

# 2021 인공지능 소수전공

## 71차시: Segmentation

2021.08.09 17:30~18:15

Seokhwan Yang

- **Classification (영상분류)**
- **Object Detection (객체 탐지)**
  - 얼굴 인식, 물체 인식 등
  - Object Tracking (객체 추적) 포함
- **Image Segmentation (영상 분할)**
  - Semantic Segmentation, Instance Segmentation 등
- **Gesture & Motion Recognition (제스처, 동작인식)**
- **Image Captioning 등 기타...**

## • 대표적인 영역과 알고리즘들

### • Object Detection

- CNN / RCNN / Fast RCNN / Faster RCNN
- SPP Net
- YOLO
- SDD
- Attention Net
- (Modern CNN): LeNet / AlexNet / VGG Nets / GoogLeNet / ResNet 등

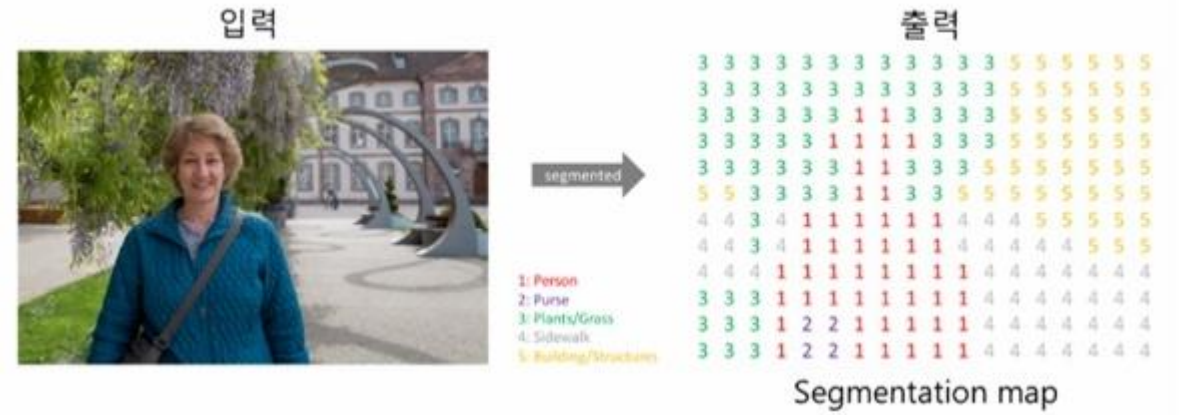
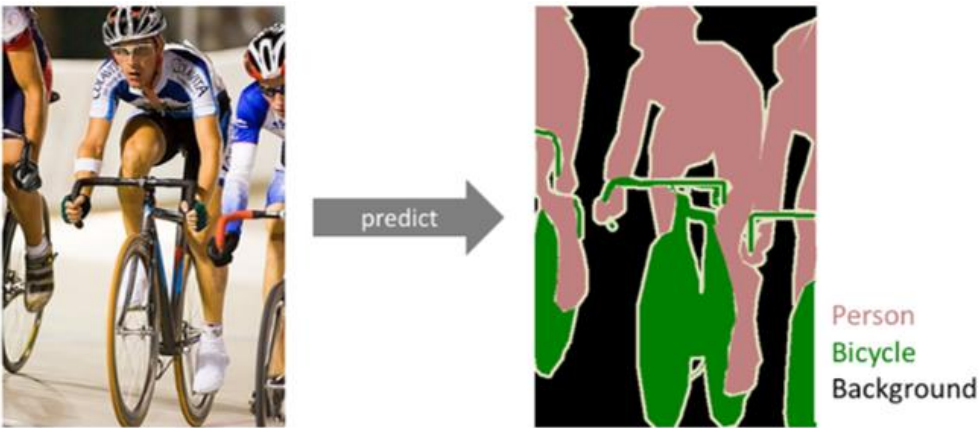
### • Semantic Segmentation

- FCN
- DeepLab v1, v2
- U-Net
- ReSeg

- Semantic Segmentation

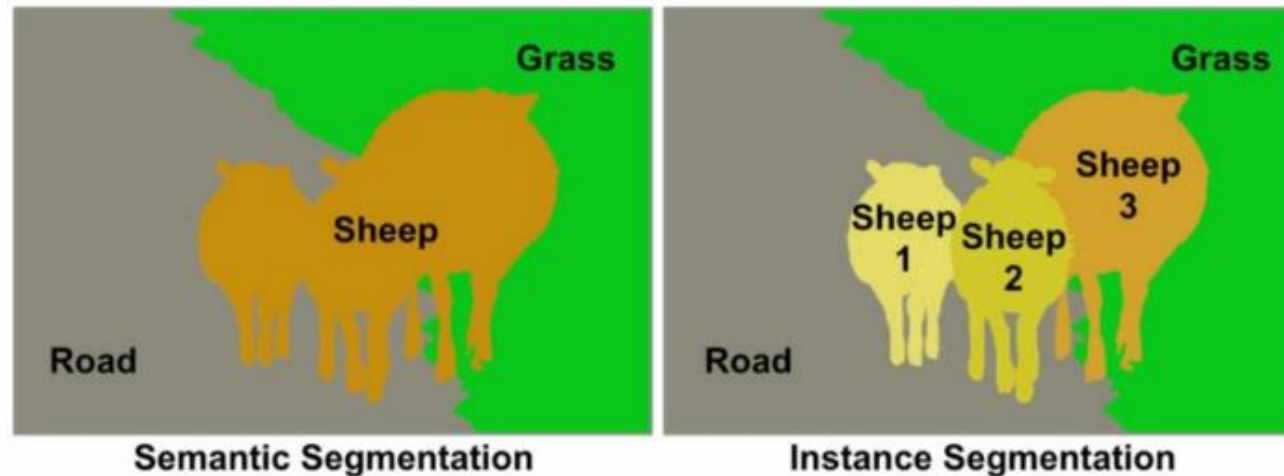
그래서 Dense Prediction이라고도 부름

- 이미지에 포함된 **모든 픽셀에 대한 레이블**을 예측하는 기술  
→ 이미지 내의 객체들을 의미 있는 단위로 분할
- 딥 러닝에서의 영상분할은 일반적으로 Semantic Segmentation을 말함

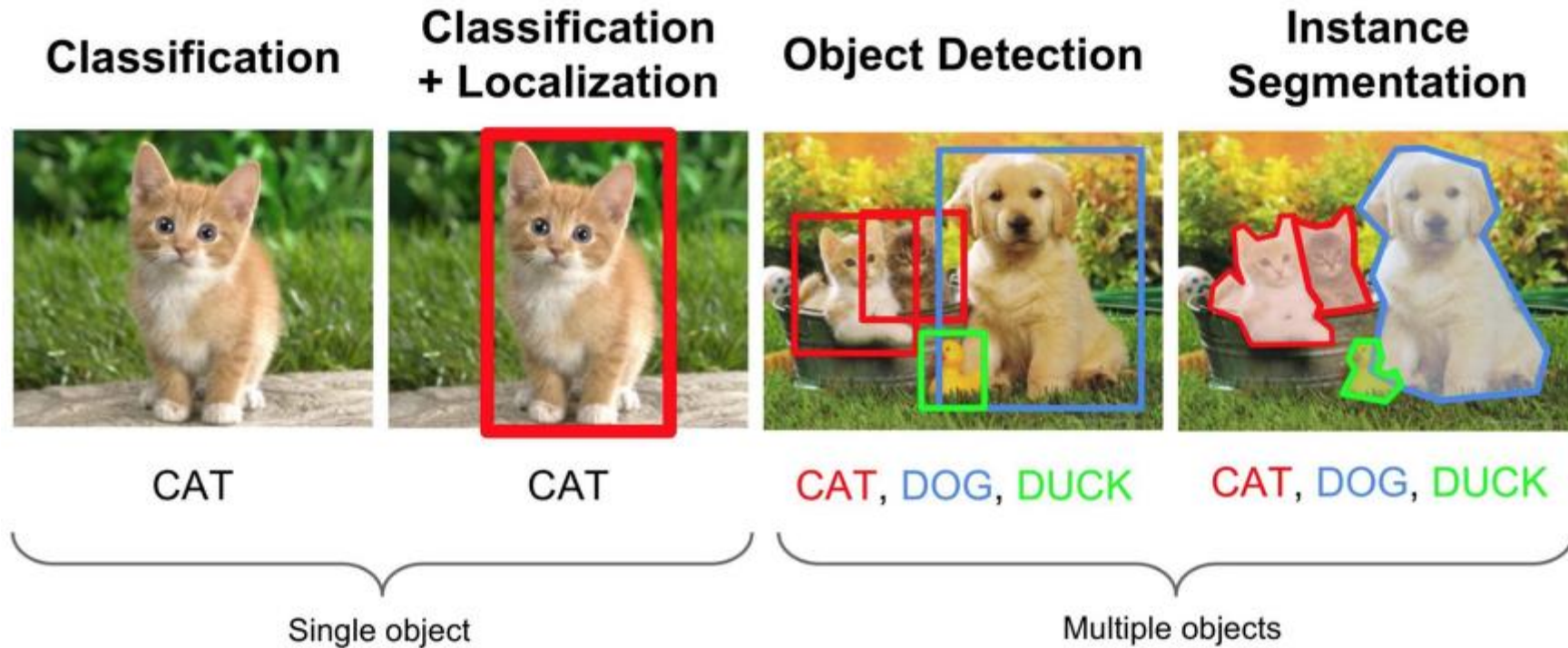


- Instance Segmentation

- Semantic Segmentation보다 더 엄격한 기준을 적용한 기술
- Semantic Segmentation과 Instance Segmentation의 비교
  - Semantic Segmentation: 같은 클래스의 인스턴스를 구분하지 않음
  - Instance Segmentation: 각각의 인스턴스를 구분함



# 영상인식 기술의 차이점 비교



- Semantic Segmentation의 목적

- 사람의 눈으로 이미지를 의미에 따라 분할하는 것은 어렵지 않은 작업
- 그러나 컴퓨터, 기계에게는 결코 쉽지 않은 작업
  - 딥러닝 기술의 발달 이전에는 거의 불가능에 가까운 작업이었음
- Semantic Segmentation을 이용하면 이미지에 포함된 각 픽셀이 어느 클래스에 속하는지 판단할 수 있음
- Semantic Segmentation은 이미지의 각 픽셀이 예측된 클래스를 나타내는 Semantic Map을 얻는 것이 실질적인 목적임



# Semantic Segmentation

입력



segmented

1: Person  
2: Purse  
3: Plants/Grass  
4: Sidewalk  
5: Building/Structures

출력



Segmentation map

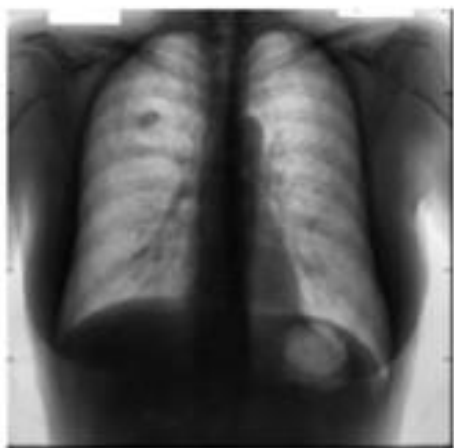


- 적용 분야

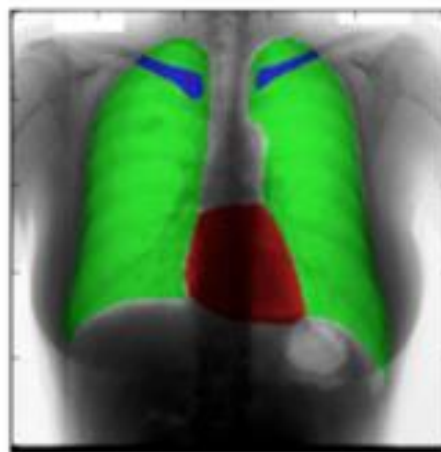
- Semantic Segmentation의 대표적인 적용 분야는 자율 주행 자동차



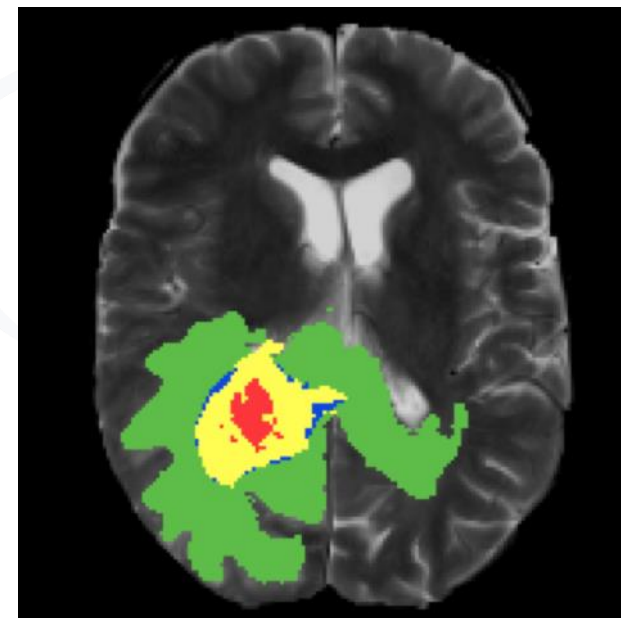
- 적용 분야
  - 의료 영상 분석



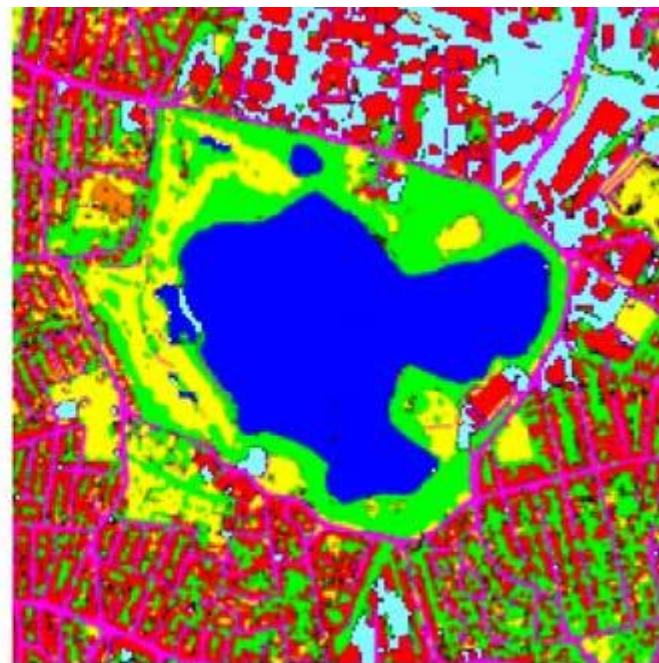
Input Image



Segmented Image



- 적용 분야
  - 위성 이미지 분석



- 대표적인 알고리즘

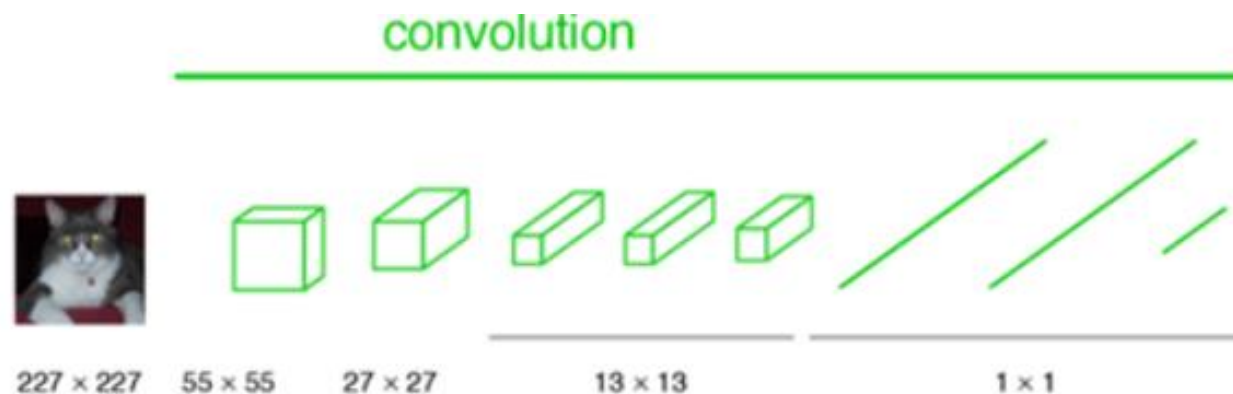
- FCN (Fully Convolutional Networks)

- CNN 알고리즘의 공간정보 소실 문제를 해결하기 위하여 개발됨
    - CNN 계열의 알고리즘은
      - 이미지에 있는 물체가 어떤 클래스에 속하는지 예측할 수 있지만
      - 그 물체가 어디에 존재하는지는 예측할 수 없음
      - 이유: CNN 모델의 마지막 부분에서 사용된 Fully Connected Layer에서 이미지의 모든 위치정보가 소실되기 때문

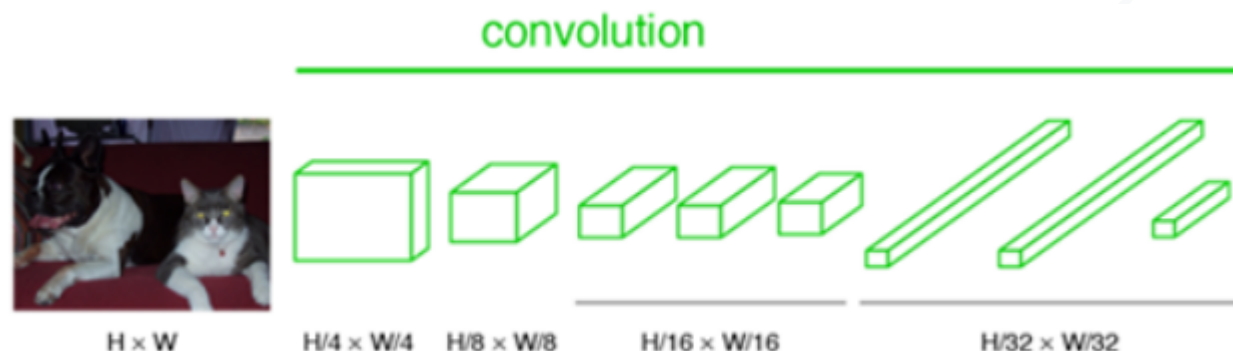
- FCN 모델의 경우
  - 위치정보가 소실되지 않게 하기 위해서
  - 또한 어떠한 크기의 입력 이미지도 허용하기 위해서
  - 고정된 크기의 인풋만을 허용하는 Fully Connected 층을 1x1 Convolution Layer로 바꿔 줌
- 결국 모든 레이어가 Convolution Layer로 구성되었고
- Fully Connected Layer의 충돌이 사라졌으므로 입력 이미지의 크기에 제한을 받지 않음

# FCN (Fully Convolutional Networks)

- Fully Connected Layers들을 1x1 크기의 Convolution Layer로 전환



- 입력 이미지의 크기에 대하여 자유로워 짐

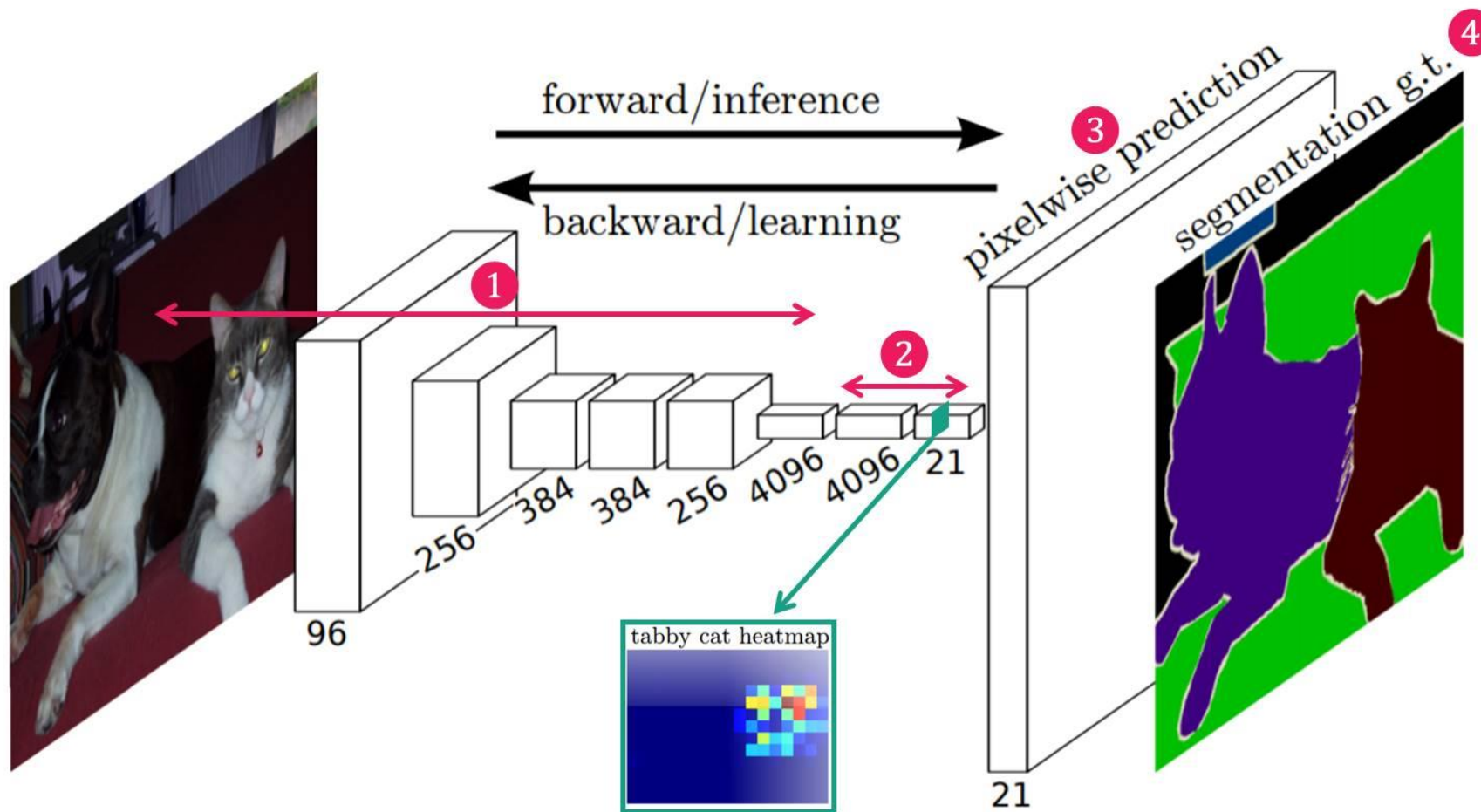




- 알고리즘의 특징

- Convolution Layer들을 거치고 나서 얻게 된 마지막 특성 맵의 개수는 훈련된 클래스의 개수와 동일함
  - 21개의 클래스로 훈련된 네트워크라면 21개의 특성 맵(=heatmap)을 산출
  - 각 특성 맵은 하나의 클래스를 대표함
  - 만약 고양이 클래스에 대한 특성 맵이라면 고양이가 있는 위치의 픽셀 값들이 높고, 강아지 클래스에 대한 특성 맵이라면 강아지 위치의 픽셀 값들이 높게 나타남
- 이와 같은 대략적인(coarse) 특성 맵들의 크기를 원래 이미지의 크기로 다시 복원해 줄 필요가 있음

# FCN (Fully Convolutional Networks)



Hitmap 설명

# FCN (Fully Convolutional Networks)

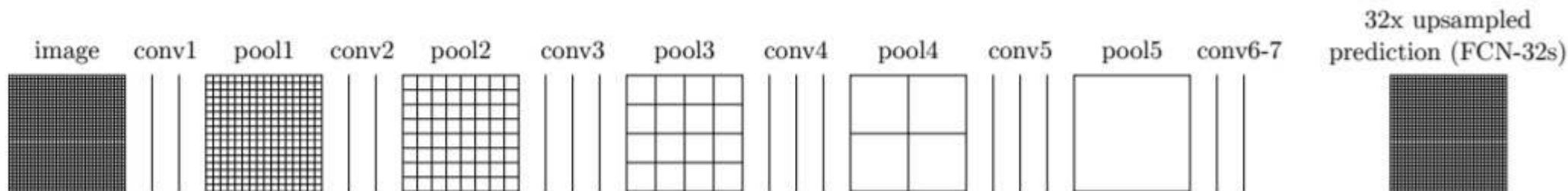


- 이미지의 모든 픽셀에 대해서 클래스를 예측하는 Dense Prediction을 해주는 것이 Semantic Segmentation의 목적임(Coarse와 Dense는 서로 반의어)
- 원래 이미지 크기로 복원하는 과정을 Up-Sampling이라고 부름
- Up-Sampling을 통해 각 클래스에 해당하는 Coarse한 특성 맵들을 원래 사이즈로 크기 조절

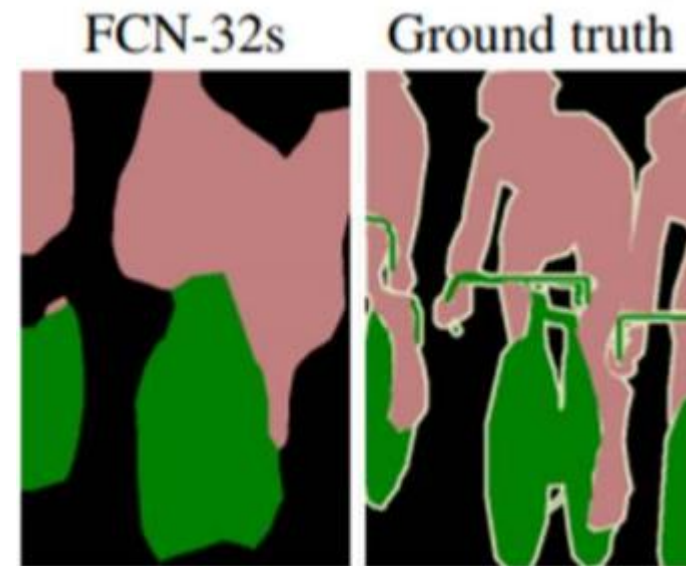
- Up-Sampling된 특성 맵들을 종합해서 최종적인 Segmentation Map 생성  
→ 즉, 각 픽셀당 확률이 가장 높은 클래스를 선정해 주는 것
  - 만약 (1, 1) 픽셀에 해당하는 클래스 당 확률 값이
  - 강아지 0.45, 고양이 0.94, 나무 0.02, 컴퓨터 0.05, 호랑이 0.21 와 같은 식이라면
  - 0.94로 가장 높은 확률을 산출한 고양이 클래스를 (1, 1) 픽셀의 클래스로 예측하는 것
  - 이런 방법으로 모든 픽셀이 어느 클래스에 속하는지 판단함

# FCN (Fully Convolutional Networks)

## • FCN-32s 모델



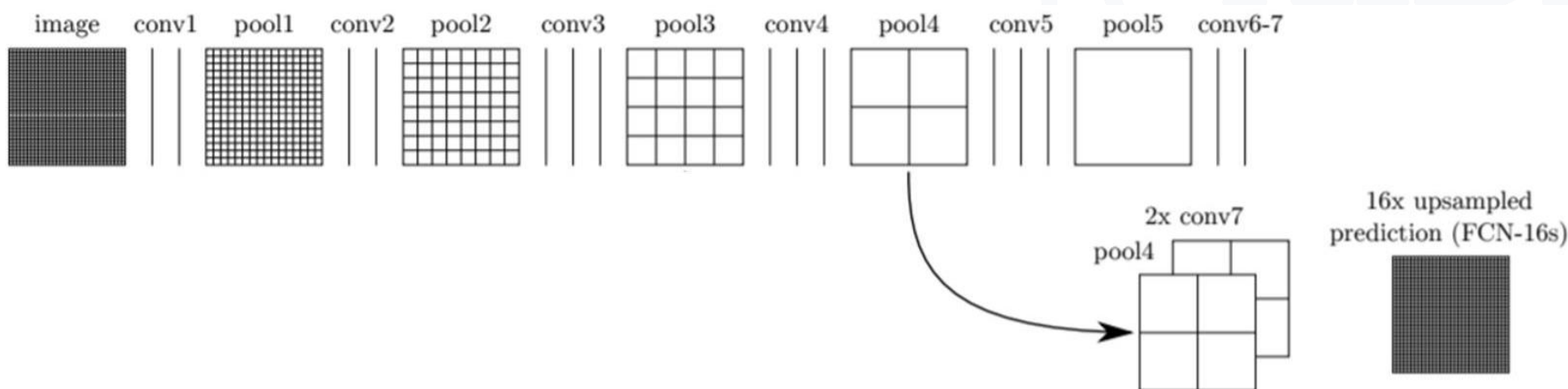
- 단순히 Up-Sampling을 수행할 경우
  - 특성 맵의 크기는 원래 이미지 크기로 복원됨  
→ 복원된 특성 맵으로부터 원래 이미지 크기의 Segmentation Map 확보 가능
  - 그러나 상세하지 못한 대략적인 Segmentation Map
  - 단계적으로 1/32 만큼 줄어든 특성 맵을 한 번에 32배로 Up-Sampling한 결과



# FCN (Fully Convolutional Networks)

- FCN-16s

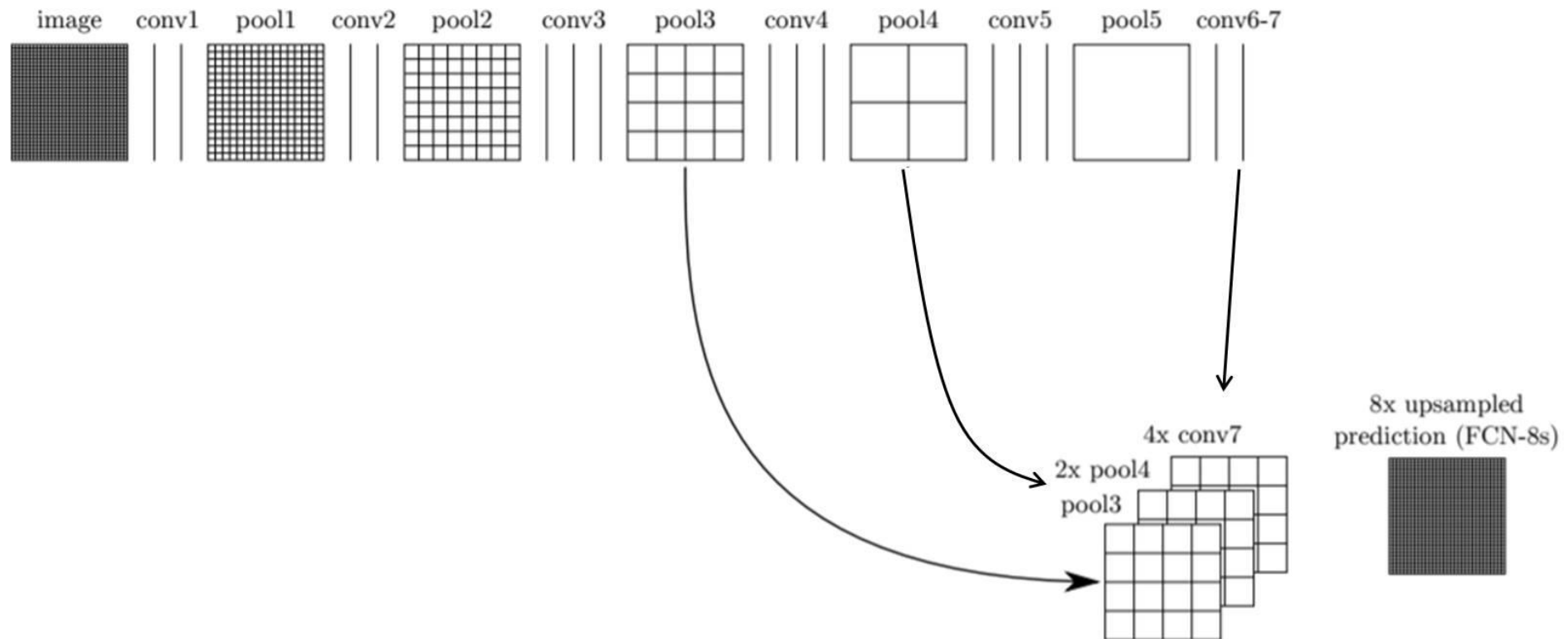
- FCN-32s보다 상세한 Segmentation Map을 얻기 위하여 FCN-32s를 개선한 모델  
→ Skip Combining 기법 제안 및 적용
- “컨볼루션 + 풀링”으로 구성된 이전 단계의 Convolution Layer들의 특성 맵을 참고하여 Up-Sampling 작업을 수행하면 더욱 정확도를 높일 수 있지 않을까? 라는 제안





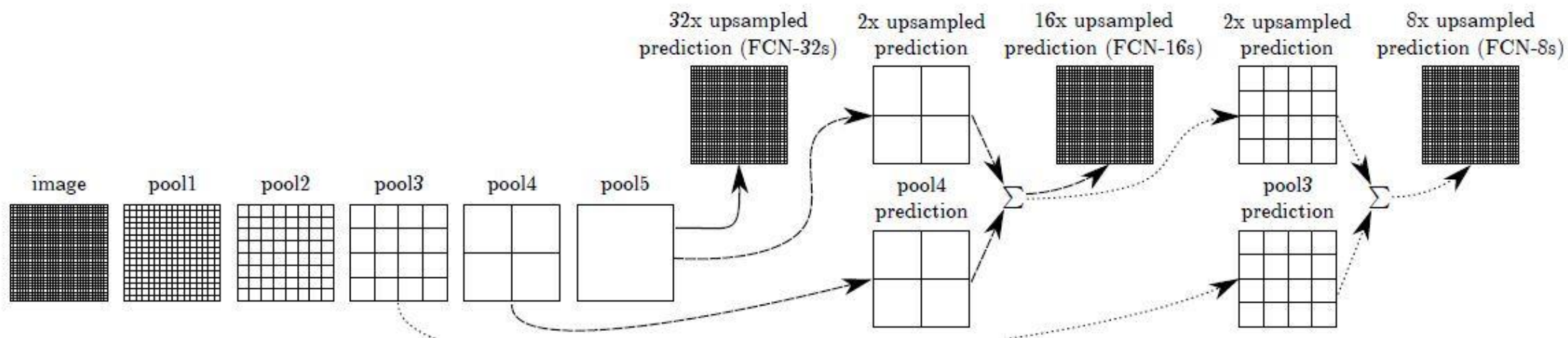
# FCN (Fully Convolutional Networks)

- FCN-8s



# FCN (Fully Convolutional Networks)

- FCN-8s



# FCN (Fully Convolutional Networks)

- FCN-32s vs FCN-16s vs FCN-8s

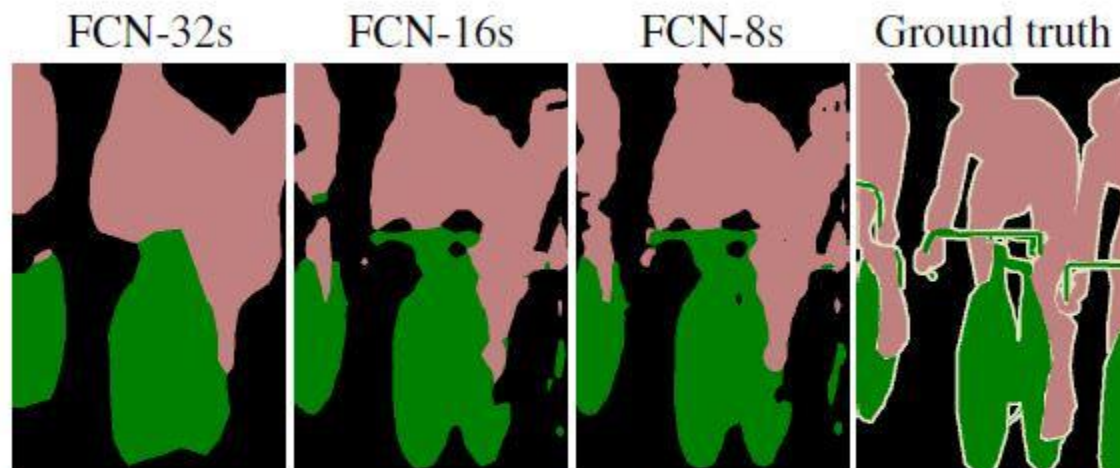


Figure 4. Refining fully convolutional nets by fusing information from layers with different strides improves segmentation detail. The first three images show the output from our 32, 16, and 8 pixel stride nets (see Figure 3).

출처: Jonathan Long 등, "Fully Convolutional Networks for Semantic Segmentation", CVPR 2015  
(FCN의 original 논문)

- SegNet

- Encoder Decode 프레임을 채택한 모델
- FCN-16s, FCN-8s 모델에서 볼 수 있는 건너뛰는 연결 방식이 없음
- FCN과 달리 Up-Sampling에서 학습 가능한 매개 변수가 사용되지 않음

## Max Pooling

Remember which element was max!

1	2	6	3
3	5	2	1
1	2	2	1
7	3	4	8

Input: 4 x 4



5	6
7	8

Output: 2 x 2



Rest of the network

## Max Unpooling

Use positions from pooling layer

1	2
3	4

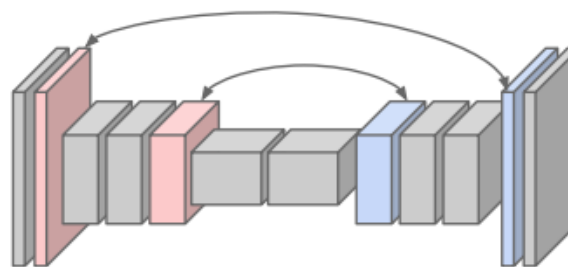
Input: 2 x 2



0	0	2	0
0	1	0	0
0	0	0	0
3	0	0	4

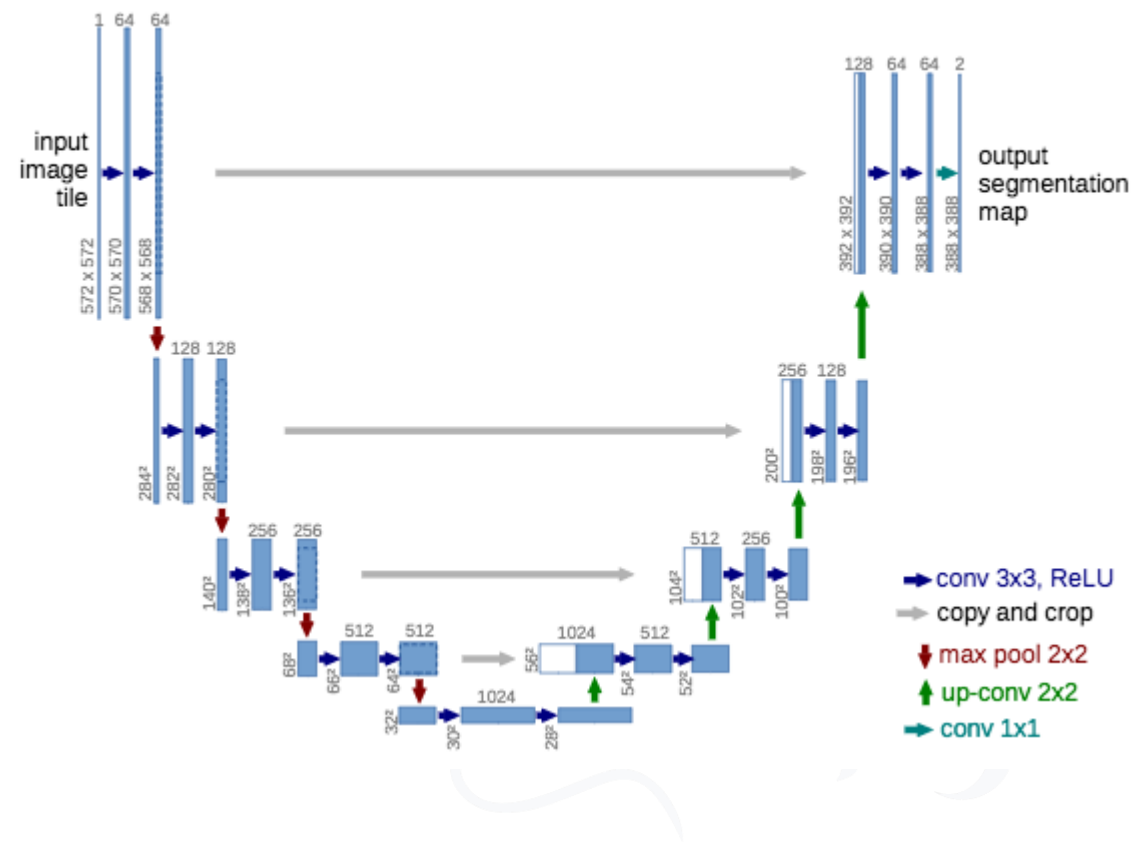
Output: 4 x 4

Corresponding pairs of  
downsampling and  
upsampling layers



## • U-Net

- 저차원, 고차원의 정보를 모두 사용하여 이미지의 특징을 추출하는 신경망
- SegNet과 마찬가지로 서로 대칭인 Encoder-Decoder 프레임워크 채택
- FCN-16s, FCN-8s와 같이 건너뛰는 연결을 가지고 있음





## • U-Net의 활용

- U-Net은 복호화 시 모든 부호화 계층의 출력을 다시 사용하므로
  - 이미지의 외곽선 추출, 객체 검출(탐지) 등 이미지를 특정 형태로 변형하는 분야에 적합
- 부호화 단계에서 복호화 단계로 정보가 전달될 때,
  - 부호화 마지막 계층의 정보만 복호화 첫 계층으로 전달되는 것이 아니라
  - 부호화 단계의 각 계층의 결과가 해당 복호화 단계의 각 계층으로 전달됨
  - 이는 차원 축소로 인한 공간 정보의 손실을 방지해 줌
    - 분류와 다른, 이미지 처리에 매우 중요한 요소로 작용
  - 공간 정보 손실 방지 → 이미지 변형 시 올바른 위치에 원하는 형태의 이미지 배치 가능

- Reference

- [A Beginner's guide to Deep Learning based Semantic Segmentation using Keras | Divam Gupta \(https://divamgupta.com/image-segmentation/2019/06/06/deep-learning-semantic-segmentation-keras.html\)](https://divamgupta.com/image-segmentation/2019/06/06/deep-learning-semantic-segmentation-keras.html)
  - [한글번역 및 정리: semantic segmentation의 목적과 대표 알고리즘 FCN의 원리 by bskyvision\(https://bskyvision.com/491\)](https://bskyvision.com/491)
- [CNN을 활용한 주요 Model - \(4\) : Semantic Segmentation \(renew.github.io/18/\)](https://renew.github.io/18/)