

Task 5.2: Minor research project

1. Introduction

Artificial Intelligence has been changing the way of software development across different verticals. Better deep learning models and algorithms are coming up owing to better hardware capabilities and increase in organically generated data. In [1] we can also see this impact on computer vision, where impact of deep learning in the domain of computer vision has been discussed at length. Object detection is a specific subproblem within the domain of computer vision where the objective is to find the exact location as bounding box coordinates on a given image frame. Over last few decades, several algorithms have been proposed for performing object detection. Earlier, classical machine learning methods based on hand crafted features like Haar Cascade as proposed by Viola-Jones in [2] and Histogram of Oriented Gradients (HOG) [3] were used across tasks related to object detection. The impact of deep learning can be seen in object detection scenarios as well, recent deep learning architectures like Faster RCNN [4], SSD [5] and YOLO [6] constitute some of the state-of-the-art object detection deep learning networks. This report focusses on summarising the working of YOLO9000 [7] and finally demonstrates the working using Road Sign Detection dataset [8].

2. The road to YOLO9000

At the time when YOLO [6] was introduced, detection systems used classifiers which used techniques like sliding window or region proposal to classify various regions of image at different scales. YOLO proposed a novel approach of reframing object detection problem as a regression problem, the algorithm uses a unified model to predict object coordinates and class probabilities using raw image pixels. The algorithm outputs multiple bounding boxes and probabilities representing associated class labels and then uses non-maximum suppression to choose dominating object bounding location and label (Shown in figure 1). An added advantage of this approach is that because the network looks at the complete image frame during the training stage, it also learns the contextual information associated with the object of interest through object's neighbourhood.

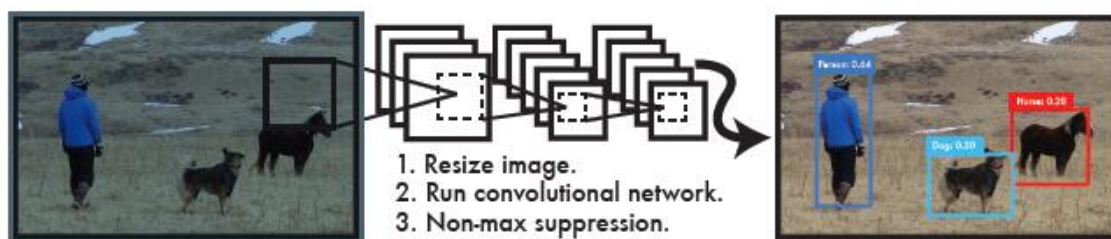


Fig 1: Image showing the working of YOLO [6].

While YOLO had its novelties, it suffered from multiple shortcomings, a significant error when compared to region proposal methods was the low recall of YOLO. YOLO9000 [7] or also known as YOLOv2 was introduced to address these shortcomings and take the YOLO architecture to at par or surpass the state-of-the-art approaches.

3. Why YOLO works?

This section discusses the additions and novel concepts introduced by YOLO9000, which make it the state-of-the-art object detection framework of the time.

YOLO9000 introduced **Batch Normalisation**, this helps the model to improve convergence and tackle regularisation issues. [7] reports 2% increase in mAP (mean Average Precision) after adding Batch Normalisation on all the convolution layers of the original YOLO architecture. Inspired from the use of priors in Faster R-CNN, YOLO9000 introduces the concept of **Anchor Boxes** to the YOLO architecture. To perform this modification, fully connected layer of the original YOLO architecture is removed, and the network learns to predicts offsets with respect to hard coded anchor boxes, this simplifies the learning process. Another key modification while introducing anchor boxes was the reducing the image resolution from 448 x 448 to 416 x 416, this forces the network to output odd number of bounding box coordinates for the feature map and the network has only one centre cell. This also means that the output feature map is now of the dimension 13 x 13 as YOLO9000 has convolution layers which down sample the input image by a factor of 32.

With these modifications, another change to the YOLO architecture was needed, the original architecture predicted class probabilities based on actual locations of the object in the image, instead of this, the architecture now predicts class probabilities and *objectness score* for every anchor box. The concept of objectness score is preserved in the modified architecture as well, where objectness score denotes the IOU of the proposed bounding box and the ground truth bounding box and given there is an object, class probabilities denote the conditional probability that of the class labels associated with the object. These changes contribute to a significant increase in recall, taking the initial recall figure of 81% to 88%. While there is an increase in recall, the accuracy of the architecture takes a slight hit going down to 69.2 mAP from 69.5 mAP.

While the introduction of anchor boxes increased recall, it had some shortcomings, the first major shortcoming was the hand picked or hard coded anchors used, this makes it difficult for the network to predict better detections for different objects. To tackle this issue, authors use KMeans to select the anchor boxes (with K value of 5). This strategy results in much higher IOU as opposed to using hard coded anchor boxes.

Because the anchor boxes are predicted as offsets, they might be shifted to an inaccurate location in the image. This unconstrained nature of the anchor boxes prohibits the network to become stable, this is tackled by introducing logistic activation. This is like the approach used in YOLOv1 and the predicted bounding box coordinates are relative to the grid cell location. This squashes the input between 0 and 1 and makes the learning process easier as the offsets are localised the given grid cells and introduces additional stability on the network. Another key experiment was the introduction of **passthrough layer**, this allows the features from previous layers to be used deeper in the network. This brings higher resolution features and stacks them with lower resolution features, this provides the detector access to an elaborated feature map induced with fine grained features to perform detection.

Another powerful addition to the training process of the improved YOLO9000 architecture is the introduction of **multi-scale training**, during the training process the network chooses an image resolution randomly after every 10 batches. This is possible because the network only consists of convolution and pooling layer, this makes the network elastic enough to be resized during the training process. This also makes the network more robust to handle various input sizes, this makes YOLO9000 ideal for real time applications where such high frame rate is preferred. The range for these resizes

values varies from 320 x 320 to 608 x 608 (multiples of 32, this makes the down sampling convenient). The result of this multi scale training is that the network achieves very **high mAP of 78.6** for high resolution of **544 x 544 at an FPS of 40** whereas it performs reasonably well at low resolutions of **288 x 288** where it achieves a **decent mAP of 69.0 with very high 91 FPS**. This provides the end user a flexibility to use the network depending on the use case (high latency vs high accuracy).

4. Possible improvements on YOLO9000

The previous section describes how and why the YOLO9000 detector came to be known as one of the state-of-the-art detectors. While the modifications to YOLOv1 architecture made several improvements to design and performance, there is still a considerable room for future improvements to the architecture of YOLO9000. This section explores the possible improvements which can make this architecture even more efficient. Section 4.1 highlight some of the work that has already been done to improve YOLO9000 and section 4.2 highlights some proposals which can lay foundations to increase the performance even more.

4.1. The YOLO family

Over the years several implementations have contributed to adding features and improvements to the YOLO9000 architecture. Some of these are from the original authors (Yolov3) while others (YOLOv4, v5, v6, v7) are from different contributors. This is just a brief overview of the other architectures in this family.

YOLOv3 [9]: YOLOv3 aims to improve the accuracy of YOLOv2 and increases the complexity of the features extracted by the convolution layer. To achieve this, the convolution layers are increased from 19 to 53. The framework for training and testing YOLO is developed in C++ is called darknet, for YOLOv3 this framework is called as Darknet-53 and for YOLOv2 Darknet-19, the suffixed number denotes the number of convolution layers in each YOLO architecture. Because YOLOv3 is deeper, it can extract higher order features and thus there is an increase in the accuracy, this is coupled with shortcut connections which enable fine grained features.

YOLOv4 [10]: YOLOv4 builds on top of YOLOv3 and introduces universal features which improve a CNN networks performance, YOLOv4 uses **Weighted-Residual-Connections, Self-adversarial-training, Mish-activation, Cross mini-Batch Normalization, Cross-Stage-Partial-connections** to improve the performance. This also uses data augmentation techniques like CutMix, MixUp, Blurring, Mossaic to improve the robustness of the model.

YOLOv5 [11]: YOLOv5 is not a published work rather a PyTorch implementation of existing concepts with minor changes. This work is more focussed on improving integration of the YOLO architecture and is compatible with popular cloud platforms like AWS, GCP and containers like Docker. The focus of this work is to provide wide support for usage of YOLO architecture apart from Darknet framework.

Note: Due to the easy integration, PyTorch compatibility and most resemblance with YOLO9000, this repository is used to demonstrate the working of YOLO on Road Sign Detection Dataset in this report.

YOLOv6 [12]: Building on top of YOLOv4, YOLOv6 improves and modifies the architecture to improve both accuracy and speed using novel quantization method which is responsible for faster FLOPs.

YOLOv7 [13]: YOLOv7 introduces trainable bag of freebies methods, re-parameterization of original model, *compound scaling* and *extend* methods and reducing parameters to increase inference speed and accuracy.

4.2 Other improvements

While the above works suggest that there is ongoing research to increase the efficiency of YOLO architectures, it is important to evaluate these architectures with the AI bias viewpoint so that we can move forward towards a *fair* object detector (YOLO architectures are significantly used in pedestrian detection scenarios, hence it is important to evaluate them on terms of *AI bias*). This will help uncover any prevalent biases and introduce training methods which can help mitigate these.

Another important area of improvement for these architectures is focus on inference using lightweight hardware. As we move towards democratization of AI more and more mobile applications and other lightweight devices (FPGA based devices) will rely on AI models. YOLO has lightweight variants such as tiny-YOLO, but even these modifications are too heavy to run in real time on low power devices, YOLO can be stripped down to a lighter network architecture like [14].

5. Experiment on Road Sign Detection Dataset

This section demonstrates how YOLO can be used to set up a training pipeline and inference pipeline using a custom dataset.

5.1 Dataset Details: The dataset [8] contains 877 samples which are divided as shown in table 1.

The dataset has these class labels:

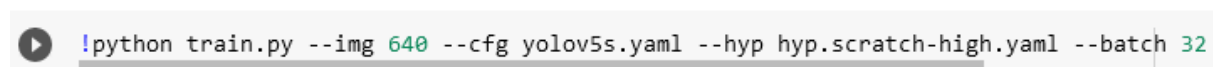
- traffic light
- stop
- speed limit
- crosswalk

Train	701
Validation	88
Test	88

Table 1: Dataset split

5.2 Training: To train the YOLO model we use the train.py script from the official YOLOv5 repository and save the trained weights to the specified directory. It is important to note that because this is just a demonstration of the training process, we have trained the model till 100 epochs, whereas in real world scenarios the training process can go up in order of thousands of epochs.

This also includes changes to the training hyperparameter config file, where the number of classes parameter must be changed to 4 to reflect updated number of classes.



```
!python train.py --img 640 --cfg yolov5s.yaml --hyp hyp.scratch-high.yaml --batch 32
```

Fig 2: Training command for YOLO

5.3 Testing: There are several ways to test the trained model, the attached notebook shows using the test script to test on a given test set as well as using the API to test which can be used in real world applications and integrated into inference pipelines.

5.4 Results: Although the network is trained on a small dataset and only for 100 epochs, we can see that the network achieves high mAP of 95.2%, 99.3% Precision and 90.1% Recall. It is important to note that the training uses pretrained weights of YOLOv5 which contributes to such high evaluation

metrics. This also presents the case of how easily YOLO can be used for real world dataset and thus is fit for real world applications with minimal finetuning.

```
!python val.py --weights runs/train/yolo_road_det3/weights/best.pt --data road_sign_data.yaml --task test --name
val: data=/content/yolov5/data/road_sign_data.yaml, weights=['runs/train/yolo_road_det3/weights/best.pt'], batch_
YOLOv5 v6.2-186-g7f097dd Python-3.7.14 torch-1.12.1+cu113 CUDA:0 (Tesla V100-SXM2-16GB, 16160MiB)

Fusing layers...
YOLOv5s summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
test: Scanning '/content/RoadSignDetectionDataset/labels/test' images and labels...88 found, 0 missing, 0 empty, 1
test: New cache created: /content/RoadSignDetectionDataset/labels/test.cache
```

Class	Images	Instances	P	R	mAP50	mAP50-95
all	88	126	0.993	0.901	0.952	0.795
trafficlight	88	20	1	0.647	0.828	0.554
stop	88	7	0.986	1	0.995	0.889
speedlimit	88	76	0.994	1	0.995	0.9
crosswalk	88	23	0.992	0.957	0.99	0.837

```
Speed: 0.3ms pre-process, 2.0ms inference, 0.9ms NMS per image at shape (32, 3, 640, 640)
Results saved to runs/val/yolo_det2
```

Fig 3: Result showing the overall mAP, Precision and Recall as well as class wise evaluation metrics.

Note: For entire training and testing process please refer to the attached notebook in Appendix – 1.

6. Conclusion

This report summarises the YOLO architecture and highlights the evolution of algorithm over different versions. This report also indicates future areas of improvement in YOLO9000 and other YOLO architectures. Finally, this report showcases the algorithm in practice using a real-world dataset, a similar model can be used in self driving car use-case to identify Road Signs.

7. References

- [1] P. Patel and A. Thakkar, "The upsurge of deep learning for computer vision applications," *Int. J. Electr. Comput. Eng. (IJECE)*, vol. 10, no. 1, p. 538, 2020.
- [2] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, 2005, vol. 1, p. I–I.
- [3] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, 2005, vol. 1, pp. 886–893 vol. 1.
- [4] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *arXiv [cs.CV]*, 2015.
- [5] W. Liu *et al.*, "SSD: Single Shot MultiBox Detector," *arXiv [cs.CV]*, 2015.
- [6] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," *arXiv [cs.CV]*, 2015.
- [7] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.
- [8] Larxel, "Road sign detection." 24-May-2020.
- [9] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement," *arXiv [cs.CV]*, 2018.
- [10] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "YOLOv4: Optimal speed and accuracy of object detection," *arXiv [cs.CV]*, 2020.
- [11] *Yolov5*.
- [12] C. Li *et al.*, "YOLOv6: A single-stage object detection framework for industrial applications," *arXiv [cs.CV]*, 2022.

- [13] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv [cs.CV]*, 2022.
- [14] J. Pedoeem and R. Huang, "YOLO-LITE: A real-time object detection algorithm optimized for non-GPU computers," *arXiv [cs.CV]*, 2018.

APPENDIX - I

The following cells require the dataset to be in its respective location.

assignment_5_2

October 7, 2022

```
[1]: gpu_info = !nvidia-smi
gpu_info = '\n'.join(gpu_info)
if gpu_info.find('failed') >= 0:
    print('Not connected to a GPU')
else:
    print(gpu_info)
```

Thu Oct 6 12:17:28 2022

```
+-----+
| NVIDIA-SMI 460.32.03      Driver Version: 460.32.03      CUDA Version: 11.2      |
+-----+-----+-----+-----+-----+-----+
| GPU  Name           Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|      Memory-Usage | GPU-Util  Compute M. |
|                                       |                      | MIG M. |
+-----+-----+-----+-----+-----+-----+
|   0   Tesla V100-SXM2...    Off   | 00000000:00:04:0  Off   |              0      |
| N/A   33C    P0     23W / 300W |      0MiB / 16160MiB |      0%      Default  |
|                                       |                      | N/A |
+-----+-----+-----+-----+-----+-----+

+-----+
| Processes:
| GPU   GI    CI          PID    Type    Process name                  GPU Memory
|       ID    ID                                   |          Usage          |
+-----+-----+-----+-----+-----+-----+
| No running processes found
+-----+
```

```
[2]: !git clone https://github.com/ultralytics/yolov5 # clone
!pip install -r yolov5/requirements.txt # install
```

```
fatal: destination path 'yolov5' already exists and is not an empty directory.
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: matplotlib>=3.2.2 in
/usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line
5)) (3.2.2)
Requirement already satisfied: numpy>=1.18.5 in /usr/local/lib/python3.7/dist-
packages (from -r yolov5/requirements.txt (line 6)) (1.21.6)
```

Requirement already satisfied: opencv-python>=4.1.1 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 7)) (4.6.0.66)

Requirement already satisfied: Pillow>=7.1.2 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 8)) (7.1.2)

Requirement already satisfied: PyYAML>=5.3.1 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 9)) (6.0)

Requirement already satisfied: requests>=2.23.0 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 10)) (2.23.0)

Requirement already satisfied: scipy>=1.4.1 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 11)) (1.7.3)

Requirement already satisfied: torch>=1.7.0 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 12)) (1.12.1+cu113)

Requirement already satisfied: torchvision>=0.8.1 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 13)) (0.13.1+cu113)

Requirement already satisfied: tqdm>=4.64.0 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 14)) (4.64.1)

Requirement already satisfied: tensorboard>=2.4.1 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 18)) (2.8.0)

Requirement already satisfied: pandas>=1.1.4 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 23)) (1.3.5)

Requirement already satisfied: seaborn>=0.11.0 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 24)) (0.11.2)

Requirement already satisfied: ipython in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 41)) (7.9.0)

Requirement already satisfied: psutil in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 42)) (5.4.8)

Requirement already satisfied: thop>=0.1.1 in /usr/local/lib/python3.7/dist-packages (from -r yolov5/requirements.txt (line 43)) (0.1.1.post2209072238)

Requirement already satisfied: cycycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=3.2.2->-r yolov5/requirements.txt (line 5)) (0.11.0)

Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=3.2.2->-r yolov5/requirements.txt (line 5)) (3.0.9)

Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=3.2.2->-r yolov5/requirements.txt (line 5)) (2.8.2)

Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib>=3.2.2->-r yolov5/requirements.txt (line 5)) (1.4.4)

Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/lib/python3.7/dist-packages (from requests>=2.23.0->-r yolov5/requirements.txt (line 10)) (1.24.3)

Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages (from requests>=2.23.0->-r yolov5/requirements.txt (line 10)) (2.10)

Requirement already satisfied: certifi>=2017.4.17 in
 /usr/local/lib/python3.7/dist-packages (from requests>=2.23.0->-r
 yolov5/requirements.txt (line 10)) (2022.6.15)

Requirement already satisfied: chardet<4,>=3.0.2 in
 /usr/local/lib/python3.7/dist-packages (from requests>=2.23.0->-r
 yolov5/requirements.txt (line 10)) (3.0.4)

Requirement already satisfied: typing-extensions in
 /usr/local/lib/python3.7/dist-packages (from torch>=1.7.0->-r
 yolov5/requirements.txt (line 12)) (4.1.1)

Requirement already satisfied: tensorboard-data-server<0.7.0,>=0.6.0 in
 /usr/local/lib/python3.7/dist-packages (from tensorboard>=2.4.1->-r
 yolov5/requirements.txt (line 18)) (0.6.1)

Requirement already satisfied: wheel>=0.26 in /usr/local/lib/python3.7/dist-
 packages (from tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18))
 (0.37.1)

Requirement already satisfied: absl-py>=0.4 in /usr/local/lib/python3.7/dist-
 packages (from tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (1.2.0)

Requirement already satisfied: werkzeug>=0.11.15 in
 /usr/local/lib/python3.7/dist-packages (from tensorboard>=2.4.1->-r
 yolov5/requirements.txt (line 18)) (1.0.1)

Requirement already satisfied: protobuf>=3.6.0 in /usr/local/lib/python3.7/dist-
 packages (from tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18))
 (3.17.3)

Requirement already satisfied: google-auth<3,>=1.6.3 in
 /usr/local/lib/python3.7/dist-packages (from tensorboard>=2.4.1->-r
 yolov5/requirements.txt (line 18)) (1.35.0)

Requirement already satisfied: setuptools>=41.0.0 in
 /usr/local/lib/python3.7/dist-packages (from tensorboard>=2.4.1->-r
 yolov5/requirements.txt (line 18)) (57.4.0)

Requirement already satisfied: grpcio>=1.24.3 in /usr/local/lib/python3.7/dist-
 packages (from tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18))
 (1.48.1)

Requirement already satisfied: google-auth-oauthlib<0.5,>=0.4.1 in
 /usr/local/lib/python3.7/dist-packages (from tensorboard>=2.4.1->-r
 yolov5/requirements.txt (line 18)) (0.4.6)

Requirement already satisfied: tensorboard-plugin-wit>=1.6.0 in
 /usr/local/lib/python3.7/dist-packages (from tensorboard>=2.4.1->-r
 yolov5/requirements.txt (line 18)) (1.8.1)

Requirement already satisfied: markdown>=2.6.8 in /usr/local/lib/python3.7/dist-
 packages (from tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (3.4.1)

Requirement already satisfied: pytz>=2017.3 in /usr/local/lib/python3.7/dist-
 packages (from pandas>=1.1.4->-r yolov5/requirements.txt (line 23)) (2022.2.1)

Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.7/dist-
 packages (from google-auth<3,>=1.6.3->tensorboard>=2.4.1->-r
 yolov5/requirements.txt (line 18)) (1.15.0)

Requirement already satisfied: rsa<5,>=3.1.4 in /usr/local/lib/python3.7/dist-
 packages (from google-auth<3,>=1.6.3->tensorboard>=2.4.1->-r
 yolov5/requirements.txt (line 18)) (4.9)

Requirement already satisfied: pyasn1-modules>=0.2.1 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (0.2.8)

Requirement already satisfied: cachetools<5.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from google-auth<3,>=1.6.3->tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (4.2.4)

Requirement already satisfied: requests-oauthlib>=0.7.0 in /usr/local/lib/python3.7/dist-packages (from google-auth-oauthlib<0.5,>=0.4.1->tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (1.3.1)

Requirement already satisfied: importlib-metadata>=4.4 in /usr/local/lib/python3.7/dist-packages (from markdown>=2.6.8->tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (4.12.0)

Requirement already satisfied: zipp>=0.5 in /usr/local/lib/python3.7/dist-packages (from importlib-metadata>=4.4->markdown>=2.6.8->tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (3.8.1)

Requirement already satisfied: pyasn1<0.5.0,>=0.4.6 in /usr/local/lib/python3.7/dist-packages (from pyasn1-modules>=0.2.1->google-auth<3,>=1.6.3->tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (0.4.8)

Requirement already satisfied: oauthlib>=3.0.0 in /usr/local/lib/python3.7/dist-packages (from requests-oauthlib>=0.7.0->google-auth-oauthlib<0.5,>=0.4.1->tensorboard>=2.4.1->-r yolov5/requirements.txt (line 18)) (3.2.0)

Requirement already satisfied: pexpect in /usr/local/lib/python3.7/dist-packages (from ipython->-r yolov5/requirements.txt (line 41)) (4.8.0)

Requirement already satisfied: pygments in /usr/local/lib/python3.7/dist-packages (from ipython->-r yolov5/requirements.txt (line 41)) (2.6.1)

Requirement already satisfied: decorator in /usr/local/lib/python3.7/dist-packages (from ipython->-r yolov5/requirements.txt (line 41)) (4.4.2)

Requirement already satisfied: backcall in /usr/local/lib/python3.7/dist-packages (from ipython->-r yolov5/requirements.txt (line 41)) (0.2.0)

Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.7/dist-packages (from ipython->-r yolov5/requirements.txt (line 41)) (5.1.1)

Requirement already satisfied: prompt-toolkit<2.1.0,>=2.0.0 in /usr/local/lib/python3.7/dist-packages (from ipython->-r yolov5/requirements.txt (line 41)) (2.0.10)

Requirement already satisfied: pickleshare in /usr/local/lib/python3.7/dist-packages (from ipython->-r yolov5/requirements.txt (line 41)) (0.7.5)

Requirement already satisfied: jedi>=0.10 in /usr/local/lib/python3.7/dist-packages (from ipython->-r yolov5/requirements.txt (line 41)) (0.18.1)

Requirement already satisfied: parso<0.9.0,>=0.8.0 in /usr/local/lib/python3.7/dist-packages (from jedi>=0.10->ipython->-r yolov5/requirements.txt (line 41)) (0.8.3)

Requirement already satisfied: wcwidth in /usr/local/lib/python3.7/dist-packages

(from prompt-toolkit<2.1.0,>=2.0.0->ipython->-r yolov5/requirements.txt (line 41)) (0.2.5)

Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.7/dist-packages (from pexpect->ipython->-r yolov5/requirements.txt (line 41)) (0.7.0)

```
[34]: import torch
from IPython.display import Image # for displaying images
import os
import random
import shutil
from sklearn.model_selection import train_test_split
import xml.etree.ElementTree as ET
from xml.dom import minidom
from tqdm import tqdm
from PIL import Image, ImageDraw
import numpy as np
import matplotlib.pyplot as plt
import glob
import cv2
```

```
[4]: from google.colab import drive
drive.mount('/content/gdrive')
!cp -f '/content/gdrive/MyDrive/sit789_assignment_5_2_data/
↳RoadSignDetectionDataset.zip' '/content/'
!rm -rf RoadSignDetectionDataset
!mkdir RoadSignDetectionDataset
!unzip /content/RoadSignDetectionDataset.zip -d /content/
↳RoadSignDetectionDataset/
```

Drive already mounted at /content/gdrive; to attempt to forcibly remount, call drive.mount("/content/gdrive", force_remount=True).

Archive: /content/RoadSignDetectionDataset.zip

```
inflating: /content/RoadSignDetectionDataset/annotations/road0.xml
inflating: /content/RoadSignDetectionDataset/annotations/road1.xml
inflating: /content/RoadSignDetectionDataset/annotations/road10.xml
inflating: /content/RoadSignDetectionDataset/annotations/road100.xml
inflating: /content/RoadSignDetectionDataset/annotations/road101.xml
inflating: /content/RoadSignDetectionDataset/annotations/road102.xml
inflating: /content/RoadSignDetectionDataset/annotations/road103.xml
inflating: /content/RoadSignDetectionDataset/annotations/road104.xml
inflating: /content/RoadSignDetectionDataset/annotations/road105.xml
inflating: /content/RoadSignDetectionDataset/annotations/road106.xml
inflating: /content/RoadSignDetectionDataset/annotations/road107.xml
inflating: /content/RoadSignDetectionDataset/annotations/road108.xml
inflating: /content/RoadSignDetectionDataset/annotations/road109.xml
inflating: /content/RoadSignDetectionDataset/annotations/road11.xml
inflating: /content/RoadSignDetectionDataset/annotations/road110.xml
inflating: /content/RoadSignDetectionDataset/annotations/road111.xml
```


[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

[illegible]

```

inflating: /content/RoadSignDetectionDataset/images/road90.png
inflating: /content/RoadSignDetectionDataset/images/road91.png
inflating: /content/RoadSignDetectionDataset/images/road92.png
inflating: /content/RoadSignDetectionDataset/images/road93.png
inflating: /content/RoadSignDetectionDataset/images/road94.png
inflating: /content/RoadSignDetectionDataset/images/road95.png
inflating: /content/RoadSignDetectionDataset/images/road96.png
inflating: /content/RoadSignDetectionDataset/images/road97.png
inflating: /content/RoadSignDetectionDataset/images/road98.png
inflating: /content/RoadSignDetectionDataset/images/road99.png

```

```

[5]: def extract_info_from_xml(xml_file):
    root = ET.parse(xml_file).getroot()

    # Initialise the info dict
    info_dict = {}
    info_dict['bboxes'] = []

    # Parse the XML Tree
    for elem in root:
        # Get the file name
        if elem.tag == "filename":
            info_dict['filename'] = elem.text

        # Get the image size
        elif elem.tag == "size":
            image_size = []
            for subelem in elem:
                image_size.append(int(subelem.text))

            info_dict['image_size'] = tuple(image_size)

        # Get details of the bounding box
        elif elem.tag == "object":
            bbox = {}
            for subelem in elem:
                if subelem.tag == "name":
                    bbox["class"] = subelem.text

                elif subelem.tag == "bndbox":
                    for subsubelem in subelem:
                        bbox[subsubelem.tag] = int(subsubelem.text)
            info_dict['bboxes'].append(bbox)

    return info_dict

```



```
[6]: print(extract_info_from_xml('/content/RoadSignDetectionDataset/annotations/
↳road4.xml'))
```

```
{'bboxes': [{'class': 'trafficlight', 'xmin': 20, 'ymin': 109, 'xmax': 81,
'ymax': 237}, {'class': 'trafficlight', 'xmin': 116, 'ymin': 162, 'xmax': 163,
'ymax': 272}, {'class': 'trafficlight', 'xmin': 189, 'ymin': 189, 'xmax': 233,
'ymax': 295}], 'filename': 'road4.png', 'image_size': (267, 400, 3)}
```

```
[7]: # Dictionary that maps class names to IDs
class_name_to_id_mapping = {"trafficlight": 0,
                             "stop": 1,
                             "speedlimit": 2,
                             "crosswalk": 3}

# Convert the info dict to the required yolo format and write it to disk
def convert_to_yolov5(info_dict):
    print_buffer = []

    # For each bounding box
    for b in info_dict["bboxes"]:
        try:
            class_id = class_name_to_id_mapping[b["class"]]
        except KeyError:
            print("Invalid Class. Must be one from ", class_name_to_id_mapping.
↳keys())

        # Transform the bbox co-ordinates as per the format required by YOLO v5
        b_center_x = (b["xmin"] + b["xmax"]) / 2
        b_center_y = (b["ymin"] + b["ymax"]) / 2
        b_width = (b["xmax"] - b["xmin"])
        b_height = (b["ymax"] - b["ymin"])

        # Normalise the co-ordinates by the dimensions of the image
        image_w, image_h, image_c = info_dict["image_size"]
        b_center_x /= image_w
        b_center_y /= image_h
        b_width /= image_w
        b_height /= image_h

        #Write the bbox details to the file
        print_buffer.append("{} {:.3f} {:.3f} {:.3f} {:.3f}".format(class_id,
↳b_center_x, b_center_y, b_width, b_height))

    # Name of the file which we have to save
    save_file_name = os.path.join("./annotations", info_dict["filename"].
↳replace("png", "txt"))
```

```

# Save the annotation to disk
print("\n".join(print_buffer), file= open(save_file_name, "w"))

```

```
[11]: %cd /content/RoadSignDetectionDataset
```

```

/content/RoadSignDetectionDataset
/content/RoadSignDetectionDataset

```

```

[12]: # Get the annotations
annotations = [os.path.join('./annotations', x) for x in os.listdir('./
↳annotations') if x[-3:] == ".xml"]
annotations.sort()

# Convert and save the annotations
for ann in tqdm(annotations):
    info_dict = extract_info_from_xml(ann)
    convert_to_yolov5(info_dict)
annotations = [os.path.join('./annotations', x) for x in os.listdir('./
↳annotations') if x[-3:] == ".txt"]

```

```
100%|          | 877/877 [00:00<00:00, 6939.27it/s]
```

```

[19]: class_id_to_name_mapping = dict(zip(class_name_to_id_mapping.values(),
↳class_name_to_id_mapping.keys()))

def plot_bounding_box(image, annotation_list):
    annotations = np.array(annotation_list)
    w, h = image.size

    plotted_image = ImageDraw.Draw(image)

    transformed_annotations = np.copy(annotations)
    transformed_annotations[:,[1,3]] = annotations[:,[1,3]] * w
    transformed_annotations[:,[2,4]] = annotations[:,[2,4]] * h

    transformed_annotations[:,1] = transformed_annotations[:,1] -
↳(transformed_annotations[:,3] / 2)
    transformed_annotations[:,2] = transformed_annotations[:,2] -
↳(transformed_annotations[:,4] / 2)
    transformed_annotations[:,3] = transformed_annotations[:,1] +
↳transformed_annotations[:,3]
    transformed_annotations[:,4] = transformed_annotations[:,2] +
↳transformed_annotations[:,4]

    for ann in transformed_annotations:
        obj_cls, x0, y0, x1, y1 = ann
        plotted_image.rectangle(((x0,y0), (x1,y1)))

```

```

        plotted_image.text((x0, y0 - 10),
↪class_id_to_name_mapping[(int(obj_cls)]))

plt.imshow(np.array(image))
plt.show()

# Get any random annotation file
annotation_file = random.choice(annotations)
with open(annotation_file, "r") as file:
    annotation_list = file.read().split("\n")[:-1]
    annotation_list = [x.split(" ") for x in annotation_list]
    annotation_list = [[float(y) for y in x ] for x in annotation_list]

#Get the corresponding image file
image_file = annotation_file.replace("annotations", "images").replace("txt",
↪"png")
assert os.path.exists(image_file)

#Load the image
image = Image.open(image_file)

#Plot the Bounding Box
plot_bounding_box(image, annotation_list)

```



```
[20]: # Read images and annotations
images = [os.path.join('images', x) for x in os.listdir('images')]
annotations = [os.path.join('annotations', x) for x in os.
↳listdir('annotations') if x[-3:] == ".txt"]

images.sort()
annotations.sort()

# Split the dataset into train-valid-test splits
train_images, val_images, train_annotations, val_annotations =
↳train_test_split(images, annotations, test_size = 0.2, random_state = 1)
val_images, test_images, val_annotations, test_annotations =
↳train_test_split(val_images, val_annotations, test_size = 0.5, random_state
↳= 1)
```

```
[21]: !mkdir images/train images/val images/test annotations/train annotations/val
↳annotations/test
```

```
[22]: #Utility function to move images
def move_files_to_folder(list_of_files, destination_folder):
    for f in list_of_files:
        try:
            shutil.move(f, destination_folder)
        except:
            print(f)
            assert False

# Move the splits into their folders
move_files_to_folder(train_images, 'images/train')
move_files_to_folder(val_images, 'images/val/')
move_files_to_folder(test_images, 'images/test/')
move_files_to_folder(train_annotations, 'annotations/train/')
move_files_to_folder(val_annotations, 'annotations/val/')
move_files_to_folder(test_annotations, 'annotations/test/')
```

```
[26]: !mv annotations labels
%cd ../yolov5
```

```
mv: cannot stat 'annotations': No such file or directory
/content/yolov5
```

```
[27]: !pwd
```

```
/content/yolov5
```

```
[31]: !python train.py --img 640 --cfg yolov5s.yaml --hyp hyp.scratch-high.yaml
↳--batch 32 --epochs 100 --data road_sign_data.yaml --weights yolov5s.pt
↳--workers 24 --name yolo_road_det
```

```

train: weights=yolov5s.pt, cfg=yolov5s.yaml,
data=road_sign_data.yaml, hyp=hyp.scratch-high.yaml, epochs=100, batch_size=32,
imgsz=640, rect=False, resume=False, nosave=False, noval=False,
noautoanchor=False, noplots=False, evolve=None, bucket=, cache=None,
image_weights=False, device=, multi_scale=False, single_cls=False,
optimizer=SGD, sync_bn=False, workers=24, project=runs/train,
name=yolo_road_det, exist_ok=False, quad=False, cos_lr=False,
label_smoothing=0.0, patience=100, freeze=[0], save_period=-1, seed=0,
local_rank=-1, entity=None, upload_dataset=False, bbox_interval=-1,
artifact_alias=latest

```

[github](https://github.com/ultralytics/yolov5): up to date with <https://github.com/ultralytics/yolov5>
YOLOv5 v6.2-186-g7f097dd Python-3.7.14 torch-1.12.1+cu113 CUDA:0 (Tesla V100-SXM2-16GB, 16160MiB)

```

hyperparameters: lr0=0.01, lrf=0.1, momentum=0.937,
weight_decay=0.0005, warmup_epochs=3.0, warmup_momentum=0.8, warmup_bias_lr=0.1,
box=0.05, cls=0.3, cls_pw=1.0, obj=0.7, obj_pw=1.0, iou_t=0.2, anchor_t=4.0,
fl_gamma=0.0, hsv_h=0.015, hsv_s=0.7, hsv_v=0.4, degrees=0.0, translate=0.1,
scale=0.9, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0,
mixup=0.1, copy_paste=0.1

```

Weights & Biases: run 'pip install wandb' to automatically track and visualize YOLOv5 runs in Weights & Biases

ClearML: run 'pip install clearml' to automatically track, visualize and remotely train YOLOv5 in ClearML

Comet: run 'pip install comet_ml' to automatically track and visualize YOLOv5 runs in Comet

TensorBoard: Start with 'tensorboard --logdir runs/train', view at <http://localhost:6006/>

Downloading <https://ultralytics.com/assets/Arial.ttf> to /root/.config/Ultralytics/Arial.ttf...

100% 755k/755k [00:00<00:00, 51.0MB/s]

Downloading

<https://github.com/ultralytics/yolov5/releases/download/v6.2/yolov5s.pt> to yolov5s.pt...

100% 14.1M/14.1M [00:00<00:00, 41.4MB/s]

Overriding model.yaml nc=80 with nc=4

	from	n	params	module	
arguments					
0	-1	1	3520	models.common.Conv	[3,
32, 6, 2, 2]					
1	-1	1	18560	models.common.Conv	[32,
64, 3, 2]					
2	-1	1	18816	models.common.C3	[64,
64, 1]					
3	-1	1	73984	models.common.Conv	[64,
128, 3, 2]					

4	-1 2	115712	models.common.C3	
[128, 128, 2]				
5	-1 1	295424	models.common.Conv	
[128, 256, 3, 2]				
6	-1 3	625152	models.common.C3	
[256, 256, 3]				
7	-1 1	1180672	models.common.Conv	
[256, 512, 3, 2]				
8	-1 1	1182720	models.common.C3	
[512, 512, 1]				
9	-1 1	656896	models.common.SPPF	
[512, 512, 5]				
10	-1 1	131584	models.common.Conv	
[512, 256, 1, 1]				
11	-1 1	0	torch.nn.modules.upsampling.Upsample	
[None, 2, 'nearest']				
12	[-1, 6] 1	0	models.common.Concat	[1]
13	-1 1	361984	models.common.C3	
[512, 256, 1, False]				
14	-1 1	33024	models.common.Conv	
[256, 128, 1, 1]				
15	-1 1	0	torch.nn.modules.upsampling.Upsample	
[None, 2, 'nearest']				
16	[-1, 4] 1	0	models.common.Concat	[1]
17	-1 1	90880	models.common.C3	
[256, 128, 1, False]				
18	-1 1	147712	models.common.Conv	
[128, 128, 3, 2]				
19	[-1, 14] 1	0	models.common.Concat	[1]
20	-1 1	296448	models.common.C3	
[256, 256, 1, False]				
21	-1 1	590336	models.common.Conv	
[256, 256, 3, 2]				
22	[-1, 10] 1	0	models.common.Concat	[1]
23	-1 1	1182720	models.common.C3	
[512, 512, 1, False]				
24	[17, 20, 23] 1	24273	models.yolo.Detect	[4,
[[10, 13, 16, 30, 33, 23], [30, 61, 62, 45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]				

YOLOv5s summary: 214 layers, 7030417 parameters, 7030417 gradients, 16.0 GFLOPs

Transferred 342/349 items from yolov5s.pt

AMP: checks passed

optimizer: SGD(lr=0.01) with parameter groups 57 weight(decay=0.0), 60 weight(decay=0.0005), 60 bias

augmentations: Blur(p=0.01, blur_limit=(3, 7)), MedianBlur(p=0.01, blur_limit=(3, 7)), ToGray(p=0.01), CLAHE(p=0.01, clip_limit=(1, 4.0), tile_grid_size=(8, 8))

train: Scanning '/content/RoadSignDetectionDataset/labels/train'
 images and labels...701 found, 0 missing, 0 empty, 0 corrupt: 100% 701/701
 [00:00<00:00, 5610.25it/s]
train: New cache created:
 /content/RoadSignDetectionDataset/labels/train.cache
val: Scanning '/content/RoadSignDetectionDataset/labels/val' images
 and labels...88 found, 0 missing, 0 empty, 0 corrupt: 100% 88/88 [00:00<00:00,
 909.22it/s]
val: New cache created:
 /content/RoadSignDetectionDataset/labels/val.cache

AutoAnchor: 5.59 anchors/target, 1.000 Best Possible Recall (BPR).
 Current anchors are a good fit to dataset
 Plotting labels to runs/train/yolo_road_det3/labels.jpg...
 Image sizes 640 train, 640 val
 Using 8 dataloader workers
 Logging results to runs/train/yolo_road_det3
 Starting training for 100 epochs...

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
0/99	6.46G	0.1083	0.02287	0.02311	87	640:
100% 22/22 [00:08<00:00, 2.70it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:02<00:00, 1.48s/it]				
	all	88	132	0.696	0.205	0.17
0.0441						

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
1/99	8.08G	0.06918	0.02491	0.01784	100	640:
100% 22/22 [00:04<00:00, 4.46it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00, 1.50it/s]				
	all	88	132	0.727	0.182	0.184
0.067						

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
2/99	8.08G	0.05932	0.02193	0.01667	112	640:
100% 22/22 [00:04<00:00, 4.75it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00, 1.46it/s]				
	all	88	132	0.623	0.195	0.125
0.0459						

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
3/99	8.08G	0.05778	0.01939	0.01546	95	640:
100% 22/22 [00:05<00:00, 3.87it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00, 1.70it/s]				

0.106	all	88	132	0.738	0.239	0.197
-------	-----	----	-----	-------	-------	-------

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
4/99	8.08G	0.05523	0.01759	0.01424	72	640:
100% 22/22 [00:05<00:00, 4.17it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.95it/s]						
	all	88	132	0.371	0.429	0.441

0.211	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
	5/99	8.08G	0.05157	0.01466	0.01197	105	640:
100% 22/22 [00:06<00:00, 3.34it/s]							
	Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:01<00:00, 1.47it/s]							
	all	88	132	0.466	0.647	0.598	

0.275	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
	6/99	8.08G	0.0483	0.01302	0.0103	93	640:
100% 22/22 [00:04<00:00, 4.85it/s]							
	Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:00<00:00, 2.27it/s]							
	all	88	132	0.832	0.629	0.787	

0.358	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
	7/99	8.08G	0.04301	0.01145	0.008366	93	640:
100% 22/22 [00:05<00:00, 3.89it/s]							
	Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:00<00:00, 2.22it/s]							
	all	88	132	0.741	0.789	0.826	

0.405	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
	8/99	8.08G	0.0423	0.01024	0.007336	93	640:
100% 22/22 [00:05<00:00, 4.03it/s]							
	Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:00<00:00, 2.04it/s]							
	all	88	132	0.863	0.802	0.882	

0.446	Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
	9/99	8.08G	0.03994	0.009085	0.006396	107	640:
100% 22/22 [00:05<00:00, 4.12it/s]							
	Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:01<00:00, 1.57it/s]							

	all	88	132	0.811	0.833	0.846
--	-----	----	-----	-------	-------	-------

0.473

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
10/99	8.08G	0.03822	0.00921	0.005511	82	640:
100% 22/22	[00:05<00:00,	4.04it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:00<00:00,	2.30it/s]			
	all	88	132	0.912	0.882	0.925

0.503

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
11/99	8.08G	0.03782	0.009302	0.005117	106	640:
100% 22/22	[00:06<00:00,	3.39it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00,	1.99it/s]			
	all	88	132	0.839	0.868	0.856

0.501

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
12/99	8.08G	0.03578	0.008653	0.004564	107	640:
100% 22/22	[00:04<00:00,	4.76it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:00<00:00,	2.33it/s]			
	all	88	132	0.931	0.914	0.931

0.527

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
13/99	8.08G	0.03421	0.008664	0.004032	81	640:
100% 22/22	[00:05<00:00,	3.83it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00,	1.81it/s]			
	all	88	132	0.826	0.917	0.909

0.541

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
14/99	8.08G	0.03359	0.008406	0.003831	81	640:
100% 22/22	[00:05<00:00,	3.88it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:00<00:00,	2.15it/s]			
	all	88	132	0.922	0.888	0.94

0.585

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
15/99	8.08G	0.03281	0.008031	0.003723	88	640:
100% 22/22	[00:05<00:00,	3.93it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00,	1.99it/s]			

	all	88	132	0.965	0.88	0.941
--	-----	----	-----	-------	------	-------

0.579

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
16/99	8.08G	0.0319	0.008042	0.003496	92	640:
100% 22/22 [00:05<00:00, 3.98it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.02it/s]						
	all	88	132	0.974	0.905	0.967

0.594

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
17/99	8.08G	0.03269	0.007651	0.003462	72	640:
100% 22/22 [00:05<00:00, 4.38it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.55it/s]						
	all	88	132	0.938	0.941	0.962

0.644

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
18/99	8.08G	0.03105	0.008038	0.003225	129	640:
100% 22/22 [00:05<00:00, 3.71it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.14it/s]						
	all	88	132	0.942	0.937	0.963

0.602

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
19/99	8.08G	0.03093	0.007821	0.00322	78	640:
100% 22/22 [00:06<00:00, 3.54it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.82it/s]						
	all	88	132	0.964	0.953	0.975

0.637

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
20/99	8.08G	0.02957	0.007462	0.002836	95	640:
100% 22/22 [00:05<00:00, 4.15it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.04it/s]						
	all	88	132	0.956	0.938	0.962

0.639

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
21/99	8.08G	0.02988	0.007722	0.00294	72	640:
100% 22/22 [00:06<00:00, 3.65it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.50it/s]						

	all	88	132	0.918	0.924	0.953
--	-----	----	-----	-------	-------	-------

0.614

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
22/99	8.08G	0.0289	0.007662	0.002786	100	640:
100% 22/22	[00:05<00:00,	3.77it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.16it/s]				
	all	88	132	0.975	0.924	0.948

0.606

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
23/99	8.08G	0.02918	0.007114	0.002713	90	640:
100% 22/22	[00:06<00:00,	3.56it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.45it/s]				
	all	88	132	0.981	0.921	0.953

0.597

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
24/99	8.08G	0.02908	0.007546	0.002646	93	640:
100% 22/22	[00:05<00:00,	4.13it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:01<00:00,	1.82it/s]				
	all	88	132	0.95	0.949	0.978

0.695

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
25/99	8.08G	0.02719	0.006949	0.002307	91	640:
100% 22/22	[00:05<00:00,	3.96it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:01<00:00,	1.87it/s]				
	all	88	132	0.951	0.944	0.965

0.672

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
26/99	8.08G	0.02712	0.007153	0.002457	95	640:
100% 22/22	[00:05<00:00,	3.89it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:01<00:00,	1.93it/s]				
	all	88	132	0.942	0.928	0.966

0.686

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
27/99	8.08G	0.02743	0.007378	0.00251	98	640:
100% 22/22	[00:05<00:00,	3.86it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.43it/s]				

	all	88	132	0.981	0.943	0.961
--	-----	----	-----	-------	-------	-------

0.717

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
28/99	8.08G	0.0266	0.00686	0.002348	107	640:
100% 22/22	[00:05<00:00,	3.80it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.33it/s]				
	all	88	132	0.963	0.95	0.974

0.723

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
29/99	8.08G	0.02687	0.007231	0.002493	75	640:
100% 22/22	[00:05<00:00,	3.97it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:01<00:00,	1.88it/s]				
	all	88	132	0.958	0.981	0.989

0.705

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
30/99	8.08G	0.02619	0.007012	0.00217	75	640:
100% 22/22	[00:06<00:00,	3.64it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:01<00:00,	1.56it/s]				
	all	88	132	0.954	0.971	0.98

0.694

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
31/99	8.08G	0.02723	0.007169	0.002381	113	640:
100% 22/22	[00:06<00:00,	3.59it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.47it/s]				
	all	88	132	0.966	0.944	0.977

0.713

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
32/99	8.08G	0.02561	0.007232	0.00223	85	640:
100% 22/22	[00:07<00:00,	3.10it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:01<00:00,	1.77it/s]				
	all	88	132	0.957	0.965	0.98

0.716

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
33/99	8.08G	0.02568	0.006736	0.002239	93	640:
100% 22/22	[00:05<00:00,	4.34it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.16it/s]				

	all	88	132	0.971	0.897	0.966
--	-----	----	-----	-------	-------	-------

0.672

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
34/99	8.08G	0.02528	0.006985	0.002148	84	640:
100% 22/22 [00:05<00:00, 3.85it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.51it/s]						
	all	88	132	0.95	0.926	0.97

0.725

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
35/99	8.08G	0.02457	0.006875	0.001967	78	640:
100% 22/22 [00:05<00:00, 4.13it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.27it/s]						
	all	88	132	0.928	0.965	0.976

0.718

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
36/99	8.08G	0.02456	0.00686	0.002048	129	640:
100% 22/22 [00:07<00:00, 3.07it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.03it/s]						
	all	88	132	0.977	0.92	0.981

0.704

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
37/99	8.08G	0.02389	0.006842	0.002111	81	640:
100% 22/22 [00:05<00:00, 3.77it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.74it/s]						
	all	88	132	0.968	0.943	0.962

0.731

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
38/99	8.08G	0.02478	0.006474	0.001945	92	640:
100% 22/22 [00:05<00:00, 4.31it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.72it/s]						
	all	88	132	0.965	0.924	0.974

0.744

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
39/99	8.08G	0.0245	0.006911	0.001883	103	640:
100% 22/22 [00:05<00:00, 4.05it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.64it/s]						

0.73	all	88	132	0.986	0.924	0.978
------	-----	----	-----	-------	-------	-------

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
40/99	8.08G	0.02318	0.006571	0.001915	87	640:
100% 22/22 [00:06<00:00, 3.34it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.36it/s]						
	all	88	132	0.968	0.948	0.977

0.757						
-------	--	--	--	--	--	--

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
41/99	8.08G	0.02357	0.006809	0.001882	120	640:
100% 22/22 [00:05<00:00, 3.76it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.97it/s]						
	all	88	132	0.983	0.941	0.98

0.728						
-------	--	--	--	--	--	--

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
42/99	8.08G	0.02328	0.00653	0.001925	78	640:
100% 22/22 [00:05<00:00, 3.97it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.02it/s]						
	all	88	132	0.922	0.979	0.974

0.712						
-------	--	--	--	--	--	--

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
43/99	8.08G	0.0228	0.006247	0.001741	80	640:
100% 22/22 [00:05<00:00, 3.93it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.35it/s]						
	all	88	132	0.948	0.962	0.981

0.719						
-------	--	--	--	--	--	--

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
44/99	8.08G	0.02229	0.00612	0.001663	79	640:
100% 22/22 [00:06<00:00, 3.43it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.10it/s]						
	all	88	132	0.889	0.974	0.972

0.738						
-------	--	--	--	--	--	--

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
45/99	8.08G	0.02211	0.006153	0.00182	90	640:
100% 22/22 [00:06<00:00, 3.50it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.24it/s]						

	all	88	132	0.979	0.933	0.975
--	-----	----	-----	-------	-------	-------

0.735

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
46/99	8.08G	0.02258	0.006025	0.001796	105	640:
100% 22/22 [00:06<00:00, 3.29it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.73it/s]						
	all	88	132	0.981	0.944	0.977

0.743

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
47/99	8.08G	0.02194	0.006173	0.001674	59	640:
100% 22/22 [00:06<00:00, 3.60it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.73it/s]						
	all	88	132	0.956	0.966	0.967

0.741

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
48/99	8.08G	0.0216	0.006345	0.001681	90	640:
100% 22/22 [00:05<00:00, 3.80it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.12it/s]						
	all	88	132	0.979	0.931	0.968

0.746

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
49/99	8.08G	0.02123	0.00614	0.001655	82	640:
100% 22/22 [00:05<00:00, 3.88it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.93it/s]						
	all	88	132	0.974	0.968	0.981

0.764

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
50/99	8.08G	0.02203	0.006111	0.001561	89	640:
100% 22/22 [00:05<00:00, 4.02it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.16it/s]						
	all	88	132	0.98	0.925	0.962

0.736

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
51/99	8.08G	0.02125	0.006039	0.001574	57	640:
100% 22/22 [00:05<00:00, 3.67it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.97it/s]						

0.748	all	88	132	0.979	0.962	0.975
-------	-----	----	-----	-------	-------	-------

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
52/99	8.08G	0.02105	0.006075	0.001573	81	640:
100% 22/22 [00:05<00:00, 4.35it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.45it/s]						
	all	88	132	0.971	0.967	0.972

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
53/99	8.08G	0.02138	0.006079	0.001534	88	640:
100% 22/22 [00:06<00:00, 3.22it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.22it/s]						
	all	88	132	0.928	0.973	0.975

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
54/99	8.08G	0.02074	0.00584	0.001547	86	640:
100% 22/22 [00:06<00:00, 3.40it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.68it/s]						
	all	88	132	0.932	0.974	0.97

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
55/99	8.08G	0.02098	0.005952	0.001354	97	640:
100% 22/22 [00:07<00:00, 2.99it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.97it/s]						
	all	88	132	0.942	0.97	0.983

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
56/99	8.08G	0.02078	0.006223	0.001337	107	640:
100% 22/22 [00:05<00:00, 3.75it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.92it/s]						
	all	88	132	0.936	0.971	0.984

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
57/99	8.08G	0.02053	0.00623	0.00144	57	640:
100% 22/22 [00:06<00:00, 3.38it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.04it/s]						

all 88 132 0.918 0.984 0.984
0.763

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
58/99	8.08G	0.02003	0.005945	0.001456	97	640:
100% 22/22 [00:05<00:00, 3.77it/s]						
Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:01<00:00, 1.74it/s]						
all	88	132	0.944	0.959	0.984	

0.763

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
59/99	8.08G	0.02022	0.005878	0.001516	97	640:
100% 22/22 [00:05<00:00, 3.67it/s]						
Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:00<00:00, 2.31it/s]						
all	88	132	0.954	0.97	0.981	

0.766

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
60/99	8.08G	0.01931	0.005838	0.001416	83	640:
100% 22/22 [00:04<00:00, 4.49it/s]						
Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:01<00:00, 1.87it/s]						
all	88	132	0.984	0.914	0.975	

0.765

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
61/99	8.08G	0.02001	0.005658	0.001352	91	640:
100% 22/22 [00:06<00:00, 3.58it/s]						
Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:01<00:00, 1.92it/s]						
all	88	132	0.951	0.97	0.984	

0.77

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
62/99	8.08G	0.01964	0.005909	0.001306	78	640:
100% 22/22 [00:05<00:00, 4.09it/s]						
Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:00<00:00, 2.03it/s]						
all	88	132	0.978	0.979	0.991	

0.782

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
63/99	8.08G	0.01921	0.005922	0.001344	81	640:
100% 22/22 [00:06<00:00, 3.60it/s]						
Class	Images	Instances	P	R	mAP50	
mAP50-95: 100% 2/2 [00:01<00:00, 1.90it/s]						

	all	88	132	0.924	0.983	0.971
--	-----	----	-----	-------	-------	-------

0.776

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
64/99	8.08G	0.018	0.005566	0.001249	108	640:
100% 22/22 [00:05<00:00, 4.07it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.62it/s]						
	all	88	132	0.965	0.934	0.976

0.755

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
65/99	8.08G	0.01843	0.005869	0.001095	95	640:
100% 22/22 [00:06<00:00, 3.38it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.26it/s]						
	all	88	132	0.937	0.984	0.974

0.775

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
66/99	8.08G	0.01883	0.005993	0.001225	85	640:
100% 22/22 [00:06<00:00, 3.24it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.48it/s]						
	all	88	132	0.959	0.98	0.985

0.767

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
67/99	8.08G	0.01945	0.005669	0.001283	98	640:
100% 22/22 [00:06<00:00, 3.60it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.08it/s]						
	all	88	132	0.966	0.95	0.984

0.78

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
68/99	8.08G	0.01815	0.005539	0.00129	87	640:
100% 22/22 [00:05<00:00, 3.76it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.92it/s]						
	all	88	132	0.962	0.933	0.984

0.779

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
69/99	8.08G	0.01852	0.005758	0.001265	102	640:
100% 22/22 [00:05<00:00, 3.82it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.68it/s]						

	all	88	132	0.94	0.962	0.973
--	-----	----	-----	------	-------	-------

0.768

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
70/99	8.08G	0.01787	0.005545	0.001143	96	640:
100% 22/22	[00:06<00:00,	3.58it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.44it/s]				
	all	88	132	0.977	0.947	0.973

0.79

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
71/99	8.08G	0.01814	0.005644	0.001157	110	640:
100% 22/22	[00:05<00:00,	4.03it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.18it/s]				
	all	88	132	0.947	0.952	0.974

0.77

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
72/99	8.08G	0.01739	0.005552	0.000976	80	640:
100% 22/22	[00:05<00:00,	3.88it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.88it/s]				
	all	88	132	0.921	0.961	0.981

0.784

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
73/99	8.08G	0.0177	0.005701	0.001083	101	640:
100% 22/22	[00:06<00:00,	3.57it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.24it/s]				
	all	88	132	0.965	0.911	0.979

0.773

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
74/99	8.08G	0.01706	0.005411	0.001092	71	640:
100% 22/22	[00:06<00:00,	3.29it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:01<00:00,	1.75it/s]				
	all	88	132	0.952	0.955	0.985

0.799

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
75/99	8.08G	0.01731	0.005856	0.001112	107	640:
100% 22/22	[00:05<00:00,	3.86it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2 [00:00<00:00,	2.63it/s]				

	all	88	132	0.95	0.976	0.988
--	-----	----	-----	------	-------	-------

0.792

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
76/99	8.08G	0.01718	0.005293	0.001084	88	640:
100% 22/22	[00:06<00:00,	3.60it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:00<00:00,	2.01it/s]			
	all	88	132	0.956	0.944	0.987

0.802

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
77/99	8.08G	0.01741	0.005503	0.001028	88	640:
100% 22/22	[00:04<00:00,	4.50it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:00<00:00,	2.95it/s]			
	all	88	132	0.95	0.982	0.989

0.783

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
78/99	8.08G	0.01702	0.005364	0.001006	81	640:
100% 22/22	[00:06<00:00,	3.28it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:00<00:00,	2.21it/s]			
	all	88	132	0.971	0.985	0.993

0.786

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
79/99	8.08G	0.01664	0.005333	0.0009713	96	640:
100% 22/22	[00:06<00:00,	3.61it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00,	1.77it/s]			
	all	88	132	0.981	0.965	0.989

0.809

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
80/99	8.08G	0.0164	0.005115	0.0009684	96	640:
100% 22/22	[00:04<00:00,	4.84it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00,	1.53it/s]			
	all	88	132	0.966	0.967	0.988

0.796

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
81/99	8.08G	0.01661	0.005311	0.0009634	90	640:
100% 22/22	[00:05<00:00,	3.78it/s]				
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:00<00:00,	2.11it/s]			

	all	88	132	0.972	0.96	0.988
--	-----	----	-----	-------	------	-------

0.798

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
82/99	8.08G	0.01619	0.00552	0.0009595	106	640:
100% 22/22 [00:06<00:00, 3.16it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.33it/s]						
	all	88	132	0.959	0.98	0.988

0.802

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
83/99	8.08G	0.01606	0.005174	0.0009515	85	640:
100% 22/22 [00:05<00:00, 3.67it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.71it/s]						
	all	88	132	0.939	0.97	0.983

0.792

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
84/99	8.08G	0.01631	0.005249	0.0009134	60	640:
100% 22/22 [00:05<00:00, 4.34it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.51it/s]						
	all	88	132	0.943	0.961	0.977

0.796

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
85/99	8.08G	0.01593	0.005192	0.0009041	98	640:
100% 22/22 [00:06<00:00, 3.39it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.18it/s]						
	all	88	132	0.974	0.928	0.976

0.794

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
86/99	8.08G	0.01652	0.005258	0.0008908	69	640:
100% 22/22 [00:05<00:00, 3.72it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.10it/s]						
	all	88	132	0.902	0.992	0.984

0.787

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
87/99	8.08G	0.0157	0.005241	0.0009021	54	640:
100% 22/22 [00:06<00:00, 3.53it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.18it/s]						

	all	88	132	0.941	0.986	0.985
--	-----	----	-----	-------	-------	-------

0.785

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
88/99	8.08G	0.01559	0.005297	0.0008223	84	640:
100% 22/22 [00:06<00:00, 3.42it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.22it/s]						
	all	88	132	0.971	0.955	0.985

0.795

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
89/99	8.08G	0.01606	0.005096	0.0008322	100	640:
100% 22/22 [00:05<00:00, 4.06it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.81it/s]						
	all	88	132	0.939	0.972	0.984

0.797

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
90/99	8.08G	0.01604	0.005375	0.0008942	75	640:
100% 22/22 [00:06<00:00, 3.48it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.62it/s]						
	all	88	132	0.972	0.969	0.989

0.794

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
91/99	8.08G	0.01549	0.00493	0.0009987	79	640:
100% 22/22 [00:06<00:00, 3.63it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.87it/s]						
	all	88	132	0.976	0.968	0.989

0.799

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
92/99	8.08G	0.015	0.005198	0.0007106	84	640:
100% 22/22 [00:05<00:00, 3.93it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.12it/s]						
	all	88	132	0.977	0.958	0.988

0.793

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
93/99	8.08G	0.01462	0.004842	0.0008006	89	640:
100% 22/22 [00:06<00:00, 3.52it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.05it/s]						

	all	88	132	0.943	0.974	0.981
--	-----	----	-----	-------	-------	-------

0.805

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
94/99	8.08G	0.01492	0.004959	0.0008217	100	640:
100% 22/22 [00:06<00:00, 3.45it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.84it/s]						
	all	88	132	0.99	0.924	0.972

0.797

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
95/99	8.08G	0.01456	0.005134	0.0008545	89	640:
100% 22/22 [00:06<00:00, 3.29it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.54it/s]						
	all	88	132	0.989	0.925	0.972

0.808

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
96/99	8.08G	0.01425	0.004754	0.0006507	68	640:
100% 22/22 [00:05<00:00, 4.09it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.66it/s]						
	all	88	132	0.992	0.928	0.971

0.802

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
97/99	8.08G	0.01453	0.004586	0.0007498	82	640:
100% 22/22 [00:05<00:00, 4.05it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.01it/s]						
	all	88	132	0.985	0.937	0.974

0.819

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
98/99	8.08G	0.01493	0.005011	0.0006505	85	640:
100% 22/22 [00:05<00:00, 3.81it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:01<00:00, 1.76it/s]						
	all	88	132	0.96	0.97	0.985

0.82

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
99/99	8.08G	0.01428	0.004918	0.0009268	90	640:
100% 22/22 [00:05<00:00, 4.35it/s]						
	Class	Images	Instances	P	R	mAP50
mAP50-95: 100% 2/2 [00:00<00:00, 2.60it/s]						

	all	88	132	0.993	0.922	0.983
--	-----	----	-----	-------	-------	-------

0.812

100 epochs completed in 0.215 hours.

Optimizer stripped from runs/train/yolo_road_det3/weights/last.pt, 14.4MB

Optimizer stripped from runs/train/yolo_road_det3/weights/best.pt, 14.4MB

Validating runs/train/yolo_road_det3/weights/best.pt...

Fusing layers...

YOLOv5s summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs

	Class	Images	Instances	P	R	mAP50
mAP50-95: 100%	2/2	[00:01<00:00,	1.27it/s]			
	all	88	132	0.96	0.97	0.985
0.823	trafficlight	88	19	0.894	0.89	0.956
0.626	stop	88	10	0.972	1	0.995
0.923	speedlimit	88	81	0.972	1	0.995
0.906	crosswalk	88	22	1	0.989	0.995

0.835

Results saved to runs/train/yolo_road_det3

```
[43]: model_name = "/content/yolov5/runs/train/yolo_road_det3/weights/best.pt"
test_model = torch.hub.load("ultralytics/yolov5", 'custom', model_name)
test_model.conf = 0.25 # confidence threshold (0-1)
test_model.iou = 0.45
test_model.eval()

test_path = "/content/RoadSignDetectionDataset/"
test_files = glob.glob(test_path + "/images/test/*")
labels = ["trafficlight", "stop", "speedlimit", "crosswalk"]

for files in test_files:
    test_image = cv2.imread(files)
    file_basename = os.path.basename(files)
    print (file_basename)
    txt_name = file_basename[:file_basename.rfind(".")] + ".txt"
    txt_file = open(test_path + "/labels/test/" + txt_name, "r")
    data = txt_file.read()
    data_into_list = data.split("\n")
    #print(data_into_list)

    results = test_model(test_image, size=640)
    results_df = results.pandas().xyxy
    print (results_df[0].name)
```



```
print("\n")
```

```
Using cache found in /root/.cache/torch/hub/ultralytics_yolov5_master  
YOLOv5 2022-10-6 Python-3.7.14 torch-1.12.1+cu113 CUDA:0 (Tesla  
V100-SXM2-16GB, 16160MiB)
```

```
Fusing layers...
```

```
YOLOv5s summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs
```

```
Adding AutoShape...
```

```
road113.png
```

```
0 speedlimit
```

```
Name: name, dtype: object
```

```
road801.png
```

```
0 speedlimit
```

```
Name: name, dtype: object
```

```
road486.png
```

```
0 speedlimit
```

```
1 trafficlight
```

```
Name: name, dtype: object
```

```
road385.png
```

```
0 speedlimit
```

```
Name: name, dtype: object
```

```
road679.png
```

```
0 speedlimit
```

```
Name: name, dtype: object
```

```
road360.png
```

```
0 speedlimit
```

```
Name: name, dtype: object
```

```
road599.png
```

```
0 speedlimit
```

```
Name: name, dtype: object
```

```
road563.png
```

```
0    speedlimit
1    crosswalk
Name: name, dtype: object
```

```
road731.png
0    speedlimit
1    speedlimit
Name: name, dtype: object
```

```
road418.png
0    speedlimit
Name: name, dtype: object
```

```
road316.png
0    speedlimit
1    speedlimit
2    crosswalk
Name: name, dtype: object
```

```
road709.png
0    speedlimit
1    speedlimit
Name: name, dtype: object
```

```
road80.png
0    stop
Name: name, dtype: object
```

```
road50.png
0    trafficlight
Name: name, dtype: object
```

```
road809.png
0    speedlimit
Name: name, dtype: object
```

```
road403.png
0    speedlimit
Name: name, dtype: object
```

road590.png
0 speedlimit
Name: name, dtype: object

road337.png
0 speedlimit
Name: name, dtype: object

road803.png
0 speedlimit
Name: name, dtype: object

road445.png
0 speedlimit
1 speedlimit
Name: name, dtype: object

road417.png
0 speedlimit
Name: name, dtype: object

road661.png
0 speedlimit
1 stop
Name: name, dtype: object

road600.png
0 speedlimit
1 stop
Name: name, dtype: object

road206.png
0 speedlimit
Name: name, dtype: object

road18.png
0 trafficlight
Name: name, dtype: object

road129.png
0 crosswalk
1 trafficlight
Name: name, dtype: object

road796.png
0 speedlimit
Name: name, dtype: object

road543.png
0 speedlimit
1 crosswalk
Name: name, dtype: object

road854.png
0 speedlimit
Name: name, dtype: object

road819.png
0 stop
1 trafficlight
2 trafficlight
Name: name, dtype: object

road321.png
0 speedlimit
1 crosswalk
2 stop
Name: name, dtype: object

road433.png
0 speedlimit
1 stop
Name: name, dtype: object

road367.png
0 speedlimit
Name: name, dtype: object

road67.png
0 stop
Name: name, dtype: object

road37.png
0 trafficlight
Name: name, dtype: object

road174.png
0 speedlimit
Name: name, dtype: object

road356.png
0 stop
Name: name, dtype: object

road199.png
0 speedlimit
Name: name, dtype: object

road494.png
0 speedlimit
Name: name, dtype: object

road408.png
0 speedlimit
Name: name, dtype: object

road315.png
0 speedlimit
1 crosswalk
Name: name, dtype: object

road293.png
0 speedlimit
1 crosswalk
Name: name, dtype: object

road804.png

0 speedlimit
Name: name, dtype: object

road318.png
0 speedlimit
1 crosswalk
Name: name, dtype: object

road764.png
0 speedlimit
Name: name, dtype: object

road91.png
0 stop
1 speedlimit
Name: name, dtype: object

road267.png
0 speedlimit
Name: name, dtype: object

road667.png
0 speedlimit
Name: name, dtype: object

road142.png
0 crosswalk
Name: name, dtype: object

road564.png
0 speedlimit
1 crosswalk
Name: name, dtype: object

road160.png
0 stop
1 crosswalk
Name: name, dtype: object

```
road665.png
0    speedlimit
Name: name, dtype: object
```

```
road672.png
0    speedlimit
1          stop
Name: name, dtype: object
```

```
road498.png
0    speedlimit
Name: name, dtype: object
```

```
road183.png
0    crosswalk
1    crosswalk
2    trafficlight
Name: name, dtype: object
```

```
road3.png
0    trafficlight
1    trafficlight
Name: name, dtype: object
```

```
road558.png
0    speedlimit
Name: name, dtype: object
```

```
road643.png
0    speedlimit
Name: name, dtype: object
```

```
road695.png
0    speedlimit
1    speedlimit
2          stop
Name: name, dtype: object
```

```
road261.png
0    speedlimit
```

Name: name, dtype: object

road609.png

0 speedlimit

1 crosswalk

Name: name, dtype: object

road789.png

0 speedlimit

Name: name, dtype: object

road490.png

0 speedlimit

Name: name, dtype: object

road328.png

0 speedlimit

Name: name, dtype: object

road164.png

0 trafficlight

Name: name, dtype: object

road43.png

0 trafficlight

1 trafficlight

2 trafficlight

3 trafficlight

4 trafficlight

5 trafficlight

6 trafficlight

7 trafficlight

8 trafficlight

Name: name, dtype: object

road617.png

0 speedlimit

Name: name, dtype: object

road157.png

0 speedlimit
Name: name, dtype: object

road143.png
0 crosswalk
Name: name, dtype: object

road841.png
0 speedlimit
1 crosswalk
Name: name, dtype: object

road194.png
0 crosswalk
1 trafficlight
Name: name, dtype: object

road47.png
0 trafficlight
Name: name, dtype: object

road96.png
0 stop
Name: name, dtype: object

road550.png
0 speedlimit
Name: name, dtype: object

road748.png
0 speedlimit
1 speedlimit
Name: name, dtype: object

road784.png
0 speedlimit
1 crosswalk
Name: name, dtype: object

road640.png
0 speedlimit
1 speedlimit
Name: name, dtype: object

road721.png
0 speedlimit
Name: name, dtype: object

road492.png
0 speedlimit
Name: name, dtype: object

road100.png
0 speedlimit
Name: name, dtype: object

road712.png
0 speedlimit
1 speedlimit
Name: name, dtype: object

road140.png
0 crosswalk
Name: name, dtype: object

road730.png
0 speedlimit
1 speedlimit
Name: name, dtype: object

road431.png
0 speedlimit
Name: name, dtype: object

road547.png
0 speedlimit
1 crosswalk
2 crosswalk
Name: name, dtype: object

```
road33.png
0    trafficlight
Name: name, dtype: object
```

```
road175.png
0    speedlimit
Name: name, dtype: object
```

```
road625.png
0    speedlimit
Name: name, dtype: object
```

```
[42]: !python val.py --weights runs/train/yolo_road_det3/weights/best.pt --data_
      ↪road_sign_data.yaml --task test --name yolo_det
```

```
val: data=/content/yolov5/data/road_sign_data.yaml,
weights=['runs/train/yolo_road_det3/weights/best.pt'], batch_size=32, imgsz=640,
conf_thres=0.001, iou_thres=0.6, max_det=300, task=test, device=, workers=8,
single_cls=False, augment=False, verbose=False, save_txt=False,
save_hybrid=False, save_conf=False, save_json=False, project=runs/val,
name=yolo_det, exist_ok=False, half=False, dnn=False
YOLOv5 v6.2-186-g7f097dd Python-3.7.14 torch-1.12.1+cu113 CUDA:0 (Tesla
V100-SXM2-16GB, 16160MiB)
```

Fusing layers...

YOLOv5s summary: 157 layers, 7020913 parameters, 0 gradients, 15.8 GFLOPs

test: Scanning '/content/RoadSignDetectionDataset/labels/test'

images and labels...88 found, 0 missing, 0 empty, 0 corrupt: 100% 88/88

[00:00<00:00, 2895.35it/s]

test: New cache created:

/content/RoadSignDetectionDataset/labels/test.cache

	Class	Images	Instances	P	R	mAP50
mAP50-95:	100% 3/3	[00:02<00:00,	1.14it/s]			
0.795	all	88	126	0.993	0.901	0.952
0.554	trafficlight	88	20	1	0.647	0.828
0.889	stop	88	7	0.986	1	0.995
0.9	speedlimit	88	76	0.994	1	0.995
	crosswalk	88	23	0.992	0.957	0.99

0.837

Speed: 0.3ms pre-process, 2.0ms inference, 0.9ms NMS per image at shape (32, 3, 640, 640)

Results saved to **runs/val/yolo_det2**

[]: