

FCN(fully convolution network)에 관하여

출처: <https://wikidocs.net/147359>

def: 이미지 분류에서 우수한 성능을 보인 CNN기반의 모델(AlexNet, VGG16, GoogLeNet)을 semantic seg. task(여러 클래스로 픽셀 분류하는 task)를 수행할 수 있도록 변형한 모델

전이 학습으로 구현함

이후 나온 segmentation seg. task들은 주로 fcn을 기반으로 만들어짐.

이미지 분류(image classification)

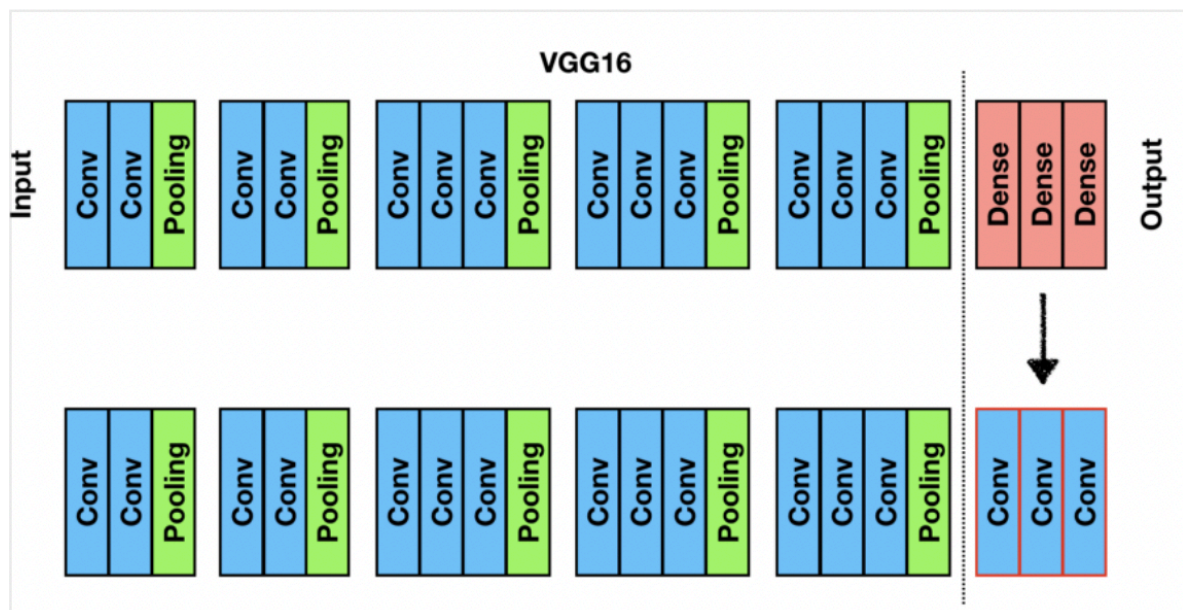
이미지 내의 모든 픽셀에서 피쳐 추출하고,

추출한 피쳐들을 classifier에 넣어 전체(total) 입력 이미지의 클래스를 예측

이미지 분할(Image seg.)

이미지를 이루는 모든 픽셀들의 class를 예측하는 task임.

이미지 분류에서 쓰인 pretrain 네트워크에서 feature extraction 레이어는 재사용하여 feature를 추출하고 fully connected layer(dense layer)를 버리고 1*1 conv와 up-scaling(transpose convolution)으로 변경하여 (fine tuning) 픽셀 클래스 분류의 이미지랑 같은 크기를 아웃풋으로 하도록 함.



-> upscaling

Fully connected layer(FC layer) 이란?(= dense layer)

CNN에서 레이어는 크게 2가지 유형으로 나뉜다.

- Convolution/pooling layer: 이미지를 (filter 통해) 분할하고 분석
- FC layer: 이미지를 분류하고 설명하는 데 가장 적합하게 예측을 하는 층

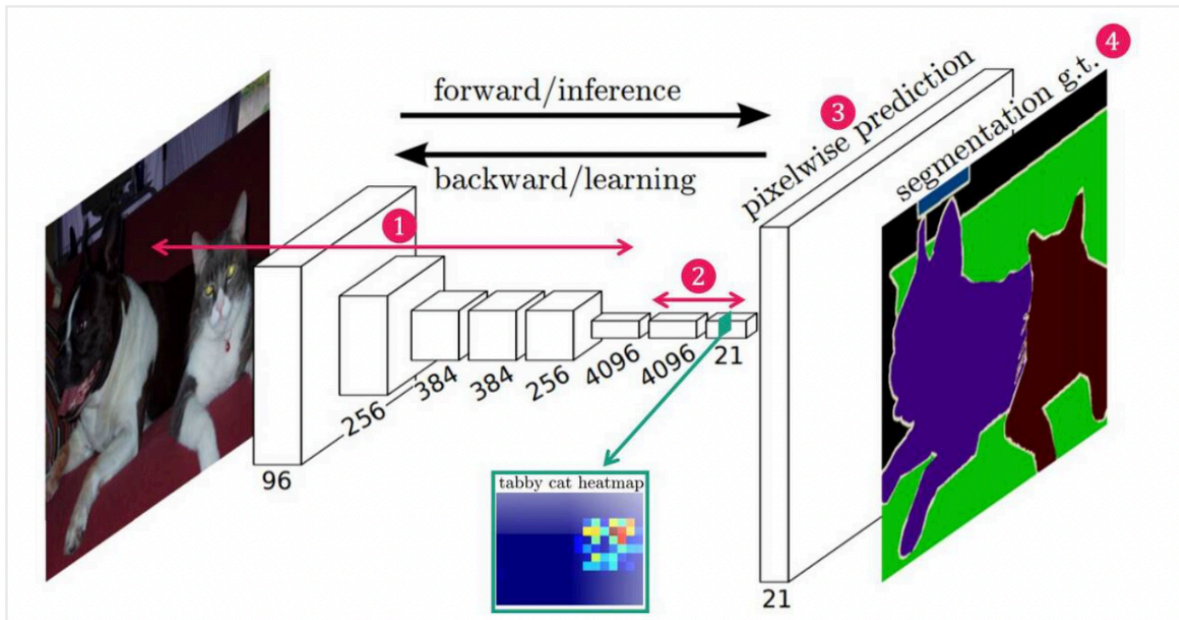
이중 FC layer는 다음과 같은 역할을 한다.

- flattening: 평탄화, 이전 레이어의 출력을 평탄화하여 다음 레이어의 입력이 될 수 있는 단일 벡터로 변환함
- 먼저 입력을 취하여 라벨을 예측하기 위해 가중치를 적용하고, 각 라벨에 대한 최종 확률을

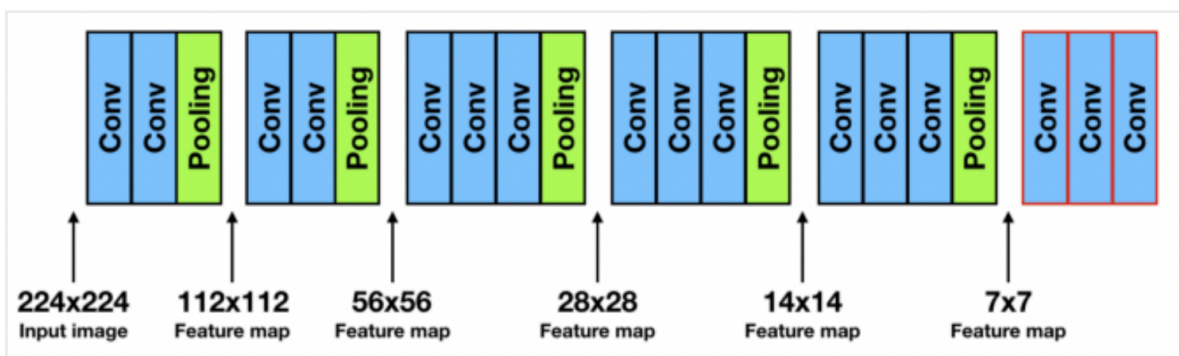
제공함.

- 즉 convolution/pooling 프로세스의 결과를 취하여 이미지를 라벨로 분류하는데 사용함.(최종 분류 레이어)

네트워크 구조

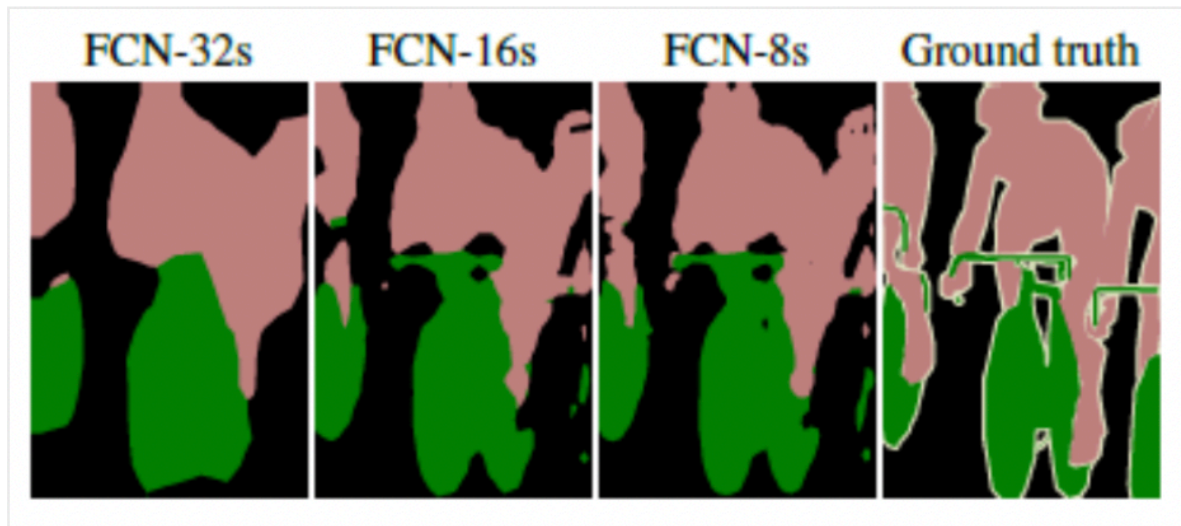


1. Convolution layer로 feature 추출
2. 1*1 convolution layer를 이용하여 feature map의 채널수를 데이터셋의 객체수(# of instance)와 동일하게 변경 (class presence heat map 추출)
3. Up-sampling: 낮은 해상도의 heat map을 upsampling(=transposed convolution) 하고, 입력 이미지와 같은 크기의 map 생성
4. 최종 피쳐 맵과 라벨 피쳐맵의 차이를 이용하여 네트워크 학습



마지막 7*7 피쳐 맵의 한 픽셀은 기존 인풋 이미지의 32*32 pixel을 대표하게 되는데, 이를 업샘플링하는 과정에서 기존 이미지의 정보가 손실될 수 있다.(뭉그러짐 문제) 따라서 업샘플링을 할 때 피쳐 추출 단계의 피쳐맵도 업샘플링에 포함(sum)하여 정보 손실을 막는 방법을 사용함.(skip architecture, 이는 파라미터의 큰 증가 없이 성능을 비약적으로 향상시킴)

어느 단계의 피쳐추출단계까지를 반영하느냐에 따라 정확도가 달라진다.(초기의 피쳐추출단계부터 고려할수록 img. Seg. 정확도가 높아진다.)



이미지 분할을 하는 fcnn은 이미지 분류 모델을 사용하기에, 이미지 분류 모델이 발달할수록, 이미지 분할 모델의 성능이 향상됨.(이미지 분류 네트워크에서 추출하는 피쳐를 이미지 분할에서도 수정없이 사용가능하기 때문)

FC layer를 삭제함으로써 이미지 분할을 구현함.

FC layer를 이용하여 이미지 분할을 하는 것은 적합하지 않음.

1. FC layer를 통과하고 나면 이미지의 위치 정보가 사라지기 때문
(컨볼루션 레이어는 위치정보가 저장되는 반면 FC layer는 위치 정보가 사라짐)
2. FC layer는 고정된 크기의 input image만 받을 수 있기 때문
(Dense layer에 가중치 개수가 고정되어 있기 때문에 바로 앞 레이어의 Feature Map의 크기도 고정됨)
3. FC layer는 파라미터 개수를 너무 많이 필요로 함
(fully connected 이기 때문)

이런 이유로 FC layer를 삭제하고 1*1 convolution으로 대체함.

FCN에서는 deconvolution을 사용하는가 아님 transposed convolution을 사용하는가?

같은 개념으로 사용하는 경향이 있는데, convolution의 역연산을 deconvolution이라고 하는데, 그 역연산 안에 transposed convolution이 포함되는 듯

<https://gaussian37.github.io/vision-segmentation-fcn/> fcnn pytorch로 구현

사용할 하이퍼파라미터(?, 손 대볼곳)

- 인코더와 디코더 구조(얼마나 복잡하게 만들것인지, 무슨 방식 사용할 것인지)
- CNN 기반 모델 뭘 채택할지 (CNN의 파라미터들)
- Skip architecture()
- Data augmentation(데이터에 따른 augmentation 방식)
-