课程实践 3： 基于 KL 变换的人像识别

实验目的：

通过对傅立叶变换原理的学习，选定一图像，编程实现 2D 的 F-变换，显示其频谱图像，相位图,实部，虚部图像,以及还原图；另选取高频区还原图像。

通过实验了解图像空间和频率之间的关系。

实验要求：

设 m 个人，每人 n 张人脸图像，即图像为 $x_{1,1},x_{1,2},\ldots,x_{1,n},x_{2,1},x_{2,2},\ldots,x_{2,n},\ldots,x_{m,1},x_{m,2},\ldots,x_{m,n}$，试用 KL 变换（PCA）或其他变换， 设计人脸特征提取算法，并设计人脸识别系统。附实验数据 yale 人脸识别（15 人 x 每人 11 幅）。

实验环境：MATLAB R2018a

原理和方法：

通常的经验是将数据集的 70%分为训练集，30%分为测试集。因此这里取每人前 8 张人脸图像做训练，后 3 张图像做测试，故训练集大小为 15×8=120 张，测试集大小为 15×3=45 张。

由于数据量较大，运行一遍代码耗时非常长，且很容易爆出警告“内存不足”而强制中止运行。不得不将原 128×128 的每张图像压缩为 64×64 以减少计算量，提高运行效率。

首先将每幅采样图像表示成列向量，按列依次排列，构成训练集矩阵 A 和测试集矩阵 B，取训练集矩阵 A 的协方差矩阵 C：

$$C=\mathrm{cov}(A)=E\{(A-\overline{A})(A-\overline{A})^T\}$$

令 e_i,λ_i 分别表示 C 的特征值和对应特征向量，即 $Ce_i=\lambda_i e_i, i=1,2,\ldots,120$。这些特征向量组成了训练集人脸特征子空间的正交基，训练集任意一幅图像都可以表示成这组正交基的线性组合，而且图像的信息主要集中在较大特征值对应的特征向量中，即丢掉较小特征值所对应的特征向量也不会影响图像重构后的质量。因此可以利用 PCA 降维，在保留绝大部分特征信息的前提下极大的减少计算量。

将特征值按从大到小排列，选择前 p 个最大特征值所对应的特征向量作为主成分。选取的特征值满足：

$$\frac{\sum_{i=1}^{p}\lambda_i}{\sum_{i=1}^{120}\lambda_i}\geq\alpha$$

经过试验，这里取 $\alpha=92\%$ 较为合适，得 p=29，即前 29 大特征向量的能量占总能量的 92%以上，因此可将原 120 维数据映射为 29 维数据而不失主要信息。这些主成分构成了投影矩阵 W：

$$W=[e_1,e_2,\ldots,e_p], p<<120$$

将训练样本 A 通过投影矩阵 W 投影到子空间中，就可以得到子空间特征，即 K-L 变换：

$$Y_A=W^TA \quad i.e. \quad y_i=W^TA_i, i=1,2,\ldots,120$$

同理再将测试样本 B 通过投影矩阵 W 投影到特征子空间中，得到测试样本的特征向量

Y_B，然后利用相似度度量准则计算测试样本的特征向量与训练样本的特征向量之间的相似度，选择相似度最大的训练样本所在的类别作为测试样本的类别。

常用相似度匹配算法：

①欧式距离

$$dist=(X_1-X_2)(X_1-X_2)^T=\sum(x_{1i}-x_{2i})^2$$

欧氏距离虽然很有用，但也有明显的缺点。

(i)它将样品的不同属性（即各指标或各变量）之间的差别等同看待，这一点有时不能满足实际要求。

(ii)它没有考虑各变量的数量级(量纲)，容易犯大数吃小数的毛病。所以，可以先对原始数据进行规范化处理再进行距离计算。

②标准化欧氏距离

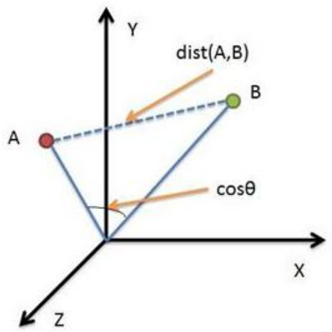
$$dist=(X_1-X_2)V^{-1}(X_1-X_2)^T=\sum\sigma_i(x_{1i}-x_{2i})^2$$

针对简单欧氏距离的缺点的一种改进方案，先将各个分量都“标准化”到均值、方差相等。

③余弦距离

$$dist=1-\frac{X_1X_2^T}{\|X_1\|\cdot\|X_2\|}=1-\cos\langle X_1,X_2\rangle$$

也称为余弦相似度，是用向量空间中两个向量夹角的余弦值作为衡量两个个体间差异的大小的度量。余弦取值范围为[-1,1]。余弦越大表示两个向量的夹角越小，余弦越小表示两向量的夹角越大。当两个向量的方向重合时余弦取最大值 1，当两个向量的方向完全相反夹角余弦取最小值-1。



从左图可以看出，欧氏距离衡量的是空间各点的绝对距离，跟各个点所在的位置坐标直接相关；而余弦距离衡量的是空间向量的夹角，更加体现在方向上的差异，而不是位置。如果保持 A 点位置不变，B 点朝原方向远离坐标轴原点，那么这个时候余弦距离 $\cos\theta$ 是保持不变的（因为夹角没有发生变化），而 A、B 两点的距离显然在发生改变，这就是欧氏距离和余弦距离之间的不同之处。

欧氏距离和余弦距离各自有不同的计算方式和衡量特征，因此它们适用于不同的数据分析模型：

欧氏距离能够体现个体数值特征的绝对差异，所以更多的用于需要从维度的数值大小中体现差异的分析，如使用用户行为指标分析用户价值的相似度或差异。

余弦距离更多的是从方向上区分差异，而对绝对的数值不敏感，更多的用于使用用户对内容评分来区分兴趣的相似度和差异，同时修正了用户间可能存在的度量标准不统一的问题（因为余弦距离对绝对数值不敏感）。

此外还有马氏距离、曼哈顿距离、切比雪夫距离、闵可夫斯基距离、马氏距离、汉明距离、杰卡德距离&杰卡德相似系数、相关系数&相关距离、信息熵等。

查阅文献知，欧氏距离和余弦距离的表现较好，因此本实验中采用这两种距离计算相似度，距离越短，相似度越高。

MATLAB 中自带的计算距离矩阵的函数有两个 pdist()和 pdist2()。前者计算一个向量自身的距离矩阵，后者计算两个向量之间的距离矩阵。具体用法可参阅 doc 文档。

最后计算两种距离算法的准确率，即测试集 45 个样本中，分对个数/总个数，比较两种相似度(距离)算法的优劣。并以表格形式输出识别结果。

实验代码和结果：

读取数据

```
A = [];  % 训练集
A_label = [];  % 训练集标签
B = [];  % 测试集
B_label = [];  % 测试集标签
for i = 1:15
    for j = 1:8 % 每人取8张做训练
        num = 10000 + i*100 + j;
        num = num2str(num);
        num(1) = '0';
        f = imread(strcat('yale/',num,'.bmp'));
        f = imresize(f,[64 64]);  % 压缩图片，提高运行速度
        A = [A,f(:)];
        A_label = [A_label,i];
    end
    for j = 9:11 % 每人取3张做测试
        num = 10000 + i*100 + j;
        num = num2str(num);
        num(1) = '0';
        f = imread(strcat('yale/',num,'.bmp'));
        f = imresize(f,[64 64]);
        B = [B,f(:)];
        B_label = [B_label,i];
    end
end
A = double(A); % 训练集，每列为一个样本
B = double(B); % 测试集，每列为一个样本
```

K-L变换 + PCA降维

```
C = cov(A'); % 总体散布矩阵
[V D] = eig(C);
D = diag(D); % 特征值
[Ds index] = sort(D,'descend'); % 降序排列特征值
for k = 1:length(Ds) % PCA 保留前p个特征值占总能量的92%
    if sum(Ds(1:k))/sum(Ds) > 0.92
        p = k;
        break
    end
end
Vs = V(:,index);
W = Vs(:,1:p); % 投影矩阵
Y_A = W'*A;  % 训练集特征提取
Y_B = W'*B;  % 测试集特征提取
```

相似度计算

```
[~,I1] = pdist2(Y_A',Y_B','seuclidean','Smallest',1);  % 法一，标准化欧氏距离,取最小
[~,I2] = pdist2(Y_A',Y_B','cosine','Smallest',1);  % 法二，余弦距离，取最小
B_testlabel1 = A_label(I1); % 法一预测测试集标签
B_testlabel2 = A_label(I2); % 法二
Accuracy1 = sum(B_testlabel1 == B_label)/length(B_label) % 法一准确率
Accuracy2 = sum(B_testlabel2 == B_label)/length(B_label) % 法二准确率
```

```
Accuracy1 =
    0.9778
Accuracy2 =
    0.9778
```

可知在此数据集上，两种距离的匹配效果是一样的，准确率都为 97.78%，即 45 个测试样本中仅有一个样本识别错误，我们进一步探究详细的识别结果。

输出具体识别结果

```
NUM = [];
for i = 1:15
    for j = 9:11
        num = 10000 + i*100 + j;
        num = int2str(num);
        num(1) = '0';
        NUM = [NUM;num];
    end
end
iftrue1 = (B_testlabel1 == B_label);
iftrue2 = (B_testlabel2 == B_label);
table = table(NUM,B_label',B_testlabel1',iftrue1',B_testlabel2',iftrue2',...
    'VariableNames',{'Name','real','predict1','iftrue1','predict2','iftrue2'})
```

table =
  45×6 table

Name	real	predict1	iftrue1	predict2	iftrue2
00109	1	1	true	1	true
00110	1	1	true	1	true
00111	1	1	true	1	true
00209	2	2	true	2	true
00210	2	2	true	2	true
00211	2	2	true	2	true
00309	3	3	true	3	true
00310	3	3	true	3	true
00311	3	3	true	3	true
00409	4	4	true	4	true
00410	4	4	true	4	true
00411	4	4	true	4	true
00509	5	5	true	5	true
00510	5	5	true	5	true
00511	5	5	true	5	true
00609	6	6	true	6	true
00610	6	6	true	6	true
00611	6	6	true	6	true
00709	7	7	true	7	true
00710	7	7	true	7	true
00711	7	7	true	7	true

00809	8	8	true	8	true
00810	8	13	false	13	false
00811	8	8	true	8	true
00909	9	9	true	9	true
00910	9	9	true	9	true
00911	9	9	true	9	true
01009	10	10	true	10	true
01010	10	10	true	10	true
01011	10	10	true	10	true
01109	11	11	true	11	true
01110	11	11	true	11	true
01111	11	11	true	11	true
01209	12	12	true	12	true
01210	12	12	true	12	true
01211	12	12	true	12	true
01309	13	13	true	13	true
01310	13	13	true	13	true
01311	13	13	true	13	true
01409	14	14	true	14	true
01410	14	14	true	14	true
01411	14	14	true	14	true
01509	15	15	true	15	true
01510	15	15	true	15	true
01511	15	15	true	15	true

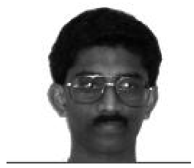
由上表可知，两种算法均把第八个人的第 10 张图像”00810.bmp”错误识别为第 13 个人。

```
>> I1(23)        >> I2(23)
ans =            ans =
   102              98
```

进一步探究可知，算法一欧氏距离是把 00810 识别为与 01306 最为相似，算法二余弦距离是把 00810 识别为与 01302 最为相似，我们来对比一下这三张图片：

00810

01302



01306

好吧，肉眼来看的话貌似确实没法理解为啥会被识别认为长得像……

参考文献：

[1] 数字图像处理与分析基础,黄爱民,中国水利水电出版社,2005 年 8 月

[2] 基于子空间特征提取的人脸识别
https://www.docin.com/p-1007151997-f3.html

[3] 几种相似性/距离(杰卡德距离和余弦距离)与其 matlab 实现
https://www.cnblogs.com/Maxnorm/p/8150388.html