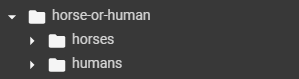
Notebook here:

<https://colab.research.google.com/drive/10_3J2S-vlbSQlpqFEWk4Wfgzs1BKWiEl?usp=sharing>



* This is the file structure. Inside the horses file is all the images of horses. Same for humans

def create\_dataset(path, shape,mode='rgb'):

numiter = 0

imgList = []

ylist = []

for i in os.listdir(path):

sub\_path = os.path.join(path, i)

print(i)

for j in os.listdir(sub\_path):

pic\_path = os.path.join(sub\_path, j)

if mode == 'grayscale':

img = Image.open(pic\_path).convert('L')

elif mode == 'rbg':

img = Image.open(pic\_path)

img = img.resize((shape))

data = np.array(img)

#Since PIL returns a color channel of 4 (RGBA where A is the transparency) but you want 3 color channels (RBG), you can just slice the A channel off

data = data[:,:,:3]

imgList.append(data)

if i == 'horses':

ylist.append(1)

elif i =='humans':

ylist.append(0)

number += 1

imgList = np.array(imgList)

ylist = np.array(ylist)

print(numiter)

return imgList, ylist

x\_train, y\_train = create\_dataset('/content/horse-or-human', (300,300), mode='rgb')

for i in os.listdir(path):

sub\_path = os.path.join(path, i)

print(i)

* What this does it to loop through all the sub directories (which is ‘horses’ and ‘humans’)

in the main dir (which is ‘horse-or-human’)

* sub\_path is ‘horse-or-human/horses’ or ‘horse-or-human/humans’

for j in os.listdir(sub\_path):

pic\_path = os.path.join(sub\_path, j)

if mode == 'grayscale':

img = Image.open(pic\_path).convert('L')

elif mode == 'rbg':

img = Image.open(pic\_path)

img = img.resize((shape))

data = np.array(img)

#Since PIL returns a color channel of 4 (RGBA where A is the transparency) but you want 3 color channels (RBG), you can just slice the A channel off

data = data[:,:,:3]

imgList.append(data)

if i == 'horses':

ylist.append(1)

elif i =='humans':

ylist.append(0)

number += 1

* Pic\_path = ‘horse-or-human/horses/horse01.jpg’ or ‘horse-or-human/humans/human01.jpg’
* Load image using PIL.Image.open()
* If want grayscale, convert it using .convert(‘L’)
* Resize the img to desired size and convert it to an array
* Change color channels from 4 to 3 (Since if you pass through PIL.Image it gives RGBA, not RGB). Slice the A (transparency) channel
* Get the y value, based on its directory (0 if human, 1 if horse)
* Number is just there for debugging (see how many examples were appended to list)

imgList = np.array(imgList)

ylist = np.array(ylist)

print(numiter)

return imgList, ylist

x\_train, y\_train = create\_dataset('/content/horse-or-human', (300,300), mode='rbg')

* After looping through all the images, convert the list of image arrays to a np array
* Also convert the list of y values to a numpy array