[11주차] Detection and Segmentation

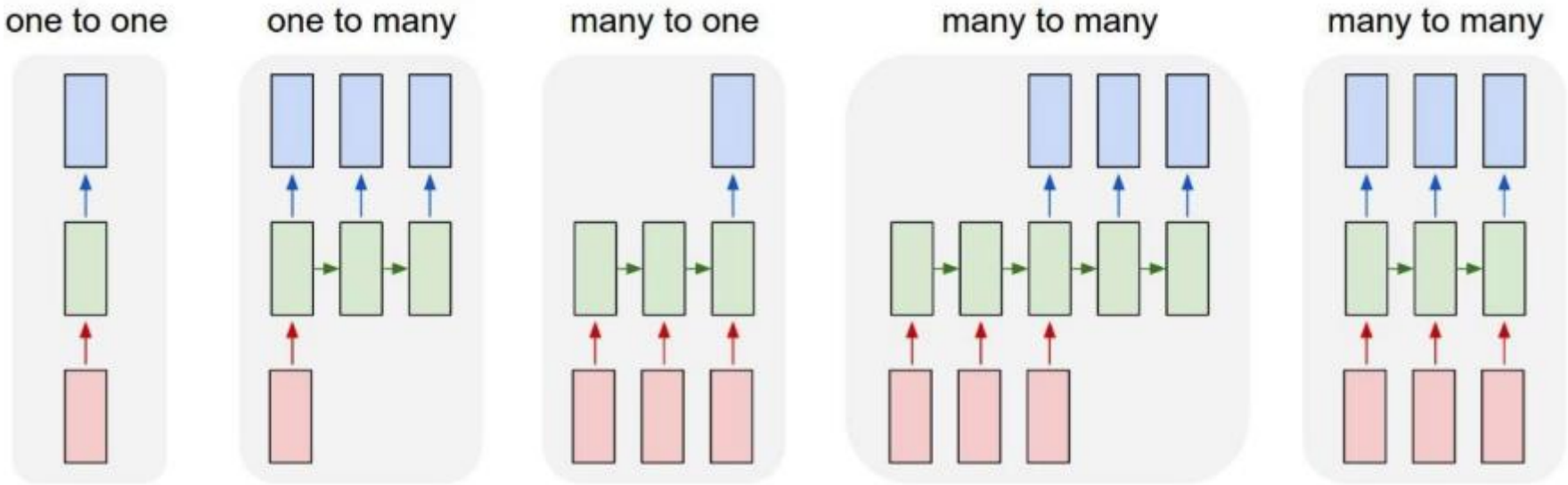
1기 구미진
1기 김지인
1기 황시은

EURON

목차

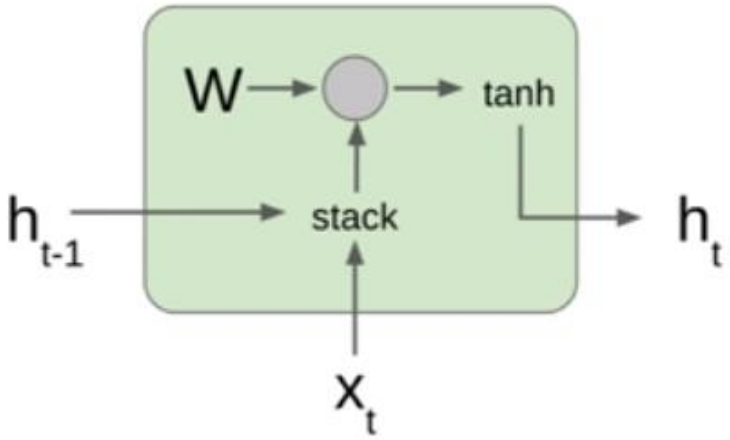
Review

RNN

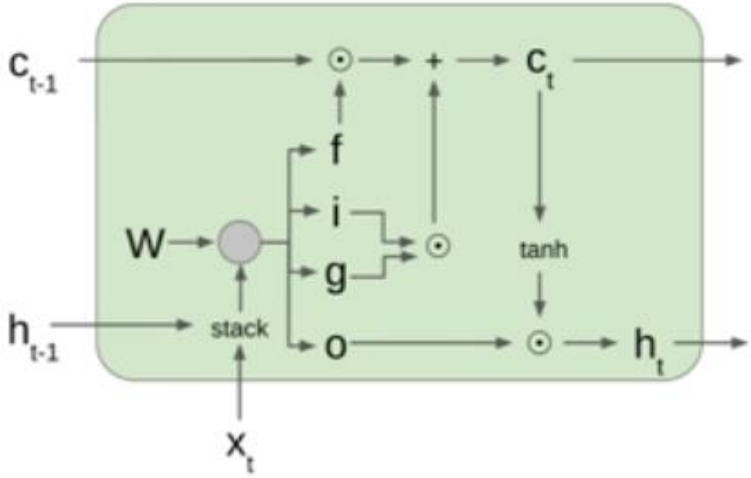


Review

RNN



LSTM
→ Long Term Dependency
문제 해결

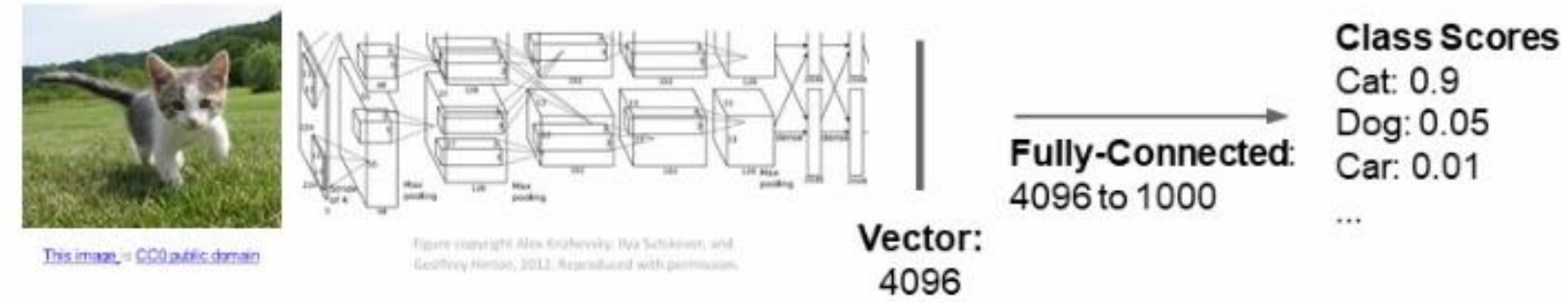


Segmentation, Localization, Detection

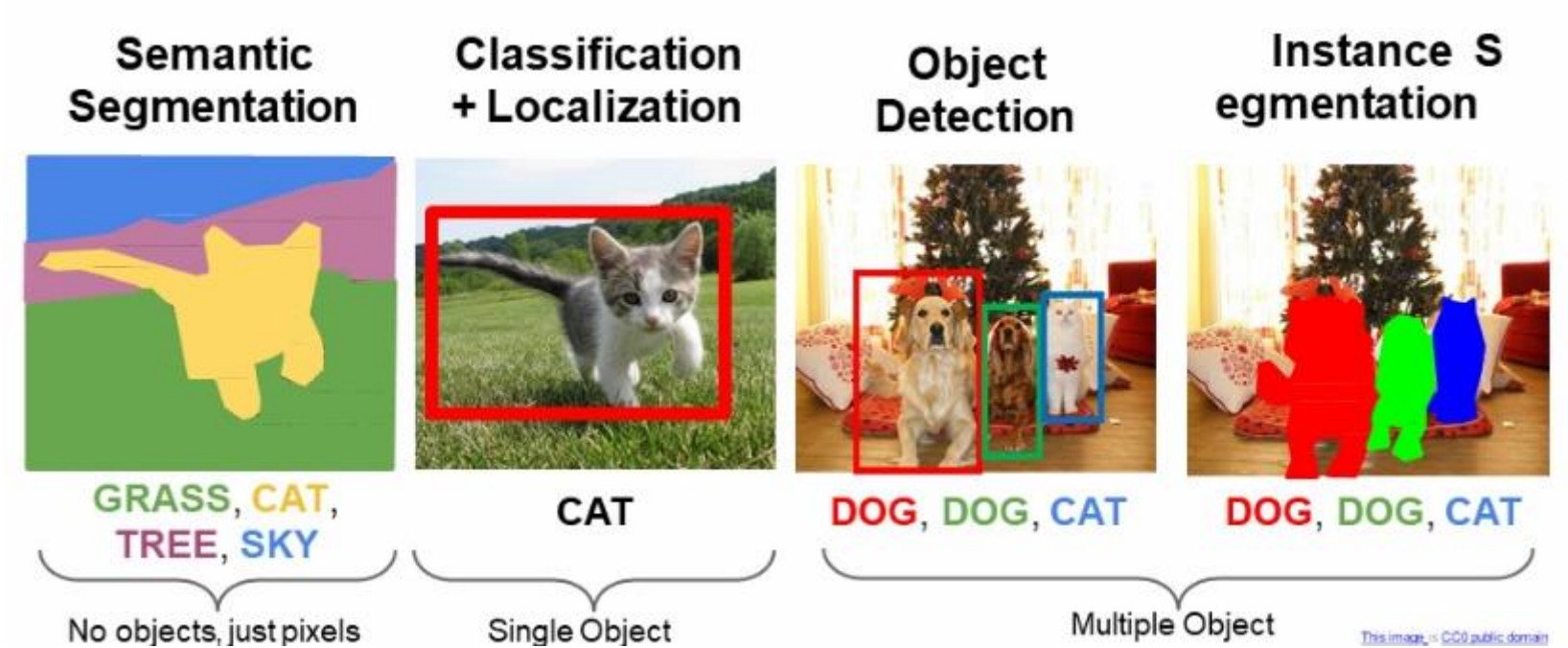
Image Classification

지금까지는

이미지 → 카테고리



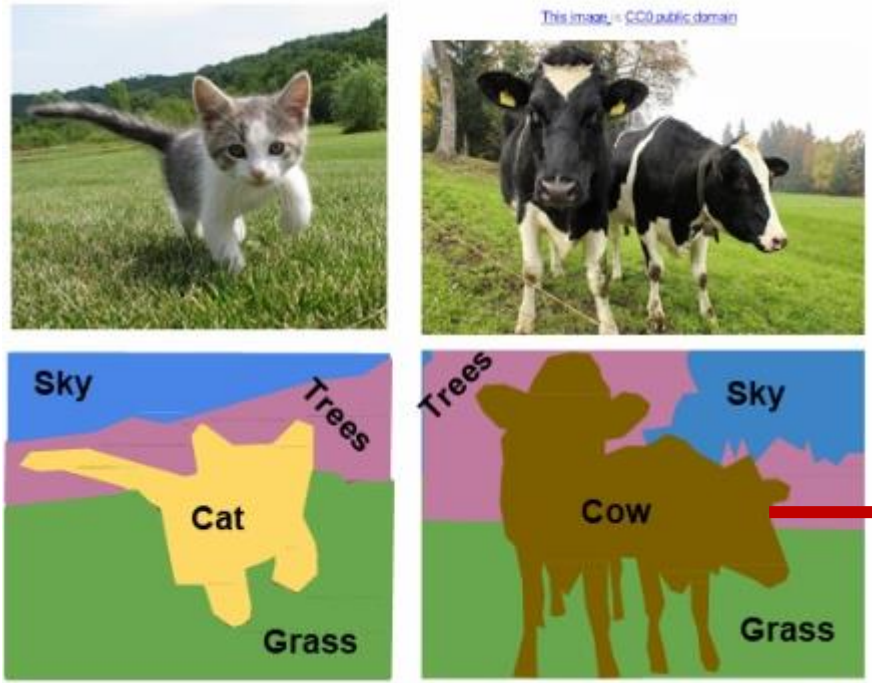
Computer Vision Tasks

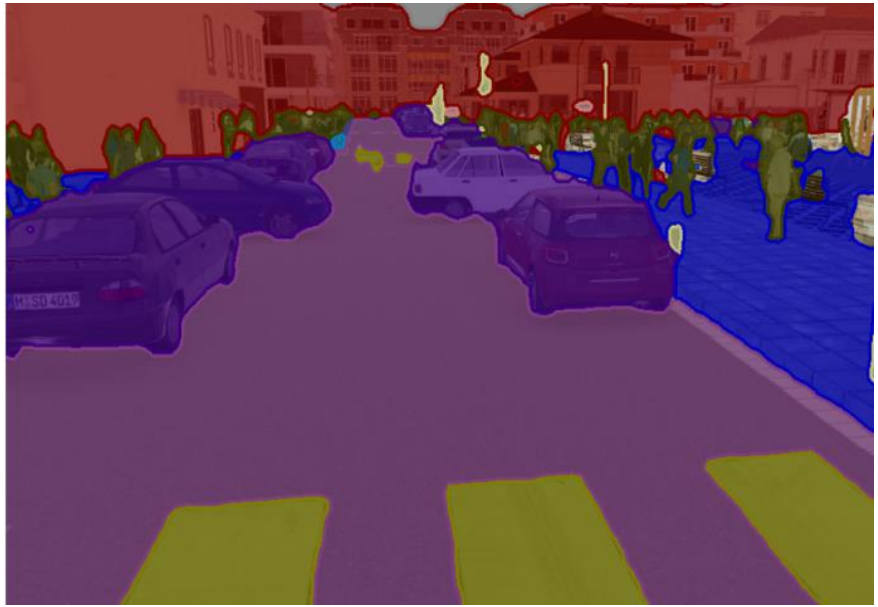


Semantic Segmentation

Input: 이미지

Output: 모든 픽셀에 대한 카테고리



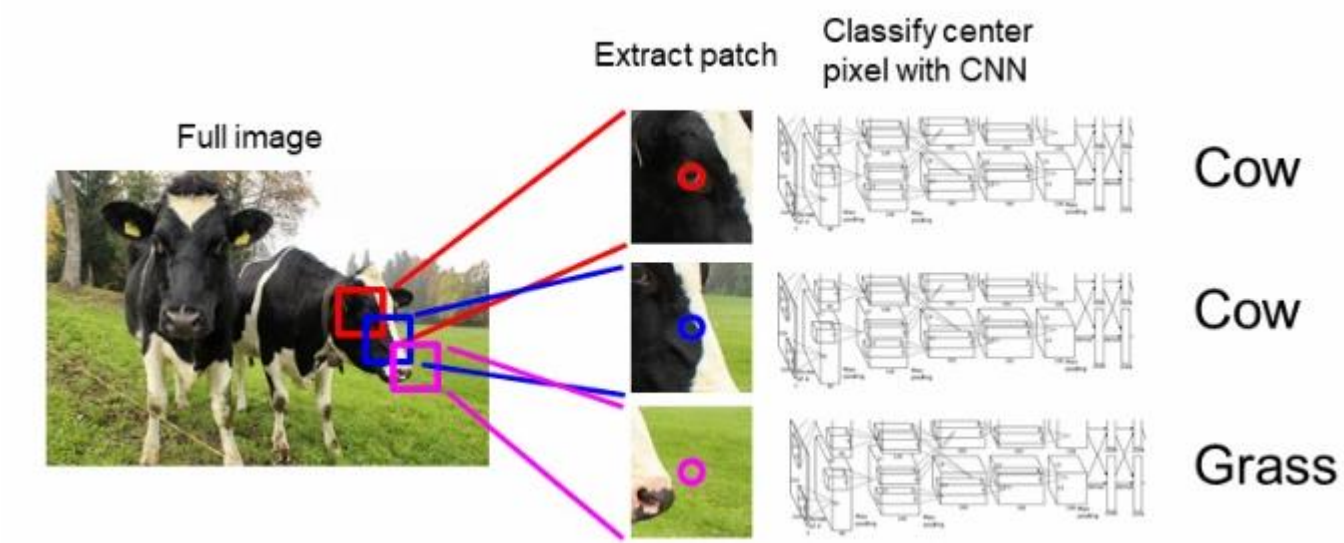


Semantic Segmentation의 한계

→ 같은 카테고리의 여러 객체 분류 X

Semantic Segmentation

Sliding Window



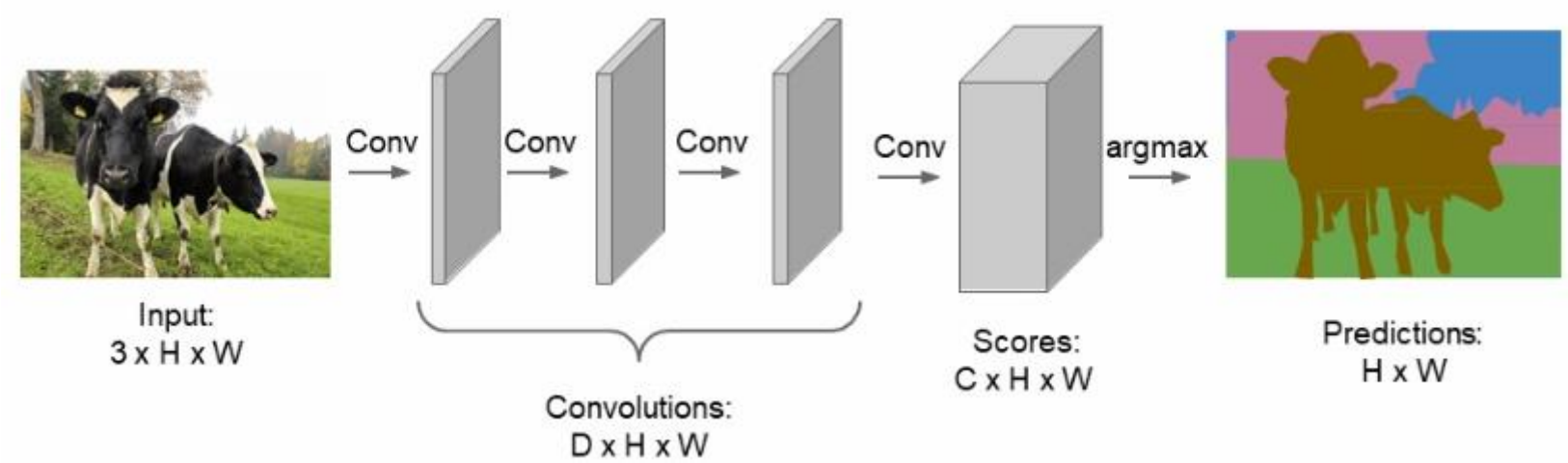
이미지를 패치 단위로 쪼개고, 각 단위가 어떤 카테고리에 속하는지 Classification

➔ 겹치는 영역에 대해서 features 공유

➔ 매우 비효율적!

Semantic Segmentation

Fully Convolutional

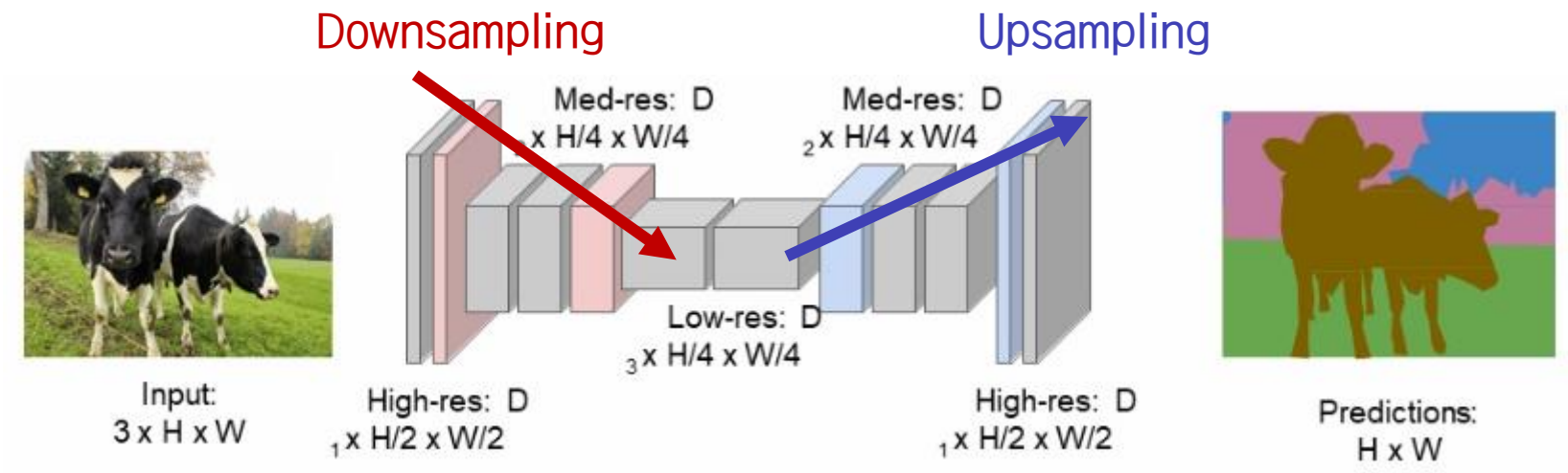


ONLY convolutional layers(Fully Connected Layer 없음)

- 공간 정보 손실 막음 → 계산량 너무 많다!

Semantic Segmentation

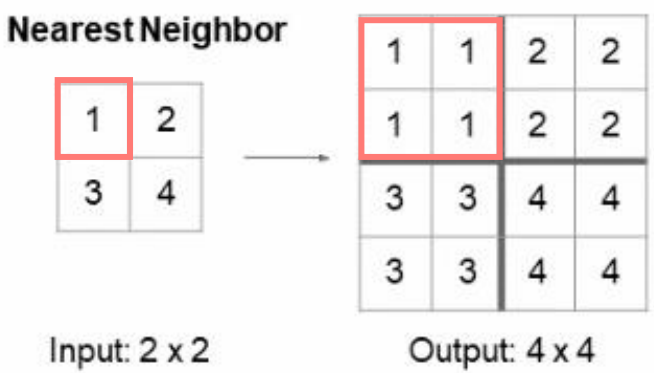
Fully Convolutional



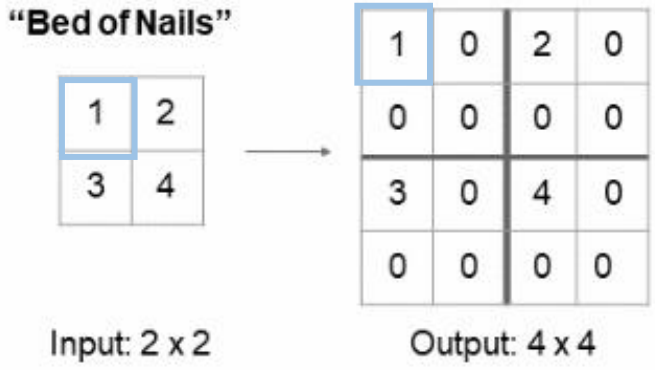
Pooling
Strided Convolution

Upsampling은?

Upsampling



숫자 동일하게 키우기!



1번째 위치에 input 값

나머지는 0으로 채운다

Max Unpooling

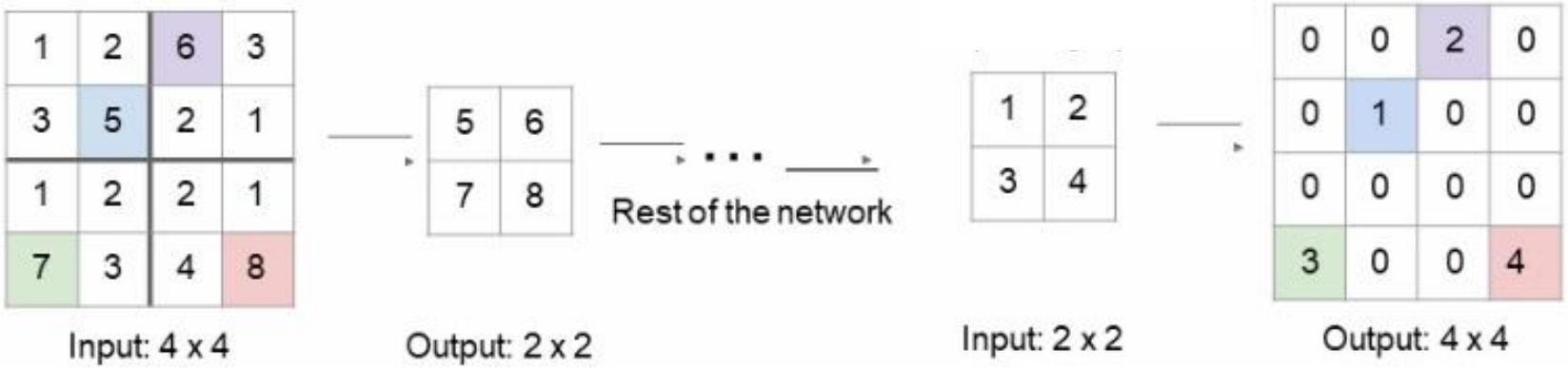
Max Pooling

: Max인 위치 기억하기

Max Unpooling

: 기억한 위치에 값 넣고

나머지는 0으로 채우기

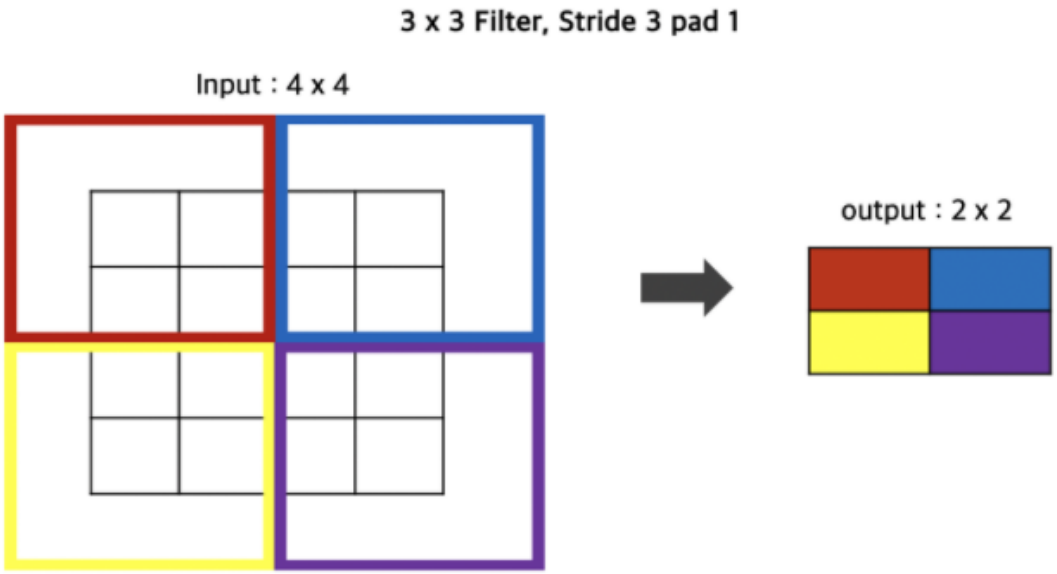


Max Pooling: feature map의 공간 정보 잃음

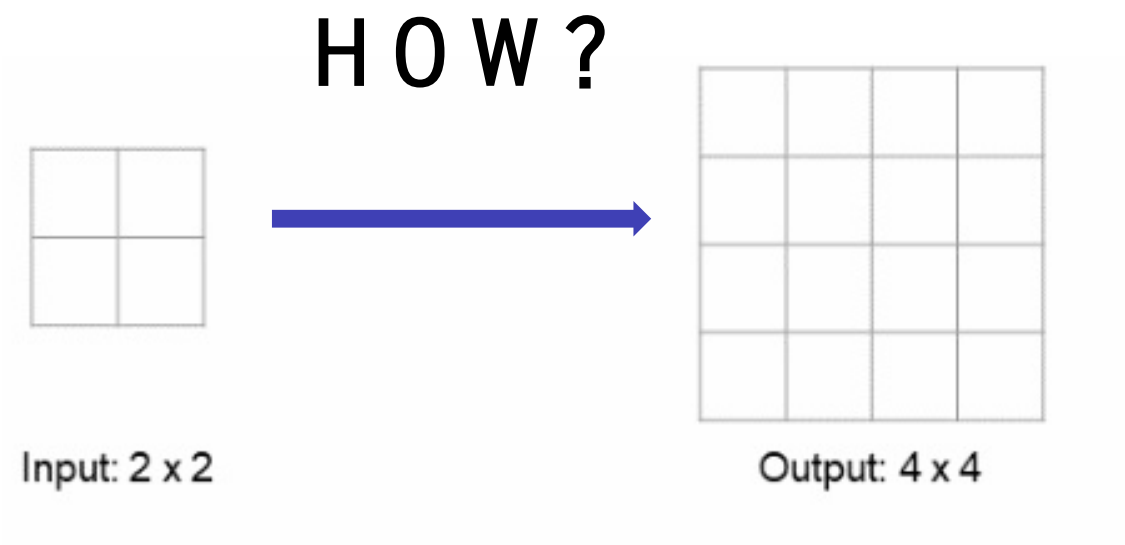
Max Unpooling: 공간 정보 균형 있게 유지

Transpose Convolution

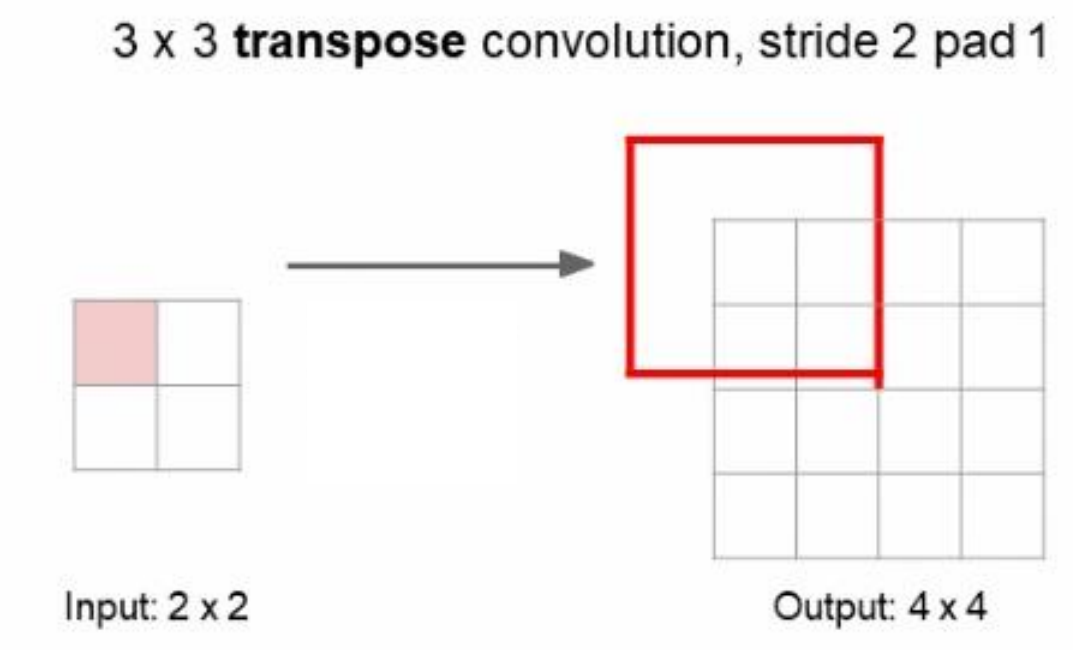
기존의 Convolution 연산



Transpose Convolution



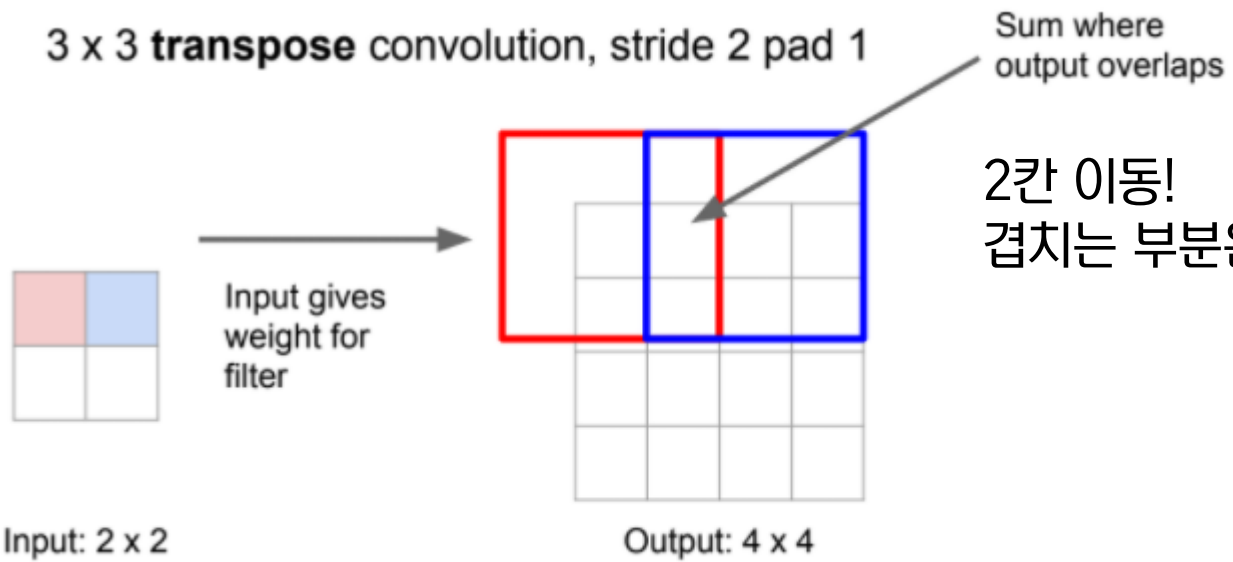
Transpose Convolution



Input: 필터에 곱해지는 가중치 역할

Output: 필터와 입력(가중치)의 곱

Transpose Convolution

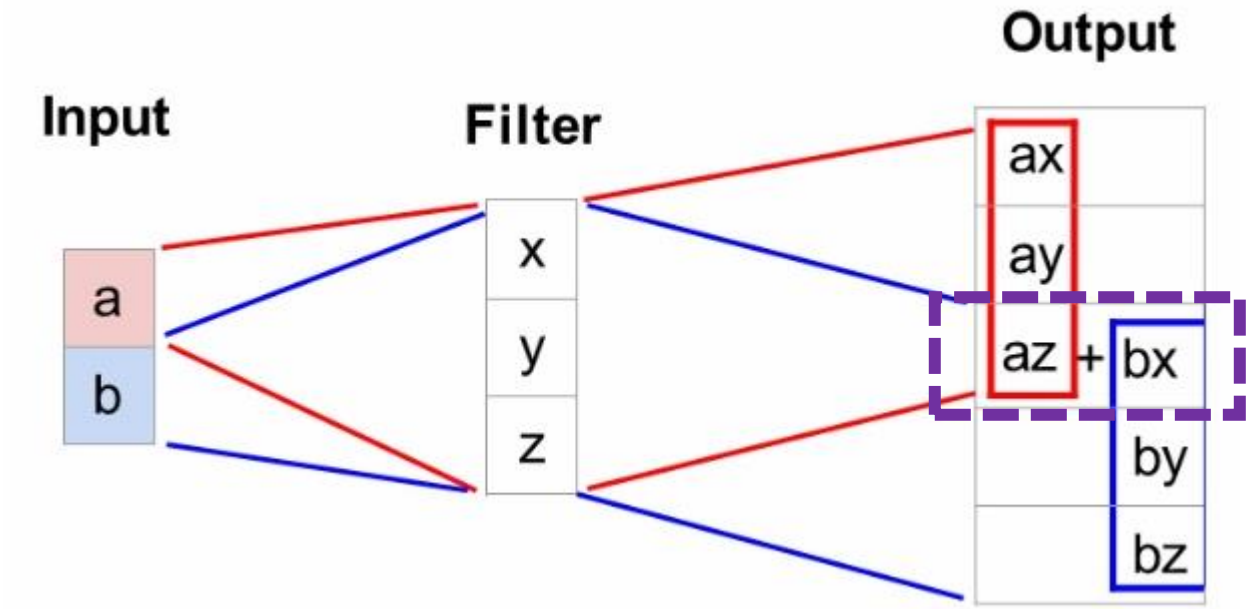


2칸 이동!
겹치는 부분은 더해준다

Other names:
Deconvolution
Upconvolution

Transpose Convolution

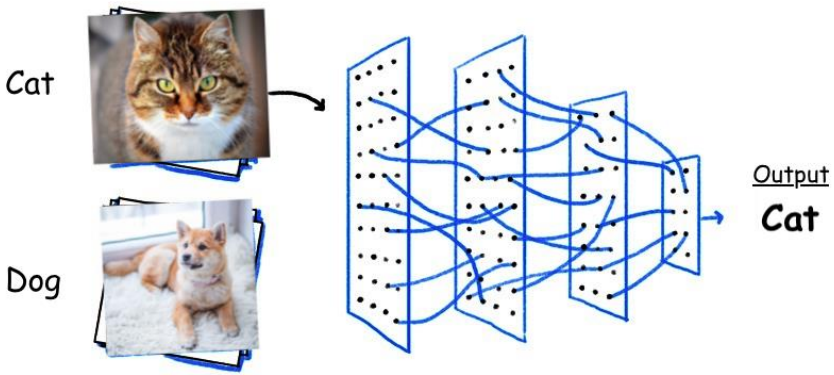
1D Example



Input → Output
으로 매핑되며 크기가
증가한다

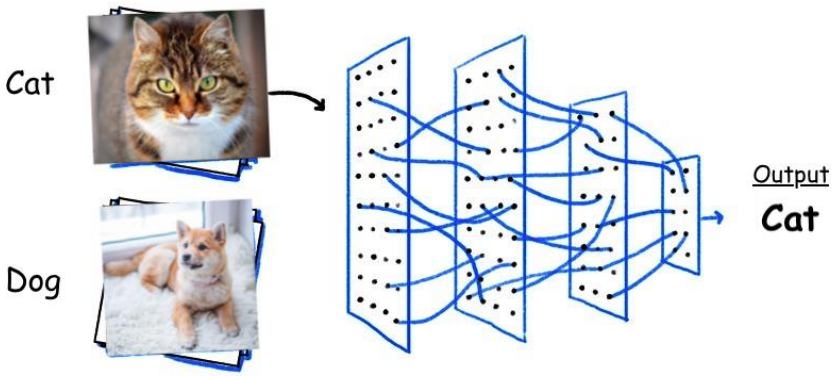
Classification + Localization

- Classification: input image의 labeling problem
- Localization: 하나의 객체가 image의 어디에 위치하는지 찾는 problem

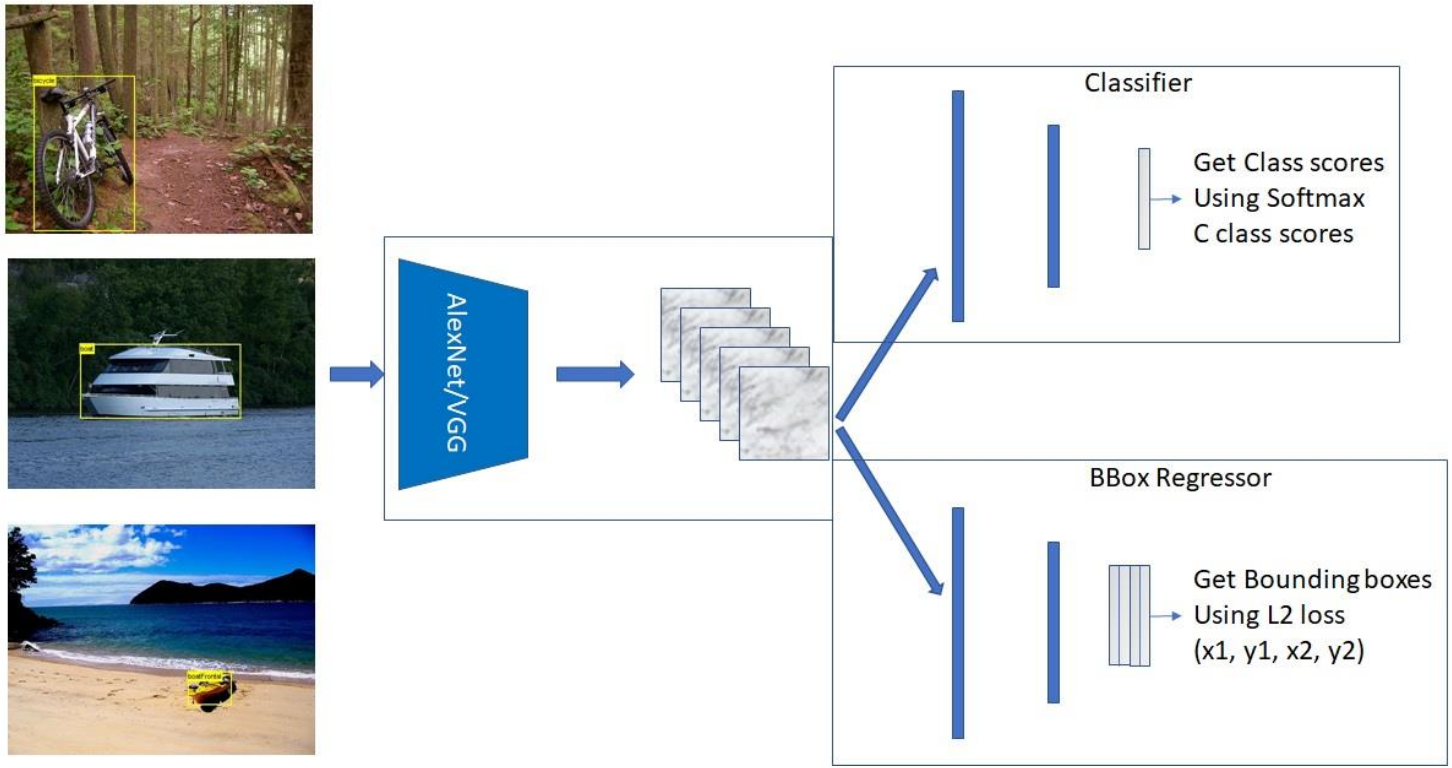


Classification + Localization

- Classification + Localization
  : image 내에서 객체 하나만을 찾아 labeling하고 그 위치를 찾아내는 것



Classification + Localization

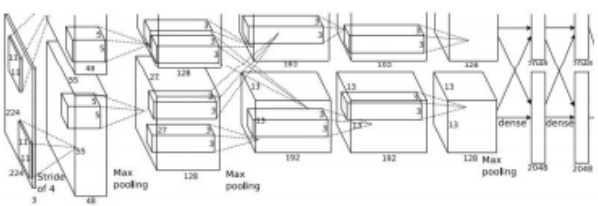


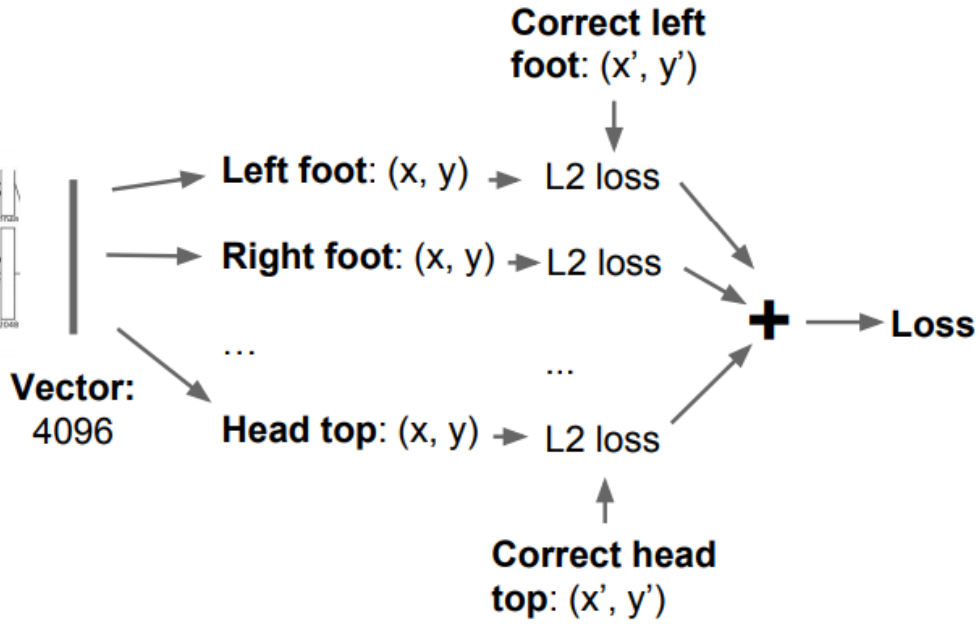
Classification + Localization

- Human pose estimation

Classification + Localization

- Human pose estimation

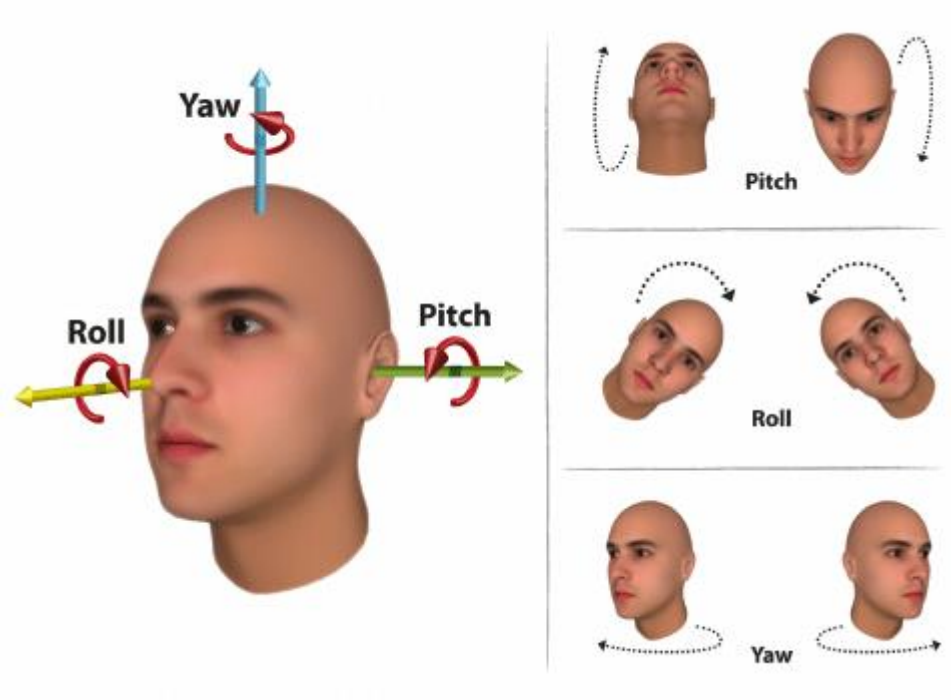




Toshev and Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks", CVPR 2014

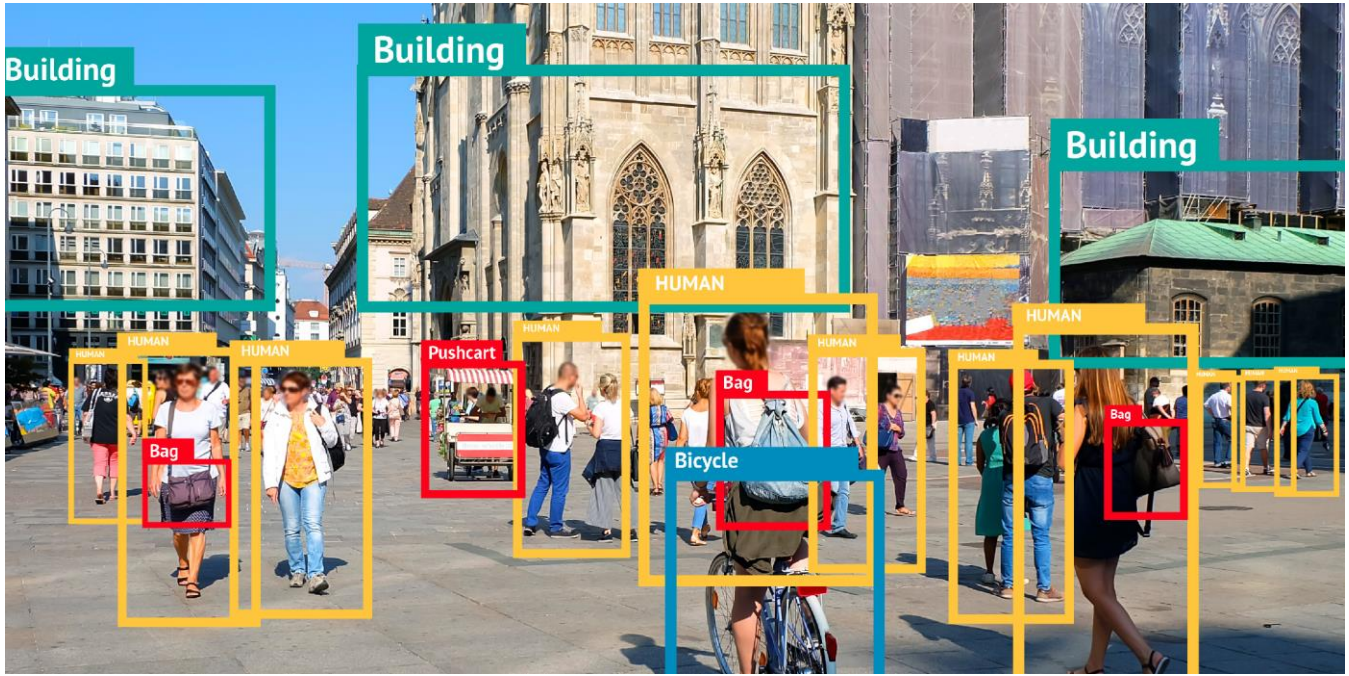
Classification + Localization

- Head pose estimation

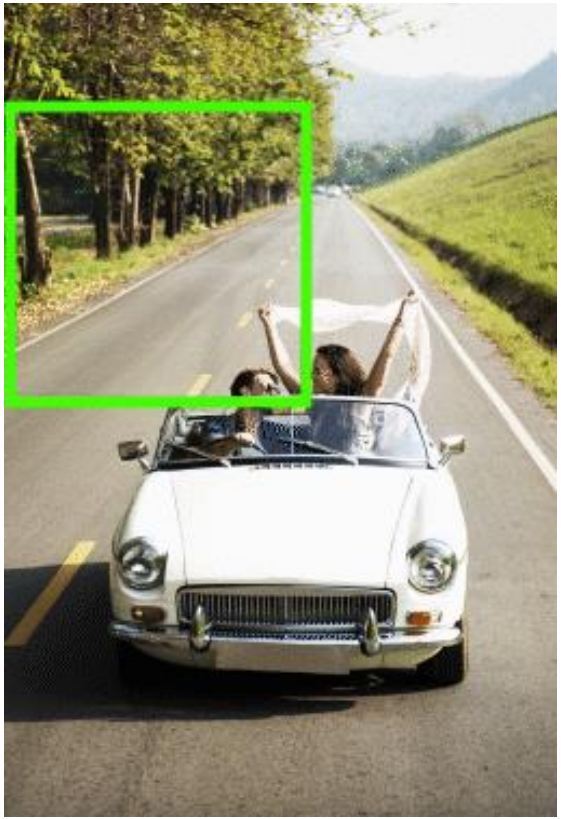


Object Detection

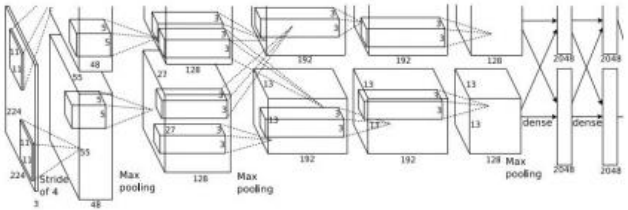
- Input image가 주어지면 해당 image 내의 객체의 bounding box와 category 예측
- Localization + classification과의 차이: 객체의 수



Object Detection



- Sliding window
  - Brute force
  - 계산량이 많음



Car?
Woman?
Man?
Background?

Region Proposal

- DL x
- Selective Search

R-CNN

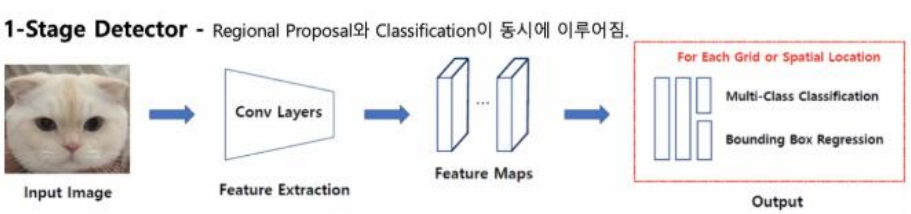
Object detection

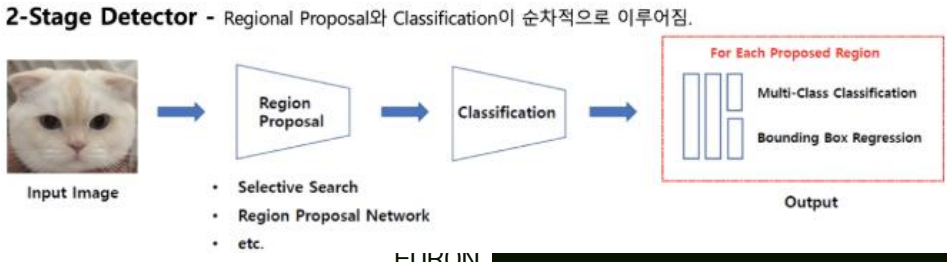
1-stage detector

- RoI영역을 먼저 추출하지 않고,
- 전체 image에 대해서 convolution network로 classification, box regression(localization)을 수행
- 간단하고 쉬운 만큼 속도가 빠르다는 장점
- YOLO, SSD …

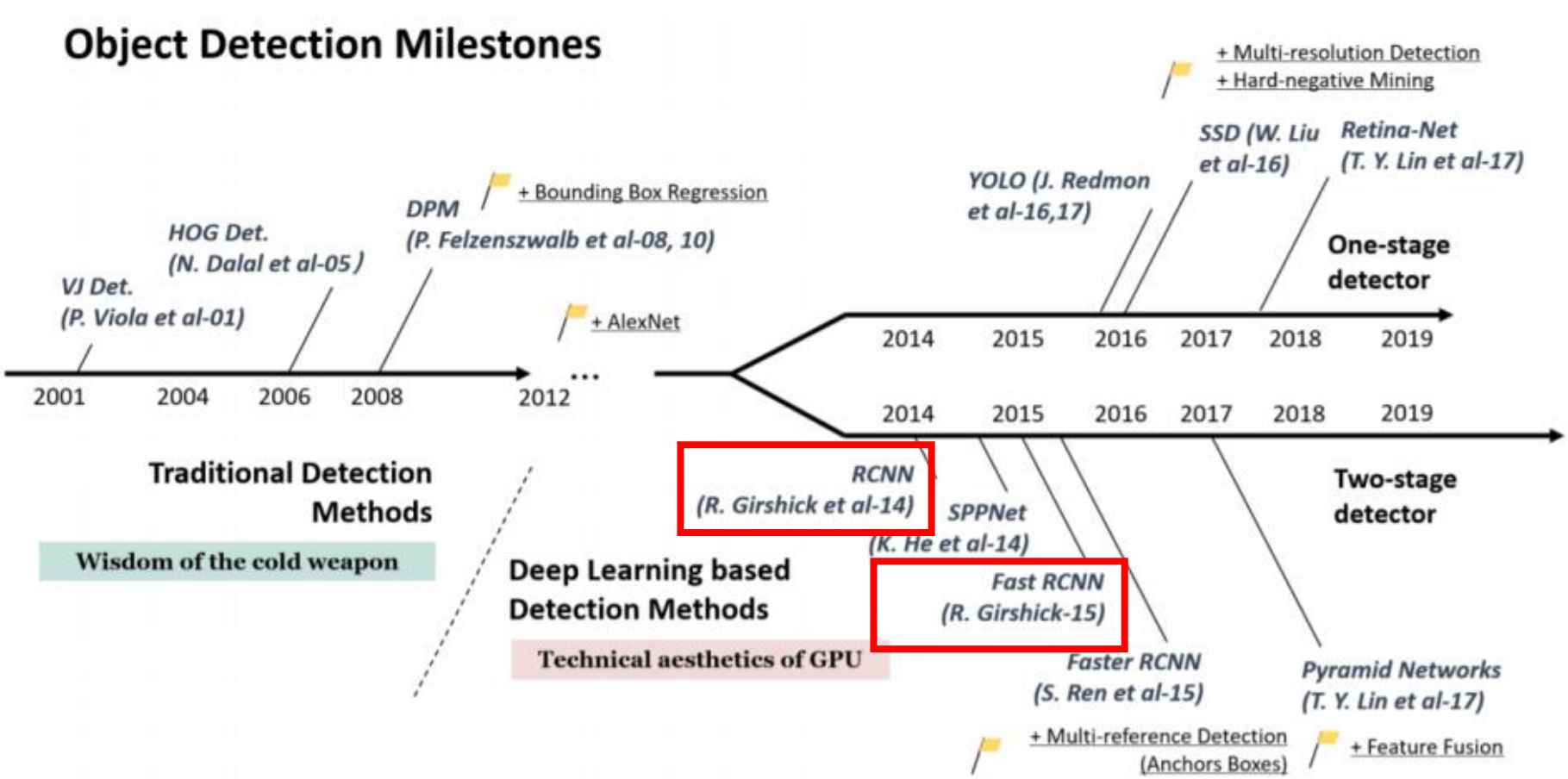
2-stage detector

- Region Proposal과 Classification이 순차적으로 이루어짐
- 비교적 정확도는 높지만 속도가 느림
- R-CNN계열 (R-CNN, Fast R-CNN, Faster R-CNN, Mask R-CNN …)



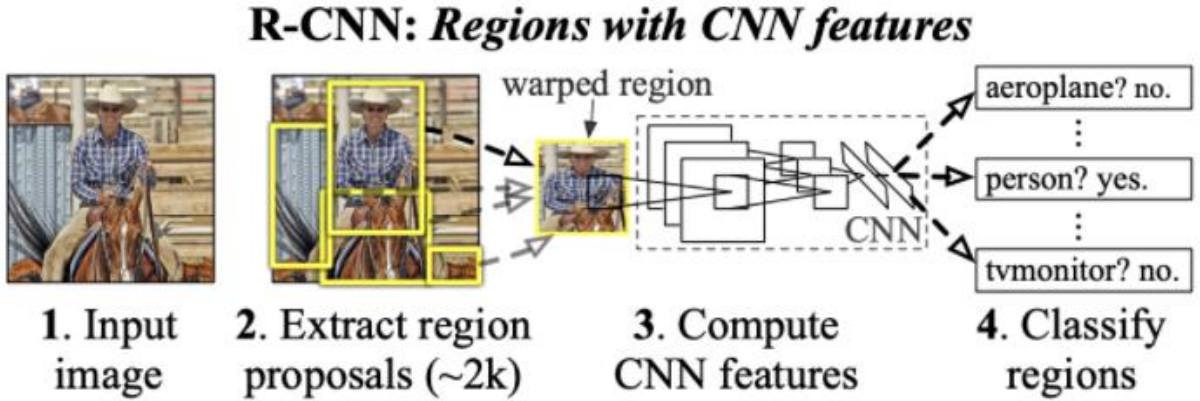


R-CNN


Object Detection Milestones
+ Multi-resolution Detection
+ Hard-negative Mining
SSD (W. Liu et al-16)
Retina-Net (T. Y. Lin et al-17)
YOLO (J. Redmon et al-16,17)
+ Bounding Box Regression
DPM (P. Felzenszwalb et al-08, 10)
HOG Det. (N. Dalal et al-05)
VJ Det. (P. Viola et al-01)
+ AlexNet
One-stage detector
2001
2004
2006
2008
2012
2014
2015
2016
2017
2018
2019
Traditional Detection Methods
Wisdom of the cold weapon
RCNN (R. Girshick et al-14)
SPPNet (K. He et al-14)
Fast RCNN (R. Girshick-15)
Faster RCNN (S. Ren et al-15)
Pyramid Networks (T. Y. Lin et al-17)
Two-stage detector
Deep Learning based Detection Methods
Technical aesthetics of GPU
+ Multi-reference Detection (Anchors Boxes)
+ Feature Fusion

R-CNN

Ross Girshick, Rich feature hierarchies for accurate object detection and semantic segmentation



1. Image를 입력 받는다.
2. Selective search 알고리즘에 의해 region proposal을 약 2000 추출한다.
   추출한 region proposal을 모두 동일한 사이즈로 warp 해준다.
3. 2000개의 warped image를 각각 CNN 모델에 넣는다.
4. 각각의 Convolution 결과에 대해 classification을 진행하여 결과를 얻는다.

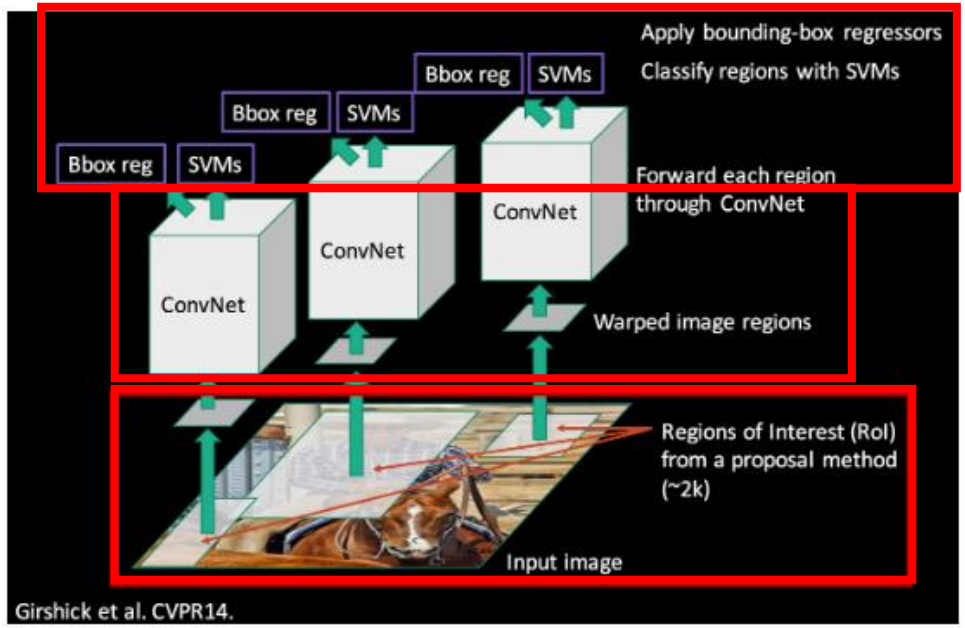
R-CNN

R-CNN의 세 가지 모듈

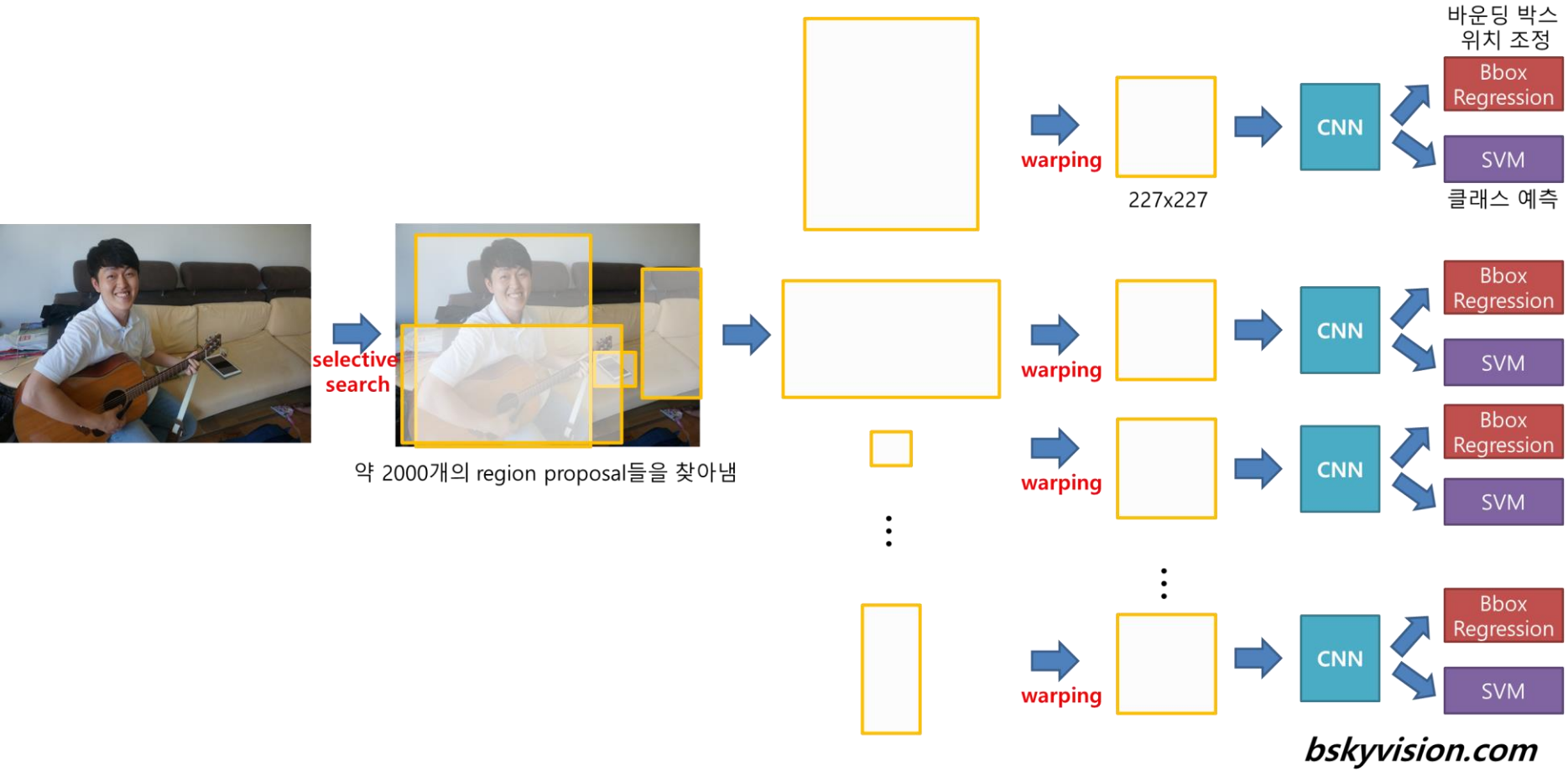
1. Region Proposal
- 기존의 Sliding window 방식의 비효율성 극복

2. CNN
- 각각의 영역으로부터 고정된 크기의 feature vector 추출

3. SVM
- classification을 위한 선형 지도학습 모델



R-CNN

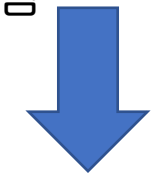

selective search
약 2000개의 region proposal들을 찾아냄
warping
227x227
CNN
바운딩 박스 위치 조정
Bbox Regression
SVM
클래스 예측
bskyvision.com

R-CNN

한계

1) RoI (Region of Interest) 마다 CNN 연산을 함으로써 속도저하

2) multi-stage pipelines으로써 연산을 공유하지 않아
   모델을 한번에 학습시키지 못함

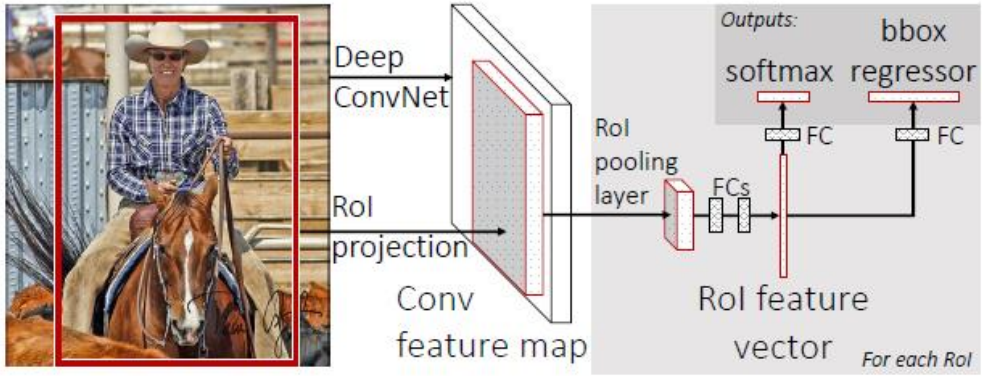


Fast R-CNN

1) RoI pooling

2) CNN 특징 추출부터 classification, bounding box regression까지
   하나의 모델에서 학습

Fast R-CNN



1-1. Selective Search를 통해서 RoI를 찾아낸다.

1-2. 전체 이미지를 CNN에 통과시켜 feature map을 추출한다.

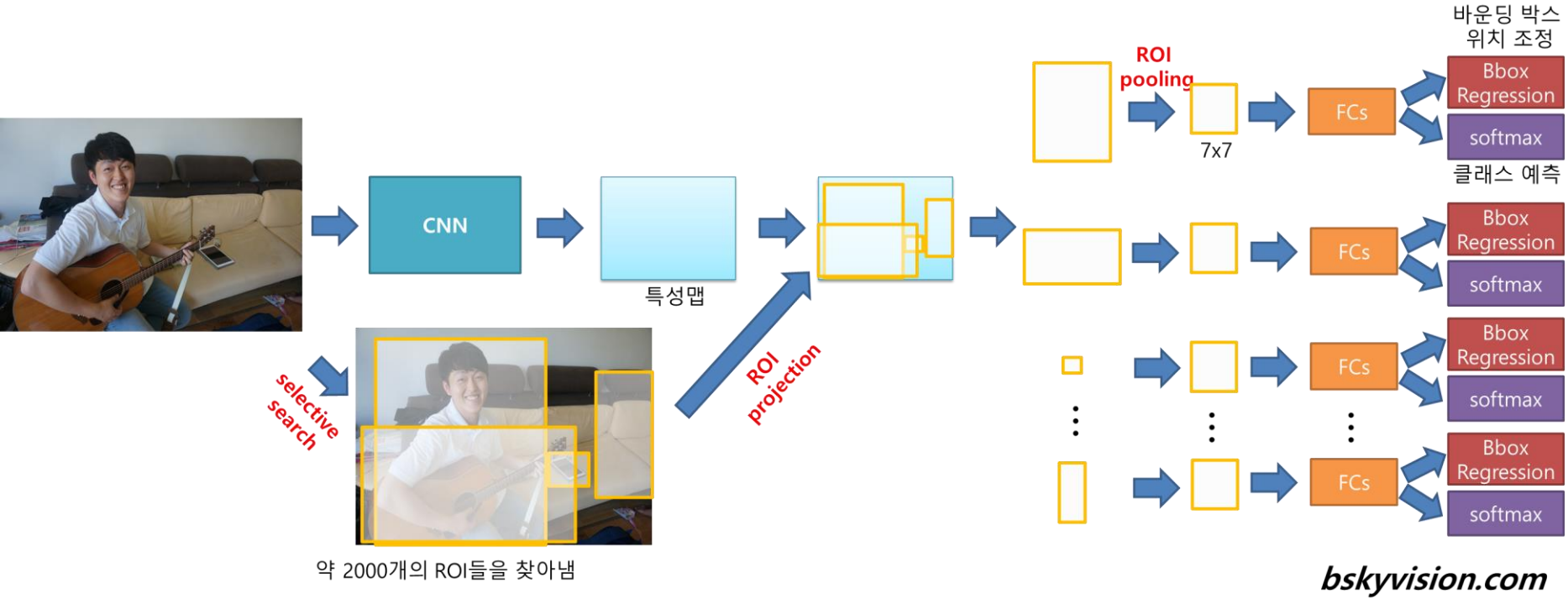
2. RoI를 feature map크기에 맞춰서 projection시킨다.

3. RoI Pooling을 진행하여 고정된 크기의 feature vector를 얻는다.

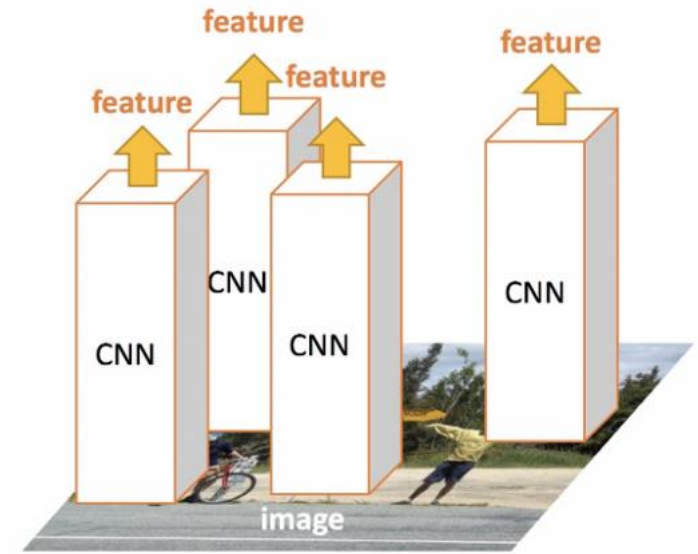
4-1. 하나는 softmax를 통과하여 RoI가 어떤 물체인지 classification 한다.

4-2. bouding box regression을 통해 박스의 위치를 조정한다.

Fast R-CNN

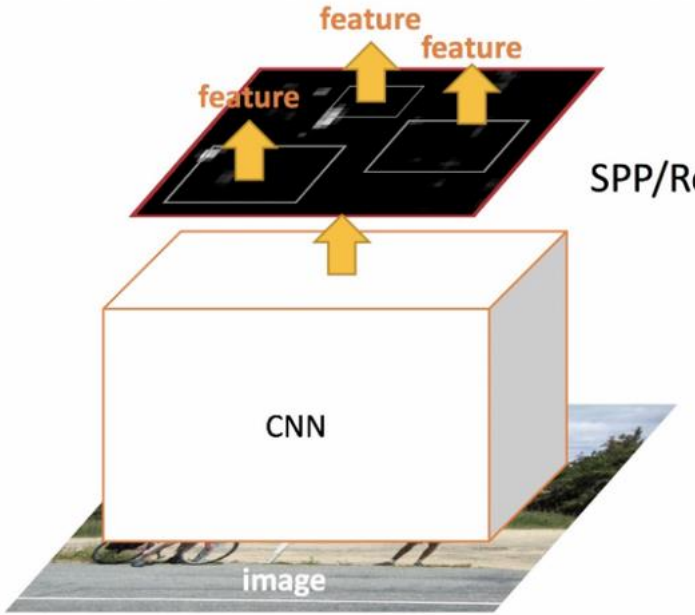

바운딩 박스
위치 조정
ROI
pooling
CNN
특성맵
selective
search
ROI
projection
7x7
FCs
Bbox
Regression
softmax
클래스 예측
약 2000개의 ROI들을 찾아냄
bskyvision.com

Fast R-CNN



R-CNN

- Extract image regions
- 1 CNN per region (2000 CNNs)
- Classify region-based features

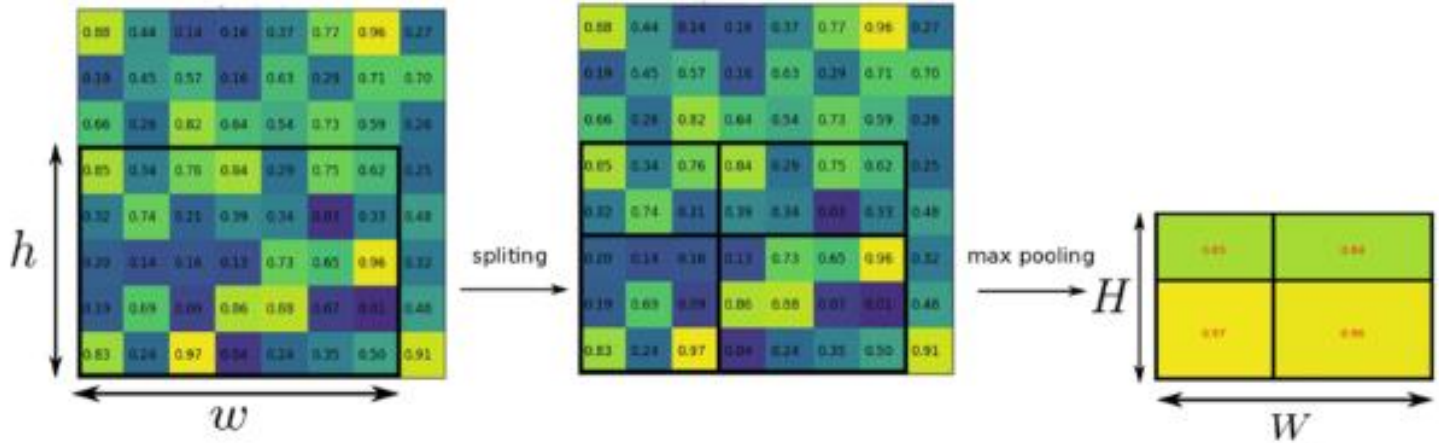


SPP-net & Fast R-CNN (the same forward pipeline)

- 1 CNN on the entire image
- Extract features from feature map regions
- Classify region-based features

Fast R-CNN

RoI Pooling: 크기가 다른 feature map의 region마다 max pooling을 진행하여 원하는 size로 맞추는 방법



Region proposal 5x7
Feature map 8x8

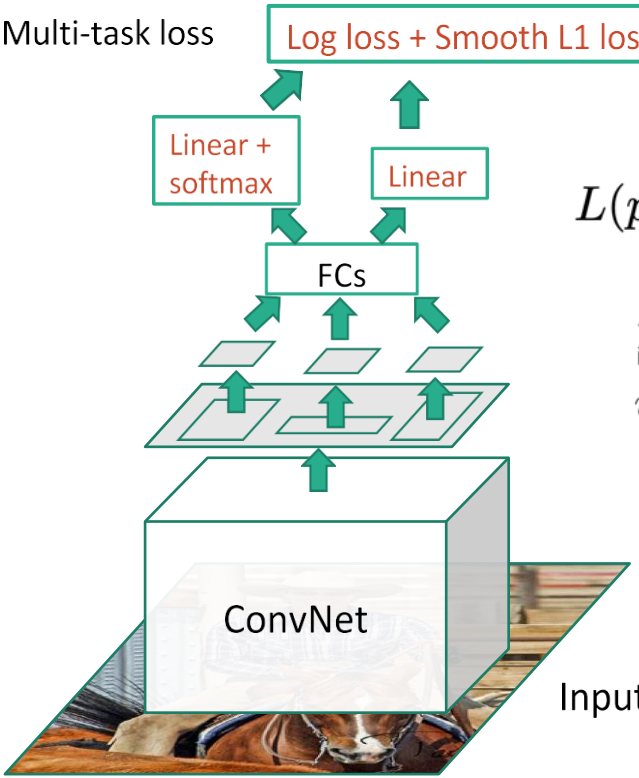
grid
(5/2 = 2, 7/2 = 3)

2x2

2000번의 CNN 연산을 1번의 CNN 연산으로!

Fast R-CNN

Multi-task loss



Loss function – multi-task Loss

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1]L_{\text{loc}}(t^u, v)$$

Classification

$p = (p_0, \ldots, p_K)$: 예측된 Class Score

in $K + 1$ categories (0은 background)

$u =$ 실제 Class Score

Bounding box Regression

$t^u = (t^u_{\text{x}}, t^u_{\text{y}}, t^u_{\text{w}}, t^u_{\text{h}})$: 예측된 tuple

$(v_{\text{x}}, v_{\text{y}}, v_{\text{w}}, v_{\text{h}})$: 실제 Bounding box 좌표 값

u가 0(background)일때 0, 나머지일때 1

$$L_{\text{cls}}(p, u) = -\log p_u$$

<분류>

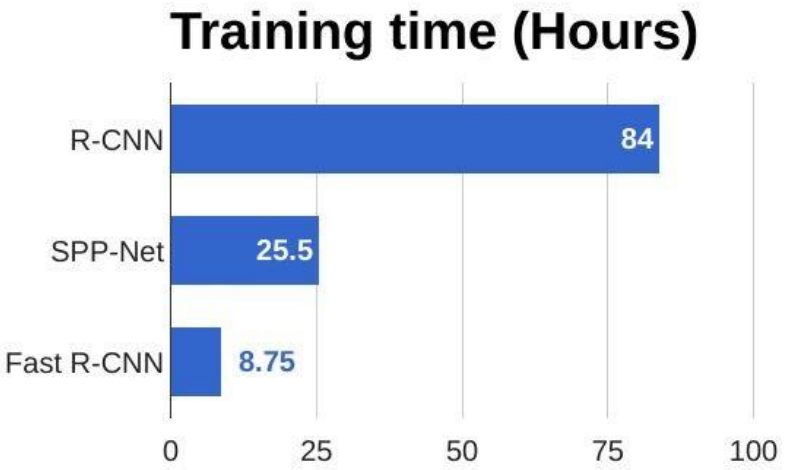
$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t^u_i - v_i)$$

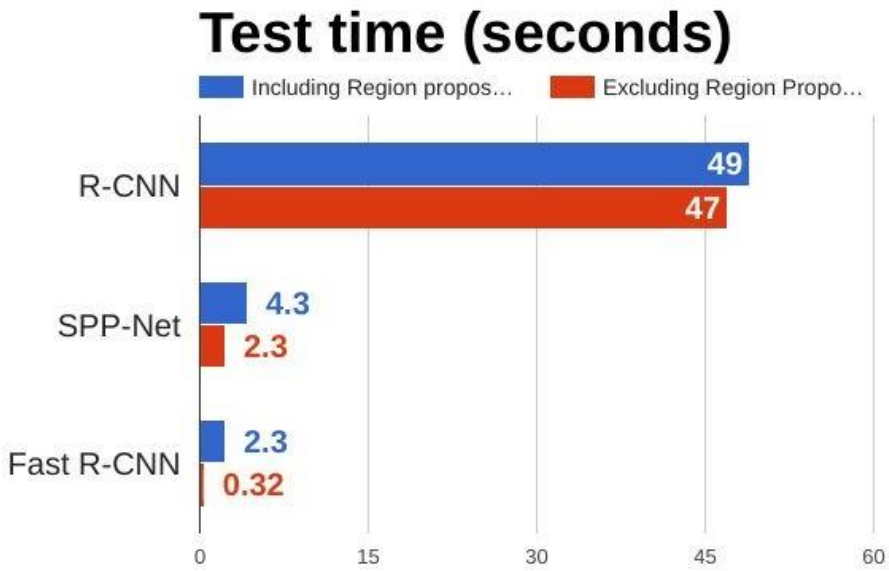
$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise} \end{cases}$$

<회귀>

Girshick, "Fast R-CNN", ICCV 2015.

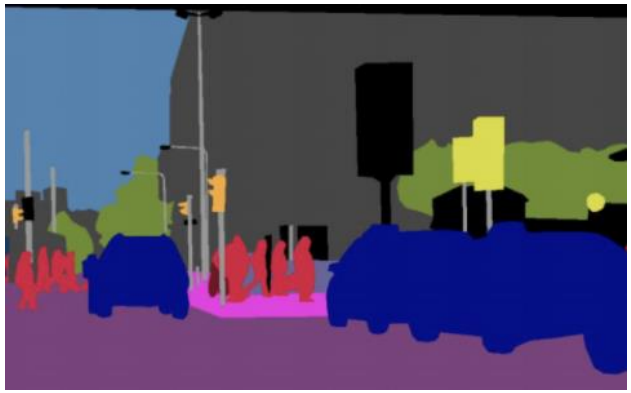
Fast R-CNN

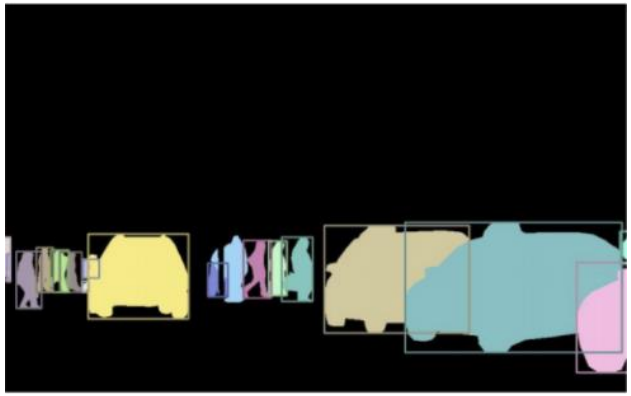




Girshick et al, “Rich feature hierarchies for accurate object detection and semantic segmentation”, CVPR 2014.
He et al, “Spatial pyramid pooling in deep convolutional networks for visual recognition”, ECCV 2014
Girshick, “Fast R-CNN”, ICCV 2015

Instance Segmentation





Semantic segmentation

같은 class의 object들에 대해
서로 구분할 수 없다는 단점

Instance segmentation

같은 class의 object들에 대해서도
서로 다른 instance로 구분

Mask R-CNN

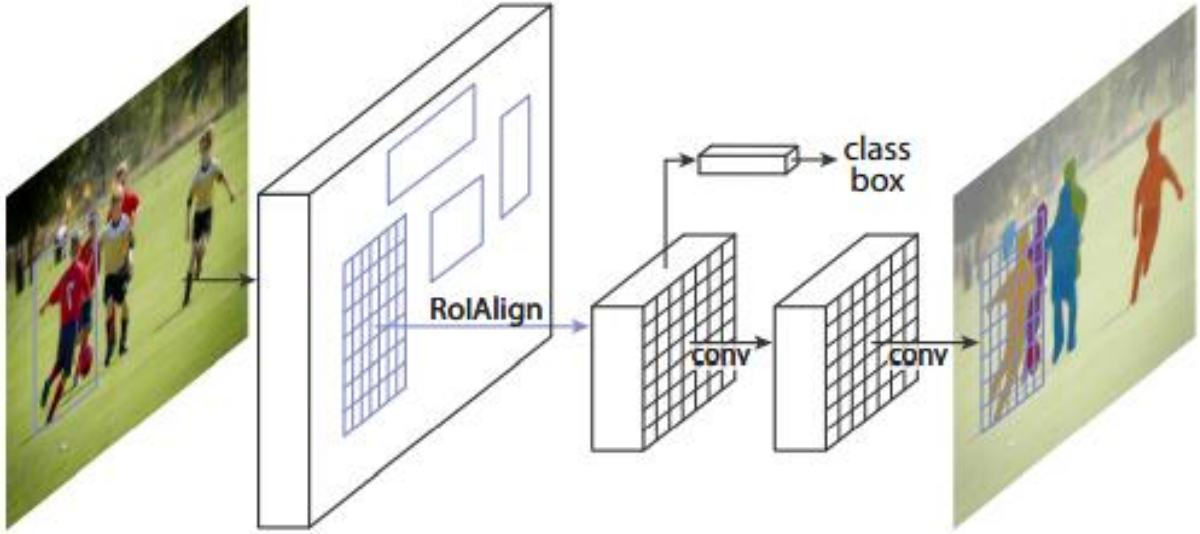
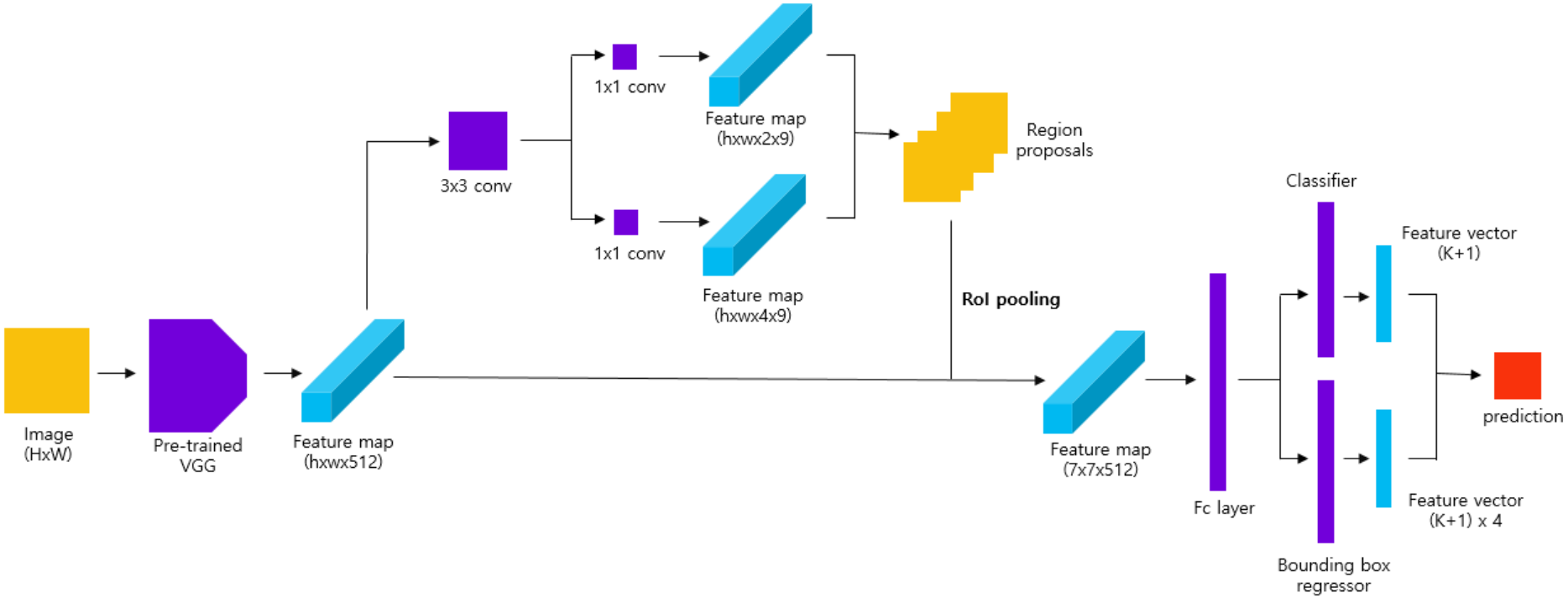


Figure 1. The **Mask R-CNN** framework for instance segmentation.

He et al, “Mask R-CNN”, arXiv 2017

Mask R-CNN

Faster R-CNN



Mask R-CNN

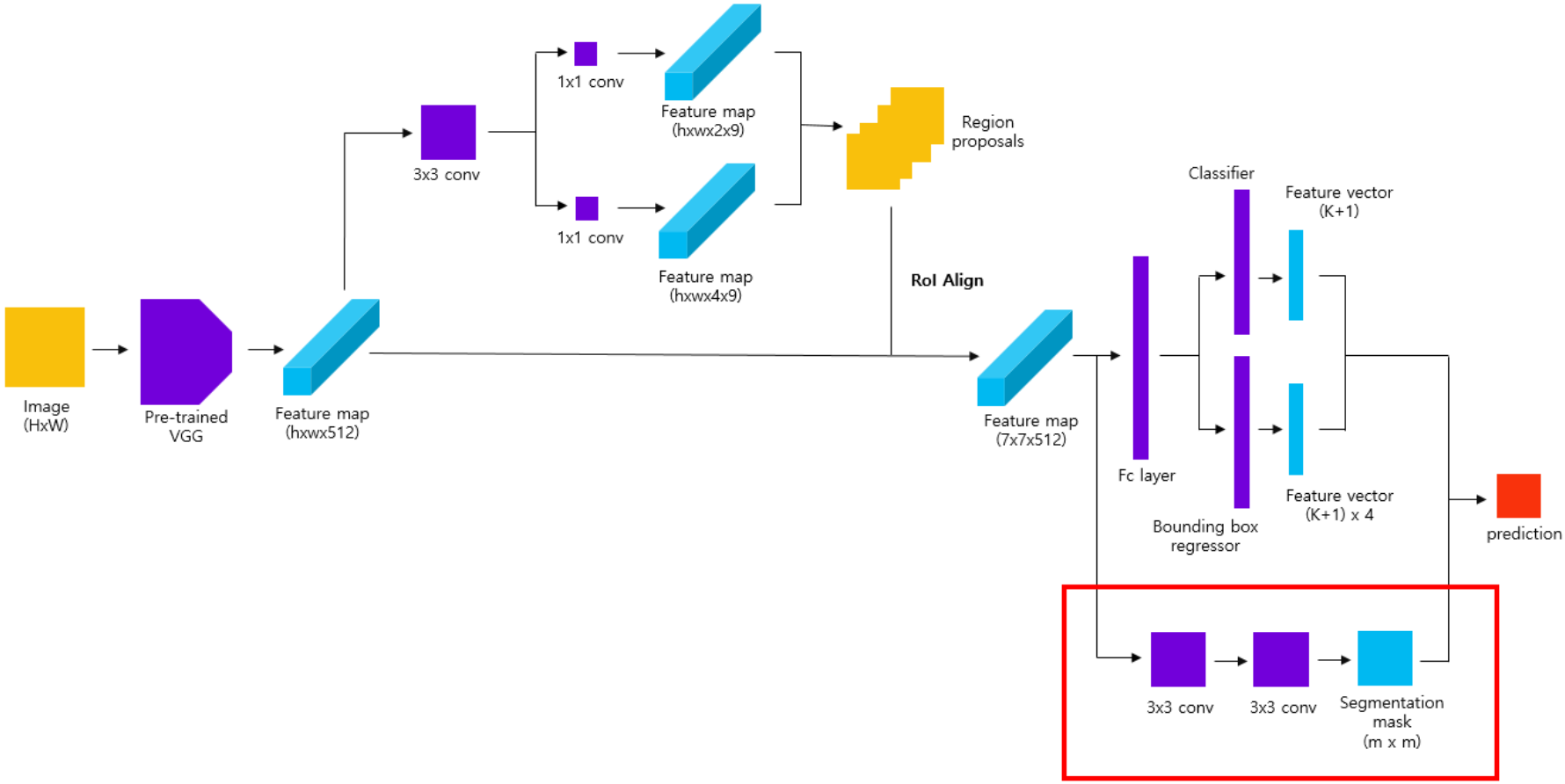
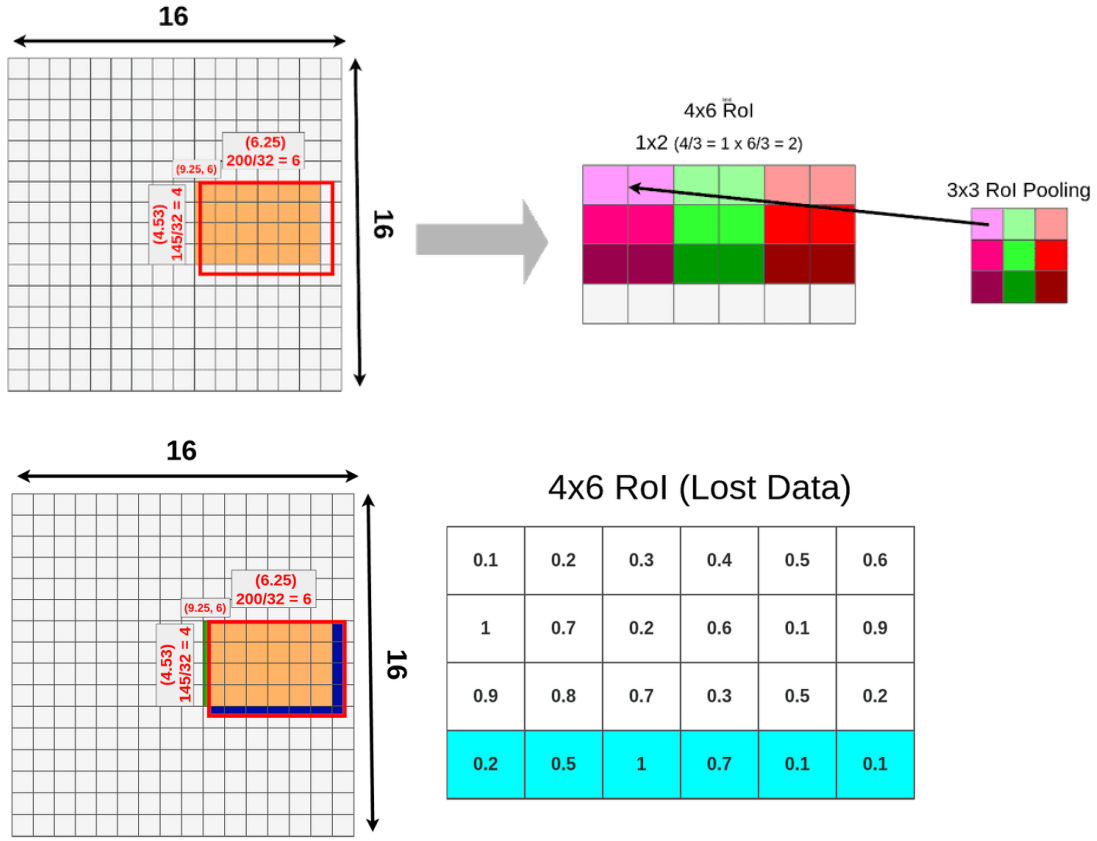


Image
(HxW)
Pre-trained
VGG
Feature map
(hxwx512)
3x3 conv
1x1 conv
Feature map
(hxwx2x9)
1x1 conv
Feature map
(hxwx4x9)
Region
proposals
RoI Align
Feature map
(7x7x512)
Fc layer
Classifier
Feature vector
(K+1)
Bounding box
regressor
Feature vector
(K+1) x 4
prediction
3x3 conv
3x3 conv
Segmentation
mask
(m x m)
Mask branch

Mask R-CNN

RoI Pooling의 문제점

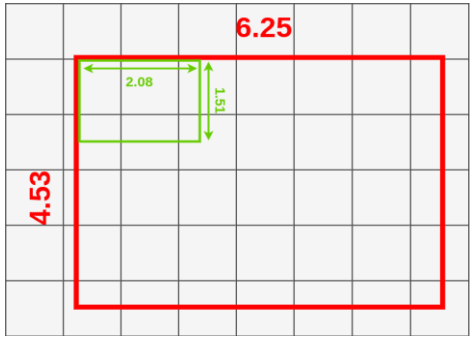


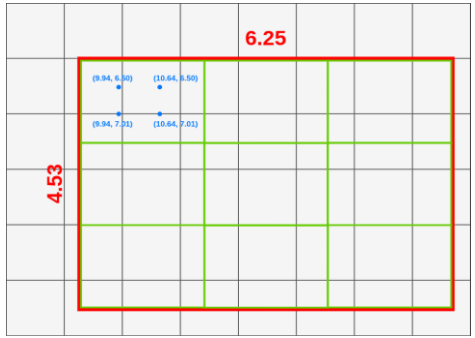
4x6 RoI (Lost Data)

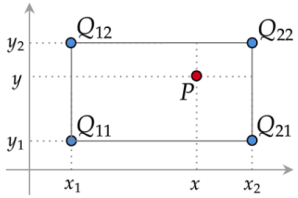
0.1	0.2	0.3	0.4	0.5	0.6
1	0.7	0.2	0.6	0.1	0.9
0.9	0.8	0.7	0.3	0.5	0.2
0.2	0.5	1	0.7	0.1	0.1

Mask R-CNN

RoI Align



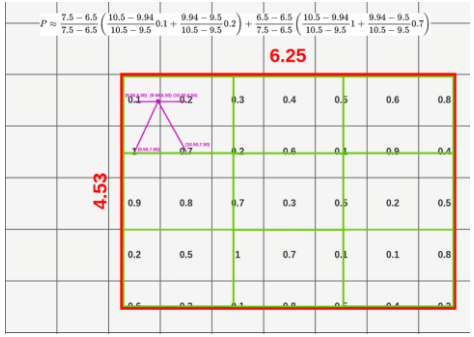


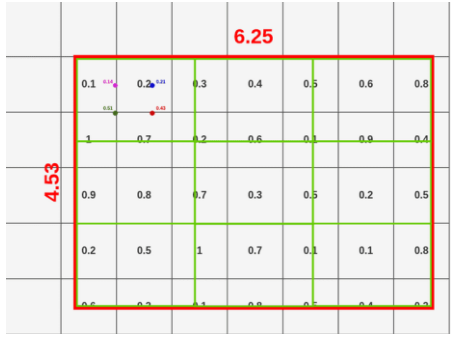


	x_1	x	x_2
y_1	Q_{11}		Q_{21}
y		P	
y_2	Q_{12}		Q_{22}

$$P \approx \frac{y_2 - y}{y_2 - y_1}\left(\frac{x_2 - x}{x_2 - x_1}Q_{11} + \frac{x - x_1}{x_2 - x_1}Q_{21}\right) + \frac{y - y_1}{y_2 - y_1}\left(\frac{x_2 - x}{x_2 - x_1}Q_{12} + \frac{x - x_1}{x_2 - x_1}Q_{22}\right)$$

Bilinear interpolation





감사합니다

Q&A