R-CNN / fast R-CNN

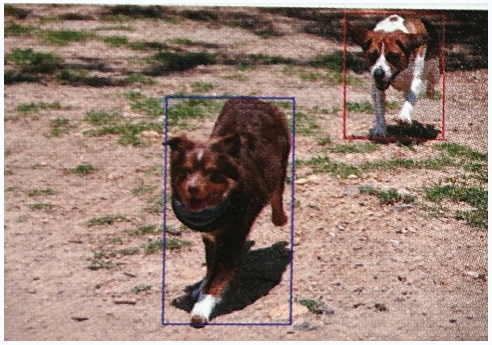
객체 탐지 (object detection)

객체를 탐지하고 영역을 인식

분류 + 지역화 (localization)

물체의 위치를 파악

- 객체 영역을 표현하는 방법



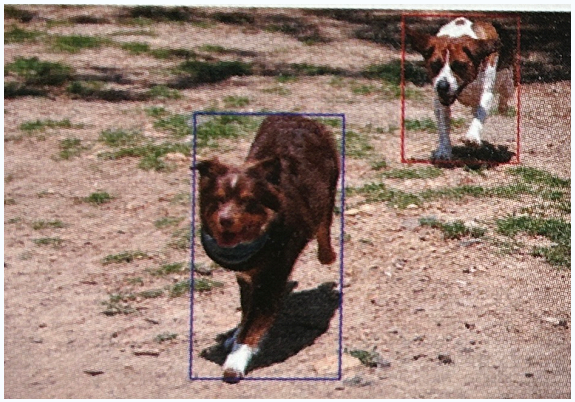
Bounding box

MASK

- 객체 영역을 표현하는 방법

Bounding box

- 직/정사각형 형태로 영역을 간단하게 표현
- 처리속도가 빠름
- 상세한 영역 파악하기 어려움



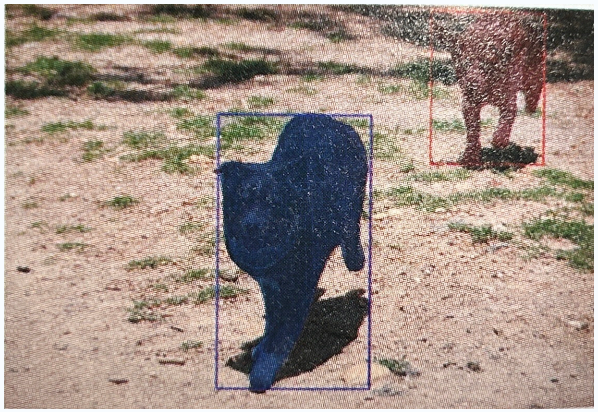
MASK(sematic segmentation, 의미론적 분할)

- 분할 (segmentation)
- 객체와 배경을 픽셀 단위로 분할
- 상세한 모양 확인 가능
- 계산비용이 높음

- 객체 영역을 표현하는 방법

Instance segmentation(객체 분할)

- 객체를 픽셀 단위로 분류하고 경계상사 추출
- 더 정확한 만큼 높은 계산 비용, 더 많은 학습데이터



R-CNN

Region based CNN

규칙기반 알고리즘

SIFT, HOG → CNN

분류 모델(ImageNet)으로
어떻게 객체탐지를 만들어낼 수 있을까?

- To solve
  - DIp로 이미지를 지역화(localizing)함
  - 적은 지역화한 데이터로 high-capacity model 을 학습시킴

R-CNN

Region based CNN

DIp로 이미지를 지역화(localizing)

build a sliding-window detector

이미지에서 물체를 찾기 위해 window의 (크기, 비율)을 임의로 마구 바꿔가면서 모든 영역에 대해서 탐색

Localizing을 위해 regression problem을 활용할 수 있음 하지만 당시 성능이 좋지 않아 폐기

R-CNN

Region based CNN

DIp로 이미지를 지역화(localizing)

build a sliding-window detector
+
CNN

To maintain spatial resolution
Typically, two convolutional and pooling layer

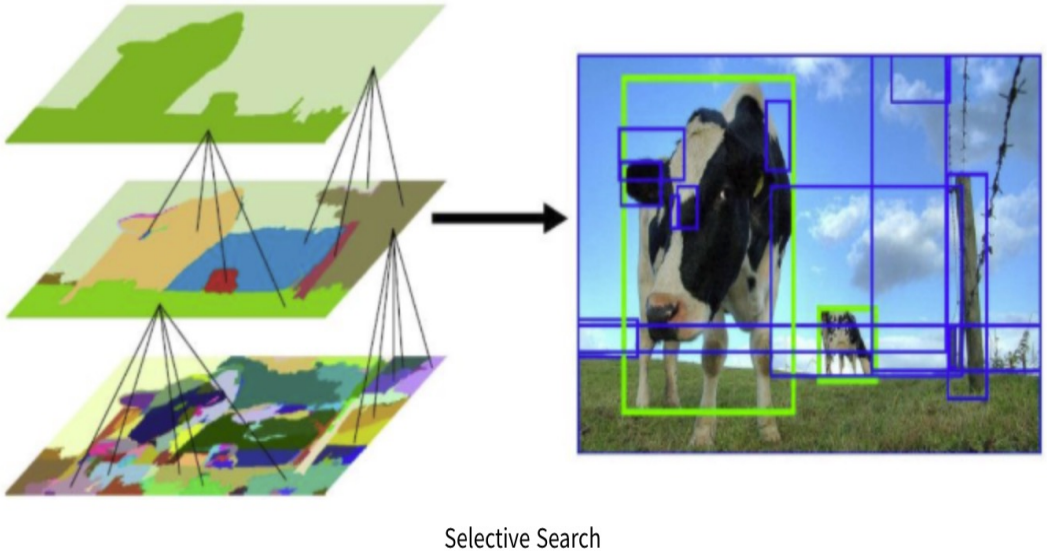
But use five convolutional layers, have very large receptive fields and strides

R-CNN

Region based CNN

Dlp로 이미지를 지역화(localizing)

"Recognition using regions" paradigm
- selective search



Selective Search

- Region proposal methods

Objectness, selective search, category-independent object proposals, constrained parametric min-cuts, multi-scale combinatorial grouping etc…

1. 색상, 질감, 영역크기 등.. 을 이용해 non-object-based segmentation을 수행한다.
이 작업을 통해 좌측 제일 하단 그림과 같이 많은 small segmented areas들을 얻을 수 있다.
2. Bottom-up 방식으로 small segmented areas들을 합쳐서 더 큰 segmented areas들을 만든다.
3. (2)작업을 반복하여 최종적으로 2000개의 region proposal을 생성한다.

R-CNN

Region based CNN

적은 지역화한 데이터로 high-capacity model 학습

➢ Fine tuning

1. ILSVRC20122 dataset을 이용한 pre-train
2. to adapt CNN to the new task(detection)
    1. 1000way classification layer -> randomly initialized (N+1)-way classification layer (1 would be for background)
3. Set learning rate 0.001 (1/10 of the initial pre-training rate) not to clobber the initialization. And starts SGD
4. If IoU < 0.5, overlap as positive and the rest as negative

- Covert the data into a form that fits with the CNN

CNN에 맞는 사이즈가 되도록 bounding box를 넓힘

R-CNN

Region based CNN

적은 지역화한 데이터로 high-capacity model 학습

➢ Fine tuning

1. ILSVRC20122 dataset을 이용한 pre-train
2. to adapt CNN to the new task(detection)
    1. 1000way classification layer -> randomly initialized (N+1)-way classification layer (1 would be for background)
3. If IoU < 0.5, overlap as positive and the rest as negative

- IoU

Intersection over Union, 객체인식모델의 성능평가 도구
요기선 실제 레이블링 된 박스와 CNN으로 얻은 영역을 나눔

R-CNN

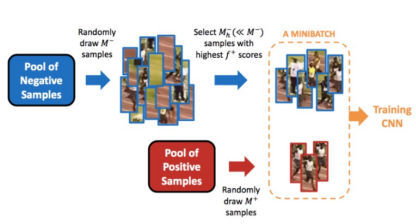
Region based CNN

적은 지역화한 데이터로 high-capacity model 학습

➢ Fine tuning

4. In each SGD iteration,
   - uniformly sample 32 positive windows and 96 background(negative) window which 128 mini-batch
5. Set learning rate 0.001 (1/10 of the initial pre-training rate) not to clobber the initialization. And starts SGD
6. Bias the sampling towards positive windows because they are extremely rare compared to background.

- Hard negative mining

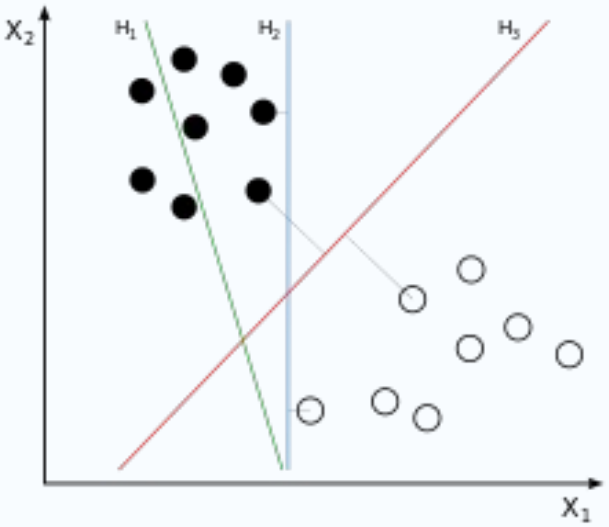


Background >> region 그대로 넣으면 background 로 편향하여 판단할 수 있음. 그래서 한 미니배치에 region 데이터를 더 많이 넣음

R-CNN

Region based CNN

SVM

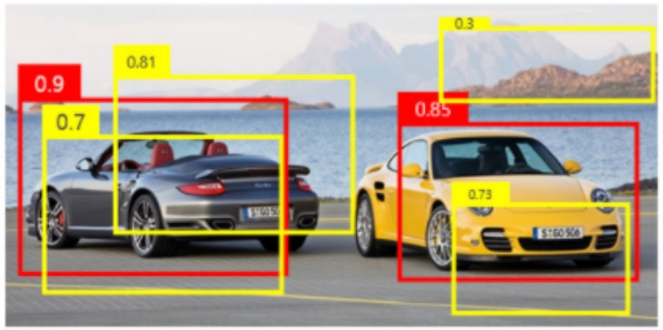


- CNN의 마지막 softmax layer를 사용하지 않고 SVM을 통해 결정(decision)을 제시한다.
  - Softmax layer는 가중치를 공유하기 때문에 클래스마다의 가치가 감소한다.

R-CNN

Region based CNN

- 추가적인 작업
  - Bounding Box Regressor
    - Selective search를 통한 객체 탐지는 위치가 부정확하므로 위치를 조절
  - Non maximum Supression
    - 여러 bounding box중에 적합한 하나를 찾음

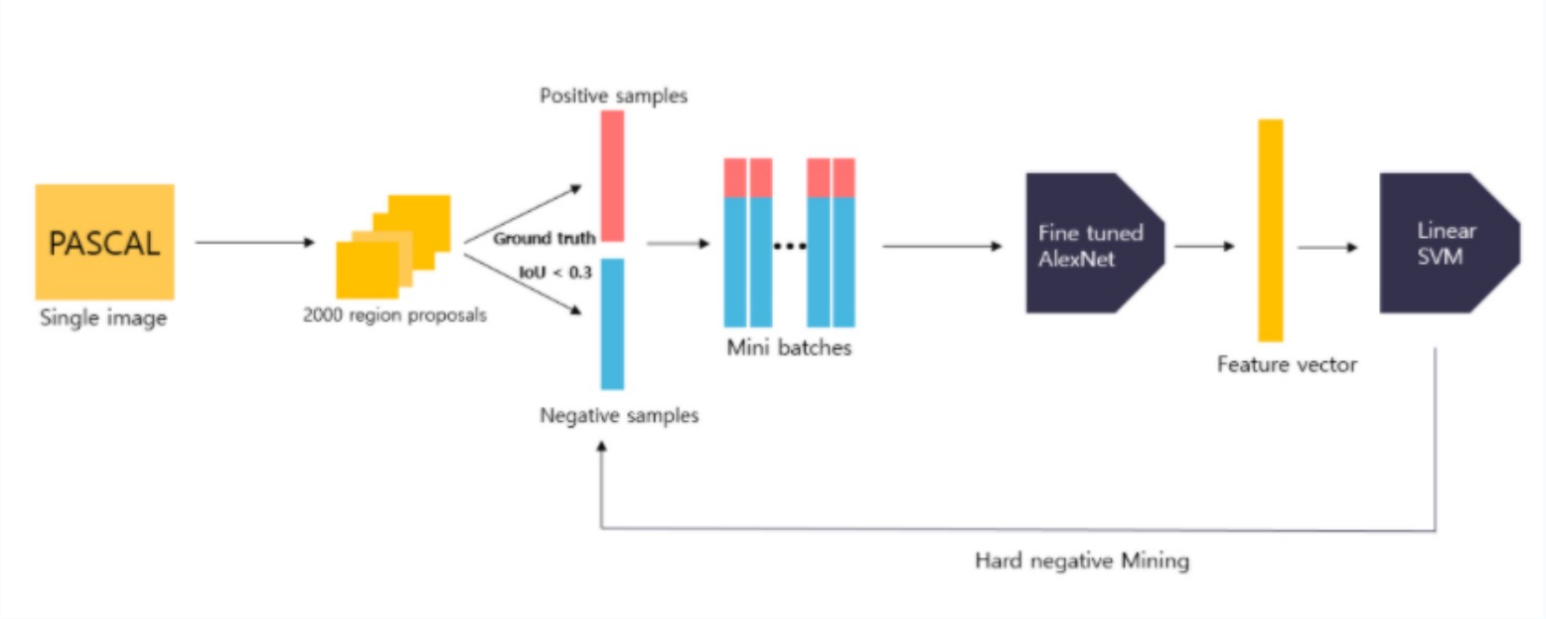


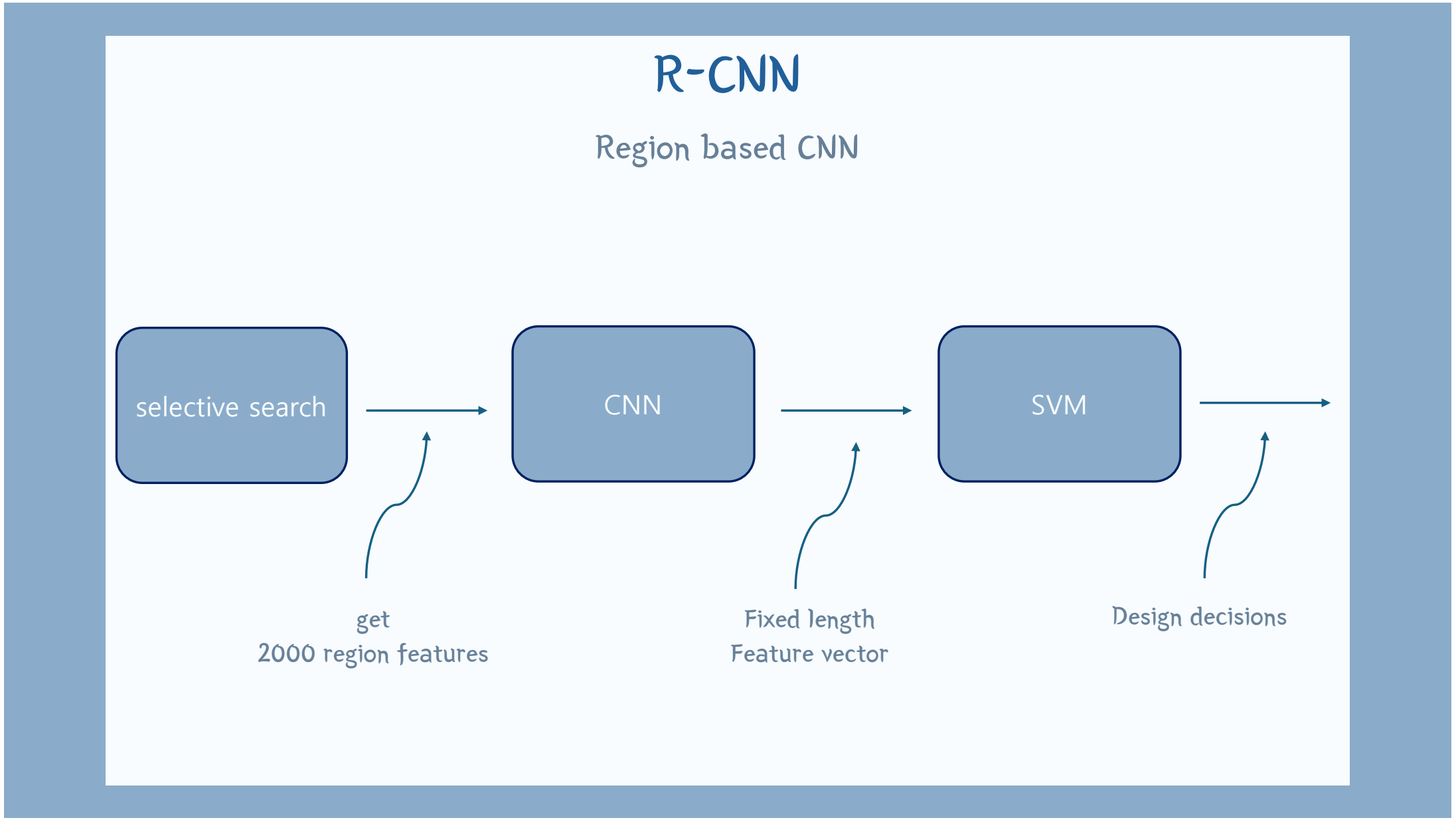
Before Non Maximum Suppression

After Non Maximum Suppression

R-CNN

Region based CNN




R-CNN
Region based CNN
selective search
CNN
SVM
get
2000 region features
Fixed length
Feature vector
Design decisions

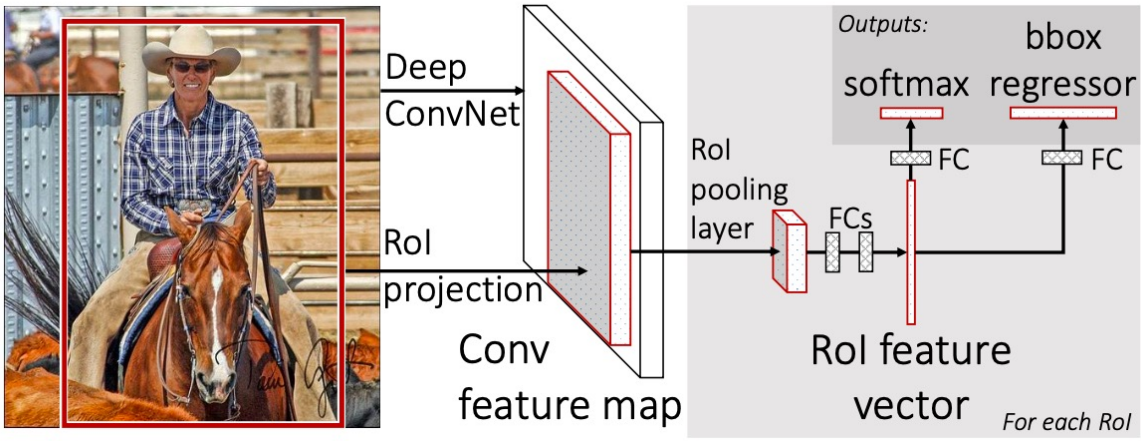
Drawbacks of R-CNN

and SPPnets

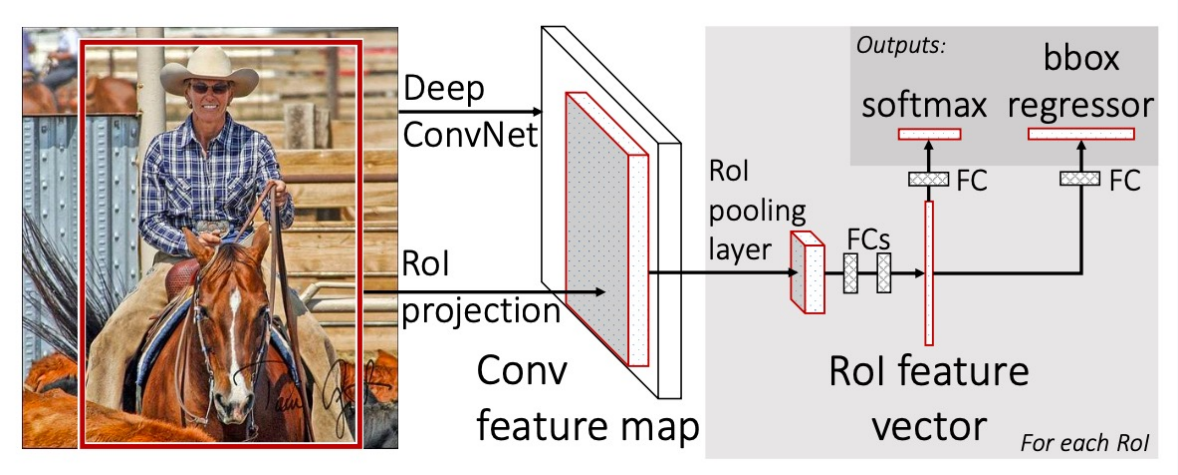
1. Multi-stage pipeline
    - Not familiar to training
2. Expensive space and time
    - To move on next stage data must be written to disk
3. Slow
    - Forward pass for each object proposal

Fast R-CNN

1. single-stage pipeline
   - Multi-task loss
2. No disk storage is required
3. Update all network layers (drawback of SPPnets)

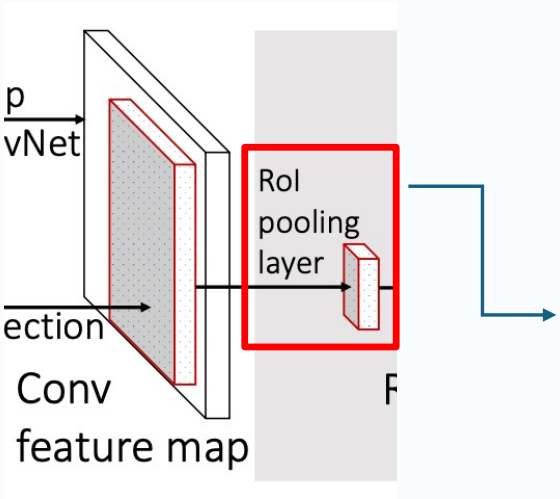


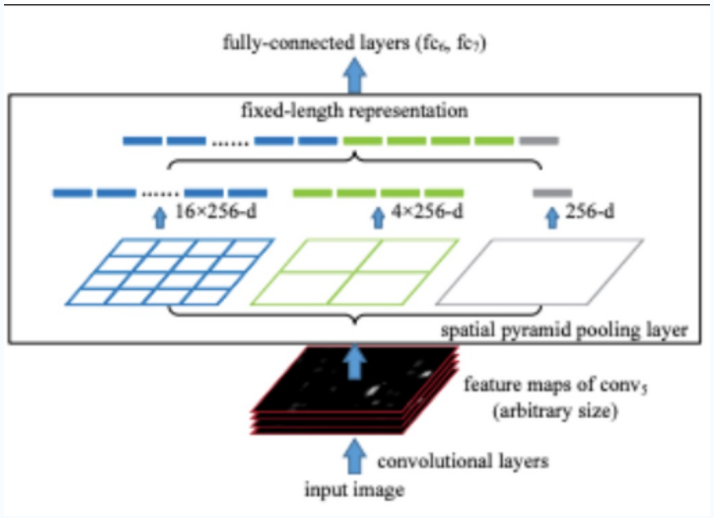
Fast R-CNN



- Input an **entire image** and set of **object proposals**
  - for single-stage
  - Object proposals are from selective search
- The CNN is pre-trained VGG16

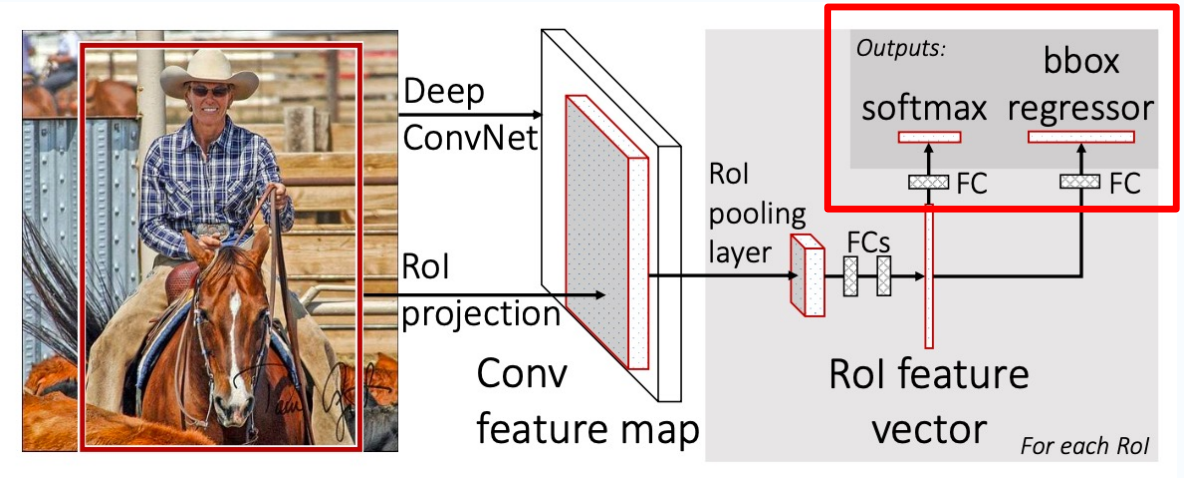
Fast R-CNN





- ROI pooling layer (Spatial pyramid pooling)
  - 이미지를 (1,1) (2,2) (3,3) … 으로 분할 (요기선 (7, 7)로만 분할)
  - 분할한 영역에서 max pooling 진행
  - 위 방법을 n번 반복 (이 모델에선 1번만 진행)
  - get fixed length feature vector

Fast R-CNN



- Replace last layer of fully connected layer
  - One is softmax with (N+1) classes
  - Other one is box regressor for Bounding Box

Fast R-CNN

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

$p = (p_0, ..., p_k)$: K+1개의 class score

u : class ground truth

$t^u = (t^u_x, t^u_y, t^u_w, t^u_h)$: u 클래스의 bounding box 좌표를 조정하는 값

(R-CNN에서 regressor output처럼 해당 좌표가 아니라 ground truth 좌표를 계산할 수 있게 해주는 값)

$v = (v_x, v_y, v_w, v_h)$: bounding box 좌표값의 ground truth

λ: hyperparameter(논문에서 $\lambda = 1$)

L_{cls}는 cross-entropy error로 계산된다.

L_{loc}는 다음과 같이 정의된다.

$L_{loc}(t^u, v) = \Sigma_{i \in \{x,y,w,h\}} smooth_{L1}(t^u_i - v_i)$

$smooth_{L1}(x)$ = $0.5x^2$ if $|x| < 1$

$smooth_{L1}(x)$ = $|x| - 0.5$ $otherwise$

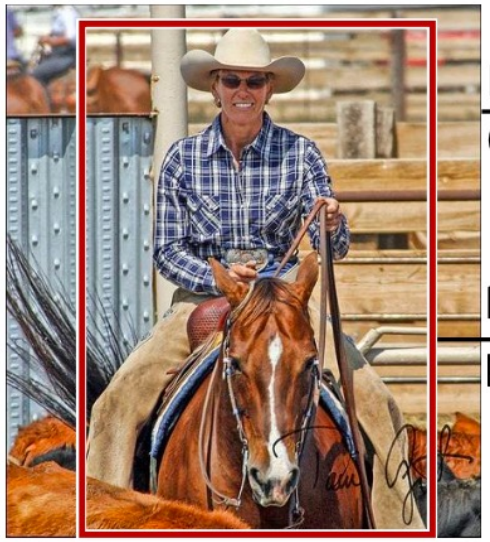
L_1 loss가 outlier에 대해 L_2 loss 보다 덜 민감하기 때문에 $smooth_{L_1}$을 사용한다.

$[u \geq 1]$는 indicator function으로 해당 클래스에 속할때만 loss를 계산하게 한다.

- Softmax와 box regressor를 동시에 학습

Fast R-CNN

입력이 2개



Deep ConvNet

RoI projection

Conv feature map

RoI pooling layer

FCs

FC

FC

Outputs:

softmax

bbox regressor

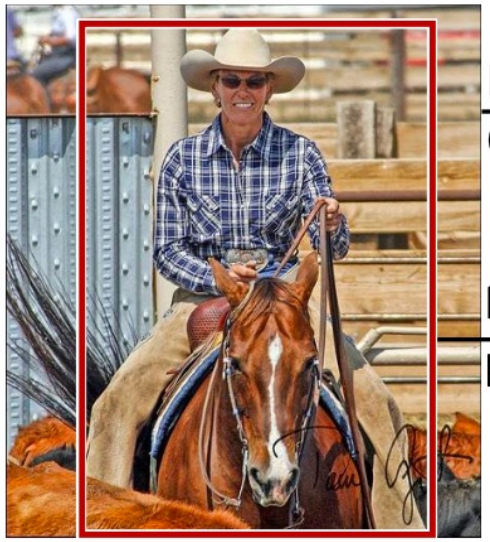
RoI feature vector

For each RoI

Fast R-CNN

입력이 2개

1. 전체 이미지
2. 객체 집합



Deep ConvNet

RoI projection

Conv feature map

RoI pooling layer

FCs

FC

FC

Outputs:

softmax

bbox regressor

RoI feature vector

For each RoI

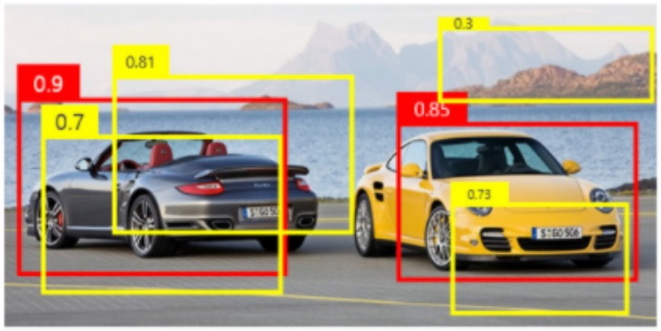
Class predict.

Points of box

크기를 똑같이

Fast R-CNN

- 추가적인 작업
  - Non maximum Supression
    - 여러 bounding box중에 적합한 하나를 찾음



Before Non Maximum Suppression

After Non Maximum Suppression

Faster R-CNN

- Too slow selective search -> RPN(region proposal Network)
  - Region을 찾는 신경망을 만든다.

➢ Mask R-CNN

➢ Cascade R-CNN