Image Analysis, Assignment 1 Report

1 Image sampling

Sample the image evenly with 5*5 pixels, the lower left pixel is a sample from (0,0) and the upper right pixel is a sample from (1,1). So the 5*5 pixels' position are

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

And to quantify with 32 different gray levels from 0 to 31, so if the intensity of a point is from 0 to 1/32, the gray level is 0, from 1/32 to 2/32, then the gray level is 1,⋯,from 31/32 to 1, the gray level is 31. So we calculate the intensity of each points above.

Code:

```
x = [0 0.25 0.5 0.75 1
    0 0.25 0.5 0.75 1
    0 0.25 0.5 0.75 1
    0 0.25 0.5 0.75 1
    0 0.25 0.5 0.75 1]
y =[1 1 1 1 1
    0.75 0.75 0.75 0.75 0.75
    0.5 0.5 0.5 0.5 0.5
    0.25 0.25 0.25 0.25 0.25
    0 0 0 0 0]
f = x+y-2.*x.*y
g = floor(f/(1/32))
g(g==32) = 31
```

resulting matrix:

```
g = 5x5

    31    24    16     8     0
    24    20    16    12     8
    16    16    16    16    16
     8    12    16    20    24
     0     8    16    24    31
```

2 Histogram equalization

$$\int_0^s p_s(t)dt = \int_0^r p_r(t)dt$$

Take T so that $p_s(s) = 1$, s∈ $[0,1]$

$\int_0^r p_r(t)dt = \int_0^s 1\,dt$ = s = T(r) = $[t^{\frac{3}{2}}]_0^r = r^{\frac{3}{2}} - 0 = r^{\frac{3}{2}}$, r∈ $[0,1]$

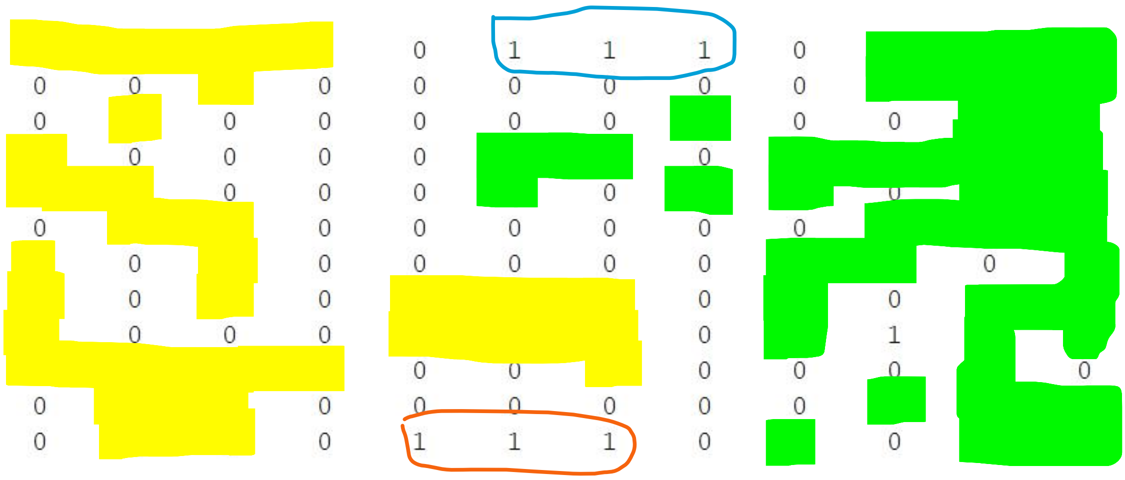
3 Neighborhood of pixels

Use the MATLAB to do the threshold.

Code and resulting g:

```
m = size(A,1);
n = size(A,2);
B = zeros(m,n);
for i=1:m
    for j=1:n
        if A(i,j)>1
            B(i,j)=1;
        else
            B(i,j)=0;
        end
    end
end
```

1	1	1	1	0	1	1	1	0	1	1	1
0	0	1	0	0	0	0	0	0	1	1	1
0	1	0	0	0	0	0	1	0	0	1	1
1	0	0	0	0	1	1	0	1	1	1	1
1	1	0	0	0	1	0	1	1	0	1	1
0	1	1	0	0	0	0	0	0	1	1	1
1	0	1	0	0	0	0	0	1	1	0	1
1	0	1	0	1	1	1	0	1	0	1	1
1	0	0	0	1	1	1	0	1	1	1	1
1	1	1	1	0	0	1	0	0	0	1	0
0	1	1	0	0	0	0	0	0	1	1	1
0	1	1	0	1	1	1	0	1	0	1	1



There are 4 8-connected components for g=1.

4 Segmentation part of OCR

Code

```
function S = im2segment(im)
```

```
A = imgaussfilt(im,0.5);% gaussion filter
S = cell(1,5);% we have 5 numbers
m = size(im,1);
n = size(im,2);
for i=1:m
     for j=1:n
        if j==1||j==n||i==1||i==m
        A(i,j)=0;
        end
     end
end

for i=1:m % threshold
    for j=1:n
        if im(:,1)==[2 7 5 0 0 0 0 10 18 7 0 0 6 16 10 0 0 8 0 20 0 14 0 14 0
10 15 10]'
        if A(i,j)>=26
            A(i,j)=1;
        else
            A(i,j)=0;
        end
        else
        if A(i,j)>=40
            A(i,j)=1;
        else
            A(i,j)=0;
        end
        end
    end
end

for i=2:m-1
    for j=2:n-1
        if A(i+1,j)==0&&A(i,j+1)==0&&A(i-1,j)==0&&A(i,j-1)==0
            A(i,j)=0;
        end
    end
end

BW= logical(A);
B = bwlabel(BW,8);% use this function to find 8-connected components
S{1} = zeros(m,n);
S{2} = zeros(m,n);
S{3} = zeros(m,n);
```

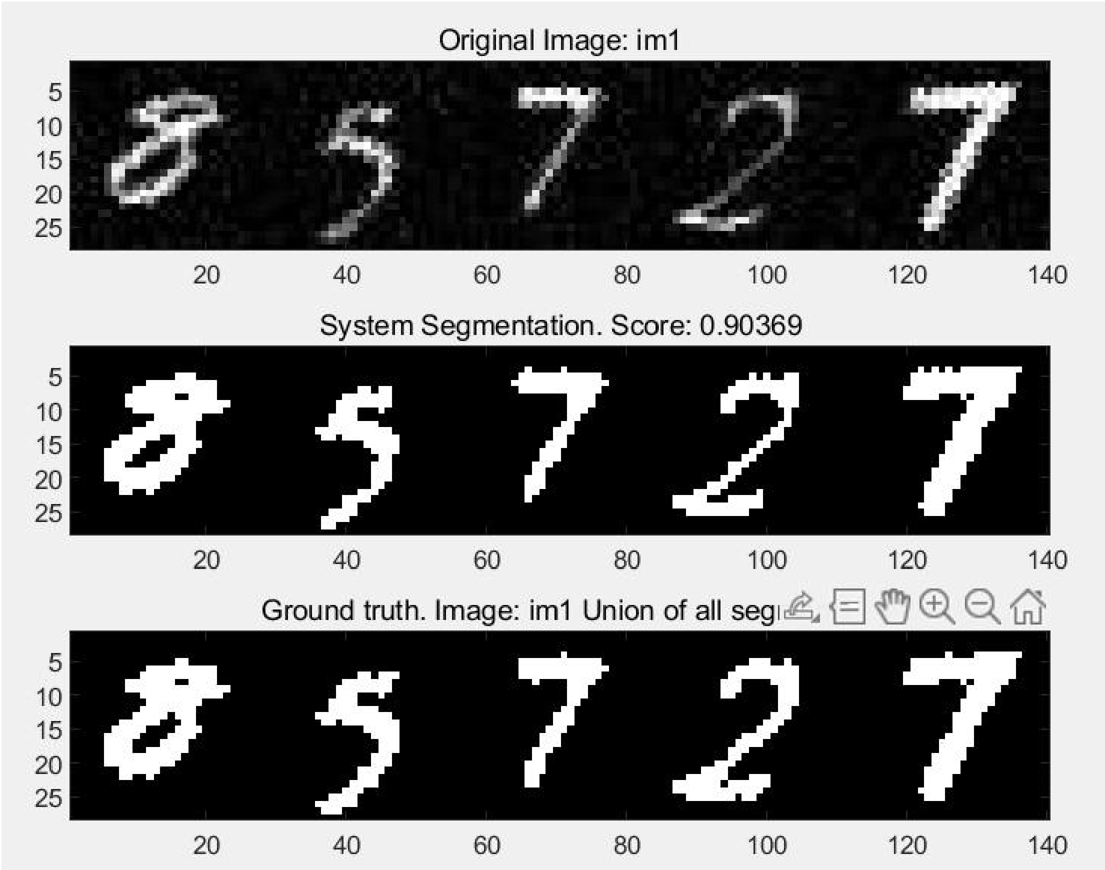
```
S{4} = zeros(m,n);
S{5} = zeros(m,n);
for i=1:m
    for j=1:n
        if B(i,j)==1
            S{1}(i,j)=1;% s1 is the first component and the first number
        elseif B(i,j)==2
            S{2}(i,j)=2;
        elseif B(i,j)==3
            S{3}(i,j)=3;
        elseif B(i,j)==4
            S{4}(i,j)=4;
        elseif B(i,j)==5
            S{5}(i,j)=5;
        end
    end
end
S={S{1},S{2},S{3},S{4},S{5}};
```

The jaccard scores for all segments in all images were

0.9512	0.8868	0.9302	0.7951	0.9379
0.9109	0.9424	0.8626	0.9658	0.9621
0.9310	0.9508	0.9380	0.9474	0.9333
0.7549	0.9137	0.9551	0.9478	0.9172
0.9187	0.9170	0.9424	0.8866	0.9448
0.9624	0.9298	0.9732	0.9527	0.9931
0.9461	0.8941	0.9204	0.9638	0.9250
0.9191	0.9598	0.7745	0.9188	0.8849
0.9268	0.9203	0.9432	0.9449	0.8901
0.8359	0.8750	0.8627	0.7574	0.8750

The mean of the jaccard scores were 0.91385

This is good!.



5 Dimensionality

In A the dimension k=2*3=6

Basis elements:

$$e1=\begin{bmatrix}1 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}\quad e2=\begin{bmatrix}0 & 1\\ 0 & 0\\ 0 & 0\end{bmatrix}\quad e3=\begin{bmatrix}0 & 0\\ 1 & 0\\ 0 & 0\end{bmatrix}\quad e4=\begin{bmatrix}0 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}\quad e5=\begin{bmatrix}0 & 0\\ 0 & 0\\ 1 & 0\end{bmatrix}\quad e6=\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 1\end{bmatrix}$$

In B the dimension k=1500*2000=3000000

The basis elements can be chosen by making every single element is 1 and the others are 0 each time, and there are 3000000 elements in total.

6 Scalar products and norm on images

The scalar product of two images is defined as $f \cdot g = \sum_{i=1}^{M}\sum_{j=1}^{N} \bar{f}(i,j)g(i,j)$

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

The norm of an image f is defined as

$||u||=\sqrt[2]{3*3+7*7+1+4*4}=5\sqrt{3}$

$||v||=1/2\sqrt{1+1+1+1}=1$

$||w||=1/2\sqrt{1+1+1+1}=1$

u · v=2+5.5=7.5

u · w=-1-1.5=-2.5

v · w=0

{v, w} is orthonormal because v · w=0, v · v=1, w · w=1

Let x1=u · v=7.5, x2=u · w=-2.5

The orthogonal projection is u1 =7.5v-2.5w=

[5 -5

-2.5 2.5]

$|u-u1|^2$=(u-u1) · (u-u1)=12.5

So the projection is not a good approximation, it is not close to u.

7 Image compression

φ1 · φ2=0 φ1 · φ3=0 φ1 · φ4=0

φ2 · φ3=0 φ2 · φ4=0 φ3 · φ4=0

φ1 · φ1=1 φ2 · φ2=1 φ3 · φ3=1 φ4 · φ4=1

Let f − fa ⊥the subspace spanned by {φ1, φ2, φ3, φ4} , so fa is the orthogonal projection

and |f − fa| 2 is as small as possible.

x1=f · φ1=50/3 x2=f · φ2=-4

x3=f · φ3=3/2 x4=f · φ4=2

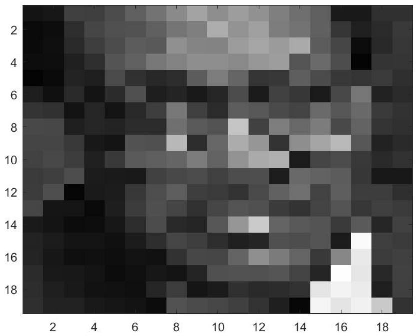
fa=x1φ1 + x2φ2 + x3φ3 + x4φ4=

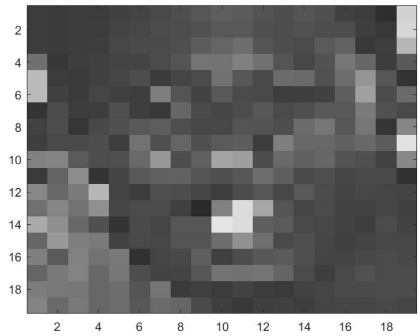
1.4167	6.2222	-0.0833
6.9722	5.5556	5.4722
2.8889	-0.6667	6.8889
3.5556	4.8889	7.5556

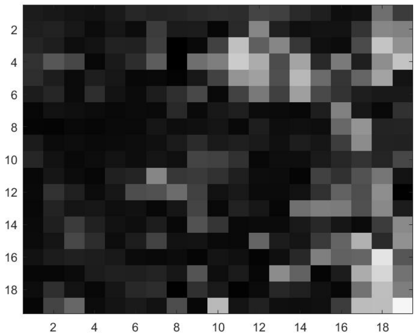
|f − fa| 2 = (f − fa) · (f − fa)=139.9722

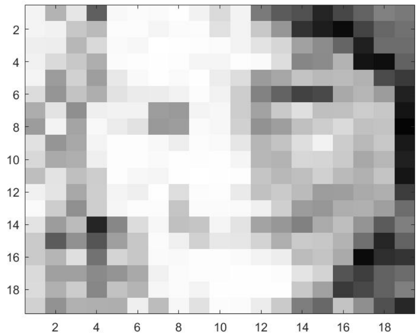
8 Image bases

```
function r = project(stacks,bases,k,i,j)% r is the norm of the difference
u = stacks{k}(:,:,j);% k is the number of stacks , j is number of images
e_1 = bases{i}(:,:,1);% i is the number of basis , each basis has 4 images
e_2 = bases{i}(:,:,2);
e_3 = bases{i}(:,:,3);
e_4 = bases{i}(:,:,4);
x_1 = sum(dot(u,e_1));% project the image onto a basis so x1 is scalar product
x_2 = sum(dot(u,e_2));% of u and each basis image
x_3 = sum(dot(u,e_3));
x_4 = sum(dot(u,e_4));
u_p = x_1 * e_1 + x_2 * e_2 + x_3 * e_3 + x_4 * e_4;% u_p is projection result
r = sqrt(sum(dot(u-u_p,u-u_p)));% r to the power of 2 =(u-up)·(u-up)
end
```

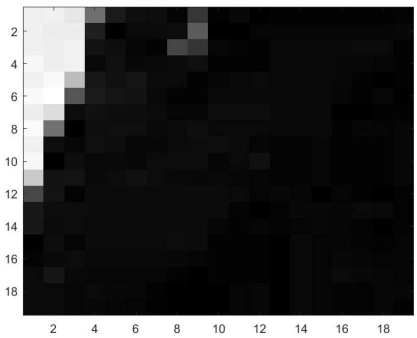


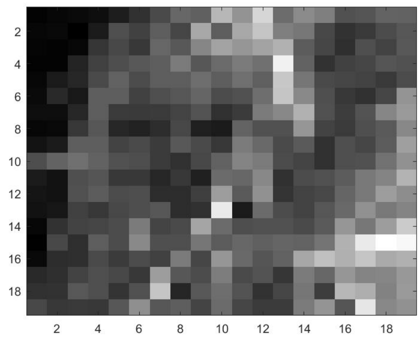


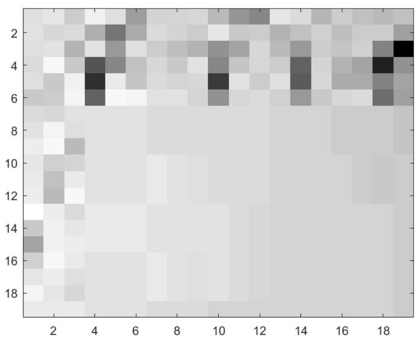


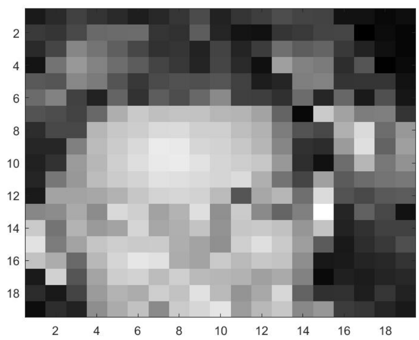


The images in stack1 look like human faces but very vague and with very low resolution.





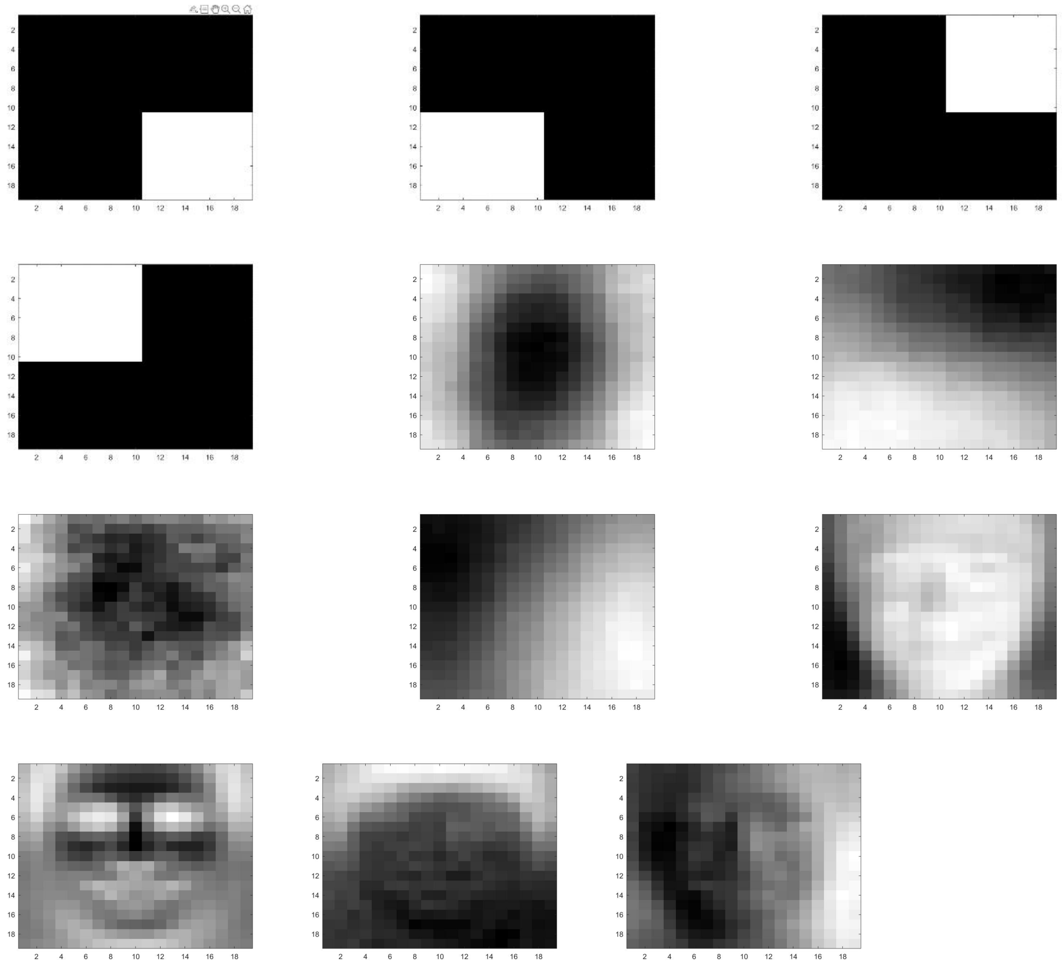




The images in stack2 have even lower resolution, sometimes they are almost black or white, I can't even tell they are human faces.

Plots of the four basis elements of each of the three bases:

Visual differences: bases 3 are most simple with only 4 squares.
Bases 1 and 2 are more complicated but the bases 1 have human face shape.



The mean of the error norms for the six combinations (two test sets against the three bases)

```
rm =

   821.0271   795.1902
   860.4754   649.2013
   944.9009   697.3214
```

Basis 1 work best for test set 1 and basis 2 work best for test 2 , because they have the least mean of error norms.