Gaiceanu Theodora

Assignment 1 – Image Analysis

1. Image sampling

The resulting matrix is:

0	0	0	0	0
0	2	4	6	8
0	4	8	12	16
0	6	12	17	23
0	8	16	23	31

As 0 ≤ x ≤ 1 and 0 ≤ y ≤ 1, the continuous image is a 5x5 matrix, with the following content:

f(0,1)	f(0.25,1)	f(0.5,1)	f(0.75,1)	f(1,1)
f(0,0.75)	f(0.25,0.75)	f(0.5,0.75)	f(0.75,0.75)	f(1,0.75)
f(0,0.50)	f(0.25,0.50)	f(0.5,0.50)	f(0.75,0.50)	f(1,0.50)
f(0,0.25)	f(0.25,0.25)	f(0.5,0.25)	f(0.75,0.25)	f(1,0.25)
f(0,0)	f(0.25,0)	f(0.5,0)	f(0.75,0)	f(1,0)

By replacing *f(x,y)* with *x(1-y)*, the values of the continuous image are obtained:

0	0	0	0	0
0	0.0625	0.1250	0.1875	0.2500
0	0.1250	0.2500	0.3750	0.5000
0	0.1875	0.3750	0.5625	0.7500
0	0.2500	0.5000	0.7500	1.0000

The discrete image has gray levels from 0 to 31, therefore the values from the continuous image need to be multiplied with 31. After the multiplication is done, the values are rounded, as a discrete image can only take integer values.

2. Histogram equalization

$$\begin{cases} p_s = 1 \Rightarrow \int_0^r p_r(t) = \int_0^s 1dt = s \\ s = T(r) \end{cases} \Rightarrow T(r) = \int_0^r p_r(t) = \int_0^r 6r(1-r)$$

$$T(r) = \int_0^r 6r - 6r^2 = 6\left(\frac{r^2}{2} - \frac{r^3}{3}\right) = 3r^2 - 2r^3$$

3. Neighbourhood of pixels

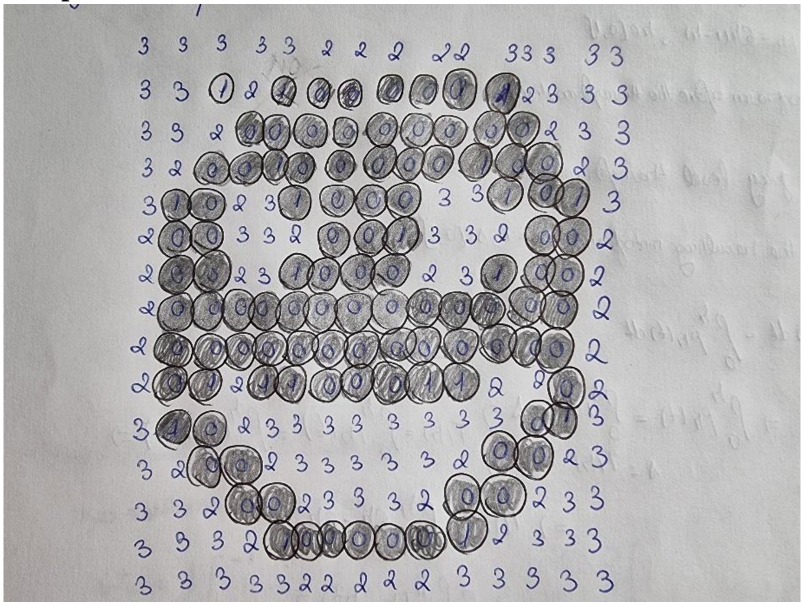


Fig. 1 The results from question 3

4. Segmentation part of OCR

```
function S = im2segment(im)
nrofsegments = 5; % could vary, probably you should estimate this
S = cell(1,nrofsegments);
m = size(im,1);
n = size(im,2);

count = 28; %indexes for selecting only a digit (140:5=28)
start = 1;
for kk = 1:nrofsegments
    %filter the image with a gaussian filter
    %in order to reduce noise
    H = fspecial('gaussian'); %gaussian filter
    filteredImg = imfilter(im,H);
    %construct a new black image
    new_image = zeros(m,n);
    %do the segmentation for only a digit - only 28 columns
    %inspecting the filtered version of the image bild, it can be observed
    %that the contour of the digit has values higher than 40
    %therefore the chosen threshold is 43
    new_image(:,start:count) = filteredImg(:,start:count)>43;
    S{kk} = new_image;
    %increment the start and stop indexes
    %in order to go to the next digit
    start = start+28;
    count = count+28;
    %S{kk}= (rand(m,n)<0.5); % this is not a good segmentation method...
end
```

```
>> inl1_test_and_benchmark

You tested 10 images in folder ../datasets/short1

The jaccard scores for all segments in all images were

    0.9390    0.8750    0.9412    0.7603    0.9441

    0.8911    0.9562    0.8605    0.9655    0.9394

    0.9474    0.9396    0.9600    0.9394    0.9322

    0.7157    0.9058    0.9101    0.9304    0.9346

    0.9414    0.9286    0.9348    0.8750    0.9580

    0.9771    0.9464    0.9530    0.9524    0.9862

    0.9602    0.8706    0.9115    0.9420    0.9153

    0.9405    0.9535    0.7647    0.9526    0.8696

    0.9106    0.8913    0.9540    0.9444    0.9195

    0.9633    0.7105    0.9000    0.9450    0.7100


The mean of the jaccard scores were 0.91139

This is good!
```

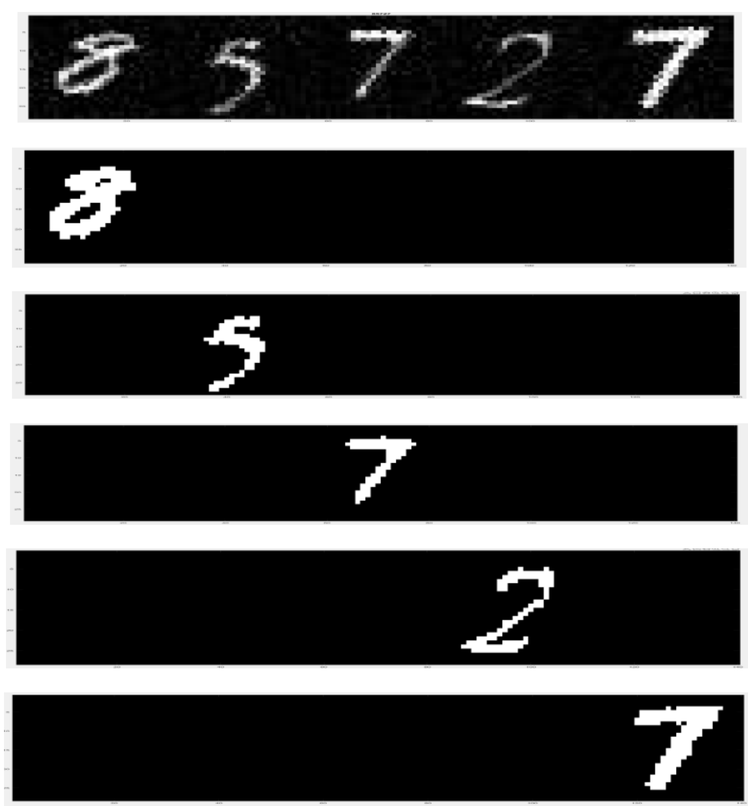


Fig. 2 Segmentation results

5. Dimensionality

It is known that the dimension of a vector space is given by the number of vectors in a basis for that vector space [1].

Since A is a set of images with 2x3 pixels, the dimension k of A is 2·3 = 6.

An example basis for A may be:

$$e_1 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ e_2 = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix},\ e_3 = \begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix},\ e_4 = \begin{pmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix},\ e_5 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix},\ e_6 = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

Since B is a set of images with 2000x3000 pixels, the dimension k of B is 2000·3000 = 6000000.

The basis elements can be chosen similar to the elements from the previous basis, which means that we will have 60000000 elements, each one having 2000 rows and 3000 columns. Each element has one digit of 1, and 5999999 of 0.

6. Scalar products and norm on images

According to [2], the scalar product for images has the following expression:

$$f \cdot \mathrm{g} = \sum_{i=1}^{M} \sum_{j=1}^{N} \overline{f(\imath,\jmath)} g(i,j)$$

This means that the scalar product is an element-wise product. The result of the scalar product of two matrices is a number.

According to [2], the norm of an image is:

$$\|f\| = \sqrt{f \cdot \mathrm{f}} = \sqrt{\sum_{i=1}^{M} \sum_{j=1}^{N} \overline{f(\imath,\jmath)} f(i,j)}$$

Therefore, the norm of an image is the square root of the scalar product of the image with itself.

$$\|u\| = \sqrt{4^2 + (-2)^2 + (-1)^2 + 5^2} = \sqrt{46}$$

$$\|v\| = \sqrt{\frac{(-1)^2}{2^2} + \frac{1}{2^2}^2 + \frac{1}{2^2}^2 + \frac{(-1)^2}{2^2}} = 1$$

$$\|w\| = \sqrt{\frac{1}{2^2}^2 + \frac{1}{2^2}^2 + \frac{(-1)^2}{2^2} + \frac{(-1)^2}{2^2}} = 1$$

$$u \cdot \mathrm{v} = 4\frac{(-1)}{2} + (-2)\frac{1}{2} + (-1)\frac{1}{2} + 5\frac{(-1)}{2} = -6$$

$$u \cdot w = 4\,\frac{1}{2} + (-2)\frac{1}{2} + (-1)\frac{(-1)}{2} + 5\frac{(-1)}{2} = -1$$

$$v \cdot w = \frac{(-1)}{2}\frac{1}{2} + \frac{1}{2}\frac{1}{2} + \frac{1}{2}\frac{(-1)}{2} + \frac{(-1)}{2}\frac{(-1)}{2} = 0$$

As it can be seen from the above expression, the scalar product of v and w is 0, and the scalar products $v \cdot v$ and $\mathrm{w} \cdot \mathrm{w}$ are 1, which means that the matrices $\{v, w\}$ are orthonormal.

Since $\{v, w\}$ are orthonormal, the coordinates are:

$$x_1 = u \cdot \mathrm{v} = -6$$

$$x_2 = u \cdot \mathrm{w} = -1$$

The orthogonal projection is:

$$\hat{\mathrm{u}} = -6v - w = \begin{bmatrix} 5/2 & -5/2 \\ -7/2 & 7/2 \end{bmatrix}$$

$$\mathrm{u} - \hat{\mathrm{u}} = \begin{bmatrix} 4 & -1 \\ -2 & 5 \end{bmatrix} - \begin{bmatrix} 2.5 & -2.5 \\ -3.5 & 3.5 \end{bmatrix} = \begin{bmatrix} 1.5 & 1.5 \\ 1.5 & 1.5 \end{bmatrix}$$

$$|\mathrm{u} - \hat{\mathrm{u}}|^2 = \begin{bmatrix} 1.5 & 1.5 \\ 1.5 & 1.5 \end{bmatrix} \cdot \begin{bmatrix} 1.5 & 1.5 \\ 1.5 & 1.5 \end{bmatrix} = 9$$

$$\frac{\|\mathrm{u} - \hat{\mathrm{u}}\|}{\|\mathrm{u}\|} = \frac{\sqrt{9}}{\sqrt{46}} = 0.4423 \rightarrow$$

The resulting projection is not a good approximation

7. Image compression

$$\Phi_1 \cdot \Phi_1 = \Phi_2 \cdot \Phi_2 = \Phi_2 \cdot \Phi_2 = 1$$

$$\Phi_1 \cdot \Phi_2 = \Phi_1 \cdot \Phi_3 = \Phi_1 \cdot \Phi_4 = \Phi_2 \cdot \Phi_3 = \Phi_2 \cdot \Phi_4 = \Phi_3 \cdot \Phi_4 = 0$$

Since every scalar product from above is 0, when $\Phi_\mathrm{i} \neq \Phi_\mathrm{j}$, and 1, when $\Phi_I = \Phi_\mathrm{j}$, the four images $\Phi_1, \Phi_2, \Phi_3, \Phi_4$ are orthonormal.

The approximate image f_a is as close to f as possible if one computes the orthogonal projection of f onto the space spanned by $\Phi_1, \Phi_2, \Phi_3, \Phi_4$.

Therefore, the four numbers (coordinates) x1, x2, x3, x4 can be calculated as follows:

$$\mathrm{x}_1 = \mathrm{f} \cdot \Phi_1$$

$$\mathrm{x}_2 = \mathrm{f} \cdot \Phi_2$$

$$\mathrm{x}_3 = \mathrm{f} \cdot \Phi_3$$

$$x_4 = f \cdot \Phi_4$$

Particularly, for the image f, the values obtained after computing the scalar products above are: x1 = 17, x2 = -4, x3 = 1.5, x4 = 1.667.

Using the values of the coordinates above, the approximate fa can be computed as:

$$f_a = x_1\Phi_1 + x_2\Phi_2 + x_3\Phi_3 + x_4\Phi_4 =$$

$$17\Phi_1 - 4\Phi_2 + 1.5\Phi_3 + 1.667\Phi_4 =$$

$$f_a = \begin{matrix} 1.305 & 6.2222 & -0.1944 \\ 6.9722 & 5.6667 & 5.4722 \\ 3.1111 & -0.5556 & 7.1111 \end{matrix}$$

$$\begin{matrix} 3.6667 & 5.1111 & 7.6667 \end{matrix}$$

In order to say if f_a is close to f, we can compute:

$$|f - f_a|^2 = 136.9722$$

$$\frac{||f - f_a||}{||f||} = \frac{\sqrt{136.9722}}{\sqrt{447}} = 0.5536$$

Thus f_a is not close to *f*.

8. Image bases

The code for the function:

```
function [up, r] = projection(u,e1,e2,e3,e4)
    %input arguments - image, the four bases elements
    %output arguments - projection, the norm
    %compute the coordinates x1, x2, x3, x4
    prod1 = u.*e1;
    x1 = sum(prod1(:)); %scalar product result
    prod2 = u.*e2;
    x2 = sum(prod2(:)); %scalar product result
    prod3 = u.*e3;
    x3 = sum(prod3(:)); %scalar product result
    prod4 = u.*e4;
    x4 = sum(prod4(:)); %scalar product result
    %compute the orthogonal projection
    up = x1*e1 + x2*e2 + x3*e3 + x4*e4;
    %compute the norm
    prod_r = (u-up).*(u-up);
    r = sqrt(sum(prod_r(:)));
end
```

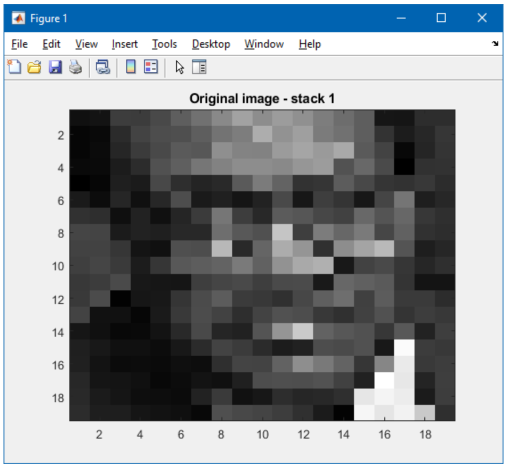


Figure 1
File Edit View Insert Tools Desktop Window Help
Original image - stack 1
2 4 6 8 10 12 14 16 18
2 4 6 8 10 12 14 16 18

(a)

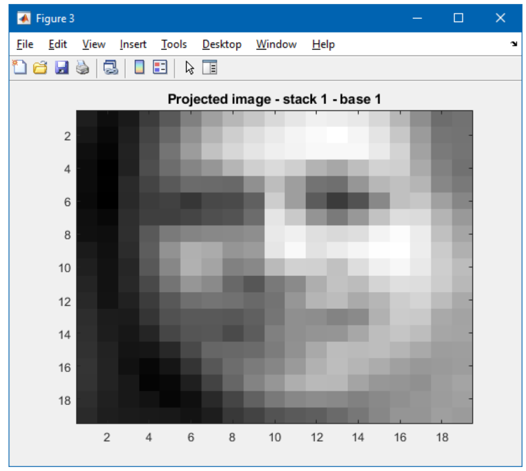


Figure 3
File Edit View Insert Tools Desktop Window Help
Projected image - stack 1 - base 1
2 4 6 8 10 12 14 16 18
2 4 6 8 10 12 14 16 18

(b)

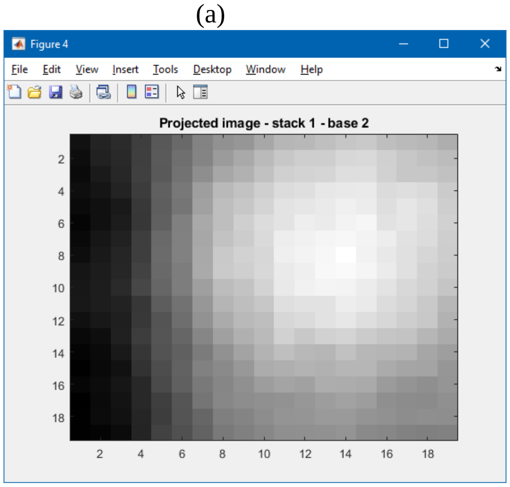


Figure 4
File Edit View Insert Tools Desktop Window Help
Projected image - stack 1 - base 2
2 4 6 8 10 12 14 16 18
2 4 6 8 10 12 14 16 18

(c)

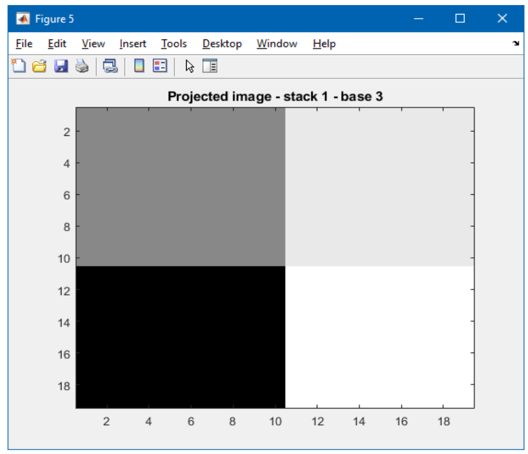


Figure 5
File Edit View Insert Tools Desktop Window Help
Projected image - stack 1 - base 3
2 4 6 8 10 12 14 16 18
2 4 6 8 10 12 14 16 18

(d)

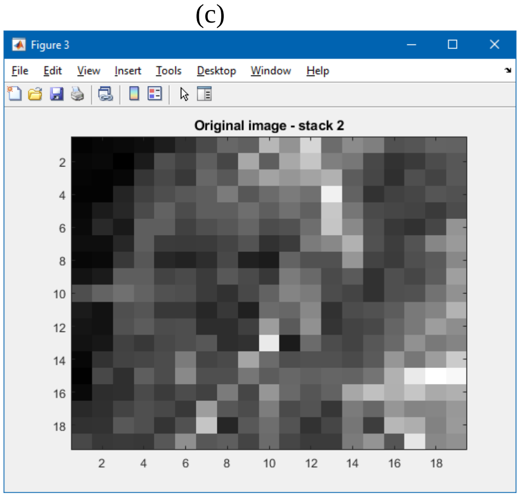


Figure 3
File Edit View Insert Tools Desktop Window Help
Original image - stack 2
2 4 6 8 10 12 14 16 18
2 4 6 8 10 12 14 16 18

(e)

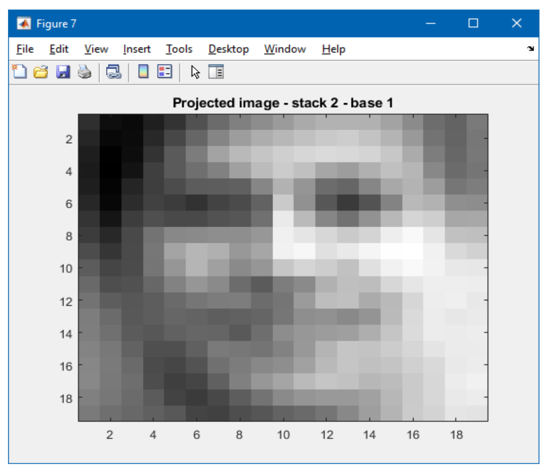
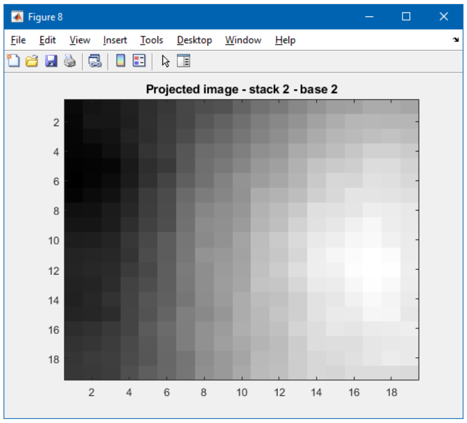
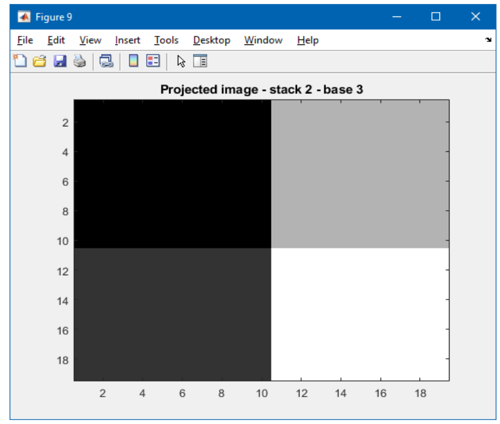


Figure 7
File Edit View Insert Tools Desktop Window Help
Projected image - stack 2 - base 1
2 4 6 8 10 12 14 16 18
2 4 6 8 10 12 14 16 18

(f)



(g)



(h)

Fig. 3 Original and projected images: (a) - Image from the first set, (b) - Projected image from the first set using base 1, (c) - Projected image from the first set using base 2, (d) - Projected image from the first set using base 3, (e) - Image from the second set, (f) - Projected image from the second set using base 1, (g) - Projected image from the second image using base 2, (h) - Projected image from the second set using base 3

The input images from the stack1 preserve more face characteristics than the input images from stack2. The basis that represents best the face characteristics is base 1. This fact can be observed for both sets of images. Base 2 seems to smooth the input image. Base 3 divides the input image intro 4 categories.

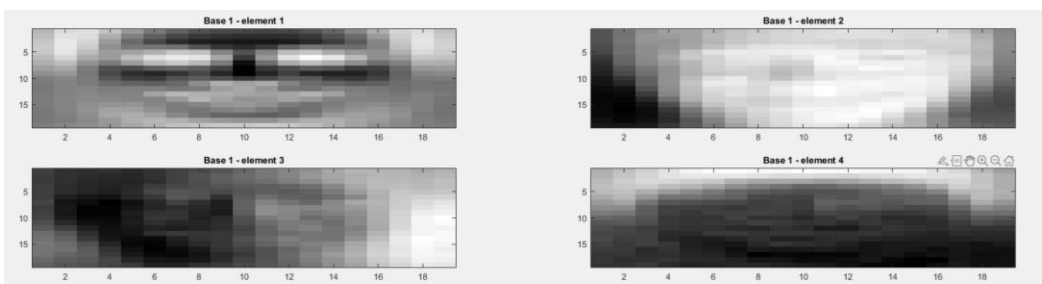


Fig. 4 Base 1

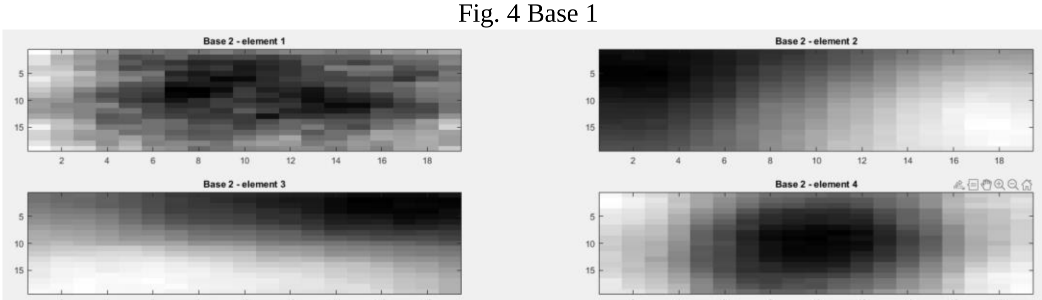


Fig. 5 Base 2

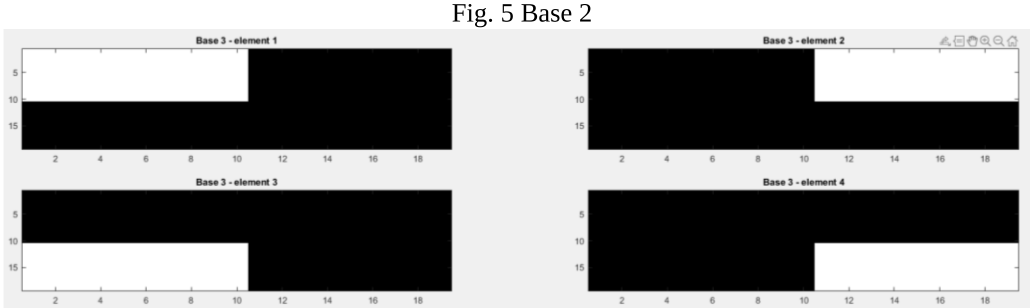


Fig. 6 Base 3

Base 1 focuses upon the face features, like eyes, mouth, form of the face. Base 2 focuses upon shadows. Base 3 aims to divide the pixels in black and white.

Means of the error norms are shown in Table 1:

	Base 1	Base 2	Base 3
Stack 1	821.0271	860.4754	944.9009
Stack 2	795.1902	649.2013	697.3214

Table 1. Means of the error norms

Considering the projected image and the mean value of the error norm, base 1 works best for stack.

Considering the projected image and the mean value of the error norm, base 2 works best for stack.

References

[1] J. He, Math 2331 - Linear Algebra Course (2016), Department of Mathematics, University of Houston.

[2] M. Oskarsson, Image Analysis Course (2021), Lund University.