

※ Object Detection

: Image에서 Concept이 있으면 어디에 얼마나 있는지?

- Vision은 Classification 보다 더 다양하다

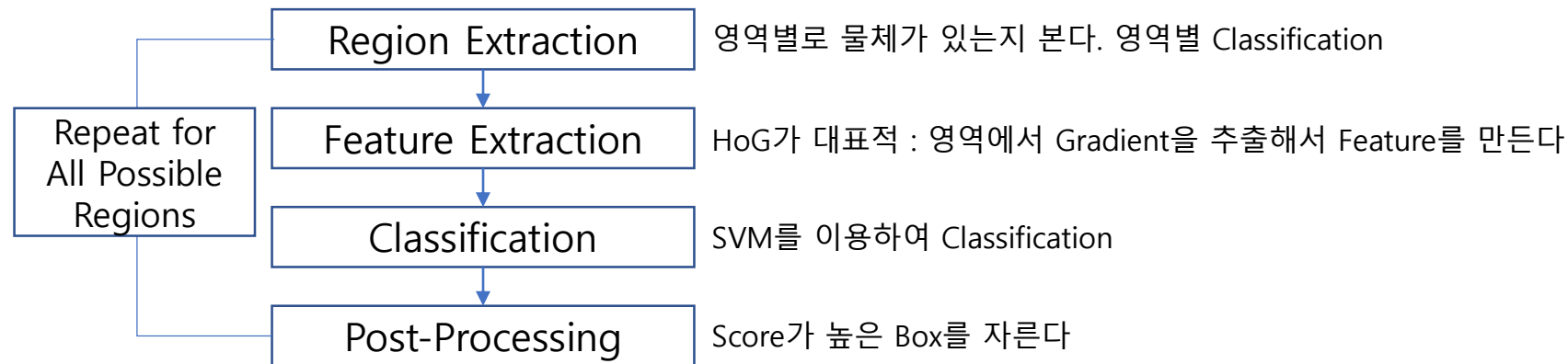
→ Classification, Object Detection, Semantic Segmentation, Pose Estimation, Activity Recognition

※ Object Detection : Classification(Label) + Localization(Object Bounding)

: 어느 영역에 있는가? 어떻게 물체를 핸들링 하는가? 물체의 다양한 크기, 형태를 찾아 구해야 함

※ Object Detection Pipeline (Traditional Object Detection Pipeline)

: Detection 순서도



모든 영역에서 반복 (Scale 와 Aspect Ratio를 변동 하면서)

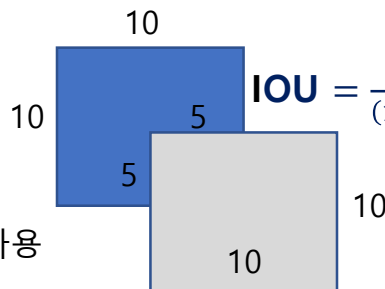
- NMS(Non Maximum Suppression) : 가까이 있는 박스를 높은 Score로 합침
- Sliding Window : 박스를 움직여 가면서 물체를 인식(전형적인 방법)
- Object Proposal : Image 안에 물체가 있는 것 같은 Bounding Box를 만들어 주는 것 (Sliding Window 대비 효율적)

※ Evaluation Metrics

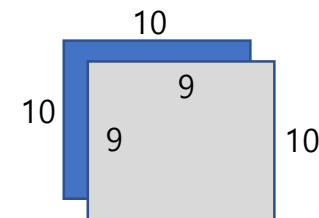
- IOU (Intersection Over Union)

$$IOU = \frac{\text{Area of Overlap}}{\text{Area of Union}}$$

Threshold로 정의하고 보편적으로 0.5 사용



$$IOU = \frac{5 \times 5}{(10 \times 10) + (10 \times 10) - (5 \times 5)} = 0.143$$

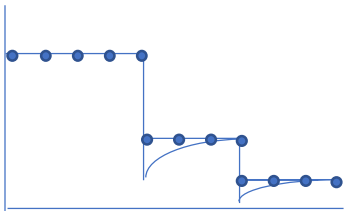


$$IOU = \frac{9 \times 9}{(10 \times 10) + (10 \times 10) - (9 \times 9)} = 0.68$$

Accuracy라고 함

※ Average Precision

: Area Under Curve (AUC)



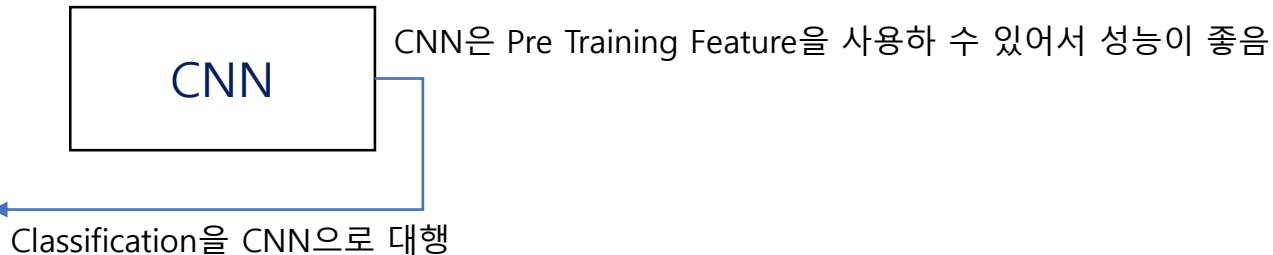
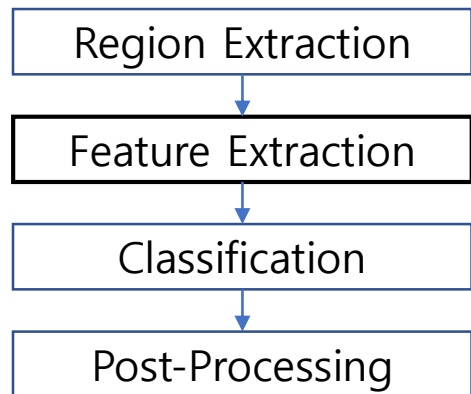
※ Precision 예측 정확도

- True Positive (TP)
- True Negative (TN)
- False Positive (FP)
- False Negative (FN)

$$\text{Precision} = \frac{\text{Correct Prediction}}{\text{All Positive Prediction}} = \frac{TP}{TP+FP} \quad \text{모델이 맞다고 한 것 중에 맞은 것}$$

$$\text{Recall} = \frac{\text{Correct Prediction}}{\text{All gt Object}} = \frac{TP}{TP+FN} \quad \text{있다고 한 것 중 맞은 것}$$

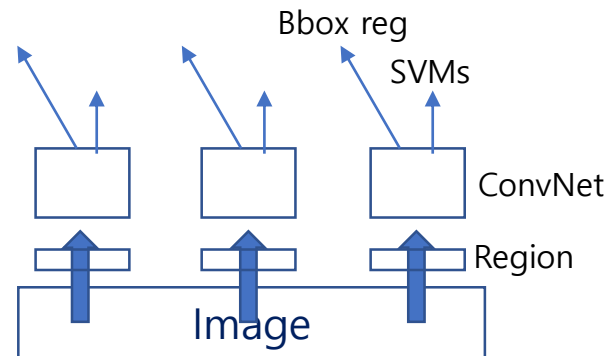
※ CNN for Classification



※ R-CNN (Region based CNN)

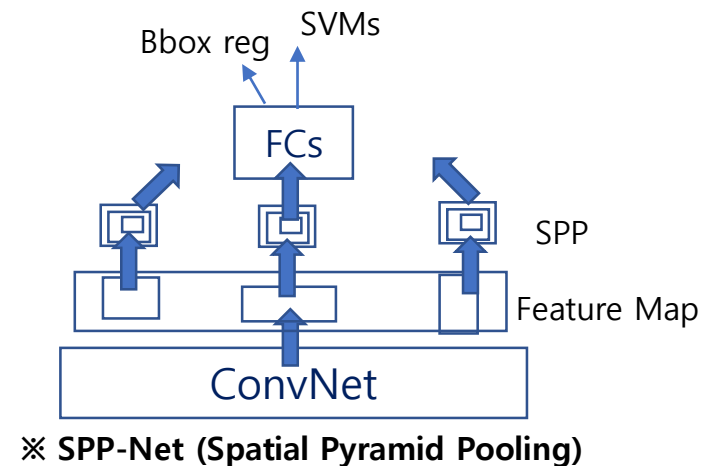
: Feature Extraction 하는 것을 convolution Network으로 바꾼 것이 특징

- Classification은 SVM 적용
- Bbox reg : Bounding Box Regression
- 시간이 오래 걸려서 실시간 사용 불가
- End to End 안됨
- 단점 : 모든 영역에서 CNN Feature를 뽑는다



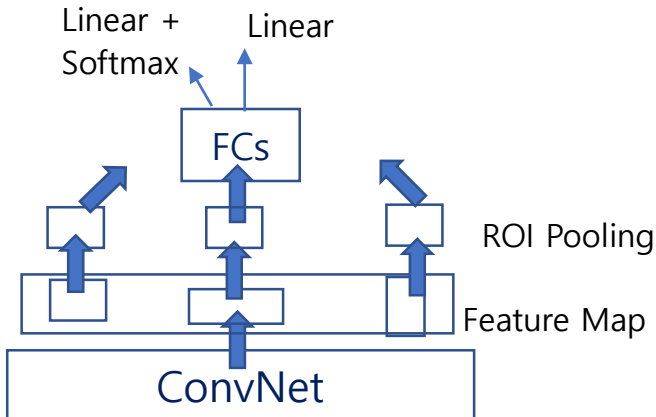
※ SPP-Net (Spatial Pyramid Pooling)

- Main Idea : 속도를 빠르게, CNN Feature 추출을 줄이자, Convolution Feature를 한번에 계산하자
- 영역별로 Pooling 함 (Feed Forward를 한번만)
- Feature 에서 Pooling Window를 만들어서 Fully Connected Layer를 만듦
- 문제점 : End to End 하지 않음. R-CNN 대비해 빠르나 아직 속도 측면에서 부족 함



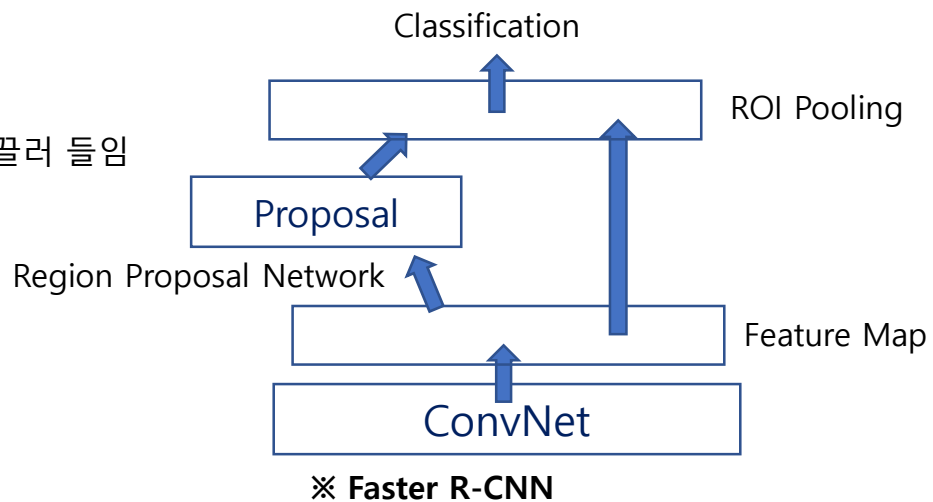
※ Fast R-CNN

- ROI Pooling : Single Level (Average Pooling)
- SVM 대신에 Softmax+Linear Classification이 들어감
- 전체를 학습하는게 차이점 (각각의 False(Bounding Box 내뱉는거)가 각각 학습했던 것을 전체로 학습 함



※ Faster R-CNN

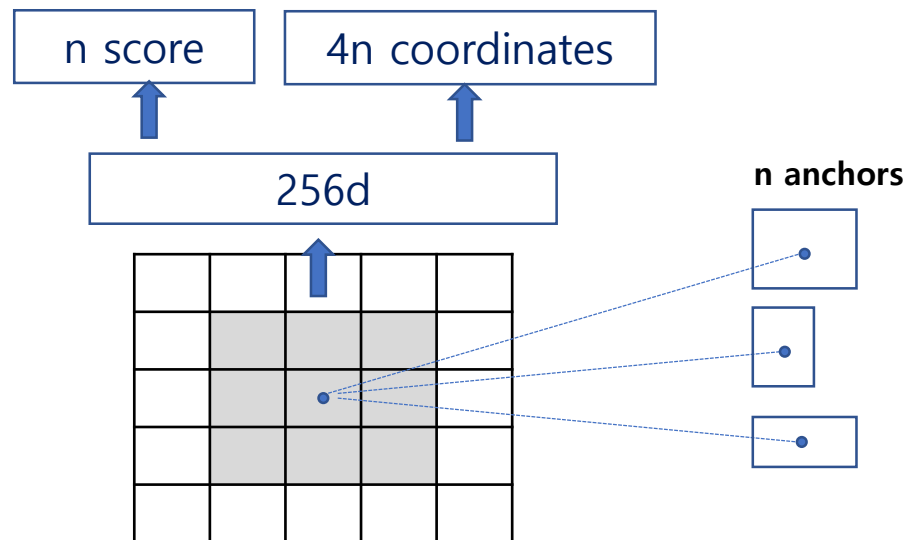
- Region Proposal 하는 것을 Model 안으로 끌러 들임
- 계산을 빠르게 하는 것이 목적
- Detector를 통해 추출된 Region을 공유



Accuracy와 Speed

※ Anchors

- 똑같은 Network을 어느 위치에 놓아도 Proposal 할 수 있다
- Region Proposal이 Model 안으로 들어와서 End to End 가 된다



※ YOLO (You Only Look Once)

- Concept : 모든 것은 하나로 Output을 뽑아 보겠다
- 1) Image를 Grid로 나눈다
- 2) Cell 마다 정해진 Bounding Box로 Object 함 (Object Score 계산)
: 선의 굵기로 Object Score를 표현
- 3) 모든 Cell별로 Bounding Box를 표현 (Region Proposal 생성과 유사)
→ YOLO는 Region Proposal Classification을 같이 함
- 4) Cell에서 가장 유사한 Prediction으로 구분
- 5) Cell Prediction Class 와 Bounding Box를 Merge

@ Output Size 구하기

- Grid
 - Bounding Box / Cell
 - Classes
- Grid x (Bounding Box + Classes)

Ex) 7x7 Grid, 2 Bounding Box/Cell, 20 Classes
: $7 \times 7 \times (2 \times 5 + 20) = 7 \times 7 \times 30 = 1470$ Outputs