COSC 6373 - HW9-ICA - Minh Nguyen #2069407

In [2]:
```
import os
import keras
from keras.models import Sequential
from keras.layers import Dense, Activation, Flatten, Input
from keras.layers import Conv2D, MaxPooling2D, UpSampling2D
import matplotlib.pyplot as plt
from keras import backend as K
import numpy as np
# from keras.preprocessing.image import ImageDataGenerator, array_to_img, im
from PIL import Image, ImageChops
from sklearn.neighbors import KernelDensity
import random
```

In [ ]:
```
# Define the convolutional autoencoder model
# input shape must be the same size as the images that will be fed into it b
# The output layer must be the same dimensions as the original image
model = Sequential()

model.add(Conv2D(16, (3, 3), padding='same', activation='relu', input_shape=
model.add(MaxPooling2D(pool_size=(4,4), padding='same')) # using pool_size (

model.add(Conv2D(8,(3, 3),activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(4,4), padding='same'))

model.add(Conv2D(3,(3, 3),activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2,2), padding='same'))

#-------------------------
model.add(Conv2D(3,(3, 3),activation='relu', padding='same'))
model.add(UpSampling2D((2, 2)))

model.add(Conv2D(8,(3, 3),activation='relu', padding='same'))
model.add(UpSampling2D((4, 4)))

model.add(Conv2D(16,(3, 3),activation='relu', padding='same'))
model.add(UpSampling2D((4, 4)))

model.add(Conv2D(3,(3, 3), activation='sigmoid', padding='same'))
#-------------------------

model.summary()

# Compile the model
model.compile(optimizer='adadelta', loss='mean_squared_error')
```

```
/Users/ndminh/miniconda3/lib/python3.12/site-packages/keras/src/layers/convolutional/base_conv.py:107: UserWarning: Do not pass an `input_shape`/`input_dim` argument to a layer. When using Sequential models, prefer using an `Input(shape)` object as the first layer in the model instead.
  super().__init__(activity_regularizer=activity_regularizer, **kwargs)
```

Model: "sequential"

Layer (type)	Output Shape	Par
conv2d (Conv2D)	(None, 96, 96, 16)	
max_pooling2d (MaxPooling2D)	(None, 24, 24, 16)	
conv2d_1 (Conv2D)	(None, 24, 24, 8)	1
max_pooling2d_1 (MaxPooling2D)	(None, 6, 6, 8)	
conv2d_2 (Conv2D)	(None, 6, 6, 3)	
max_pooling2d_2 (MaxPooling2D)	(None, 3, 3, 3)	
conv2d_3 (Conv2D)	(None, 3, 3, 3)	
up_sampling2d (UpSampling2D)	(None, 6, 6, 3)	
conv2d_4 (Conv2D)	(None, 6, 6, 8)	
up_sampling2d_1 (UpSampling2D)	(None, 24, 24, 8)	
conv2d_5 (Conv2D)	(None, 24, 24, 16)	1
up_sampling2d_2 (UpSampling2D)	(None, 96, 96, 16)	
conv2d_6 (Conv2D)	(None, 96, 96, 3)	

Total params: 3,738 (14.60 KB)

Trainable params: 3,738 (14.60 KB)

Non-trainable params: 0 (0.00 B)