Assignment 4

Mohamed Faisal 20190711

Fatima salih 20190744

The dataset used:

Hard Hat Sample Dataset The Hard Hat dataset is an object detection dataset of workers in workplace settings that require a hard hat. Annotations also include examples of just "person" and "head," for when an individual may be present without a hard hart. The classes are:

- head
- helmet
- person

The dataset size is 240 samples (210 train-20 validate-10 test) The dataset link : Hard Hat Sample - v2 augmented-416x416 (roboflow.com)

Yolov5s Architecture used:

YOLOv5-s Model Architecture

ReLU activation function is used in middle/hidden layers and the sigmoid activation function is used in the final detection layer. Optimization Functions used are SGD and Adam Loss Function used Binary Cross-Entropy with Logits Loss Model Summary: 283 layers, 7260488 parameters, 7260488 gradients

```
# clone YOLOv5 repository
!git clone https://github.com/ultralytics/yolov5  # clone repo
%cd yolov5
!git reset --hard 886f1c03d839575afecb059accf74296fad395b6


# install dependencies as necessary
!pip install -qr requirements.txt  # install dependencies (ignore errors)
import torch
```

```
import torch

from IPython.display import Image, clear_output  # to display images
from utils.google_utils import gdrive_download  # to download models/datasets

# clear_output()
print('Setup complete. Using torch %s %s' % (torch.__version__, torch.cuda.get_device_properties(0) if torch.cuda.is_availab


#follow the link below to get your download code from from Roboflow
!pip install -q roboflow
from roboflow import Roboflow
rf = Roboflow(model_format="yolov5", notebook="roboflow-yolov5")


%cd /content/yolov5

!pip install roboflow

from roboflow import Roboflow
rf = Roboflow(api_key="Oh10X1sKUjzj0py521En")
project = rf.workspace("fu-ai").project("hard-hat-sample-37zr8")
dataset = project.version(2).download("yolov5")



# this is the YAML file Roboflow wrote for us that we're loading into this notebook with our data
%cat {dataset.location}/data.yaml


# define number of classes based on YAML
import yaml
with open(dataset.location + "/data.yaml", 'r') as stream:
    num_classes = str(yaml.safe_load(stream)['nc'])


#this is the model configuration we will use for our tutorial
%cat /content/yolov5/models/yolov5s.yaml

     cat: /content/yolov5/models/yolov5s.yaml: No such file or directory
```

```
#customize iPython writefile so we can write variables
from IPython.core.magic import register_line_cell_magic

@register_line_cell_magic
def writetemplate(line, cell):
    with open(line, 'w') as f:
        f.write(cell.format(**globals()))



%%writetemplate /content/yolov5/models/custom_yolov5s.yaml

# parameters
nc: {num_classes}  # 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, BottleneckCSP, [128]],
   [-1, 1, Conv, [256, 3, 2]],  # 3-P3/8
   [-1, 9, BottleneckCSP, [256]],
   [-1, 1, Conv, [512, 3, 2]],  # 5-P4/16
   [-1, 9, BottleneckCSP, [512]],
   [-1, 1, Conv, [1024, 3, 2]],  # 7-P5/32
   [-1, 1, SPP, [1024, [5, 9, 13]]],
   [-1, 3, BottleneckCSP, [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, BottleneckCSP, [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, BottleneckCSP, [256, False]],  # 17 (P3/8-small)

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

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

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

# train yolov5s on custom data for 100 epochs
# time its performance
%%time
%cd /content/yolov5/
!python train.py --img 416 --batch 16 --epochs 100 --data {dataset.location}/data.yaml --cfg ./models/custom_yolov5s.yaml --


# first, display our ground truth data
print("GROUND TRUTH TRAINING DATA:")
Image(filename='/content/yolov5/runs/train/yolov5s_results3/test_batch0_labels.jpg', width=900)
```

GROUND TRUTH TRAINING DATA:

```
# print out an augmented training example
print("GROUND TRUTH AUGMENTED TRAINING DATA:")
Image(filename='/content/yolov5/runs/train/yolov5s_results3/train_batch0.jpg', width=900)
```

GROUND TRUTH AUGMENTED TRAINING DATA:

```
# trained weights are saved by default in our weights folder
%ls runs/
```

```
train/
```

```
%ls runs/train/yolov5s_results/weights
```

```
best.pt  last.pt
```

```
# 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_results3/weights/best.pt --img 416 --conf 0.4 --source /conte


alse, augment=False, classes=None, conf_thres=0.4, device='', exist_ok=False, img_size=416, iou_thres=0.45, name='exp',
 torch 1.11.0+cu113 CUDA:0 (Tesla T4, 15109.75MB)
```

```
/dist-packages/torch/functional.py:568: UserWarning: torch.meshgrid: in an upcoming release, it will be required to pass
nsors, **kwargs)  # type: ignore[attr-defined]
s, 7251912 parameters, 0 gradients, 16.8 GFLOPS
v5/Hard-Hat-Sample-2/test/images/000008_jpg.rf.TPtjOiq3AcNMEojFF8PO.jpg: 416x416 Done. (0.018s)
v5/Hard-Hat-Sample-2/test/images/000011_jpg.rf.PC57UVL5iXuLZuTpyuzy.jpg: 416x416 4 helmets, Done. (0.024s)
v5/Hard-Hat-Sample-2/test/images/000034_jpg.rf.gsQ2cbICzAK5a83rpPcw.jpg: 416x416 Done. (0.020s)
v5/Hard-Hat-Sample-2/test/images/000047_jpg.rf.PabvlROii29pDIOmdSTE.jpg: 416x416 1 helmet, Done. (0.020s)
v5/Hard-Hat-Sample-2/test/images/000054_jpg.rf.3qm73oJKaPPb5t83ZZBr.jpg: 416x416 Done. (0.020s)
v5/Hard-Hat-Sample-2/test/images/000073_jpg.rf.LvqbSW3nTL1aRMDLwxyj.jpg: 416x416 8 helmets, Done. (0.020s)
v5/Hard-Hat-Sample-2/test/images/000076_jpg.rf.psRdB6QEQMjnvWI2Gewn.jpg: 416x416 Done. (0.021s)
v5/Hard-Hat-Sample-2/test/images/000084_jpg.rf.3vV3vKF1iwZqkLx1rl26.jpg: 416x416 6 helmets, Done. (0.019s)
v5/Hard-Hat-Sample-2/test/images/000097_jpg.rf.BUQCk47SqieWAbg0qyBA.jpg: 416x416 1 helmet, Done. (0.020s)
ov5/Hard-Hat-Sample-2/test/images/000098_jpg.rf.IExBMipCviMUWjMFToj9.jpg: 416x416 1 head, 2 helmets, Done. (0.021s)
tect/exp
```

```
#display inference on ALL test images
#this looks much better with longer training above

import glob
from IPython.display import Image, display

for imageName in glob.glob('/content/yolov5/runs/detect/exp/*.jpg'): #assuming JPG
    display(Image(filename=imageName))
    print("\n")
```


helmet 0.72
helmet 0.72
helmet 0.65


helmhead 0.84


helmet 0.92

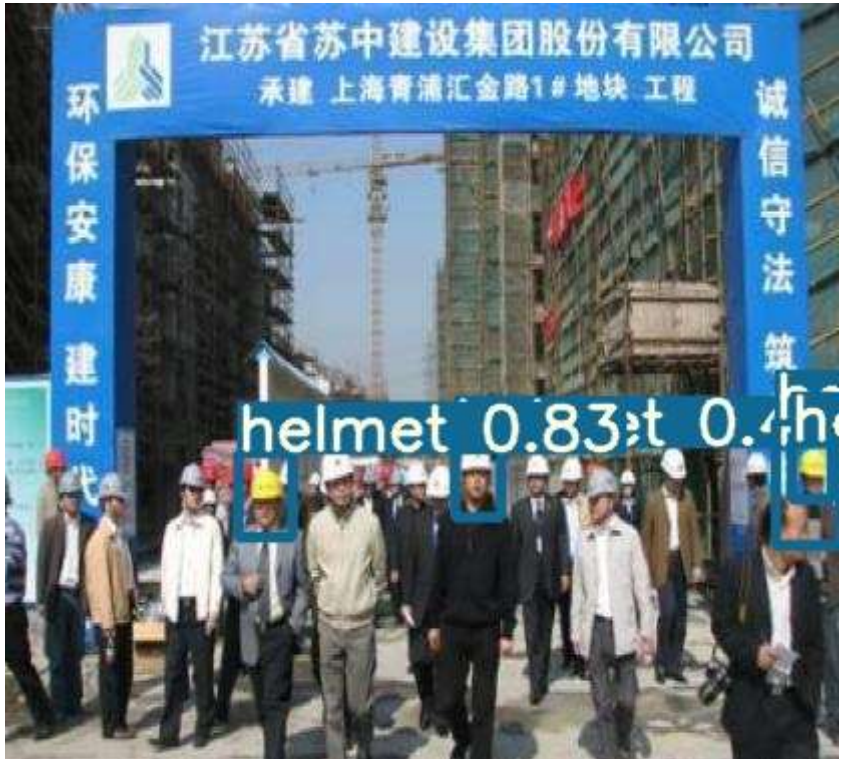

编号：3081543 红动中国 (www.redocn.com) 一颗松


helmet 0.8
helmet
helmet 0.7


遂宁新闻网


成都消防


helme
红网交通
jt.rednet.cn


江苏省苏中建设集团股份有限公司
承建 上海青浦汇金路1#地块 工程
环保安康建时代
诚信守法筑
helmet 0.83