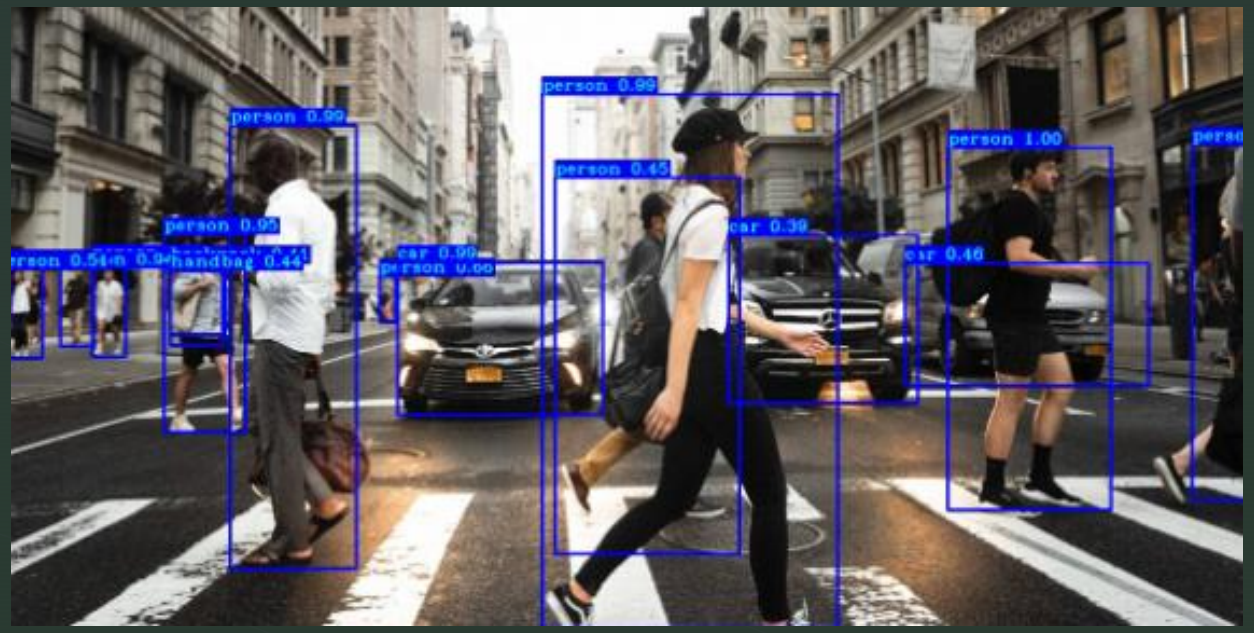
YOLO: Real-Time Object Detection



Under supervision of Prof.Kalhor

By: Mohammadali Shakerdargah

Fall 2021

Outlines

- Definition of Object-Detection
- History of Object-Detection
- YOLO
- How to train with YOLOv5
- Examples

Definition of Object-Detection

- Object-Detection definition:
  - Object detection is a phenomenon in computer vision that involves the detection of various objects in digital images or videos.
  - Object detection combines the tasks of object classification and localization.
- Object-Detection Categories:
  - Two stage detectors: Networks separating the tasks of determining the location of objects and their classification.
    - Faster R-CNN, G-RCNN
  - One stage detectors: Networks which predict bounding boxes and class scores at once.
    - YOLO, SSD network, RetinaNet

Definition of Object-Detection

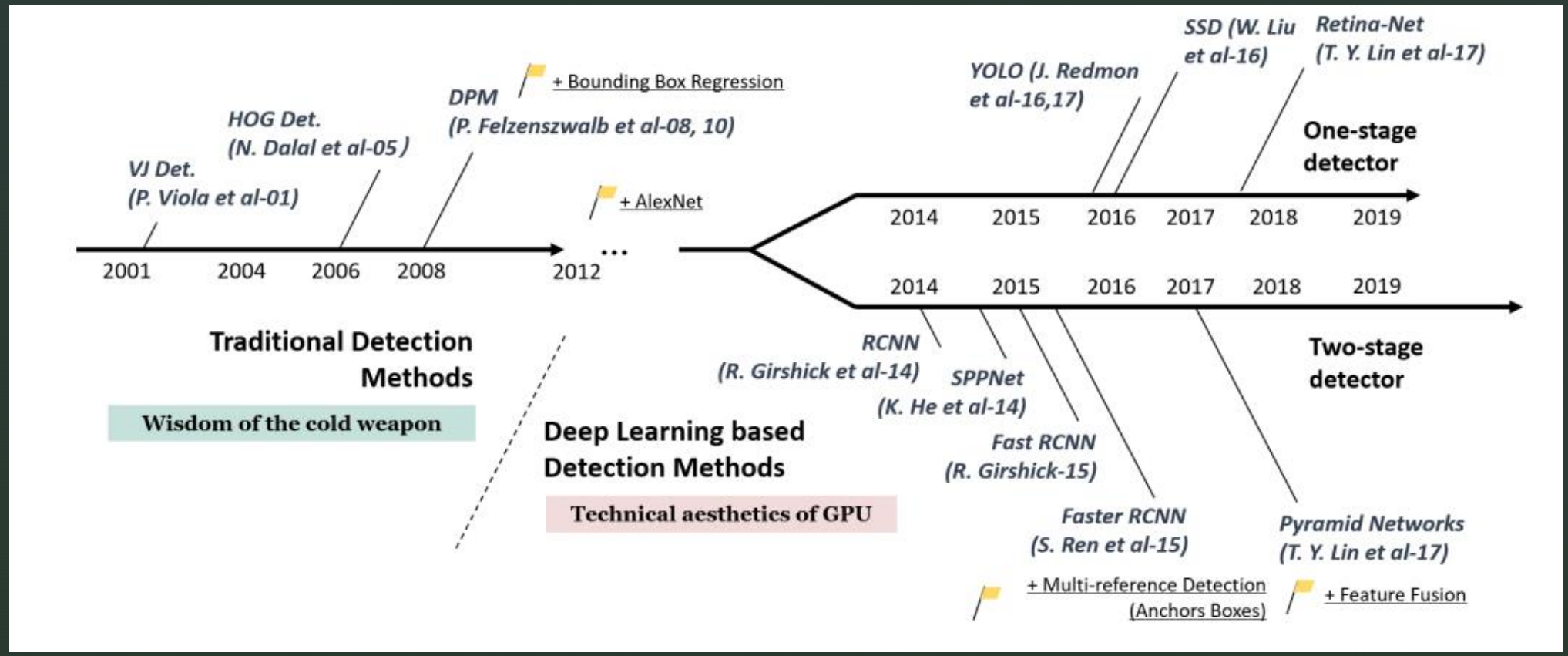
- Two stage detectors :
  - Involves
    - Object region proposal
    - Object classification based on features extracted from the proposed region
  - Achieve the highest detection accuracy but are typically slower
    - Total low performance
  - Usually not end-to-end trainable
  - Famous Networks:
    - Region convolutional neural network (RCNN)
    - Faster R-CNN
    - Mask R-CNN.
    - Granulated RCNN (G-RCNN)

Definition of Object-Detection

- One stage detectors :
  - Prioritize inference speed and are super fast but not as good at recognizing irregularly shaped objects or a group of small objects – Only advantage over multi-stage detectors.

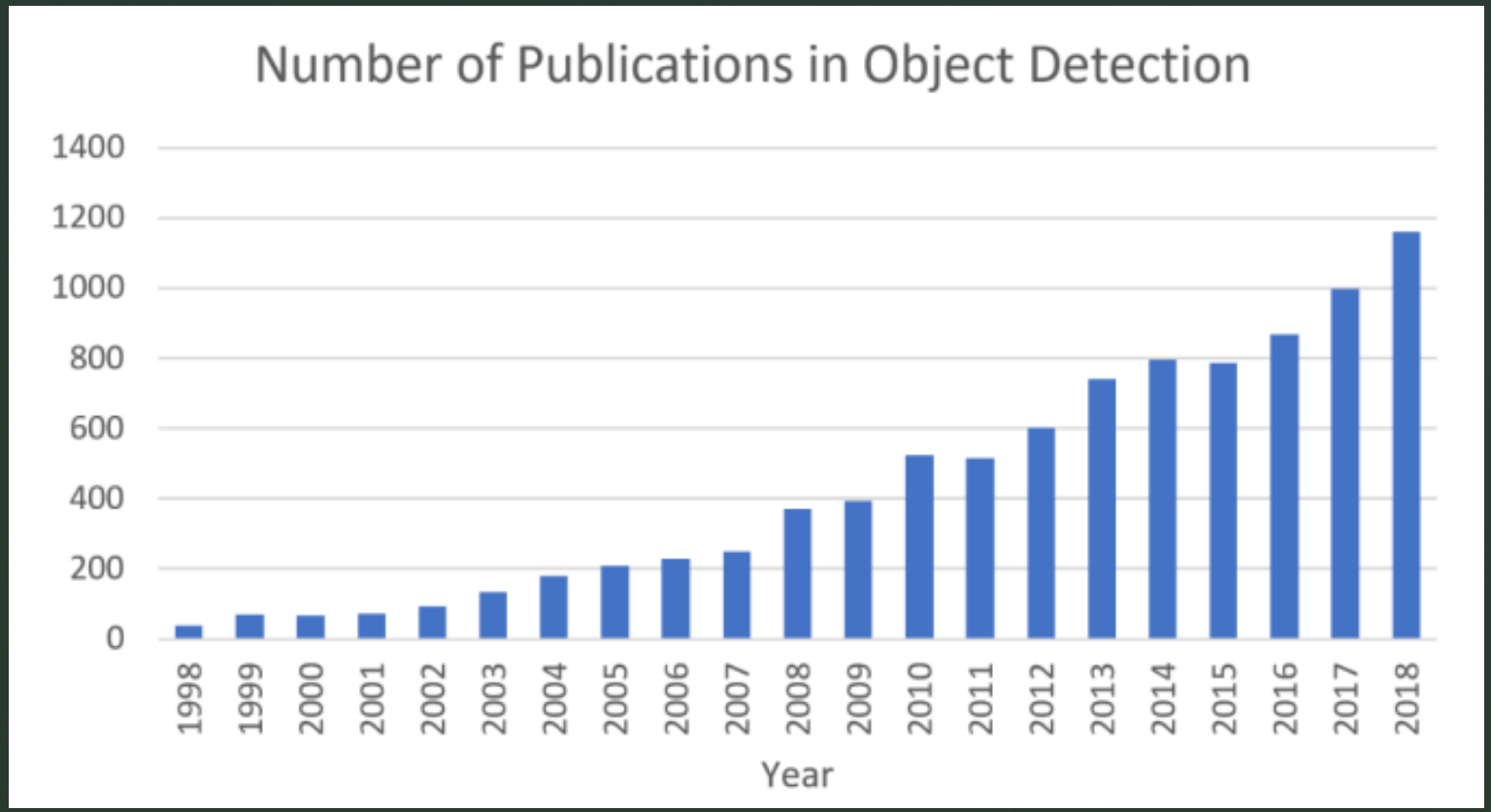
  - Famous Networks:
    - YOLO
    - SSD
    - RetinaNet

History of Object Detection



History of Object Detection

- Object detection is becoming a big deal.



YOLO

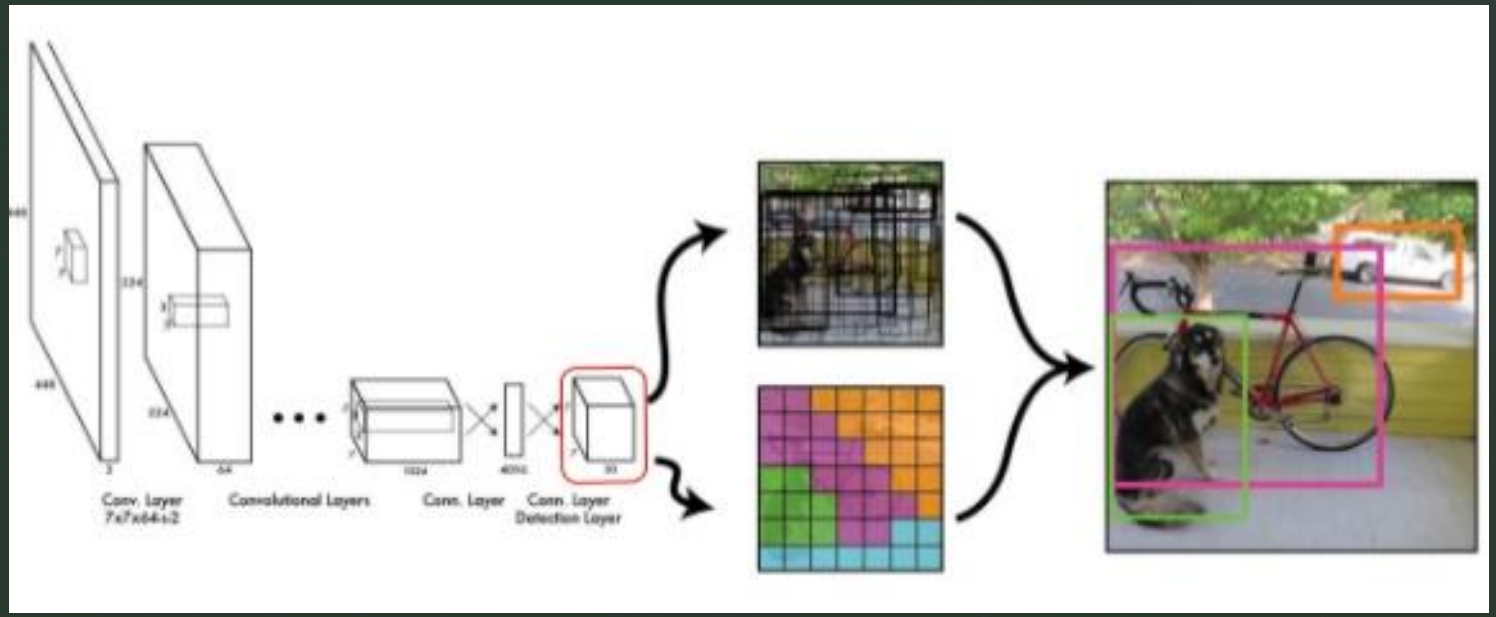
- YOLO version:
  - Version 1, by Joseph Redmon, 2016
    - You Only Look Once: Unified, Real-time Object Detection
  - Version 2, by Joseph Redmon, 2017
    - YOLO9000
    - Better, Faster, Stronger
  - Version 3, by Joseph Redmon, 2018
    - An incremental improvement

YOLO

- YOLO version:
  - Version 4, by Alexey Bochkovskiy, 2020
    - Optimal speed and accuracy of object detection
  - Version 5, by Glenn Jocher, 2020
    - A huge advancement in time consumption
    - Better accuracy

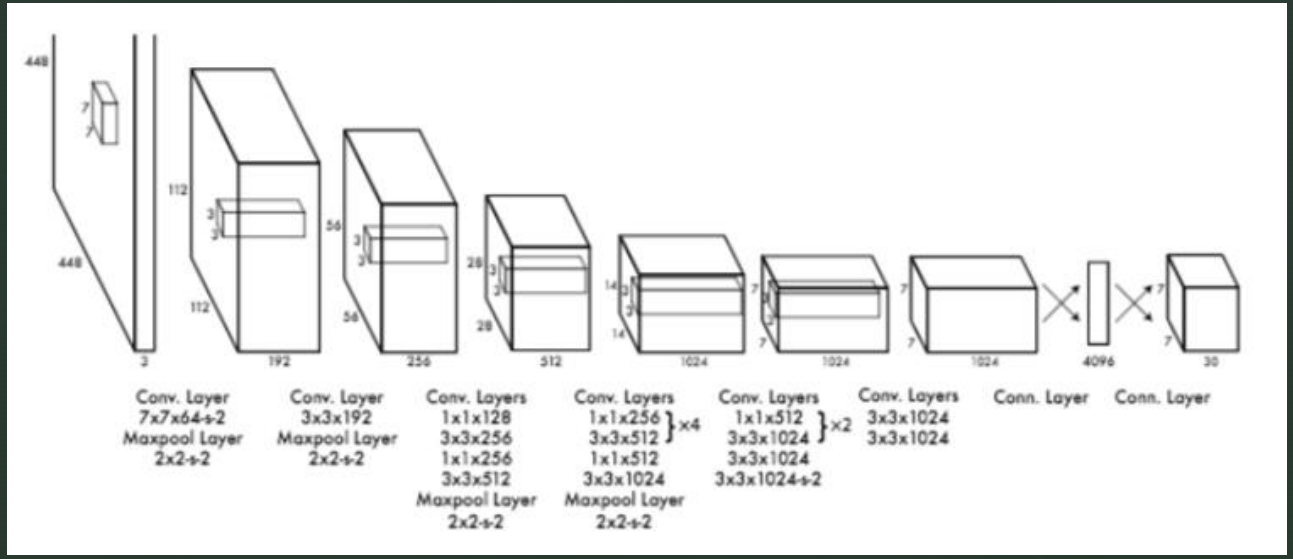
YOLO

- YOLO version1:
  - Introduced in 2015-2016 by Joseph Redmon
  - Its algorithm is based on regression
  - Fast enough for real-time systems.



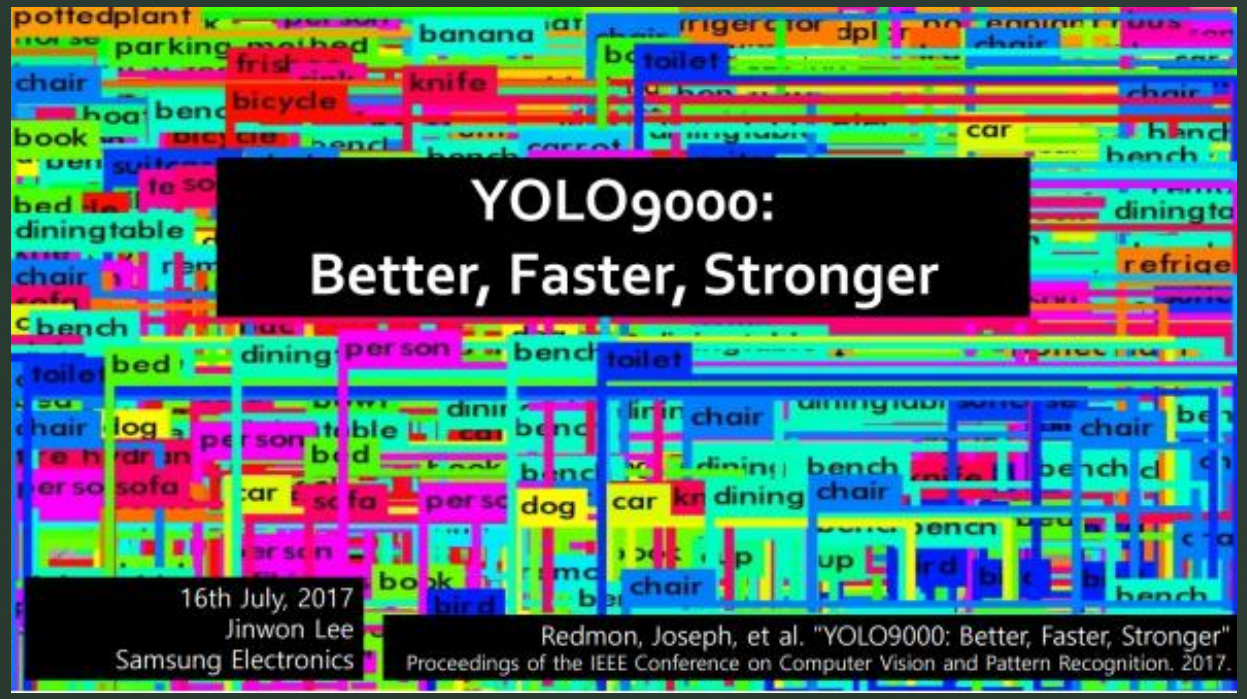
YOLO

- YOLO version1 Design:
  - YOLO is implemented as a convolution neural network and has been evaluated on the PASCAL VOC detection dataset. It consists of a total of 24 convolutional layers followed by 2 fully connected layers.



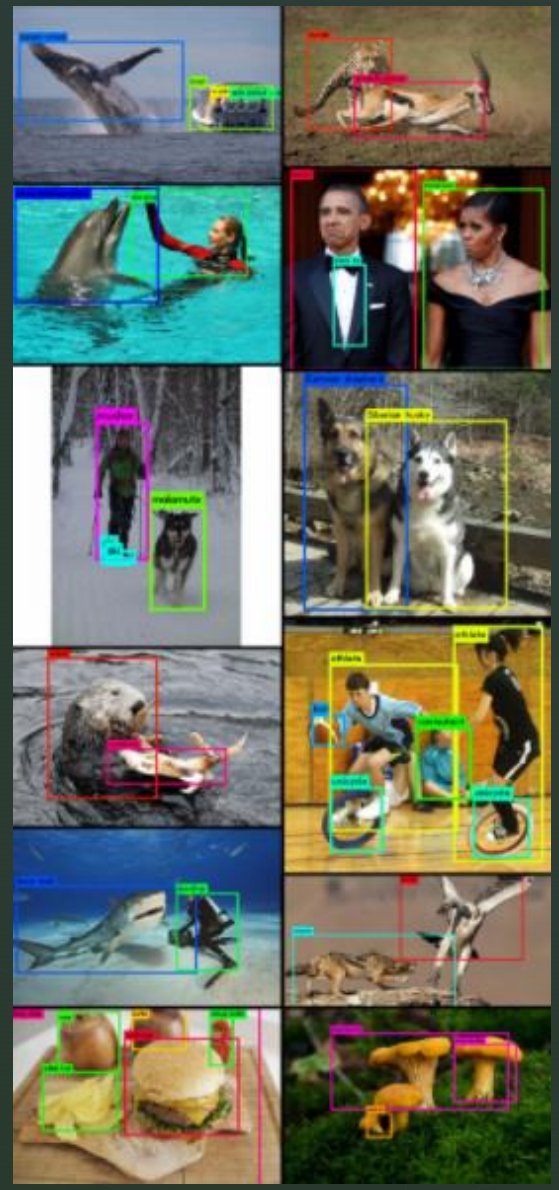
YOLO

- YOLO version2:
  - Why 9000?
  - Why Better?
  - Why Faster?
  - Why Stronger?



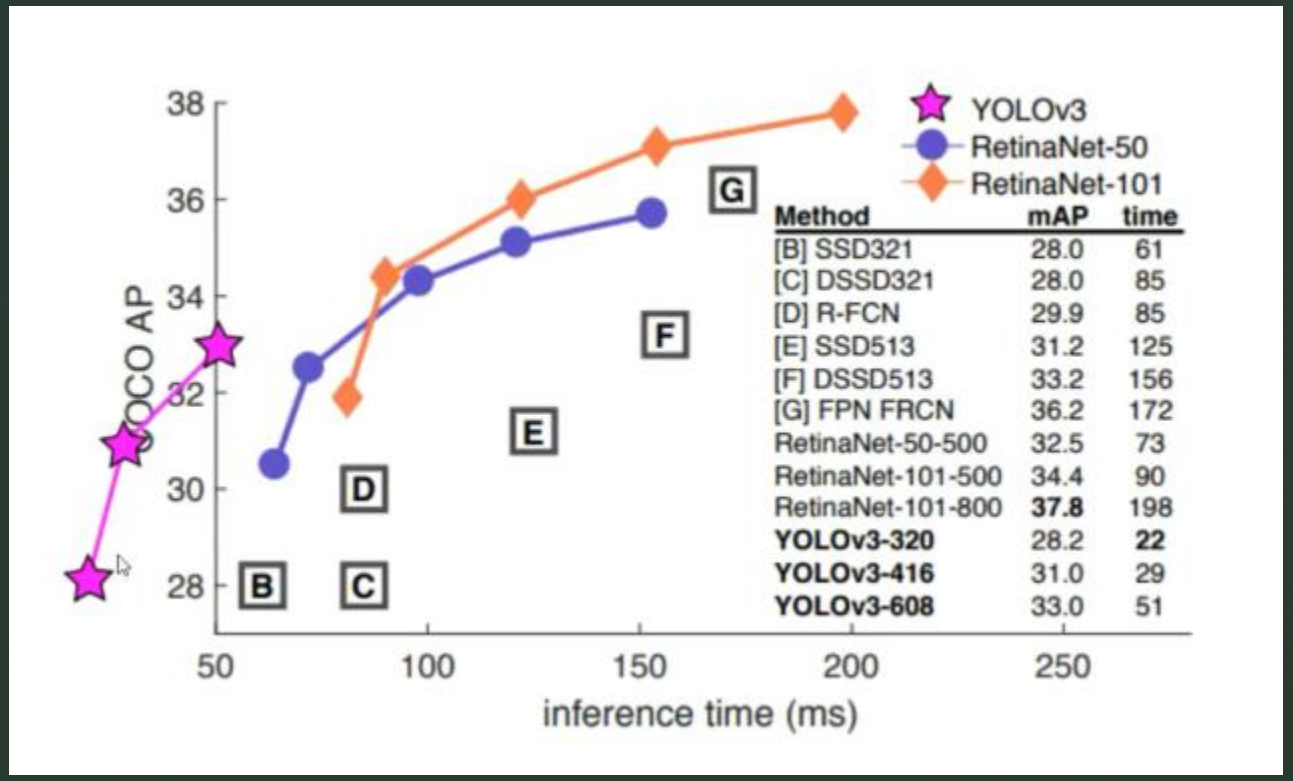
YOLO

- YOLO version2:
  - It is an enhanced version of the YOLO with the objective of improving the accuracy significantly while making it faster.
  - YOLO9000 is built on top of YOLOv2 but trained with joint dataset combining the COCO detection dataset and the top 9000 classes from ImageNet.



YOLO

- YOLO version3: An incremental improvement



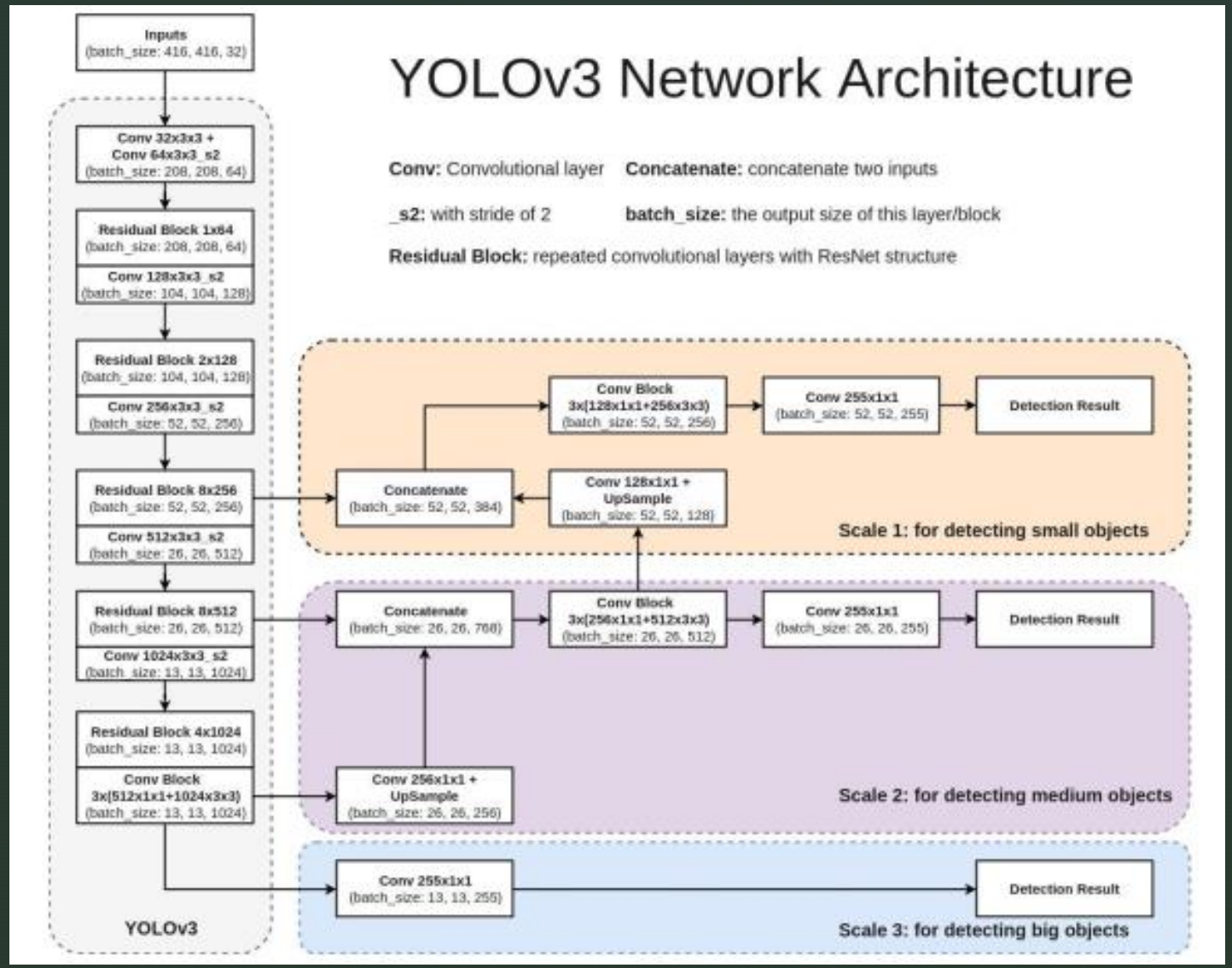
Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

YOLO

- YOLO version3:
  - multilabel classification
  - No Softmax
  - logistic regression
  - cross-entropy loss for each label

YOLO

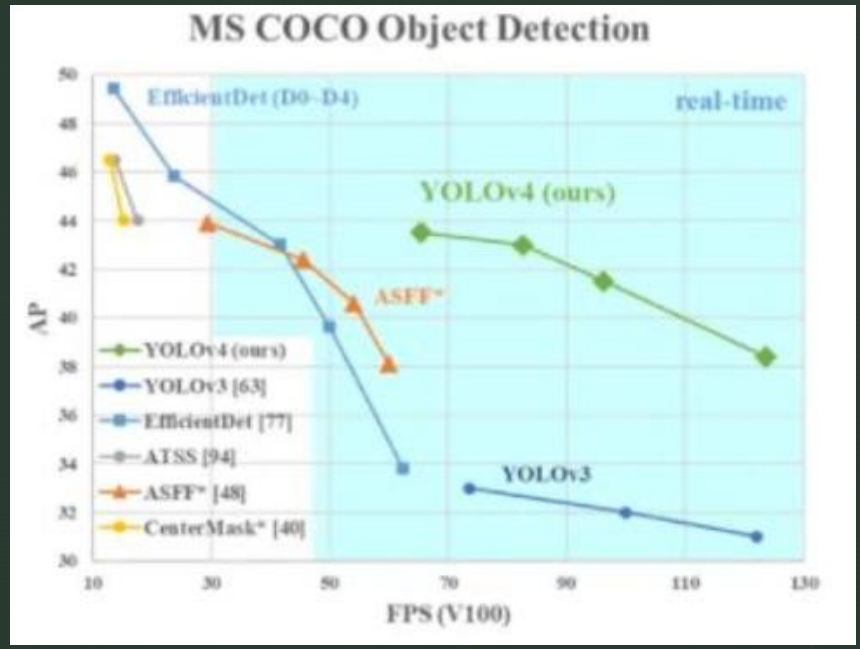
- YOLO version3:



YOLO

- YOLOv4 and YOLOv5:
  - Introduced by
    - Alexey Bochkovskiy
    - Glenn Joche

- YOLOv4:
  - Train
  - Bag-of-freebies
  - Bag-of-specials

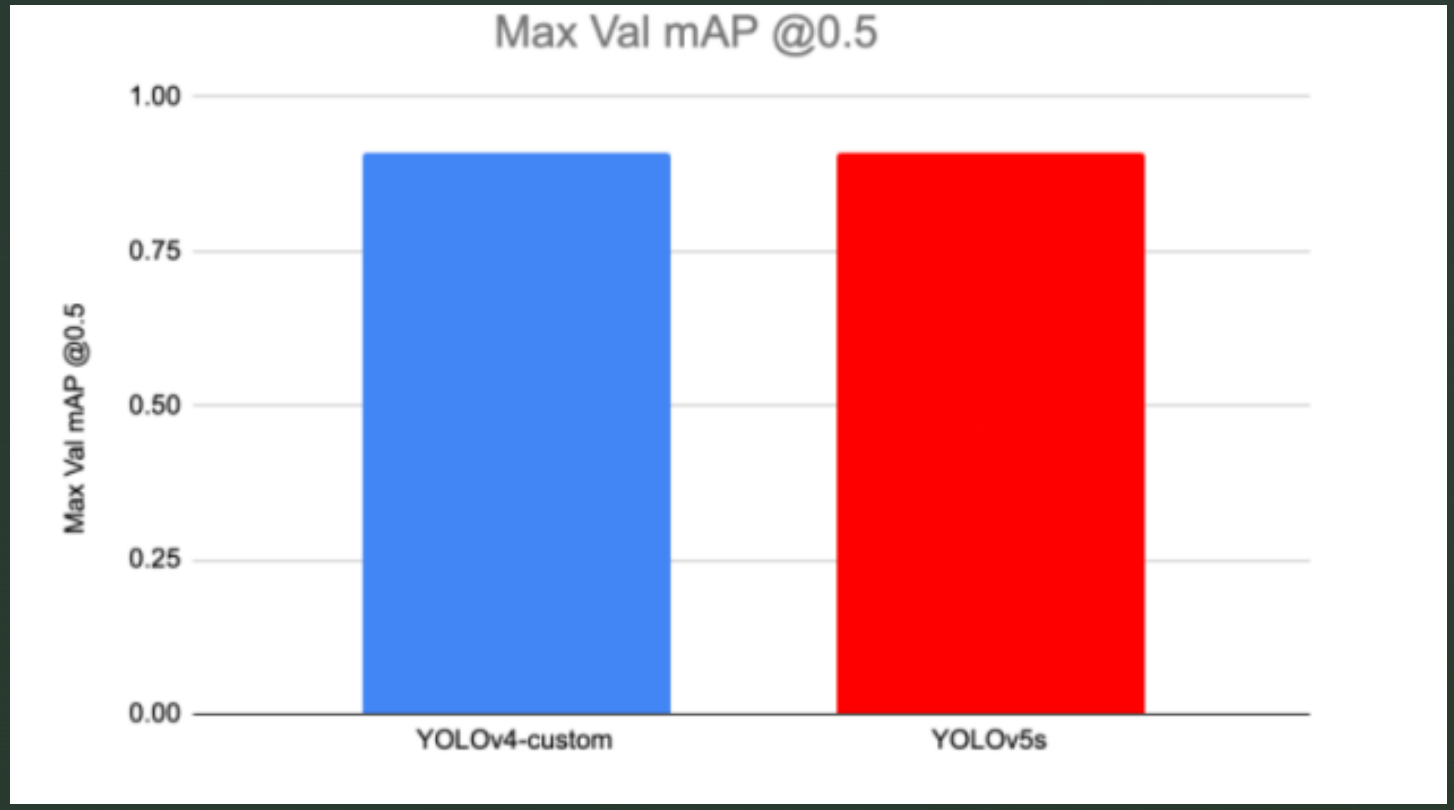


YOLO

- YOLOv4 and YOLOv5:
  - Time: YOLOv5 is so much faster

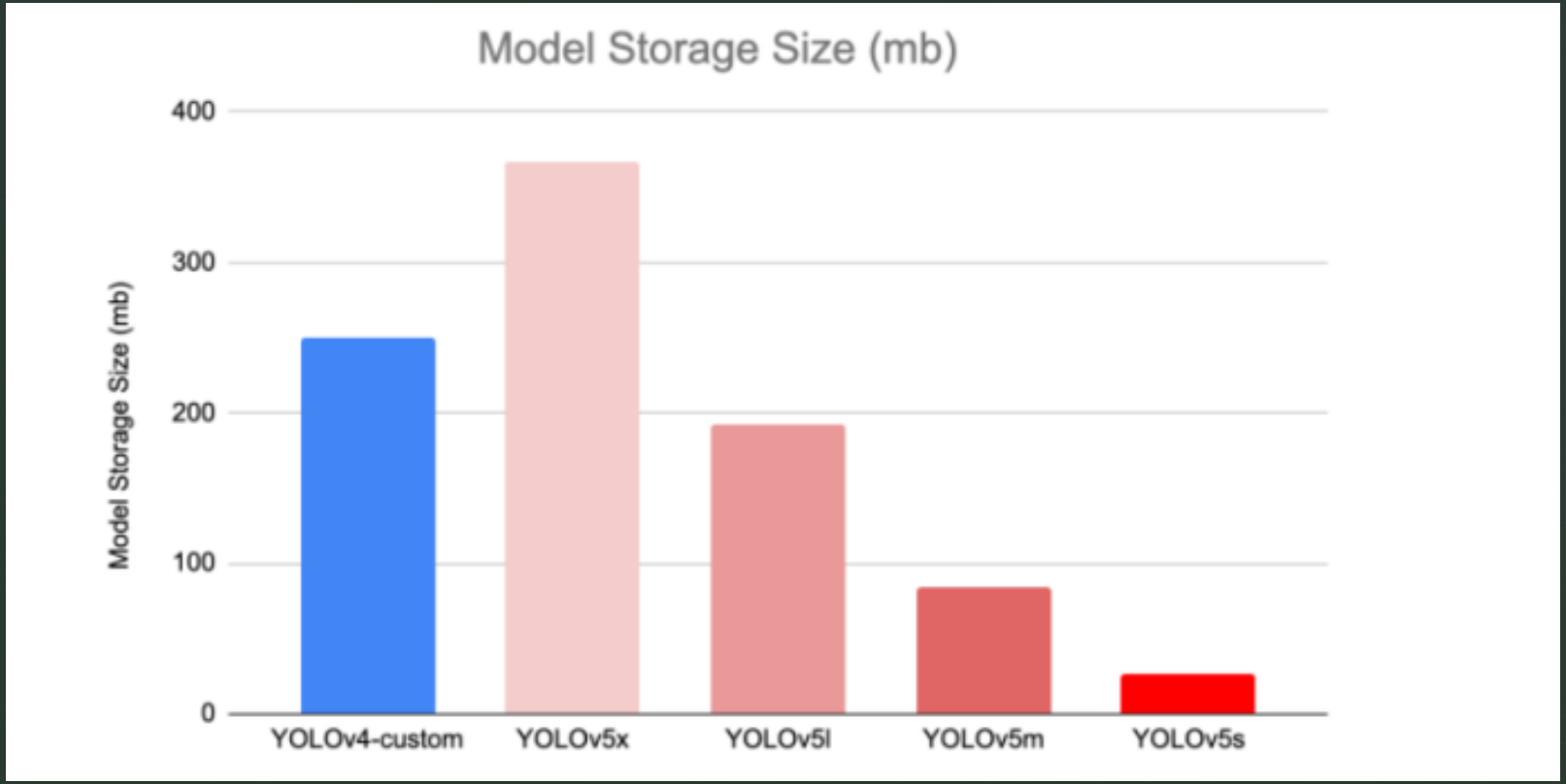
YOLO

- YOLOv4 and YOLOv5:
  - Accuracy: Almost the same.



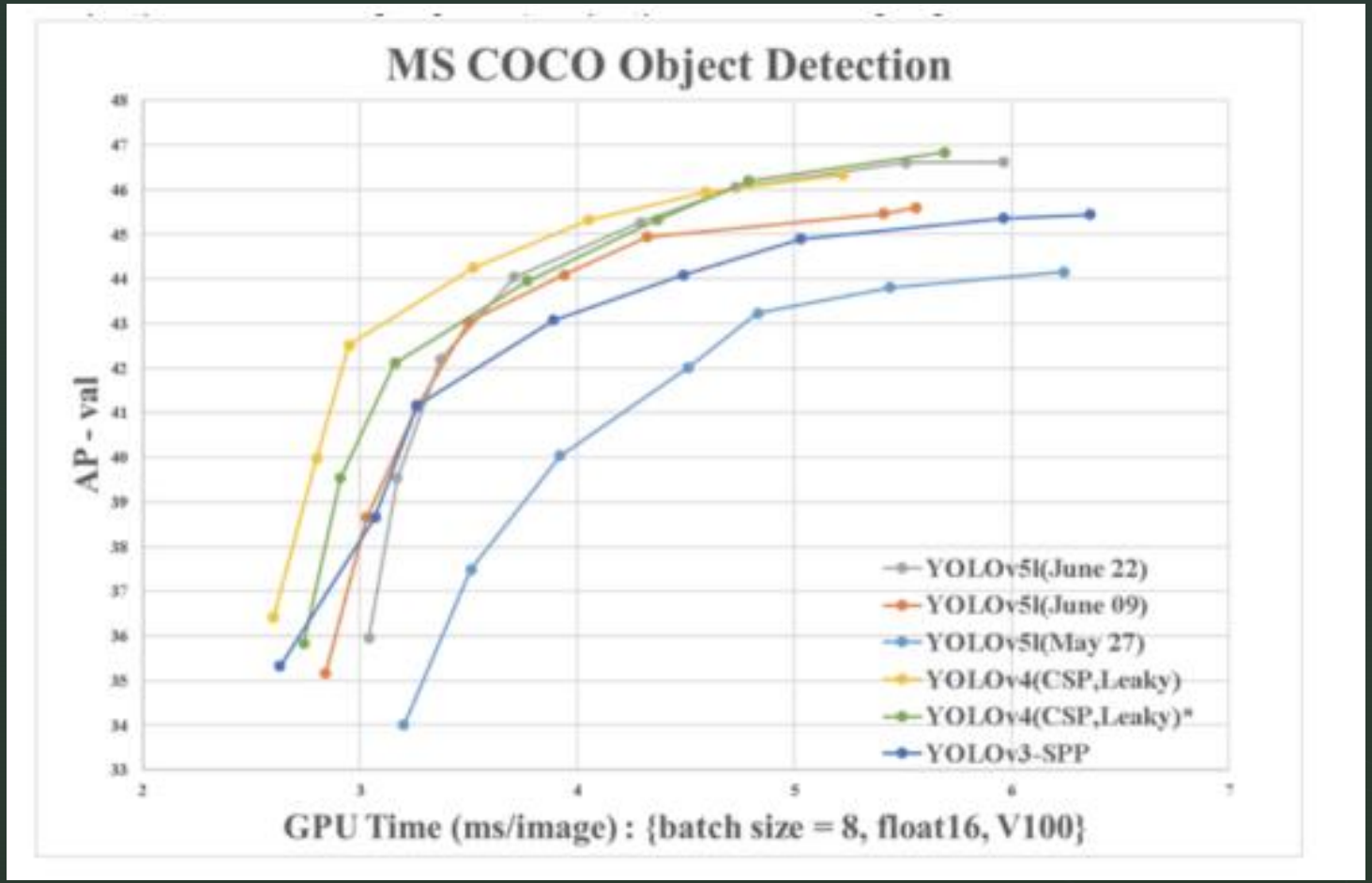
YOLO

- YOLOv4 and YOLOv5:
  - Space usage: YOLOv5 is so much faster



YOLO

- YOLOv4 and YOLOv5 on MS COCO:



How to train with YOLOv5

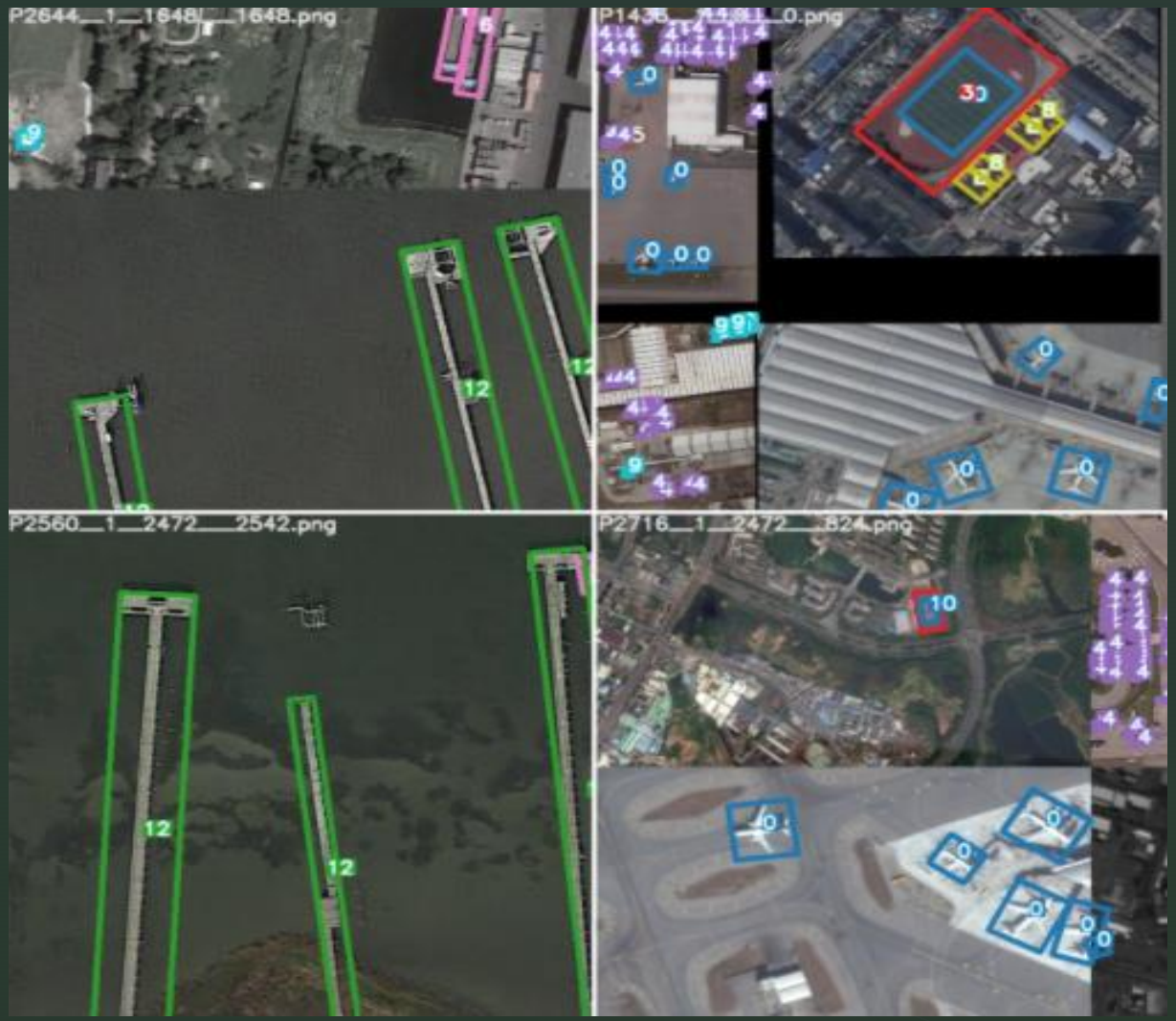
- Install Dependencies and import libraries
- Load data | Roboflow
- Model configuration and architecture
- Train
  - Img
  - Batch
  - Epoch
  - Data
  - Cfg
  - Weights
  - Name
  - No-save
  - Chache
- Test

Checkout: Link

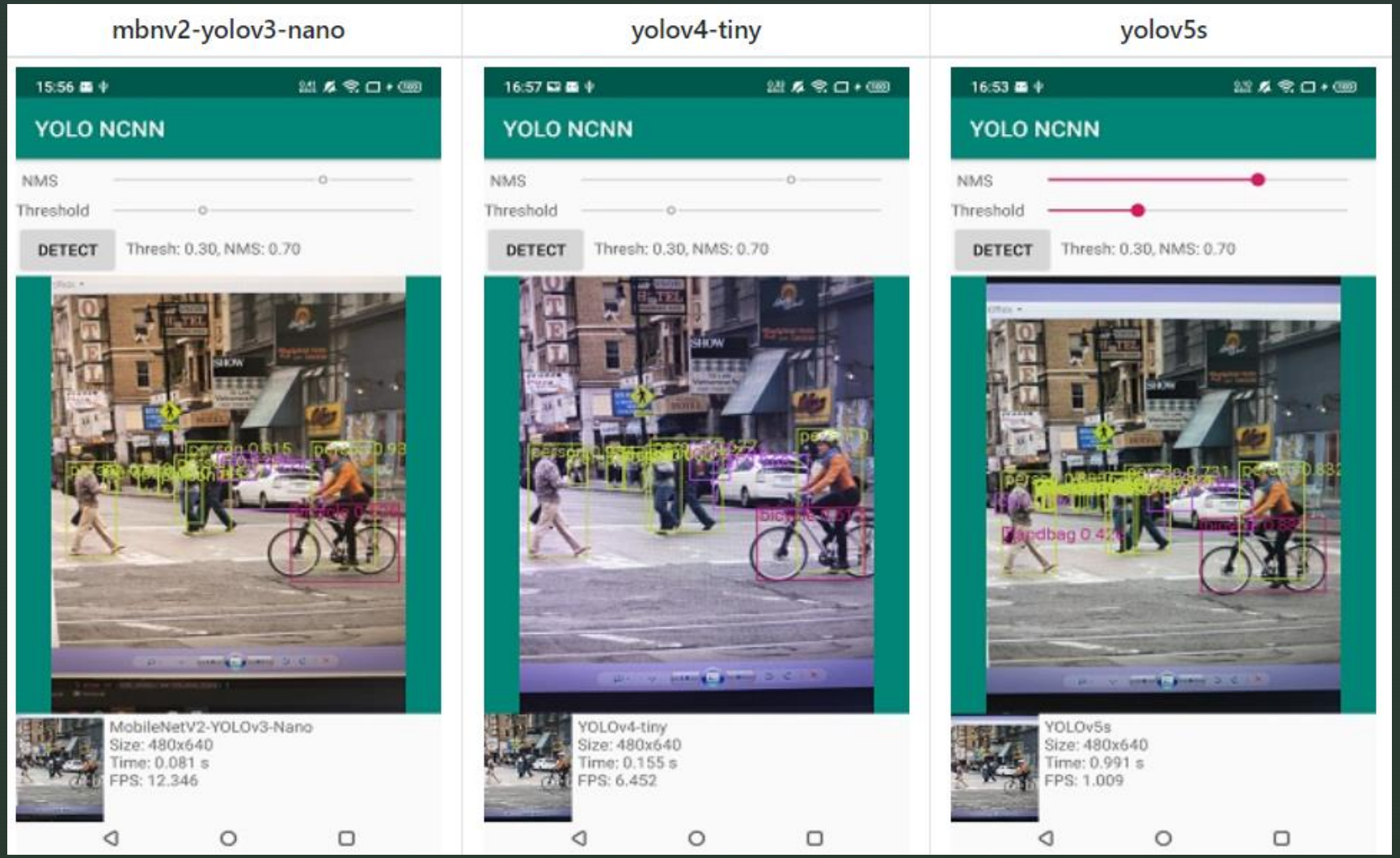
Examples

Using DOTA_OBB Data-set

Examples

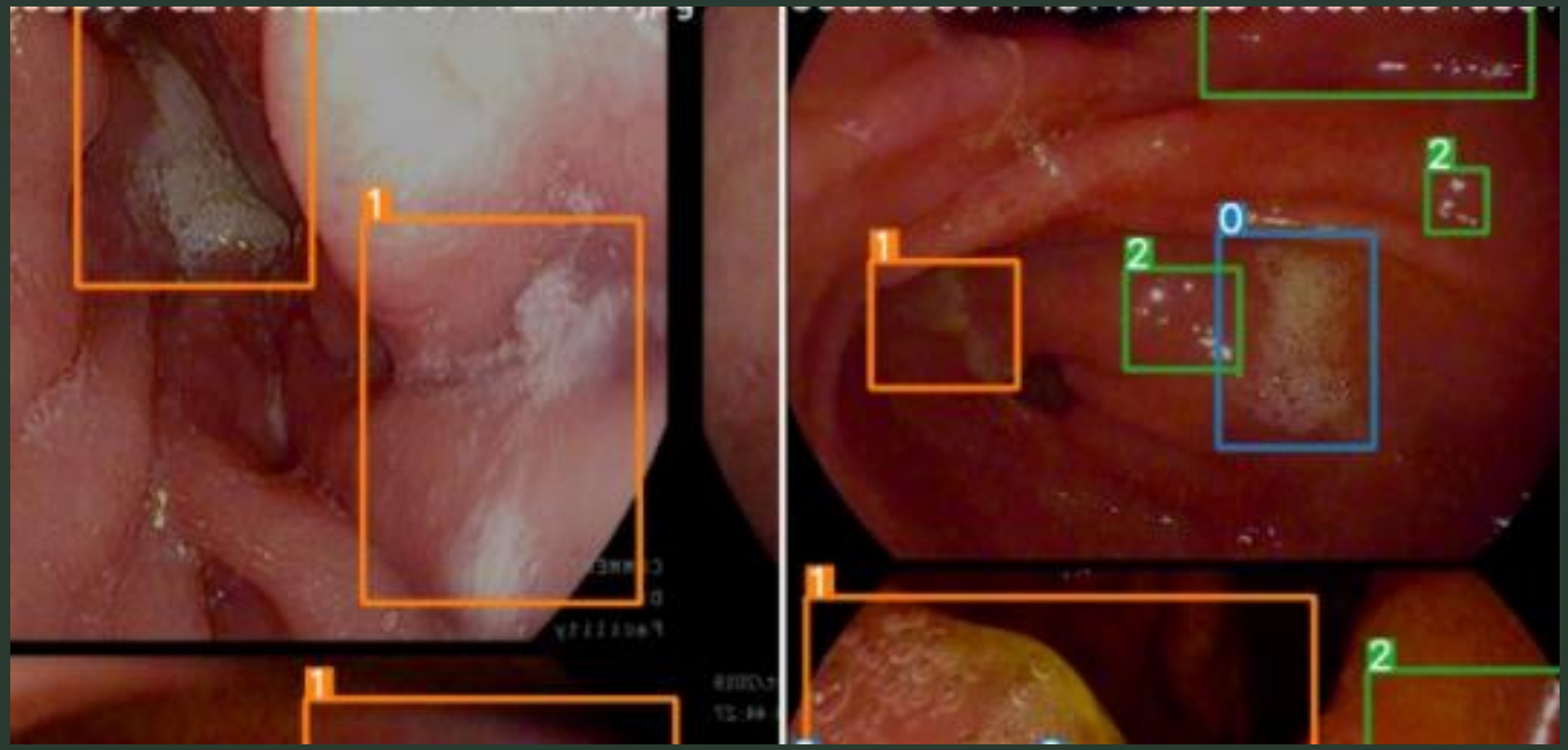

P2644__1__1648__1648.png
P2560__1__2472__2542.png
P2716__1__2472__824.png

Examples


mbnv2-yolov3-nano
yolov4-tiny
yolov5s
YOLO NCNN
NMS
Threshold
DETECT
Thresh: 0.30, NMS: 0.70
MobileNetV2-YOLOv3-Nano
Size: 480x640
Time: 0.081 s
FPS: 12.346
YOLOv4-tiny
Size: 480x640
Time: 0.155 s
FPS: 6.452
YOLOv5s
Size: 480x640
Time: 0.991 s
FPS: 1.009

Examples

Detection of digestive endoscopy targets in the medical field



Examples

Detection of people, traffic lights, cars and etc.

Examples

Detection of smoke and fire

Thank you for your attention