



SNIPER

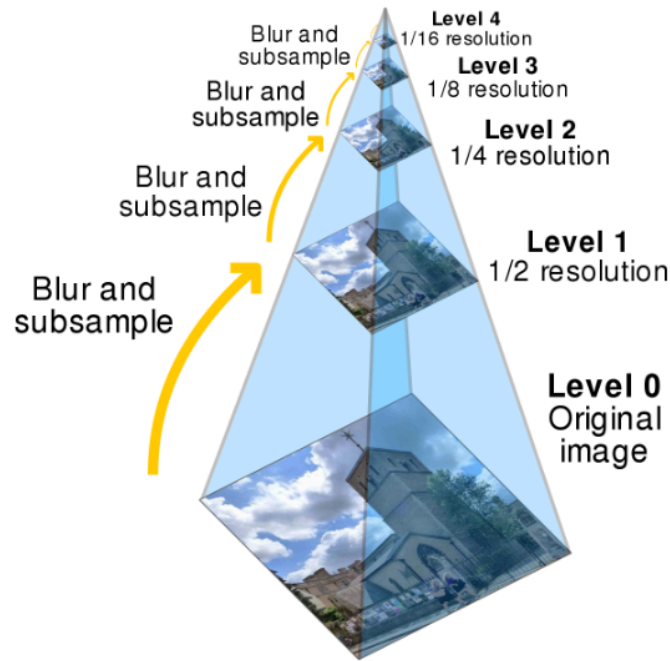
<https://github.com/mahyarnajibi/SNIPER.git>

: Efficient Multi-Scale Training

SNIPER (Scale Normalization for Image Pyramids with Efficient Resampling)

객체 탐지에서 **다중 스케일 학습의 효율성**을 크게 향상 시킨 방식

※ **Image pyramid** : Input Image의 Size를 단계별로 변화시켜 쌓은 형태



object detection → scale-invariance 가 중요함

Feature Pyramid Network - FPN

[논문 요약]

1. 요약

- 기존 다중 스케일 학습, 전체 image pyramid를 처리 → 계산량 증가 → 문제
- **"Chips" 사용** : 이미지에서 객체 주변의 작은 영역을 샘플링하여 필요없는 영역을 무시하고, 적절한 크기만으로 처리하여 효율성 높임.
⇒ 기존보다 3배 빠르게 학습하면서 높은 성능 유지

2. SNIPER 알고리즘

2.1) Chip 생성

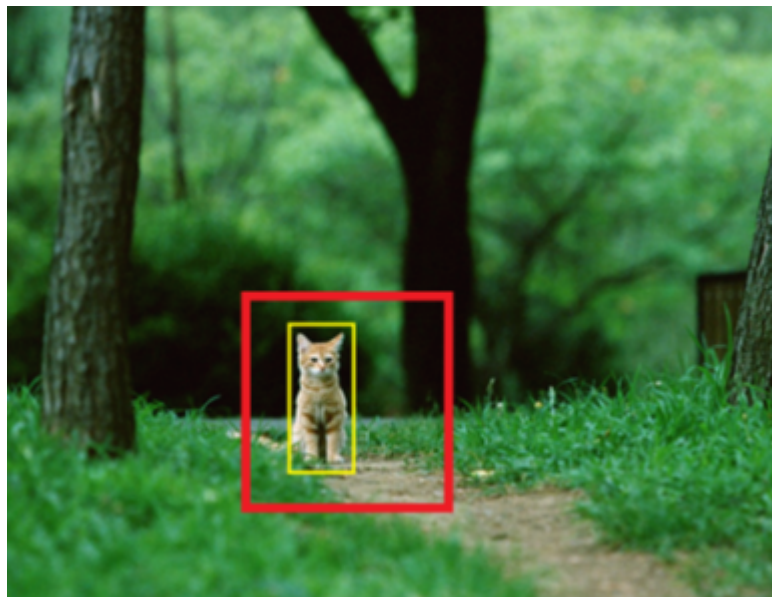
- 다양한 스케일에서 **512 x 512 픽셀 크기의 chips**를 생성
- 원본 이미지를 **특정 해상도로 조정 후**, 격자 형태로 작은 chips을 나눠 배치
⇒ 전체 이미지가 아닌, **객체가 존재하는 영역만 집중적으로** 학습할 수 있음.



예시)

배경에 비해 상대적으로 고양이가 작음.

이 사진을 이용하여 작은 고양이부터 큰 고양이까지 검출가능한 모델을 학습시키기 위해, Image pyramid를 사용하는 것이 무리가 있음.



위 처럼 빨간 박스 = chip

Ground Truth가 포함 + 객체가 있는 region \Rightarrow chip 명명

(chip = 512×512 사이즈로 고정)

\rightarrow 픽셀 관점, 연산량을 대폭 줄어둘게 한다고 함

2.2) Positive chip selection

- 특정 스케일에서 적절한 크기의 객체만 포함된 chips 선택
- 작은 객체 → 고해상도 / 큰 객체 → 저해상도 학습 (불필요한 연산 줄임)
- 각 chips 내부에 있는 객체가 해당 칩의 유용한 학습 대상

2.3) Negative chip selection

- 객체가 포함되지 않은 배경 부분을 처리할 필요 없음

⇒ Negative Mining이란, 객체가 존재하지 않는 배경 영역(negative samples)을 효과적으로 걸러내어 불필요한 연산을 줄이는 방법

- 배경이 검출 오류 (False positive)를 유발할 수 있기에

일부 부정 샘플(배경 영역)도 학습

⇒ 간단한 Region Proposal Network (RPN)을 먼저 훈련하여 배경 Chip을 선택하고 학습에 포함

2.4) Label Assignment (레이블 할당)

- end-to-end 학습
- RPN이 예측한 영역에 대해 라벨을 할당하고 객체 유무를 구별해 학습 진행
- bbox target들은 chip 내부의 모든 gt box에 기반
- 바운딩 박스 회귀도 수행
- gt bbox와 0.5 이상의 iou를 가지는 모든 proposal들을 positive로 표시하고 bbox target을 배정함.

2.5) 장점

- 단일 GPU에서도 큰 배치 크기를 활용할 수 있어, 배치 정규화도 효과적으로 사용할 수 있음.
- 기존 방식과 달리, 작은 해상도에서 학습하므로 연산량 감소
- 3배 더 빠른 학습 속도 / COCO 데이터 셋에서 기존 기법과 동등한 성능 유지

[실험]

데이터: COCO dataset

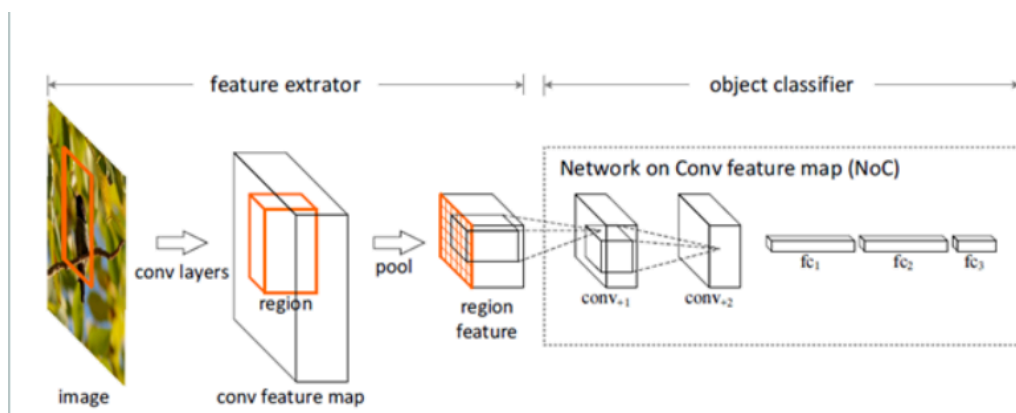
→ 123,000 img(훈련 및 validation)/ 20,288 img(test)

- 118,000 장을 훈련 하고 5,000장을 recall 보고

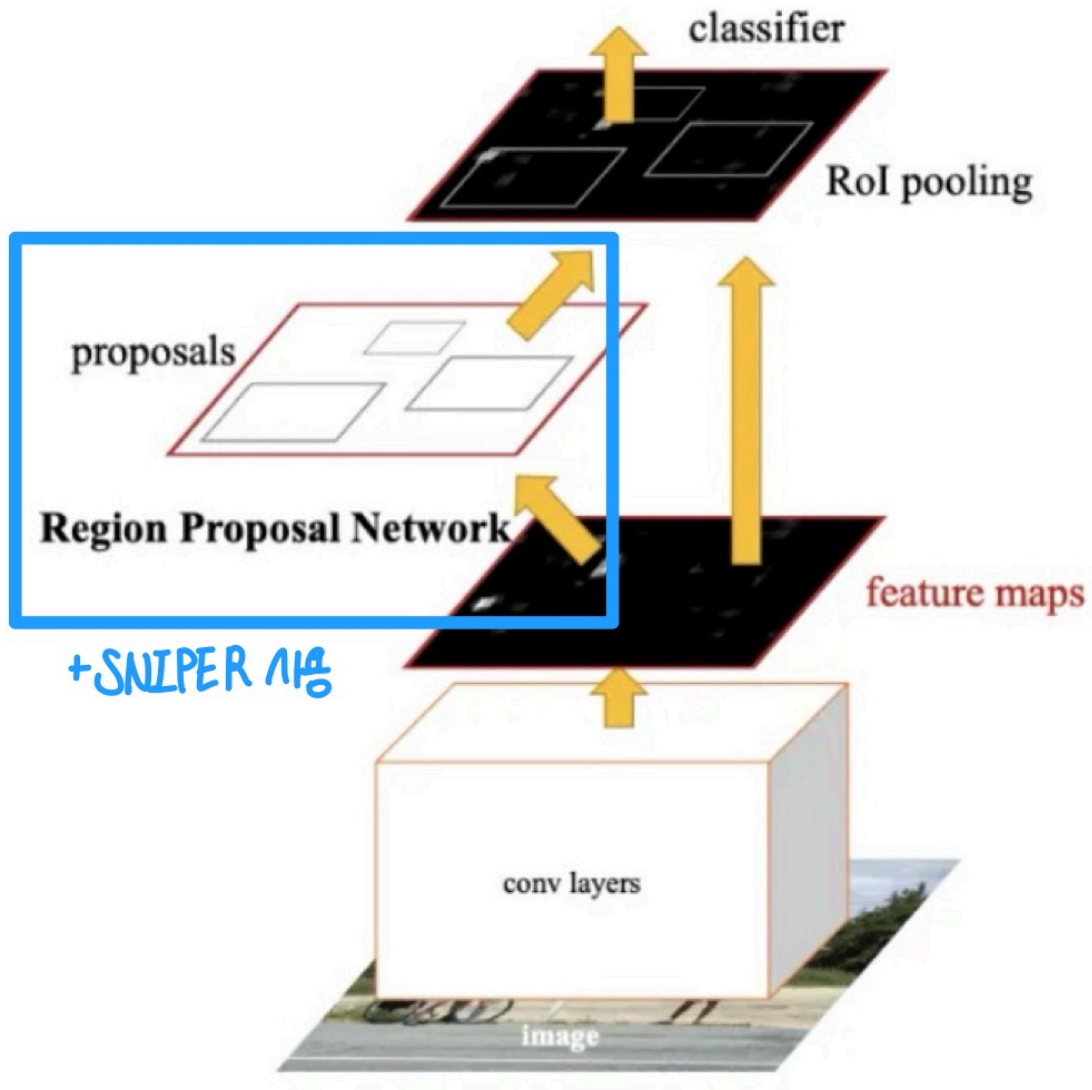
아키텍처: Faster R-CNN (ResNet-101) 모델

Hyper-parameter)

- batch_size : 128
- lr_rate : 0.015
- epoch : 6 (1 epoch 당, 11,000번 반복)



출처:[[Part V. Best CNN Archite.. : 네이버블로그](#)]



[결과]

Method	Backbone	AP	AP ⁵⁰	AP ⁷⁵	AP ^S	AP ^M	AP ^L
SNIPER	ResNet-101	46.1	67.0	51.6	29.6	48.9	58.1
SNIPER 2 scale	ResNet-101	43.3	63.7	48.6	27.1	44.7	56.1
SNIPER w/o negatives	ResNet-101	43.4	62.8	48.8	27.4	45.2	56.2

Table 2: The effect training on 2 scales (1.667 and max size of 512). We also show the impact in performance when no negative mining is performed.

SNIPER_기본 버전

- 여러 스케일 훈련 → 다양한 해상도 활용해 성능 최적화
- Negative Mining 기법을 적용 → 불필요한 부분 줄이고 효율성 높임

⇒ 결과 : AP 46.1

SNIPER_2 scale

- 1.667배와 512 크기에서만 훈련
- 여러 스케일을 사용하지 않으므로 일반적인 SNIPER보다 성능이 다소 낮음 ⇒ 결과 **AP 43.3**

SNIPER_w/o negatives

- Negative mining을 적용하지 않은 경우
- 불필요한 배경 영역도 학습하게 되어 효율성이 낮아짐

⇒ AP 감소

※ AP(Average Precision)

: 객체 탐지 모델의 성능을 평가하는 대표적인 지표

		실제 정답	
		True	False
분류 결과	True	True Positive	False Positive
	False	False Negative	True Negative

분류성능평가지표 - Precision(정밀도), Recall(재현율) and Accuracy(정확도).