

Report of the assignment:

- **Number of layers and parameters:**

```
Model Summary: 283 layers, 7101300 parameters, 7101300 gradients, 16.5 GFLOPS
```

- **Details of each layer:**

```
# parameters
nc: 80 # number of classes
depth_multiple: 0.33 # model depth multiple
width_multiple: 0.50 # layer channel multiple

# anchors
anchors:
  - [10,13, 16,30, 33,23] # P3/8
  - [30,61, 62,45, 59,119] # P4/16
  - [116,90, 156,198, 373,326] # P5/32

# YOLOv5 backbone
backbone:
  # [from, number, module, args]
  [[-1, 1, Focus, [64, 3]], # 0-P1/2
  [-1, 1, Conv, [128, 3, 2]], # 1-P2/4
  [-1, 3, C3, [128]],
  [-1, 1, Conv, [256, 3, 2]], # 3-P3/8
  [-1, 9, C3, [256]],
  [-1, 1, Conv, [512, 3, 2]], # 5-P4/16
  [-1, 9, C3, [512]],
  [-1, 1, Conv, [1024, 3, 2]], # 7-P5/32
  [-1, 1, SPP, [1024, [5, 9, 13]]],
  [-1, 3, C3, [1024, False]], # 9
  ]

# YOLOv5 head
head:
  [[-1, 1, Conv, [512, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 6], 1, Concat, [1]], # cat backbone P4
  [-1, 3, C3, [512, False]], # 13

  [-1, 1, Conv, [256, 1, 1]],
  [-1, 1, nn.Upsample, [None, 2, 'nearest']],
  [[-1, 4], 1, Concat, [1]], # cat backbone P3
  [-1, 3, C3, [256, False]], # 17 (P3/8-small)

  [-1, 1, Conv, [256, 3, 2]],
  [[-1, 14], 1, Concat, [1]], # cat head P4
  [-1, 3, C3, [512, False]], # 20 (P4/16-medium)

  [-1, 1, Conv, [512, 3, 2]],
  [[-1, 10], 1, Concat, [1]], # cat head P5
  [-1, 3, C3, [1024, False]], # 23 (P5/32-large)

  [[17, 20, 23], 1, Detect, [nc, anchors]], # Detect(P3, P4, P5)
  ]
```

- **Backbone: feature extraction**
- **Neck: perform aggregation to the features**
- **Head: generate predictions from the anchor boxes for object detection.**

- **Activation functions:**

The Leaky ReLU activation function is used in middle/hidden layers and the sigmoid activation function is used in the final detection layer

- **Optimizers:** SGD and ADAM

- **Anchor boxes:** There are 3 anchor boxes

- **Loss function:** Binary Cross-Entropy with Logits Loss function



Predicted image:

- **In the assignment:**

```
# when we ran this, we saw .007 second inference time. That is 140 FPS on a TESLA P100!
# use the best weights!
%cd /content/yolov5/
!python detect.py --weights /content/yolov5/runs/train/yolov5s_results7/weights/best.pt --img 416 --conf 0.4 --source /content/yolov5/test/images

/content/yolov5
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.4, device='', exist_ok=False, img_size=416, iou_thres=0.45, name='exp', project='YOLOv5 v4.0-126-g886f1c0 torch 1.11.0+cu113 CUDA:0 (Tesla T4, 15109.75MB)')

Fusing layers...
/usr/local/lib/python3.7/dist-packages/torch/functional.py:568: UserWarning: torch.meshgrid: in an upcoming release, it will be required to pass the in
return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Model Summary: 224 layers, 7266973 parameters, 0 gradients, 17.0 GFLOPS
image 1/118 /content/yolov5/test/images/000000000009.jpg: 320x416 Done. (0.046s)
image 2/118 /content/yolov5/test/images/000000000025.jpg: 288x416 1 giraffe, Done. (0.039s)
image 3/118 /content/yolov5/test/images/000000000030.jpg: 288x416 Done. (0.015s)
image 4/118 /content/yolov5/test/images/000000000034.jpg: 288x416 1 zebra, Done. (0.016s)
image 5/118 /content/yolov5/test/images/000000000042.jpg: 320x416 Done. (0.015s)
image 6/118 /content/yolov5/test/images/000000000049.jpg: 416x320 2 persons, Done. (0.037s)
image 7/118 /content/yolov5/test/images/000000000061.jpg: 320x416 2 persons, Done. (0.018s)
image 8/118 /content/yolov5/test/images/000000000064.jpg: 416x320 1 clock, 1 stop sign, Done. (0.015s)
image 9/118 /content/yolov5/test/images/000000000071.jpg: 288x416 Done. (0.017s)
image 10/118 /content/yolov5/test/images/000000000072.jpg: 416x288 1 giraffe, Done. (0.038s)
image 11/118 /content/yolov5/test/images/000000000073.jpg: 416x384 Done. (0.037s)
image 12/118 /content/yolov5/test/images/000000000074.jpg: 288x416 Done. (0.018s)
image 13/118 /content/yolov5/test/images/000000000078.jpg: 416x416 Done. (0.016s)
image 14/118 /content/yolov5/test/images/000000000081.jpg: 288x416 1 aeroplane, Done. (0.016s)
image 15/118 /content/yolov5/test/images/000000000092.jpg: 288x416 Done. (0.016s)
image 16/118 /content/yolov5/test/images/000000000094.jpg: 288x416 2 cars, Done. (0.016s)
image 17/118 /content/yolov5/test/images/000000000109.jpg: 288x416 1 person, Done. (0.016s)
image 18/118 /content/yolov5/test/images/000000000127.jpg: 320x416 Done. (0.016s)
image 19/118 /content/yolov5/test/images/000000000133.jpg: 320x416 Done. (0.016s)
image 20/118 /content/yolov5/test/images/000000000138.jpg: 384x416 1 clock, Done. (0.038s)
image 21/118 /content/yolov5/test/images/000000000142.jpg: 416x320 1 orange, Done. (0.016s)
image 22/118 /content/yolov5/test/images/000000000143.jpg: 352x416 1 person, Done. (0.037s)
```







■ Bonus:

```
# when we ran this, we saw .007 second inference time. That is 140 FPS on a TESLA P100!
# use the best weights!
%cd /content/yolov5/
!python detect.py --weights /content/yolov5/runs/train/yolov5s_results/weights/best.pt --img 416 --conf 0.4 --source /content/yolov5/test/images

/content/yolov5
Namespace(agnostic_nms=False, augment=False, classes=None, conf_thres=0.4, device='', exist_ok=False, img_size=416, iou_thres=0.45, name='exp', project='runs/dt
YOLOv5 v4.0-126-g886f1c8 torch 1.11.0+cu113 CUDA:0 (Tesla T4, 15109.75MB)

Fusing layers...
/usr/local/lib/python3.7/dist-packages/torch/functional.py:568: UserWarning: torch.meshgrid: in an upcoming release, it will be required to pass the indexing ar
    return _VF.meshgrid(tensors, **kwargs) # type: ignore[attr-defined]
Model Summary: 224 layers, 7091668 parameters, 0 gradients, 16.4 GFLOPS
image 1/899 /content/yolov5/test/images/000090623.jpg.rf.f0956cd690e13eeb4614b8bf89df3de.jpg: 416x416 1 1, 1 9, Done. (0.015s)
image 2/899 /content/yolov5/test/images/000617146.jpg.rf.f6fa8b23166ca61c408bfe4b4c32cc2.jpg: 416x416 1 0, 1 2, 1 7, Done. (0.016s)
image 3/899 /content/yolov5/test/images/002147484.jpg.rf.79c3ff8a0cac123c8faf22e078616943.jpg: 416x416 1 14, 1 4, 1 9, Done. (0.015s)
image 4/899 /content/yolov5/test/images/003571947.jpg.rf.4953b6196850dd8954c6c2941963ca36.jpg: 416x416 1 10, 1 11, Done. (0.016s)
image 5/899 /content/yolov5/test/images/003929802.jpg.rf.bbd05947c8bb699912e528f394ea122f.jpg: 416x416 1 13, 2 4s, Done. (0.015s)
image 6/899 /content/yolov5/test/images/004416543.jpg.rf.66e201ed8da25aaf40c7ba522032b80c.jpg: 416x416 1 1, 1 7, Done. (0.015s)
image 7/899 /content/yolov5/test/images/006431818.jpg.rf.1d81de238d72a3119fes299a12a25dccc.jpg: 416x416 1 1, 1 8, 1 9, Done. (0.015s)
image 8/899 /content/yolov5/test/images/006645394.jpg.rf.d06196079c85fb56366c068144769aef.jpg: 416x416 1 1, 1 11, 1 4, Done. (0.015s)
image 9/899 /content/yolov5/test/images/007841444.jpg.rf.9b516c8fca247b1f6384c19a289bad33.jpg: 416x416 1 1, 1 10, 1 9, Done. (0.015s)
image 10/899 /content/yolov5/test/images/007987475.jpg.rf.0a0ea98dbb92231c57d64e85b166fe.jpg: 416x416 1 0, 2 5s, Done. (0.016s)
image 11/899 /content/yolov5/test/images/010208206.jpg.rf.2cf748708744390810ce26b07cc733ca.jpg: 416x416 2 12s, 1 13, Done. (0.016s)
image 12/899 /content/yolov5/test/images/012156352.jpg.rf.f2abes7e3c0f0ab52e0b9e85906940e4.jpg: 416x416 1 11, 1 7, 1 9, Done. (0.015s)
image 13/899 /content/yolov5/test/images/013432108.jpg.rf.a309fb3dc1f34911dd7fb0610ec41ef7.jpg: 416x416 1 0, 1 8, Done. (0.015s)
image 14/899 /content/yolov5/test/images/014502708.jpg.rf.adf8e925c9fc830c58c3dd6f09f2185.jpg: 416x416 2 13s, 1 7, Done. (0.015s)
image 15/899 /content/yolov5/test/images/015012266.jpg.rf.5f3e6d917bd87df1dd1b102a463e3f.jpg: 416x416 2 0s, 1 12, Done. (0.023s)
image 16/899 /content/yolov5/test/images/015307783.jpg.rf.f126979318efa3a82ca2e0297d9e24d3.jpg: 416x416 1 11, 1 14, Done. (0.015s)
image 17/899 /content/yolov5/test/images/015884271.jpg.rf.eaffc811e7d0ecb51cd8177af14d739d.jpg: 416x416 1 12, 1 13, 1 14, Done. (0.015s)
image 18/899 /content/yolov5/test/images/016710983.jpg.rf.7541a09568643f705ee7542eed05566d.jpg: 416x416 1 4, 1 9, Done. (0.015s)
image 19/899 /content/yolov5/test/images/017701459.jpg.rf.h271aa422f4h9770h2f5f0he254a5535.jpg: 416x416 1 10, 2 11s, 1 2, Done. (0.016s)
```





