

# 基于余类零空间与最近距离的人脸识别算法<sup>①</sup>

原豪杰, 孙桂玲, 郑博文, 李志晟

(南开大学 电子信息与光学工程学院, 天津 300350)

**摘要:** 提出了一种新的人脸识别算法, 即基于余类零空间与最近距离的人脸识别算法. 通过构建不同类别的人脸图像的余类零空间与子空间, 可以将不同类别的人脸最大化地区别出来. 本算法的主要思想在于: 测试图像与所属类别图像的子空间之间的距离最小, 而与所属类别的图像的余类零空间距离最大. 本算法基于 ORL 数据集与 AR 数据集进行了测试. 从这些人脸数据集上的测试结果可以看出, 本文提出的算法在 PCA 降维方法的基础上, 比一些常见的算法所使用的判别方式更有效, 如最近邻分类器(NN)所使用的最近距离判别方式、最近空间分类器(NS)所使用的最近空间距离判别方式、最近最远子空间分类器(NFS)所使用的最近最远空间距离判别方式等.

**关键词:** 余类零空间; 人脸识别; 最近空间; 分类器; 子空间

## Face Recognition Using Null-Space Combined with Nearest Space Distance Classifier

YUAN Hao-Jie, SUN Gui-Ling, ZHENG Bo-Wen, LI Zhi-Sheng

(College of Electronic Information and Optical Engineering, Nankai University, Tianjin 300350, China)

**Abstract:** This paper presents a new scheme for face recognition, namely face recognition using null-space combined with nearest space distance classifier. By constructing the null-spaces and the subspaces of different types of human face images, different types of human face images are distinguished at the maximum degree. This idea considers that a test image has the shortest distance from its own class-specific subspace and has the farthest distance from its own class-specific null-space. The proposed classifier is evaluated on ORL database and AR database. Experiments on these databases demonstrate that the proposed scheme is more effective than some discriminants used by common classifiers, such as nearest distance used by nearest neighbor classifier, nearest space distance used by nearest space classifier and nearest-farthest subspace distance used by nearest-farthest subspace classifier.

**Key words:** null-space; face recognition; nearest space; classifier; subspace

人脸识别是一项重要的技术. 我们可以借助计算机技术, 通过从人脸图像中提取出来的信息快速识别出不同的人. 人脸识别技术被广泛用于国家安全、军事安全以及公共安全中. 同时, 人脸识别的研究也将在认知科学、生理学、心理学以及其它相关学科中起到非常重要的作用.

目前已经有很大许多譬如计算机科学、数学科学、神经科学等众多科学的研究者加入到了人脸识别的研究中. 目前一些比较优秀的、成型的人脸识别算法包括: 基于 PCA 人脸识别算法、相关匹配算法 (correlation

matching algorithm)、基于神经网络的算法、基于小波变换的算法以及基于离散余弦变换的算法等.

基于 PCA 的人脸识别算法在人脸识别中有非常重要的地位<sup>[1,2]</sup>, 其算法步骤主要分为三个阶段, 即特征提取阶段、训练阶段、识别阶段. 许多基于 PCA 的算法侧重前两个阶段的研究, 如 2DPKA<sup>[3]</sup>与 BDPKA<sup>[4]</sup>以及另外一些改进的 PCA 算法<sup>[5-7]</sup>. 针对第三阶段的研究, J.Mi 提出了最远子空间分类器算法以及最近最远子空间分类器算法(NFS)<sup>[8]</sup>, 此外还有一些另外的分类器算法<sup>[9,10]</sup>. 本文提出了侧重于第三个阶段的一种新型

① 基金项目:高等学校博士学科点专项科研基金(20130031110032)

收稿时间:2016-07-16;收到修改稿时间:2016-08-25 [doi:10.15888/j.cnki.csa.005694]

分类器算法.

本文提出的算法主要分为三个步骤:

第一步利用 PCA 算法以及 PCA 的改进算法提取人脸图像的特征向量或特征矩阵, 构建属于不同类别的人脸子空间, 并计算测试图像与所有子空间的距离;

第二步基于不同类别人脸之间的关系构建不同类别的人脸余类零空间并计算测试图像与所有余类零空间的距离;

第三步根据第一步计算出的距离与第二步计算出的距离判别出测试图像的分类.

我们在两个人脸数据集上对算法进行了测试, 实验结果表明本文算法在各种测试环境下由于其它同类别算法. 余类零空间算法首先被用在 MIMO(多入多出技术)系统的预编码中<sup>[11,12]</sup>. 在 MIMO 系统中, 多用户的 MIMO 广播信道被分解成多个独立的单用户的广播通道, 系统可以在高速数据流中将多样性最大化. 结合 MIMO 预编码技术与人脸识别技术, 我们会发现二者之间很多的共同点. 在 MIMO 系统中, 我们将多用户通道分解成多个独立的单用户通道, 而在人脸识别中, 我们将不同类别的人脸数据分解到多个独立的子空间与余类零空间中. MIMO 系统中单用户中的多样性也与人脸识别中某一类别的人脸的多样性不谋而合.

本文第一部分首先介绍 PCA 及分类器的基本原理, 第二部分详细介绍本文提出的算法, 第三部分对算法进行试验并给出结果讨论.

## 1 PCA算法与分类器基本原理

### 1.1 PCA 算法基本原理

PCA 算法在图像压缩中是一种理想的正交变换. PCA 算法首先在原始图像中使用 K-L 变换得到一系列正交基底, 通过保留部分重要的正交基底, 可以得到图像的特征矩阵. PCA 算法极大地降低了图像的维度, 但其却几乎不影响人脸识别的性能.

假设在一个训练集中有  $N$  幅人脸图像, 其中每张图像可以用一个二维的  $m \times n$  矩阵表示. 对于一幅图像  $\mathbf{X}_i$ , 我们将矩阵的每一列附加到前一列中即可将一幅图像转换为一个维度为  $D$  的列向量(其中  $D=m \times n$ ).

将一个训练集中的  $N$  幅图像表示成矩阵  $\mathbf{X} = (\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N)$ , 其中每一个列向量表示一个图像. 矩阵  $\mathbf{X}$  的互相关矩阵可以表示为:

$$\mathbf{W}_L = \frac{1}{N} \sum_{i=1}^N (\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})^T \quad (1)$$

其中  $\bar{\mathbf{X}} = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i$  是所有训练图像的均值图像. 互相关矩阵的维度为  $D \times D$ .

计算互相关矩阵的特征值与特征向量, 在计算得到的所有特征向量中, 取前  $r$  个最大特征值对应的特征向量, 可得到特征脸空间  $\mathbf{Q} = (\mathbf{Q}_1, \mathbf{Q}_2, \dots, \mathbf{Q}_r) \subset \mathfrak{R}^{D \times r}$ . 这里取到的每一个特征向量被称为一个特征脸. 将训练集  $\mathbf{X}$  中的每一个图像向量映射到特征脸空间中, 即得到基于特征脸的特征图像  $\mathbf{Y}_i \in \mathfrak{R}^{r \times 1}$ ,  $\mathbf{Y}_i \in \mathfrak{R}^{r \times 1}$  可以表示为:

$$\mathbf{Y}_i = \mathbf{Q}^T \mathbf{Z}_i, \quad i = 1, 2, \dots, R \quad (2)$$

这里的  $\mathbf{Z}_i$  是  $\mathbf{X}_i$  的零均值化图像.

### 1.2 BDPCA 与 2DPCA 算法基本原理

传统 PCA 算法中, 图像首先被转换为一个向量. 而在 2DPCA 中则直接从原始图像矩阵中提取特征矩阵, 这大大减少了特征提取与训练的时间, 并且 2DPCA 的识别性能比 PCA 更好.  $N$  个训练图像可以表示为  $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ , 其中每个图像矩阵  $\mathbf{X}_i \in \mathfrak{R}^{m \times n}$  的维度为  $m \times n$ . 总散度矩阵表示为:

$$\mathbf{S}^{row} = \frac{1}{Nm} \sum_{i=1}^N (\mathbf{X}_i - \mathbf{M})^T (\mathbf{X}_i - \mathbf{M}) \quad (3)$$

其中矩阵  $\mathbf{M} = \frac{1}{N} \sum_{i=1}^N \mathbf{X}_i \in \mathfrak{R}^{m \times n}$  是所有训练图像的均值图像. 选取散度矩阵  $\mathbf{S}^{row}$  的前  $k_{row}$  个最大的特征值对应的特征向量构成映射矩阵  $\mathbf{W}_{row}$ , 使用公式  $\mathbf{Y}_i = (\mathbf{X}_i - \mathbf{M})\mathbf{W}_{row}$  将训练图像映射到  $\mathbf{W}_{row}$  即可从训练图像  $\mathbf{X}_i$  中得到特征矩阵  $\mathbf{Y}_i$ .

### 1.3 BDPCA 算法基本原理

BDPCA 通过公式(4)直接从训练图像矩阵  $\mathbf{X}_i$  中获取特征矩阵  $\mathbf{Y}_i$ :

$$\mathbf{Y}_i = \mathbf{W}_{col}^T \mathbf{X}_i \mathbf{W}_{row} \quad (4)$$

其中  $\mathbf{W}_{col}$  与  $\mathbf{W}_{row}$  分别为左映射矩阵与右映射矩阵.

给定训练图像集  $\{\mathbf{X}_1, \mathbf{X}_2, \dots, \mathbf{X}_N\}$ , 其中  $N$  为训练集样本的个数, 每幅图像矩阵的维度为  $m \times n$ . 首先定义行总散度矩阵, 可表示为:

$$\mathbf{S}_i^{row} = \frac{1}{Nm} \sum_{i=1}^N (\mathbf{X}_i - \bar{\mathbf{X}})^T (\mathbf{X}_i - \bar{\mathbf{X}}) \quad (5)$$

其中  $\bar{\mathbf{X}}$  为所有训练图像的均值图像矩阵. 选取  $\mathbf{S}_i^{row}$  矩阵的前  $k_{row}$  个特征值对应的特征向量可构成左映射矩阵  $\mathbf{W}_{row}$ . 同理, 可以定义列总散度矩阵  $\mathbf{S}_i^{col}$ , 可表示为:

$$\mathbf{S}_i^{col} = \frac{1}{Nn} \sum_{i=1}^N (\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})^T \quad (6)$$

选择  $\mathbf{S}_i^{col}$  矩阵的前  $k_{col}$  个特征值对应的特征向量可构成右映射矩阵  $\mathbf{W}_{col}$ 。最终，用公式  $\mathbf{Y}_i = \mathbf{W}_{col}^T \mathbf{X}_i \mathbf{W}_{row}$  即可从训练图像  $\mathbf{X}_i$  中得到特征矩阵  $\mathbf{Y}_i$ 。

事实上，BDPCA 是 2DPCA 的一般化的方法，而 2DPCA 可被看做 BDPCA 的一种特殊情况，将 BDPCA 中的  $\mathbf{W}_{col}$  以单位矩阵  $\mathbf{I}_m$  来代替即为 2DPCA。

文章从这里开始，文中提到的测试图像向量均指测试图像经 PCA 或其改进算法得到的特征向量。测试图像矩阵均指测试图像经 PCA 改进算法得到的特征矩阵。

#### 1.4 分类器

目前有许多基于 PCA 特征的分类器。一种常见的分类器为基于 Frobenius 距离的最近邻分类器(NN)。在 NN 分类器的基础上，有许多改进的分类器。

根据测试图像矩阵与所有训练图像矩阵的距离，即可判断出图像的分类。常见的距离定义有 Frobenius 距离，Yang 距离，Assembled Matrix 距离(AMD)。

给定两个图像矩阵  $\mathbf{A} = (a_{ij})_{k_{col} \times k_{row}}$  与  $\mathbf{B} = (b_{ij})_{k_{col} \times k_{row}}$ 。Frobenius 距离可表示为：

$$d_F(\mathbf{A}, \mathbf{B}) = \left( \sum_{i=1}^{k_{col}} \sum_{j=1}^{k_{row}} (a_{ij} - b_{ij})^2 \right)^{1/2} \quad (7)$$

Yang 距离<sup>[3]</sup>可以表示为：

$$d_Y(\mathbf{A}, \mathbf{B}) = \sum_{j=1}^{k_{row}} \left( \sum_{i=1}^{k_{col}} (a_{ij} - b_{ij})^2 \right)^{1/2} \quad (8)$$

AMD 距离可以表示为：

$$d_{AMD}(\mathbf{A}, \mathbf{B}) = \left( \sum_{i=1}^{k_{row}} \left( \sum_{j=1}^{k_{col}} (a_{ij} - b_{ij})^{p_1} \right)^{p_2 / p_1} \right)^{1/p_2} \quad (9)$$

Li 和 Lu 提出了 NN 分类器的一种改进分类器为最近线(NL)分类器<sup>[13]</sup>。

作为 NN 与 NL 分类器的扩展，Chien 和 Wu<sup>[14]</sup>提出了最近平面(NP)与最近空间(NS)分类器。下面这简单介绍 NS 分类器<sup>[15]</sup>的原理。

假设训练集有  $K$  种类别的人脸，每种类别包含  $p$  个图像向量。将这  $p$  个图像向量组合成一个矩阵代表第  $i$  个类别的子空间，可表示为：

$$\mathbf{A}_i = [\mathbf{a}_i^{(1)} \mathbf{a}_i^{(2)} \dots \mathbf{a}_i^{(p)}] \quad (10)$$

假设  $\mathbf{h}$  为测试图像向量，我们需要计算  $\mathbf{h}$  与由  $\mathbf{A}_i$  展成的空间的距离，即计算  $\mathbf{h}$  与其预测向量  $\hat{\mathbf{h}}_i$  的距离：

$$d_i = \|\mathbf{h} - \hat{\mathbf{h}}_i\|_2 \quad (11)$$

其中  $\hat{\mathbf{h}}_i$  可以简单地通过最小二乘法得到：

$$\hat{\mathbf{h}}_i = \mathbf{A}_i (\mathbf{A}_i^T \mathbf{A}_i)^{-1} \mathbf{A}_i^T \mathbf{h}_i \quad (12)$$

最后，我们在得到的  $K$  个距离中找到最小的距离  $d_i$ ， $d_i$  对应的类别即判定为测试图像向量对应的类别。

J.Mi 提出了最近最远子空间分类器(NFS)<sup>[8]</sup>，第一步，对于第  $i$  种类别，除第  $i$  种类别的  $N-1$  种训练集构建“leave-one-class-out”子空间：

$$\mathbf{B}_i = [\mathbf{A}_1 \mathbf{A}_2 \dots \mathbf{A}_{i-1} \mathbf{A}_{i+1} \dots \mathbf{A}_N] \quad (13)$$

测试图像向量与“leave-one-class-out”子空间的距离可表示为：

$$l_i = \|\mathbf{h}_i - \bar{\mathbf{h}}_i\|_2 \quad (14)$$

其中  $\bar{\mathbf{h}}_i = \mathbf{B}_i (\mathbf{B}_i^T \mathbf{B}_i)^{-1} \mathbf{B}_i^T \mathbf{h}_i$ 。最后，找到最小的值  $j_i$  对应的类别即判定为测试图像向量所属的类别。值  $j_i$  可表示为：

$$j_i = \frac{d_i}{l_i} \quad (15)$$

## 2 余类零空间与最近空间算法

回顾 NS 算法，算法将测试图像判别为最小距离  $d_i$  对应的类别。我们能否找到另外一个与  $d_i$ 、 $l_i$  无关的，判别效果更好的距离  $val_i$ ，我们将测试图像判别为最大的  $val_i$  对应的类别？进一步，结合距离  $d_i$  与  $val_i$ ，我们可以将测试图像判别为最大的  $val_i / d_i$ 。这里的距离  $d_i$  可以通过 NS 距离来计算。

通过距离  $val_i$  来对测试图像进行判别的算法与使用距离  $d_i$  来判别的 NS 算法是两种完全不同的算法。两个距离所表示的含义以及采取的判别策略有本质差别。

### 2.1 多次余类零空间距离分类器

在得到训练图像的特征矩阵  $\mathbf{Y}_i$  后，将矩阵中的每一列附加到前一列中，可将矩阵转换为一个列向量。所有的特征列向量可以表示为  $\mathbf{H}^T \in \mathfrak{R}^{D \times N}$ 。  $D$  为每个特征列向量的维度， $N$  为训练图像的个数。 $\mathbf{H}^T$  中的每一列均表示一个特征列向量。 $\mathbf{H}^T$  共包含  $K$  种类别，每种类别包括  $S$  张人脸图。将  $\mathbf{H}^T$  转置后得到  $\mathbf{H}$ ， $\mathbf{H}$  可表示为  $\mathbf{H} = [\mathbf{H}_1^T \mathbf{H}_2^T \dots \mathbf{H}_k^T]^T \in \mathfrak{R}^{N \times D}$ 。每一个子矩阵  $\mathbf{H}_i$  表示第  $i$  种类别的特征矩阵，并且  $\mathbf{H}_i \in \mathfrak{R}^{S \times D}$ 。矩阵  $\mathbf{H}$  的右逆矩阵可表示为：

$$\mathbf{H}_{right}^{\dagger} = \mathbf{H}^T (\mathbf{H} \mathbf{H}^T)^{-1} \quad (16)$$

其中  $\mathbf{H}_{right}^\dagger \in \mathfrak{R}^{D \times N}$ . 将  $\mathbf{H}_{right}^\dagger$  分解为  $K$  个子矩阵可得到  $\mathbf{H}_{right}^\dagger = [\mathbf{H}_{1,right}, \mathbf{H}_{2,right}, \dots, \mathbf{H}_{K,right}]$ . 每一个子矩阵  $\mathbf{H}_{i,right}$  与第  $i$  种类型的特征矩阵  $\mathbf{H}_i$  一一对应. 矩阵  $\mathbf{H}_{i,right}$  的大小为  $\mathbf{H}_{i,right} \in \mathfrak{R}^{D \times S}$ . 一般情况下, 当  $D > S$  时, 将每个子矩阵  $\mathbf{H}_{i,right}$  进行 QR 分解, 可得到:

$$\mathbf{H}_{i,right} = \mathbf{Q}_{i,right} \times \mathbf{R}_{i,right} \quad (17)$$

其中  $\mathbf{Q}_{i,right}$  是一个正交矩, 并且  $\mathbf{Q}_{i,right} \in \mathfrak{R}^{D \times D}$ .  $\mathbf{R}_{i,right}$  为一个上三角矩阵, 并且  $\mathbf{R}_{i,right} \in \mathfrak{R}^{D \times S}$ . 选取  $\mathbf{Q}_{i,right}$  的前  $S$  列可构建一个新的矩阵  $\mathbf{P}_{i,right} \in \mathfrak{R}^{D \times S}$ .

下面, 我们将会证明  $\mathbf{H}_{j(j \neq i)} \times \mathbf{P}_{i,right} = 0$  成立.

由右逆矩阵的特性可知  $\mathbf{H} \times \mathbf{H}_{right}^\dagger = \mathbf{I}$ , 进一步可以得到  $\mathbf{H}_i \times \mathbf{H}_{i,right} = \mathbf{I}$ ,  $\mathbf{H}_{j(j \neq i)} \times \mathbf{H}_{i,right} = 0$ . 这里的矩阵  $\mathbf{H}_{i,right}$  是  $\mathbf{H}_{right}^\dagger$  的子矩阵. 定义矩阵  $\mathbf{T}$  为  $\mathbf{H}_{j(j \neq i)} \times \mathbf{Q}_{i,right} = \mathbf{T}$ , 我们可以得  $\mathbf{T} \times \mathbf{R}_{i,right} = 0$ , 其中  $\mathbf{T} \in \mathfrak{R}^{S \times D}$ , 并且  $\mathbf{R}_{i,right} \in \mathfrak{R}^{D \times S}$  的对角线元素不为 0.  $\mathbf{T} \times \mathbf{R}_{i,right} = 0$  可以扩展为如下形式:

$$\begin{bmatrix} \mathbf{T}_{(1,1)} & \mathbf{T}_{(1,2)} & \dots & \mathbf{T}_{(1,S)} & \mathbf{T}_{(1,S+1)} & \dots & \mathbf{T}_{(1,D)} \\ \mathbf{T}_{(2,1)} & \mathbf{T}_{(2,2)} & \dots & \mathbf{T}_{(2,S)} & \mathbf{T}_{(2,S+1)} & \dots & \mathbf{T}_{(2,D)} \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{T}_{(S,1)} & \mathbf{T}_{(S,2)} & \dots & \mathbf{T}_{(S,S)} & \mathbf{T}_{(S,S+1)} & \dots & \mathbf{T}_{(S,D)} \end{bmatrix} \times \begin{bmatrix} \mathbf{R}_{(1,1)} & \mathbf{R}_{(1,2)} & \dots & \mathbf{R}_{(1,S)} \\ 0 & \mathbf{R}_{(2,2)} & \dots & \mathbf{R}_{(2,S)} \\ 0 & 0 & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{R}_{(S,S)} \\ 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 \end{bmatrix} = 0 \quad (18)$$

其中  $\mathbf{R}_{(1,1)}, \mathbf{R}_{(2,2)}, \dots, \mathbf{R}_{(S,S)}$  均不等于 0. 矩阵  $\mathbf{T}$  的左边部分可以表示为  $\mathbf{H}_{j(j \neq i)} \times \mathbf{P}_{i,right}$ .

将矩阵  $\mathbf{T}$  的第一行依次乘以右矩阵  $\mathbf{R}_{i,right}$  的每一列, 将会依次得到  $\mathbf{T}_{(1,1)} = \mathbf{T}_{(1,2)} = \dots = \mathbf{T}_{(1,S)} = 0$ . 同理可得  $\mathbf{T}_{(2,1)} = \mathbf{T}_{(2,2)} = \dots = \mathbf{T}_{(2,S)} = 0$ . 最终, 我们将会得到  $\mathbf{T}$  的左边部分为 0, 即:

$$\mathbf{H}_{j(j \neq i)} \times \mathbf{P}_{i,right} = 0 \quad (19)$$

我们将  $\mathbf{P}_{i,right}$  称作第  $i$  类人脸图像的余类零空间 (此零空间与除第  $i$  类外余下类别人脸图像乘积为 0). 计算得到所有  $K$  类人脸图像的余类零空间后, 可得到余类零空间矩阵  $\mathbf{P} = [\mathbf{P}_{1,right}, \mathbf{P}_{2,right}, \dots, \mathbf{P}_{K,right}]$ ,  $\mathbf{P}$  的大小为  $\mathbf{P} \in \mathfrak{R}^{D \times N}$ .

简明地概括余类零空间的性质为: 第  $i$  类人脸的余类零空间与所有不属于第  $i$  类人脸的图像矩阵相乘

的结果为零, 而与第  $i$  类人脸的图像矩阵相乘的结果不为零.

进一步可推导出来, 由第  $i$  类人脸的图像矩阵线性组合成的图像矩阵与所有不属于第  $i$  类人脸的余类零空间相乘的结果为零, 而与第  $i$  类人脸的余类零空间相乘的结果不为零.

事实上, 来自同一个类别的人脸图像以很高的概率属于同一个子空间, 即同一个类别中的人脸图像以很高的概率为这一类别中人脸图像的线性组合. 如果一个向量  $\mathbf{h}$  是  $\mathbf{H}_{j(j \neq i)}$  中向量的线性组合, 根据公式(19)可得到  $\|\mathbf{h} \times \mathbf{P}_{i,right}\|_2 = 0$ . 这里的  $\|\cdot\|_2$  为求矩阵的二范数. 所以, 如果一个测试图像向量  $\mathbf{h}$  属于第  $j$  种类型, 那么  $\|\mathbf{h} \times \mathbf{P}_{i,right}\|_2$  的值将远远小于  $\|\mathbf{h} \times \mathbf{P}_{j,right}\|_2$ .

在得到余类零空间矩阵  $\mathbf{P}$  后, 计算测试图像向量  $\mathbf{h}$  与所有余类零空间的零空间距离, 可得到:

$$Val_i = \|\mathbf{h} \times \mathbf{P}_{i,right}\|_2 \quad (20)$$

不同的  $i$  表示不同的余类零空间.  $Val_i$  表示第  $i$  个零空间距离. 在所有的  $Val_i$  中找到最大的  $Val$ ,  $Val$  对应的类型即判定为测试图像向量所属的类型.

余类零空间距离算法的流程如图 1 所示.

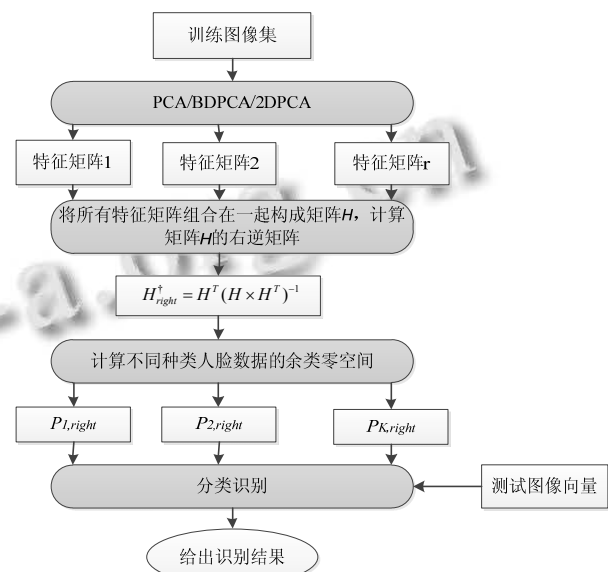


图 1 余类零空间距离算法流程图

当人脸数据集的数据量比较小时, 例如总共有 20 种类型的人脸图像, 每种类型的人脸包含 14 张人脸向量, 每张人脸向量的维度为 300. 前文提出的一次余类零空间距离分类器(null-space distance classifier 简称 NSD)识别效果很好. 当人脸数据集的数据量较大时,

例如, AR 人脸数据集总共包含 100 种人脸图像, 当在这样大的数据集上运用 NSD 算法时, 算法结果较差.

人脸图像的维度较大时, 意味着矩阵  $\mathbf{H}$  的行数较大, 计算右逆矩阵时  $\mathbf{H}^T(\mathbf{H}\times\mathbf{H}^T)^{-1}$  的复杂度将很大. 同时, 在计算过程中, 很容易出现奇异矩阵, 奇异矩阵的出现将导致计算得到的结果有很大的偏差. 在这种情况下, 我们将不能使用上述提到的算法.

本文在上面提出的 NSD 算法的基础上提出了多次余类零空间距离分类器算法(multi-time null-space distance classifier 简称 MNSD). MNSD 算法不仅解决了 NSD 算法不适合大数据集的问题, 而且相较于 NSD, 降低了算法的复杂度. MNSD 算法将不同类型的人脸数据集分成  $P$  组, 每组包括  $Q$  种类型的人脸数据.  $\mathbf{H}$  可表示为:

$$\mathbf{H} = [\mathbf{H}_{1,1}^T \mathbf{H}_{1,2}^T \dots \mathbf{H}_{1,Q}^T, \dots, \mathbf{H}_{P,1}^T \mathbf{H}_{P,2}^T \dots \mathbf{H}_{P,Q}^T]^T \in \mathfrak{R}^{N \times D} \quad (21)$$

$\mathbf{H}_t = [\mathbf{H}_{t,1}^T \mathbf{H}_{t,2}^T \dots \mathbf{H}_{t,Q}^T]^T \in \mathfrak{R}^{(S \times Q) \times D}$  ( $1 \leq t \leq P$ ) 表示其中的一组. 对包括 100 类人脸图像的 AR 数据集, 我们可以将其分为 10 组, 每组包含 10 类人脸图像, 即  $Q$  和  $P$  的值分别为 10 和 10. 在将原有数据集进行分组后, 在每一组中使用 NSD 算法, 选择出每一组中与测试图像向量最匹配类别. 调用  $P$  次 NSD 算法后, 将得到  $P$  类与测试图像向量最匹配的类别. 下一步, 将这  $P$  类人脸特征向量组合成一个新的组, 新的组可以表示为:

$$\mathbf{H}^{new} = [\mathbf{H}_{1,max_i}, \mathbf{H}_{2,max_i}, \dots, \mathbf{H}_{P,max_i}] \quad (22)$$

$\mathbf{H}^{new}$  的大小为  $\mathbf{H}^{new} \in \mathfrak{R}^{(S \times P) \times D}$ , 其中  $\mathbf{H}_{t,max_i} \in \mathfrak{R}^{S \times D}$  ( $1 \leq t \leq P$ ) 为组  $\mathbf{H}_t = [\mathbf{H}_{t,1}^T \dots \mathbf{H}_{t,Q}^T]^T$  中与测试图像向量最匹配的类别.

在这个新的组上使用 NSD 算法, 即可判断出测试图像类别. 如果这个新的组的维度仍然较大, 我们可以对这个新的组继续分割, 然后重复上面的过程即可. MNSD 算法的流程如表 1 所示.

表 1 MNSD 算法

步骤	操作
	将训练集的人脸特征图像分为 $P$ 个组, 每个组包含 $Q$ 类人脸特征图像, 每一类包括 $S$ 张图像.
1	$\mathbf{H} = [\mathbf{H}_{1,1}^T \mathbf{H}_{1,2}^T \dots \mathbf{H}_{1,Q}^T, \dots, \mathbf{H}_{P,1}^T \mathbf{H}_{P,2}^T \dots \mathbf{H}_{P,Q}^T]^T \in \mathfrak{R}^{N \times D}$ 其中 $\mathbf{H}_t = [\mathbf{H}_{t,1}^T \mathbf{H}_{t,2}^T \dots \mathbf{H}_{t,Q}^T]^T$ 代表第 $t$ 组数据.
2	对测试图像向量 $\mathbf{h}$ , 在每一组 $\mathbf{H}_t$ 中使用 NSD 算法选取每一组中与 $\mathbf{h}$ 最匹配的类别 $\mathbf{H}_{t,max_i}$ .
3	将每一组中与 $\mathbf{h}$ 最匹配的类别组合为一个新的矩阵

步骤	操作
	$\mathbf{H}^{new} = [\mathbf{H}_{1,max_i}, \mathbf{H}_{2,max_i}, \dots, \mathbf{H}_{P,max_i}]$ .
4	如果 $\mathbf{H}^{new}$ 中的类别数大于 $Q$ , 返回步骤 1 继续迭代过程, 如果 $\mathbf{H}^{new}$ 中的类别数小于 $Q$ , 使用 NSD 算法在 $\mathbf{H}^{new}$ 中判别出 $\mathbf{h}$ 所属的类别即可.

## 2.2 余类零空间与最近空间距离分类器

在 MNSD 算法基础上, 进一步, 我们可以利用  $Val_i$  与  $d_j$  结合起来进行分类. 新的距离被定义为  $Val_i / d_i$ . 所有  $Val_i / d_i$  中, 最大的对应的类别即判定为测试图像向量所属的类别. 我们将此算法称为余类零空间与最近空间距离算法(null-space combined with nearest distance method 简称 NSND). 如果使用 NS 算法计算  $d_j$ , 我们称算法为 NSND(结合 NS). NSND(结合 NS)算法流程如表 2 所示.

表 2 NSND (结合 NS)

步骤	操作
	对每一类特征矩阵 $\mathbf{H}_i$ 以及测试图像向量 $\mathbf{h}$ , 计算 $\mathbf{h}$
1	在由 $\mathbf{H}_i$ 扩展生成的子空间上的映射 $\hat{\mathbf{h}}_i = \mathbf{A}_i(\mathbf{A}_i^T \mathbf{A}_i)^{-1} \mathbf{A}_i^T \mathbf{h}$ (where $\mathbf{A}_i = \mathbf{H}_i^T$ ).
2	计算 $\mathbf{h}$ 与 $\hat{\mathbf{h}}_i$ 的距离 $d_i = \ \mathbf{h} - \hat{\mathbf{h}}_i\ _2$ .
3	用 NSD 或 MNSD 算法计算 $\mathbf{h}$ 与所有余类零空间的零空间距离 $\mathbf{Val} = [Val_1, Val_2, \dots, Val_K]$ .
4	计算判别变量 $j_i = Val_i / d_i$ .
5	最大的 $j_i = Val_i / d_i$ 对应的类别即判定为 $\mathbf{h}$ 所属的类别.

本文提出的 MNSD 算法的思想在于用不同类别的余类零空间最大化不同类别人脸图像的差异性. 将最近空间方法结合到余类零空间的方法后, 本文提出的 NSND 算法不仅利用了同一类别中不同图片之间的相似性, 而且最大化利用了不同类别之间的差异性.

## 2.3 算法时间复杂度分析

MNSD 算法的计算量主要包括求解矩阵的右逆矩阵, 求解多个矩阵的 QR 分解.

对矩阵  $\mathbf{H} = [\mathbf{H}_1^T \mathbf{H}_2^T \dots \mathbf{H}_K^T]^T \in \mathfrak{R}^{N \times D}$ , 计算其右逆矩阵的时间复杂度为  $O(DN^2)$ . 对需要进行 QR 分解的矩阵  $\mathbf{H}_{i,right} \in \mathfrak{R}^{D \times S}$ , 计算  $K$  个这样的矩阵的 QR 分解的时间复杂度为  $O(2KDS^2)$ , 整个算法的复杂度为  $O(DN^2 + 2KDS^2)$ .

然而, 上述的计算过程都只需要进行一次计算即可. 即将余类零空间全部计算出来后即可应用于所有测试样本.

对每一个样本向量, 算法需要计算其与所有余类零空间乘积的二范数, 即对计算得到的所有零空间  $\mathbf{P} = [\mathbf{P}_{1,right}, \mathbf{P}_{2,right} \dots \mathbf{P}_{K,right}]$ , 其中  $\mathbf{P}_{i,right} \in \mathbb{R}^{D \times S}$ , 则一个样本向量与其相乘求二范数的算法复杂度为  $O(KDS)$ .

本文算法的运行时间与 NS 算法相当, 但比 NFS 算法快许多. 因为 NFS 算法需要做多次大维度矩阵的最小二乘法, 而本文算法只需要对一个大维度矩阵做一次右逆矩阵计算.

### 2.4 本文算法与同类算法的关系

为了更好地将本文算法与其它算法进行比较, 有必要说明本文算法与其它算法的一些关系. 与本文比较的其它算法有 NN 算法、NS 算法、NFS 算法等. 本文算法与其它算法的关系图如图 2 所示.

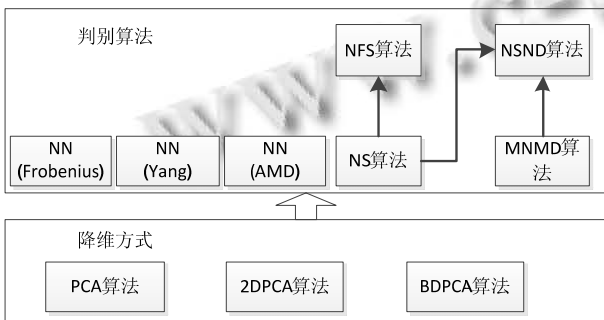


图 2 本文算法与其它算法的关系图

从图 2 可以看到, 在 PCA、2DPCA 与 BDPCA 三种降维的方式基础上, 将七种算法进行了比较.

NFS 算法在 NS 算法的基础上增加了最远距离的判别手段, 本文的 NSND 算法在 MNMMD 算法的基础上结合了 NS 算法的判别手段.

特别的, 当采用 PCA 降维方法时, 降维后的图像为向量形式, 不能采用 NN(Yang) 与 NN(AMD) 算法, 因此, 在使用 PCA 降维方法时, 共对比了五种算法.

## 3 实验结果与讨论

我们用 ORL 人脸数据集与 AR 人脸数据集对本文

提出的算法进行了验证. 我们用多种基于 PCA 的算法进行图像特征提取. 在 PCA、2DPCA 与 BDPCA 三种不同的特征提取方法的基础上, 本文算法与 NN 算法、NS 算法、NFS 算法等进行了比较.

### 3.1 ORL 数据集实验结果

ORL 人脸数据集共包含 400 张人脸图像, 每种类别包含 10 张图像. 图像之间年龄、灯光、表情、面部细节等不尽相同, 每张图像的大小为 112\*93. 图 3 展示了 ORL 数据集中的部分图像.



图 3 ORL 数据集中部分图像

对于每种类别, 实验随机选取  $S$  张图片作为训练集, 用多次实验的平均准确率作为评价标准(运行 20 次的平均结果). 我们对不同的  $S$  进行了测试. 使用特征提取算法为 BDPCA 时, 选取的参数为  $k_{row}=12, k_{col}=12$ . 使用特征提取算法为 2DPCA 时, 选取的参数为  $k_{row}=12$ . 对本文提出的算法, 我们将特征图像分为 8 组, 每组包含 5 类人脸数据.

我们将本文的 MNSD, NSND(结合 NS)与如下算法进行了比较: NS, NFS, NN(AMD), NN(Frobenius 距离), NN(Yang 距离). 实验结果见表 3, 表 4 与表 5. 在这些表中, 我们可以看到不同的算法与不同的训练图像个数对实验结果的影响. 表 3 展示了使用 PCA 作为特征提取方法的实验结果, 表 4 展示了使用 2DPCA 作为特征提取方法的实验结果, 表 5 展示了使用 BDPCA 作为特征提取方法的实验结果.

表 3 在 ORL 数据集上基于 PCA 特征的比较结果

训练数据个数	NS	NN(Frobenius)	NFS	MNSD	NSND(结合 NS)
2	0.8031	0.8109	0.6694	0.8273	<b>0.8419</b>
3	0.8747	0.8780	0.8236	0.8936	<b>0.9061</b>
4	0.9158	0.9244	0.8950	<b>0.9408</b>	0.9371
5	0.9525	0.9467	0.9270	0.9557	<b>0.9645</b>
6	0.9600	0.9525	0.9312	0.9666	<b>0.9719</b>
7	0.9750	0.9675	0.9600	0.9695	<b>0.9833</b>

8	0.9700	0.9725	0.9350	<b>0.9781</b>	0.9762
9	0.9700	0.9800	0.9850	<b>0.9875</b>	0.9800

表 4 在 ORL 数据集上基于 2DPCA 特征的比较结果

训练数据个数	NS	NN(Frobenius)	NN(Yang)	NN(AMD)	NFS	MNSD	NSND(结合 NS)
2	0.8294	0.8164	0.8199	0.7838	0.8525	0.8397	<b>0.8544</b>
3	0.8850	0.8808	0.8860	0.9006	0.9057	0.9009	<b>0.9121</b>
4	0.9460	0.9267	0.9119	0.9190	0.9317	0.9288	<b>0.9491</b>
5	0.9579	0.9420	0.9485	0.9170	<b>0.9690</b>	0.9595	0.9636
6	0.9688	0.9438	0.9551	0.9538	0.9650	0.9656	<b>0.9700</b>
7	<b>0.9834</b>	0.9598	0.9617	0.9600	0.9800	0.9717	0.9767
8	0.9752	0.9702	0.9608	0.9652	0.9675	0.9719	<b>0.9825</b>
9	0.9750	<b>0.9850</b>	0.9663	0.9750	0.9702	0.9800	0.9800

表 5 在 ORL 数据集上基于 BDPCA 特征的比较结果

训练数据个数	NS	NN(Frobenius)	NN(Yang)	NN(AMD)	NFS	MNSD	NSND(结合 NS)
2	0.8304	0.8208	0.8166	0.7940	0.8287	0.8355	<b>0.8541</b>
3	0.8981	0.8970	0.8919	0.8470	0.8976	0.9055	<b>0.9254</b>
4	0.9400	0.9310	0.9282	0.8972	0.9472	0.9471	<b>0.9500</b>
5	0.9630	0.9510	0.9440	0.9190	0.9583	0.9583	<b>0.9645</b>
6	0.9694	0.9614	0.9599	0.9476	0.9656	0.9672	<b>0.9725</b>
7	0.9791	0.9666	0.9683	0.9500	0.9736	0.9783	<b>0.9808</b>
8	0.9827	0.9626	0.9784	0.9726	0.9812	0.9806	<b>0.9888</b>
9	<b>0.9950</b>	0.9900	0.9830	0.9800	0.9790	0.9812	0.9900

从表 3 中, 我们可以看到, 当作为训练的图像的个数为 2 到 9 时, 本文的算法的识别率最高。

从表 4 可以看到, 除了 5、7、9, 本文算法在别的训练个数时, 取得的结果最好。

从表 5 可以看到, 除了 9, 在别的训练个数情况下, 本文算法的结果最好。

尽管本文算法在某些训练个数的情况下不是最好的, 但这些不是最好的结果与最好的结果很接近。

### 3.2 AR 数据集实验结果

我们在 AR 数据集中选取了包含 50 个男性与 50 个女性的共 100 类人脸图片。每一类人脸图片包含 14 张图片。每张图片的尺寸为 60\*43。图 4 展示了 AR 数据集中的部分人脸图片。



图 4 AR 数据集中的部分图像

对于每种类别, 实验随机选取了  $s$  张图片作为训练数据集, 用多次实验的平均准确率作为评价标准(运行 20 次的平均结果)。我们对不同的  $s$  进行了测试。使用特征提取算法为 BDPCA 时, 选取的参数为  $k_{row} = 20$ ,  $k_{col} = 20$ 。使用特征提取算法为 2DPCA 时, 选取的参数为  $k_{row} = 10$ 。对本文提出的算法, 我们将特征图像分为 10 组, 每组包含 10 类人脸数据。我们将本文的 MNSD, NSND(结合 NS)与如下算法进行了比较: NS, NFS, NN(AMD), NN(Frobenius 距离), NN(Yang 距离)。

图 5 展示了使用 PCA 作为特征提取方法的结果。

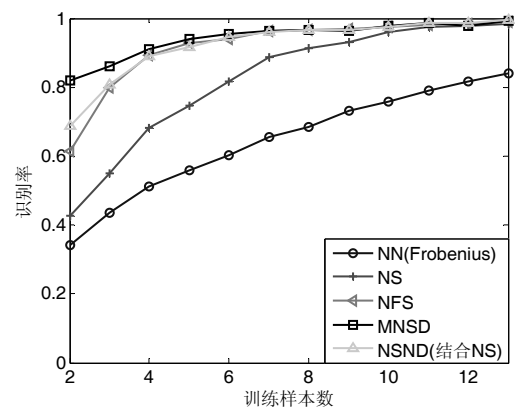


图 5 在 AR 数据集上基于 PCA 特征的算法识别率比较

从图 5 的结果可以看出, 本文的 MNSD 与 NSND(结合 NS)的识别率最高.

图 6 展示了使用 2DPCA 作为特征提取方法的结果. 图 7 展示了使用 BDPCA 作为特征提取方法的结果.

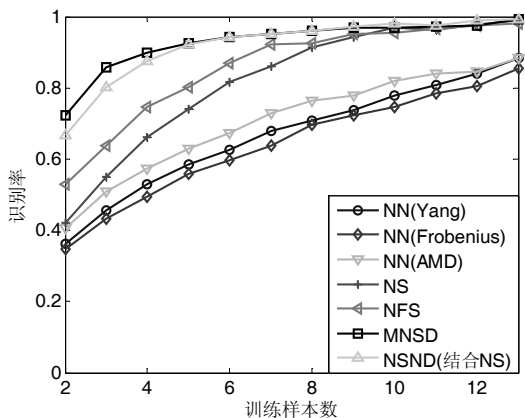


图 6 AR 数据集上基于 2DPCA 特征的算法识别率比较

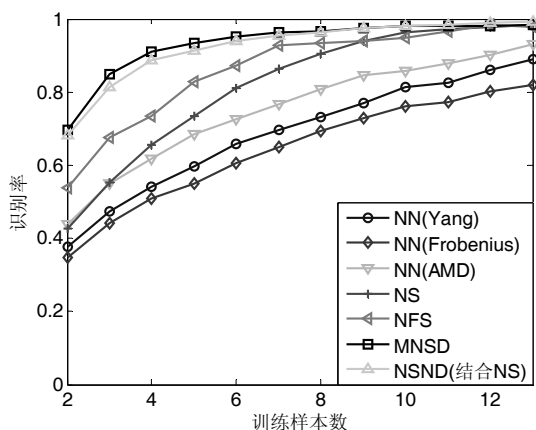


图 7 AR 数据集上基于 BDPCA 特征的算法识别率比较

从图 6 与图 7 可以看出, 在平均识别率的指标下, 本文的 MNSD 与 NSND(结合 NS)的表现最好.

结合以上的 ORL 数据集与 AR 数据集的比较结果可以看出, 本文的算法在 AR 数据集上比在 ORL 数据集上更有效. 原因如下:

① AR 数据集的维度比 ORL 数据集的维度大, 而本文提出的算法在数据集的维度较大时更有效.

② AR 数据集中图像的变化性比 ORL 中图像的变化性大, 例如灯光、面部表情等. 本文提出的算法在图像变化性较大时更有效.

从上述的实验结果可以看出, 本文提出的算法比传统的算法更有效. 在得到训练图像的特征矩阵后, 我们计算不同类别特征矩阵的余类零空间. 相比较于

图像的特征矩阵, 这些独立的余类零空间可以将不同类别的人脸最大化地分离开来. 传统的分类器主要利用了图像之间的相似性而没有充分利用不同类别图像之间的差异性. 本文提出的算法不仅利用了同一类别图像之间的相似性, 而且充分利用了不同类别图像之间的差异性.

#### 4 结语

本文提出了一种新型的使用余类零空间与最近空间距离的分类器. 我们在 ORL 数据集与 AR 数据集上进行了测试. 测试结果表明本文算法性能优异, 即使当训练数据较小时也可取得较高的识别率. 此外, 本文的降维算法使用的是 PCA 及其改进算法, 当使用别的降维算法, 如独立成分分析算法(ICA)、线性判别式分析算法(LDA)、局部保留投影算法(LPP)时算法的性能将是继续研究的地方. 另外在算法结果中可以看出, NSND(结合 NS)算法并非在所有情况下都好于本文的另一种 MNSD 算法, 如何调整余类零空间距离与最近空间距离的权重以便得到更好的识别效果也是算法有待改进的地方.

#### 参考文献

- 1 Sirovich L, Kirby M. Low-dimensional procedure for characterization of human faces. *Journal of the Optical Society of American*, 1987, 33(4): 591-596.
- 2 Li C, Liu J, Wang A, et al. Matrix reduction based on generalized PCA method in face recognition. *International Conference on Digital Home*. IEEE. 2014. 35-38.
- 3 Yang J, Zhang D, Frangi AF, et al. Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2004, 26(1): 131-137.
- 4 Zuo W, Zhang D, Wang K. Bidirectional PCA with assembled matrix distance metric for image recognition. *IEEE Trans. on Systems Man and Cybernetics Society*, 2006, 36(4): 863-872.
- 5 Zhou C, Wang L, Zhang Q, et al. Face recognition based on PCA and logistic regression analysis. *Optik-International Journal for Light and Electron Optics*, 2014, 125(20): 5916-5919.
- 6 Tian X, Tian M. Face recognition using a hybrid algorithm based on improved PCA. *Proc. of the 2013 International*



- Conference on Information Technology and Computer Application Engineering. ITCAE. 2014. 289–292.
- 7 Luan X, Fang B, Liu L, et al. Extracting sparse error of robust PCA for face recognition in the presence of varying illumination and occlusion. *Pattern Recognition*, 2014, 47(2): 495–508.
- 8 Mi JX, Huang DS, Wang B, et al. The nearest-farthest subspace classification for face recognition. *Neurocomputing*, 2013, 113(7): 241–250.
- 9 He R, Yang N, Wang XK, et al. Direct sparse nearest feature classifier for face recognition. *Life System Modeling and Intelligent Computing-International Conference on Life System Modeling and Simulation*. 2010. 386–394.
- 10 Naseem I, Togneri R, Bennamoun M. Linear Regression for Face Recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2010, 32(11): 2106–2112.
- 11 Choi LU, Murch RD. A transmit preprocessing technique for multiuser MIMO systems using a decomposition approach. *IEEE Trans. on Wireless Communications*, 2004, 3(1): 20–24.
- 12 Zu K, de Lamare RC, Haardt M. Generalized design of low-complexity block diagonalization type precoding algorithms for multiuser MIMO systems. *IEEE Trans. on Communications*, 2013, 61(10): 4232–4242.
- 13 Li SZ, Lu J. Face recognition using the nearest feature line method. *IEEE Trans. on Neural Networks*, 1999, 10(2): 439–443.
- 14 Chien JT, Wu CC. Discriminant waveletfaces and nearest feature feature classifiers for face face recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2002, 24(12): 1644–1649.
- 15 Shi Q, Eriksson A, Hengel AVD et al. Is face recognition really a compressive sensing problem? *IEEE Conference on Computer Vision and Pattern Recognition*. 2011. 553–560.