University of Washington

Classical and Deep Methods for Computer Vision
EE P 596

Report: Object Detection in Street using YOLOv5s

Authors: Ziwei Tao

December 15, 2021

Content

Introduction

This project focuses on object detection in different street scenes, the targets would be detected including people, animals and transportations. Firstly, there will be a series of training on COCO dataset based on three different models YOLOv5 network. Secondly, a new dataset is designed and created for training, testing on this network. Thirdly, some comparisons between three models in YOLOv5 trained on this new dataset will be executed for the purpose of observing the detection performance of each model.

The reason for choosing this topic is that I am interested in object detection, a technology related to computer vision and image processing. It is widely used in face detection, face recognition, image annotation, activity recognition and vehicle counting and provides diverse applications for people, which help improve the quality of human life.

Current study about object detection not only focused on normal target detection such as real-time vehicle and distance detection but also abnormal detection. For instance, Y.Li and X.He proposed an abnormal target detection algorithm based on the YOLOv5 framework, which has a good performance on the task of detecting and classifying the covid-19 on the chest radiograph [1]. There are also some works on comparing different algorithms in object detection. For example, since the single-stage detector YOLO is not as good as the two-stage detectors like Faster R-CNN in terms of accuracy, one study shows the fusion of these two algorithms by using the Kalman filter has better detection accuracy [2].

Method

Baseline

In this project, the baseline is as the Figure 1 shows. COCO dataset and Street Dataset are trained firstly, then obtain two files containing parameters from each training, and use these parameters in the process of detection. Street dataset are constantly adjusted in this process in order to achieve object detection. All training were processed on a Cloud GPU, Google Colaboratory [8].
In addition, the new dataset will be used for the process of training in YOLOv5s, YOLOv5m, YOLOv5l models separately for doing evaluation on detection performance for each model.

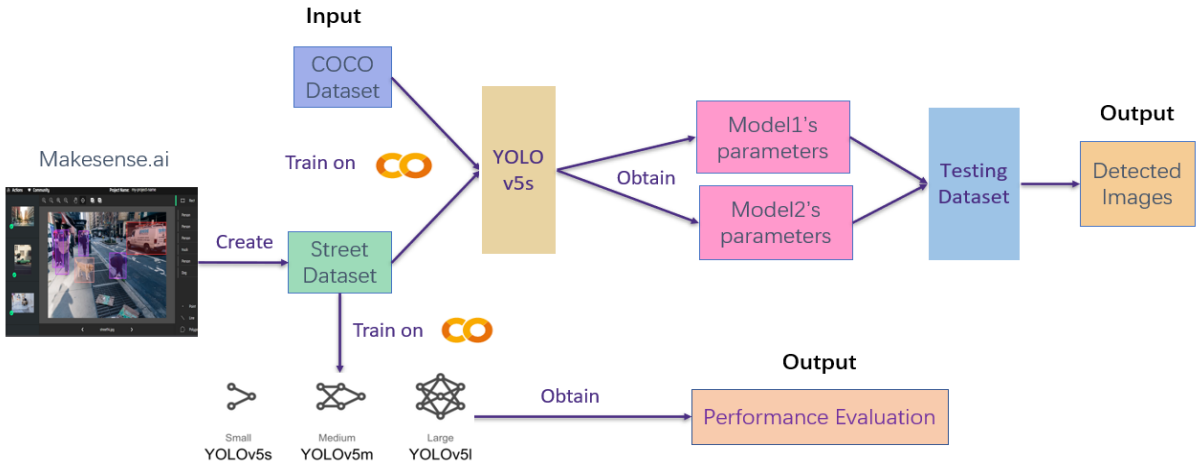

Figure 1: Baseline

Network

The network used in this project is YOLOv5, the structure of this network is shown in Figure 2. The YOLO object detector is high-speed due to its one-stage structure, which can predict all bounding boxes and class probabilities for those boxes simultaneously, using features from the entire image. It is suitable for end-to-end training in real-time while maintaining good average precision [3].

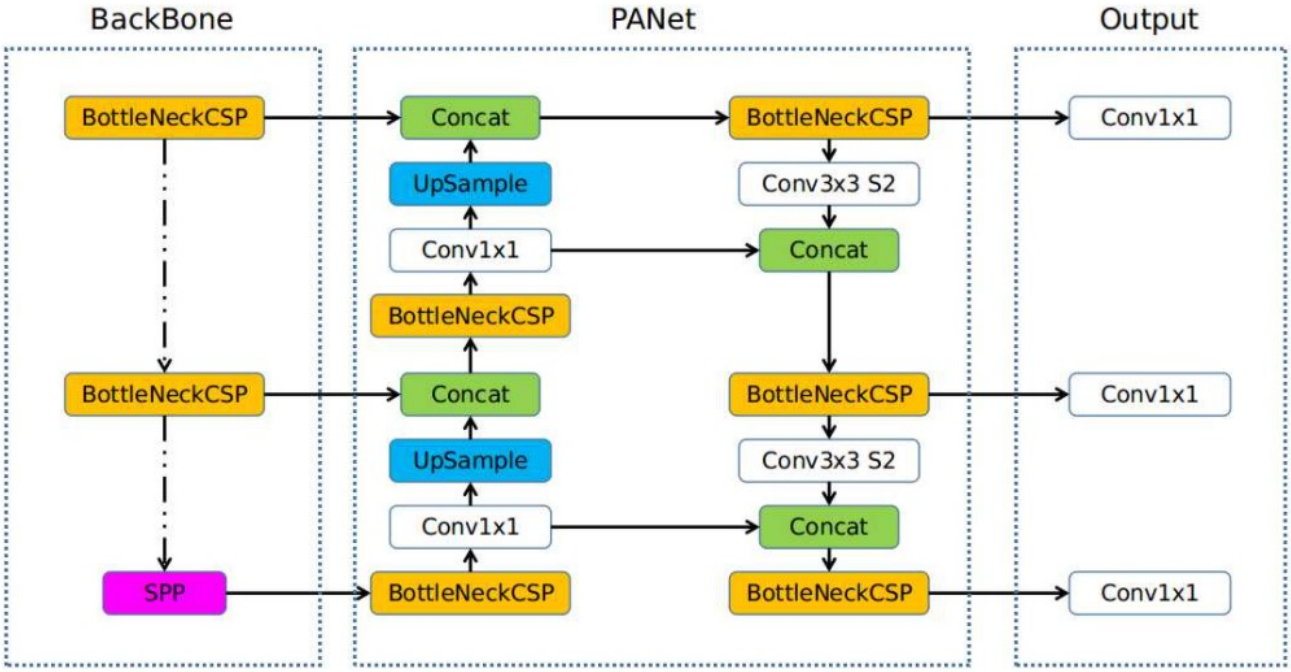

Figure 2: YOLOv5 network structure [4]

In the first process of this project, three different pretrained YOLOv5 model were download from the repository on Github [5], which were used to do testing, and then YOLOv5s model was chosen in the next step for training on COCO dataset and Street dataset separately. YOLOv5s has 270 layers, 7.2M parameters and 15.9 GFLOPs.

Datasets

In order to achieve object detection on street, a new dataset was created for training.
The Street dataset was made through an online tool, called *makesense.ai* and each picture was labeling manually. It contains 50 images and 7 classes totally. The classes are person, dog, car, trunk, traffic light, bus, bicycle. COCO dataset includes all these classes and this project only uses 128 images from COCO dataset

Table 1: Two datasets trained in the project

Dataset	Number of images	Classes
COCO	128	80
Street	50	7

Training configuration

The following table illustrates the parameters of the training process based on COCO dataset and Street dataset.

Table 2: Training configuration on two datasets

Training models	Fusing Layers	GFLOPs	Batch-size	Epochs	Software/Tool
Best1.pt	213	16.5	10	300	Google Colaboratory (Tesla K80, 11441.1875MB)
Best2.pt	213	15.9	10	300	Google Colaboratory (Tesla K80, 11441.1875MB)

Table 3: Training configuration on 3 models

Training models	Layers	GFLOPs	Batch-size	Epochs	Software/Tool
YOLOv5s	270	15.9	10	200	Tesla K80, 11441.1875MB
YOLOv5m	369	48.1	10	200	Tesla K80, 11441.1875MB
YOLOv5l	468	108.0	10	200	Tesla K80, 11441.1875MB

Results

Detection Performance

The performance metric used in this project is mean average precision (mAP) and precision, as shown in table 3. YOLOv5 uses $G_{IOULOss}$ as the loss function, which improves the detection speed and accuracy to a certain extent and is calculated as the following equation shows.

$$\mathrm{mAP}=\frac{1}{N}\sum_{i=1}^{N}\mathrm{AP}_i \quad [6]$$

AP_i is the AP in the i-th class, and N stands for the total number of classes being evaluated. For COCO, AP is the average over multiple IoU. For example, AP@0.5:0.95 corresponds to the average AP for IoU from 0.5 to 0.95

$$G_{IOULoss} = 1 - I_{IOU} + \frac{\left|\frac{C}{A \cup B}\right|}{|C|}$$

$$I_{IOU} = \left|\frac{A \cap B}{A \cup B}\right|$$

A is the prediction box and B is the real box. C is the minimum area containing A and B.

Although, some objects in images could be detected successfully based on the training model using Street dataset, it could be seen that training model based on COCO dataset has more accuracy than the model trained on Street dataset since it has higher mAP and precision. This result can be observed from the detected images as well, which are Fig3.

For the left one in the Fig3, which applying the models trained on COCO dataset, some small details could be detected and classified correctly like whether the object is a real people or a photo and for the transportations, which one is car or truck.

Figure 3: Image detection using different model's parameters.

There are some reasons causing this, one is that the size of Street dataset is small to this large network. Small datasets usually require models that have low complexity to avoid overfitting the model to the data [6]. Another reason is that the classes is not enough which may make feature extraction and representation less effective.

Table 3: Performance metrics

YOLOv5s	mAP@0.5	mAP@0.5:0.95	Precision
Training model on Street Dataset	0.82	0.64	0.91
Training model on COCO Dataset	0.98	0.83	0.96

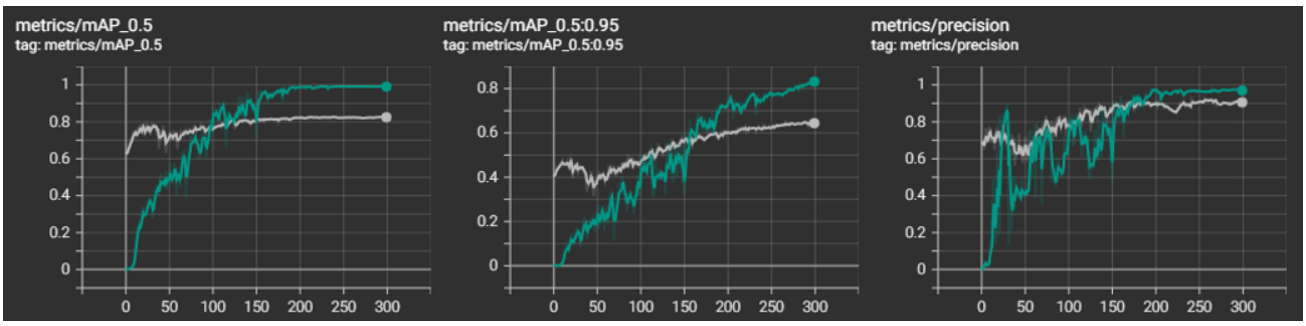

Figure 4: Line chart for metrics

Ablation study

In this project, after achieving object detection using Street dataset in training process. From previous part, the performance metrics show that the trained model on Street Dataset have less accurate in

detection and classification. Instead of enlarge the size and quality of the Street dataset, a test was tried on comparison between different models in order to see which one could perform better in object detection, all the training process using street dataset. The training time is longer as the model become larger. It could be seen that the YOLOv5l has the best performance on detection.

Table 4: Performance metrics on different models

Street Dataset	mAP@0.5	mAP@0.5:0.95	Precision
YOLOv5s	0.9879	0.7736	0.9581
YOLOv5m	0.9841	0.7648	0.9624
YOLOv5l	0.9904	0.8025	0.9659

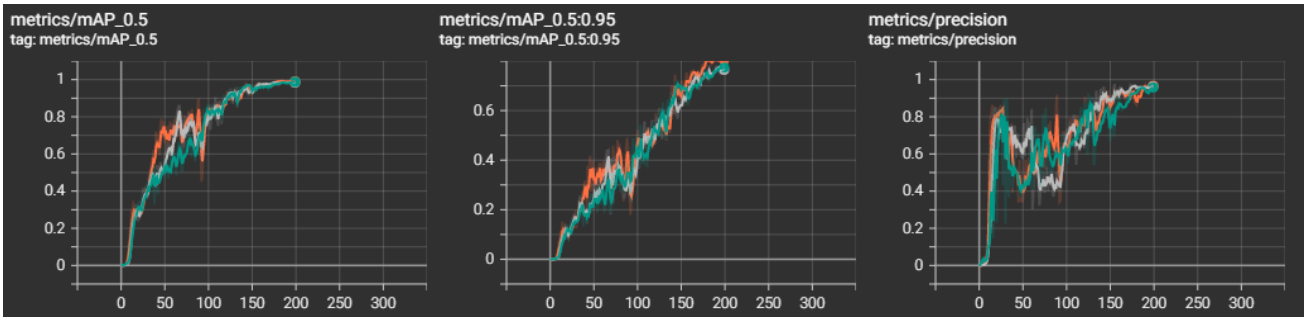

Figure 5: Line chart about metrics for 3 YOLOv5 models

Conclusion

In conclusion, this project successfully achieved object detection in street using YOLOv5 network. However, the new trained model using the new dataset has less mAP and higher loss than the trained model from COCO dataset. The training applying the new dataset based on diverse YOLOv5 models has different performance detection, larger model YOLOv5l has higher precision than the smaller one like YOLOv5s, however, in terms of mAP, the model having less complexity is more suitable for smaller dataset. In the future, the new dataset could be optimized in terms of size and quality.

Since the large-scale spread and mutation of COVID-19 epidemic have increased difficulties in its fast detection and effective diagnosis [7], instead of doing testing uses the different models in YOLOv5, more comparison between different algorithm in object detection like Faster R-CNN and EfficientDet SSD could be completed in the next process in order to show the speed, robustness and detection accuracy separately and then make combination of these network for the face mask detection in the future.

References

[1] Y. Li and X. He, "COVID-19 Detection in Chest Radiograph Based on YOLO v5," in *2021 IEEE International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)*, 2021, pp. 344-347.

[2] J. Fan, J. Lee, I. Jung and Y. Lee, "Improvement of Object Detection Based on Faster R-CNN and YOLO," in *2021 36th International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, 2021, pp. 1-4.

[3] Wu, S., Li, X. and Wang, X., "IoU-aware single-stage object detector for accurate localization." in Image and Vision Computing, May 2020, vol. 97, p.103911.

[4] "Overview of model structure about YOLOv5" https://github.com/ultralytics/yolov5/issues/280 (accessed Dec. 4, 2021)

[5] "glenn-jocher YOLOv5 release v6.0" https://github.com/ultralytics/yolov5/tree/v6.0 (accessed by Dec. 1, 2021)

[6] Catal, C. and Diri, B., "Investigating the effect of dataset size, metrics sets, and feature selection techniques on software fault prediction problem." In *Information Sciences.*, 2019, vol. 179, no. 8, pp. 1040-1058.

[7] Yang, G., Feng, W., Jin, J., Lei, Q., Li, X., Gui, G. and Wang, W., "Face Mask Recognition System with YOLOV5 Based on Image Recognition," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)* ChengDu, China, Dec. 2021, pp. 1398-1404.

[8] *Colaboratory,* Google [Online]. Available: https://colab.research.google.com/

Appendix A

The following are dataset file stored as a format yaml

```yaml
# YOLOv5 🚀 by Ultralytics, GPL-3.0 license
# COCO128 dataset https://www.kaggle.com/ultralytics/coco128 (first 128 images from COCO train2017)
# Example usage: python train.py --data coco128.yaml
# parent
# ├── yolov5
# └── datasets
#     └── coco128  ← downloads here


# Train/val/test sets as 1) dir: path/to/imgs, 2) file: path/to/imgs.txt, or 3) list: [path/to/imgs1, path/to/imgs2, ..]
path: ../datasets/coco128  # dataset root dir
train: images/train2017  # train images (relative to 'path') 128 images
val: images/train2017  # val images (relative to 'path') 128 images
test:  # test images (optional)

# Classes
nc: 80  # number of classes
names: ['person', 'bicycle', 'car', 'motorcycle', 'airplane', 'bus', 'train', 'truck', 'boat', 'traffic light',
        'fire hydrant', 'stop sign', 'parking meter', 'bench', 'bird', 'cat', 'dog', 'horse', 'sheep', 'cow',
        'elephant', 'bear', 'zebra', 'giraffe', 'backpack', 'umbrella', 'handbag', 'tie', 'suitcase', 'frisbee',
        'skis', 'snowboard', 'sports ball', 'kite', 'baseball bat', 'baseball glove', 'skateboard', 'surfboard',
        'tennis racket', 'bottle', 'wine glass', 'cup', 'fork', 'knife', 'spoon', 'bowl', 'banana', 'apple',
        'sandwich', 'orange', 'broccoli', 'carrot', 'hot dog', 'pizza', 'donut', 'cake', 'chair', 'couch',
        'potted plant', 'bed', 'dining table', 'toilet', 'tv', 'laptop', 'mouse', 'remote', 'keyboard', 'cell phone',
        'microwave', 'oven', 'toaster', 'sink', 'refrigerator', 'book', 'clock', 'vase', 'scissors', 'teddy bear',
        'hair drier', 'toothbrush']  # class names


# Download script/URL (optional)
download: https://github.com/ultralytics/yolov5/releases/download/v1.0/coco128.zip
```

```
train: localdata\images\train # train images (relative to 'path') 50 images
val: localdata\images\train # val images (relative to 'path') 50 images
test: # 7 test images (optional)

# Classes
nc: 7 # number of classes
names: ["Car", "Person", "Dog", "bicycle", "traffic light", "bus", "truck"]
```