



深度學習 Deep Learning (11)

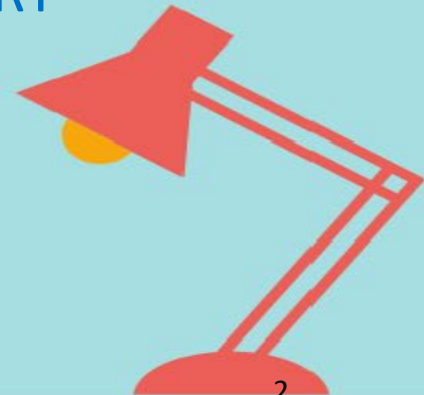
112-1

朱學亭老師



課程大綱

- W1-課程介紹/Introduction
- W2-Python/Colab and TensorFlow
- W3-Numpy/Pandas and PyTorch
- W4-Sklearn and 機器學習
- W5-神經網路, TensorFlow, PyTorch
- W6-載客熱點預測
- W7-自動光學檢查(AOI)-1
- W8-自動光學檢查(AOI)-2
- W9-Midterm presentation
- W10-RNN
- W11-Object detection
- W12- PyTorch
- W13-NLP1-Word2Vec
- W14-NLP2-Seq2Seq, Attention
- W15-NLP3-Transformer, BERT
- W16-AICUP 1
- W17-AICUP 2
- W18-Final presentation



CV jobs

Computer Vision Tasks

Classification



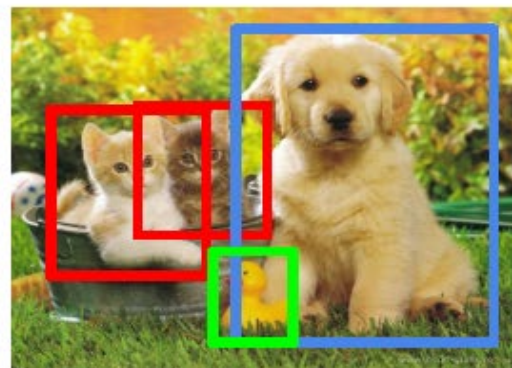
CAT

**Classification
+ Localization**



CAT

Object Detection



CAT, DOG, DUCK

**Instance
Segmentation**

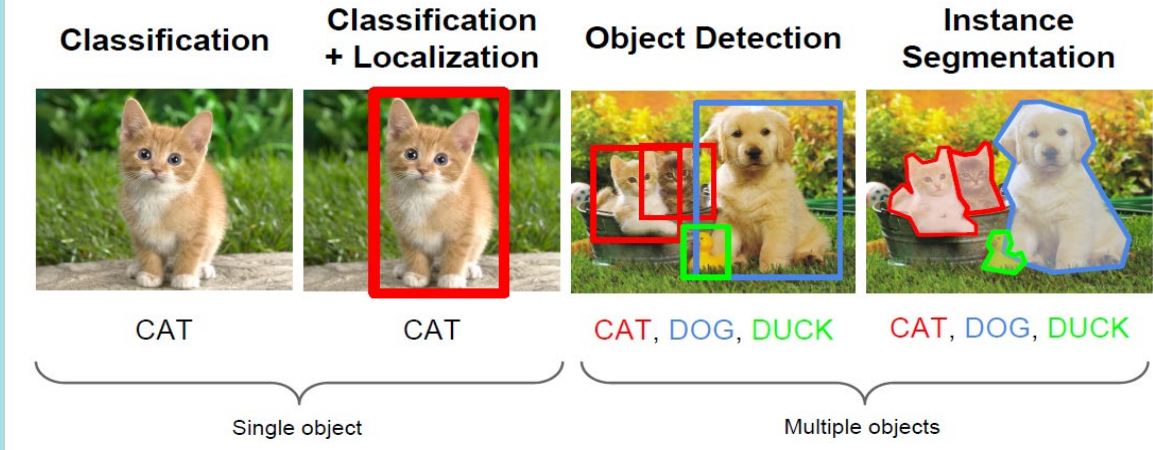


CAT, DOG, DUCK

Single object

Multiple objects

Computer Vision Tasks



- 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.



YOLOv5 Tutorial




The screenshot shows the Google Colab interface for the YOLOv5 Tutorial. The top bar includes the Colab logo and the title "YOLOv5 Tutorial". Below the title is a navigation menu with options: 檔案 (Files), 編輯 (Edit), 檢視畫面 (View), 插入 (Insert), 執行階段 (Runtime), 工具 (Tools), and 說明 (Help). The left sidebar contains a table of contents with sections like Setup, Detect, Validate, Train, Visualize, and Environments. The main content area displays the YOLOv5 v6.2 logo by ultralytics, followed by buttons to "Run on Gradient", "Open in Colab", and "Open in Kaggle". Below these buttons, a text block states: "This YOLOv5 notebook by Ultralytics presents simple train, validate and predict examples to help start your AI adventure. See GitHub for community support or contact us for professional support." The "Setup" section is expanded, showing instructions to clone the GitHub repository and install dependencies. A code block at the bottom shows the terminal commands:

```
[ ] !git clone https://github.com/ultralytics/yolov5 # clone
    %cd yolov5
```



1. Setup

 KLDL-21-YoloV5 Quick Tutorial ☆

PRO 檔案 編輯 檢視畫面 插入 執行階段 工具 說明 最近於 2月11日 編輯

☰ 目錄

🔍 Topic 21: YoloV5 Quick Tutorial

{x} Setup

📁 Detect

Validate

Train

Select YOLOv5 🚀 logger

Visualize

Comet Logging and Visualization 🌟 NEW

ClearML Logging and Automation 🌟 NEW

Local Logging


Environments

Status

Appendix

+ 區段

+ 程式碼 + 文字




Topic 21: YoloV5 Quick Tutorial

This [YOLOv5](#) 🚀 notebook by [Ultralytics](#) presents simple train, validate and predict examples to help start your community support.

▼ Setup

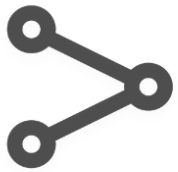
Clone GitHub [repository](#), install [dependencies](#) and check PyTorch and GPU.



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

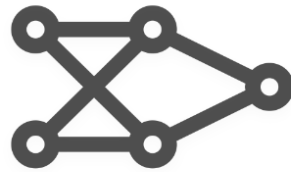


YOLOv5 models



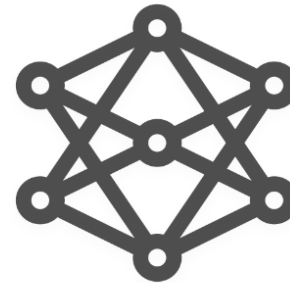
Small
YOLOv5s

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



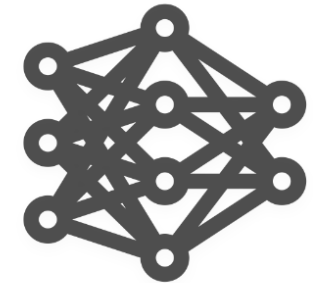
Medium
YOLOv5m

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}

2. Detect

`detect.py` runs YOLOv5 inference on a variety of sources, downloading models automatically from the [latest releases](#) to `runs/detect`. Example inference sources are:

```
python detect.py --source 0 # webcam
img.jpg # image
vid.mp4 # video
screen # screenshot
path/ # directory
'path/*.jpg' # glob
'https://youtu.be/Zgi9glksQhc' # YouTube
'rtsp://example.com/media.mp4' # RTSP, RTMP, HTTP stream
```

```
[ ] !python detect.py --weights yolov5s.pt --img 640 --conf 0.25 --source data/images
# display.Image(filename='runs/detect/exp/zidane.jpg', width=600)
```



3. Validate

3. Validate

Validate a model's accuracy on the [COCO](#) dataset's `val` or `test` splits. Models are downloaded automatically. To show results by class use the `--verbose` flag.

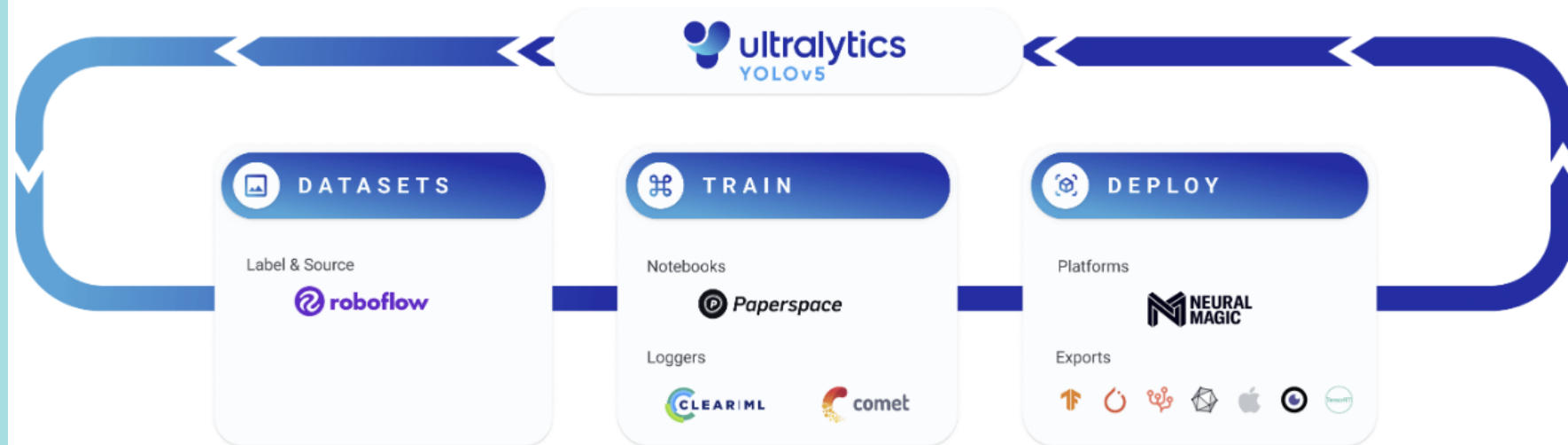
```
[ ] # Download COCO val
    torch.hub.download_url_to_file('https://ultralytics.com/assets/coco2017val.zip', 'tmp.zip')
    !unzip -q tmp.zip -d ../datasets && rm tmp.zip # unzip

[ ] # Validate YOLOv5s on COCO val
    !python val.py --weights yolov5s.pt --data coco.yaml --img 640 --half
```



4. Train

4. Train



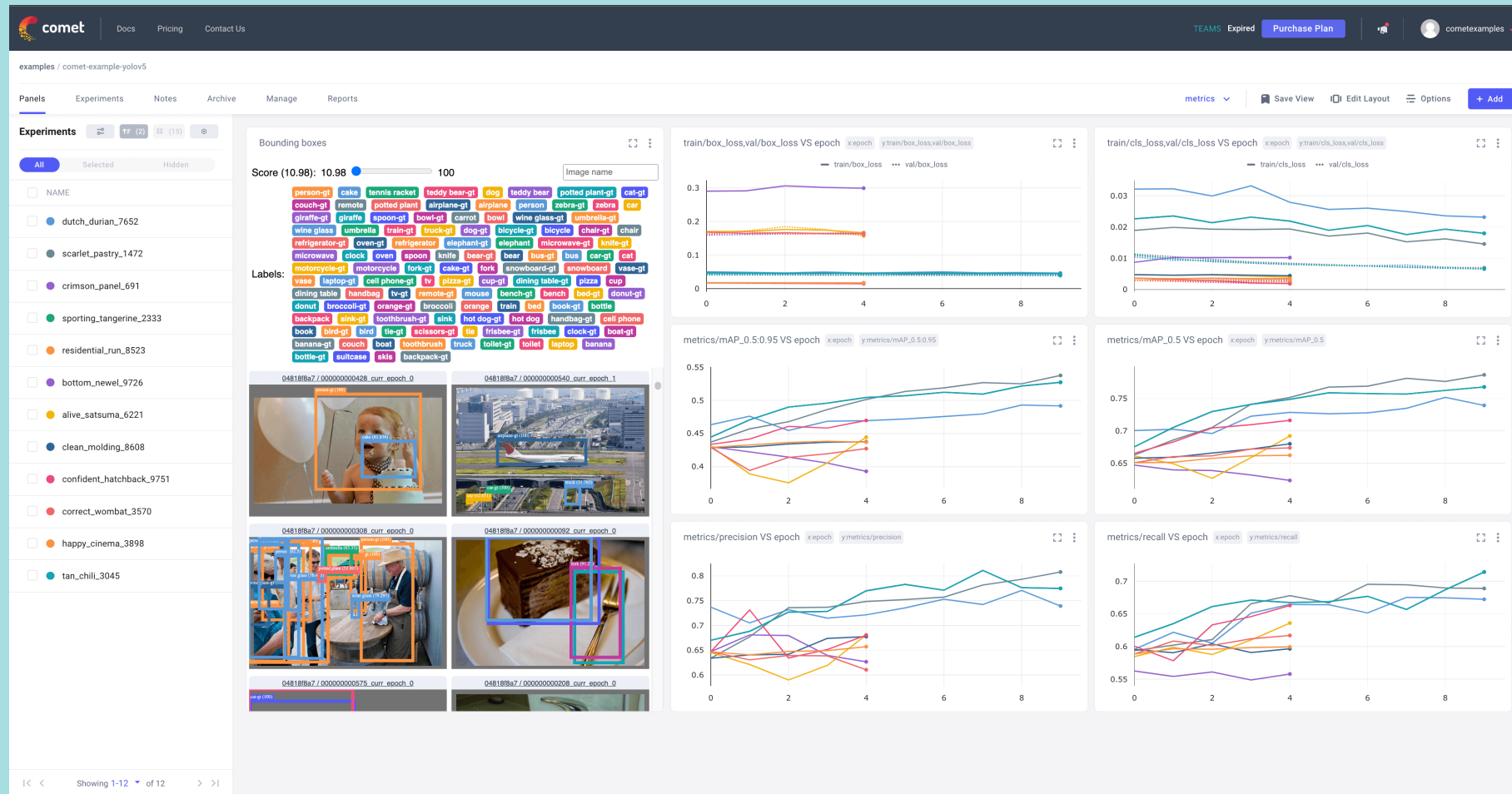
Close the active learning loop by sampling images from your inference conditions with the `roboflow` pip package

Train a YOLOv5s model on the [COCO128](#) dataset with `--data coco128.yaml`, starting from pretrained `--weights yolov5s.pt`, or from randomly initialized `--weights '' --cfg yolov5s.yaml`.

- **Pretrained Models** are downloaded automatically from the [latest YOLOv5 release](#)
- **Datasets** available for autodownload include: [COCO](#), [COCO128](#), [VOC](#), [Argoverse](#), [VisDrone](#), [GlobalWheat](#), [xView](#), [Objects365](#), [SKU-110K](#).
- **Training Results** are saved to `runs/train/` with incrementing run directories, i.e. `runs/train/exp2`, `runs/train/exp3` etc.



5. Visualize



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:

- Traffic Light;
- Stop;
- Speedlimit;
- Crosswalk.



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

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

Deep Learning Course

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



Data from <https://www.kaggle.com/datasets/andrewmyd/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



Thanks!

Q&A

