R-CNN(Rich hierarchies for accurate object detection and semantic segmentation) 논문 정리

object detection

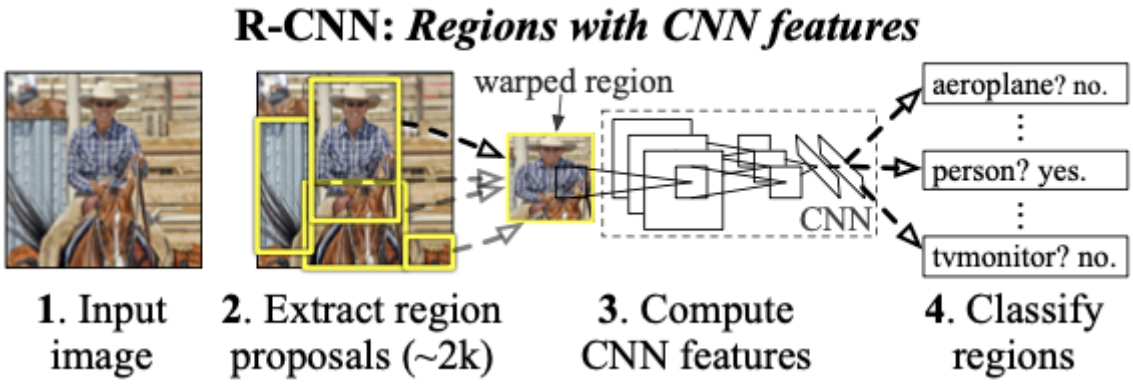
말 그대로 객체를 추적하여 위치를 알아내고 분류를 하는 것으로 classification과 localization이 결합된 형태이다.

R-CNN

R-CNN은 region proposal과 CNN을 결합해서 사용했기 때문에 R-CNN으로 명명되었으며

R-CNN은 object detection(객체감지) 분야에서 Relu함수, overfitting을 방지하기 위한 방안으로 dropout, pooling 등을 최초로 적용시킨 모델으로 R-CNN이 나오기 이전의 object detection 모델들 보다 성능을 30%정도 향상시키고 mAP(mean average precision)를 53.3% 를 달성한 R-CNN 계열의 시초가 되는 모델이다. 그리고 여기서 CNN 부분은 2012년 개최된 ILSVRC 대회에서 우승을 차지한 CNN 구조인 alexnet 을 사용한다.

region with CNN feature



위 그림은 R-CNN의 순서를 나타낸 것입니다.

첫번째로 input image를 준비합니다.

그 다음 2000개의 region proposal을 추출합니다.

여기서 image를 CNN에 넣기 전에 fine tuning을 통해 CNN을 학습 시켜주는데, fine tuning을 할때는 ground-truth box와 IoU가 0.5 이상이면 positive sample, 0.5이하이면 negative sample로 classify 했으나 이후 training svm에서는 IoU가 0.3이상이면 positive sample, 0.3이하이면 negative sample로 classfiy 했습니다.

이렇게 fine tuning을 할 때와 training svm을 할 때 positive sample과 negative sample을 다르게 설정하였을 때 성능이 더 우수했기 때문에 이런 방식을 채택하였습니다.

설정한 positive sample과 negative sample로 미니 배치를 여러개 만든 후

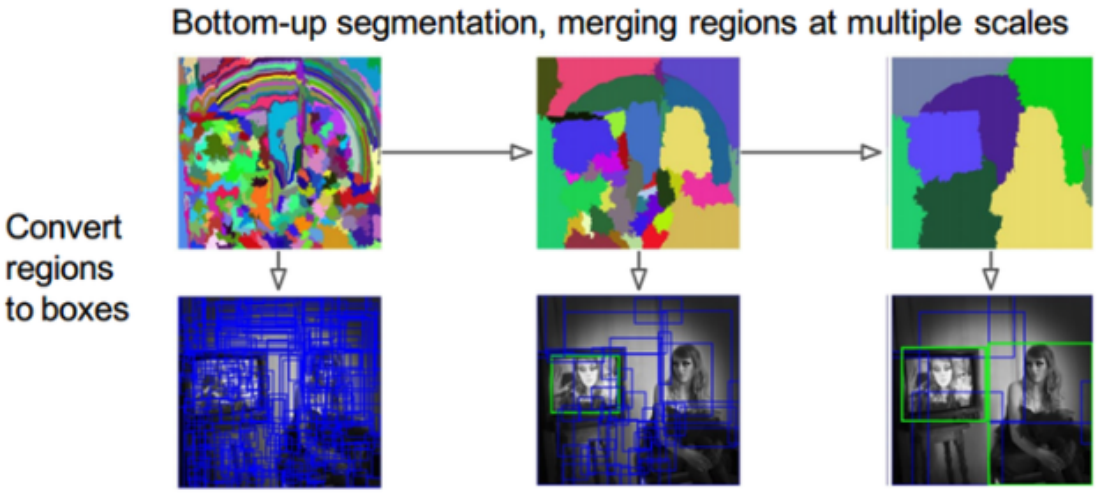
alexnet의 227x227로 fix 되어있는 input image를 맞춰주기 위해 warp 시켜서 CNN에 넣어줍니다.

CNN에서 나온 feature vector를 svm을 사용하여 스코어링(region 분류) 해줍니다.

이후에 greedy NMS(non-maximum suppression)을 적용합니다.

1. extract region proposal

selective search 알고리즘을 사용해 2000개의 bottom-up region proposal을 추출합니다.



selective search

1. 각 픽셀간의 유사도를 통해 첫번째 그림과 같은 small segmented areas 들을 만듭니다
2. bottom-up 방식으로 이전에 만든 small segmented areas들을 합쳐서 더 큰 segmented areas들을 만듭니다.

3. 두 번째 과정을 반복해서 2000개의 region proposal을 추출합니다.

2. CNN

Alexnet에 넣기 전에 추출한 2000개의 region proposal들을 warp 시켜서 resizing 시켜준 다음 Alexnet에 넣습니다.

다음으로 fine tuning을 진행하기전에 ground-truth box와 region proposal의 IoU를 비교하여 0.5보다 큰 box를 positive sample, 0.5보다 작은 box를 negative sample로 나눕니다.

이렇게 해서 32개의 positive sample + 96개의 negative sample = 128개의 이미지로 이루어진 미니배치를 만듭니다.

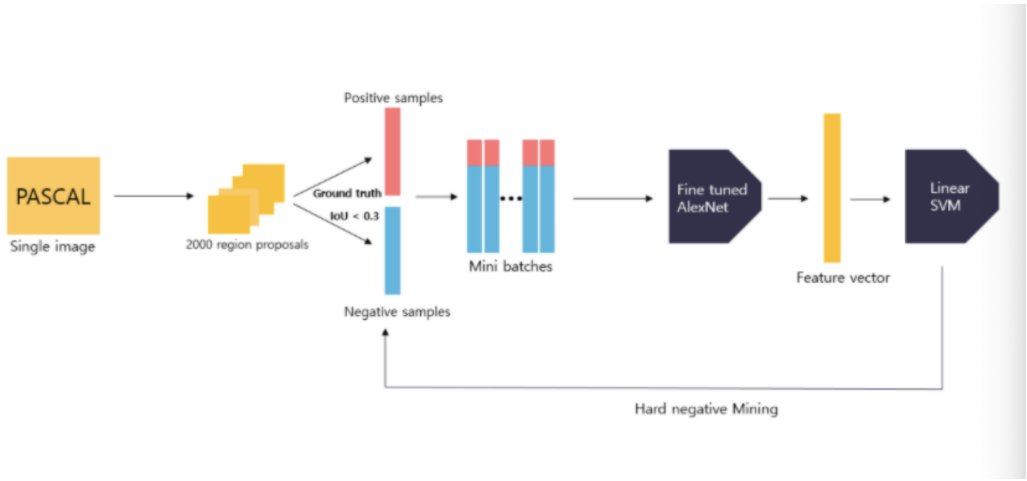
이렇게 해서 만든 배치들을 통해 fine tuning을 진행합니다.

fine tuning을 진행하면 많은 학습데이터를 얻을 수 있기 때문에 overfitting을 방지할 수 있습니다.

그리고 fine tuning을 진행하기 위해서 기존의 softmax layer를 N+1 classification을 수행하게 하기 위해 수정합니다.

마지막으로 4096개의 feature vector를 추출합니다.

3. SVM classifier



Svm classifier에서는 ground-truth box를 positive sample, region proposal과 ground-truth box의 IoU를 비교했을 때 0.3 보다 작은 box를 negative sample로 나눕니다

이때 IoU값이 0.3보다 큰 것은 무시합니다.

fine tuning 때와는 다르게 이렇게 분류하는 이유는 fine tuning을 할 때와 svm을 훈련시킬 때 positive sample과 negative sample 을 분류하는 기준을 다르게 설정했을 때 더 성능이 좋았기 때문에 다르게 나눕니다.

svm은 이진 분류이기 때문에 classify하려는 객체 종류 만큼의 svm이 필요합니다.

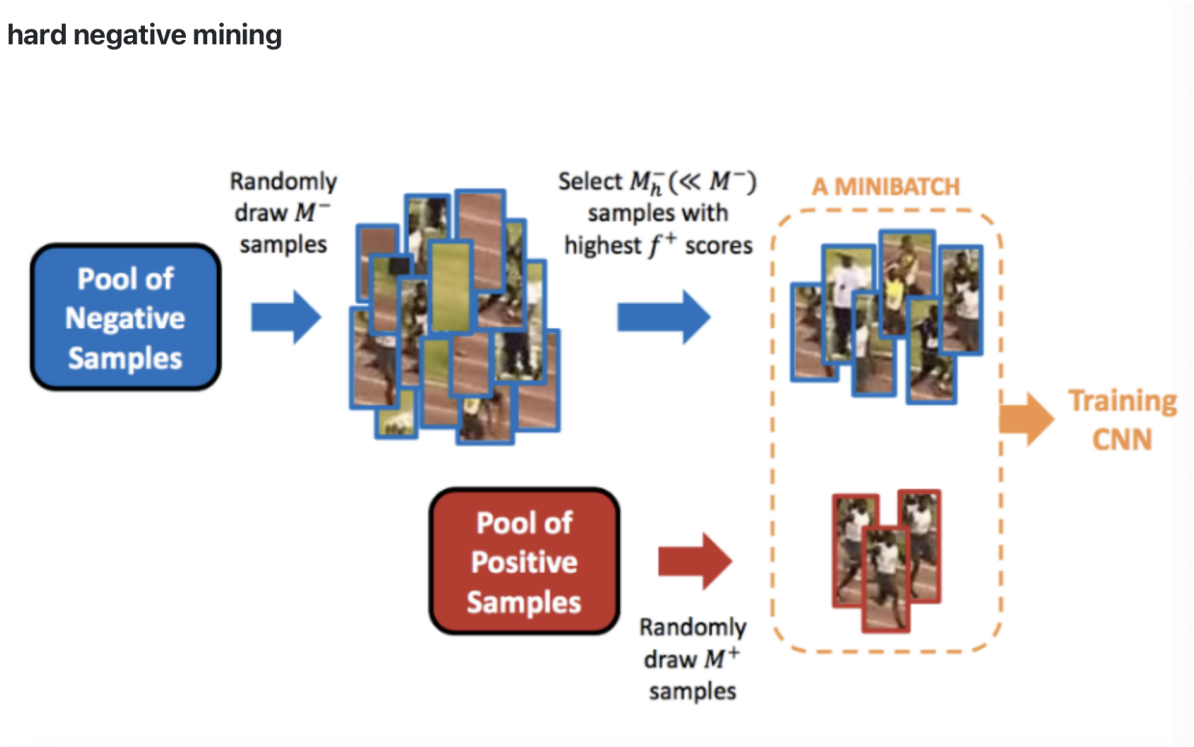
그리고 한차례 학습이 끝난 이후에 hard negative mining 기법을 적용하여 재학습을 수행합니다.

svm에서 output으로 confidence score를 반환합니다.

그리고 bounding box regression을 통해 selective search를 통해 얻은 부정확한 객체의 위치 정보를 수정합니다.

svm에서 반환한 confidence score를 이용해서 Greedy nms(greedy non-maximum suppression)을 수행합니다.

4. Hard negative mining



이미지에서 object는 positive sample로, 나머지 배경은 negative sample이 됩니다.

이때, 모델이 bounding box를 배경이라고 예측하고 실제로 배경인 경우 true negative sample라고 합니다.

반면에 모델이 사람이라고 예측했지만. 실제로 배경인 경우 false positive sample에 해당합니다.

그리고 찾으려고 하는 object를 찾으려고 했는데 못찾는 것을 false negative라고 합니다.

여기서 positive sample보다 negative sample이 더 많은 클래스 불균형 때문에 false positive sample error를 주로 범하게 됩니다.

hard negative mining을 통해 epoch마다 학습데이터에 false positive sample을 넣어서 추가로 학습을 진행하여 false positive error를 줄입니다.