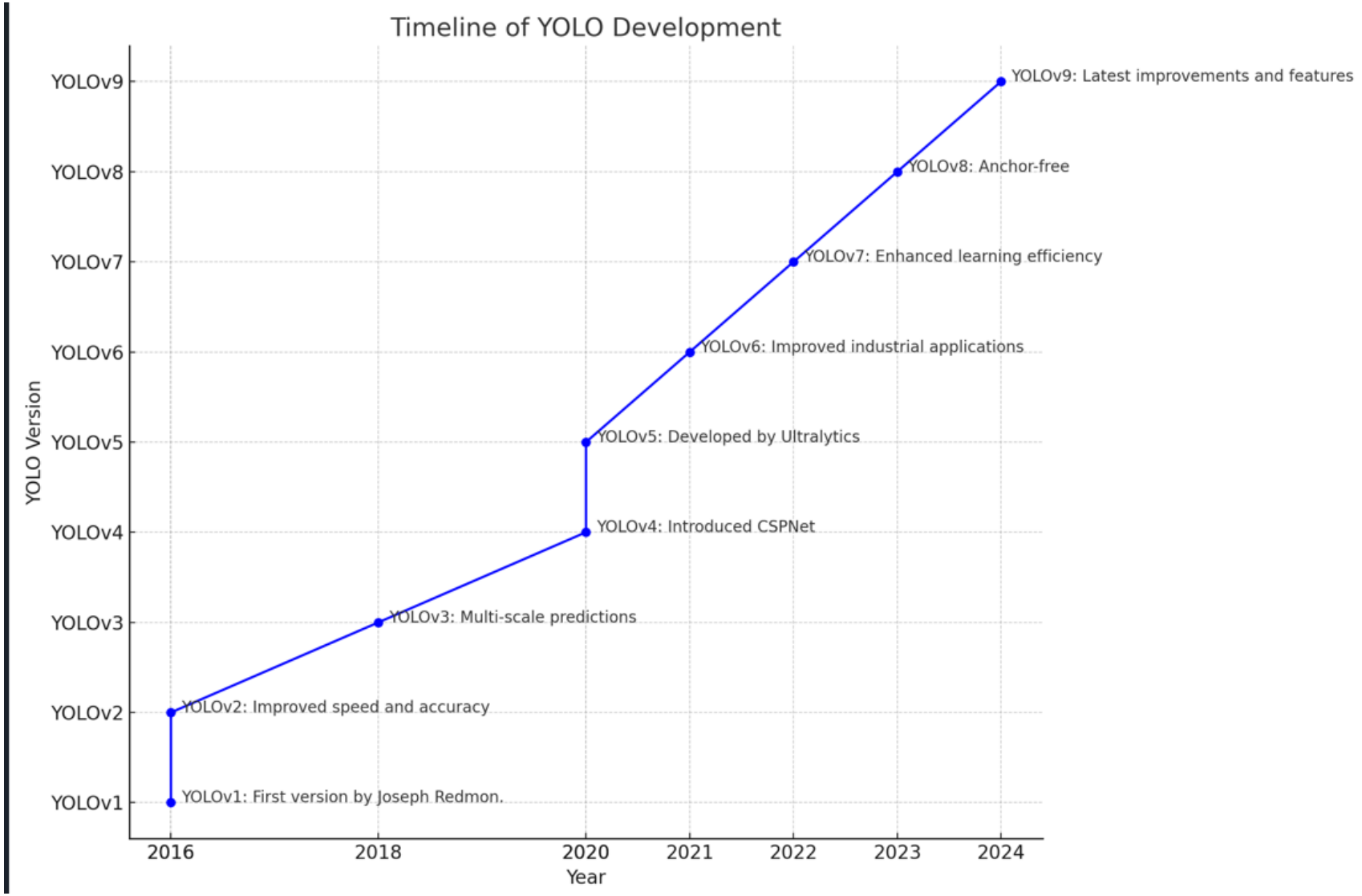
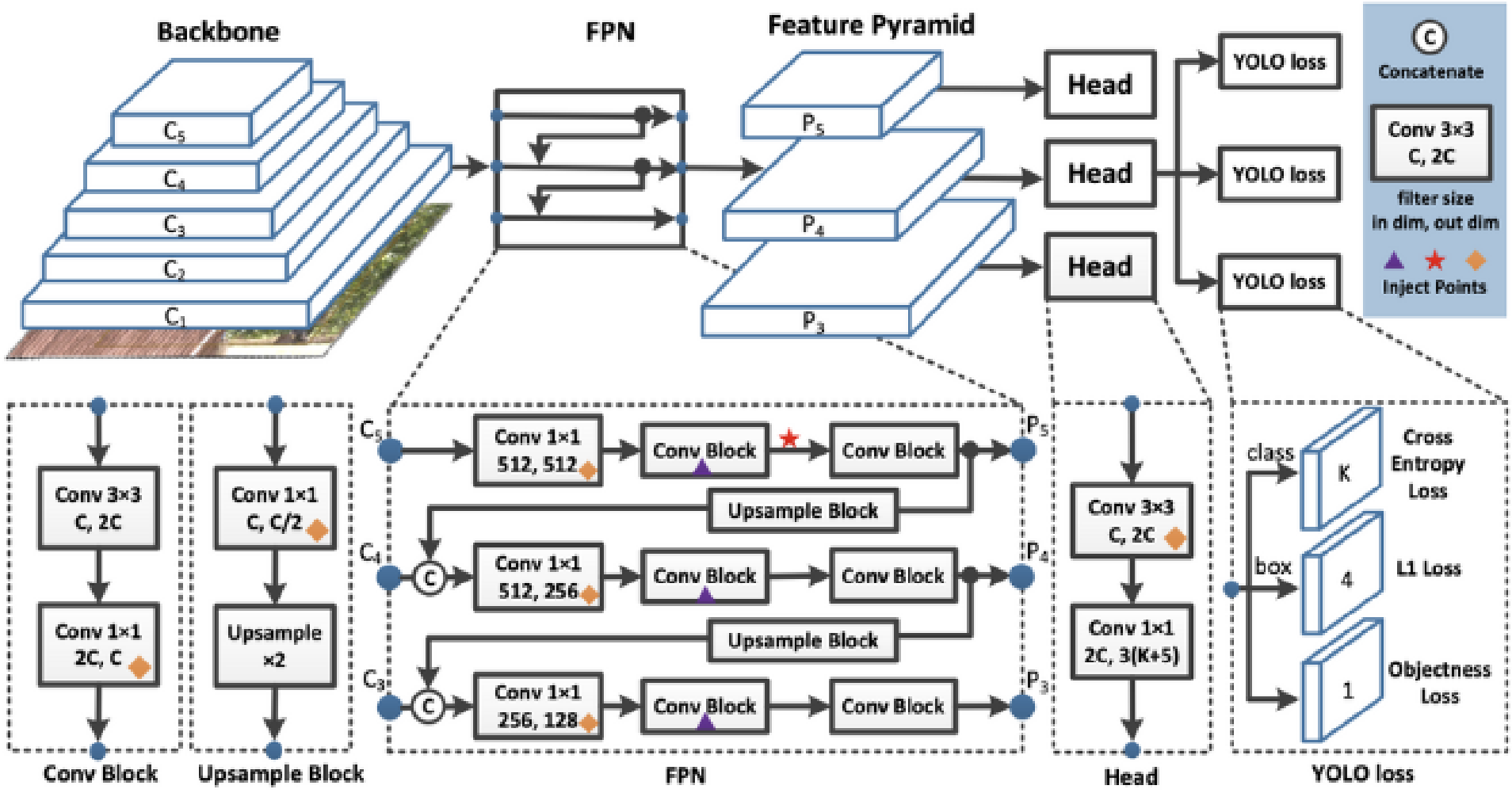
Homework Project :
Object Detection with YOLO

YOLO Models from YOLOv1 to YOLOv9



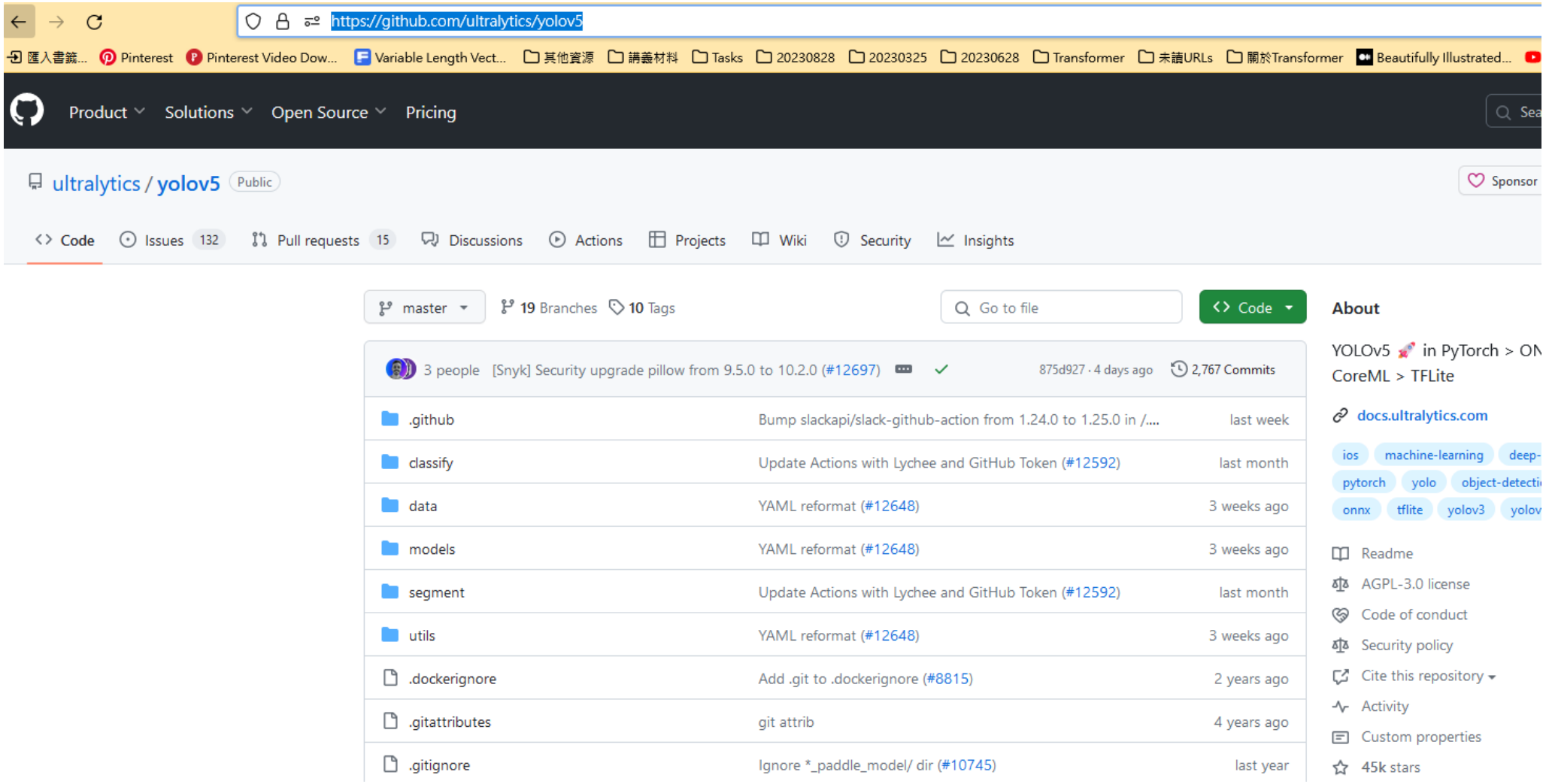


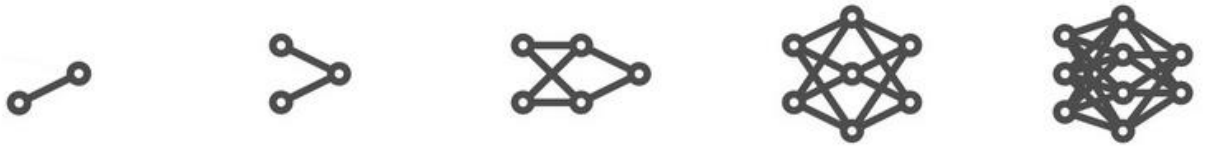
*A graphical depiction of the **PP-YOLO** object detection network*

YOLO Models from YOLOv1 to YOLOv9

- **2015** - **YOLOv1**: First version by Joseph Redmon.
- **2016** - **YOLOv2**: Improved speed and accuracy, introduced anchor boxes.
- **2018** - **YOLOv3**: Multi-scale predictions, Darknet-53 backbone.
- **2020** - **YOLOv4**: Introduced CSPNet, optimized for speed and accuracy.
  2020 - **YOLOv5**: Developed by Ultralytics, focused on usability and deployment.
- **2021** - **YOLOv6**: Improved industrial applications, anchor-free detection.
- **2022** - **YOLOv7**: Enhanced learning efficiency, reduced parameters and computation.
- **2023** - **YOLOv8**: Anchor-free, faster NMS, and semantic segmentation.
- **2024** - **YOLOv9**: Latest improvements and features, pushing the boundaries.

https://github.com/ultralytics/yolov5





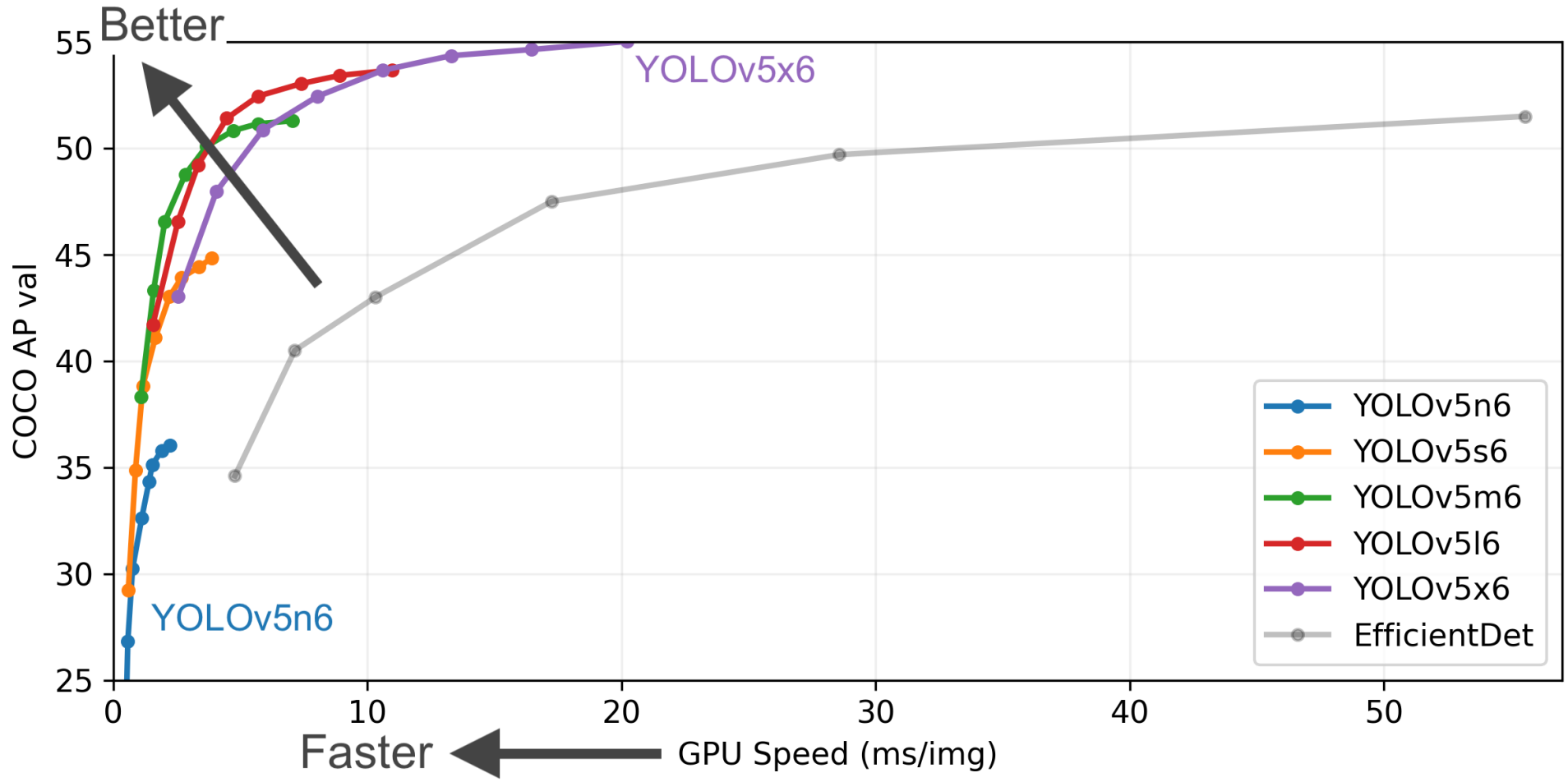
Nano	Small	Medium	Large	XLarge
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
4 MB_{FP16} 6.3 ms_{V100} 28.4 mAP_{COCO}	14 MB_{FP16} 6.4 ms_{V100} 37.2 mAP_{COCO}	41 MB_{FP16} 8.2 ms_{V100} 45.2 mAP_{COCO}	89 MB_{FP16} 10.1 ms_{V100} 48.8 mAP_{COCO}	166 MB_{FP16} 12.1 ms_{V100} 50.7 mAP_{COCO}

Pretrained Checkpoints

Model	size (pixels)	mAP^{val} 50-95	mAP^{val} 50	Speed CPU b1 (ms)	Speed V100 b1 (ms)	Speed V100 b32 (ms)	params (M)	FLOPs @640 (B)
YOLOv5n	640	28.0	45.7	**45**	**6.3**	**0.6**	**1.9**	**4.5**
YOLOv5s	640	37.4	56.8	98	6.4	0.9	7.2	16.5
YOLOv5m	640	45.4	64.1	224	8.2	1.7	21.2	49.0
YOLOv5l	640	49.0	67.3	430	10.1	2.7	46.5	109.1
YOLOv5x	640	50.7	68.9	766	12.1	4.8	86.7	205.7
YOLOv5n6	1280	36.0	54.4	153	8.1	2.1	3.2	4.6
YOLOv5s6	1280	44.8	63.7	385	8.2	3.6	12.6	16.8
YOLOv5m6	1280	51.3	69.3	887	11.1	6.8	35.7	50.0
YOLOv5l6	1280	53.7	71.3	1784	15.8	10.5	76.8	111.4
YOLOv5x6 + TTA	1280 1536	55.0 **55.8**	72.7 **72.7**	3136 -	26.2 -	19.4 -	140.7 -	209.8 -

https://docs.ultralytics.com/zh/yolov5/tutorials/train_custom_data/

Why YOLOv5


Better
55
50
45
40
35
30
25
COCO AP val
YOLOv5x6
YOLOv5n6
YOLOv5n6
YOLOv5s6
YOLOv5m6
YOLOv5l6
YOLOv5x6
EfficientDet
0
10
20
30
40
50
Faster
GPU Speed (ms/img)

Quick Start: Setup Environment

- Create & activate conda environment for yolov5

```
> conda create -n yolov5 python=3.8
> conda activate yolov5
```

- Clone yolov5(https://github.com/ultralytics/yolov5.git)

```
> git clone https://github.com/ultralytics/yolov5.git
> cd yolov5
```

- Install requirements

```
> pip install -r yolov5/requirements.txt
```

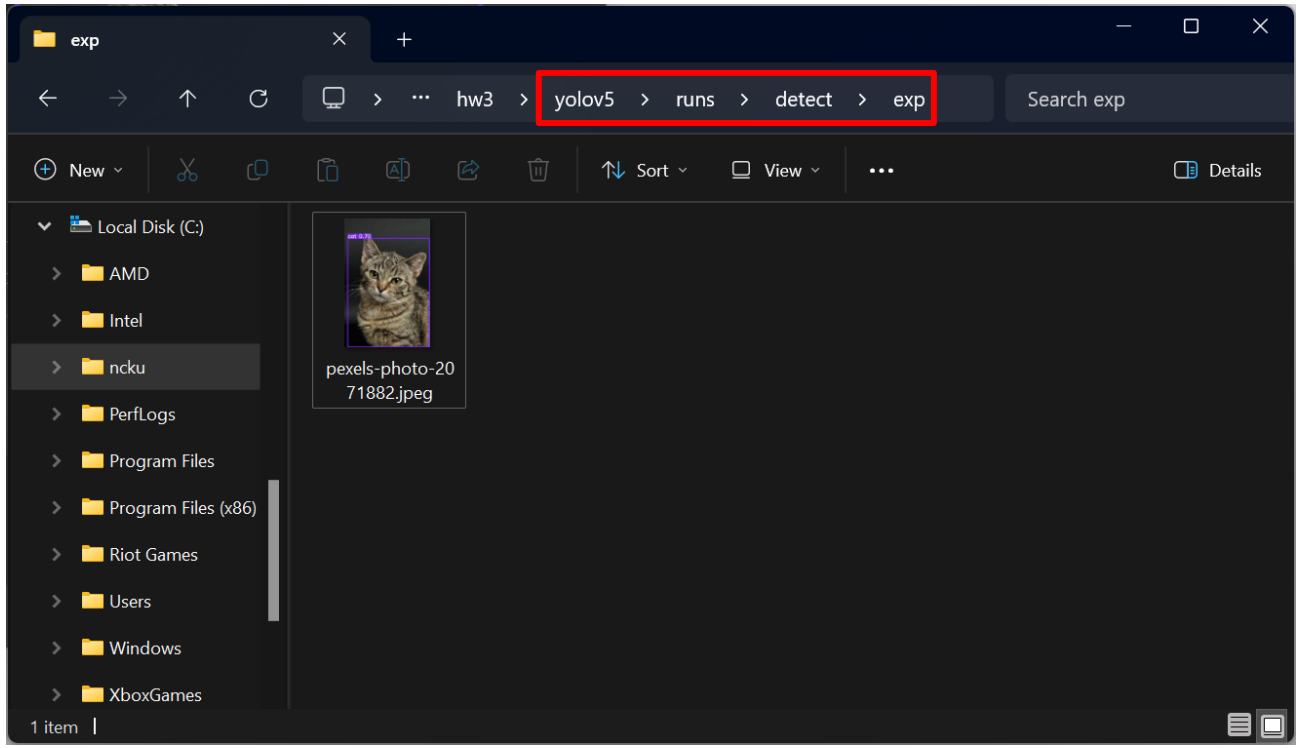
Quick Start: Inference

- Inference with detect.py
  - detect.py runs inference on a variety of sources, downloading models automatically from the latest YOLOv5 release and saving results to runs/detect.

```
> python detect.py --weights yolov5s.pt --source 0                               # webcam
                                                 img.jpg                         # image
                                                 vid.mp4                         # video
                                                 screen                          # screenshot
                                                 path/                           # directory
                                                 list.txt                        # list of images
                                                 list.streams                    # list of streams
                                                 'path/*.jpg'                    # glob
                                                 'https://youtu.be/LNwODJXcvt4'  # YouTube
```

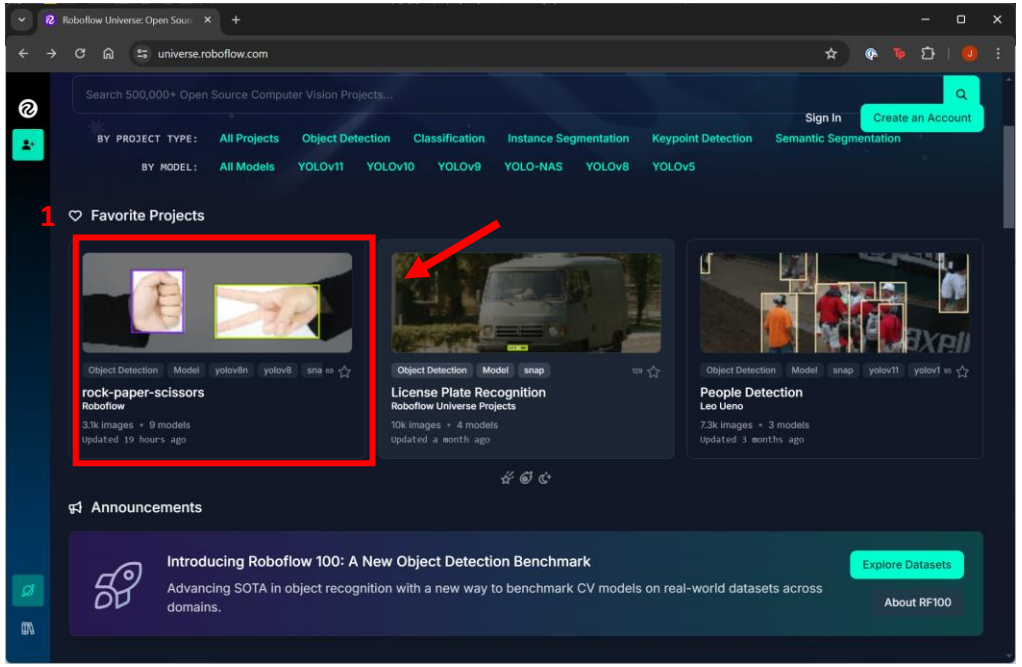
Quick Start: Inference

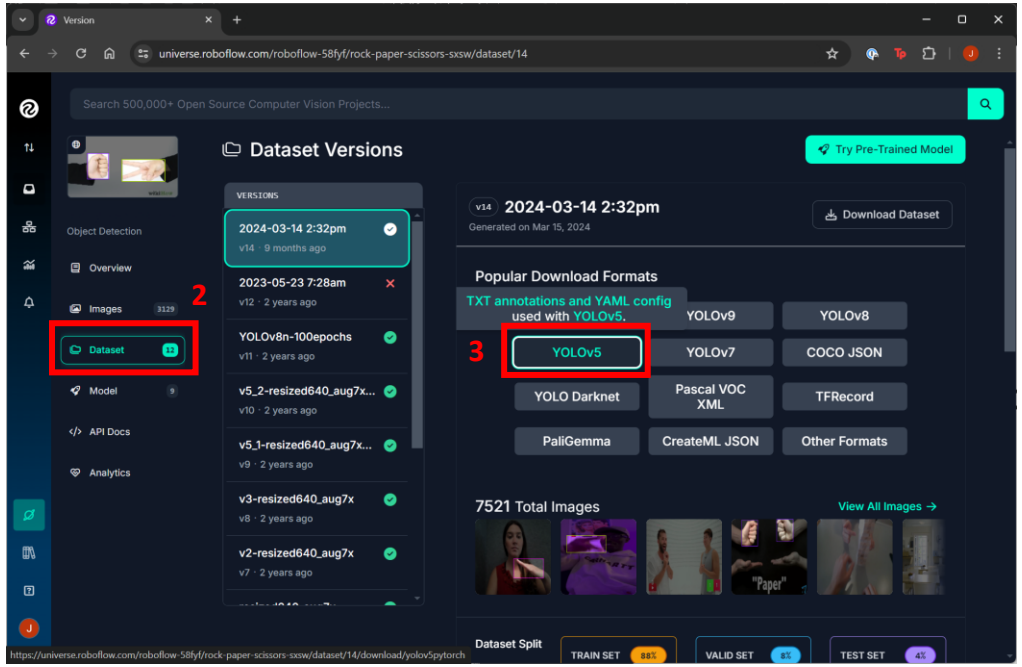
- The inference result should be under the path yolov5/run/detect/exp*



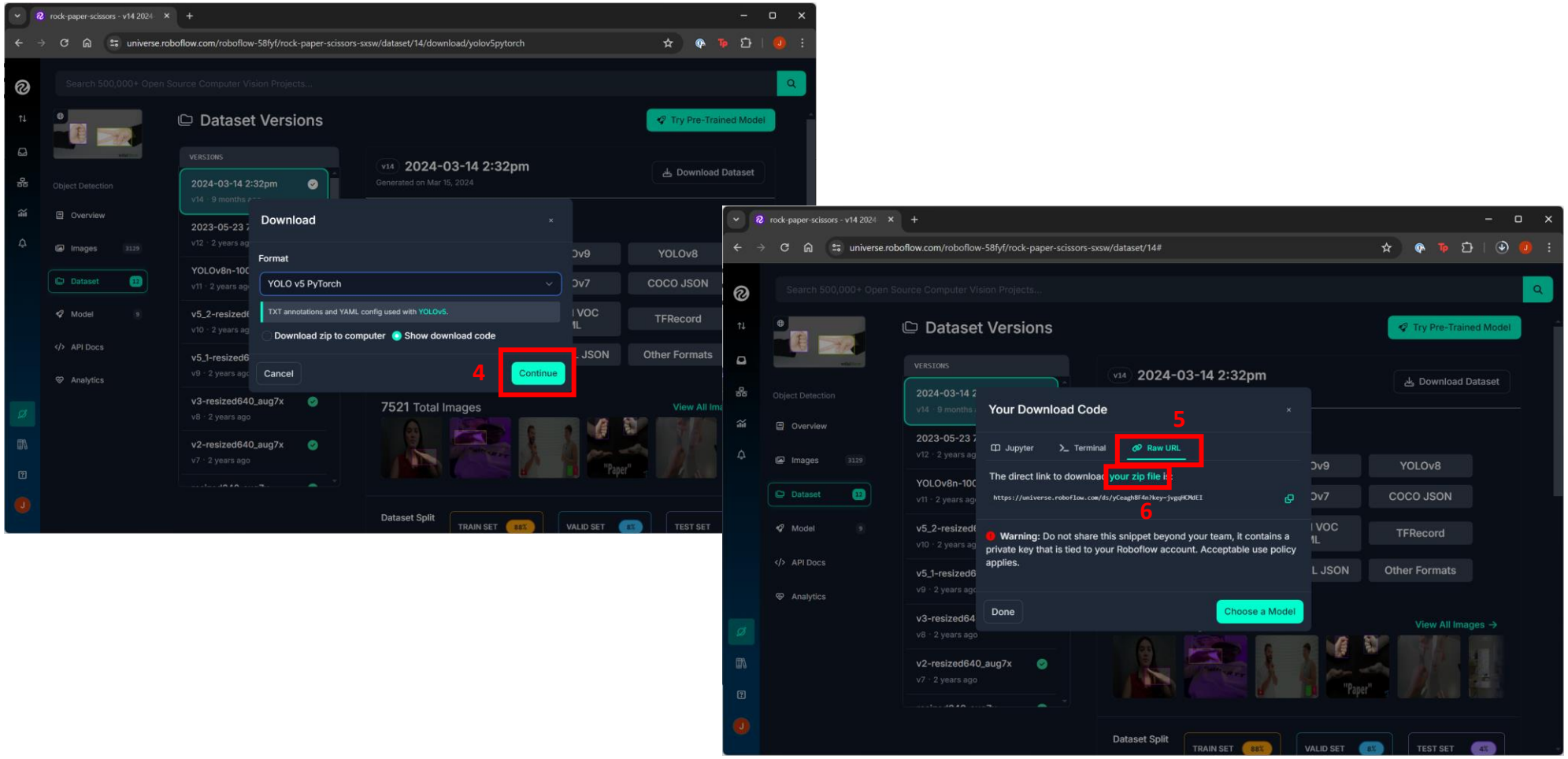
Tutorial: Download Custom Data

- A huge amount of objection detection dataset are available in RoboFlow Universe: https://universe.roboflow.com/
- Let's use rock-paper-scissors as an example(make sure to sign-in)





Tutorial: Download Custom Data



Tutorial: Download Custom Data

- Unzip the downloaded file & move its content to the ***yolov5*** folder.
- Your project should have the following structure in the end.
  - HW3
    - yolov5
      - test/
      - train/
      - valid/
      - data.yaml
      - …
    - …

Tutorial: Train w/ Custom Data

- Run the following command to train yolo5s on our custom data. (feel free to modify the parameters highlighted in red)

```
> python train.py --name custom --data data.yaml --weights yolov5s.pt --epochs 3 --batch-size 16
```

- See the ***parse opt*** function in train.py for details of other hyper parameters.
- The trained model weight should be saved under runs/train/custom*/weights/best.pt **(* is a number depends on how many time you've run train.py before)**

Tutorial: Validation

- We can run validate our model's performance with *val.py*:

```
  # Run validation on custom model
> python val.py --name custom --data data.yaml --weights runs/train/custom*/weights/best.pt --task test --augment
```

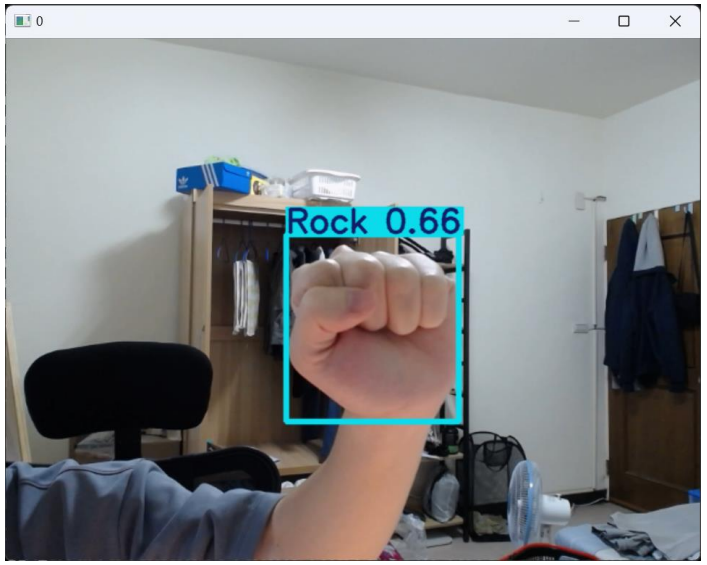
- The result should look like below:

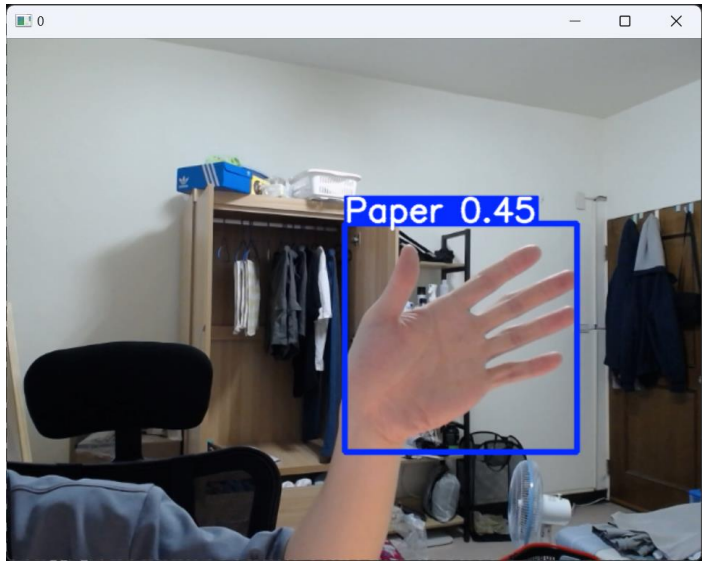
```
Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs
test: Scanning C:\ncku\hw3\yolov5\test\labels.cache... 304 images, 118 backgrounds, 0 corrupt: 100%|██████████| 304/304 [00:00<?, ?it/s]
                 Class     Images  Instances          P          R      mAP50   mAP50-95: 100%|██████████| 10/10 [00:28<00:00,  2.82s/it]
                   all        304        204       0.77      0.811      0.853      0.531
                 Paper        304         72      0.718      0.736      0.794      0.441
                  Rock        304         65      0.813      0.862      0.891      0.566
              Scissors        304         67      0.779      0.836      0.875      0.585
Speed: 0.5ms pre-process, 90.6ms inference, 0.4ms NMS per image at shape (32, 3, 640, 640)
Results saved to runs\val\custom
```

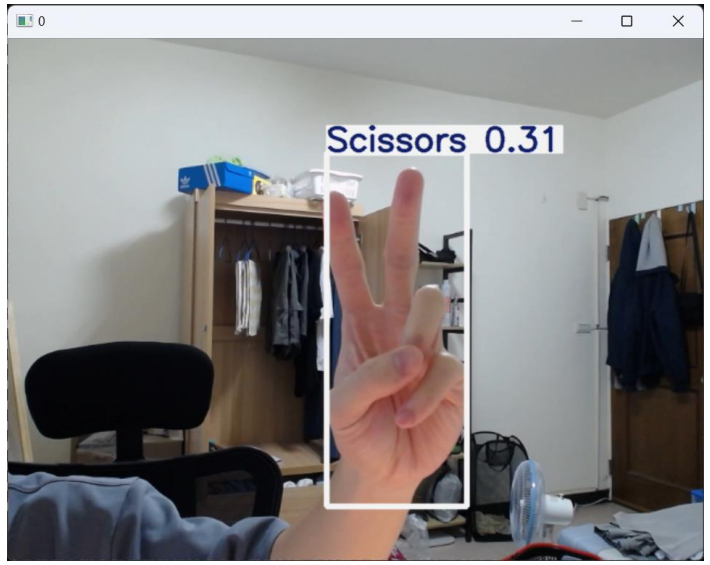
Tutorial: Run the Model

- Finally, we can run our model with *detect.py*:

```
  # Inference with webcam
> python detect.py --weights runs/train/custom*/weights/best.pt --source 0
```







Objective

- For this homework, you need to follow the tutorial above and prepare a report in the form of ppt or pdf detailing each steps.
  - Collect custom data from internet, ex. Roboflow Universe
  - Train yolov5 model with custom data
  - Validate the model
  - Run inference