CBAM 활용 객체 인식 모델 개선 연구

컴퓨터비전 특론 어플리케이션 연구 과제

인공지능학과 2023320001 윤준영
인공지능학과 2023320004 강대열
컴퓨터공학과 2023407001 김현창
스마트항공모빌리티학과 2022318010 윤석빈

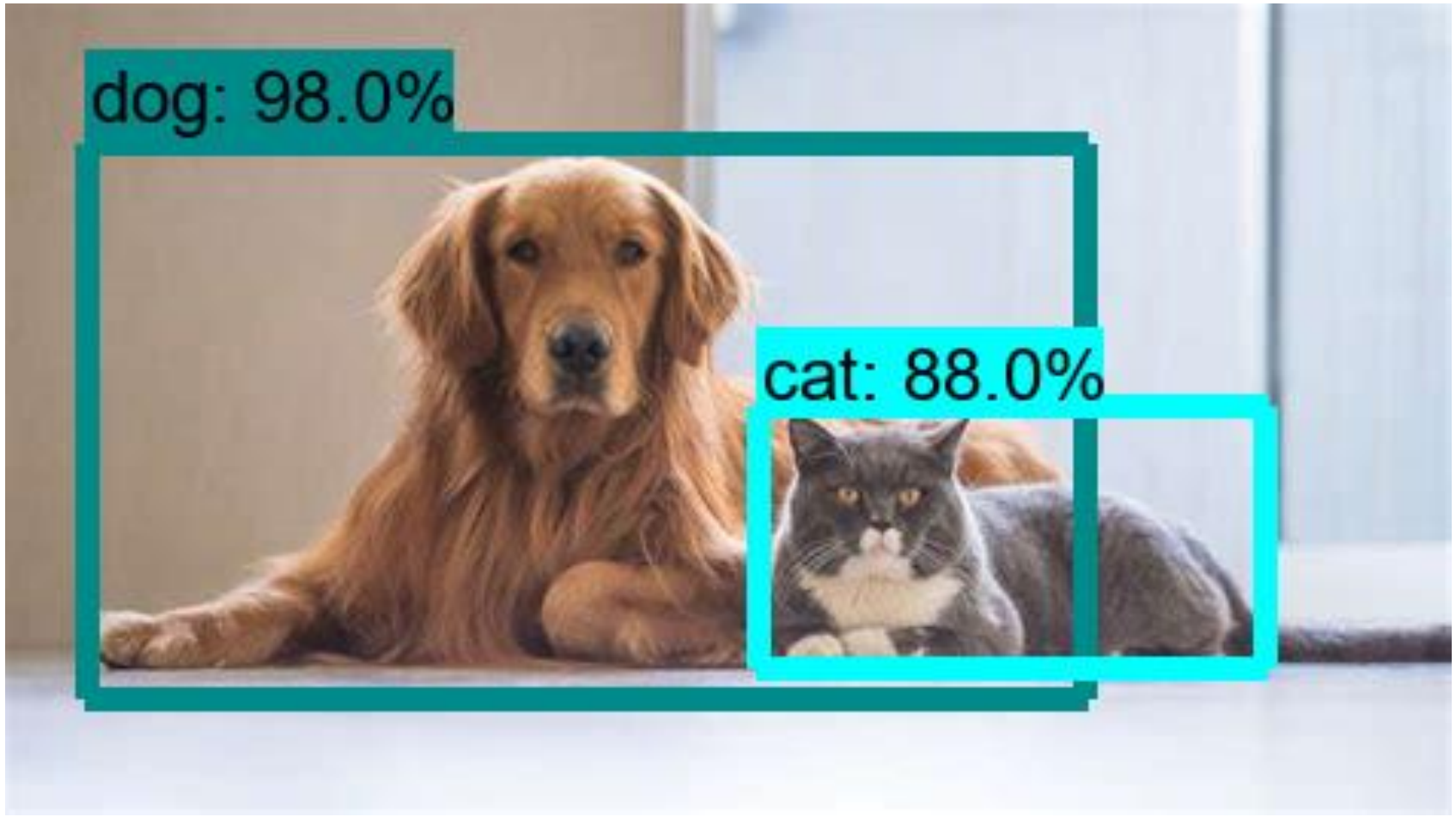
연구 선정 배경

- **수업과 연계성:** Convolutional Neural Network(CNN)을 활용한 Object detection은 컴퓨터비전에서 다루는 중요한 개념이며, 프로젝트 수행을 통해 심도 있는 학습이 가능함

- **높은 활용성:** 객체인식(object detection)은 컴퓨터비전 분야에서 가장 많이 활용되는 기술로 자율주행, 보안 감시, 의료 분야 등에서 많이 사용됨

- **연구의 대표성:** 본 프로젝트를 통해 연구하려는 알고리즘은 YOLO와 Convolutional Block Attention Module (CBAM)이며 이는 딥러닝 기반 객체인식 성능 개선에 기여함

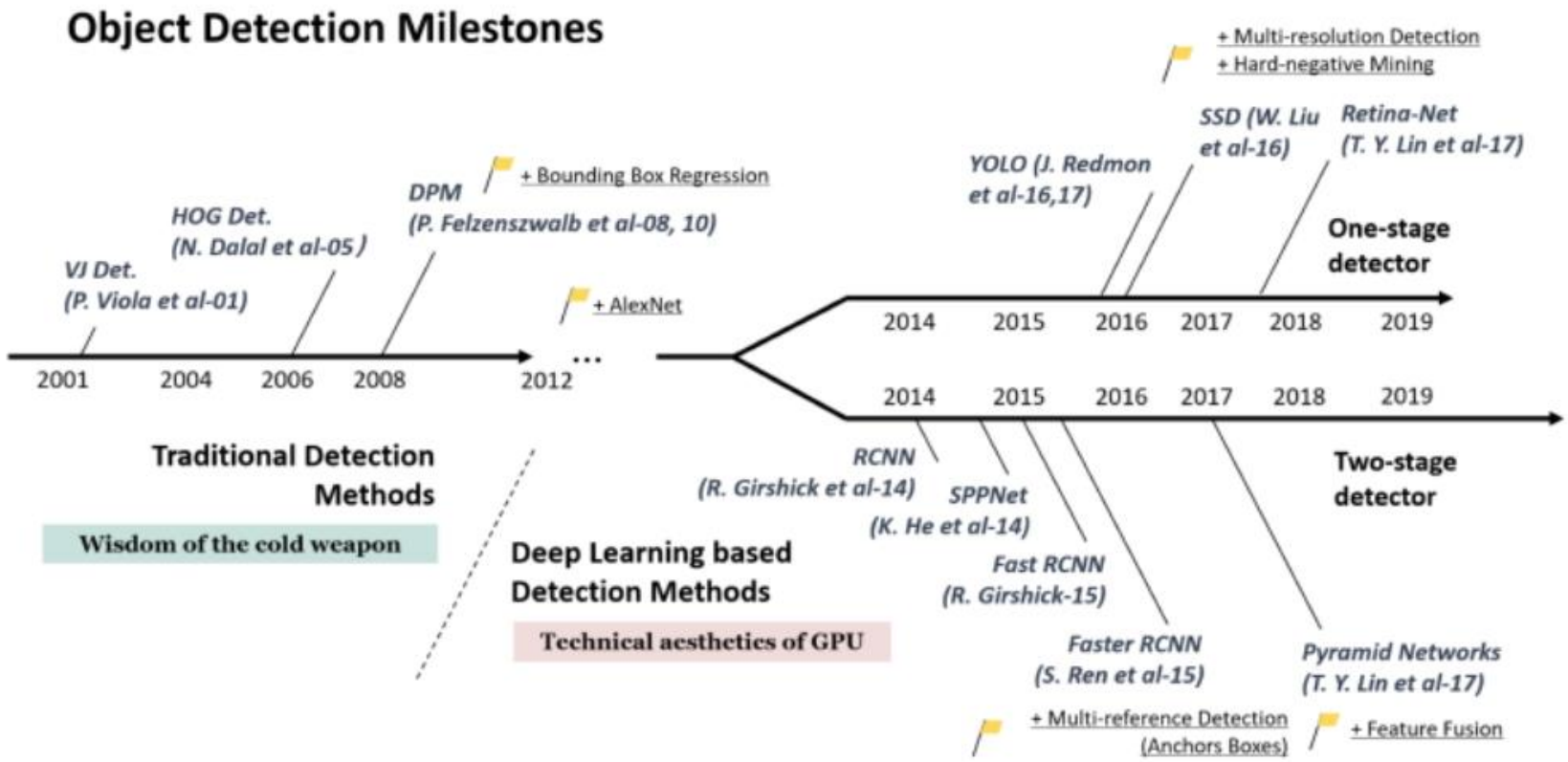
객체 인식(Object Detection)



*객체 인식(Object Detection) 기술은 이미지 내에서 특정 객체의 위치와 종류를 찾고 분류하는 컴퓨터 비전의 한 분야

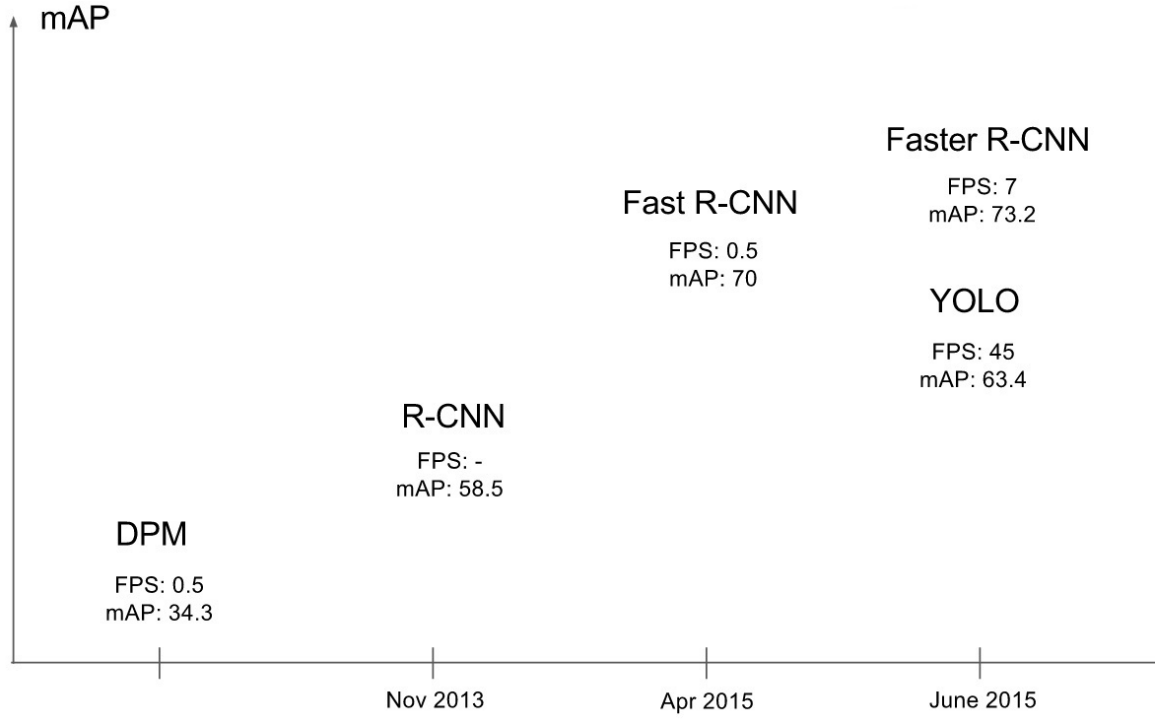
객체 인식(Object Detection) 기술 발전

객체 인식(Object Detection) 기술은 이미지 내에서 특정 객체의 위치와 종류를 찾고 분류하는 컴퓨터 비전의 한 분야



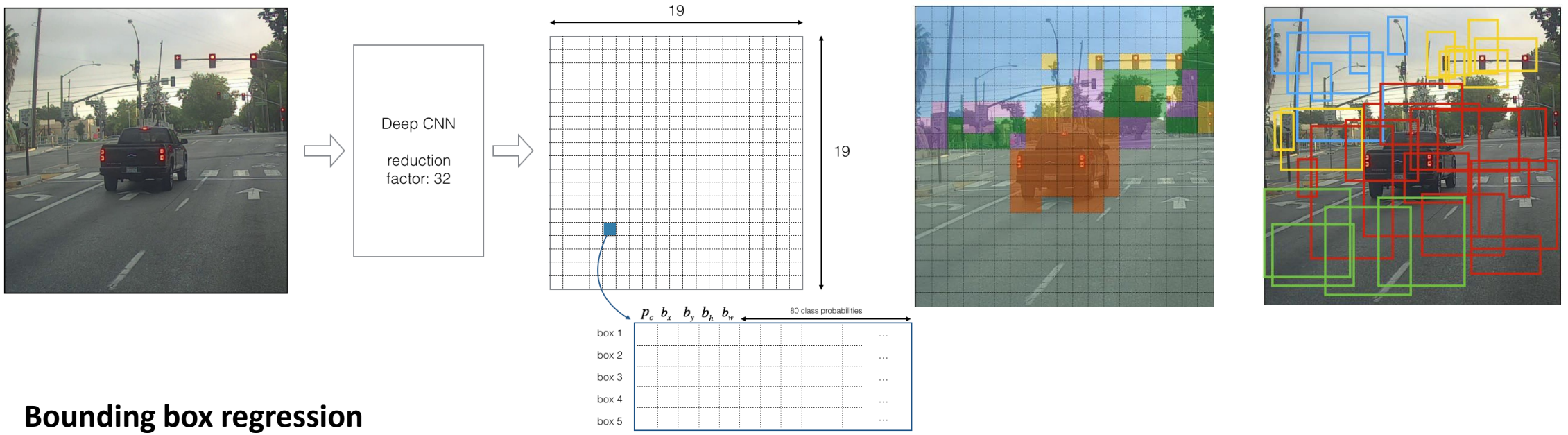
YOLO

- **YOLO(You Only Look Once) 개요**
  - YOLO는 실시간 객체 탐지를 위한 딥러닝 기반 알고리즘
  - 이미지를 한 번만 보고 객체의 위치와 클래스를 동시에 예측하여 빠른 속도와 높은 정확도를 제공
  - 대상의 일반적인 특징을 학습하기 때문에 다른 영역으로의 확장에서도 뛰어난 성능을 보임



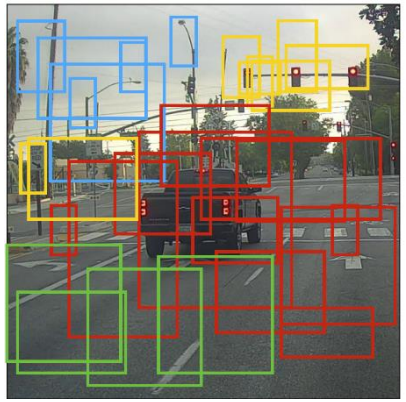
YOLOv3 – Detection

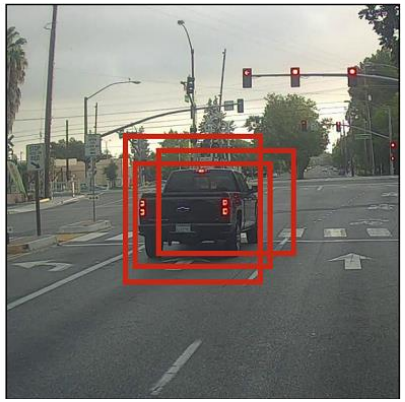
- **Grid Cell**
  - 이미지를 S x S 크기의 Grid cell로 분할
  - Feature Network에서 Cell마다 B개의 Bounding boxes, Confidence score, class probability를 예측



- **Bounding box regression**
  - Anchor box는 dataset에서 계산된 bounding box prior
  - Anchor box로 크기와 위치를 실제 객체가 있는 부분으로 조절하여 bounding box 결정

YOLOv3 – Detection





- **Two bounding box filtering methods**

  1. First method

  - 신뢰도가 낮은 Bounding box 제거
  - Confidence 정보를 이용하여 thresholding

  2. Second method

  - Bounding box가 겹치면 하나를 제외하고 나머지를 지워주어야 함
  - IoU값을 통해 non-maximal 한 값들을 모두 지워줌 -> NMS(비최대억제)

$$IOU(Intersection\ of\ Union) = \frac{Area\ of\ Intersection}{Area\ of\ Union}$$

- **Classification**
  - Classification에 대한 확률이 가장 높은 classification 값을 해당 bounding box의 객체라고 판단
  - 즉 conf의 정보는 이 anchor box가 존재할 확률, classification의 확률은 특정 label이 붙을 확률을 의미

YOLOv3 – Feature Network

- **Network design**
  - Darknet53 backbone: 53 Convolution layers for feature detection
  - Pyramid, Skip connection: 3 scale prediction

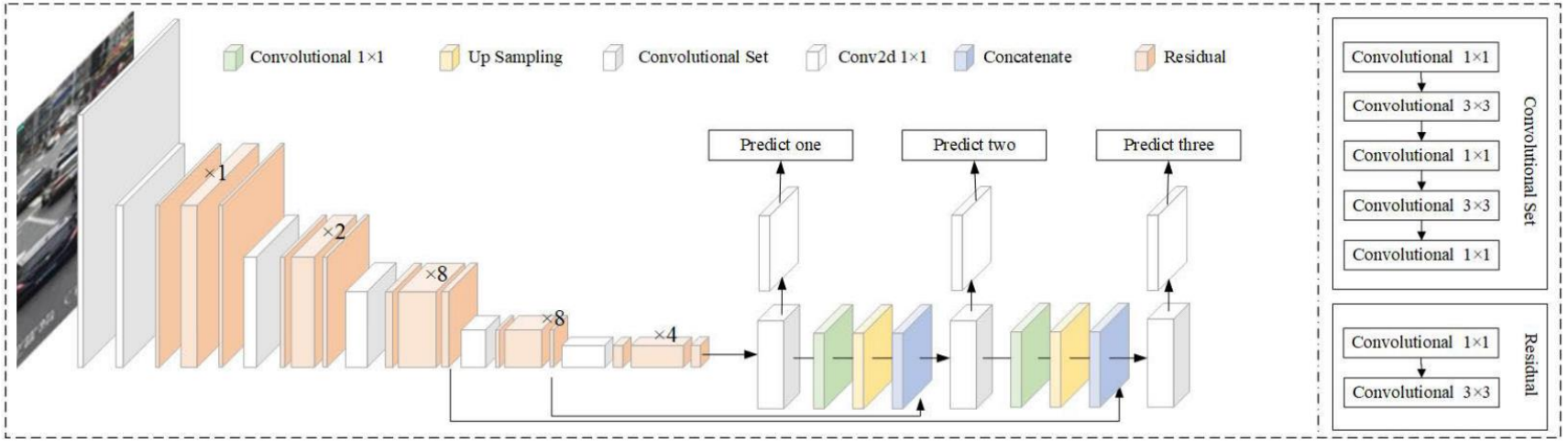
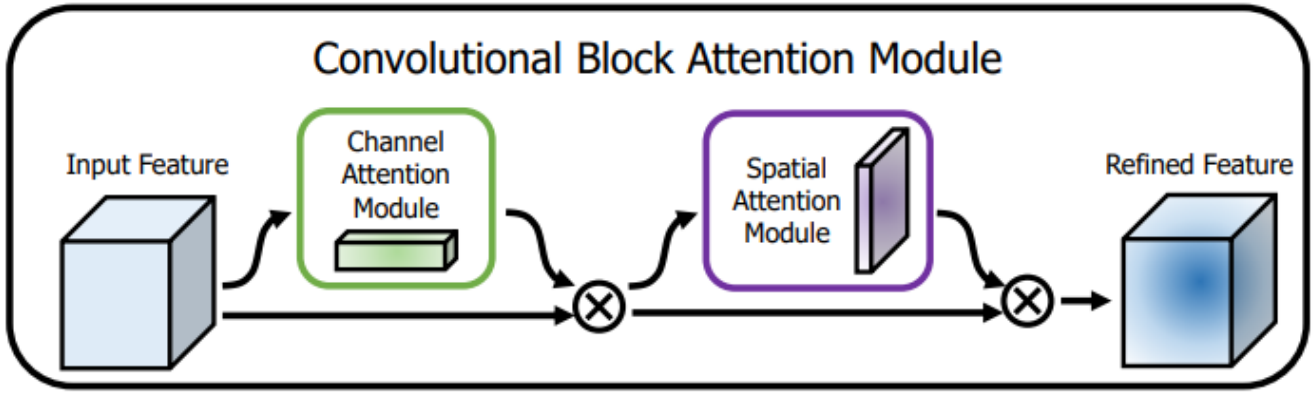


FIGURE 2. Structure detail of YOLOv3.It uses Darknet-53 as the backbone network and uses three scale predictions.

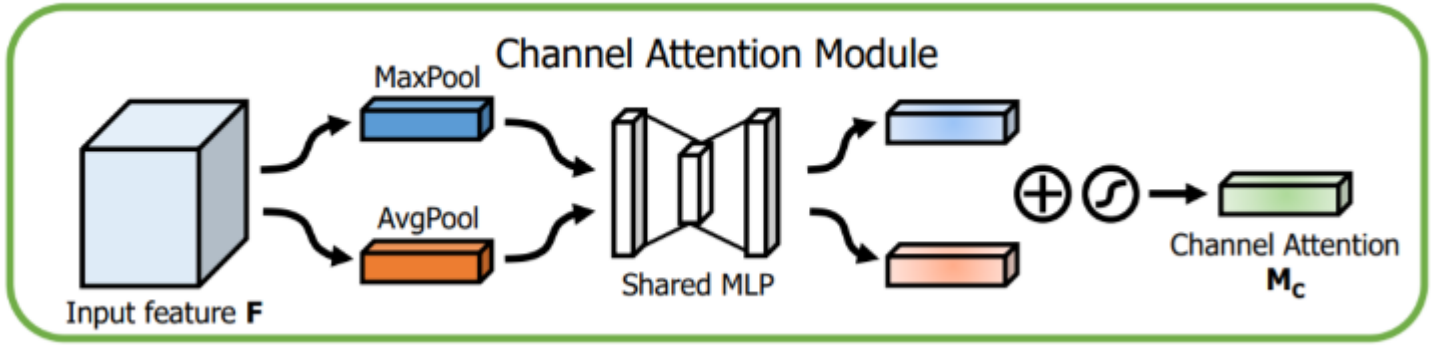
Convolutional Block Attention Module (CBAM)

- **CBAM 개요**
  - CBAM은 Convolutional Neural Networks (CNN)의 성능 향상을 위한 모듈로, 중요 영역 주목을 위한 attention mechanism을 사용
  - Light하면서도, general한 모듈이기에 어떠한 CNN architecture에도 자유롭게 부착 가능
  - 다양한 모델에서 classification이나 detection에서 성능 향상이 검증됨
- **구성 요소**
  - Channel Attention: 각 채널의 중요성을 파악하여 특징을 가중치로 조절
  - Spatial Attention: 이미지의 공간 정보를 활용하여 중요한 영역에 집중



Convolutional Block Attention Module (CBAM)

- **Channel Attention – Focus on 'WHAT'**
  - Channel Attention Module은 Input feature F의 내부 채널 관계를 활용하여 Channel Attention Map 생성
  - AVG Pooling과 Max pooling을 통해 인코딩 진행
  - 위 두 Pooling을 통해 인코딩 된 2개 벡터들을 MLP(Multi-Layer Perceptron)에 통과시켜 비선형성 부여 (같은 Channel로부터 나온 벡터이기 때문에, 동일한 MLP를 거침)
  - 이후 Sigmoid 함수를 거쳐 확률화 된 값으로 인코딩 진행 M_c(1X1XC)

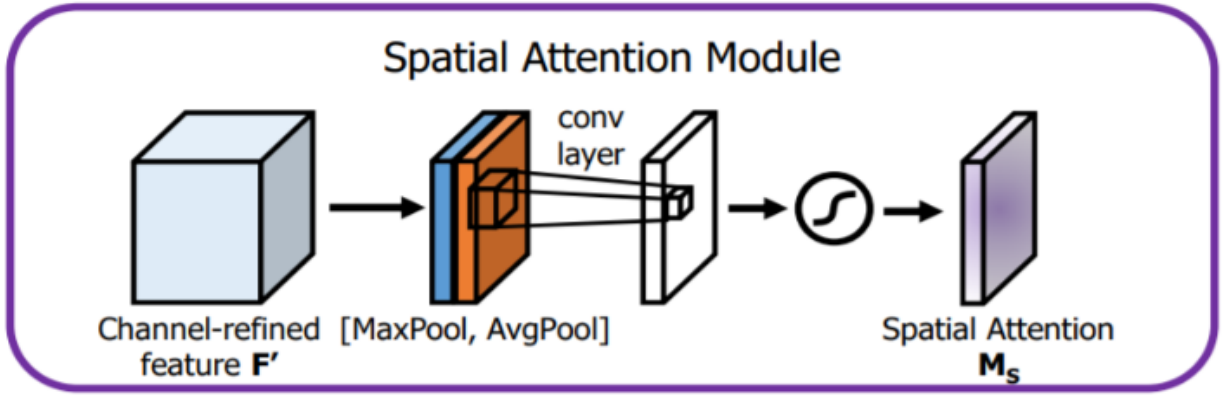


$$M_c = \sigma\left(MLP\left(AvgPool(F)\right) + MLP\left(MaxPool(F)\right)\right)$$
$$= \sigma(W_1\left(W_0\left(F^c_{avg}\right)\right) + W_1(W_0(F^c_{max}))$$

Convolutional Block Attention Module (CBAM)

- **Spatial Attention – Focus on 'WHERE'**
  - Channel Attention Module을 통해 Input feature F에 Mc를 곱해주어 F`를 생성
  - F`로부터 Spatial Attention Map 생성
  - Channel Attention과 마찬가지로, AVG Pooling과 Max Pooling 진행

    (그 후 벡터를 concatenate하여 $H \times W \times 2$ 생성)
  - 7×7 Convolution 진행 후 $M_s(H \times W \times 1)$ 생성



$$M_s(F) = \sigma(f^{7\times7}([AvgPool(F); MaxPool(F)]))$$
$$= \sigma\left(f^{7\times7}([F^s_{avg}; F^s_{max}]\right)$$

Convolutional Block Attention Module (CBAM)

- **성능 및 장점**
  - CBAM은 다양한 CNN 구조에 적용 가능하며, 모델의 복잡성이나 연산량을 크게 증가시키지 않음

예시) CBAM을 ResNet의 ResBlock에 결합할 수 있음

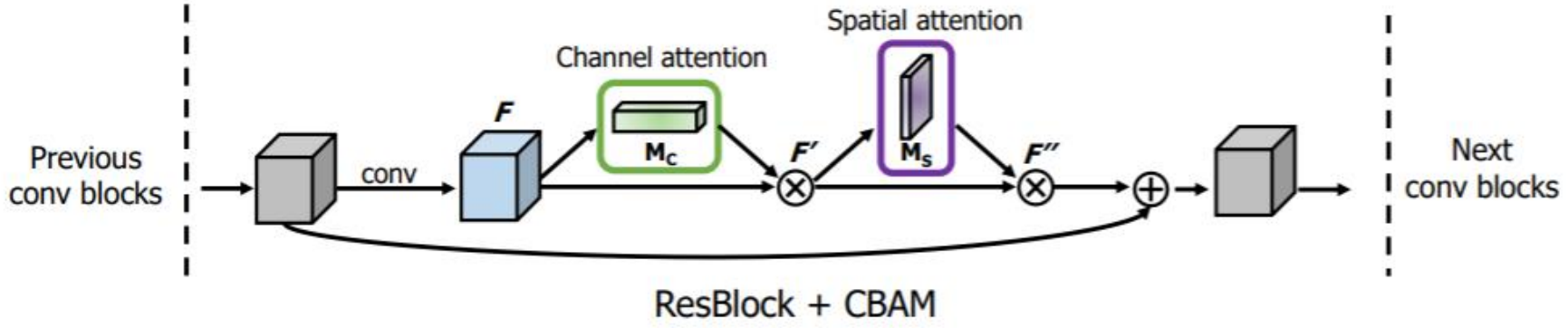
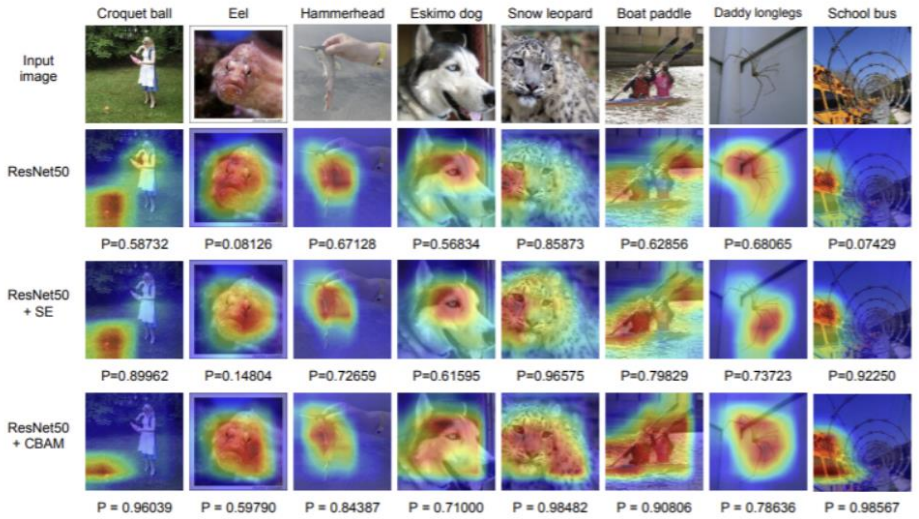


Fig. 3: **CBAM integrated with a ResBlock in ResNet[5].** This figure shows the exact position of our module when integrated within a ResBlock. We apply CBAM on the convolution outputs in each block.

Convolutional Block Attention Module (CBAM)

- **성능 및 장점**
  - CBAM은 여러 벤치마크 데이터셋에서 기존 CNN 모델보다 성능 향상을 보임



Description	Param.	GFLOPs	Top-1 Error(%)	Top-5 Error(%)
ResNet50 + channel (SE [28])	28.09M	3.860	23.14	6.70
ResNet50 + channel	28.09M	3.860	22.80	6.52
ResNet50 + channel + spatial (1x1 conv, k=3)	28.10M	3.862	22.96	6.64
ResNet50 + channel + spatial (1x1 conv, k=7)	28.10M	3.869	22.90	6.47
ResNet50 + channel + spatial (avg&max, k=3)	28.09M	3.863	22.68	6.41
ResNet50 + channel + spatial (avg&max, k=7)	28.09M	3.864	**22.66**	**6.31**

Table 2: **Comparison of different spatial attention methods**. Using the proposed channel-pooling (*i.e.* average- and max-pooling along the channel axis) along with the large kernel size of 7 for the following convolution operation performs best.

Description	Top-1 Error(%)	Top-5 Error(%)
ResNet50 + channel (SE [28])	23.14	6.70
ResNet50 + channel + spatial	**22.66**	**6.31**
ResNet50 + spatial + channel	22.78	6.42
ResNet50 + channel & spatial in parallel	22.95	6.59

Table 3: **Combining methods of channel and spatial attention**. Using both attention is critical while the best-combining strategy (*i.e.* sequential, channel-first) further improves the accuracy.

YOLOv3 모델에 CBAM 적용 사전 사후 성능 검사

- ❑ 기존 object detection에서 주로 사용되는 YOLO 모델에 CBAM을 적용하고자 함
- ❑ COCO dataset을 이용하여 학습 및 검증 (train : val : test = 118k : 5k : 41k images)

- ❑ 실험 1 – Darknet backbone에 CBAM layer 추가
- ❑ 실험 2 – Darknet backbone에 CBAM concatenate

- ❑ Environments
  - Ubuntu 20.04
  - RTX 3060 12GB
  - CUDA 11.7

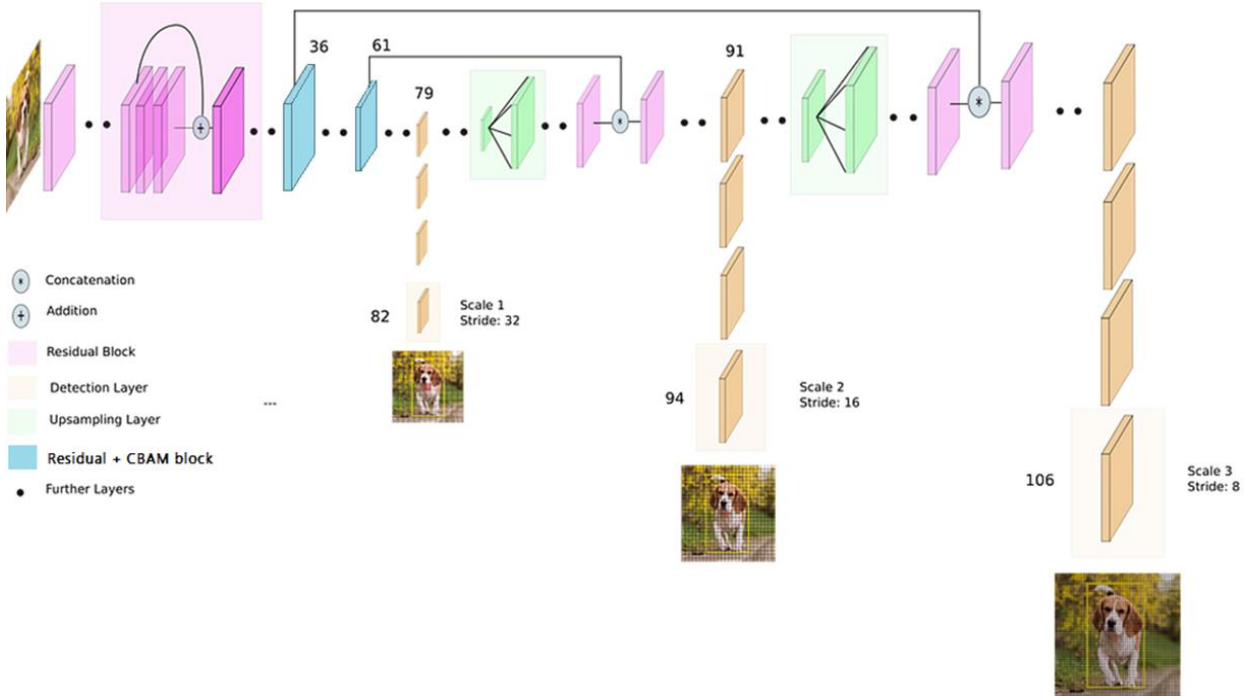
- ❑ Metrics
  - Precision(정밀도) – P(TP/Detection)
  - Recall(재현율) – P(TP/Ground Truth)
  - AP(Average Precision) – Area under P-R curve
  - mAP(mean Average Precision) – Mean AP per class
    - mAP0.5 – IoU threshold = 0.5
    - mAP0.5:0.95 – Average mAP at 10 IoU(0.5~0.95)

실험 1: Darknet Backbone에 CBAM 적용 사전 사후 성능 검사

❑ Darknet53 Backbone에 CBAM 적용한 후 YOLOv3 head에 입력

- Skip connection 분기 전 CBAM 적용

	Original	CBAM
Parameters	61.9M	62.1M
Operations	156.6 GFLOPs	157.0 GFLOPs
Epochs	50	50
Training time	19m08s/epoch	19m36s/epoch
image size	320	320

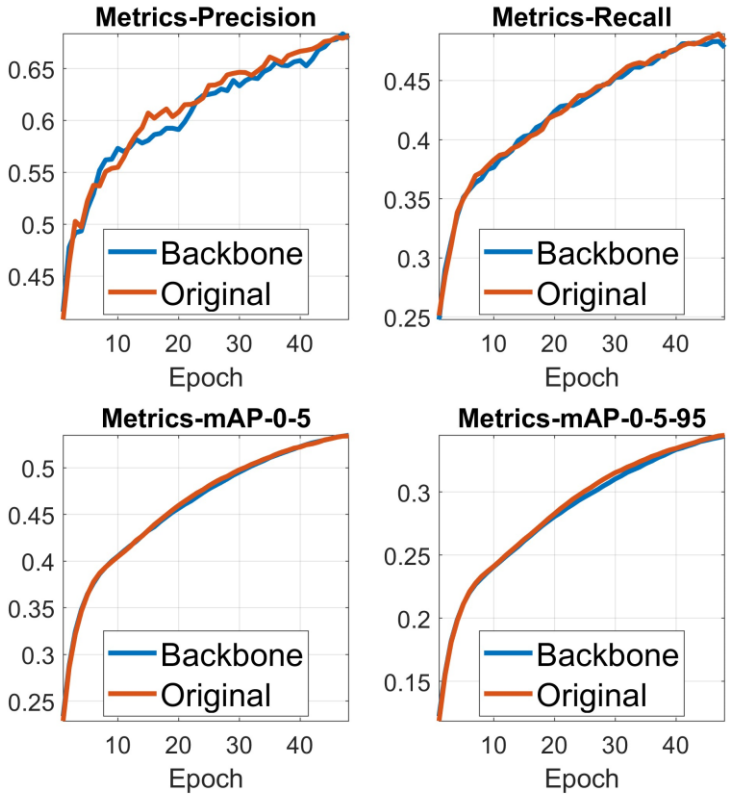


실험 1: Darknet Backbone에 CBAM 적용 사전 사후 성능 검사

❑ Darknet53 Backbone에 CBAM 적용한 후 YOLOv3 head에 입력

	Area:	all	small	medium	large
Average Precision @0.5:0.95	Original	0.345	0.160	0.377	0.501
	CBAM	0.344	0.160	0.375	0.498
Average Recall @0.5:0.95	Original	0.512	0.284	0.569	0.700
	CBAM	0.510	0.286	0.566	0.701

❑ Slightly low performance on every epochs/metrics

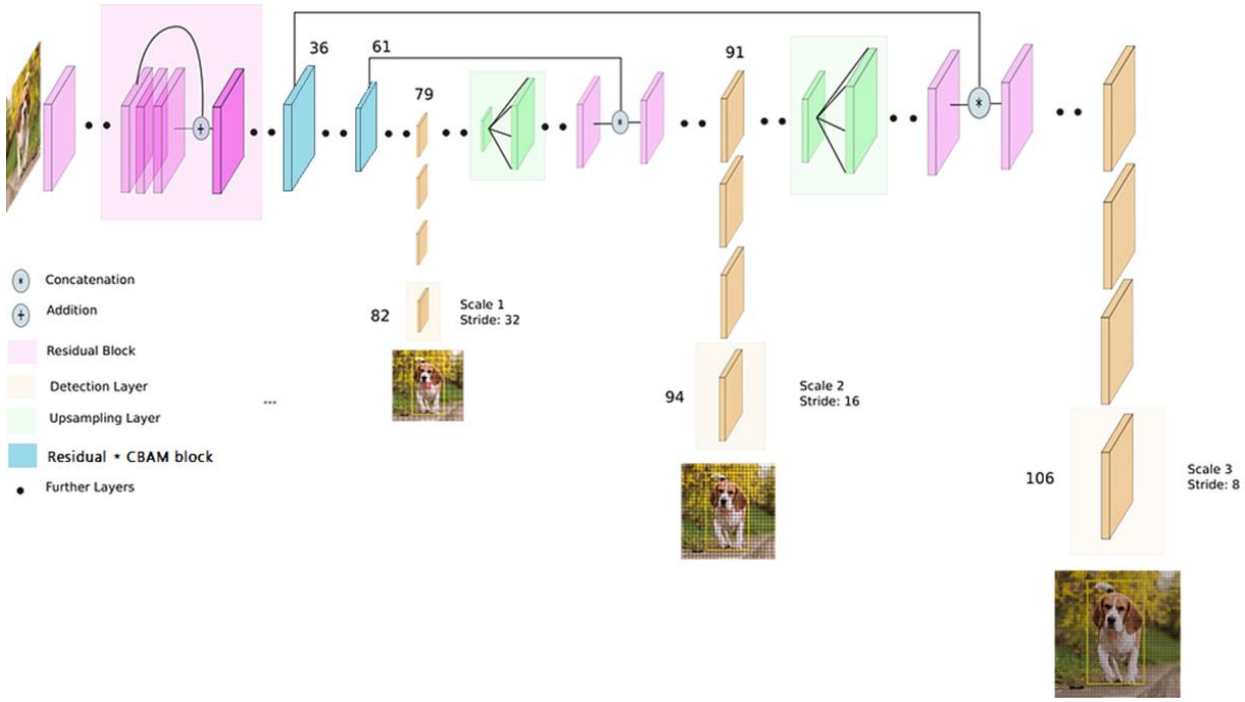


실험 2: Darknet Backbone에 CBAM 적용 사전 사후 성능 검사 - Concat

❑ Darknet53 Backbone에 CBAM 적용한 후 concatenate하여 YOLOv3 head에 입력

- Skip connection 분기 전 CBAM 적용
- 이전 Residual block과 concatenate

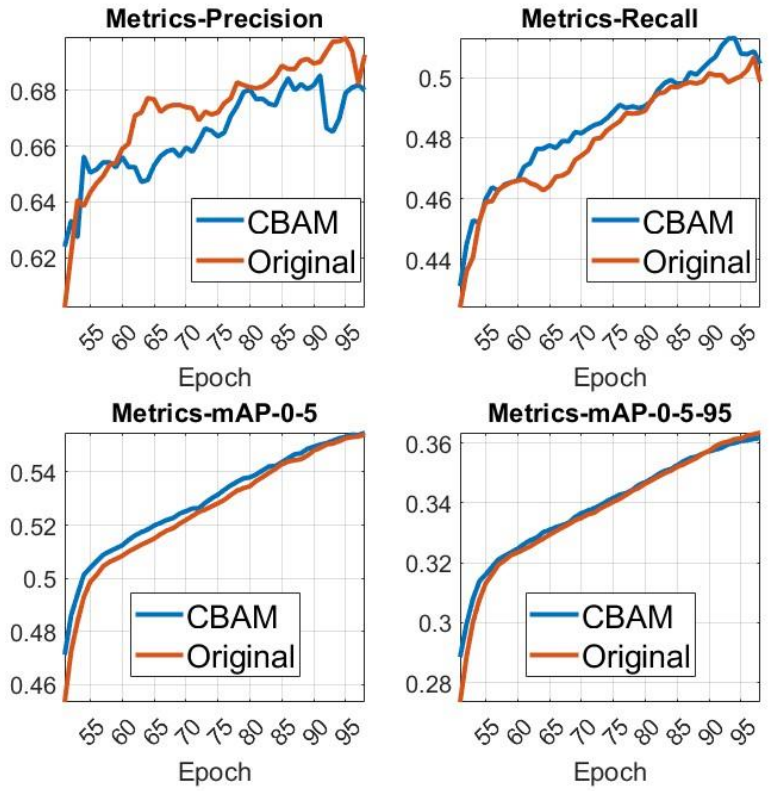
	Original	CBAM
Parameters	61.9M	68.2M
Operations	156.6 GFLOPs	165.4 GFLOPs
Epochs	100	100
Training time	19m08s/epoch	22m08s/epoch
image size	320	320



실험 2: Darknet Backbone에 CBAM 적용 사전 사후 성능 검사 - Concat

❑ Darknet53 Backbone에 CBAM 적용한 후 concatenate하여 YOLOv3 head에 입력

	Area:	all	small	medium	large
Average Precision @0.5:0.95	Original	0.363	0.171	0.401	0.529
	CBAM	0.361	0.175	0.396	0.523
Average Recall @0.5:0.95	Original	0.527	0.302	0.584	0.719
	CBAM	0.527	0.307	0.586	0.718

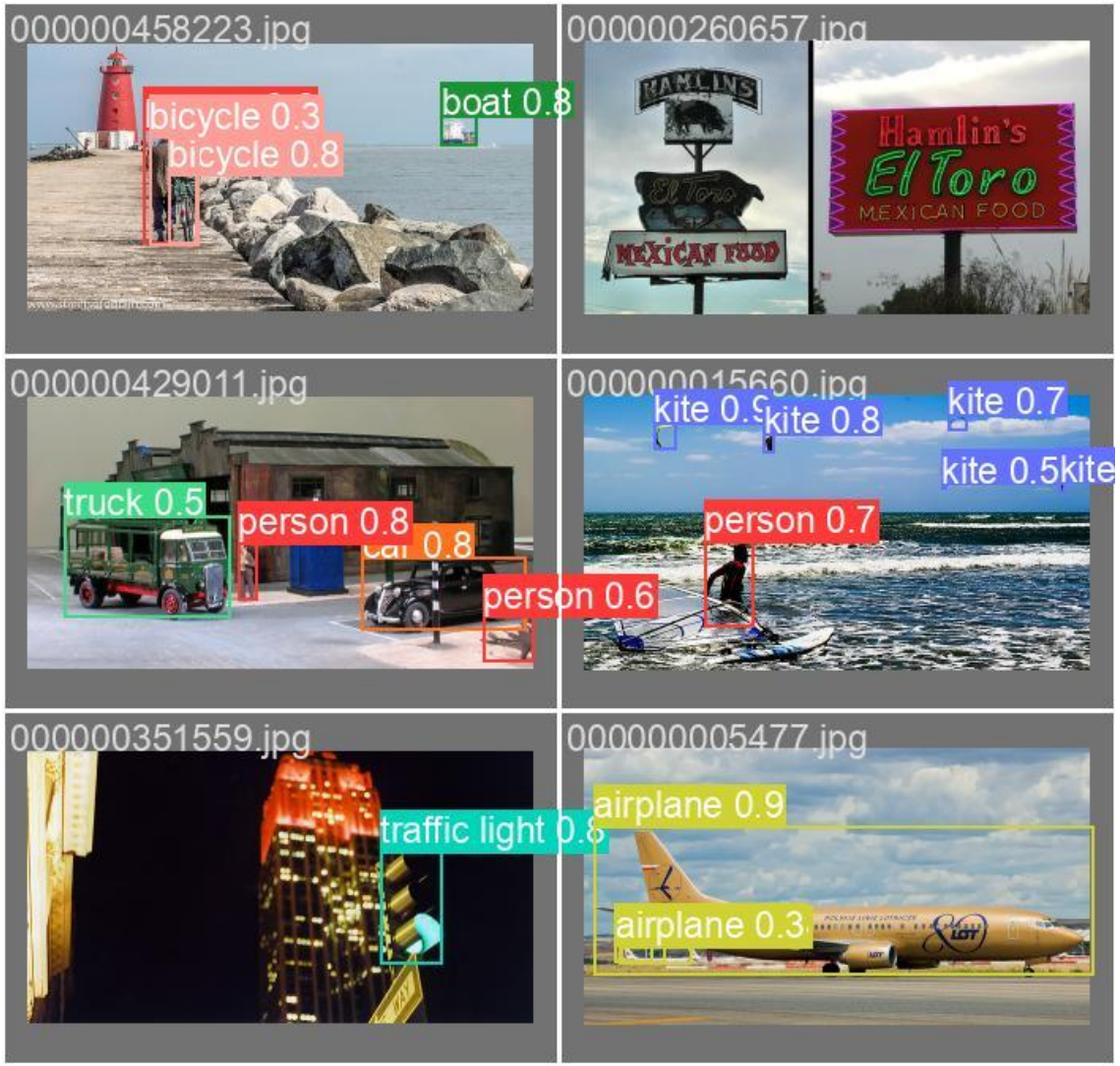


❑ Slightly low Precision, Slightly high Recall

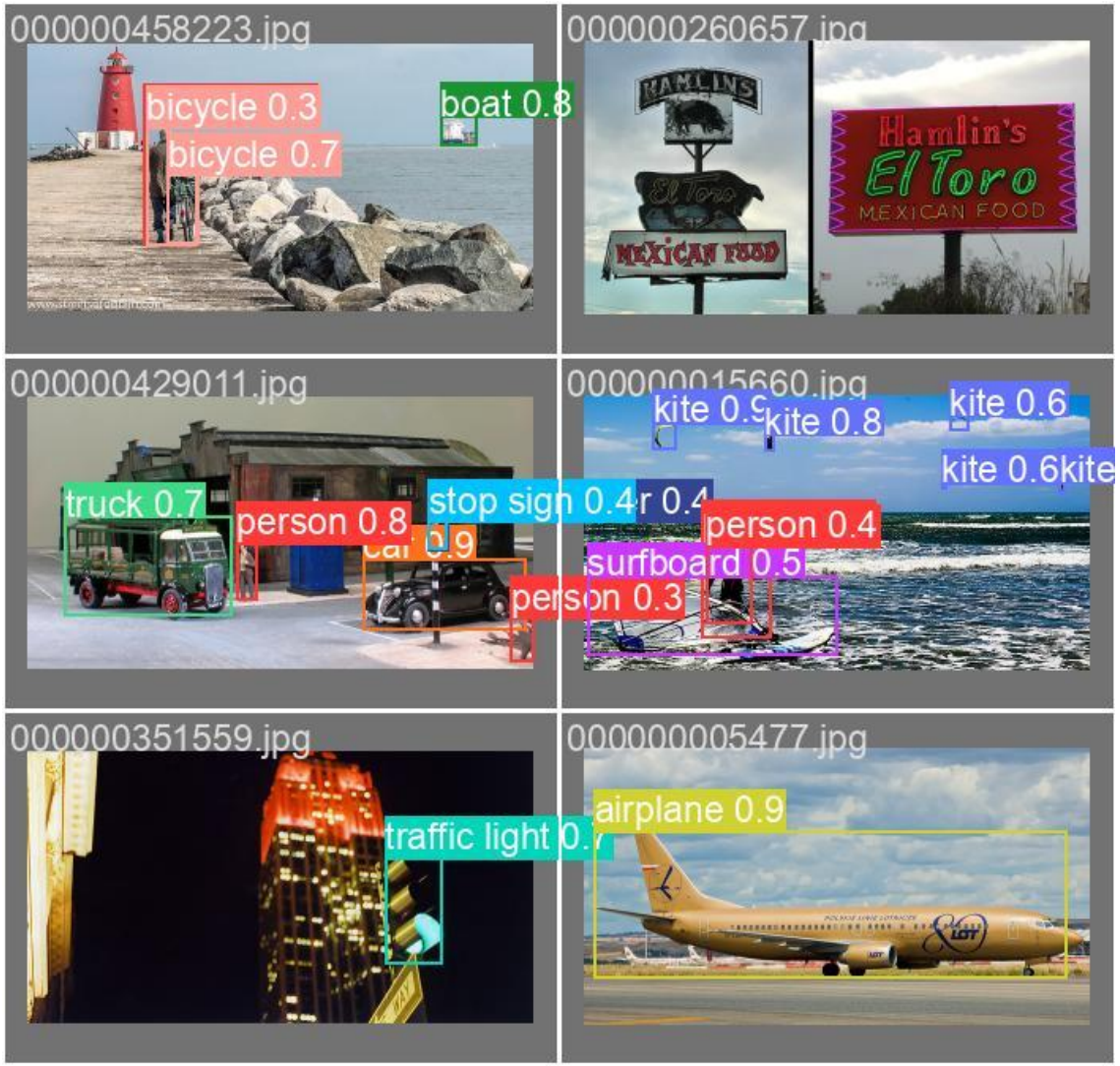
❑ Slightly high on mAP0.5

❑ 0.4%p increased AP & 0.5%p increased AR on small objects

4 Experiments



CBAM



Original

5 Conclusion

❑ YOLOv3에 CBAM 적용

- ➢ 기존의 architecture를 backbone으로 이용하여 architecture customization을 진행
- ➢ Yolo algorithm은 anchor 기반의 알고리즘으로써 anchor의 크기에 따라서 detection 성능이 결정이 된다는 단점이 존재.
- ➢ 이의 극복 방안으로써 CBAM 모듈을 이용하여 small object의 경우에 anchor의 한계를 조금이나마 개선하는 모습을 보여줌
- ➢ Big object의 경우에는 전반적으로 CBAM 모듈의 추가의 여부와 관계 없이 성능이 비슷
- ➢ Cnn과 attention 모두를 이용한 DETR architecture 또한 실험을 진행하였지만, 모델 자체가 무거워, 비교 모델로써 적당하지 않았음.

❑ YOLOv4에는 CBAM의 SAM(Spatial Attention Module)외 여러 다른 모듈이 적용

- ➢ 10%p(AP) 이상의 성능 향상

Reference

YOLOv3: An Incremental Improvement, Redmond et al., ArXiv 2018

https://baozoulin.gitbook.io/neural-networks-and-deep-learning

CBAM: Convolutional Block Attention Module, Woo et al., ECCV 2018

Mini-YOLOv3: Real-Time Object Detector for Embedded Applications, Mao et al., 2019

https://github.com/ultralytics/yolov3

https://github.com/elbuco1/CBAM

Improved YOLOv4 Marine Target Detection Combined with CBAM, Fu et al., Symmetry 2021

A Novel Improved YOLOv3-SC Model for Individual Pig Detection, Hao et al., Sensors 2022

Thank you