Turning images into arrays for PyTorch ¶

In [2]:
```
import numpy as np
from PIL import Image
import matplotlib.pyplot as plt
import os
```

In [13]:
```
# in locations
incorrect = r'D:\data\face_mask\FaceMaskDetection_12k\Cropped\mask_weared_incorrect'
correct = r'D:\data\face_mask\FaceMaskDetection_12k\Cropped\with_mask'
without = r'D:\data\face_mask\FaceMaskDetection_12k\Cropped\without_mask'

# out locations
output_X_fp =  r'D:\data\face_mask\FaceMaskDetection_12k\Cropped\images'
output_y_fp =  r'D:\data\face_mask\FaceMaskDetection_12k\Cropped\labels'
```

In [11]:
```
# Turns directory of images into array of images (as pixel vals) and array of labels
def image_dir_to_list_of_arrays(input_image_dir, label: int, verbose=False):

    img_array_list = []
    for root, subdirectories, files in os.walk(input_image_dir):
        for f in files:

            im_fp = os.path.join(input_image_dir, f)

            im = Image.open(im_fp)
            im_arr = np.array(im)

            img_array_list.append(im_arr)

    length = len(img_array_list)

    if verbose:
        print('{:>5} images'.format(length))

    # list to np array
    imgs_as_np_array = np.array(img_array_list)

    # generate labels
    label_list = [label] * length
    label_array = np.array(label_list).astype(int)

    return imgs_as_np_array, label_array
```

In [12]:
```
# process correctly worn mask images
correct_array, correct_label_array = image_dir_to_list_of_arrays(
    correct,
    label=1,
    verbose=True
)
```

```
2994 images
```

In [14]:
```
# process correctly worn mask images
incorrect_array, incorrect_label_array = image_dir_to_list_of_arrays(
    incorrect,
    label=0,
    verbose=True
)
```

```
2994 images
```

In [15]:
```
# process correctly worn mask images
without_array, without_label_array = image_dir_to_list_of_arrays(
    without,
    label=0,
    verbose=True
)
```

```
2994 images
```

In [16]:
```
# concatinating arrays
FMD_12k_X = np.concatenate([correct_array, incorrect_array, without_array])
FMD_12k_y = np.concatenate([correct_label_array, incorrect_label_array, without_label_array])
```

In [17]:
```
FMD_12k_X.shape
```

Out[17]:
```
(8982, 112, 112, 3)
```

In [18]:
```
FMD_12k_y.shape
```

Out[18]:
```
(8982,)
```

In [19]:
```
# saving output to file
np.save(output_X_fp, FMD_12k_X)
np.save(output_y_fp, FMD_12k_y)
```

In [ ]:
```

```