RetinaNet

에이아이스쿨(AISchool) 대표
양진호 (솔라리스)
http://aischool.ai
http://solarisailab.com

TensorFlow Object Detection API에서 제공하는 다양한 Object Detection을 위한 최신 모델들

- TensorFlow Object Detection API는 다음과 같은 최신 Object Detection 모델의 다양한 backbone을 이용한 구현을 제공합니다.

① Faster R-CNN

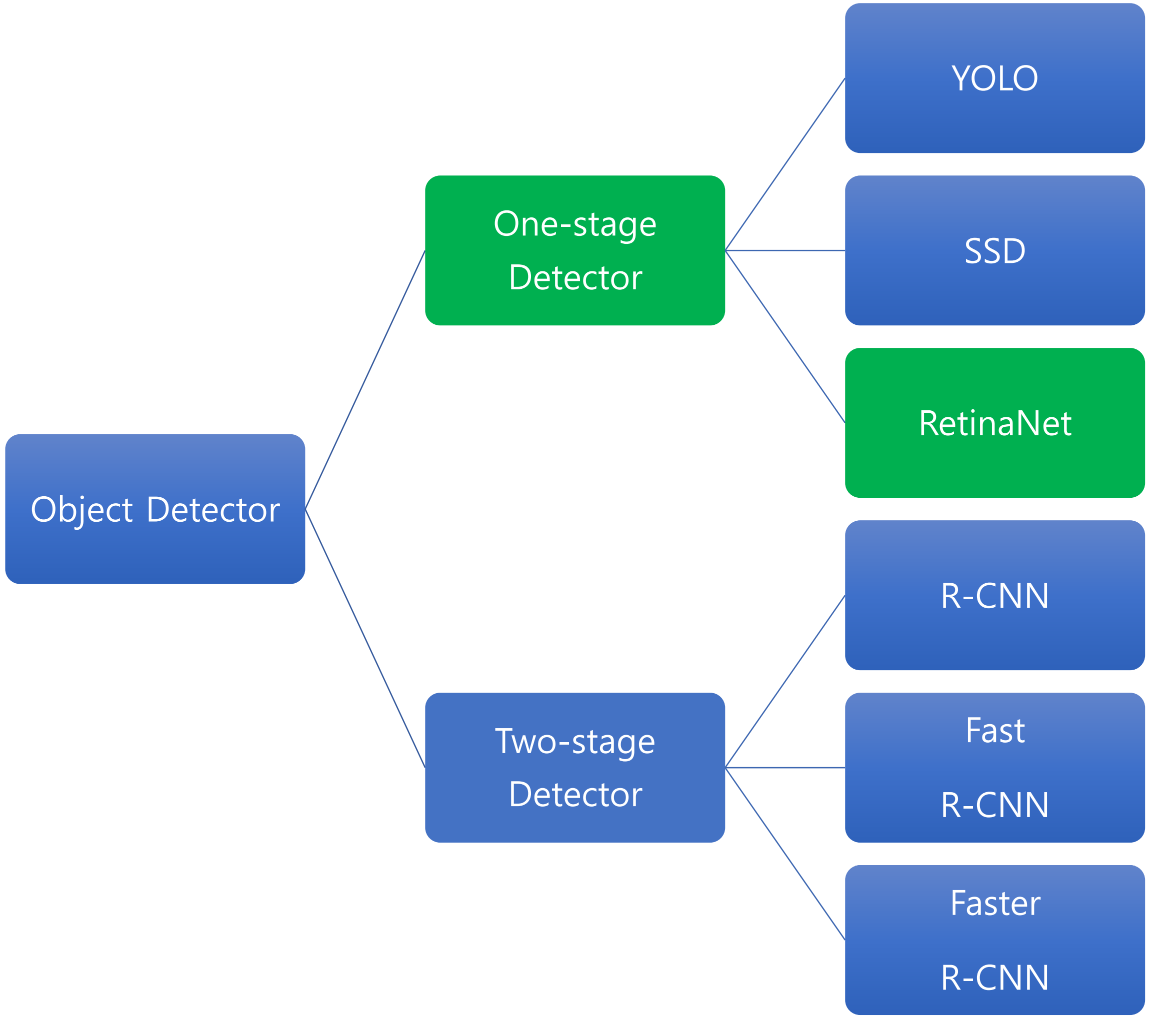
② SSD(Single Shot Multi-box Detector)

③ RetinaNet

④ CenterNet

⑤ EfficientDet

One-stage Detector vs Two-Stage Detector



AI School.

RetinaNet

- Lin, Tsung-Yi, et al. "Focal loss for dense object detection." Proceedings of the IEEE international conference on computer vision. 2017.
- https://arxiv.org/pdf/1708.02002.pdf

Focal Loss for Dense Object Detection

Tsung-Yi Lin   Priya Goyal   Ross Girshick   Kaiming He   Piotr Dollár
Facebook AI Research (FAIR)

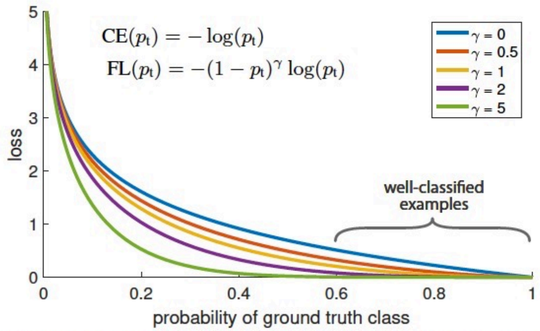


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1-p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

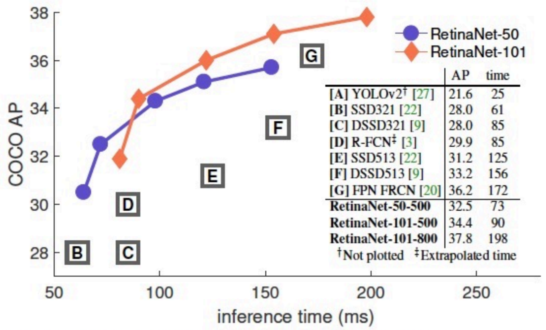


Figure 2. Speed (ms) versus accuracy (AP) on COCO `test-dev`. Enabled by the focal loss, our simple one-stage *RetinaNet* detector outperforms all previous one-stage and two-stage detectors, including the best reported Faster R-CNN [28] system from [20]. We show variants of RetinaNet with ResNet-50-FPN (blue circles) and ResNet-101-FPN (orange diamonds) at five scales (400-800 pixels). Ignoring the low-accuracy regime (AP<25), RetinaNet forms an upper envelope of all current detectors, and an improved variant (not shown) achieves 40.8 AP. Details are given in §5.

Abstract

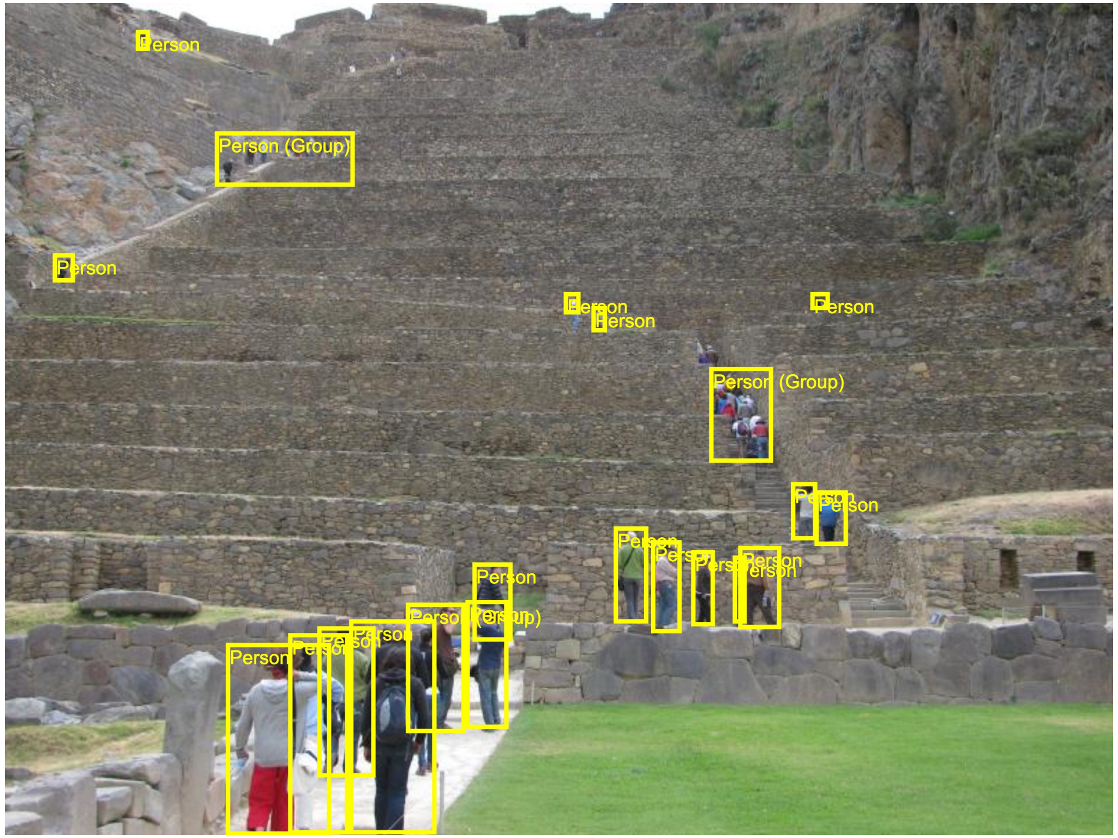
AI School.

기존 One-Stage Detector 계열의 문제점

- 기존 One-Stage Detector의 문제점

① 작은 Object를 잘 찾지 못함

② Foreground와 Background 간의 Class Imbalance 문제가 존재



Reference : https://storage.googleapis.com/openimages/web/visualizer/index.html?set=train&type=detection&c=%2Fm%2F01g317&id=3389eeaee4430d91

Focal Loss의 핵심 아이디어

- Focal Loss의 핵심 아이디어 : 쉬운 Example(Background)의 가중치를 낮추고 어려운 Example의 가중치를 높여서 Loss를 계산하자!

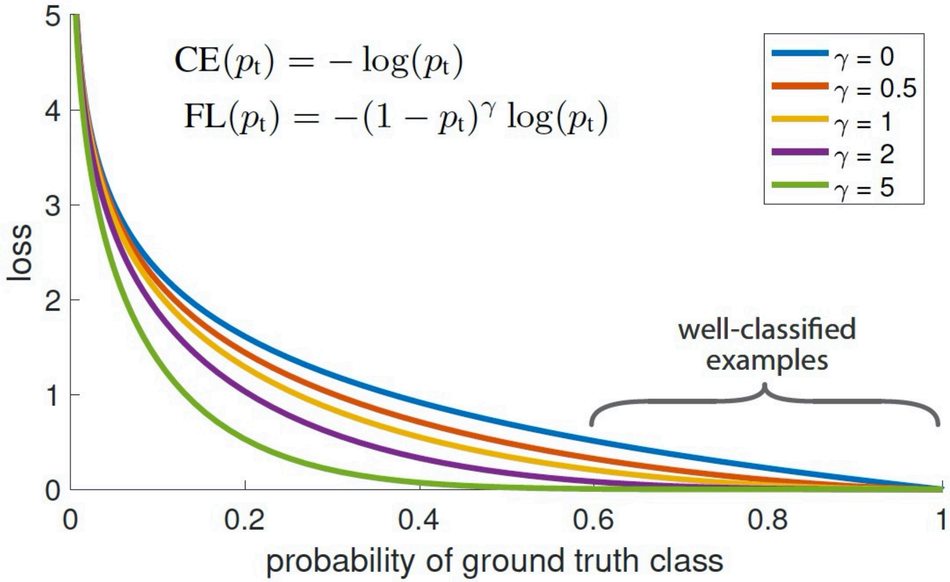
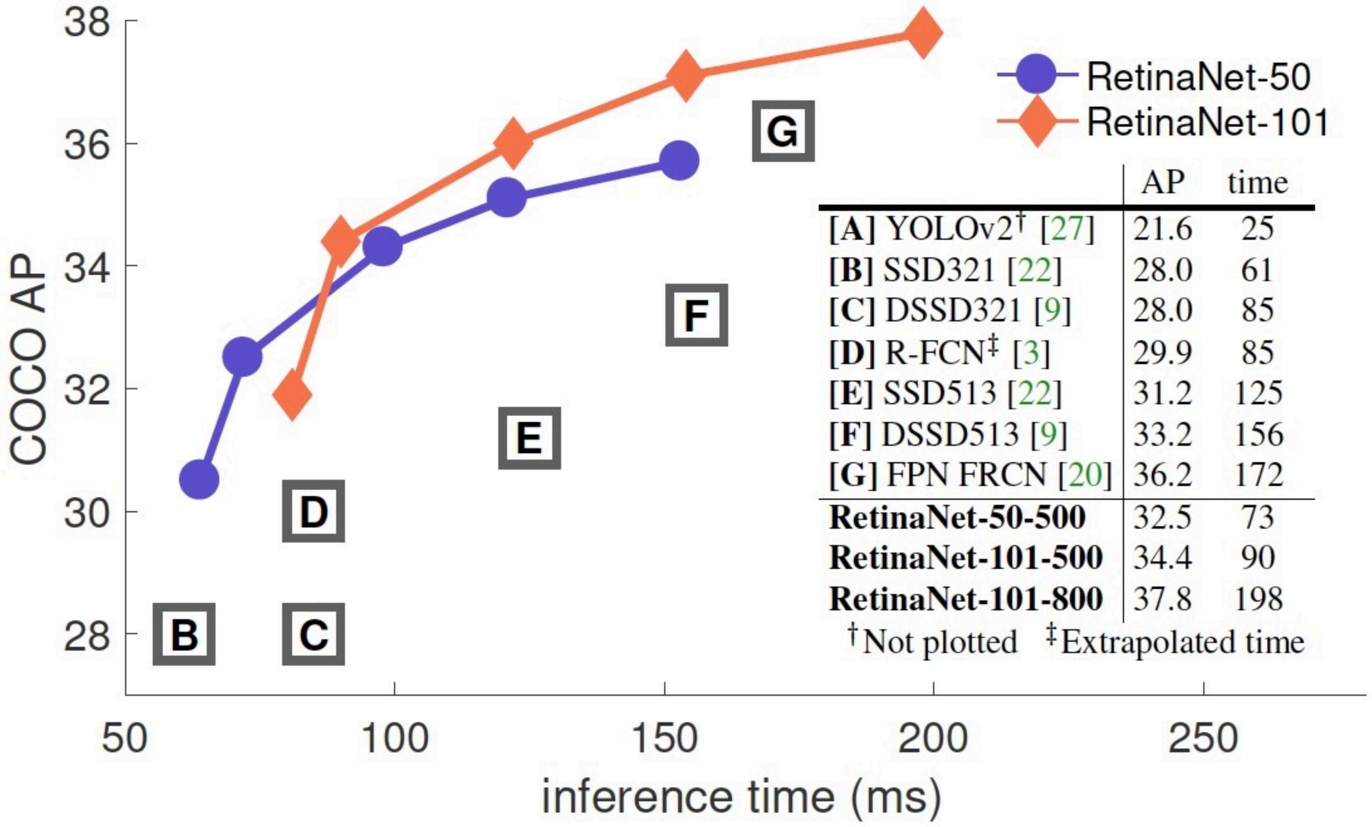


Figure 1. We propose a novel loss we term the *Focal Loss* that adds a factor $(1 - p_t)^\gamma$ to the standard cross entropy criterion. Setting $\gamma > 0$ reduces the relative loss for well-classified examples ($p_t > .5$), putting more focus on hard, misclassified examples. As our experiments will demonstrate, the proposed focal loss enables training highly accurate dense object detectors in the presence of vast numbers of easy background examples.

RetinaNet Performance

- 2017년 기준으로 속도도 빠르면서 기존의 One-Stage Detector 계열과 Two-Stage Detector 계열의 모든 모델의 성능을 뛰어넘음



AI School.

Class Imblance

- Both classic one-stage object detection methods, like booste d detectors [37, 5] and DPMs [8], and more recent methods, l ike SSD [22], face a large class imbalance during training.
- These detectors evaluate $10^4 - 10^5$ candidate locations per i mage but only a few locations contain objects.

Focal Loss

- We introduce the focal loss starting from the cross entropy (CE) loss for binary classification

$$\mathrm{CE}(p, y) = \begin{cases} -\log(p) & \text{if } y = 1 \\ -\log(1-p) & \text{otherwise.} \end{cases} \quad (1)$$

- In the above $y \in \{\pm 1\}$ specifies the ground-truth class and $p \in [0,1]$ is the model's estimated probability for the class with label $y = 1$. For notational convenience, we define p_t:

$$p_{\mathrm{t}} = \begin{cases} p & \text{if } y = 1 \\ 1-p & \text{otherwise,} \end{cases} \quad (2)$$

and rewrite $\mathrm{CE}(p, y) = \mathrm{CE}(p_{\mathrm{t}}) = -\log(p_{\mathrm{t}})$.

Balanced Cross Entropy & Focal Loss

- A common method for addressing class imbalance is to introduce a weighting factor $\alpha \in [0,1]$ for class 1 and $1 - \alpha$ for class -1.
- We write the **α-balanced CE loss** as:

$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t). \tag{3}$$

- While α balances the importance of positive/negative examples, it does not differentiate between easy/hard examples.

$$\mathrm{FL}(p_t) = -(1 - p_t)^\gamma \log(p_t). \tag{4}$$

- (we found γ=2 to work best in our experiments).

Focal Loss

- For instance, with $\gamma = 2$, an example classified with $p_t = 0.9$ would have 100x lower loss compared with CE and with $p_t \approx 0.968$ it would have 1000x lower loss.

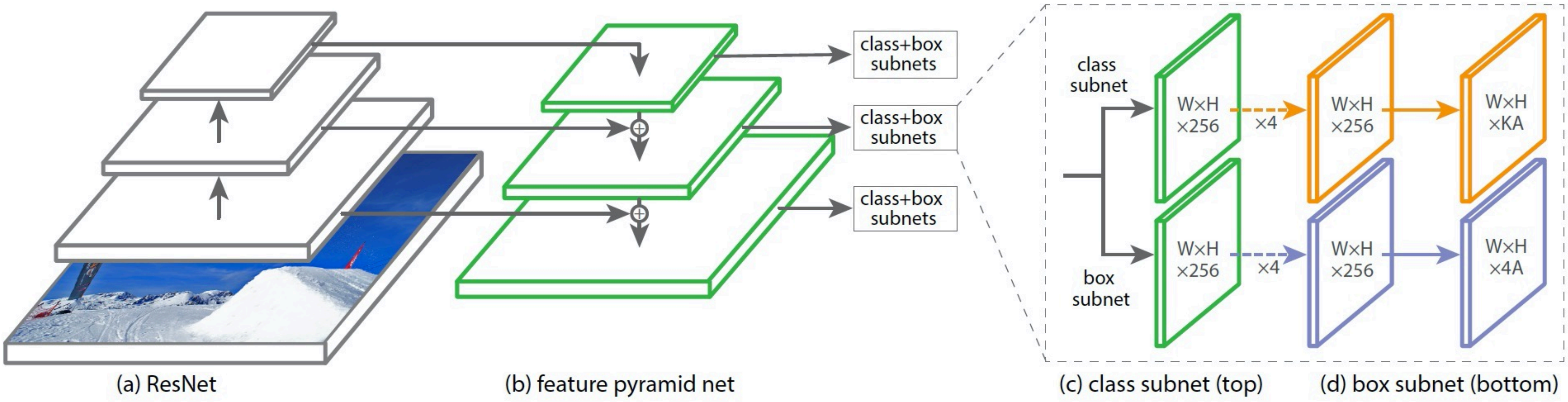
$$\mathrm{CE}(p_t) = -\alpha_t \log(p_t). \qquad (3)$$

- In practice we use an α -balanced variant of the focal loss:

$$\mathrm{FL}(p_t) = -\alpha_t (1 - p_t)^{\gamma} \log(p_t). \qquad (5)$$

RetinaNet Architecture

- Classification Subnet + Bounding Box Regression Subnet
- RPN과 동일하게 3개의 Aspect Ratio와 3개의 Scale의 Anchor 사용 (Anchor Difference 계산)



RetinaNet Peformance

- 2017년 기준 State-of-the-art

	backbone	AP	AP_{50}	AP_{75}	AP_S	AP_M	AP_L
Two-stage methods							
Faster R-CNN+++ [16]	ResNet-101-C4	34.9	55.7	37.4	15.6	38.7	50.9
Faster R-CNN w FPN [20]	ResNet-101-FPN	36.2	59.1	39.0	18.2	39.0	48.2
Faster R-CNN by G-RMI [17]	Inception-ResNet-v2 [34]	34.7	55.5	36.7	13.5	38.1	52.0
Faster R-CNN w TDM [32]	Inception-ResNet-v2-TDM	36.8	57.7	39.2	16.2	39.8	**52.1**
One-stage methods							
YOLOv2 [27]	DarkNet-19 [27]	21.6	44.0	19.2	5.0	22.4	35.5
SSD513 [22, 9]	ResNet-101-SSD	31.2	50.4	33.3	10.2	34.5	49.8
DSSD513 [9]	ResNet-101-DSSD	33.2	53.3	35.2	13.0	35.4	51.1
RetinaNet (ours)	ResNet-101-FPN	39.1	59.1	42.3	21.8	42.7	50.2
RetinaNet (ours)	ResNeXt-101-FPN	**40.8**	**61.1**	**44.1**	**24.1**	**44.2**	51.2

Table 2. **Object detection** *single-model* results (bounding box AP), *vs*. state-of-the-art on COCO `test-dev`. We show results for our RetinaNet-101-800 model, trained with scale jitter and for 1.5× longer than the same model from Table 1e. Our model achieves top results, outperforming both one-stage and two-stage models. For a detailed breakdown of speed versus accuracy see Table 1e and Figure 2.

ReitnaNet의 장점과 단점

- **장점** :

① 후속 모델 Object Detection 모델 들에도 큰 영향을 끼치는 Focal Loss라는 새로운 개념을 제안

- **단점** :

① 여전히 1장의 Image에 대해 약 100K(100,000)장의 Anchor Prediction과 NMS(Non-Maximum Supression) 과정이 필요함

Thank you!