

# Face Mask Detection with YOLOv5

```
In [ ]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import tensorflow as tf
from tensorflow import keras
import cv2
import seaborn as sns
import PIL

import os
for dirname, _, filenames in os.walk('/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

Preparing directories

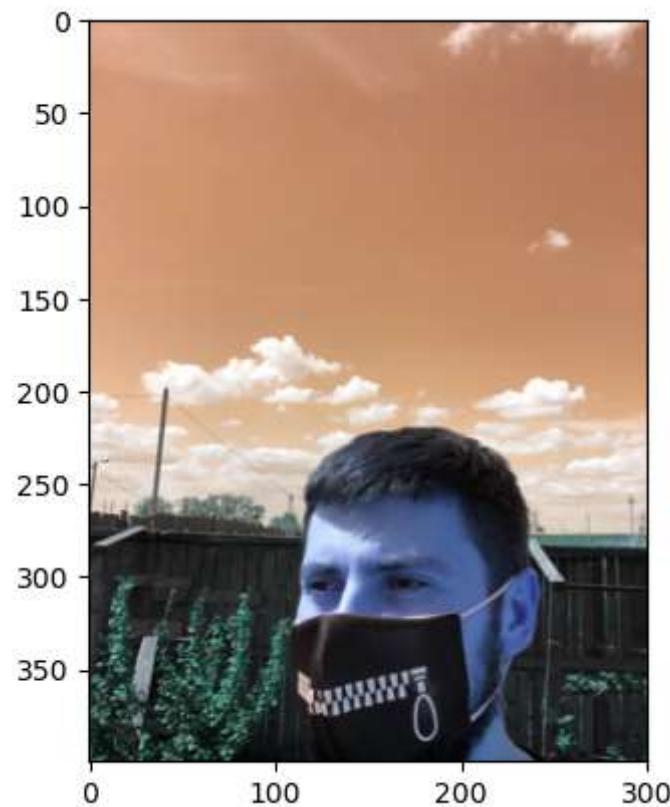
```
In [2]: # Path
img_dir = '/input/face-mask-detection/images'
"""
annotation_directory contains files, and each file is associated to only one image,
and it contains the height and width of the image and also xmin,ymin,xmax, and ymax of each boundary box
inside the image
"""
annotation_dir = '/input/face-mask-detection/annotations'

input_dir = '/input/face-mask-detection'
output_dir = '/working/'
```

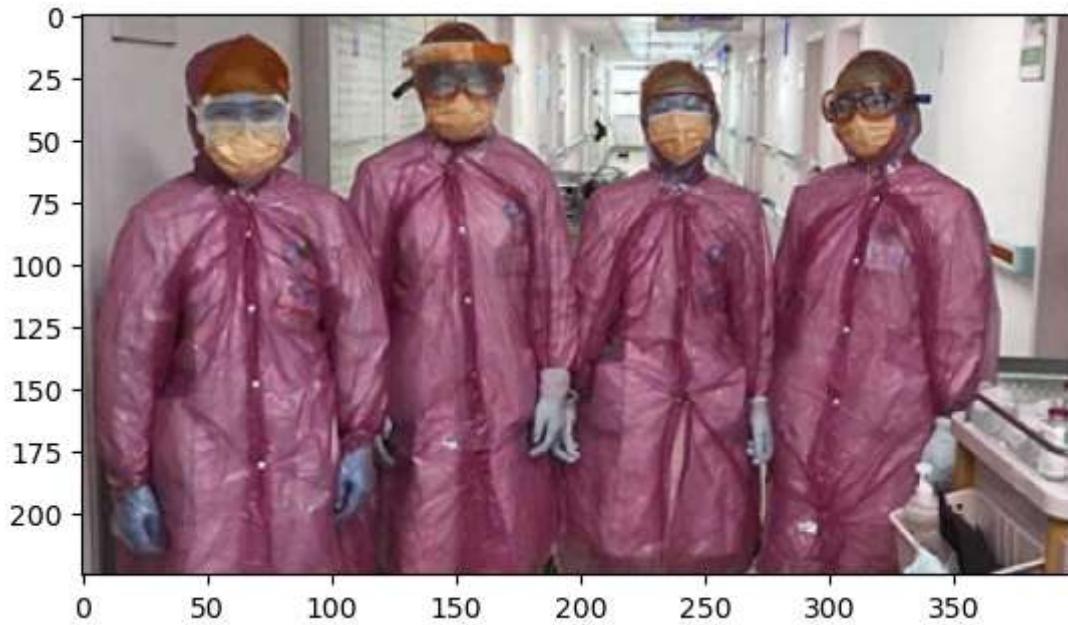
Viewing some images

```
In [3]: # sample visualization
for idx, image in enumerate(os.listdir(img_dir)):
    img = cv2.imread(os.path.join(img_dir, image))
    plt.imshow(img)
    plt.show()

    if idx == 4:
        break
```







Storing images paths

```
In [4]: img_file_path = []
for img in os.listdir(img_dir): # img here is the name of the image not the image itself
    image = cv2.imread(os.path.join(img_dir,img), 0) # 0 for grayscale
    img_file_path.append(f'{img}')
```

libpng warning: iccp: Not recognizing known sRGB profile that has been edited

Libraries needed for reading XML files

```
In [5]: import xml.etree.ElementTree as ET
"""
ElementTree module provides
a way to work with Extensible Markup Language (XML) documents as a tree-like structure of elements.
"""
```

Reading XML files

```
In [6]: df = { 'name': [],
          'label': [],
          'width': [],
          'height': [],
```

```

'xmin': [],
'ymin': [],
'xmax': [],
'ymax': []}
"""

The glob.glob() function returns a list of all the pathnames that match the specified pattern.
The resulting list can then be used to process or analyze the files that match the pattern.
"""
for idx, anno in enumerate(glob.glob(annotation_dir + '/*.xml')):
    trees = ET.parse(anno)

    #print(anno) print/view the annotation to understand the following code
    root = trees.getroot()
    width, height = [], []
    for item in root.iter():
        if item.tag == 'size':
            for attr in list(item):
                if attr.tag == 'width':
                    width = int(round(float(attr.text)))
                if attr.tag == 'height':
                    height = int(round(float(attr.text)))

        if item.tag == 'object':
            for attr in list(item):
                if 'name' in attr.tag:
                    label = attr.text
                    df['label'] += [label]
                    df['width'] += [width]
                    df['height'] += [height]
                    #dataset['name']+=[anno.split('/')[-1][0:-4]]
                    df['name'] += [anno.split('/')[-1][0:-4]]

                if 'bndbox' in attr.tag:
                    for dim in attr:
                        if dim.tag == 'xmin':
                            xmin = int(round(float(dim.text)))
                            df['xmin'] += [xmin]

                        if dim.tag == 'ymin':
                            ymin = int(round(float(dim.text)))
                            df['ymin'] += [ymin]
                        if dim.tag == 'xmax':
                            xmax = int(round(float(dim.text)))
                            df['xmax'] += [xmax]
                        if dim.tag == 'ymax':

```

```
ymax = int(round(float(dim.text)))
df['ymax'] += [ymax]
```

## Viewing data frame of images

```
In [7]: df1 = pd.DataFrame(df)
df1.head()
```

```
Out[7]:
```

	name	label	width	height	xmin	ymin	xmax	ymax
0	makssksksss737	with_mask	400	226	28	55	46	71
1	makssksksss737	with_mask	400	226	98	62	111	78
2	makssksksss737	mask_weared_incorrect	400	226	159	50	193	90
3	makssksksss737	with_mask	400	226	293	59	313	80
4	makssksksss737	with_mask	400	226	352	51	372	72

## Describing data frame

```
In [8]: df1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4072 entries, 0 to 4071
Data columns (total 8 columns):
 #   Column   Non-Null Count  Dtype  
--- 
 0   name     4072 non-null   object 
 1   label    4072 non-null   object 
 2   width    4072 non-null   int64  
 3   height   4072 non-null   int64  
 4   xmin     4072 non-null   int64  
 5   ymin     4072 non-null   int64  
 6   xmax     4072 non-null   int64  
 7   ymax     4072 non-null   int64  
dtypes: int64(6), object(2)
memory usage: 254.6+ KB
```

## Maping labels to integers

```
In [9]: label_map = { 'without_mask': 0,
                  'with_mask': 1,
                  'mask_weared_incorrect': 2}

df1['class'] = df1['label'].map(label_map)
```

```
In [10]: df1.head()
```

```
Out[10]:
```

	name	label	width	height	xmin	ymin	xmax	ymax	class
0	makssssksksss737	with_mask	400	226	28	55	46	71	1
1	makssssksksss737	with_mask	400	226	98	62	111	78	1
2	makssssksksss737	mask_weared_incorrect	400	226	159	50	193	90	2
3	makssssksksss737	with_mask	400	226	293	59	313	80	1
4	makssssksksss737	with_mask	400	226	352	51	372	72	1

### Splitting the data

```
In [11]: # split train, test, val data
from sklearn.model_selection import train_test_split

train, test = train_test_split(img_file_path, test_size=0.2, random_state=101)
train, val = train_test_split(train, test_size=0.15, random_state=101)
```

### Preparing yolo v5 model

```
In [12]: # yolo v5
!git clone https://github.com/ultralytics/yolov5
%cd yolov5
!pip install -qr requirements.txt
```

```
Cloning into 'yolov5'...
remote: Enumerating objects: 16008, done.
remote: Counting objects: 100% (41/41), done.
remote: Compressing objects: 100% (28/28), done.
remote: Total 16008 (delta 22), reused 22 (delta 13), pack-reused 15967
Receiving objects: 100% (16008/16008), 14.60 MiB | 32.43 MiB/s, done.
Resolving deltas: 100% (10988/10988), done.
/kaggle/working/yolov5
ERROR: pip's dependency resolver does not currently take into account all the packages that are installed. This behavior is the source of the following dependency conflicts.
opentelemetry-api 1.18.0 requires importlib-metadata~=6.0.0, but you have importlib-metadata 6.7.0 which is incompatible.
pymc3 3.11.5 requires numpy<1.22.2,>=1.15.0, but you have numpy 1.23.5 which is incompatible.
pymc3 3.11.5 requires scipy<1.8.0,>=1.7.3, but you have scipy 1.11.1 which is incompatible.
```

Preparing some new directories

```
In [13]: # create a separate folder structure for train, validation and test image and validation files
os.chdir('/working')
"""
The os.chdir() function is used to change the working directory
of the Python script to the specified path.
"""

os.mkdir('./yolov5/data/train')
os.mkdir('./yolov5/data/val')
os.mkdir('./yolov5/data/test')
os.mkdir('./yolov5/data/train/images')
os.mkdir('./yolov5/data/train/labels')
os.mkdir('./yolov5/data/val/images')
os.mkdir('./yolov5/data/val/labels')
os.mkdir('./yolov5/data/test/images')
os.mkdir('./yolov5/data/test/labels')
```

Copying the image data in the yolov5 folder

```
In [14]: def open_image_file(image_items, folder_name):

    for image in image_items:
        img = PIL.Image.open(f'{img_dir}/{image}')
        img1 = img.resize((640, 480))
        _ = img1.save(f'{output_dir}/yolov5/data/{folder_name}/images/{image}')

open_image_file(train, 'train')
open_image_file(val, 'val')
open_image_file(test, 'test')
```

### Resizing boxes to match with the new images size

```
In [15]: df1['xmin'] = (640/df1['width']) * df1['xmin']
df1['ymin'] = (480/df1['height']) * df1['ymin']
df1['xmax'] = (640/df1['width']) * df1['xmax']
df1['ymax'] = (480/df1['height']) * df1['ymax']
df1.head()
```

```
Out[15]:
```

	name	label	width	height	xmin	ymin	xmax	ymax	class
0	makssssksksss737	with_mask	400	226	44.8	116.814159	73.6	150.796460	1
1	makssssksksss737	with_mask	400	226	156.8	131.681416	177.6	165.663717	1
2	makssssksksss737	mask_weared_incorrect	400	226	254.4	106.194690	308.8	191.150442	2
3	makssssksksss737	with_mask	400	226	468.8	125.309735	500.8	169.911504	1
4	makssssksksss737	with_mask	400	226	563.2	108.318584	595.2	152.920354	1

```
In [16]: df1[['xmin', 'ymin', 'xmax', 'ymax']] = df1[['xmin', 'ymin', 'xmax', 'ymax']].astype('int')
```

```
In [17]: WIDTH = 640
HEIGHT = 480
```

### Converting from left upper corner and right bottom corner of boxes to just midpoint, height, and width of boxes

```
In [18]: df1['x_center'] = (df1['xmin']+df1['xmax'])/(2*WIDTH)
df1['y_center'] = (df1['ymin']+df1['ymax'])/(2*HEIGHT)
df1['box_width'] = (df1['xmax']-df1['xmin'])/ WIDTH
df1['box_height'] = (df1['ymax']-df1['ymin'])/ HEIGHT
```

```
In [19]: df1.head()
```

Out[19]:

	name	label	width	height	xmin	ymin	xmax	ymax	class	x_center	y_center	box_width	box_height
0	makssksksss737	with_mask	400	226	44	116	73	150	1	0.091406	0.277083	0.045312	0.070833
1	makssksksss737	with_mask	400	226	156	131	177	165	1	0.260156	0.308333	0.032813	0.070833
2	makssksksss737	mask_weared_incorrect	400	226	254	106	308	191	2	0.439063	0.309375	0.084375	0.177083
3	makssksksss737	with_mask	400	226	468	125	500	169	1	0.756250	0.306250	0.050000	0.091667
4	makssksksss737	with_mask	400	226	563	108	595	152	1	0.904687	0.270833	0.050000	0.091667

In [20]: `df1 = df1.astype(str)`

Viewing some images paths

In [21]: `img_file_path[:5]`

Out[21]:

```
['makssksksss810.png',
 'makssksksss848.png',
 'makssksksss145.png',
 'makssksksss825.png',
 'makssksksss30.png']
```

Writing info for each images

In [22]:

```
def copy_label(label_items, folder_name):
    file_name = [x.split('.')[0] for x in img_file_path]
    for name in file_name:
        data = df1[df1.name == name]
        box_list = []
        for idx in range(len(data)):
            row = data.iloc[idx]
            box_list.append(row['class']+ " "+row['x_center']+ " "+row['y_center']+ " "+row['box_width']+ " "+row['box_height'])

        text = "\n".join(box_list)
        with open(f'{output_dir}/yolov5/data/{folder_name}/labels/{name}.txt', 'w') as file:
            file.write(text)
```

In [23]: `copy_label(train, 'train')`  
`copy_label(val, 'val')`  
`copy_label(test, 'test')`

```
In [24]: os.chdir('/working/yolov5/data/train/labels')
```

```
In [25]: cat makssksksss0.txt
```

```
0 0.1828125 0.3364583333333336 0.059375 0.1020833333333333  
1 0.40078125 0.3322916666666665 0.0796875 0.11875  
0 0.66875 0.3145833333333333 0.06875 0.1375
```

### Creating yaml file

```
In [26]: # Configure .yaml file  
yaml_file = """train: /working/yolov5/data/train/images  
val: /working/yolov5/data/val/images  
  
nc: 3  
names: [without_mask, with_mask, mask_weared_incorrect]"""\n  
with open('/working/yolov5/data/data.yaml', 'w') as f:  
    f.write(yaml_file)  
  
%cat /working/yolov5/data/data.yaml
```

### Training

```
In [28]: !python /working/yolov5/train.py --img 640 --epochs 100 --batch 32 --data /working/yolov5/data/data.yaml --weights yolov5s.pt  
# image size
```

wandb: WARNING ! wandb is deprecated and will be removed in a future release. See supported integrations at <https://github.com/ultralytics/yolov5#integrations>.

wandb: (1) Create a W&B account

wandb: (2) Use an existing W&B account

wandb: (3) Don't visualize my results

wandb: Enter your choice: (30 second timeout)

wandb: W&B disabled due to login timeout.

train: weights=yolov5s.pt, cfg=, data=/kaggle/working/yolov5/data/data.yaml, hyp=../../../../data/hyps/hyp.scratch-low.yaml, epochs=100, batch\_size=32, imgsz=640, rect=False, resume=False, nosave=False, noval=False, noautoanchor=False, noplots=False, evolve=None, bucket=, cache=ram, image\_weights=False, device=, multi\_scale=False, single\_cls=False, optimizer=SGD, sync\_bn=False, workers=8, project=../../../../runs/train, name=exp, 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: up to date with <https://github.com/ultralytics/yolov5> ✓

YOLOv5 🚀 v7.0-227-ge4df1ec Python-3.10.12 torch-2.0.0 CUDA:0 (Tesla T4, 15110MiB)

**hyperparameters:** lr0=0.01, lrf=0.01, momentum=0.937, weight\_decay=0.0005, warmup\_epochs=3.0, warmup\_momentum=0.8, warmup\_bias\_lr=0.1, box=0.05, cls=0.5, cls\_pw=1.0, obj=1.0, 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=0.5, shear=0.0, perspective=0.0, flipud=0.0, fliplr=0.5, mosaic=1.0, mixup=0.0, copy\_paste=0.0

**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, 15.9MB/s]

Downloading <https://github.com/ultralytics/yolov5/releases/download/v7.0/yolov5s.pt> to yolov5s.pt...

100%|██████████| 14.1M/14.1M [00:00<00:00, 133MB/s]

Overriding model.yaml nc=80 with nc=3

	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   21576 models.yolo.Detect [3, [[10, 13, 16, 30, 33, 23], [30, 61, 62,
45, 59, 119], [116, 90, 156, 198, 373, 326]], [128, 256, 512]]
Model summary: 214 layers, 7027720 parameters, 7027720 gradients, 16.0 GFLOPs

```

Transferred 343/349 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

WARNING ! DP not recommended, use torch.distributed.run for best DDP Multi-GPU results.

See Multi-GPU Tutorial at [https://docs.ultralytics.com/yolov5/tutorials/multi\\_gpu\\_training](https://docs.ultralytics.com/yolov5/tutorials/multi_gpu_training) to get started.

**albumentations:** 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 /kaggle/working/yolov5/data/train/labels... 579 images, 0 backgr

**train:** New cache created: /kaggle/working/yolov5/data/train/labels.cache

**train:** Caching images (0.1GB ram): 16%|██████████| 91/579 [00:00<00:03, 146.89libpng warning: iCCP: Not recognizing known sRGB profile that has been edited

**train:** Caching images (0.5GB ram): 100%|██████████| 579/579 [00:03<00:00, 146.62

**val:** Scanning /kaggle/working/yolov5/data/val/labels... 103 images, 0 background

**val:** New cache created: /kaggle/working/yolov5/data/val/labels.cache

**val:** Caching images (0.1GB ram): 100%|██████████| 103/103 [00:01<00:00, 89.02it/

**AutoAnchor:** 5.62 anchors/target, 0.999 Best Possible Recall (BPR). Current anchors are a good fit to dataset ✓

Plotting labels to ../../runs/train/exp/labels.jpg...

Image sizes 640 train, 640 val

Using 2 dataloader workers

Logging results to ../../runs/train/exp

Starting training for 100 epochs...

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
0/99	3.73G	0.1119	0.05595	0.03718	20	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.00815	0.102	0.0118 0.00326
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
1/99	4.96G	0.08458	0.06164	0.02396	16	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.693	0.119	0.023 0.00603

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
2/99	4.97G	0.07263	0.04648	0.01974	36	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.76	0.171	0.111 0.0365
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
3/99	4.97G	0.07204	0.04382	0.0185	32	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.81	0.182	0.169 0.0663
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
4/99	4.97G	0.06954	0.03892	0.01967	8	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.86	0.186	0.27 0.101
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
5/99	4.97G	0.06554	0.03726	0.01701	10	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.546	0.428	0.357 0.121
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
6/99	4.97G	0.05824	0.03737	0.01616	18	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.568	0.474	0.378 0.148
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
7/99	4.97G	0.05382	0.03538	0.01471	13	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.776	0.453	0.515 0.222
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
8/99	4.97G	0.04899	0.03492	0.01162	34	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.709	0.417	0.434 0.215
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
9/99	4.97G	0.04794	0.03367	0.009953	13	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.765	0.445	0.502 0.246
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
10/99	4.97G	0.04744	0.0335	0.008764	36	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.845	0.499	0.576 0.288

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
11/99	4.97G	0.04296	0.03307	0.008266	21	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.846	0.497	0.602 0.34
12/99	4.97G	0.0421	0.03169	0.008069	14	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.791	0.522	0.597 0.324
13/99	4.97G	0.04059	0.03019	0.007749	16	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.887	0.532	0.616 0.348
14/99	4.97G	0.03939	0.03112	0.006818	17	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.924	0.483	0.654 0.366
15/99	4.97G	0.03869	0.03047	0.007046	24	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.934	0.514	0.662 0.387
16/99	4.97G	0.03939	0.03409	0.006668	130	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.922	0.529	0.659 0.359
17/99	4.97G	0.03795	0.03088	0.006535	30	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.907	0.549	0.652 0.371
18/99	4.97G	0.03638	0.03041	0.006486	24	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.935	0.544	0.665 0.389
19/99	4.97G	0.0359	0.02952	0.006187	12	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.916	0.543	0.637 0.367

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
20/99	4.97G	0.03548	0.03099	0.006632	25	640: 1
	Class all	Images 103	Instances 428	P 0.925	R 0.559	mAP50 0.665 0.398
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
21/99	4.97G	0.03385	0.02925	0.005943	13	640: 1
	Class all	Images 103	Instances 428	P 0.935	R 0.567	mAP50 0.665 0.406
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
22/99	4.97G	0.03435	0.03276	0.005438	75	640: 1
	Class all	Images 103	Instances 428	P 0.953	R 0.541	mAP50 0.657 0.403
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
23/99	4.97G	0.03249	0.02903	0.005717	27	640: 1
	Class all	Images 103	Instances 428	P 0.905	R 0.561	mAP50 0.68 0.415
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
24/99	4.97G	0.03279	0.0297	0.005596	22	640: 1
	Class all	Images 103	Instances 428	P 0.911	R 0.547	mAP50 0.693 0.416
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
25/99	4.97G	0.03207	0.02836	0.005395	31	640: 1
	Class all	Images 103	Instances 428	P 0.611	R 0.658	mAP50 0.694 0.444
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
26/99	4.97G	0.03299	0.02945	0.005758	29	640: 1
	Class all	Images 103	Instances 428	P 0.795	R 0.658	mAP50 0.71 0.442
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
27/99	4.97G	0.03258	0.02882	0.005455	43	640: 1
	Class all	Images 103	Instances 428	P 0.932	R 0.553	mAP50 0.668 0.416
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
28/99	4.97G	0.03234	0.02895	0.006175	45	640: 1
	Class all	Images 103	Instances 428	P 0.635	R 0.699	mAP50 0.714 0.436

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
29/99	4.97G	0.03088	0.02766	0.004996	25	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.657	0.679	0.709 0.427
30/99	4.97G	0.03074	0.02516	0.004811	14	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.618	0.645	0.695 0.41
31/99	4.97G	0.02989	0.0271	0.004506	13	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.649	0.714	0.709 0.452
32/99	4.97G	0.03061	0.02737	0.00545	26	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.687	0.667	0.692 0.416
33/99	4.97G	0.02968	0.02739	0.005746	12	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.682	0.727	0.743 0.464
34/99	4.97G	0.03001	0.0258	0.004774	7	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.615	0.699	0.687 0.434
35/99	4.97G	0.02964	0.02707	0.004496	37	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.633	0.694	0.733 0.445
36/99	4.97G	0.02852	0.02538	0.00495	8	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.587	0.686	0.735 0.463
37/99	4.97G	0.02844	0.02759	0.004317	36	640: 1
		Class Images	Instances	P	R	mAP50
		all 103	428	0.667	0.717	0.728 0.463

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
38/99	4.97G	0.02881	0.02633	0.004267	19	640: 1
	Class all	Images 103	Instances 428	P 0.612	R 0.753	mAP50 0.732 0.443
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
39/99	4.97G	0.02753	0.02591	0.003903	20	640: 1
	Class all	Images 103	Instances 428	P 0.622	R 0.707	mAP50 0.742 0.457
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
40/99	4.97G	0.02829	0.02821	0.004556	80	640: 1
	Class all	Images 103	Instances 428	P 0.636	R 0.766	mAP50 0.761 0.464
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
41/99	4.97G	0.02776	0.02686	0.004309	43	640: 1
	Class all	Images 103	Instances 428	P 0.632	R 0.739	mAP50 0.75 0.475
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
42/99	4.97G	0.02742	0.02643	0.003938	12	640: 1
	Class all	Images 103	Instances 428	P 0.683	R 0.714	mAP50 0.739 0.478
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
43/99	4.97G	0.02676	0.02522	0.004204	12	640: 1
	Class all	Images 103	Instances 428	P 0.72	R 0.748	mAP50 0.744 0.472
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
44/99	4.97G	0.02749	0.0262	0.003971	17	640: 1
	Class all	Images 103	Instances 428	P 0.786	R 0.682	mAP50 0.721 0.451
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
45/99	4.97G	0.02649	0.02379	0.003736	11	640: 1
	Class all	Images 103	Instances 428	P 0.718	R 0.783	mAP50 0.764 0.485
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
46/99	4.97G	0.0264	0.02477	0.003197	20	640: 1
	Class all	Images 103	Instances 428	P 0.833	R 0.742	mAP50 0.771 0.489

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
47/99	4.97G	0.02657	0.02474	0.003236	6	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.771	0.768	0.801 0.513
48/99	4.97G	0.02632	0.02734	0.00295	39	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.874	0.721	0.819 0.51
49/99	4.97G	0.02595	0.02538	0.003158	34	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.847	0.743	0.806 0.481
50/99	4.97G	0.02583	0.02445	0.003061	24	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.888	0.737	0.814 0.514
51/99	4.97G	0.02727	0.02694	0.00338	121	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.828	0.747	0.79 0.511
52/99	4.97G	0.02525	0.02438	0.002879	38	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.86	0.72	0.782 0.496
53/99	4.97G	0.02599	0.02504	0.003247	21	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.865	0.745	0.807 0.497
54/99	4.97G	0.02507	0.02472	0.002994	24	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.876	0.712	0.783 0.487
55/99	4.97G	0.02537	0.02516	0.002595	32	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.846	0.675	0.764 0.501

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
56/99	4.97G	0.02502	0.02354	0.002543	16	640: 1
	Class all	Images 103	Instances 428	P 0.817	R 0.713	mAP50 0.779 0.508
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
57/99	4.97G	0.02429	0.0246	0.002732	46	640: 1
	Class all	Images 103	Instances 428	P 0.924	R 0.721	mAP50 0.788 0.516
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
58/99	4.97G	0.02407	0.02442	0.002724	16	640: 1
	Class all	Images 103	Instances 428	P 0.862	R 0.752	mAP50 0.816 0.523
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
59/99	4.97G	0.02367	0.0233	0.002455	11	640: 1
	Class all	Images 103	Instances 428	P 0.855	R 0.749	mAP50 0.811 0.522
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
60/99	4.97G	0.025	0.02496	0.002666	18	640: 1
	Class all	Images 103	Instances 428	P 0.824	R 0.764	mAP50 0.811 0.513
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
61/99	4.97G	0.02357	0.02198	0.00253	9	640: 1
	Class all	Images 103	Instances 428	P 0.894	R 0.779	mAP50 0.827 0.527
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
62/99	4.97G	0.02353	0.02405	0.002239	22	640: 1
	Class all	Images 103	Instances 428	P 0.901	R 0.76	mAP50 0.821 0.518
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
63/99	4.97G	0.02394	0.02369	0.00273	13	640: 1
	Class all	Images 103	Instances 428	P 0.913	R 0.731	mAP50 0.816 0.53
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
64/99	4.97G	0.02302	0.0233	0.002911	39	640: 1
	Class all	Images 103	Instances 428	P 0.879	R 0.733	mAP50 0.793 0.511

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
65/99	4.97G	0.02404	0.02393	0.002654	15	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.906	0.758	0.823 0.535
66/99	4.97G	0.02224	0.02182	0.001667	30	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.929	0.744	0.824 0.536
67/99	4.97G	0.02336	0.02393	0.002246	10	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.886	0.715	0.805 0.514
68/99	4.97G	0.02234	0.02204	0.002036	13	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.939	0.731	0.829 0.523
69/99	4.97G	0.02253	0.02234	0.00206	17	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.888	0.762	0.816 0.533
70/99	4.97G	0.02243	0.02248	0.001936	14	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.828	0.755	0.814 0.538
71/99	4.97G	0.02181	0.02065	0.001955	9	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.958	0.719	0.811 0.536
72/99	4.97G	0.02182	0.02321	0.001736	33	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.898	0.753	0.827 0.53
73/99	4.97G	0.02111	0.02321	0.001585	22	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.851	0.77	0.824 0.526

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
74/99	4.97G	0.02121	0.0217	0.001872	6	640: 1	
	Class all	Images 103	Instances 428	P 0.955	R 0.777	mAP50 0.867	0.56
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
75/99	4.97G	0.02121	0.02199	0.001948	10	640: 1	
	Class all	Images 103	Instances 428	P 0.926	R 0.789	mAP50 0.86	0.563
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
76/99	4.97G	0.02142	0.02231	0.001549	13	640: 1	
	Class all	Images 103	Instances 428	P 0.917	R 0.734	mAP50 0.853	0.563
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
77/99	4.97G	0.02097	0.02327	0.001944	25	640: 1	
	Class all	Images 103	Instances 428	P 0.882	R 0.731	mAP50 0.834	0.551
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
78/99	4.97G	0.02115	0.02212	0.001642	53	640: 1	
	Class all	Images 103	Instances 428	P 0.934	R 0.755	mAP50 0.857	0.567
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
79/99	4.97G	0.02096	0.02288	0.001463	31	640: 1	
	Class all	Images 103	Instances 428	P 0.941	R 0.774	mAP50 0.838	0.565
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
80/99	4.97G	0.02076	0.02116	0.001664	25	640: 1	
	Class all	Images 103	Instances 428	P 0.932	R 0.772	mAP50 0.848	0.561
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
81/99	4.97G	0.02043	0.02077	0.001639	35	640: 1	
	Class all	Images 103	Instances 428	P 0.947	R 0.764	mAP50 0.849	0.565
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size	
82/99	4.97G	0.02177	0.02239	0.001735	30	640: 1	
	Class all	Images 103	Instances 428	P 0.88	R 0.785	mAP50 0.849	0.56

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
83/99	4.97G	0.0204	0.02266	0.001486	49	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.919	0.776	0.842 0.549
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
84/99	4.97G	0.01979	0.02143	0.00151	20	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.939	0.767	0.853 0.562
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
85/99	4.97G	0.021	0.02715	0.001409	42	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.919	0.771	0.832 0.543
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
86/99	4.97G	0.01902	0.0185	0.001263	5	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.92	0.764	0.846 0.564
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
87/99	4.97G	0.02028	0.02192	0.001555	22	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.922	0.742	0.835 0.554
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
88/99	4.97G	0.01964	0.02206	0.001241	29	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.916	0.773	0.835 0.553
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
89/99	4.97G	0.01926	0.01968	0.001275	27	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.884	0.767	0.832 0.553
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
90/99	4.97G	0.01926	0.02005	0.001307	18	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.859	0.773	0.832 0.557
Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
91/99	4.97G	0.0192	0.01953	0.001486	34	640: 1
	Class all	Images	Instances	P	R	mAP50
		103	428	0.897	0.78	0.835 0.556

Epoch	GPU_mem	box_loss	obj_loss	cls_loss	Instances	Size
92/99	4.97G	0.01954	0.0214	0.001271	41	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.901	0.773	0.831 0.551
93/99	4.97G	0.01945	0.02096	0.001255	31	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.93	0.773	0.839 0.559
94/99	4.97G	0.01928	0.02207	0.001265	34	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.909	0.778	0.827 0.563
95/99	4.97G	0.01995	0.02014	0.001251	22	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.928	0.744	0.832 0.564
96/99	4.97G	0.01901	0.02078	0.001113	10	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.92	0.768	0.842 0.568
97/99	4.97G	0.01961	0.02399	0.001192	78	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.924	0.772	0.847 0.58
98/99	4.97G	0.01932	0.02092	0.001142	20	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.922	0.753	0.842 0.569
99/99	4.97G	0.01843	0.02064	0.001399	33	640: 1
	Class	Images	Instances	P	R	mAP50
	all	103	428	0.928	0.752	0.836 0.564

100 epochs completed in 0.215 hours.

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

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

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

Fusing layers...

Model summary: 157 layers, 7018216 parameters, 0 gradients, 15.8 GFLOPs

Class	Images	Instances	P	R	mAP50	
all	103	428	0.923	0.772	0.847	0.581
without_mask	103	75	0.936	0.787	0.847	0.498
with_mask	103	335	0.954	0.92	0.959	0.703
mask_weared_incorrect	103	18	0.88	0.611	0.734	0.541

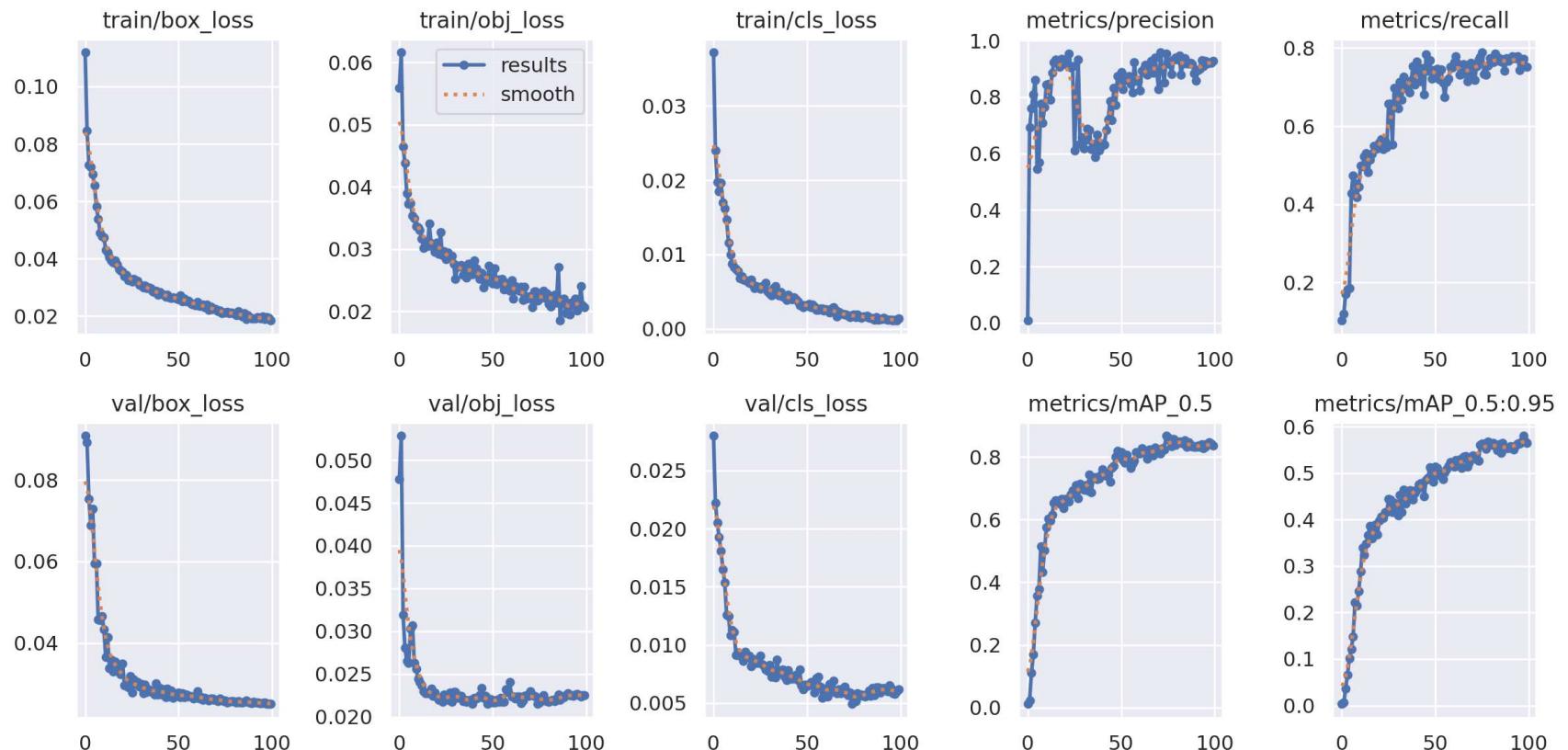
Results saved to ../../runs/train/exp

Displaying performance of model

In [29]: `from IPython.display import Image as Display`

In [30]: `Display(filename='/working/yolov5/runs/train/exp/results.png', width=1080)`

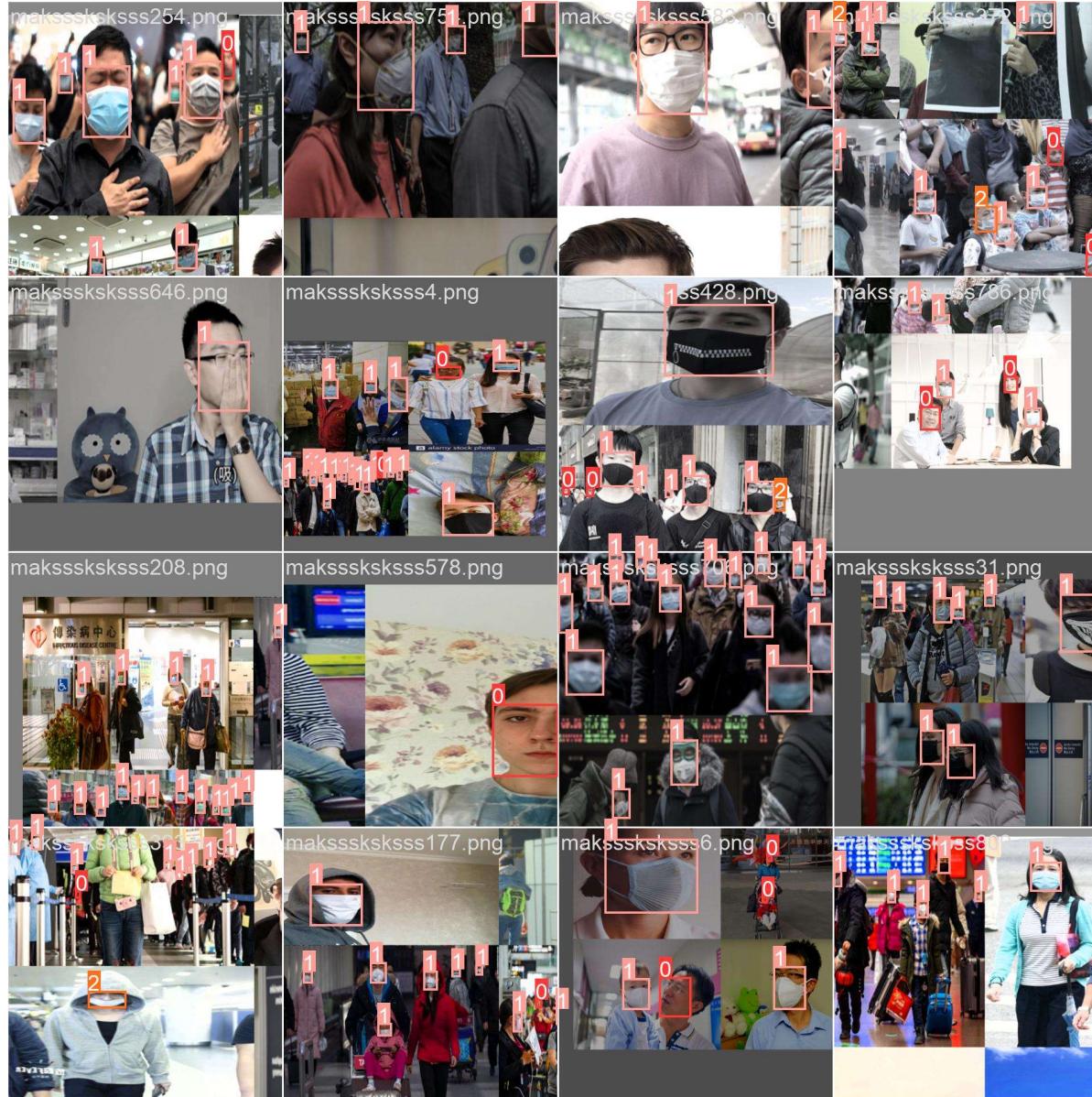
Out[30]:



Displaying Training and Validation batches

```
In [31]: Display(filename='/working/yolov5/runs/train/exp/train_batch0.jpg', width=600)
```

Out[31]:



```
In [32]: from PIL import Image, ImageOps  
Image.open("/working/yolov5/runs/train/exp/val_batch1_pred.jpg")
```

Out[32]:

