



deeplearning.ai

Object Detection

Object detection

object detection을 수행하기 위해 ConvNet을 사용하는 방법을 알아보자. sliding window detection algorithm을 이용한다.

Car detection example

Training set: 우선 closely cropped examples of cars로 labeled training set을 만들어야 한다.



x	y
	1
	1
	1
	0
	0

이와 같은 labeled training set이 주어지면 그 다음은 이런 cropped image가 차인지 아닌지를 판별하는 ConvNet을 학습한다.



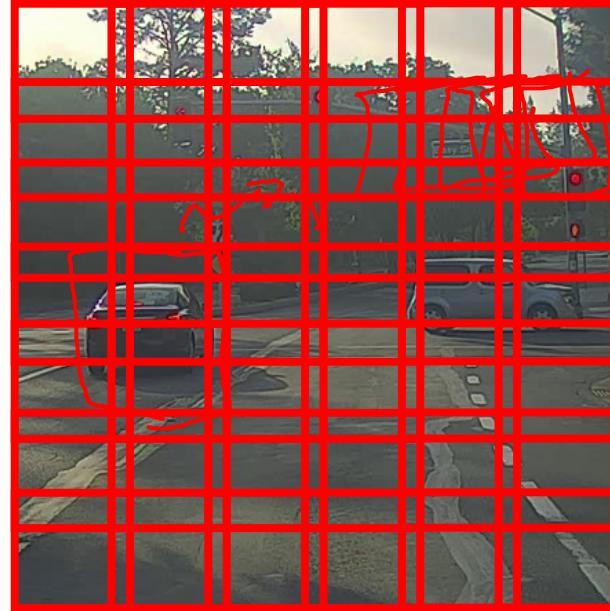
→ ConvNet → y

이 ConvNet을 학습하고 나면 이것을 sliding windows detection에 사용할 수 있다.

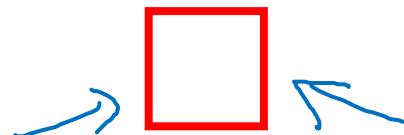
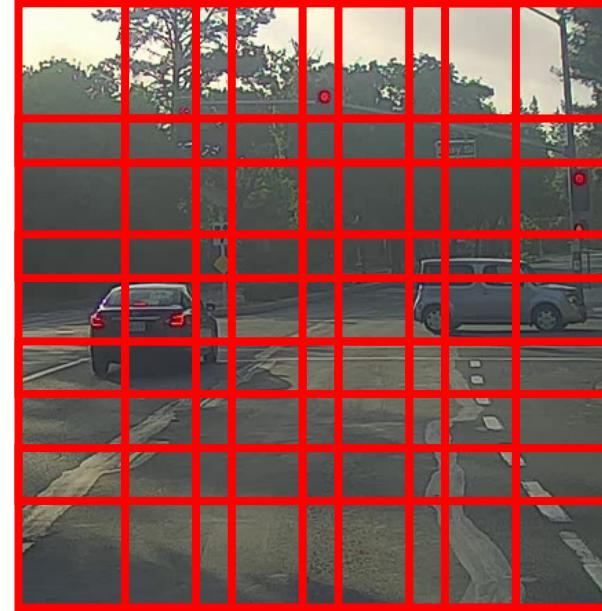
차가 있으면 1
차가 없으면 0

Sliding windows detection

→ ConvNet → 0



→ ConvNet



- 요런 테스트 이미지가 있을 때,
 1. window size를 선택하고
 2. 이 window를 이동시키면서
 3. 해당 영역을 ConvNet에 입력하고
 4. 각 포지션에 대해 0/1을 classify한다

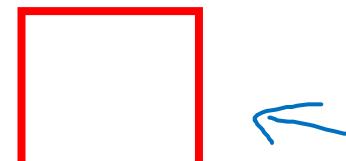
computation cost
computation cost

Sliding windows detection의 단점:
계산 비용이 너무 크다.

앞의 과정이 끝나면 좀 더 큰
window로 동일한 과정을 반복한다.

딥러닝 이전에는 꽤 단순한 classifier를 사용했기 때문에 슬라이딩 윈도우 방식이 가능했지만 ConvNet은 계산이 비싸기 때문에 불가능. coarser stride를 쓰면 계산량은 줄겠지만 성능이 떨어짐(이미지 안에서 object를 상당히 정확하게 localize할 수 없다는 얘기)

다행히 좋은 대안이 있다. sliding windows object detector can be implemented convolutionally or much more efficiently



똑같이 반복

Andrew Ng