IML: Programming Assignment #3

Christian Wiskott
a01505394@unet.univie.ac.at

University of Vienna — January 21, 2021

Abstract

Task 1 was completed and uploaded to the Gitlab repository. Due to time constraints task 2 was not completed thus will not be mentioned in this report. The file of task 1 requires no input parameters and can simply be run with a standard IDE or from the command-line. The print statements produce the relevant results.

Task 1: Dimensionality Reduction

The objective of this task was to use the "*Labeled Faces in the Wild*" dataset to reconstruct the contained images using *PCA* and *autoencoders* as dimensionality-reduction techniques and to evaluate their performances.

Subtask 1: Data Loading and Data Preparation

The dataset contains 2370 images of 34 different people, each with a size of 62 x 47 Pixels. Fig. 1 shows 10 images from different people taken from the dataset. They were plotted using plt's grayscale and show relatively fine facial features.

For further analysis the dataset was split into a stratified train-test-split using *test_size=0.3* and *train_size=0.7*.

Subtask 2: Dimensionality Reduction Using PCA

The class *PCA_custom* was created to compute the k most important components (*PCs*), project the data onto them and compute the reconstructed images. This was implemented in the style of sklearn's *fit, fit_transform* and *transform* functions to calculate the projections onto the PCs.

With the fitted model via:

PCA_custom fitting

```
pca = PCA_custom(k=d)
xtrain_pca = pca.fit_transform(xtrain)
xtest_pca = pca.transform(xtest)
```

the first five PCs were plotted as images (see fig. 2). By studying these images intently, one can notice subtle differences w.r.t. the facial features.

Fig. 3 shows 10 chosen reconstructed faces for each d. As d increases, the resulting images become increasingly similar to the original images. But due to the increased computational cost of higher number of components, there is of course a trade-off between accuracy and cost.

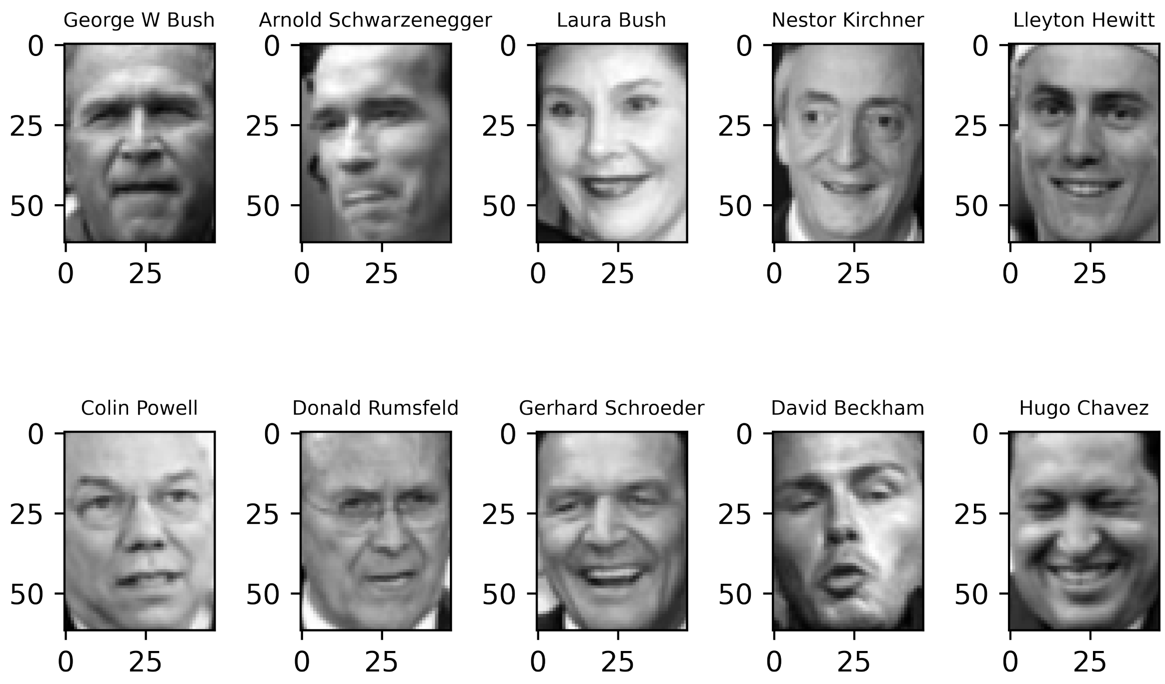

Figure 1: Original images taken from the dataset. Shows ten different people using plt's grayscale.

Reconstruction is done by multiplying the computed projections with the components of the PCA-class plus the mean computed during the fit.

Reconstruction
```
  reconstructed = (xtrain_pca @ pca.components_) + pca.mean_
```

In order to measure the quality of the reconstruction, sklearn's *LogisticRegression* classifier was fitted via:

Logistic Regression fitting
```
clf = LogisticRegression(max_iter=1000)
clf.fit(xtrain_pca, ytrain)
```

and the resulting classification scores were computed. Table 1 shows the corresponding results. Until *d=160* both scores generally increase as the number of components increases. But with higher number of components the model is clearly over-fitted and the validation score begins to deteriorate. The best *d* is 160 as this maximizes the validation score with 62.17 %.

The runtime for the entire subtask 2 $\sim 2\ min$, using the university server.

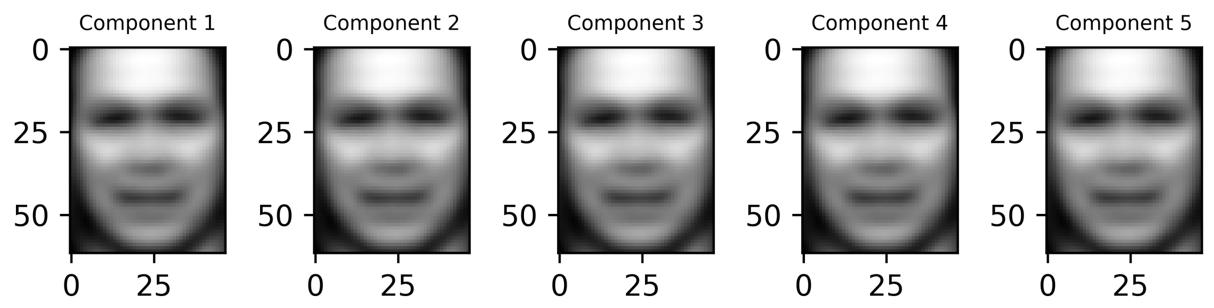

Figure 2: Plot of the first five principal components as images.

Subtask 3: Dimensionality Reduction Using Autoencoders

In order to use autoencoders for the dimensionality reduction and reconstruction of the images, the *MLPRegression* class was fitted via:

MLPRegressor fitting

```
regr = MLPRegressor(hidden_layer_sizes=((500, 100, d, 100, 500)),
        max_iter=500, activation='relu', random_state=0)
regr.fit(xtrain, xtrain)
```

For encoding and decoding the images, the provided *encode* and *decode* functions were used. Fig. 4 shows 10 reconstructed images using the autoencoders for each dimention d. For computing the classification scores, the *LogisticRegression* class was fitted with the training data and the respective scores were computed for each d. Those can be seen in table 2.

The runtime for the entire subtask 3 $\sim 19\ min$, using the university server.

Conclusion

The comparison of both techniques shows that the PCA seems to provide a better performance due to faster computation and higher maximal validation scores. The visual inspection shows that both methods lead to similar results when given the same number of features or the dimension for autoencoders. Due to the faster computation of PCA, it is possible to increase the number of components and thus to obtain better results as compared to the autoencoders.

d	Acc_{Train} [%]	Acc_{Test} [%]
5	24.71	24.61
10	30.08	28.97
20	38.28	33.47
40	57.32	47.54
80	73.48	55.56
160	86.80	62.17
320	96.98	57.81
640	100.0	48.81

Table 1: Comparison of training- and test classification accuracies while using a different number of PCA components d. As d increases beyond 160, the model exhibts overfitting and the validation score begins to drop. The data was scaled via the *MinMaxScaler*.

d	Acc_{Train} [%]	Acc_{Test} [%]
40	38.58	33.48
80	43.34	35.44

Table 2: Classification accuracies of training- and validation set using autoencoders and different values of dimension d.

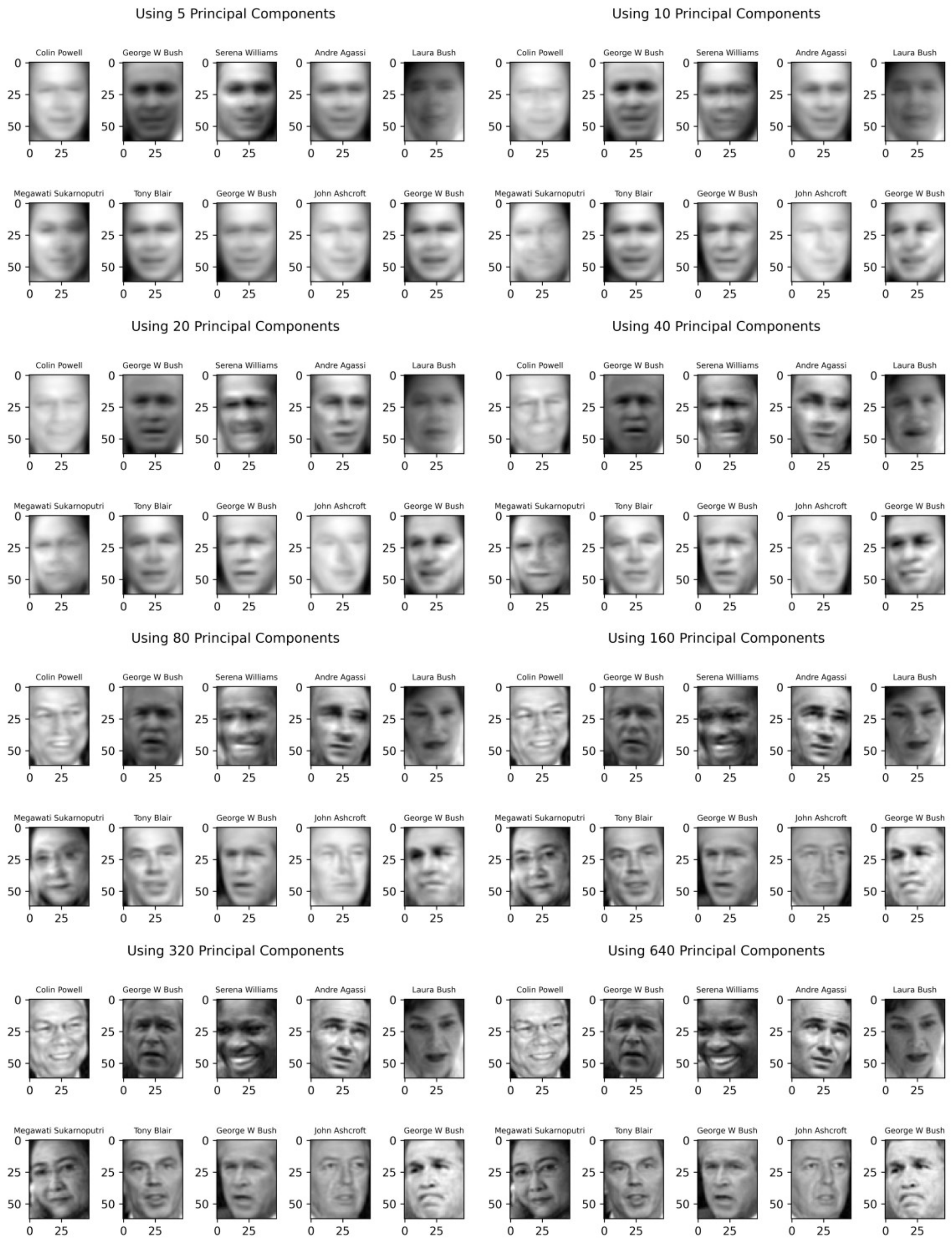
Using 5 Principal Components
Using 10 Principal Components
Using 20 Principal Components
Using 40 Principal Components
Using 80 Principal Components
Using 160 Principal Components
Using 320 Principal Components
Using 640 Principal Components
Colin Powell
George W Bush
Serena Williams
Andre Agassi
Laura Bush
Megawati Sukarnoputri
Tony Blair
George W Bush
John Ashcroft
George W Bush

Figure 3: Plot of the same 10 reconstructed images using the PCs of the *PCA_custom* class for increasing number of components d. The more components are used the finer the facial features. The faces become recognizable with around 160 components.

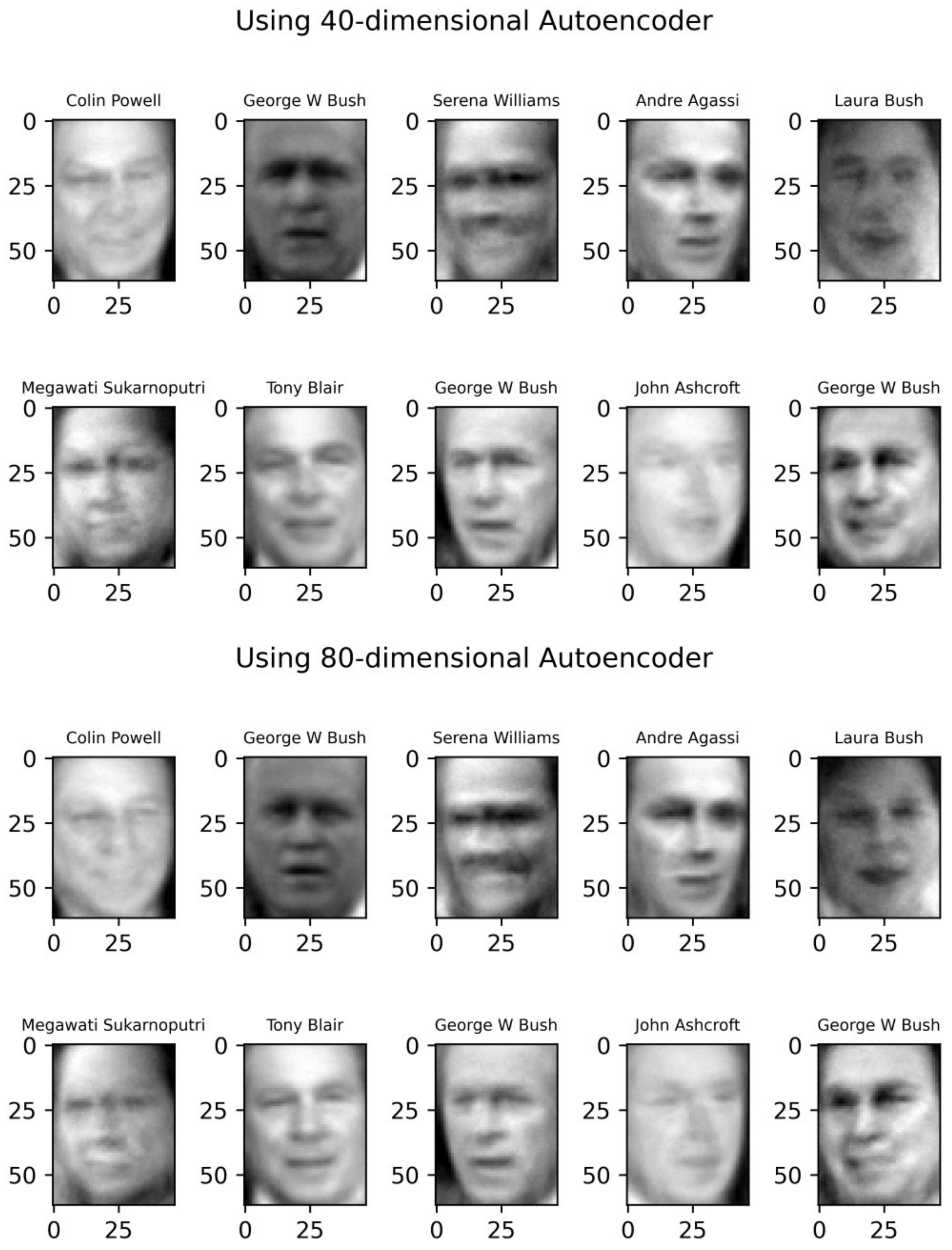

Figure 4: 10 chosen faces using autoencoders with different dimension d.