Computer Homework 5

Matrix and Tensor Decompositions

Dr. Sepideh Hajipour

Sharif University of Technology

Electrical Engineering

Nikoo Moradi

400101934

Date: July 2, 2024

Contents

1

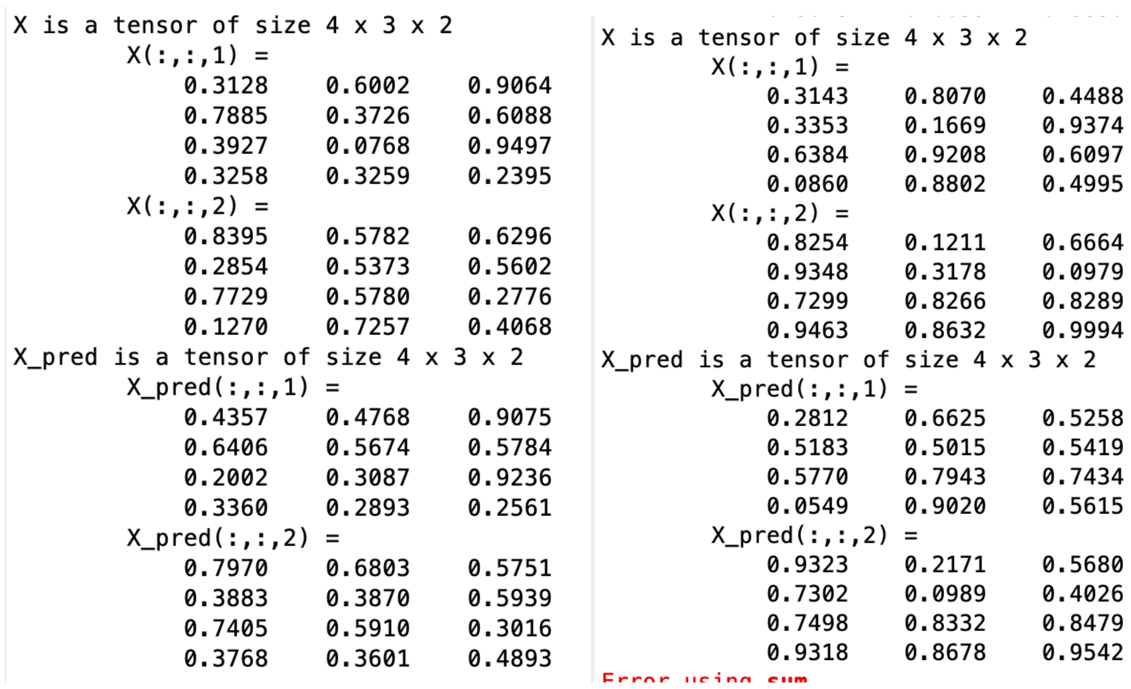
Here is the pseudo code of my costume HOOI algorithm for Tucker decomposition:

Algorithm 1 Higher-Order Orthogonal Iteration (HOOI)

```
1: Initialize factor matrices $A_1, A_2, A_3$ using HOSVD
2: for $i = 1$ to 3 do
3:     $temp \leftarrow$ mode-$i$ unfolding of $X$
4:     $[U, \sim, \sim] \leftarrow \mathrm{svd}(temp)$
5:     $A_i \leftarrow U(:, 1 : R_i)$
6: end for
7: for $iter = 1$ to $max\_iterations$ do
8:     $A_{\mathrm{old}} \leftarrow A$
9:     $Z1 \leftarrow ttm(X, \{A_2^T, A_3^T\}, [2, 3])$
10:    $temp \leftarrow$ mode-1 unfolding of $Z1$
11:    $[U, \sim, \sim] \leftarrow \mathrm{svd}(temp)$
12:    $A_1 \leftarrow U(:, 1 : R_1)$
13:    $Z2 \leftarrow ttm(X, \{A_1^T, A_3^T\}, [1, 3])$
14:    $temp \leftarrow$ mode-2 unfolding of $Z2$
15:    $[U, \sim, \sim] \leftarrow \mathrm{svd}(temp)$
16:    $A_2 \leftarrow U(:, 1 : R_2)$
17:    $Z3 \leftarrow ttm(X, \{A_1^T, A_2^T\}, [1, 2])$
18:    $temp \leftarrow$ mode-3 unfolding of $Z3$
19:    $[U, \sim, \sim] \leftarrow \mathrm{svd}(temp)$
20:    $A_3 \leftarrow U(:, 1 : R_3)$
21:    Check convergence:
22:    if $\max\left(\frac{\|A_1 - A_{\mathrm{old}_1}\|_F}{\|A_{\mathrm{old}_1}\|_F}, \frac{\|A_2 - A_{\mathrm{old}_2}\|_F}{\|A_{\mathrm{old}_2}\|_F}, \frac{\|A_3 - A_{\mathrm{old}_3}\|_F}{\|A_{\mathrm{old}_3}\|_F}\right) < tol$ then
23:        break
24:    end if
25: end for
26: $U1 \leftarrow A_1$
27: $U2 \leftarrow A_2$
28: $U3 \leftarrow A_3$
29: $G \leftarrow ttm(X, \{A_1^T, A_2^T, A_3^T\}, [1, 2, 3])$
30: return $G, U1, U2, U3$
```

The HOOI function was implemented as described. To verify its correctness, we generated an arbitrary tensor and computed its factor matrices and core tensor. Using these, we reconstructed the tensor and compared it with the original. The results are presented, including two examples on the next page.

While ALS-based algorithms do not guarantee convergence, this algorithm successfully performs the Tucker decomposition.



```
X is a tensor of size 4 x 3 x 2
        X(:,:,1) =
            0.3128    0.6002    0.9064
            0.7885    0.3726    0.6088
            0.3927    0.0768    0.9497
            0.3258    0.3259    0.2395
        X(:,:,2) =
            0.8395    0.5782    0.6296
            0.2854    0.5373    0.5602
            0.7729    0.5780    0.2776
            0.1270    0.7257    0.4068
X_pred is a tensor of size 4 x 3 x 2
        X_pred(:,:,1) =
            0.4357    0.4768    0.9075
            0.6406    0.5674    0.5784
            0.2002    0.3087    0.9236
            0.3360    0.2893    0.2561
        X_pred(:,:,2) =
            0.7970    0.6803    0.5751
            0.3883    0.3870    0.5939
            0.7405    0.5910    0.3016
            0.3768    0.3601    0.4893
```

```
X is a tensor of size 4 x 3 x 2
        X(:,:,1) =
            0.3143    0.8070    0.4488
            0.3353    0.1669    0.9374
            0.6384    0.9208    0.6097
            0.0860    0.8802    0.4995
        X(:,:,2) =
            0.8254    0.1211    0.6664
            0.9348    0.3178    0.0979
            0.7299    0.8266    0.8289
            0.9463    0.8632    0.9994
X_pred is a tensor of size 4 x 3 x 2
        X_pred(:,:,1) =
            0.2812    0.6625    0.5258
            0.5183    0.5015    0.5419
            0.5770    0.7943    0.7434
            0.0549    0.9020    0.5615
        X_pred(:,:,2) =
            0.9323    0.2171    0.5680
            0.7302    0.0989    0.4026
            0.7498    0.8332    0.8479
            0.9318    0.8678    0.9542
Error using sum
```

Figure 1: Example use of costume HOOI function

2

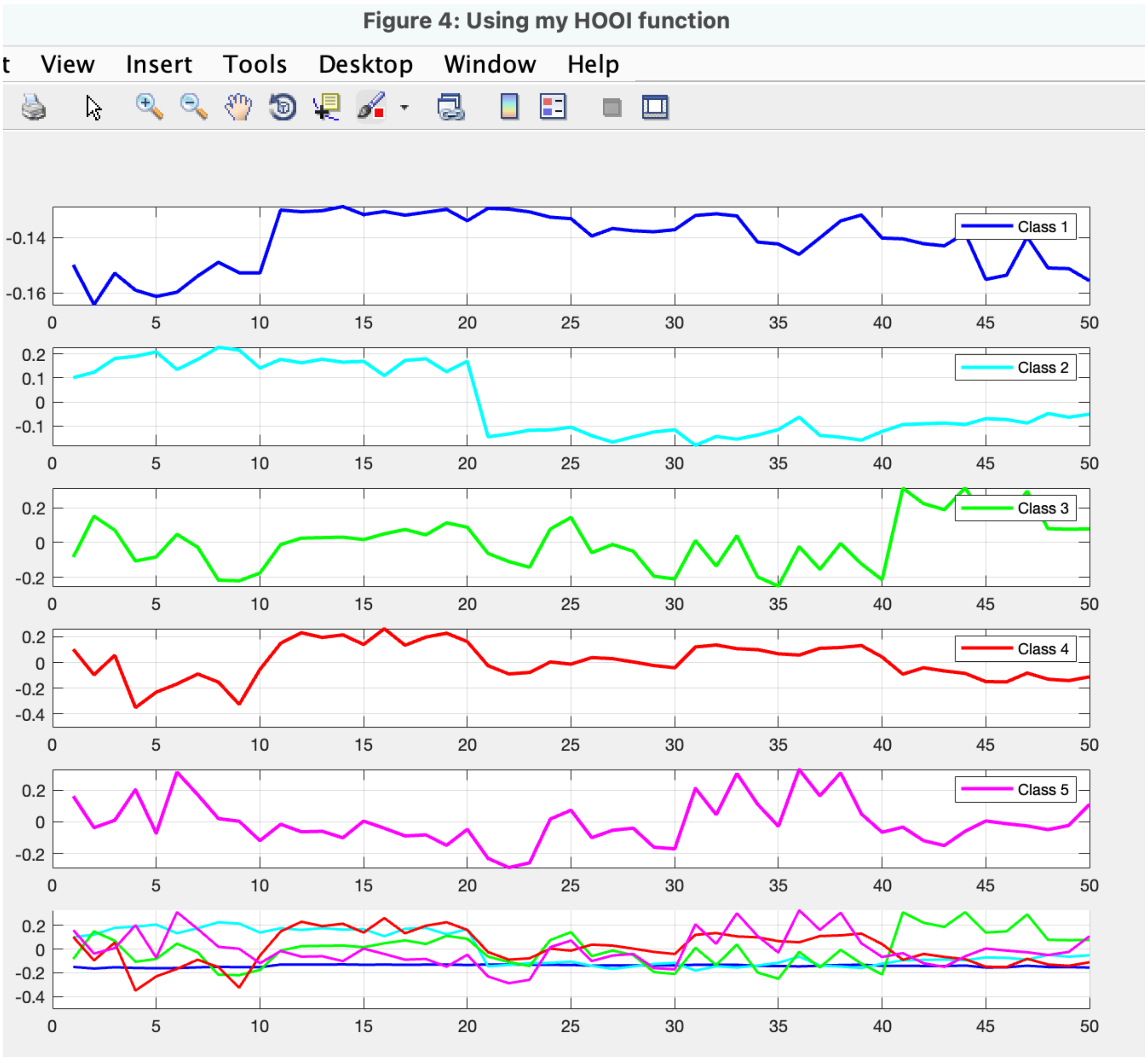


Figure 2: costume HOOI

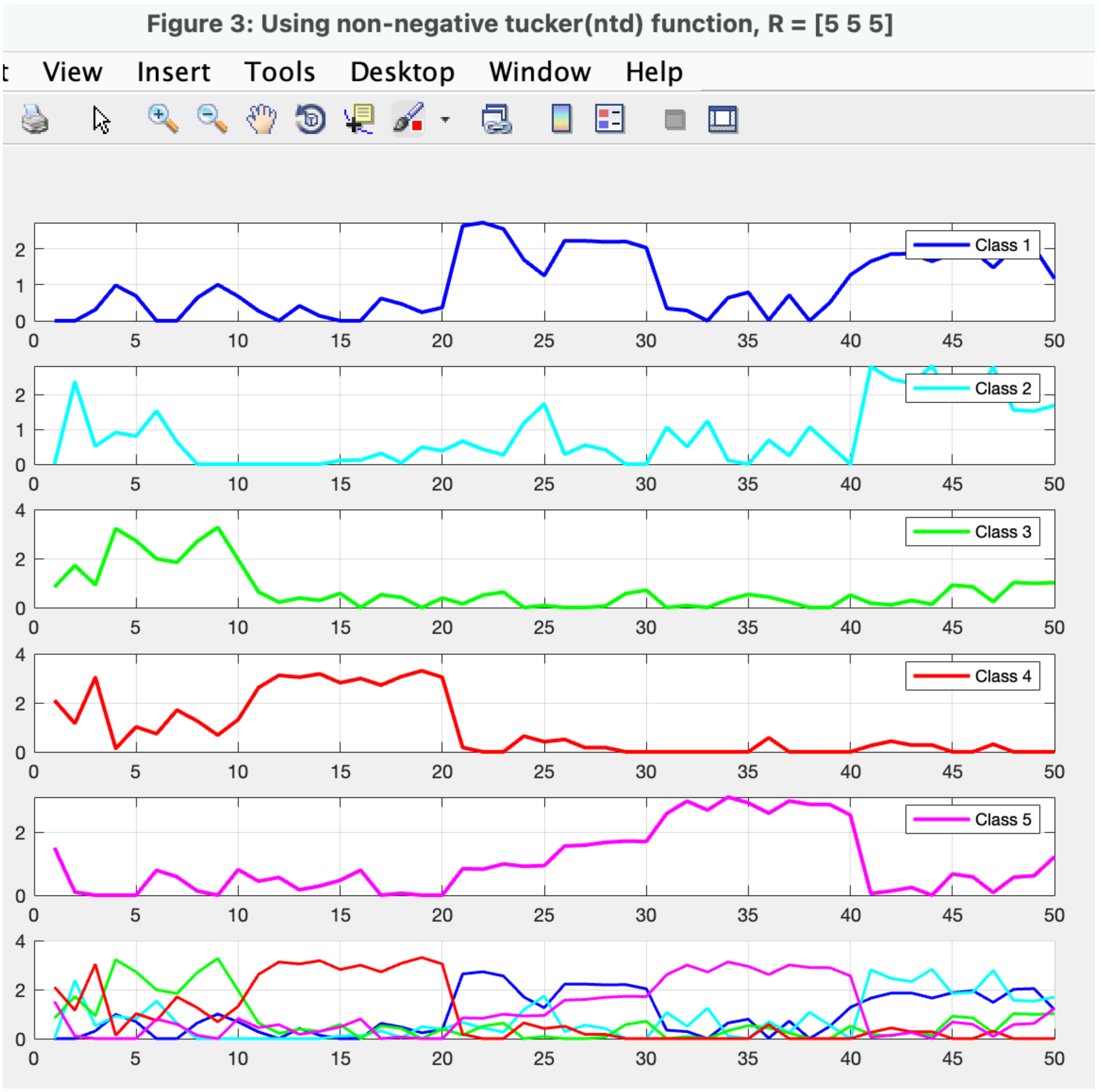


Figure 3: NNTD, R = [5 5 5]

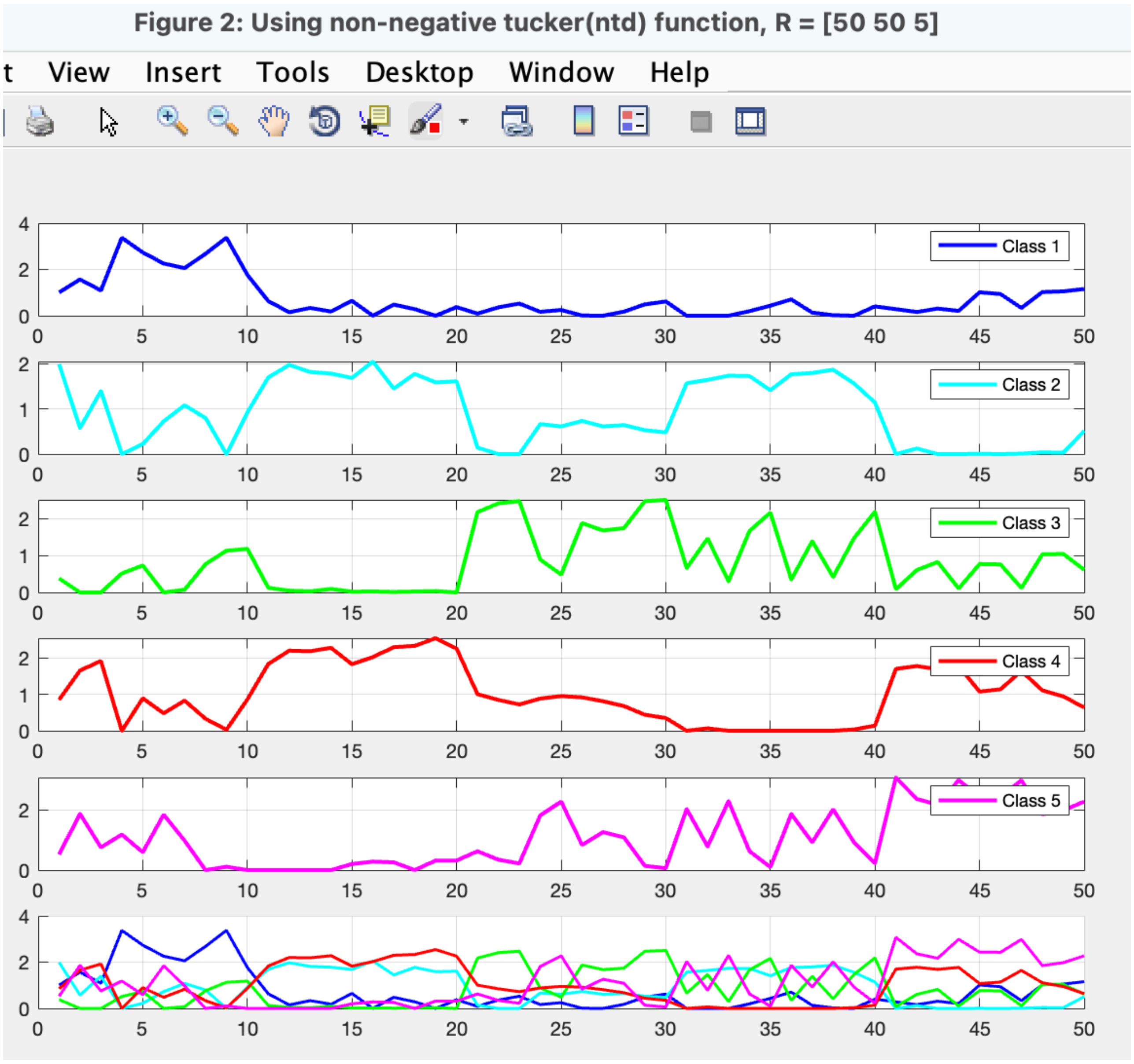


Figure 4: NNTD, R = [50 50 5]

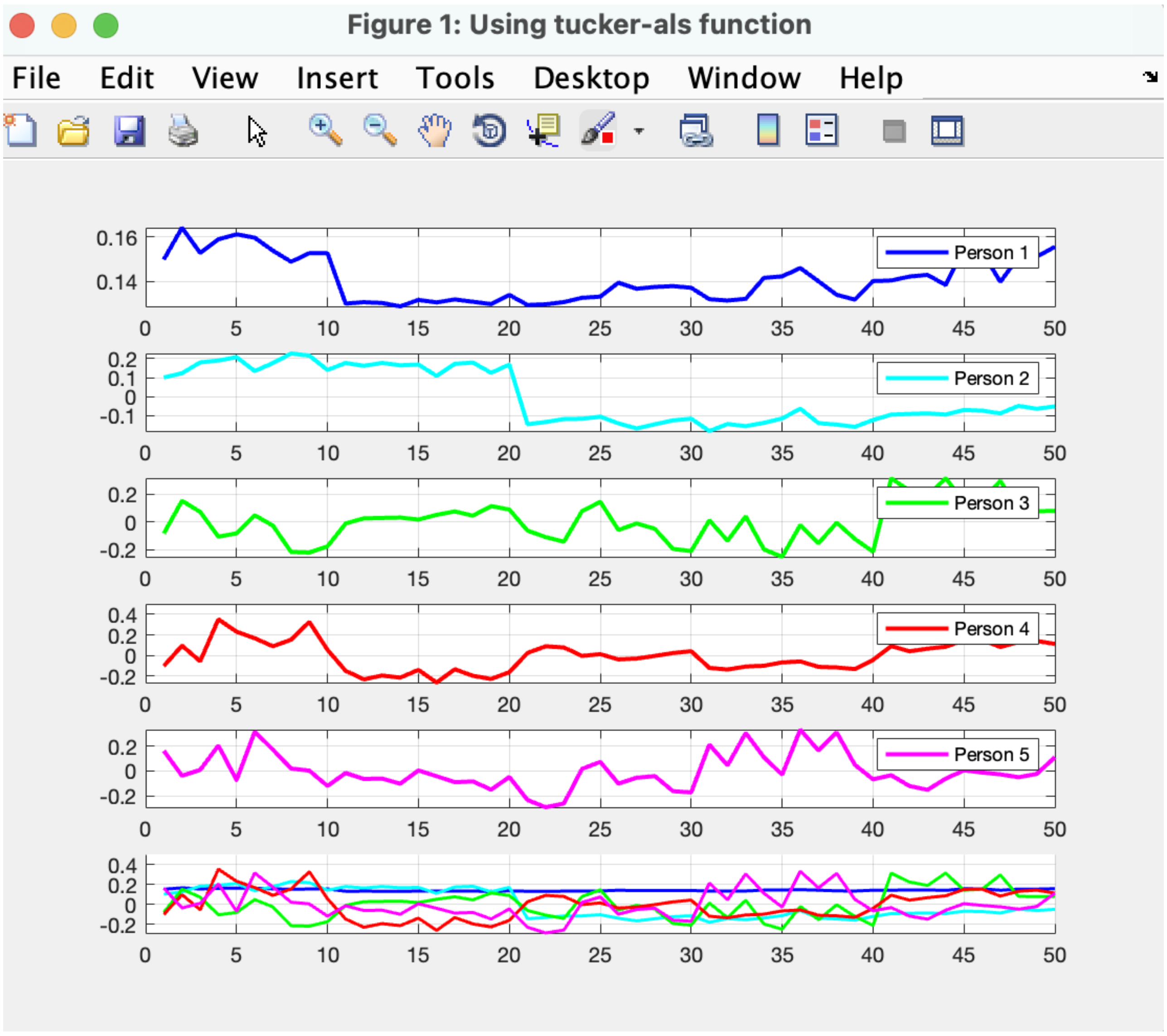


Figure 5: tucker_als

For the Tucker decomposition with both positive and negative values, we used ALS(tucker_als), HOOI(costume) and non-negative tucker(NNTD) methods. For the Non-negative Tucker (NNTD), we used the given function. Adjusting tensor dimensions, configurations $R = [5, 5, 5]$ and $R = [50, 50, 5]$ provided the best separation.

As observed, the Non-negative Tucker cases show significantly better separation than the regular Tucker decomposition.

So we can conclude that Clustering with Non-negative Tucker is much better than with regular Tucker, partly because the data tensor contains only positive values.

3

3.A

Using Tucker decomposition and reducing the rank of the Illumination mode removes shadows from the image, resulting in an output image of the person without shadows. The more the rank of the Illumination mode is reduced, the more the shadows disappear. Here we show three examples.

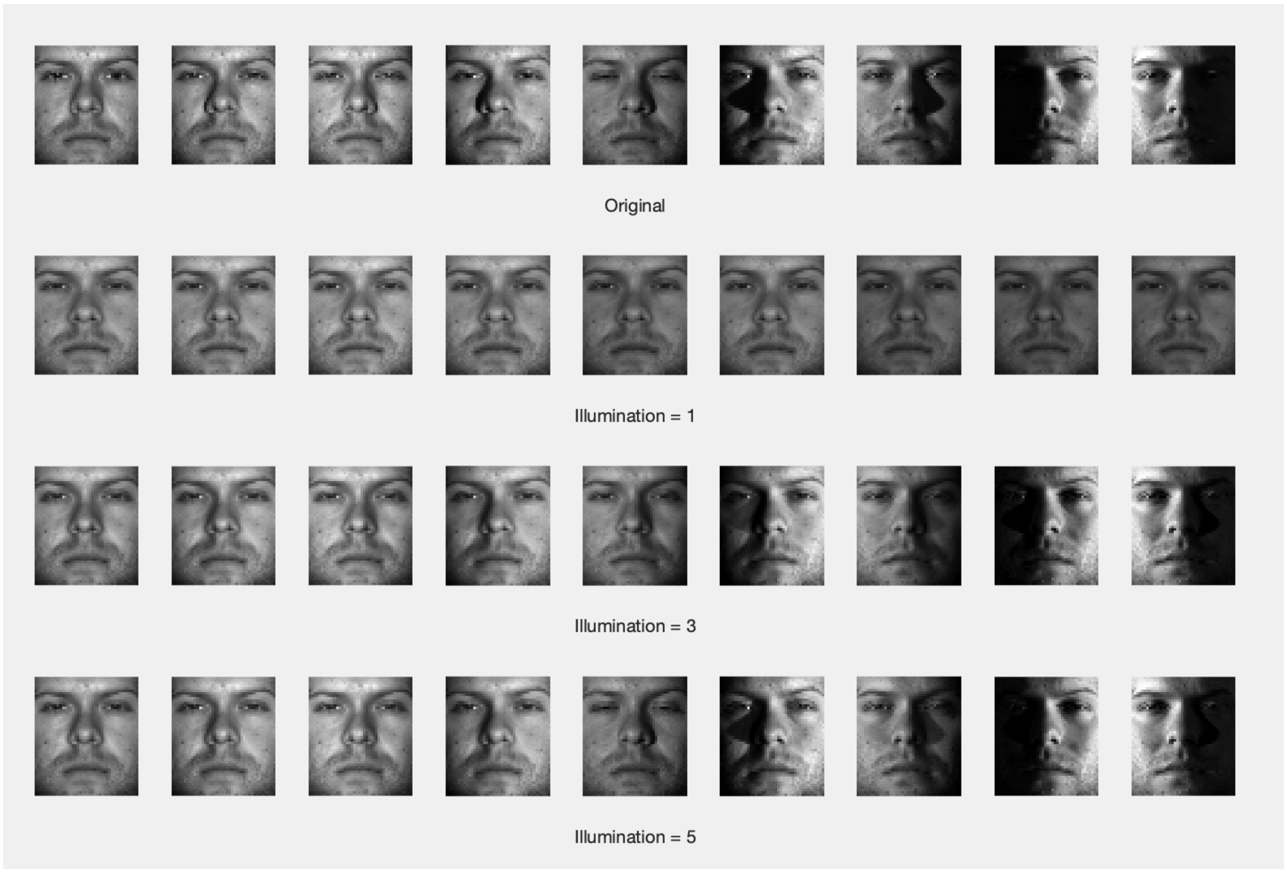


Figure 6: person 1

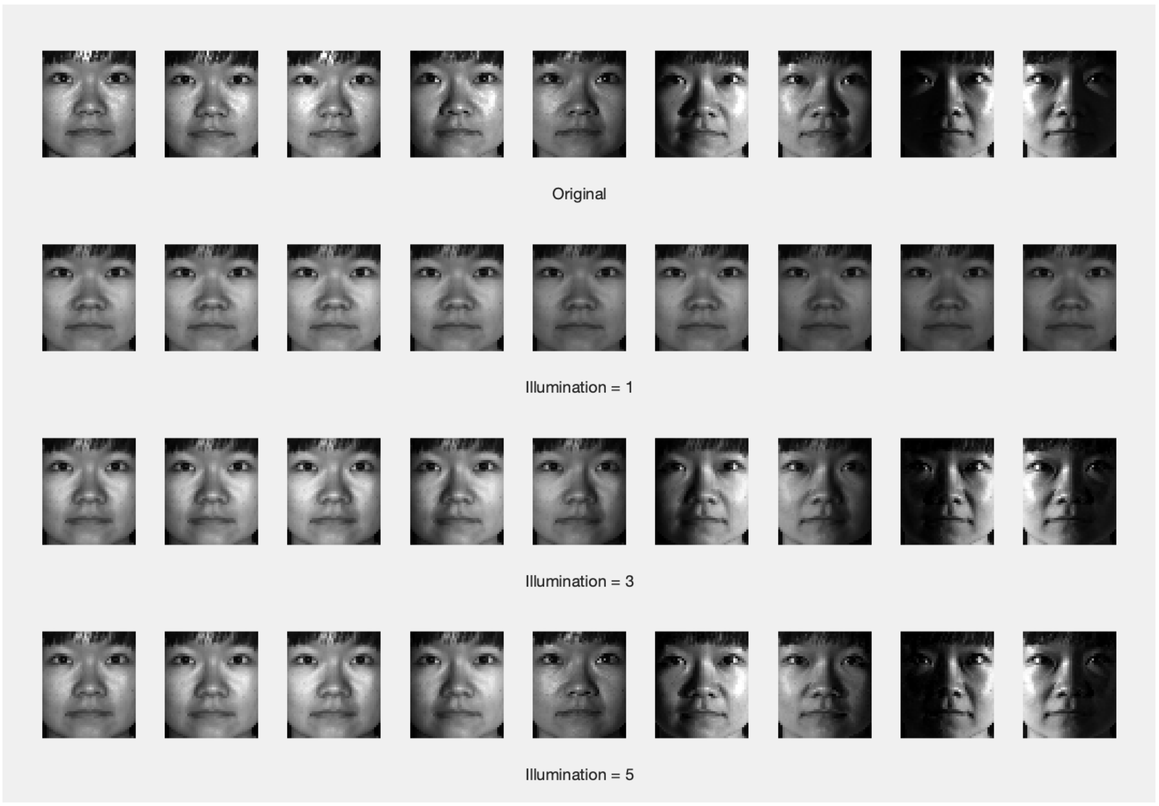


Figure 7: person 5

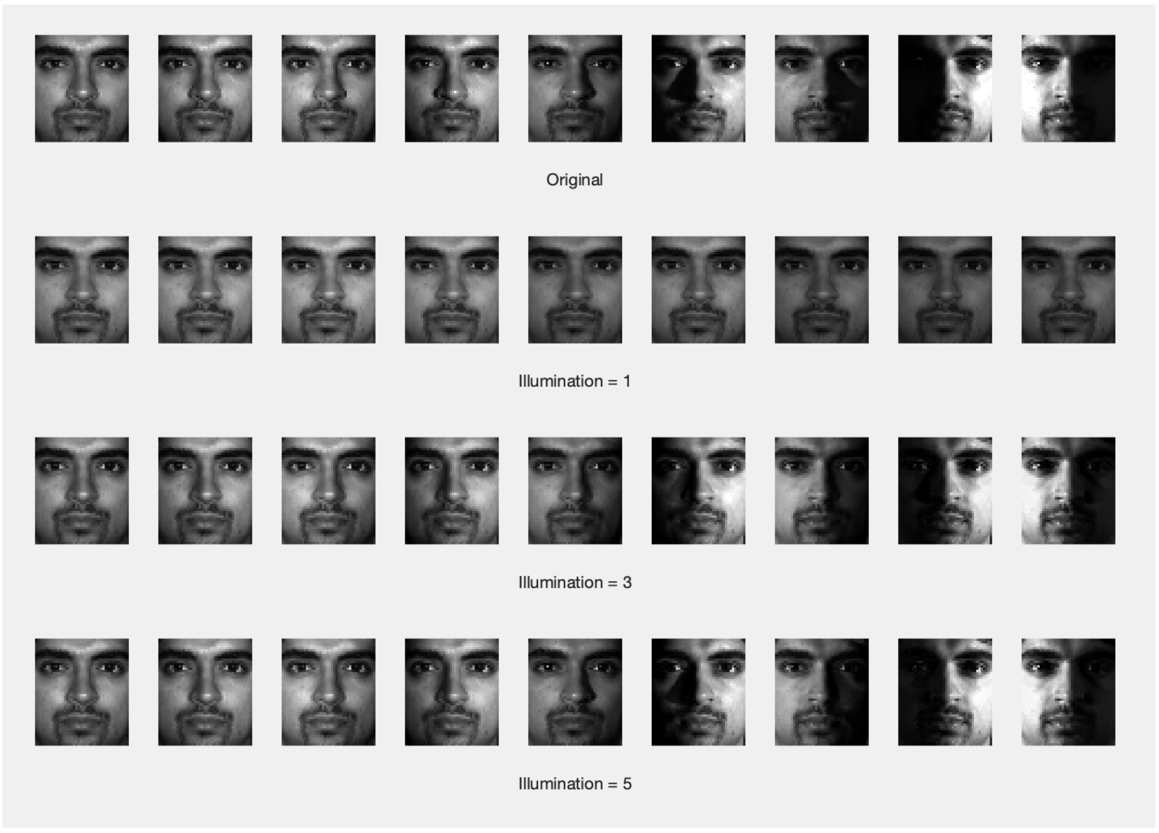


Figure 8: person 9

3.B

In the SVD decomposition, the more we reduce the rank in SVD decomposition, the more image details are removed, resulting in an image with fewer details. Additionally, SVD decomposition removes information from all parts of the image and is not as effective as Tucker decomposition in removing shadows.Here we show three examples:

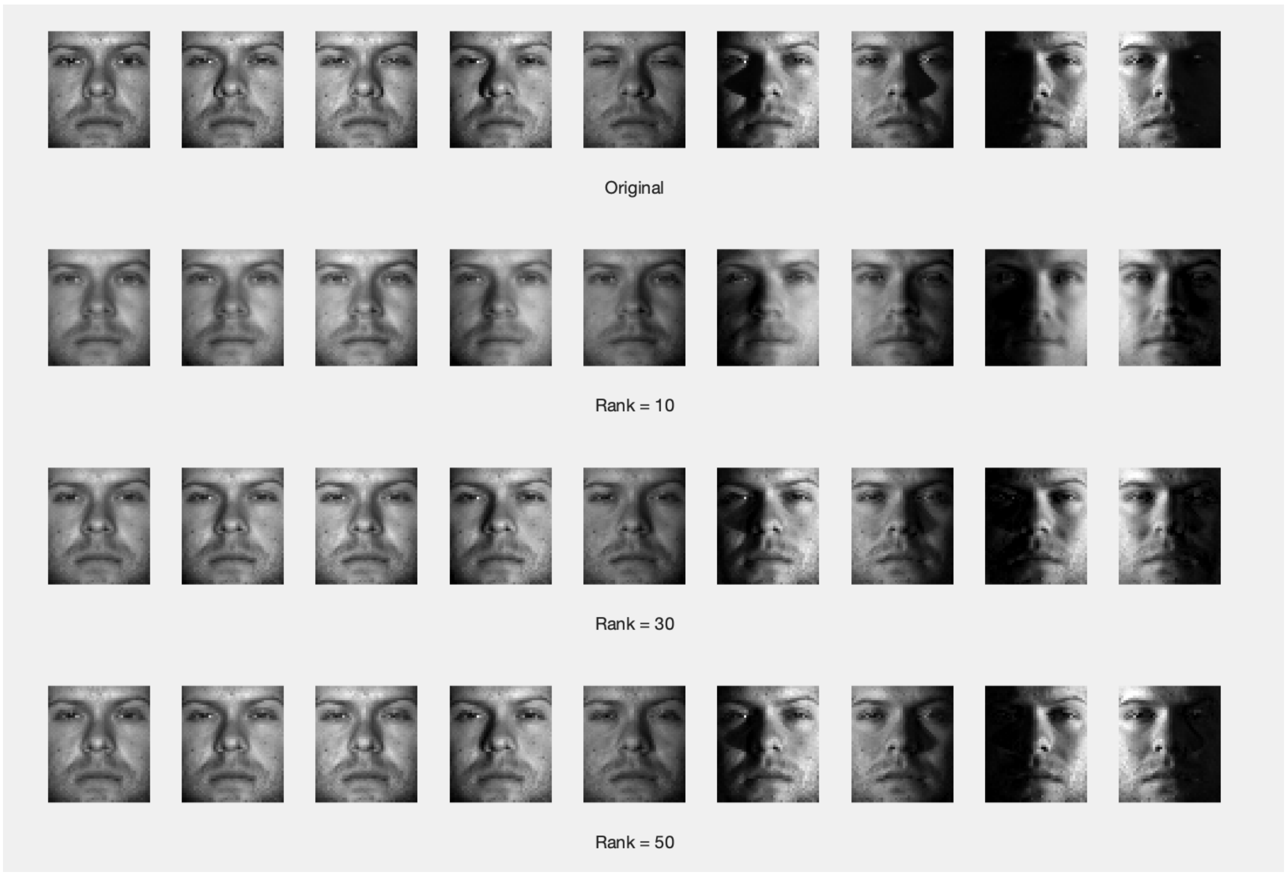


Figure 9: person 1

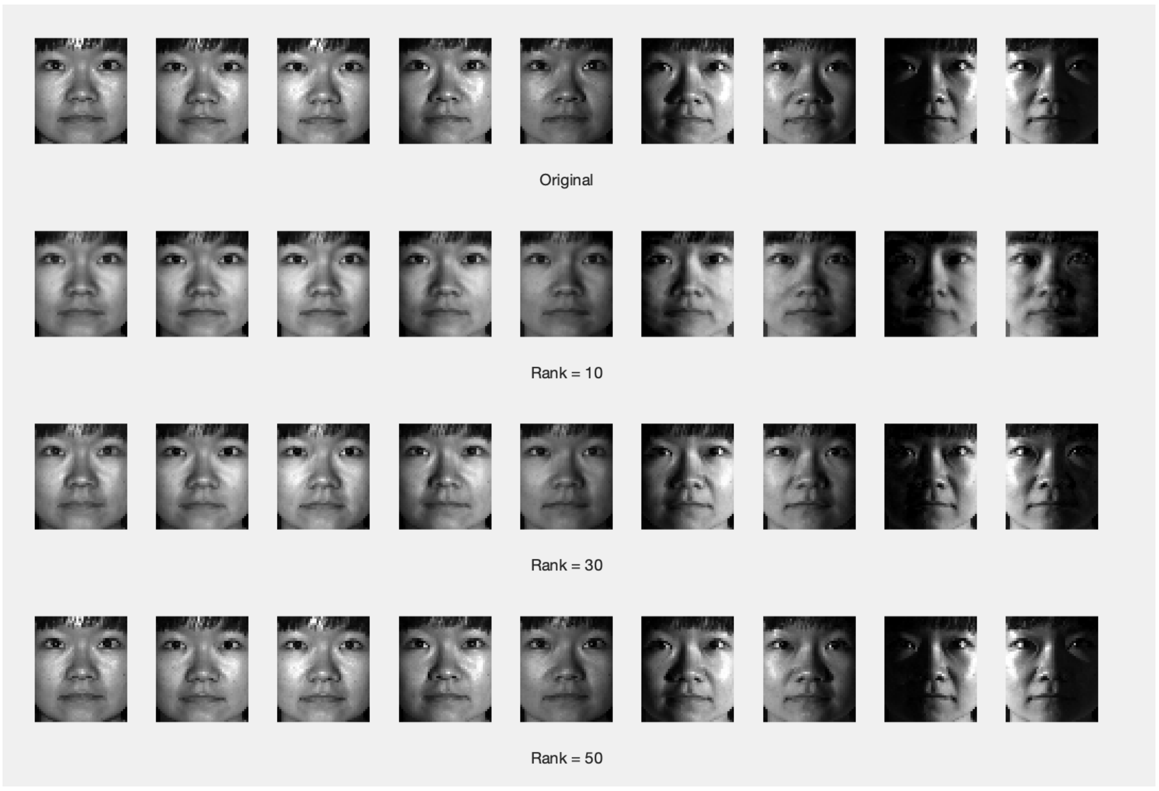


Figure 10: person 5

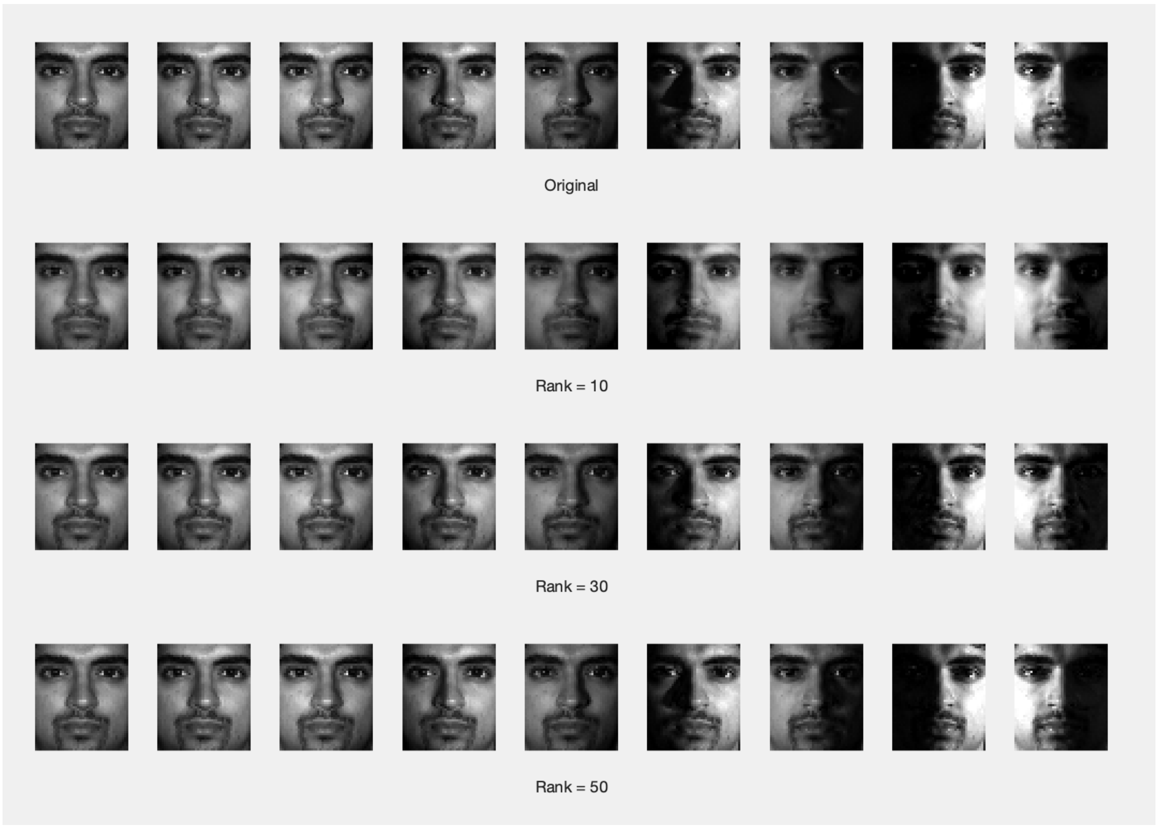


Figure 11: person 9

3.C

In the SVD case, compared to Tucker decomposition, the error is less, but the image obtained from Tucker decomposition is better. In SVD decomposition, reducing the rank removes information from all parts of the image, leading to a less detailed image. In contrast, in Tucker decomposition, the resulting image resembles the original image more, and reducing the rank in the Illumination mode removes shadows better and creates a clearer image. Although SVD decomposition introduces fewer errors, Tucker decomposition removes more shadow information and provides better clarity.

In Tucker decomposition, shadows are removed more effectively, and the resulting image resembles the original image more closely.