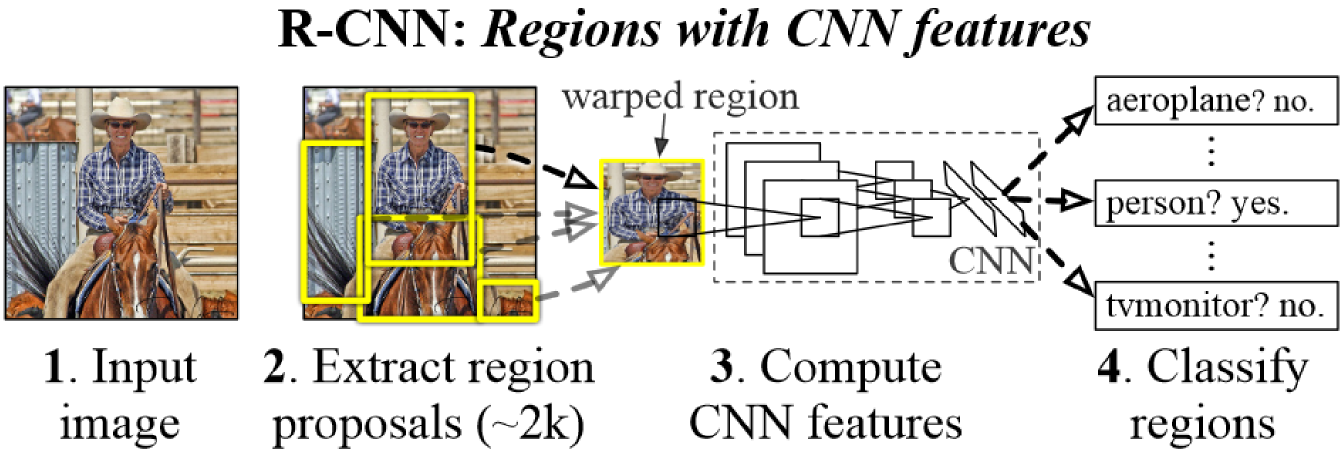
Rich feature hierarchies for accurate object detection and semantic segmentation



- "This paper is the first to show that a CNN can lead to dramatically higher object detection performance on PASCAL VOC as compared to systems based on simpler HOG-like features.” Page 2
- 💡 HOG

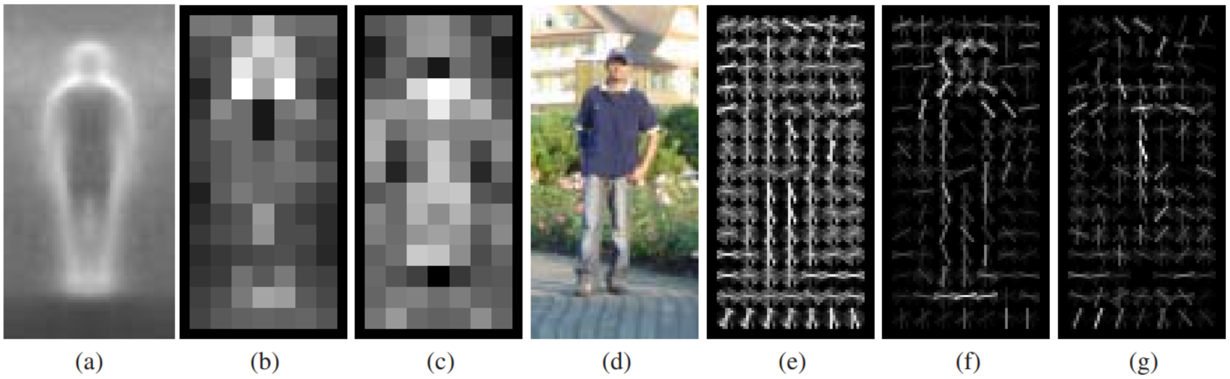


Figure 6. Our HOG detectors cue mainly on silhouette contours (especially the head, shoulders and feet). The most active blocks are centred on the image background just *outside* the contour. (a) The average gradient image over the training examples. (b) Each “pixel” shows the maximum positive SVM weight in the block centred on the pixel. (c) Likewise for the negative SVM weights. (d) A test image. (e) It’s computed R-HOG descriptor. (f,g) The R-HOG descriptor weighted by respectively the positive and the negative SVM weights.

- "we focused on two problems: localizing objects with a deep network and training a high-capacity model with only a small quantity of annotated detection data.” Page 2
- "Unlike image classification, detection requires localizing (likely many) objects within an image.” Page 2

localizing objects

1. 가장 greedy한 방법: sliding window
   이미지 크기: $H \times W$
   박스 크기: $h \times w$
   가능한 x 위치: $W - w + 1$
   가능한 y 위치: $H - h + 1$
   가능한 박스 위치: $(W - w + 1)(H - h + 1)$

$$\sum_{h=1}^{H} \sum_{w=1}^{W} (W - w + 1)(H - h + 1)$$

   is equal to

$$\frac{H(H+1)}{2} \frac{W(W+1)}{2}$$

2. selective search

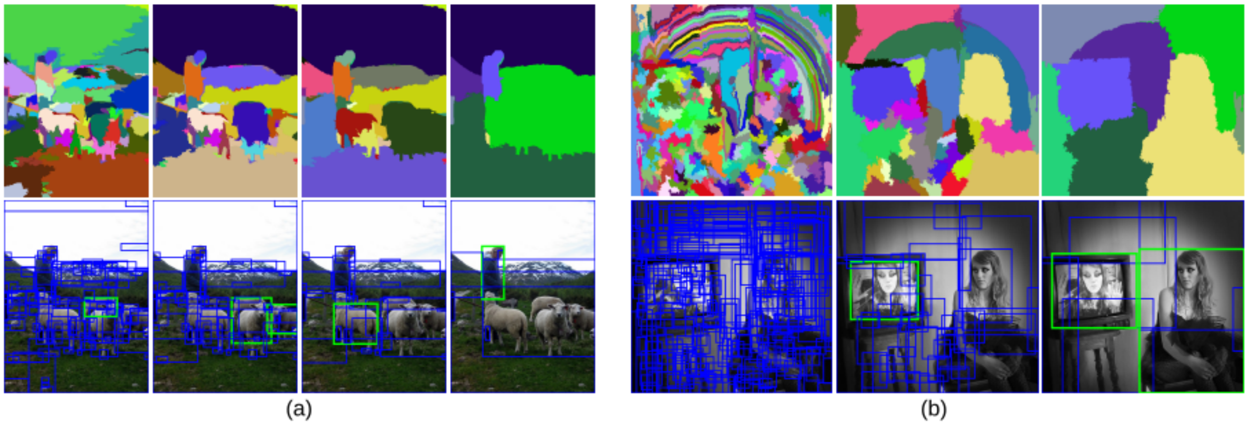


Figure 2: Two examples of our selective search showing the necessity of different scales. On the left we find many objects at different scales. On the right we necessarily find the objects at different scales as the girl is contained by the tv.

   2000개의 bounding box를 뽑음
   region proposal -> cnn

- R-CNN이라 부름
- "Instead, we solve the CNN localization problem by operating within the "recognition using regions" paradigm [21], which has been successful for both object detection [39] and semantic segmentation" Page 2

Proposal-based vs Proposal-free

1. proposal-based model
   - Two-stage model

    - `R-CNN`, `Fast R-CNN` , `Faster R-CNN`, `R-FCN`
2. proposal-free model
    - Single-stage model
    - `YOLO`, `SSD`, `DETR`

region proposal된 2000개 각각을 cnn에 넣은 후 SVM을 통해 classify

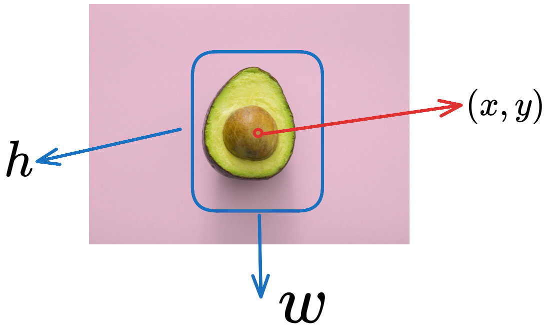
- "At test time, our method generates around 2000 category-independent region proposals for the input image, extracts a fixed-length feature vector from each proposal using a CNN, and then classifies each region with category-specific linear SVMs." Page 2

bounding-box regression 추가

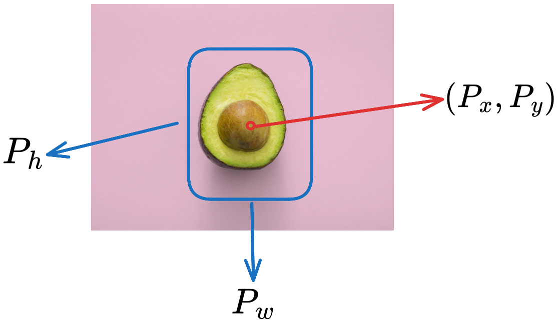
- As an immediate consequence of this analysis, we demonstrate that a simple bounding-box regression method significantly reduces mislocalizations, which are the dominant error mode.

bounding box regression

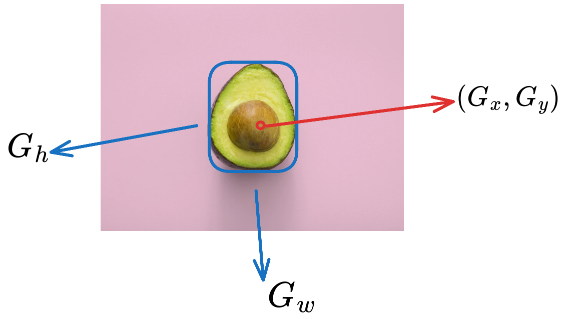
bounding box는 x, y, h, w로 구성이 됨



proposal로 나온 bounding box



ground truth



transformation

$$t_x = (G_x - P_x)/P_w \quad (6)$$
$$t_y = (G_y - P_y)/P_h \quad (7)$$
$$t_w = \log(G_w/P_w) \quad (8)$$
$$t_h = \log(G_h/P_h). \quad (9)$$

$$d_x = (\hat{G}_x - P_x)/P_w$$

$$d_y = (\hat{G}_y - P_y)/P_h$$

$$d_w = \log(\hat{G}_w/P_w)$$

$$d_h = \log(\hat{G}_h/P_h)$$

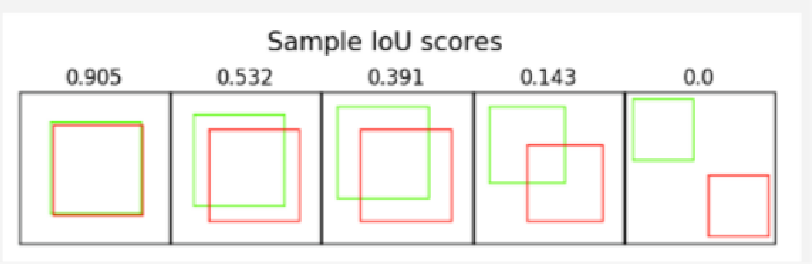
- "Our goal is to learn a transformation that maps a proposed box P to a ground-truth box G.” Page 12

regularized least squares objective (ridge regression)

$$\mathbf{w}_\star = \underset{\hat{\mathbf{w}}_\star}{\operatorname{argmin}} \sum_i^N (t_\star^i - \hat{\mathbf{w}}_\star^{\mathrm{T}} \phi_5(P^i))^2 + \lambda \left\| \hat{\mathbf{w}}_\star \right\|^2 . \quad (5)$$

https://velog.io/@claude_ssim/%EA%B8%B0%EA%B3%84%ED%95%99%EC%8A%B5-Linear-Regression-Ridge-Regression

IoU(Intersection over Union)

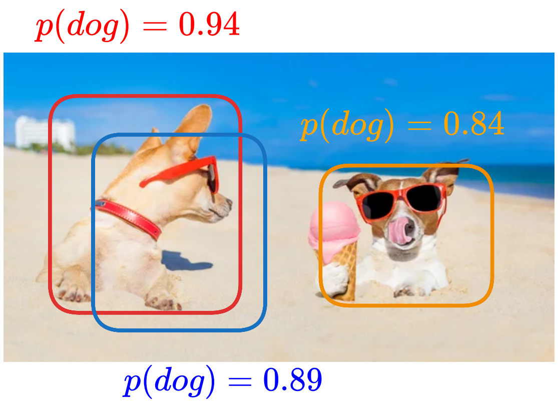


처음 라벨링 할 때 threshold를 0.3

- "Less clear is how to label a region that partially overlaps a car. We resolve this issue with an IoU overlap threshold, below which regions are defined as negatives. The overlap threshold, 0.3, was selected by a grid search over {0, 0.1, . . . , 0.5}" Page 4
- "We bias the sampling towards positive windows because they are extremely rare compared to background." Page 4

non-maximum suppression

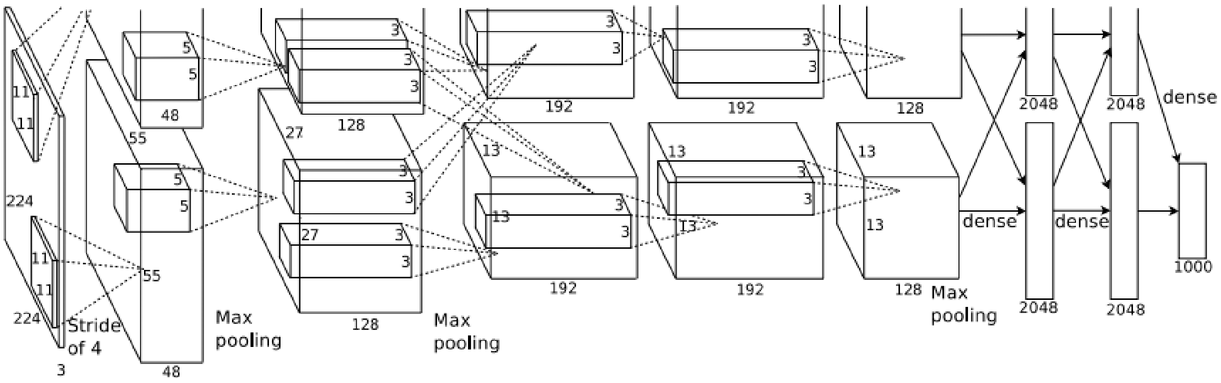
- Then, for each class, we score each extracted feature vector using the SVM trained for that class. Given all scored regions in an image, we apply a greedy non-maximum suppression



IoU (🟥🟦) = 0.75
IoU (🟥🟧) = 0
threshold = 0.5

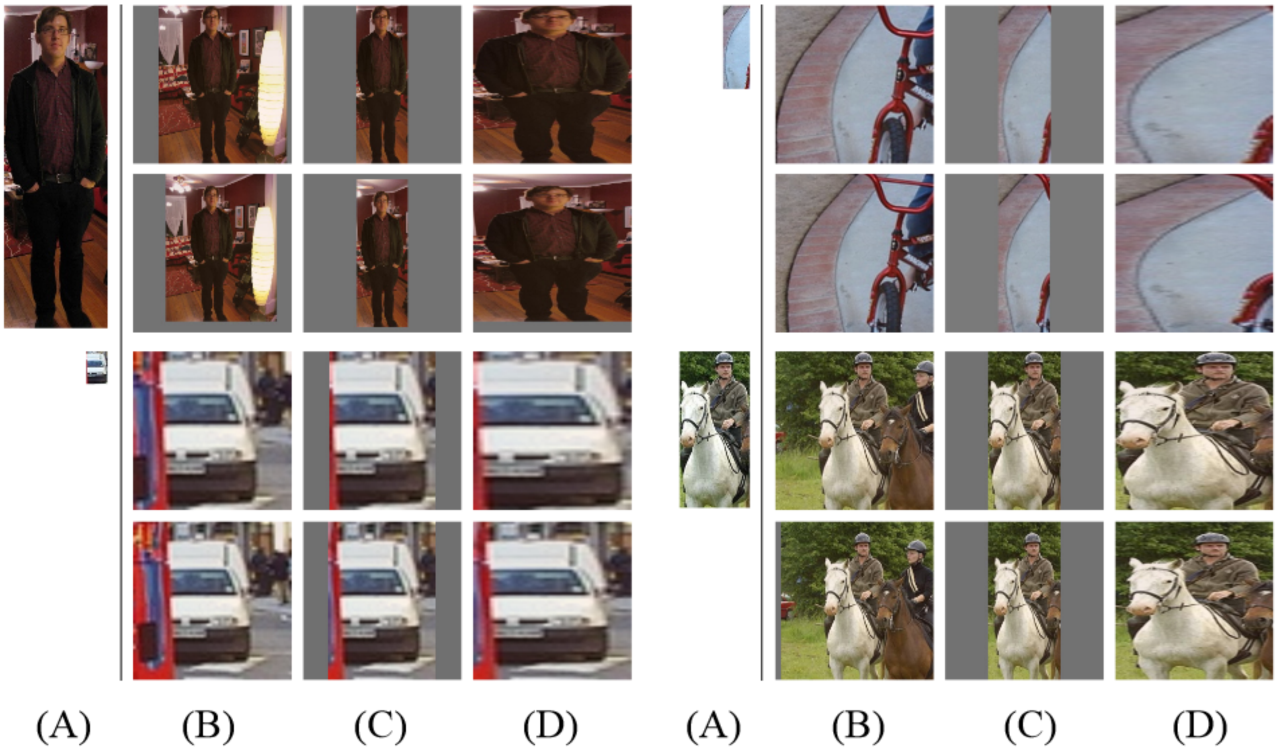
cnn (feautre extraction)

image net



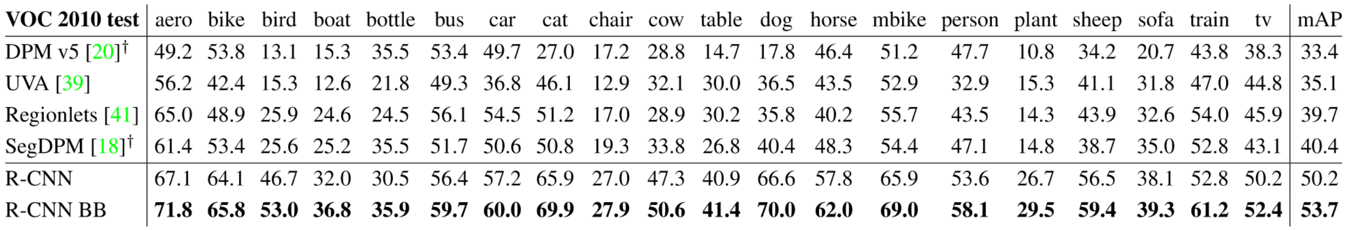
- "we must first convert the image data in that region into a form that is compatible with the CNN (its architecture requires inputs of a fixed 227 × 227 pixel size)." Page 3

227 * 227 사이즈로 맞춰주기



- "Within each column and example proposal, the top row corresponds to p = 0 pixels of context padding while the bottom row has p = 16 pixels of context padding." Page 11

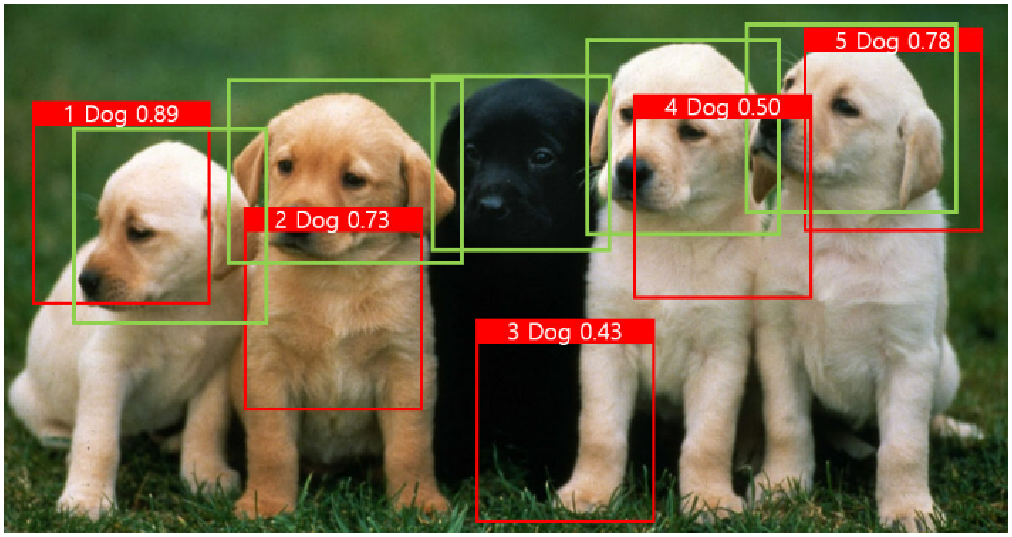
experiments



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**

mAP

	positive(predict)	negative(predict)
positive(g.t)	TP (있는 걸 있다고 함)	FN (있는데 없다고 함)
negative(g.t)	FP (없는데 있다고 함)	TN (없는걸 없다고 함)



1,4,5는 TP

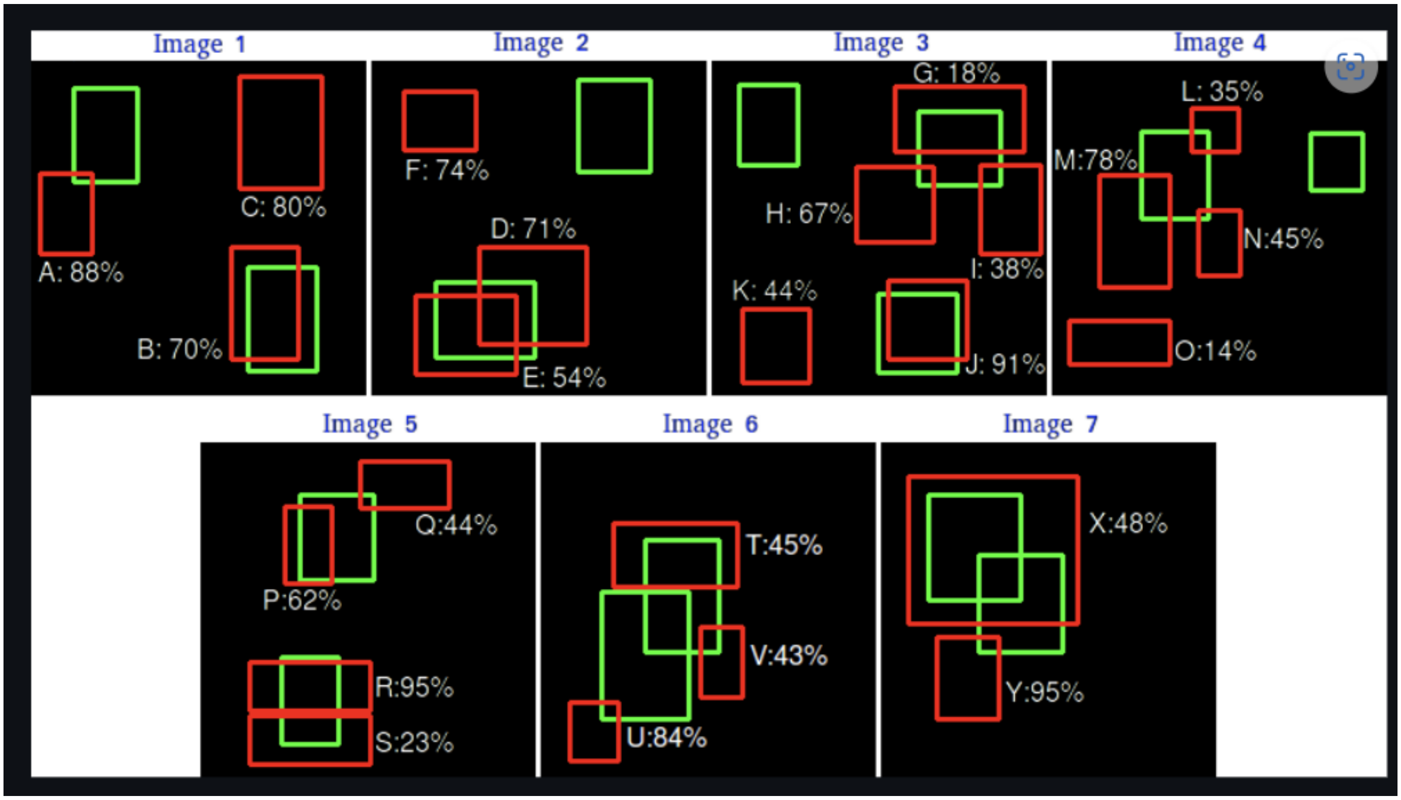
3은 FP

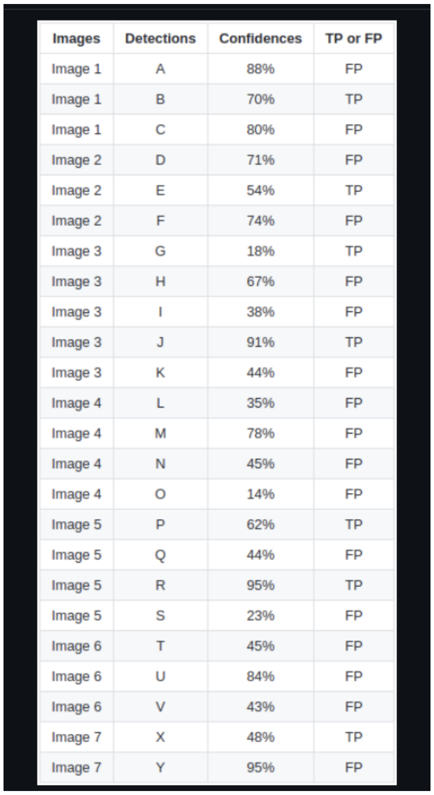
가운데 검정 강아지 박스는 FN

TN은 object detection에서 고려 x(없는 걸 없다고 판정하는 모델이 아니기에)

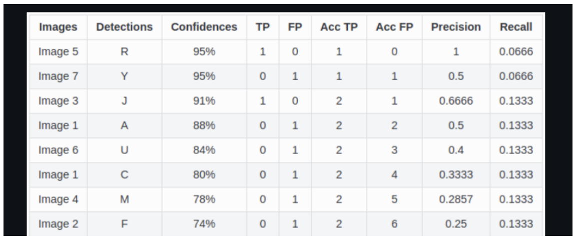
Precision: $\frac{TP}{TP+FP}$

Recall: $\frac{TP}{TP+FN}$



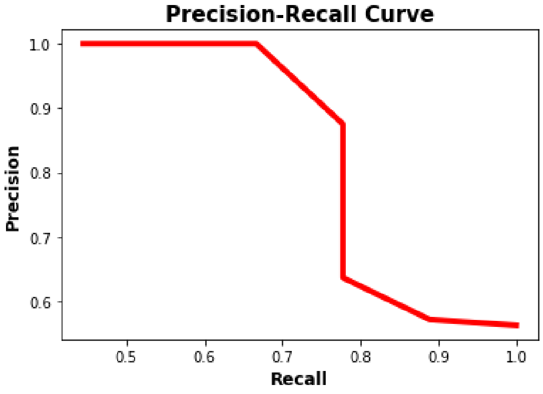


Images	Detections	Confidences	TP or FP
Image 1	A	88%	FP
Image 1	B	70%	TP
Image 1	C	80%	FP
Image 2	D	71%	FP
Image 2	E	54%	TP
Image 2	F	74%	FP
Image 3	G	18%	TP
Image 3	H	67%	FP
Image 3	I	38%	FP
Image 3	J	91%	TP
Image 3	K	44%	FP
Image 4	L	35%	FP
Image 4	M	78%	FP
Image 4	N	45%	FP
Image 4	O	14%	FP
Image 5	P	62%	TP
Image 5	Q	44%	FP
Image 5	R	95%	TP
Image 5	S	23%	FP
Image 6	T	45%	FP
Image 6	U	84%	FP
Image 6	V	43%	FP
Image 7	X	48%	TP
Image 7	Y	95%	FP

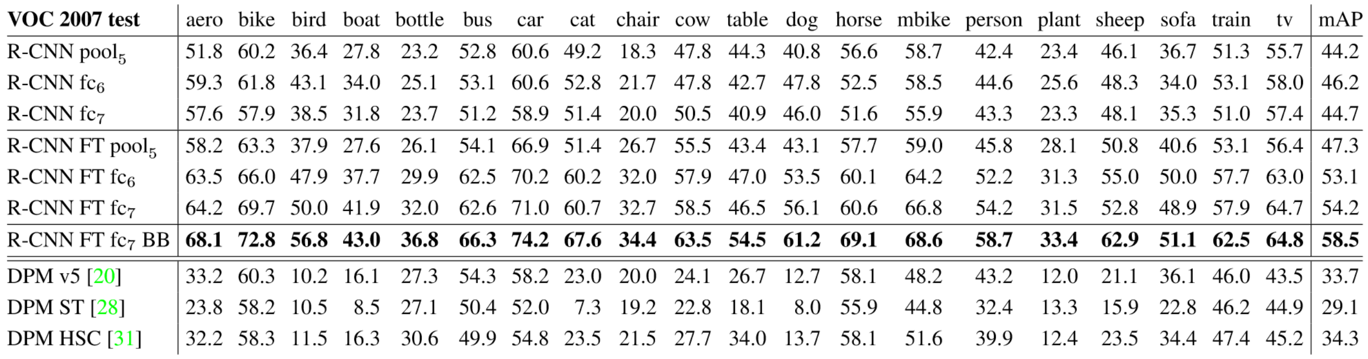


Images	Detections	Confidences	TP	FP	Acc TP	Acc FP	Precision	Recall
Image 5	R	95%	1	0	1	0	1	0.0666
Image 7	Y	95%	0	1	1	1	0.5	0.0666
Image 3	J	91%	1	0	2	1	0.6666	0.1333
Image 1	A	88%	0	1	2	2	0.5	0.1333
Image 6	U	84%	0	1	2	3	0.4	0.1333
Image 1	C	80%	0	1	2	4	0.3333	0.1333
Image 4	M	78%	0	1	2	5	0.2857	0.1333
Image 2	F	74%	0	1	2	6	0.25	0.1333

curve 아래의 면적을 Average Precision이라고 함.



Object Detection의 정의와 Metric mAP(mean Average Precision) (tistory.com)



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

cnn만으로도 생각보다 결과가 좋다.

- "More surprising is that removing both fc7 and fc6 produces quite good results even though pool5 features are computed using only 6% of the CNN’s parameters.” Page 6

fine-tuning하게되면, fc layer들의 성능이 좋아짐

- "The boost from fine-tuning is much larger for fc6 and fc7 than for pool5, which suggests that the pool5 features learned from ImageNet are general and that most of the improvement is gained from learning domain-specific non-linear classifiers on top of them.” Page 7

fine tuning

- "A second challenge faced in detection is that labeled data is scarce and the amount currently available is insufficient for training a large CNN. The conventional solution to this problem is to use unsupervised pre-training, followed by supervised fine-tuning” Page 2

ILSVRC(pretrain) => PASCAL(fine-tuning)

- "Aside from replacing the CNN’s ImageNetspecific 1000-way classification layer with a randomly initialized (N + 1)-way classification layer (where N is the number of object classes, plus 1 for background), the CNN architecture is unchanged. For VOC, N = 20 and for ILSVRC2013, N = 200.” Page 3