基于PCA的人脸识别

Enviroment & Usage

```
macOS 10.14.2
Matlab R2018b

% 首先运行random_gene.m脚本生成训练集和测试集
% 先选择适当的 k 值，然后运行training.m或者改进版本training_imp.m进行训练
% 运行testing.m进行测试
% 选择适当的 k 值及对应的子图行数和列数，运行eigenface_display.m
```

算法描述

利用 PCA 生成训练集的 Eigenface

> **Principal components analysis** 主成分分析，是一种分析、简化数据集的技术。用于减少数据集的维数，同时保持数据集中的对方差贡献最大的特征。

算法步骤

1、初始化一个训练集矩阵，每一行为不同训练数据在同一个维度的不同坐标，每一列代表一个训练数据。

在剑桥大学ORL人脸数据库中，随机选取每个人（共40个人）中的7张不同人脸（共10张）的平均图像作为训练数据，每张图像的像素点数目（98 x 115）为初始维数，形成初始矩阵 *B* （10,304 x 40）

2、进行零均值处理（方差假设为1，每列减去均值）

$B = B - E(B)$，其中 *E(B)* 为所有训练数据的均值，这一步减少了数据分布的分散性，有利用投影空间的构造效果。

3、求取协方差矩阵

协方差矩阵表示不同随机变量之间的相互关系，图像中也即求任意两个像素之间的关系。如果两个随机变量的协方差为正或为负，表明两个变量之间具有相关性，如果为零则表示两个变量不相关。通过计算协方差矩阵，可以获得不同像素之间的关系。且要生成投影子空间，需要一组正交的基向量，所以构造协方差矩阵 *C*（10,304 x 10,304），利用其对称性获得的特征向量都是正交的。

$$C = \frac{1}{N-1}BB^*\text{，}N\text{为训练数据的个数（}40\text{）}$$

4、求取协方差矩阵的特征值与特征向量，选择主成分

由奇异值分解（singular value decomposition）定理可知，对称矩阵可以分解成同一组特征向量来表示，一个特征向量对应一个特征值，特征值越大，表明该特征向量占有的信息量越大，其越适合作为投影子空间的基向量。故将特征值降序排列，并根据对应的特征向量选择前 *k* 个作为基向量构成投影子空间 *V* （10,304 x k）即为特征脸，这一步也可以消除线性相关的向量之间的冗余性。

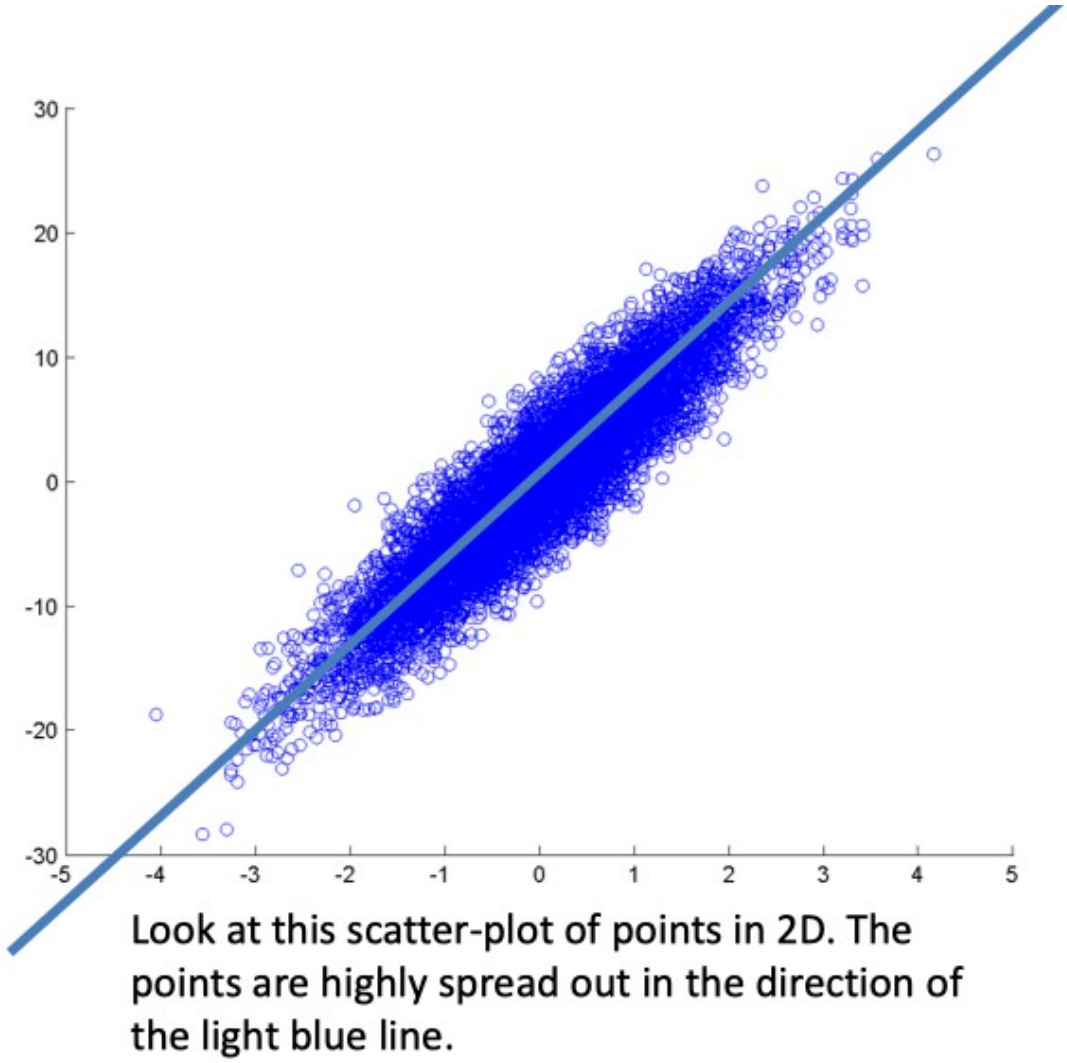
5、将原始矩阵投射到该子空间得到降维矩阵 *E* （k x 40），即每个训练数据的 *k* 个主要特征

$$E = V^*B$$

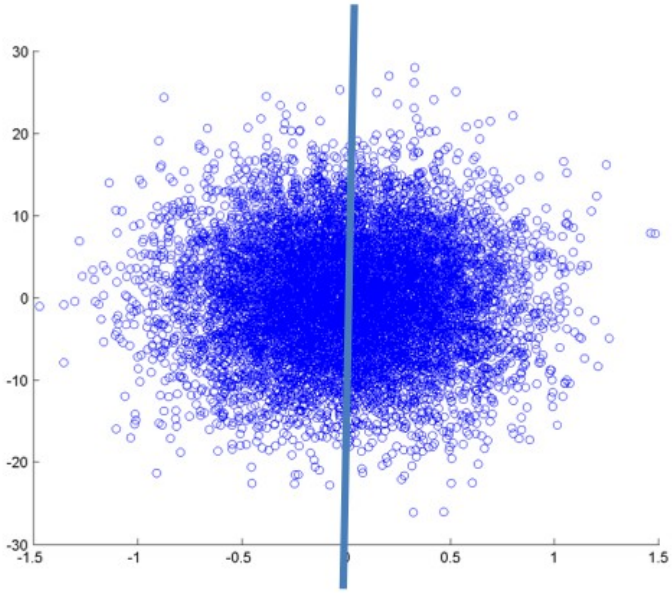
6、将得到的 *E(B)* ， *V* 和 *E* 保存

利用 PCA 对原始数据进行主成分提取，得到降维后的数据，用以下例子可以更直观的表示：

假设数据集为二维的，其散点图如下所示



由于数据集在Y轴坐标的方差较大，故可以将原始数据（x，y）降维成 y轴一维坐标，而忽略 x 轴的坐标



This is how the data would look if they were rotated in such a way that the major axis of the ellipse (the light blue line) now coincided with the Y axis. As the spread of the X coordinates is relatively insignificant (**observe the axes!**), we can approximate the data points by their projections onto the Y-axis (i.e. their Y coordinates alone!). This was not possible prior to rotation!

二范数最小匹配

> 将每个人剩余的三张图片作为测试集进行匹配

算法步骤

1、将测试图片转换为向量（10,304 x 1），并利用上一步保存的 *E(B)* 的进行零均值处理

2、利用 *V* 进行降维投影，得到 k 个主要特征的系数（k x 1）

3、与 *E* 中每一列计算二范数（即欧式距离），并得到最小值的下标即为匹配的类别

4、如果匹配的类别和测试图片所属的类别相同，更新正确数量

5、得到正确率 =（识别正确的图像数）/ 120

部分代码

提取主要特征

随机将数据集分为训练集和测试集

```
% random_gene.m
clear;

for x = 1:40
    % 每个目录随机选取7个作为训练样本，剩余3个作为测试样本
    idx = randperm(10);
    training_set(x,:) = idx(1:7);
    testing_set(x,:) = idx(8:10);
end

save random_gene.mat training_set testing_set
```

获得每类人脸的平均图像

```
for x = 1:40
     % temp_set d*7
     temp_set = [];
     for y = training_set(x,:)
          temp_mat = imread(['../att_faces/s',num2str(x),'/',num2str(y),'.pgm']);
          temp_mat = reshape(temp_mat,[d,1]); %将图片转化为一个列向量
          temp_set = [temp_set temp_mat];
     end
     % Img_Mat d*40
     Img_Mat = [Img_Mat mean(temp_set,2)];
end
```

每列减去均值

```
% differ_mat d*N
differ_mat = [];
img_mean = mean(Img_Mat,2);
% 40张平均图像
num_img = size(Img_Mat,2);

for i = 1:num_img
     temp_mat = double(Img_Mat(:,i)) - img_mean;
     differ_mat = [differ_mat temp_mat];
end
```

生成对称矩阵并求特征值和特征向量

```
% 对称矩阵 C_mat 10,304 x 10,304
C_mat = (1/(num_img-1)).*(differ_mat * differ_mat');
[eiv eic] = eig(C_mat);   %求取特征向量eiv以及特征值eic
```

```
% 对称矩阵 C_mat 40 x 40
C_mat = differ_mat' * differ_mat;
[eiv eic] = eig(C_mat);    %求取特征向量eiv以及特征值eic
```

降序排列特征值，并得到特征脸

```
% 降序排列特征值
[dd,ind] = sort(diag(eic),'descend');
eic_sort = eic(ind,ind);
eiv_sort = eiv(:,ind);
% 特征脸 Vk_mat d*k
Vk_mat = eiv_sort(:,1:k);
```

因为 *N* 小于 *d* ，所以 d x d 的协方差矩阵的秩最多为 *N-1* ，故可以采用改进的算法，消除一些线性相关列的冗余性。计算 40 * 40 协方差矩阵的特征值和特征向量，根据

$$\mathbf{V}=\mathbf{XW},\mathbf{X}\in R^{d\times N},\mathbf{W}\in R^{N\times N},\mathbf{V}\in R^{N\times N}$$

d

得到特征脸，可以验证对于 *W* 最大的 *k* 个特征向量对应于 *V* 的最大的特征向量

```
% Wk_mat N*k
Wk_mat = eiv_sort(:,1:k);

% 特征脸 Vk_mat d*k
Vk_mat = differ_mat * Wk_mat;

% normalize columns of Vk_mat
Vk_mat = normc(Vk_mat);
```

得到降维后的eigen-coefficents，即每类人脸提取的特征并保存相关中间结果

```
% Ei_Face k*N
Ei_Face = Vk_mat' * differ_mat ;      %得到投影子空间的坐标

save training.mat img_mean Vk_mat Ei_Face d
save eigenvector_sort.mat eiv_sort differ_mat img_mean
```

测试阶段

```
% testing.m
clear;
load training.mat;
load random_gene.mat;
```

```
tic;
% 记录识别正确数
correct_num = 0;
for x = 1:40
    for y = testing_set(x,:)
        temp_mat = imread(['../att_faces/s',num2str(x),'/',num2str(y),'.pgm']);
        % 显示测试图像
        % figure,
        % subplot(1,2,1),imshow(temp_mat);
        % title('Test Image');

        %************投影降维度测试图片*************************
        temp_mat = reshape(temp_mat,d,1);
        temp_mat = double(temp_mat) - img_mean;
        project_test = [];
        % project_test k*1
        project_test = Vk_mat' * temp_mat;

        %***************计算二范数*****************************
        com_dist = [];
        % Ei_Face k*40
        % i = 1:40
        for i = 1:size(Ei_Face,2)
            vec_dist = norm(project_test - Ei_Face(:,i),2);
            com_dist = [com_dist vec_dist];
        end
        %************筛选出距离最小的样本图片*********************
        [match_min,match_index] = min(com_dist);
        if match_index == x
            correct_num = correct_num+1;
        end

        % 显示识别图像，用于N=280时的全局训练
        % directories = ceil(match_index / 10);
        % subject = mod(match_index,10);
        % if subject == 0
        %     subject = 10;
        % end
        % recognize_img =
imread(['../att_faces/s',num2str(directories),'/',num2str(subject),'.pgm']);
        % subplot(1,2,2),imshow(recognize_img);
        % title('Recognized Image');
    end
end
t1 = toc;
disp(['识别正确的图像数: ',num2str(correct_num),'/120']);
disp(['识别系统的正确率: ',num2str(correct_num/120)]);
disp(['测试用时(s): ',num2str(t1)]);
```

显示图像结果和重构误差

```
% eigenface_display.m
clear;
load eigenvector_sort.mat;

k = 80;
row = 112;
col = 92;

% Vk_mat d*k
Vk_mat = eiv_sort(:,1:k);

% Ei_Face k*N
Ei_Face = Vk_mat' * differ_mat ;      %得到投影子空间的坐标

% 将 k 个特征脸组成全新的矩阵显示
space = 2; %间距的大小
subplot_row = 8; %子图行数
subplot_col = 10; %子图列数
immat = zeros(space*(subplot_row+1)+row*subplot_row,space*
(subplot_col+1)+col*subplot_col);
immat = uint8(immat);
for ii = 1:subplot_row
     for kk = 1:subplot_col
          index = (ii-1)*subplot_col+kk;
          temp_mat = Vk_mat(:,index);
          temp_mat = reshape(temp_mat,[row col]);
          temp_max = max(max(temp_mat));
          temp_min = min(min(temp_mat));
          temp_range = temp_max - temp_min;
          temp_mat = round(255*(temp_mat - temp_min)/temp_range);
          immat((ii-1)*row+1+ii*space:ii*(row+space),(kk-1)*col+kk*space+1:kk*
(col+space)) = temp_mat;
     end
end
figure
imshow(immat)


% 重构40张平均图像
% project_sample d*N
project_sample = [];
project_sample = Vk_mat * Ei_Face;
% 计算重构误差，二范数表示
disp(['k: ',num2str(k)]);
err=norm(project_sample-differ_mat,2);
```

```
disp(['重构误差（欧式距离）：',num2str(err)]);
project_sample = project_sample + img_mean;
% 显示重构的人脸平均图像
space = 2; %间距的大小
subplot_row = 5; %子图行数
subplot_col = 8; %子图列数
immat = zeros(space*(subplot_row+1)+row*subplot_row,space*
(subplot_col+1)+col*subplot_col);
immat = uint8(immat);
for ii = 1:subplot_row
     for kk = 1:subplot_col
          index = (ii-1)*subplot_col+kk;
          temp_mat = project_sample(:,index);
          temp_mat = reshape(temp_mat,[row col]);
          temp_max = max(max(temp_mat));
          temp_min = min(min(temp_mat));
          temp_range = temp_max - temp_min;
          temp_mat = round(255*(temp_mat - temp_min)/temp_range);
          immat((ii-1)*row+1+ii*space:ii*(row+space),(kk-1)*col+kk*space+1:kk*
(col+space)) = temp_mat;
     end
end
figure
imshow(immat)
```

性能测试表格

每类人脸随机选取7张作为训练集，下标如下（40 x 7）

7	**10**	**1**	**5**	**4**	**6**	**3**
7	4	8	6	9	2	10
1	9	3	7	5	2	10
9	1	7	10	5	4	8
6	9	3	1	7	5	4
6	7	5	3	8	4	2
10	2	6	5	7	4	3
6	4	10	9	5	7	2
9	1	5	2	4	10	8
10	8	9	6	3	1	5

3	5	8	10	4	6	2
9	5	7	1	8	6	4
8	9	7	2	6	10	5
10	5	8	1	6	3	7
6	10	5	9	4	3	1
10	7	1	9	8	4	2
4	7	5	9	2	8	6
10	3	4	2	7	8	6
8	2	3	5	7	9	10
6	7	2	3	1	8	4
1	7	9	6	2	4	8
3	4	5	9	8	2	10
6	8	9	10	3	7	4
1	2	8	7	10	9	6
4	3	2	8	6	1	7
8	7	9	6	1	4	10
4	8	3	2	7	1	6
5	6	8	9	2	7	4
10	6	8	7	5	9	3
1	9	2	5	4	3	10
6	10	2	9	3	1	7
9	2	10	1	8	5	4
4	2	10	1	7	9	6
2	1	8	4	10	9	6
4	1	6	5	2	10	8
10	4	2	8	7	3	5
2	7	6	3	5	8	4
2	4	7	9	3	5	6

10	8	6	2	4	7	9
7	5	4	3	10	6	9

剩余的作为测试集 （40 x 3）

8	**9**	**2**
1	5	3
6	8	4
3	2	6
10	2	8
9	1	10
8	9	1
8	1	3
3	6	7
7	4	2
1	7	9
2	10	3
4	1	3
9	2	4
7	8	2
3	6	5
10	3	1
1	5	9
6	4	1
9	5	10
3	5	10
7	1	6
5	1	2
5	4	3
5	10	9

2	3	5
9	5	10
1	10	3
2	4	1
6	8	7
8	5	4
7	3	6
5	8	3
7	5	3
3	7	9
9	1	6
10	1	9
8	10	1
3	1	5
2	8	1

采用提取 10,304 x 10,304 协方差矩阵最大的 *k* 个特征向量的方法 （其中训练时间包括每次计算特征值和特征向量的时间，实际每次计算的结果相同，只是投影子空间的基向量（eigenfaces）和降维后的eigen-coefficients不同）

k 值	**训练时间(s)**	**测试时间(s)**	**正确率(%)**
25	222.2428	1.1482	95.83
50	238.2676	1.1589	96.67
80	245.7515	1.248	96.67
100	221.8019	1.1639	96.67

采用提取 40 x 40 协方差矩阵最大的 *k* 个特征向量的方法（k<N，其中训练时间包括每次计算特征值和特征向量的时间）

k 值	**训练时间(s)**	**测试时间(s)**	**正确率(%)**
10	0.60125	0.25029	86.67
20	0.65003	0.28471	95
30	1.7987	0.2634	96.67

```
>> training_imp
k: 10
训练用时(s): 0.60125
>> testing
识别正确的图像数: 104/120
识别系统的正确率: 0.86667
测试用时(s): 0.25029
>> training_imp
k: 20
训练用时(s): 0.65003
>> testing
识别正确的图像数: 114/120
识别系统的正确率: 0.95
测试用时(s): 0.28471
>> training_imp
k: 30
训练用时(s): 1.7987
>> testing
识别正确的图像数: 116/120
识别系统的正确率: 0.96667
测试用时(s): 0.2634
```

根据以上结果可以发现：

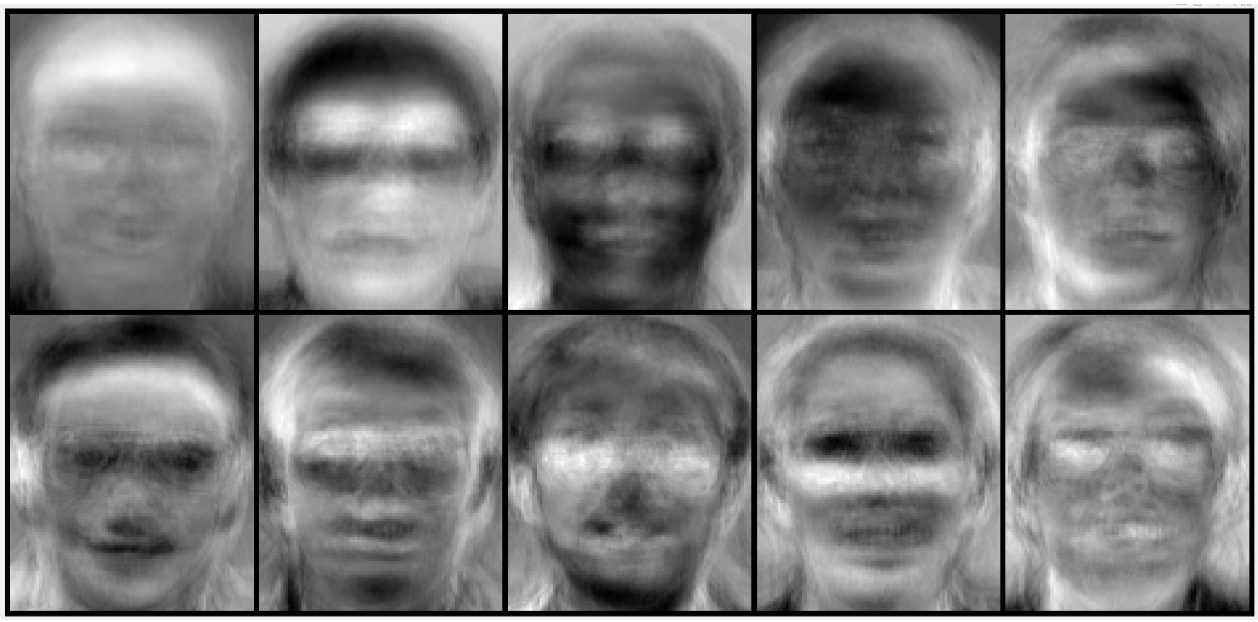
1. 由于计算特征值和特征向量的代价为 $O(d^3)$，$d_{\text{为训练数据维数}}$，所以使用改进后的方法 $O(N^3 + d * N^2)$，$N_{\text{为训练数据个数}}$，$d_{\text{为数据的维数}}$ 计算时间明显减少。
2. 随着 *k* 值的增加，训练时间发生了较小的增加，并且正确率也再不断提高。
3. 因为 *N* 小于 *d*，所以协方差矩阵的秩最多为 *N-1*，当 *k* 值到达一定值（*N-1*）后，增加 *k* 值并不会导致正确率的提高，而是保持在一个稳定的水平上。

图像结果

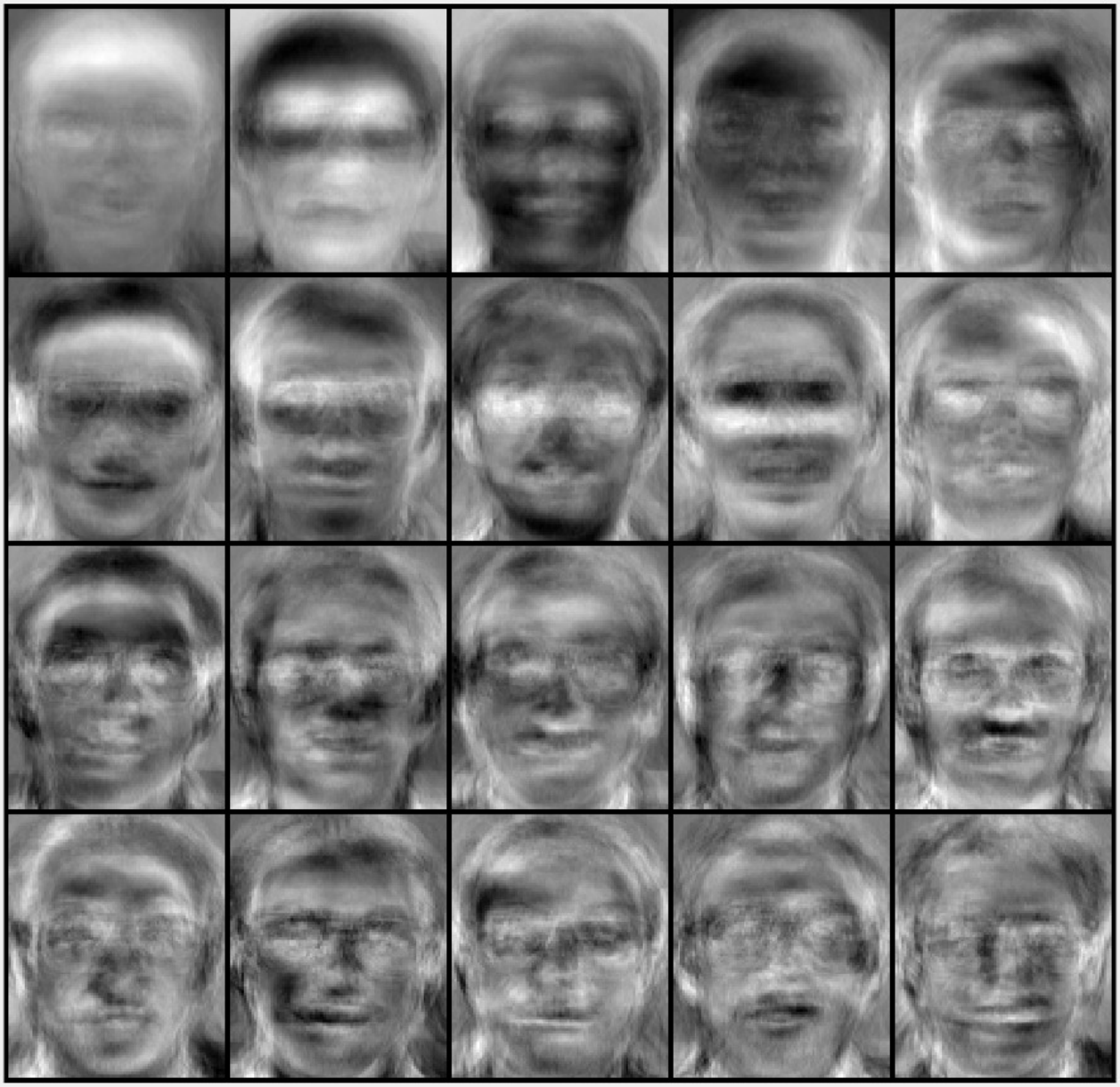
消除子图之间的空白区域

每类人脸7张训练图片的平均图像

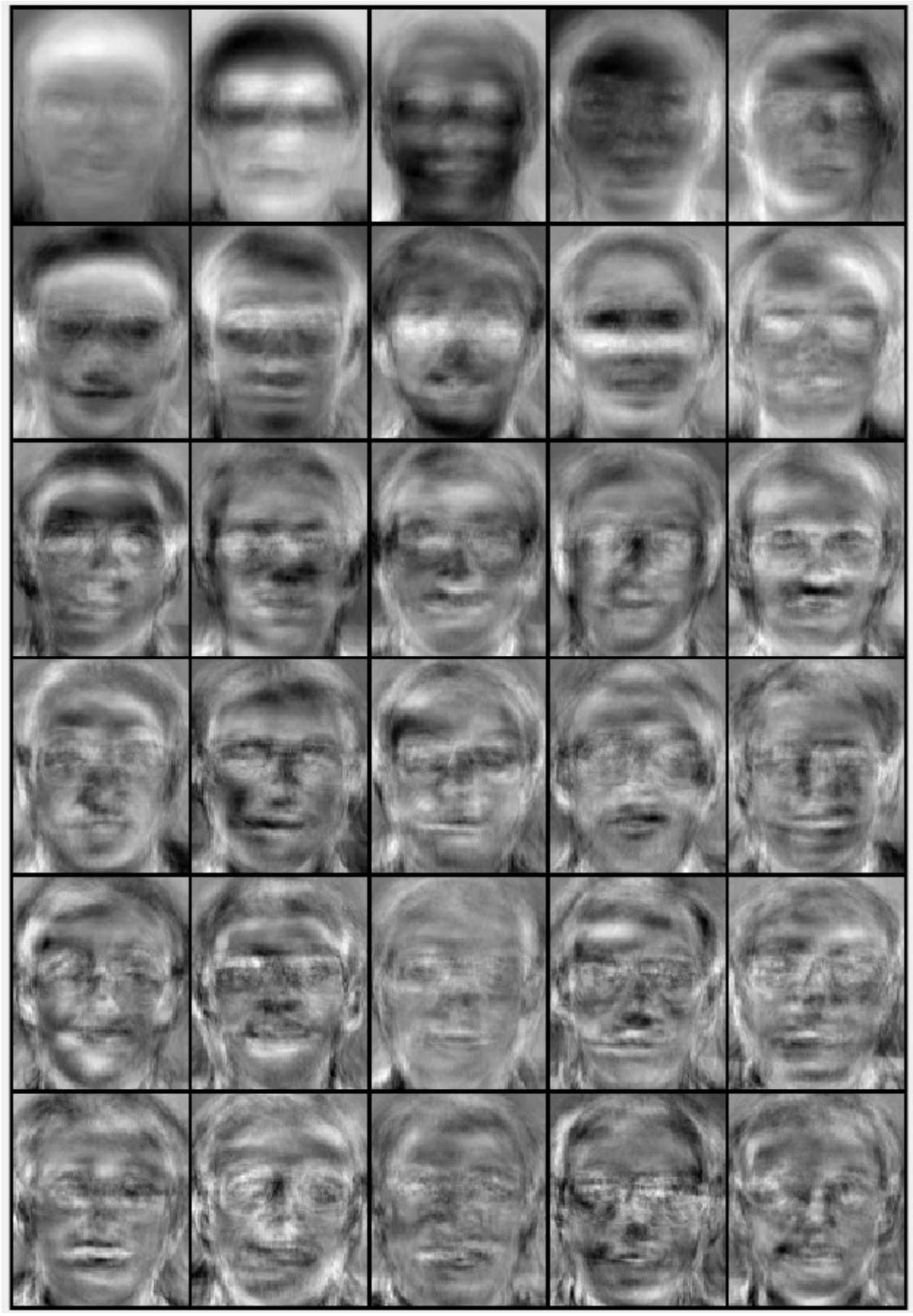
当 k = 10 时的特征脸



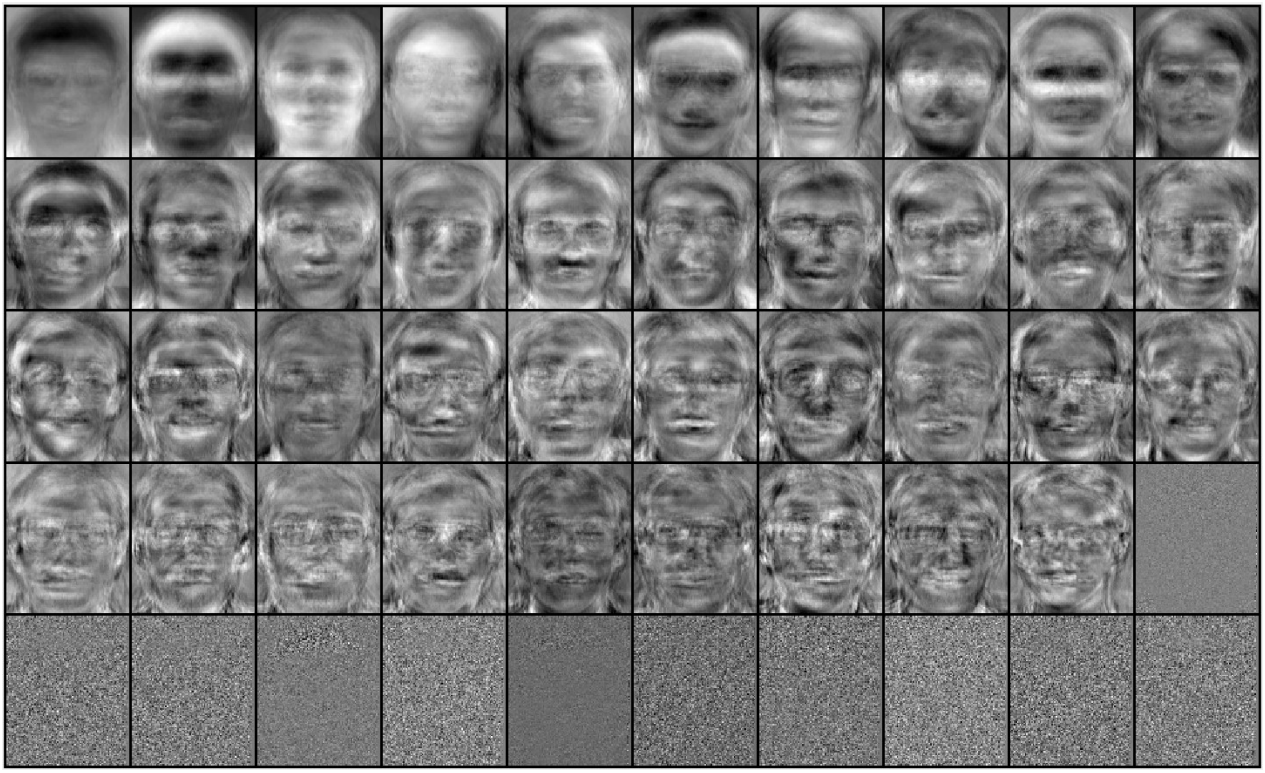
当 k = 20 时的特征脸



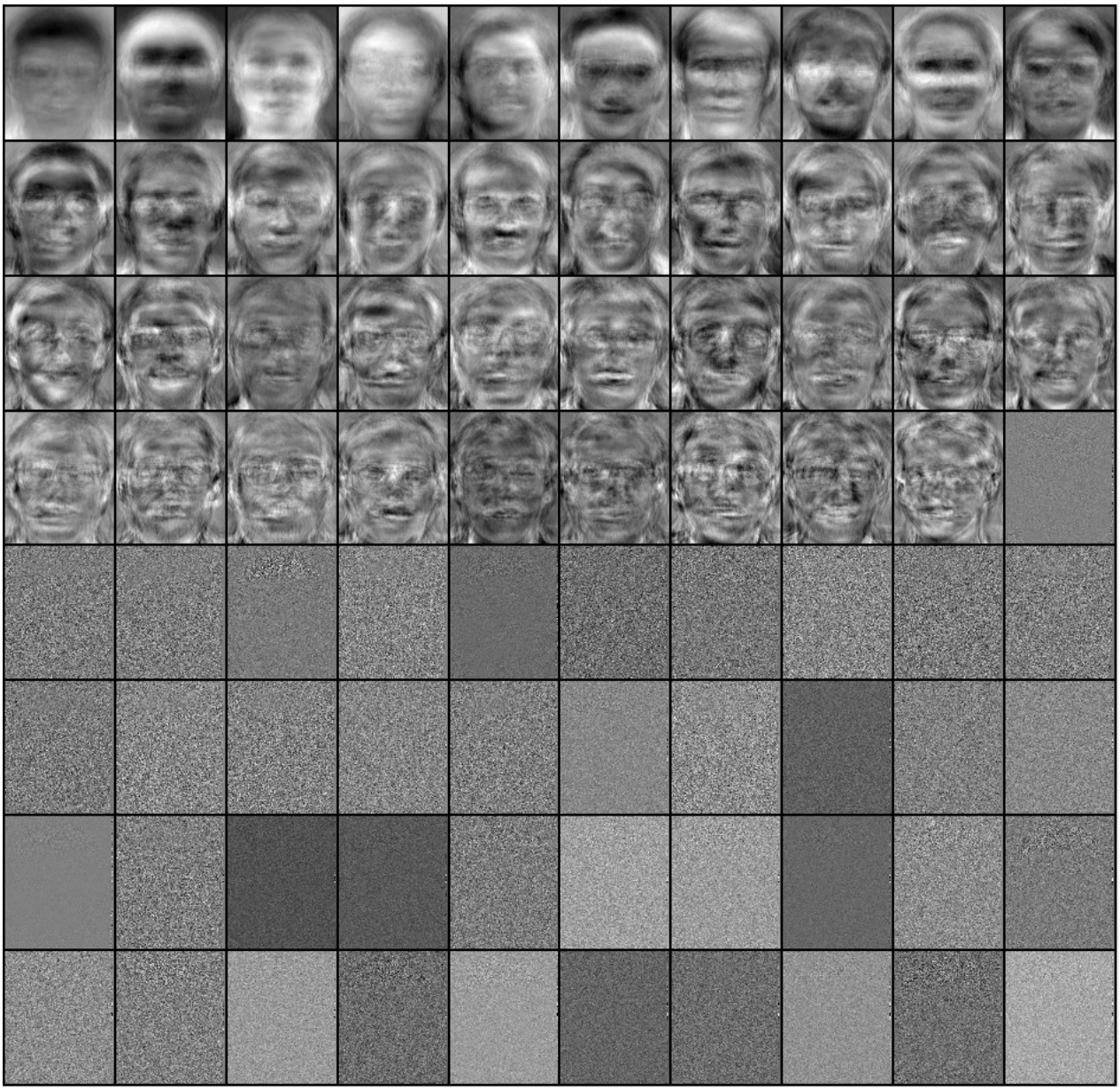
当 k = 30 时的特征脸



当 k = 50 时的特征脸



当 k = 80 时的特征脸



结果分析

1. 特征值越大所对应的特征向量，经过reshape后显示的特征脸，高频信息丢越严重，表示其为更加主要的特征。
2. 随着特征值的不断降低，特征脸包括了人脸中更多的高频信息。
3. 由于首先减去了所有数据的均值，非零的特征值对应的特征向量数量最多为 $min(N-1, d)$ 即 39 个。

当 k = 10 时的重构图像

当 k = 20 时的重构图像

当 k = 30 时的重构图像

当 k = 50 时的重构图像

当 k = 80 时的重构图像

```
k: 10
重构误差（欧式距离）: 2548.189
>> eigenface_display
k: 20
重构误差（欧式距离）: 1654.6156
>> eigenface_display
k: 30
重构误差（欧式距离）: 1142.0629
>> eigenface_display
k: 50
重构误差（欧式距离）: 7.2162e-11
>> eigenface_display
k: 80
重构误差（欧式距离）: 7.2607e-11
```

```
>> eigenface_display
k: 39
重构误差（欧式距离）: 7.528e-11
>> eigenface_display
k: 38
重构误差（欧式距离）: 759.7306
```

k	重构误差（二范数）
10	2548.189
20	1654.6156
30	1142.0629
38	759.730580290793
39	7.52797383764799e-11
50	7.21620586658817e-11
80	7.26069554719961e-11

结果分析

1. 将降维后的数据进行重构，随着 *k* 的增加，重构误差减少，重构效果更好。
2. 随着 *k* 的增加，重构的人脸图片会保留更多的细节，即高频信息。
3. 当 k 增加到 39（*N-1*）时，重构误差显著减少到可以忽略的程度，这是因为特征人脸库中最多只包括39个非零特征值对应的特征向量，每一个人脸可以由这39个特征脸线性组合得到。
4. 当 k 大于 39后，重构误差不会发生显著变化。

总结

Eigenface 提供了一种简单而廉价的方式来实现人脸识别：

1. 特征脸充分降低了人脸图像表示的复杂度，本次实验中，每张人脸可以由 N-1 个主成分表示，且信息损失极小。
2. 其训练过程完全自动，每个训练集只需计算一次特征值和特征向量
3. 降维后的数据可以减少测试过程中利用欧式距离匹配的计算代价
4. 可以处理大型数据集：当数据库较大时，即训练数据的个数N大于数据的维数d时，可以对维数为d的协方差矩阵进行特征提取。

缺点：

1. 当 *k* 取得较大时，图像的重构效果较好，但较小的 *k* 相当于图像去噪，有利于图像的识别。