

YOLO를 이용하여 객체 탐지를 해보기

DEEP LEARNING AND COMPUTER VISION

09

APPLICATION

ARTIFICIAL INTELLIGENCE

ARTIFICIAL INTELLIGENCE (AI)
REFERS TO THE SIMULATION OF
HUMAN INTELLIGENCE IN MACHINES
AND THE PROCESSING OF DATA
AND INFORMATION BY THESE
MACHINES AND MACHINES
SYSTEMS.

The term may also refer to
the way machines, their systems,
software and/or hardware, learn
from data as learning and
problem-solving.

Artificial intelligence (AI) refers
to the simulation of human
intelligence.

Notice

Artificial intelligence (AI) refers
to the simulation of human
intelligence.

ARTIFICIAL INTELLIGENCE (AI)
REFERS TO THE SIMULATION OF
HUMAN INTELLIGENCE IN MACHINES
AND THE PROCESSING OF DATA
AND INFORMATION BY THESE
MACHINES AND MACHINES
SYSTEMS.

이 교육과정은 교육부 ‘성인학습자 역량 강화 교육콘텐츠 개발’ 사업의 일환으로써
교육부로부터 예산을 지원 받아 고려사이버대학교가 개발하여 운영하고 있습니다.
제공하는 강좌 및 학습에 따르는 모든 산출물의 저작권은 교육부, 한국교육학술정보원,
한국원격대학협의회와 고려사이버대학교가 공동 소유하고 있습니다.

Receptive Field

- 입력 이미지에 3×3의 컨볼루션을 수행합니다.
첫 번째 컨볼루션 레이어의 한 점은 이미지의 3×3영역에 해당합니다.
Feature의 영역에 영향을 미치는 이미지의 영역을 receptive field라고 합니다.
- 두 번째 3×3 컨볼루션 레이어를 추가합니다.
이 때 두 번째 컨볼루션 레이어의 한 점에 해당하는 receptive field의 넓이는 얼마일까요?
레이어를 더 증가시킬 경우에 대해서도 생각해 봅시다.

학습목표

Artificial Intelligence (AI) refers
to the simulation of human

GOALS

Artificial Intelligence has
risen to the position of
human intelligence in many
fields and is expected to have
a great impact on human
society.

The technology will be applied
to many fields and will
have a great impact on human
society.

- 1 완전 연결 레이어를 컨볼루션 레이어로
변경할 수 있다.
- 2 Grid와 Anchor를 이용한
객체 탐지의 원리를 설명할 수 있다.
- 3 YOLO를 이용해 객체탐지를
실행할 수 있다.

1 Grid와 Anchor를 이용한
Object detection

2 YOLO의 상세 구성요소와
EfficientDet 소개

3 객체 탐지 실습

"The more things you know, the further
you are from the truth."
- William Shakespeare, Twelfth Night
- "The more things you know, the further
you are from the truth."
- William Shakespeare, Twelfth Night

CONTENTS

학습내용

Artificial Intelligence (AI) refers
to the simulation of human

1차시

Grid와 Anchor를 이용한
Object detection

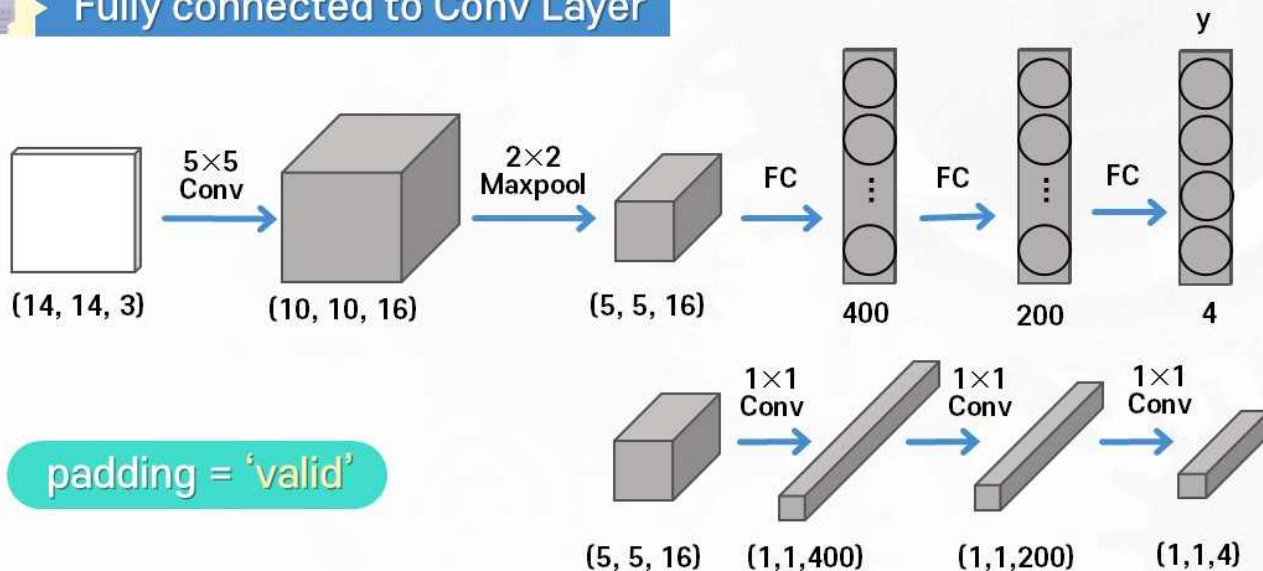
YOLO의 아이디어들



01 Grid와 Anchor를 이용한 Object detection

01 FC to Conv

Fully connected to Conv Layer



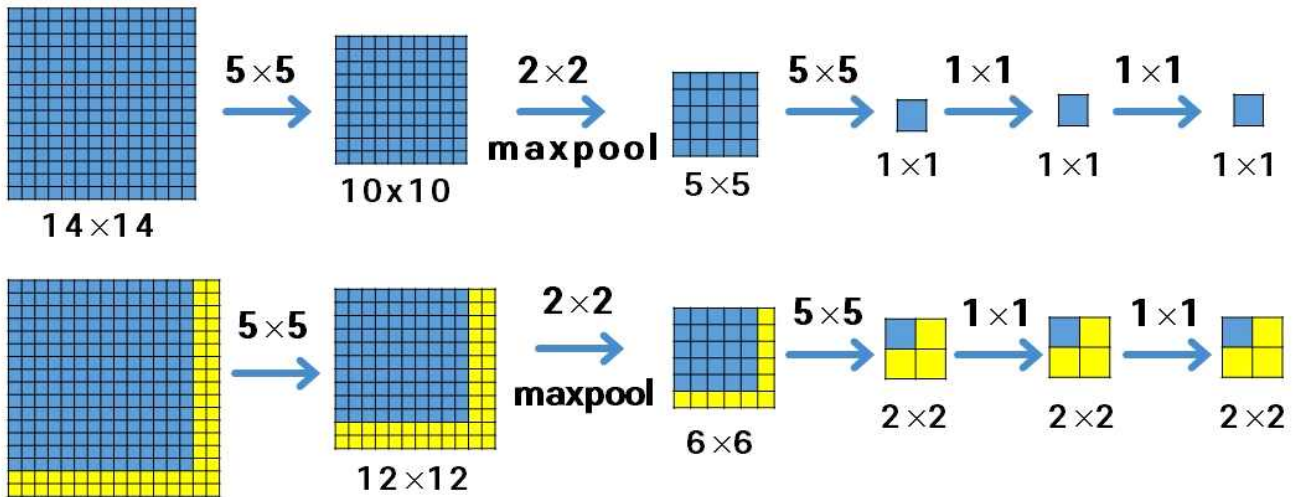
01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) starts
in the simulation of human

01 FC to Conv



다른 크기의 입력에 대한 Conv Layer



01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) starts
in the simulation of human

01 FC to Conv



다른 크기의 입력에 대한 Conv Layer

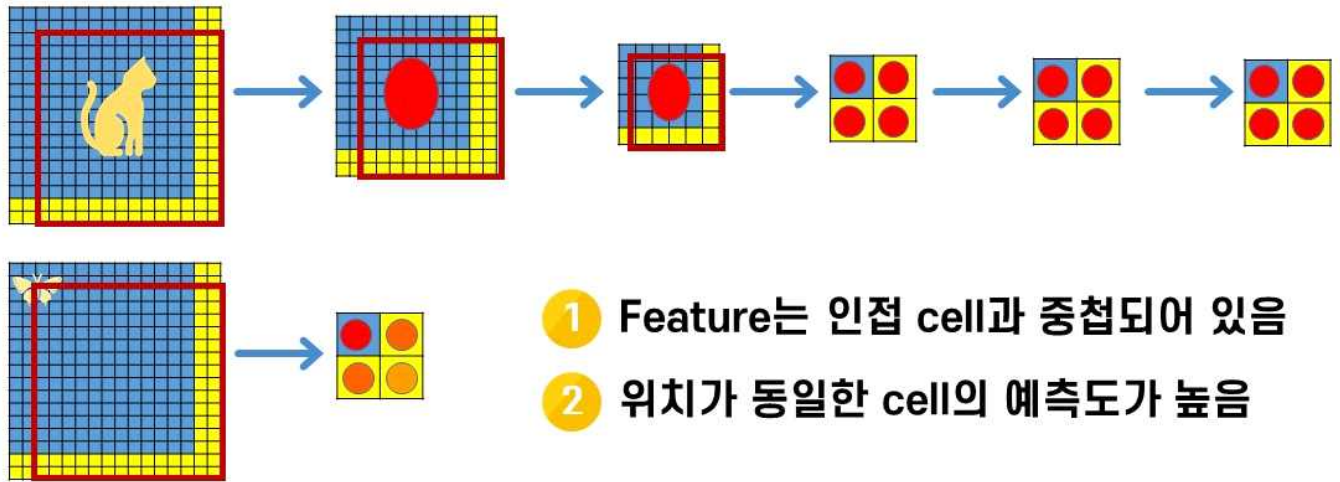
- 1 두 네트워크는 동일함 (파라미터 수 동일)
- 2 입력, 출력의 width, height만 다름

01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) refers to the simulation of human intelligence in the computers.

01 FC to Conv

Feature is overlapped

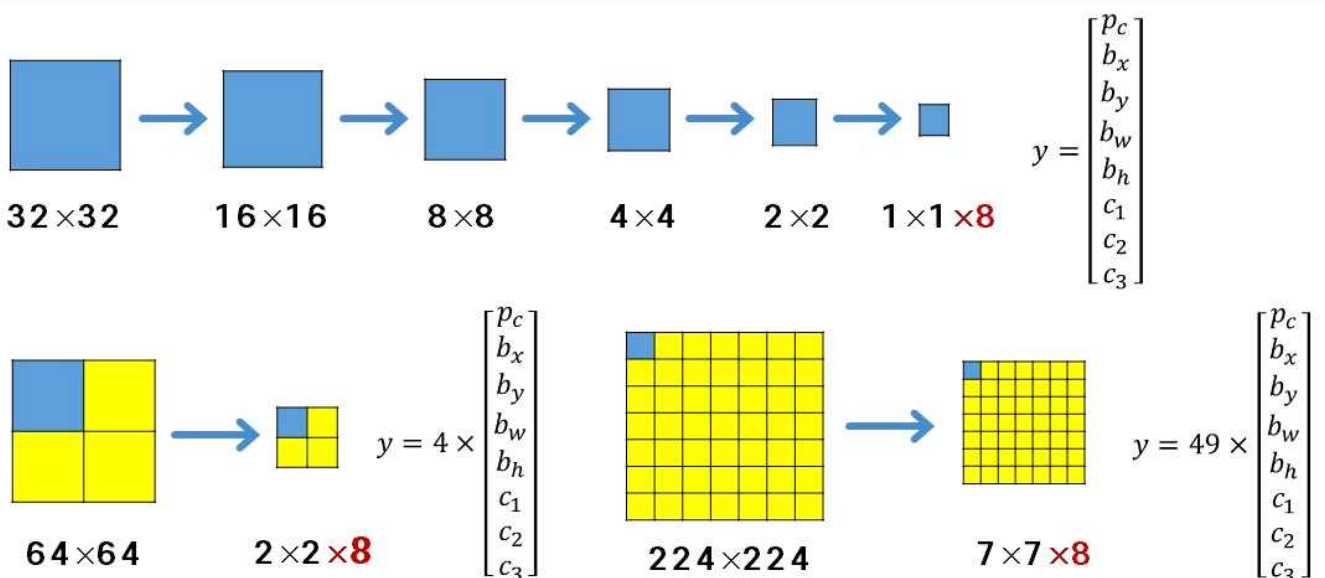


01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) refers to the simulation of human intelligence in the computers.

02 Grid

YOLO grid

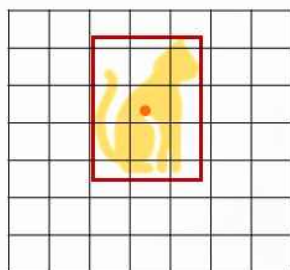


01 Grid와 Anchor를 이용한 Object detection

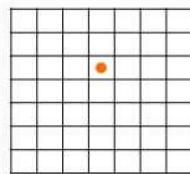
Artificial Intelligence (AI) refers to the simulation of human

02 Grid

▶ 학습 시 cell의 지정

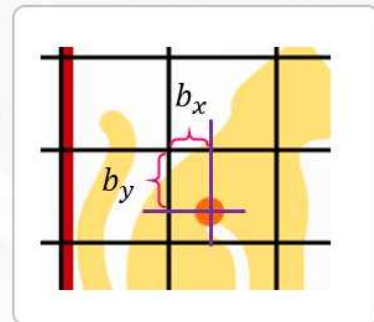


224×224



7×7×8

$$y = S \times S \times \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$



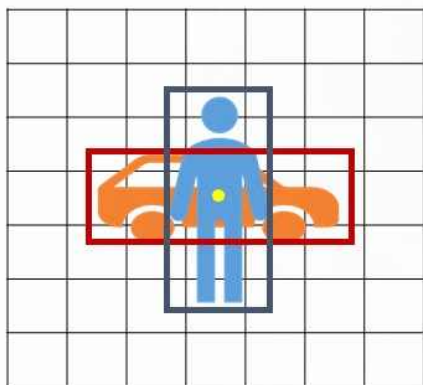
- ▶ 학습 시 center의 위치에 해당하는 cell에 결과를 입력
추론 시에는 학습된 cell 쪽에 결과가 도출될 확률이 높아짐
- ▶ center의 좌표값 b_x , b_y 는 Grid 좌상단 좌표의 offset으로 입력
Grid내의 cell의 위치에 상대적으로 학습하게 됨으로서, 이동 대칭성 확보

01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) refers to the simulation of human

03 Anchor

▶ 한 cell 내의 복수 객체의 인식



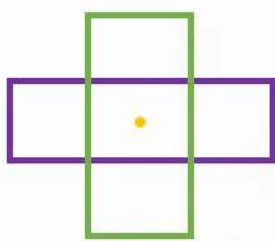
- ▶ 한 cell의 출력 결과가 1개일 경우, 복수의 중첩된 객체의 감지가 불가능
 - ▶ **Anchor box**를 도입하여, 복수의 출력을 가능하게 함

01 Grid와 Anchor를 이용한 Object detection

Artificial intelligence (AI) refers to the simulation of human

03 Anchor

Anchor Box (Prior box)



Anchor 1



Anchor 2

$$y = 2 \times \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \end{bmatrix}$$

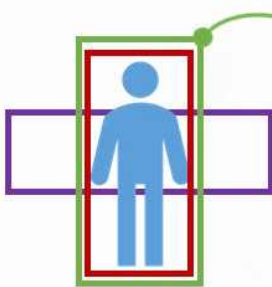
- ❖ 미리 전체 이미지의 크기에 따라 object가 많은 형태의 Anchor를 선정
이 예제에서는 Anchor box를 총 2개 설정
- ❖ Anchor box의 수 만큼 결과를 증가
8 개의 결과x2 만큼 한 cell의 출력 결과의 텐서 수가 증가

01 Grid와 Anchor를 이용한 Object detection

Artificial intelligence (AI) refers to the simulation of human

03 Anchor

Anchor Box를 이용한 학습



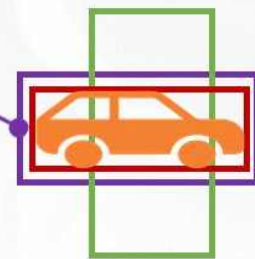
Anchor 1



Anchor 2



Ground truth



- ❖ Ground truth와 Anchor의 IoU가
최대인 Anchor에 해당하는 곳으로 학습 데이터를 입력

사람인 경우 Anchor 1쪽이 IoU가 큼 → 첫 번째 결과쪽으로 입력

자동차인 경우 Anchor 2쪽이 IoU가 큼 → 두 번째 결과쪽으로 입력

01 Grid와 Anchor를 이용한 Object detection

Artificial intelligence (AI) refers to the simulation of human intelligence in the computers.

03 Anchor



Anchor Box에 따른 출력 텐서의 수

$$y = 7 \times 7 \times 2 \times \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \end{bmatrix} \quad y = S \times S \times A \times \begin{bmatrix} p_c \\ b_x \\ b_y \\ b_w \\ b_h \\ c_1 \\ c_2 \\ c_3 \\ \dots \end{bmatrix}$$

❖ 총 출력텐서의 수 grid의 cell수($S \times S$) \times Anchor 수(A) \times (5+클래스 수)

cell = 7×7 , Anchor = 2개, class = 3인 경우 $7 \times 7 \times 2 \times (5+3) = 784$

cell = 19×19 , Anchor = 9개, class = 80인 경우 $19 \times 19 \times 9 \times (5+80) = 276165$

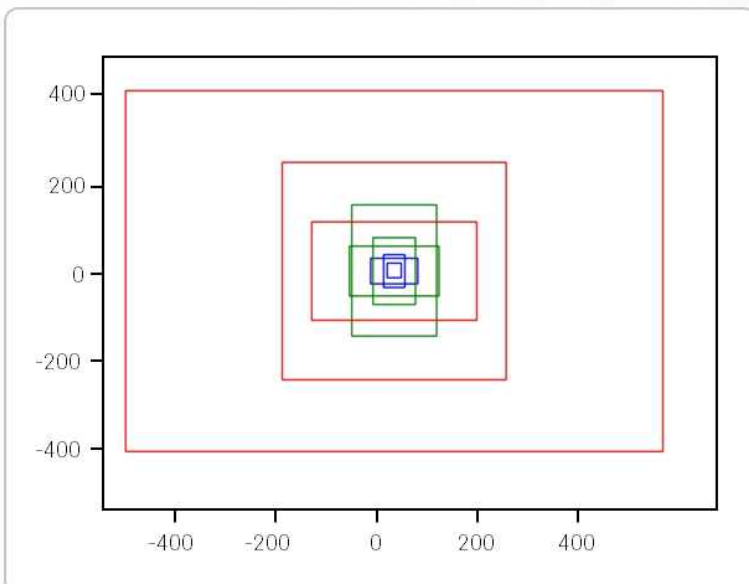
01 Grid와 Anchor를 이용한 Object detection

Artificial intelligence (AI) refers to the simulation of human intelligence in the computers.

03 Anchor



Default anchor boxes



❖ K-means clustering를 통하여 사전에 분류

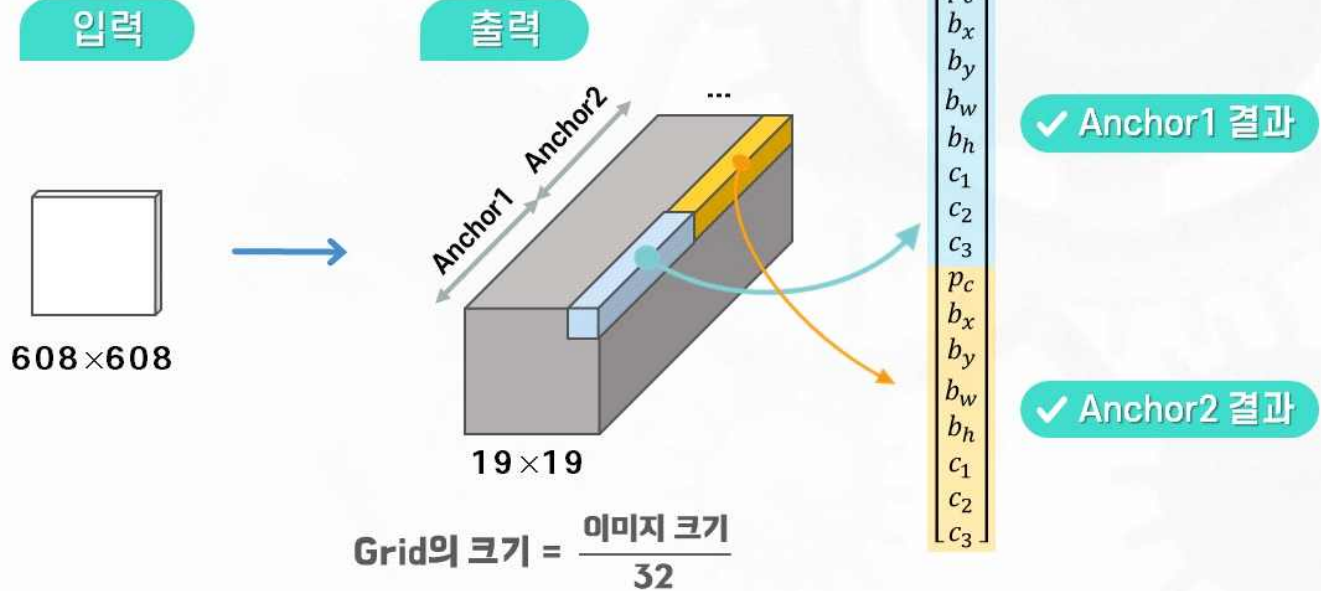
- 통상적인 사용은 이것으로도 **OK!**
- custom한 객체를 학습할 때, 학습데이터의 bounding box로부터 간단히 계산 가능
- Small, medium, large 크기로 3개씩

01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) starts
in the context of human

03 Anchor

출력 텐서 시각화



01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) starts
in the context of human

04 YOLO

YOLO – the algorithm



01 Grid와 Anchor를 이용한 Object detection

Artificial intelligence (AI) relies
on the assistance of humans

04 YOLO

YOLO – the algorithm

Full ConNet

이미지부터, 분류 및
Boudning box regression까지
전체 네트워크를 CNN으로
구성하여 학습

Grid와 Anchor

Non Max Suppression

01 Grid와 Anchor를 이용한 Object detection

Artificial intelligence (AI) relies
on the assistance of humans

04 YOLO

YOLO – the algorithm

Full ConNet

Grid와 Anchor

Non Max Suppression

한 번의 CNN으로
추론 도출
Grid와 cell별 anchor로
분류 결과 및
bounding box 결과를 출력

01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) refers to the simulation of human

04 YOLO

YOLO – the algorithm

Full ConNet



Grid와 Anchor



Non Max Suppression

최종 출력 텐서에
NMS를 적용하여
중복된 box들 중 높은
confidence score만 선택

01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) refers to the simulation of human

04 YOLO

Grid별 출력 및 NMS를 통한 결과 출력

Joseph Redmon et al You Only Look Once: Unified, Real-Time Object Detection [2016] arXiv:1506.02640



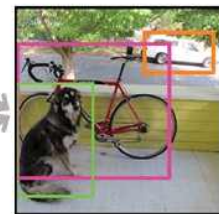
SxS grid on input



Bounding boxes + confidence



Class probability map



Final detections

01 Grid와 Anchor를 이용한 Object detection

Artificial Intelligence (AI) refers to the simulation of human

04 YOLO

Grid별 출력 및 NMS를 통한 결과 출력

Joseph Redmon et al You Only Look Once: Unified, Real-Time Object Detection [2016] arXiv:1506.02640



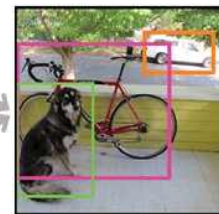
SxS grid on input



Bounding boxes + confidence



Class probability map



Final detections

01 Grid와 Anchor를 이용한 Object detection

Artificial intelligence (AI) relies
on the assistance of humans

04 YOLO

YOLO – You only look once (2016~2019)

- 1 2016년 YOLO 발표 (J. Redmon)**
높은 성능과 빠른 속도로 주목받음
v2, 9000 (2016) | v3 (2018)
- 2 2020년 v4 (A. Bochkovskiy)**
Single GPU에서 실시간 감지 가능
- 3 선행연구의 핵심 기술요소들 반영**
독창적이면서도, 다른 선행 연구들의
장점을 지속적으로 반영 중

01 Grid와 Anchor를 이용한 Object detection

Artificial intelligence (AI) relies
on the assistance of humans

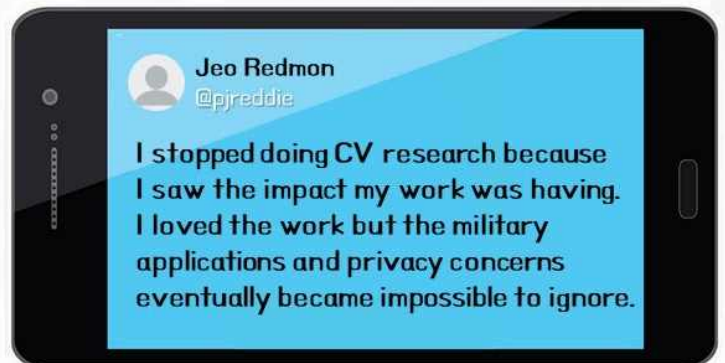
04 YOLO

YOLO – You only look once (2016~2019)



“어렵고 다른 사람들이
대신 해 줄 것이기 때문에,
우리의 일에 대한
사회적 영향에 대해
생각할 필요가 없다는 것은
정말 잘못된 논리이다.”

- Joseph Redmon



본문 출처 : Joseph Redmon 트위터 계정

2차시

YOLO의 상세 구성요소와
EfficientDet 소개

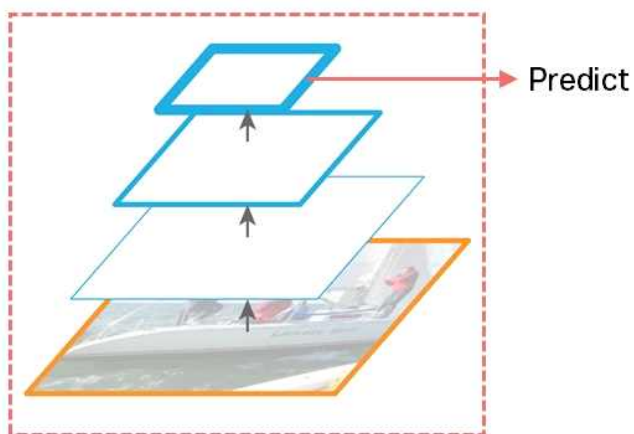
The art of object detection



02 YOLO의 상세 구성요소와 EfficientDet 소개

01 Backbone

Backbone network



Backbone ConvNet

이미지 출처 : Tsung-Yi Lin et.al
Pyramid Networks for Object Detection (2017) arXiv:1612.03144

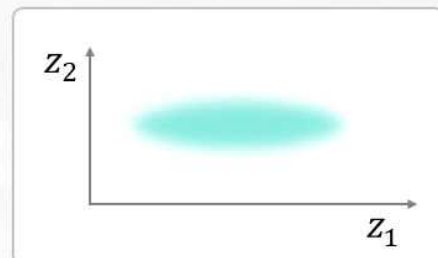
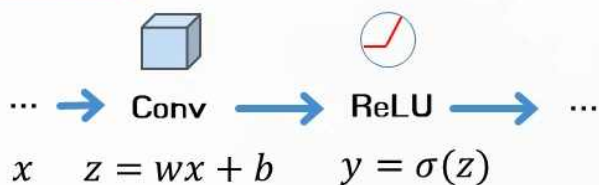
Backbone이란?

- CNN classifier에서 최종 FC레이어를 제외한 특징 추출 부분
- Object detection에서는 여러 가지 backbone 활용 가능
- 성능과 속도를 고려하여 적합한 네트워크를 선정
- YOLOv4의 경우에는 CSPDarknet52를 주로 사용

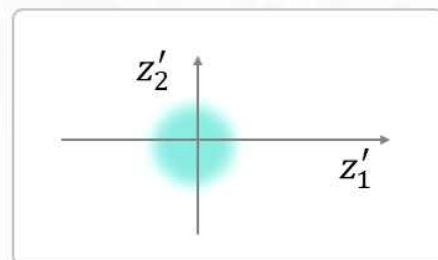
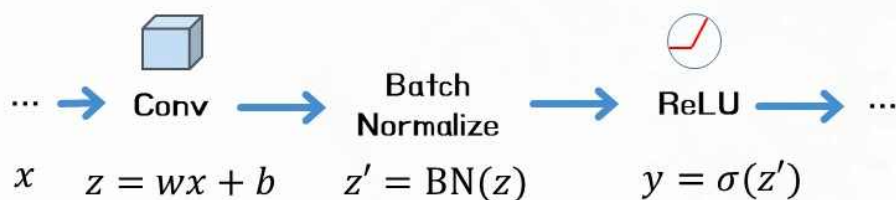
01 Backbone

Batch normalization이란? – 레이어의 출력값에 대한 정규화

기존



Batch Normalization



01 Backbone

Batch normalization

$$\mu = \frac{1}{n} \sum z_i$$

학습 데이터의 평균

$$\sigma^2 = \frac{1}{n} \sum (z_i - \mu)^2$$

학습 데이터의 분산

$$\hat{z} = \frac{z - \mu}{\sqrt{\sigma^2}}$$

정규화

$$\text{BN}(z) \equiv \gamma \hat{z} + \beta$$

Scale and shift
(선형 변환)

02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) relies
on the collection of human

01 Backbone

Batch normalization

1 학습 시 평균 및 분산 계산

- 학습 데이터 batch에서 계산
- 실행(inference)시에는 학습되어 있는 평균과 분산을 이용
- **non-trainable param = channel 수 × 2**

2 γ 와 β 는 학습 파라미터

- Conv channel별로 공통 파라미터
- **params = channel수 × 2**

3 효과

- 학습 속도 및 성능 향상
- Overfitting 방지

02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) relies
on the collection of human

01 Backbone

Batch normalization

$$\mu = \frac{1}{n} \sum z_i$$

학습 데이터의 평균

$$\sigma^2 = \frac{1}{n} \sum (z_i - \mu)^2$$

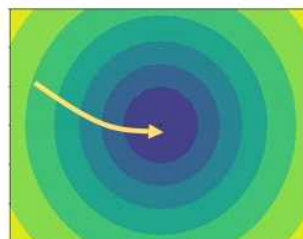
학습 데이터의 분산

$$\hat{z} = \frac{z - \mu}{\sqrt{\sigma^2}}$$

정규화

$$\text{BN}(z) \equiv \gamma \hat{z} + \beta$$

Scale and shift
(선형 변환)



배치 정규화에 따른
손실함수의 변화

02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) starts
in the simulation of human

01 Backbone

Batch normalization code

```
from tensorflow.keras.layers import Conv2D
from tensorflow.keras.layers import BatchNormalization, Activation

# 기존
Conv2D(64, (3,3), activation='relu'),

# 기존과 동일 코드
Conv2D(64, (3,3)),
Activation('relu'),

# Batch Normalization
Conv2D(64, (3,3)),
BatchNormalization(),
Activation('relu'),
```

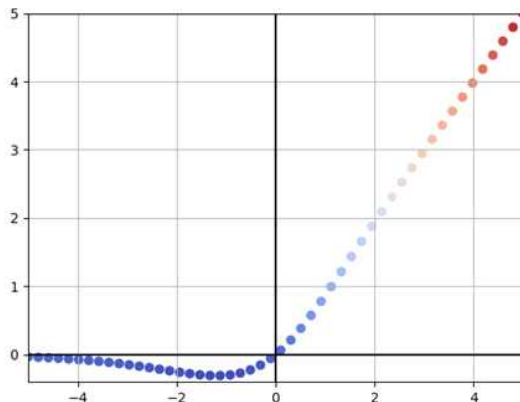
02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) starts
in the simulation of human

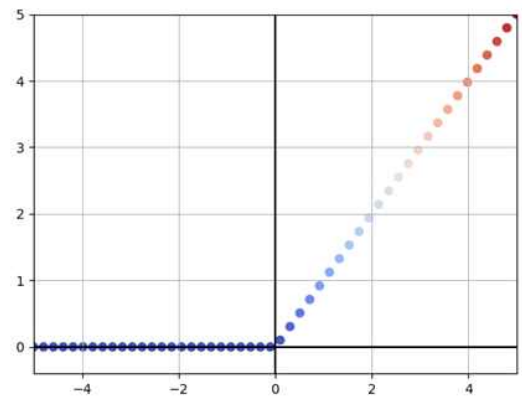
01 Backbone

Mish activation function (2020)

Mish



ReLU



성능 ↑
속도 ↓

$$\sigma(x) \equiv x \tanh(\text{softplus}(x)) = x \tanh(\log(1 + e^x))$$

02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) starts
in the collection of human

01 Backbone

Mish Keras implementation

```
from tensorflow.keras.layers import Layer
from tensorflow.keras import backend as K

class Mish(Layer):
    def __init__(self, **kwargs):
        super(Mish, self).__init__(**kwargs)
        self.supports_masking = True

    def call(self, inputs):
        return inputs * K.tanh(K.softplus(inputs))

    def get_config(self):
        config = super(Mish, self).get_config()
        return config

    def compute_output_shape(self, input_shape):
        return input_shape
```

02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) starts
in the collection of human

01 Backbone

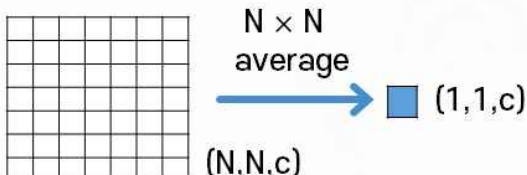
darknet backbone

1 YOLO v1 – VGG-19

2 YOLO v2 – Darknet-19

- vgg-19와 유사
- 1x1 conv를 활용하여 속도와 성능 개선
- Global average pooling 사용

Global average pooling



Type	Filters	Size/Stride	Output
Convolutional	32	3×3	224×224
Maxpool		$2 \times 2/2$	112×112
Convolutional	64	3×3	112×112
Maxpool		$2 \times 2/2$	56×56
Convolutional	128	3×3	56×56
Convolutional	64	1×1	56×56
Convolutional	128	3×3	56×56
Maxpool		$2 \times 2/2$	28×28
Convolutional	256	3×3	28×28
Convolutional	128	1×1	28×28
Convolutional	256	3×3	28×28
Maxpool		$2 \times 2/2$	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Convolutional	256	1×1	14×14
Convolutional	512	3×3	14×14
Maxpool		$2 \times 2/2$	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	512	1×1	7×7
Convolutional	1024	3×3	7×7
Convolutional	1000	1×1	7×7
Avgpool		Global	1000
Softmax			

이미지 출처 : J. Redmon
YOLO9000: Better, Faster, Stronger [2016] arXiv:1612.08242

02 YOLO의 상세 구성요소와 EfficientDet 소개

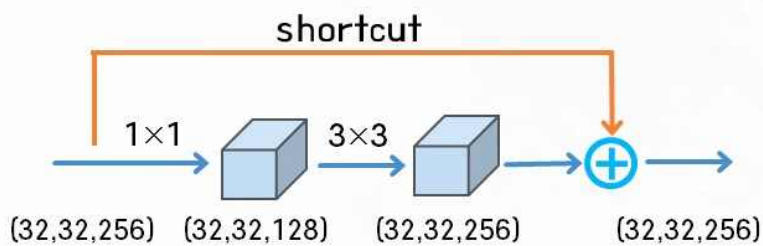
01 Backbone

darknet backbone

3 YOLO v3 – Darknet-53

- resnet과 비슷한 skip connection 도입
- layer수 증가에 따른 성능 향상

darknet Resblock



Type	Filters	Size	Output
Convolutional	32	3 × 3	256 × 256
Convolutional	64	3 × 3 / 2	128 × 128
Convolutional	32	1 × 1	
1x Convolutional	64	3 × 3	
Residual			128 × 128
Convolutional	128	3 × 3 / 2	64 × 64
Convolutional	64	1 × 1	
2x Convolutional	128	3 × 3	
Residual			64 × 64
Convolutional	256	3 × 3 / 2	32 × 32
Convolutional	128	1 × 1	
8x Convolutional	256	3 × 3	
Residual			32 × 32
Convolutional	512	3 × 3 / 2	16 × 16
Convolutional	256	1 × 1	
8x Convolutional	512	3 × 3	
Residual			16 × 16
Convolutional	1024	3 × 3 / 2	8 × 8
Convolutional	512	1 × 1	
4x Convolutional	1024	3 × 3	
Residual			8 × 8
Avgpool		Global	
Connected		1000	
Softmax			

이미지 출처 : <https://arxiv.org/pdf/1804.02767.pdf>

02 YOLO의 상세 구성요소와 EfficientDet 소개

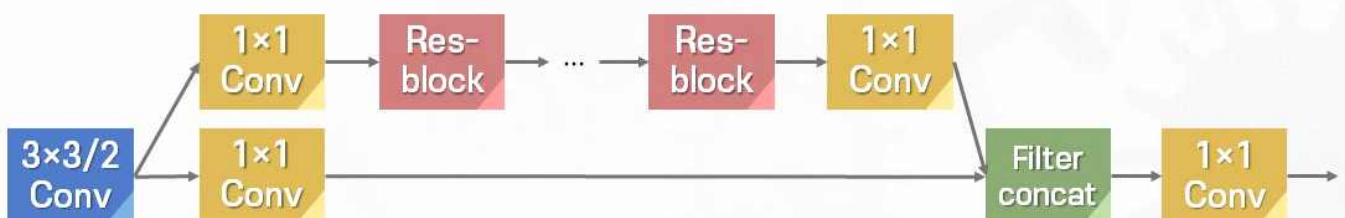
01 Backbone

darknet backbone

4 YOLO v4 – CSPdarknet-53

- Cross stage partial network 도입
- layer수 증가에 따른 성능 향상

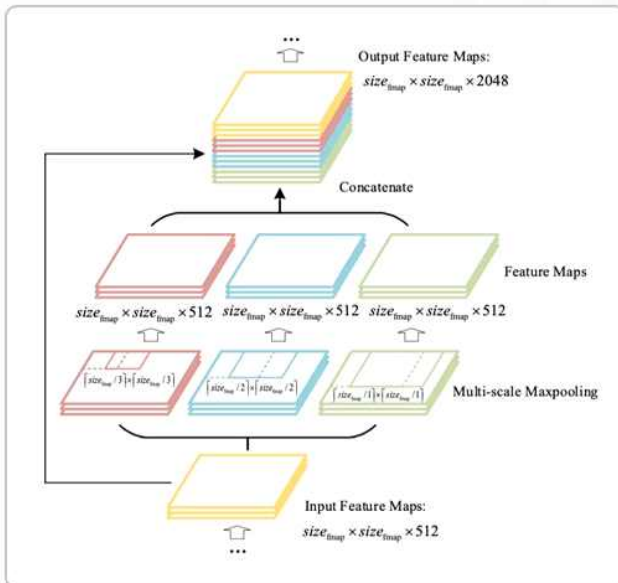
Cross Stage Partial Network



02 Neck



SPP (Spatial Pyramid Pooling)



이미지 출처 : <https://arxiv.org/ftp/arxiv/papers/1903/1903.08589.pdf>

1 NIN(network in network)과 유사

- Inception module처럼 다양한 크기 도입
- 13×13 , 9×9 , 5×5 , 1×1 maxpooling (not the Conv)
- stride=1 같은 $W \times H$ 출력
- backbone 다음 레이어로 사용

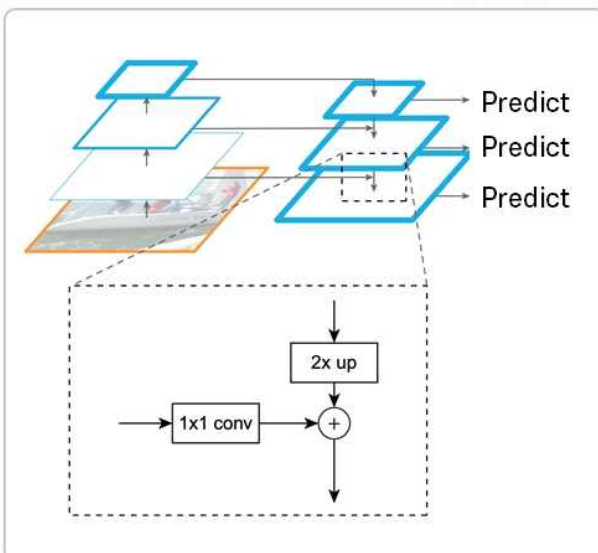
2 SPP의 효과

- 여러 scale의 feature를 detection에 전달
- 보다 다양한 크기의 감지에 유용
- COCO 기준 2.7% 성능(AP50) 향상

02 Neck



Feature Pyramid Network (FPN)



이미지 출처 : Tsung-Yi Lin et.al
Feature Pyramid Networks for Object Detection [2017] arXiv:1612.03144

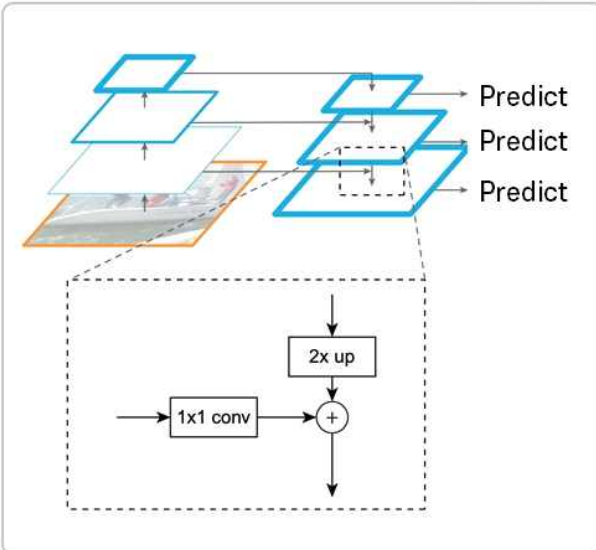
1 FPN이란?

backbone의
최종레이어 및 사이즈 변경 직전의
feature layer들을 가지고
복수의 detection 수행

02 Neck



Feature Pyramid Network (FPN)



이미지 출처 : Tsung-Yi Lin et.al
Feature Pyramid Networks for Object Detection [2017] arXiv:1612.03144

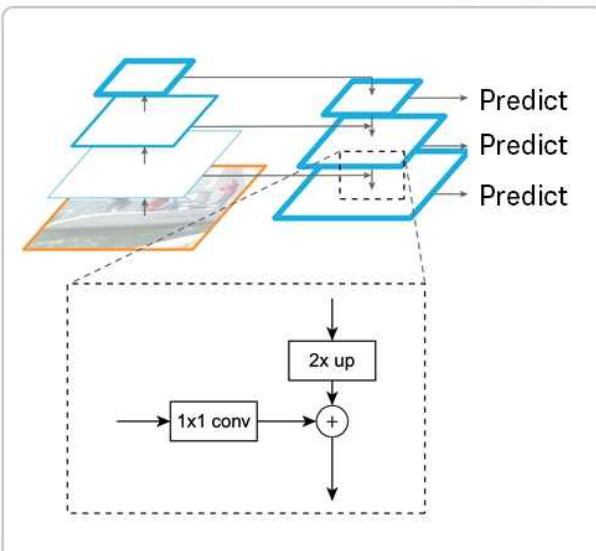
2 레이어 별로 다른 크기의 감지

- 최종 레이어 :
모든 정제된 feature + **대략적** 위치
- 중간 레이어 :
덜 정제된 feature + **상세** 위치

02 Neck



Feature Pyramid Network (FPN)



이미지 출처 : Tsung-Yi Lin et.al
Feature Pyramid Networks for Object Detection [2017] arXiv:1612.03144

3 중간 레이어의 feature를 보강

상위 레이어의 feature를
up-sample하여 더하기
(YOLO의 경우 : **concatenate**)

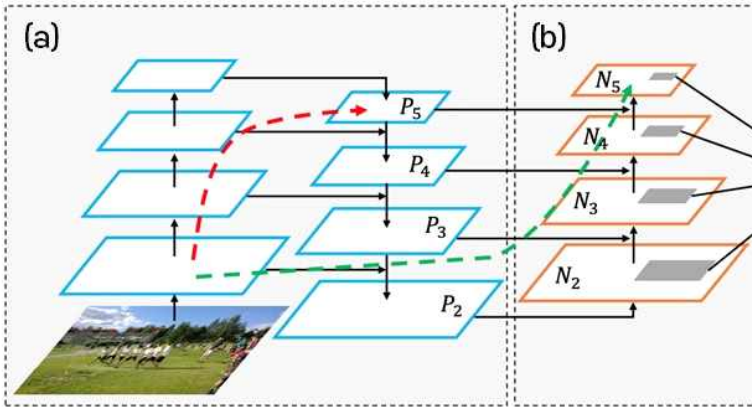
02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) relies on the collection of human

02 Neck

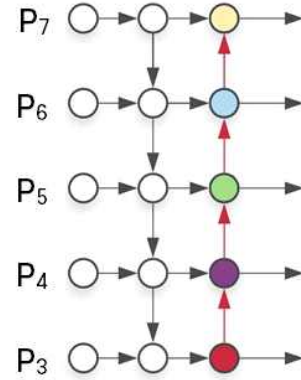


Path Aggregation Network



- FPN의 Layer들을 다시 역방향으로 더하거나 concatenate(YOLO의 경우)하여 최종 head로 전달

이미지 출처 : <https://arxiv.org/pdf/1803.01534.pdf>, <https://arxiv.org/pdf/1911.09070.pdf>



02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) relies on the collection of human

03 Loss



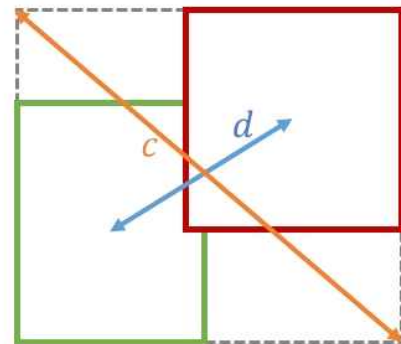
DIoU (distance-IoU)

IoU as Loss function

$$L_{IoU} = 1 - IoU$$

Distance-IoU Loss

$$L_{DIOU} = L_{IoU} + \frac{(\hat{b}_x - b_x)^2 + (\hat{b}_y - b_y)^2}{c^2}$$



02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) starts
in the collection of human

03 Loss

CloU (Complete-IoU)

Complete-IoU Loss

$$v = \frac{4}{\pi^2} \left(\tan^{-1} \frac{\hat{w}}{\hat{h}} - \tan^{-1} \frac{w}{h} \right)^2$$

Aspect ratio의 차이를 0~1 사이로 정규화

$$\alpha = \frac{v}{(1 - IoU) + v}$$

IoU가 작을 때에는 가중치를 적게 만듦

$$L_{CloU} = L_{DIoU} + \alpha v$$

IoU Loss와 비교해서 성능 향상

02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial Intelligence (AI) starts
in the collection of human

03 Loss

Loss Function

Box Loss

$$\lambda_{scale} = 2 - \frac{w_{GT} h_{GT}}{w_{input} h_{input}}$$

$$L_{box} = \sum 1^{obj} \lambda_{scale} L_{CloU}$$

$$= \sum \lambda_{scale} L_{CloU} \quad \text{for Obj}=1$$

Class Loss

$$L_{class} = \sum 1^{obj} \sum H$$

$$= \sum \sum H \quad \text{for Obj}=1$$

($\sum H$: sum of binary cross entropy)

03 Loss

Loss Function

Confidence Loss

$$L_{conf}^{obj} = \sum 1^{obj} \cdot -\log \hat{p}_c = \sum -\log \hat{p}_c$$

$$= \sum -\log \hat{p}_c \quad (\text{for Obj}=1)$$

$$L_{conf}^{noobj} = \sum 1^{noobj} \cdot -\log(1 - \hat{p}_c)$$

$$= \sum -\log(1 - \hat{p}_c)$$

(for Obj=0 and max_iou < iou_loss_thres)

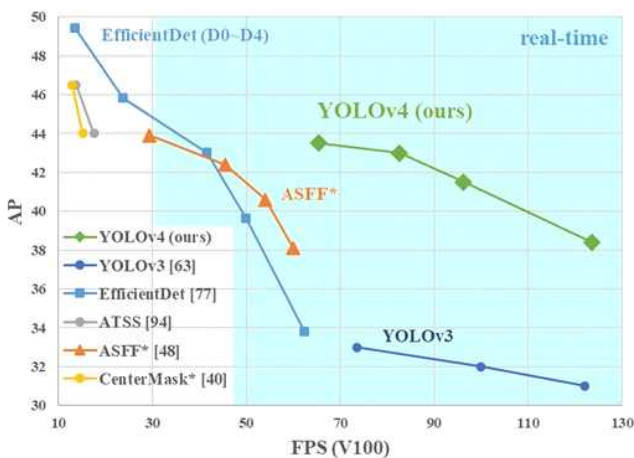
Total Loss

$$L = L_{box} + L_{class} + L_{conf}^{obj} + L_{conf}^{noobj}$$

04 EfficientDet

YOLO의 성능

MS COCO Object Detection



이미지 출처 :

1 빠른 실행 속도

- 적절한 성능 대비 높은 수행 속도
- Single GPU에서 실시간 동영상 지원, 일반 개발자들이 사용하기에 적절
- 가장 높은 성능을 원한다면 **EfficientDet** 등을 추천

2 모바일 대응

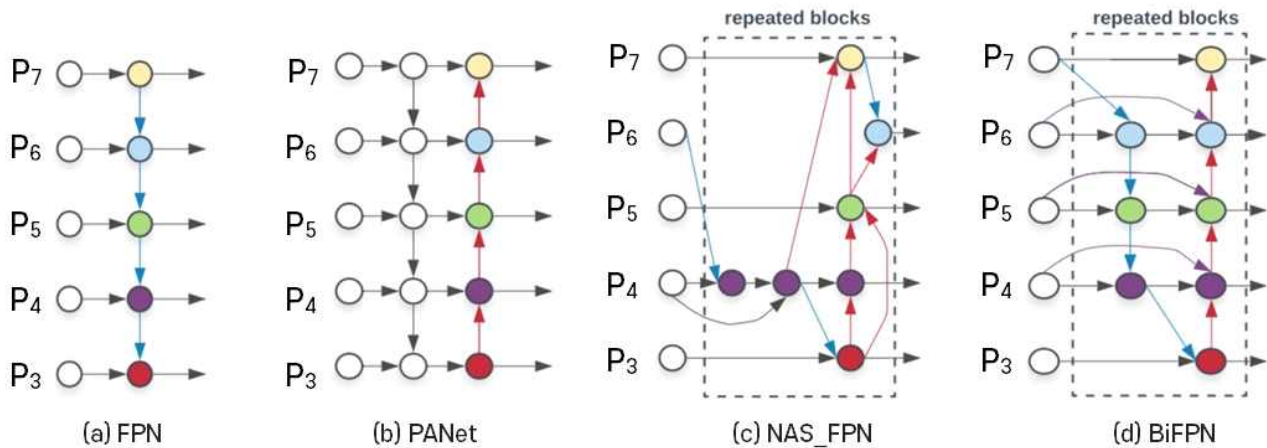
- OpenCV 등 지원
- Yolo tiny

04 EfficientDet



EfficientDet – Feature network design

AutoML의 방법론을 Object detection에 적용



이미지 출처 Mingxing Tan et. al EfficientDet: Scalable and Efficient Object Detection (2019) arXiv:1911.09070

04 EfficientDet

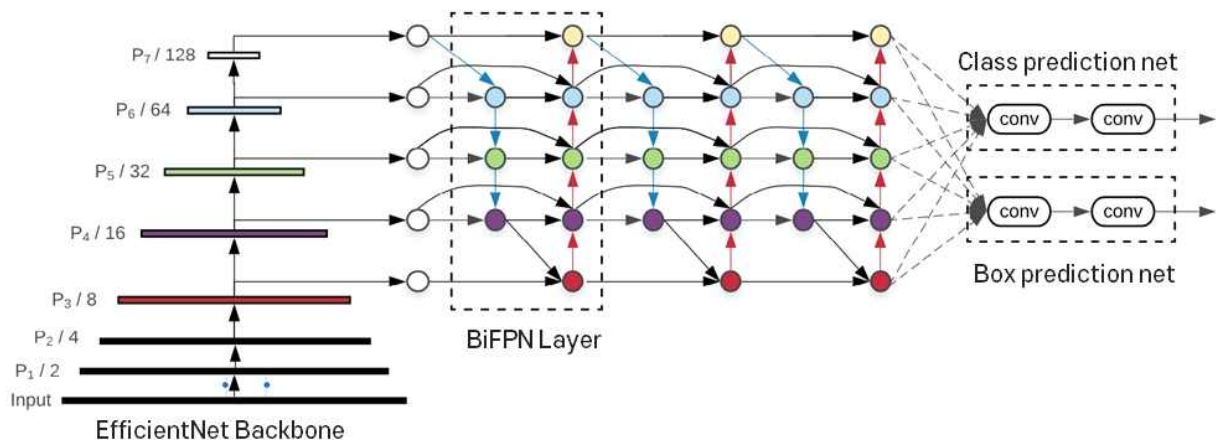


EfficientDet – Architecture

Mingxing Tan et. al
EfficientDet: Scalable and Efficient Object Detection (2019) arXiv:1911.09070

Backbone - EfficientNet, Feature network -BiFPN

shared class/box prediction network

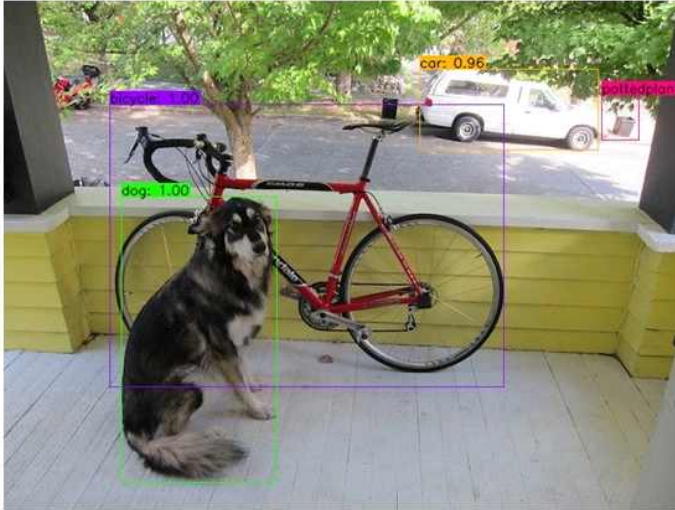


02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial intelligence (AI) relies
on the collection of human

05 실행

Object detection 실행 결과



Pascal VOC 2012 학습 결과

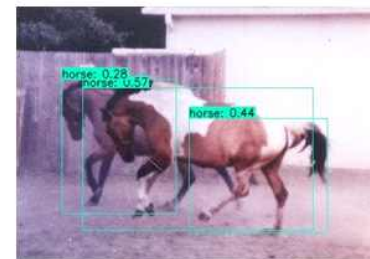
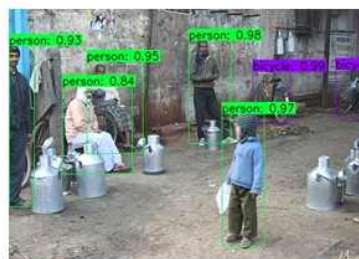
원본 이미지 출처 : <https://github.com/AlexeyAB/darknet.git>

02 YOLO의 상세 구성요소와 EfficientDet 소개

Artificial intelligence (AI) relies
on the collection of human

05 실행

Object detection 실행 결과



이미지 출처 : Pascal VOC dataset

05 실행



Object detection 실행 결과



1 거리 영상

Pascal VOC 2012로 학습

2 강아지 영상

COCO로 학습

(YOLOv4의 사전 학습)

동영상 원본 출처 :

Youtube German Korb

- No copyright

Youtube CHANNEL PMI

- Create Commons Licence

Aristotle recognized 120 values
in the simulation of human

APPLICATION

3차시

객체탐지 실습

직접 객체탐지 해보기



실습 PRACTICE

Artificial Intelligence (AI) refers to the simulation of human intelligence in machines that are programmed to think and learn like humans. AI can be used in various ways, such as in the form of chatbots, recommendation systems, and self-driving cars.

This page will help you to understand the basic concepts of AI and how it can be used in various ways. It will also provide you with some examples of AI applications and how they work.

영상처리 실습

❖ YOLO를 이용한 객체탐지 실습



SUMMARY

학습정리

- ❖ Grid와 Anchor를 이용하여, 객체탐지를 위한 컨볼루션 신경망을 구성할 수 있습니다.
- ❖ Batch normalization과 Mish 활성화함수는 CNN backbone의 학습 속도 및 성능향상에 효과적입니다.
- ❖ YOLO의 backbone은 CSPdarknet-53을 기본적으로 사용합니다.

학습정리

- ❖ Spacial Pyramid Pooling레이어를 통하여 다양한 scale의 이미지 감지에 도움을 줄 수 있습니다.
- ❖ Feature Pyramid Network(PAN)은 backbone의 출력을 3단계 크기로 출력합니다.
- ❖ EfficientDet은 최근 등장한 고성능의 객체 감지 모델로서 높은 성능이 필요할 때에 적합합니다.

케라스 랑 놀면 뭐 할까?

Global average pooling,
Mish, Batch normalization

MNIST CNN 모델 코드를 다시 불러옵니다.

Batch normalization을 추가하고
Mish activation 함수를
적용합니다.

최종 Fully Connected 레이어를 제거하고
Convolution을 추가한 후에
Global average pooling을 적용합니다.

기존 모델과 학습시간 및 인식률을 비교해 봅시다.

참고 문헌

REFERENCE

강의에 필요한 URL

- 구글 코랩: colab.research.google.com
- 실습파일: github.com/kotech1/computervision

참고 URL

- **YOLO v1, v2(9000), v3, v4 논문:**
<https://pjreddie.com/media/files/papers/yolo.pdf>
<https://arxiv.org/abs/1612.08242>
<https://pjreddie.com/media/files/papers/YOLOv3.pdf>
<https://arxiv.org/abs/2004.10934>
- **YOLO 해설(Jonathan Hui):**
https://medium.com/@jonathan_hui/yolov4-c9901eaa8e61
- **Andrew Ng의 객체탐지 유튜브 강좌(한글자막):**
www.youtube.com/playlist?list=PLkDaE6sCZn6GI29AoE31iwdVwSG-KnDzF

☎ 서체 출처 : 에스코어드림체-(주)에스코어, 나눔글꼴체-(주)네이버, 배달의민족체-우아한형제들

참고 문헌

REFERENCE

강의에 사용된 시청각 자료 및 논문

- **Joseph Redmon et.al**
You Only Look Once: Unified, Real-Time Object Detection (2016)
arXiv:1506.02640
- **Joseph Redmon 트위터 계정**
<https://twitter.com/pjreddie/status/1230524770350817280>
- **Tsung-Yi Lin et.al**
Pyramid Networks for Object Detection (2017) arXiv:1612.03144
- **J. Redmon YOLO9000: Better, Faster, Stronger (2016)**
arXiv:1612.08242
- **YOLOv3: An Incremental Improvement**
Joseph Redmon, Ali Farhadi
<https://arxiv.org/pdf/1804.02767.pdf>

☎ 서체 출처 : 에스코어드림체-(주)에스코어, 나눔글꼴체-(주)네이버, 배달의민족체-우아한형제들

참고 문헌

REFERENCE

The copyright notice for this document is as follows: This document is copyrighted by the author(s). All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without prior written permission from the author(s).

강의에 사용된 시청각 자료 및 논문

- **DC-SPP-YOLO: Dense Connection and Spatial Pyramid Pooling Based YOLO for Object Detection**
Zhanchao Huang, Jianlin Wang
<https://arxiv.org/ftp/arxiv/papers/1903/1903.08589.pdf>
- **Path Aggregation Network for Instance Segmentation**
Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, Jiaya Jia
<https://arxiv.org/pdf/1803.01534.pdf>
- **EfficientDet: Scalable and Efficient Object Detection**
Mingxing Tan, Ruoming Pang, Quoc V. Le
<https://arxiv.org/pdf/1911.09070.pdf>
- <https://github.com/AlexeyAB/darknet>
- **Pascal VOC dataset**

서체 출처 : 에스코어드림체-(주)에스코어, 나눔글꼴체-(주)네이버, 배달의민족체-우아한형제들

참고 문헌

REFERENCE

The copyright notice for this document is as follows: This document is copyrighted by the author(s). All rights reserved. No part of this document may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording, or by any information storage or retrieval system, without prior written permission from the author(s).

강의에 사용된 시청각 자료 및 논문

- <https://github.com/AlexeyAB/darknet>
- **Pascal VOC dataset**
- **Youtube German Korb channel**
<https://youtu.be/N1H6O-yx7ZY>
- **Youtube CHANNEL PMI**
https://youtu.be/W2_fxTpg82s

서체 출처 : 에스코어드림체-(주)에스코어, 나눔글꼴체-(주)네이버, 배달의민족체-우아한형제들