Ex3: PCA - sklearn

Eigenfaces

- Sử dụng bộ dữ liệu các khuôn mặt sklearn.datasets.fetch_lfw_people, lấy min_faces_per_person=60
- Áp dụng PCA: chúng ta sẽ giảm chiều dữ liệu còn 150 chiều (gốc là ~3000 chiều)
- Trực quan hóa dữ liệu gốc và sau khi giảm chiều

In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.decomposition import PCA
```

In [2]:
```
from sklearn.datasets import fetch_lfw_people
```

In [3]:
```
faces = fetch_lfw_people(min_faces_per_person=60)
```

In [4]:
```
type(faces)
```

Out[4]:
```
sklearn.utils.Bunch
```

In [5]:
```
print(faces.target_names)
print(faces.images.shape)
```

```
['Ariel Sharon' 'Colin Powell' 'Donald Rumsfeld' 'George W Bush'
 'Gerhard Schroeder' 'Hugo Chavez' 'Junichiro Koizumi' 'Tony Blair']
(1348, 62, 47)
```

In [6]:
```
faces.images[0].shape
```

Out[6]:
```
(62, 47)
```

In [7]:
```
faces.data[0].size
```

Out[7]:
```
2914
```

In [8]:
```
from sklearn.decomposition import PCA
```

In [9]:
```
pca = PCA(150)
pca.fit(faces.data)
```

Out[9]:
```
PCA(copy=True, iterated_power='auto', n_components=150, random_state=None,
  svd_solver='auto', tol=0.0, whiten=False)
```

In [10]:
```
pca.explained_variance_ratio_
```

Out[10]:
```
array([0.1878269 , 0.14550328, 0.07100105, 0.06029004, 0.05040162,
       0.02936219, 0.02469285, 0.02047788, 0.01968621, 0.01891433,
       0.01561243, 0.01469926, 0.01214477, 0.01095759, 0.01042821,
       0.0097199 , 0.00906832, 0.00877006, 0.00813053, 0.00704724,
       0.00682862, 0.00647883, 0.00603494, 0.00578391, 0.00532262,
       0.00520684, 0.00500025, 0.00476452, 0.00452436, 0.00425207,
       0.00405164, 0.0038007 , 0.00359863, 0.00350867, 0.00347822,
       0.00324879, 0.00314475, 0.00310487, 0.00307685, 0.00289922,
       0.00282612, 0.00274748, 0.00272808, 0.0025997 , 0.00246546,
       0.002382  , 0.00235044, 0.0023152 , 0.00227269, 0.00221832,
       0.00210567, 0.00205797, 0.00202996, 0.0020065 , 0.00195754,
       0.00195432, 0.00188095, 0.00182764, 0.00176693, 0.00175856,
       0.00174926, 0.0016632 , 0.00161306, 0.00158565, 0.00156576,
       0.00152887, 0.00149985, 0.00146118, 0.00145263, 0.00141041,
       0.00140521, 0.00136414, 0.00136122, 0.00131623, 0.00129271,
       0.00125563, 0.00124939, 0.00123093, 0.00120673, 0.00118767,
       0.00117411, 0.00115423, 0.00113158, 0.00110182, 0.00108854,
       0.00107467, 0.00105291, 0.00103628, 0.00101902, 0.00101126,
       0.00098039, 0.0009782 , 0.00095506, 0.00094166, 0.00092525,
       0.00092251, 0.00088968, 0.00087129, 0.00085994, 0.00085601,
       0.00085047, 0.00082554, 0.00081631, 0.00080155, 0.00078466,
       0.00077574, 0.00076287, 0.00074693, 0.00074172, 0.00073724,
       0.00072111, 0.00070959, 0.00069912, 0.00069208, 0.00068837,
       0.00068219, 0.00067487, 0.0006563 , 0.00065156, 0.00063699,
       0.00062868, 0.00061479, 0.00060915, 0.00060389, 0.00059112,
       0.00058045, 0.00057742, 0.00057093, 0.00056657, 0.00055678,
       0.00055088, 0.00054134, 0.00053235, 0.00051981, 0.0005119 ,
       0.00051024, 0.00050068, 0.00049567, 0.00049261, 0.00048454,
       0.00047829, 0.00046883, 0.00046106, 0.00045559, 0.00044983,
       0.00044792, 0.00044113, 0.0004285 , 0.00042473, 0.00041473],
      dtype=float32)
```

In [17]:
```
x = sum(pca.explained_variance_ratio_)
```

In [18]:
```
x
```

Out[18]:
```
0.9456852865987457
```

In [11]:
```
components = pca.transform(faces.data)
projected = pca.inverse_transform(components)
```

In [12]:
```
# Plot the results
fig, ax = plt.subplots(2, 10, figsize=(10, 2.5),
                       subplot_kw={'xticks':[], 'yticks':[]},
                       gridspec_kw=dict(hspace=0.1, wspace=0.1))
for i in range(10):
    ax[0, i].imshow(faces.data[i].reshape(62, 47), cmap='binary_r')
    ax[1, i].imshow(projected[i].reshape(62, 47), cmap='binary_r')

ax[0, 0].set_ylabel('full-dim\ninput')
ax[1, 0].set_ylabel('150-dim\nreconstruction')
```

Out[12]:
```
Text(0,0.5,'150-dim\nreconstruction')
```

In [ ]:
```
```