Kissipo Learning for Deep Learning

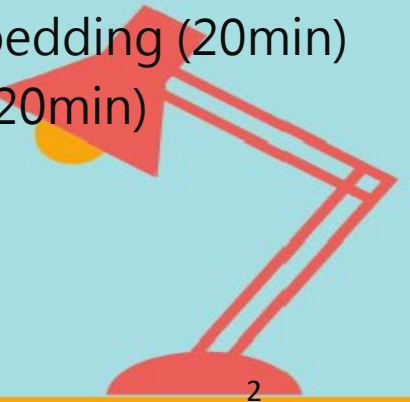
Topic 22: Using YoloV5 for RSD (20min)

Hsueh-Ting Chu

KLDL-W8-T22

Topics

- Topic 01: Introduction to Deep Learning (20min)
- Topic 02: Kissipo Learning for Deep Learning (20min)
- Topic 03: Python quick tutorial (20min)
- Topic 04: Numpy quick tutorial (15min)
- Topic 05: Pandas quick tutorial (15min)
- Topic 06: Scikit-learn quick tutorial (15min)
- Topic 07: OpenCV quick tutorial (15min)
- Topic 08: Image Processing basics (20min)
- Topic 09: Machine Learning basics (20min)
- Topic 10: Deep Learning basics (20min)
- Topic 11: TensorFlow overview (20min)
- Topic 12: CNN with TensorFlow (20min)
- Topic 13: RNN with TensorFlow (20min)
- Topic 14: PyTorch overview (20min)
- Topic 15: CNN with PyTorch (20min)
- Topic 16: RNN with Pytorch (20min)
- Topic 17: Introduction to AOI (20min)
- Topic 18: AOI simple Pipeline (A) (20min)
- Topic 19: AOI simple Pipeline (B) (20min)
- Topic 20: Introduction to Object detection (20min)
- Topic 21: YoloV5 Quick Tutorial (20min)
- Topic 22: Using YoloV5 for RSD (20min)
- Topic 23: Introduction to NLP (20min)
- Topic 24: Introduction to Word Embedding (20min)
- Topic 25: Name prediction project (20min)



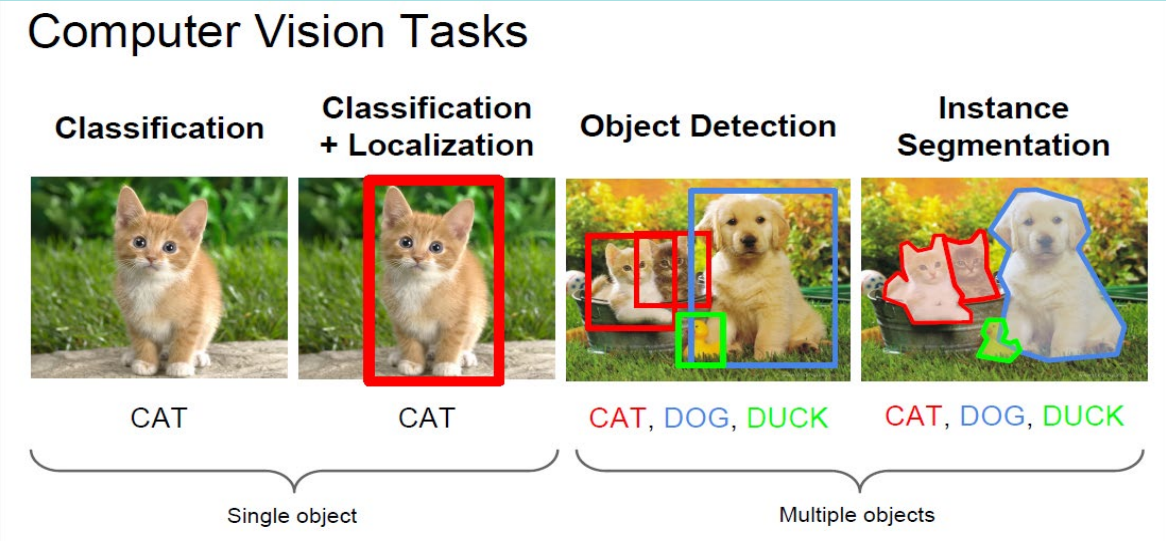
Course Schedule

- W1 - Course Introduction
- W2 - DL Programming Basics(1)
- W3 - DL Programming Basics(2)
- W4 - DL with TensorFlow
- W5 - Midterm
- W6 - DL with PyTorch
- W7 - AOI hands-on project
- W8 - RSD hands-on project
- W9 - NLP hands-on project
- W10 - Final exam

DL: Deep Learning
AOI: Automated Optical Inspection
RSD: Road Sign Detection
NLP: Natural Language Processing

Week 8 Topics

- Topic 20: Introduction to Object detection (20min)
- Topic 21: YoloV5 Quick Tutorial (20min)
- Topic 22: Using YoloV5 for RSD (20min)



- Classification
- Positioning (Classification + Localization): mark the position and size of an object.
- Object Detection: Mark the location and size of multiple objects.
- Semantic Segmentation: Do not distinguish instances.
- Instance Segmentation: Mark instances.

- Road sign recognition is the basic function of automatic driving system.

Road Sign Detection Dataset

About this Dataset

This dataset contains **877** images of **4 distinct classes** for the objective of **road sign detection**.

Bounding box annotations are provided in the PASCAL VOC format

The classes are:

- Trafic Light;
- Stop;
- Speedlimit;
- Crosswalk.

https://www.kaggle.com/datasets/andrewmvd/road-sign-detection

1. Preparing YoloV5

KLDL-22-Using YoloV5 for RSD

檔案 編輯 檢視畫面 插入 執行階段 工具 說明 已儲存所有變更

目錄

+ 程式碼 + 文字

Topic 22: Using YoloV5 for RSD (Road Sign Detection)

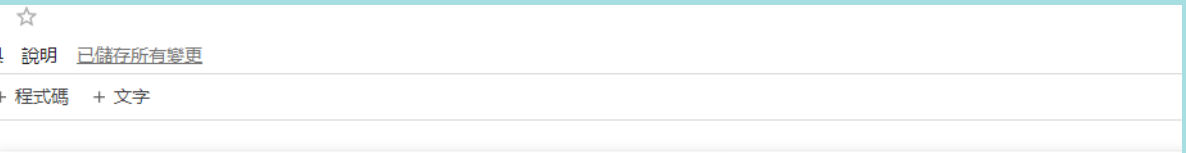
Part-1: Preparing YoloV5

Part-2: Preprocessing data

Part-3 Train a YoloV5 model

Part-4 Predict a Test image with the trained YoloV5 model

區段



Topic 22: Using YoloV5 for RSD (Road Sign Detection)

Data from https://www.kaggle.com/datasets/andrewmvd/road-sign-detection

Part-1: Preparing YoloV5

```
[ ]
```

```
[ ] !git clone https://github.com/ultralytics/yolov5  # clone
    %cd yolov5
    %pip install -qr requirements.txt  # install

    import torch
    import utils
    display = utils.notebook_init()  # checks
```

2. Preprocessing data

Part-2: Preprocessing data

If the following command does not work, please download it, put it on your Google drive, and set up sharing

Download from: https://drive.google.com/file/d/1z5QTlNnZaA2e6uQXIa5On6x6gHmEyzQK/view?usp=sharing

```
[ ]  %%bash
     pip install --upgrade gdown
     gdown https://drive.google.com/uc?id=1z5QTlNnZaA2e6uQXIa5On6x6gHmEyzQK
     unzip roadsign.zip
     rm roadsign.zip
```

```
[ ]  !pwd
```

```
[ ]  !touch roadsign.yaml
```

```
# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1
path: roadsign
train: images
val: images
test: images

# number of classes
nc: 4
```

path: roadsign
train: images
val: images
test: images

number of classes
nc: 4

class names
names: [
'trafficlight',
'stop',
'speedlimit',
'crosswalk']

3. Train a YoloV5 model

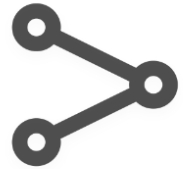
Part-3 Train a YoloV5 model

```
[ ]  # Train YOLOv5s on Drone for 10 epochs
     !python train.py --img 640 --batch 32 --epochs 10 --data roadsign.yaml --weights yolov5s.pt --cache
```

```
[ ]  !pwd
```

4. Predict a Test image with the trained YoloV5 model

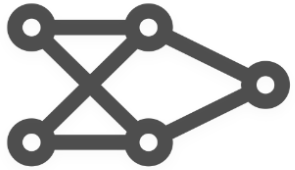
YOLOv5 models



Small

YOLOv5s

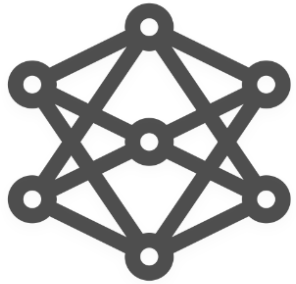
15 MB_{FP16}
2.4 ms_{V100}
37.0 mAP_{COCO}



Medium

YOLOv5m

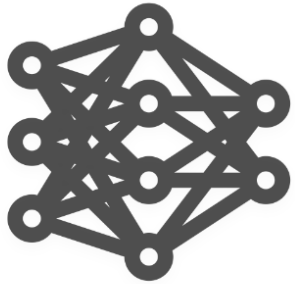
42 MB_{FP16}
3.4 ms_{V100}
44.3 mAP_{COCO}



Large

YOLOv5l

92 MB_{FP16}
4.4 ms_{V100}
47.7 mAP_{COCO}



XLarge

YOLOv5x

170 MB_{FP16}
6.9 ms_{V100}
50.8 mAP_{COCO}

Thanks!

Q&A