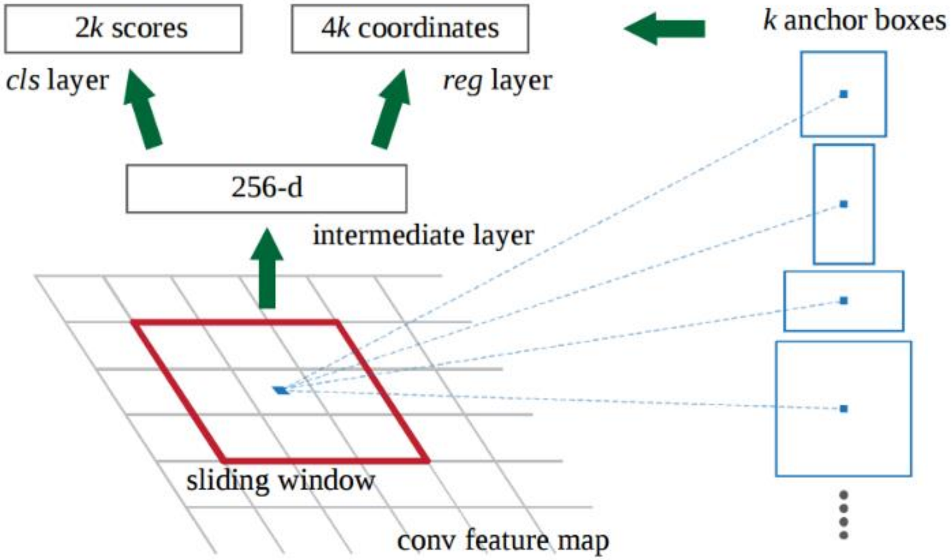
Bridging the Gap Between Anchor-based and Anchor-free Detection via Adaptive Training Sample Selection

ZUM internet Corp.

Search & AI Team
김병조

Anchor-based vs Anchor-free

- Anchor-based detector
  - R-CNN, RetinaNet ..
  - 미리 세팅해 놓은 수 많은 Anchor에서 category, coordinates 예측
- Anchor-free detector
  - FCOS, CornetNet ..
  - Anchor 없이 예측



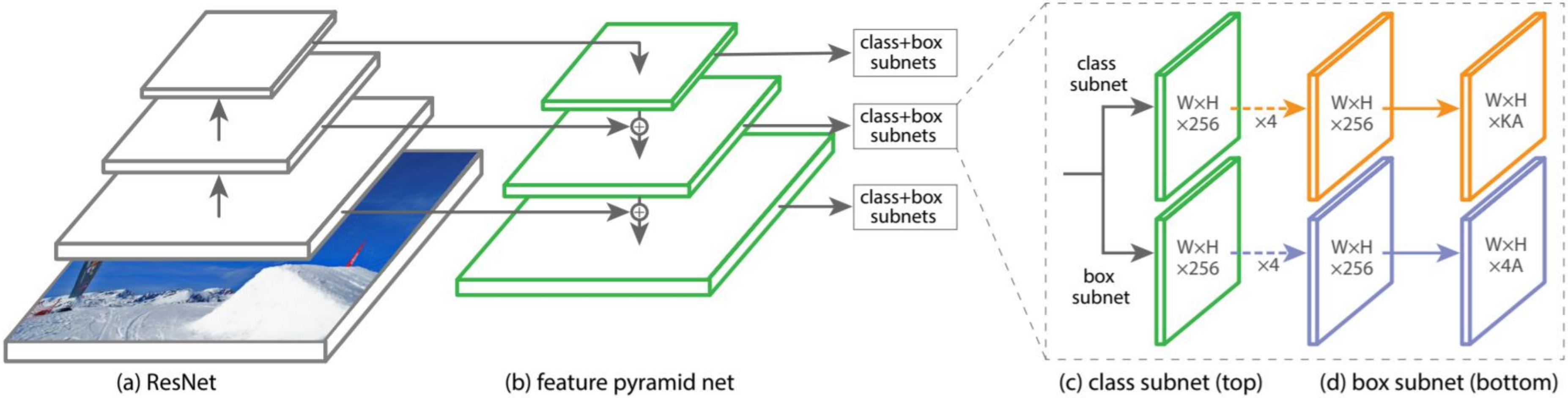
Anchor-based vs Anchor-free

- Anchor-based detection: RetinaNet
- Anchor-free detection: FCOS

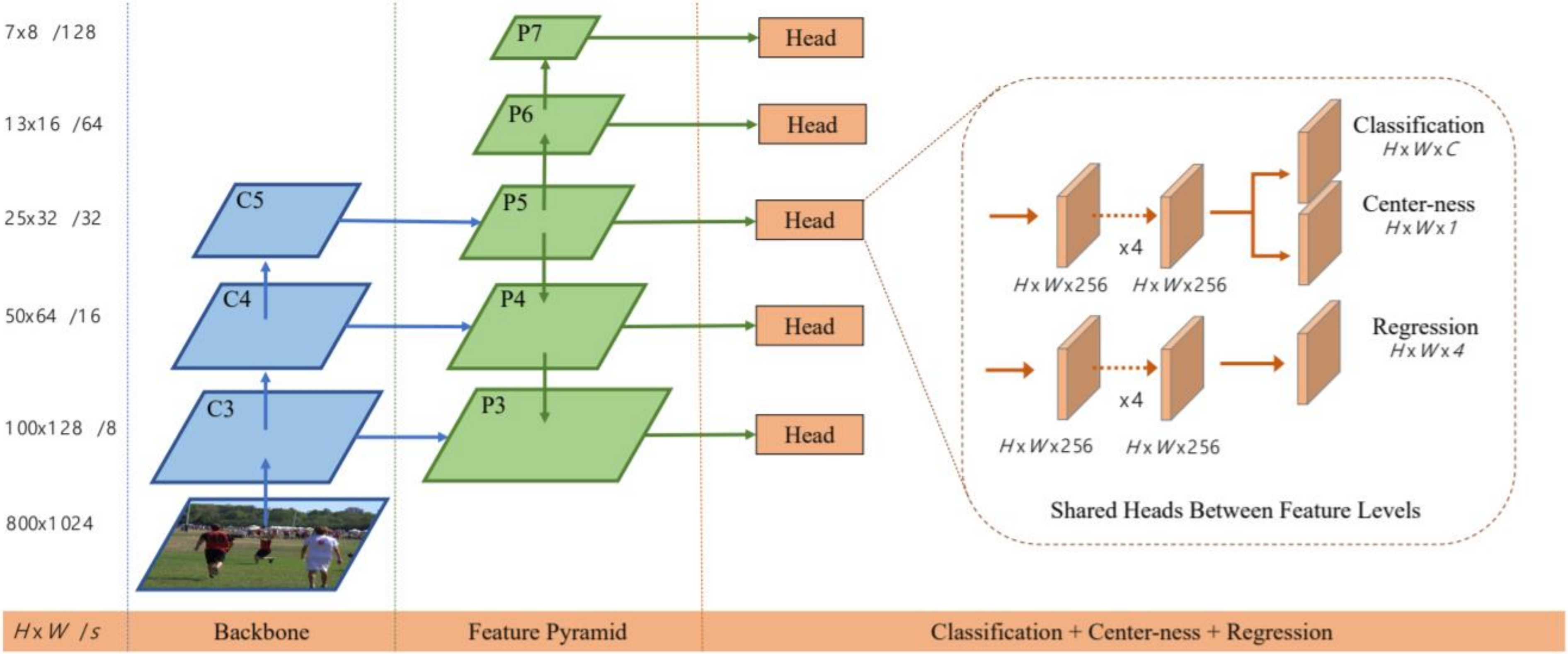
- Location 마다 Anchor(point)의 개수
- **Positive, Negative Sample 정의**
- **Regression starting status**

두 차이에 집중

Anchor-based (RetinaNet)

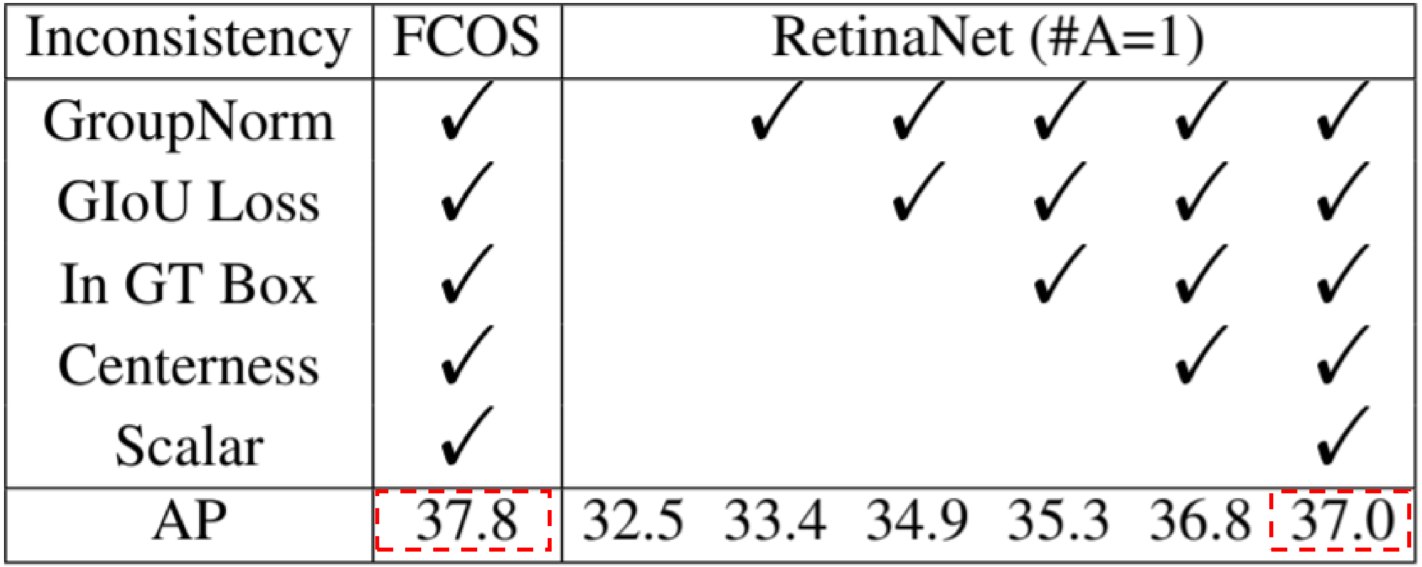


Anchor-free (FCOS)



RetinaNet vs FCOS

- Location 마다 Anchor(point)의 개수
  - Location 마다 Anchor 하나로 고정 -> RetinaNet(#A=1)
  - FCOS와 유사한 모델이 됨
- Inconsistency Removal



Inconsistency	FCOS	RetinaNet (#A=1)					
GroupNorm	✓		✓	✓	✓	✓	✓
GIoU Loss	✓			✓	✓	✓	✓
In GT Box	✓				✓	✓	✓
Centerness	✓					✓	✓
Scalar	✓						✓
AP	37.8	32.5	33.4	34.9	35.3	36.8	37.0

Positive samples only in GT box

Add centerness branch

for using identical heads (not sharing)

RetinaNet vs FCOS

- Essential Difference
  - Classification sub-task

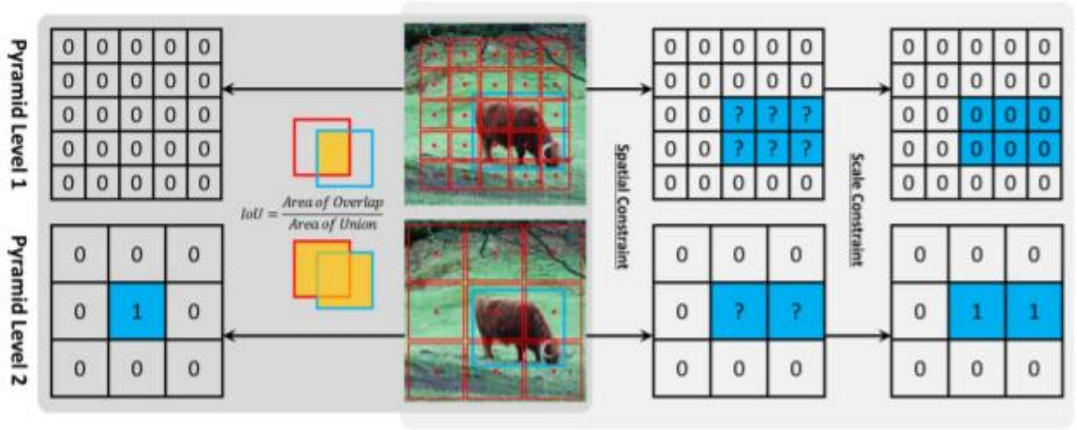
    Positive, Negative samples 정의
  - Regression sub-task

    Regression Starting status

    Box 또는 Point에서 시작되는 Regression

RetinaNet vs FCOS

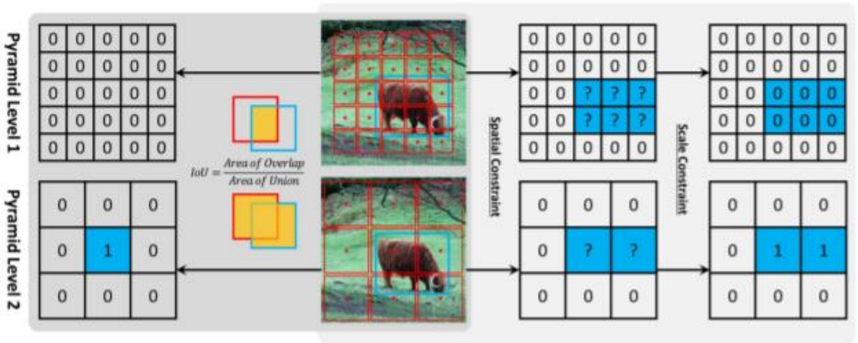
- Classification sub-task



(a) RetinaNet (b) FCOS

Figure 1: Definition of positives (1) and negatives (0). Blue box, red box and red point are ground-truth, anchor box and anchor point. (a) RetinaNet uses IoU to select positives (1) in spatial and scale dimension simultaneously. (b) FCOS first finds candidate positives (?) in spatial dimension, then selects final positives (1) in scale dimension.

RetinaNet vs FCOS

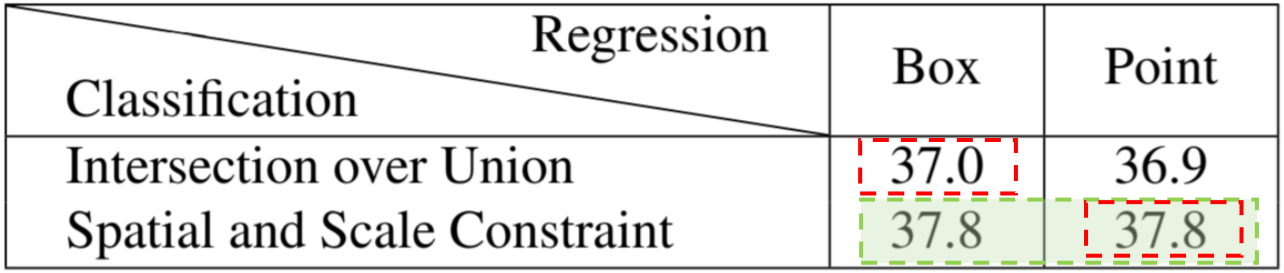


(a) RetinaNet (b) FCOS

- RetinaNet
  - IoU를 이용
  - Positive sample if IoU > threshold else Negative sample
  - Negative sample은 학습에서 제외
- FCOS
  - Spatial과 Scale 측면으로 고려
  - Anchor point가 ground box에 속해 있고, 각 pyramid level마다 정해진 scale range를 이용하여 positive sample 선택
  - 즉 Spatial로 후보 정하고, Scale로 선택

RetinaNet vs FCOS

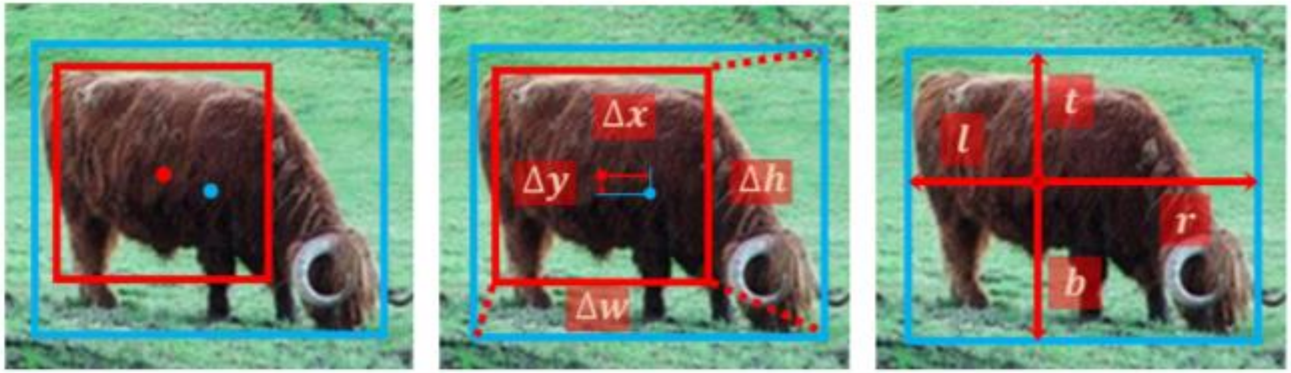
- IoU vs Spatial and Scale Constraint
  - IoU는 Spatial, Scale을 한번에 고려
  - Anchor-based, Anchor-free detector의 본질적인 차이



Classification \ Regression	Box	Point
Intersection over Union	37.0	36.9
Spatial and Scale Constraint	37.8	37.8

RetinaNet vs FCOS

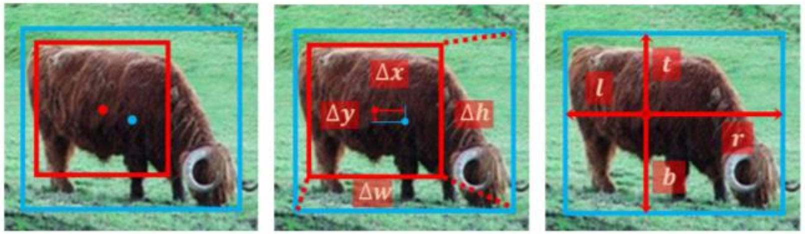
- Regression sub-task



(a) Positive sample (b) RetinaNet (c) FCOS

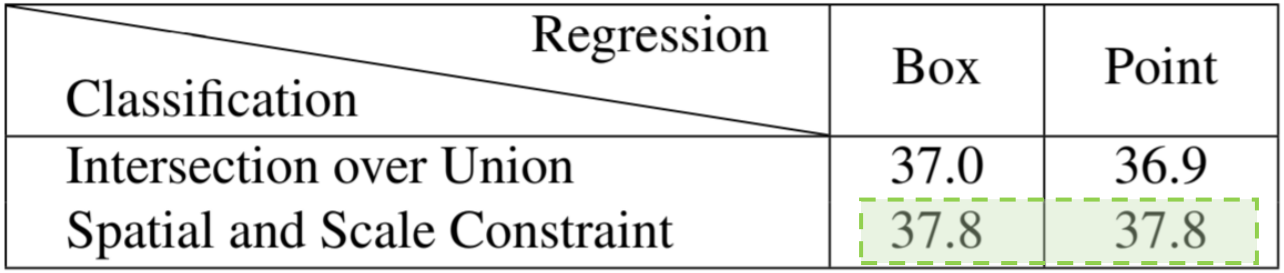
Figure 2: (a) Blue point and box are the center and bound of object, red point and box are the center and bound of anchor. (b) RetinaNet regresses from anchor box with four offsets. (c) FCOS regresses from anchor point with four distances.

RetinaNet vs FCOS



(a) Positive sample (b) RetinaNet (c) FCOS

- 결정된 Positive sample의 Bounding Box 찾기
- RetinaNet
  - Anchor box로 부터 4개의 offset에 대한 regression
- FCOS
  - Point로 부터 4개의 distance에 대한 regression



Regression / Classification	Box	Point
Intersection over Union	37.0	36.9
Spatial and Scale Constraint	37.8	37.8

Positive Sample 정의 방법을 바꿨더니 똑같은 37.8 성능 -> **Regression Starting Status는 본질적인 차이가 아니구나!**

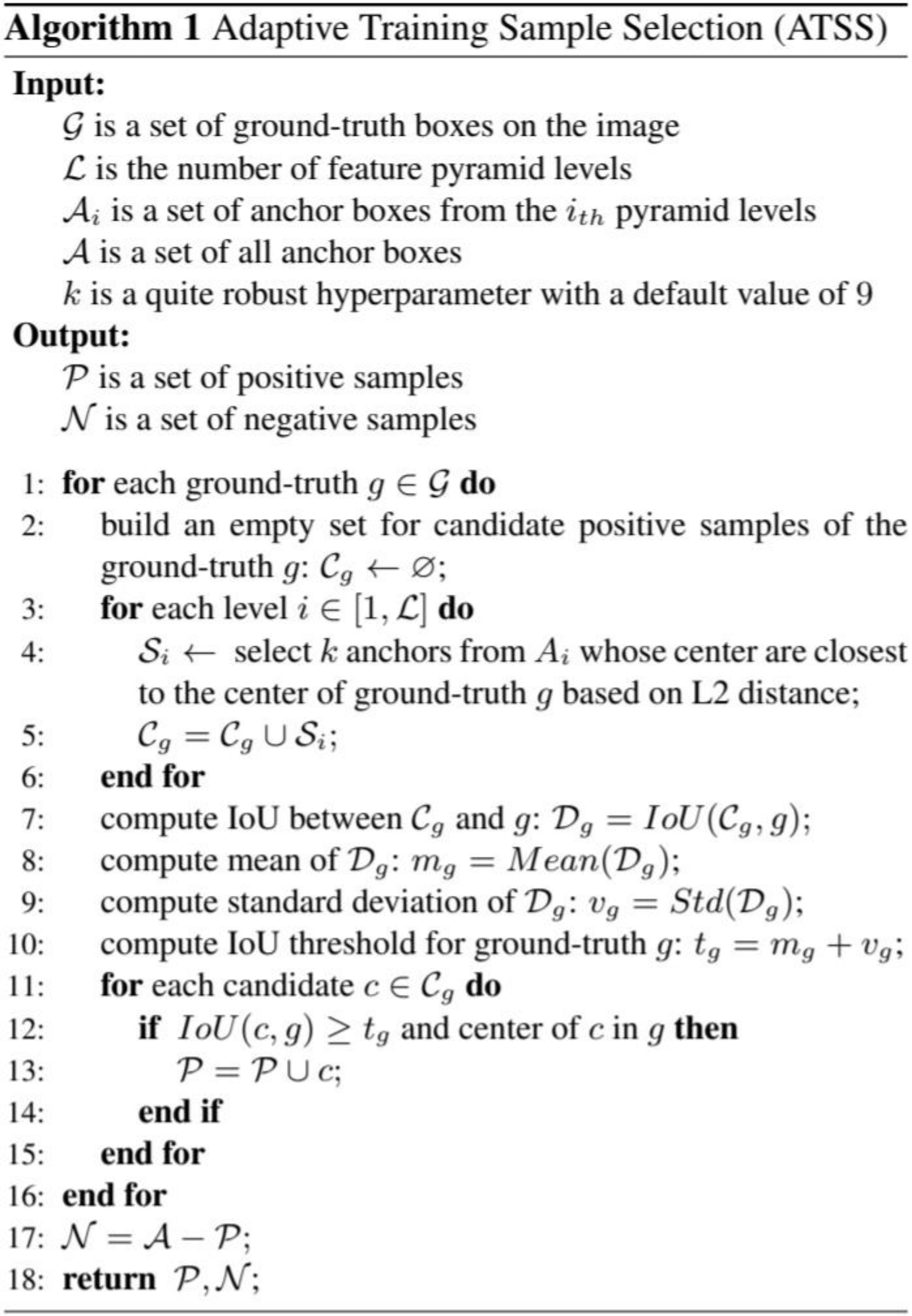
Positive Sample, Negative Sample 정의하는 방법이 제일 중요하구나!

ATSS

Anchor에 의존하지 않고,
Hyperparameter을 최소화 하면서
Positive/Negative sample을 가려내보자

모든 ground-truth g마다 loop:

- 각 pyramid level 마다 loop:
  - positive sample 후보 선택
- 최종 positive sample 선택



Algorithm 1 Adaptive Training Sample Selection (ATSS)

Input:

$\mathcal{G}$ is a set of ground-truth boxes on the image
$\mathcal{L}$ is the number of feature pyramid levels
$\mathcal{A}_i$ is a set of anchor boxes from the i_{th} pyramid levels
$\mathcal{A}$ is a set of all anchor boxes
k is a quite robust hyperparameter with a default value of 9

Output:

$\mathcal{P}$ is a set of positive samples
$\mathcal{N}$ is a set of negative samples

```
for each ground-truth $g \in \mathcal{G}$ do
    build an empty set for candidate positive samples of the ground-truth $g$: $\mathcal{C}_g \leftarrow \varnothing$;
    for each level $i \in [1, \mathcal{L}]$ do
        $\mathcal{S}_i \leftarrow$ select $k$ anchors from $\mathcal{A}_i$ whose center are closest to the center of ground-truth $g$ based on L2 distance;
        $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$;
    end for
    compute IoU between $\mathcal{C}_g$ and $g$: $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$;
    compute mean of $\mathcal{D}_g$: $m_g = Mean(\mathcal{D}_g)$;
    compute standard deviation of $\mathcal{D}_g$: $v_g = Std(\mathcal{D}_g)$;
    compute IoU threshold for ground-truth $g$: $t_g = m_g + v_g$;
    for each candidate $c \in \mathcal{C}_g$ do
        if $IoU(c, g) \geq t_g$ and center of $c$ in $g$ then
            $\mathcal{P} = \mathcal{P} \cup c$;
        end if
    end for
end for
$\mathcal{N} = \mathcal{A} - \mathcal{P}$;
return $\mathcal{P}, \mathcal{N}$;
```

ATSS

- ground-truth box g의 positive sample 후보를 정한다.
  - 각 pyramid level 마다 g와 가장 가까운 k개의 anchor box를 구한다.
  - g의 center 그리고 anchor box의 center와의 L2 distance 이용
  - 결국 하나의 ground-truth box g는 k×L개의 positive sample 후보(Cg)가 생긴다.

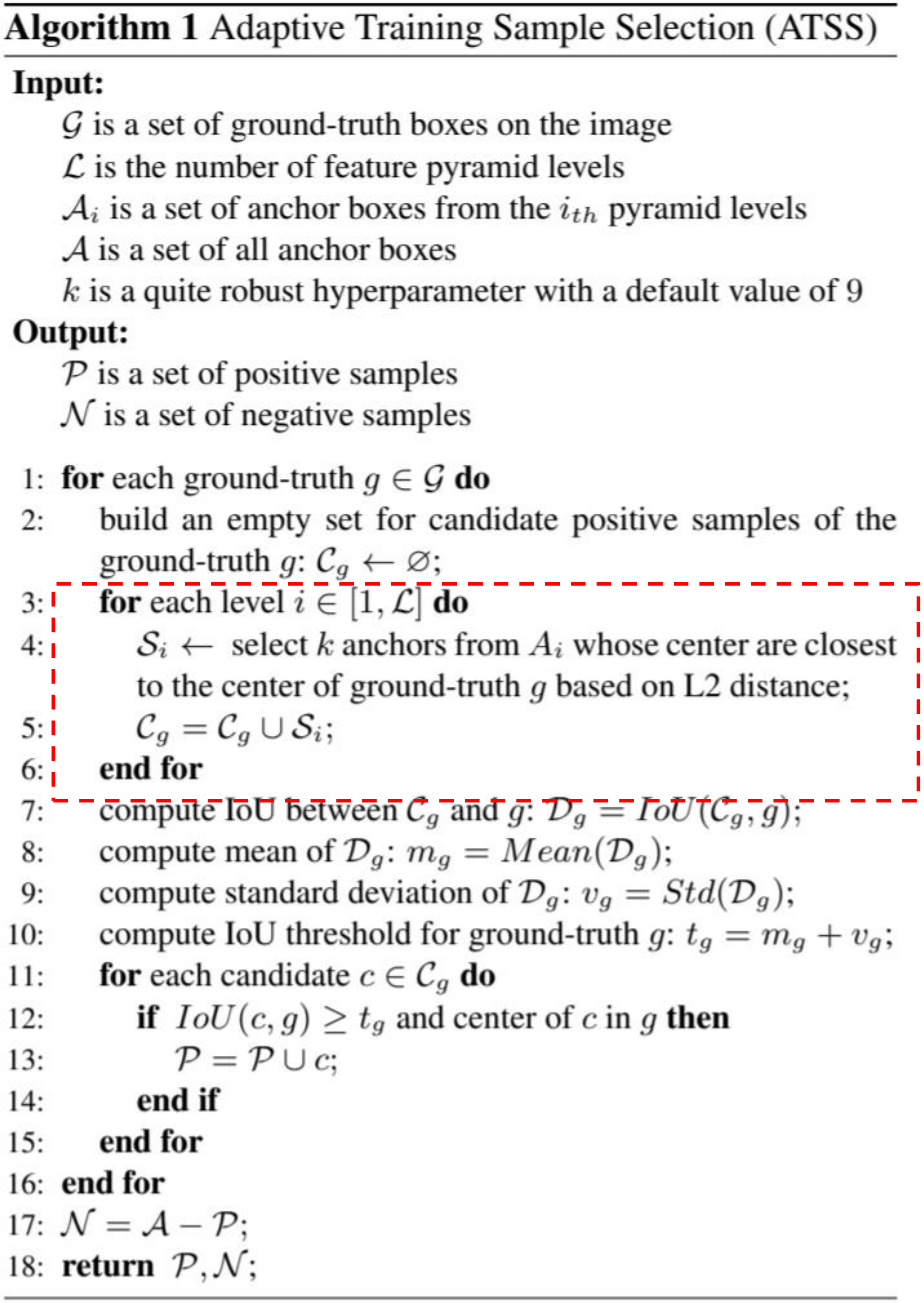
anchor box와 object 사이의 center distance based 후보 선택

중심의 거리가 가까울 수록 좋은 검출

-> center distance 기반

hyperparameter-free

k 가 유일한 hyperparameter



Algorithm 1 Adaptive Training Sample Selection (ATSS)

Input:

$\mathcal{G}$ is a set of ground-truth boxes on the image
$\mathcal{L}$ is the number of feature pyramid levels
$\mathcal{A}_i$ is a set of anchor boxes from the i_{th} pyramid levels
$\mathcal{A}$ is a set of all anchor boxes
k is a quite robust hyperparameter with a default value of 9

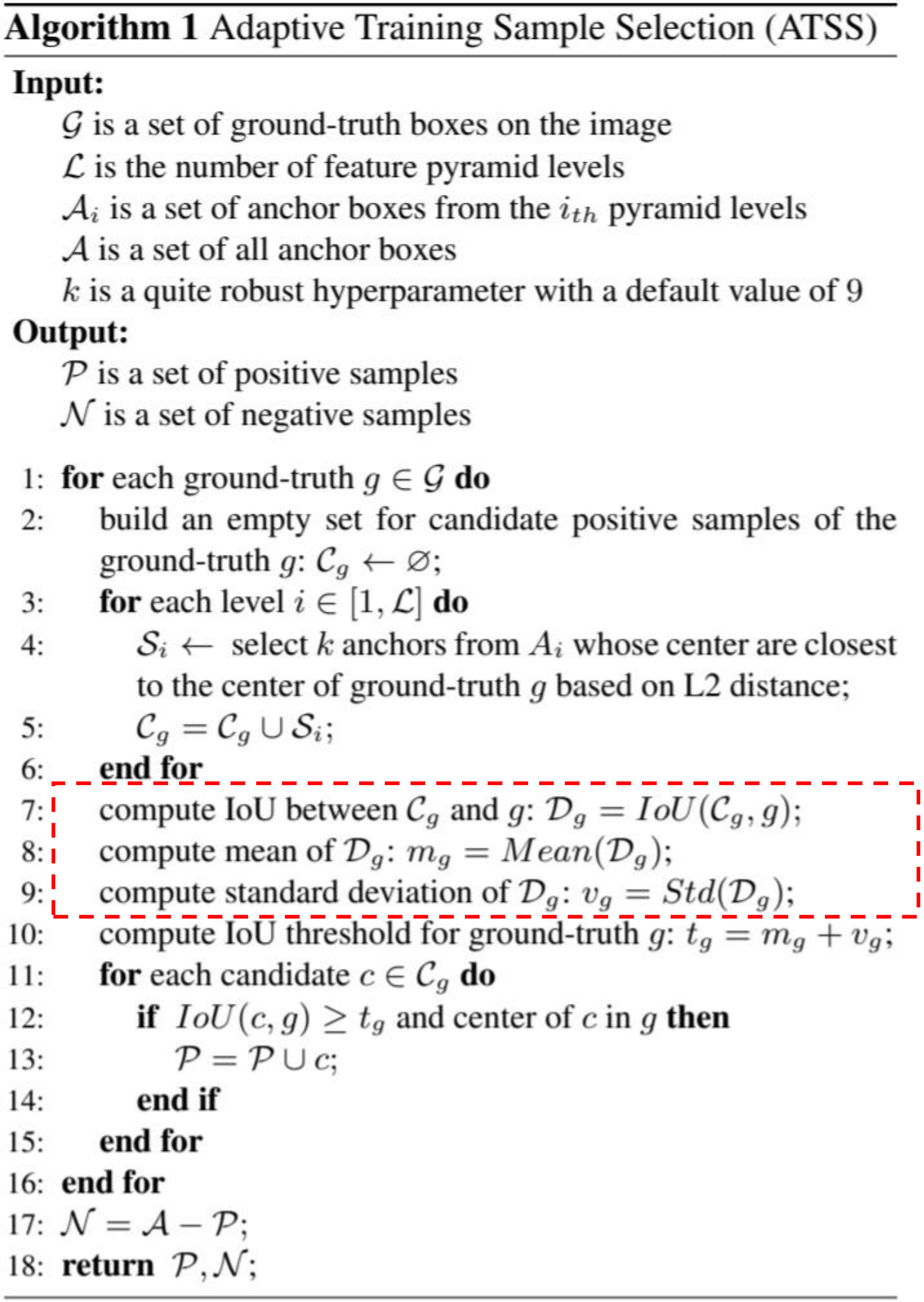
Output:

$\mathcal{P}$ is a set of positive samples
$\mathcal{N}$ is a set of negative samples

```
for each ground-truth $g \in \mathcal{G}$ do
    build an empty set for candidate positive samples of the ground-truth $g$: $\mathcal{C}_g \leftarrow \varnothing$;
    for each level $i \in [1, \mathcal{L}]$ do
        $\mathcal{S}_i \leftarrow$ select $k$ anchors from $\mathcal{A}_i$ whose center are closest to the center of ground-truth $g$ based on L2 distance;
        $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$;
    end for
    compute IoU between $\mathcal{C}_g$ and $g$: $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$;
    compute mean of $\mathcal{D}_g$: $m_g = Mean(\mathcal{D}_g)$;
    compute standard deviation of $\mathcal{D}_g$: $v_g = Std(\mathcal{D}_g)$;
    compute IoU threshold for ground-truth $g$: $t_g = m_g + v_g$;
    for each candidate $c \in \mathcal{C}_g$ do
        if $IoU(c, g) \geq t_g$ and center of $c$ in $g$ then
            $\mathcal{P} = \mathcal{P} \cup c$;
        end if
    end for
end for
$\mathcal{N} = \mathcal{A} - \mathcal{P}$;
return $\mathcal{P}, \mathcal{N}$;
```

ATSS

- ground truth g와 후보들(Cg)간의 IoU를 계산한다. Dg 그 후 mean(mg)과 standard deviation(vg)을 구한다.



Algorithm 1 Adaptive Training Sample Selection (ATSS)

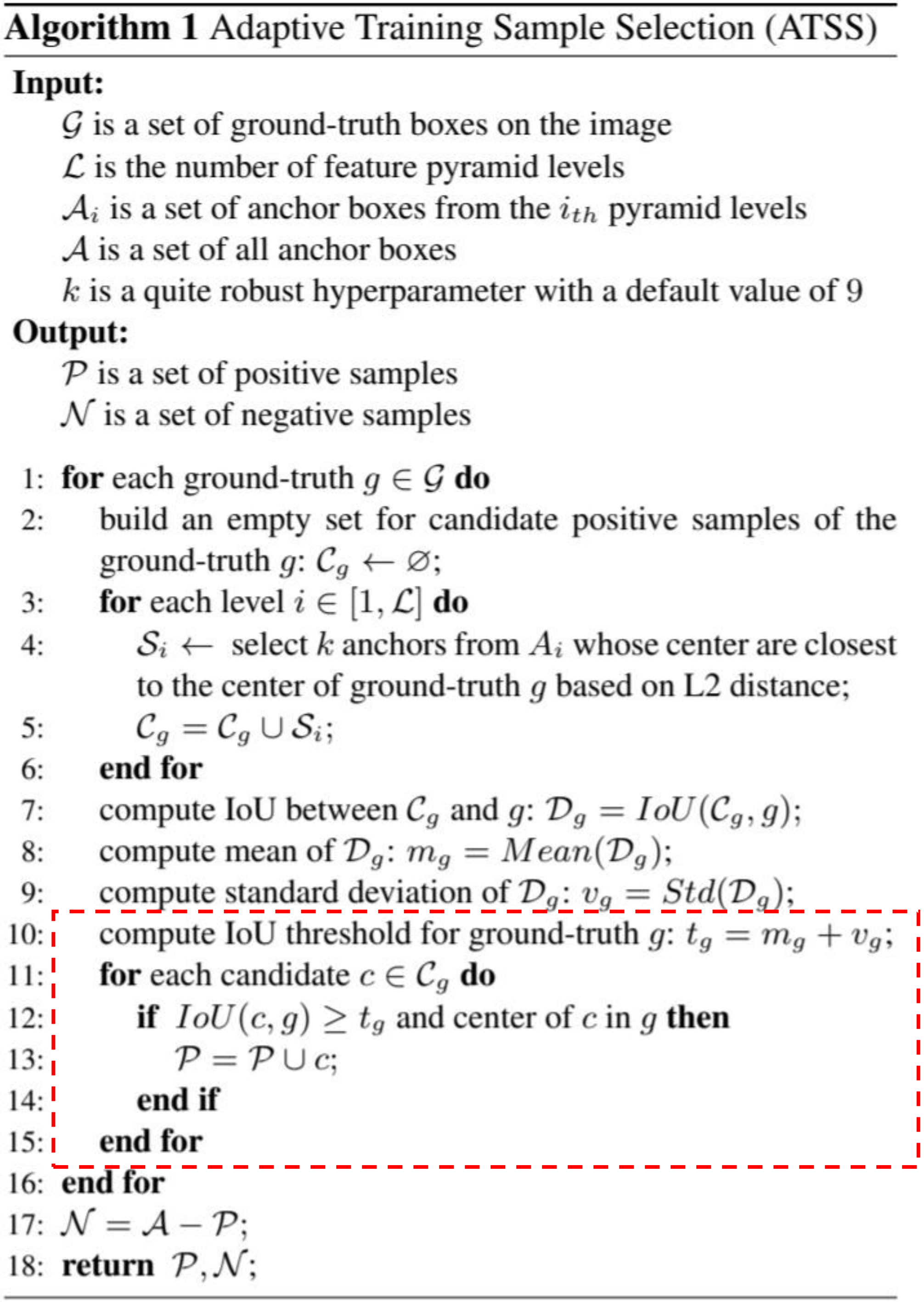
```
Input:
    $\mathcal{G}$ is a set of ground-truth boxes on the image
    $\mathcal{L}$ is the number of feature pyramid levels
    $\mathcal{A}_i$ is a set of anchor boxes from the $i_{th}$ pyramid levels
    $\mathcal{A}$ is a set of all anchor boxes
    $k$ is a quite robust hyperparameter with a default value of 9
Output:
    $\mathcal{P}$ is a set of positive samples
    $\mathcal{N}$ is a set of negative samples

for each ground-truth $g \in \mathcal{G}$ do
    build an empty set for candidate positive samples of the ground-truth $g$: $\mathcal{C}_g \leftarrow \varnothing$;
    for each level $i \in [1, \mathcal{L}]$ do
        $\mathcal{S}_i \leftarrow$ select $k$ anchors from $\mathcal{A}_i$ whose center are closest to the center of ground-truth $g$ based on L2 distance;
        $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$;
    end for
    compute IoU between $\mathcal{C}_g$ and $g$: $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$;
    compute mean of $\mathcal{D}_g$: $m_g = Mean(\mathcal{D}_g)$;
    compute standard deviation of $\mathcal{D}_g$: $v_g = Std(\mathcal{D}_g)$;
    compute IoU threshold for ground-truth $g$: $t_g = m_g + v_g$;
    for each candidate $c \in \mathcal{C}_g$ do
        if $IoU(c, g) \geq t_g$ and center of $c$ in $g$ then
            $\mathcal{P} = \mathcal{P} \cup c$;
        end if
    end for
end for
$\mathcal{N} = \mathcal{A} - \mathcal{P}$;
return $\mathcal{P}, \mathcal{N}$;
```

ATSS

- ground truth g와의 IoU가 특정 threshold 값 보다 큰 후보를 최종 positive sample(P)로 선택한다.
    - positive sample의 center가 ground truth g안에 있는 경우에만 선택
    - threshold tg=mg+vg
    - 하나의 anchor box가 여러 ground-truth box의 positive sample이 된다면, 가장 높은 IoU를 가진 쪽으로 선택된다.

중심이 object에 있어야 선택

중심이 object 밖에 있는 anchor
-> 좋지 않은 후보



Algorithm 1 Adaptive Training Sample Selection (ATSS)

```
Input:
    $\mathcal{G}$ is a set of ground-truth boxes on the image
    $\mathcal{L}$ is the number of feature pyramid levels
    $\mathcal{A}_i$ is a set of anchor boxes from the $i_{th}$ pyramid levels
    $\mathcal{A}$ is a set of all anchor boxes
    $k$ is a quite robust hyperparameter with a default value of 9
Output:
    $\mathcal{P}$ is a set of positive samples
    $\mathcal{N}$ is a set of negative samples

for each ground-truth $g \in \mathcal{G}$ do
    build an empty set for candidate positive samples of the ground-truth $g$: $\mathcal{C}_g \leftarrow \varnothing$;
    for each level $i \in [1, \mathcal{L}]$ do
        $\mathcal{S}_i \leftarrow$ select $k$ anchors from $\mathcal{A}_i$ whose center are closest to the center of ground-truth $g$ based on L2 distance;
        $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$;
    end for
    compute IoU between $\mathcal{C}_g$ and $g$: $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$;
    compute mean of $\mathcal{D}_g$: $m_g = Mean(\mathcal{D}_g)$;
    compute standard deviation of $\mathcal{D}_g$: $v_g = Std(\mathcal{D}_g)$;
    compute IoU threshold for ground-truth $g$: $t_g = m_g + v_g$;
    for each candidate $c \in \mathcal{C}_g$ do
        if $IoU(c, g) \geq t_g$ and center of $c$ in $g$ then
            $\mathcal{P} = \mathcal{P} \cup c$;
        end if
    end for
end for
$\mathcal{N} = \mathcal{A} - \mathcal{P}$;
return $\mathcal{P}, \mathcal{N}$;
```

ATSS

- ground truth g와의 IoU가 특정 threshold 값 보다 큰 후보를 최종 positive sample(P)로 선택한다.
    - positive sample의 center가 ground truth g안에 있는 경우에만 선택
    - threshold tg=mg+vg

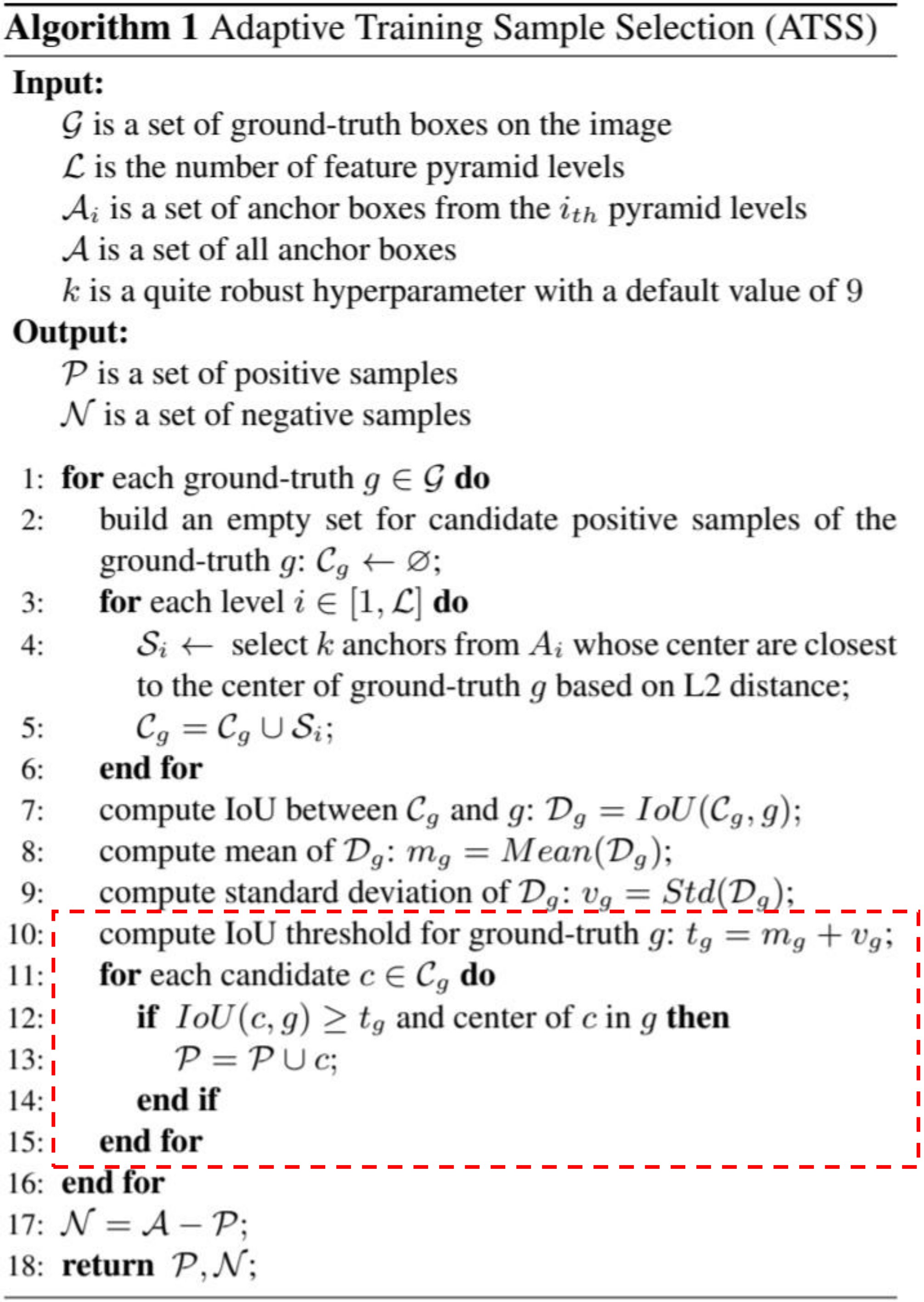
IoU threshold에 tg=mg+vg를 사용

mean ↑: high quality
-> threshold ↑

mean ↓: low quality
-> threshold ↓

standard deviation ↑: 대부분의 pyramid level에서도 검출 가능
-> threshold ↑

standard deviation ↓ : 몇몇 pyramid level에서만 검출 가능
-> threshold ↓

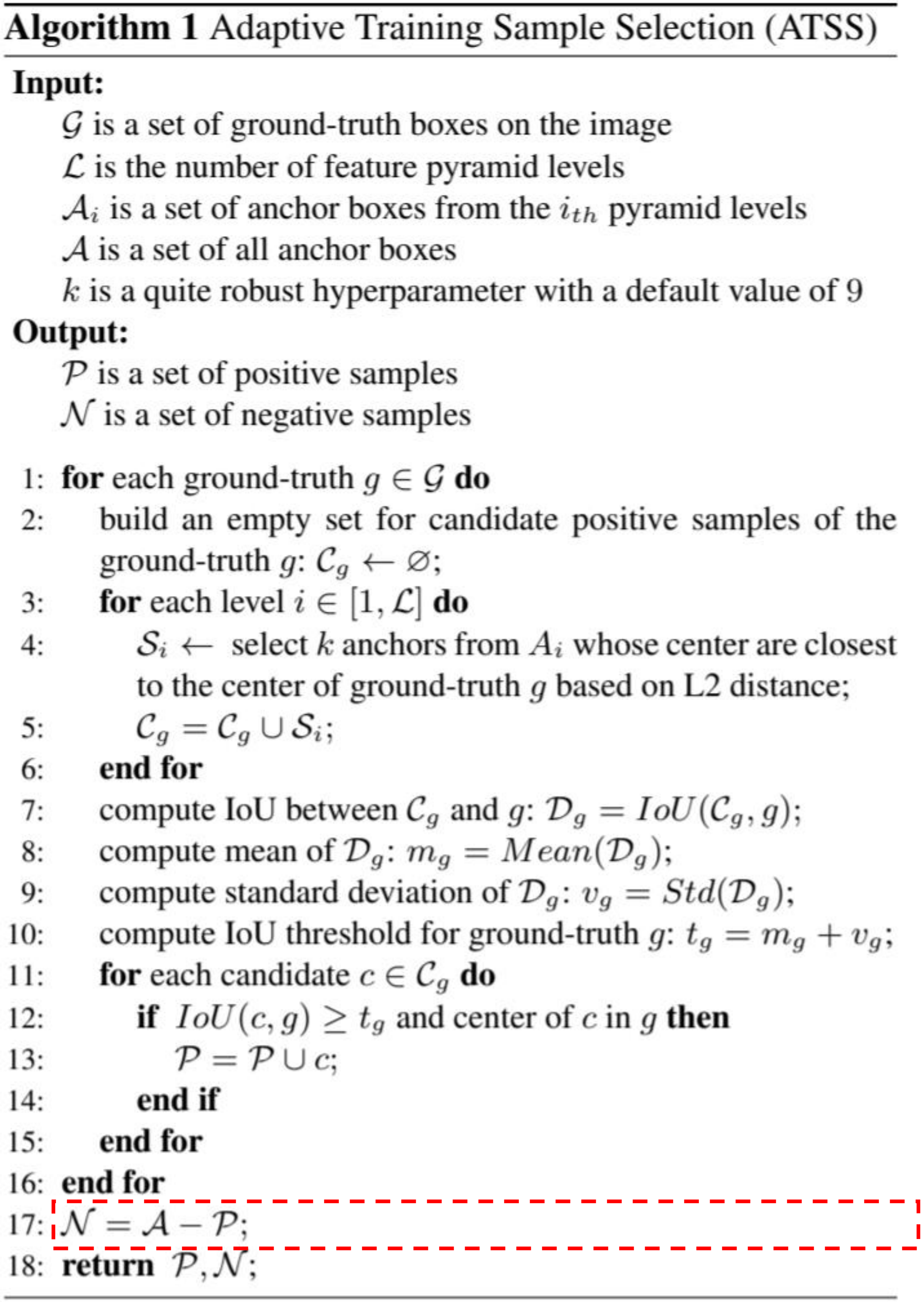


Algorithm 1 Adaptive Training Sample Selection (ATSS)

```
Input:
    $\mathcal{G}$ is a set of ground-truth boxes on the image
    $\mathcal{L}$ is the number of feature pyramid levels
    $\mathcal{A}_i$ is a set of anchor boxes from the $i_{th}$ pyramid levels
    $\mathcal{A}$ is a set of all anchor boxes
    $k$ is a quite robust hyperparameter with a default value of 9
Output:
    $\mathcal{P}$ is a set of positive samples
    $\mathcal{N}$ is a set of negative samples

for each ground-truth $g \in \mathcal{G}$ do
    build an empty set for candidate positive samples of the ground-truth $g$: $\mathcal{C}_g \leftarrow \varnothing$;
    for each level $i \in [1, \mathcal{L}]$ do
        $\mathcal{S}_i \leftarrow$ select $k$ anchors from $\mathcal{A}_i$ whose center are closest to the center of ground-truth $g$ based on L2 distance;
        $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$;
    end for
    compute IoU between $\mathcal{C}_g$ and $g$: $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$;
    compute mean of $\mathcal{D}_g$: $m_g = Mean(\mathcal{D}_g)$;
    compute standard deviation of $\mathcal{D}_g$: $v_g = Std(\mathcal{D}_g)$;
    compute IoU threshold for ground-truth $g$: $t_g = m_g + v_g$;
    for each candidate $c \in \mathcal{C}_g$ do
        if $IoU(c, g) \geq t_g$ and center of $c$ in $g$ then
            $\mathcal{P} = \mathcal{P} \cup c$;
        end if
    end for
end for
$\mathcal{N} = \mathcal{A} - \mathcal{P}$;
return $\mathcal{P}, \mathcal{N}$;
```

ATSS

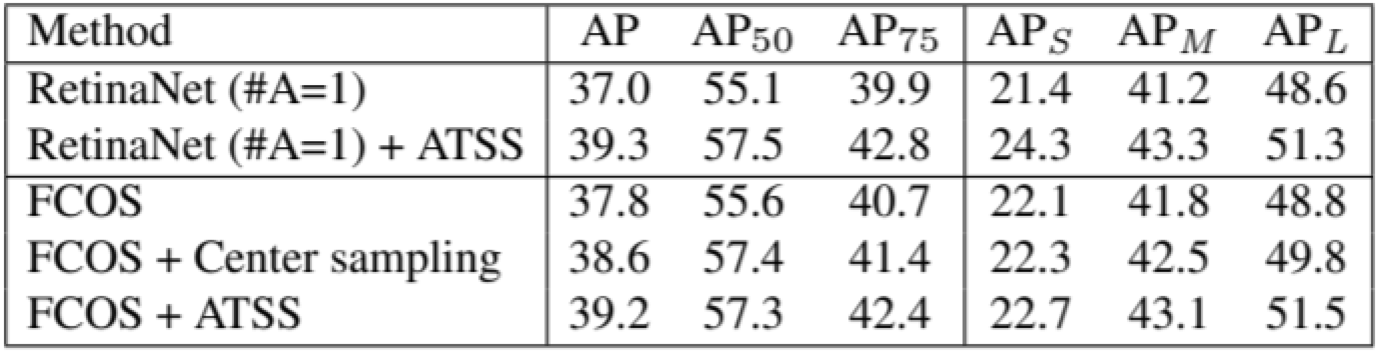
- 전체 anchor box에서 positive sample로 선택 받지 못한 anchor들은 negative sample이 된다.



Algorithm 1 Adaptive Training Sample Selection (ATSS)

```
Input:
    $\mathcal{G}$ is a set of ground-truth boxes on the image
    $\mathcal{L}$ is the number of feature pyramid levels
    $\mathcal{A}_i$ is a set of anchor boxes from the $i_{th}$ pyramid levels
    $\mathcal{A}$ is a set of all anchor boxes
    $k$ is a quite robust hyperparameter with a default value of 9
Output:
    $\mathcal{P}$ is a set of positive samples
    $\mathcal{N}$ is a set of negative samples

for each ground-truth $g \in \mathcal{G}$ do
    build an empty set for candidate positive samples of the ground-truth $g$: $\mathcal{C}_g \leftarrow \varnothing$;
    for each level $i \in [1, \mathcal{L}]$ do
        $\mathcal{S}_i \leftarrow$ select $k$ anchors from $\mathcal{A}_i$ whose center are closest to the center of ground-truth $g$ based on L2 distance;
        $\mathcal{C}_g = \mathcal{C}_g \cup \mathcal{S}_i$;
    end for
    compute IoU between $\mathcal{C}_g$ and $g$: $\mathcal{D}_g = IoU(\mathcal{C}_g, g)$;
    compute mean of $\mathcal{D}_g$: $m_g = Mean(\mathcal{D}_g)$;
    compute standard deviation of $\mathcal{D}_g$: $v_g = Std(\mathcal{D}_g)$;
    compute IoU threshold for ground-truth $g$: $t_g = m_g + v_g$;
    for each candidate $c \in \mathcal{C}_g$ do
        if $IoU(c, g) \geq t_g$ and center of $c$ in $g$ then
            $\mathcal{P} = \mathcal{P} \cup c$;
        end if
    end for
end for
$\mathcal{N} = \mathcal{A} - \mathcal{P}$;
return $\mathcal{P}, \mathcal{N}$;
```

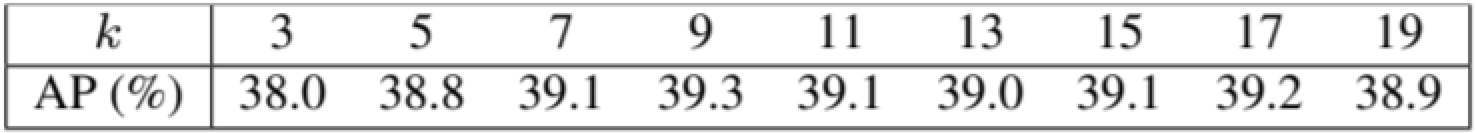
Experiments



Method	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
RetinaNet (#A=1)	37.0	55.1	39.9	21.4	41.2	48.6
RetinaNet (#A=1) + ATSS	39.3	57.5	42.8	24.3	43.3	51.3
FCOS	37.8	55.6	40.7	22.1	41.8	48.8
FCOS + Center sampling	38.6	57.4	41.4	22.3	42.5	49.8
FCOS + ATSS	39.2	57.3	42.4	22.7	43.1	51.5

- RetinaNet의 기법과 비교 했을 때 좋은 성능, overhead도 적음
- Center sampling은 ATSS의 lite 버전 (center distance로 후보 선택 + Scale Constraint)
- Scale Constraint에는 각 level 마다 scale 관련 hyperparameter가 존재해서 그냥 ATSS 사용하는게 더 좋음
- Hyperparameter 줄이니 여러 metric에서 좋은 성능을 낼 수 있었음

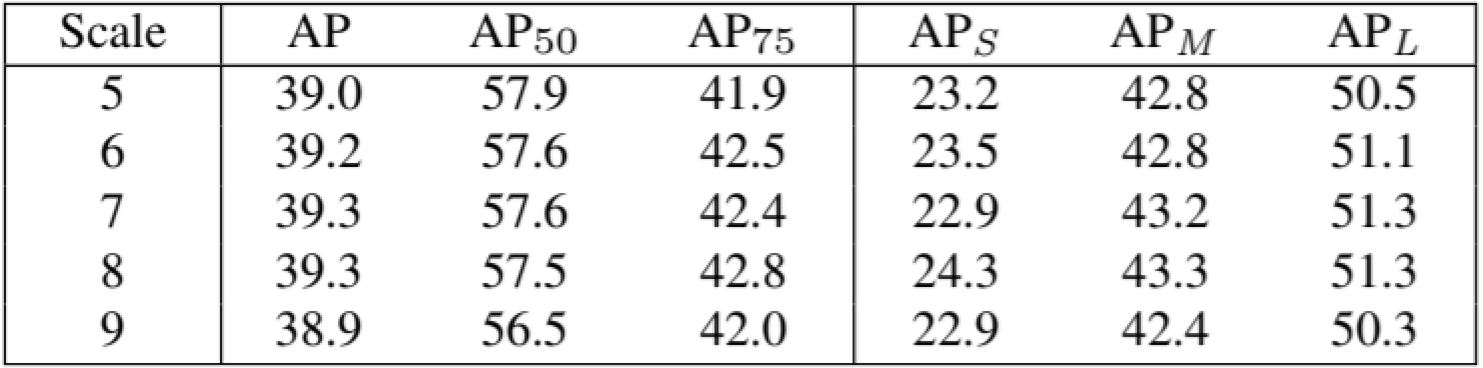
Experiments



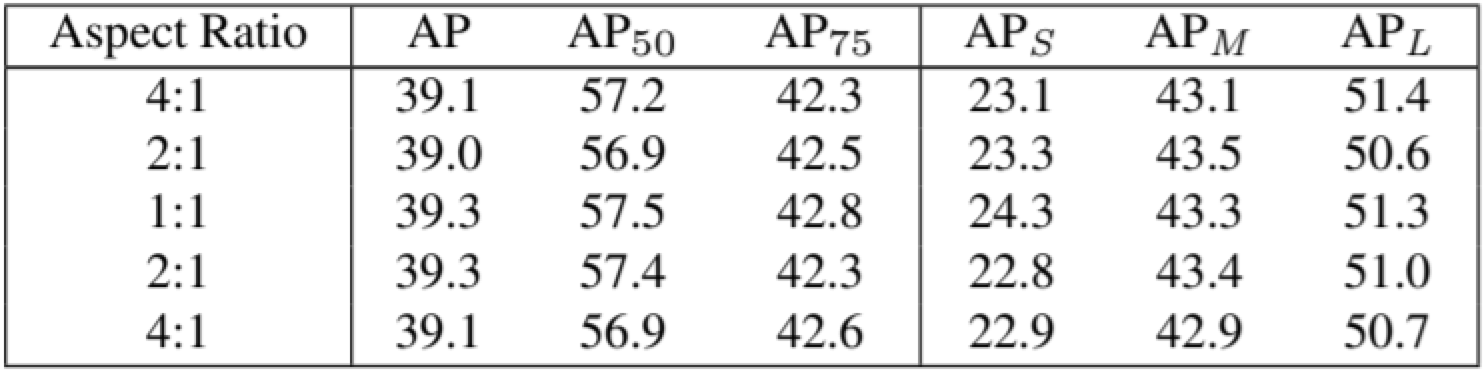
k	3	5	7	9	11	13	15	17	19
AP (%)	38.0	38.8	39.1	39.3	39.1	39.0	39.1	39.2	38.9

- 유일한 hyperparameter k 실험
- k가 매우 큰 경우: low-quality 후보들이 선택되어 성능 감소
- k가 매우 작은 경우: 매우 적은 수의 후보들만 선택되어 (statistical 불안정 (mean, standard deviation 사용)) 성능 감소
- k가 적당한 경우에는 성능 비슷함: quite robust 하며, hyperparameter free 하다 라고 주장

Experiments



Scale	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
5	39.0	57.9	41.9	23.2	42.8	50.5
6	39.2	57.6	42.5	23.5	42.8	51.1
7	39.3	57.6	42.4	22.9	43.2	51.3
8	39.3	57.5	42.8	24.3	43.3	51.3
9	38.9	56.5	42.0	22.9	42.4	50.3



Aspect Ratio	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
4:1	39.1	57.2	42.3	23.1	43.1	51.4
2:1	39.0	56.9	42.5	23.3	43.5	50.6
1:1	39.3	57.5	42.8	24.3	43.3	51.3
2:1	39.3	57.4	42.3	22.8	43.4	51.0
4:1	39.1	56.9	42.6	22.9	42.9	50.7

- ATSS 방식도 anchor을 사용하기 때문에 anchor의 크기에 관한 실험
- 결론: Scale, Ratio 변경시켜도 비슷한 성능 냄
- 이 또한 ATSS는 Robust 하다고 주장

Discussion

- ATSS는 단순히 positive/negative sample 정의하는 방법론이기 때문에, 타 모델들과 상호보완적이다.

- 많은 Anchor 개수 모델 > Anchor 1개 모델
- 많은 Anchor 개수 모델 + ATSS == Anchor 1개 모델 + ATSS

- Anchor 여러 개 쓰려면 어떻게 해야 할까?
- Anchor 여러 개 쓴 정확한 이유가 무엇일까?

VIPriors

- Image Classification
  - Subset of ImageNet 2012
  - 50 images per class
  - Base: ResNet-50 -> 31.16% Accuracy (Top-1)
- Object Detection
  - Subset of MS COCO 2017
  - Base: Faster R-CNN -> 0.049 AP@0.50:0.95

VIPriors

- Challenges open: March 11, 2020
- Challenges close: July 3, 2020
- Winners announced: July 17, 2020

- 계획..?
- 약 14주

QnA