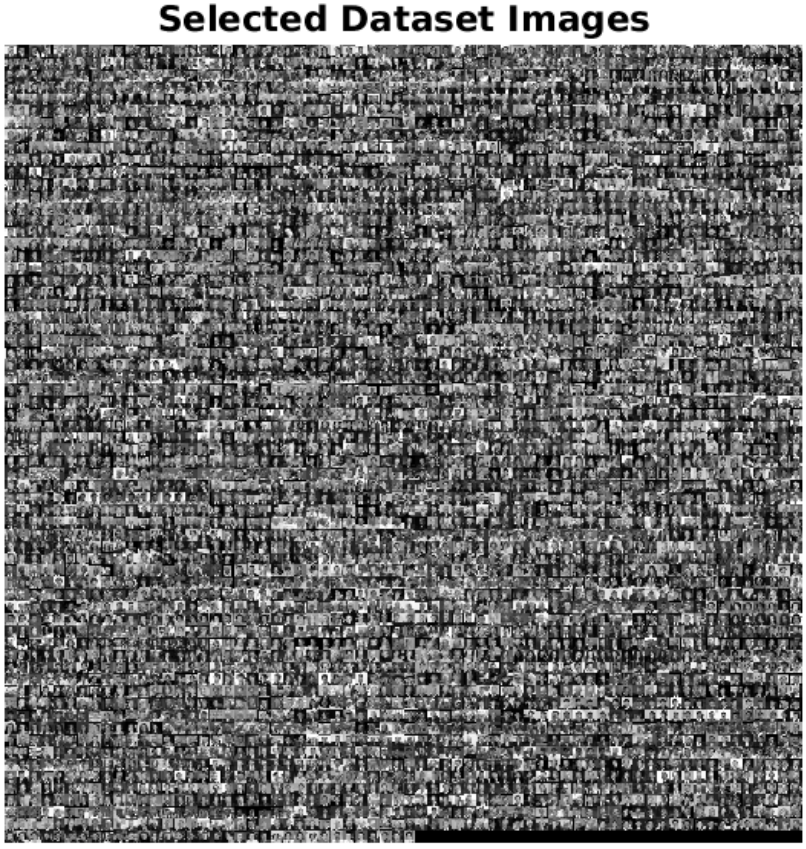
Facial Recognition Using SVD, Eigenfaces, and SVM

With HPC Integration in MATLAB



Joel Maldonado

19-Dec-2024

Table of Contents

1. Introduction and Exploratory Data Analysis (EDA)

Facial recognition systems have become integral in various fields, including security, authentication, and human-computer interaction. The primary challenge in facial recognition lies in dealing with high-dimensional image data. Here, we harness the power of Singular Value Decomposition (SVD) to reduce dimensionality and extract the most discriminative featuresoften referred to as eigenfaces. By integrating MATLABs capabilities with High-Performance Computing (HPC) resources, we aim to scale this process to large datasets efficiently.

In the Exploratory Data Analysis (EDA) phase, we begin by examining the dataset and its characteristics. We standardize image formats, inspect a sample of images, and evaluate data quality. This helps ensure consistent preprocessing and informs the choice of dimensionality reduction and classification techniques.

Figure 1: A sample of the processed dataset images.

2. Singular Value Decomposition and Eigenfaces

$$A = U \Sigma V^T$$

$$A_k = U_k \Sigma_k V_k^T$$

In the context of facial recognition, each image is first transformed into a vectorized form and stacked into a large data matrix A, where each column represents one face image. Applying SVD to A decomposes it into: U (left singular vectors) which represent the directions of maximum variance in the image space, \Sigma (singular values) which quantify the importance of each direction, and V^T (right singular vectors) which provide the coordinates of each image in the new feature space.

By retaining only the top k singular values and their corresponding vectors, we reduce the dimensionality while preserving the most significant information. The columns of U corresponding to the top k singular values are known as eigenfaces. These eigenfaces are essentially the principal components of the face space, capturing features like facial contours, eyes, noses, and mouth positions that best explain the variance in the dataset.

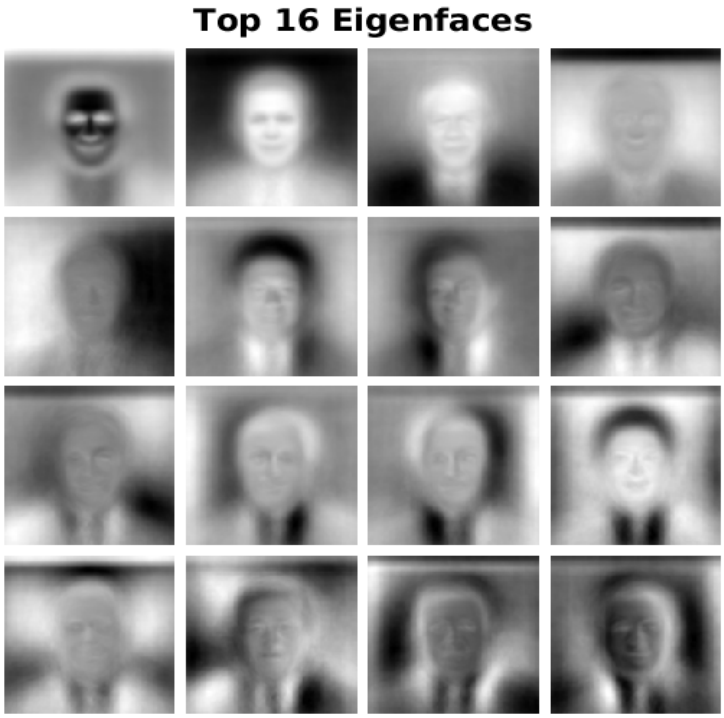


Figure 2: Top 16 extracted eigenfaces using SVD.

3. Feature Extraction and Classification with SVM

Once we have extracted the eigenfaces, each face image can be represented as a linear combination of these eigenfaces. Projecting the original images onto the reduced eigenface space yields a feature vector of significantly smaller dimension, capturing the most discriminative facial features.

These low-dimensional feature vectors form the input to a supervised classifier. In this project, we employ a Support Vector Machine (SVM) classifier. SVMs are chosen for their robustness and effectiveness in high-dimensional spaces, and when combined with a one-vs-one or Error-Correcting Output Code (ECOC) scheme, they can handle multiple classes.

The result is a facial recognition pipeline: Raw images → Preprocessing → SVD (for eigenfaces) → Feature Extraction (projecting onto eigenfaces) → SVM Classification.

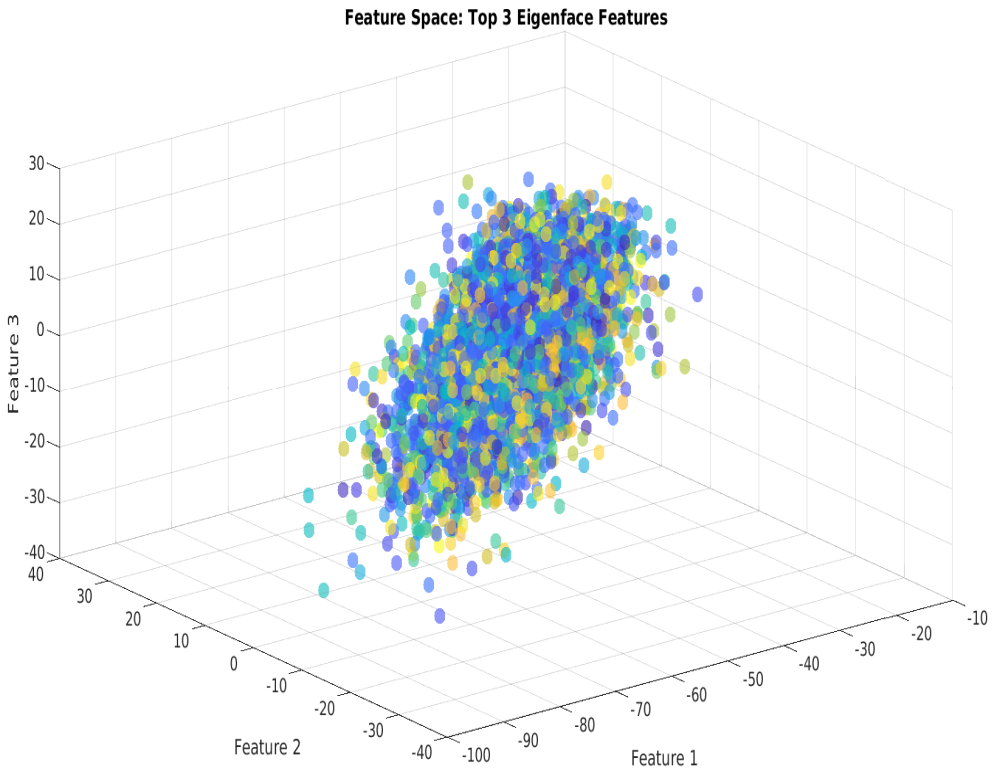


Figure 3: Visualization of the top 3 PCA/SVD-derived features of the dataset.

4. HPC Integration and Scalability

High-Performance Computing (HPC) resources enable scaling the SVD and classification stages to very large datasets. MATLABs Parallel Computing Toolbox allows distributing computations across multiple workers, either on a local multicore machine or on a computing cluster. By utilizing parfor loops, distributed arrays, and parallel-enabled functions (like fitcecoc with parallel options), we significantly reduce computation times.

In this project, the following HPC techniques are employed:1. Parallelized Reading of Images: Using parallel loops to speed up I/O and image preprocessing.2. Parallelized SVD Computation: Although SVD can be computationally expensive, especially on large datasets, MATLAB can offload these computations to multiple cores or to GPU resources if available.3. Parallelized Training of Classifiers: The classification (fitcecoc) can run in parallel, reducing training time significantly.

These HPC approaches ensure that the pipeline can handle increasingly large datasets without incurring prohibitive runtime costs.

5. Model Evaluation

To assess model performance, we compute a confusion matrix and ROC curves. The confusion matrix visualizes how well the classifier distinguishes among different individuals, and the ROC curve shows the trade-off between True Positive and False Positive rates. In practice, evaluating on subsets of classes helps gauge classifier effectiveness and reveals areas needing improvement.

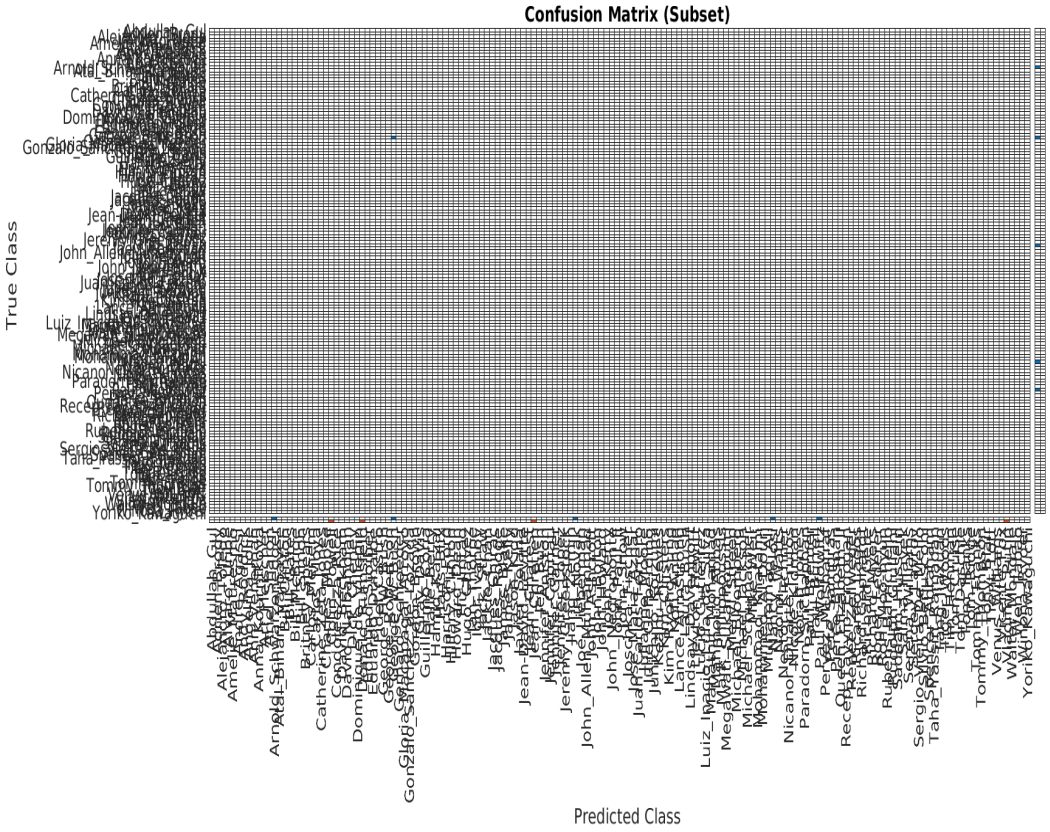


Figure 4: Confusion Matrix for a subset of 5 classes.

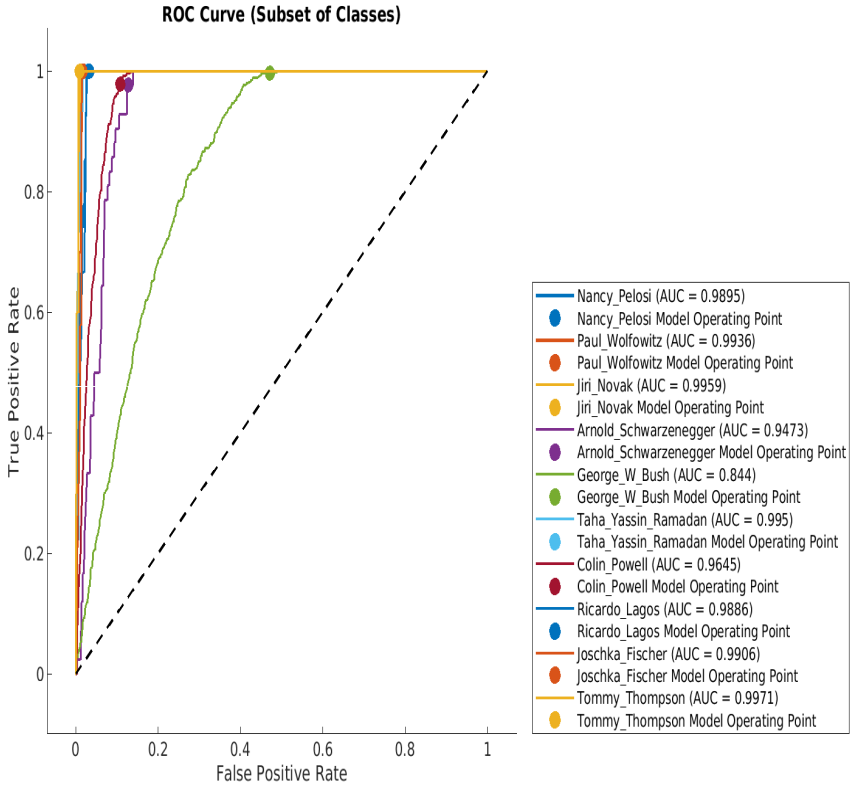


Figure 5: ROC Curve for a subset of 10 classes, illustrating classifier sensitivity and specificity.

6. Conclusion and Future Work

This project demonstrates a scalable facial recognition pipeline that leverages SVD for eigenface extraction and SVM for classification. By integrating MATLABs parallel computing capabilities, we ensure that the approach scales well, handling larger datasets efficiently.

Future work could explore advanced feature extraction methods, such as deep learning-based embeddings (e.g., using CNNs), and incorporate GPU acceleration for even faster computations. These enhancements can further improve both the accuracy and efficiency of the facial recognition system.