Lab Course Machine Learning
— Problem Set 1—

Oliver Stoll, Anton Hopmann

May 8, 2022

Part 1: Implementation

Assignment 1 PCA

The implementation of the principle component analysis class is completed within a few steps. First, we implement the initialisation of the class, in which we calculate the center of the data C, by which we then center the data. Having centered the data, we can calculate the covariance matrix on the transposed input matrix and extract the eigenvalues and eigenvectors with the eigendecomposition. The eigenvalues and eigenvectors are then sorted by the significance of the eigenvalues in decending order and stores as D and U.

The projection is now only a matrix multiplication of the centered data and the first m principle components.

Subsequently, in the denoise function, only a projection of the data and then a matrix multiplication of the projection with the first m principle components (transposed and centered) is required to project them back from the smaller space and thereby denoise it from noise existent in less relevant principle components.

Assignment 2 γ-index

The γ-index is implemented by calculating the distances to all neighbors for each data point. These distances are then sorted in ascending order and the first k values (excluding the data point itself) are extracted. A list of such a vector for each data point is then returned.

Assignment 3 AUC

The function returning the area under the curve for the receiver operating curve is implemented using the false- and true positive rate calculated beforehand. These rates are then used to plot the curve using matplotlib if needed and to calculate the area under the curve with np.trapz. The calculated AUC is then returned.

Assignment 4

Part 2: Application

Assignment 5 USPS

For this application of the pca, we first load the usps data set .mat file using scipy.io. The data and labels are then extracted from the file, where the data needs to be transposed to fit the normal convention of the data points being the first axis.

To solve the visualization, denoising and example images plotting for all four noise scenarios (no noise, low gaussian noise, high gaussian noise and outliers noise), we loop over these four scenarios with a list of appropriate noise variances. As noise variances, 0.2 was chosen for the the low gaussian noise scenario, 0.5 for the high gaussian noise scenario and 0.8 for the outliers.

At the beginning of the loop, the noise is added by adding np.random.normal() with the selected noise level. The exception to this is the outliers noise scenario, where the noise was only added to five images. The pca was then calculated using the implementation in 1.1 and the principle values and components were extracted.

These principle values were then plotted as a bar plot, once for all values and once for the first 25 values using plt.bar(). In figure 1, we can see an example of this visualisation. Here, due to the noise added we can see that the principle values in descending order do not near zero. The first 5 principle components for each scenario were visualized using plt.imshow().

Then, the comparison plot of the first 10 images was created by plotting three rows of the first 10 images, original, with added noise and finally denoised. In figure 2, we can see an example of this for the outliers scenario. The first 5 images are the outliers, which are distorted beyond human recognition with a noise variance factor of 0.8. The principle component analysis can denoise the images so that even though the noisy examples denoised are less clear than the normal denoised examples, some form of shape can now be recognised that previously was fully hidden in the noise.

Some experimentation on the optimal number of principal components to be included for denoising had to be done to find, that a parameter either less or greater than 25 seems to visually decrease the quality of the denoising reconstruction.

Assignment 6

Assignment 7

Assignment 8

Assignment 9

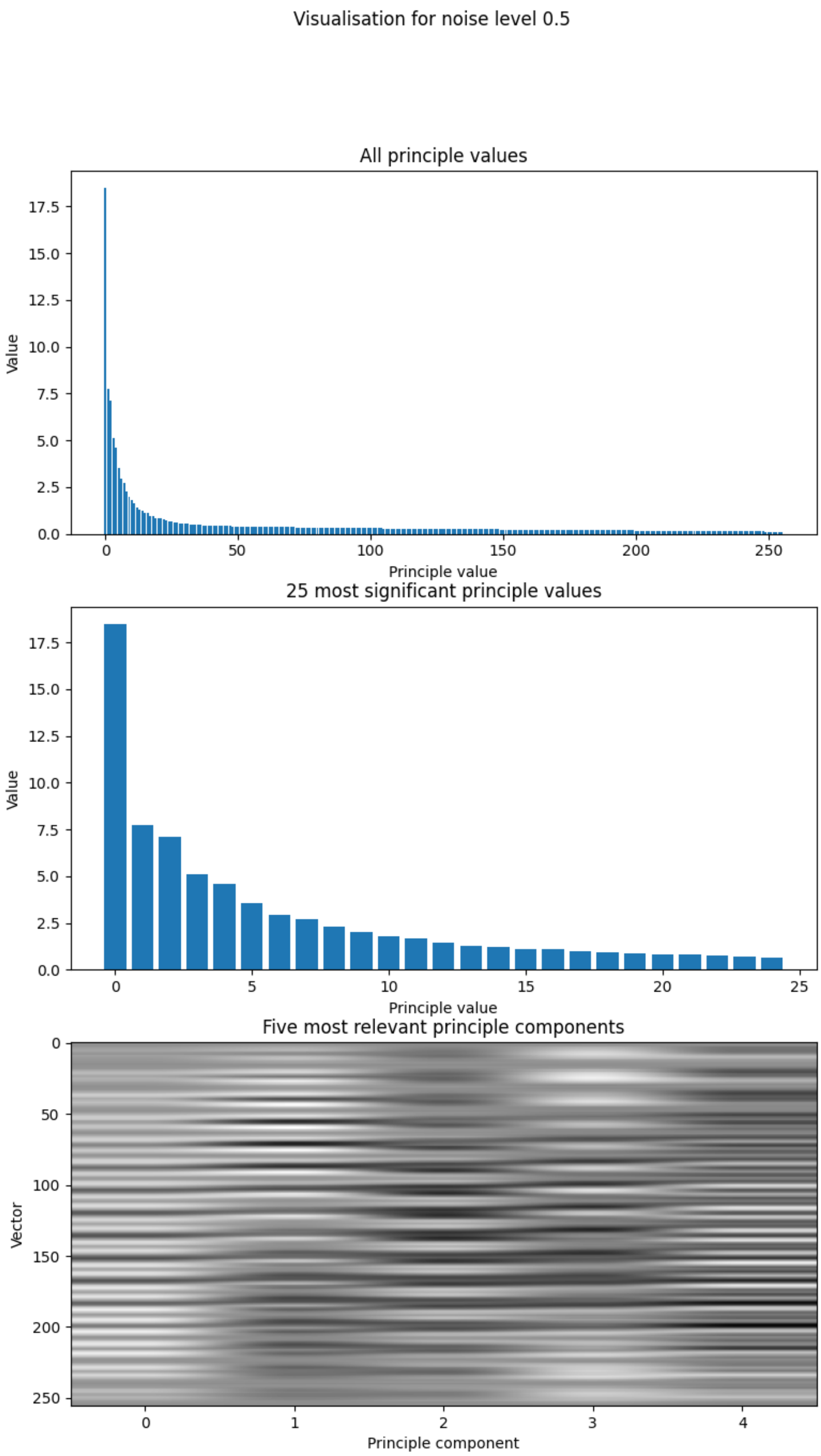

Figure 1: Example visualisation of the principle values and components for the high gaussian noise scenario

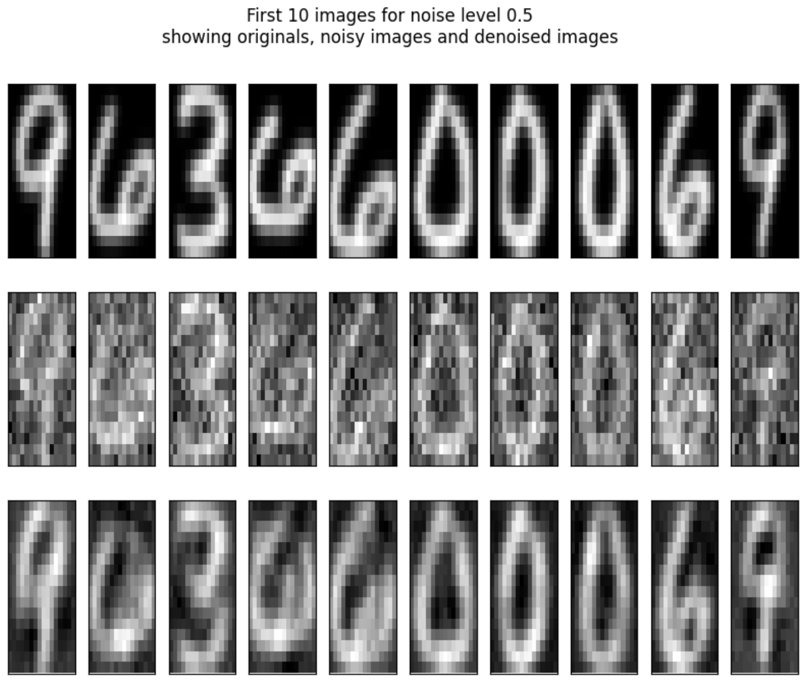

Figure 2: Example visualisation of the first 10 images for the high gaussian noise scenario