

■ Semantic Segmentation

⇒ 모든 픽셀에 대해 픽셀이 어떤 클래스인지 분류한다

⇒ 기존 detection 과는 픽셀을 분류 하고 싶다 → 3가지 레키

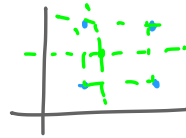
- 1. Convolutionalization
- 2. Upsampling (deconvolution)
- 3. skip connection architecture

1. 기존에 flatten 해서 1D 벡터를 soft max에 이용했다,
하지만 기존 FC 레이어 이거나 1개당 detection 이고, 여러 레이어는
여러 번 필요! $\Rightarrow \therefore$ FC layer의 1D를 제외하고 모두 Conv layer
만 사용!

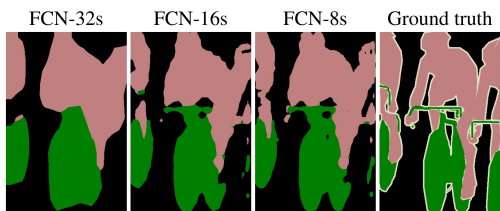
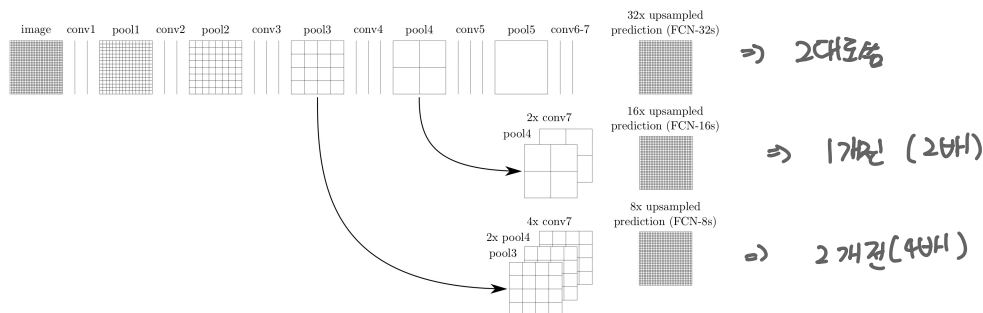
2. Pooling layer로 feature size 줄여줄만큼 다시 키워준다.

(Pooling 줄일 size 줄어 parameter \downarrow , compu \downarrow
그래서 같은 크기 conv 사용한다고 줄였다 \Rightarrow 된다.

⇒ bilinear interpolation 사용



3. => size 줄였다 키워면 (보통은 쓰더라도) 그림 뚫고 그려서 손으로
이름 해줘야해 이런 pooling 단계에서 생기는 Layer 활용 리플



⇒ FCN-8s 가 Ground truth에 가까운

⇒ [Coarse => (거친) , 굵직한 분류 (개, 자동차, 비행기)
Fine => (미세한) , 세세한 분류 (포대, 패기, 비둘, 날도...)]

⇒ patch => 이미지에서 object는 그대로 두고 background 제거한것
patch wise training => 관심이 object 안에 computation을 back 처리 하면
오래걸리니 이를 제거하고 리플

비교할때 PCNN에 왜? (object detect 할?) => 객체 나온 거라 그와 비슷함만큼
비교 (비교대상 없었음)

질문? FCN-4s 한것보다 Computational time 때문?

↳ 너무이런 feature는 만들 특징 잘 반영
안되고 이상하게 나올