Face recognition with HOSVD

An application of HOSVD and T-HOSVD

Sonia Spinelli

University of Trento

December 10, 2024

Table of contents

The Classification Problem

Tensor Representation and HOSVD

Face Recognition Algorithm

Description of the database

Results with HOSVD

Results with T-HOSVD

The Classification Problem

Face recognition is the process of identifying a person from a digital image or a video frame.

Our Purpose

The objective of this work is to distinguish between different individuals whose photos are part of the dataset, using the data structure of a tensor.

Sample images from the dataset.

The Classification Problem

We will decompose the dataset into three dimensions:

- **The image:** capturing the raw pixel information.
- **The expression:** representing variations in illumination, viewing angles, and other factors in photos of the same individual.
- **The person:** identifying the unique individual.

Our approach

We utilize the *High-Order Singular Value Decomposition (HOSVD)*. This method enables us to extract the key components in a structured, multidimensional framework, facilitating precise classification.

Tensor Representation and HOSVD

- A dataset contains images of n_p persons photographed in n_e different expressions.
- Each image contains a face and is represented as a matrix $n_1 \times n_2$.
- We can assume that the columns of the images are stacked, so that each image is represented by a vector in $\mathbb{R}^{n_i}$, with $n_i = n_1 n_2$ being the number of pixels. Typically n_i is considerably larger than n_e and n_p.

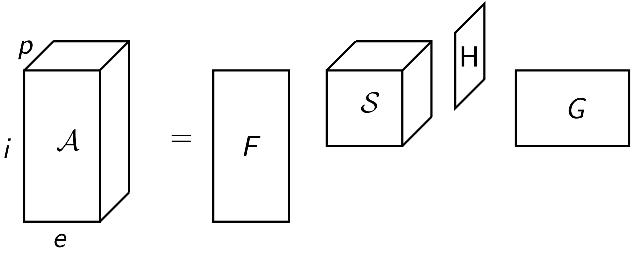
We can store the dataset as a tensor $\mathcal{A} \in \mathbb{R}^{n_i \times n_e \times n_p}$.
We refer to the different modes as the image mode (i), the expression mode (e) and the person mode (p), respectively.

Tensor Representation and HOSVD

High-Order Singular Value Decomposition

$$\mathcal{A} = \mathcal{S} \times_i F \times_e G \times_p H$$

- $\mathcal{S} \in \mathbb{R}^{n_e n_p \times n_e \times n_p}$ is the *core tensor*.
- $F \in \mathbb{R}^{n_i \times n_e n_p}$, $G \in \mathbb{R}^{n_e \times n_e}$ and $H \in \mathbb{R}^{n_p \times n_p}$ are orthogonal matrices, called *factor matrices*.



Tensor Representation and HOSVD

Algorithm 1 HOSVD

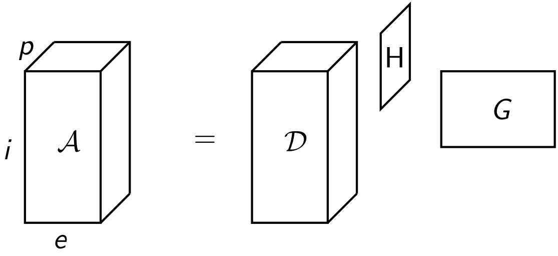
```
Input: tensor 𝒜 with K modes.
Output: core tensor 𝒮, factor matrices U_1, ..., U_K.
for k = 1, 2, ..., K do
    Compute the thin SVD of the unfolding 𝒜_(k) = U_k Σ_k V_k^⊤
end for
Initialize the core tensor 𝒮 ← 𝒜
for k = 1, 2, ..., K do
    Compute the core tensor by projecting 𝒜 onto U_k:
    𝒮 ← 𝒮 ×_k U_k^⊤ (tensor contraction along mode k)
end for
```

Tensor Representation and HOSVD

The HOSVD can be interpreted in different ways depending on what it is to be used for.
For instance, we can use the following interpretation

$$\mathcal{A} = \mathcal{D} \times_e G \times_p H$$

where $\mathcal{D} = \mathcal{S} \times_i F$.

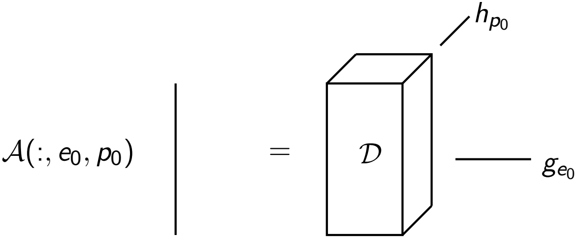


Tensor Representation and HOSVD

The image of person p_0 in expression e_0 can be synthesized by multiplication of the tensor $\mathcal{D}$ by h_{p_0} and g_{e_0} in their respective modes:

$$\mathcal{A}(:, e_0, p_0) = \mathcal{D} \times_e g_{e_0} \times_p h_{p_0}$$

where $g_{e_0} = G(e_0, :)$ and $h_{p_0} = H(p_0, :)$.



Face Recognition Algorithm

Our objective

Given an image of an unknown person, represented by a vector in $\mathbb{R}^{n_i}$, determine which of the n_p persons it represents.

For our classification problem we write the HOSVD in the following form:

$$\mathcal{A} = \mathcal{C} \times_p H, \quad \text{where } \mathcal{C} = \mathcal{S} \times_i F \times_e G.$$

For a particular expression e we have

$$\mathcal{A}(:, e, :) = \mathcal{C}(:, e, :) \times_p H.$$

We can identify the tensors $\mathcal{A}(:, e, :)$ and $\mathcal{C}(:, e, :)$ with the matrices A_e and C_e. So, for all the expressions, we have

$$A_e = C_e H^\top \text{ with } e = 1, \dots, n_e.$$

Face Recognition Algorithm

The p-th column of A_e is

$$a_p^{(e)} = C_e h_p^\top$$

which contains the image of person p and expression e.

The columns of C_e are basis vectors for expression e, the p-th row of H holds the coordinates of the image of person p in all expression bases.

Face Recognition Algorithm

The classification algorithm
Let us assume that $z \in \mathbb{R}^{n_i}$ is an image of an unknown person in an unknown expression and that we want to classify it. In this setup, z is called a *test image*.

Idea: We would like to find the most similar $a_p^{(e)}$ and classify the image z as a photo of the person p. So we want to minimize

$$\|a_p^{(e)} - z\|_2 = \|C_e h_p^\top - z\|_2.$$

We need to find the coordinates α_e of the test image z in expression basis e. These can be obtained by solving the least square problem

$$\min_{\alpha_e} ||C_e \alpha_e - z||_2.$$

Face Recognition Algorithm

Once we have α_e, we can compute the distance between α_e and the coordinates of the person p in all expression bases h_p.

$$d(e,p) = \|\alpha_e - h_p^\top\|_2$$

For each person, we calculate the minimum distance among all the distances associated with the different expressions.
Then, we select the person with the overall minimum distance among all the people.

Face Recognition using HOSVD

The algorithm is the following:

Algorithm 2 Classification Algorithm (preliminary version)

Input: Test image z.
for $e = 1, 2, \ldots, n_e$ **do**
    Solve $\min_{\alpha_e} \|C_e \alpha_e - z\|_2$
    for $p = 1, 2, \ldots, n_p$ **do**
        Compute distance $d_{e,p} = \|\alpha_e - h_p^\top\|_2$
    end for
end for
Compute $d_p = \min d_{e,p}$
Classify as the person $\hat{p} = \arg\min d_p$.

Face Recognition Algorithm

For each test image z we must solve n_e least squares problems with $C_e \in \mathbb{R}^{n_i \times n_p}$.
Since we have assumed that $n_i >> n_e n_p$, we can use the following equation

$$C_e = FB_e \text{ with } F \in \mathbb{R}^{n_i \times n_e n_p}, B_e \in \mathbb{R}^{n_e n_p \times n_p}$$

where B_e is the matrix obtained as

$$B_e = (\mathcal{S} \times_e G)(:, e, :) = F^\top C_e.$$

Then, we can enlarge the matrix F so that it becomes square and orthogonal:

$$\hat{F} = FF^\perp \quad \hat{F}^\top \hat{F} = I.$$

Face Recognition Algorithm

Since $\hat{F}$ is orthogonal we can insert it inside the norm and it won't change the euclidean norm:

$$
\begin{aligned}
\|C_e\alpha_e - z\|_2^2 &= \left\|\hat{F}^\top(C_e\alpha_e - z)\right\|_2^2 \\
&= \left\|\begin{bmatrix} F^\top \\ (F^\perp)^\top \end{bmatrix}(C_e\alpha_e - z)\right\|_2^2 \\
&= \left\|F^\top(C_e\alpha_e - z)\right\|_2^2 + \left\|(F^\perp)^\top(C_e\alpha_e - z)\right\|_2^2 \\
&= \left\|B_e\alpha_e - F^\top z\right\|_2^2 + \left\|(F^\perp)^\top z\right\|_2^2.
\end{aligned}
$$

Face Recognition Algorithm

Solving the least squares problem is equivalent to computing

$$\min_{\alpha_e} \|B_e \alpha_e - F^\top z\|_2 \quad e = 1, \ldots, n_e.$$

Since $B_e \in \mathbb{R}^{n_e n_p \times n_p}$, while $C_e \in \mathbb{R}^{n_i \times n_p}$, the computation is lighter.
Furthermore, it is possible to compute the thin QR decomposition of each matrix B_e to further reduce the work.

$$B_e = Q_e R_e, \text{ for } e = 1, 2, \ldots, n_e$$

with an orthonormal matrix Q_e and an upper triangular matrix R_e.

Face Recognition Algorithm

The final algorithm is the following:

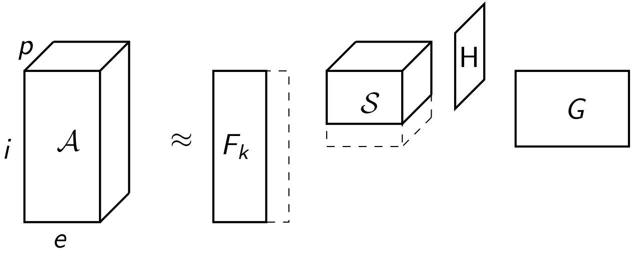
Algorithm 3 Classification Algorithm

Input: Test image $\mathbf{z}$.
Compute $\hat{z} = F^{\top} z$.
for $e = 1, 2, \ldots, n_e$ **do**
    Compute and save the thin QR decompositions of B_e.
    Solve $R_e \alpha_e = Q_e^{\top} \hat{z}$ for α_e.
    for $p = 1, 2, \ldots, n_p$ **do**
      Compute distance $d_{e,p} = \|\alpha_e - h_p^{\top}\|_2$.
    end for
end for
Compute $d_p = \min d_{e,p}$
Classify as the person $\hat{p} = \arg\min d_p$.

Face Recognition using T-HOSVD

Can we use an approximation of these images?

We can approximate the images by reducing the number of pixels and use the T-HOSVD with a truncation in the first mode, the image mode.



Face Recognition using T-HOSVD

Define $F_k = F(:, 1:k)$ for some value of k that we assume is much smaller than n_i. Then the enlarged orthogonal matrix becomes

$$\widehat{F} = F_k \tilde{F}^{\perp} \quad \widehat{F}^{\top}\widehat{F} = I.$$

We truncate the tensor similarly:

$$\widehat{\mathcal{C}} = (\mathcal{S} \times_e G)(1:k, :, :) \times_i F_k.$$

We have the following relation:

$$\|\widehat{\mathcal{C}} - \mathcal{C}\|_F^2 = \sum_{\nu=k+1}^{n_i} \sigma_\nu^{(i)}.$$

Face Recognition using T-HOSVD

By applying the same computations as before but using $\widehat{\mathcal{C}}$, we have

- $\widehat{C}_e = \widehat{\mathcal{C}}(:, e, :) = F_k \widehat{B}_e$
- $\widehat{B}_e = F_k^\top \widehat{C}_e \in \mathbb{R}^{k \times n_p}$

and the least square problem has become

$$\min_{\alpha_e} \| \widehat{B}_e \alpha_e - F_k^\top z \|_2 \quad e = 1, \ldots, n_e$$

where the matrix F has been substituted with the new F_k and B_e with $\widehat{B}_e$.

Description of the database

The ORL database

- Contains 400 grayscale images in PGM format of 40 persons.
- Each image is represented as a 64×64 matrix pixels.
- Each person has been photographed in 10 expressions.

Data are normalized. We artificially split the dataset into training and testing set, allocating 9 photos per individual for training the algorithm and 1 photo per individual for testing.

So we end up with the following parameters

- $n_p = 40$ people,
- $n_e = 9$ expressions,
- $n_i = 64 \times 64 = 4096$ pixels.

Description of the database

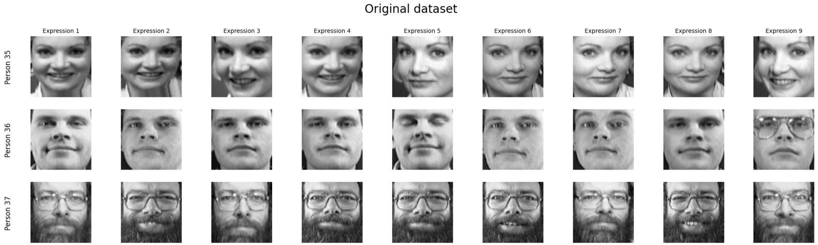


Figure: Some photos of the training dataset

HOSVD

We compute the HOSVD as

$$\mathcal{A} = \mathcal{C} \times_p H, \quad \text{where } \mathcal{C} = \mathcal{S} \times_i F \times_e G.$$

$\mathcal{C}$ has dimension $(n_e n_p, n_e, n_p) = (360, 9, 40)$.

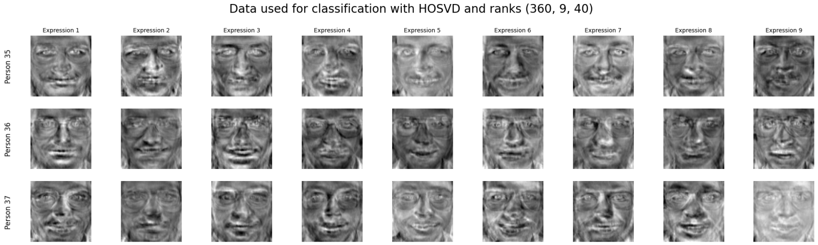


Figure: Images reconstructed without the identity mode

HOSVD

Now, we can distinguish two scenarios:

1. **Classification on the Training Set:** Given an image from the training dataset, classify it by comparing it to the learned representations.
   - Useful to evaluate how well the T-HOSVD model has learned the dataset.
   - High accuracy here demonstrates the model's ability to capture the structure of the training data.
2. **Classification on the Testing Set:** Given a new, unseen image, classify it using the trained model.
   - Reflects the model's generalization capability in real-world applications.
   - Lower accuracy may indicate overfitting or insufficient generalization of the model.

HOSVD

Classification on the Training Set:

- Number of misclassified images: 0
- Accuracy: 100.00%

Classification on the Testing Set:

- Number of misclassified images: 3
- Accuracy: 92.50%

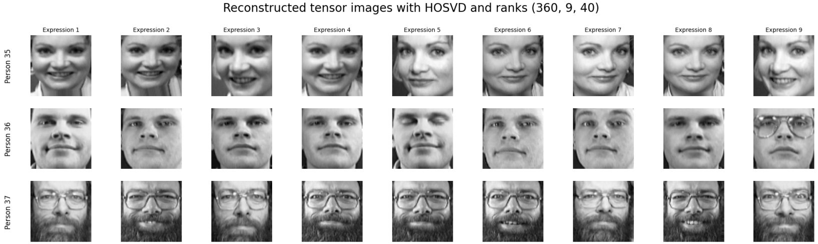


Figure: Images fully reconstructed

T-HOSVD

Let's try to decrease the first-mode rank in order to reduce the image complexity and perform a lower rank approximation. We start with $k = 10$:

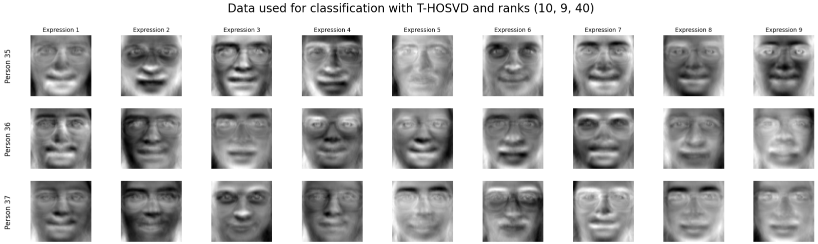


Figure: Images reconstructed without the identity mode.

T-HOSVD

Classification on the Training Set:

- Number of misclassified images: 91
- Accuracy: 74.72%

Classification on the Testing Set:

- Number of misclassified images: 16
- Accuracy: 60.00%.

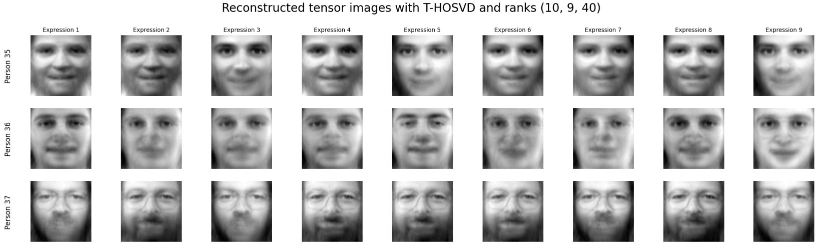
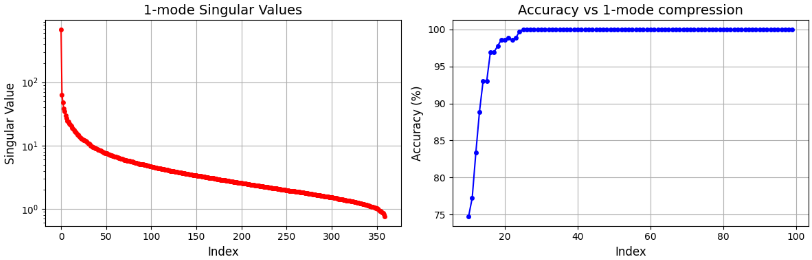


Figure: Images fully reconstructed.

T-HOSVD

Classification on the Training Set:
The 1-mode singular values reveals a rapid initial decrease, followed by a more gradual decline. This indicates that the first singular values capture the majority of the information in the data.



When classifying on the training set the accuracy reaches 100% after k=25, meaning that, in order to distinguish the people in the database, it is sufficient a low k.

T-HOSVD

By fixing k as 25:

- Number of misclassified images: 0
- Accuracy: 100.00%

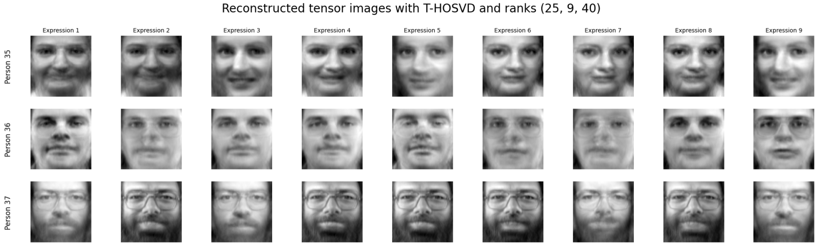
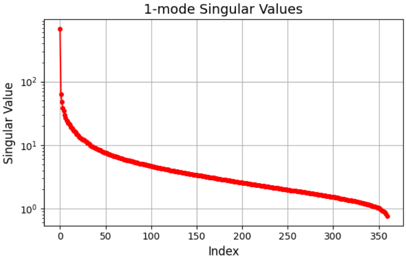


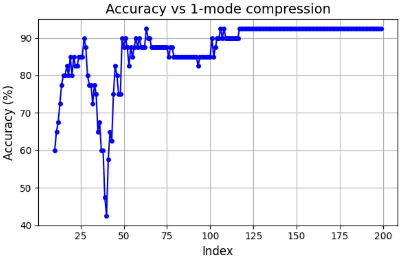
Figure: Images fully reconstructed.

T-HOSVD

Classification on the Testing Set:

On the other hand, when we are classifying new images, the accuracy stabilizes after index 50, indicating that prior to this point, the classification results appear not to have consistent correctness.





This shows that, in order to have a generalizable model, we have to increase k.

T-HOSVD

By fixing k as 50:

- Number of misclassified images: 5
- Accuracy: 87.50%

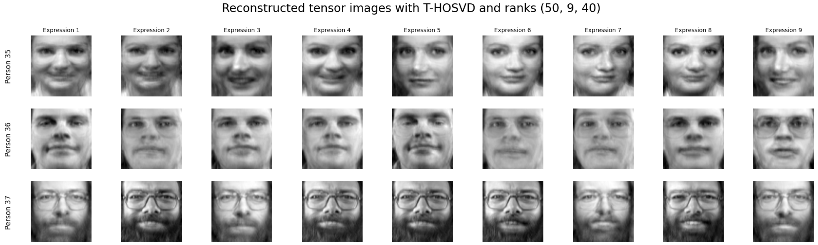


Figure: Images fully reconstructed.

Conclusion

- The problem of face recognition can be effectively tackled using a tensor-based approach.
- By applying the T-HOSVD algorithm, we can significantly reduce the complexity of the image representation while retaining essential features.

- This method achieves reliable performance when classifying images from the training set.
- Additionally, it can be extended to classify unseen data, though this may require adjusting the compression level to balance accuracy and computational efficiency.

Thank you
for your attention!

Code and references

Code:

Github repo: Click here for the code
Colab repo: Click here for the code

References:

- Brandoni, D. (2018). Tensor Decompositions for Face Recognition (Master's thesis). Alma Mater Studiorum University of Bologna, Master's Degree in Mathematics. Academic Year 2017/2018. Advisor: Prof. Valeria Simoncini.
- Eldén, L. (2007). Matrix Methods in Data Mining and Pattern Recognition. Society for Industrial and Applied Mathematics (SIAM), Linköping University, Linköping, Sweden.