Car Object Detection

학과: 컴퓨터공학전공
학번: 2017108246
발표자: 고대은

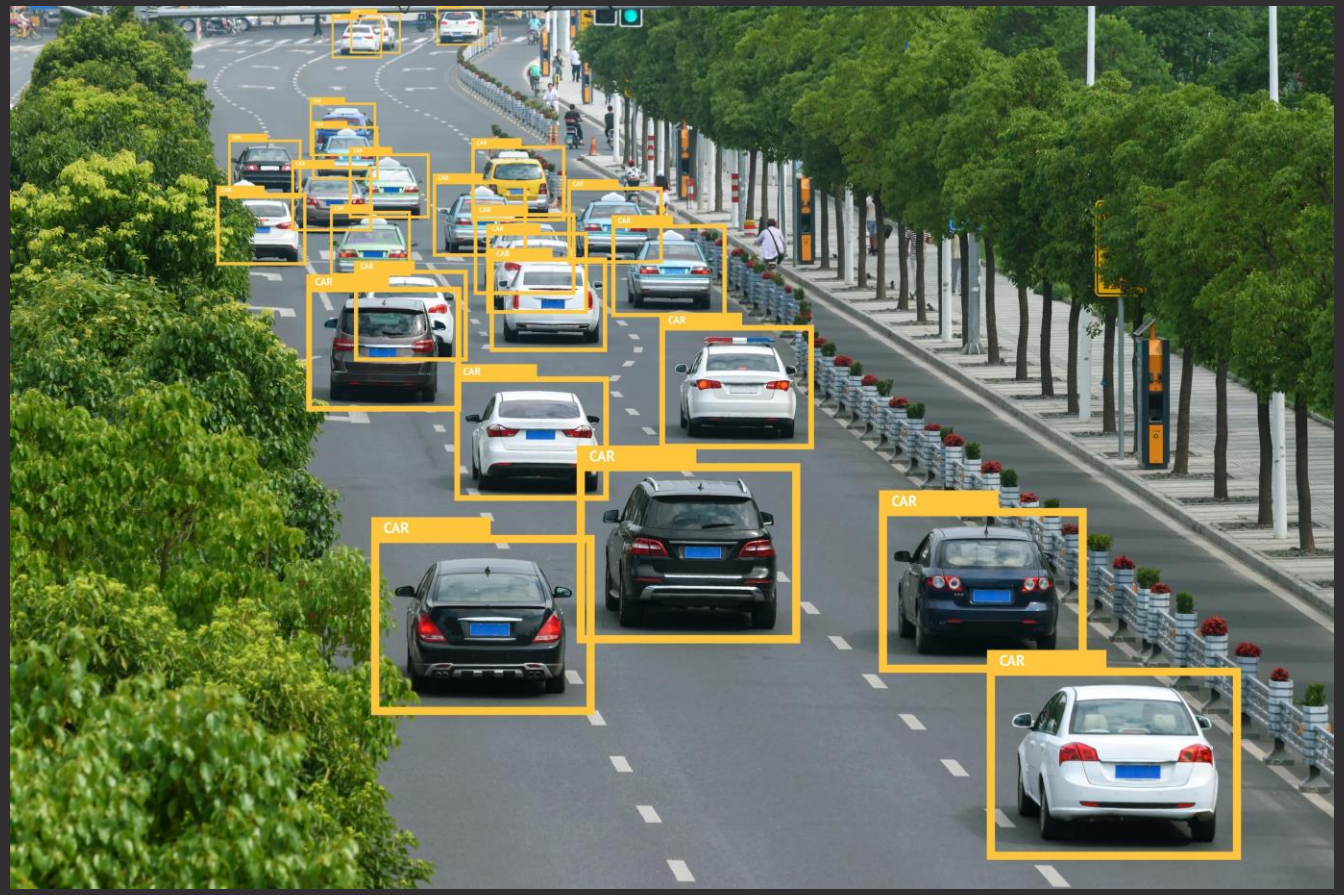
목차

객체 탐지

YOLOv5

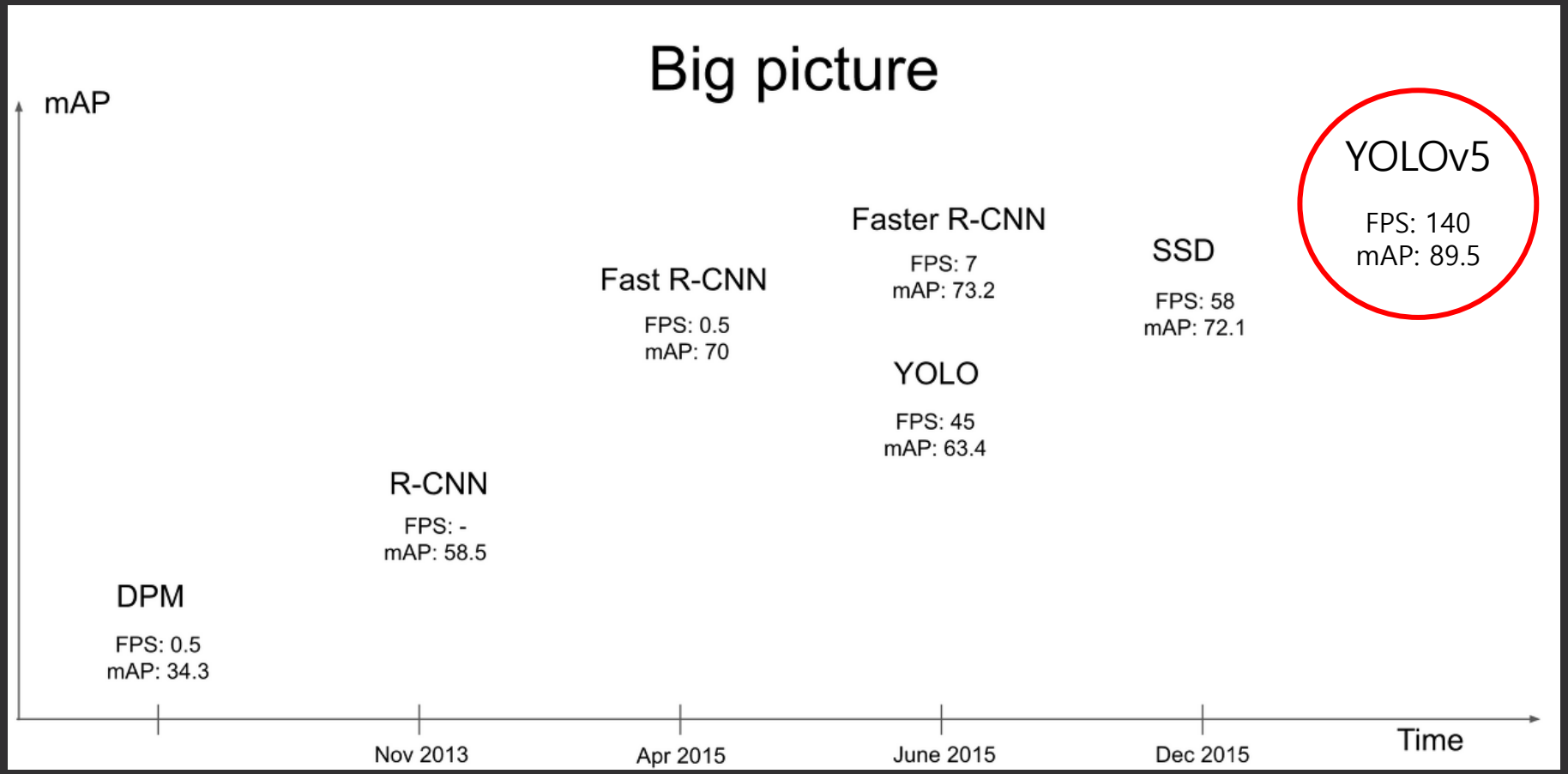
예제 코드

객체 탐지



이미지 내에
특정 객체가 어떤 위치에 있는지
찾는 것

YOLOv5

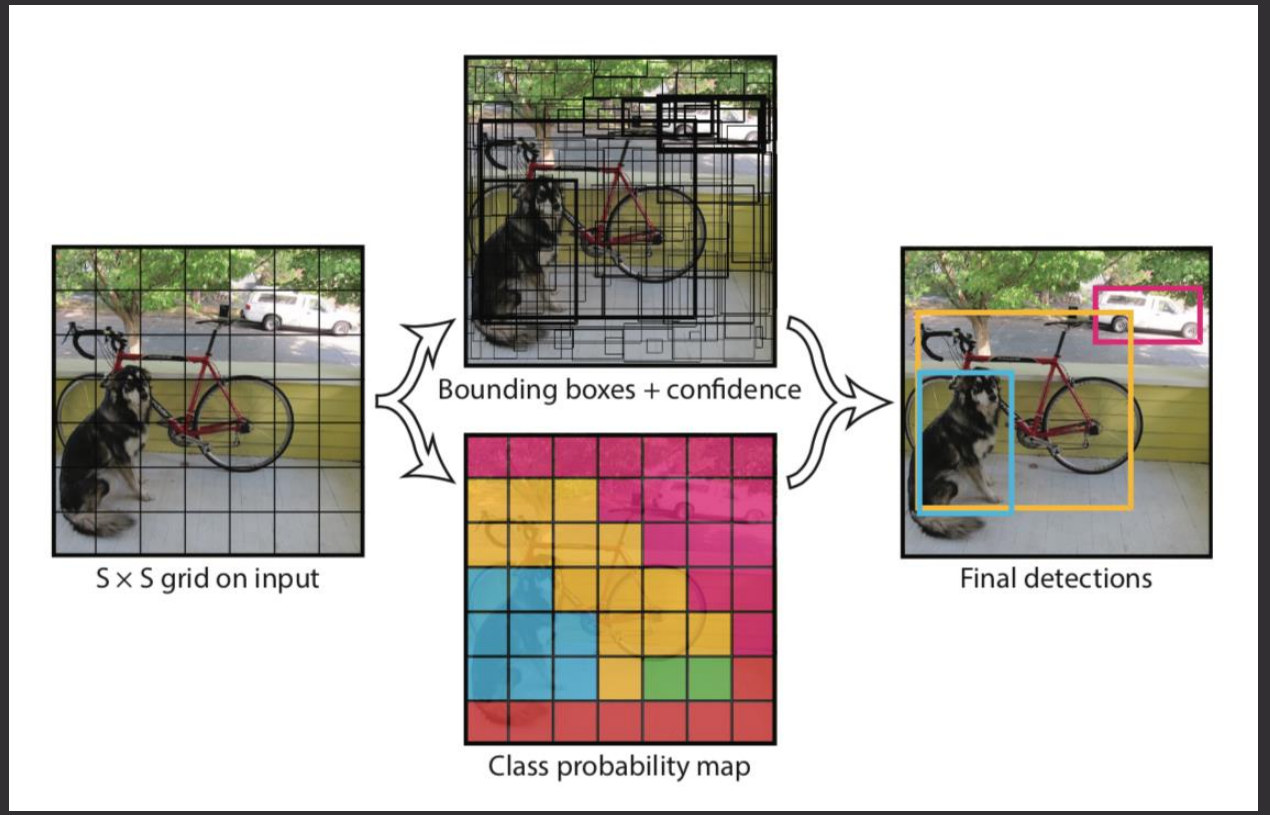

Big picture
mAP
DPM
FPS: 0.5
mAP: 34.3
R-CNN
FPS: -
mAP: 58.5
Fast R-CNN
FPS: 0.5
mAP: 70
Faster R-CNN
FPS: 7
mAP: 73.2
YOLO
FPS: 45
mAP: 63.4
SSD
FPS: 58
mAP: 72.1
YOLOv5
FPS: 140
mAP: 89.5
Nov 2013
Apr 2015
June 2015
Dec 2015
Time

YOLOv5

Region Proposal

Classification

바운딩 박스 계산



그리드 단위로 분류

예제 코드

```
import os, time, random
import numpy as np
import pandas as pd
import cv2, torch
from tqdm.auto import tqdm
import shutil as sh

from IPython.display import Image, clear_output
import matplotlib.pyplot as plt
%matplotlib inline
```

라이브러리 설치

```
%%time

!git clone https://github.com/ultralytics/yolov5  # clone repo
!pip install -U pycocotools
!pip install -qr yolov5/requirements.txt  # install dependencies
!cp yolov5/requirements.txt ./
```

YOLOv5 설치

학습 데이터 및 라벨 형태

```yaml
#data.yaml
train: ../train/images
val: ../valid/images

nc: 0
names: ['car']
```

Label map

Image

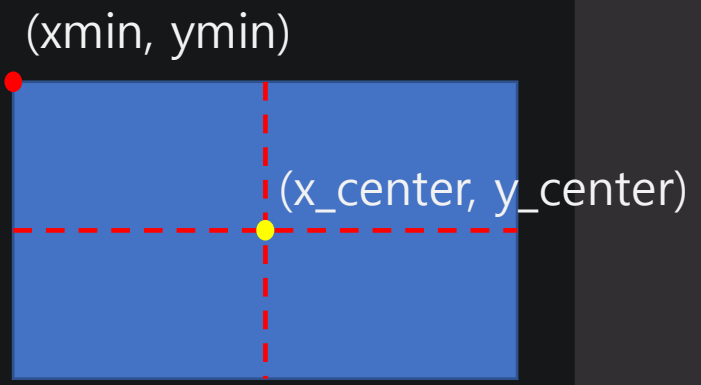
```
# label.txt
0 0.484375 0.4254807692307692 0.5216346153846154 0.21153846153846154
```

class x_center y_center w y

Label - bounding box

데이터 전처리 – 객체 중앙값 계산

```
img_h, img_w, num_channels = (380, 676, 3)
df = pd.read_csv('../input/car-object-
detection/data/train_solution_bounding_boxes (1).csv')
df.rename(columns={'image':'image_id'}, inplace=True)
df['image_id'] = df['image_id'].apply(lambda x: x.split('.')[0])
df['x_center'] = (df['xmin'] + df['xmax'])/2
df['y_center'] = (df['ymin'] + df['ymax'])/2
df['w'] = df['xmax'] - df['xmin']
df['h'] = df['ymax'] - df['ymin']
df['classes'] = 0
df['x_center'] = df['x_center']/img_w
df['w'] = df['w']/img_w
df['y_center'] = df['y_center']/img_h
df['h'] = df['h']/img_h
df.head()
```



데이터 전처리 – 객체 중앙값 계산

image	xmin	ymin	xmax	ymax
vid_4_1000.jpg	281.2590449	187.0350708	327.7279305	223.225547
vid_4_10000.jpg	15.16353111	187.0350708	120.3299566	236.4301802
vid_4_10040.jpg	239.1924747	176.7648005	361.9681621	236.4301802
vid_4_10020.jpg	496.4833575	172.3632561	630.0202605	231.5395753

	image_id	xmin	ymin	xmax	ymax	x_center	y_center	w	h	classes
0	vid_4_1000	281.259045	187.035071	327.727931	223.225547	0.450434	0.539817	0.068741	0.095238	0
1	vid_4_10000	15.163531	187.035071	120.329957	236.430180	0.100217	0.557191	0.155572	0.129987	0
2	vid_4_10040	239.192475	176.764800	361.968162	236.430180	0.444645	0.543678	0.181621	0.157014	0
3	vid_4_10020	496.483358	172.363256	630.020261	231.539575	0.833213	0.531451	0.197540	0.155727	0
4	vid_4_10060	16.630970	186.546010	132.558611	238.386422	0.110347	0.559122	0.171491	0.136422	0

데이터 전처리 - csv to txt

```
source = 'training_images'
if True:
    for fold in [0]:
        val_index = index[len(index)*fold//5:len(index)*(fold+1)//5]
        for name,mini in tqdm(df.groupby('image_id')):
            if name in val_index:
                path2save = 'val2017/'
            else:
                path2save = 'train2017/'
            if not os.path.exists('/tmp/convertor/fold{}/labels/'.format(fold)+path2save):
                os.makedirs('/tmp/convertor/fold{}/labels/'.format(fold)+path2save)
            with open('/tmp/convertor/fold{}/labels/'.format(fold)+path2save+name+".txt", 'w+') as f:
                row = mini[['classes','x_center','y_center','w','h']].astype(float).values
                row = row.astype(str)
                for j in range(len(row)):
                    text = ' '.join(row[j])
                    f.write(text)
                    f.write("\n")
            if not os.path.exists('/tmp/convertor/fold{}/images/{}'.format(fold,path2save)):
                os.makedirs('/tmp/convertor/fold{}/images/{}'.format(fold,path2save))
            sh.copy("/kaggle/input/car-object-
detection/data/{}/{}.jpg".format(source,name),'/tmp/convertor/fold{}/images/{}/{}.jpg'.format(fold,path2save,name))
```

Training

Hyper parameter

Label map

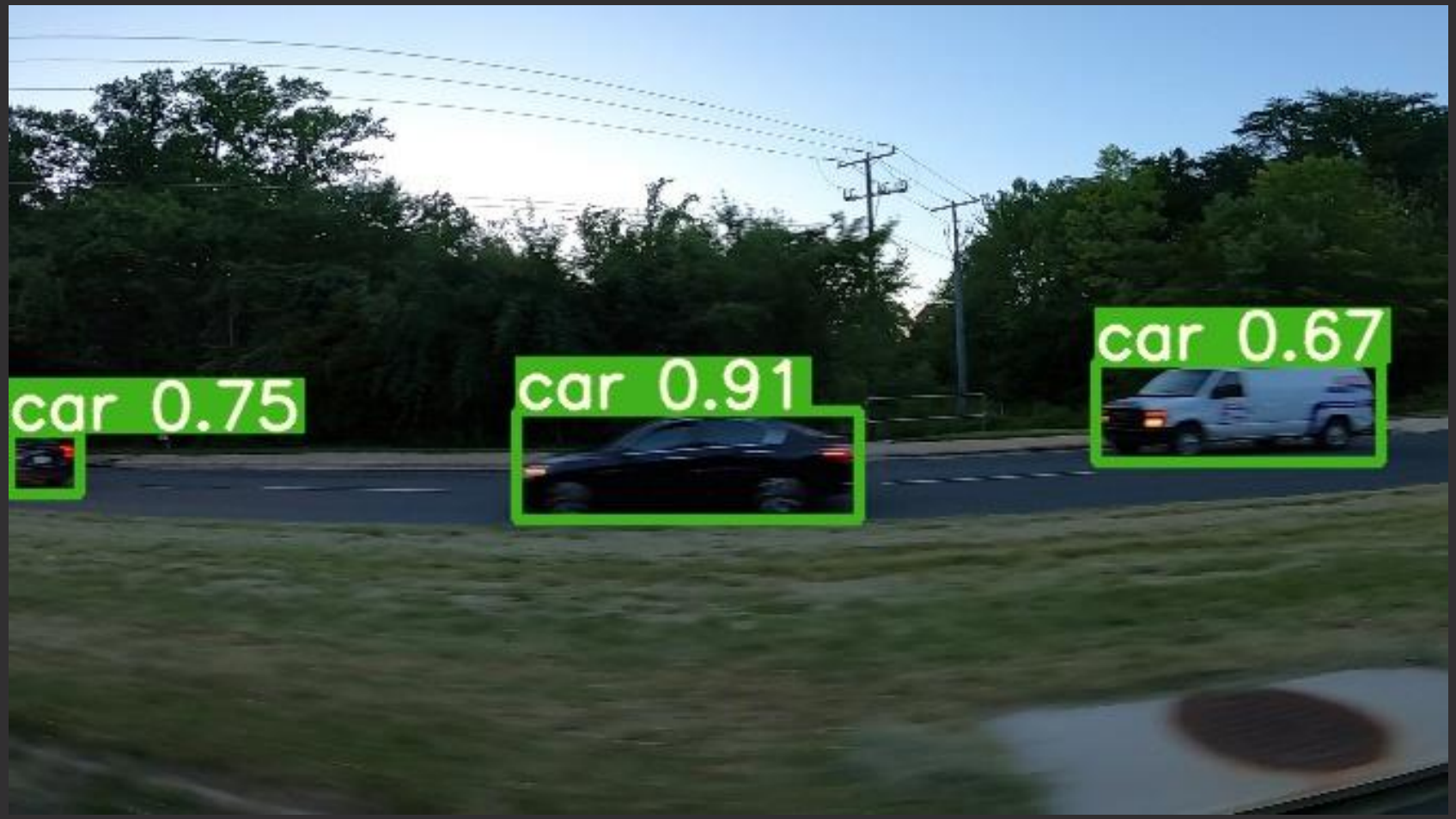
```
!python train.py --batch 2 --epochs 1 --data ../input/yolov5-config/car.yaml --cfg ../input/yolov5-config/yolov5x.yaml --name yolov5x_fold0_new
```

Prediction

학습이 완료된 모델

```
!python yolov5/detect.py --weights yolov5/yolov5s.pt --img 676 --conf 0.4 --source /kaggle/input/car-object-detection/data/testing_image
```

결과 확인



Kaggle

https://www.kaggle.com/sshikamaru/car-object-detection/code
https://www.kaggle.com/balraj98/yolo-v5-car-object-detection

Q&A