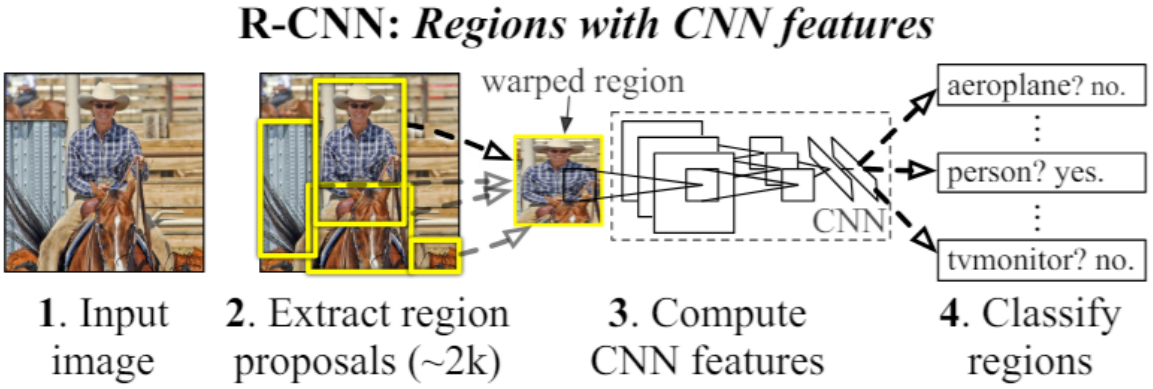
1. R-CNN

Abstract

- PASCAL VOC 데이터셋을 사용한 객체 탐지 성능은 정체되어 있었으며, 가장 높은 성능을 보이는 방법들은 일반적으로 다양한 저수준 이미지 특성과 고수준 컨텍스트를 결합한 복잡한 앙상블 시스템이었음
- R-CNN 논문은 VOC 2012에서 이전 최고 결과보다 평균 정밀도(mAP)를 30%이상 향상시킨 간단하면서도 확장 가능한 탐지 알고리즘을 제안함
- 두가지 주요 통찰력을 결합
    - 고용량의 컨볼루션 신경망(CNN)을 하향식 영역 제안에 적용하여 객체를 지역화하고 분할할 수 있음
    - 레이블이 지정된 교육 데이터가 부족할 때, 보조 작업을 위한 지도 학습 전처리 후 도메인 특화 파인 튜닝을 수행하면 상당한 성능 향상을 가져올 수 있음
- 영역 제안과 CNN을 결합하기 때문에 R-CNN(영역을 가진 CNN)특성으로 부름
- 슬라이딩 윈도우 탐지기인 OverFeat와 비교했고, R-CNN이 200 클래스 ILSVRC2013탐지 데이터셋에서 OverFeat보다 훨씬 높은 성능을 보인다는 것을 발견

Intoroduction

- 지난 10년동안 다양한 시각 인식 작업에서의 진보는 주로 SIFT와 HOG같은 특징 사용에 크게 의존해왔지만, 대표적인 시각 인식 작업인 PASCAL VOC 객체 검출에서의 성능을 보면, 2010년부터 2012년까지 진전이 더딘 것으로 인정받고 성공적인 방법들의 작은 변형을 사용하거나, 앙상블 시스템을 구축함으로써 소규모의 성과를 얻었음
- SIFT & HOG
    - 블록형 방향 히스토그램으로 대략적으로 원숭이의 시각 경로에서 첫 번째 대뇌 피질영역인 V1의 복잡한 세포와 연관될 수 있음
    - 인식이 몇 단계 아래에서 일어난다는 것을 알고 있으며, 시각 인식을 위해 더욱 유익한 특징을 계산하기 위한 계층적이고 다단계적인 과정이 있을 수 있음을 시사
- 90년대 CNN은 널리 사용되었으나 유행하지 못 했고, 2012년 Krizhevsky 등은 ImageNet 대규모 시각 인식 챌린지에서 훨씬 높은 이미지 분류 정확도를 보여주면서 CNN에 대한 관심을 다시 불러일으켰음 → 120만 개의 레이블이 붙은 이미지에서 큰 CNN을 훈련시키는 것과 LeCun의 CNN에 몇 가지 변형을 추가했음



1. 입력 이미지를 받음
2. 약 2000개의 생성된 영역 제안을 추출
   → 노란색 박스부분
3. 각 제안에 대해 큰 컨볼루션 신경망(CNN)을 사용하여 특징을 계산
   → 각 영역의 이미지를 변형시켜 CNN을 통해 특징을 추출
4. 각 영역을 클래스별 선형 SVM을 사용하여 분류
   - R-CNN은 PASCAL VOC 2010에서 평균 정밀도(mAP)53.7% 달성
   - ILSVRC2013 200 클래스 검출 데이터셋에서 R-CNN의 mAP는 31.4%로 이전 최고 결과였던 OverFeat의 24.3%보다 크게 향상

- 이 논문은 CNN이 PASCAL VOC에서 HOG같은 단순한 특징을 기반으로 한 시스템보다 훨씬 높은 객체 검출 성능을 이끌어 낼 수 있음을 처음으로 보여줌
  - 해당 결과를 달성하기 위해 두가지 문제에 초점을 맞춤
    1. 깊은 네트워크로 객체를 정확히 위치시킴
    2. 은 양의 주석이 달린 검출 데이터로 고용량 모델을 훈련시키는 것
- 이미지 분류와는 달리 검출은 이미지 내에서 객체를 위치시키는 것을 요구함
  - 물체 감지와 의미적 분할 모두에서 성공한 '영역을 이용한 이식' 패러다임 내에서 작동하여 CNN 지역화 문제를 해결
    - 위 이미지에서와 같이 입력 받은 이미지에 대해 약 2000개의 카테고리 독립영역의 이미지를 생성하고, CNN을 사용하여 각 제안에서 고정길이 특징 벡터를 추출한 다음, 카테고리별 선형 SVM을 사용하여 각 영역을 분류
    - 아핀 이미지 변형을 사용하여 영역의 모양에 관계없이 각 영역 제안에서 고정 크기의 CNN 입력을 계

- 대규모 보조 데이터 세트(ILSVRC)에 대한 지도된 사전 학습과 소규모 데이터 세트(PASCAL)에 대한 도메인별 미세 조정이 데이터가 부족할 때 고용량 CNN을 학습하는데 효과적인 패러다임이라는 것을 보여줌
  - 미세 조정을 통해 탐지 성능을 향상시키면 mAP성능이 8% 포인트 향상
- 유일한 클래스별 연산은 상대적으로 작은 행렬-벡터 곱셈과 탐욕적 최대억제
  - 이 연산 특성은 모든 카테고리에 걸쳐 공유되는 특징과 이전에 사용된 영역특징보다 두 오더 낮은 차원에서 비롯

Object Detection with R-CNN

- 객체 검출 시스템은 세 가지 모듈로 구성
  1. 카테고리에 독립적인 영역 제안을 생성
     a. 검출기가 사용할 수 있는 후보 검출 집합을 정의
  2. 각 영역에서 고정 길이의 특징 벡터를 추출하는 큰 컨볼루션 신경망
  3. 클래스별 선형 SVM집합
- Module Design

Figure 2: Warped training samples from VOC 2007 train.

  - 객체성, 선택적 검색, 카테고리에 독립적인 객체 제안, 제한된 파라메트릭 최소컷(CPMC), 다중 스케일 조합 그룹화 등은 정기적으로 배치된 정사각형 영역에 CNN을 적용하여 유사세포를 감지하는데 이는 영역 제안의 특별한 경우임
    - R-CNN은 특정 영역 제안 방법에 대해 무지한 상태지만, 이전 검출 작업과의 비교를 통제하기 위해 선택적 검색을 사용함
    - 선택적 검색 : 이미지에서 잠재적인 관심 영역을 도출하기 위해 다양한 스케일과 색상 전략을 사용
  - 특징추출
    - CNN의 Caffe 구현을 사용하여 각 영역 제안에서 4096차원 특징 벡터를 추출

        - 특징은 평균이 빼진 227 x 227 RGB 이미지를 다섯 개의 컨볼루션 레이어와 두 개의 완전 연결 레이어를 통해 전파함으로써 계산
    - 영역 제안에 대한 특징을 계산하기 위해 그 영역의 이미지 데이터를 CNN과 호환되는(277 x 277) 픽셀 크기로 변환
        - 후보 영역의 크기나 종횡비에 상관없이, 그 주위의 타이트한 경계 상자 안의 모든 픽셀을 필요한 크기로 변형
            - 젼형하기 전, 타이트한 경계 상자를 확장하여 변형된 크기에서 원래 상자 주위에 정확히 p 픽셀의 변형된 이미지 컨텍스트가 있도록함(논문에서는 p = 16을 사용)
                - 원본 영역 주위에 픽셀을 추가하여 모든 영역이 같은 크기의 컨텍스트를 갖도록함
            - 위 이미지는 변형된 훈련 영역의 무작위 샘플을 보여줌

Object Proposal Transformation

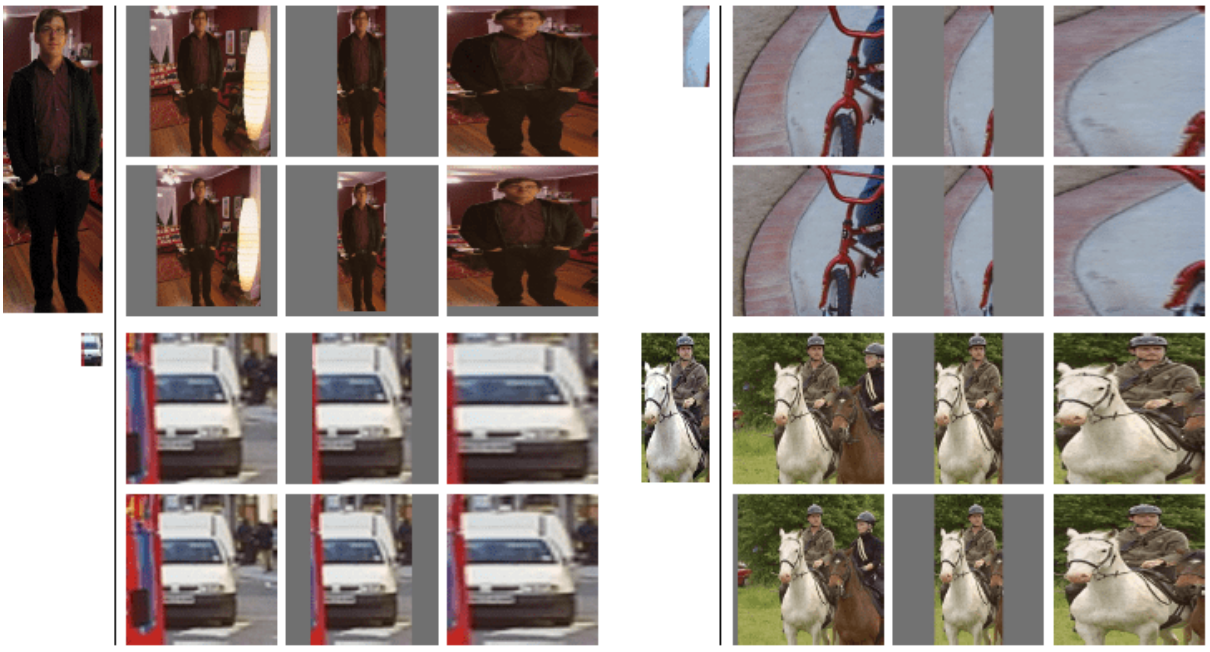


Figure 7: Different object proposal transformations. (A) the original object proposal at its actual scale relative to the transformed CNN inputs; (B) tightest square with context; (C) tightest square without context; (D) warp. Within each column and example proposal, the top row corresponds to $p = 0$ pixels of context padding while the bottom row has $p = 16$ pixels of context padding.

- CNN은 고정된 input size의 image를 넣어야 합니다. 직사각형이든 더 크거나 작은 사이즈를
  227 * 227 이미지로 변환할 수 있는 transformation 변형을 제시함
- padding(p) = 16 일 때 3 ~ 5 mAP정도의 성능이 더 좋았음
- input size에서 이미지의 사이즈가 오히려 더 작을경우 빈 data공간은 mean으로 대체
- Test-time Detection
  - selective searchs fast mode 방식으로 test image에서 약 2000개의 region proposal을 추출한 정보 + Warp + CNN + SVM으로 점수를 측정 후 greedy non-maximum suppression을 적용
- Run-time analysis
  - R-CNN은 2가지 특성에서 효율성을 증명함

        - 모든 CNN Parameter는 모든 카테고리를 공유
        - CNN을 활용한 feature vector는 low-dimension(저차원)
    - features, SVM weights, non-maximum suppression 사이의 dot product가 유일한 computatuon이므로 hashing을 이용한 방식에 의존하지 않은 저차원 feature 계산이 가능함
- Training
    - Supervised Pre-training
        - ILSVRC 2012 classification dataset을 pre-train에 사용
    - Domain-specific fine-tuning
        - SGD를 이용한 wrap된 region proposals만을 사용했으며,
          AlexNet의 마지막 1000 classification을 N+1(N개의 클래스, 1개의 Background) classification으로 수정함 나머지는 동일
    - if ground-truth box, IoU overlap ≥ 0.5 : Positive else : negative
    - SGD learning rate = 0.001
- Object category classifiers
    - IOU overlap threshold를 통해 negative를 잘 설정하는 것이 중요
    - 0.3이 임계값으로 선택되었는데 만약 임계값이 0.5면 5 mAP, 0이면 4mAP가 감소
    - standard hard negative mining method를 통해 빠르게 수렴하는 방식으로 채택

Positive vs. negative examples and softmax

- fine-tunning을 할 때와 training SVM을 할 때의 positive, negative example을 다르게 설정하는 이유
    - fine-tuning에서는 IoU가 0.5이상이면 positive이고 그 외 negative로 놓는 반면 SVM으로
      훈련 시킬 때는 0.3 IoU이상이면 Positive이고 그 외 negative로 놓음으로써 더 성능이 잘 나옴
- fine-tuning 이후에 SVM으로 훈련하는 이유
    - softmax 방식의 regression classifier는 VOC 2007 기준 mAP 50.9%로 성능을 떨어트렸기 때문에 SVM으로 훈련 classifier를 선택

- 훈련과정

▼ Fine-Tuning 과정

- 사전 훈련된 모델
  - 먼저 큰 데이터셋에서 사전 훈련된 CNN 모델을 사용, 이 모델은 일반적인 이미지 인식 작업에 대한 지식을 이미 가지고 있음
  - Region Proposal
    - 입력 이미지에서 Region Proposal 알고리즘(예:Selective Search)을 사용하여 여러 객체 후보 영역을 추출
  - Feature Extraction
    - 각 후보 영역(region proposal)을 CNN에 입력하여 고차원 feature map을 추출
  - Fine-Tuning
    - 추출된 feature map을 사용하여 네트워크의 마지막 레이러를 재훈련 각 후보 영역에 대해 IoU값이 0.5이상인 경우를 positive 예제로, 그 외에는 negative 예제로 설정
    - 네트워크는 각 클래스에 대한 확률을 추력하도록 softmax 분류기를 사용
    - 손실 함수로는 일반적으로 교차 엔트로피(cross-entropy) 손실 사용
  - Backpropagation
    - 역전파를 통해 네트워크의 가중치를 업데이트하여 모델을 최적화

▼ SVM 훈련과정

- Feature Extraction
  - Fine-Tuning을 마친 CNN을 사용하여 각 region proposal에 대해 feature vector를 추출
- Positive/Negative Example 설정
  - SVM훈련을 위해 positive 예제를 IoU값이 0.3 이상인 경우로 설정하고 그 외는 negative 예제로 설정
  - 이는 fine-tuning에서 설정한 기준보다 더 낮아 다양한 경우를 학습하게 함
- SVM 훈련

        - 추출된 feature vector를 사용하여 각 클래스에 대한 SVM 분류기를 훈련
        - SVM은 각 클래스에 대해 최적의 초평면(hyperplane)을 찾아 데이터를 분류
        - 이때, 서포트 벡터(support vector)로 불리는 데이터 포인트들이 초평면을 정의
    - Margin Maximization
        - SVM은 클래스 간의 마진(margin)을 최대화하여 분류기의 일반화 성능을 높임
        - 손실 함수로는 힌지 손실(hinge loss)을 사용
- 요약
    - Fine-Tuning
        - 사전 훈련된 CNN을 사용하여 특정 작업에 맞게 미세 조정
        - Resion Proposal을 통해 후보 영역 추출
        - 추출된 feature map을 기반으로 네트워크의 마지막 레이어를 재훈련
        - 손실 함수로 교차 엔트로피 사용
    - SVM 훈련
        - Fine-Tuning을 마친 CNN으로부터 추출된 feature vector를 사용
        - Positive 예제와 negative 예제를 새로 설정 (IoU 0.3 이상 : Positive)
        - 각 클래스에 대해 SVM 분류기를 훈련하여 최적의 초평면을 찾아 분류
        - 손실함수로 힌지 손실 사용

Visualization, abalation, and modes of error

- Visualizing learned features
    - region proposals에 대한 unit activation 계산 → activation 내림차순으로 정렬 → non-maximum suppression 수행 → 고득점 region 표시
    - 위 과정을 통해 각 layer마다의 unit들에 대해 어떻게 학습하는지 볼 수 있음
    - 이를 통해 네트워크는 모양, 질감, 색, 물성의 분산 표현들을 결합하여 표현을 배우는 것으로 나타남 → 작은 특징들로부터 큰 특징을 배움
- Abalation studies

- Performance layer-by-layer,without fine-tuning
  - Detection 성능에 중요한 layer를 이해하기 위해 fine-tuning없이 수행했을 때 fc6, fc7 없이도 비교적 잘 수행됨
  - 이를 통해 중요한 부분은 Convolutional layer에 있다는 사실을 알게되었고 그리고 이것은 DPM에서 pool5 features 위에서 sliding-window detector 를 수행할 수 있게 함

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.

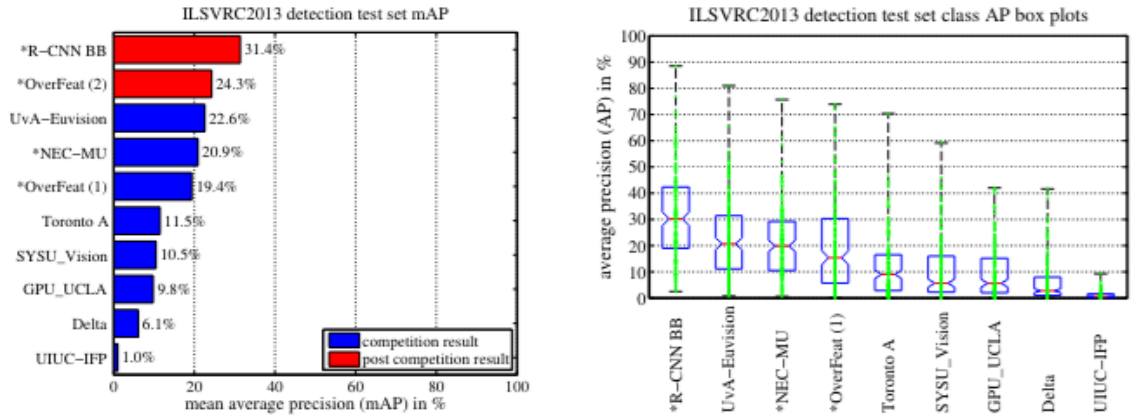


Figure 3: (Left) Mean average precision on the ILSVRC2013 detection test set. Methods preceeded by * use outside training data (images and labels from the ILSVRC classification dataset in all cases). **(Right) Box plots for the 200 average precision values per method.** A box plot for the post-competition 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).

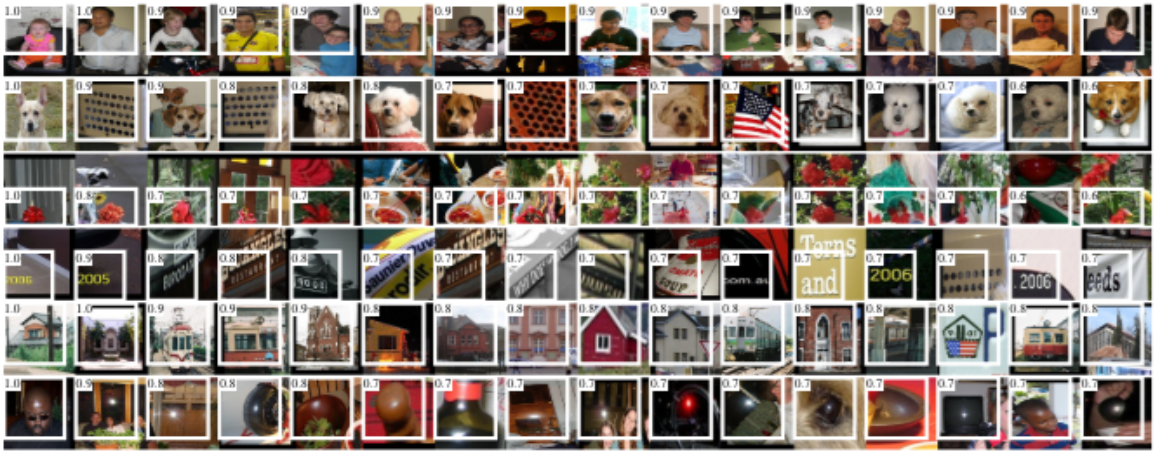


Figure 4: Top regions for six pool$_5$ units. Receptive fields and activation values are drawn in white. Some units are aligned to concepts, such as people (row 1) or text (4). Other units capture texture and material properties, such as dot arrays (2) and specular reflections (6).

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].

- perfomance layer-by-layer, with fine-tuning
  - 8% 향상된 54.2%의 결과를 Table2를 통해 확인 가능
- Comparison to recent feature learning methods
  - HOG-Based DPM과의 비교를 수행하는 부분으로 R-CNN이 더 좋다는 성능 지표를 나타냄
  - DPM v5 : HOG만 사용
  - DPM ST : Sketch Token 히스토그램 특성을 사용한 증식 augment
  - DPM HSC : HOG 대신 histogram of sparse code(HSC)사용
- Network arxchitectures
  - 기존의 방식 모델인 T-Net을 O-Net으로 통칭하여 같은 방식을 적용, O-Net이 훨씬 좋음
  - 단점 : 시간이 7배 더 걸림
- Bounding-box regression
  - localization error를 줄일 수 있는 방식으로 DPM에서 사용한 dounding box regression 방식을 도입
  - Bounding-box regression
    - localizational performance를 향상시키기 위해 Bounding-Box Regression을 사용하고자 했으며 이는 selective search proposal

scoring 후에 새 bounding box를 예측하고자 할 때 사용

$$P = (P_x, P_y, P_w, P_h)$$

- 중앙 값을 x,y로 설정 각 Grounding-truth bounding box를 G로 명시
- box P를 G에 대응할 수 있도록 변형시키는 것이 목적
  - 4개의 function으로 P에 대한 x, y, w, h의 함수 d를 사용하여 변형, 예측된 ground-truth box를 G^로 설정

$$\hat{G}_x = P_w d_x(P) + P_x$$
$$\hat{G}_y = P_h d_y(P) + P_y$$
$$\hat{G}_w = P_w \exp(d_w(P))$$
$$\hat{G}_h = P_h \exp(d_h(P))$$

$$d_x(P) = proposal$$
P의 $pool5\ feature$들에 대한 선형함수로$\Phi_5(P)$아래와 같이 표현가능
$$d_x(P) = w_x^T \Phi_5(P)$$
이를통해 $\mathbf{w}_*$ 를 최적화를 찾는 $ridge\,regression$을 사용
$$\mathbf{w}_* = \arg\min_{\hat{\mathbf{w}}_*} \sum_i (t_*^i - \hat{\mathbf{w}}_*^T \Phi_5(P^i))^2 + \lambda \|\hat{\mathbf{w}}_*\|^2$$

- t = regression traget

$$t_x = (G_x - P_x)/P_w$$
$$t_y = (G_y - P_y)/P_h$$
$$t_w = log(G_w/P_w)$$
$$t_h = log(G_h/P_h).$$

- regularization은 중요하지 않음
- P를 아무거나 선택할 수 없기 때문에 Ground Truth box G의 IoU Overlap을 최대화하는 것을 P로 선택 → 임계값 0.6 선택
- bounding box regression은 여러번 수행해도 결과 향상이 없어 한번만 사용

◦ 요약

- Bounding-box 설정
  - 후보 영역과 Ground-truth bounding box의 정의
  - 변환 함수 : 후보 영역을 Ground-truth에 맞추기 위한 변환 함수 정의

- Ridge Regression : 변환 함수를 최적화하기 위한 Ridge Regression 사용
- Regression Target : 변환 함수가 예측해야 하는 목표 값 설정

▪ 주요 포인트

- Regularization : Ridge Regression의 정규화 항은 중요한 역할을 하지 않음
- IoU Overlap : 후보 영역을 선택할 때 Ground-truth와의 IoU를 최대화 하는 것을 목표로 함
- Bounding-box Regression을 여러 번 수행해도 결과가 크게 향상되지 않기 때문에 한 번만 수행

ILSVRC2013 Detection DataSet

- ILSVRC 2013 Detection dataset은 PASVAL VOC와 달리 어떻게 선택할지에 대한 어려운 부분이 있기에 Section으로 나눠 설명

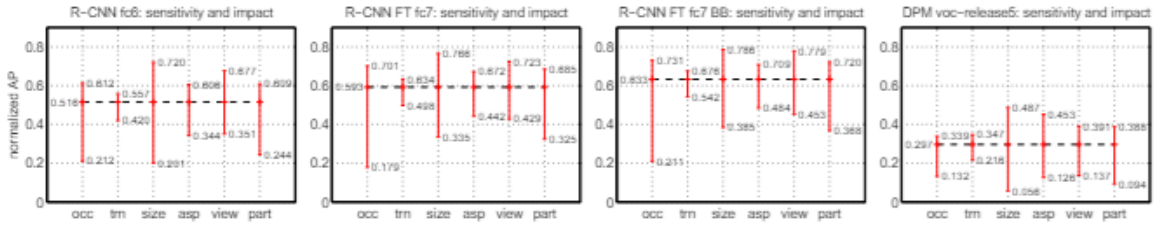


Figure 6: Sensitivity to object characteristics. Each plot shows the mean (over classes) normalized AP (see [23]) for the highest and lowest performing subsets within six different object characteristics (occlusion, truncation, bounding-box area, aspect ratio, viewpoint, part visibility). We show plots for our method (R-CNN) with and without fine-tuning (FT) and bounding-box regression (BB) as well as for DPM voc-release5. Overall, fine-tuning does not reduce sensitivity (the difference between max and min), but does substantially improve both the highest and lowest performing subsets for nearly all characteristics. This indicates that fine-tuning does more than simply improve the lowest performing subsets for aspect ratio and bounding-box area, as one might conjecture based on how we warp network inputs. Instead, fine-tuning improves robustness for all characteristics including occlusion, truncation, viewpoint, and part visibility.

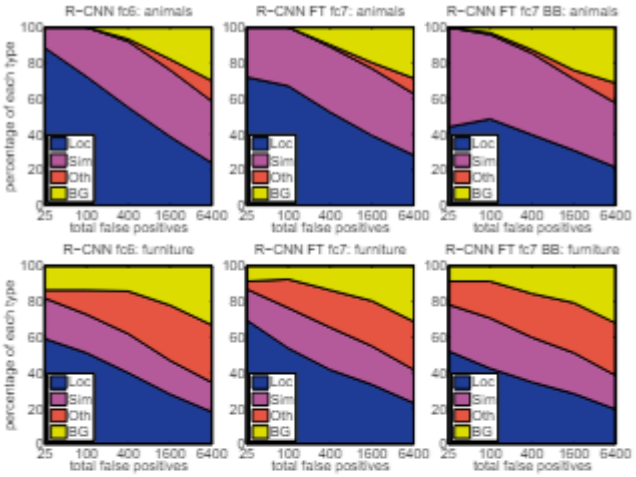


Figure 5: Distribution of top-ranked false positive (FP) types. Each plot shows the evolving distribution of FP types as more FPs are considered in order of decreasing score. Each FP is categorized into 1 of 4 types: Loc—poor localization (a detection with an IoU overlap with the correct class between 0.1 and 0.5, or a duplicate); Sim—confusion with a similar category; Oth—confusion with a dissimilar object category; BG—a FP that fired on background. Compared with DPM (see [23]), significantly more of our errors result from poor localization, rather than confusion with background or other object classes, indicating that the CNN features are much more discriminative than HOG. Loose localization likely results from our use of bottom-up region proposals and the positional invariance learned from pre-training the CNN for whole-image classification. Column three shows how our simple bounding-box regression method fixes many localization errors.

- Dataset Overview
  - train : 395,918
  - val : 20,121
  - test : 40,152
  - val과 test는 완전히 주석처리가 되어 있지만 train dataset의 경우 negative image들을 포함한 완전하지 않은 주석처리의 상태이기 때문에 hard negative mining이 불가능
  - 이를 위한 전략으로 val과 train dataset 중 positive images들을 사용하여 val1, val2로 나누고 class마다의 숫자 불균형을 맞추기 위한 a randomized local search를 사용
- Region Proposal
  - PASCAL과 같이 Selective Search의 Fast Mode를 사용하여 val1, val2, test에 적용했지만 selective search는 크기에 불변함으로 ILSVRC의 각 이미지의 폭을 500pixel로 미리 맞춤

- Training data
  - Negative example을 전혀 사용하지 않은 상태로 RCNN의 3가지 절차인 CNN Fine-Tuning, SVM traing, bounding-box regressor traning을 수행
- Validation on evaluation
  - 최대한 tuning 없이 R-CNN의 결과를 보기 위한 2가지 파일을 제출
    - bounding-box regression과 함께한 데이터와 그렇지 않은 상태의 데이터
- Ablation study
  - Ablation study란 모델이나 알고리즘의 feature들을 제거하면서 성능을 연구하는 것으로 아래 이미지에서 관찰이 가능

test set	val_2	val_2	val_2	val_2	val_2	val_2	test	test
SVM training set	val_1	$val_1+train_{.5k}$	$val_1+train_{1k}$	$val_1+train_{1k}$	$val_1+train_{1k}$	$val_1+train_{1k}$	$val+train_{1k}$	$val+train_{1k}$
CNN fine-tuning set	n/a	n/a	n/a	val_1	$val_1+train_{1k}$	$val_1+train_{1k}$	$val_1+train_{1k}$	$val_1+train_{1k}$
bbox reg set	n/a	n/a	n/a	n/a	n/a	val_1	n/a	val
CNN feature layer	fc_6	fc_6	fc_6	fc_7	fc_7	fc_7	fc_7	fc_7
mAP	20.9	24.1	24.1	26.5	29.7	**31.0**	30.2	**31.4**
median AP	17.7	21.0	21.4	24.8	29.2	**29.6**	29.0	**30.3**

Table 4: ILSVRC2013 ablation study of data usage choices, fine-tuning, and bounding-box regression.

  - train의 데이터 수를 늘릴고 fine-tuning과 bounding-box regression을 할수록 더 좋아지는 mAP 관찰 가능