CS156 LBA

Armin Hamp

Principle Component Analysis

Original Code by Joel Grus, author of the blog referenced in the preclass-work: https://github.com/joelgrus/shirts/blob/master/visuals.py (https://github.com/joelgrus/shirts/blob/master/visuals.py)

In [140]:
```
#import libraries
from PIL import Image
import PIL.ImageOps
from matplotlib import pyplot as plt
from glob import glob
import numpy as np
import pylab as pl
import pandas as pd
from sklearn.decomposition import PCA
import random
```

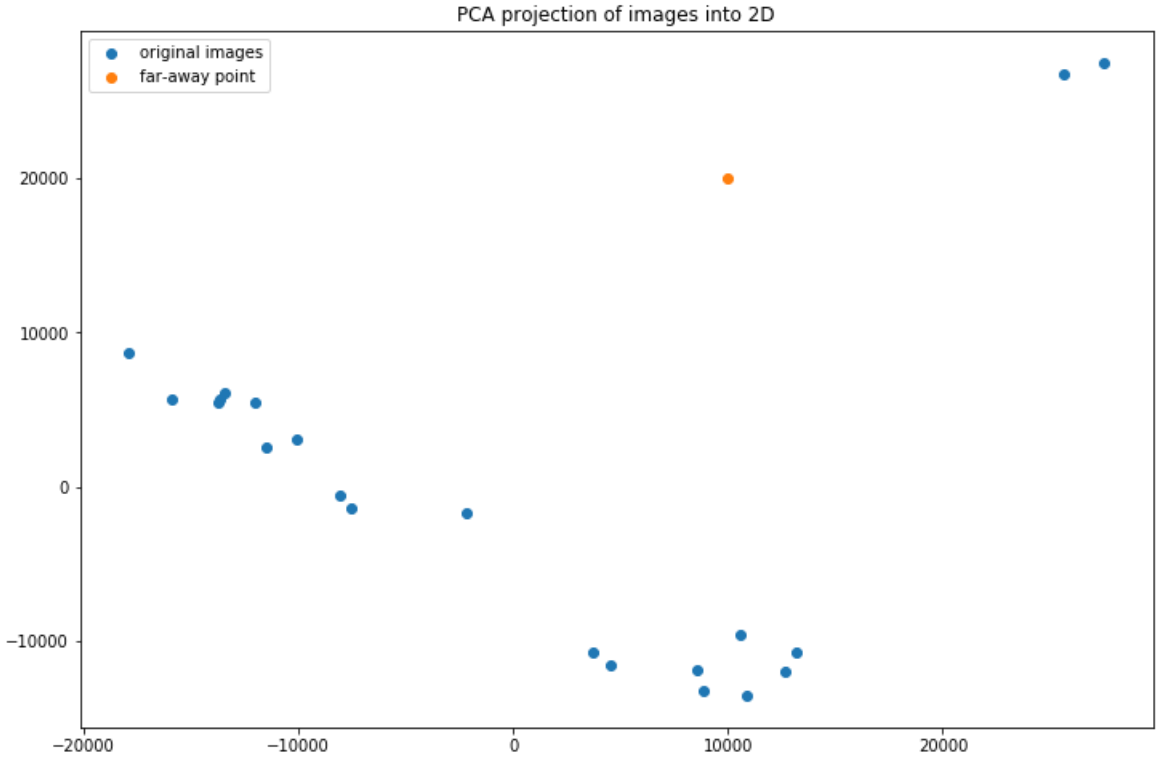
In [141]:
```
new_size=(512,384) #this size allows me to keep the original ratio
def img_to_array(filename):
    """
    takes a filename and turns it into a numpy array of RGB pixels
    """
    img = Image.open(filename)
    img = img.resize(new_size)
    img=img.rotate(270, expand=True)
    img = list(img.getdata())
    img = map(list, img)
    img = np.array(list(img))
    s = img.shape[0] * img.shape[1]
    #print(img.shape[0], img.shape[1])
    img_wide = img.reshape(1, s)
    return img_wide[0]
def array_to_image(data):
    d
```

In [142]:
```
#fetch picture addresses
pics=glob('/Users/Armin/Documents/Minerva Year 3/Semester 2/CS156/LBA/Pictures/*')
```

In [171]:
```
#convert pictures to raw data
raw_data = [img_to_array(filename) for filename in pics]

#display a randomly selected, resized image
a=raw_data[random.randint(0,21)].reshape(512,384,3)
plt.figure(figsize=(10,10))
plt.axis('off')
plt.imshow(a)
plt.show()
```

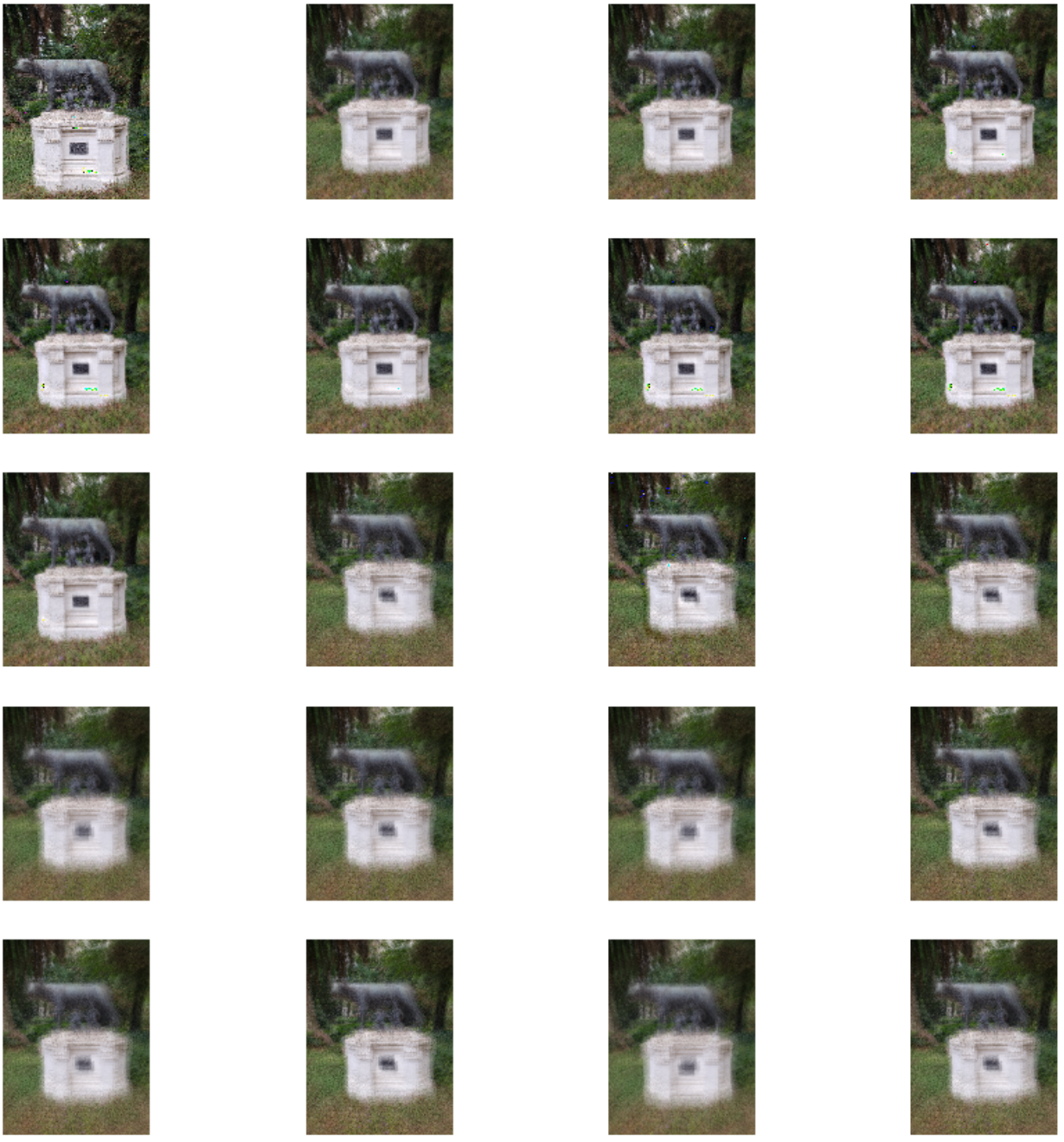
In [144]:
```
# Apply PCA transformation to data
N_COMPONENTS = 2
pca = PCA(n_components=N_COMPONENTS, svd_solver="randomized") #create model
pca_data0= pca.fit_transform(raw_data) # fit model and apply transformation
pca_data = pca_data0.transpose() #transpose matrix for plotting
```

In [169]:
```
#plot pca transformed images in 2D
plt.figure(figsize=(12,8))
plt.scatter(pca_data[0],pca_data[1], label="original images")
plt.title("PCA projection of images into 2D")

far_point=[10000,20000]
plt.scatter(far_point[0],far_point[1], label="far-away point")
plt.legend()
plt.show()
```



In [156]:

```
#reconstruct images from their 2D representation
reconstructions=pca.inverse_transform(pca_data0)
restored_imgs = [i.reshape(512,384,3) for i in reconstructions]

#display all reconstructed images
fig=plt.figure()
#fig.axis('off')
for i in range(1, 21):
    fig.add_subplot(5, 4, i)
    plt.imshow(restored_imgs[i].astype('uint8'))
    plt.axis('off')
fig.set_figheight(15)
fig.set_figwidth(15)
plt.show()
```



In [166]:

```
#far away point
far=[10000,20000]
reconstructed_far=pca.inverse_transform(far).reshape(512,384,3)
plt.figure(figsize=(10,10))
plt.axis('off')
plt.imshow(reconstructed_far.astype('uint8'))
plt.show()
```