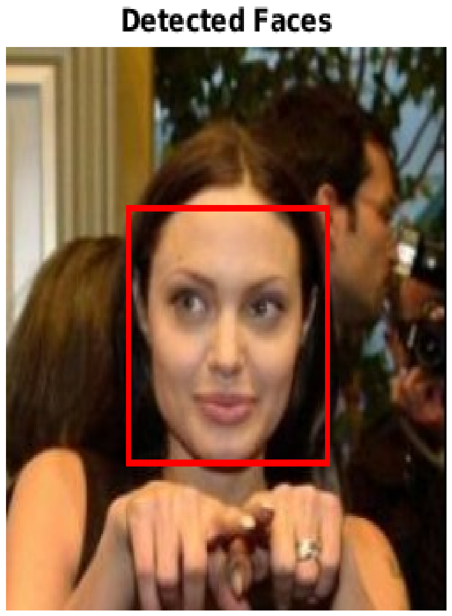
Facial Recognition Using SVD, Eigenfaces, and SVM

With HPC Integration – A MATLAB Generated Report



Joel Maldonado

19-Dec-2024

Table of Contents

1. Data Exploration

The initial phase of this project involves exploring the dataset to understand its structure and distribution. This exploration guides preprocessing decisions and ensures that feature extraction methods capture the key variations in the data. The dataset contains more than 13,000 images of faces collected from the web. Each face is labeled with a person's name, and some individuals have as many as 530 images. Such imbalances can lead to overfitting if not addressed properly.

We begin by examining the distribution of images per person. The histogram below shows the number of images each person has. Most individuals have between 10 and 60 images.

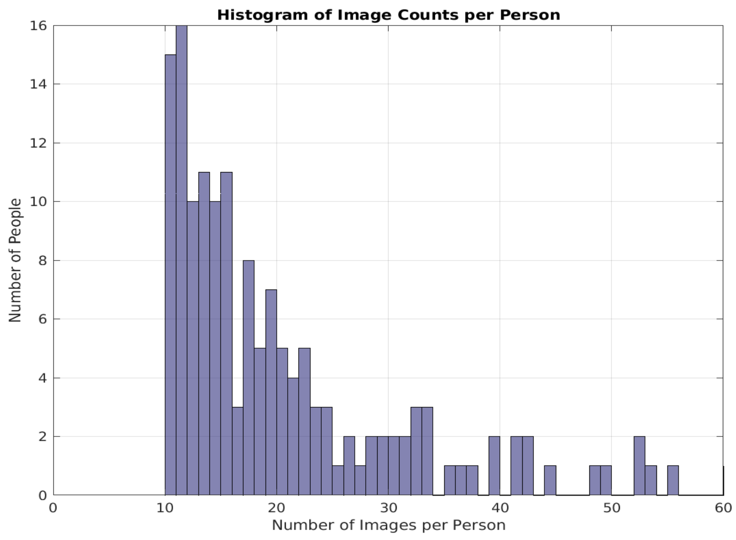


Figure 1: Histogram showing the distribution of image counts per person.

To understand intra-class variability, consider the collage of Angelina Jolie. The images vary in pose, lighting, and expression, which highlights the complexity of face recognition even for a single individual.

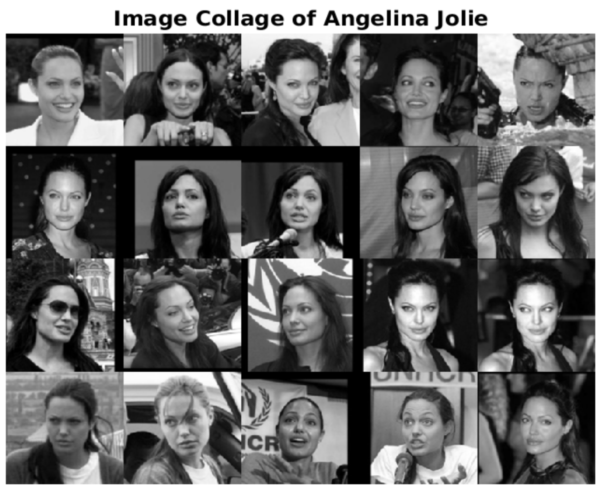


Figure 2: Image collage of Angelina Jolie.

Beyond one person, the dataset contains a diverse range of individuals from various backgrounds. The montage below offers a glimpse into the rich variety of faces, which is crucial for building a robust model.

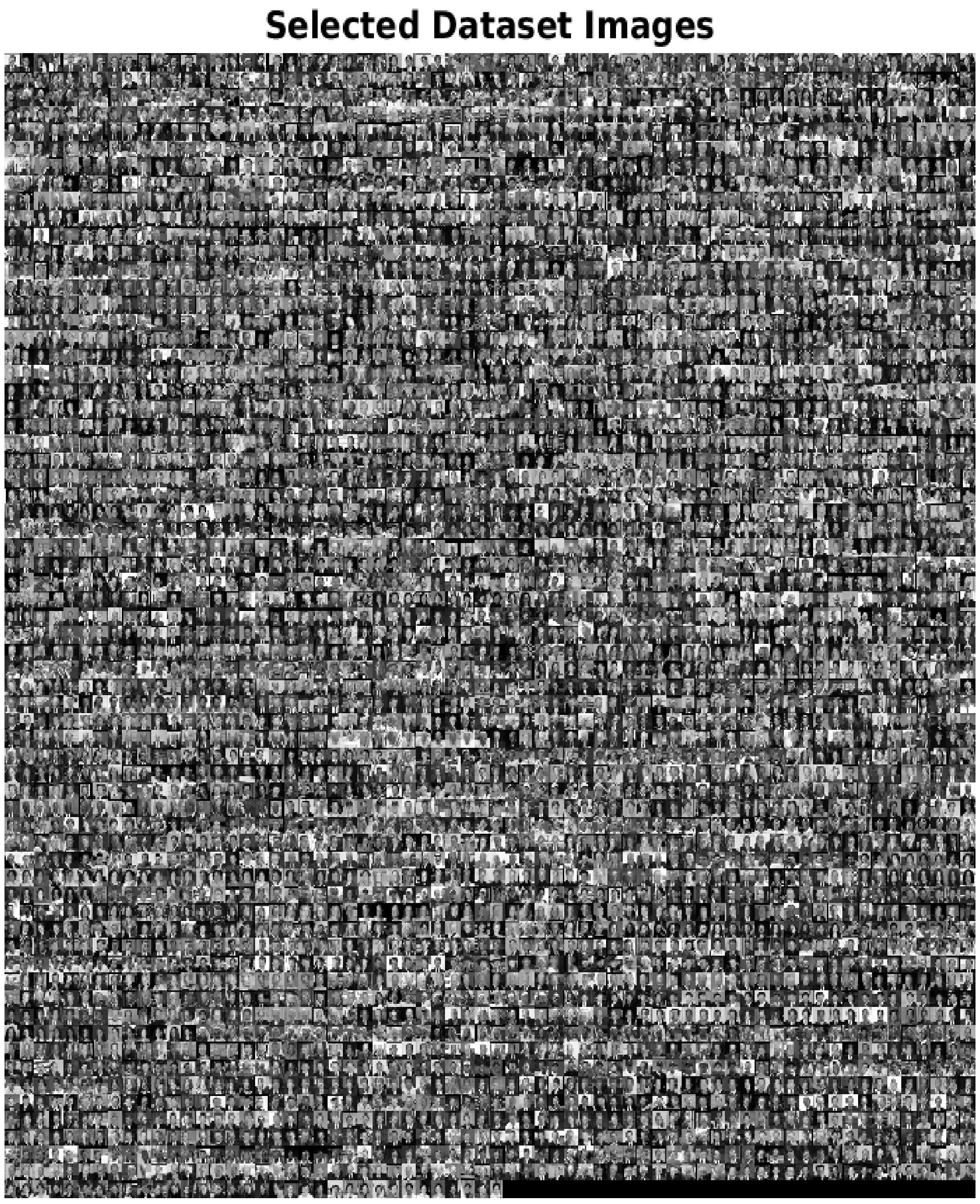


Figure 3: Montage of different individuals, illustrating dataset diversity.

2. Singular Value Decomposition and Eigenfaces

Singular Value Decomposition (SVD) decomposes the data matrix into three components: U, Σ, and V^T. When applied to our collection of face images, SVD reveals directions of maximum variance, known as eigenfaces. These eigenfaces serve as a compact basis for representing facial images.

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

Here, A is the data matrix where each column is a vectorized face. U contains eigenfaces, Σ holds singular values indicating the importance of each eigenface, and V^T gives coordinates of faces in this reduced space. By choosing only the top k eigenfaces, we capture the most critical variations and reduce dimensionality.

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

The resulting eigenfaces highlight essential facial features (eyes, nose, mouth), capturing primary variations. Below are the top 16 eigenfaces extracted using SVD. Each eigenface is a "building block" of any face in the dataset.

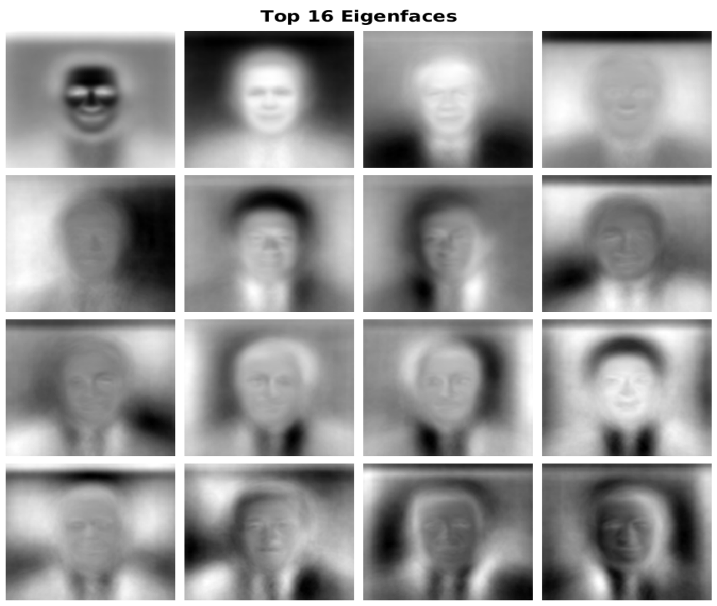


Figure 4: Top 16 extracted eigenfaces.

3. Feature Extraction and Classification with SVM

Projecting each face image onto these eigenfaces yields a feature vector. This lower-dimensional representation is more tractable and retains the most discriminative information. Next, a Support Vector Machine (SVM) classifier is trained on these feature vectors. SVMs are robust and perform well in complex classification tasks.

By using a multi-class strategy (e.g., one-vs-one or ECOC), we extend SVM for multiple person recognition. The pipeline is thus: Raw Image → Preprocessing → SVD (Eigenfaces) → Feature Extraction → SVM Classification.

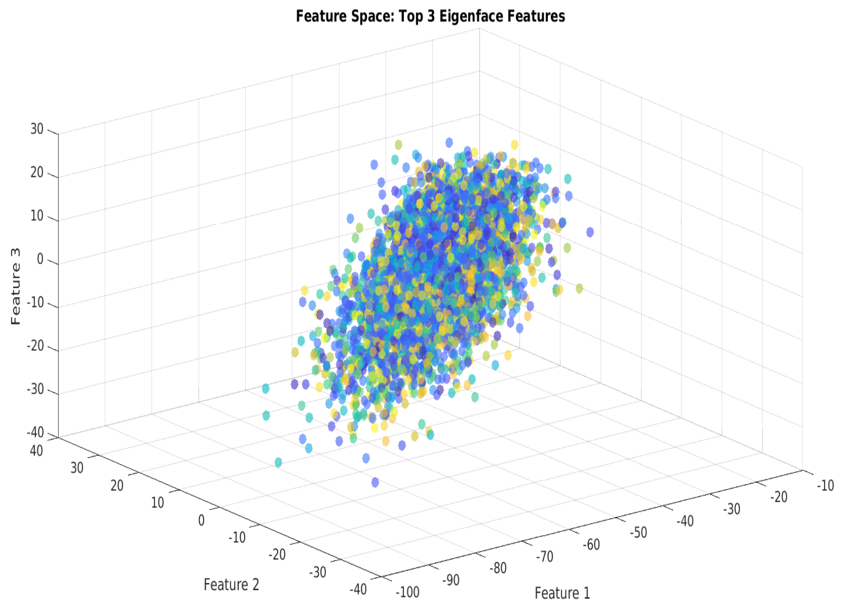


Figure 5: Visualization of the top 3 eigenface-based features.

4. Discussion of Other Approaches and State of the Art

While SVD and SVM form a strong baseline, the field of face recognition has evolved. PCA, LDA, and deep learning approaches provide alternative or complementary strategies.

Principal Component Analysis (PCA) is closely related to SVD and widely used for extracting eigenfaces. LDA optimizes class separability, often improving classification performance.

Deep learning methods (e.g., CNNs like FaceNet or VGG-Face) learn complex, hierarchical features directly from the data. They typically outperform classical methods but require large datasets and high computational power.

5. Results and Metrics of Face Recognition

We evaluate the trained model using confusion matrices and ROC curves. The confusion matrix shows correct and misclassified faces. ROC curves illustrate performance trade-offs in terms of true and false positive rates.

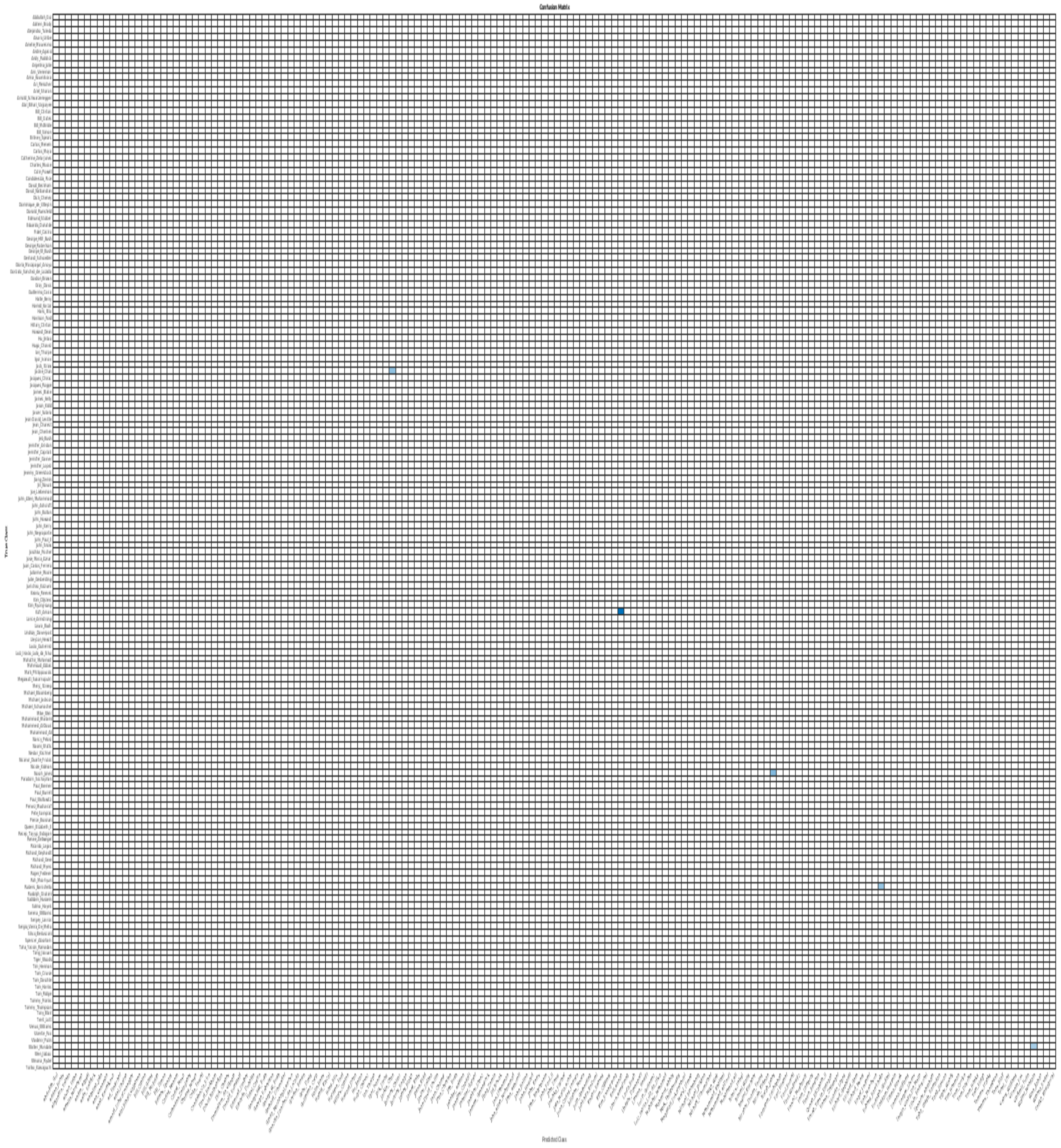


Figure 6: Confusion Matrix for a subset of classes.

Below is an ROC curve for selected classes. Higher AUC values indicate better discrimination between individuals.

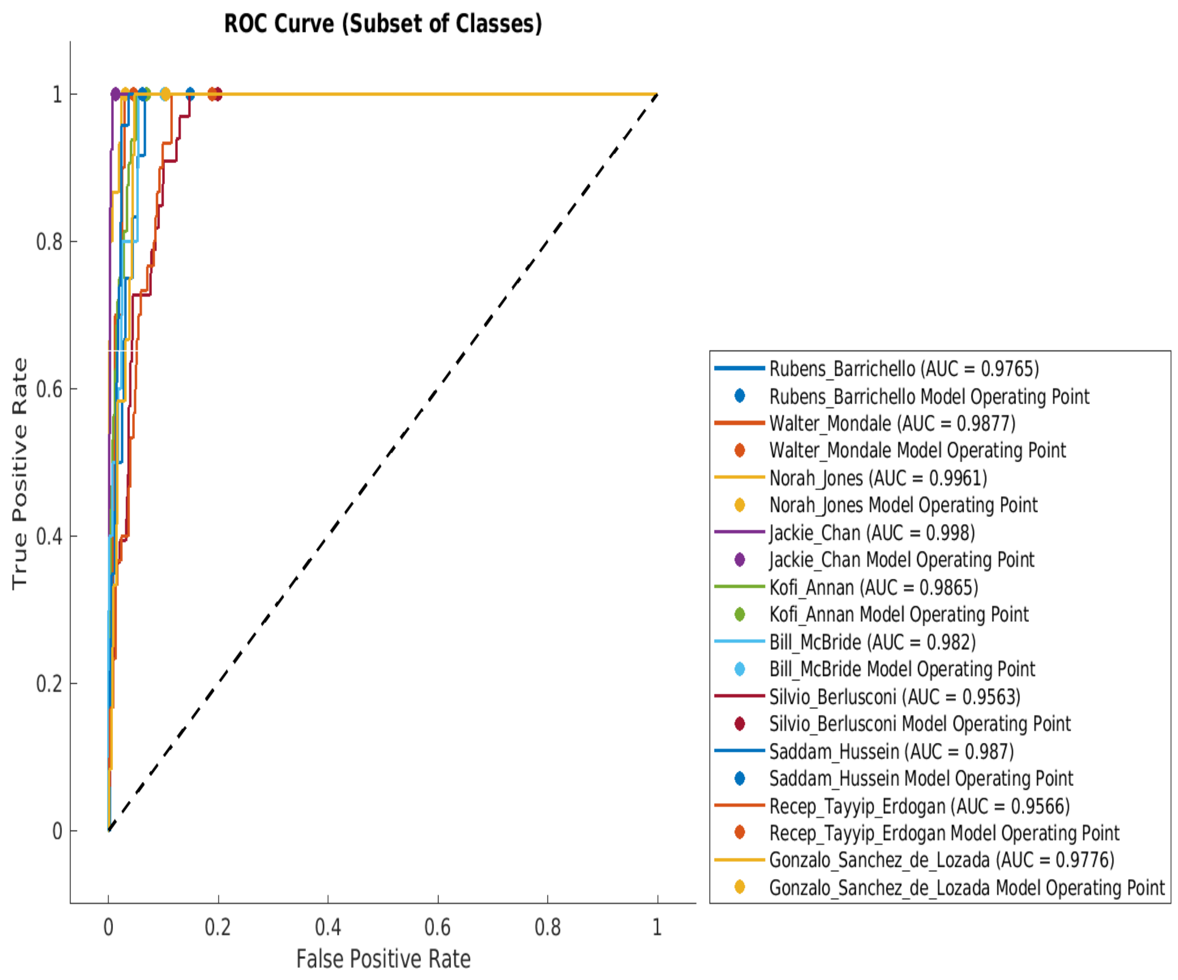


Figure 7: ROC Curve for selected classes.

6. Conclusion and Future Work

This project demonstrates a scalable facial recognition pipeline using SVD (for eigenface extraction) and SVM for classification. Parallelization techniques accelerate data handling and model training, making the approach suitable for larger datasets.

Future work may focus on advanced segmentation before applying SVD, such as using MATLAB's 2024b "imsegsam" function, or alternative methods like bounding boxes and skin color masks to isolate facial features. This preprocessing could improve recognition accuracy.

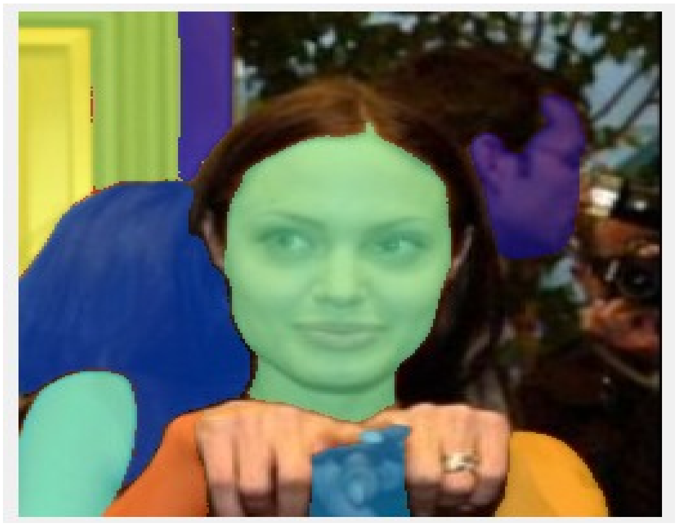


Figure 8: Attempted segmentation using "imsegsam" (MATLAB 2024b).

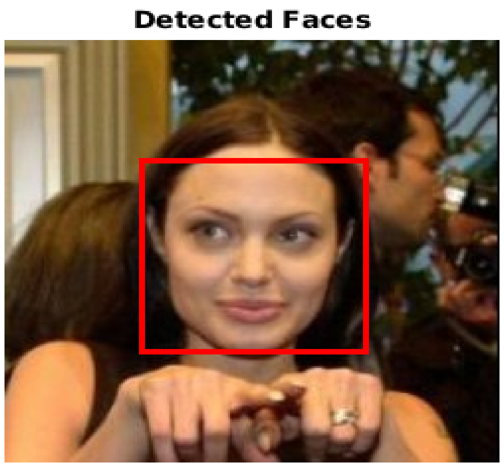


Figure 9: Bounding box around a detected face.

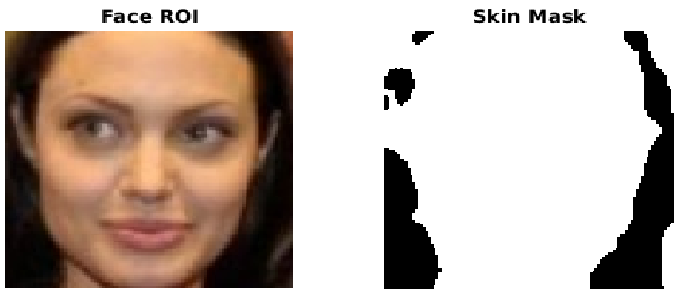


Figure 10: Applying a skin-color mask to segment the face region.

Expanding the dataset, incorporating more robust segmentation, and exploring deep learning models are all potential avenues for further research. Such enhancements promise to improve both the accuracy and scalability of future facial recognition systems.

7. HPC Integration and Scalability

High-Performance Computing (HPC) resources streamline the pipeline. MATLAB’s Parallel Computing Toolbox can parallelize image preprocessing, SVD computations, and SVM training. This reduces training time significantly and supports scaling to even larger datasets.

HPC integration ensures that methods like SVD and classification can run efficiently. Leveraging multiple cores or a cluster environment can turn a time-consuming process into a more manageable and repeatable computation.