





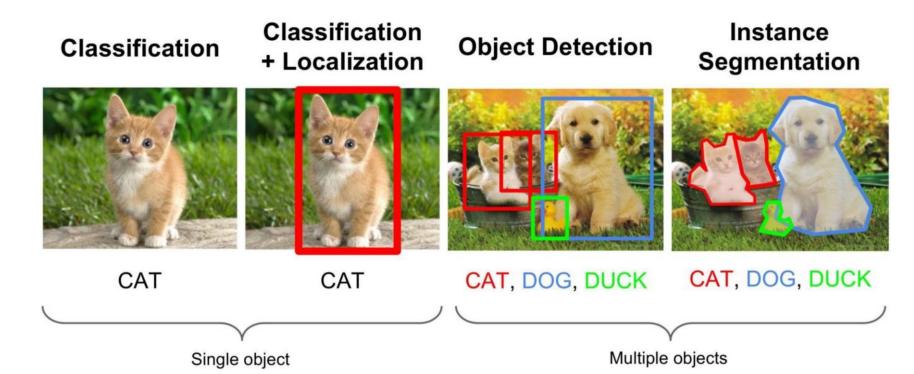
• 객체 탐지의 정의

객체탐지/이미지분류

객체탐지 알고리즘

객체탐지(Object Detection)의 정의

- 객체 탐지는 컴퓨터 비전과 이미지 처리와 관련된 컴퓨터 기술
- 디지털 <mark>이미지와 비디오</mark>로 특정한 계열의 <mark>시맨틱 객체 인스턴스</mark> (예: 인간, 건물, 자동차)를 감지하는 일을 다룸
- 잘 연구된 객체 탐지 분야로는 얼굴 검출, 보행자 검출이 포함됨







● 객체탐지/이미지분류

객체탐지 알고리즘

객체탐지와 이미지분류

- 이미지분류 (Image Classification) 에서는 이미지에 주 대상 물체가 하나만 있다고 가정하며, 모델의 역할은 이 대상 물체의 카테고리가 무엇인지 식별하는 것임
- 객체탐지(Object Detection) 에서는 이미지에 하나 이상의 대상이 존재하고, 이미지 상에서 대상의 카테고리 뿐만 아니라 위치도 함께 특정해야 하는 경우



● 객체탐지/이미지분류

객체탐지 알고리즘

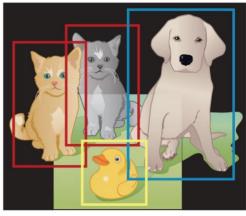
객체탐지와 이미지분류

- 이미지분류 (Image Classification) 에서는 이미지에 주 대상 물체가 하나만 있다고 가 정하며, 모델의 역할은 이 대상 물체의 카테고리가 무엇인지 식별하는 것임
- 사물탐지(Object Detection) 에서는 이미지에 하나 이상의 대상이 존재하고, 이미지 상에서 대상의 카테고리 뿐만 아니라 위치도 함께 특정해야 하는 경우

Image classification



Object detection (classification and localization)



Cat, Cat, Duck, Dog

Cat

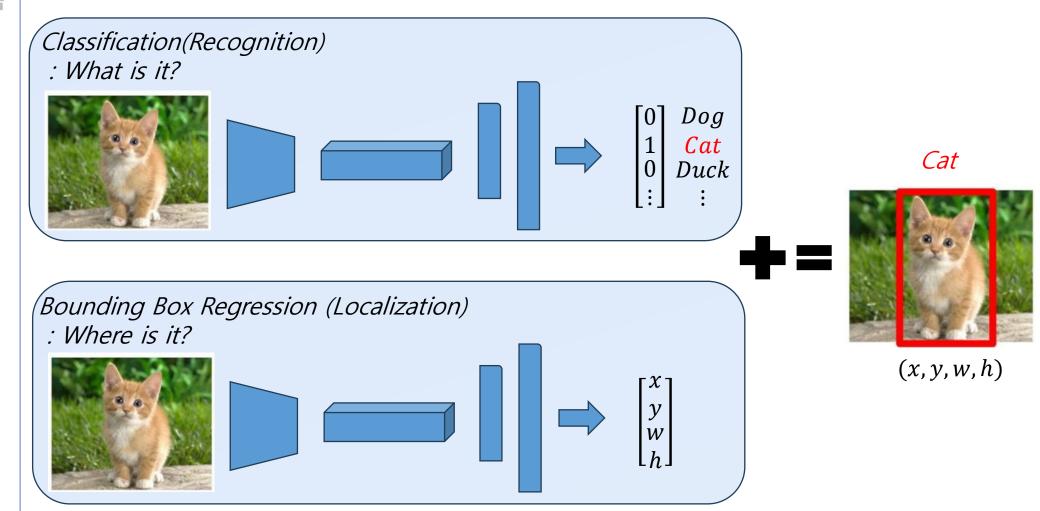


● 객체탐지/이미지분류

객체탐지 알고리즘

객체탐지는 분류와 회귀를 동시에 하는 task!

- 객체의 class를 '분류'하고 동시에 bounding box를 찾는 '회귀' 를 함께!





● 객체탐지/이미지분류

객체탐지 알고리즘

정리

Object Detection(객체 탐지) Image Classification(이미지 분류) 이미지에 있는 (하나 이상의) 대상의 위치를 나타내는 이미지에 있는 (하나뿐인) 대상의 클래스를 예 박스와 해당 물체의 클래스를 예측하는 것이 목표 측하는 것이 목표 · 입력: 하나 이상의 대상이 있는 이미지 ㆍ 입력: 하나의 대상이 있는 이미지 · 출력: 각 대상의 위치를 특정하는 경계박스(좌표)와 · 출력: 클래스 레이블 (고양이,개,등) · 출력 예: 각 클래스에 속할 확률 (고양이 84%) 해당 물체의 클래스 레이블 · *출력 예:* - box1의 좌표(x,y,w,h)와 클래스에 속할 확률 - box2의 좌표(x,y,w,h)와 클래스에 속할 확률 (참고 : 좌표(x,y,w,h)에서 x와 y는 경계 박스의 중심점 좌표고, w와 h는 경계 박스의 너비와 높이다.)





• 객체탐지 알고리즘

객체탐지 알고리즘

<목차>

- 1) Region Proposal
- 2) 1-stage detector VS 2-stage detector
- 3) NMS(Non-Max Suppression)
- 4) mAP



• 객체탐지 알고리즘

Region Proposal (영역제안)

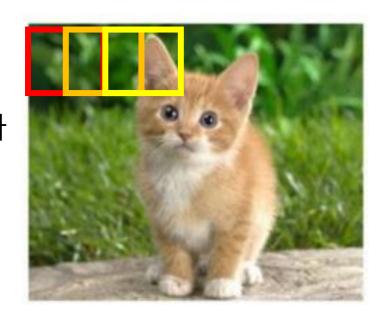
- 객체가 있을 만한 영역들의 후보군들을 여러 개 추출해주는 것
- 객체들을 나타내는 픽셀들만을 찾고 탐지하기 위해 Region Proposal이 객체가 '**있을 법한**' 픽셀의 영역들을 대거 추출해 추천해주는 방식
- Region Proposal에는 3가지 방법을 소개하겠다.
 - (1) Sliding Window
 - (2) Selective Search
 - (3) RPN



• 객체탐지 알고리즘

Region Proposal - Sliding Window

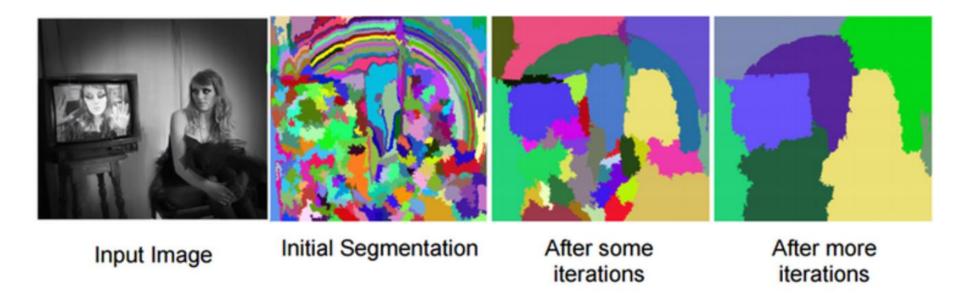
- 특정한 모양의 윈도우를 이미지 왼쪽 상단에서
 오른쪽 하단으로 점진적으로 이동하면서 객체가
 있을 만한 Region들을 Proposal(제안)해주는
 것.
- 객체가 없는 지역도 슬라이딩을 무조건 하기 때문에 슬라이딩 윈도우를 하는 데 시간이 많이 걸릴 뿐더러 객체를 잘 탐지할 확률도 낮아지게 됨.





• 객체탐지 알고리즘

Region Proposal - Selective Search



- 일종의 알고리즘으로 이미지 픽셀의 컬러, 무늬, 크기, 형태에 따라 유사한 Region을 계층적 그룹핑 방법으로 계산하는 방식.
- 객체가 들어있을 만한 여러 개의 바운딩 박스를 만들고 반복적으로 Selective Search 알고 리즘을 이용해 최적의 바운딩 박스를 선정해 Region Proposal을 해주는 것이다.
- 초기에 픽셀 개별로 Segmentation된 부분을 유사도에 따라 Segmentation 그룹핑을 계속 반복하는 것이다.



• 객체탐지 알고리즘

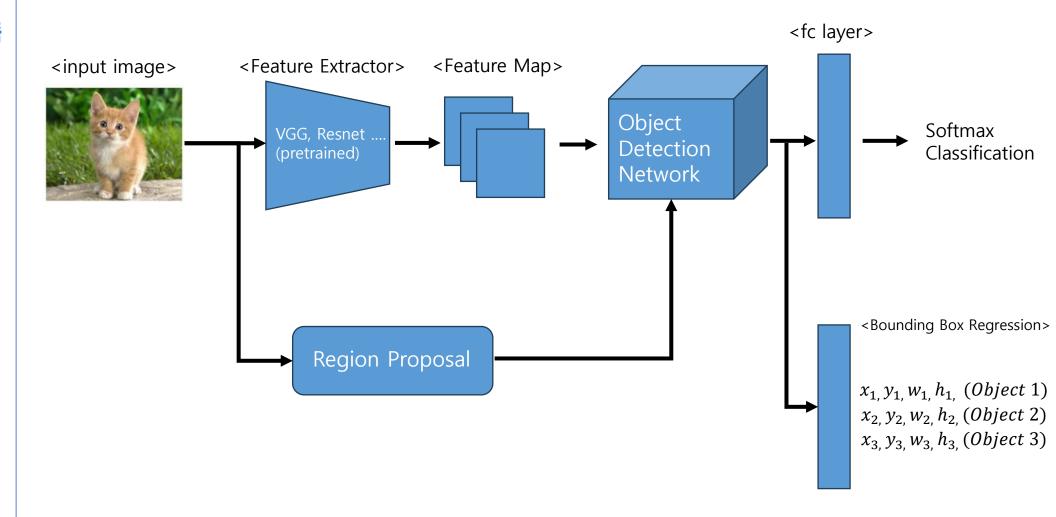
Region Proposal - RPN

- CNN을 직접 사용하는 Region Proposal 기법의 대표적인 예로는 Region Proposal Network (RPN)가 있다.
- 이미지의 특징 맵을 CNN을 통해 추출한다.
- 이 특징 맵을 기반으로 특정 영역에 객체가 있을 확률과 해당 영역의 경계를 예측한다.
- RPN은 학습 가능한 네트워크로, 이미지의 다양한 영역을 효율적으로 스캔하여 객체가 있을 가능성이 높은 후보 영역을 제안한다.
- RPN은 CNN을 사용하여 후보 영역을 직접 생성하고, 이러한 후보 영역을 객체 탐지 및 분류 단계로 넘긴다.



• 객체탐지 알고리즘

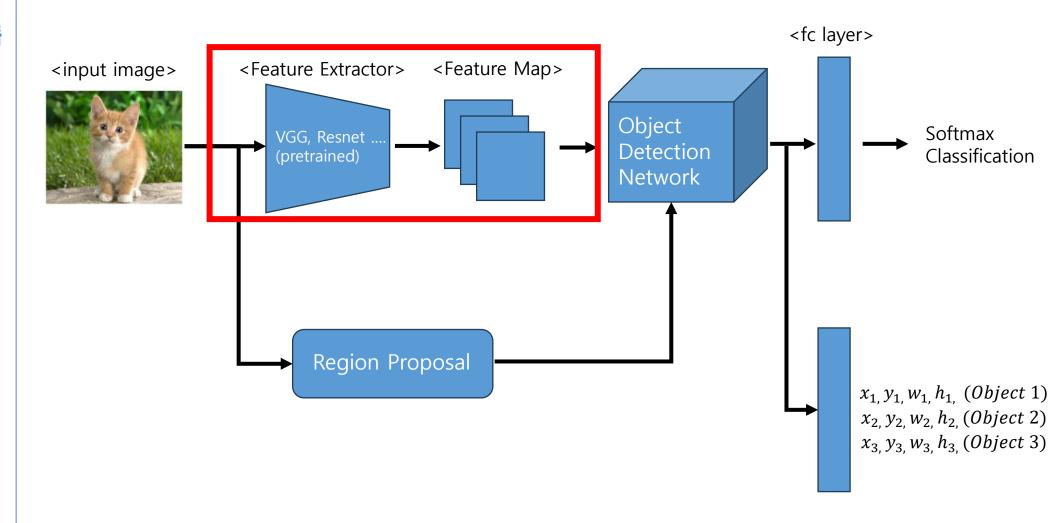
전체적인 파이프라인





• 객체탐지 알고리즘

Feature Extraction/Mapping

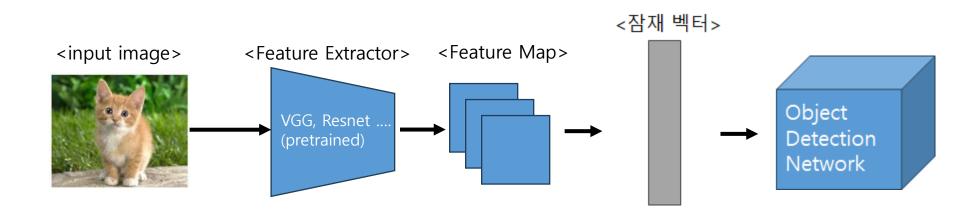




• 객체탐지 알고리즘

Region Proposal - Feature Extraction/Mapping

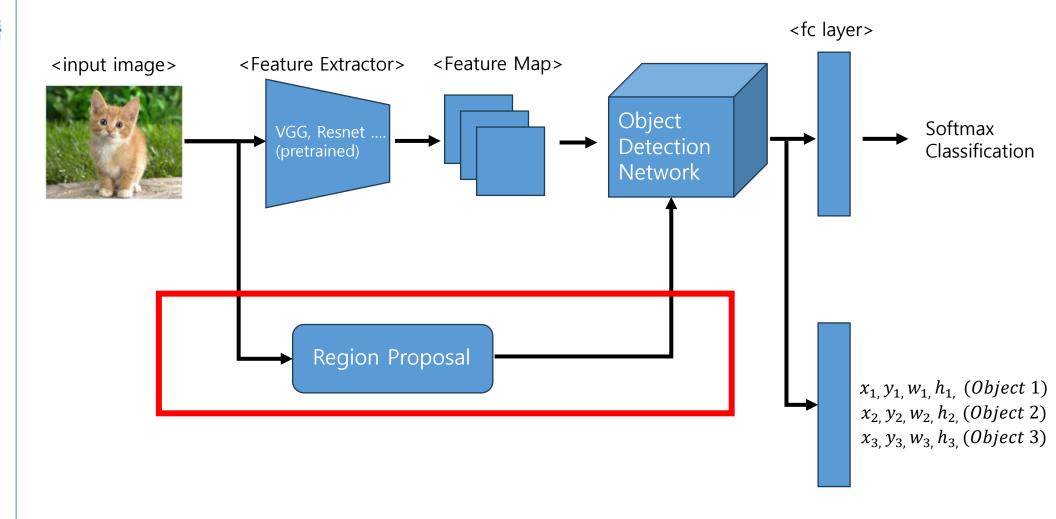
- 이 모듈에는 특징 추출을 위해 <mark>사전 학습된 합성곱</mark> 신경망이 포함되며,
 일반화성능이 높은 이미지 분류 모델을 선택해서 사용함
- 모듈에서 추출된 특징을 이용해 물체의 유형을 분류하는데 사용함
- MS COCO나 ImageNet 데이터셋을 학습한 사전학습모델은 대부분 물체의 특징을 잘 추출할 수 있음 (즉, 일반화 성능이 좋음)





• 객체탐지 알고리즘

Region Proposal - Rol





• 객체탐지 알고리즘

Region Proposal - Rol

- 딥러닝 모델이 이미지를 관찰하고 추가 분석이 필요한 <mark>관심 영역(Rol)</mark>를 제안
- Rol는 시스템이 이미지의 해당 위치에 높은 확률(Objectness score, 물체존재 확신도)로 물체가 존재한다고 판단한 영역
- 모델은 각각 물체 Objectness score가 매겨진 많은 수의 박스 정보를 출력.
- Objectness score가 높은 영역은 다음 단계로 정보가 전달되고, 낮은 영역은 추가 분석을 하지 않고 해당 영역의 정보를 폐기한다.

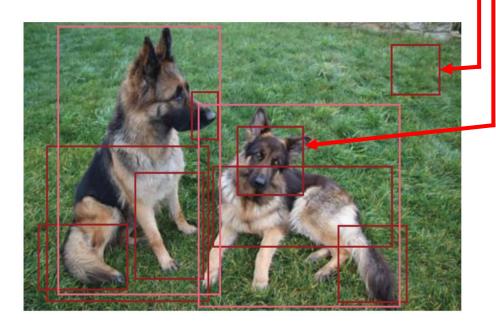


• 객체탐지 알고리즘

Region Proposal - Rol

물체 존재 확신도가 높음

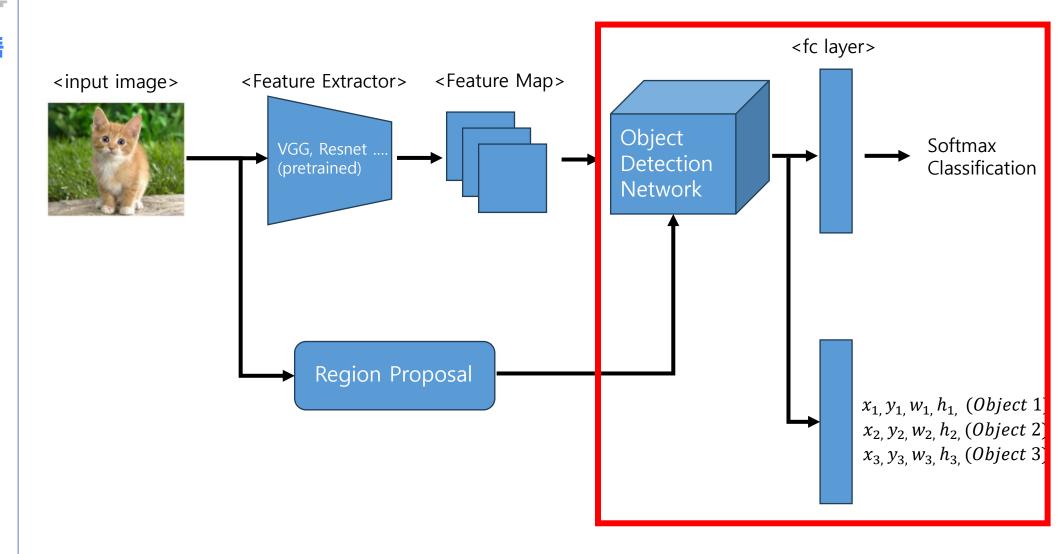
물체 존재 확신도가 낮음 (배경)





• 객체탐지 알고리즘

Prediction





• 객체탐지 알고리즘

Prediction

- 이 단계에서는 물체를 포함했을 가능성이 높다고 판단된 모든 영역 (Region Proposal을 통과한 영역)을 신경망이 분석하며 각 영역마다 다음 두가지를 예측함
 - 경계 박스 예측과 클래스 예측



• 객체탐지 알고리즘

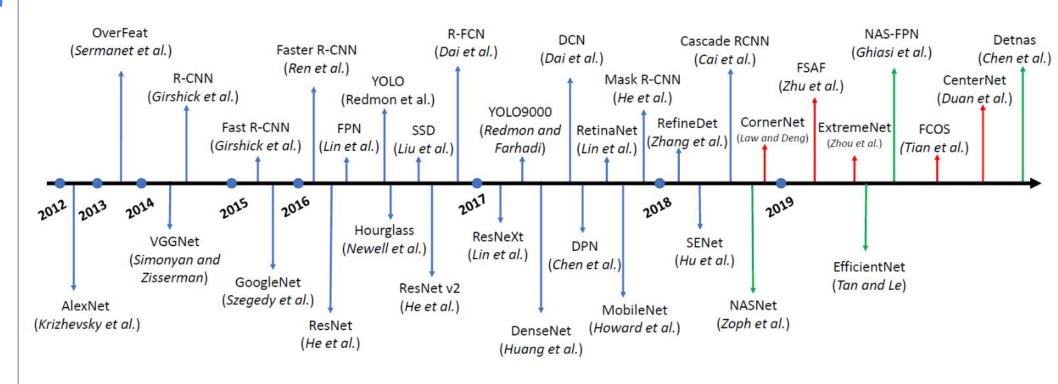
Prediction

- 이 단계에서는 물체를 포함했을 가능성이 높다고 판단된 모든 영역 (Region Proposal을 통과한 영역)을 신경망이 분석하며 각 영역마다 다음 두가지를 예측함
 - 경계 박스 예측과 클래스 예측
 - 경계 박스(Bounding Box, BB) 예측 : 물체를 감싸는 박수의 좌표를 예측함. 경계 박스의 좌표는 튜플(x,y,w,h)로 표현됨. x와 y는 경계 박스의 중심점 좌표이고, w와 h는 경계 박스의 너비와 높이임
 - 클래스 예측: 해당 영역의 물체가 각 클래스에 속할 확률을 예측. (소프트맥스 함수 사용)



• 객체탐지 알고리즘

발전 과정





• 객체탐지 알고리즘

발전 과정

```
R-CNN → OverFeat → MultiBox → SPP-Net → MR-CNN → DeepBox → AttentionNet →
                                                                      ICCV' 15
                                                                                                           ICCV' 15
                                                    ECCV' 14
                                                                                       ICCV' 15
                                  CVPR' 14
                 ICLR' 14
   2013.11
 Fast R-CNN → DeepProposal → Faster R-CNN → OHEM → YOLO v1 → G-CNN → AZNet →
                                                                                                CVPR' 16
                                                                                                              CVPR' 16
                                                                                 CVPR' 16
                           ICCV' 15
    ICCV' 15
 Inside-OutsideNet(ION) \rightarrow HyperNet \rightarrow CRAFT \rightarrow MultiPathNet(MPN) \rightarrow SSD \rightarrow
                                                                                                            GBDNet →
                                                                                               ECCV' 16
                                                                                                              ECCV' 16
                                                                         BMVC' 16
                                                      CVPR' 16
                                      CVPR' 16
            CVPR' 16
 CPF \rightarrow MS-CNN \rightarrow R-FCN \rightarrow PVANET \rightarrow DeepID-Net \rightarrow NoC \rightarrow DSSD \rightarrow TDM \rightarrow YOLO v2 \rightarrow
                                                                                                              CVPR' 17
                                                                                               CVPR' 17
                                                          PAMI' 16
                                                                        TPAMI' 16
                                                                                    arXiv' 17
                                        NIPSW' 16
                           NIPS' 16
ECCV' 16
            ECCV' 16
Feature Pyramid Net(FPN) → RON → DCN → DeNet → CoupleNet → RetinaNet → DSOD →
                                                                                                  ICCV' 17
                                                                                                                ICCV' 17
                                                                                ICCV' 17
                                                                 ICCV' 17
                                       CVPR' 17
                                                   ICCV' 17
             CVPR' 17
Mask R-CNN \rightarrow SMN \rightarrow YOLO v3 \rightarrow SIN \rightarrow STDN \rightarrow RefineDet \rightarrow MLKP \rightarrow Relation-Net \rightarrow
                                                               CVPR' 18
                                                                                                            CVPR' 18
                      ICCV' 17
                                                  CVPR' 18
                                                                             CVPR' 18
                                                                                            CVPR' 18
    ICCV' 17
                                     arXiv' 18
Cascade R-CNN → RFBNet → CornerNet → PFPNet → Pelee → HKRM → R-DAD →
                                                                                                             AAAI' 19
                                                                                               AAAI' 19
                                                                       NIPS' 18
                          ECCV' 18
                                         ECCV' 18
                                                          ECCV' 18
       CVPR' 18
```



● 객체탐지 알고리즘

1 stage VS 2 stage

[1-stage(single stage) detector]

- regional proposal 과 classification이 동시 진행하는 것으로 객체의 검출과 분류, 바운딩 박스 regression을 한번에 하는 방법
- 1-stage detector의 예로는 YOLO(You only look once), SSD(Single-Shot Multibox Detector)가 있다.

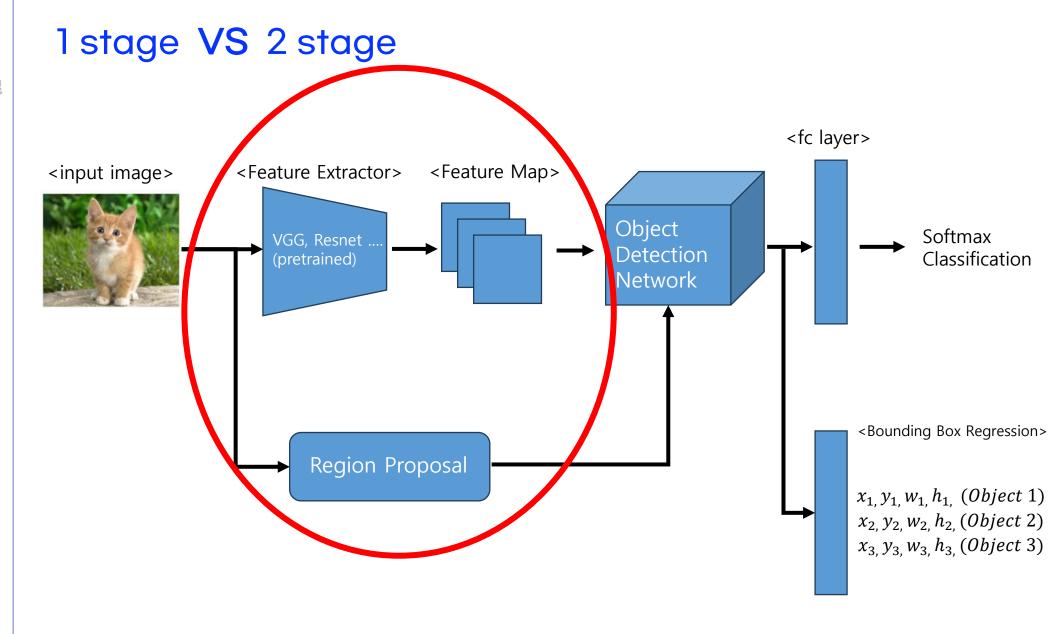
[2-stage detector]

- Regional proposal과 classification이 순차적으로 진행되는 방법
- 2-stage detector의 예로는 R-CNN, Fast R-CNN, Faster R-CNN 등이 있다.

두가지 방법의 차이는 <mark>속도</mark>(1-stage를 통해서 한번에 진행하는 것이 빠르다.)



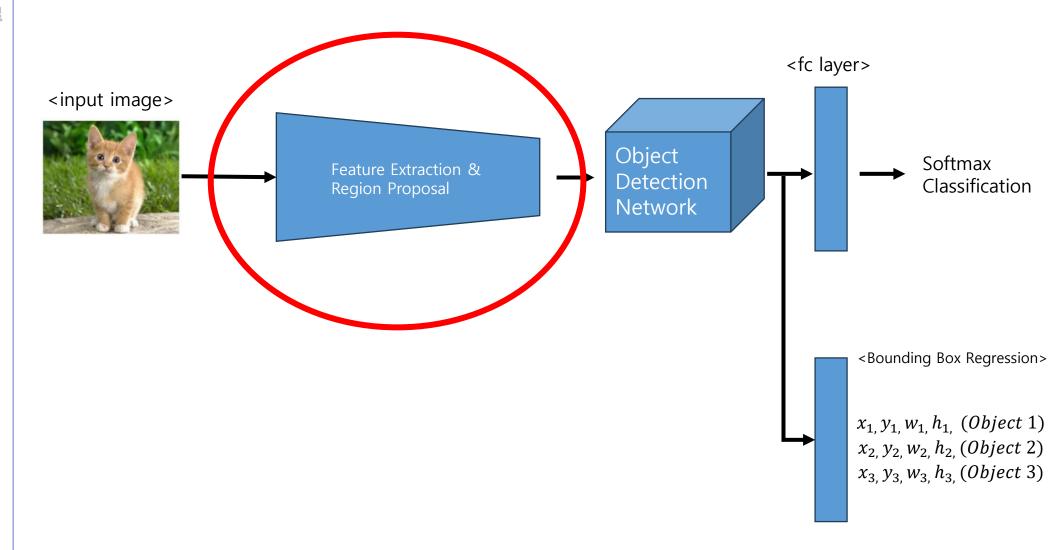
• 객체탐지 알고리즘





• 객체탐지 알고리즘

1 stage VS 2 stage

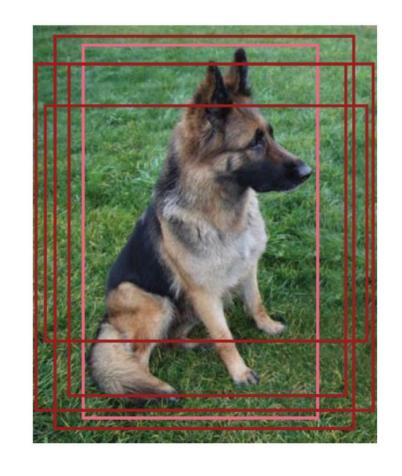




• 객체탐지 알고리즘

Region Proposal의 이슈

- 대부분 물체마다 여러 경계 박스가 생김
- 오른쪽 그림과 같이 개 이미지를 보면, 신경망이 물체는 잘 찾아낸 것을 볼 수 있음
- 예를 들어 이전 단계에서 개가 포함된 Rol가
 5개 제안되었기 때문에 개 주변에 경계 박스도
 5개 표시됨. 물체의 위치와 종류가 모두 정확하게 파악됐지만, 이 상태로는 우리가 원하는
 결과가 되지 못함

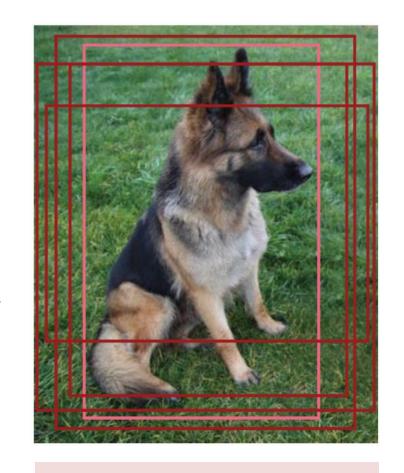




• 객체탐지 알고리즘

Region Proposal의 이슈

- 대부분 물체마다 여러 경계 박스가 생김
- 오른쪽 그림과 같이 개 이미지를 보면, 신경망이 물체는 잘 찾아낸 것을 볼 수 있음
- 예를 들어 이전 단계에서 개가 포함된 Rol가
 5개 제안되었기 때문에 개 주변에 경계 박스도
 5개 표시됨. 물체의 위치와 종류가 모두 정확하게 파악됐지만, 이 상태로는 우리가 원하는
 결과가 되지 못함



객체 1개에 여러 개의 경계 박스를 예측함. 같은 객체를 포함하는 여러 개의 경계 박스를 객체를 가장 잘 포함하는 박스로 통합 필요



• 객체탐지 알고리즘

NMS (Non-Maximum Suppression)

- 객체 탐지 알고리즘의 문제점 중 하나는 같은 물체를 여러 번 탐지한다는 것이며, 이 때문에 물체 하나에 경계박스가 여러 개 표시될 수 있음.
- NMS는 객체 하나에 경계박스가 하나만 남도록 하는 기법
- 이름에서 짐작할 수 있듯이 점수가 가장 높은 경계박스만 남기고 나머지는 배제하는 방식

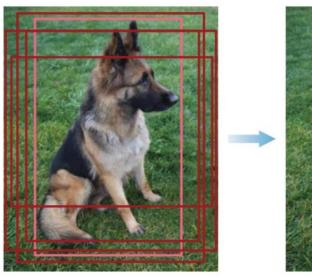


• 객체탐지 알고리즘

NMS (Non-Maximum Suppression)

- 객체 탐지 알고리즘의 문제점 중 하나는 같은 물체를 여러 번 탐지한다는 것이며, 이 때문에 물체 하나에 경계박스가 여러 개 표시될 수 있음.
- NMS는 객체 하나에 경계박스가 하나만 남도록 하는 기법
- 이름에서 짐작할 수 있듯이 점수가 가장 높은 경계박스만 남기고 나머지는

배제하는 방식



redictions before NMS



After applying non-maximum suppression



• 객체탐지 알고리즘

NMS (Non-Maximum Suppression)

- (1) Confidence Score
- 특정 바운딩 박스안에 있는 객체가 어떤 물체의 클래스일 확률
- Softmax 단계로 도출된 확률값 X IoU
- 즉, 각 바운딩 박스안에 물체가 있을 확률

(2) IoU

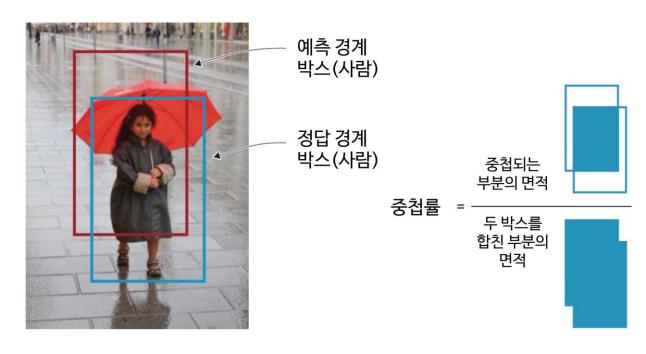
- 2개의 바운딩 박스가 중첩되는 정도를 나타내는 값



• 객체탐지 알고리즘

중첩률(IoU)

- 2개의 경계박스가 중첩되는 정도를 나타내는 값
- 정답 경계박스(BB ground truth)와 예측 경계박스 (BB predicted)가 있다고 할때 loU를 계산해서 해당 탐지 결과가 유효한지, 아닌지 결정할 수 있음

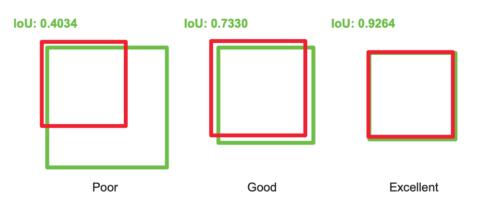




• 객체탐지 알고리즘

중첩률(IoU)

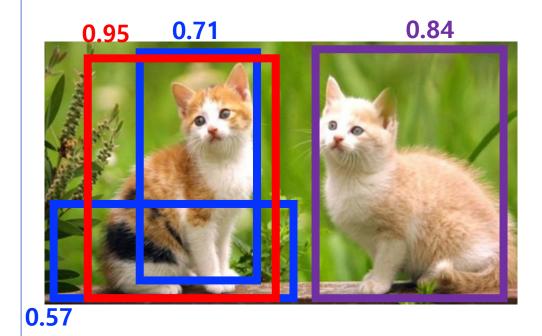
- IoU는 O(두 경계박스가 전혀 겹치지 않음)부터 1(두 경계박스가 완전히 겹침)까지의 값을 가지며 값이 클수록 좋음
- 중첩률은 정확한 예측을 정의하는 용도로 사용됨
 - 중첩률이 설정된 임계값보다 크면 정확한 예측이 됨을 의미함
 - 이 임계값은 상황에 따라 조정 가능하지만 일반적으로 0.5가 많이 사용됨





• 객체탐지 알고리즘

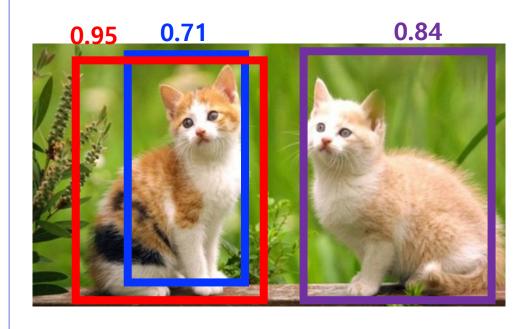
NMS (Non-Maximum Suppression)



- Confidence Score Threshold: 0.6
- IoU Threshold: 0.5

● 객체탐지 알고리즘

NMS (Non-Maximum Suppression)

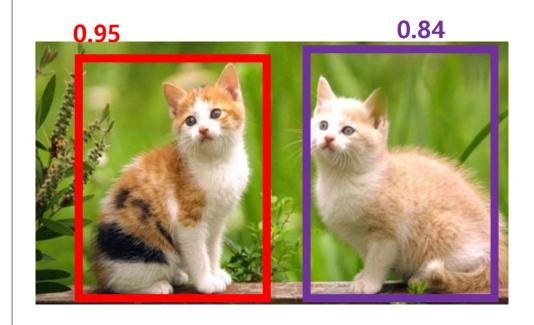


- Confidence Score Threshold: 0.6
- IoU Threshold: 0.5



● 객체탐지 알고리즘

NMS (Non-Maximum Suppression)



- Confidence Score Threshold: 0.6
- IoU Threshold: 0.5



객체탐지/이미지분류

• 객체탐지 알고리즘

mAP (mean-Average Precision)

- mAP를 구하기 위해서는 Precision과 Recall의 개념을 알아야 한다.
- Precision: '모델의 입장'에서 모델이 True라고 예측했을 때 실제로 True일 비율
- Recall: '실제 정답의 입장'에서 실제로 True일 때 모델이 True라고 예측한 비율

		실제 정답	
		Positive	Negative
실험 결과	Positive	True positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

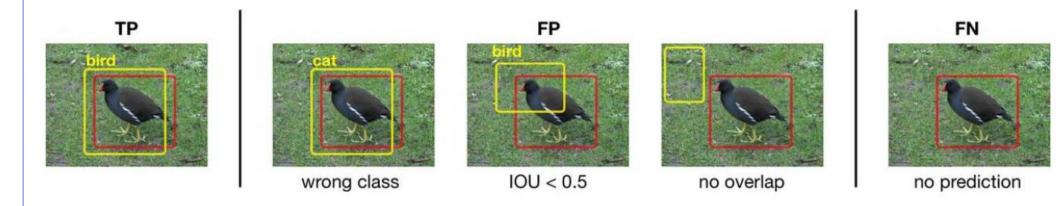


객체탐지/이미지분류

• 객체탐지 알고리즘

mAP (mean-Average Precision)

- Precision: '모델의 입장'에서 모델이 True라고 예측했을 때 실제로 True일 비율
- Recall: '실제 정답의 입장'에서 실제로 True일 때 모델이 True라고 예측한 비율



$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$



• 객체탐지 알고리즘

Precision과 Recall의 Trade-off 관계

- 두 개의 값을 모두 베스트로 끌어올릴 수 없다.
- 둘 중 하나의 값을 끌어올리기에만 몰두하다 보면 보기에는 멀쩡해 보이지만
 알고 보면 심각한 오류를 발견할 수 있다.
- 그렇기에 Precision과 Recall의 균형을 조절하는 것이 중요하다.



• 객체탐지 알고리즘

Precision을 100%로 올리려 한다면

- Precision은 모델이 "사과"라고 예측한 것 중에서 실제로 사과인 것의 비율
- Precision을 100%로 만들기 위해서는 모델이 사과라고 예측한 것 중에 틀린 예측이 없어야 한다.

<u>예시</u>

- 1. 사과 10개, 배 10개가 있다고 가정한다.
- 2. 모델이 사과라고 예측한 것 중 9개가 실제 사과이고, 1개가 배일 경우 Precision은 90%가 된다. (9/10)
- 3. 그러나 Precision을 100%로 만들기 위해서는 모델이 사과라고 예측한 것 중 틀린 예측이 없어야 한다.
- 어떻게 해야 할까? -



객체탐지/이미지분류

• 객체탐지 알고리즘

Precision을 100%로 올리려 한다면

- 모델이 정말 확실한 것만 "사과"라고 예측하고, 나머지는 "사과가 아니다"라고 예측한다.
- 예를 들어, 모델이 확실한 사과 5개만 사과라고 예측하고, 나머지 5개 사과와 10개 배는 모두 "사과가 아니다"라고 예측하면 Precision은 100%가 된다. (5/5)
- 그러나 이렇게 하면 모델이 많은 사과를 놓칠 수 있다. Recall, 즉 모델이 실제 사과를 찾아내는 능력이 낮아질 수 있다. 모델이 사과라고 예측한 것 중 틀린 예 측은 없지만, 실제 사과를 많이 놓쳤기 때문에 좋은 모델이라고 할 수 없다.



객체탐지/이미지분류

• 객체탐지 알고리즘

Recall을 100%로 올리게 된다면

- Recall은 모델이 실제로 존재하는 사과 중에서 얼마나 많은 사과를 찾는 비율
- Recall을 100%로 만들기 위해서는 모델이 실제 사과를 하나도 빠짐없이 찾아내야 한다.

예시

- 1. 사과 10개, 배 10개가 있다고 가정한다.
- 2. 모델이 실제 사과 9개를 사과라고 예측하고, 나머지 1개 실제 사과를 배라고 예측하면 Recall은 90%가 된다. (9/10)
- 3. 그러나 Recall을 100%로 만들기 위해서는 모델이 실제 사과 10개를 완벽하게 사과라고 예측해야 한다.
- 어떻게 해야 할까? -



객체탐지/이미지분류

• 객체탐지 알고리즘

Recall을 100%로 올리려 한다면

- 모델이 모든 과일을 "사과"라고 예측해야 한다.
- 예를 들어 모델이 사과 10개와 배 10개를 모두 사과라고 예측하면 Recall은 100%가 된다. (10/10)
- 그러나 이렇게 하면 모델이 배도 사과로 예측한다. 모델이 실제 사과를 모두 찾아냈지만, 많은 잘못된 예측을 하기 때문에 좋은 모델이라고 할 수 없다.

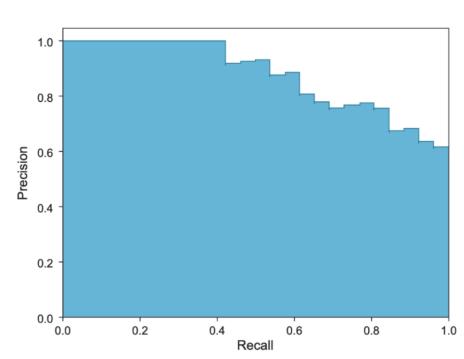


• 객체탐지 알고리즘

Precision-Recall(PR) 곡선

재현율(Recall) =
$$\frac{TP}{TP+FN}$$
 정밀도(Precision) = $\frac{TP}{TP+FP}$

- TP와 FP를 정의했으니 이제 주어진 클래스에 대한 탐지결과의 Precision과 Recall을 계산할 수 있음
- 모든 대상 클래스에 대한 Precision와
 Recall을 계산하고 나면, 다음과 같은 PR
 곡선을 그릴 수 있음
- 매번 예측마다 TP와 FP를 계산해 가면서 모든 예측에 대해 Precision과 Recall의 변화를 계산하여 그래프를 그리는 것





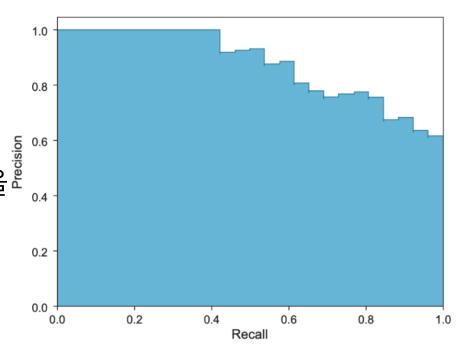
• 객체탐지 알고리즘

Precision-Recall(PR) 곡선

재현율(Recall) =
$$\frac{TP}{TP+FN}$$
 정밀도(Precision) = $\frac{TP}{TP+FP}$

- Recall이 증가할 때에도 Precision이 하락하지 않는다면 임계값을 변화시켜도
Precision이 하락하지 않는다는 의미이므

로 모델의 성능이 뛰어나다는 의미
- 하나의 Class에 대해 Precision과 Recall을
계산해 PR 곡선을 구하고, PR곡선 아래의
면적을 AP(Average Precision)라고 한다.



mAP(mean Average Precision)

: 모든 Class에 대해 PR Curve를 구해 AP를 구하고 이 AP의 평균을 mAP라고 한다.



• 객체탐지 알고리즘

mAP 계산법

- 1) 각 Bounding Box와 Confidence Score를 계산
- 2) Precision과 Recall을 계산
- 3) 각 분류 클래스마다 threshold를 변화시키며 PR곡선을 그림
- 4) PR 곡선의 AUC를 계산하여 각 분류 클래스마다 AP를 계산
- 5) 각 클래스의 AP의 평균(mAP)을 계산



• 객체탐지 알고리즘

Object Detection 모델의 성능을 평가하는 2가지 metric

- <mark>mloU</mark> : 모든 클래스에 대한 각 Object에 그려지는 모든 Bounding Box의 loU를 평균낸 것

- mAP: 모든 클래스에 대한 AP를 평균낸 것





