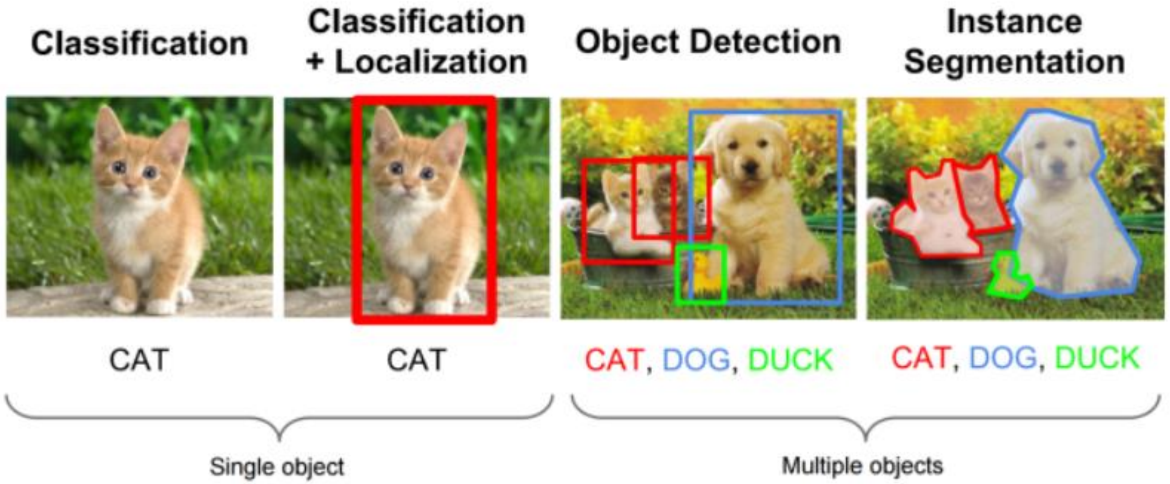
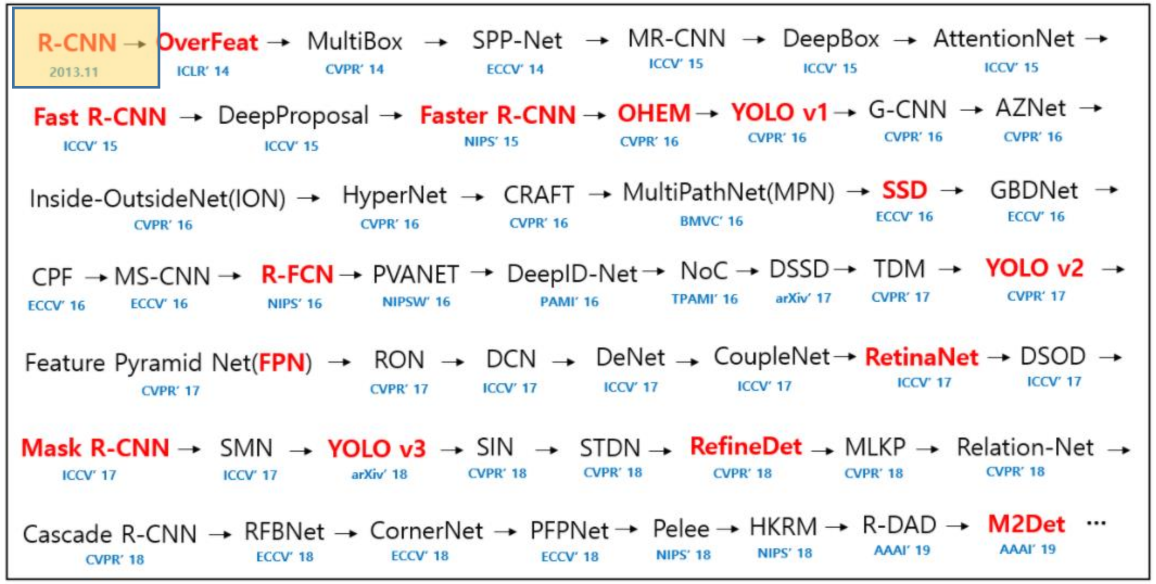
# RCNN

Rich feature hierarchies for accurate object detection and semantic segmentation

# Object Detection이란?





R-CNN → OverFeat → MultiBox → SPP-Net → MR-CNN → DeepBox → AttentionNet →
2013.11 ICLR' 14 CVPR' 14 ECCV' 14 ICCV' 15 ICCV' 15 ICCV' 15

Fast R-CNN → DeepProposal → Faster R-CNN → OHEM → YOLO v1 → G-CNN → AZNet →
ICCV' 15 ICCV' 15 NIPS' 15 CVPR' 16 CVPR' 16 CVPR' 16 CVPR' 16

Inside-OutsideNet(ION) → HyperNet → CRAFT → MultiPathNet(MPN) → SSD → GBDNet →
CVPR' 16 CVPR' 16 CVPR' 16 BMVC' 16 ECCV' 16 ECCV' 16

CPF → MS-CNN → R-FCN → PVANET → DeepID-Net → NoC → DSSD → TDM → YOLO v2 →
ECCV' 16 ECCV' 16 NIPS' 16 NIPSW' 16 PAMI' 16 TPAMI' 16 arXiv' 17 CVPR' 17 CVPR' 17

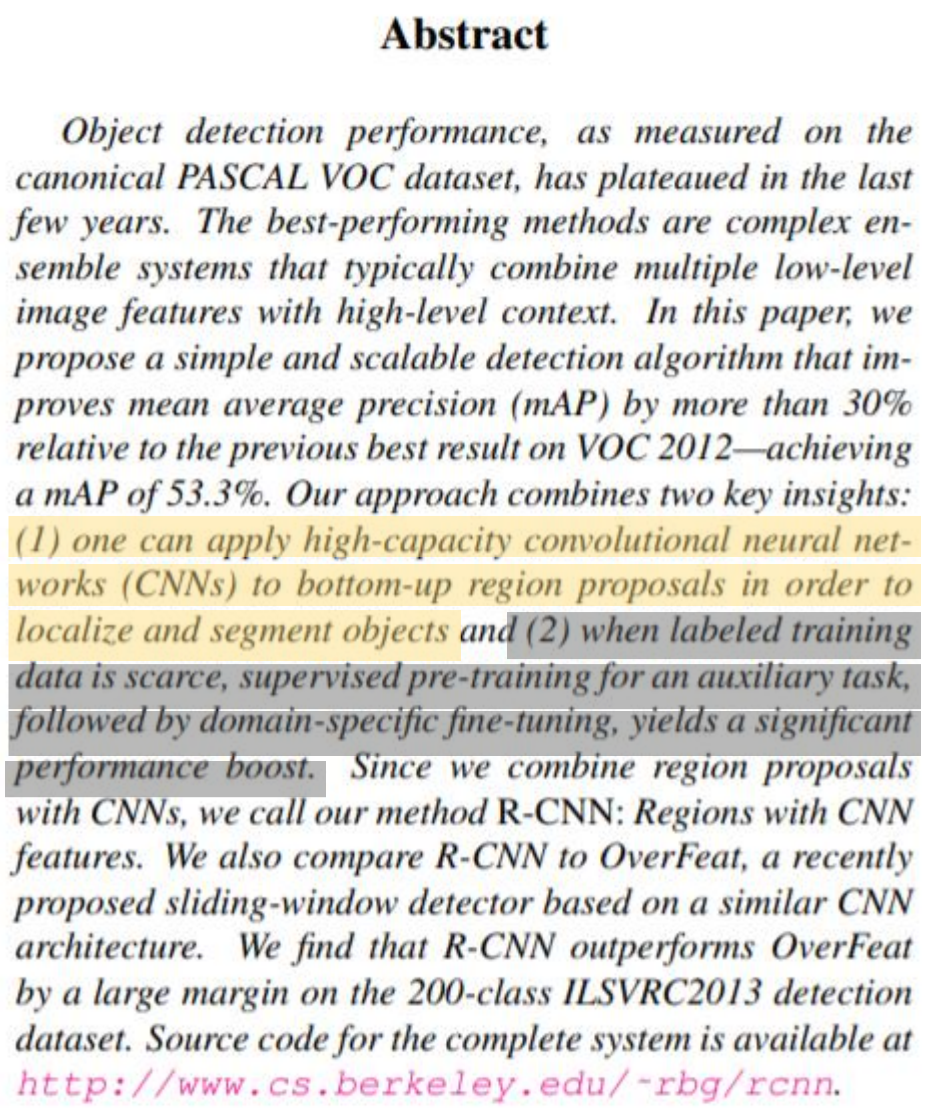
Feature Pyramid Net(FPN) → RON → DCN → DeNet → CoupleNet → RetinaNet → DSOD →
CVPR' 17 CVPR' 17 ICCV' 17 ICCV' 17 ICCV' 17 ICCV' 17 ICCV' 17

Mask R-CNN → SMN → YOLO v3 → SIN → STDN → RefineDet → MLKP → Relation-Net →
ICCV' 17 ICCV' 17 arXiv' 18 CVPR' 18 CVPR' 18 CVPR' 18 CVPR' 18 CVPR' 18

Cascade R-CNN → RFBNet → CornerNet → PFPNet → Pelee → HKRM → R-DAD → M2Det …
CVPR' 18 ECCV' 18 ECCV' 18 ECCV' 18 NIPS' 18 NIPS' 18 AAAI' 19 AAAI' 19

**Object Detection** = Classification + Localization

: 여러가지 object에 대한 classification과 그 object들의 위치 정보를 파악



## Abstract

*Object detection performance, as measured on the canonical PASCAL VOC dataset, has plateaued in the last few years. The best-performing methods are complex ensemble systems that typically combine multiple low-level image features with high-level context. In this paper, we propose a simple and scalable detection algorithm that improves mean average precision (mAP) by more than 30% relative to the previous best result on VOC 2012—achieving a mAP of 53.3%. Our approach combines two key insights: (1) one can apply high-capacity convolutional neural networks (CNNs) to bottom-up region proposals in order to localize and segment objects and (2) when labeled training data is scarce, supervised pre-training for an auxiliary task, followed by domain-specific fine-tuning, yields a significant performance boost. Since we combine region proposals with CNNs, we call our method* R-CNN*: Regions with CNN features. We also compare R-CNN to OverFeat, a recently proposed sliding-window detector based on a similar CNN architecture. We find that R-CNN outperforms OverFeat by a large margin on the 200-class ILSVRC2013 detection dataset. Source code for the complete system is available at* http://www.cs.berkeley.edu/~rbg/rcnn.

## Abstract

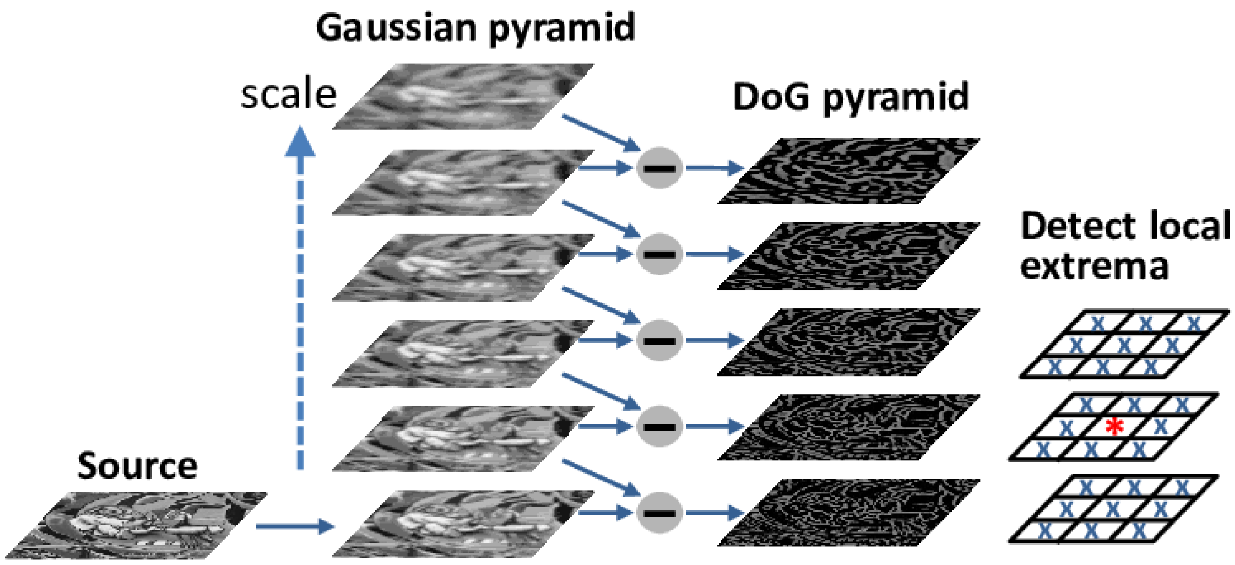
1. 상향식 방식의 지역 제안 (Region proposal)에 CNN 적용 (Region Proposal + CNN)

2. Labeled 훈련 데이터가 적은 경우 ➔ Domain-specific fine-tuning 을 통한 지도 사전 훈련을 적용 (Pre-train + Fine-tuning)

## Introduction

CNN으로 Image Classification에서 좋은 성능을 냈지만 Object detection 에서는? ➔ 이전 Object Detection Task에 사용되던 SIFT, HOG와 비교했을 때 성능이 많이 향상됨

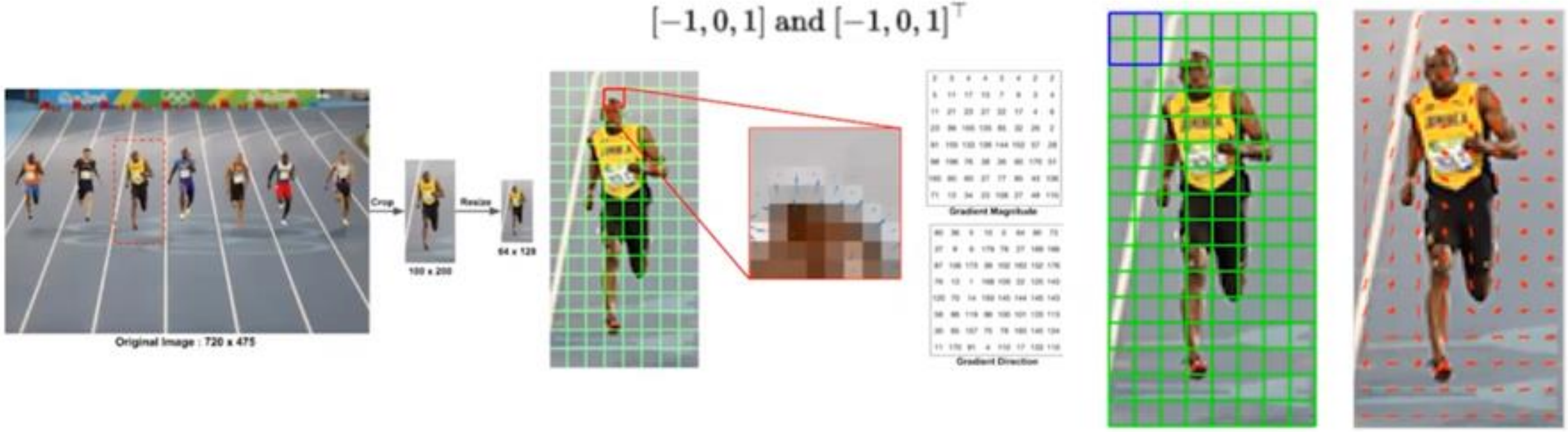
# SIFT & HOG - SIFT



이미지의 scale(크기) 및 rotation(회전)에 robust한 특징점을 추출하는 알고리즘

1. Scale-space extrema detection
2. Keypoint localization
3. Orientation assignment
4. Keypoint descriptor

# SIFT & HOG - HOG



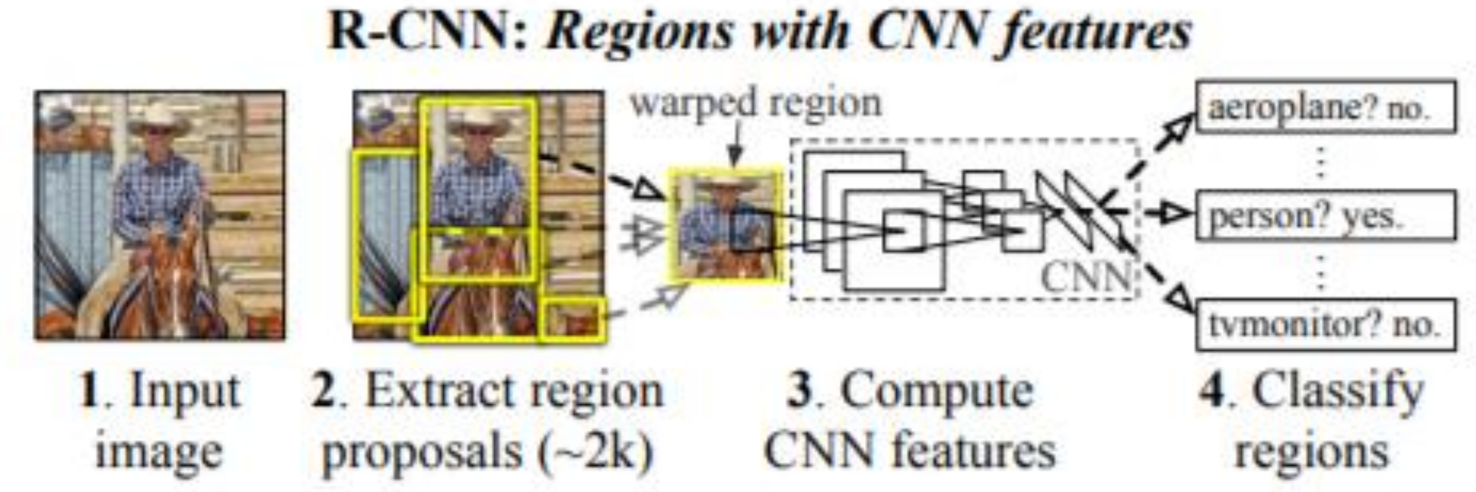
$[-1, 0, 1]$ and $[-1, 0, 1]^\top$

픽셀의 변화량의 각도와 크기를 고려하여 히스토그램 형태의 feature를 추출하는 방법

전처리 ➔ Gradient 이미지 계산 ➔ 8*8 셀 내의 histogram of gradient 계산

➔ 16*16 블록 정규화 ➔ Hog 형상 벡터 계산

# R-CNN의 대략적인 흐름



2. Selective search방식을 이용해서 region proposal 후 warp

3. Large CNN을 이용하여 지역의 feature vector를 뽑아냄

4. SVM을 학습시킴 (mAP의 성능 향상을 위해 softmax 대신 사용)

    4-1. 이후 CNN을 통해 나온 bbox에 대해 bbox regression 을 진행

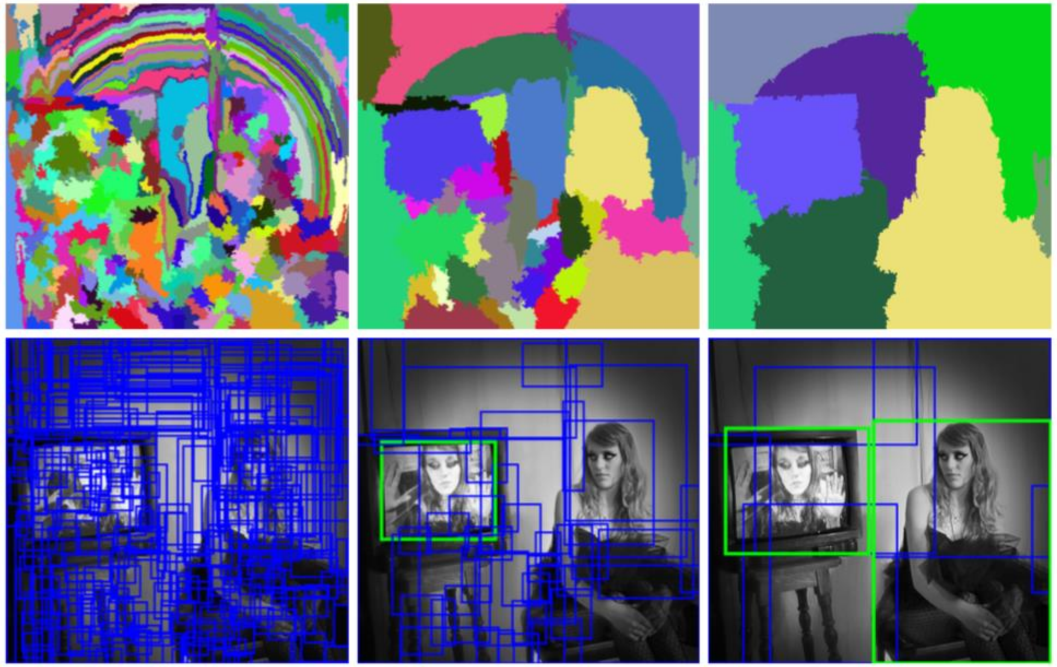
# Region Proposal이 나오게 된 과정

- Localizing의 필요성 : annotated 된 dataset이 많지 않았음

- Approach 1 : localization as a regression problem : Not efficient
- Approach 2 : Sliding window detector : to much computation
- Approach 3 : Region Proposal

  + pre-training + domain specific fine tuning

# Region Proposal – Selective Search



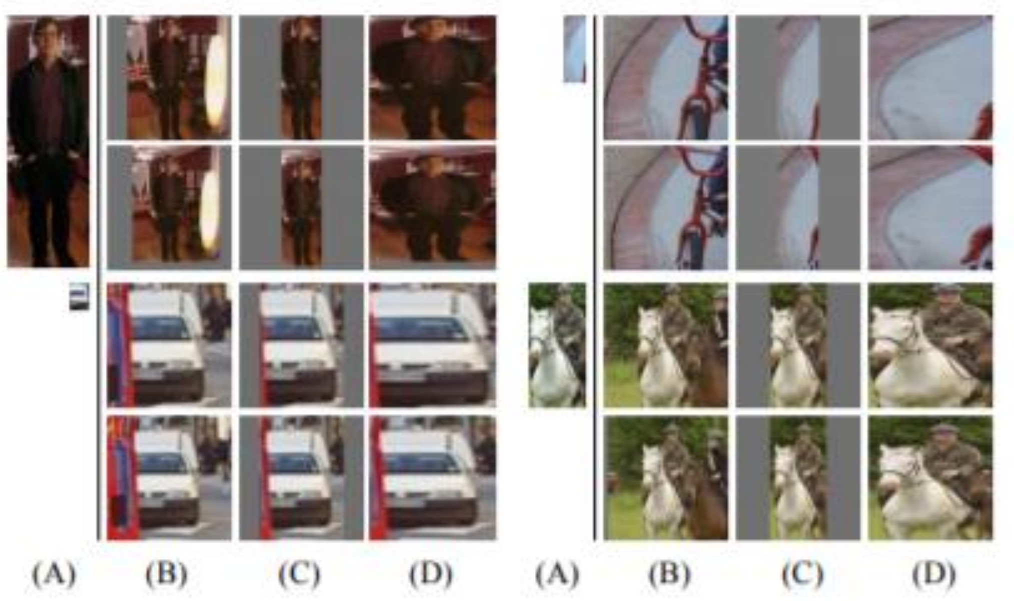
Hiearchical algorith으로 물체의 서로 다른 크기와 모호한 경계를 결정

1. “Efficient GraphBased Image Segmentation” 방법으로 초기 영역들을 지정
2. 이웃한 bbox 끼리의 유사도 계산
3. Greedy 알고리즘으로 유사도가 높은 bbox끼리 합쳐짐.

Fast Mode를 사용

Selective Search를 통해 약 2000개의 region proposal을 제안

# Region Proposal - Warping



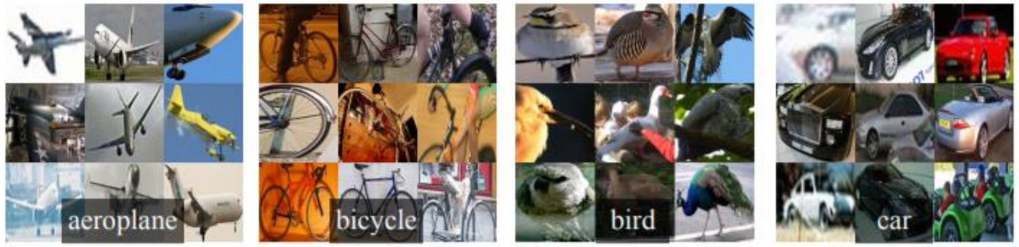


**Figure 2: Warped training samples** from VOC 2007 train.

CNN의 고정된 input size (227*227)를 맞춰줌

(A) : Original Input

(B) : Context(object+background) 맞춤 square로 resize

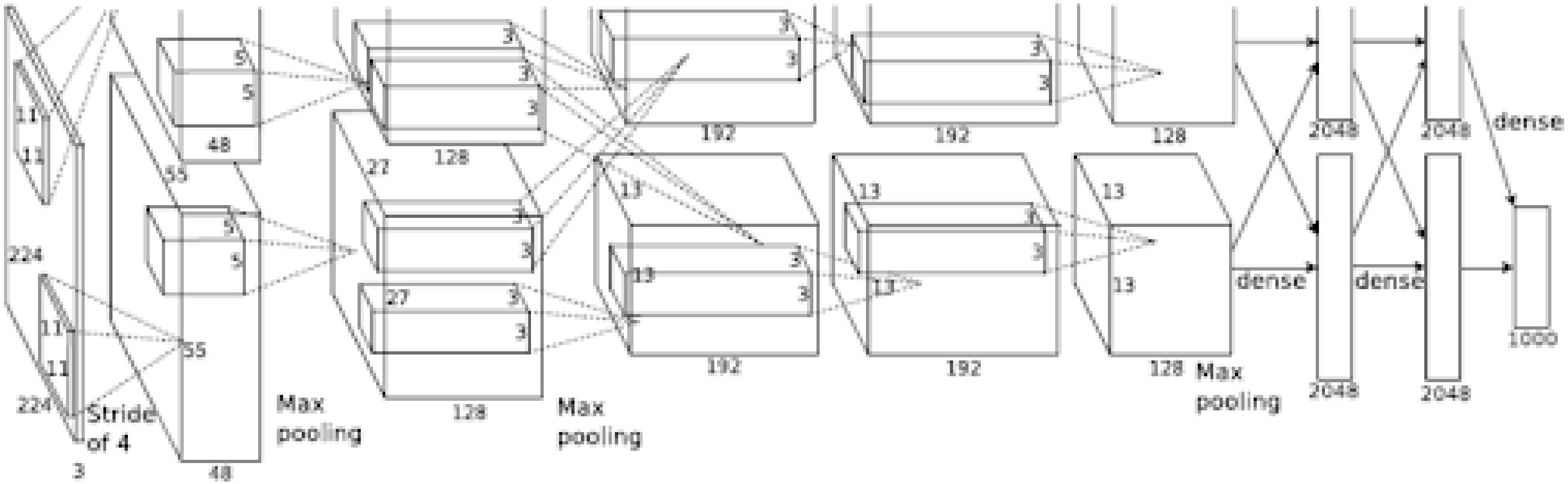
(C) : Context를 제외한 가장 tight한 square

( background를 최대한 제거 )

(D) : Warp ( Square 맞춤 사이즈로 늘림 ) ( p = 16 )

Context padding을 통해 mAP를 3~5% 개선

# Feature Extraction

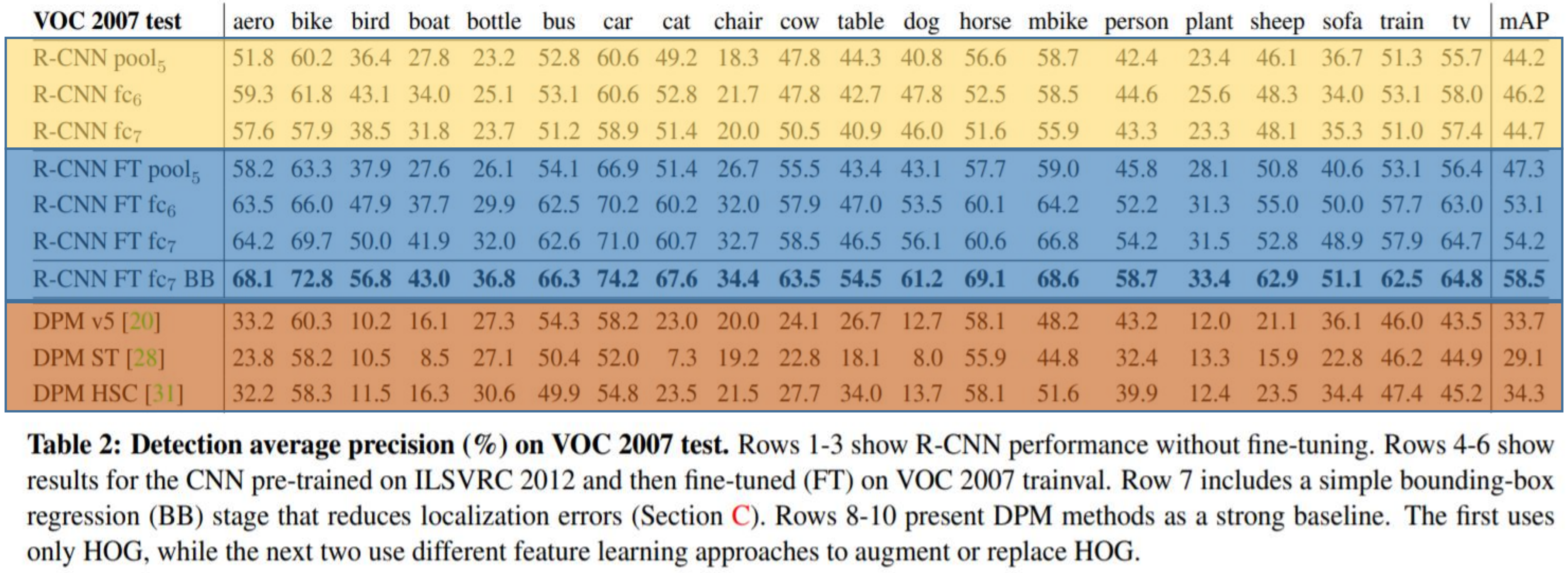


Caffe 구조를 통해 각각의 region proposal에 대해 4096 차원의 Feature Vector extract

Feature Extraction 시 AlexNet의 architecture를 차용 : 5 conv layers + 2 Fully Connected Layers

마지막에는 SVM을 사용하여 classification 진행

# Pre-training & Domain Specific Fine Tuning



| 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.

Without Fine-Tuning

Pre-Trained + Fine-Tuning

HOG (+a)

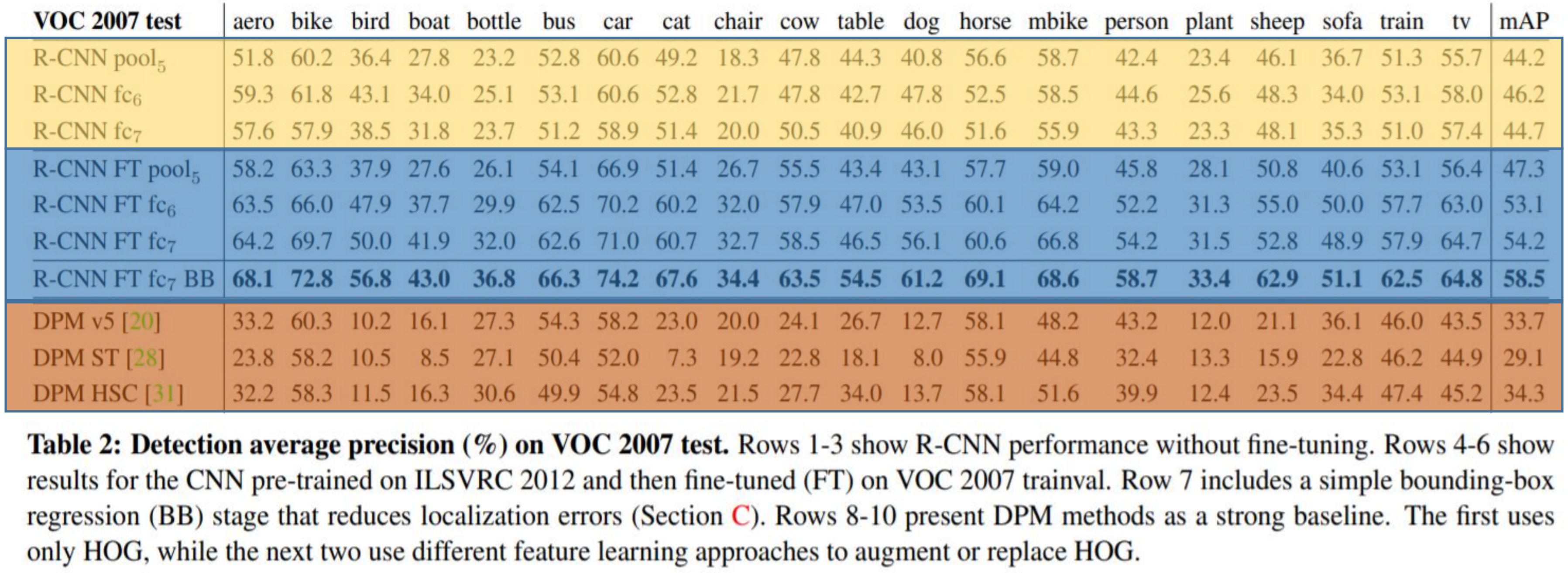
Warped된 proposal에 대해 SGD(Stochastic Gradient Descent)를 이용

전체 카테고리개수 (N) + 1 개(for background)의 분류를 진행 ( VOC(N = 20), ILSVRC(N=200) )

Ground-Truth bbox와 IoU>=0.5 이상인 proposal에 대해 training 진행

각각의 SGD iteration에 대해서 32개의 positive window + 96개의 background window = 128 size mini batch

# Pre-training & Domain Specific Fine Tuning



| 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.

Without Fine-Tuning

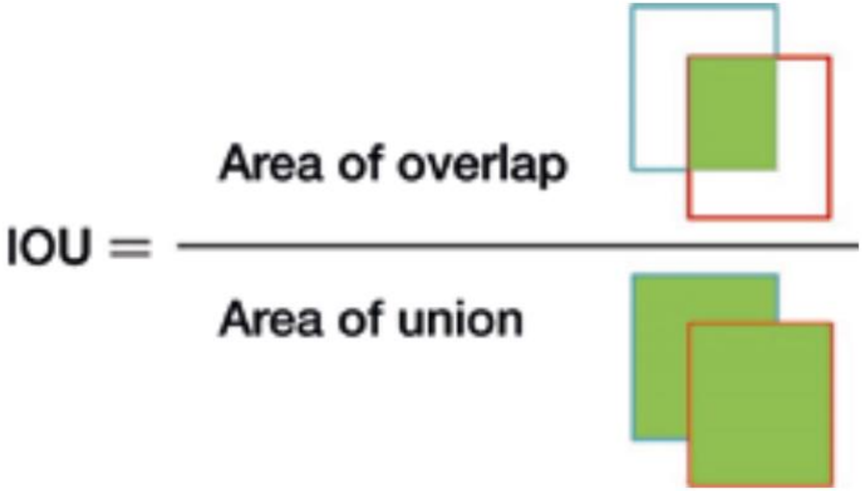
Pre-Trained + Fine-Tuning

HOG (+a)

Pre-Training (ILSVRC 2012) + Fine Tuning (VOC 2007) ➔ 약 8%의 mAP 개선

Annotated Data가 scarce하지만 high – capacity CNN architecture에서의 성능을 높임

# IoU ( Intersection over Union )



두 bbox가 얼마나 겹치는지를 0~1 사이의 값으로 나타냄

논문에서는 Ground Truth와 Proposed Region 사이의 값이 0.5이상인 경우를 객체로 보고 같은 class로 라벨링

# NMS ( Non – maximum suppression )

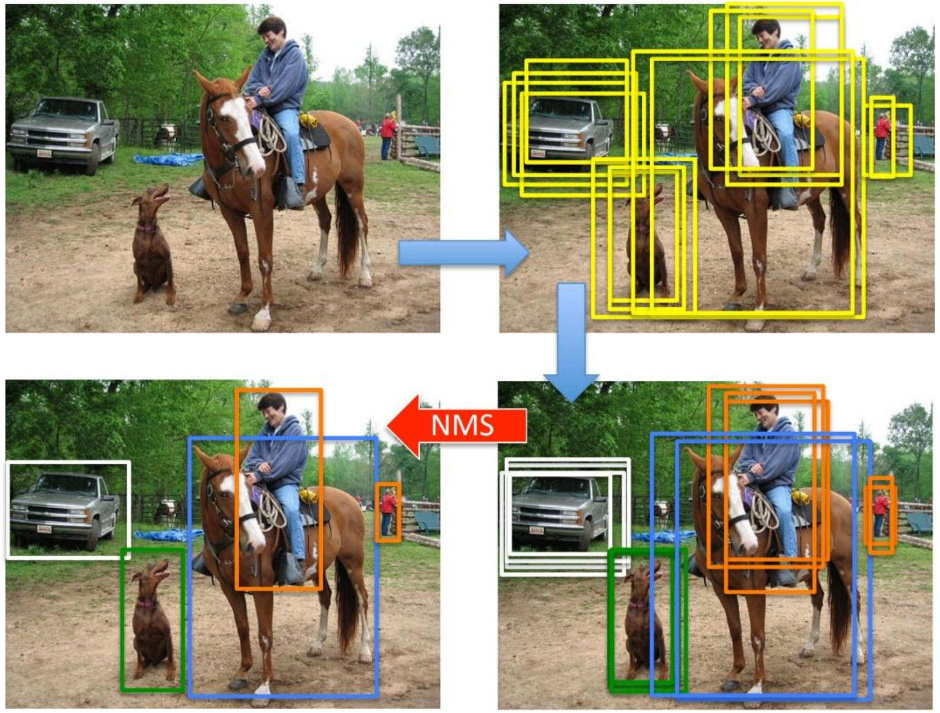


Figure 3. In object detection, first category independent region proposals are generated. These region proposals are then assigned a score for each class label using a classification network and their positions are updated slightly using a regression network. Finally, non-maximum-suppression is applied to obtain detections.

**중복되는 BBox를 제거하기 위한 기법**

1. 동일한 클래스에 대해 내림차순으로 confidence를 정렬
2. 가장 confidence가 높은 bbox와 IoU가 일정 이상인 bbox는 동일한 물체를 detect했다고 판단 후 삭제.

# Result

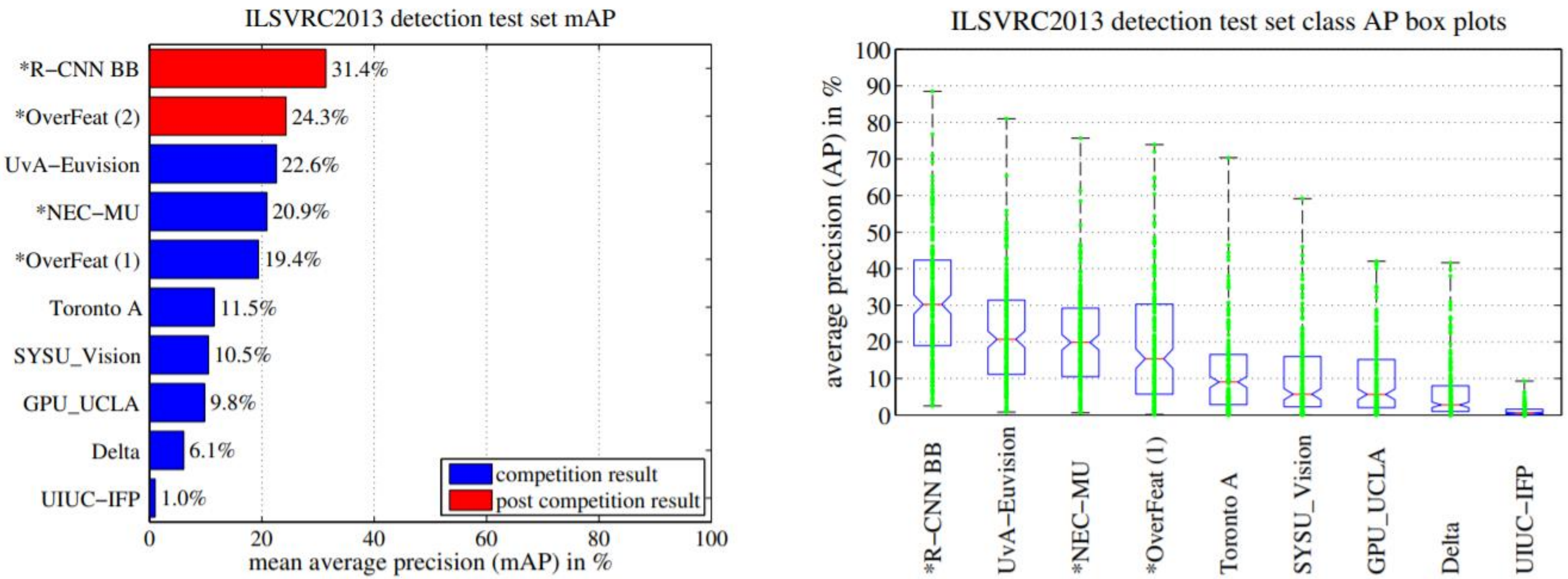
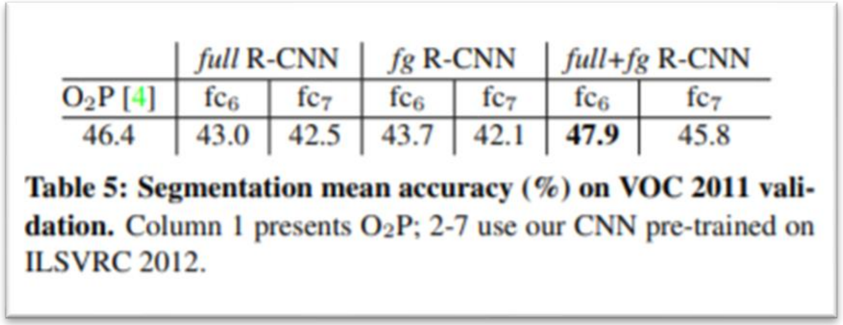


**Figure 3: (Left) Mean average precision on the ILSVRC2013 detection test set.** Methods preceeded by * use outside training data (images and labels from the ILSVRC classification dataset in all cases). **(Right) Box plots for the 200 average precision values per method.** A box plot for the post-competition OverFeat result is not shown because per-class APs are not yet available (per-class APs for R-CNN are in Table 8 and also included in the tech report source uploaded to arXiv.org; see `R-CNN-ILSVRC2013-APs.txt`). The red line marks the median AP, the box bottom and top are the 25th and 75th percentiles. The whiskers extend to the min and max AP of each method. Each AP is plotted as a green dot over the whiskers (best viewed digitally with zoom).

(좌) : Bbox Regression을 한 R-CNN이 OverFeat 보다 약 7% mAP 향상  (우) : AP 성능 box plot 비교

단, overfeat의 경우 성능이 9배 빠르게 되는데 그 이유는 image warp를 하지 않아서 overlapping 되는 window간의 computation이 공유되기 때문

# Result

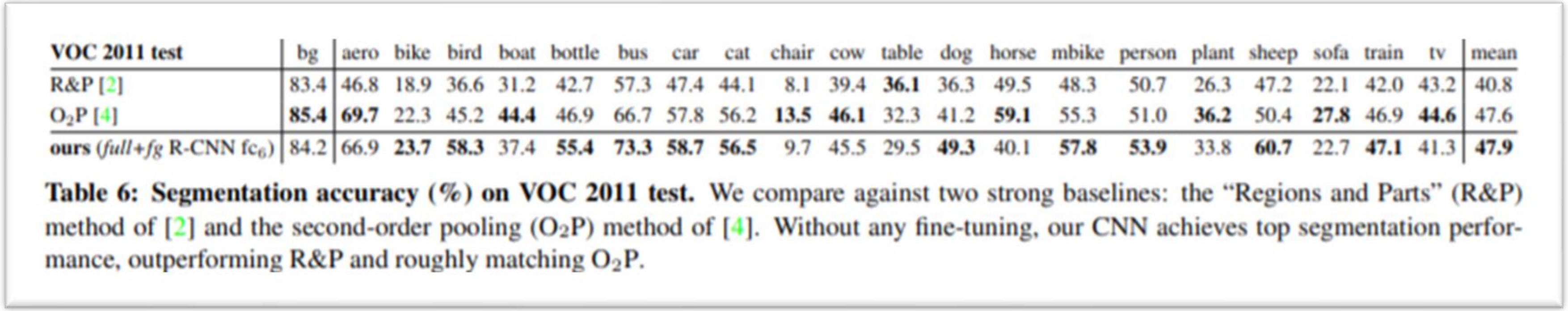


| | *full* R-CNN | | *fg* R-CNN | | *full+fg* R-CNN | |
|---|---|---|---|---|---|---|
| $O_2P$ [4] | $fc_6$ | $fc_7$ | $fc_6$ | $fc_7$ | $fc_6$ | $fc_7$ |
| 46.4 | 43.0 | 42.5 | 43.7 | 42.1 | **47.9** | 45.8 |

**Table 5: Segmentation mean accuracy (%) on VOC 2011 validation.** Column 1 presents $O_2P$; 2-7 use our CNN pre-trained on ILSVRC 2012.

full+fg R-CNN이 fc6까지만 사용한 경우 O2P보다 좋은 성능을 냄

배경은 살려두고 foreground에는 masking을 해준 것



| **VOC 2011 test** | bg | aero | bike | bird | boat | bottle | bus | car | cat | chair | cow | table | dog | horse | mbike | person | plant | sheep | sofa | train | tv | mean |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| R&P [2] | 83.4 | 46.8 | 18.9 | 36.6 | 31.2 | 42.7 | 57.3 | 47.4 | 44.1 | 8.1 | 39.4 | **36.1** | 36.3 | 49.5 | 48.3 | 50.7 | 26.3 | 47.2 | 22.1 | 42.0 | 43.2 | 40.8 |
| $O_2P$ [4] | **85.4** | **69.7** | 22.3 | 45.2 | **44.4** | 46.9 | 66.7 | 57.8 | 56.2 | **13.5** | **46.1** | 32.3 | 41.2 | **59.1** | 55.3 | 51.0 | **36.2** | 50.4 | **27.8** | 46.9 | **44.6** | 47.6 |
| **ours** (*full+fg* R-CNN $fc_6$) | 84.2 | 66.9 | **23.7** | **58.3** | 37.4 | **55.4** | **73.3** | **58.7** | **56.5** | 9.7 | 45.5 | 29.5 | **49.3** | 40.1 | **57.8** | **53.9** | 33.8 | **60.7** | 22.7 | **47.1** | 41.3 | **47.9** |

**Table 6: Segmentation accuracy (%) on VOC 2011 test.** We compare against two strong baselines: the "Regions and Parts" (R&P) method of [2] and the second-order pooling ($O_2P$) method of [4]. Without any fine-tuning, our CNN achieves top segmentation performance, outperforming R&P and roughly matching $O_2P$.

full+fg+fc6 R-CNN을 R&P, O2P와 비교했을 때

21개 중 총 11개의 카테고리에 대해 outperform한 결과를 보임