RetinaNet

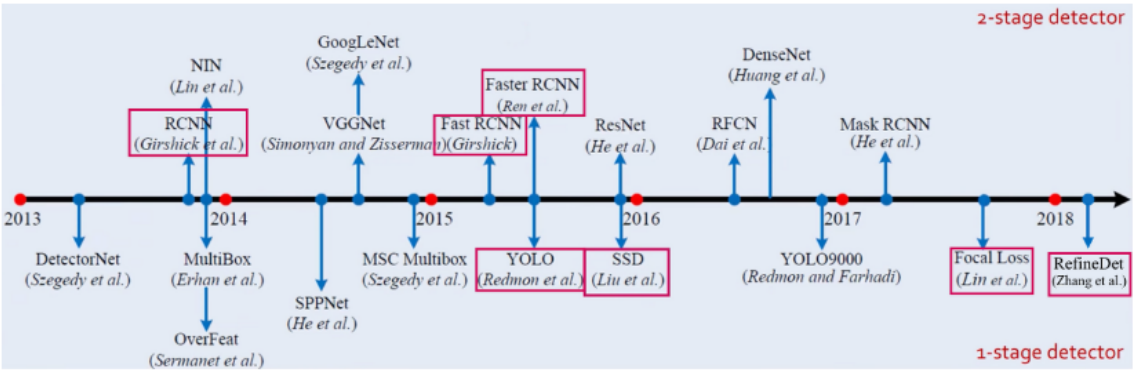
Object Detection 모델은 이미지 내의 객체를 추정하고 IOU threshold에 따라서 positive (객체) / negative sample (배경)로 구분하고, 이를 활용하여 학습

일반적으로 이미지 내 객체 수가 적기 때문에 객체 영역이 배경영역에 비해 적으므로 positive / negative sample 사이에 큰 차이가 발생하여 class imbalance 문제가 발생

Class Imbalance

- 첫 번째로 대부분의 sample이 easy negative, 즉 모델이 class를 예측하기 쉬운 sample이기 때무에 유용한 기여를 하지 못해 학습이 비효율적으로 진행
- 두 번째로 easy negative의 수가 압도적으로 많기 때문에 학습에 끼치는 영향력이 커져 모델의 성능이 하락

1-Stage Detector vs 2-Stage Detector



1-Stage Detector vs 2-Stage Detector

Regional Proposal

기존에는 이미지에서 Object Detection을 위해 Sliding Window 방식을 사용

Sliding Window 방식은 이미지에서 모든 영역을 다양한 크기의 Window로 탐색하는 것

- 비효율성 개선을 위해 '물체가 있을 만한' 영역을 빠르게 찾아내는 알고리즘
- Object의 위치를 찾는 Localization 문제

- Selective Search, Edge Boxes
  - Selective Search는 비슷한 질감, 색, 강도를 갖는 인접 픽셀로 구성된 다양한 크기의 Window 생성

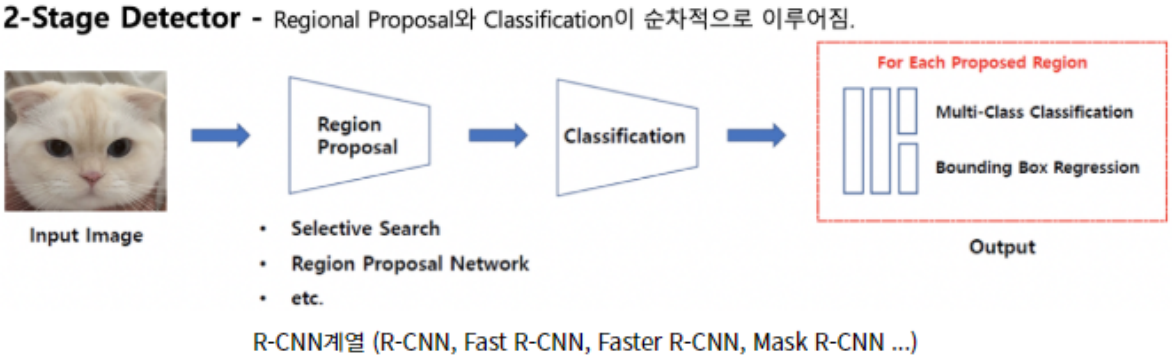
1-Stage Detector

Regional Proposal과 Classification이 동시에 이루어짐

Classification과 Localization 문제를 동시에 해결하는 방법

비교적 빠르지만 정확도가 낮음

2-Stage Detector



Regional Proposal과 Classification이 순차적으로 이루어짐

Classification과 Localization 문제를 순차적으로 해결하는 방법

비교적 느리지만 정확도가 높음

2-Stage Detector 계열의 모델

- 첫 번째로 2-Stage Cascade, 즉 Region Proposals를 추려내는 방법을 적용하여 대부분의 Background Sample을 걸러주는 방법 사용
  예를 들어, Selective Search, Edge Boxes, Deepmask, RPN 등
- 두 번째로 positive/negative sample의 수를 적절하게 유지하는 Sampling Heuristic 방법을 적용
  예를 들어, Hard Negative Mining, OHEM
- 1-Stage Detector는 Region Proposal 과정이 없기 때문에 전체 이미지를 빽빽하게 순회하면서 Sampling 하는 Dense Dampling 방법을 수행
  - 2-Stage Detector에 비해 훨씬 더 많은 후보 영역을 생성
  - Class Imbalance 문제가 2-Stage Detector보다 더 심각
  - 기존의 Sampling Heuristic 방법을 적용해도 여전히 배경으로 쉽게 분류된 Sample이 압도적으로 많기 때문에 학습이 비효율적으로 진행

해당 논문에서는 학습 시 Training Imbalance가 주된 문제로 보고, 이러한 문제를 해결하여 1-Stage Detector에서 적용할 수 있는 새로운 Loss-Function을 제시

Preview

RetinaNet 논문에서는 Focal Loss라는 새로운 Loss Function을 제시

Focal Loss란 Cross Entropy Loss에 Class에 따라 변하는 동적인 Scaling Factor를 추가한 형태

이러한 Loss Function을 통해 학습 시 Easy Example의 기여도를 자동적으로 Down-Weight 하며, Hard Example에 대해서 가중치를 높혀서 학습을 집중시킬 수 있음

Focal Loss의 효과를 실험하기 위해서 1-Stage Detector인 RetinaNet 설계

ResNet-101-FPN을 기본 모델로 하며, Anchor Boxes를 적용하여 기존의 2-Stage Detecor에 비해 높은 성능

Main Ideas

CE Loss

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

$$p_t = \begin{cases} p, & \text{if y} = 1 \\ 1-p, & otherwise \end{cases}$$

y∈[1,−1] : ground truth class

p∈[0,1]: 모델이 y=1이라고 예측한 확률

CE Loss는 Sample에 대한 예측 결과를 동등하게 가중치

어떠한 Sample이 쉽게 분류될 수 있음에도 불구하고 작지 않은 Loss를 유발

많은 수의 Easy Example의 Loss가 더해지면 보기 드문 Class를 압도해버려 학습이 제대로 이뤄지지 않음

Balanced Cross Entropy

CE Loss의 문제를 해결하기 위해 가중치 파라미터인 α∈[0,1]를 곱해준 Balanced Cross Entropy 등장

$$CE(p_t) = -\alpha log(p_t)$$

y=1일 때 α를 곱해주고, y=−1일 때 1−α를 곱해주는 방식

positive/negative sample 사이의 균형을 잡아주지만, easy/hard sample에 대해서는 균형을 잡지 못함

논문에서는 Balanced Cross Entropy를 baseline으로 삼고 실험을 진행

Focal Loss

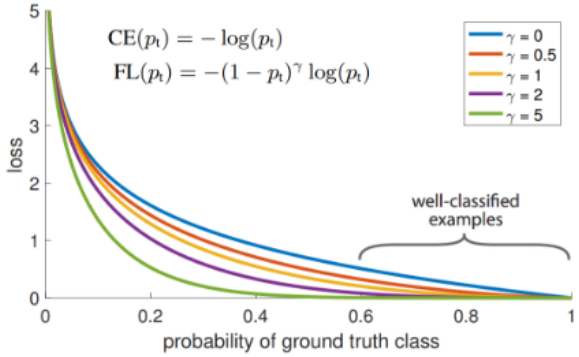
$$FL(p_t) = \begin{cases} -(1-p_t)^{\gamma} log(p_t), & \text{if y} = 1 \\ -(1-(1-p_t))^{\gamma} log(1-p_t), & otherwise \end{cases}$$

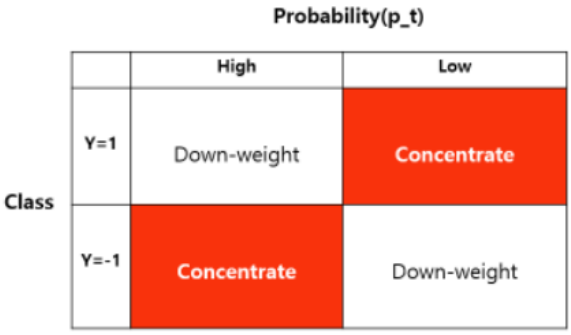
1-Stage Detector 모델에서 Foreground와 Background Class 사이에 발생하는 극단적인 Class Imbalance 문제를 해결하는데 사용

이진 분류에서 사용되는 Cross Entropy Loss Function으로부터 비롯

Easy Example을 Down-Weight하여 Hard Negative Sample에 집중하여 학습하는 Loss Function

Modulating Factor (1−pt)^γ와 Tunable Focusing Parameter γ를 CE에 추가한 형태





서로 다른 γ∈[0,5]값에 따른 Loss

파란색 선은 CE를 의미, 파란색 선은 경사가 완만하여 pt가 높은 Example과 낮은 Example 사이의 차이가 크지 않다는 것을 확인할 수 있음

반면 Focal Loss는 Focusing Parameter γ에 따라 pt가 높은 Example과 낮은 Example 사이의 차이가 상대적으로 크다는 것을 확인할 수 있음

즉, y=1인 Class임에도 pt가 낮은 경우와, y=-1임에도 pt가 높은 경우 Focal Loss가 높음

반대의 경우에는 Down-Weight되어 Loss값이 낮게 나타남

Focal loss의 두 가지 특성

1) pt와 Modulating Factor와의 관계

Example이 잘못 분류되고, pt가 작으면, Modulating Factor는 1과 가까워지며, Loss는 영향을 받지 않음

반대로 pt 값이 크면 Modulating Factor는 0에 가까워지고, 잘 분류된 Example의 Loss는 Down-Weight됨

2) focusing parameter γ의 역할

Focusing Parameter γ는 Easy Example을 Down-Weight하는 정도를 부드럽게 조정

γ=0인 경우, Focal Loss는 CE와 같으며, γ가 상승할수록 modulating factor의 영향력이 커짐

논문에서는 실험 시 γ=2일 때 가장 좋은 결과

직관적으로 봤을 때, Modulating Factor는 Easy Example의 기여도를 줄이고 Example이 작은 Loss를 받는 범위를 확장시키는 기능

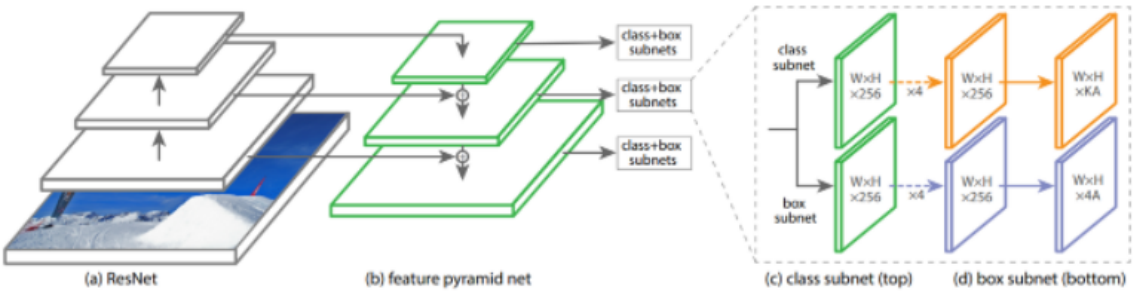
예를 들어 γ=2, pt=0.9일 때, CE에 비해 100배 적은 Loss를 가지며, pt=0.968일 때는 1000배 적은 Loss를 가짐
이는 잘못 분류된 Example을 수정하는 작업의 중요도를 상승시킴을 의미

RetinaNet = ResNet-101 + FPN + Focal Loss

논문에서는 Focal Loss를 실험하기 위해 RetinaNet이라는 1-Stage Detector을 설계

하나의 Backbone Network와 각각 Classification과 Bounding Box Regression을 수행하는 2개의 Subnetwork로 구성

Training RetinaNet



1) Feature Pyramid by ResNet + FPN

먼저 이미지를 Backbone Network에 입력하여 서로 다른 5개의 Scale을 가진 Feature Pyramid를 출력

- Backbone Network는 ResNet 기반의 FPN(Feature Pyramid Network)를 사용
- pyramid level은 P3~P7로 설정
    - Input : image

- Process : feature extraction by ResNet + FPN
- Output : feature pyramid(P5~P7)

2) Classification by Classification subnetwork

1)번 과정에서 얻은 각 Pyramid Level별 Feature Map을 Classification Subnetwork에 입력

- Subnet는 3x3(xC) Conv Layer - ReLU - 3x3(xKxA) Conv Layer로 구성
  - K는 분류하고자 하는 Class의 수
  - A는 Anchor Box의 수 논문에서는 A=9로 설정

마지막으로 얻은 Feature Map의 각 Spatial Location(Feature Map의 Cell)마다 Sigmoid Activation Function을 적용

이를 통해 Channel 수가 KxA인 5개(Feature Pyramid의 수)의 Feature Map을 얻을 수 있음

- Input : feature pyramid(P5~P7)
- Process : classification by classification subnetwork
- Output : 5 feature maps with KxA channel

3) Bounding box regression by Bounding box regression subnetwork

1)번 과정에서 얻은 각 Pyramid Level별 Feature Map을 Bounding Box Regression Subnetwork에 입력

- 해당 Subnet은 Classification Subnet과 마찬가지로 FCN(Fully Convolutional Network)

Feature Map이 Anchor Box별로 4개의 좌표값(x, y, w, h)을 encode하도록 channel 수를 조정

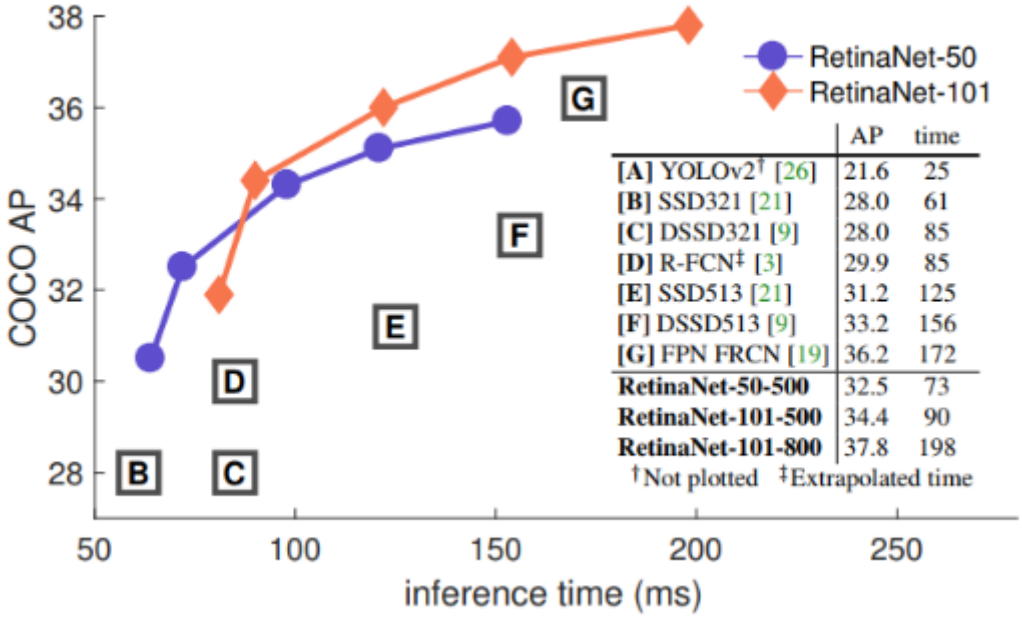
최종적으로 Channel 수가 4xA인 5개의 Feature Map을 얻을 수 있음

- Input : feature pyramid(P5~P7)
- Process : bounding box regression by bounding box regression subnet
- Output : 5 feature maps with 4xA channel

Inference

Inference 시에는 속도를 향상시키기 위해 각 FPN의 Pyramid Level에서 가장 점수가 높은 1000개의 prediction만을 사용

2개의 Subnetwork의 출력 결과에서 모든 level의 예측 결과는 병합
Non maximum suppression(threshold=0.5)를 통해 최종 예측 결과를 산출



	AP	time
[A] YOLOv2† [26]	21.6	25
[B] SSD321 [21]	28.0	61
[C] DSSD321 [9]	28.0	85
[D] R-FCN‡ [3]	29.9	85
[E] SSD513 [21]	31.2	125
[F] DSSD513 [9]	33.2	156
[G] FPN FRCN [19]	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	37.8	198

†Not plotted ‡Extrapolated time

RetinaNet을 COCO 데이터셋을 통해 학습시킨 후 서로 다른 Loss Function을 사용하여 AP 값을 측정

- CE loss는 30.2%, Balanced Cross Entropy는 31.1%, Focal loss는 34% AP 값

SSD 모델을 통해 positive/negative 비율을 1:3으로, NMS threshold=0.5로 설정한 OHEM과 성능을 비교

- Focal Loss를 사용한 경우의 AP값이 3.2% 더 높게 나타남

이를 통해 Focal loss가 Class Imbalance 문제를 기존의 방식보다 효과적으로 해결

OHEM은 Class Imbalance 문제를 해결하기 위해 별도의 네트워크(readonly RoI network)를 설계
RetinaNet은 Loss Function을 수정하는 비교적 단순한 방법으로 성능을 향상