**과제 3**

**Object Detection 에서 크게 1 stage 와 2 stage 로 종류가 나뉜다.**

**이들의 예시 모델과 1 stage, 2 stage의 차이점, 장단점을 서술하기**

1. **전반적 차이점**

**Object Detection 문제를 두 Task로 구분**

-> 사물의 위치를 Bounding Box를 통해 나타내는 **Localization**

-> 그 Box안의 사물이 무엇인지 분류하는 **Classification**

이 **두 Task**를 모두 해결해야 하는게 **Object Detection**

-> 동시에 해결하면 : 1-stage Detector ( ex) YOLO )

-> 순차적으로 해결하면 : 2-stage Detector ( ex) R-CNN류 )

1-stage의 특성 : 두 Task를 **동시에 해결**하므로, 속도가 비교적 **빠르지만, 정확도가 낮다**

2-stage의 특성 : **순차적으로 해결**하기에, 속도가 비교적 **느리지만, 정확도가 높다.**

1. **예시 모델과 비교**

**YOLO (1-stage) vs R-CNN (2-stage)**

**1. Summary**

**YOLO : 24개의 Convolutional layer, 2개의 F.C layer로 구성된 Network**

-> 두 Task를 이 Network를 통해 **한번에 수행**

( **확률**값과 관련된 **Regression**으로 문제 해결 )

-> 단일 파이프라인으로 **직관적이며, 최적화가 용이하고 속도가 빠르다.**

**R-CNN : 크게 3가지 과정으로 이루어져 있다.**

**1) Region Proposal**

-> 이미지에서 물체의 영역을 찾아낸다

-> 사용된 알고리즘 : **Selective Search**

( 객체와 주변의 차이를 파악해, Box를 조정해나감 )

**2) CNN**

-> 1) 에서 나온 Box를 **일정 사이즈로 Reshape한 뒤, CNN에** 넣는다 (Alex-Net과 유사)

-> softmax에 넣지않고, **Feature로 이용해 다른 Classifier에** 넣는다.

**3) SVM**

-> 2) 에서 나온 결과물을 SVM에 넣어 Classification을 진행

+ Box의 위치를 조정하기위한 회귀모델 사용

( 실제 Box와 선택된 Box간의 차이 줄이기)

**2. Compare**

**1) Pipe-Line**

**YOLO )** **간단한 파이프라인 (구조)**

-> Neural Network가 파이프 라인의 전부 -> 직관적 이해 가능

-> 극적으로 빠른 속도, 최적화의 용이함, 역전파 가능

**R-CNN )** **복잡한 파이프라인(구조)**

-> 3 단계에 걸쳐, 여러 모델을 복합적으로 사용함 -> 매우 복잡한 구조

-> 1 단계에서 나온 Box안의 모든 이미지를 CNN에 넣어 학습 -> 느림

-> Stage를 나눠서 진행한다

(SVM,회귀모델의 학습 결과로 CNN 업데이트 할 수 없다.(역전파불가))

**2-1) Localization**

**YOLO )**

이미지를 S \* S 의 Grid로 나누고, Box와, Box에 대한 Object의 존재확률,

예측 할 때, 정답일 확률을 동시에 계산해가며 Box 추출

**R-CNN )**

이미지에서 Selective Search 알고리즘을 통해 Box를 추출 이후 회귀모델을

통해 지속적 위치 조정

**2-2) Classification**

**YOLO )**

이미지를 S \* S 의 Grid로 나누고, Box와, Box에 대한 Object의 존재확률,

예측할 때, 정답일 확률을 동시에 계산해가며 Box 추출

**R-CNN )**

나온 Box를 Reshape하고 CNN에 넣는다 (AlexNet과 유사)

-> Softmax 대신 다른 Classifier(SVM)에 넣는다

**3) Conclusion**

**YOLO )** **Image의 Global한 탐색**

-> 배경에 대한 error가 작지만 높은 Localization error

**Object의 일반적 특징을 학습한다**

-> 강력한 일반화, 작은 Object에 약하다

**빠르지만, 분류 성능이 조금 떨어진다.**

**R-CNN ) Box를 Seletive Search를 통해 구한다**

-> 배경에 대한 error가 존재하나 강력

**느리지만 대체적인 분류 성능이 높다.**