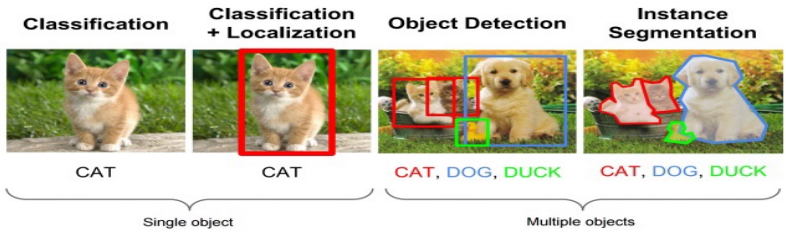
Object Detection



- classification : single object 에 대해서 object의 클래스를 분류하는 문제.
- classification + Localization : single object에 대해서 object의 위치를 bounding box로 찾고 (localization) + 클래스를 분류하는 문제
- object detection : multiobject 에서 각각의 object 에 대해 classfication + localization을 수행
- instance segmentation : object detection 과 유사하지만 다른점은 object의 위치를 bounding box가 아닌 실제 edge로 찾는것

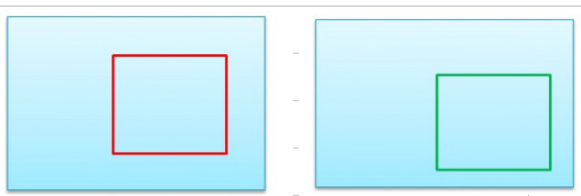
* Localization : 물체가 존재하는 bounding box을 찾아 내는것
주의: 그 안에 있는 물체가 하나로 인식, 고양이 3마리가 있어도 고양이 1, 2, 3이 아닌 고양이로 인식. 1개의 물체만 인식을 때 주로 사용

* bounding box: 네 변이 이미지 상에서 수직/수평 방향을 향한 직사각형 모양의 박스

Object Detection의 평가

학습하고자 하는 데이터의 일부 혹은 실제값

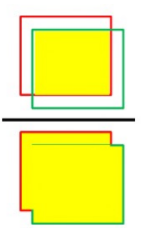
1. IoU(Intersection Over Union)
- 모델이 예측한 결과와 Ground Truth와 얼마나 정확하게 겹치는가를 나타내는 지표
구하는 방법
- 개별 box가서로 겹치는 영역 / 전체 box의 합집합 영역



0.5 이상이면 제대로 검출다!

$$IoU = \frac{area(B_{gt} \cap B_p)}{area(B_{gt} \cup B_p)}$$

*다른값으로 지정할수 있다 (0.5만은...)



IoU 계산

예측된 바운더리박스, ground truth 간 겹치는 부분의 면적 / 합집합 면적

Map(mean Average Precision)

Precison(정밀도)
- 모델이 True라고 예측한 것 중 정답도 True인 것의 비율
Recall(재현율)
- 실제 정답이 True인 것중에서 모델이 True라고 예측한 것의 비율

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

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

TP = True positive
TN = True negative
FP = False positive
FN = False negative

실제 상황 (ground truth)	예측 결과 (predict result)	
	Positive	Negative
Positive	TP(true positive) **옳은 검출**	FN(false negative) **검출되어야 할 것이 검출되지 않았음**
Negative	FP(false positive) **틀린 검출**	TN(true negative) **검출되지 말아야 할 것이 검출되지 않았음**

ex) 이미지에 10 마리의 고양이가 있고, 모델이 6마리의 고양이를 검출.
5개는 TP, 1개는 FP라고 가정
그럼 모델이 검출하지 못한 고양이는 4마리는 FN이 됨.
Precision = 5/6, Recall = 5/10 ⇒ Precision은 좋아보이나... Recall은 글쎄...??
⇒ 어느 한 값으로 알고리즘의 성능을 파악하기는 불가능, 두 값을 종합해서 알고리즘을 평가하기 위한것이 AP이다.

참고로 TP,FP를 결정해주는 것은 IoU임

AP(Average Precison)
- precision recall 그래프에서 그래프 선 아래쪽의 면적
- precision과 recall은 반비례 관계를 가지기 때문에 ap라는 지표를 사용
ap의 계산
- recall을 0부터 0.1 단위로 증가시켜 1까지 (0,0.1,0.2...,1) 로 증가시킬 때 필연적으로 precision이 감소하는데, 각 단위 마다 Precision 값을 계산하여 평균을 내어 계산한다. 즉 11가지의 recall값에 따른 precision 값들의 평균 AP를 의미하며, 하나의 class마다 하나의 AP값을 계산할 수 있다.

mAP
- 전체 class에 대해 AP를 계산하여 평균을 낸 값

1-stage detector VS 2-stage detector

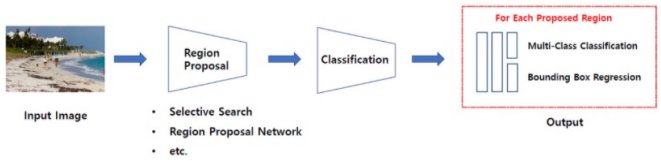
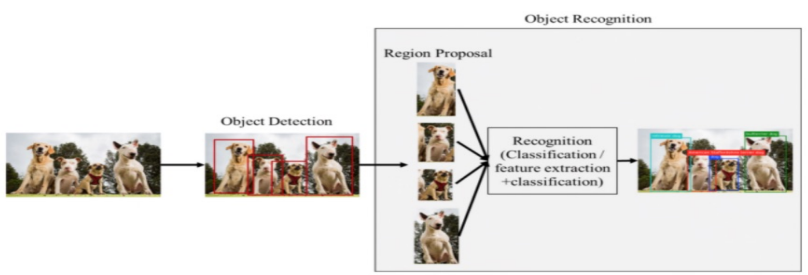


그림5. 2-stage Detector의 전체적인 구조 (출처 : hoya012 .github.io)

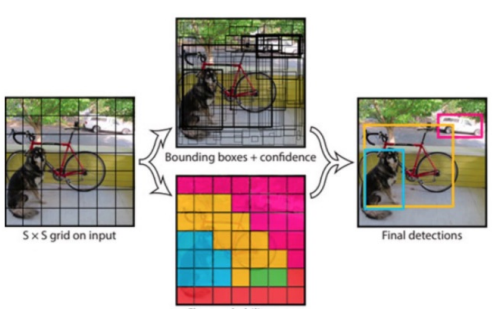
그림6. 1-stage Detector의 전체적인 구조 (출처 : hoya012 .github.io)

2-stage detector



Selecte search, region proposal network 와 같은 알고리즘 및 네트워크를 통해 object가 있을만한 영역을 우선 뽑아냄.
이 영역을 RoI(region of interest)라고 한다.
이런 영역을 우선 뽑아내고 나면 각 영역들을 convolutuon network를 통해 classification, box regression(localization)을 수행.

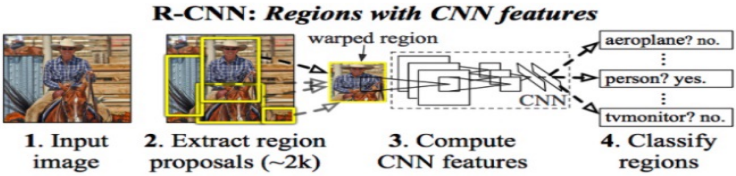
1-stage detector



1-stage는 2-stage와 다르게 RoI영역을
먼저 추출하지 않고 전체 이미지에서 classification, box regression(localization)을 수행
특정 object를 담고 있는 RoI에서 classification, localization을 수행하는 것보다 여러 object가 섞여 있는 전체 image에서 이를 수행하는게 더 정확도는 떨어진다.
하지만 간단하고 쉬운만큼 속도가빠른 장점이 있다.

* 1보다 2가 비교적 느리지만 정확도가 높다.

R-CNN



R-CNN은 image classfiacation을 수행하는 CNN과 localization을 위한 regional proposal알고리즘을 연결한 모델.

수행과정
1. 이미지를 입력받는다.(inpuut)
2. Selective search알고리즘에 의해 regional proposal output 약 2000개를 추출한다.
추출한 regional proposal output을 모두 동일 사이즈인(224 * 224)로 찌그러트린다.(이때 박스의 비율은 고려 x)
    * 동일사이즈로 만들어 주는 이유: convolution layer에는 input size가 동일하지 않는데, 마지막 FC layer에서의 input size는 고정이므로 convolution layer에 대한 input size도 동일해야 한다.
    그래서 입력에서 부터 동일한 사이즈로 넣어주어서 아웃풋 사이즈도 동일하게 하는것
3. 미리 이지미넷 데이터를 통해 학습시켜놓은 CNN을 통과시켜 feature vector를뽑아냄
4. 이 추출된 벡터를 가지고 각각의 클래스 마다 학습시켜놓은 SVM classifier를 통과
5. 바운딩 박스 리그레이션을 적용하여 박스의 위치를 조정

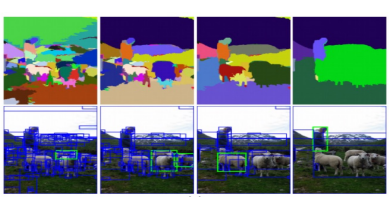
과정을 수행하기 위한 단계
1. Region Proposal
- 주어진 이미지에서 물체가 있을법한 위치를 찾는 것
2. CNN : 각각의 영역으로부터 고정된 크기의 Feature Vector를 뽑아낸다.
3. SVN: classfication 을 위한 선형 지도학습 모델

1. Regoin proposal
- 기존의 sliding window방식의 비효율성을 극복
* sliding window

<Sliding Window> 좌 : 모든 영역에 대해 탐색 / 우 : 크기와 비율을 변형

이미지에서 물체를 찾기 위해 windows(크기,비율)을 임의로 바꿔가면서 모든 영역에 대해 탐색
이렇게 임의로 모든 영역을 탐색하는 것은 너무 느리기 때문에 R-CNN에서는 이를 극복하기 위해 Selective search를 사용한다.

Selective search

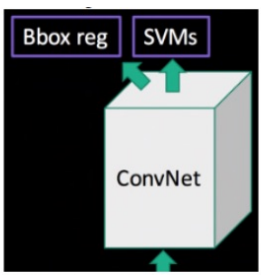


초기 후보 영역을 다양한 크기와 비율로 생성
모든 영역에 대해 유사도를 계산한 후 유사한 영역과 근접한 pixel을 점점 Merge, grouping을 한다.
작업을 반복하여 2000개의 regoin proposal을 생성
생성되면 모두cnn에 넣기전에 같은 사이즈로 warp해준다.

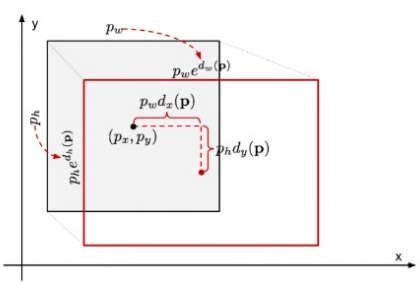
2. CNN
Warp작업을 통해 모두 244 * 244사이즈가 되면 CNN모델에 넣는다.
최종적으로 cnn을 각각 거쳐 region proposal로부터 4906-dementional feature vector를 뽑아내고,
이를 통해 고정길이 feature vector를 만들어 낸다.

3. SVM
Cnn모델로 부터 feature가 추출되면 liner svm을 통해 classfication을 진행
Classfication로 softmax를 사용하지 않고 svm을 사용한 것은 svm이 더 좋은 성능을 나타내었기 때문,,,
Svm은 cnn으로 부터 각각의 feature vector들의 점수를 class 별로 메기고, 객체인지 아닌지, 객체라면 어떤 객체인지 등
판별하는 역할을 하는 classifier임

3-1. Bounding Box Regression -보완 방법
- selective search만 이용하니 localization성능이 안좋았ㅇ음,,,
그래서 물체를 정확하게 감싸도록 조정해주는 선형회귀모델(bounding box regression) 을 넣음



R-CNN 구조 (3-1. Bounding Box Regression)



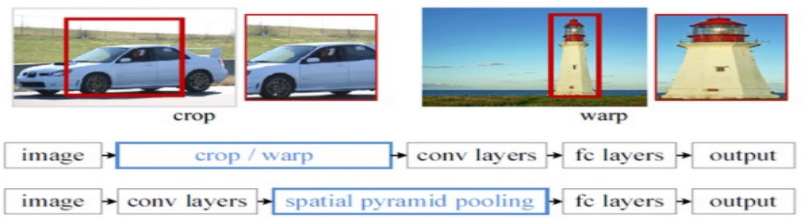
실제 회색 박스로 예측했을 때, 정답과 가까운 빨간색 박스로 보완해주기 위해서 사용,,,

단점

1. 오래걸린다
- selecives search에 해당하는 region proposal만큼 cnn을 돌려야하고 큰 저장공간 연장,, 그래서 느림,,쒸익쒸익
2. 복잡함
- 학습이 세단계의 multi-stage로 구성,,

SPPnet

- R-CNN의 속도 저하의 원인인 region proposal마다 CNN feature map생성을 보완
구조를 region proposal에 바로 cnn을 적용하는 것이 아니라 우선 이미지에 cnn을 적용하여 생성한 feature map을 Region proposal에 사용



SPPnet은 spatial pyramid pooling이라는 특징을 가지는 구조를 활용하여 임의 사이즈의 이미지를 모두 활용 할 수 도록함
SPP layer은 이미지의 사이즈와 상관없이 특징을 잘 반영할 수 있도록 여러 크기의 bin을 만들고 그 bin값을 활용하는 구조
속도 향상과 고정된 이미지만을 필요로 하지 않은 장점을 가지게 되었음,

하지만,, 이것도 단점이 있었구,,,
그래서 R-CNN, SPPnet의 장점을 가져오고 단점을 보완하고자 한것이 Fase R -CNN,,,