More info found here:

<https://www.tensorflow.org/api_docs/python/tf/keras/preprocessing/image/ImageDataGenerator>

| Unzip files with  local\_zip = '/content/horse-or-human.zip'  zip\_ref = zipfile.ZipFile(local\_zip, 'r')  zip\_ref.extractall('/content/horse-or-human')  Local\_zip is the path to zip file  For .extractall method, pass in the directory you want the unzipped files to be in  with zipfile.ZipFile(path\_to\_zip\_file, 'r') as zip\_ref: zip\_ref.extractall(directory\_to\_extract\_to)  List all files in a directory with  os.listdir('/content/horse-or-human/horses') |
| --- |

Perform Augmentations on the train dataset using ImageDataGenerator

train\_datagen = ImageDataGenerator(

rescale=1/255,

rotation\_range = 40, #Range of degree to rotate the image by

width\_shift\_range=0.2, #How much ratio to shift image left and right

height\_shift\_range=0.2, #How much ratio to shift image up and down

shear\_range=0.2, #How much to flatten

zoom\_range=0.2, #How much to zoom in

horizontal\_flip=True, #Flip the image about the y axis

fill\_mode='nearest') #Attempt to recreate any missing pixels in the img

validation\_datagen = ImageDataGenerator(rescale=1/255)

For test dataset, just do rescaling only

You cannot pass in this train\_datagen for training. You need a generator to pass data for training

# Flow training images in batches of 128 using train\_datagen generator. The training labels will be auto created for you (Based on the name of the subdirectory 'horse' and 'human')

train\_generator = train\_datagen.flow\_from\_directory(

'/content/horse-or-human/', # This is the source directory for training images. Inside this folder contains 2 subdirectories (1 for horse and 1 for human).

target\_size=(300, 300), # All images will be resized to 300x300

batch\_size=128,

# Since we use binary\_crossentropy loss, we need binary labels

class\_mode='binary')

validation\_generator = validation\_datagen.flow\_from\_directory(

'/content/validation-horse-or-human/',

target\_size=(300, 300),

batch\_size=32, #1 epoch = 1 entire pass on the data set = batch\_size \* steps\_per\_epoch. 1 step\_per\_epoch = 1 batch\_size amount of pictures processed

class\_mode='binary')

Over here, specify the

* source directory (Containing all the images)
* Target size (The images will be reshaped to this)
* Class\_mode ("categorical", "binary", "sparse", "input", or None)
* Batch\_size
* Class\_mode (Based on the type of labels)

| Exercise  !wget --no-check-certificate \  https://storage.googleapis.com/laurencemoroney-blog.appspot.com/validation-horse-or-human.zip \  -O /content/validation-horse-or-human.zip  !wget --no-check-certificate \  https://storage.googleapis.com/laurencemoroney-blog.appspot.com/horse-or-human.zip \  -O /content/horse-or-human.zip  Model answer: <https://colab.research.google.com/drive/1zVDiCv6MZf9692VL5Pr1rUbpp3T0c6su?usp=sharing> |
| --- |

\*Its very important to choose the right optimizer.

When Adam is used, the accuracy isn't that good (50%+ after 2 epochs).

When RMSProp is used, the accuracy is very good (80%+ after 2 epochs)