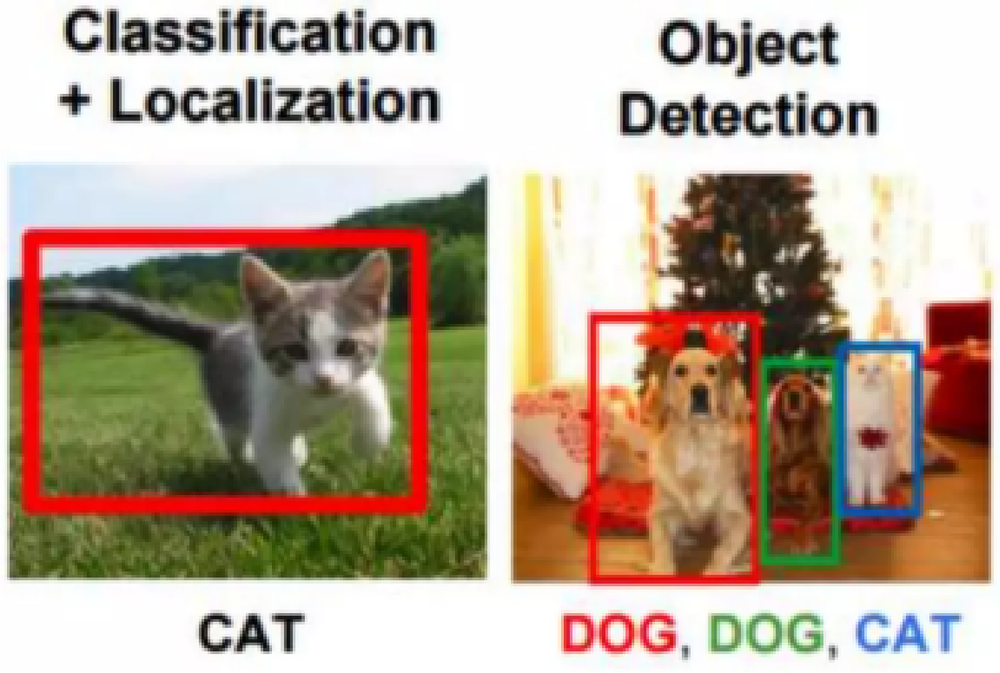
Faster R-CNN

Towards Real-Time Object Detection
with Region Proposal Networks

Introduction



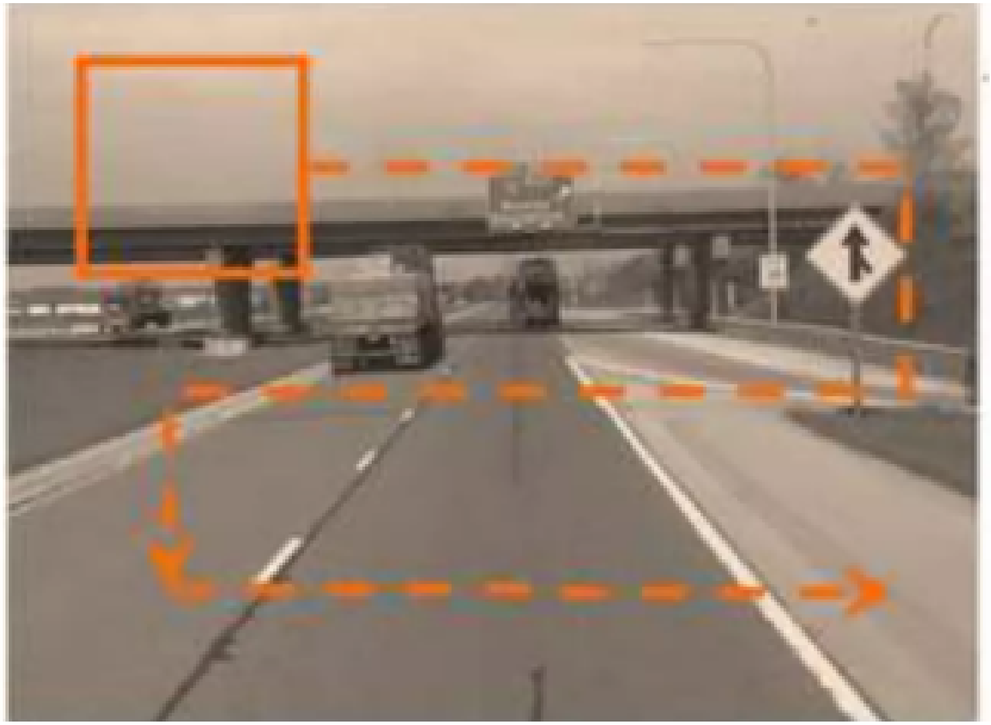
Object Detection

입력 영상내에 존재하는 모든 카테고리에 대해
classification 과 localization을 수행

Object 0 ~ N 개

Naïve Approach

Sliding window apporach



모든 크기의 영역 (different scale & ratio)
에 대해 sliding window 방식으로 이미지를 모두 탐색하면서 classification을 수행

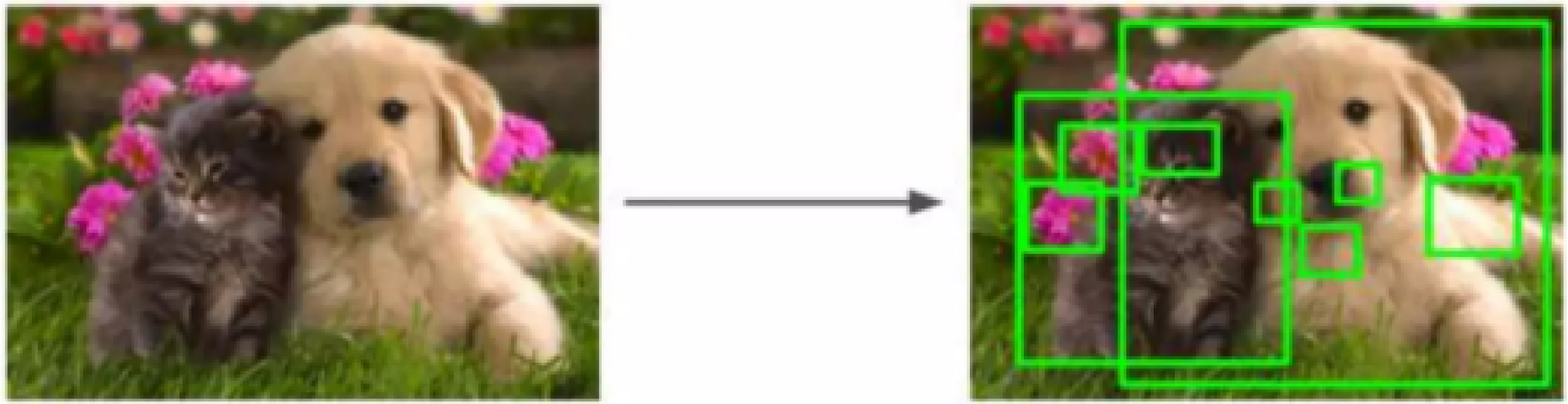
비효율적

Region Proposals

Region proposal algorithm

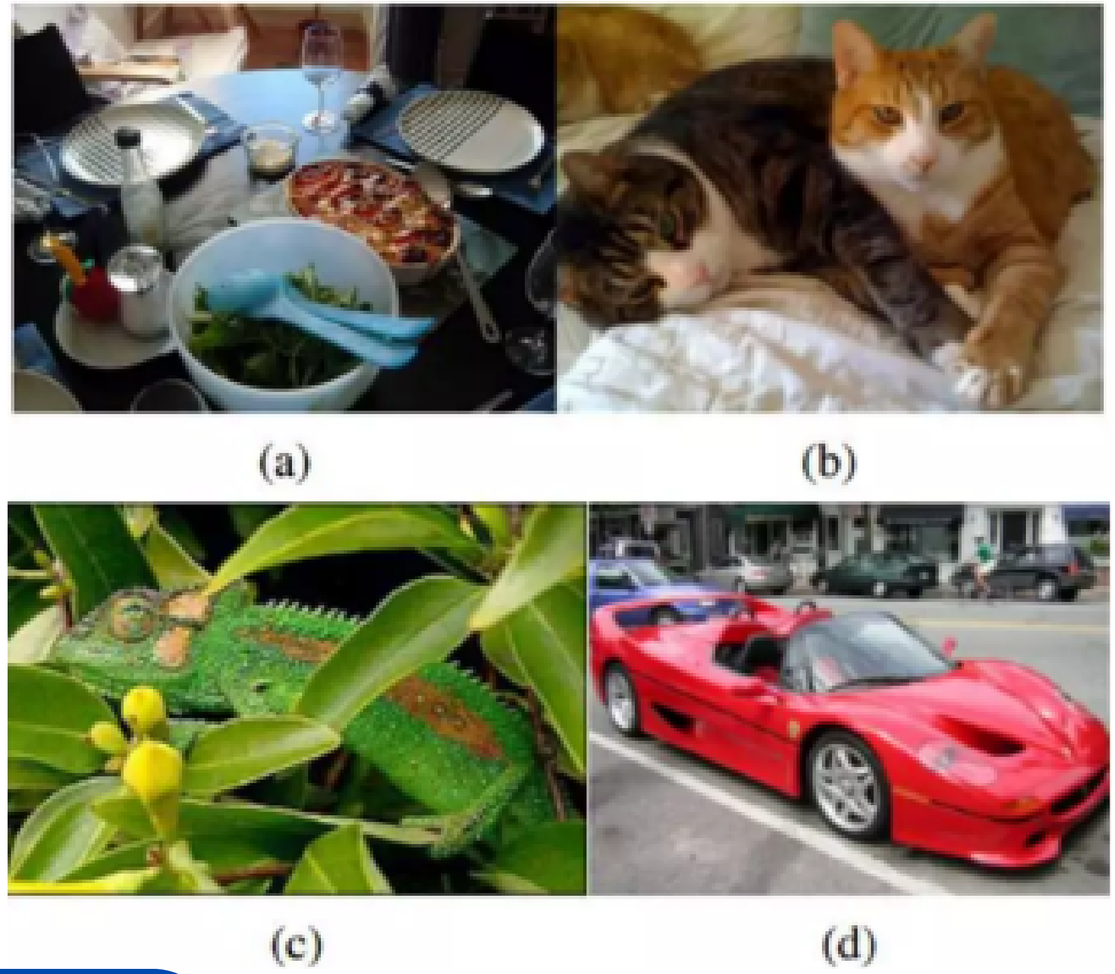
물체가 있을 법한 영역을 찾아내는 알고리즘

Search space를 줄임



Selective Search

Goal



영상은 계층적 구조를 가지므로 적절한 알고리즘을 사용하여 크기에 상관없이 대상을 찾아낸다

컬러, 무늬, 질감, 명암 등 다양한 기준에 따라 segmentation

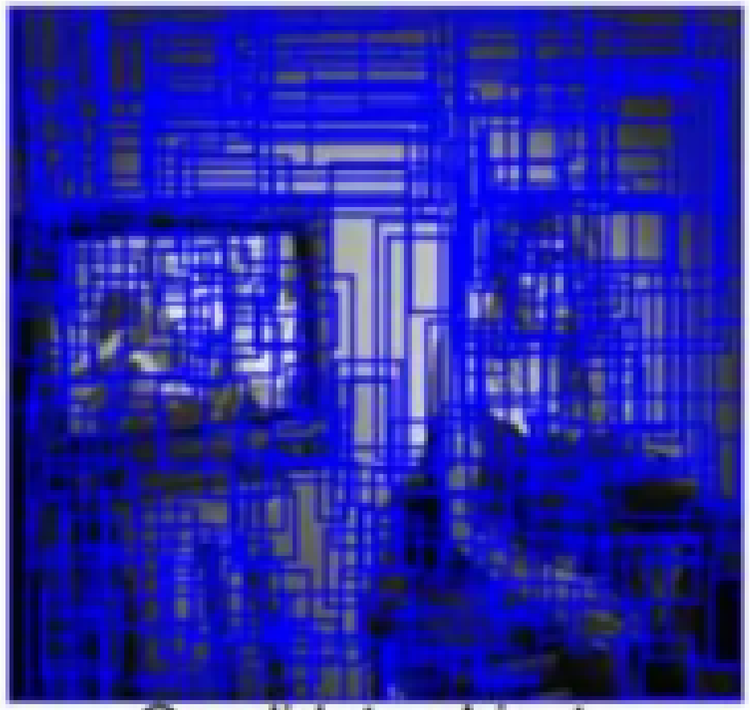
Selective Search

sub - segmentation

각각의 객체가 1개의 영역에 할당이 될 수 있도록 많은초기 영역을 생성한다.

Input Image

Segmentation

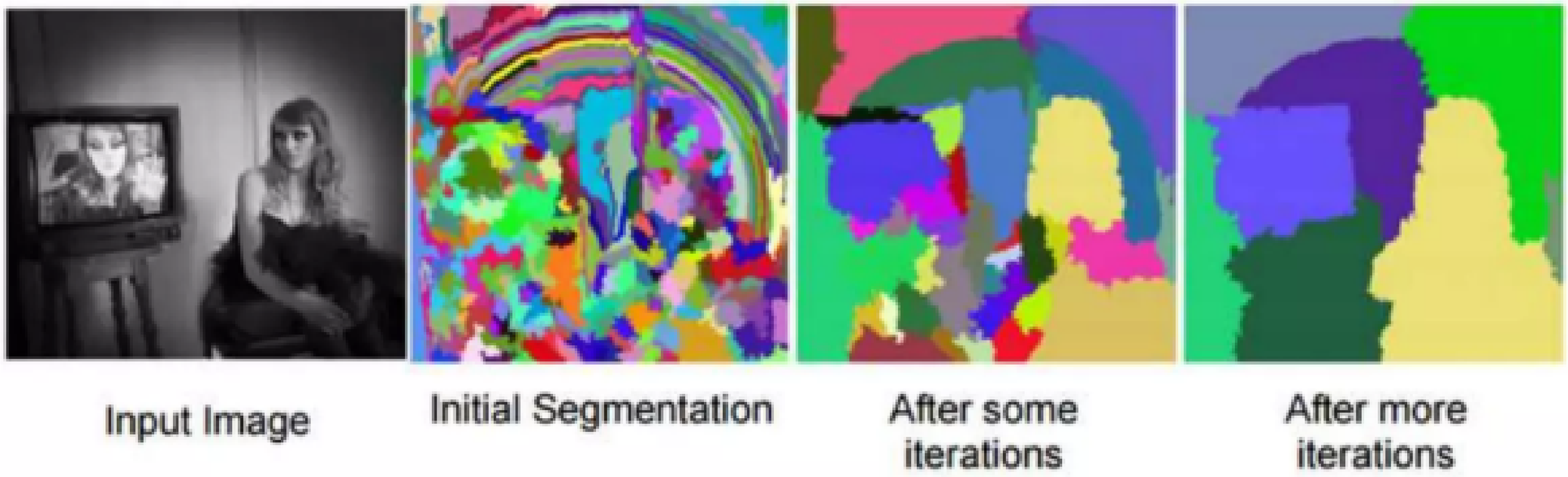


Candidate objects

Selective Search

Greedy algorithm

여러 영역으로부터 가장 비슷한 영역을 고르고,
이것들을 좀 더 큰 영역으로 통합을 하며, 1개의 영역이 남을 때 까지 반복한다.



초기의 작은 영역들이 유사도에 따라 점점 통합이 되는 것을 확인할 수 있다.

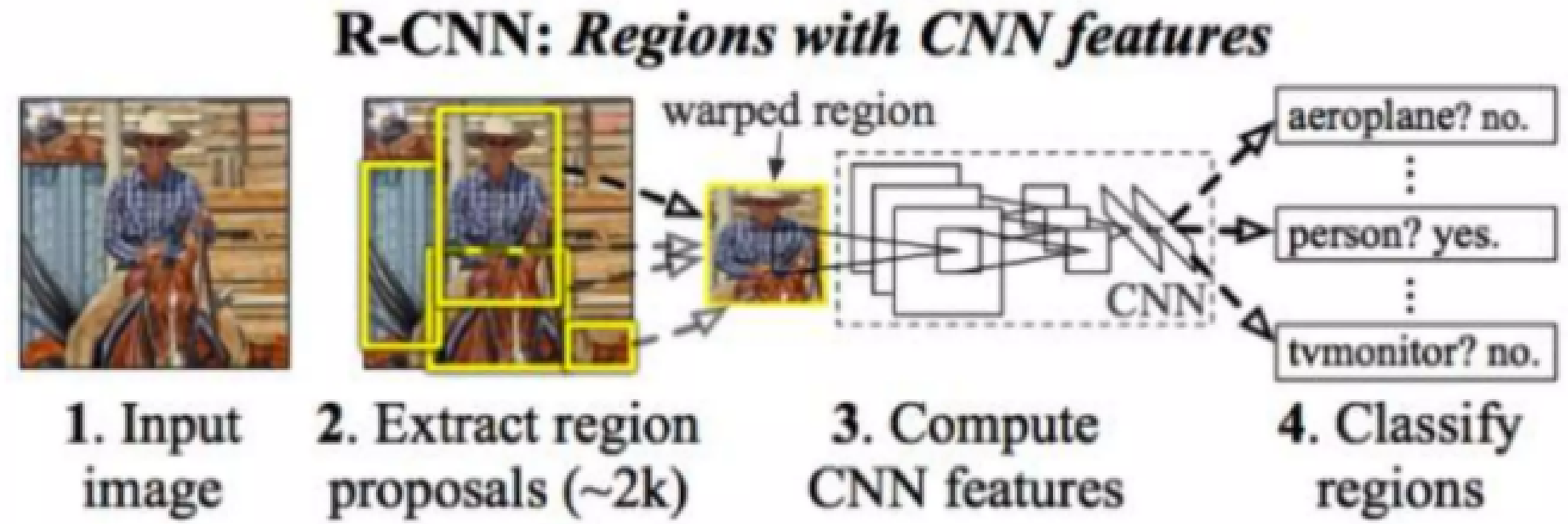
Selective Search

Region of Interest (ROI)

통합된 영역들을 바탕으로 후보 영역들을 만들어 낸다.
이 과정을 통합적으로 보여주는 과정은 아래와 같다.

R - CNN

Region Proposal + Convolutional Neural Network (CNN)



R - CNN

문제점

Localization 성능이 취약

CNN이 positional invariance한 특성을 가지고 있음

Region proposal 내에서 물체가 중앙이 아닌 다른곳에
위치하고 있어도 CNN이 높은 classification score을 에측하기 때문

R - CNN

해결

bounding-box regression

위치보정

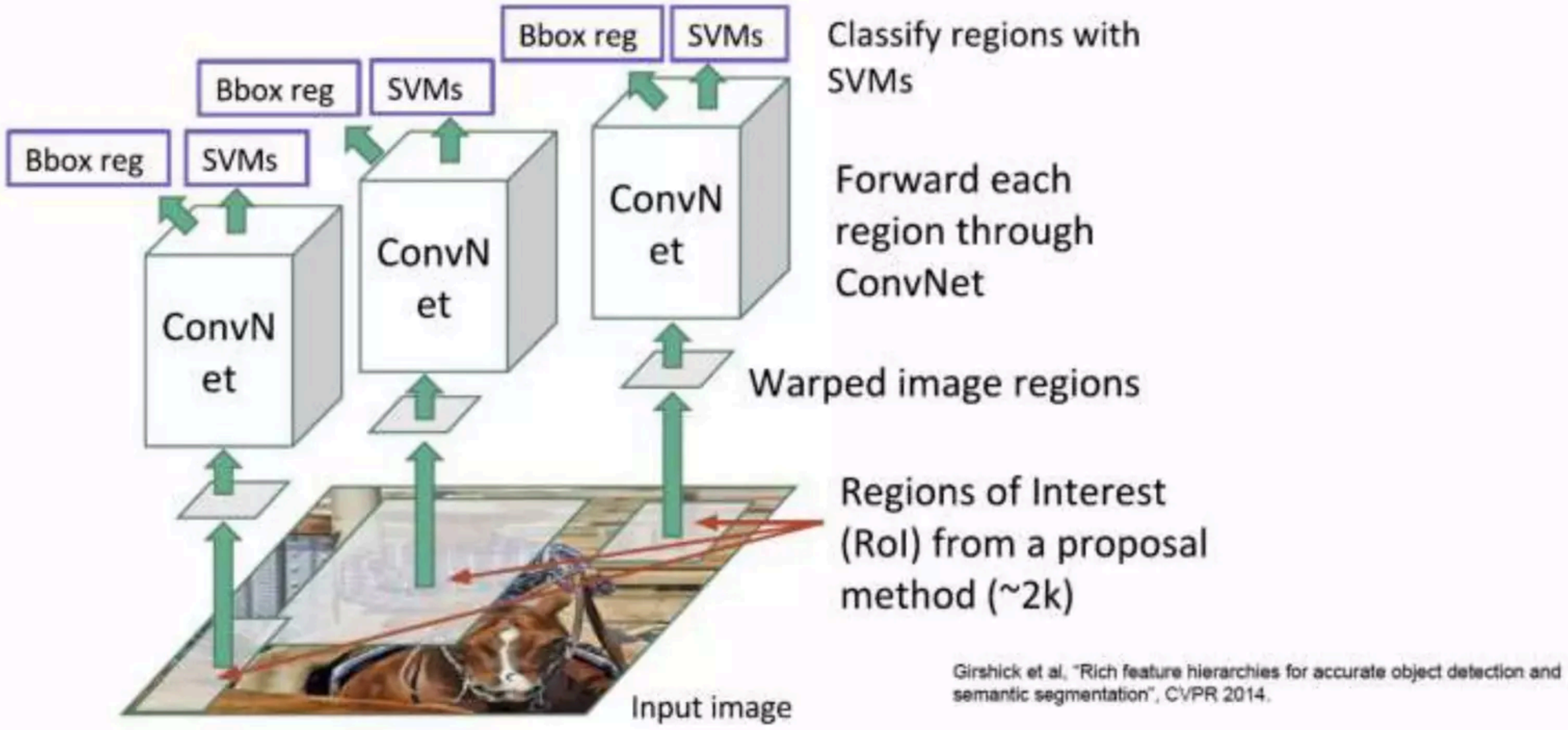
region proposal P, 정답위치 G

P를 G로 mapping 할 수 있는 변환을 학습

Kind of a refinement step

R - CNN

Summary



R - CNN

R - CNN 과 타 알고리즘과의 성능 비교

VOC 2010 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
DPM v5 [20]†	49.2	53.8	13.1	15.3	35.5	53.4	49.7	27.0	17.2	28.8	14.7	17.8	46.4	51.2	47.7	10.8	34.2	20.7	43.8	38.3	33.4
UVA [39]	56.2	42.4	15.3	12.6	21.8	49.3	36.8	46.1	12.9	32.1	30.0	36.5	43.5	52.9	32.9	15.3	41.1	31.8	47.0	44.8	35.1
Regionlets [41]	65.0	48.9	25.9	24.6	24.5	56.1	54.5	51.2	17.0	28.9	30.2	35.8	40.2	55.7	43.5	14.3	43.9	32.6	54.0	45.9	39.7
SegDPM [18]†	61.4	53.4	25.6	25.2	35.5	51.7	50.6	50.8	19.3	33.8	26.8	40.4	48.3	54.4	47.1	14.8	38.7	35.0	52.8	43.1	40.4
R-CNN	67.1	64.1	46.7	32.0	30.5	56.4	57.2	65.9	27.0	47.3	40.9	66.6	57.8	65.9	53.6	26.7	56.5	38.1	52.8	50.2	50.2
R-CNN BB	**71.8**	**65.8**	**53.0**	**36.8**	**35.9**	**59.7**	**60.0**	**69.9**	**27.9**	**50.6**	**41.4**	**70.0**	**62.0**	**69.0**	**58.1**	**29.5**	**59.4**	**39.3**	**61.2**	**52.4**	**53.7**

Table 1: Detection average precision (%) on VOC 2010 test. R-CNN is most directly comparable to UVA and Regionlets since all methods use selective search region proposals. Bounding-box regression (BB) is described in Section C. At publication time, SegDPM was the top-performer on the PASCAL VOC leaderboard. †DPM and SegDPM use context rescoring not used by the other methods.

R-CNN

문제점

- Test 속도가 느림
  - 모든 region proposal에 대해 CNN을 계산
  - GPU(K40)에서 13s / image
- SVM과 bounding box regressor의 학습이 분리
  - CNN 학습 과정 후, SVM과 bounding box regressor의 학습이 나중에 진행됨 (post-hoc)
- 학습 과정이 복잡함
  - 다단계 training pipeline
  - GPU(K40)에서 84시간 (VOC2007, 5000 images)

Fast R - CNN

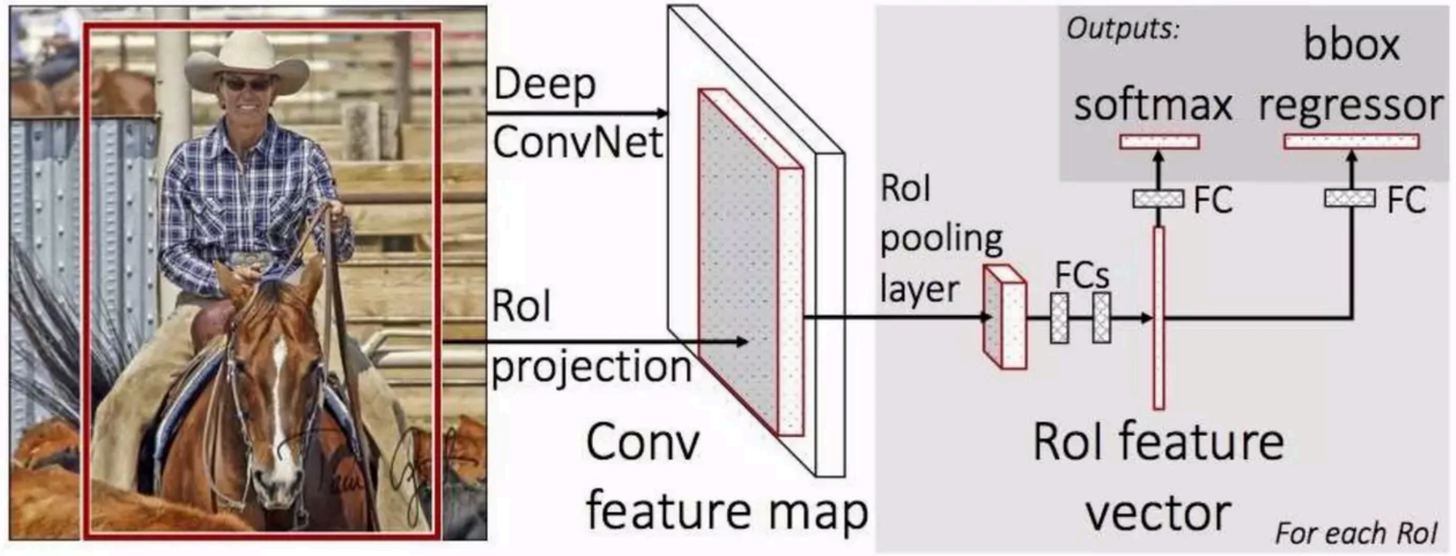
특징: 속도개선

같은 image의 proposal들이 convolution layer를 공유

ROI Pooling 도입

전체 network가 End-to-end로 한번에 학습

~160x faster than R - CNN



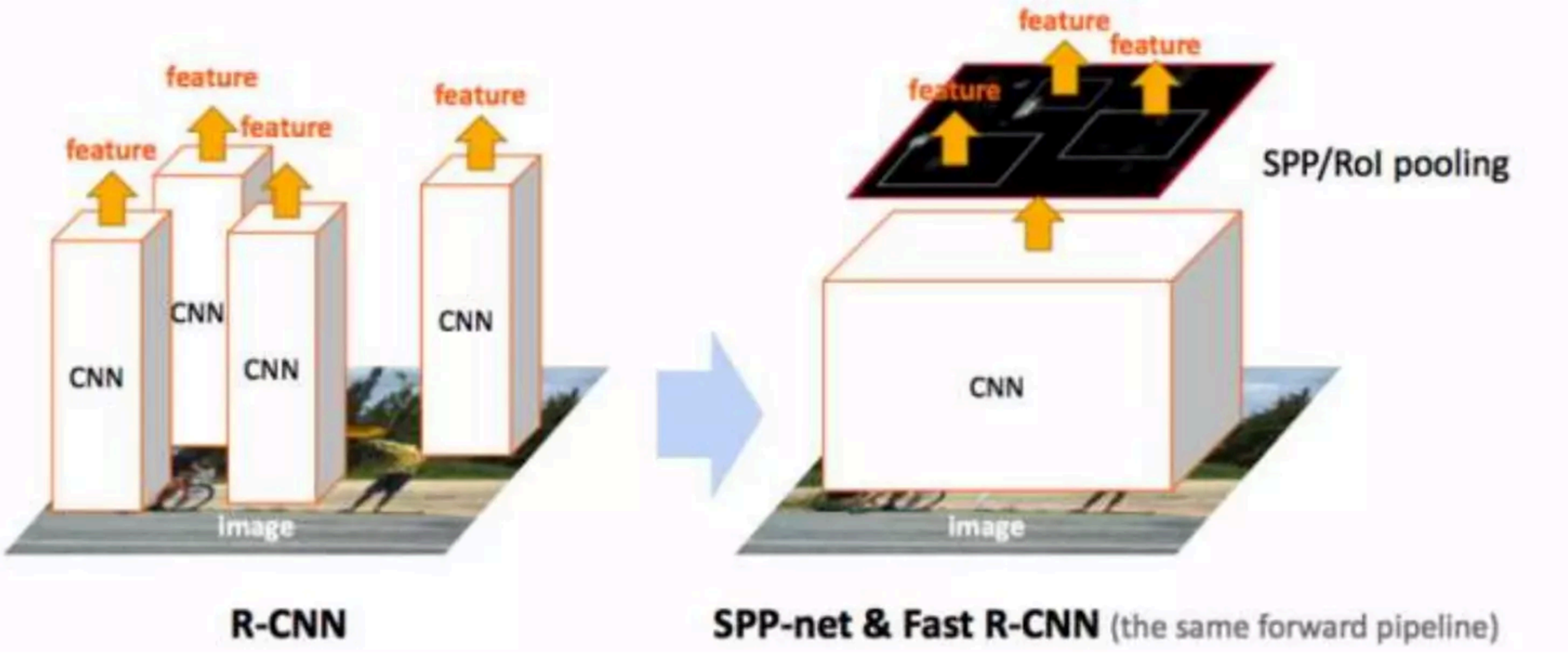
Fast R - CNN

R-CNN 과 feature 추출방법 비교

R-CNN은 각각 region마다 crop하여 cnn연산

Fast R-CNN은 cropping을 image level이 아닌 feature map level에서 수행

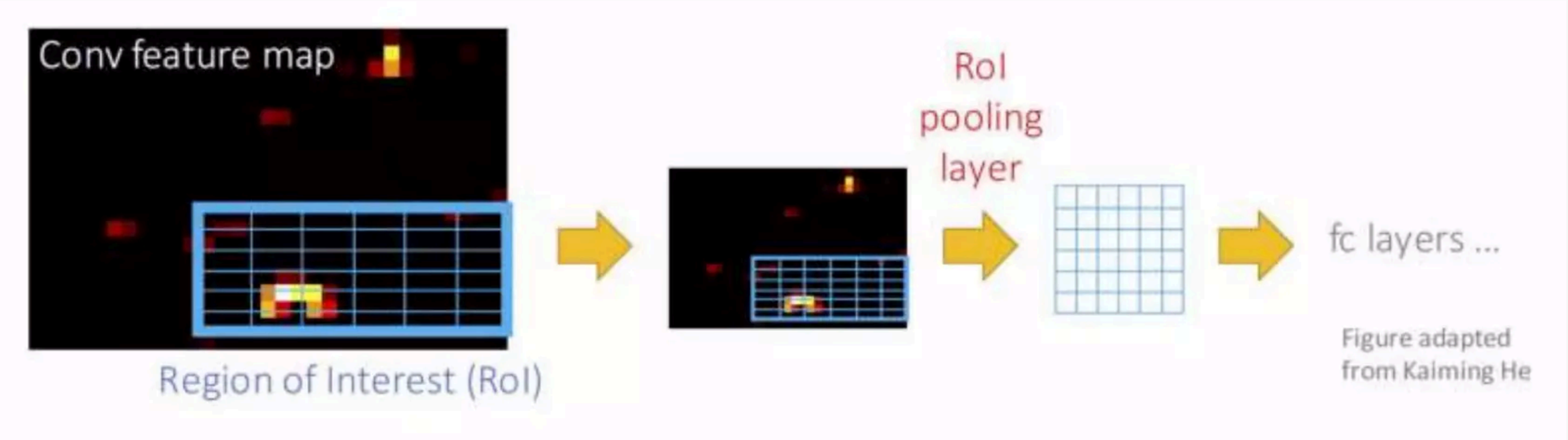
CNN 연산 2,000번 ~> 1번



Fast R - CNN

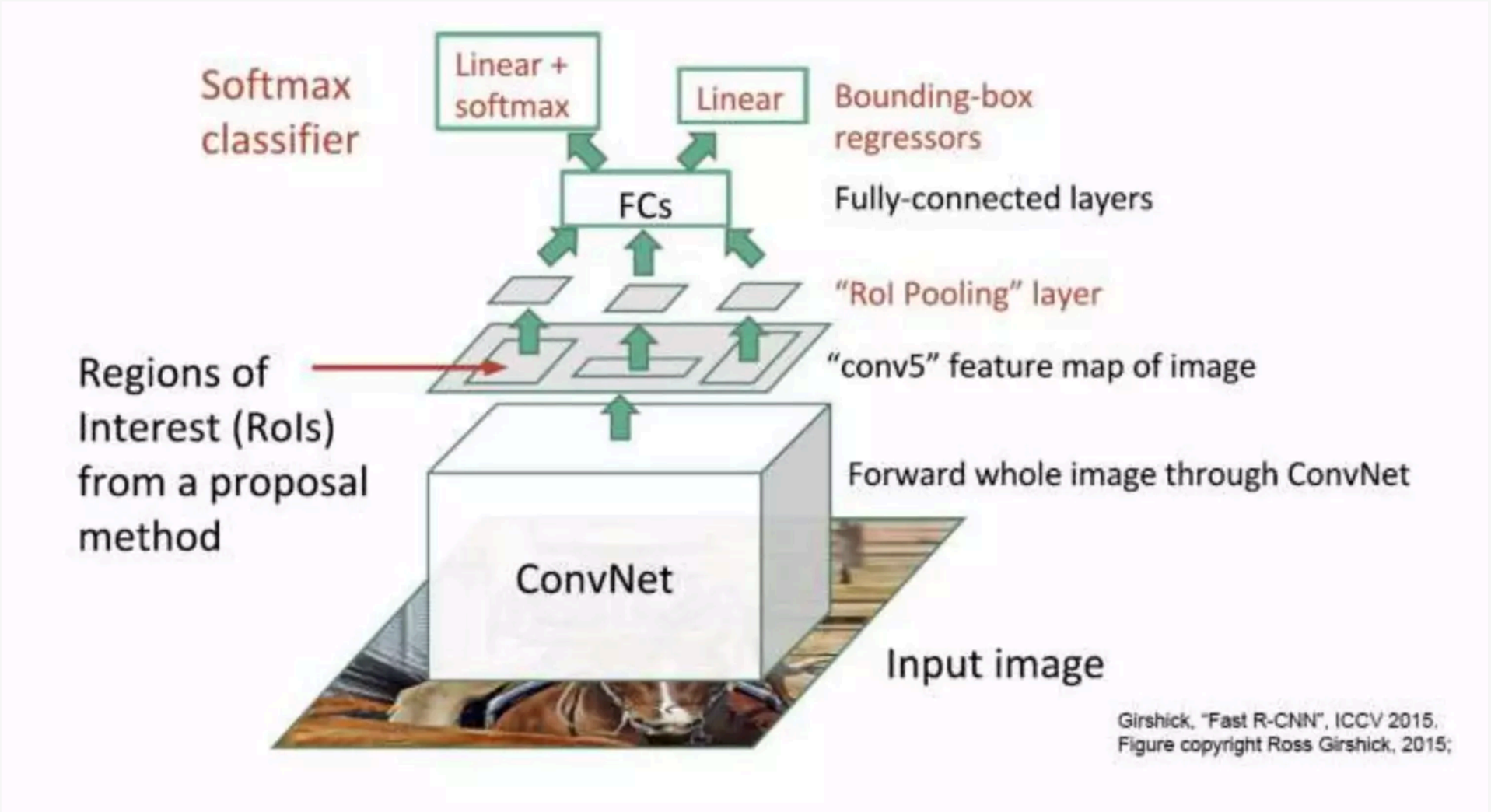
ROI pooling

- Selective Search 통해 찾은 ROI 영역
- 전체 image 를 convolution해 나온 feature map에서 ROI 영역만 pooling하여 fc layer에 넣는다.



Fast R - CNN

전체 구조



Fast R - CNN

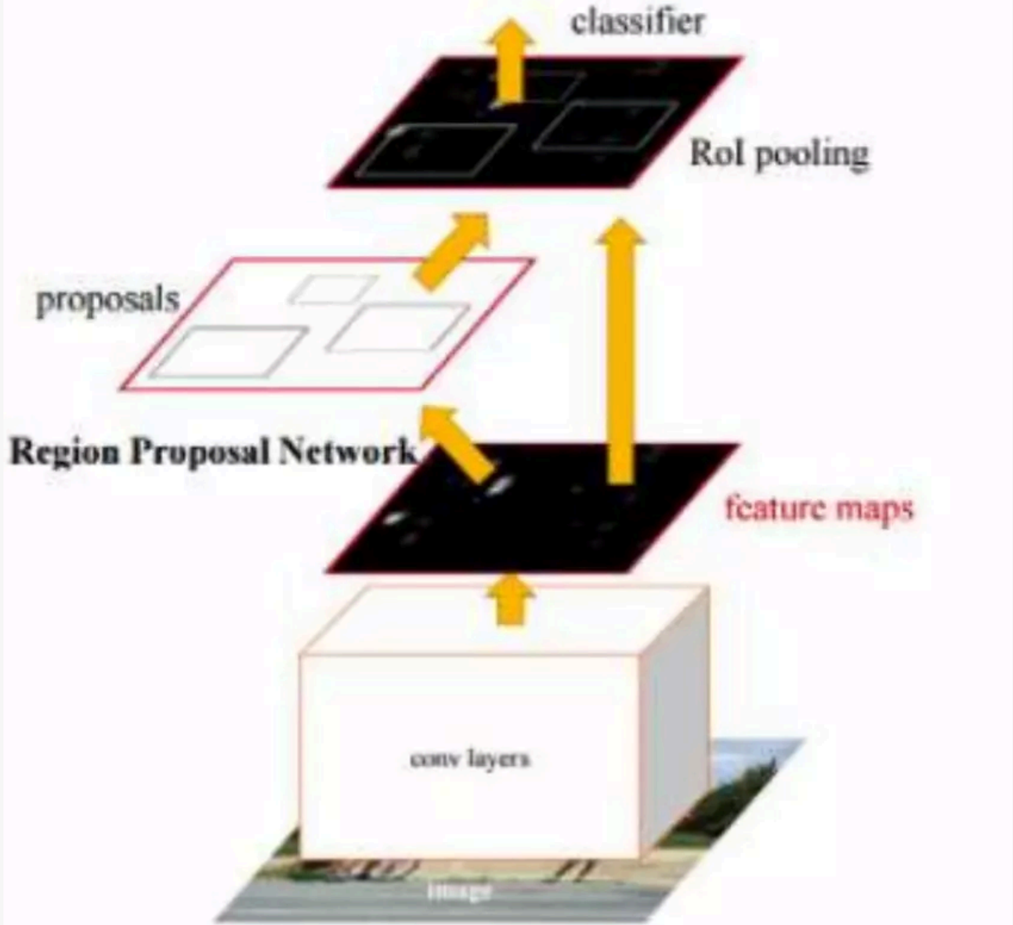
문제점

- Region proposal 계산이 NN밖에서 일어난다.
- Region proposal (Selective Search) 이 전체 성능의 bottleneck이 된다.
- SS가 느린 이유중 하나는 GPU가 아니라 CPU로 계산하기 때문.
  => GPU 연산을 사용하자 (Faster R - CNN)

Fast R - CNN

RPN : Region Proposal Network

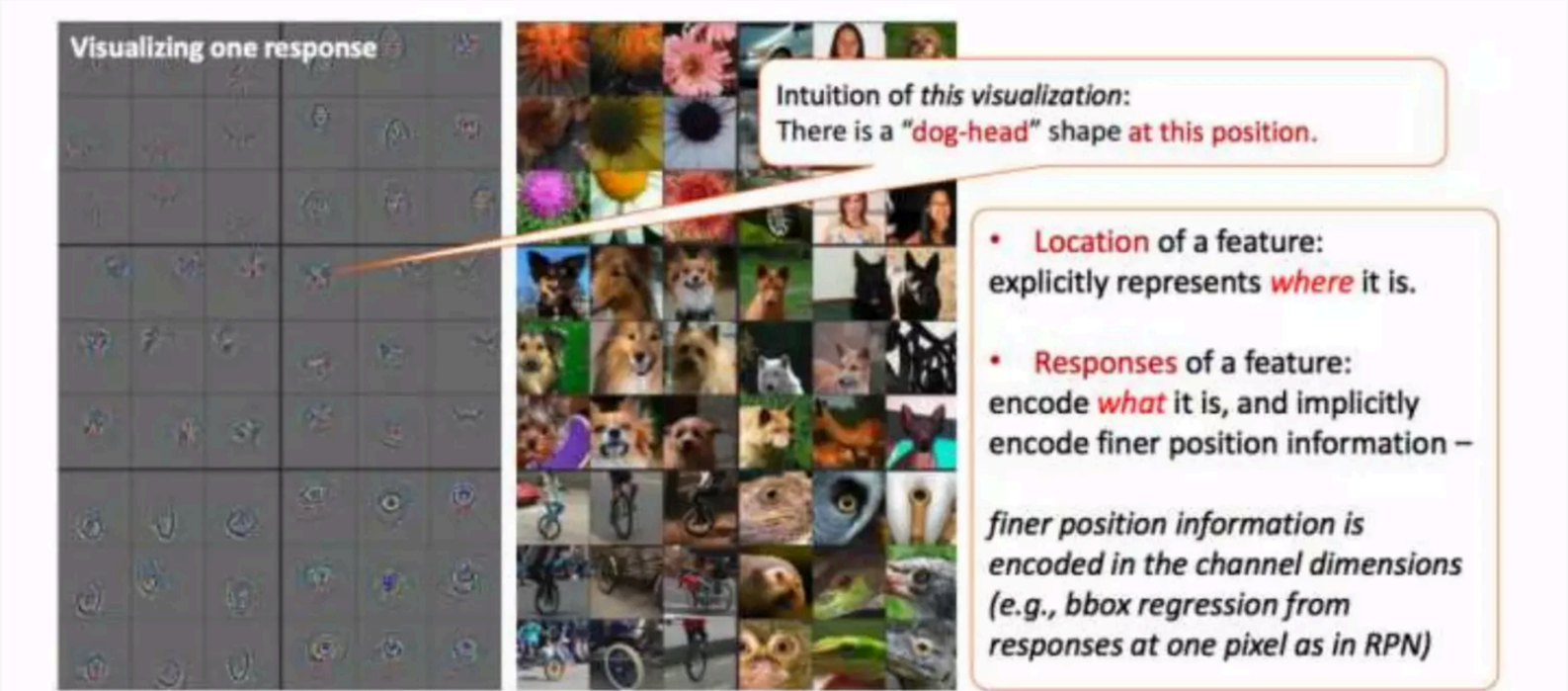
- Fast R - CNN 에서 Bottleneck 이었던 region proposal 생성을 CNN 내부에 설계



Faster R - CNN

Region Proposals from Feature Map

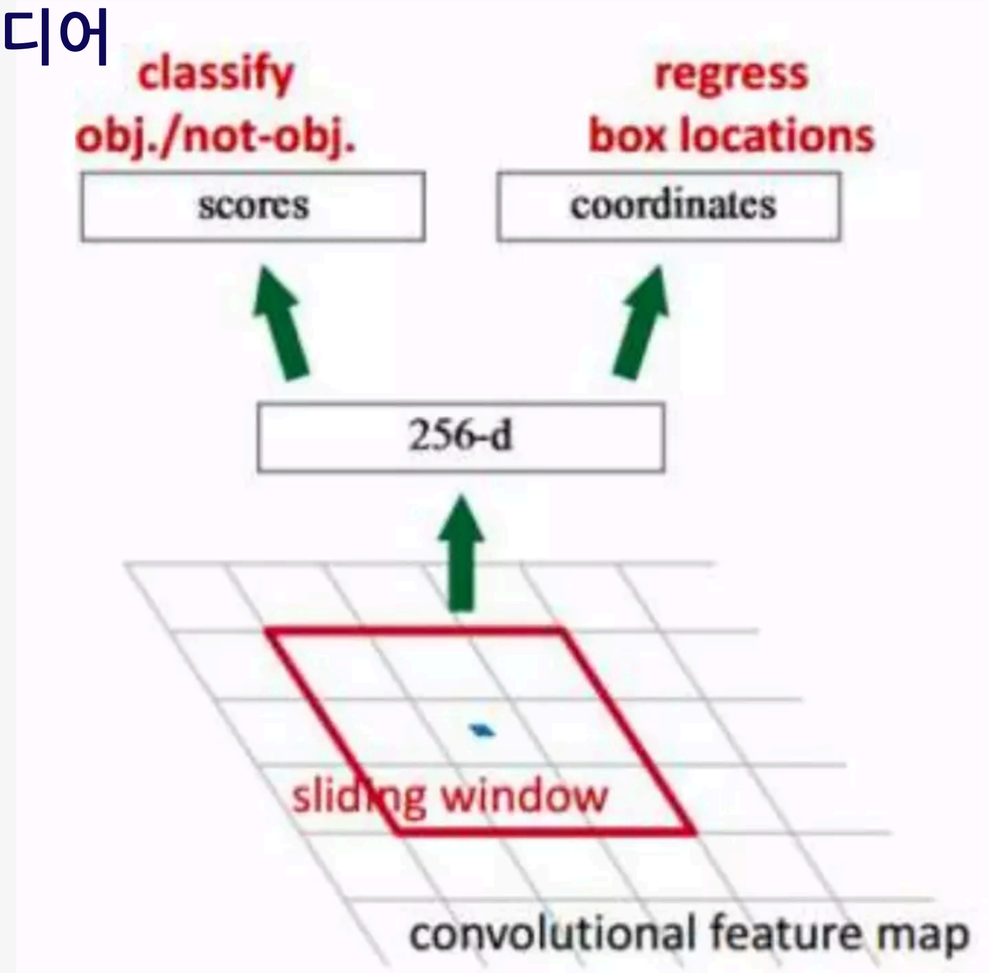
- Classification 또는 detection을 수행하는 CNN이 가진 feature map이 대략적인 물체의 위치정보를 가지고 있다면,
  이를 잘 학습하면 물체의 대략적인 위치를 잡아낼 수 있을것
- feature map activation 시각화



Faster R - CNN

RPN: Region Proposal Network

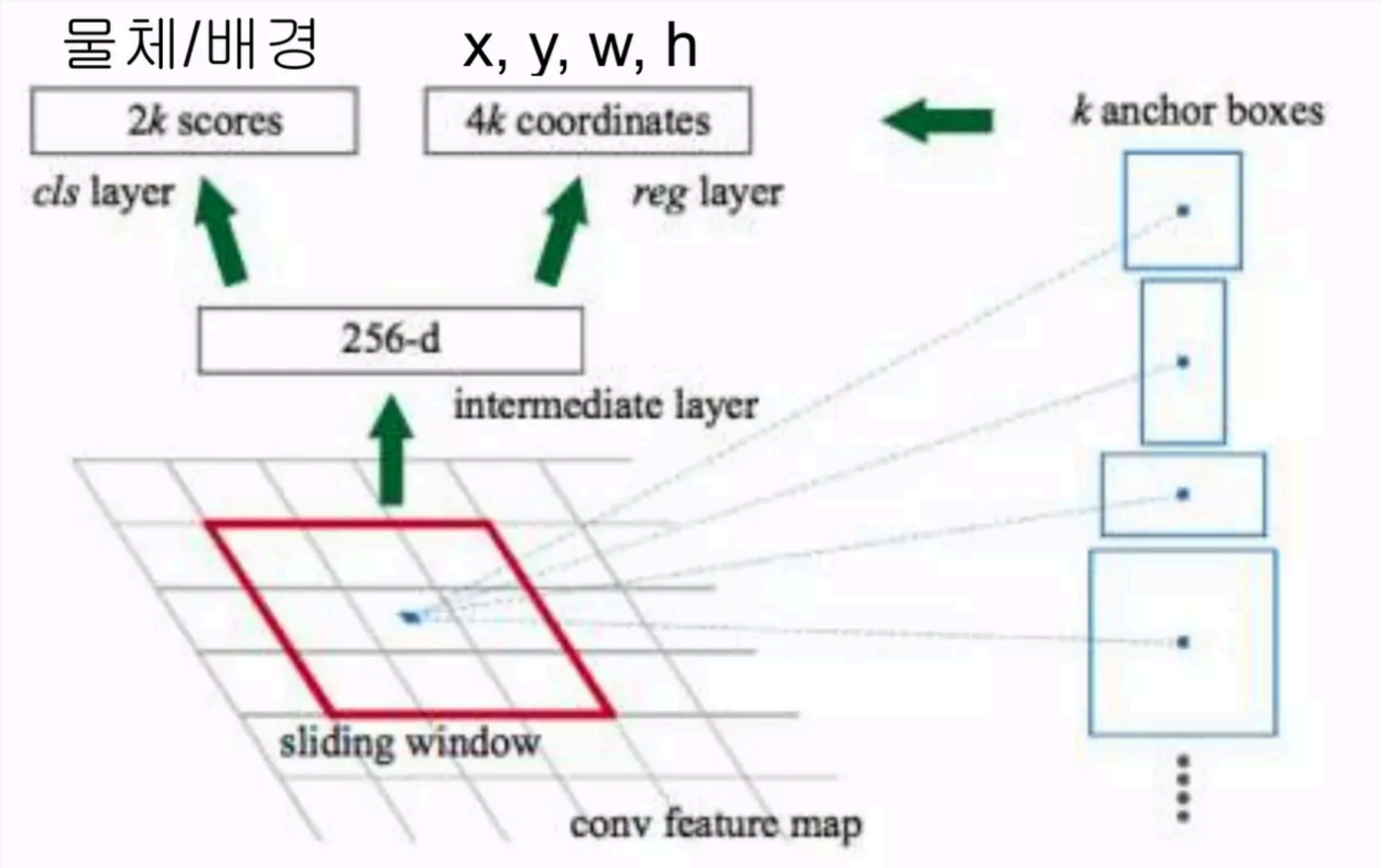
- feature map 정보를 활용해 물체가 존재하는 위치를 출력으로 가지는 네트워크인 region proposal network(RPN)를 학습해보자는 아이디어
- 입력: N x N 크기의 작은 window 영역
- 출력: binary classification
- Bounding - box regression 또한 사용



Faster R - CNN

RPN: Region Proposal Network

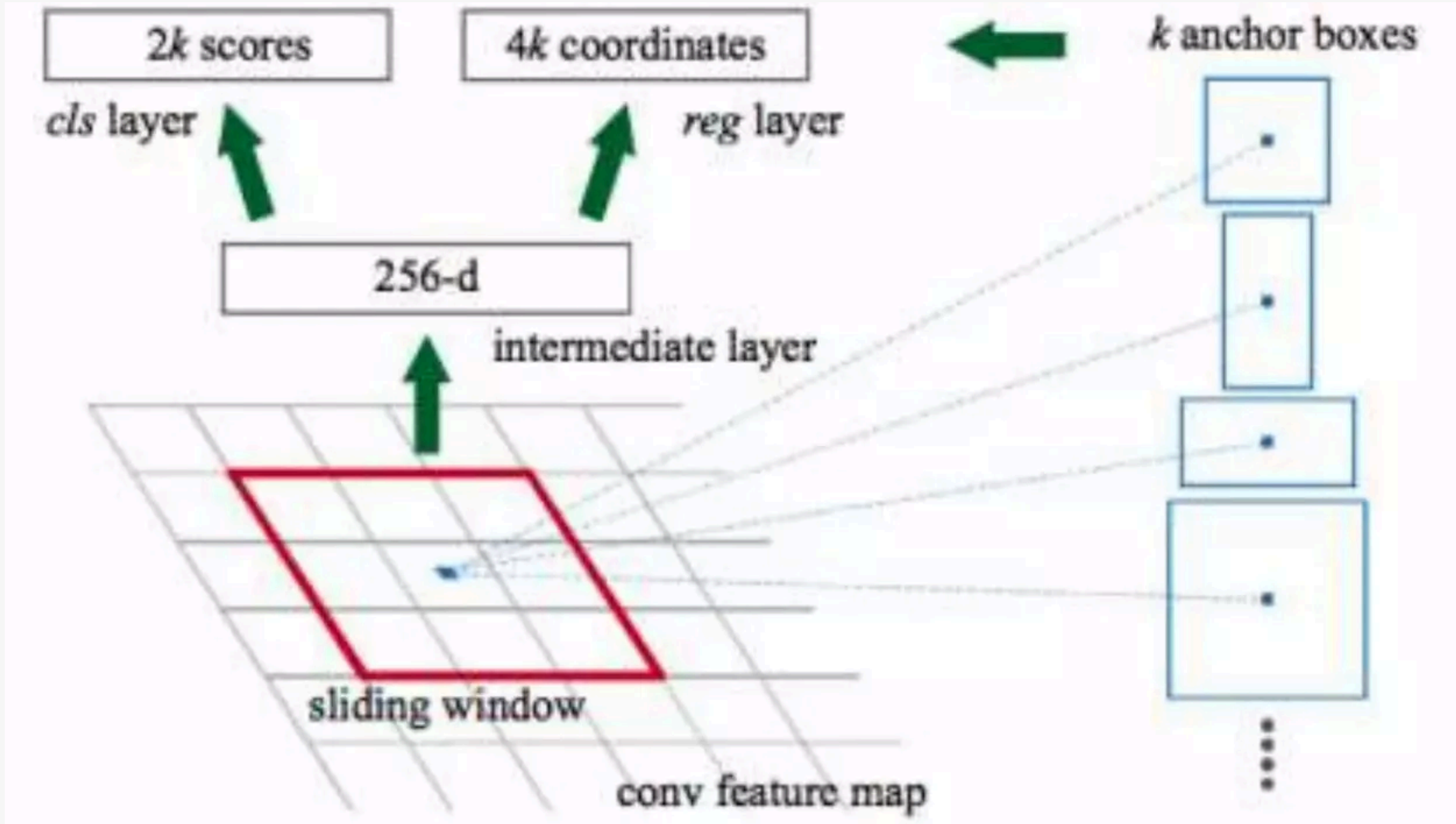
- 하지만, object들의 크기와 비율은 다양
- 미리 정의된 여러 크기와 비율의 reference box k



Faster R - CNN

Anchor 개수

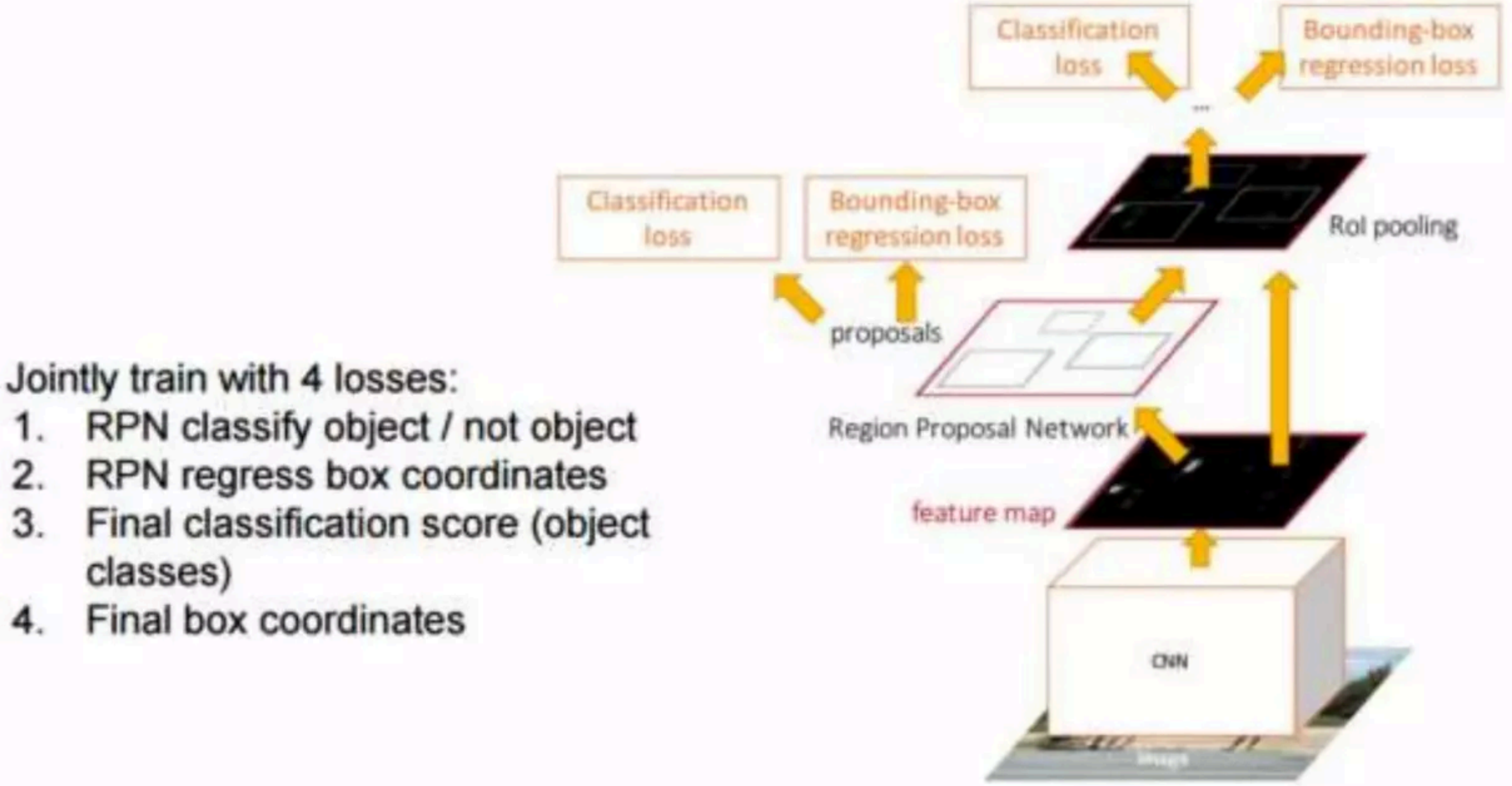
- feacher map의 크기가 W x H 일때
- 총 W x H x K 개의 anchor를 가진다.
- 논문에서는 3가지 크기 (128, 256, 512) , 3가지 비율 (2:1, 1:1, 1:2)
- K = 9



Faster R - CNN

Alternating Optimization

- RPN과 Fast R-CNN이 서로 convolution feature를 공유한 상태에서 번갈아 가며 학습을 진행하는 형태
- 복잡한 학습 과정



Faster R - CNN

Joint Optimization

- RPN의 loss function 과 Fast R-CNN의 loss function을 합쳐 multi_task loss로 둔 뒤 한번에 학습
- 동일하거나 높은 성능이 나올 수 있음을 실험적으로 증명

Jointly train with 4 losses:

1. RPN classify object / not object
2. RPN regress box coordinates
3. Final classification score (object classes)
4. Final box coordinates

Faster R - CNN

Experiments

- RPN을 사용했을때 상당한 속도 향상을 보인다.

Table 5: **Timing** (ms) on a K40 GPU, except SS proposal is evaluated in a CPU. "Region-wise" includes NMS, pooling, fully-connected, and softmax layers. See our released code for the profiling of running time.

model	system	conv	proposal	region-wise	total	rate
VGG	SS + Fast R-CNN	146	1510	174	1830	0.5 fps
VGG	RPN + Fast R-CNN	141	**10**	47	**198**	**5 fps**
ZF	RPN + Fast R-CNN	31	**3**	25	**59**	**17 fps**

Faster R - CNN

Experiments

- 3 scales, 3 ratios를 사용 했을때 가장 성능이 잘 나온다
- Anchor를 9개로 잡은 이유

Table 8: Detection results of Faster R-CNN on PASCAL VOC 2007 test set using **different settings of anchors**. The network is VGG-16. The training data is VOC 2007 trainval. The default setting of using 3 scales and 3 aspect ratios (69.9%) is the same as that in Table 3.

settings	anchor scales	aspect ratios	mAP (%)
1 scale, 1 ratio	128^2	1:1	65.8
	256^2	1:1	66.7
1 scale, 3 ratios	128^2	{2:1, 1:1, 1:2}	68.8
	256^2	{2:1, 1:1, 1:2}	67.9
3 scales, 1 ratio	$\{128^2, 256^2, 512^2\}$	1:1	**69.8**
3 scales, 3 ratios	$\{128^2, 256^2, 512^2\}$	{2:1, 1:1, 1:2}	**69.9**

Faster R - CNN

Performance

- R-CNN, Fast R-CNN, Faster R-CNN의 성능 비교
- 거의 실시간에 가까운 속도로 동작할 수 있다

per image

system	time	07 data	07+12 data
R-CNN	~50s	66.0	-
Fast R-CNN	~2s	66.9	70.0
Faster R-CNN	198ms	**69.9**	**73.2**

detection mAP on PASCAL VOC 2007, with VGG-16 pre-trained on ImageNet

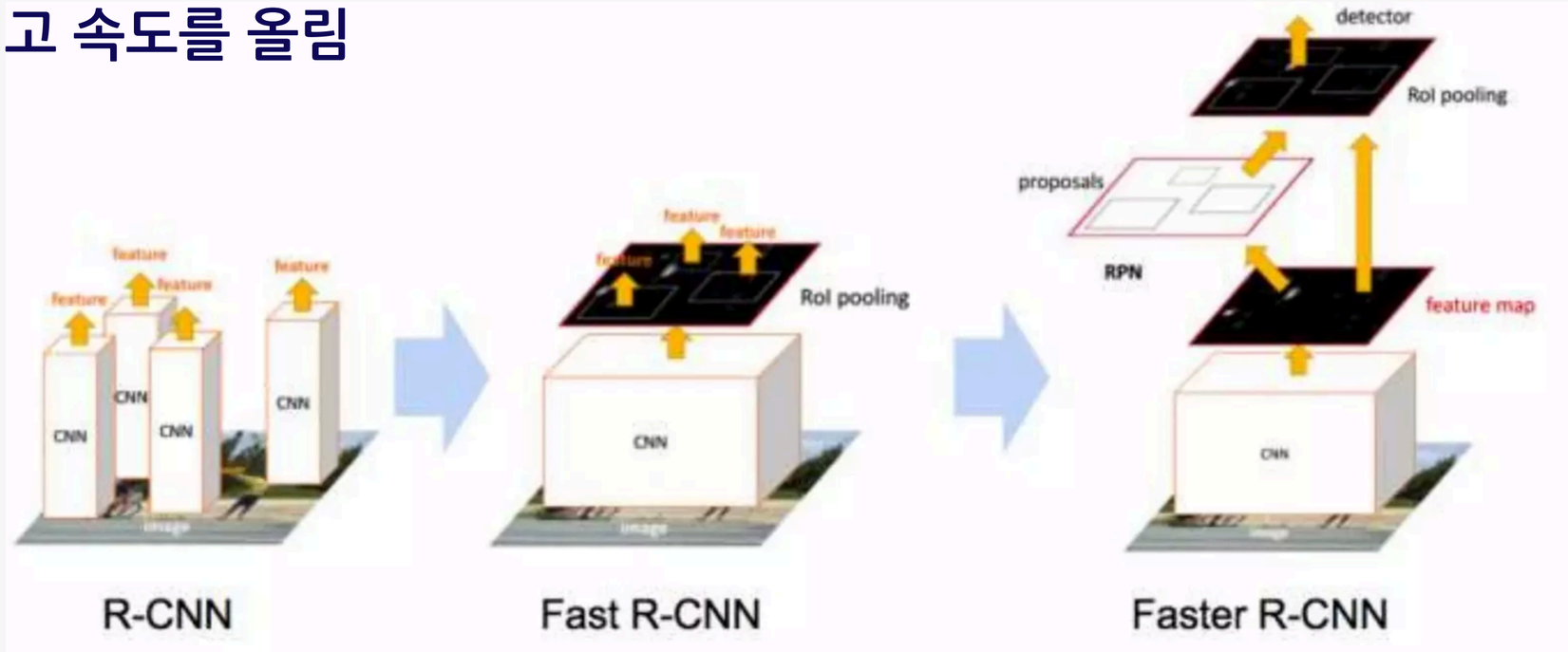
SUMMARY

R-CNN

Region proposal과 Classification CNN 을 결합하여 높은 성능의 Object detection을 수행

Fast R-CNN

ROI Pooling 을 사용해 연산을 줄이고 속도를 올림



SUMMARY

Faster R-CNN

- Region proposal 알고리즘으로 생기는 bottleneck을 개선
- Conv feature map으로부터 region proposal을 직접 생성할 수 있는 RPN(region proposal network)를 제안