

## Notebook for processing and combining 4 datasets in order to make a balanced dataset for training models.

```
import os, random, pickle, json
import cv2
import numpy as np
import torch
import matplotlib.pyplot as plt
```

```
from config import Config
from process_images import ImageUtils
```

```
%matplotlib inline
%config IPCompleter.greedy=True
%config Completer.use_jedi = False
```

```
# init config holding constants
config = Config()
config.im_size = 128
```

```
#init image utils
im_utils = ImageUtils(config)
```

### Load WWMR

```
from process_images import WWMRImageProcessor
```

```
wwmr_image_proc = WWMRImageProcessor(config)
```

```
# process and store tensors for WWMR
wwmr_image_proc.process_images()
```

Loading image data...

```
100 {'path': '../data/real/WWMR-DB - Part 1\\Subject 12\\Mask Or
Respirator Correctly Worn\\Non-Medical Mask/0012_MRCW_NMDM_0045.jpg',
'size': (4224, 3136, 3), 'mask_status': 'MRCW'}
```

```
200 {'path': '../data/real/WWMR-DB - Part 1\\Subject 16\\Mask Or
Respirator Not Worn/0016_MRNW_0090.jpg', 'size': (3264, 2448, 3),
'mask_status': 'MRNW'}
```

```
300 {'path': '../data/real/WWMR-DB - Part 1\\Subject 4\\Mask Or
Respirator On The Forehead\\Non-Medical Mask/0004_MRFH_NMDM_0000.jpg',
'size': (5184, 3880, 3), 'mask_status': 'MRFH'}
```

```
400 {'path': '../data/real/WWMR-DB - Part 1\\Subject 7\\Mask Or
Respirator Not Worn/0007_MRNW_0000.jpg', 'size': (1280, 960, 3),
'mask_status': 'MRNW'}
```

446 files processed

100 {'path': '../data/real/WWMR-DB - Part 2\\Subject 20\\Mask Or Respirator On The Tip Of The Nose\\Surgical Mask/0020\_MRTN\_SRG\_M\_0045.jpg', 'size': (3264, 2448, 3), 'mask\_status': 'MRTN'}

200 {'path': '../data/real/WWMR-DB - Part 2\\Subject 27\\Mask Or Respirator Correctly Worn\\Disposable Respirator Without Valve/0027\_MRCW\_DRNV\_0045.jpg', 'size': (3264, 2448, 3), 'mask\_status': 'MRCW'}

300 {'path': '../data/real/WWMR-DB - Part 2\\Subject 32\\Mask Or Respirator On The Forehead\\Non-Medical Mask/0032\_MRFH\_NMDM\_0090.jpg', 'size': (4608, 3456, 3), 'mask\_status': 'MRFH'}

400 {'path': '../data/real/WWMR-DB - Part 2\\Subject 36\\Mask Or Respirator On The Forehead\\Surgical Mask/0036\_MRFH\_SRG\_M\_0000.jpeg', 'size': (3088, 2320, 3), 'mask\_status': 'MRFH'}

500 {'path': '../data/real/WWMR-DB - Part 2\\Subject 41\\Mask Or Respirator Correctly Worn\\Surgical Mask/0041\_MRCW\_SRG\_M\_0000.jpg', 'size': (640, 480, 3), 'mask\_status': 'MRCW'}

535 files processed

Getting bounding box info...

0001\_MRHN\_DRNV\_0000 not in im sizes  
0001\_MRHN\_DRNV\_0045\_B not in im sizes  
0001\_MRHN\_DRNV\_0045\_F not in im sizes  
0001\_MRHN\_DRNV\_0090\_B not in im sizes  
0001\_MRHN\_DRNV\_0090\_F not in im sizes  
0001\_MRHN\_DRWV\_0000 not in im sizes  
0001\_MRHN\_DRWV\_0045\_B not in im sizes  
0001\_MRHN\_DRWV\_0045\_F not in im sizes  
0001\_MRHN\_DRWV\_0090\_B not in im sizes  
0001\_MRHN\_DRWV\_0090\_F not in im sizes  
0001\_MRHN\_NMDM\_0000 not in im sizes  
0001\_MRHN\_NMDM\_0045\_B not in im sizes  
0001\_MRHN\_NMDM\_0045\_F not in im sizes  
0001\_MRHN\_NMDM\_0090\_B not in im sizes  
0001\_MRHN\_NMDM\_0090\_F not in im sizes  
0001\_MRHN\_SRG\_M\_0000 not in im sizes  
0001\_MRHN\_SRG\_M\_0045\_B not in im sizes  
0001\_MRHN\_SRG\_M\_0045\_F not in im sizes  
0001\_MRHN\_SRG\_M\_0090\_B not in im sizes  
0001\_MRHN\_SRG\_M\_0090\_F not in im sizes  
0010\_MRHN\_SRG\_M\_0000 not in im sizes  
0010\_MRHN\_SRG\_M\_0045\_B not in im sizes  
0010\_MRHN\_SRG\_M\_0045\_F not in im sizes  
0010\_MRHN\_SRG\_M\_0090\_B not in im sizes  
0010\_MRHN\_SRG\_M\_0090\_F not in im sizes  
0011\_MRHN\_DRNV\_0000 not in im sizes  
0012\_MRHN\_NMDM\_0000 not in im sizes

0012\_MRHN\_NMDM\_0045\_B not in im sizes  
0012\_MRHN\_NMDM\_0045\_F not in im sizes  
0013\_MRHN\_SRGM\_0000 not in im sizes  
0013\_MRHN\_SRGM\_0045\_B not in im sizes  
0013\_MRHN\_SRGM\_0045\_F not in im sizes  
0013\_MRHN\_SRGM\_0090\_B not in im sizes  
0013\_MRHN\_SRGM\_0090\_F not in im sizes  
0014\_MRHN\_SRGM\_0000 not in im sizes  
0014\_MRHN\_SRGM\_0045\_B not in im sizes  
0014\_MRHN\_SRGM\_0045\_F not in im sizes  
0014\_MRHN\_SRGM\_0090\_B not in im sizes  
0014\_MRHN\_SRGM\_0090\_F not in im sizes  
0015\_MRHN\_DRNV\_0000 not in im sizes  
0015\_MRHN\_DRNV\_0045\_B not in im sizes  
0015\_MRHN\_DRNV\_0045\_F not in im sizes  
0015\_MRHN\_DRNV\_0090\_B not in im sizes  
0015\_MRHN\_DRNV\_0090\_F not in im sizes  
0015\_MRHN\_SRGM\_0000 not in im sizes  
0015\_MRHN\_SRGM\_0045\_B not in im sizes  
0015\_MRHN\_SRGM\_0045\_F not in im sizes  
0015\_MRHN\_SRGM\_0090\_B not in im sizes  
0015\_MRHN\_SRGM\_0090\_F not in im sizes  
0016\_MRHN\_NMDM\_0000 not in im sizes  
0016\_MRHN\_NMDM\_0045\_F not in im sizes  
0016\_MRHN\_NMDM\_0090\_B not in im sizes  
0016\_MRHN\_NMDM\_0090\_F not in im sizes  
0017\_MRHN\_DRNV\_0000 not in im sizes  
0017\_MRHN\_DRNV\_0045\_B not in im sizes  
0017\_MRHN\_DRNV\_0045\_F not in im sizes  
0017\_MRHN\_DRNV\_0090\_B not in im sizes  
0017\_MRHN\_DRNV\_0090\_F not in im sizes  
0017\_MRHN\_DRWV\_0000 not in im sizes  
0017\_MRHN\_DRWV\_0045\_B not in im sizes  
0017\_MRHN\_DRWV\_0045\_F not in im sizes  
0017\_MRHN\_DRWV\_0090\_B not in im sizes  
0017\_MRHN\_DRWV\_0090\_F not in im sizes  
0017\_MRHN\_NMDM\_0000 not in im sizes  
0017\_MRHN\_SRGM\_0000 not in im sizes  
0017\_MRHN\_SRGM\_0045\_B not in im sizes  
0017\_MRHN\_SRGM\_0045\_F not in im sizes  
0017\_MRHN\_SRGM\_0090\_B not in im sizes  
0017\_MRHN\_SRGM\_0090\_F not in im sizes  
0018\_MRHN\_NMDM\_0000 not in im sizes  
0019\_MRHN\_SRGM\_0000 not in im sizes  
0019\_MRHN\_SRGM\_0045\_B not in im sizes  
0019\_MRHN\_SRGM\_0045\_F not in im sizes  
0019\_MRHN\_SRGM\_0090\_B not in im sizes  
0019\_MRHN\_SRGM\_0090\_F not in im sizes  
0002\_MRHN\_NMDM\_0000 not in im sizes  
0002\_MRHN\_NMDM\_0045\_B not in im sizes

0002\_MRHN\_NMDM\_0045\_F not in im sizes  
0002\_MRHN\_NMDM\_0090\_B not in im sizes  
0002\_MRHN\_NMDM\_0090\_F not in im sizes  
0002\_MRHN\_SRGM\_0000 not in im sizes  
0002\_MRHN\_SRGM\_0045\_B not in im sizes  
0002\_MRHN\_SRGM\_0045\_F not in im sizes  
0002\_MRHN\_SRGM\_0090\_B not in im sizes  
0002\_MRHN\_SRGM\_0090\_F not in im sizes  
0020\_MRHN\_SRGM\_0000 not in im sizes  
0020\_MRHN\_SRGM\_0045\_B not in im sizes  
0020\_MRHN\_SRGM\_0045\_F not in im sizes  
0020\_MRHN\_SRGM\_0090\_B not in im sizes  
0020\_MRHN\_SRGM\_0090\_F not in im sizes  
0021\_MRHN\_SRGM\_0000 not in im sizes  
0022\_MRHN\_SRGM\_0000 not in im sizes  
0023\_MRHN\_NMDM\_0000 not in im sizes  
0023\_MRHN\_SRGM\_0045\_B not in im sizes  
0023\_MRHN\_SRGM\_0045\_F not in im sizes  
0024\_MRHN\_NMDM\_0000 not in im sizes  
0024\_MRHN\_NMDM\_0045\_B not in im sizes  
0024\_MRHN\_NMDM\_0045\_F not in im sizes  
0024\_MRHN\_NMDM\_0090\_B not in im sizes  
0024\_MRHN\_NMDM\_0090\_F not in im sizes  
0025\_MRHN\_SRGM\_0000 not in im sizes  
0025\_MRHN\_SRGM\_0045\_B not in im sizes  
0025\_MRHN\_SRGM\_0045\_F not in im sizes  
0025\_MRHN\_SRGM\_0090\_B not in im sizes  
0025\_MRHN\_SRGM\_0090\_F not in im sizes  
0026\_MRHN\_SRGM\_0000 not in im sizes  
0026\_MRHN\_SRGM\_0045 not in im sizes  
0026\_MRHN\_SRGM\_0090 not in im sizes  
0027\_MRHN\_DRNV\_0000 not in im sizes  
0027\_MRHN\_DRNV\_0045\_B not in im sizes  
0027\_MRHN\_DRNV\_0045\_F not in im sizes  
0027\_MRHN\_DRNV\_0090\_B not in im sizes  
0027\_MRHN\_DRNV\_0090\_F not in im sizes  
0028\_MRHN\_SRGM\_0000 not in im sizes  
0029\_MRHN\_SRGM\_0000 not in im sizes  
0003\_MRHN\_SRGM\_0000 not in im sizes  
0003\_MRHN\_SRGM\_0045\_B not in im sizes  
0003\_MRHN\_SRGM\_0045\_F not in im sizes  
0003\_MRHN\_SRGM\_0090\_B not in im sizes  
0003\_MRHN\_SRGM\_0090\_F not in im sizes  
0030\_MRHN\_DRNV\_0000 not in im sizes  
0030\_MRHN\_DRNV\_0045\_B not in im sizes  
0030\_MRHN\_DRNV\_0045\_F not in im sizes  
0030\_MRHN\_DRNV\_0090\_B not in im sizes  
0030\_MRHN\_DRNV\_0090\_F not in im sizes  
0030\_MRHN\_DRWV\_0000 not in im sizes  
0030\_MRHN\_DRWV\_0045\_B not in im sizes

0030\_MRHN\_DRWV\_0045\_F not in im sizes  
0030\_MRHN\_DRWV\_0090\_B not in im sizes  
0030\_MRHN\_DRWV\_0090\_F not in im sizes  
0030\_MRHN\_SRGM\_0000 not in im sizes  
0030\_MRHN\_SRGM\_0045\_B not in im sizes  
0030\_MRHN\_SRGM\_0045\_F not in im sizes  
0030\_MRHN\_SRGM\_0090\_B not in im sizes  
0030\_MRHN\_SRGM\_0090\_F not in im sizes  
0031\_MRHN\_SRGM\_0000 not in im sizes  
0032\_MRHN\_NMDM\_0000 not in im sizes  
0032\_MRHN\_NMDM\_0045\_B not in im sizes  
0032\_MRHN\_NMDM\_0045\_F not in im sizes  
0032\_MRHN\_NMDM\_0090\_B not in im sizes  
0032\_MRHN\_NMDM\_0090\_F not in im sizes  
0033\_MRHN\_SRGM\_0000 not in im sizes  
0033\_MRHN\_SRGM\_0045\_B not in im sizes  
0033\_MRHN\_SRGM\_0045\_F not in im sizes  
0033\_MRHN\_SRGM\_0090\_B not in im sizes  
0033\_MRHN\_SRGM\_0090\_F not in im sizes  
0034\_MRHN\_DRNV\_0000 not in im sizes  
0034\_MRHN\_DRNV\_0045\_B not in im sizes  
0034\_MRHN\_DRNV\_0045\_F not in im sizes  
0034\_MRHN\_DRNV\_0090\_B not in im sizes  
0034\_MRHN\_DRNV\_0090\_F not in im sizes  
0034\_MRHN\_SRGM\_0000 not in im sizes  
0034\_MRHN\_SRGM\_0045\_B not in im sizes  
0034\_MRHN\_SRGM\_0045\_F not in im sizes  
0034\_MRHN\_SRGM\_0090\_B not in im sizes  
0034\_MRHN\_SRGM\_0090\_F not in im sizes  
0035\_MRHN\_NMDM\_0000 not in im sizes  
0035\_MRHN\_NMDM\_0045\_B not in im sizes  
0035\_MRHN\_NMDM\_0045\_F not in im sizes  
0035\_MRHN\_NMDM\_0090\_F not in im sizes  
0036\_MRHN\_SRGM\_0000 not in im sizes  
0036\_MRHN\_SRGM\_0045\_B not in im sizes  
0036\_MRHN\_SRGM\_0045\_F not in im sizes  
0036\_MRHN\_SRGM\_0090\_B not in im sizes  
0036\_MRHN\_SRGM\_0090\_F not in im sizes  
0037\_MRHN\_NMDM\_0000 not in im sizes  
0038\_MRHN\_SRGM\_0000 not in im sizes  
0038\_MRHN\_SRGM\_0045\_B not in im sizes  
0038\_MRHN\_SRGM\_0045\_F not in im sizes  
0038\_MRHN\_SRGM\_0090\_B not in im sizes  
0038\_MRHN\_SRGM\_0090\_F not in im sizes  
0039\_MRHN\_SRGM\_0000 not in im sizes  
0039\_MRHN\_SRGM\_0045\_B not in im sizes  
0039\_MRHN\_SRGM\_0045\_F not in im sizes  
0039\_MRHN\_SRGM\_0090\_B not in im sizes  
0039\_MRHN\_SRGM\_0090\_F not in im sizes  
0004\_MRHN\_DRNV\_0000 not in im sizes

0004\_MRHN\_DRNV\_0045\_B not in im sizes  
0004\_MRHN\_DRNV\_0045\_F not in im sizes  
0004\_MRHN\_DRNV\_0090\_B not in im sizes  
0004\_MRHN\_DRNV\_0090\_F not in im sizes  
0004\_MRHN\_DRWV\_0000 not in im sizes  
0004\_MRHN\_DRWV\_0045\_B not in im sizes  
0004\_MRHN\_DRWV\_0045\_F not in im sizes  
0004\_MRHN\_DRWV\_0090\_B not in im sizes  
0004\_MRHN\_DRWV\_0090\_F not in im sizes  
0004\_MRHN\_NMDM\_0000 not in im sizes  
0004\_MRHN\_NMDM\_0045\_B not in im sizes  
0004\_MRHN\_NMDM\_0045\_F not in im sizes  
0004\_MRHN\_NMDM\_0090\_B not in im sizes  
0004\_MRHN\_NMDM\_0090\_F not in im sizes  
0004\_MRHN\_SRGM\_0000 not in im sizes  
0004\_MRHN\_SRGM\_0045\_B not in im sizes  
0004\_MRHN\_SRGM\_0045\_F not in im sizes  
0004\_MRHN\_SRGM\_0090\_B not in im sizes  
0004\_MRHN\_SRGM\_0090\_F not in im sizes  
0040\_MRHN\_DRWV\_0000 not in im sizes  
0040\_MRHN\_DRWV\_0045\_B not in im sizes  
0040\_MRHN\_DRWV\_0045\_F not in im sizes  
0040\_MRHN\_DRWV\_0090\_B not in im sizes  
0040\_MRHN\_DRWV\_0090\_F not in im sizes  
0040\_MRHN\_NMDM\_0000 not in im sizes  
0040\_MRHN\_NMDM\_0045\_B not in im sizes  
0040\_MRHN\_NMDM\_0045\_F not in im sizes  
0040\_MRHN\_NMDM\_0090\_B not in im sizes  
0040\_MRHN\_NMDM\_0090\_F not in im sizes  
0041\_MRHN\_SRGM\_0000 not in im sizes  
0041\_MRHN\_SRGM\_0045\_B not in im sizes  
0041\_MRHN\_SRGM\_0045\_F not in im sizes  
0041\_MRHN\_SRGM\_0090\_B not in im sizes  
0041\_MRHN\_SRGM\_0090\_F not in im sizes  
0042\_MRHN\_DRNV\_0000 not in im sizes  
0042\_MRHN\_DRNV\_0045\_B not in im sizes  
0042\_MRHN\_DRNV\_0045\_F not in im sizes  
0042\_MRHN\_DRNV\_0090\_F not in im sizes  
0005\_MRHN\_NMDM\_0000 not in im sizes  
0005\_MRHN\_NMDM\_0045\_B not in im sizes  
0005\_MRHN\_NMDM\_0045\_F not in im sizes  
0005\_MRHN\_NMDM\_0090\_B not in im sizes  
0005\_MRHN\_NMDM\_0090\_F not in im sizes  
0006\_MRHN\_DRNV\_0000 not in im sizes  
0006\_MRHN\_DRNV\_0045\_B not in im sizes  
0006\_MRHN\_DRNV\_0045\_F not in im sizes  
0006\_MRHN\_DRNV\_0090\_B not in im sizes  
0006\_MRHN\_DRNV\_0090\_F not in im sizes  
0006\_MRHN\_SRGM\_0000 not in im sizes  
0006\_MRHN\_SRGM\_0045\_B not in im sizes

```
0006_MRHN_SRGM_0045_F not in im sizes
0006_MRHN_SRGM_0090_B not in im sizes
0006_MRHN_SRGM_0090_F not in im sizes
0007_MRHN_SRGM_0000 not in im sizes
0008_MRHN_NMDM_0000 not in im sizes
0008_MRHN_NMDM_0045_B not in im sizes
0008_MRHN_NMDM_0045_F not in im sizes
0008_MRHN_NMDM_0090_B not in im sizes
0008_MRHN_NMDM_0090_F not in im sizes
0009_MRHN_SRGM_0000 not in im sizes
0009_MRHN_SRGM_0045_B not in im sizes
0009_MRHN_SRGM_0045_F not in im sizes
0009_MRHN_SRGM_0090_B not in im sizes
0009_MRHN_SRGM_0090_F not in im sizes
Writing image data to ../data/real/im_data.json
Cropping and resizing and storing tensors...
```

```
100
200
300
400
500
600
700
800
900
torch.Size([981, 3, 128, 128])
saving ../data/real/x_128.pt
Done!
```

```
#load the processed data
```

```
wwmr_x, wwmr_y, wwmr_im_names = wwmr_image_proc.load_data()
```

```
wwmr_x.shape
```

```
torch.Size([981, 3, 128, 128])
```

```
#Get class counts: 0=no mask, 1=mask correct, 2=mask incorrect
print(np.unique(wwmr_y, return_counts=True))
```

```
(array([0, 1, 2], dtype=int64), array([113, 152, 716], dtype=int64))
```

```
# augment class 2 since there are additional instances of class 0 and 1 in the other datasets
```

```
wwmr_x, wwmr_y, wwmr_im_names = im_utils.flip_augment_class(wwmr_x,
wwmr_y, wwmr_im_names, class_idx_to_flip=2, num=280)
```

```
print(np.unique(wwmr_y, return_counts=True))
```

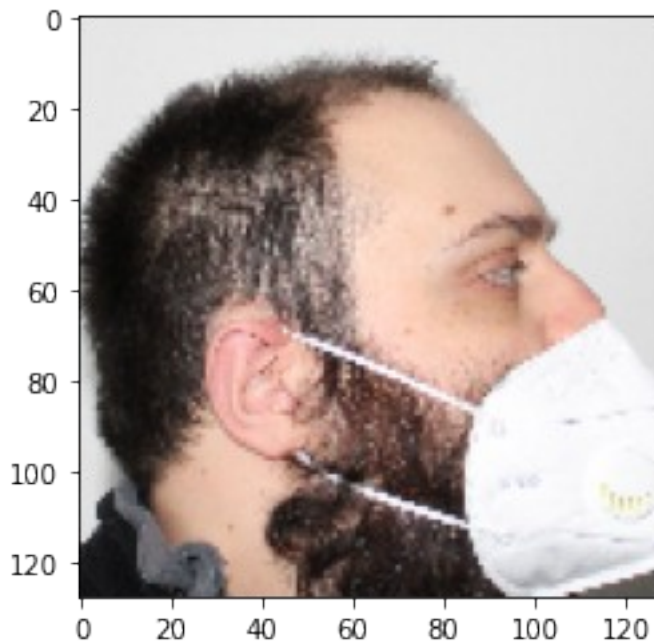
```
Flipping 280 instances of class index 2
New dataset size: 1261
Shuffling...
(array([0., 1., 2.], dtype=float32), array([113, 152, 996],
dtype=int64))
```

```
#save the augmented data
```

```
torch.save(wwmr_x, '%s/x_%d.pt' % (config.wwmr_path, config.im_size))
torch.save(wwmr_y, '%s/y.pt' % (config.wwmr_path))
with open('%s/im_names.txt' % config.wwmr_path, 'w') as o:
    o.write('\n'.join(wwmr_im_names))
```

```
#show random image
```

```
im_index = random.randint(0, len(wwmr_im_names)-1)
im_utils.show_image(wwmr_x[im_index])
```



## Load FMD data

```
from process_images import FMDImageProcessor
```

```
fmd_image_proc = FMDImageProcessor(config)
```

```
# process and store tensors for FMD
```

```
fmd_image_proc.process_images()
```

```
50 25
100 50
150 81
```



```

200 110
250 135
300 167
350 194
400 218
450 244
500 272
550 301
600 329
650 359
700 392
750 408
800 432
850 459
torch.Size([462, 3, 128, 128])
saving ../data/face_mask_detection/x_128.pt

#load the processed data

fmd_x, fmd_y, fmd_im_names = fmd_image_proc.load_data()
fmd_x.shape

torch.Size([462, 3, 128, 128])

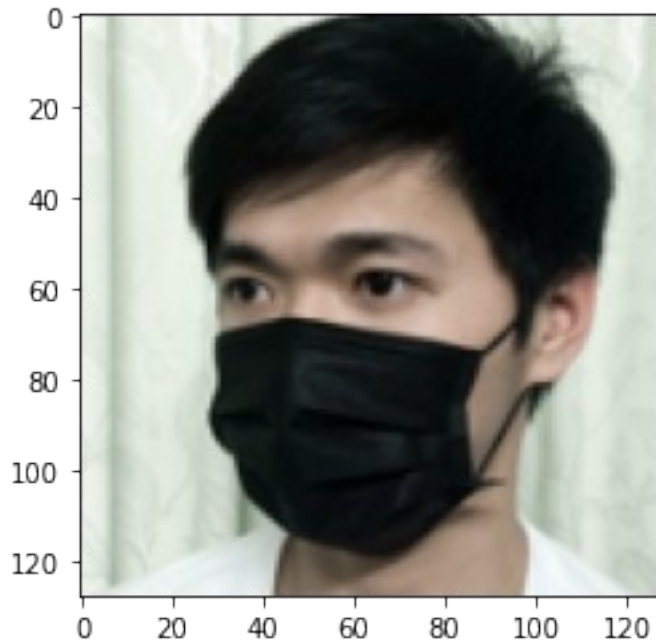
#class counts are now 113+64=177, 152+379=531, 996+19=1015
print(np.unique(fmd_y, return_counts=True))

(array([0, 1, 2], dtype=int64), array([ 64, 379, 19], dtype=int64))

#show random image

im_index = random.randint(0, len(fmd_im_names)-1)
image_proc.show_image(fmd_x[im_index])

```



### Load MFN data

```
from process_images import MFNImageProcessor

mfn_image_proc = MFNImageProcessor(config)

# these are only 450 instances of masks worn correctly
mfn_image_proc.process_images(num_imgs=450)

saving ../data/masked_face_net/x_128.pt

#load the processed data

mfn_x, mfn_y, mfn_im_names = mfn_image_proc.load_data()
mfn_x.shape

torch.Size([450, 3, 128, 128])

#class counts are now 177, 531+450=981, 1015
print(np.unique(mfn_y, return_counts=True))

(array([1], dtype=int64), array([450], dtype=int64))

#show random image

im_index = random.randint(0, len(mfn_im_names)-1)
image_proc.show_image(mfn_x[im_index])
```



### Load GAN faces data

```
from process_images import GANImageProcessor

gan_image_proc = GANImageProcessor(config)

# process and store tensors for GAN

gan_image_proc.process_images(num_imgs=800)

saving ../data/gan_faces/x_128.pt

#load the processed data

gan_x, gan_y, gan_im_names = gan_image_proc.load_data()
gan_x.shape

torch.Size([800, 3, 128, 128])

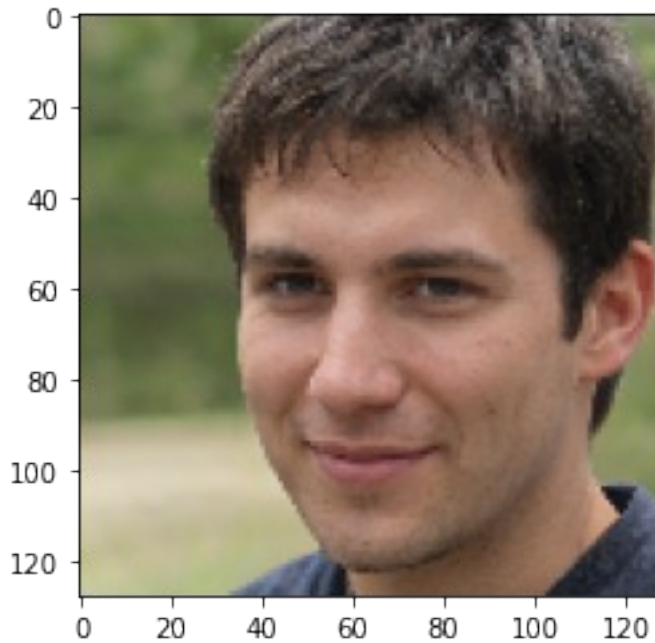
#class counts are now roughly 1000 each: 177+800=977, 531+450=981, 1015

print(np.unique(gan_y, return_counts=True))

(array([0], dtype=int64), array([800], dtype=int64))

#show random image

im_index = random.randint(0, len(gan_im_names)-1)
image_proc.show_image(gan_x[im_index])
```



### Combine and shuffle

*# combine the 4 datasets and shuffle together*

```
x = torch.cat((wwmr_x, fmd_x, mfn_x, gan_x))
y = torch.cat((wwmr_y, fmd_y, mfn_y, gan_y))

im_names = wwmr_im_names.copy()
im_names.extend(fmd_im_names)
im_names.extend(mfn_im_names)
im_names.extend(gan_im_names)

print(x.shape, y.shape, len(im_names))

np.unique(y, return_counts=True)

torch.Size([2973, 3, 128, 128]) torch.Size([2973]) 2973
(array([0., 1., 2.], dtype=float32), array([ 977,  981, 1015],
dtype=int64))

#shuffle

idx = np.arange(y.size(0))
np.random.shuffle(idx)

x = x[idx]
y = y[idx]
im_names = [im_names[i] for i in idx.tolist()]
```

```
#store combined data
```

```
torch.save(x, '%s/x_%d.pt' % (config.combined_data_path,  
config.im_size))  
torch.save(y, '%s/y.pt' % (config.combined_data_path))  
with open('%s/im_names.txt' % (config.combined_data_path), 'w') as o:  
    o.write('\n'.join(im_names))
```