

**本 科 毕 业 设 计**

基于流形学习子空间的人脸识别算法

|  |  |
| --- | --- |
| 姓 名 | 姜希成 |
| 学 院 | 信息与电气工程学院 |
| 专 业 | 计算机科学与技术 |
| 年 级 | 2015 |
| 学 号 | 20152203031 |
| 指导教师 | 王庆军 |

2019年5月4日

独 创 声 明

本人郑重声明：所呈交的毕业设计，是本人在指导老师的指导下，独立进行研究工作所取得的成果，成果不存在知识产权争议。尽我所知，除文中已经注明引用的内容外，本设计不含任何其他个人或集体已经发表或撰写过的作品成果。对本文的研究做出重要贡献的个人和集体均已在文中以明确方式标明。

此声明的法律后果由本人承担。

作者签名：

年 月 日

毕业设计使用授权声明

本人完全了解鲁东大学关于收集、保存、使用毕业设计的规定。

本人愿意按照学校要求提交设计的印刷本和电子版，同意学校保存设计的印刷本和电子版，或采用影印、数字化或其它复制手段保存设计；同意学校在不以营利为目的的前提下，建立目录检索与阅览服务系统，公布设计的部分或全部内容，允许他人依法合理使用。

（保密论文在解密后遵守此规定）

论文作者：

年 月 日

**毕业设计开题报告**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | 姜希成 | | 学 院 | 信息与电气工程学院 | | 年级 | 2015 | 学号 | 20152203031 |
| 题 目 | | 基于流形学习子空间的人脸识别算法 | | | | | | | |
| 课题来源 | | 科研 | | 课题类别 |  | | | | |
| **选题意义（包括科学意义和应用前景，研究概况，水平和发展趋势，列出主要参考文献目录）：**  TODO：  **主要参考文献：**  [1] 古楠楠，樊明宇，王迪，韩志. 流形学习若干关键问题与算法研究[M]. 北京: 首都经济贸易大学出版社，2015. | | | | | | | | | |
| **研究主要内容和预期结果（说明具体研究内容和拟解决的关键问题，预期结果和形式，如在理论上解决哪些问题及其价值，或应用的可能性及效果）：**  TODO：  主要研究内容：  预期结果： | | | | | | | | | |
| **拟采取的研究方法和技术路线（包括理论分析、计算，实验方法和步骤及其可行性论证，可能遇到的问题和解决方法，以及研究的进度与计划）：**  研究方法：实验法、经验总结法、文献研究法、功能分析法  技术线路： TODO：  研究的进度与计划：2019年3月1日，选题。  2019年3月1日---2019年3月6日，查阅并整理相关资料。  2019年3月6日---2019年3月10日，完成开题报告。  2019年3月10日---2019年4月20日，完成程序创建与前期功能实现。  2019年4月20日---2019年5月1日，完善程序功能，完成论文初稿。 | | | | | | | | | |
| **指导教师意见（对论文选题的意义、应用性、可行性、进度与计划等内容进行评价，填写审核结果：同意开题、修改后再开题、不同意开题）：**  该生对本课题相关的知识与理论研究比较透彻，参考了许多的文献资料，具有一定的研究价值。  　　本课题结构合理，内容完整，主要观点突出，并且时效性强，是学生学习方向的延续，对于提高学生的能力有利。  　　同意该课题开题。  签名：  年 月 日 | | | | | | | | | |
| **学院毕业设计领导小组意见：**        （签章）  年 月 日 | | | | | | | | | |

**毕业设计结题报告**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 姓 名 | 姜希成 | | | 学 院 | 信息与电气工程学院 | | | 年级 | 2015 | 学号 | 20152203031 |
| 题 目 | | 基于流形学习子空间的人脸识别算法 | | | | | | | | | |
| 课题来源 | | 科研 | | | | 课题类别 |  | | | | |
| **本课题完成情况介绍（包括研究过程、实验过程、结果分析、存在的问题及应用情况等。）**  研究过程： TODO：  结果分析：  存在的问题：（1）。（2）。（3）。 | | | | | | | | | | | |
| **指导教师评语：**  该生对本课题相关的知识与理论研究比较透彻，本设计基本实现了需求所描述的功能。  　　设计选题新颖，功能全面，应用到的技术较多，时效性强。  　　同意结题。  签名：  年 月 日 | | | | | | | | | | | |
| **学院毕业设计领导小组意见：**  （公章）  年 月 日 | | | | | | | | | | | |
| 指导教师  评定成绩 | | |  | | | | | | | | |

**毕业设计成绩评定表**

学院：信息与电气工程学院 学号：20152203031

|  |  |  |  |
| --- | --- | --- | --- |
| 姓 名 | 姜希成 | 总成绩 |  |
| 题 目 | 基于流形学习子空间的人脸识别算法 | | |
| 评  阅  人  评  语 |  | | |
| 评定成绩： 评阅人（签名）：  年 月 日 | | |
| 答  辩  小  组  评  语 |  | | |
| 答辩成绩： 答辩组成员（签名）： 年 月 日 | | |

注：总成绩=指导教师评定成绩（40%）+评阅人评定成绩（20%）+答辩成绩（40%），将总成绩由百分制转换为五级制，填入本表相应位置。

目 录

[1.人脸识别技术的简单介绍 1](#_Toc7509030)

[1.1人脸识别的发展情况 2](#_Toc7509031)

[1.1.1人脸识别的技术及应用简介 2](#_Toc7509032)

[1.1.2人脸识别技术的优势 2](#_Toc7509033)

[1.1.3人脸识别技术的市场前景 2](#_Toc7509034)

[1.1.4人脸识别系统的研究内容 2](#_Toc7509035)

[1.2流形学习 2](#_Toc7509036)

[1.2.1流形学习的研究背景 3](#_Toc7509037)

[1.2.2流形学习的发展情况 3](#_Toc7509038)

[1.2.3流形学习的定义 3](#_Toc7509039)

[1.2.4本文用到的流形学习算法 4](#_Toc7509040)

[2本文要用到的其他算法的详细介绍 4](#_Toc7509041)

[2.1规范正交基 4](#_Toc7509042)

[2.1.1施密特(Schmidt)正交化 4](#_Toc7509043)

[2.1.2单位化 5](#_Toc7509044)

[2.2迪杰斯特拉(Dijkstra)算法 5](#_Toc7509045)

[2.2.1迪杰斯特拉(Dijkstra)算法原理 5](#_Toc7509046)

[2.2.2迪杰斯特拉(Dijkstra)算法步骤 5](#_Toc7509047)

[2.3 ISOMAP算法 6](#_Toc7509048)

[2.3.1 ISOMAP算法步骤 6](#_Toc7509049)

[2.3.2 ISOMAP算法流程 6](#_Toc7509050)

[2.3.3 ISOMAP算法的优点 7](#_Toc7509051)

[2.3.4 ISOMAP算法的缺点 7](#_Toc7509052)

[2.4 MDS算法 7](#_Toc7509053)

[2.4线性图嵌入算法(LGE) 8](#_Toc7509054)

[2.4.1 LGE算法推导 8](#_Toc7509055)

[2.4.2 LGE算法流程 9](#_Toc7509056)

[2.5正交线性图嵌入算法(OLGE) 9](#_Toc7509057)

[3等距映射算法（IsoProjection） 9](#_Toc7509058)

[3.1 IsoProjection算法推导 9](#_Toc7509059)

[3.2 IsoProjection算法流程 10](#_Toc7509060)

[3.3 IsoProjection算法优点 10](#_Toc7509061)

[3.4 IsoProjection算法缺点 10](#_Toc7509062)

[4领域保护嵌入算法（NPE） 11](#_Toc7509063)

[4.1 NPE算法推导 11](#_Toc7509064)

[4.2 NPE算法流程 11](#_Toc7509065)

[4.3 NPE算法优点 11](#_Toc7509066)

[4.4 NPE算法缺点 12](#_Toc7509067)

[5正交的IsoProjection (OIsoProjection)和NPE(ONPE) 12](#_Toc7509068)

[5.1 OIsoProjection 12](#_Toc7509069)

[5.1.1 OIsoProjection算法推导 12](#_Toc7509070)

[5.1.2 OIsoProjection算法流程 13](#_Toc7509071)

[5.1.3 OIsoProjection算法优点 13](#_Toc7509072)

[5.1.4 OIsoProjection算法缺点 13](#_Toc7509073)

[5.2 ONPE 14](#_Toc7509074)

[5.2.1 ONPE算法推导 14](#_Toc7509075)

[5.2.2 ONPE算法流程 15](#_Toc7509076)

[5.2.3 ONPE算法优点 15](#_Toc7509077)

[5.2.4 ONPE算法缺点 15](#_Toc7509078)

[6实验 15](#_Toc7509079)

[6.1常用的人脸数据库介绍 15](#_Toc7509080)

[6.1.1 FERET人脸数据库 16](#_Toc7509081)

[6.1.2 Yale人脸数据库 16](#_Toc7509082)

[6.1.3 Yale B人脸数据库 16](#_Toc7509083)

[6.1.4 ORL人脸数据库 16](#_Toc7509084)

[6.1.5 AR人脸数据库 17](#_Toc7509085)

[6.1.6 XM2VTS人脸数据库 17](#_Toc7509086)

[6.1.7 CMU PIE人脸数据库 17](#_Toc7509087)

[6.2实验结果 17](#_Toc7509088)

[6.2.1 NPE 17](#_Toc7509089)

[6.2.2 ONPE 18](#_Toc7509090)

[6.2.3 IsoProjection 18](#_Toc7509091)

[6.2.4 OIsoProjection 18](#_Toc7509092)

[6.2.5各个算法的性能对比 18](#_Toc7509093)

[7结论 18](#_Toc7509094)

[7.1 NPE与IsoProjection 18](#_Toc7509095)

[7.2 ONPE与OIsoProjection 19](#_Toc7509096)

[8结束语 19](#_Toc7509097)

[9附录 19](#_Toc7509098)

[9.1 算法详细推导流程 19](#_Toc7509099)

[9.1.1 OIsoProjection算法详细推导过程 19](#_Toc7509100)

[9.1.2 ONPE算法详细推导过程 20](#_Toc7509101)

[9.2 人脸数据库 21](#_Toc7509102)

[9.3 主要算法的核心代码 22](#_Toc7509103)

[9.3.1 IsoProjection算法 22](#_Toc7509104)

[9.3.2 NPE算法 24](#_Toc7509105)

[9.3.3 LGE算法 25](#_Toc7509106)

[参考文献 28](#_Toc7509107)

[致 谢 31](#_Toc7509108)

基于流形学习子空间的人脸识别算法

姜希成

（信息与电气工程学院，计算机科学与技术，2015级2班，20152203031）

**摘 要：** 本文研究了等距映射算法（IsoProjection）、邻域保护嵌入算法（NPE）以及正交的等距映射算法（OIsoProjection）和正交的邻域保护嵌入算法（ONPE），并在ORL人脸库上进行试验，在本文最后将对这些算法进行对比，研究每种算法的优缺点。

**关键字：**人脸识别，流形学习，NPE，ONPE，IsoProjection，OIsoProjection。

**Face recognition algorithm based on manifold learning subspace**

Jiang XiCheng

(School of Information and Electrical Engineering, Ludong University)

**Abstract:** In this paper, isometric mapping algorithm (IsoProjection), Neighborhood protection embedding algorithm (NPE) and orthogonal isometric mapping algorithm (OIsoProjection) and orthogonal neighborhood protection embedding algorithm (ONPE) are introduced and tested on ORL face library. At the end of this article, we will compare these algorithms and study the advantages and disadvantages of each algorithm.

**Key words:** Face recognition, Manifold Learning, NPE, ONPE, IsoProjection, OIsoProjection.

# 1.人脸识别技术的简单介绍

人脸识别技术和扩展已应用于生活的各个方面，本文将介绍如何使用流形学习子空间方法来实现人脸识别技术，

TODO：完善

## 1.1人脸识别的发展情况

### 1.1.1人脸识别的技术及应用简介

人脸识别是基于人脸特征信息的身份识别。人脸识别技术始于20世纪60年代，到现在已经有了近60年的发展史，在这60年中人脸识别技术得到突飞猛进的发展，相继出现了许多经典算法、经典思想和经典人脸库，如今人脸识别技术已经达到了比肉眼还要准确的地步，我相信在未来人脸识别技术一定能我们创造更加便利的生活条件。

### 1.1.2人脸识别技术的优势

人脸识别是生物识别的一种，生物识别因为具有易检测、唯一性和终身不变的特点，所以十分适合互联网时代用户对安全的需求，如今生物识别技术的识别速度更快，准确率更高，因此也更具有研究的价值。

目前主流的生物识别技术有人脸识别、指纹识别、虹膜识别、语音识别、静脉识别等。相对于其他的生物识别技术，人脸识别技术的成本更低、稳定性更好、准确率更高，因此也更具有发展优势，所以在多数应用场景中都会首选人脸识别技术。

### 1.1.3人脸识别技术的市场前景

市场前景是判断一项技术是否有研究价值的重要指标。

人脸识别市场前景非常好，发展势头也非常迅速，2017年全球人脸识别的市场规模超过40.5亿美元，预计2022年达到77.6亿美元，复合年增长率高达13.9%。

TODO：总结

### 1.1.4人脸识别系统的研究内容

TODO：

## 1.2流形学习

流形学习的本质在于根据有限的离散样本学习和发现嵌入在高维空间中的低维光滑流形,从而揭示隐藏在高维数据中的内在低维结构,以实现非线性降维或者可视化。

TODO：完善

### 1.2.1流形学习的研究背景

随着互联网时代的发展，在各个研究领域，每时每刻都在快速生成大量的数据，而在这些数据背后的规律却难以发现，人们虽然获取了海量的信息却发现自己正处于“数据丰富，知识匮乏”的尴尬境地，因此如何从海量信息中提取自己所需的知识是当今各个领域共同面临的巨大挑战。

在许多实际应用中，尤其是在人脸识别中，往往要对成千上万张图片进行处理，而每张图片又有极高的维度，这种高维的特质往往隐藏了数据间关系的本质，对于传统的数据分析方法往往会造成“维数灾难”，这种情况就需要通过降维来把高维空间的数据间的关系映射到低纬度空间，这样就可以更加方便快速的处理数据。因此降维就成了这一任务吸引了许多科研人员的注意，也成为了如人脸识别、机器学习和数据挖掘等领域的热门研究问题。而流形学习就是解决这类问题的方案之一。

自从流形学习方法被提出到现在，研究它的工作就一直在进行着，特别是近年来随着数据挖掘和人脸识别等技术的高速发展，“维度灾难”的问题就成了相关研究领域的重大障碍。本文将利用流行学习方法解决这类问题。

### 1.2.2流形学习的发展情况

1984年斯坦福大学统计系的Hastie在一份技术报告中首次提出主曲线和主曲面的概念。1995年Bregler和Omohundro首次提出“流形学习”的概念，主要应用于图像插值和语音识别中。2000年Seung和Lee在《科学》杂志上发表《认知的流行模式》一文，提出了视觉感知的流行结构假说。就在同年同期的《科学》上还刊登了另外两篇著名的文章，它们提出了两个经典算法LLE和Isomap。随着后期学者不断地深入研究，又有许多经典算法相继被提出，比如Belkin和Niyogi提出拉普拉斯特征映射算法（LE），使高维空间相近的点映射到低微空间时也相近；Donoho和Grimes提出海森特征映射（HLLE），是对LLE算法的扩展；He和Hiyogi提出局部保持投影算法（LPP），是对LE算法的线性扩展；Zhang和Zha提出局部切空间校准算法（LTSA），基于“局部拟合，全局整合”的思想；Lin和Zha提出黎曼流行（RML），利用局部黎曼正交坐标系将高维空间的数据映射到低微本质空间中去。这些经典算法被提出后，又有许多学者为了弥补这些算法的缺陷而相继提出了很多经典的改进算法。

此外，很多学者发现不同的流形学习算法之间存在着一定的联系，又提出了一些框架将多种流形学习算法纳入其中，比如经典的核主成分分析框架（KPCA）将MDS、LLE、LE、Isomap和谱聚类进行统一，GA框架在将LE、LLE、LPP和Isomap等流形学习方法纳入其中的同时又将PCA和LDA等传统线性降维方法统一进去。

### 1.2.3流形学习的定义

TODO：参考论文：基于流形学习子空间的人脸识别方法研究-王

### 1.2.4小结

本文主要用的NPE算法和IsoProjection算法并详细介绍算法实现原理，本文会在后面介绍正交的NPE和IsoProjection算法，并对比这些算法的优缺点。

TODO：完善

# 2流形学习相关算法

NPE和IsoProjection算法的实现需要用到LGE算法，IsoProjection算法因为涉及最短路径选择所以用到迪杰斯特拉算法，而IsoProjection算法又是基于ISOMAP算法的改进，ISOMAP算法又涉及到MDS算法。通过LGE算法构造OLGE算法要用到规范正交基。

TODO：完善

## 2.1规范正交基

定义：设是一个向量空间。

（1）若是向量空间V的一个基，且是两两正交的向量组，则称是向量空间V的正交基。

（2）若是向量空间V的一个基，两两正交，且都是单位向量，则称是向量空间V的一个规范正交基（或标准正交基）。

求法：设是向量空间V的一个基，要求V的一个规范正交基，也就是要找一组两两正交的单位向量，使与等价。这一过程称为把基规范正交化。

可以先进行施密特正交化在进行单位化。本节接下来将详细介绍施密特正交化和单位化的计算方法。

### 2.1.1施密特(Schmidt)正交化

令：

；

;

……

，

则易验证两两正交，且与等价。且满足：对任何，向量组与等价。

### 2.1.2单位化

令：

则是V的一个规范正交基。

施密特正交化过程中可将中的任一线性无关的向量组化为与之等价的正交向量组；在经过单位化，得到与等价的规范正交向量组.

## 2.2迪杰斯特拉(Dijkstra)算法

迪杰斯特拉(Dijkstra)提出了一个按路径长度递增的次序产生最短路径的算法。该算法是典型最短路径算法，用于计算一个节点到其他节点的最短路径。

它的主要特点是以起始点为中心向外层层扩展 (广度优先搜索思想)，直到扩展到终点为止。

### 2.2.1迪杰斯特拉(Dijkstra)算法原理

1. 通过 Dijkstra 计算图 G 中的最短路径时，需要指定起点 s（即从顶点 s 开始计算）。
2. 此外，引进两个集合 S 和 U。S 的作用是记录已求出最短路径的顶点 (以及相应的最短路径长度)，而 U 则是记录还未求出最短路径的顶点 (以及该顶点到起点 s 的距离)。
3. 初始时，S 中只有起点 s；U 中是除 s 之外的顶点，并且 U 中顶点的路径是“ 起点 s 到该顶点的路径”。然后，从 U 中找出路径最短的顶点，并将其加入到 S 中；接着，更新 U 中的顶点和顶点对应的路径。 然后，再从 U 中找出路径最短的顶点，并将其加入到 S 中；接着，更新 U 中的顶点和顶点对应的路径。重复该操作，直到遍历完所有顶点。

### 2.2.2迪杰斯特拉(Dijkstra)算法步骤

1. 初始时，S 只包含起点 s；U 包含除 s 外的其他顶点，且 U 中顶点的距离为“起点 s 到该顶点的距离”。例如，U 中顶点 v 的距离为 的长度，然后 s 和 v 不相邻，则 v 的距离为∞。
2. 从 U 中选出“ 距离最短的顶点 k”，并将顶点 k 加入到 S 中；同时，从 U 中移除顶点 k。
3. 更新 U 中各个顶点到起点 s 的距离。之所以更新 U 中顶点的距离，是由于上一步中确定了 k 是求出最短路径的顶点，从而可以利用 k 来更新其它顶点的距离；例如，的距离可能大于 的距离。
4. 重复步骤 2 和3，直到遍历完所有顶点。

## 2.3 ISOMAP（等度量映射）算法

Isomap 通过 “改造一种原本适用于欧氏空间的算法”，达到了“将流形映射到一个欧氏空间” 的目的。

Isomap 所改造的这个方法是 MDS，它的目的就是使得降维之后的点两两之间的距离尽量不变。只是 MDS 是针对欧氏空间设计的，对于距离的计算也是使用欧氏距离来完成的。如果数据分布在一个流形上的话，欧氏距离就不适用了。

Isomap 把 MDS 中原始空间中距离的计算从欧氏距离转换为了流形上的测地距离。当然，如果流形的结构事先不知道的话，这个距离是没法算的，于是 Isomap 通过将数据点连接起来构成一个邻接 Graph 来离散地近似原来的流形，而测地距离也相应地通过 Graph 上的最短路径来近似了。



所以低维嵌入流形上的测地线距离（红色）不能用高维空间的直线距离计算，但能用近邻距离来近似。

TODO：参考论文，再加一个ISOMAP算法步骤。

### 2.3.1 ISOMAP算法步骤

ISOMAP算法分为三步：

1. 构造邻域图。
2. 计算测地距离。
3. 数据嵌入。使用MDS算法。

### 2.3.2 ISOMAP算法流程

输入：样本集；近邻参数k；低维空间维数d。

过程：

确定的k近邻；

与k近邻之间的距离设 置为欧式距离，与其他点的距离设置为无穷大；

调用最短路径算法（迪杰斯特拉算法）计算任意两样本点之间的距离；

将作为MDS算法的输入；

MDS算法的输出；

输出：样本集D在低维空间的投影。

### 2.3.3 ISOMAP算法的优点

ISOMAP算法最主要的优点就是使用 “测地距离”，而不是使用原始的欧几里得距离，这样可以更好的控制数据信息的流失，能够在低维空间中更加全面的将高维空间的数据表现出来。

### 2.3.4 ISOMAP算法的缺点

虽然相对于MDS算法而言，Isomap算法可以更好的保持和表示数据内部的几何结构，但Isomap算法还存在着一些不足之处，其中最严重的一点是算法本身的拓扑不稳定性，即Isomap算法在图中可能会建立错误的连接，这将严重的影响算法的执行。其次，Isomap算法针对的是凸形的流形，对于非凸形的流形将不能处理。再者，对于流形中含有“孔洞”的情况，一般的Isomap算法也不能很好的处理。

TODO：抄的，改善，引出IsoProjection

## 2.4 MDS（多维缩放）算法

MDS 与 PCA 一样，是一种有效的降维方式，其可获得样本间相似性的空间表达。MDS 的原理可以简述为，利用样本的成对相似性，构建一个低维空间，使每对样本在高维空间的距离与在构建的低维空间中的样本相似性尽可能保持一致。

MDS 算法的核心思想是：降维前后，各自样本间的距离是不变的。由此可得到如下关系：原空间中的两样本之间的距离等于降维后这两样本之间的距离 ，即。所以：

，得到：

，，其中B为降维后样本的内积矩阵，。

对B做特征值分解 ，这里V是特征向量矩阵、A是由特征值构成的对角阵。此时，把A中的特征值排序后，把其中每个非0特征值拿出来构成对角矩阵，其对应的特征向量也需按特征值的大小改变排列顺序，组成新的特征向量矩阵。最终，通过这个公式完成降维操作：。

另外，在现实应用中为了有效降维，往往只需要降维后两样本间的距离应尽可能和原空间中两样本的距离相近就好了，不需要强行一致，因此上面特征值构成的对角阵和特征向量矩阵有了一些变化。

变化为：本来特征值取的是所有非0的特征值排序。现在变成排序好后，从大到小取特征值与其对应的特征向量。比如要降维到5维，就从大到小取5个特征值。

## 2.5线性图嵌入算法(LGE)

为基于图的子空间学习提供一般框架。该算法将由NPE和IsoProjection调用。

TODO：简介

### 2.5.1 LGE算法推导

数据集，n表示样本数量，m表示样本维度。LGE算法采用无向有权图描述数据集的流形结构，顶点集对应数据，，表示边的权重，W矩阵是对称的。LGE算法在保持图的邻接关系的前提下，寻找X的低维表示。令表示从图到实线的映射，则LGE的目标函数如下：

变换后可得：

L是图的拉普拉斯矩阵，D是对角矩阵，对应W矩阵第列（行）所有元素之和，即。

令（消除嵌入时的量化影响），则：

若图到实线的映射为线性，则，则：

则最优向量对应于以下最小特征值所对应的特征向量。

矩阵通常是奇异的，所以LGE算法要先把X集投影到PCA空间，使变成可逆矩阵，然后求解特征值问题。经过PCA处理后上式可通过奇异值分解进行求解：

可解出，设为PCA空间特征向量，则LGE算法的解为：

矩阵是非对称的，所以LGE算法的解往往是非正交的。本节会在接下来介绍正交的LGE算法(OLGE)。

许多常用的线性子空间算法如LPP等，都可以通过定义其权重矩阵W将该算法统一于LGE框架下，通过进行求解。

TODO：参考论文：直接线性图嵌入算法及其人脸识别中的应用

### 2.5.2 LGE算法流程

TODO：源码的算法流程

## 2.6正交线性图嵌入算法(OLGE)

为基于图的子空间学习（正交基矢量）提供一般框架。该算法将由ONPE和OIsoProjection调用。

在LGE算法的基础添加正交迭代获得。

TODO:联系LGE，正交；源码

## 2.7等距映射算法（IsoProjection）

IsoProjection是ISOMAP的线性近似，

TODO：简介，

### 2.7.1 IsoProjection算法推导

设为输入数据集，其中，寻找投影矩阵W使得，其中。Y就是降维后的数据集。

定义一个距离矩阵D，表示到之间的测地距离，定义矩阵S令，令，I为单位矩阵，E为元素全为1的列向量，则可得内积矩阵：。则IsoProjection的目标函数如下：

使用拉格朗日算法将上式的解向量问题转化为求解下式的特征值和特征向量的问题。

所以要求是非奇异矩阵，因此IsoProjection的最佳投影向量就是上式最大特征值所对应的特征向量。

IsoProjection算法求出来的投影向量往往是非正交的，这种非正交的性质会在从高维空间向低维空间的投影过程中扰乱数据的流形结构，而且会使算法本身对降维后的子空间的维数十分敏感，很难估计样本的内蕴维数。

### 2.7.2 IsoProjection算法流程

IsoProjection算法分五步执行：

1. 先使用PCA算法对样本进行降维。因为是奇异矩阵，所以要进行去奇异处理。使用PCA算法的主要目的是将变为非奇异矩阵，然后在低维空间使用IsoProjection算法。
2. 使用K-近邻方法构造近邻图G。
3. 使用迪杰斯特拉算法计算最短路径矩阵D。
4. 计算最佳投影矩阵W。
5. 特征提取。令，则。

TODO：源码中好像没有用到PCA

## 2.8领域保护嵌入算法（NPE）

TODO：简介。参考论文：

### 2.8.1 NPE算法推导

令NPE算法的权重矩阵为S，w为一投影向量，使得满足。则NPE算法的目标函数如下：

对该目标函数做出如下推导：

上式中，令。则简化后的目标函数如下：

则NPE算法的最佳投影向量就是以下特征方程的最小特征值所对应的特征向量。

TODO：再补充的详细一点

### 2.8.2 NPE算法流程

NPE算法的实现分为三步：

1. 构造近邻图。使用K-近邻法寻找与数据点欧氏距离最近的个近邻点。
2. 确定权值。用近邻对各个数据点进行重构。
3. 计算特征映射。在低维空间中，对各个进行数据点重构，即在保持重构权值不变的情况下，使重构误差最小，计算出降维矩阵。

TODO：对每一步再详细介绍

# 3正交化算法

TODO：调成正交的程序，识别率有问题。

## 3.1 OIsoProjection

OIsoProjection算法是在IsoProjection算法的基础上，得到一组正交基向量，因此该算法在保留IsoProjection算法线性的特点的同时又能够保持高维数据的流形结构。

OIsoProjection算法希望通过一组正交基改造目标函数，正交基的求法也相类似，都是通过拉格朗日乘子法引入正交基约束条件来推导出一个特征方程，然后求出该特征方程的最大特征值对应的特征向量以求出正交基，再利用该正交基构造特征方程进行特征提取。

### 3.1.1 OIsoProjection算法推导

首先要计算出一组正交基，则目标函数如下：

则向量可以通过求解特征方程的最大特征值所对应的特征向量获得。

若已有d-1个正交基向量。则可以利用如下正交约束条件获得：。

利用拉格朗日乘子法得到如下方程：

令：，则：

方程两边同时左乘：

得：

将(9.1.1.2)式方程两边同时左乘，得到d-1个式子：

为方便求解做出如下定义：

；

；

；

；

所以就可以用矩阵形式表示这个方程了：

得：

将(9.1.1.2)两边同乘，得：

矩阵表示：

由和得：

得：

得特征方程：

可通过求解上式最大特征值对应的特征向量得到正交基。

所以，OIsoProjection算法的正交投影向量就是特征方程的最大特征值所对应的特征向量。

### 3.1.2 OIsoProjection算法流程

OIsoProjection算法的前三步与IsoProjection算法的步骤相同，此处不再赘述。直接从第四步开始介绍：

4．计算正交投影向量。

通过5.2.1中的算法计算出一组正交基，具体流程如下：

1. 通过求解特征方程的最大特征值所对应的特征向量获得计算。
2. 通过求解特征方程的最大特征值所对应的特征向量获得，。

5. 特征提取。

令通过OIsoProjection算法得到的最佳投影矩阵为d维的，通过PCA算法得到的投影矩阵为，则令，则通过线性嵌入：

可以得到对任意x的d维特征。

## 3.2 ONPE

ONPE算法在NPE算法的基础上添加了正交迭代处理。改进的思路与OIsoProjection算法类似，算法的推导过程也与OIsoProjection算法类似。

TODO：基于NPE；改进思路，算法特点

### 3.2.1 ONPE算法推导

定义邻域保护函数为：

最小化该函数将得到NPE算法，而ONPE算法既要找到一组正交向量又要满足上式最小，因此ONPE算法的目标函数如下：

可以通过求解的最小特征对应的特征向量获得。

若已有k-1个正交向量，则第k正交向量满足一下条件：

利用拉格朗日乘子法得到如下方程：

为方便求解做出如下定义：

；

；

；

经过对目标函数进行一系列运算后（因推导过程冗长繁琐故将详细推导过程放于附录9.1.2），得到如下函数：

令：

则：

通过求解这个特征方程就能得到一组正交投影向量。

所以，ONPE算法的正交投影向量就是特征方程的最大特征值所对应的特征向量。

### 3.2.2 ONPE算法流程

ONPE算法的前两步与NPE算法一样，这里从第三步开始介绍：

3.计算正交投影向量。通过5.2.1中的算法计算出一组正交投影向量。

1. 通过求解的最小特征对应的特征向量获得。
2. 通过求解的最小特征对应的特征向量获得。

4. 特征提取。

令通过ONPE算法得到的最佳投影矩阵为d维的，通过PCA算法得到的投影矩阵为，则令，则通过线性嵌入：

可以得到对任意x的d维特征。

TODO：参考论文：基于支持向量机和流形学习的分类方法研究\_陶晓燕

# 4算法实现

NPE算法、ONPE算法、IsoProjection算法和OIsoProjection算法在ORL人脸库上的识别率。

TODO：完善

## 4.1常用的人脸数据库介绍

使用公开人脸数据库对程序进行测评是评判算法好坏的主要依据。本节将详细介绍目前人脸识别领域常用的几个人脸数据库。

人脸库的图像示例和URL在附录9.2中。

TODO：完善，是不是要删掉几个人脸库？

### 6.1.1 FERET人脸数据库

FERET项目从1993年运行至1997年，由美国国防部Counterdrug技术开发项目办公室和美国国防部高级研究计划局(DARPA)发起。该项目的主要任务是研究开发一个可以协助安全、情报和执法人员执行任务的自动的人脸识别系统。

FERET是目前应用最广泛的人脸数据库，该库由14051个正面、左右侧面的人脸头像组成。FERET库为多姿态、不同光照的灰度人脸图像，FERET库收纳人脸的不同光照、不同角度和不同表情的成像，该库包含以下子集：

1. fa: 正面的人脸图像
2. fb: 是对应于fa中的图像数秒后获取的。
3. ba: 类似于fa中的正面人脸图像。
4. bj: 是对应于ba中的图像数秒后获取的。
5. bk: 在不用灯光的条件下，对应于ba中的图像数秒后获取的。
6. ra: 各种角度的姿态库，角度是连续的，但是角度是没有确切统计过的。

### 6.1.2 Yale人脸数据库

Yale人脸数据库是耶鲁大学计算视觉与控制中心创建的。该库由15个人的人脸图像组成，共有165张图像，每个人有11张图像，分别基于不同的光照变化、表情变化和姿态变化，如眨眼、正常、瞌睡、惊奇、悲伤和高兴，还有佩戴眼镜的人脸图像。

### 6.1.3 Yale B人脸数据库

Yale B人脸库是来自不同光照和不同姿态下的单光源人脸图像，该库由10个人的5850张图像组成。该库主要用于在光照和姿态问题上的建模与分析。

该库包含了在64种光照条件下的9种不同姿态共576种不同的观察条件。该库还包括每种姿态下获得的包含背景的图像，所以，该库中共有5760+90=5850张图像。该库压缩后大约1GB。

### 6.1.4 ORL人脸数据库

ORL人脸库由剑桥大学AT&T实验室创建，该库由40个人，每人10张图像，共400张图像组成。图像均在不同光照、不同角度和不同表情下获得。每张图片的分辨率均为像素。该库的面部表情表情变化主要是笑/不笑和睁眼/闭眼，还有一些其他的面部细节。拍摄的是受试者的正面、垂直头像，允许倾斜或旋转。

### 6.1.5 AR人脸数据库

AR人脸库由西班牙巴塞罗那计算机视觉中心创建，该库包含126人4000多张彩色图像，图像分两次拍摄获得，并且严格控制采集环境，如摄像机参数、摄像机距离和光照环境等。

AR库中的图像都是白种人图像，该库的控制条件是：中性表情；笑；愤怒；尖叫；左侧光照；右侧光照；两侧光照；墨镜；墨镜和左侧光照；墨镜和右侧光照；围巾；围巾和左侧光照；围巾和右侧光照。

### 6.1.6 XM2VTS人脸数据库

英国的XM2VTS人脸库是受欧洲ACTS项目的研究计划M2VTS资助建立的身份认证资料数据库，是一个大范围的多模数据库。该库数据是由英国Surrey大学的295名志愿者在4个不同的时间段的图像和语音视频片段。该数据库是商业收费数据库。

XM2VTS有2360张图像，每张图片的分辨率均为像素，控制条件为：低头；抬头；戴眼镜；不戴眼镜；愤怒；平和；侧脸；正脸；有妆；无妆等等。

该库还有293人的三维模型，以VRML格式存储。

### 6.1.7 CMU PIE人脸数据库

CMU PIE是卡耐基梅隆大学在2000年10月到12月收集的，有68个人的41368张图像，包含大量关于位置、表情和光照变化的图像。该库对基于位置的人脸识别产生很大影响。

## 4.2实验结果

分别运行NPE、ONPE、IsoProjection和OIsoProjection算法，并在下一节对比这些算法，分析这些算法的优缺点。

TODO：完善

### 6.2.1 NPE

以下为NPE算法的运行结果：

每类训练样本数为：9

最佳投影维数为：26

NPE的识别率为：75.00%

程序运行时间为：37.76s

程序运行时间为：37.76s

可见NPE算法的识别率为75%左右。

TODO：分析

### 6.2.2 ONPE

以下为ONPE算法的运行结果：

TODO：

### 6.2.3 IsoProjection

以下为IsoProjection算法的运行结果：

每类训练样本数为：9

最佳投影维数为：27

IsoP的识别率为：85.00%

程序运行时间为：32.90s

可见IsoProjection算法的识别率为85%左右。

TODO：分析

### 6.2.4 OIsoProjection

以下为OIsoProjection算法的运行结果：

TODO：

### 4.3各个算法的性能对比

# 5结束语

TODO：

# 6附录

TODO：附录的简介

## 9.1 算法详细推导流程

### 9.1.1 OIsoProjection算法详细推导过程

TODO：基于流形学习算法的新生儿疼痛表情识别[D].

### 9.1.2 ONPE算法详细推导过程

条件不再赘述，详见5.2.1。

对于以下方程：

令，则：

得：

式两边同时左乘，得到k-1个式子：

5.2.1中定义过：

；

；

；

所以就可以用矩阵形式表示这个方程了：

得：

式两边同时左乘，得：

通过式和式得：

得：

得特征方程：

## 9.2 人脸数据库

TODO：人脸库的图像示例&URL

## 9.3 主要算法的核心代码

详见中国浙江大学计算机科学学院的蔡登教授的个人主页：

<http://www.cad.zju.edu.cn/home/dengcai/Data/DimensionReduction.html>

### 9.3.1 IsoProjection算法

if options.k <= 0 % Always supervised!

Label = unique(options.gnd);

nLabel = length(Label);

G = zeros(nSmp,nSmp);

for i=1:nLabel

classIdx = find(options.gnd==Label(i));

D = EuDist2(data(classIdx,:),[],1);

G(classIdx,classIdx) = D;

end

maxD = max(max(G));

% effectively infinite distance

INF = maxD\*INFratio;

D = INF\*ones(nSmp,nSmp);

for i=1:nLabel

classIdx = find(options.gnd==Label(i));

D(classIdx,classIdx) = G(classIdx,classIdx);

end

Else

switch lower(options.NeighborMode)

case {lower('KNN')}

D = EuDist2(data);

maxD = max(max(D));

% effectively infinite distance

INF = maxD\*INFratio;

[dump,iidx] = sort(D,2);

iidx = iidx(:,(2+options.k):end);

for i=1:nSmp

D(i,iidx(i,:)) = 0;

end

D = max(D,D');

D = sparse(D);

D = dijkstra(D, 1:nSmp);

D = reshape(D,nSmp\*nSmp,1);

infIdx = find(D==inf);

D = reshape(D,nSmp,nSmp);

case {lower('Supervised')}

Label = unique(options.gnd);

nLabel = length(Label);

G = zeros(nSmp,nSmp);

maxD = 0;

for idx=1:nLabel

classIdx = find(options.gnd==Label(idx));

nSmpClass = length(classIdx);

D = EuDist2(data(classIdx,:),[],1);

if maxD < max(max(D))

maxD = max(max(D));

end

if options.k >= nSmpClass

G(classIdx,classIdx) = D;

else

[dump,iidx] = sort(D,2);

iidx = iidx(:,(2+options.k):end);

for i=1:nSmpClass

D(i,iidx(i,:)) = 0;

end

D = max(D,D');

D = sparse(D);

D = dijkstra(D, 1:nSmpClass);

G(classIdx,classIdx) = D;

end

end

% effectively infinite distance

INF = maxD\*INFratio;

D = INF\*ones(nSmp,nSmp);

for i=1:nLabel

classIdx = find(options.gnd==Label(i));

D(classIdx,classIdx) = G(classIdx,classIdx);

end

end

end

S = D.^2;

sumS = sum(S);

H = sumS'\*ones(1,nSmp)/nSmp;

TauDg = -.5\*(S - H - H' + sum(sumS)/(nSmp^2));

TauDg = max(TauDg,TauDg');

[eigvector,eigvalue]=LGE(TauDg, [], options, data);

eigIdx = find(eigvalue < 1e-3);

eigvalue (eigIdx) = [];

eigvector(:,eigIdx) = [];

### 9.3.2 NPE算法

if options.k <= 0 % Always supervised!

W = zeros(nSmp,nSmp);

for ii=1:nSmp

idx = find(options.gnd==options.gnd(ii));

idx(find(idx==ii)) = [];

% shift ith pt to origin

z=data(idx,:)-repmat(data(ii,:),length(idx),1);

C = z\*z'; % local covariance

% regularlization

C = C + eye(size(C))\*tol\*trace(C);

tW = C\ones(length(idx),1); % solve Cw=1

tW = tW/sum(tW); % enforce sum(w)=1

W(idx,ii) = tW;

end

M = (eye(size(W)) - W);

M = M\*M';

M = max(M,M');

M = sparse(M);

Else

switch lower(options.NeighborMode)

case {lower('KNN')}

Distance = EuDist2(data,[],0);

[sorted,index] = sort(Distance,2);

neighborhood = index(:,2:(1+options.k));

case {lower('Supervised')}

Label = unique(options.gnd);

nLabel = length(Label);

neighborhood = zeros(nSmp,options.k);

for idx=1:nLabel

classIdx=find(options.gnd==Label(idx));

Distance=EuDist2(data(classIdx,:),[],0);

[sorted,index] = sort(Distance,2);

neighborhood(classIdx,:) = classIdx(index(:,2:(1+options.k)));

end

end

W = zeros(options.k,nSmp);

for ii=1:nSmp

% shift ith pt to origin

z = data(neighborhood(ii,:),:)-repmat(data(ii,:),options.k,1);

C = z\*z'; % local covariance

%regularlization

C = C + eye(size(C))\*tol\*trace(C);

W(:,ii) = C\ones(options.k,1);% solve Cw=1

% enforce sum(w)=1

W(:,ii) = W(:,ii)/sum(W(:,ii));

end

M = sparse(1:nSmp,1:nSmp,ones(1,nSmp),nSmp,nSmp,4\*options.k\*nSmp);

for ii=1:nSmp

w = W(:,ii);

jj = neighborhood(ii,:)';

M(ii,jj) = M(ii,jj) - w';

M(jj,ii) = M(jj,ii) - w;

M(jj,jj) = M(jj,jj) + w\*w';

end

M = max(M,M');

M = sparse(M);

end

M = -M;

for i=1:size(M,1)

M(i,i) = M(i,i) + 1;

end

[eigvector, eigvalue] = LGE(M, [], options, data);

eigIdx = find(eigvalue < 1e-10);

eigvalue (eigIdx) = [];

eigvector(:,eigIdx) = [];

### 9.3.3 LGE算法

MAX\_MATRIX\_SIZE = 1600;

EIGVECTOR\_RATIO = 0.1;

% SVD

if bPCA

[U, S, V] = mySVD(data);

[U, S, V]=CutonRatio(U,S,V,options);

eigvalue\_PCA = full(diag(S));

if bD

data = U\*S;

eigvector\_PCA = V;

DPrime = data'\*D\*data;

DPrime = max(DPrime,DPrime');

else

data = U;

eigvector\_PCA = V\*spdiags(eigvalue\_PCA.^-1,0,length(eigvalue\_PCA),length(eigvalue\_PCA));

end

else

if ~bChol

if bD

DPrime = data'\*D\*data;

else

DPrime = data'\*data;

end

switch lower(options.ReguType)

case {lower('Ridge')}

if options.ReguAlpha > 0

for i=1:size(DPrime,1)

DPrime(i,i) = DPrime(i,i) + options.ReguAlpha;

end

end

case {lower('Tensor')}

if options.ReguAlpha > 0

DPrime = DPrime + options.ReguAlpha\*options.regularizerR;

end

case {lower('Custom')}

if options.ReguAlpha > 0

DPrime = DPrime + options.ReguAlpha\*options.regularizerR;

end

end

DPrime = max(DPrime,DPrime');

end

end

WPrime = data'\*W\*data;

WPrime = max(WPrime,WPrime');

% Generalized Eigen

dimMatrix = size(WPrime,2);

if ReducedDim > dimMatrix

ReducedDim = dimMatrix;

end

if isfield(options,'bEigs')

bEigs = options.bEigs;

else

if (dimMatrix > MAX\_MATRIX\_SIZE) && (ReducedDim < dimMatrix\*EIGVECTOR\_RATIO)

bEigs = 1;

else

bEigs = 0;

end

end

if bEigs

%disp('use eigs to speed up!');

option = struct('disp',0);

if bPCA && ~bD

[eigvector, eigvalue] = eigs(WPrime,ReducedDim,'la',option);

else

if bChol

option.cholB = 1;

[eigvector, eigvalue] = eigs(WPrime,R,ReducedDim,'la',option);

else

[eigvector, eigvalue] = eigs(WPrime,DPrime,ReducedDim,'la',option);

end

end

eigvalue = diag(eigvalue);

else

if bPCA && ~bD

[eigvector, eigvalue] = eig(WPrime);

else

[eigvector, eigvalue] = eig(WPrime,DPrime);

end

eigvalue = diag(eigvalue);

[junk, index] = sort(-eigvalue);

eigvalue = eigvalue(index);

eigvector = eigvector(:,index);

if ReducedDim < size(eigvector,2)

eigvector = eigvector(:, 1:ReducedDim);

eigvalue = eigvalue(1:ReducedDim);

end

end

if bPCA

eigvector = eigvector\_PCA\*eigvector;

end

for i = 1:size(eigvector,2)

eigvector(:,i) = eigvector(:,i)./norm(eigvector(:,i));

end

%function CutonRatio

function [U, S, V]=CutonRatio(U,S,V,options)

if ~isfield(options, 'PCARatio')

options.PCARatio = 1;

end

eigvalue\_PCA = full(diag(S));

if options.PCARatio > 1

idx = options.PCARatio;

if idx < length(eigvalue\_PCA)

U = U(:,1:idx);

V = V(:,1:idx);

S = S(1:idx,1:idx);

end

elseif options.PCARatio < 1

sumEig = sum(eigvalue\_PCA);

sumEig = sumEig\*options.PCARatio;

sumNow = 0;

for idx = 1:length(eigvalue\_PCA)

sumNow = sumNow + eigvalue\_PCA(idx);

if sumNow >= sumEig

break;

end

end

U = U(:,1:idx);

V = V(:,1:idx);

S = S(1:idx,1:idx);

end

# 参考文献

[1] Deng Cai, Xiaofei He, and Jiawei Han. Isometric Projection[R]. Department of Computer Science, University of Illinois at Urbana-Champaign: Yahoo! Research Labs, 2006.

[2] 徐勇，范自柱，张大鹏. 基于稀疏算法的人脸识别[M]. 北京: 国防工业出版社，2014.

[3] 王庆军，张汝波，刘冠群. 一种应用于人脸识别的核正交等度规映射算法[N]. 光电子激光，2010(11).

[4] 王庆军. 流形学习算法分析及应用研究[D]. 哈尔滨工程大学: 计算机科学与技术学院，2008.

[5] 王庆军. 基于流形学习子空间的人脸识别方法研究[D]. 哈尔滨工程大学: 计算机科学与技术学院，2011.

[6] 王庆军，张汝波，潘海为. 基于核正交局部判别嵌入的人脸识别[N]. 光电子激光，2010(9).

[7] 王庆军，张汝波，楼宋江，吕海燕. 一种核正交局部敏感辨别分析算法[N]. 小型微型计算机系统，2009(11).

[8] 王庆军，张汝波. 基于LogGabor和正交等度规映射的人脸识别[N]. 计算机科学，2011(2).

[9] 古楠楠，樊明宇，王迪，韩志. 流形学习若干关键问题与算法研究[M]. 北京: 首都经济贸易大学出版社，2015.

[10] 秦鸿，李泰峰，郭亨艺，许毅. 人脸识别技术在图书馆的应用研究[N]. 大学图书馆学报，2018(6).

[11] 徐蓉，姜峰，姚鸿勋. 流形学习概述[N]. 智能系统学报，2006(1).

[12] 高小方. 流形学习方法中的若干问题分析[N]. 计算机科学，2009(4).

[13] 王自强，钱旭，孔敏. 流形学习算法综述[N]. 计算机工程与应用，2008(35).

[14] 李波. 基于流形学习的特征提取方法及其应用研究[D]. 中国科学技术大学: 中国科学技术大学研究生院，2008.

[15] 赵连伟，罗四维，赵艳敞，刘蕴辉. 高维数据流形的低维嵌入及嵌入维数研究[D]. 软件学报，2005(16).

[17] 张军平. 流形学习及应用[D]. 中国科学院自动化研究所： 中国科学院研究生院，2003.

[18] 孟德宇，徐晨，徐宗本. 基于Isomap的流形结构重建方法[N]. 计算机学报，2010(3).

[19] 李小丽，薛清福. 几种流形学习算法的比较研究[N]. 电脑与信息技术，2009(3).

[20] 王靖. 流形学习的理论与方法研究[D]. 浙江大学: 理学院，2006.

[21] 程起才，王洪元，刘爱萍，冯燕. 基于ISOMAP的一种多流形学习算法[N]. 微电子学与计算机，2009(10).

[22] 詹宇斌. 流形学习理论与方法及其应用研究[D]. 国防科学技术大学: 国防科学技术大学研究生院，2011.

[23] 黄结. 基于正交局部敏感辨别分析的人脸识别方法研究[D]. 鲁东大学: 信息与电气工程学院，2013.

[24] 吴赣昌. 线性代数[M]. 北京: 中国人民大学出版社，2011.

[25] 陈江峰. 线性图嵌入算法及其应用[D]. 北京交通大学: 信号与信息处理，2012.

[26] 陈江峰，袁保宗. 直接线性图嵌入算法及其在人脸识别中的应用[N]. 电子与信息学报，2010(6).

[27] 严蔚敏，吴伟民. 数据结构（C语言版）[M]. 北京: 清华大学出版社，2007.

[28] 郭鹤楠. 基于NPE和LDCRF的人体运动识别[D]. 吉林大学: 计算机科学与技术(生物信息学)，2011.

[29] 刘嘉敏，李连泽，罗甫林，刘亦哲，刘玉梅. 相关NPE算法的人脸识别研究[N]. 计算机应用研究，2015(6).

[30] 胡凡君，张勤，李鹏，苗爱敏，邹勋，陈霍兴. 基于NPE算法的环网柜故障检测方法研究[N]. 自动化仪表，2017(10).

[31] 仝一君，王力. 基于NPE算法的语音特征提取应用研究[N]. 通信技术，2014(11).

[32] 左加阔. 基于流形学习算法的新生儿疼痛表情识别[D]. 南京邮电大学: 信号与信息处理，2011.

[33] 陶晓燕，姬红兵，景志宏. 一种用于人脸识别的正交邻域保护嵌入算法[N]. 西安电子科技大学学报（自然科学版），2008(6).

[34] 陶晓燕. 基于支持向量机和流形学习的分类方法研究[D]. 西安电子科技大学: 模式识别与智能系统，2008.

[35] 刘韵佳，赵荣珍，王雪冬. 基于Schur分解和正交邻域保持嵌入算法的故障数据集降维方法[N]. 中国机械工程，2017(21).

[36] 刘韵佳. 基于Schur-ONPE的转子故障数据集降维方法研究[D]. 兰州理工大学: 机械制造及其自动化，2017.

[37] 陈法法，杨晓青，陈保家，程珩，肖文荣. 基于正交邻域保持嵌入与多核相关向量机的滚动轴承早期故障诊断[N]. 计算机集成制造系统，2018(8).

[38] 李锋，汤宝平，董绍江. 基于正交邻域保持嵌入特征约简的故障诊断模型[N]. 仪器仪表学报，2011(3).

[39] 苗爱敏. 数据局部时空结构特征提取与故障检测方法[D]. 浙江大学，2014.

[40] 季云峰，冯立元，匡亮. 基于改进的有监督正交邻域保持嵌入的故障辨识[N]. 机械传动，2017(1).

[41] 陈江峰. 线性图嵌入算法及其应用[D]. 北京交通大学: 信号与信息处理，2012.

# 致 谢

因为

TODO：