YOLO

You Only Look Once: Unified, Real-Time Object Detection

(Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, 2016, CVPR)

IVPG Lab Seminar 2022.03.30
세종대학교 지능기전공학부
18학번 장윤정

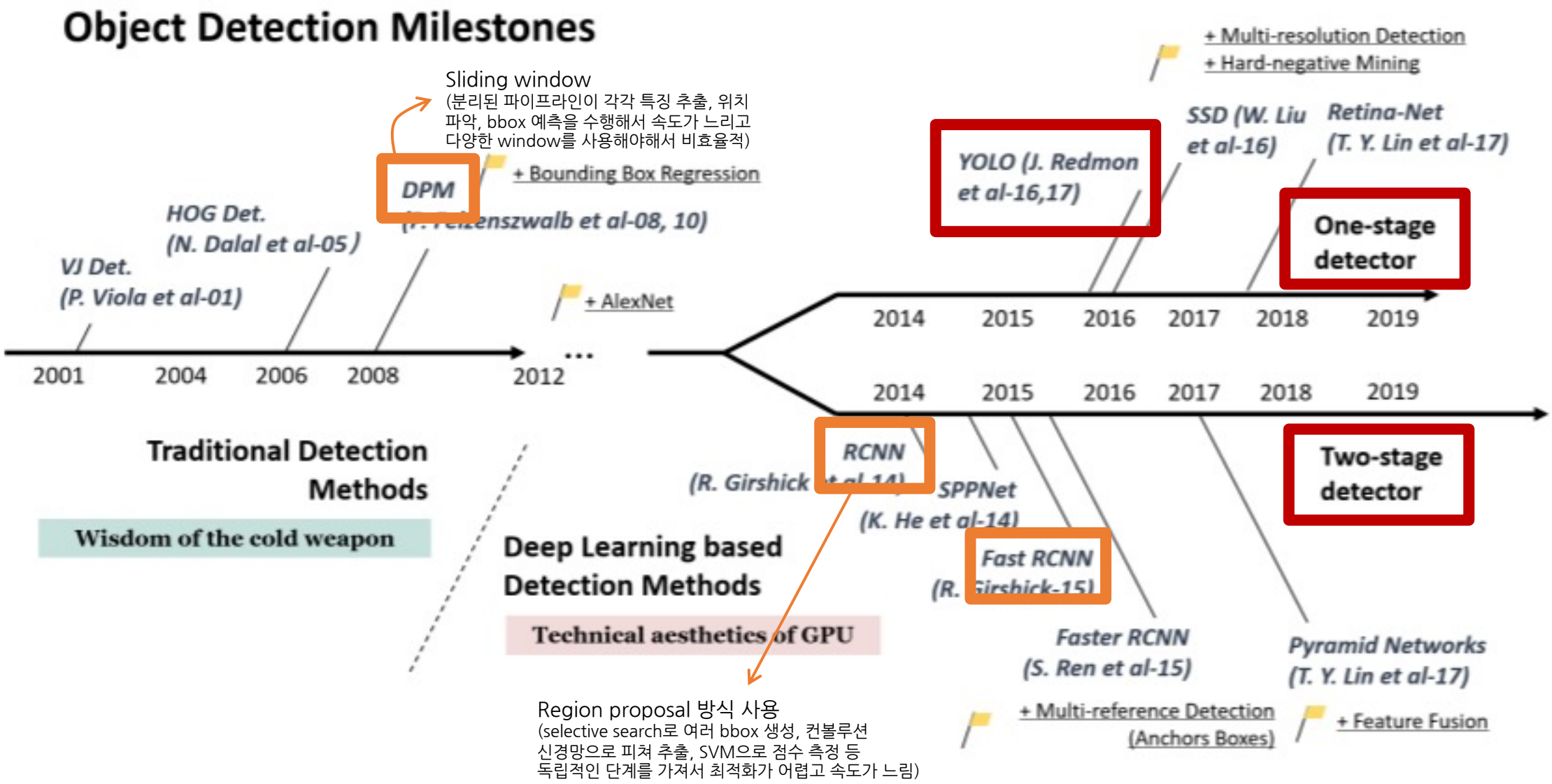
이미지를 보는 횟수는 한 번이면 된다

You Only Look Once: Unified, Real-Time Object Detection

실시간 시스템 가능 (속도에 자신 있는 모델)

Classification & Localization
단계가 합해진 단일 네트워크 모델

Introduction



Introduction

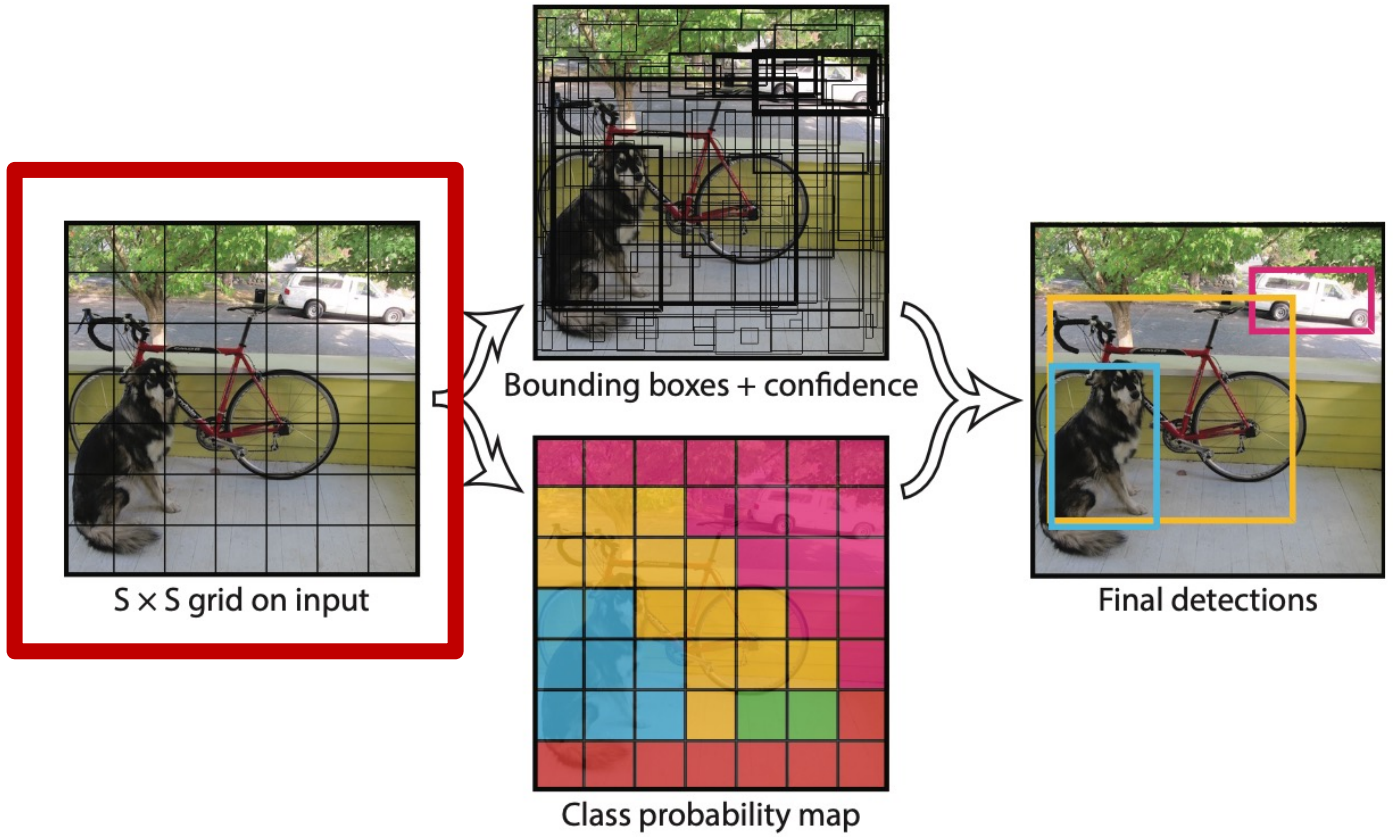
▪ YOLO의 장점

- 당시에 성능이 비슷한 모델들(20 FPS 이하) 중 가장 빠른 속도 (45 FPS)
- 당시에 다른 Real-Time object detection model보다 2배 가량의 mAP 개선 (63.4 mAP)
- 학습 및 테스트 과정에서 이미지 전체를 보기 때문에, 배경을 물체로 인식하는 Background error 감소
- 물체의 일반적인 표현을 학습하기 때문에 새로운 도메인에 대한 일반화 측면에서 더 robust 함

▪ YOLO의 한계

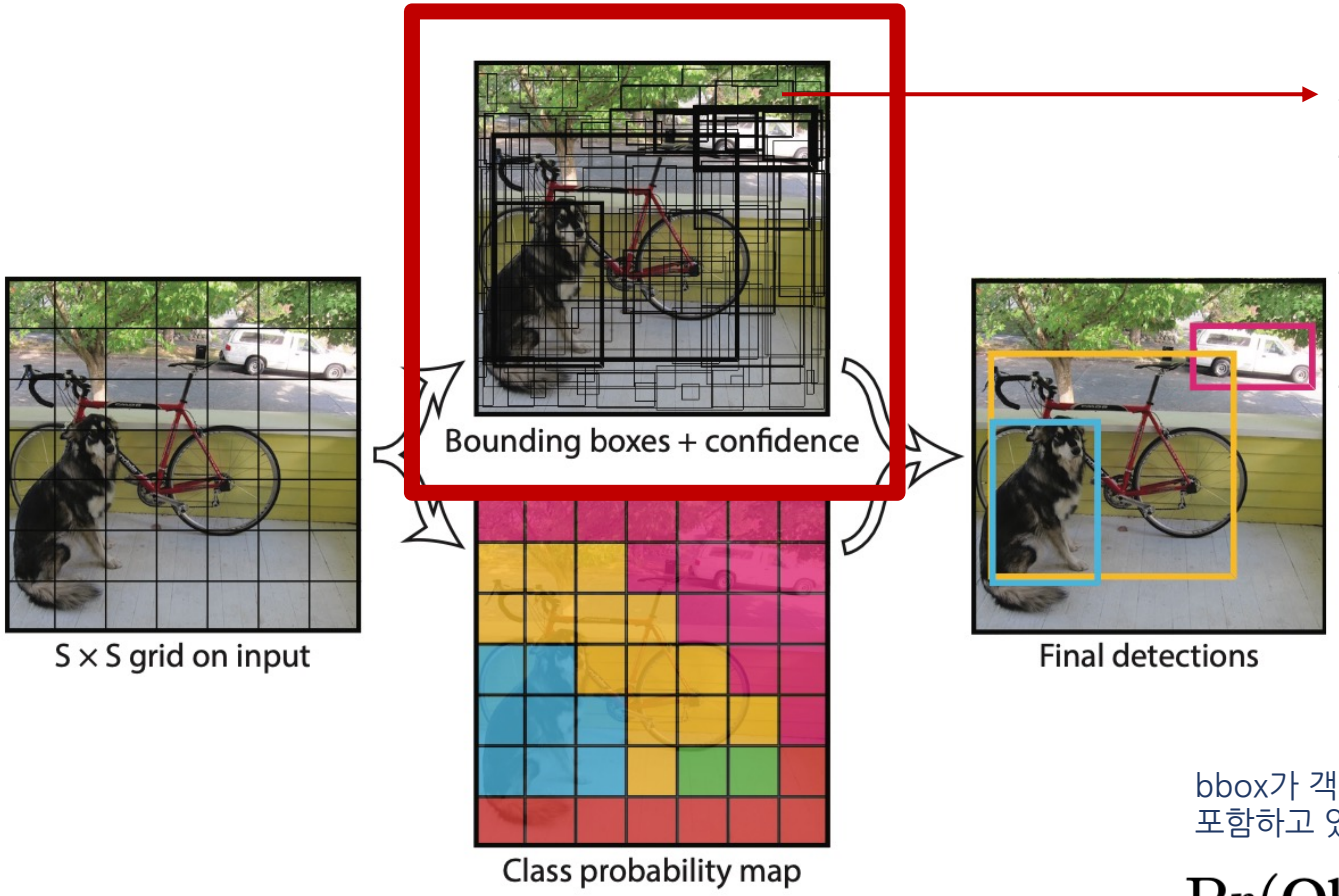
- 당시 SOTA 모델 (73.2 mAP)에 비해서는 정확도가 떨어짐
- 하나의 그리드 셀 마다 하나의 물체만 검출 가능해서 두 개 이상의 물체가 붙어 있는 경우 잘 검출하지 못함
  (e.g. 새 떼처럼 작은 물체가 몰려 있는 경우)
- 학습 데이터에 존재하는 bbox들을 학습하는 것이므로, 전혀 새로운 종횡비를 가진 물체가 등장하면 정확도 감소
- 큰 bbox에 비해 작은 bbox는 위치가 조금만 달라져도 IOU가 크게 변하는데,
  loss 계산 시에 이를 특별히 고려하지 않기 때문에 작은 크기의 물체에서 localization이 잘못되는 경우 발생

Unified Detection



1. 이미지를 S x S의 그리드로 나눔 (In the paper : 7 x 7)

Unified Detection



각각의 bbox는 5개의 예측치로 이루어짐
- (x, y) : bbox의 중심을 그리드 셀 내 상대 위치로 표현한 값(0~1)
- (w, h) : bbox를 이미지 전체 크기에 대해 상대 너비와 상대 높이로 표현한 값 (0~1)
- confidence

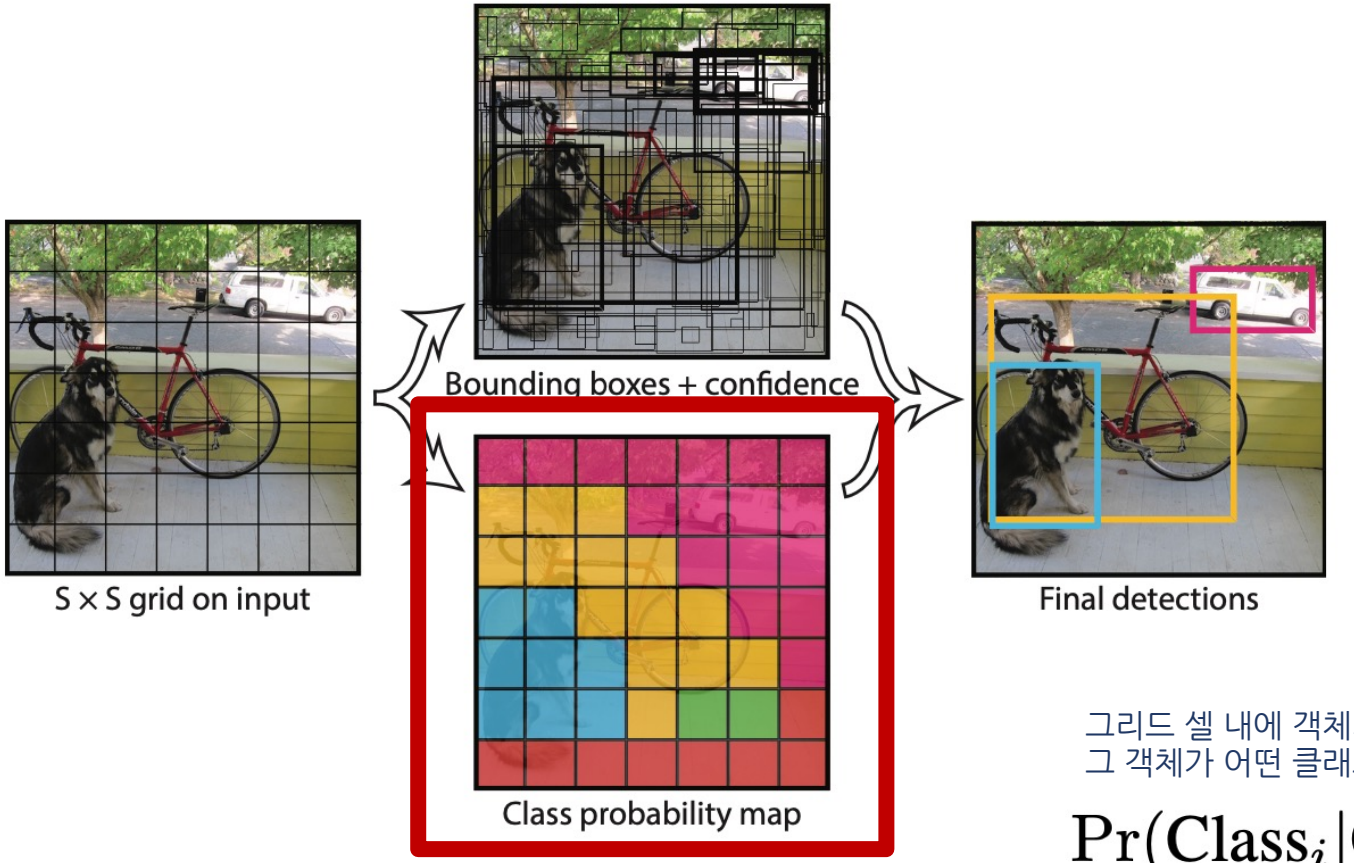
bbox가 객체를 포함하고 있는지,
포함하고 있다면 GT와 비교한 IOU는 얼마나 되는지

$$\Pr(\text{Object}) * \text{IOU}_{\text{pred}}^{\text{truth}}$$

2. 각각의 그리드 셀은 B개의 **bounding box(이하 bbox)**와 그 bbox 대한 **confidence score** 예측

(In the paper : B = 2 → 한 이미지 당 98개(7x7x2)의 bbox)

Unified Detection



그리드 셀 내에 객체가 있다는 조건 하에,
그 객체가 어떤 클래스에 속하는지에 대한 확률

$$\Pr(\text{Class}_i | \text{Object})$$

3. 각각의 그리드 셀은 C개의 **conditional class probabilities**를 가짐

(In the paper : C = 20 → PASCAL VOC dataset class)

Unified Detection

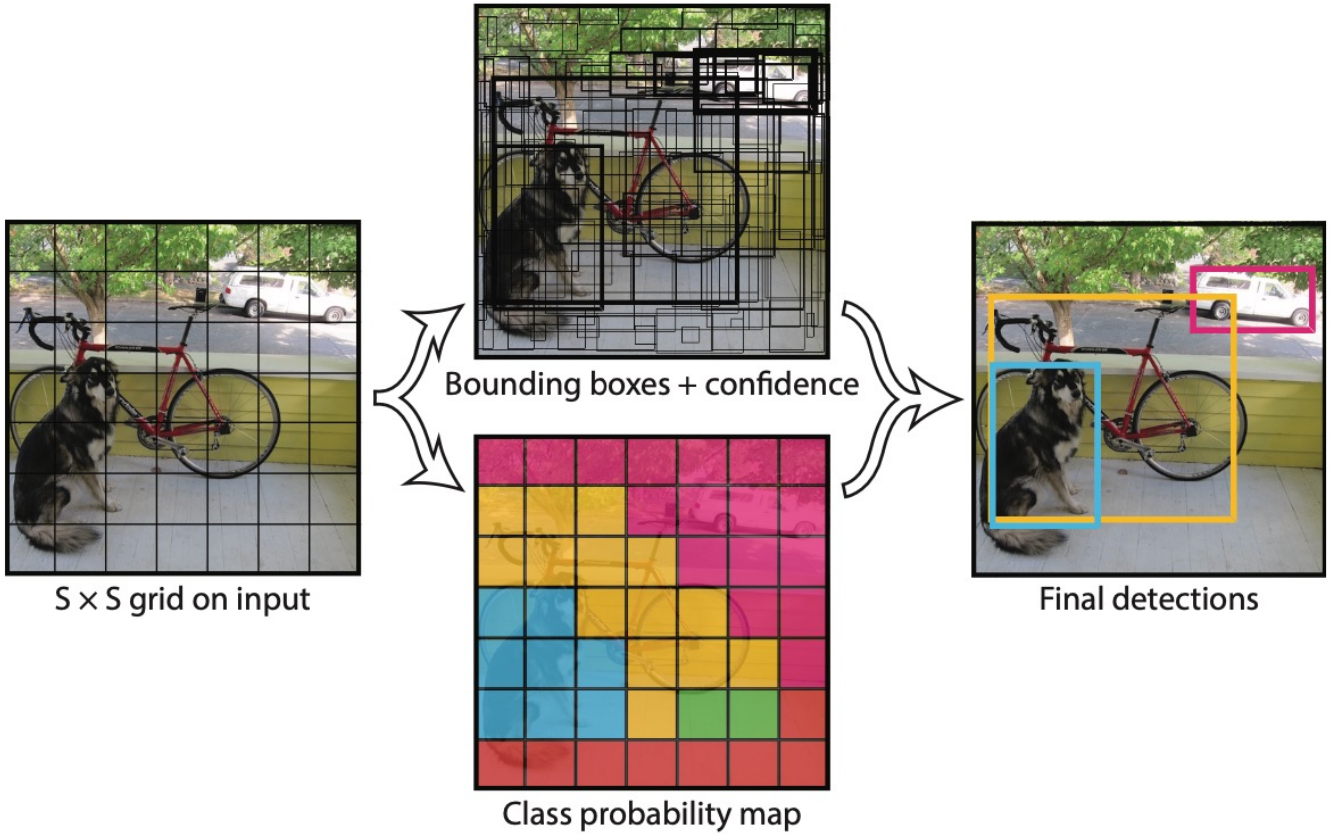


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor.

→ In the paper : 한 이미지 당 최종 예측 텐서는 7 x 7 x 30

Network Design

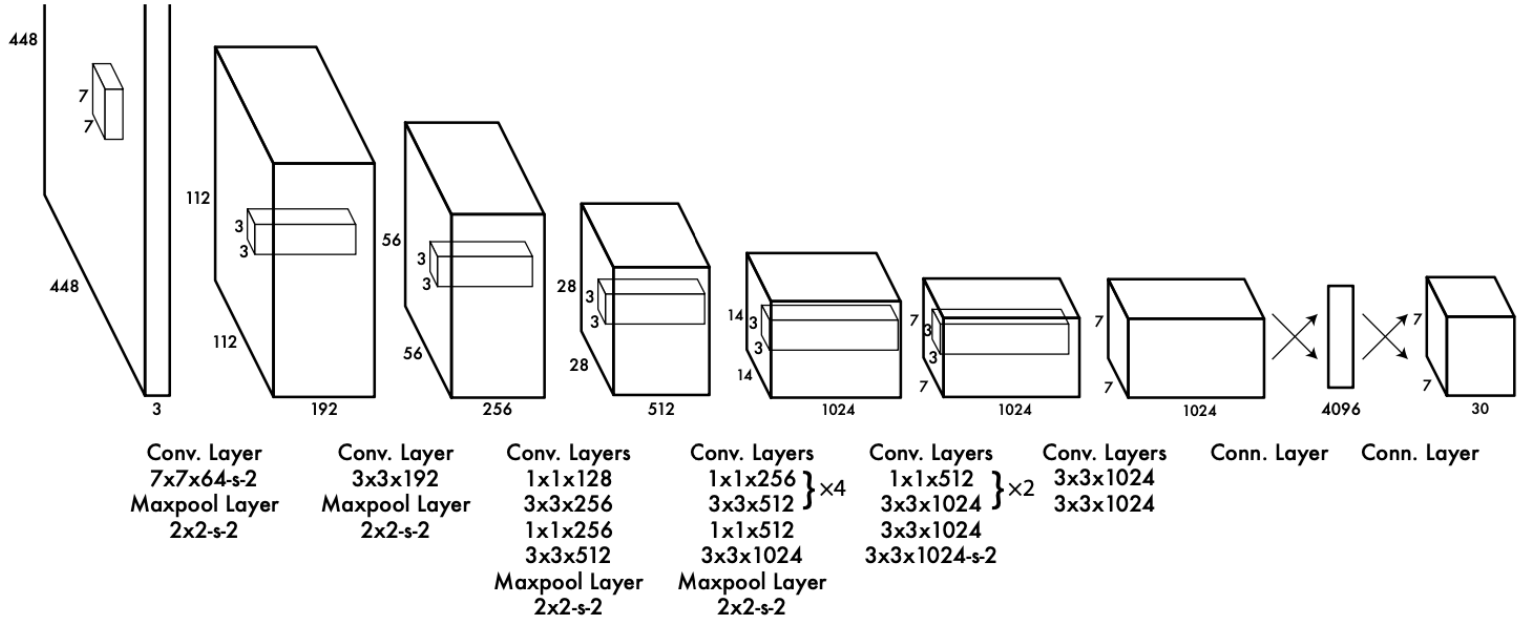


Figure 3: The Architecture. Our detection network has 24 convolutional layers followed by 2 fully connected layers. Alternating 1×1 convolutional layers reduce the features space from preceding layers. We pretrain the convolutional layers on the ImageNet classification task at half the resolution (224×224 input image) and then double the resolution for detection.

- **24 conv layer + 2 fc layer** (GoogLeNet의 구조를 기반으로 함)
  - 20 conv layer : pretrained with 1000 class ImageNet dataset
  - 4 conv layer + 2 fc layer : fine-tuned with 20 class PASCAL VOC dataset
- 중간에 1x1 reduction layer와 3x3 conv layer의 결합으로 GoogLeNet의 인셉션 구조를 대신하며 연산량 감소
- 네트워크의 최종 아웃풋은 **7x7x30의 예측 텐서**
- Fast YOLO는 더 빠른 속도를 위해 24 conv layer → 9 conv layer

Training

- 마지막 계층 (linear activation function)을 제외한 모든 계층에 **leaky ReLU** 적용 $\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases}$
- Dataset : **PASCAL VOC** 2007, 2012
- Epochs : 135
- Batch size : 64
- Momentum : 0.9
- Decay = 0.0005
- Learning rate : 0.001 (1~3 epoch) → 0.01 (4~74 epoch) → 0.001 (74~104 epoch) → 0.0001 (105~135 epoch)
- Overfitting을 방지하기 위해
  - Dropout = 0.5
  - Data augmentation : original image의 20%까지 random scaling, random translation

Loss function

- YOLO의 loss는 SSE(Sum-Squared Error, 오차제곱합)를 기반으로 함 (최적화 용이)
- 문제 1) SSE는 localization loss와 classification loss의 가중치를 동일하게 취급함

  → classification loss보다 localization loss의 가중치를 증가시켜 줌 (balancing parameter : λ_coord = 5)
- 문제 2) 이미지 내 대부분의 그리드 셀에는 객체가 없기 때문에 (confidence score = 0) 불균형 초래

  → 객체가 없는 그리드 셀의 confidence loss를 객체가 있는 그리드 셀에 비해 감소시켜 줌 (balancing parameter : λ_noobj = 0.5)
- 문제 3) SSE는 크기가 큰 bbox와 작은 bbox에 대해 동일한 가중치로 loss를 계산함

  → bbox의 너비와 높이에 square root를 취해서 크기가 커짐에 따라 그 증가율이 감소해서 큰 bbox의 loss에 대한 가중치를 감소

그리드 셀 i의 j번째 bbox predictor가 사용되는지를 의미

$$\lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[(x_i - \hat{x}_i)^2 + (y_i - \hat{y}_i)^2 \right]$$

(1) 물체가 존재하는 그리드 셀 i의 bbox predictor j에 대해, x와 y의 loss 계산

$$+ \lambda_{\text{coord}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left[\left(\sqrt{w_i} - \sqrt{\hat{w}_i} \right)^2 + \left(\sqrt{h_i} - \sqrt{\hat{h}_i} \right)^2 \right]$$

(2) 물체가 존재하는 그리드 셀 i의 bbox predictor j에 대해, w와 h의 loss 계산

$$+ \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{obj}} \left(C_i - \hat{C}_i \right)^2$$

1

(3) 물체가 존재하는 그리드 셀 i의 bbox predictor j에 대해, confidence score의 loss 계산

$$+ \lambda_{\text{noobj}} \sum_{i=0}^{S^2} \sum_{j=0}^{B} \mathbb{1}_{ij}^{\text{noobj}} \left(C_i - \hat{C}_i \right)^2$$

0

(4) 물체가 존재하지 않는 그리드 셀 i의 bbox predictor j에 대해, confidence score의 loss 계산

그리드 셀 i안에 객체가 존재하는지 여부(존재하면 1, 없으면 0)

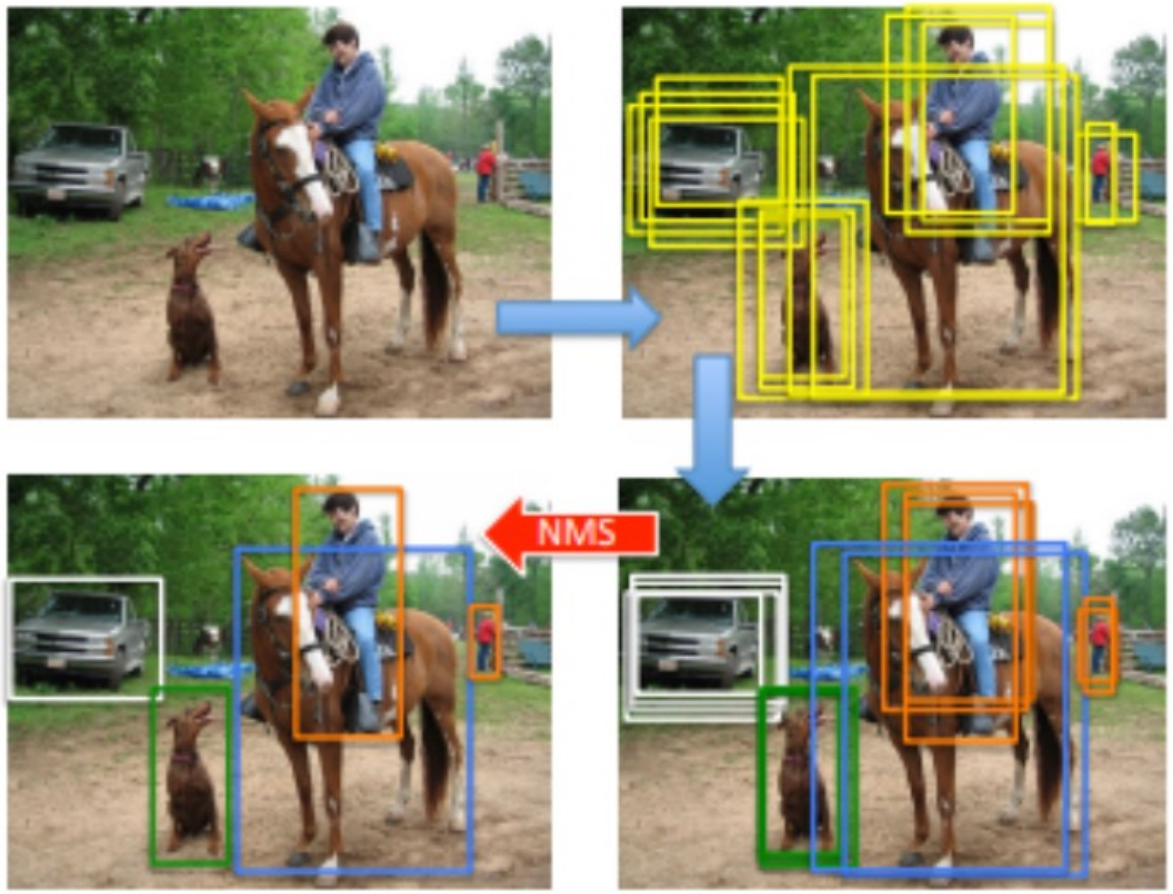
$$+ \sum_{i=0}^{S^2} \mathbb{1}_{i}^{\text{obj}} \sum_{c \in \text{classes}} \left(p_i(c) - \hat{p}_i(c) \right)^2$$

(5) 물체가 존재하는 그리드 셀 i에 대해, conditional class probability의 loss 계산

Inference

- Training과 마찬가지로, Inference에서도 테스트 이미지를 하나의 신경망에 넣어서 계산하면 됨 → **빠른 속도**
- 객체의 크기가 크거나 그리드 셀 경계에 인접해 있는 경우, 한 객체에 대한 bbox가 여러 개 생기는 multiple detections 발생

  → **NMS(non-maximum suppression)** 방법으로 개선 가능 (mAP 2~3% 향상)



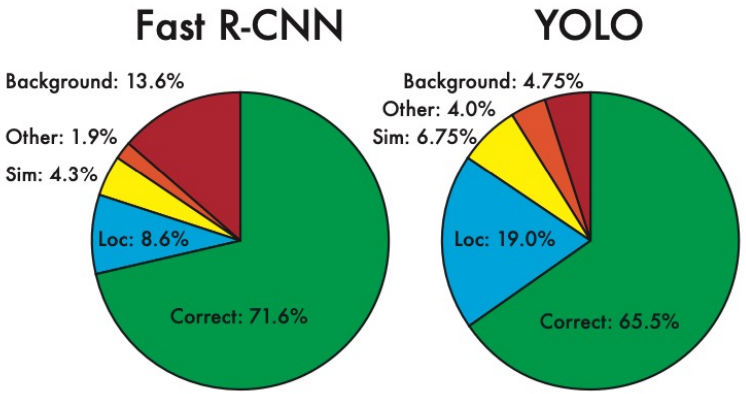
Experiments

[Experiment #1]

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [31]	2007	16.0	100
30Hz DPM [31]	2007	26.1	30
Fast YOLO	2007+2012	52.7	**155**
YOLO	2007+2012	**63.4**	45
Less Than Real-Time			
Fastest DPM [38]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[28]	2007+2012	73.2	7
Faster R-CNN ZF [28]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

- Real-Time Detector들과 비교하면 mAP가 2배 이상 좋음
- Real-Time이 아닌 Detector들과 비교해도 mAP가 더 좋거나 큰 차이 없음
- YOLO 네트워크를 VGG-16으로 바꿔서 학습했더니 mAP는 더 좋아지지만 FPS가 떨어져서 실시간 객체 검출 불가

[Experiment #2]



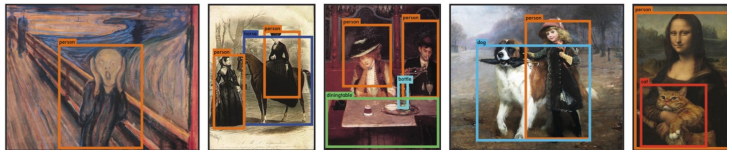
- Fast R-CNN과 비교해보면, localization error는 더 크지만 background error(배경에 아무 물체가 없는데 물체가 있다고 판단하는 false positive error)를 3배 가량 줄임

[Experiment #3]

	mAP	Combined	Gain
Fast R-CNN	71.8	-	-
Fast R-CNN (2007 data)	**66.9**	72.4	.6
Fast R-CNN (VGG-M)	59.2	72.4	.6
Fast R-CNN (CaffeNet)	57.1	72.1	.3
YOLO	63.4	**75.0**	**3.2**

- Fast R-CNN이 다른 모델들과 앙상블 했을 때보다 YOLO와 앙상블 했을 때가 가장 성능이 좋음

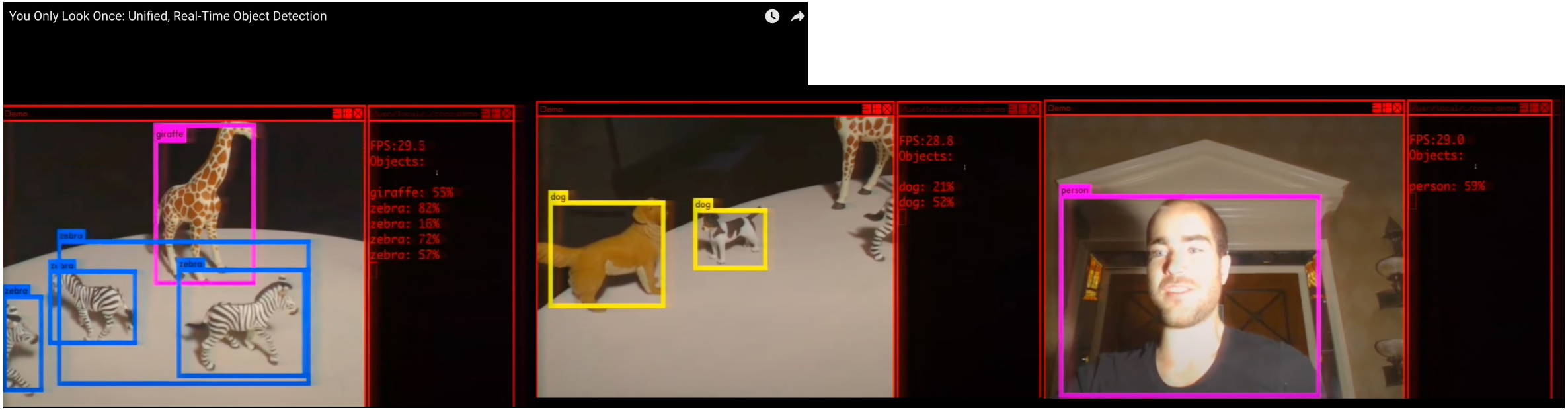
[Experiment #4]



- 피카소, 예술 작품 데이터셋으로 실험해 본 결과, 다른 모델들에 비해 YOLO가 가장 성능이 좋음
- 일반화가 잘 되었다고 주장

Conclusion (Contribution)

- **YOLO**는 네트워크가 **단순**하면서도 **빠르고 정확한** Object detection을 위한 Unified 모델
- Object detection task에 대해 **regression problem**으로 관점 전환
- **Unified Architecture** : classification과 localization을 단일 신경망 네트워크로 수행하는 one-stage object detector의 포문을 엶
- 당시에 DPM, R-CNN 계열 모델들 보다 **속도**를 월등히 개선
- 여러 도메인에 대해 **일반화**가 잘 되므로 Real-Time computer vision application에 활용할만한 가치가 있음



Reference

- https://www.youtube.com/watch?v=Ae-p7QVOdbA (허훈 - YOLO(you only look once) 논문 리뷰)
- https://www.youtube.com/watch?v=eTDcoeqj1_w (PR-016: You only look once: Unified, real-time object detection)
- https://www.youtube.com/watch?v=cNFpo7kDf-s (박경찬 - YOLO)
- https://www.youtube.com/watch?v=O78V3kwBRBk ([Paper Review] You Only Look Once : Unified, Real-Time Object Detection)
- https://www.youtube.com/watch?v=8DjIJc7xH5U (십분딥러닝_14_YOLO(You Only Look Once))
- https://www.youtube.com/watch?v=lxyCUfn_p4Q https://www.youtube.com/watch?v=ccnL_ODHfys (PIEW9 논문 리뷰)
- https://bkshin.tistory.com/entry/%EB%85%BC%EB%AC%B8-%EB%A6%AC%EB%B7%B0-YOLOYou-Only-Look-Once (Baek Kyun Shin)

감사합니다 ☺

IVPG Lab Seminar 2022.03.30

세종대학교 지능기전공학부

18학번 장윤정