Homework 1

Crepaldi Giorgio, Fiandro Andrea, Francois Jeanpierre

November 20, 2018

Contents

Chapter 1

Principal Component Visualization

1.0.1 Data standardization

The first step is to *standardize* the data, in order to apply correctly the PCA algorithm. We found that the function *transform* provided by the *sklearn* library include this preprocessing operation.

1.0.2 Image reprojection

The next step is to choose one single image and reproject it using only some components. This part is implemented with these lines of code:

Reproject the image with a specific number of PC

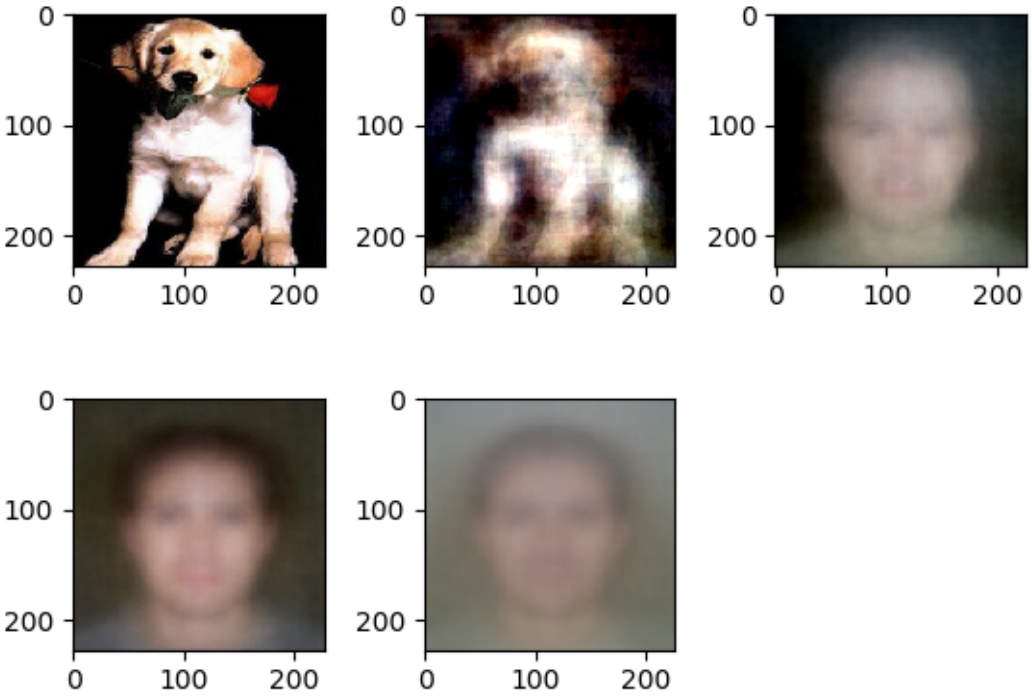
```
pca_x = PCA(n_pca)
projected = pca_x.fit_transform(input_matrix)
x_inv = pca_x.inverse_transform(projected)
```

Print the new image beside the original one

```
fig = plt.figure()
fig.add_subplot(1,2,1)
plt.imshow(np.reshape(input_matrix[nImg,:]/255.0,(227,227,3)))
fig.add_subplot(1,2,2)
plt.imshow(np.reshape(x_inv[nImg,:]/255.0,(227,227,3)))
plt.show()
```

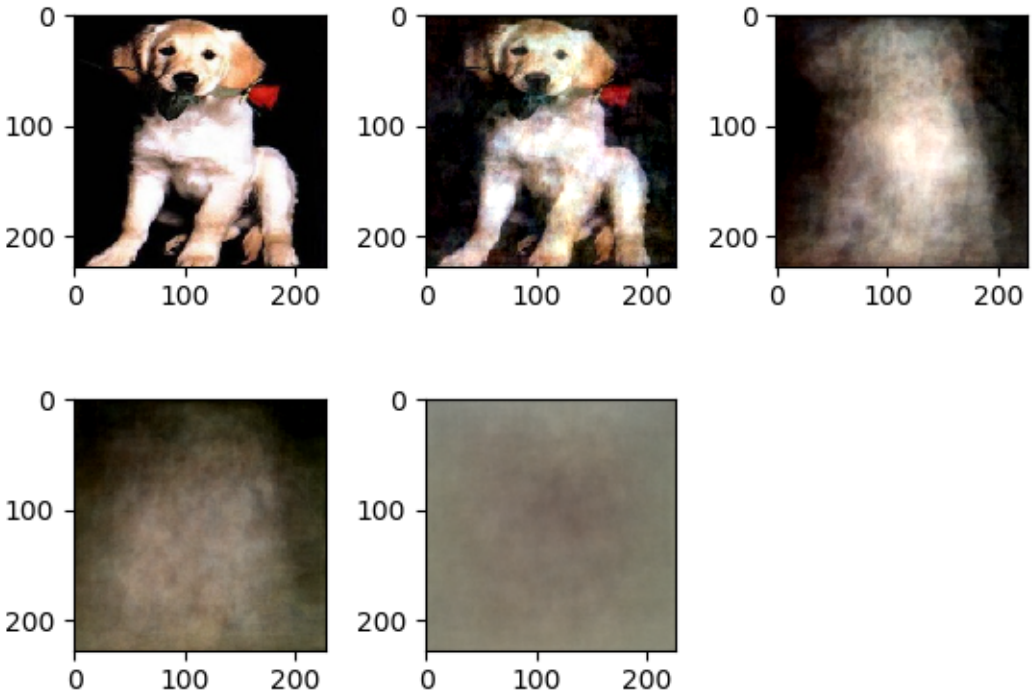
Image reprojection using the whole dataset This is the representation of a random image reprojected using:

- the first 60 PC
- the first 6 PCA
- the first 2 PCA
- the last 6 PCA



As we can see, with the first images we have a good representation of the original one. But then we start getting pictures that looks like a human face. This behaviour depends on our dataset that is not balanced and includes a lot of faces and few dogs.

Image reprojection using only dogs Here we tried to reproject images using only dog pictures in our dataset. This is the result that we get:

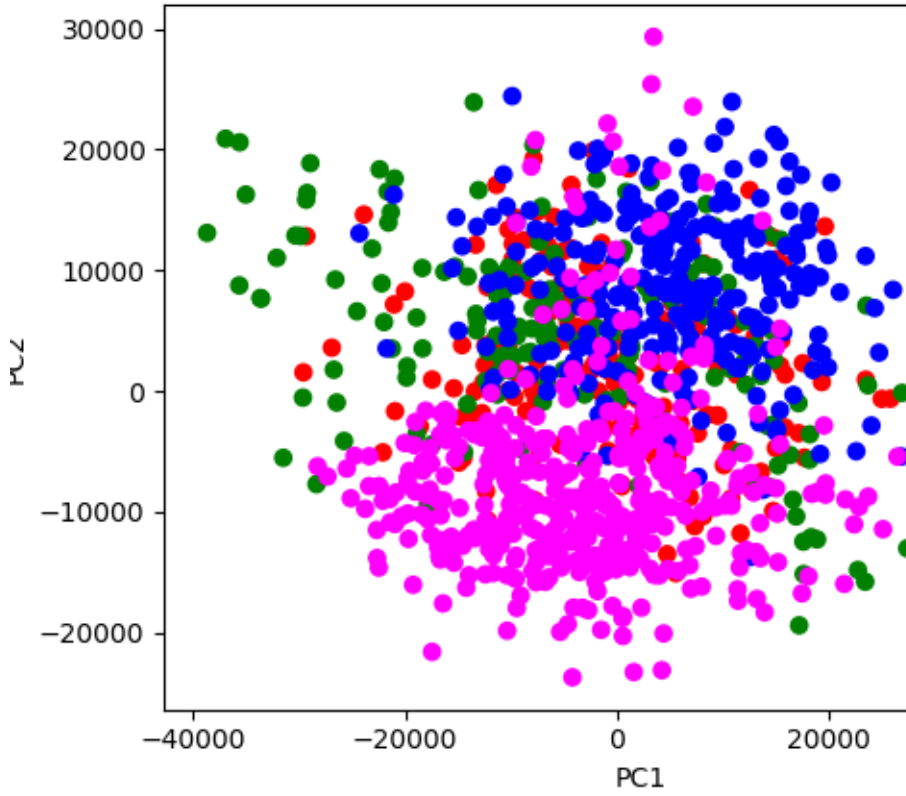


In this scenario we obviously have a better result.

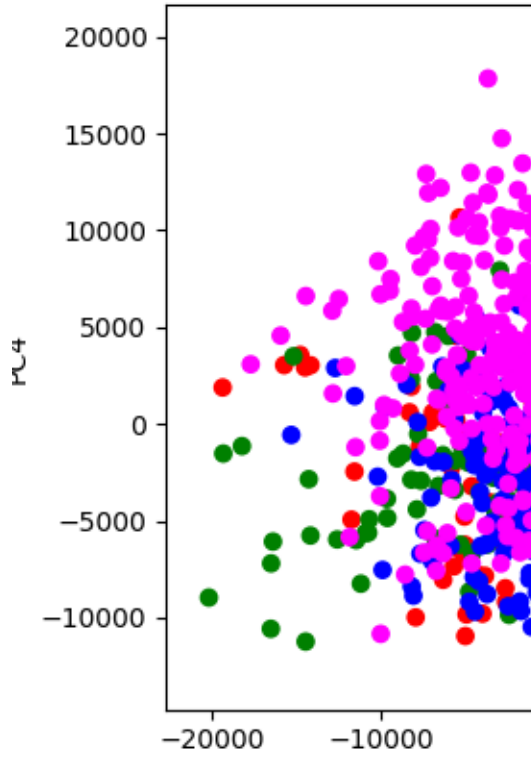
1.0.3 Scatter plot showing different classes

First of all, we define an array that, for each sample, define the class. In this case we have 4 classes:

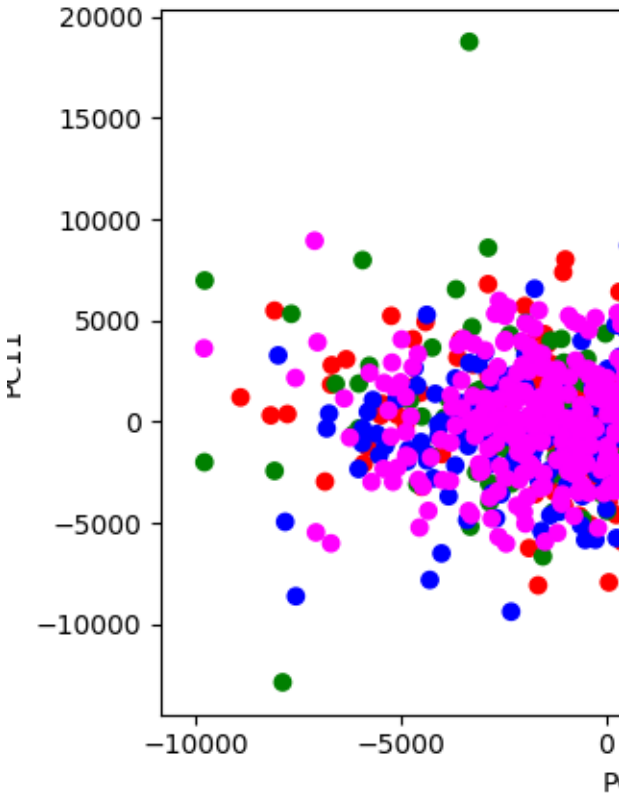
- Dogs
- Guitar
- House
- Person



Scatter plot with first 2 PC



Scatter plot with third and fourth principal components



Scatter plot with tenth and eleventh components