



[논문리뷰] Faster-RCNN [2015]

Introduction

☀ 본 논문에서는 RPN(Region Proposal Network) 기법을 제안합니다. RPN은 객체 탐지 네트워크와 함께, 합성곱 피쳐들을 공유하기 때문에, 영역 추정에 거의 비용이 들지 않습니다.

RPN(Region Proposal Network)는 경계 박스와 클래스 점수를 동시에 예측하는 CNN입니다.

DNN으로 영역 추정을 하도록 하는 알고리즘을 통해, 계산 비용이 거의 들지 않게 합니다.

RPN과 Fast R-CNN 객체 탐지 네트워크를 결합하기 위해, regress region bounds와 objectness끼리 합성곱 feature를 공유할 수 있고, 속도 또한 빨라집니다.

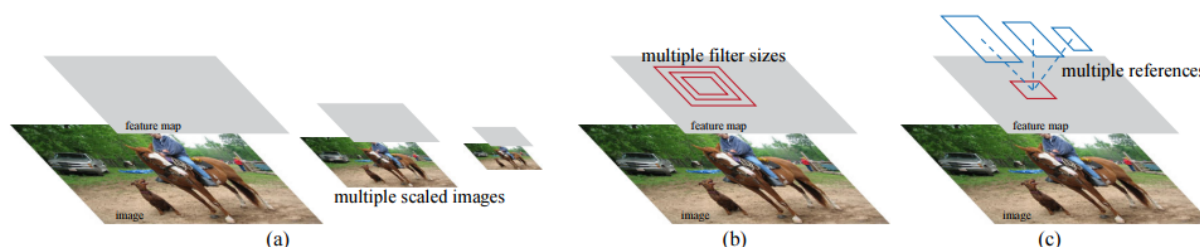


Figure 1: Different schemes for addressing multiple scales and sizes. (a) Pyramids of images and feature maps are built, and the classifier is run at all scales. (b) Pyramids of filters with multiple scales/sizes are run on the feature map. (c) We use pyramids of reference boxes in the regression functions.

Related Work

Object Proposals

- Region Proposal기법에는 여러 논문이 있습니다. 픽셀을 그룹핑하는 선택적 기법이나 슬라이딩 윈도우를 사용하는 EdgeBoxes등이 있습니다. 하지만, RPN은 객체 탐지 모델과 독립적으로 외부 모듈을 사용합니다.

Deep Networks for Object Detection

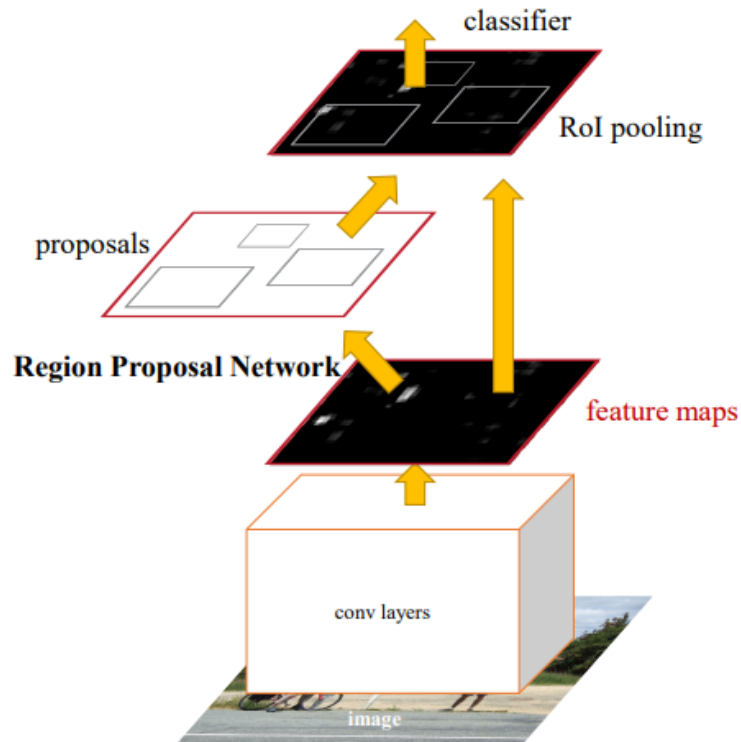
- R-CNN은 object categories 와 background를 분류하기 위해 proposal region 내에서 CNN으로 end-to-end로 학습합니다.

Faster R-CNN

Faster R-CNN은 두 가지 모듈로 이루어져 있습니다. 첫 번째 모듈은 영역 추정을 위한 깊은 합성곱 네트워크입니다. 두 번째 모듈은 객체 탐지 모듈입니다.

전체 네트워크는 객체 탐지를 위한 단일(Single), 통합(unified) 구조입니다.

Attention 메커니즘과 유사하게, RPN Module은 Fast R-CNN한테 주목할 부분을 알려줍니다.



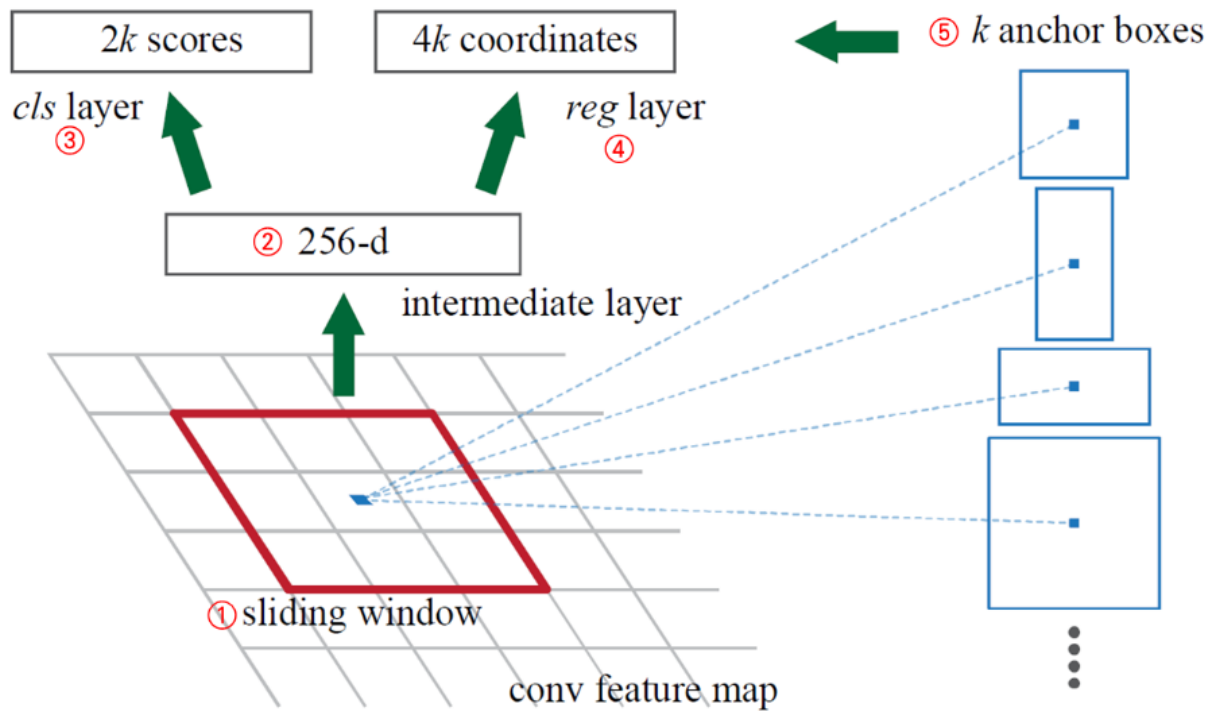
1. CNN을 활용하여, 전체 이미지에 합성곱 연산을 수행합니다.
2. 1의 과정을 통하여 얻은 피쳐 맵을 RPN과 Classifier에 전달합니다. **(Share Feature Map)**
3. RPN은 Feature Map을 기반으로 영역 추정을 수행합니다.
이어, 영역 추정 결과를 Pooling합니다.
4. 최종적으로, 영역 추정 경계 박스를 활용하여, 객체 탐지를 수행합니다.
Faster R-CNN은 영역 추정과 이미지 분류를 모두의 통합 네트워크에서 수행합니다.

Region Proposal Networks

RPN(Region Proposal Network)는 크기에 상관없이 이미지 전체를 입력받습니다. 마지막 CNN layer를 사용하여 작은 슬라이드 CNN feature map을 slide합니다.

각 경계 박스는 객체가 있는지 여부를 점수로 나타냅니다.

RPN 합성곱 네트워크로 처리합니다. RPN 과 R-CNN 객체 탐지는 Feature Map을 공유합니다.



1. 각 slide window는 작은 차원의 Feature로 Mapping됩니다.
2. 첫 번째는 분류 계층, 두 번째는 경계 박스 회귀 계층으로 반환됩니다.

Anchors

각 슬라이딩 윈도우의 중심 위치마다 여러 경계 박스 영역을 예측합니다. 각 슬라이딩 윈도우 위치마다 최대 예측할 수 있는 경계 영역 개수는 k 개입니다.

그렇기에, 회귀 계층(reg layer)은 좌표값을 $4k$ 개 찾습니다. 그리고 분류 계층(cls layer)은 $2k$ 개 점수값을 얻습니다. 경계 박스 하나는 객체일 확률과 아닐 확률로 두 가지 확률값이 존재하기 때문입니다.

RPN에서 사용하는 형태의 경계 박스를 앵커 박스(Anchor box)라고 합니다.

3가지 Scale과 3가지 비율을 활용하여 Anchor box를 생성합니다.



Translation-Invariant Anchors

Faster R-CNN의 중요한 특성은 위치 불변성(Translation-Invariant)입니다.

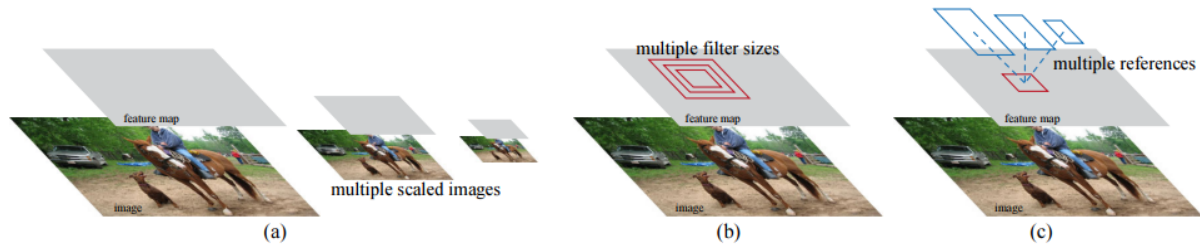
위치 불변성(Translation-Invariant)이란?

- 이미지 내의 객체 위치가 변하더라도, 같은 객체로 인식하는 특성입니다.

- Faster R-CNN의 RPN은 슬라이딩 윈도우 방식으로 이미지의 전체 영역을 훑기에 위치가 변하여도 같은 객체를 잘 인식합니다.

Multi-Scale Anchors as Regression References

멀티 스케일(Multi-Scale) 예측에는 세 가지 방식이 존재합니다.



(a) Image Pyramid Mechanism

- 입력 이미지를 다양한 크기로 조절하여, 각 스케일마다 피쳐 맵을 구합니다.
(효율적인 방법이지만 시간이 오래 걸립니다.)

(b) Filter Pyramid Mechanism

- 두 번째로는 필터(Sliding Window)를 다양한 크기(Multi Scale)로 사용하는 방법이 존재합니다.
feature map은 하나이지만, 다양한 필터를 사용하여, 풀링하는 방식입니다.

(c) Anchor Pyramid Mechanism

- 다양한 크기의 앵커 박스를 활용해 객체 분류, 경계 박스 회귀를 수행합니다. 앵커 피라미드 방식을 적용하여, 추가 연산 없이 feature를 공유할 수 있어 효율적입니다.

Loss Function

RPN을 훈련하기 위해, 각 앵커 박스마다 이진 분류를 수행합니다. 앵커 박스에 객체가 있는지 없는지 여부를 이진 분류하는 것이죠. 분류를 하려면 앵커 박스에 positive label이 있어야 합니다. 두 가지를 positive label로 감안했는데요, (i) 실제 경계 박스(ground-truth box)와 IoU가 가장 큰 앵커, 또는 (ii) 실제 경계 박스(ground-truth box)와 IoU가 0.7이 넘는 앵커입니다. 여기서 주목할 점이 있습니다. 실제 경계 박스 하나마다 여러 앵커 박스를 positive label로 할당할 수 있다는 점입니다. 다시 말해, 객체 위치를 정확히 나타내는 앵커 박스 단 하나만 positive label로 사용하는 게 아닙니다. 실제 경계 박스와 IoU가 높다면 여러 앵커 박스를 positive label로 간주하는 겁니다. 보통은 (ii) 조건으로 앵커 박스를 찾지만, (ii) 조건을 만족하는 앵커 박스가 없으면 (i) 조건으로 positive label 앵커 박스를 찾습니다. 또한, IoU가 0.3보다 작은 앵커 박스는 negative label로 간주합니다. positive도 negative도 아닌 앵커 박스는 훈련에서 제외하기 때문에 객체 탐지에 영향을 미치지 않습니다.

$$L(\{p_i\}, \{t_i\}) = \underbrace{\frac{1}{N_{cls}} \sum_i \frac{L_{cls}(p_i, p_i^*)}{\text{⑤}}}_{\text{⑦}} + \underbrace{\lambda \frac{1}{N_{reg}} \sum_i p_i^* \frac{L_{reg}(t_i, t_i^*)}{\text{⑥}}}_{\text{⑧}} \quad \text{③} \quad \text{④}$$

Training RPNs

RPN은 확률적 경사 하강법(SGD)와 역전파 end-to-end 훈련을 할 수 있습니다. 여러 앵커 박스를 만들어 영역 추정을 한다고 했는데, 그럴 때, 모든 앵커 박스마다 손실 함수를 적용하여 negative label에 치우친 결과를 나타냅니다.

Sharing Features for RPN and Fast R-CNN

RPN과 Fast R-CNN은 detector로 Fast R-CNN을 사용합니다. RPN과 Fast R-CNN이 feature map을 공유하기에 매우 효율적입니다. RPN과 Fast R-CNN은 합성곱 feature를 서로 다른 방법으로 훈련합니다. RPN과 Fast R-CNN이 Feature map을 공유하면서도, 여러 방법으로 분리해서 사용합니다.

Alternating training

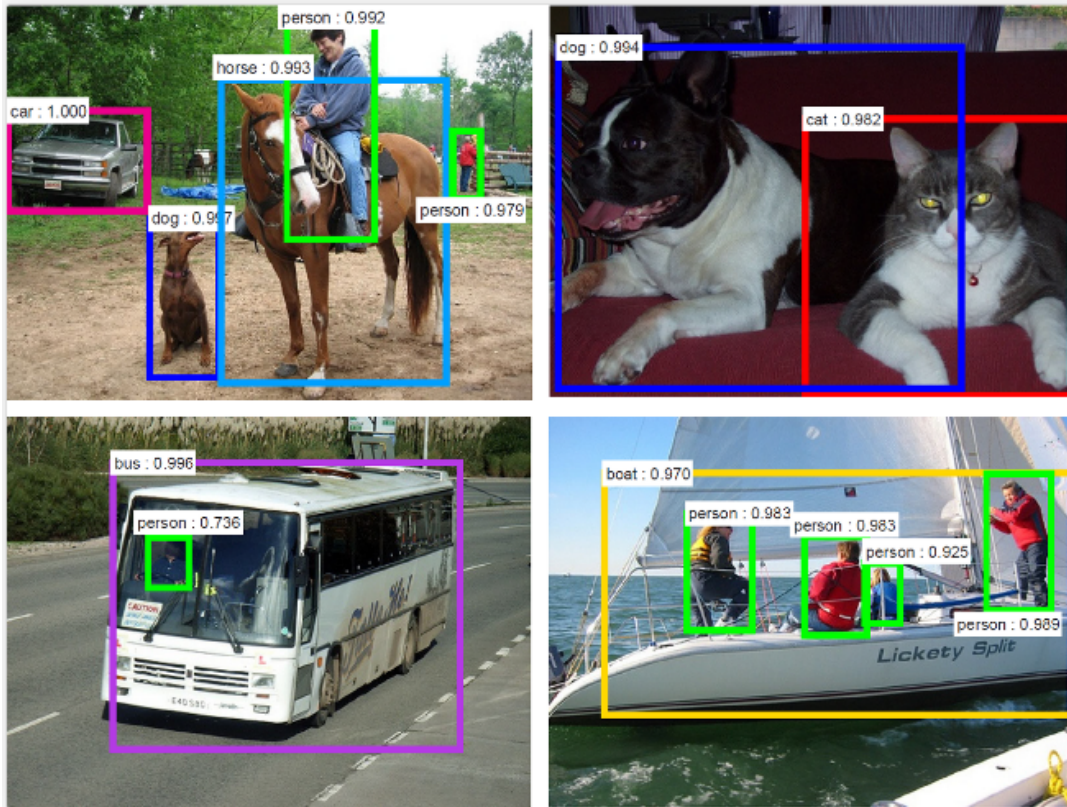
- 해당 논문에서 사용하는 방식입니다. 먼저 **RPN을 훈련하고, 이어서 영역 추정 경계 박스**를 사용하여, Fast R-CNN을 훈련합니다. **Fast R-CNN으로 튜닝된 네트워크는 다시 RPN을 초기화**하는데 사용합니다. 그러면 **RPN과 Faster R-CNN이 feature를 공유하면서 훈련이 가능합니다.**

4-Step Alternating Training

- feature를 공유하면서 훈련하도록 실용적인 4단계 알고리즘이 있습니다.
 - RPN을 활용하여 훈련합니다.
이 네트워크는 이미지넷으로 사전 훈련된 모델로 초기화하고 영역 추정 작업을 위해 end-to-end fine tuning을 합니다.
 - 첫 번째 단계에서 생성한 영역 추정 경계 박스를 활용하여, 독립적인 Fast R-CNN을 training 합니다.
 - RPN 훈련 초기화를 위해 Fast R-CNN을 사용합니다. 다만, 공유된 합성곱 계층을 고정하고, RPN 계층만 fine tuning합니다.
 - 공유된 합성곱 계층을 고정한 채로, Fast R-CNN 계층만을 Fine tuning합니다.
이러한 방식으로 RPN과 Fast R-CNN은 같은 합성곱 계층을 공유하면서 단일 네트워크 구성이 가능합니다.

Implementations Details

Faster R-CNN에서는 단일 스케일(single scale) 이미지에 대해 영역 추정과 객체 탐지 훈련 및 테스트를 수행합니다. 3가지 scale과 3가지 feature를 앵커 박스로 활용합니다.



anchor	$128^2, 2:1$	$128^2, 1:1$	$128^2, 1:2$	$256^2, 2:1$	$256^2, 1:1$	$256^2, 1:2$	$512^2, 2:1$	$512^2, 1:1$	$512^2, 1:2$
proposal	188×111	113×114	70×92	416×229	261×284	174×332	768×437	499×501	355×715

앵커 박스 종류별 영역 추정 경계 박스 평균 크기 (ZF-net 사용 시)

Experiments

train-time region proposals			test-time region proposals		mAP (%)
method	# boxes		method	# proposals	
① SS	2000		SS	2000	58.7
② EB	2000		EB	2000	58.6
③ RPN+ZF, shared	2000		RPN+ZF, shared	300	59.9

① 선택적 탐색(selective search) 기법과 ② EdgeBoxes 기법은 훈련/테스트 단계에서 모두 영역 추정 경계 박스를 2,000개 만듭니다. 각각 mAP 58.7%, 58.6%입니다. 반면 (RPN, 피쳐 공유, 베이스라인 ZF-net을 적용한) Faster R-CNN은 mAP 59.9%를 달성했습니다. Faster R-CNN은 테스트 단계에서는 영역 추정 경계 박스 개수가 최대 300개입니다. NMS를 적용해서 더 적은 경계 박스를 남길 수도 있습니다. Faster R-CNN은 피쳐를 공유하고, 테스트 단계에서 경계 박스 개수도 적기 때문에 선택적 탐색(SS)이나 EdgeBoxes(EB)에 비해 더 빠릅니다.

Conclusion

본 논문에서는 빠르고, 정확한 영역 추정을 하기 위해, RPN을 제안합니다. RPN과 객체 탐지기가 합성곱 피쳐를 공유함으로써, 영역 추정 비용을 크게 줄입니다.

Faster R-CNN은 통합된 딥러닝 기반 객체 탐지 시스템으로, 실시간 객체 탐지가 가능하며, 전체적인 정확도도 기존 모델들보다 훨씬 뛰어납니다.

Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks

State-of-the-art object detection networks depend on region proposal algorithms to hypothesize object locations. Advances like SPPnet and Fast R-CNN have reduced the running time of these detection networks, exposing region proposal computation as a bottleneck. In this work,

😊 <https://arxiv.org/abs/1506.01497>



논문 리뷰 - Faster R-CNN 훑아보기

Faster R-CNN은 기존 Fast R-CNN에 영역 추정 네트워크(RPN)를 더해 속도와 성능을 끌어올린 모델입니다. Faster R-CNN에 와서야 비로소 모든 객체 탐지 구조를 딥러닝으로 훈련할 수 있었습니다. 본 글에서 주요 내용 위주로 Faster R-CNN 논문을 번역/정리했습니다

📖 <https://bkshin.tistory.com/entry/%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0-Faster-R-CNN-%ED%86%BA%EC%95%84%EB%B3%B4%EA%B8%B0>

0

