Assignment 1 - FMAN20

Maks Epsteins

September 21, 2021

Exercise 1

First the matrix was created using the following Julia function (Julia was used since language was not specified):

```
matrix = [x*(1-y) for x = 0:0.25:1, y = 1:-0.25:0]
```

Which results in the following matrix:

$$matrix = \begin{bmatrix} 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0625 & 0.125 & 0.1875 & 0.25 \\ 0.0 & 0.125 & 0.25 & 0.375 & 0.5 \\ 0.0 & 0.1875 & 0.375 & 0.5625 & 0.75 \\ 0.0 & 0.25 & 0.5 & 0.75 & 1.0 \end{bmatrix}$$

Then the quantization is made with the following code:

```
gray_level = 32

#Does the quantization by multiplying each value with the gray level - 1 and then rounding.
quantization(matrix, gray_level) = map(e -> round((gray_level - 1)*e), matrix)

println(quantization(matrix, gray_level))
```

This code prints the following matrix:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 2 & 4 & 6 & 8 \\ 0 & 4 & 8 & 12 & 15 \\ 0 & 6 & 12 & 17 & 23 \\ 0 & 8 & 15 & 23 & 31 \end{bmatrix}$$

Exercise 2

$P_r = 6r(1-r), r \in [0,1]$
$P_s = 1$
The graylevel transformation which outputs a flat histogram is the following:
$T(r) = \int_0^r P_r(t)dt = \int_0^r 6t(1-t)dt = [3t^2 - 2t^3]_0^r = 3r^2 - 2r^3$

Exercise 3

3.1 Matrix with circles around each element with intensity 0 or 1

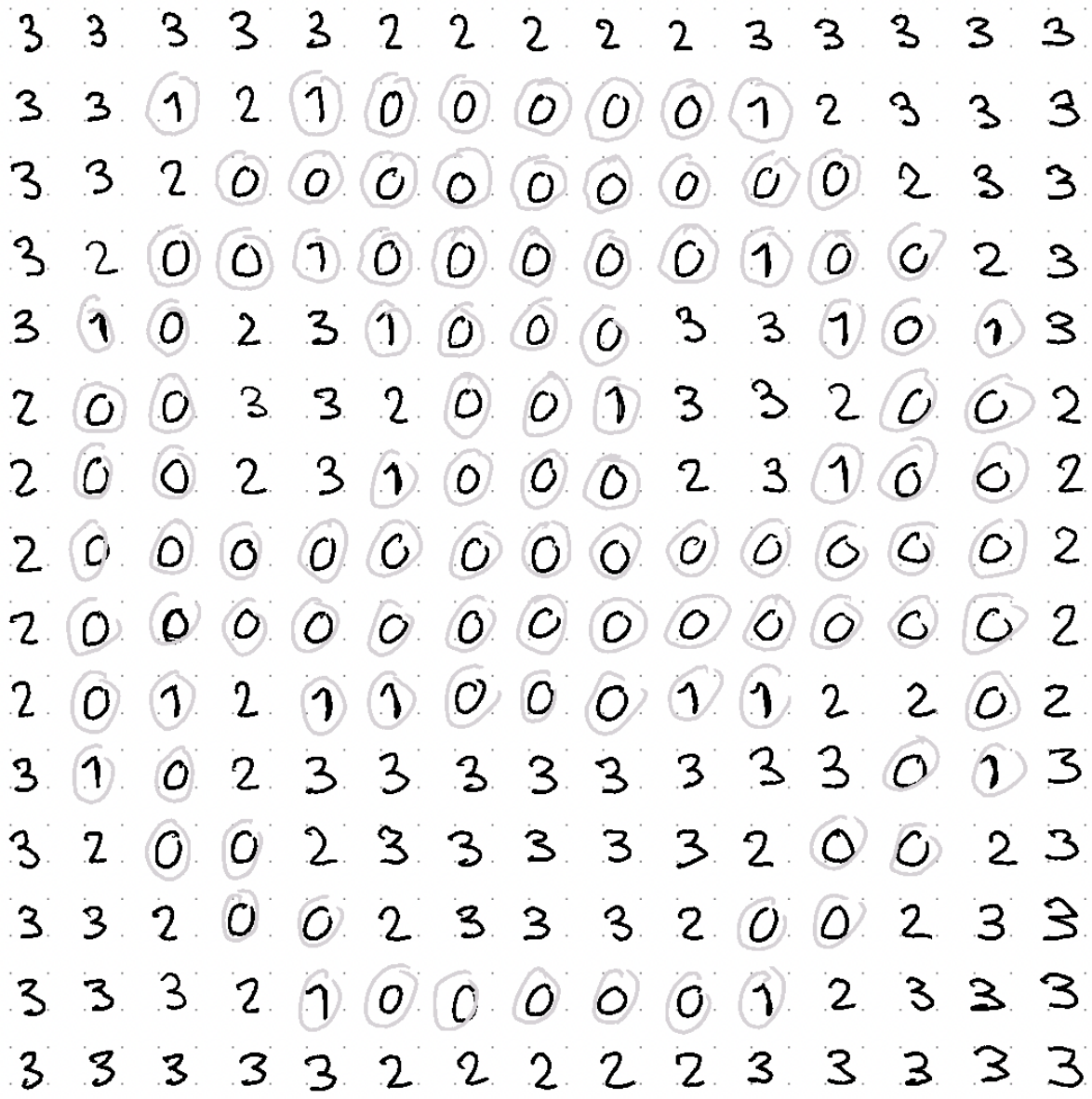

Figure 1: 3a

3.2 Matrix with circles around each element with intensity 0 or 1 and at least two 4-neighbours with the intensity 0 or 1.

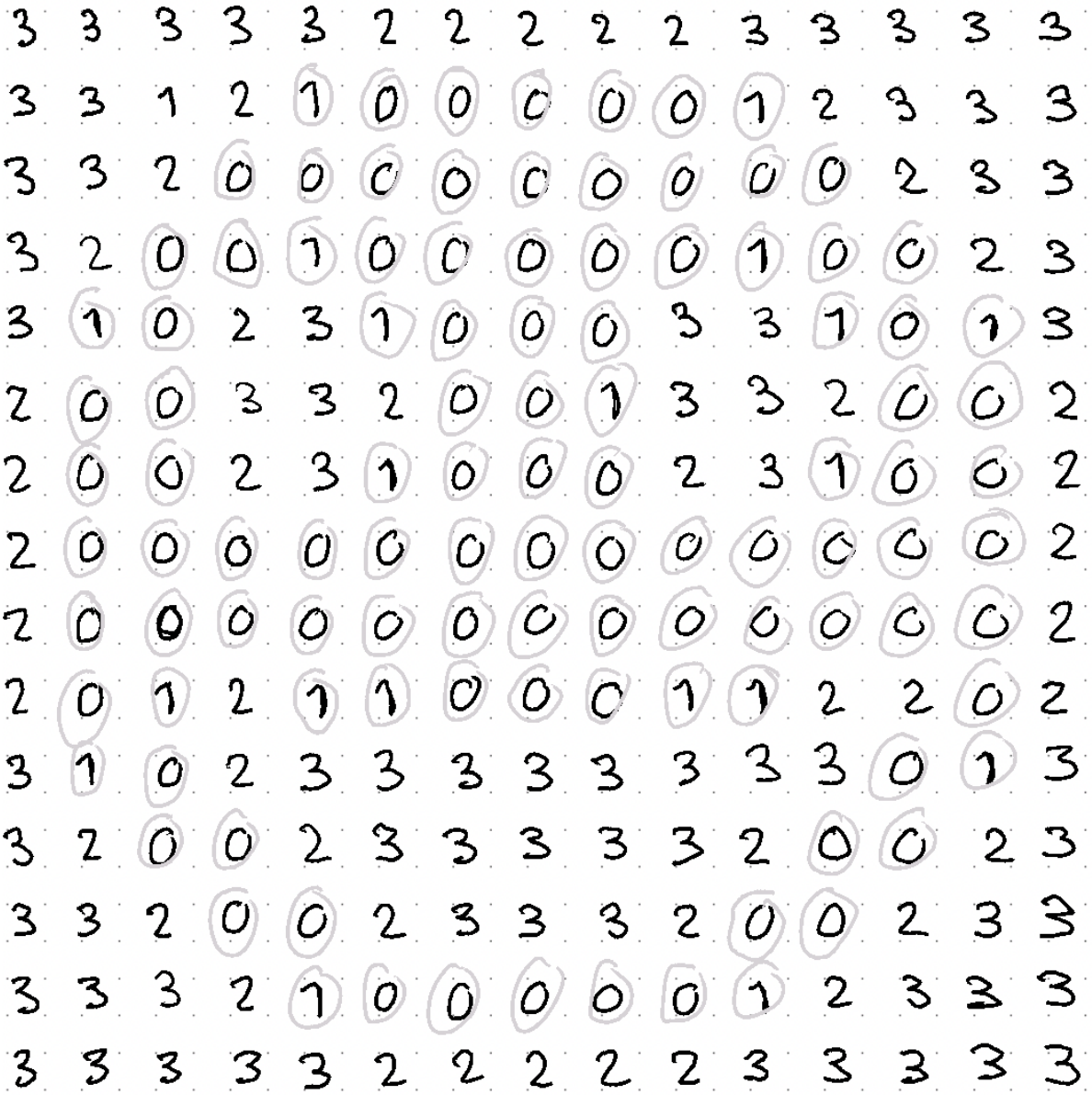

Figure 2: 3b

Exercise 4

4.1 The im2segment.m code

```
function S = im2segment(im)

nrofsegments = 5;
S = cell(1,nrofsegments);
m = size(im,1);
n = size(im,2);

for r = 1:m
    for c=1:n
        %Set threshold value at 27 because that value gave the best results
        if im(r, c) >= 27
            im(r, c) = 1;
        else
            im(r,c) = 0;
        end
    end
end

%I use a convolution kernel in order to calculate the amount of neighbors at each index
%to try to fix some errors from the threshholding.
neighborKernel = [1,1,1;1,0,1;1,1,1];
nbrNeighbors = conv2(im, neighborKernel, 'same');

%These loops should be pretty self explanatory
for r = 1:m
    for c=1:n
        if nbrNeighbors(r, c) <= 2
            im(r, c) = 0;
        elseif nbrNeighbors(r, c) >= 6
            im(r, c) = 1;
        end
    end
end

im = bwlabel(im);
%I use the k-means segmentation method to segment the numbers
[im,~] = imsegkmeans(uint8(im), nrofsegments + 1);
%blackZone is a variable checks which number is assigned to the background of the image.
blackZone = mode(mode(im));

for kk = 1:nrofsegments
    currentVal = 0;
    S{kk} = zeros(m,n);
    for c = 1:n
```

```
        for r = 1:m
            if im(r,c) ~= blackZone && currentVal == 0
                currentVal = im(r,c);
            end
            if im(r,c) == currentVal
                S{kk}(r,c) = 255;
                im(r,c) = blackZone;
            else
                S{kk}(r,c) = 0;
            end
        end
    end
end
```

4.2 An example of the ground truth and the conducted segmentation

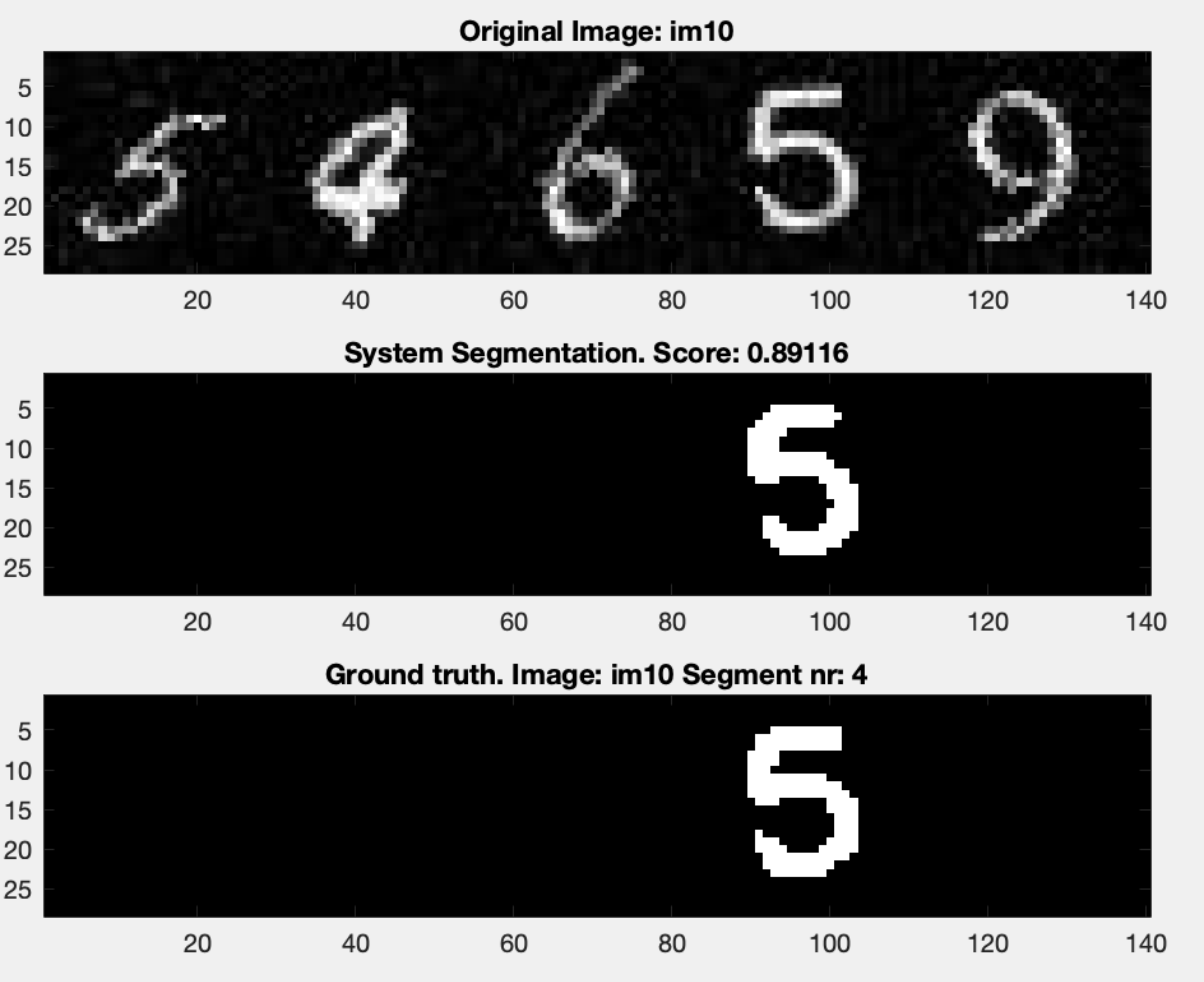

Figure 3: Result from running the segmentation algorithm

4.3 The results of the jaccard scores

```
You tested 10 images in folder ../datasets/short1
The jaccard scores for all segments in all images were
    0.9162    0.8636    0.8989    0.8226    0.9220
    0.8544    0.9085    0.8647    0.9020    0.9624
    0.8674    0.9341    0.9015    0.9259    0.9167
    0.7619    0.9281    0.9022    0.9391    0.8765
    0.9180    0.9202    0.9137    0.8400    0.9122
    0.9078    0.9298    0.9286    0.9527    0.9156
    0.9167    0.9176    0.8824    0.9007    0.8760
    0.9294    0.9213    0.8462    0.9231    0.8592
    0.8672    0.8786    0.9371    0.9291    0.8778
    0.9107    0.8205    0.9140    0.9099    0.7900

The mean of the jaccard scores were 0.89629
You can do better
```

Figure 4: Text results from the benchmark script

Exercise 5

5.1 The dimension k of the matrix A and example basis for said matrix

The dimension of the A matrix is k = 6 since 2*3 = 6. An example basis for A is:

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

5.2 Dimension of matrix B and choice of bases

The dimension of the B matrix is k = $6 \cdot 10^6$ since 2000*3000 = $6 \cdot 10^6$. The basis elements for the B matrix can be chosen by having 2000*3000 basis elements with a singular 1 at a unique index in each matrix, while the rest of the elements in said matrix are set to 0.

Exercise 6

6.1 Definition of Scalar Product

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

6.2 Definition of image norm

$$||f|| = \sqrt{f \cdot f} = \sqrt{\sum_{i=1}^{M} \sum_{i=j}^{N} f(i,j) \cdot g(i,j)}$$

6.3 Calculations of the l_2 norms and and dot products for each vector

$$||u||_2 = \sqrt{4^2 + (-1)^2 + (-2)^2 + 5^2} \approx 6.7823$$
$$||v||_2 = \sqrt{(-1/2)^2 + (1/2)^2 + (1/2)^2 + (-1/2)^2} = 1$$
$$||w||_2 = \sqrt{(-1/2)^2 + (1/2)^2 + (1/2)^2 + (-1/2)^2} = 1$$

$$u \cdot v = 4 \cdot (-1/2) + (-1) \cdot 1 + (-2) \cdot 1/2 + 5 \cdot (-1) = -9$$
$$u \cdot w = 4 \cdot 1/2 + (-1) \cdot (-1/2) + (-2) \cdot 1/2 + 5 \cdot (-1) = -3.5$$
$$v \cdot w = 1/2(-1 \cdot 1 + (-1) \cdot 1 + 1 \cdot 1 + (-1) \cdot (-1)) = 0$$

6.4 Proof of orthonormality

$$v \cdot w = 0, v \cdot v = 1, w \cdot w = 1$$

=> v,w is an ON-basis in $\mathbb{R}^2$. The dot products were calculated in Julia using the LinearAlgebra package.

6.5 Orthogonal projection of u on vectors v and w

The subspace spanned by v,w is denoted with Ω.

$$u_\Omega = x_1 \cdot v + x_2 \cdot w$$

$$x_1 = u \cdot v = -6, x_2 = u \cdot w = -1$$
$$u_\Omega = -6 \cdot v - 1 \cdot w = 1/2 \cdot \begin{bmatrix} 5 & -5 \\ -7 & 7 \end{bmatrix}$$

The above projection I would say is deficient since the values in the matrix are quite different from the original values in the u matrix.

Exercise 7

7.1 Table of tested dot products calculated using Julia

	ϕ_1	ϕ_2	ϕ_3	ϕ_4
ϕ_1	1	0	0	0
ϕ_2	-	1	0	0
ϕ_3	-	-	1	0
ϕ_4	-	-	-	1

7.2 Calculating x-values

The x values are determined as follows:

$$x_n = f \cdot \phi_n$$

Below are the calculations to determine the x-values:

$$x_1 = f \cdot \phi_1 = 17, x_2 = f \cdot \phi_2 = -4, x_3 = f \cdot \phi_3 = 3/2, x_4 = f \cdot \phi_4 = 5/3$$

7.3 Calculation of f_a

$$f_a = x_1 \cdot \phi_1 + x_2 \cdot \phi_2 + x_3 \cdot \phi_3 + x_4 \cdot \phi_4 = \begin{bmatrix} 1.30556 & 6.22222 & -0.194444 \\ 6.97222 & 5.6667 & 5.47222 \\ 3.11111 & -0.555556 & 7.11111 \\ 3.66667 & 5.11111 & 7.66667 \end{bmatrix}$$

The matrix is not similar to the matrix it is supposed to approximate.

Exercise 8

8.1 Solution written in Matlab

```
load('assignment1bases.mat')

norms = {zeros(400,1), zeros(400,1), zeros(400,1)};
for i = 1:3
    e = {bases{i}(:,:,1); bases{i}(:,:,2); bases{i}(:,:,3); bases{i}(:,:,4)};
    for j = 1:size(stacks{1},3)
        u = stacks{1}(:,:,j);
        %Calculate the dot product of u on each basis
        x = [sum(dot(u, e{1})), sum(dot(u, e{2})), sum(dot(u, e{3})), sum(dot(u, e{4}))];
        u_p = zeros(size(x(1)));
        for k = 1:4
            u_p = u_p + x(k)*e{k};
        end
        v = u - u_p;
        %Calculate the l2-norm using the definition
        norms{i}(j) = sqrt(sum(dot(v,v)));
    end
end

means = [mean(norms{1}) mean(norms{2}) mean(norms{3})];
disp(means)
```

8.2 Image from each stack

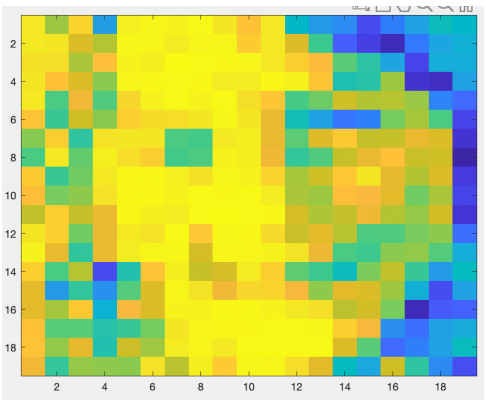

Figure 5: Image of Stack 1 which appears to consist of faces

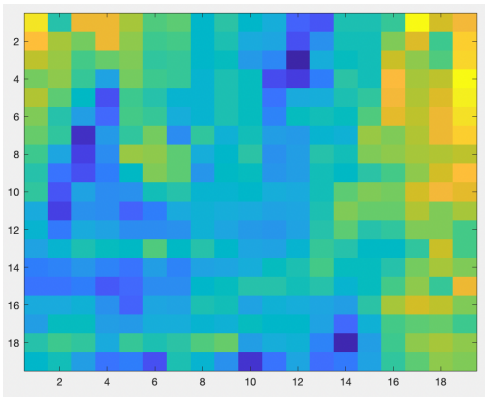

Figure 6: Image from Stack 2 which appears to consist of some sort of heat maps

8.3 Plots of the bases

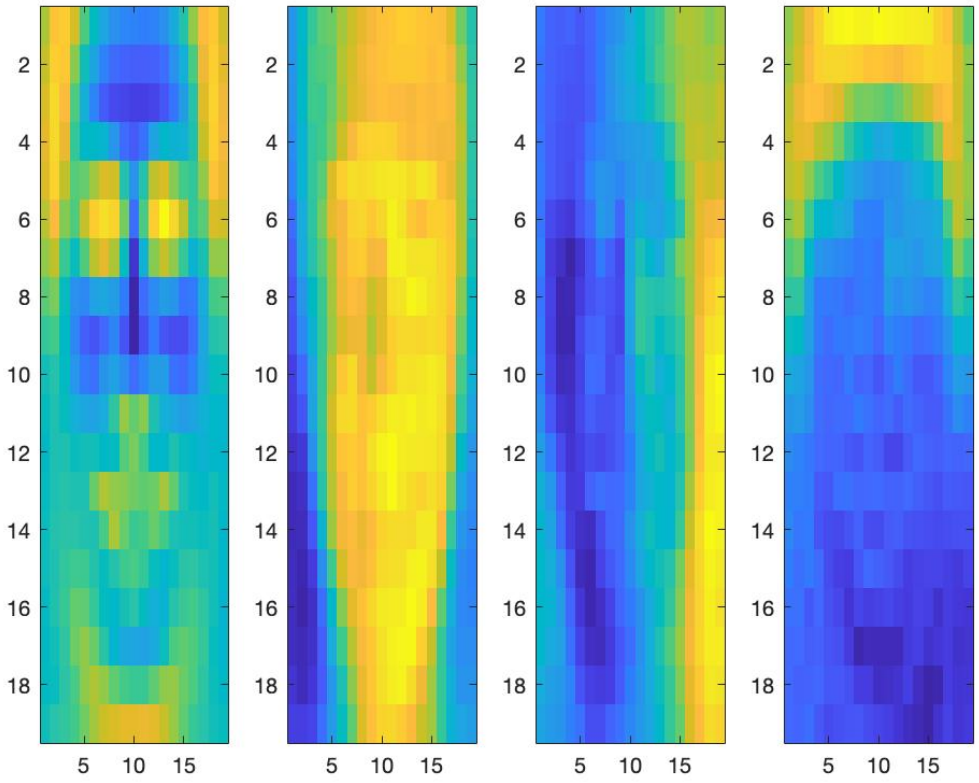

Figure 7: bases 1-4

These bases appear to resemble faces and should therefore work best for the images in stack 1 that also appeared to consist of faces.

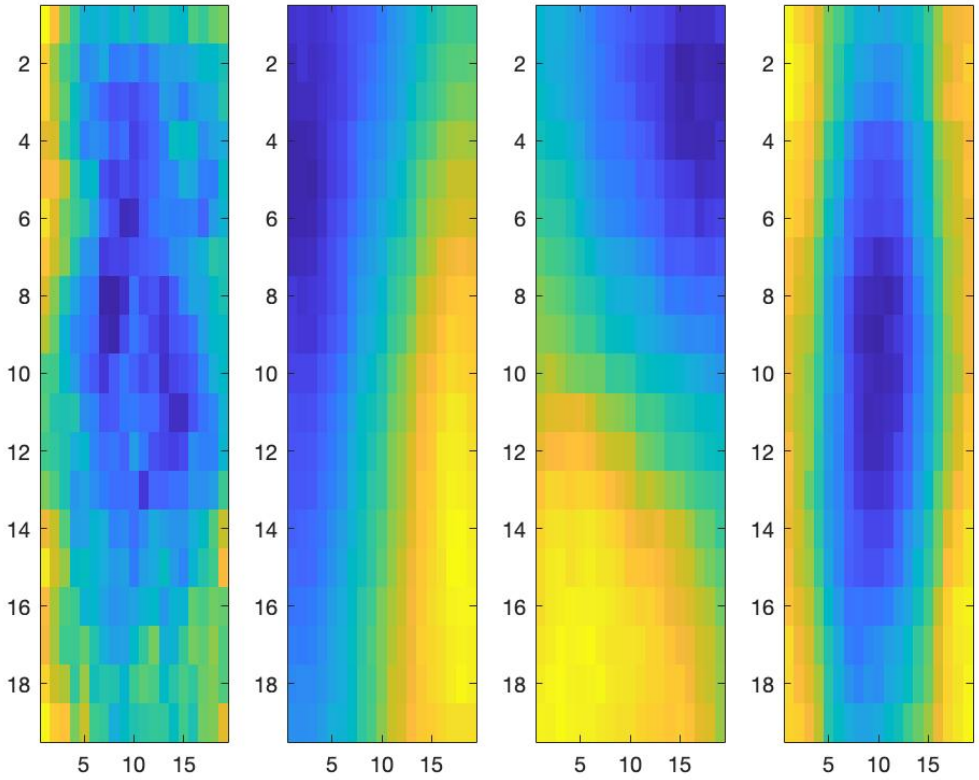

Figure 8: bases 5-8

These bases appear to resemble heat-maps just like the second stack does, so these bases should work best for the second stack.

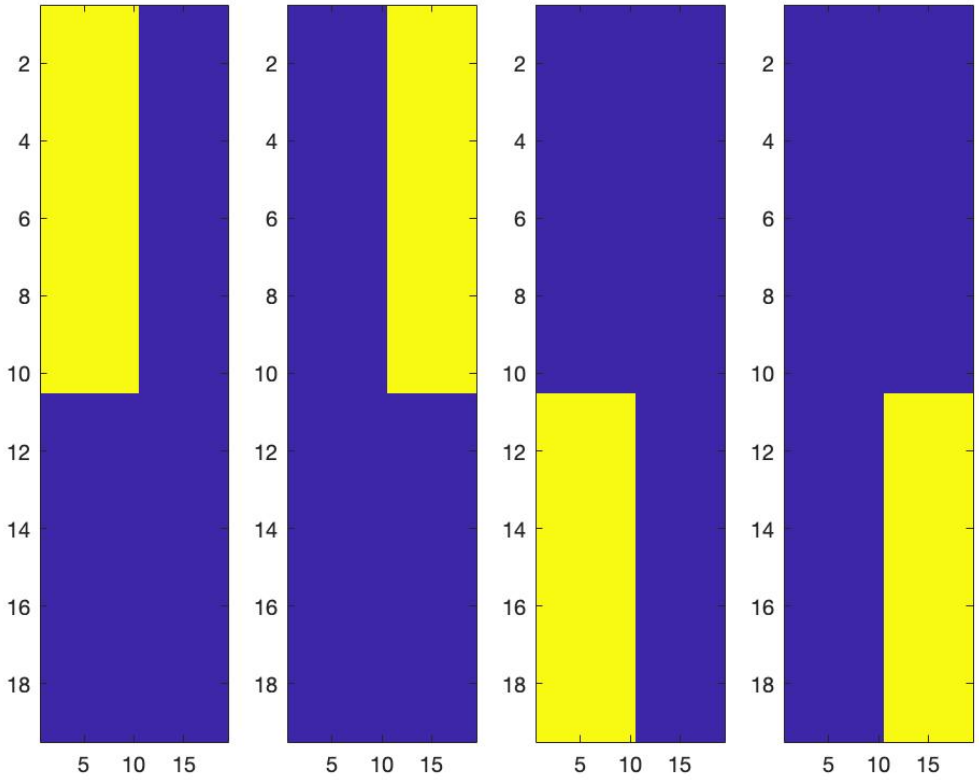

Figure 9: bases 9-12

These bases do not appear to resemble anything so probably won't work well for either set.

8.4 Print out of the error norms

	norm 1	norm 2	norm3
Stack 1	821.0271	860.4754	944.9009
Stack 2	795.1902	649.2013	697.3214

For the first stack I cant really see any norm which stands out, even though in theory I think the first 4 bases should work the best which they do but only with a slight margin. This is most likely due to the variety in the images in stack 1. For stack 2 the 5-8 bases were the best which was as expected, however there as well the second norm was just slightly better. This might be due to the more "uniform" appearance of the heat maps resembling gradients same as the stack of images.