Computer Assignment 2

Ioannis Michalopoulos

p3352314

In this project, we explore a method using Singular Value Decomposition (SVD) to classify handwritten digits. SVD is a powerful technique commonly used for dimensionality reduction and feature extraction. By decomposing each class of handwritten digits into its singular vectors, we aim to capture the essential characteristics of the digits and use these representations for classification.

Below you can see some sample images of digits from the dataset to illustrate the diversity and quality of handwritten samples used in our analysis.

Then we want to classify the data in the correct digit class. Here are the steps that we follow to achieve that.

SVD Basis Vectors: We compute the Singular Value Decomposition (SVD) for each digit class and use only the left singular vectors.

Classification Function: We create a function that classifies a given digit by comparing it to different digit classes represented by their SVD basis vectors. For each class, it calculates how well the digit matches by measuring the residual (error) when projected onto a subspace defined by a specified number of basis vectors from that class's SVD. The function identifies the digit class with the smallest residual, indicating the closest match.

Below is the method in more detail.

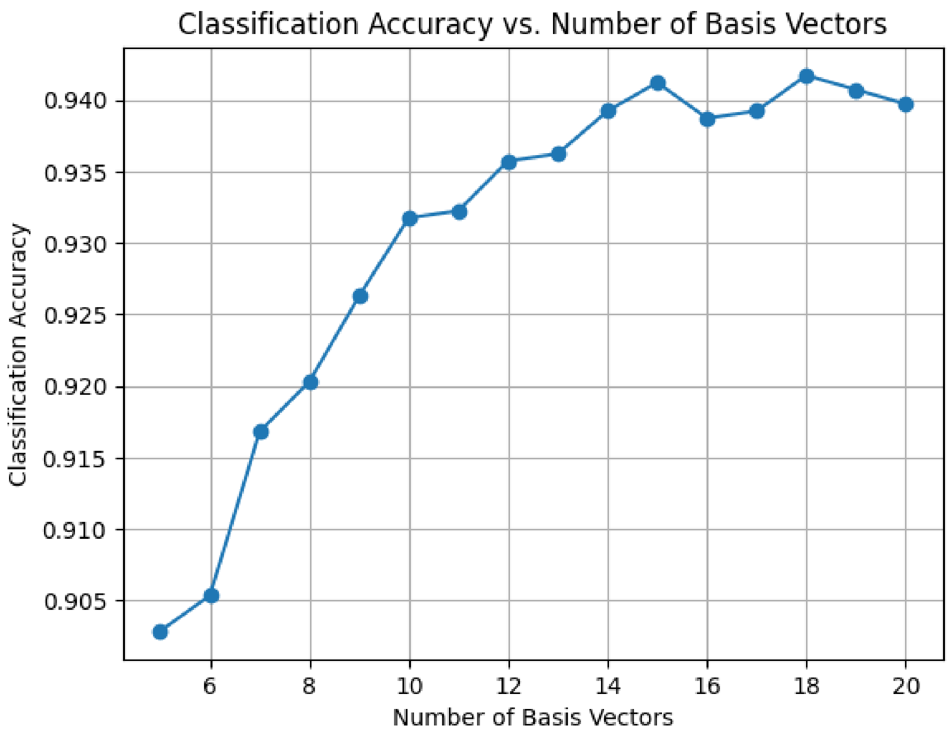
For each digit class represented by its SVD basis vectors $U_k = [u_1, u_2 \dots u_k]$:

- We project the given digit vector z onto the subspace spanned by U_k
- The projection is calculated using the least squares method to minimize $||z - U_k\alpha||_2^2$, where α are the coefficients.
- Since the columns of U_k are orthogonal, the solution α to this problem is given by α= $U_k^T z$
- The norm of the residual vector (error) is $||(I - U_k U_k^T)Z||_2$, which represents the norm of the projection of the unknown digit onto the subspace orthogonal to span

Accuracy Evaluation Function: We develop a function to classify all digits in our dataset using a specified number of basis vectors. This function then assesses the accuracy of the classification by comparing the predicted labels with the actual labels.

These functions allow us to effectively classify handwritten digits using SVD basis vectors and evaluate the accuracy of our classification method.

Afterwards, we tune the number of basis vectors for the digit classification task by evaluating the accuracy of the classifier across a range of values for the number of basis vectors.



From the graph above, 18 is a good number of basis vectors to ensure high accuracy (approximately 95%).

Below is a table with the accuracy per digit

Accuracy for Digit:0 = 98.89%
Accuracy for Digit:1 = 98.11%
Accuracy for Digit:2 = 89.9%
Accuracy for Digit:3 = 90.36%
Accuracy for Digit:4 = 92.5%
Accuracy for Digit:5 = 88.12%
Accuracy for Digit:6 = 95.88%
Accuracy for Digit:7 = 95.92%
Accuracy for Digit:8 = 92.17%
Accuracy for Digit:9 = 93.22%

From these results, the most difficult digit to identify seems to be the number 5 (88,12%) and the easiest one is the zero (98,89%).

Check the singular values of the different classes

We conducted an analysis focusing on the digits 2 and 5, which exhibited the lowest classification accuracy among all classes. By examining the singular values of these digits, we aimed to determine whether reducing the number of basis vectors could enhance classification performance. However, the singular values didn't give a clear indication that fewer vectors would help.

Through a series of experiments, we observed that decreasing the number of basis vectors had varying effects on performance. Specifically, for digit 5, reducing the basis vectors resulted in a noticeable decline in accuracy, indicating that more basis vectors are crucial for accurately classifying this digit. In contrast, for digit 2, there was a slight improvement in performance when using fewer basis vectors. The accuracy increased to 90.4% with 7 basis vectors, compared to the previous 89.9% achieved with 18 basis vectors.

This small change suggests that for some digits, fewer basis vectors can still work well and might even be a bit better, while also reducing the computation needed.

Here are some examples of digits misclassified by the model. Many of these misclassified digit examples exhibit such poor quality that they would challenge human classifiers as well.