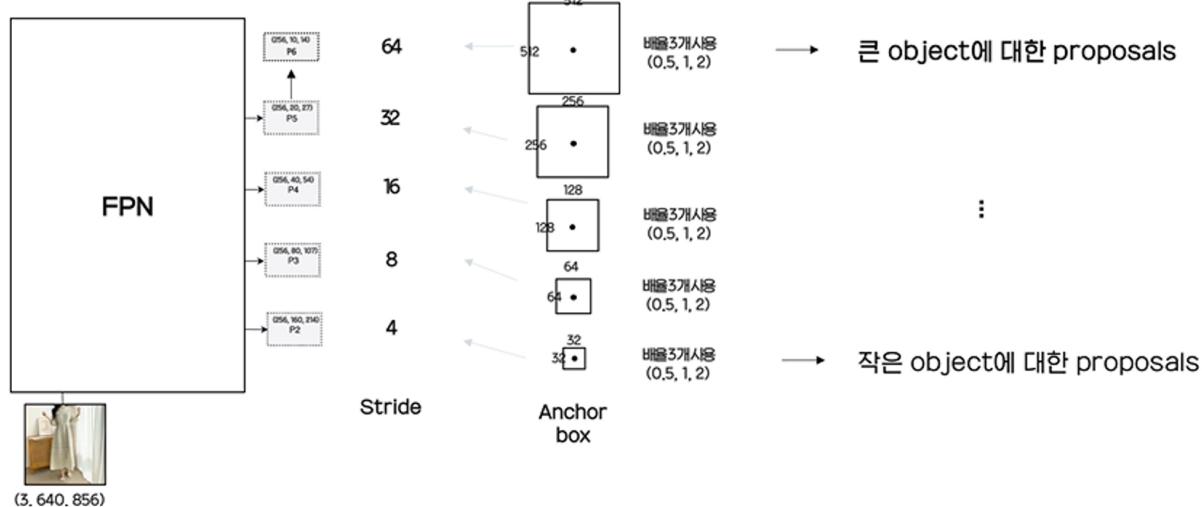
FPN을 통해 구한 P2~P6 (P6은 P5를 max pooling)은 원본 대비 크기가 4배, 8배, 16배, 32배, 64배 감소되었다.

즉 P6에서 stride=1은 원본 이미지에서 stride=64와 동일한 효과이다.

따라서 이에 맞는 anchor box 사이즈를 정의해준다.

▪ Anchor box

- Receptive field를 활용한 grid cell 정의



기본적으로 배율 0.5, 1, 2를 가지는 anchor box 3개를 가지고, 각 map마다 anchor box의 크기를 설정해준다.

이후

(1) anchor box에 object가 있는가? - objectness => binary cross entropy

(2) GT box에 가까운 좌표인가? - localization => L1 loss

수행 목적에 맞게 손실함수를 정해준다.

Pooling

- ROI pooling
- ROI align

FPN을 사용하며 multi scalar feature map이 생성되었으므로 region proposals을 어떤 scale과 매칭시킬지 결정해야 함

$$k = [k_0 + \log_2(\sqrt{wh}/224)]$$

k = pyramid level의 index

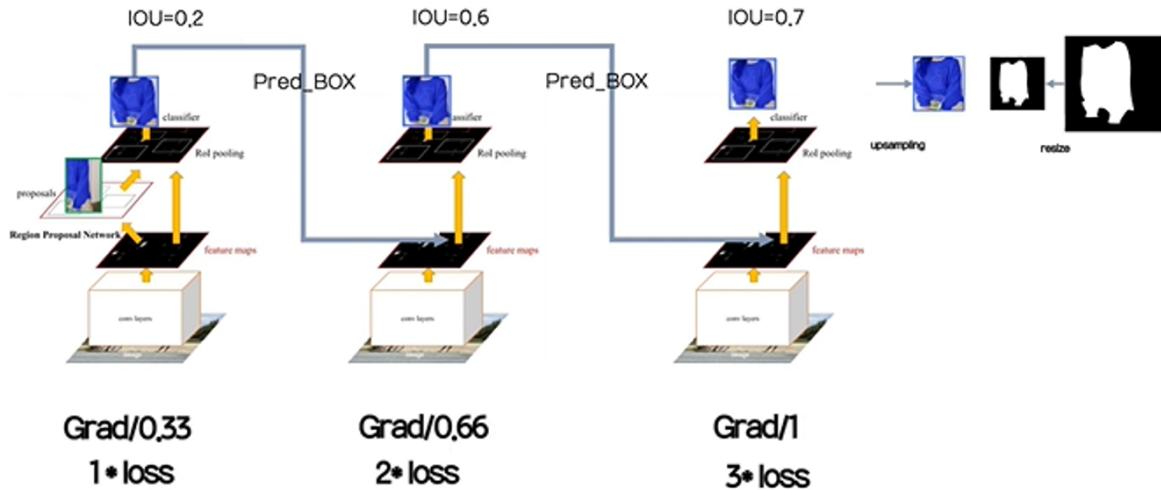
k_0 = target level

w,h = region proposal의 width, height

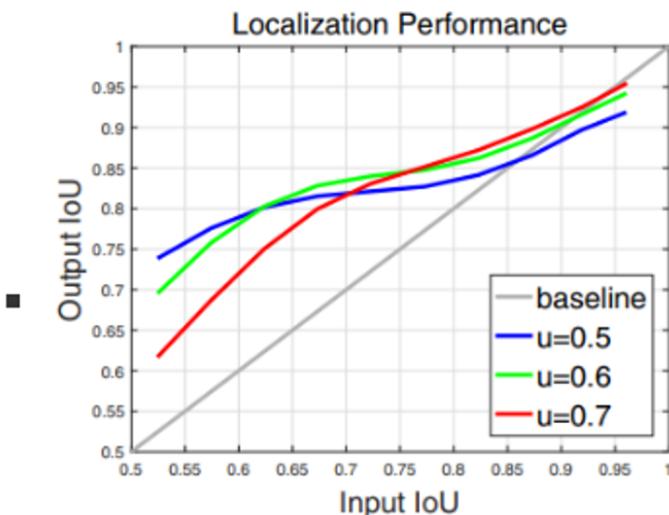
위 공식을 통해 고정된 크기의 feature map을 얻을 수 있음

3. 논문에서 제시한 아이디어

Cascade R-CNN



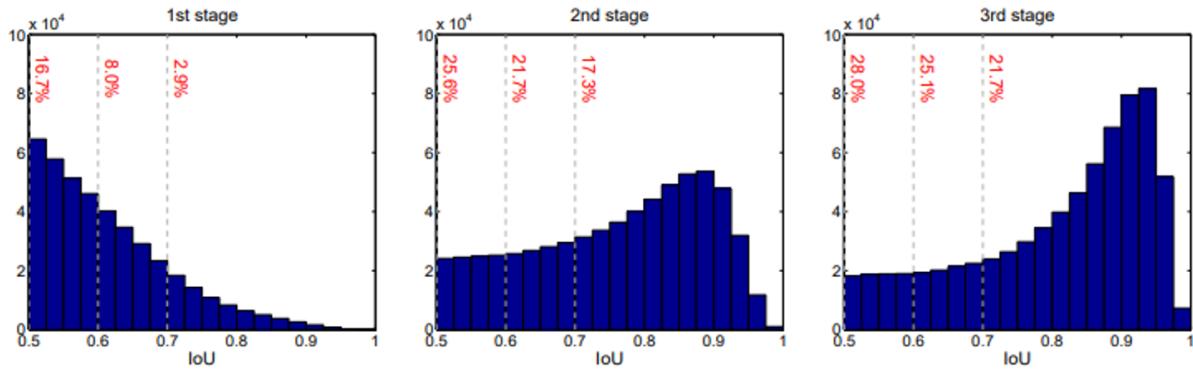
Faster R-CNN에서 이전의 Predict 값을 다음 proposal로 대체하여 사용한다.



위 그래프를 보면 input IoU 보다 output IoU 값이 잘 나온다는 것을 알 수 있다. (회색 라인 위쪽에 위치한다)

즉 낮은 값의 IOU에 훈련된 bounding box regressor는 더 높은 IOU를 생성하므로, 검출기 품질(IOU 임계값)이 증가하더라도 positive examples 집합을 유지할 수 있다.

- > 과적합 없음, 더 깊은 검출기는 더 좋은 예제를 가지게 된다.



위 그래프를 보면 1st stage에 0.7 이상은 2.9% 밖에 되지 않는다. 오버피팅이 빨리 될 수 밖에 없다.

stage가 깊어질 수록 더 좋은 예제를 가지는 것을 알 수 있다.

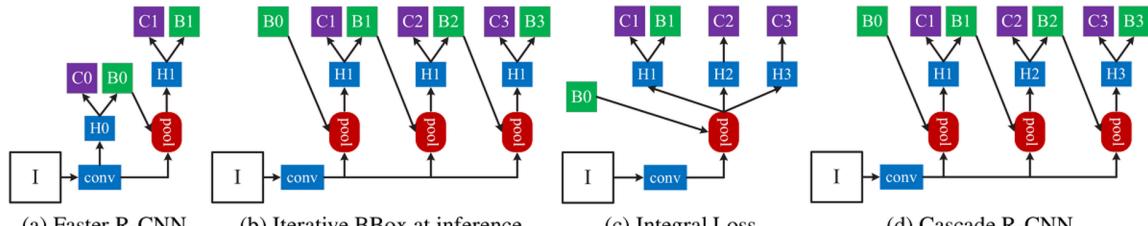


Figure 3. The architectures of different frameworks. “I” is input image, “conv” backbone convolutions, “pool” region-wise feature extraction, “H” network head, “B” bounding box, and “C” classification. “B0” is proposals in all architectures.

(b) iterative BBox at inference

$$f'(x, b) = f \circ f \circ \cdots \circ f(x, b)$$

regressor f 가 $u=0.5$ 에서 반복적으로 학습되면 (classifier가 H_1 로 모두 동일하다)
bounding box의 분포가 상당히 변하는 문제가 있었다. 논문에서는 두 번 이상 반복하는
것은 의미가 없다고 말했다.

(c) Intergral Loss

$$L_{cls}(h(x), y) = \sum_{u \in U} L_{cls}(h_i(x), y_u)$$

U : IOU thresholds

$h(x)$: classifier

classifier를 양상블(ensemble)하는 방식(H_1, H_2, H_3)이다.

이 모델의 문제는 u 에 따라 positive sample의 개수 차이가 크기 때문에 쉽게 overfitting 되었다.

(d) cascade r-cnn

$$f(x, b) = f_T \circ f_{T-1} \circ \cdots \circ f_1(x, b)$$

T: total number of cascode stages

각 regressor f 가 T단계의 샘플 분포에 따라 최적화 된다.

$$b^t = f_{t-1}(x^{t-1}, b^{t-1})$$

g : the ground truth object for x^t

$\lambda = 1$: trade-off coefficient

$[\cdot]$: the indicator function

y^t : the label of x^t given u^t

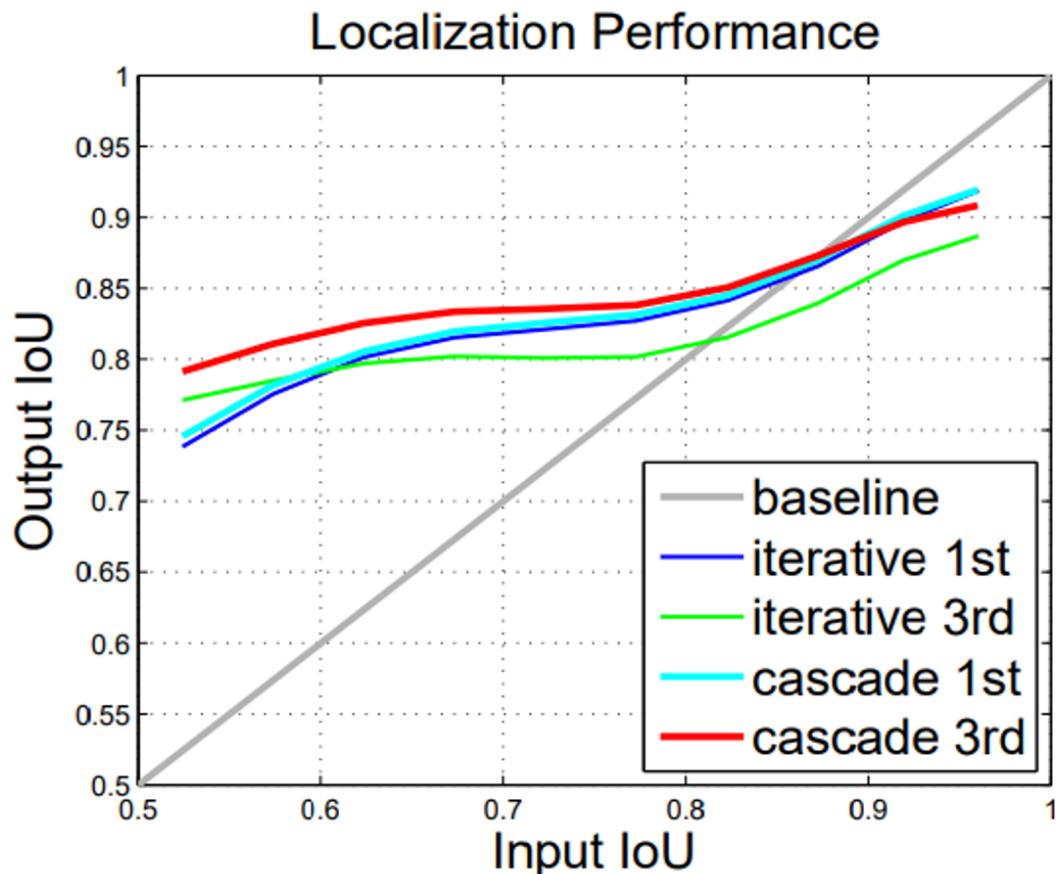
5. 결론

	AP	AP ₅₀	AP ₆₀	AP ₇₀	AP ₈₀	AP ₉₀
FPN+ baseline	34.9	57.0	51.9	43.6	29.7	7.1
<i>Iterative BBox</i>	35.4	57.2	52.1	44.2	30.4	8.1
<i>Integral Loss</i>	35.4	57.3	52.5	44.4	29.9	6.9
Cascade R-CNN	38.9	57.8	53.4	46.9	35.8	15.8

Table 1. The comparison with *iterative BBox* and *integral loss*.

iterative bbox, intergral loss도 약간의 성능 향상이 있음을 확인할 수 있다.

낮은 iou 임계값에서는 이득이 미미하지만 높은 임계값에서는 상당하다.



1번 보다 3번 반복되었을 때 iterative bbox는 성능이 감소했으나 cascade 는 더 증가하였다.