import numpy as np  
import matplotlib.pyplot as plt  
from tensorflow.keras.layers import Input, Dense  
from tensorflow.keras.models import Model  
from tensorflow.keras.preprocessing.image import ImageDataGenerator  
  
# Define the path to your CelebA dataset  
data\_dir = 'path/to/celeba-dataset' # Replace 'path/to/celeba-dataset' with the actual path  
  
# Define image dimensions and batch size  
img\_width, img\_height = 64, 64  
batch\_size = 256  
  
# Use ImageDataGenerator to load and preprocess images  
datagen = ImageDataGenerator(rescale=1. / 255)  
  
train\_generator = datagen.flow\_from\_directory(  
 data\_dir,  
 target\_size=(img\_width, img\_height),  
 batch\_size=batch\_size,  
 class\_mode=None,  
 shuffle=True)  
  
# Define the autoencoder model  
encoding\_dim = 64 # Size of the encoded representations  
input\_img = Input(shape=(img\_width \* img\_height \* 3,)) # For CelebA images  
encoded = Dense(encoding\_dim, activation='relu')(input\_img)  
decoded = Dense(img\_width \* img\_height \* 3, activation='sigmoid')(encoded)  
  
autoencoder = Model(input\_img, decoded)  
  
# Compile the autoencoder  
autoencoder.compile(optimizer='adam', loss='binary\_crossentropy')  
  
# Train the autoencoder  
autoencoder.fit(train\_generator, steps\_per\_epoch=len(train\_generator), epochs=50)  
  
# Display original and reconstructed images  
n = 10 # Number of images to display  
plt.figure(figsize=(20, 4))  
for i in range(n):  
 # Original images  
 ax = plt.subplot(2, n, i + 1)  
 plt.imshow(train\_generator[i])  
 plt.gray()  
 ax.get\_xaxis().set\_visible(False)  
 ax.get\_yaxis().set\_visible(False)  
  
 # Reconstructed images  
 ax = plt.subplot(2, n, i + 1 + n)  
 reconstructed\_img = autoencoder.predict(train\_generator[i])  
 plt.imshow(reconstructed\_img.reshape(img\_width, img\_height, 3))  
 plt.gray()  
 ax.get\_xaxis().set\_visible(False)  
 ax.get\_yaxis().set\_visible(False)  
  
plt.show()

---------------------------------------------------------------------------  
FileNotFoundError Traceback (most recent call last)  
<ipython-input-5-88cf1ed02360> in <cell line: 17>()  
 15 datagen = ImageDataGenerator(rescale=1. / 255)  
 16   
---> 17 train\_generator = datagen.flow\_from\_directory(  
 18 data\_dir,  
 19 target\_size=(img\_width, img\_height),  
  
/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py in flow\_from\_directory(self, directory, target\_size, color\_mode, classes, class\_mode, batch\_size, shuffle, seed, save\_to\_dir, save\_prefix, save\_format, follow\_links, subset, interpolation, keep\_aspect\_ratio)  
 1647 and `y` is a numpy array of corresponding labels.  
 1648 """  
-> 1649 return DirectoryIterator(  
 1650 directory,  
 1651 self,  
  
/usr/local/lib/python3.10/dist-packages/keras/src/preprocessing/image.py in \_\_init\_\_(self, directory, image\_data\_generator, target\_size, color\_mode, classes, class\_mode, batch\_size, shuffle, seed, data\_format, save\_to\_dir, save\_prefix, save\_format, follow\_links, subset, interpolation, keep\_aspect\_ratio, dtype)  
 561 if not classes:  
 562 classes = []  
--> 563 for subdir in sorted(os.listdir(directory)):  
 564 if os.path.isdir(os.path.join(directory, subdir)):  
 565 classes.append(subdir)  
  
FileNotFoundError: [Errno 2] No such file or directory: 'path/to/celeba-dataset'