Detection and Segmentation

Semantic Segmentation (의미분할)

- input an image and then output a decision of a category for every pixel in the image
- don't differentiate instances, only care about pixels
- IDEA: Sliding Window — window 마다 classification (계산 비용이 too expensive) Don't do it

: Fully Convolutional — classification decision per pixel
put 'cross entropy loss' on every pixel
ground truth category label과 loss 계산

Input Image

Expected Output (Source – Ref. 3)

Pooling, Strided Convolution ... → downsampling & upsampling inside the network ⇒ 길이만 효율적 계산

"Unpooling"

1. "Nearest Neighbor"

1	2
3	4

→

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

2. "Bed of Nails"

1	2
3	4

→

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

3. Max Unpooling

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

→

5	6
7	8

→ ... →

1	2
3	4

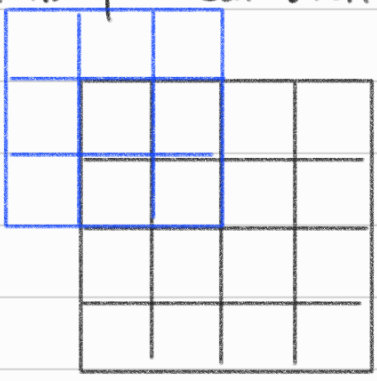
→

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

<MAX POOLING>

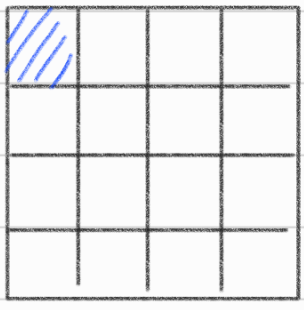
* Corresponding pairs of downsampling and unsampling layers

4. Transpose Convolution



<input>

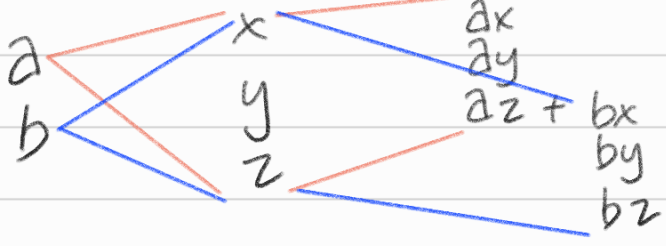
Dot product btw. filter and input



<output>

ex.

Input Filter Output



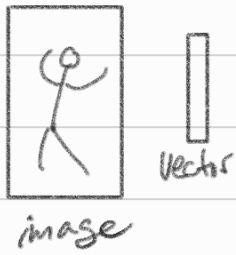
즉, output은 가중합이 적용된 결과를 겹친 중복이 있어야 함

Convolution = Matrix multiplication

Classification + Localization ⇒ Image 내 대상의 위치

have two loss function ① Softmax for ground truth category label
② L2 loss for bounding box

Human Pose Estimation 14개 parts로



left foot L2 loss
right foot L2 loss Σ ⇒ Loss
⋮ ⋮

Object Detection (객체 탐지)

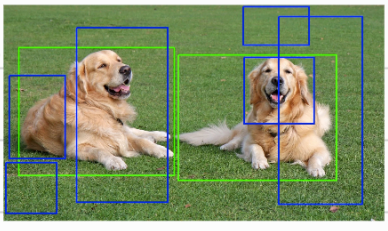
image 마다 object 개수도 종류도 다르니까 challenging

Sliding Window

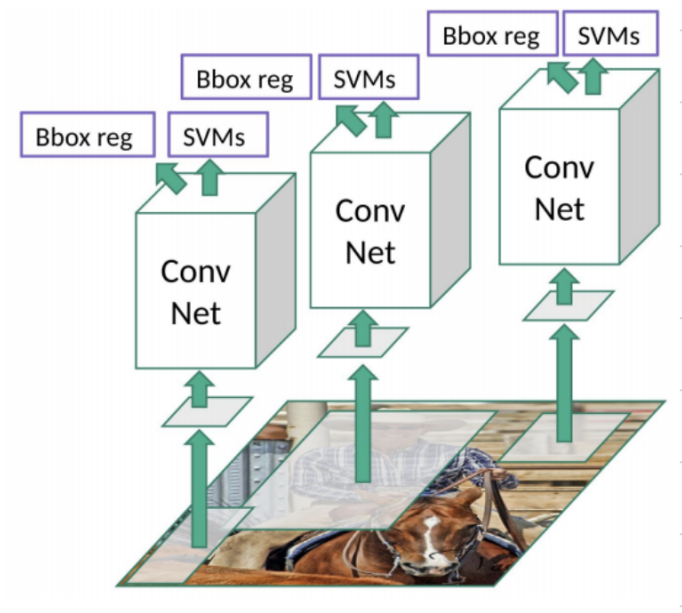
Apply a CNN to many different crops of the image, CNN classifies each crop as object or background

problem: need to apply CNN to huge # of locations and scales, Very computationally EXPENSIVE

Region Proposal - find blobby image regions that are likely to contain objects (not DL)



R-CNN (Region Proposal과 CNN 결합)



4. Classify regions with SVMs

3. Forward each region through convnet

2. CNN은 같은 크기 input 기대하므로 warp

1. RoI (Regions of Interest) from a proposed method

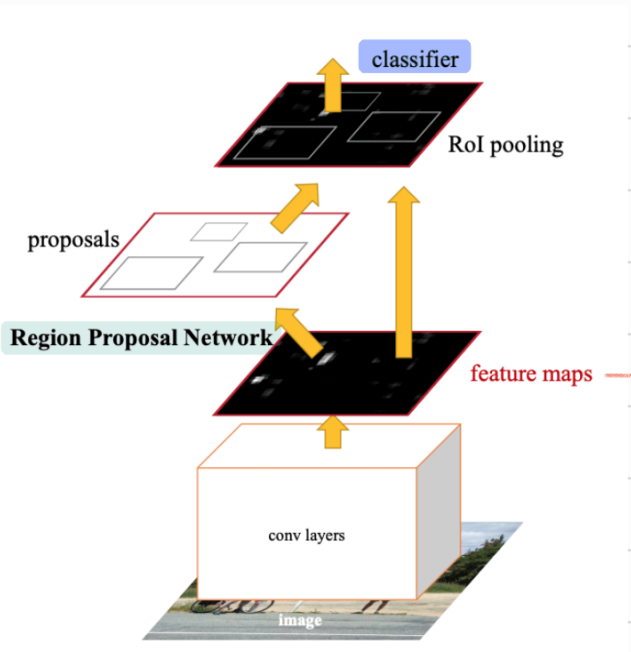
Problem: 여전히 비효율적인 연산 (too expensive), memory 비효율, too slow

Fast R-CNN

1. CNN layers 통해 feature map으로 변환
2. RoI from proposal method

3. ROI Pooling layers
4. Softmax classifier , Bounding-box regressors

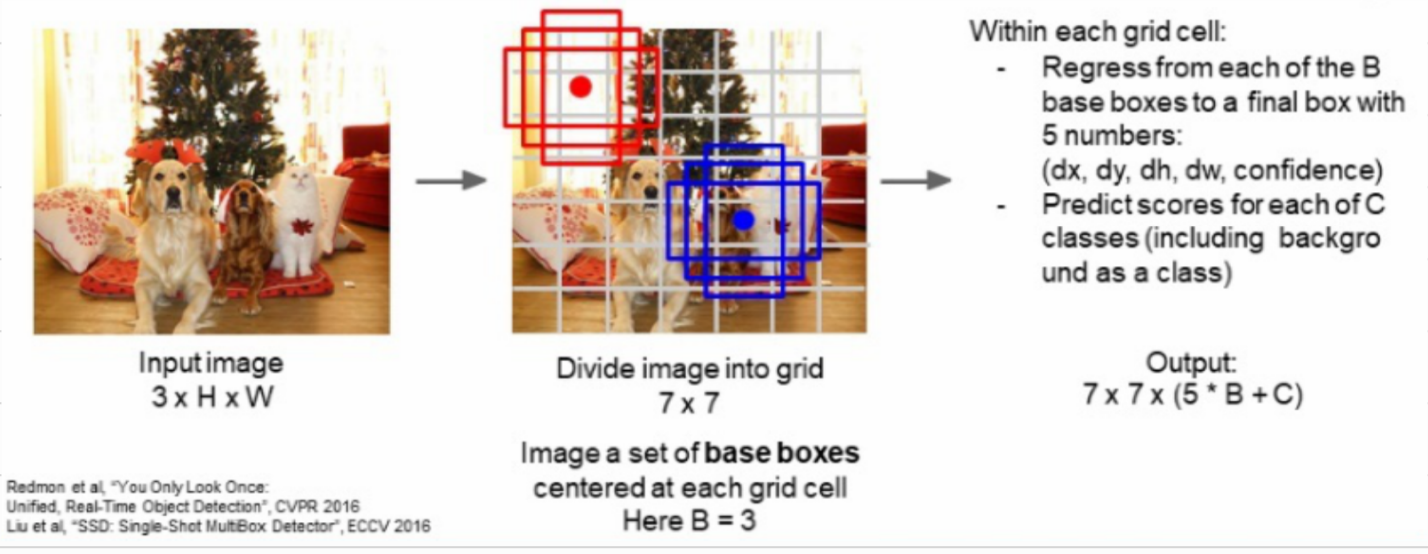
Faster R-CNN



Jointly train 4 losses:
1. RPN classify object / not object (binary)
2. RPN regress box coordinates
3. Final classification score
4. Final box coordinates

feature maps → represents entire high resolution img.

YOLO (you only look once)



Divide input image into coarse grid (ex. 7×7)
Imagine some set of bounding boxes (tall, square, wide)

Dense Captioning

Instance Segmentation

predict the locations and identities of objects in that image, predict a whole segmentation mask for each of objects and predict which pixels in the input image corresponds to each object instance (Hybrid of semantic segmentation & object detection)

* Mask R-CNN