



# Kissipo Learning for Deep Learning

## Topic 21: YoloV5 Quick Tutorial (20min)

Hsueh-Ting Chu

KLDL-W8-T21

# 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

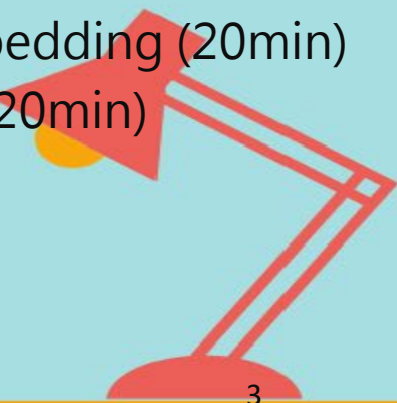
RSD: Road Sign Detection

NLP: Natural Language Processing



# 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)



# Week 8 Topics

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



# CV jobs

## Computer Vision Tasks

**Classification**



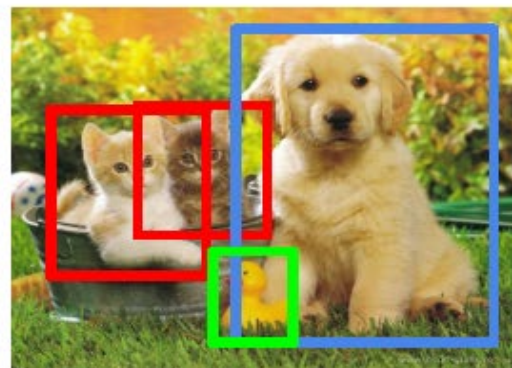
CAT

**Classification  
+ Localization**



CAT

**Object Detection**



CAT, DOG, DUCK

**Instance  
Segmentation**



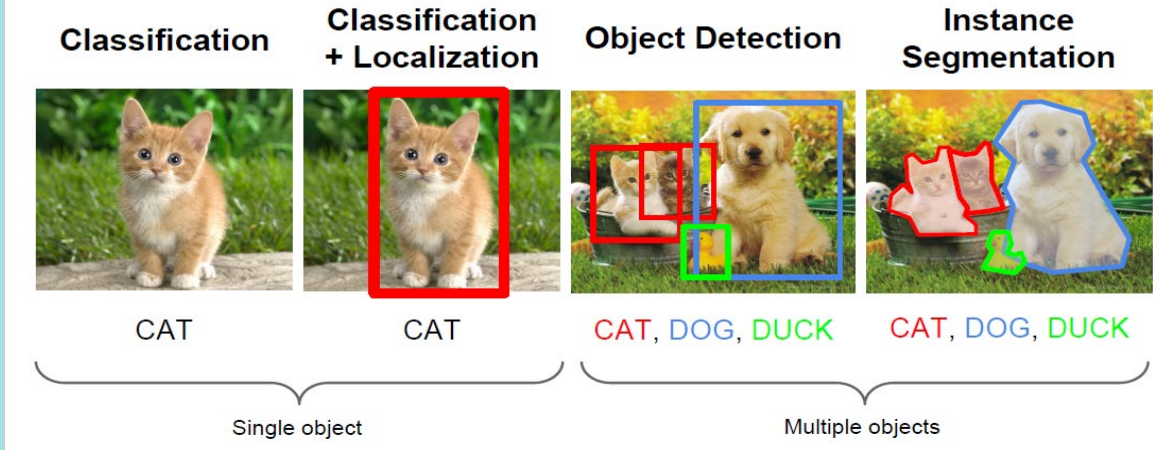
CAT, DOG, DUCK

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 this, 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, with a code block containing the following 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

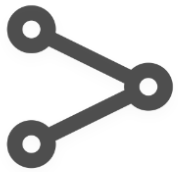
Clone GitHub [repository](#), install [dependencies](#) and check PyTorch and GPU.

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



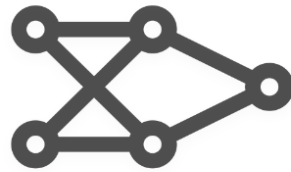


# YOLOv5 models



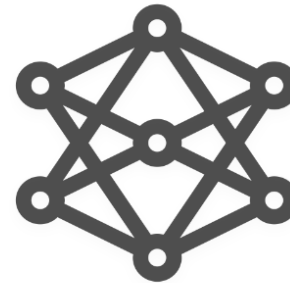
Small  
**YOLOv5s**

15 MB<sub>FP16</sub>  
2.4 ms<sub>V100</sub>  
37.0 mAP<sub>COCO</sub>



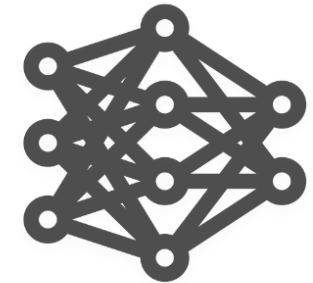
Medium  
**YOLOv5m**

42 MB<sub>FP16</sub>  
3.4 ms<sub>V100</sub>  
44.3 mAP<sub>COCO</sub>



Large  
**YOLOv5l**

92 MB<sub>FP16</sub>  
4.4 ms<sub>V100</sub>  
47.7 mAP<sub>COCO</sub>



XLarge  
**YOLOv5x**

170 MB<sub>FP16</sub>  
6.9 ms<sub>V100</sub>  
50.8 mAP<sub>COCO</sub>

## 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.

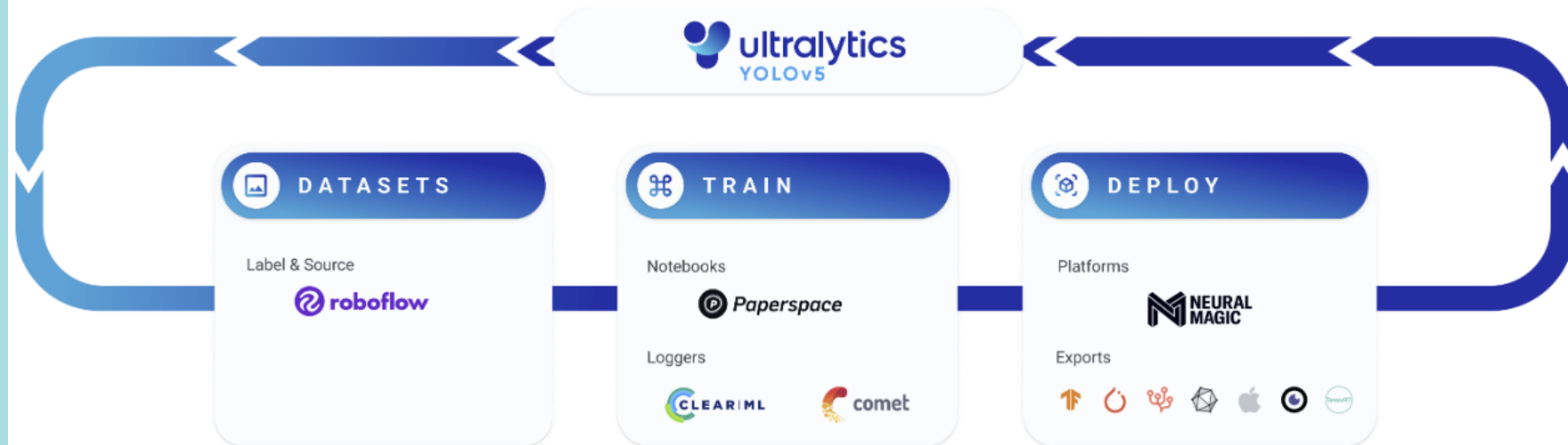
```
[ ] # Download COCO val
    torch.hub.download_url_to_file('https://ultralytics.com/assets/coco2017val.zip', 'tmp.zip')
    !unzip -q tmp.zip -d ../datasets && rm tmp.zip # unzip

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



# 4. Train

## 4. Train



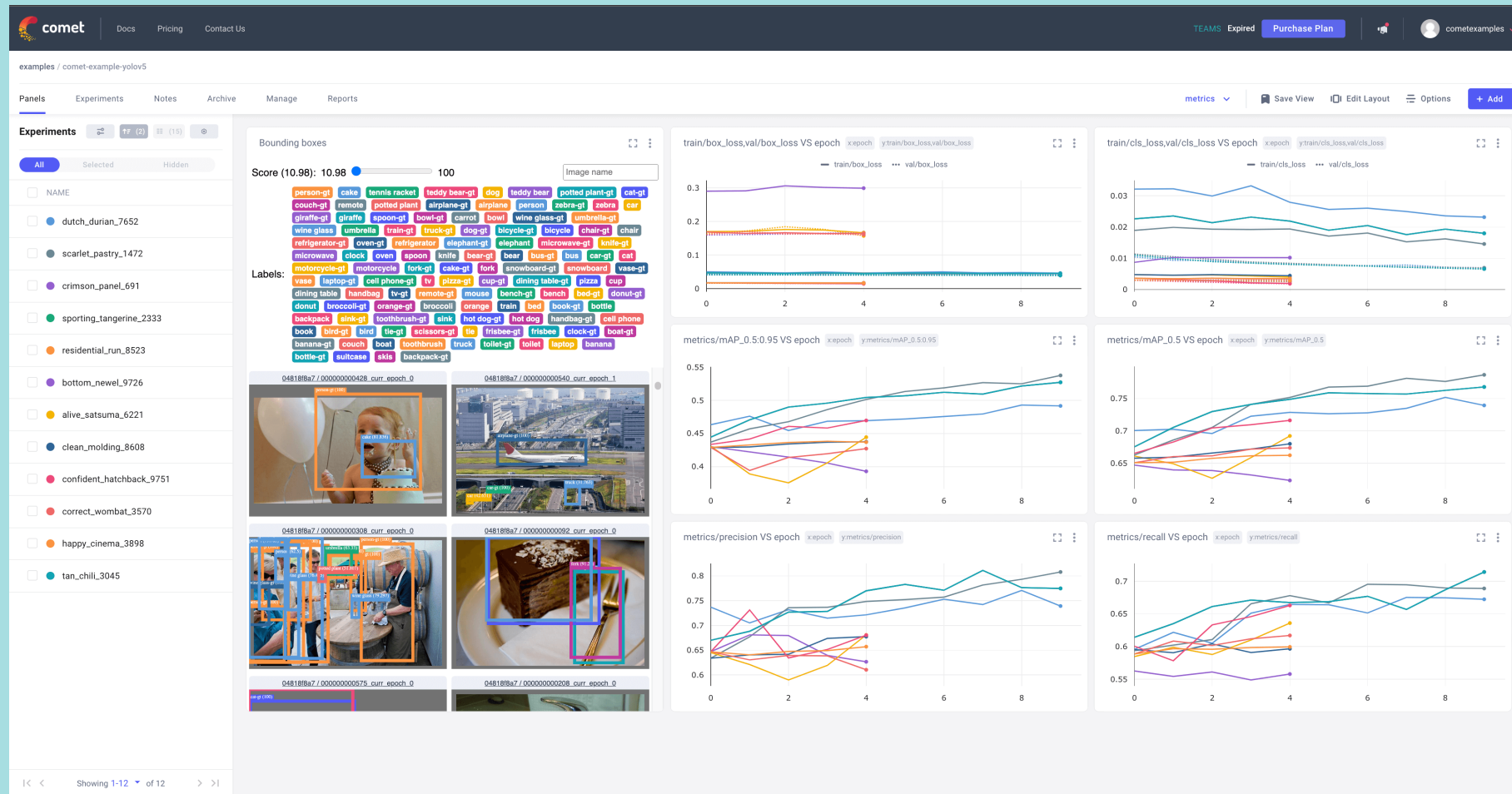
Close the active learning loop by sampling images from your inference conditions with the `roboflow` pip package

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





Thanks!

Q&A

