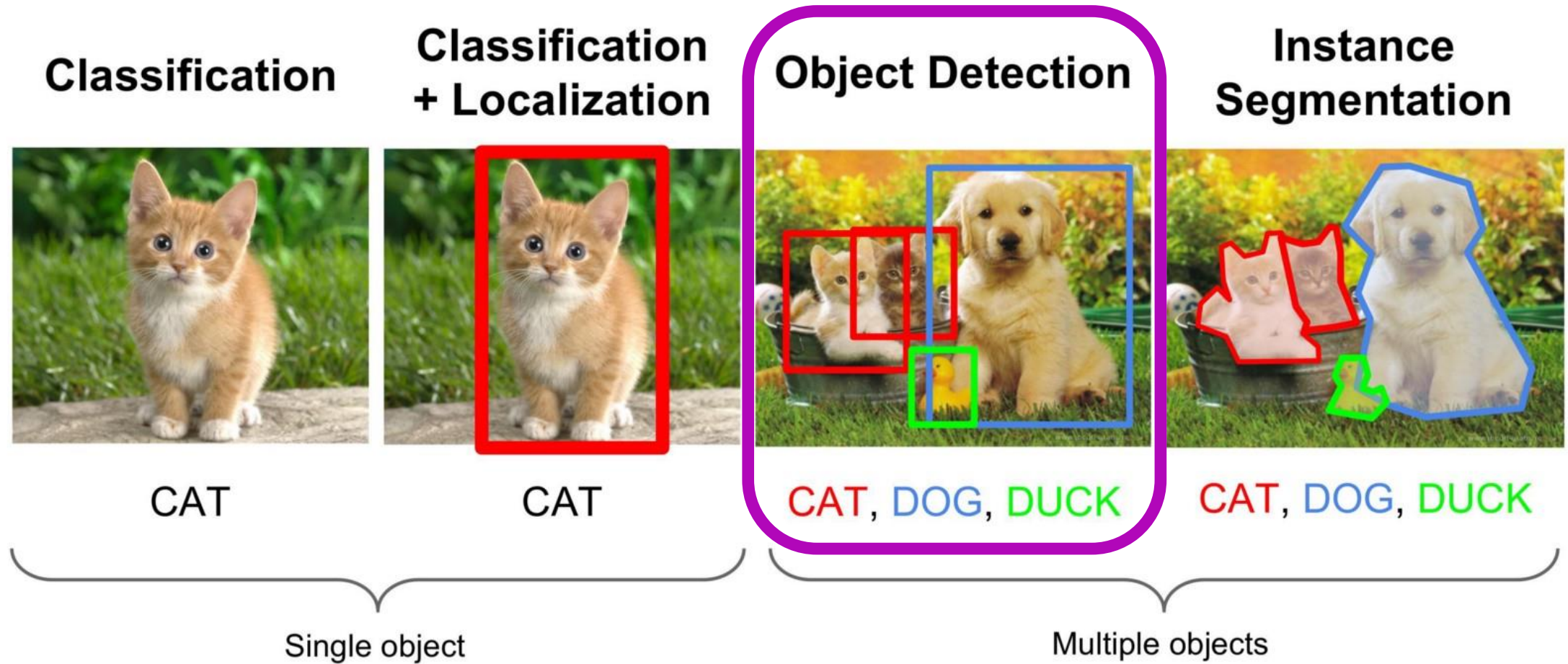


Rich feature hierarchies for accurate object detection and semantic segmentation (R-CNN)

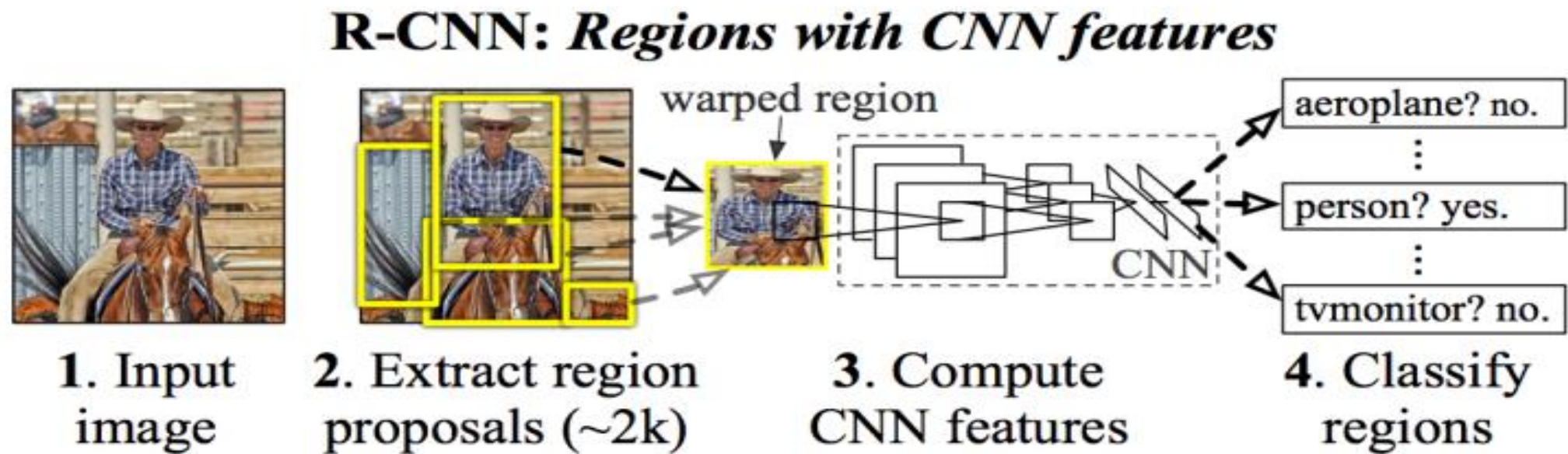
Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik
UC Berkeley

Sunwoo Kim
swkim@dongguk.edu

Computer Vision Task



Structure of R-CNN



1. Generating around 2000 category-independent region proposals.
2. Extracting a fixed-length feature vector from each proposal using a CNN
3. Classify each region with category-specific linear SVMs.

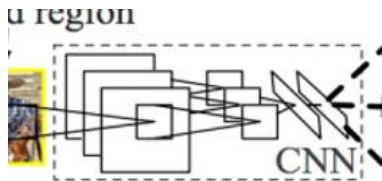
Three modules



2. Extract region proposals (~2k)

Module 1 (Region Proposals)

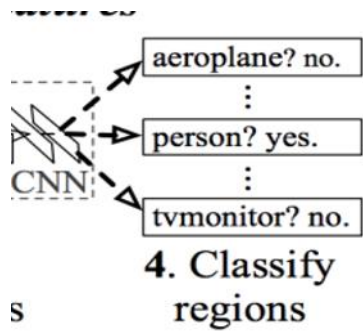
Generating category-independent region proposals.



3. Compute CNN features

Module 2 (Feature Extraction)

Large convolutional neural network that extracts a fixed-length feature vector from each region



4. Classify regions

Module 3

A set of class-specific linear SVMs.

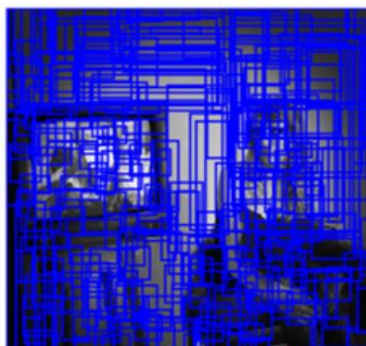
Region Proposals (Selective search)



Input Image



Segmentation



Candidate objects

가능한 많은 수의 초기영역 설정
(모든 객체에 박스가 그려질 수 있도록)



Input Image



Initial Segmentation

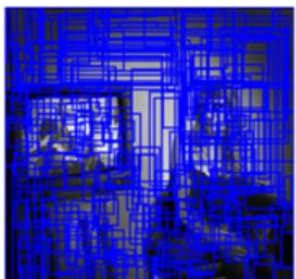


After some
iterations

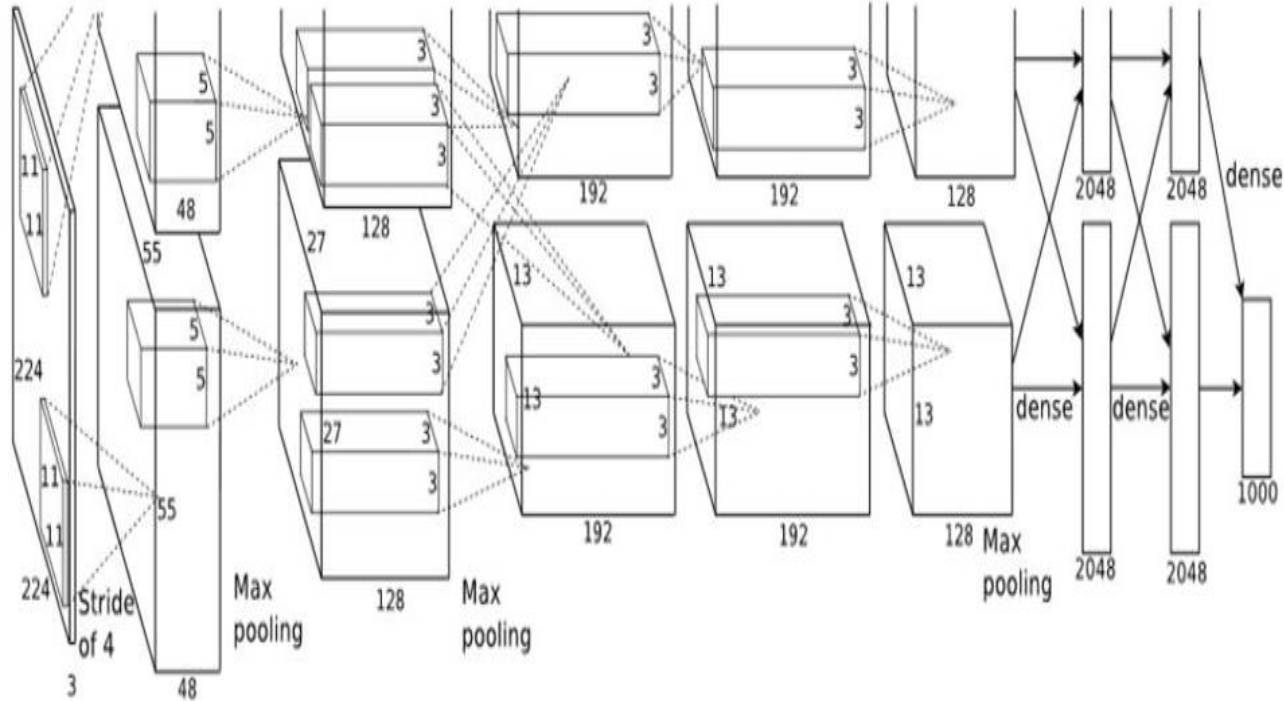


After more
iterations

유사도 큰 영역끼리 통합을 해 나간다.



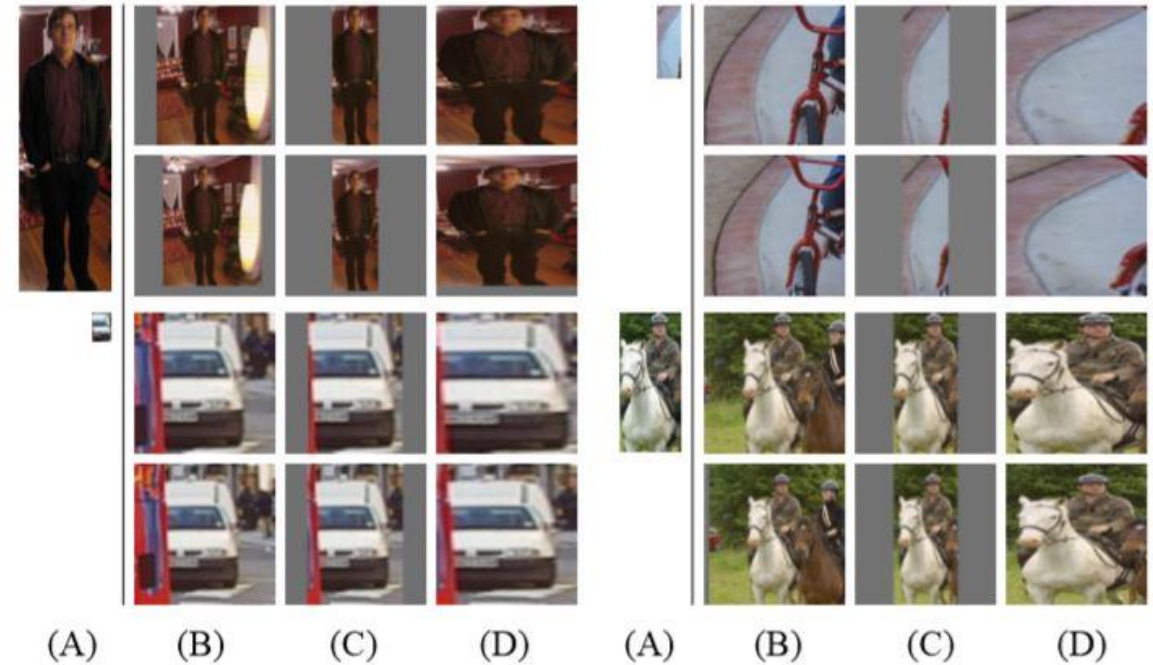
Feature Extraction



<Structure of AlexNet>

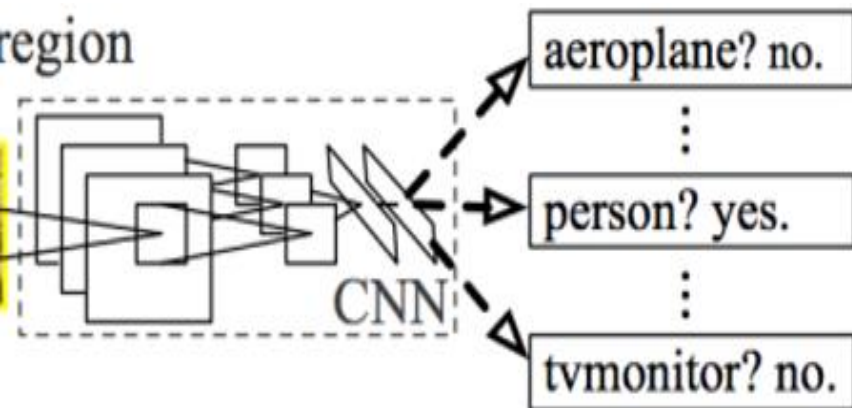
구성 : 5 Conv layer + 2 FC layer

입력 : mean-subtracted 227x227 RGB



Region proposals를 227x227 Image로
변형시키기 전, bounding box를 키워서 bounding
box영역이 tight하지 않게 만들어준다.
P(pixel : 0-16사이로 조절)

Test-time Detection



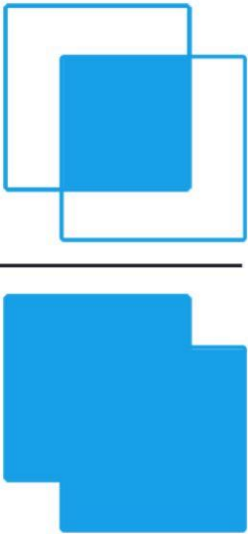
1. Region proposals을 CNN을 통과시키고 SVM을 이용하여 나온 feature vector를 이용하여 scoring(점수를 매김)한다.

2. Greedy non-maximum suppression을 각 클래스에 대하여 독립적으로 실행하여 하나의 region proposal만을 남긴다.

Greedy non-maximum suppression

NMS? : mAP올리고 연산량을 줄이기
위해서

1. Score가 제일 높은 region proposal
을 ground truth로 설정
2. 나머지 region proposal에 대해서
IoU를 구하고 threshold이상의 값을
가진다면 제거.


$$\text{IoU} = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

< Intersection of Union >

Training

Supervised pre-training

ImageNet2012 데이터를 이용하여 트레이닝 하는데 box label is not available
(단순 classification을 하여 가중치를 초기설정 해놓는 과정이라 생각)

Domain-specific fine-tuning

새로운 task(detection)가 주어지는데 warped proposal window만을 이용하여 CNN의 SGD train을 한다.
(여기서 FCL 마지막 단을 randomly initialized $[N+1]$ 개로 replace해서 사용 / ILSVRC2013에서 $N = 200$
즉, N 은 클래스 개수를 의미한다. $+1 == BG$)

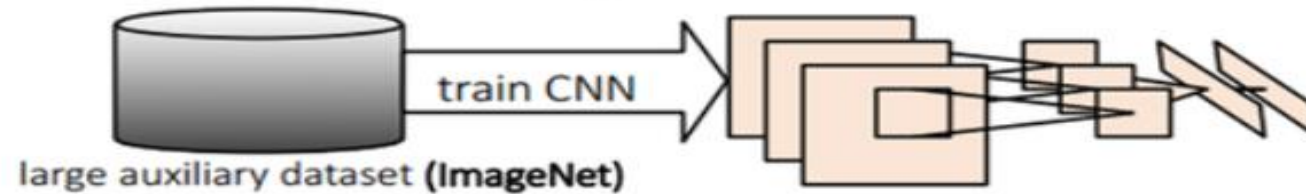
Ground-truth box에 대해 IoU overlap은 0.5 이상인 데이터 만을 positive로 그 이하는
negative 즉, BG(background)로 한다.

SGD learning rate=0.001로 (pre-train에선 0.01) 이게 fine-tuning이 initialization을 clobbering
하는 것을 방지

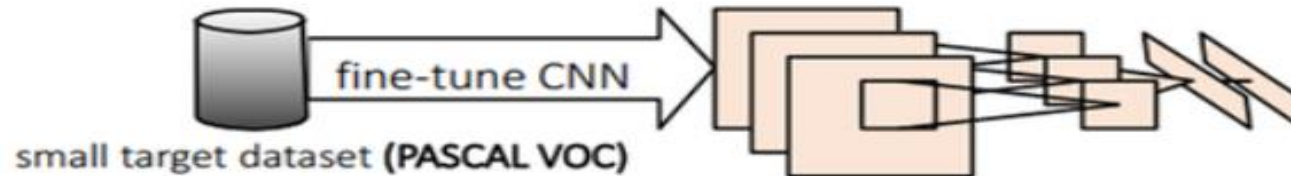
각 SGD iteration 마다 32개 p-w / 96개 b-w로 mini-batch 128로 맞추고, positive window가 훨씬
적기때문에 이 쪽으로 편향

Training

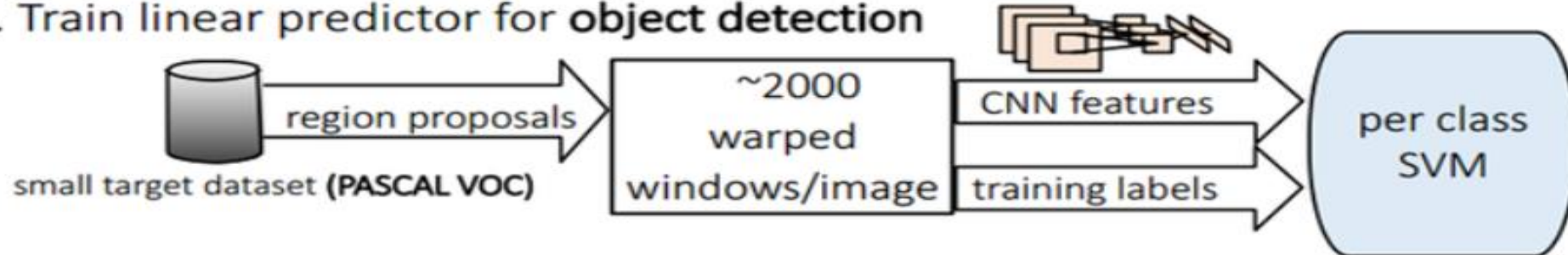
1. Pre-train CNN for image classification



2. Fine-tune CNN for object detection



3. Train linear predictor for object detection



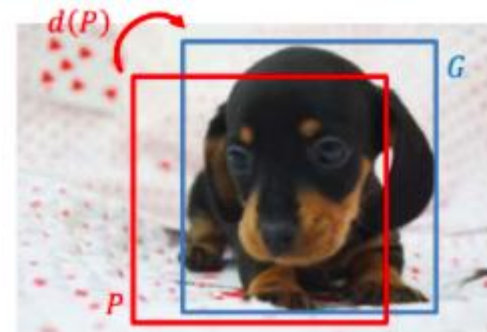
Bounding box regression

Each function $d_\star(P)$ (where \star is one of x, y, h, w) is modeled as a linear function of the pool_5 features of proposal P , denoted by $\phi_5(P)$. (The dependence of $\phi_5(P)$ on the image data is implicitly assumed.) Thus we have $d_\star(P) = \mathbf{w}_\star^T \phi_5(P)$, where \mathbf{w}_\star is a vector of learnable model parameters. We learn \mathbf{w}_\star by optimizing the regularized least squares objective (ridge regression):

$\{(P^i, G^i)\}_{i=1, \dots, N}$, where $P^i = (P_x^i, P_y^i, P_w^i, P_h^i)$

$$G = (G_x, G_y, G_w, G_h)$$

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^T \phi_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_\star\|^2.$$



$$\hat{G}_x = P_w d_x(P) + P_x$$

$$\hat{G}_y = P_h d_y(P) + P_y$$

$$\hat{G}_w = P_w \exp(d_w(P))$$

$$\hat{G}_h = P_h \exp(d_h(P)).$$

$$t_x = (G_x - P_x) / P_w$$

$$t_y = (G_y - P_y) / P_h$$

$$t_w = \log(G_w / P_w)$$

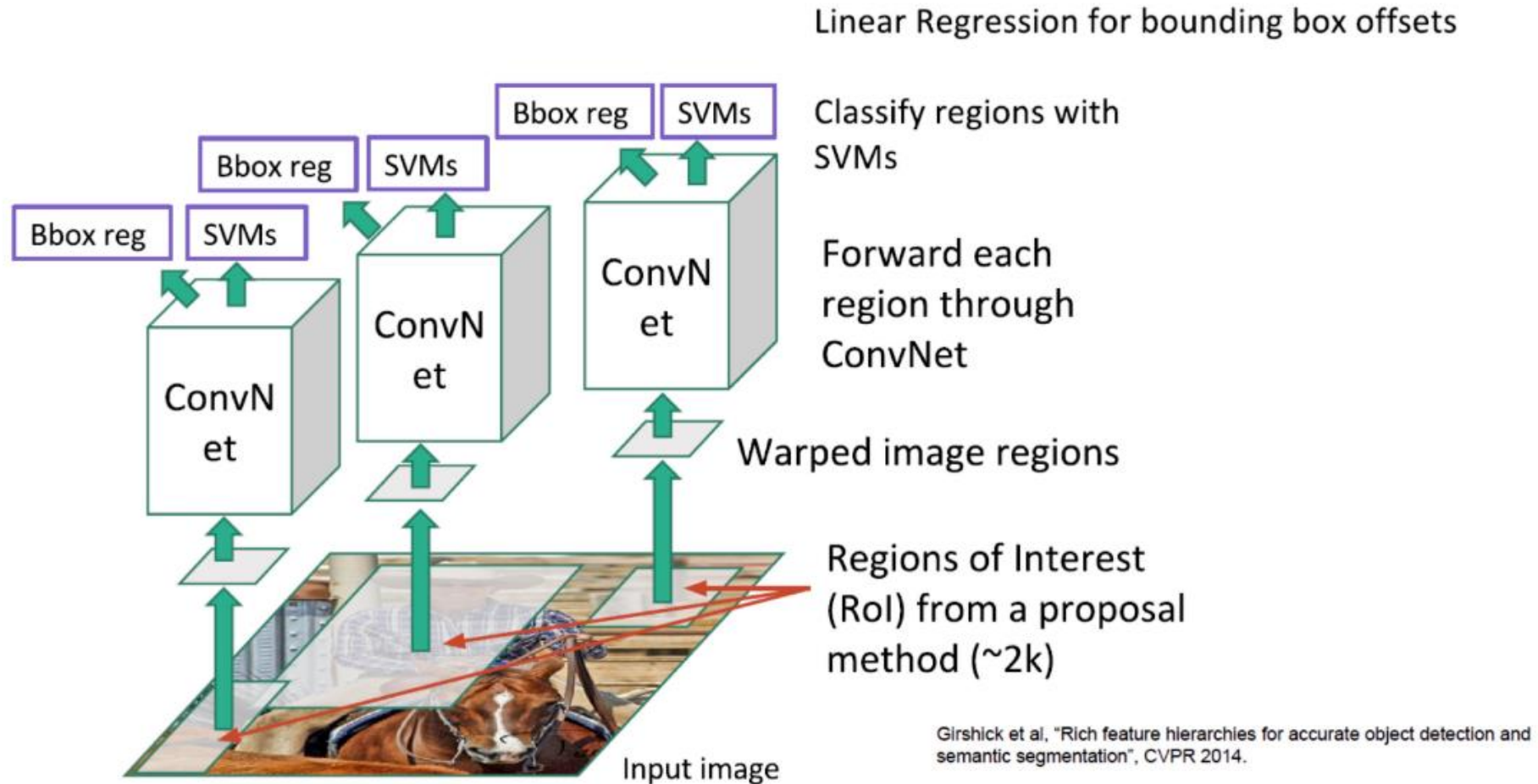
$$t_h = \log(G_h / P_h).$$

Bounding box regression

Two subtle issues

1. Regularization이 굉장히 중요 $\lambda = 1000$
2. (P, G)가 잘 선택되어야 함 \rightarrow 만약 뿔힌 P가 모든 G로부터 멀면 이 Task가 의미가 없어서 P가 적어도 한 개의 G box 근처에 있을 때만 only learning(기준은 maximum IoU Overlap 기준으로 0.6이 Threshold 이하는 모두 버림.)

Review of the Structure



Results

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN pool ₅	51.8	60.2	36.4	27.8	23.2	52.8	60.6	49.2	18.3	47.8	44.3	40.8	56.6	58.7	42.4	23.4	46.1	36.7	51.3	55.7	44.2
R-CNN fc ₆	59.3	61.8	43.1	34.0	25.1	53.1	60.6	52.8	21.7	47.8	42.7	47.8	52.5	58.5	44.6	25.6	48.3	34.0	53.1	58.0	46.2
R-CNN fc ₇	57.6	57.9	38.5	31.8	23.7	51.2	58.9	51.4	20.0	50.5	40.9	46.0	51.6	55.9	43.3	23.3	48.1	35.3	51.0	57.4	44.7
R-CNN FT pool ₅	58.2	63.3	37.9	27.6	26.1	54.1	66.9	51.4	26.7	55.5	43.4	43.1	57.7	59.0	45.8	28.1	50.8	40.6	53.1	56.4	47.3
R-CNN FT fc ₆	63.5	66.0	47.9	37.7	29.9	62.5	70.2	60.2	32.0	57.9	47.0	53.5	60.1	64.2	52.2	31.3	55.0	50.0	57.7	63.0	53.1
R-CNN FT fc ₇	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN FT fc ₇ BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
DPM v5 [20]	33.2	60.3	10.2	16.1	27.3	54.3	58.2	23.0	20.0	24.1	26.7	12.7	58.1	48.2	43.2	12.0	21.1	36.1	46.0	43.5	33.7
DPM ST [28]	23.8	58.2	10.5	8.5	27.1	50.4	52.0	7.3	19.2	22.8	18.1	8.0	55.9	44.8	32.4	13.3	15.9	22.8	46.2	44.9	29.1
DPM HSC [31]	32.2	58.3	11.5	16.3	30.6	49.9	54.8	23.5	21.5	27.7	34.0	13.7	58.1	51.6	39.9	12.4	23.5	34.4	47.4	45.2	34.3

Table 2: Detection average precision (%) on VOC 2007 test. Rows 1-3 show R-CNN performance without fine-tuning. Rows 4-6 show results for the CNN pre-trained on ILSVRC 2012 and then fine-tuned (FT) on VOC 2007 trainval. Row 7 includes a simple bounding-box regression (BB) stage that reduces localization errors (Section C). Rows 8-10 present DPM methods as a strong baseline. The first uses only HOG, while the next two use different feature learning approaches to augment or replace HOG.

VOC 2007 test	aero	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	mAP
R-CNN T-Net	64.2	69.7	50.0	41.9	32.0	62.6	71.0	60.7	32.7	58.5	46.5	56.1	60.6	66.8	54.2	31.5	52.8	48.9	57.9	64.7	54.2
R-CNN T-Net BB	68.1	72.8	56.8	43.0	36.8	66.3	74.2	67.6	34.4	63.5	54.5	61.2	69.1	68.6	58.7	33.4	62.9	51.1	62.5	64.8	58.5
R-CNN O-Net	71.6	73.5	58.1	42.2	39.4	70.7	76.0	74.5	38.7	71.0	56.9	74.5	67.9	69.6	59.3	35.7	62.1	64.0	66.5	71.2	62.2
R-CNN O-Net BB	73.4	77.0	63.4	45.4	44.6	75.1	78.1	79.8	40.5	73.7	62.2	79.4	78.1	73.1	64.2	35.6	66.8	67.2	70.4	71.1	66.0

Table 3: Detection average precision (%) on VOC 2007 test for two different CNN architectures. The first two rows are results from Table 2 using Krizhevsky et al.’s architecture (T-Net). Rows three and four use the recently proposed 16-layer architecture from Simonyan and Zisserman (O-Net) [43].

Thanks