

비전 목표 비교

| | | |
|-----------------|-----------------------|--------------------------------------|
| single object | Classification | ⇒ 단지 무엇인지 / 고양이 / 개 |
| | + localization | ⇒ 위치 정보 + 네모 상자 치기 |
| multiple object | object detection | ⇒ cat / dog / duck / bird 찾기 + 네모 박스 |
| | instance segmentation | ⇒ pixel 단위 구분 분류 |

Object detection 방식

- 2 stage Detector ⇒ localization → classification 순차적 진행
 (Region proposal) (feature extract + Classifi, regression)
 ↓
 ex) RNN
- 1 stage Detector ⇒ (local + classi) 한번에 진행
 ↓
 xolo (속도가 빠르나 정확도 ↓ trade off)

비전 RNN → Fast RNN → Faster RNN
 feature 뽑기 1번만 CPU → GPU 여러번
 region proposal 제안 (RPN)

어디곳에 물체 있는지 찾는 것 빠르게 가능

Region proposal method

Sliding Window

faster RNN (GPU로 4배)

원본에 \square 를 이동하며 움직여 있을 범위의 이미지 찾아내기 (CPU 메모리 잡아먹음)

Selective Search (RNN, Fast RNN)

인접한 영역(region) 끼리 유사성 측정해 큰영역으로 확대

Precision, recall

실제

| | |
|----|----|
| TP | FP |
| FN | TN |

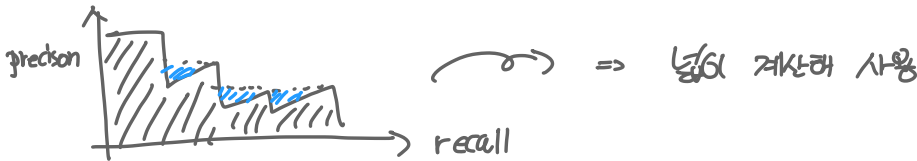
($\begin{matrix} \text{있다고 생각} & \text{없다고 생각} \\ \text{있다고 잘못} & \text{없다고 잘못} \end{matrix}$)

precision (정확도) : $\frac{TP}{TP+FP}$ (모델이 탐지한 물체 개수)

recall (재현율) : $\frac{TP}{TP+FN}$ (실제 정답 개수)

→ 모델이 많은 영역에 대해 물체 존재한다 → 재현율 ↑, 정확도 ↓
 , 매우 확실한데만 " → 재현율 ↓, 정확도 ↑

이 둘은 반비례 하기 때문에 둘다 고려하고 싶을 ⇒ average precision



* TF 같이 아키텍처? IO 50% 아키텍처 기준으로

■ NMS (Non Maximum Suppression)

↳ Obj Det 에서 중복된 박스 줄이는 법 (겉근거 제거, 나머지 합침)

■ R-CNN

1. selective search 은 Region 뽑음 (2000개 정도)
2. 얻은 region을 동일한 크기의 이미지로 잘라냄 (각 RoI에 대해 wrap 수행)
3. 잘라온 CNN 의 forward로 feature 뽑음
4. SVM 써서 3의 결과의 class 뽑음
(이때 각 class에 대해 독립적으로 훈련된 binary SVM 사용) (bad...)
5. 해당 feature 는 regressor 에 넣어 bounding box 예측

Dis \Rightarrow 느림 (CPU), end to end 안됨 (CNN과 SVM, BBR), training testing
구분 필요

■ Fast R-CNN

\Rightarrow 이미지를 한번 CNN에 넣음, 다만 나온 feature map으로
Projection 시켜서 사용 존재만 필요한 위치를 feature map에서 잘라냄

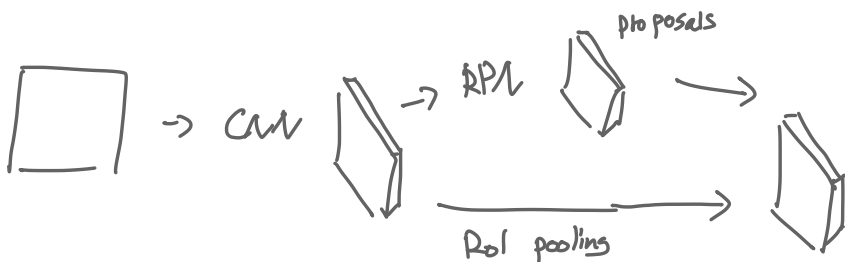
\Rightarrow SVM \rightarrow Soft max 사용

* RoI Pooling Layer 사용 \rightarrow 크기 달라도 고정된 크기 2×2 생성
(max pooling 사용)

Dis 여전히 CPU 써 느리다

Faster R-CNN

- ⇒ Region Proposal Network (RPN) GPU에서 실행됨
- ⇒ end to end로 훈련 + backward 해 가중치 업데이트 가능



- RPN

- ⇒ feature map 보다 작은 feature map을 여러 개 생성
- (K개의 anchor box 사용
sliding window 가리 regression과 classification 수행됨)

* Anchor 방식 사용하냐? 9개나? (w h x b)

⇒ 정복도 대개 9개 써야한다. (너무 많으면 느리고, 누가 담당 할지도 모름)

질문! R-CNN, Fast, Faster R-CNN에 비교해서 어떤 장점이 있는 거지?

특속 내려고 할때마다 걸리면 많아요?

아카이브에 올리면 누가 비호하게/해서 안 가져가나요?

- RoI pooling \rightarrow 2-stage (YOLO 방식) selection 해야 한다
2-stage이기 때문에, selection 하기 다음에
classification 하기 때문이다.