

# Lab2 Final Report

## YOLO : Object Detection Model 생성 및 성능 분석

정보컴퓨터공학부 202355514 강지원

### 1 주요 내용

본 Lab에서는 Pascal VOC 데이터셋을 이용하여 object detection 수행 모델인 YOLO를 구현하고 성능을 분석한다.

### 2 모델 설명

YOLO는 You Only Look Once의 약어로, 이미지 전체를 한 번만 보고 그 안의 어떤 객체가 있으며 어디에 위치하는지를 예측하는 객체 탐지를 위한 알고리즘이다. YOLO는 객체 탐지를 위한 단일 회귀 문제(single regression problem)로 재정의하여 이미지 픽셀로부터 바로 bounding box 좌표와 클래스 확률을 예측한다. 격자 셀(grid cell)은 자신이 담당하는 영역 안에 어떤 객체가 있는지와 그 위치를 예측한다. 각 셀은 두개의 바운딩 박스를 예측하며, 박스는 ( $x, y, w, h, confidence$ )로 구성된다. 여기서  $confidence$ 는 신뢰도로, 예측한 박스가 실제 객체와 얼마나 겹치는지를 나타내는 값이다.

#### 2.1 YOLO : 모델 구조

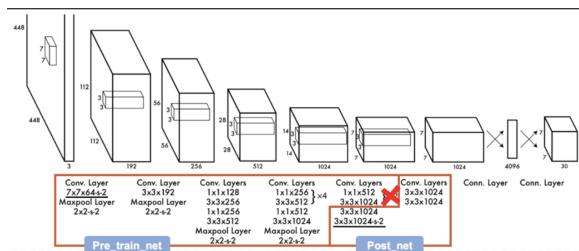


Figure 1: YOLO 모델 구조

전체 네트워크는 22개의 Convolution layer, 4개의 Max Pooling layer, 2개의 Fully Connected Layer로 구성되어 있으며, 모든 Convolution layer 후에는 LeakyReLU 활성화 함수를 적용한다. (마지막 출력층 제외) 모델의 전반부 self.pre\_train\_net은 입력 이미지로부터 특징을 추출하는 부분으로, 1번쨰와 20번째 합성곱 계층에서 stride=2를 사용해 출력 크기를 절반으로 줄여 중요한 특징만을 추출한다. 이 과정에서 4개의 max pooling layer가 사용되며 최종적으로  $7 \times 7 \times 1024$  크기의 feature map이 생성된다.

모델의 후반부 self.post\_net에서는 깊은 연산을 통해 객체의 위치 정보와 분류에 필요한 고수준의 표현을 학습한다.

마지막으로 self.FC1, self.FC2에서 Fully Connected Layer를 통해 최종 예측을 수행한다.

### 3 손실 함수

#### 3.1 Classification : compute\_prob\_error()

객체가 존재하는 셀에 대해서만 예측된 클래스 확률과 실제 클래스 확률의 차이를 계산한다.

#### 3.2 No-object confidence : not\_contain\_obj\_error()

객체가 존재하는 셀에 대해서만 예측된 클래스 확률과 실제 클래스 확률의 차이를 계산한다.

### 3.3 Object confidence, bounding box regression : contain\_obj\_error()

객체가 존재하는 셀에 대해서만 예측된 클래스 확률과 실제 클래스 확률의 차이를 계산한다.

### 4 모델학습

PASCAL VOC 데이터셋을 사용하였으며, 배치 크기는 10으로 설정하였다. 모델은 사전 학습된 pre\_train.pt를 불러온 후, 학습률 1e-4, momentum 0.9, weight decay 5e-4로 설정한 SGD(Stochastic Gradient Descent) 옵티마이저를 사용해 학습하였다. 앞서 정의한 yoloLoss의 파라미터를  $\lambda_{coord}=5$ ,  $\lambda_{noobj}=0.5$ 로 설정하였다. 학습은 15 epoch 동안 수행되었다. 또한, 기울기 폭발 방지를 위해 Gradient Clipping을 추가 적용해 값이 지나치게 커지는 것을 막았다.

학습 과정에서 avg Loss 값은 240에서 시작해 0.54에 도달하여 안정적으로 수렴하는 모습을 보였다.

### 5 성능 분석

#### 5.1 시각화

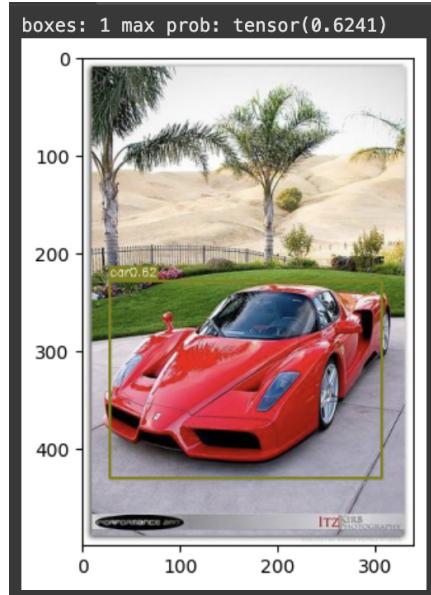


Figure 2: 시각화 결과

학습된 모델의 성능을 육안으로 확인하기 위해 Bounding Box 시각화 과정을 수행하였다. 각 박스의 좌표와 클래스 정보를 이용해 이미지 위에 예측 결과를 시각적으로 표시하였다. 또한 탐지된 객체 개수, 그에 따른 max confidence를 추가로 출력해서 확인해보았다.

#### 5.2 mAP

학습된 모델의 성능을 검증하기 위해 테스트 데이터셋을 이용해 객체 탐지 수행 후, 각 클래스별 Average Precision(AP) 값을 계산하여 모델의 성능을 평가해 보았다. 클래스 별 결과는 63.7% ~ 76.4% 사이의 값을 가지며 따라서 mAP는 69.0%로 나타났다.

```
/content/drive/MyDrive/yolo/lab2/utils/Generate_dataset  
63.74% = aeroplane AP  
64.94% = bicycle AP  
64.48% = bus AP  
69.19% = car AP  
75.24% = cat AP  
76.40% = dog AP  
mAP = 69.00%  
<Figure size 640x480 with 0 Axes>
```

Figure 3: mAP 지표 결과

## 6 결론

본 실험에서는 YOLO(Object Detection) 모델을 구현하고, PASCAL VOC 데이터셋을 기반으로 학습 및 성능 평가를 수행하였다. 모델은  $\lambda_{coord}=5$ ,  $\lambda_{noobj}=0.5$ 로 설정된 커스텀 손실 함수를 사용하였으며, Gradient Clipping을 적용하여 안정적인 학습을 달성하였다. 실험 결과, 시각화 지표와 69.00%의 mAP 성능을 통해 위치 및 클래스의 구분이 대부분 잘 이루어지고 있음을 알 수 있었다. 이를 통해 모델 구조, 손실 함수, 학습 절차가 올바르게 구현되었음을 확인하였다.

- [1] S. Ren, K. He, R. Girshick, and J. Sun, “Object Detection with Region Proposal Network,” , 2015. <https://arxiv.org/abs/1506.01497>