

手写体古籍汉字识别研究

(申请清华大学工学硕士学位论文)

培养单位：电子工程系

学 科：信息与通信工程

研 究 生：冯 继 雄

指 导 教 师：彭 良 瑞 副 教 授

二〇一五年六月

Research on Handwritten Historical Chinese Character Recognition

Thesis Submitted to
Tsinghua University
in partial fulfillment of the requirement
for the degree of
Master of Science
in
Information and Communication Engineering
by
Feng Jixiong

Thesis Supervisor : Associate Professor Peng Liangrui

June, 2015

关于学位论文使用授权的说明

本人完全了解清华大学有关保留、使用学位论文的规定，即：

清华大学拥有在著作权法规定范围内学位论文的使用权，其中包括：（1）已获学位的研究生必须按学校规定提交学位论文，学校可以采用影印、缩印或其他复制手段保存研究生上交的学位论文；（2）为教学和科研目的，学校可以将公开的学位论文作为资料在图书馆、资料室等场所供校内师生阅读，或在校园网上供校内师生浏览部分内容。

本人保证遵守上述规定。

（保密的论文在解密后应遵守此规定）

作者签名：_____

导师签名：_____

日 期：_____

日 期：_____

摘要

中文古籍记载着灿烂而悠久的中华文明，具有巨大的研究价值。在建设数字图书馆的过程中，文档识别技术能有效降低古籍全文数字化的成本。然而，将OCR（Optical Character Recognition，光学字符识别）技术直接应用于古籍文档时遇到了极大的困难，其原因主要包括古籍汉字的类别多、字体变形丰富、版面格式多样、版面破损较严重以及训练样本少等。

本文提出了GP-STM（Gaussian Process Style Transfer Mapping，高斯过程迁移学习映射）古籍汉字识别模型，根据迁移学习的理念，将非线性的高斯过程回归方法应用在风格迁移映射中，充分利用已有的印刷体汉字样本来增强手写体古籍汉字的识别性能；

本文还提出了基于张量分解的古籍汉字识别算法，将训练样本的特征向量由特征、样本和类别三个维度组成一个张量，然后通过张量的Tucker分解保留对分类有用的信息，对手写体古籍汉字进行测试，取得了较好的实验结果；

本文的工作还包括一个在线古籍图像识别系统原型，通过.NET Framework网站后台编程技术，将现有的版面预处理、汉字字符切割和汉字识别的相关研究成果整合到网站的后台程序中，预期可为古籍研究者提供相关古籍识别的网络服务。

关键词：高斯过程；迁移学习；张量分解；在线古籍识别系统

Abstract

Historical Chinese documents which reflect Chinese civilization are invaluable collections in many libraries both in China and abroad. Historical Chinese document recognition technology is very important to facilitate the related full text digitalization projects for digital libraries.

However, it is a very challenging problem in OCR (Optical Character Recognition) research field due to large character set of historical Chinese characters, variant font types, versatile document layout styles, the degradation of image quality, and the lack of labeled training samples.

This thesis focuses on historical Chinese character recognition with the following innovations and works:

The proposed GP-STM historical Chinese character recognition model, which uses non-linear Gaussian process regression for style transfer mapping, incorporates the printed Chinese characters to enhance the accuracy of OCR;

The proposed recognition method based on tensor decomposition, which builds a tensor from the feature vectors of training samples in dimensions of feature, sample and class to get helpful information for classification through Tucker decomposition, achieves a good result in experiments;

The developed on-line historical document recognition prototype system, which integrates the pre-processing, character segmentation and recognition methods with .NET Framework, hopefully makes the web service accessible to related researchers.

Key words: Gaussian Process; Transfer Learning; Tensor Decomposition; On-line Recognition System

目 录

第1章 引言	1
1.1 古籍识别概述	1
1.1.1 古籍识别背景	1
1.1.2 古籍汉字识别的关键问题	2
1.1.3 古籍汉字的识别方法	3
1.2 本文研究的主要内容和主要贡献	3
1.2.1 主要研究内容	3
1.2.2 各章内容简介	4
1.2.3 本文的主要贡献	4
第2章 文献调研与相关工作	5
2.1 本章引论	5
2.2 古籍数字化进展	5
2.2.1 古籍数字化简介	5
2.2.2 古籍数字化项目介绍	6
2.2.2.1 国际敦煌项目IDP	6
2.2.2.2 欧洲IMPACT古籍识别项目	7
2.3 汉字识别常用方法	8
2.3.1 预处理	8
2.3.2 特征提取	8
2.3.3 分类器设计	9
2.4 迁移学习及应用	9
2.4.1 迁移学习介绍	9
2.4.2 线性迁移学习方法及其在古籍汉字识别中的应用	10
2.5 高斯过程	10
2.6 本章小结	11
第3章 GP-STM模型介绍	12
3.1 本章引论	12
3.2 模型基本假设	12
3.2.1 回归问题	13
3.2.2 高斯过程	14

3.2.3 相同协方差矩阵	15
3.3 GP-STM模型	16
3.3.1 模型概述	16
3.3.2 变换公式	17
3.4 GP-STM参数的优化方法	18
3.4.1 最大似然概率	18
3.4.2 核函数的偏导数	19
3.5 本章小结	19
第4章 实验结果及分析	20
4.1 本章引论	20
4.2 实验样本介绍	20
4.2.1 敦煌古籍数据库介绍	20
4.2.2 中科院手写汉字数据库介绍	20
4.3 实验设置	21
4.4 与传统方法对比	22
4.5 STM训练集比例与识别率的关系	23
4.6 核函数选择	24
4.6.1 核函数介绍	24
4.6.2 不同核函数的对比	25
4.7 GP-STM模型理解	25
4.7.1 协方差矩阵拟合	26
4.7.2 可视化展示	26
4.7.3 识别结果分析	28
4.8 本章小结	28
第5章 基于张量分解的字符识别	29
5.1 本章引论	29
5.2 张量简介	29
5.2.1 张量运算	29
5.2.2 张量的CP分解	30
5.2.3 张量的Tucker分解	30
5.3 识别过程	31
5.3.1 预处理	31
5.3.2 训练	32
5.3.3 识别	32

5.3.4 改进的识别方法	33
5.4 实验结果分析	33
5.4.1 手写数字MNIST数据库	33
5.4.1.1 识别率与压缩率	33
5.4.1.2 不同特征的对比	34
5.4.1.3 结果分析	34
5.4.2 敦煌古籍数据库	35
5.5 本章小结	36
第6章 在线古籍识别系统	37
6.1 本章引论	37
6.2 在线OCR平台介绍	37
6.3 系统应用方案选择	38
6.3.1 B/S和C/S结构	38
6.3.2 .NET Framework平台	38
6.3.3 C#图像编程	39
6.4 系统构成与实现	40
6.5 系统使用演示	41
6.6 总结	41
第7章 结论	43
7.1 研究结论	43
7.2 需要进一步开展的工作	44
参考文献	45
致 谢	49
声 明	50
附录A 汉字识别常用技术	51
A.1 预处理	51
A.1.1 字符切分	52
A.2 字符的特征提取和降维	52
A.2.1 特征提取	52
A.2.2 特征降维	55
A.3 分类器设计	55
A.3.1 二次判别函数	56
A.3.2 MQDF分类器	57

目 录

附录 B 汉字BIG5编码以及存储格式	58
B.1 BIG5编码	58
B.2 PNT格式介绍	58
附录 C MATLAB编写.NET组件概述	59
个人简历、在学期间发表的学术论文与研究成果	61

主要符号对照表

A	风格迁移映射的系数、张量
$A \times_n F$	张量A与矩阵F的n模式乘积
$A_{(1)}, A_{(2)}, A_{(3)}$	张量A的1、2、3模式展开
b	风格迁移映射的常数项
β	STM的高层参数
c	类别数
C_X, C_Y	图像在横、纵方向的投影的总和
d, d'	映射前后特征向量的维数
ϕ	矩阵的特征向量
$f(i,j)$	图像f在第i行第j列的像素值
γ	STM的高层参数，控制b更像零向量
GP-STM	高斯过程风格转换映射(Gaussian Process STM)
H	图像的高 (Height)
h	MQDF截断维数缺省的特征值
h_X, h_Y	图像在横、纵方向的投影函数
HMM	隐马尔可夫模型 (Hidden Markov Model)
HOG	方向梯度直方图(Histogram of Oriented Gradient)
k	MQDF的截断数
K	GP-STM的核 (协方差矩阵)
$k(i,j)$	GP-STM的核函数
λ	矩阵的特征值
L_1, \dots, L_4	像素点最近笔画的对应点
L_X, L_Y	像素点横竖方向最近笔画的内切圆
LBP	局部二值模式 (Local Binary Pattern)
LDA	线性判别分析 (Linear Discriminant Analysis)
μ_i	第i类的均值向量
MQDF	改进的二次分类器
N	特征向量的个数
N_i	第i类的特征向量个数
$N(\mu, K)$	正态分布，均值向量为 μ ，协方差矩阵为K
PCA	主成分分析 (Principal Component Analysis)

主要符号对照表

PNT	汉字样本库的一种文件存储格式
$\rho(i, j)$	像素(i,j)点的线密度
s	源向量 (Source Vector)
S	源数据集 (Source Dataset)
Σ	协方差矩阵
S_B	类间散度矩阵
S_T	全体散度矩阵
S_W	类内散度矩阵
SIFT	尺度不变特征转换 (Scale-Invariant Feature Transform)
STM	风格转换映射(Style Transfer Mapping)
SVM	支持向量机(Support Vector Machine)
t	目标向量 (Target Vector)
T	目标数据集 (Target Dataset)
θ	GP-STM的参数
W	图像的宽 (Width); 也作映射矩阵
W_{opt}	最优的映射
WDH	加权方向编码直方图 (Weighted Direction Code Histogram)
\mathbf{x}^*	一个新的特征向量
$\mathbf{x}_1, \dots, \mathbf{x}_N$	N个特征向量

第1章 引言

中华文明历史悠久，古籍文卷更是浩如烟海。古籍中所记载的政治、文化、军事、地理、医药等各方面的内容，具有巨大的研究价值，众多学者不惜终其一生，研其要义。随着信息技术的发展，为了更好地保护古籍，越来越多的图书馆将古籍数字化。本研究试图从古籍汉字的识别作为切入点，为古籍全文数字化的研究提供便利和帮助。

1.1 古籍识别概述

古籍识别是指计算机古籍文档图像识别，即把古籍文档的数码照片，通过计算机处理，生成可供计算机编辑和检索的文本^[1]。下面先介绍一下古籍识别的背景、问题和常见方法。

1.1.1 古籍识别背景

当今，信息技术已经扩展到社会的方方面面，古籍数字化旨在将古籍读入计算机，让研究者能够在电脑上阅读、检索、标注古籍。这样做对古籍的保存非常有帮助，因为古籍经过拍照后就不用被经常翻动，而且研究者在研究过程中不受时间和空间的限制，因此目前世界上很多图书馆都在积极推动古籍数字化。

在数字化的过程中，首先扫描拍照，用高清照相机拍摄古籍的全貌（如图1.1所示）；其次版面处理，勾勒出文字、图片、注释等区域，在文字版面中找出单独的汉字；再次汉字识别，应用OCR技术或者人工来录入文字；最后编辑入库，将可编辑、检索的文本文档整理后存入数据库中。

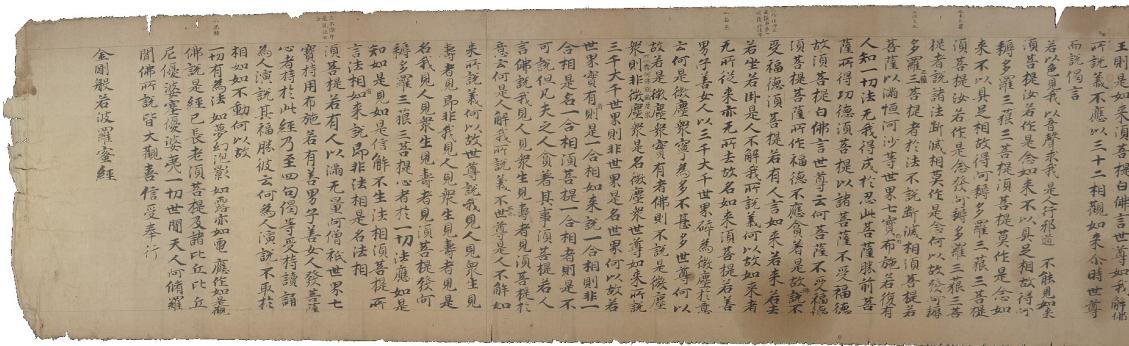


图 1.1 古籍图像的示例

在古籍数字化的大潮中，已经产生了海量的古籍图片，一些问题接踵而来：图片作为多媒体数据，占用存储空间极大，不能检索和编辑。作为古籍数字化工程的重要一环，古籍识别恰恰可以解决这些问题。它能够高效地把这些图像处理为文本，避免了低效率的人工录入，具有重要的研究意义。

1.1.2 古籍汉字识别的关键问题

手写汉字识别已经取得了很大的进步，但古籍汉字识别仍然存在困难，这主要有三个原因：

首先，标记数据不足。标记数据是研发识别系统的基础资源，每个标记点由特征向量和类别标签构成，字符分类器往往需要利用其类别标签来建立模型，然而由于标记需要较大的人力物力，目前还没有完整的标记古籍字符库，现有从敦煌古籍得到的一些标记样本如表1.1所示（其中PNT存储格式详见附录B.2），标记样本的不足使得难以用古籍汉字样本直接建立字符分类器。

表 1.1 敦煌古籍样本统计

数据集	4548.pnt	4603.pnt	4627.pnt	4628.pnt	5383.pnt
字符数	11,154	6,322	7,992	6,139	7,686
类别数	1,372	510	1,583	1,161	1,139
数据集	5390.pnt	5405.pnt	5419.pnt	5472.pnt	总计
字符数	8,373	2,204	3,983	5,760	59,613
类别数	1,596	514	1,036	1,524	3,068

其次，书写风格不同。不同古籍文档具有不同的书写风格，甚至同一个古籍文档的同一个字也有不同写法（如图1.2所示），这为古籍汉字识别增加了难度。



图 1.2 同一个古籍文档中的“我”字

第三，笔画残缺冗余。因为年代久远和保存不善，字符图像被不同程度损坏，造成笔画残缺；同时，由于笔画粘连较为严重，在字符分割时无法做到有效分割，导致笔画冗余，影响识别。

1.1.3 古籍汉字的识别方法

古籍汉字识别首先借鉴传统的汉字识别技术。传统的汉字识别主要采用提取特征向量、降维、用分类器分类的方法。特征向量的种类有许多种，比如常见的有方向线索^[2]、统计图、LBP^[3]和HOG^[4]等等，主要目的尽可能多地提取信息，同时又要兼顾噪声干扰，增加鲁棒性。降维是把高维空间中的特征向量转换成低维空间中的特征向量，目的是在维数减少的同时尽可能多地保留有利于分类的信息。常用的降维方式有PCA（Principal Component Analysis，主成分分析）和线性判别分析（LDA，Linear Discriminative Analysis）等。分类器的输入是特征向量，输出是类别。一般而言，分类器需要训练，即把带有标签的特征向量通过处理，找出不同类之间的差异，得到合适的参数，进而便于对未知类别的特征向量分类。常见的分类器有线性判别函数、MQDF、SVM、神经网络和HMM等^[5]。识别过程如图1.3所示。



图 1.3 古籍汉字识别的一般过程

现代印刷体汉字或手写体汉字的数据库较多，比如HCL2000^[6]、HIT-MW^[7]和CASIA-HWDB^[8]。为了利用这些数据库来帮助古籍汉字的识别，可以考虑以下几种方法。常用的方法是寻找古籍汉字和印刷体汉字的公共特征，这可以由多任务学习来解决^[9,10]。另一个方法是在迁移学习的框架内看待这个问题，把古籍汉字所在的特征域变换到印刷体汉字所在的特征域^[11,12]，比如语音识别就经常采用迁移学习方法^[12]，这可以为古籍汉字识别提供新的思路。

1.2 本文研究的主要内容和主要贡献

1.2.1 主要研究内容

本文研究手写体古籍汉字识别，引入迁移学习的概念，构建识别模型，提高手写体古籍汉字的识别率。古籍文档图像的版面处理、字符分割不是本文的研究内容，本文的研究对象是分割好的单个字符图像，此外，特征提取、降维和分类器都会用到，但不作为研究重点。

1.2.2 各章内容简介

本文其余部分组织如下：第2章中介绍文献调研和相关工作；第3章中介绍GP-STM模型的框架和参数选取；第4章中介绍为验证GP-STM模型开展的实验及其结果；第5章中介绍基于张量分解的字符识别方法；第6章中介绍在线古籍图像识别系统；第7章是结论和未来工作展望。

1.2.3 本文的主要贡献

本文重点研究基于高斯过程风格转换映射（GP-STM）的古籍汉字识别方法。与此同时，本文还介绍了一些与古籍识别相关的其他研究探索。主要贡献如下：

- 提出了GP-STM（Gaussian Process Style Transfer Mapping，高斯过程迁移学习映射）古籍汉字识别模型，基于迁移学习的思想，在风格迁移映射中应用非线性的高斯过程回归方法，充分利用已有的印刷体汉字数据库来增强手写体古籍汉字的识别性能；
- 提出了基于张量分解的古籍汉字识别算法，将训练样本的特征向量由特征、样本和类别三个维度组成一个张量，然后通过张量的Tucker分解保留对分类有用的信息，对手写体古籍汉字进行测试；
- 开发了一个在线古籍图像识别系统的原型，通过.NET Framework网站后台编程技术，将现有的版面预处理、汉字字符切割和汉字识别的相关研究成果整合到网站的后台程序中，预期可为古籍研究者提供相关古籍识别的网络服务。

第2章 文献调研与相关工作

2.1 本章引论

本章中将会详细介绍与本文研究相关的文献和工作，从古籍数字化的进展开始介绍，详述汉字识别的常用方法，再介绍迁移学习的概念及其在古籍汉字识别中的应用，最后介绍高斯过程的特点及其在非线性迁移学习中的应用。

2.2 古籍数字化进展

古籍识别伴随着古籍数字化的兴起而成为一项研究热点。一些重大的古籍识别项目也随之展开，下面先介绍古籍数字化的概念和主要技术，然后再介绍两个国际上具有典型意义的古籍数字化项目。

2.2.1 古籍数字化简介

随着人类进入以计算机技术为核心的信息时代，越来越多的知识以数字格式保存，数字格式的视频、图像和文本等信息在复制、传播和储存上都非常方便。古籍因为其年代久远，不易保存，在使用上存在诸多不便，从保护和利用古籍的目的来看，用信息技术将古籍书卷的文本、插图转化为数字格式，进而以数据库的形式保存，同时提供查询和检索等服务，这一系列工作共同构成了古籍数字化^[13]。

古籍数字化主要包含两个方面内容，一个是数字化技术，即用信息技术获取数字格式的图文信息；另一个是建设古籍书目的数据库，这需要考虑主题分类、著录和主题标引等图书馆学概念；二者的发展相辅相成，其中数字化技术的发展又可以细化为三个阶段^[14]：

第一阶段是探索阶段，从20世纪70年代到90年代，这一阶段遇到的主要问题是古籍录入计算机时缺少相应的大字符集支持，因此不少研究者认为古籍数字化技术的基础工作是汉字字符集的设计，比如在1993年的古籍现代化技术研讨会上，有学者提出的大字符集中文计算机平台；第二阶段是产品输出阶段，从20世纪90年代到21世纪初，这一阶段在图文录入和识别方面进行了探索，光学字符识别（Optical Character Recognition, OCR）扫描录入技术成为一种重要的录入方式，然而由于OCR技术在处理较大版面的古籍时比较麻烦，同时其录入速度慢，这种

方式逐渐被日益成熟的数码相机拍摄取代，OCR技术可以用在后期的图像处理中^[15,16]；第三阶段是21世纪初到现在，古籍数字化在自动化识别^[17,18]、全文检索和知识库等技术都取得了较大的发展，一些少数民族语言的古籍也开始参与到古籍数字化的研究中，大型的古籍数据库可以通过网络的方式供读者浏览，比如大学数字图书馆国际合作计划(China Academic Digital Associative Library, CADAL)^①。

在古籍数字化的概念被提出后，许多数字化技术被提出并被用于实践。比如王婷婷等使用二值化、对比度拉伸和直方图均衡等方法对古籍图像进行增强处理，提高了古籍图像的质量^[19]；姜哲等采用自顶向下与自底向上的方法结合、手动修正与自动处理结合的方法，对《四库全书》进行版面分析，并且取得了良好的实验效果^[20]；朱雷针对现有字符切割方法噪声大等的不足，提出了整体阈值与局部阈值相结合的算法和快速非递归连通域生成及合并算法，使古籍图像的汉字切分取得了较好成果^[21]；衡中青从目录学角度出发，用内容挖掘的方法，对地方志《方志物产·广东》进行信息系统的构建、物产研究和引书研究^[22]；贾雪莎结合汉字的特点，采用“先聚类后检索”的策略，设计了古籍图像的局部或全局检索方法^[23]；张彩录等采用Shannon理论，讨论了印刷体古籍汉字字域选择所受的约束、字符特征的性能所达到的上限和用统计特性来提高系统识别率，其完成的古籍识别系统对720万字的《续资治通鉴长编》能达到35字/秒的识别速度，95%的识别率^[24]；李璐等深入研究古籍全文数据库的原理，并较为完整地展现了《四库全书》电子版的开发过程^[1]。

2.2.2 古籍数字化项目介绍

目前，国内外已完成或正在进行一些古籍数字化的工程，下面重点介绍中外合作的国际敦煌项目（International Dunhuang Project, IDP）和欧洲IMPACT（IMProving ACCess to Text）项目。

2.2.2.1 国际敦煌项目IDP

敦煌古籍与殷墟甲骨文、南阳汉画像石和北京猿人头骨并列中国近代学术史上的四大史料发现，有关专家估计和统计数据表明，敦煌古籍中佛经占85%，经史子集及各种世俗文书占15%，为研究中古时期的中国和中西亚国家的政治、经济、文化、社会以及丝绸之路提供了翔实可信的资料，是中华民族的珍贵遗产^[25]。然而由于历史原因，敦煌古籍有很多散落在英国、法国、日本和美国等国家（见表2.1统计）。

① CADAL主页：<http://www.cadal.zju.edu.cn/index>

表 2.1 IDP图片分布

总计 ^①	英国	中国	法国	德国	俄罗斯	日本	敦煌
454,490	158,834	127,572	69,492	55,684	22,672	17,364	2,872

① 资料来自<http://idp.nlc.gov.cn/idp.a4d>

IDP是一个由多个国家协作研究的项目，旨在使世界各地的研究者能自由地从互联网上获取敦煌及丝绸之路东段其他考古遗址出土的写本、绘画、纺织品以及艺术品的信息与图像，同时通过开展研究项目鼓励研究者来使用^①。目前，IDP网站上提供的数字化敦煌古籍为图像形式，尚缺少全文数据，因此亟需研究古籍识别技术，最终实现古籍全文检索，便于研究者使用。

IDP的研究项目主要有敦煌和复原艺术、藏文和密教古籍编目项目、中国古文书学研究和丝绸之路的相关研究。其中保护修复是IDP的核心工作。自1994年IDP成立以来，IDP还出版了《早期中文著作的正字现象：来自新发现的写本证据》、《丝绸之路：商贸、旅行、战争与信仰》和《大英图书馆保存保护科学》等著作，为敦煌学的研究起到了很好的促进作用。

2.2.2.2 欧洲IMPACT古籍识别项目

近年来，欧洲的图书馆扫描了百万计的古籍文档，广大古籍研究者却迟迟难以获取这些文档，其中的障碍有三点：首先，没有一个统一的专业化机构来指导，缺乏效率；其次，制作全文电子书的成本非常高，比如人工打字往往需要每页1欧元，这样一本书将需要近1,000欧元；最后，OCR技术还有待进一步提高，识别效果还不能尽如人意。2008年1月，IMPACT在欧盟委员会资助下成立^②，旨在让研究者更方便地得到古籍，同时清除欧洲文化遗产大规模数字化进程中的这些障碍。

IMPACT项目为期4年，与26个国家和地区的图书馆、研究机构及商业赞助合作。项目将分为4个子项目。首先是业务联系（Operation Context），用来保障项目的合作；其次是文本识别（Text Recognition），用来开发古籍的OCR技术；然后是图像增强（Enhancement and Enrichment），旨在使OCR结果更加精确可靠；最后是容量建造（Capacity Building），用来激励和支持IMPACT成果。IMPACT项目在研究中，制作了许多和自动化识别有关的工具，比如自适应OCR引擎、在线协作系统、图像增强工具、字符分割工具和校对工具等。

① IDP主页：<http://idp.nlc.gov.cn/>

② IMPACT主页：<http://www.impact-project.eu/home/>

2.3 汉字识别常用方法

汉字识别是模式识别的重要分支，根据书写者可以分为印刷体汉字识别和手写体汉字识别，其中手写体汉字根据书写方式又可分为联机和脱机手写汉字，脱机手写体汉字识别因为其完全没有任何笔画顺序信息，同时有着不同书写风格，因此极具挑战，一般来讲，汉字识别分为预处理、特征提取和模式分类三个部分^[26]。其具体的技术细节详见附录A。

2.3.1 预处理

在预处理中，为了减小计算量，首先做的是二值化和降噪处理，然后做字符切分。字符切分是从文档图像中分割出单独的汉字字符，从而便于汉字识别。目前字符的切分算法主要有直接切分和基于识别的切分^[26]。

直接切分根据字符的物理特征（宽高等）来进行切分，其中有几种方法。第一种是根据字符之间的白色空隙来分割。在一些印刷体中，每个字的大小都是基本相同的，因此可以找到一个固定的字长为字符分割提供基准。第二种是根据投影分割。对一行字进行水平投影，得到一个直方图。它可以为找到字符之间的空隙来提供线索，也可以表示一些笔画。如果字符之间有重叠，投影直方图也会在适合切分的点有一个极小值，在很多情况下可以检查出来。第三种是连通域分析，当字符排列较紧密且大小不一时，前两种方法效果比较差，这个时候可以考虑进行连通域分析，找到所有连通的黑色区域^[27]。

基于识别的切分主要分为两个步骤：找到切分的窗以及选择最优的切分方法^[28]。找到窗的一种方法是搜索整个图像，但是很显然这样做的复杂度会比较高，特别是对汉字来说时间的开销基本是无法接受的；也可以根据图像的特征描述来进行切分，这里又细分为隐马尔可夫模型与非马尔可夫方法，其中隐马尔可夫模型会选择使后验概率最大化的路径^[29]。对于重叠字符可以采用Viterbi动态算法^[30]。在古籍识别中采用基于识别的切分可以取得较好的实验成果^[31,32]。

2.3.2 特征提取

为了有效地提取字符图像的本质特征，一般汉字在分类识别前会进行特征提取，而不是单纯用像素来识别。常用的特征分为结构特征和统计特征^[26]。

结构特征针对汉字的某个局部，是汉字识别研究初期常用的方法，和人们书写汉字的过程类似，能较好地反映汉字的结构特点，主要方法有：特征点^[33]、笔画^[34,35]和部件分解^[36,37]。结构特征的不足是基本单元提取困难，通常要先进行细化处理，计算量大且抗干扰性较差^[26]。

统计特征针对整个汉字提取特征，这样做抗干扰性更强^[26]。这类特征有弹性网格^[38,39]、方向线索^[40,41]、Gabor特征^[42]和矩特征^[43]。在识别手写汉字的时候，往往会将多种特征结合，采用PCA和LDA等方法进行特征降维处理后再送入分类器分类。

2.3.3 分类器设计

汉字识别面临的是类别数为几千甚至上万的分类问题。常用的方法有改进的二次判别函数（MQDF，Modified Quadratic Discrimination Function）^[44]、支持向量机（SVM，Support Vector Machine）^[45]、人工神经网络^[46]和隐马尔可夫模型（HMM，Hidden Markov Model）^[47,48]。为了利用每种分类器的优点，可以采用多分类器组合的方法，通过串行^[49]、并行^[50]或者混合方法组合多个分类器，提高汉字识别精度。

2.4 迁移学习及应用

本文的研究中引入了迁移学习的思想，尝试用其他汉字字库训练分类器，通过迁移学习的手段，来构建古籍汉字的识别系统。下面先介绍迁移学习的概念。

2.4.1 迁移学习介绍

传统的模式识别算法在解决问题时，通常会用大量数据来训练统计模型，假定测试的数据集与训练集有一样的统计特性。如果处理一组具有不同统计特性的数据，需要重新训练这组数据的模型，从而对其进行分类或预测。在现实世界中，很多任务是相似的，因此很有必要探索不同任务之间的相互关系，迁移学习就在此背景下提出。

迁移学习旨在利用已解决的问题的一些知识，来帮助解决未知问题^[11]。所谓的迁移（Transfer）就是知识的迁移，而学习就在于如何选取已解决问题的知识，并且较好地用在未知问题上。为了方便讨论，称已解决的问题的集合为源域（Source Domain），称需要解决的问题的集合为目标域（Target Domain）。

在迁移学习中，有三个主要的问题，即迁移内容、迁移方式与应用范围。迁移中可以仅仅是将源域的数据直接或者重新赋予权重，直接用于目标域中^[51]；也可以是把源特征向量变换为目标特征向量，比如文本识别^[52]；还可以是找出不同任务的模型参数之间的相关性，比如高斯过程共用高层参数^[53]。

最近，迁移学习技术被用在了很多研究领域，比如NLP（Natural Language Processing，自然语言处理）问题^[51]、文本识别^[52]、图像分类^[54]和室内WiFi定

位^[55] 等等。

2.4.2 线性迁移学习方法及其在古籍汉字识别中的应用

迁移学习在手写汉字识别中的应用非常重要，因为手写汉字往往风格不同，不同数据库的样本特征向量差异较大。但是在现实世界中，人在识字的时候就可以在已知一种字体的情况下，迅速认识另一种字体的字。如果能将迁移学习用于这类问题，很多已有的知识就能帮助新的识别问题。

一种线性迁移学习方法，即风格迁移映射（Style Transfer Mapping, STM）方法首先被用于联机手写汉字识别：通过线性映射，不同书写风格的特征向量被映射到一个无风格差异的空间，然后再用一个与风格无关的分类器进行识别；STM假定不同风格的特征向量可以由线性映射来转换，线性变换的系数可以通过加入最小平方误差函数来求解；STM还可用于半监督学习，即用来训练线性系数的数据可以是部分带标签的，这就需要通过迭代和置信度来充分利用字体风格信息^[56]。

把不同书写者换做是不同字体，比如手写体古籍汉字和繁体印刷体汉字，则可以通过STM构建古籍汉字的分类器，因为古籍汉字具有极少的训练样本，难以训练有效的分类器，而繁体印刷体汉字则样本充足，通过STM方法，把古籍汉字的特征向量变换为印刷体风格的特征向量，再用印刷体汉字的分类器分类，即可达到较好的识别效果^[57]。

2.5 高斯过程

高斯过程是多元联合高斯分布在无穷维上的扩展，即任何n个变量服从n元联合高斯分布^[58-60]。用高斯过程来预测数据的方法已经提出了很长时间，可以追溯到20世纪40年代，时间稍近一点的是克里金插值（Kriging），在地质统计学中，克里金插值是一种由先验协方差建模的插值方法，而不是一个分段多项式样条函数进行拟合的拟合函数，在适当的假设下给出了先验，克里金的均值是最佳线性无偏预测，该方法被广泛使用在空间领域分析与计算机实验^①。克里金插值其实就是高斯过程回归方法，只是在推导的时候稍有不同。在地质统计学中，高斯过程一般用于处理低维问题，并且不太关注其概率方面的解释^[61]。

最开始在统计学中用高斯过程是在1978年，O'Hagan用高斯过程定义函数的初始分布，得到一维曲线拟合方法^[62]。在机器学习领域，在监督学习中应用高斯过程还是最近的事，它随着神经网络的反向传播算法而为人们所知^[63]，随后，

^① 来自Wiki百科<http://en.wikipedia.org/wiki/Kriging>

Neal（曾提出贝叶斯插值）发现当贝叶斯神经网络的节点数限制在无穷个时，这个神经网络结构会变成高斯过程^[64]。这个结果把高斯过程带到机器学习领域中，为人们广泛使用。

高斯过程回归在机器学习中有很多应用和研究领域，比如减小训练和预测的计算量、混合高斯过程^[65]、非平稳协方差函数^[66]等。通过理论分析，线性回归模型是高斯过程模型的特例^[58]。此外，高斯过程的参数可以通过多任务学习来得到^[53,67]，高斯过程的核函数也可以通过学习得到^[68]。

2.6 本章小结

本章从文献调研的角度介绍了古籍数字化及其项目、汉字识别常用方法、迁移学习及其应用以及高斯过程的相关研究，为随后章节的展开奠定了基础。

第3章 GP-STM模型介绍

3.1 本章引论

高斯过程风格迁移映射（Gaussian Process Style Transfer Mapping, GP-STM）是将高斯过程应用在STM中的一种非线性迁移学习方法。本章将会从模型的假设、模型介绍以及其参数的优化算法进行讨论。

在本文的研究中，可以将印刷体繁体汉字字库的特征向量集合作为源域，古籍字符的特征向量集合作为目标域。这样引入印刷体字库的好处是，两个域类别相同，字形相似，便于迁移，同时印刷体字库拥有足量的数据来训练MQDF分类器。

为了便于表示，将字符*i*在源域的特征向量记作 \mathbf{s}_i ，在目标域的特征向量记作 \mathbf{t}_i ，则源数据集（Source Dataset）为

$$\mathcal{S} = \{\mathbf{s}_i \in \mathbb{R}^d \mid i = 1, \dots, n\}, \quad (3-1)$$

其中*n*是源数据集大小，*d*是特征向量维数。同时，对应的目标数据集（Target Dataset）为

$$\mathcal{T} = \{\mathbf{t}_i \in \mathbb{R}^d \mid i = 1, \dots, n\}. \quad (3-2)$$

由于古籍样本难以训练出一个适当的分类器，需要借用印刷体汉字样本进行辅助。图C.1是古籍汉字识别中容易想到的两种迁移学习的模式。图3.1(a)表示迁移映射特征向量，即通过某种变换方式，目标数据集中的特征向量被映射到源数据集中，然后通过源数据集的分类器识别。图3.1(b)表示迁移映射分类器，即通过分类器的参数变换，源数据集训练的分类器将能够用于目标数据集的分类。在本文的研究中，选取第一种（即迁移映射特征向量）方法。

3.2 模型基本假设

在研究工作中，通常需要将所研究问题的关键要素提炼出来，并忽略一些次要因素，由此提出相应的假设条件，构建模型对问题进行分析求解。在本文的研究中，根据古籍汉字的特点，结合STM方法和高斯过程模型，同时考虑到计算量和实现难度，提出三个假设：即回归假设、高斯过程假设和同协方差矩阵假设。下面就这三个假设进行论述。

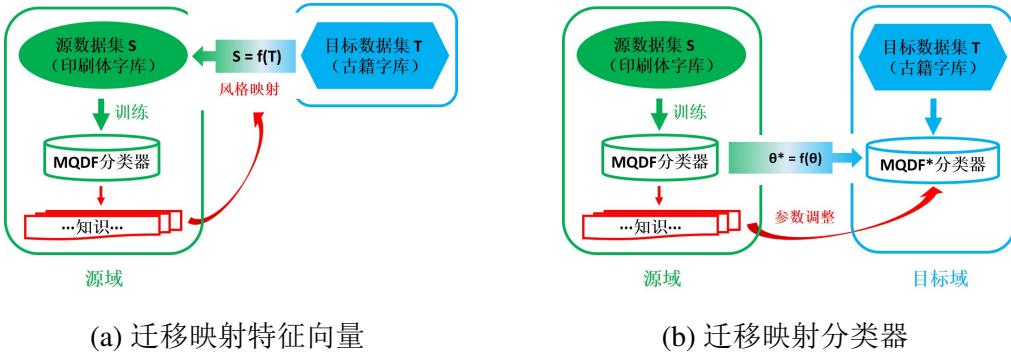


图 3.1 古籍汉字识别的两种迁移学习模式

3.2.1 回归问题

本文采用的方法是图C.1(a)，因此需要搭建一个特征向量空间的映射关系，已有的方法是STM方法。

STM方法首先被用于联机手写汉字识别，通过线性映射，不同书写风格的特征向量被映射到一个无风格差异的空间，然后再用一个与风格无关的分类器进行识别^[56]。STM假定不同风格的特征向量可以由线性映射来转换。对于任何 $\mathbf{t}_i \in \mathcal{T} (i = 1, \dots, n)$ ，希望有一个映射，可以置信度 f_i 将 \mathbf{t}_i 变换为 $\mathbf{s}_i \in \mathcal{S}$ ，也就是

$$\mathbf{s}_i = \mathbf{A}\mathbf{t}_i + \mathbf{b}. \quad (3-3)$$

系数 \mathbf{A} 和 \mathbf{b} 的正则项约束为

$$\min_{\mathbf{A} \in \mathbb{R}^{d \times d}, \mathbf{b} \in \mathbb{R}^d} \sum_{i=1}^n f_i \|\mathbf{A}\mathbf{t}_i + \mathbf{b} - \mathbf{s}_i\|^2 + \beta \|\mathbf{A} - \mathbf{I}\|_F^2 + \gamma \|\mathbf{b}\|^2, \quad (3-4)$$

这里 β 和 γ 是高层参数， $\|\cdot\|_F$ 是矩阵的Frobenius范数。这个针对 \mathbf{A} 和 \mathbf{b} 的优化问题有闭式解：

$$\begin{aligned} \mathbf{A}^* &= \frac{\sum_{i=1}^n f_i \mathbf{s}_i \mathbf{t}_i^T - \frac{1}{f} \hat{\mathbf{s}} \hat{\mathbf{t}}^T + \beta \mathbf{I}}{\sum_{i=1}^n f_i \mathbf{t}_i \mathbf{t}_i^T - \frac{1}{f} \hat{\mathbf{t}} \hat{\mathbf{t}}^T + \beta \mathbf{I}}, \\ \mathbf{b}^* &= \frac{1}{f} (\hat{\mathbf{s}} - \mathbf{A}^* \hat{\mathbf{t}}), \end{aligned} \quad (3-5)$$

这里 \mathbf{I} 是单位矩阵，式中的其他项为如下计算式：

$$\hat{f} = \sum_{i=1}^n f_i + \gamma, \quad \hat{\mathbf{s}} = \sum_{i=1}^n f_i \mathbf{s}_i, \quad \hat{\mathbf{t}} = \sum_{i=1}^n f_i \mathbf{t}_i.$$

在STM中，目标数据集 \mathcal{T} 中的特征向量 $\mathbf{t}_i \in \mathbb{R}^d$ 要映射到源数据集 \mathcal{S} 中的 $\mathbf{s}_i \in \mathbb{R}^d$ ，是一个多对多的映射，因此可以看作是一个多维输出的回归问题。

3.2.2 高斯过程

STM实质是一种向量空间的线性回归模型，本文中的研究尝试从非线性回归的方向去拓展，而高斯过程恰好是一种典型的非线性回归模型，它可以由一般的非线性回归推导出来。为了简单起见，先假定回归的输出是一维的，即 $s_i \in \mathbb{R}^1$ ，则线性回归的表达式是：

$$s_i = \mathbf{w}^T \mathbf{t}_i = \sum_{j=1}^d w_j t_{ij}. \quad (3-6)$$

这里 \mathbf{w} 是线性系数。

为了创建非线性映射，最简单的方法是把输入 \mathbf{t}_i 转为 $\phi(\mathbf{t}_i)$ ，这里 $\phi = (\phi_1, \dots, \phi_M)^T$ 是一组非线性基函数， $\phi_j(j = 1, \dots, M)$ 能把 \mathbb{R}^d 的向量映射到 \mathbb{R}^1 的数量。这样非线性映射就变成了：

$$s_i = \mathbf{w}^T \phi(\mathbf{t}_i) = \sum_{j=1}^M w_j \phi_j(\mathbf{t}_i). \quad (3-7)$$

和STM的线性回归系数约束的式（3-4）类似，归一化平方误差可以写为：

$$\min_{\mathbf{w} \in \mathbb{R}^M} \sum_{i=1}^n \|\mathbf{w}^T \phi(\mathbf{t}_i) - s_i\|^2 + \lambda \|\mathbf{w}\|^2. \quad (3-8)$$

这个误差函数对系数 \mathbf{w} 的偏导数是线性的，所以最优的 \mathbf{w} 有一个闭式解：

$$\mathbf{w}^* = \Phi^T (\Phi \Phi^T + \lambda \mathbf{I})^{-1} \mathbf{s}, \quad (3-9)$$

其中 $\mathbf{s} = (s_1, \dots, s_n)^T$ ，设计矩阵 $\Phi = (\phi(\mathbf{t}_1), \dots, \phi(\mathbf{t}_n))^T$ ，大小是 $n \times M$ 。

定义核 $\mathbf{K} = \Phi \Phi^T + \lambda \mathbf{I}$ ，是一个 $n \times n$ 的对称矩阵，每个元素是

$$K_{ij} = k(\mathbf{t}_i, \mathbf{t}_j) = \phi(\mathbf{t}_i)^T \phi(\mathbf{t}_j) + \lambda \delta(\mathbf{t}_i, \mathbf{t}_j), \quad (3-10)$$

其中 $\delta(\mathbf{t}_i, \mathbf{t}_j)$ 是Kronecker函数。

式（3-9）可以用来改写式（3-7）：

$$s_* = \mathbf{k}(\mathbf{t}_*)^T \mathbf{K}^{-1} \mathbf{s}, \quad (3-11)$$

其中 $\mathbf{k}(\mathbf{t}_i) = \Phi \cdot \phi(\mathbf{t}_i) = (k(\mathbf{t}_1, \mathbf{t}_i), \dots, k(\mathbf{t}_n, \mathbf{t}_i))^T$ 。在非线性回归的各种方法中，高斯过程在研究中更加方便，有丰富的研究成果。因此在众多的非线性映射中本研究选择了高斯过程，也假定字符特征向量之间可以由高斯过程回归映射。

3.2.3 相同协方差矩阵

高斯过程是一种非线性回归方法，它在3.2.2节的基础上，增加了协方差矩阵的一个约束。首先考虑 $s_i \in \mathbb{R}^1$ ，高斯过程假定 $\mathbf{s} = (s_1, \dots, s_n)^T$ 服从 n 元联合高斯分布，且其均值向量为 $\boldsymbol{\mu} \in \mathbb{R}^n$ ，即：

$$\mathbf{s} \sim \mathcal{N}(\boldsymbol{\mu}, \mathbf{K}). \quad (3-12)$$

其中，协方差矩阵 \mathbf{K} 由核函数定义，最常用的是单高斯核函数，比如式(3-13)所示：

$$k(\mathbf{t}_i, \mathbf{t}_j) = \theta_0 \exp\left[\frac{-\|\mathbf{t}_i - \mathbf{t}_j\|^2}{2\theta_1^2}\right] + \theta_2 \delta(\mathbf{t}_i, \mathbf{t}_j), \quad (3-13)$$

这里 $\theta_i (i = 1, 2, 3)$ 是非负参数， $\delta(\mathbf{t}_i, \mathbf{t}_j)$ 是Kronecker δ 函数。协方差矩阵由所有的 $k(\mathbf{t}_i, \mathbf{t}_j)$, $i = 1, \dots, n, j = 1, \dots, n$ 组成，如式(3-14)所示：

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{t}_1, \mathbf{t}_1) & \cdots & k(\mathbf{t}_1, \mathbf{t}_n) \\ \vdots & \ddots & \vdots \\ k(\mathbf{t}_n, \mathbf{t}_1) & \cdots & k(\mathbf{t}_n, \mathbf{t}_n) \end{bmatrix}. \quad (3-14)$$

如果有一个新的 $\mathbf{t}_* \in \mathcal{T}$ 并且需要预测对应的 $s_* \in \mathcal{S}$ ，高斯过程假定：

$$\begin{bmatrix} \mathbf{s} \\ s_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \mu_* \end{bmatrix}, \begin{bmatrix} \mathbf{K} & \mathbf{K}_*^T \\ \mathbf{K}_* & \mathbf{K}_{**} \end{bmatrix}\right), \quad (3-15)$$

其中

$$\mathbf{K}_* = [k(\mathbf{t}_*, \mathbf{t}_1), \dots, k(\mathbf{t}_*, \mathbf{t}_n)], \quad \mathbf{K}_{**} = k(\mathbf{t}_*, \mathbf{t}_*). \quad (3-16)$$

显然， s_* 最佳估计值需要让条件概率 $p(s_* | \mathbf{s})$ 最大。根据联合高斯分布的特性，这个条件概率仍然服从联合高斯分布：

$$s_* | \mathbf{s} \sim \mathcal{N}\left(\mu_* + \mathbf{K}_* \mathbf{K}^{-1}(\mathbf{s} - \boldsymbol{\mu}), \mathbf{K}_{**} - \mathbf{K}_* \mathbf{K}^{-1} \mathbf{K}_*^T\right). \quad (3-17)$$

因此 s_* 的最佳估计是这个分布的均值：

$$s_* = \mu_* + \mathbf{K}_* \mathbf{K}^{-1}(\mathbf{s} - \boldsymbol{\mu}), \quad (3-18)$$

这样就得到了这个一元回归问题的解。此外还可以得出这样估计的置信度，即式(3-17)中的方差：

$$var(s_*) = \mathbf{K}_{**} - \mathbf{K}_* \mathbf{K}^{-1} \mathbf{K}_*^T. \quad (3-19)$$

根据式(3-19), 一维的输出有 $s_i = \mathbf{k}(\mathbf{t}_i)^T \mathbf{K}^{-1} \mathbf{s}$ 。事实上, 本研究是 d 维到 d 维的映射, 因此输出 d 维。简单起见, 假定输出的每一维是独立的 (其实特征向量在提取的时候就希望每一维最好不相关), 由此可用不同的核得 d 个方程来组成 d 个输出:

$$\begin{cases} s_{*1} = \mathbf{k}_1(\mathbf{t}_*)^T \mathbf{K}_1^{-1} \mathbf{s}_1, \\ s_{*2} = \mathbf{k}_2(\mathbf{t}_*)^T \mathbf{K}_2^{-1} \mathbf{s}_2, \\ \dots, \\ s_{*d} = \mathbf{k}_d(\mathbf{t}_*)^T \mathbf{K}_d^{-1} \mathbf{s}_d. \end{cases} \quad (3-20)$$

然而, 如果每一维都有一个独立的核 (协方差矩阵), 那么模型会非常复杂, 计算量也会非常大^[53]。于是, 本研究假定 $\mathbf{K}_1 = \mathbf{K}_2 = \dots = \mathbf{K}_d = \mathbf{K}$, 也就是相同协方差矩阵假设。

3.3 GP-STM模型

在3.2节的基础上, 可以很自然地导出本文的研究重点: GP-STM。下面将具体介绍。

3.3.1 模型概述

GP-STM模型可被认为是一种“转导式迁移学习”, 即给定源数据集 \mathcal{S} 、相应的源分类器 C_S 、目标数据集 \mathcal{T} 、相应的目标分类器 C_T , 转导式迁移学习旨在用 \mathcal{S} 、 \mathcal{T} 的一小部分和 C_S 的知识来提高 C_T 分类器的性能^[11]。

在GP-STM模型中, 源数据集和目标数据集中的特征向量在风格上存有差异。通过GP-STM模块, 可以把目标数据集中的特征向量映射到源数据集中。这样, 就可以用源分类器来有效地识别目标数据集中的特征向量。

为了在识别系统中添加GP-STM模块, 本文将目标数据集随机分为两块: 大约5%的特征向量称为“STM训练集”, 用来训练GP-STM模块的参数; 剩余95%的特征向量称为“STM测试集”。在随机划分“STM训练集”和“STM测试集”时, 可以随机挑选特征向量, 也可以随机挑选类别。模型结构图如图3.2所示。

这个模型通过以下几步来搭建: (1) 用源数据集中的特征向量来训练分类器, 称为“源分类器”; (2) 用“STM训练集”和一部分源数据集中的数据来估计GP-STM模型的参数; (3) “目标分类器”由GP-STM模型和源分类器构成, 来识别目标数据集。识别过程是: 给定一个“STM测试集”中的特征向量, 首先由GP-STM模型映射到源数据集, 然后用源分类器识别。

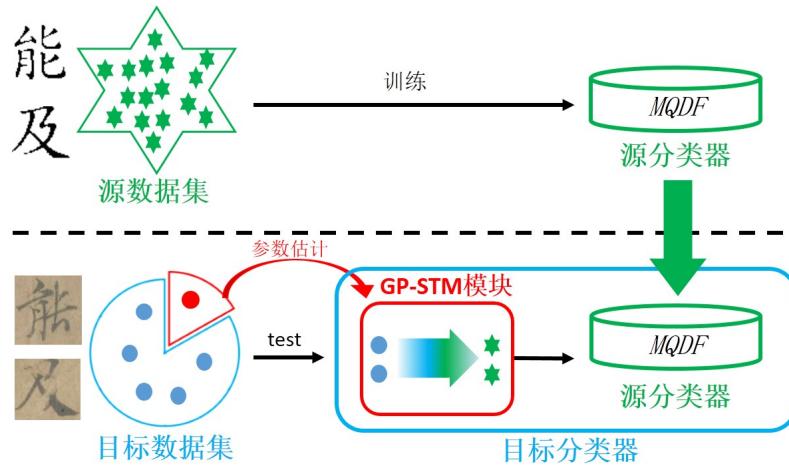


图 3.2 GP-STM识别模型，用来识别目标数据集中的字符

3.3.2 变换公式

在3.2.3节中假定 $\mathbf{s}_i \in \mathbb{R}^1$ ，现在将其由1维扩展到d维，即 $\mathbf{s}_i (i = 1, \dots, n)$ 可以写成 (s_{i1}, \dots, s_{id}) 。记 $\boldsymbol{\mu}_* = (\mu_1, \dots, \mu_d)$ 为特征向量每一维的均值向量，源数据矩阵 $\mathbf{S} = (\mathbf{s}_1, \mathbf{s}_2, \dots, \mathbf{s}_n)^T$ ，再记均值矩阵 $\boldsymbol{\Omega}$ 是 $(\bar{s}, \dots, \bar{s})^T$ ，那么根据3.2.3节中的同协方差矩阵假设条件，式 (3-19) 变为

$$\mathbf{s}_*^T = \boldsymbol{\mu}_*^T + \mathbf{K}_* \mathbf{K}^{-1} (\mathbf{S} - \boldsymbol{\Omega}). \quad (3-21)$$

根据式 (3-21)，可以把均值向量 $\boldsymbol{\mu}_*$ 当作是一个粗估计， $\mathbf{K}_* \mathbf{K}^{-1} (\mathbf{S} - \boldsymbol{\Omega})$ 项当作是修正项。在GP-STM模型中，源数据集中的向量 \mathbf{s}_* 需要和目标数据集中的向量 \mathbf{t}_* 相近，风格需要做一些改变。因此，这里设 $\boldsymbol{\mu}_*$ 为 $\bar{\mathbf{s}} + (\mathbf{t}_* - \bar{\mathbf{t}})$ 。GP-STM模型对源数据集中的向量 \mathbf{s}_* 的估计是：

$$\mathbf{s}_* = \bar{\mathbf{s}} + (\mathbf{t}_* - \bar{\mathbf{t}}) + \mathbf{K}_* \mathbf{K}^{-1} (\mathbf{S} - \bar{\mathbf{S}}). \quad (3-22)$$

$\mathbf{K}^{-1} (\mathbf{S} - \bar{\mathbf{S}})$ 可以提前计算，记为矩阵 \mathbf{A} 。 \mathbf{K}_* 是 \mathbf{t}_* 的非线性函数，可以写为 $f(\mathbf{t}_*)$ 。如果将 $\bar{\mathbf{s}} + (\mathbf{t}_* - \bar{\mathbf{t}})$ 写为 $\mathbf{b}(\mathbf{t}_*)$ 那么式 (3-22) 将会是

$$\mathbf{s}_* = \mathbf{A} f(\mathbf{t}_*) + \mathbf{b}(\mathbf{t}_*). \quad (3-23)$$

GP-STM模型和STM模型的式 (3-3) 很相似，不同的地方在于加了一个非线性的 f 函数。GP-STM模型训练和测试的算法如算法1所示。

算法1 GP-STM模型的训练和测试算法

-
- | | |
|-----|--|
| 输入: | 1. STM训练集特征向量对 $(\mathbf{s}_i, \mathbf{t}_i)_{i=1}^n$
2. 核函数 $k(\mathbf{t}_i, \mathbf{t}_j)$
3. 源分类器 $G(s, k)$
4. 一个新的目标数据集中的特征向量 \mathbf{t}_* |
| 训练: | 1. 计算每个 $\mathbf{s}_i, i = 1, \dots, n$ 的均值向量 $\bar{\mathbf{s}}$
2. 计算每个 $\mathbf{t}_i, i = 1, \dots, n$ 的均值向量 $\bar{\mathbf{t}}$
3. 利用 $(\mathbf{s}_i, \mathbf{t}_i)_{i=1}^n$ 通过式 (3-26) 学习最优的参数 θ |
| 测试: | \mathbf{t}_* 所属的类别是 $\arg \min_{k=1}^N G(\mathbf{A}f(\mathbf{t}_*) + \mathbf{b}(\mathbf{t}_*), k)$ |
-

3.4 GP-STM参数的优化方法

在高斯过程中，构成协方差矩阵的核函数有三个参数 $\theta = (\theta_0, \theta_1, \theta_2)$ ，需要预先设定。这样就会涉及到参数的优化问题，即选取合适的 θ 使得回归得更加准确。根据最大后验概率法则，最优的 θ 使得 $p(\theta|\mathbf{S}, \mathbf{T})$ 最大。如果对 θ 没有任何先验知识的话，可以用最大似然法则，即最大化 $\ln p(\mathbf{S}|\theta, \mathbf{T})$ ，或写为 $\ln p(\mathbf{S}|\mu, \mathbf{K})$ ：

$$\theta_{opt} = \arg \max_{\theta} \ln p(\mathbf{S}|\mu, \mathbf{K}). \quad (3-24)$$

3.4.1 最大似然概率

给定一个数据集 $\mathbf{S} = (s_1, \dots, s_d)^T$ ，假定它们从多元高斯分布中独立地抽样出来，可以用最大似然概率来估计这个高斯分布^[58]：

$$\begin{aligned} \ln p(\mathbf{S}|\mu, \mathbf{K}) &= \sum_{i=1}^d \ln p(\mathbf{r}_i|\mu, \mathbf{K}) \\ &= -\frac{1}{2} \sum_{i=1}^d (\mathbf{r}_i - \mu)^T \mathbf{K}^{-1} (\mathbf{r}_i - \mu) - \frac{d}{2} \log |\mathbf{K}| - \frac{nd}{2} \log (2\pi), \end{aligned} \quad (3-25)$$

其中 \mathbf{r}_i 是 \mathbf{S} 的列向量， n 是 \mathbf{r}_i 的长度， μ 是 \mathbf{S} 的列向量的均值。

为了获取最优参数，需要找到对数似然函数对参数 θ 的梯度：

$$\begin{aligned} \frac{\partial}{\partial \theta_k} \ln p(\mathbf{S}|\mu, \mathbf{K}) &= \frac{\partial}{\partial \theta_k} \sum_{i=1}^d \ln p(\mathbf{r}_i|\mu, \mathbf{K}) \\ &= \frac{1}{2} \sum_{i=1}^d (\mathbf{r}_i - \mu)^T \mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_k} \mathbf{K}^{-1} (\mathbf{r}_i - \mu) - \frac{d}{2} \text{tr} \left(\mathbf{K}^{-1} \frac{\partial \mathbf{K}}{\partial \theta_k} \right) \\ &= \frac{1}{2} \text{tr} \left(\left(\sum_{i=1}^d \alpha_i \alpha_i^T - d \mathbf{K}^{-1} \right) \frac{\partial \mathbf{K}}{\partial \theta_k} \right), \\ &= \frac{1}{2} \text{tr} \left((\mathbf{A} \mathbf{A}^T - d \mathbf{K}^{-1}) \frac{\partial \mathbf{K}}{\partial \theta_k} \right), \end{aligned} \quad (3-26)$$

其中 $\alpha_i = \mathbf{K}^{-1}(\mathbf{r}_i - \boldsymbol{\mu})$, $\mathbf{A} = (\alpha_1, \dots, \alpha_d) = \mathbf{K}^{-1}(\mathbf{S} - \boldsymbol{\Omega})$, $\text{tr}(\cdot)$ 表示求矩阵秩的函数。

3.4.2 核函数的偏导数

对于任何 $\theta_k \in \boldsymbol{\theta}$, 核函数对参数的偏导数 $\partial \mathbf{K} / \partial \theta_k$ 是

$$\frac{\partial \mathbf{K}}{\partial \theta_k} = \begin{bmatrix} \frac{\partial k(\mathbf{t}_1, \mathbf{t}_1)}{\partial \theta_k} & \dots & \frac{\partial k(\mathbf{t}_1, \mathbf{t}_n)}{\partial \theta_k} \\ \vdots & \ddots & \vdots \\ \frac{\partial k(\mathbf{t}_n, \mathbf{t}_1)}{\partial \theta_k} & \dots & \frac{\partial k(\mathbf{t}_n, \mathbf{t}_n)}{\partial \theta_k} \end{bmatrix}. \quad (3-27)$$

对于单高斯核式 (3-13), 核函数对每个参数的偏导数是:

$$\begin{aligned} k(\mathbf{t}_i, \mathbf{t}_j) &= \theta_0 \exp\left[\frac{-\|\mathbf{t}_i - \mathbf{t}_j\|^2}{2\theta_1^2}\right] + \theta_2 \delta(\mathbf{t}_i, \mathbf{t}_j), \\ \frac{\partial k(\mathbf{t}_i, \mathbf{t}_j)}{\partial \theta_0} &= \exp\left[\frac{-\|\mathbf{t}_i - \mathbf{t}_j\|^2}{2\theta_1^2}\right], \\ \frac{\partial k(\mathbf{t}_i, \mathbf{t}_j)}{\partial \theta_1} &= \frac{\|\mathbf{t}_i - \mathbf{t}_j\|^2}{\theta_1^3} \theta_0 \exp\left[\frac{-\|\mathbf{t}_i - \mathbf{t}_j\|^2}{2\theta_1^2}\right], \\ \frac{\partial k(\mathbf{t}_i, \mathbf{t}_j)}{\partial \theta_2} &= \delta(\mathbf{t}_i, \mathbf{t}_j). \end{aligned} \quad (3-28)$$

3.5 本章小结

本章提出了古籍汉字识别中的一个新模型: GP-STM模型。首先针对古籍汉字样本的特殊性, 提出了三条假设, 然后通过推导给出了模型的变换公式, 最后讨论了模型的参数一种优化方法。在实际应用过程中, 选取参数可以结合实验选取和用优化算法选取, 保证模型的性能。

第4章 实验结果及分析

4.1 本章引论

本章将会用一些实验来验证GP-STM模型的有效性。首先是介绍实验的样本数据库，其次是介绍实验中的参数配置，最后是各个实验的展现。

4.2 实验样本介绍

实验中采用了两组实验样本数据库来测试GP-STM模型：敦煌古籍汉字数据库，以及中科院自动化所的联机手写汉字 CASIA OLHWDB1.1数据库。下面将分别予以介绍。

4.2.1 敦煌古籍数据库介绍

敦煌古籍是发现于莫高窟的古代书卷，多为晋朝（公元4世纪）到元朝（公元11世纪）的佛经。这些古籍对研究中国古代宗教、政治、经济和文化有着重要的价值。近些年来，敦煌古籍文档被大量数字化，保存为高清图片，亟需识别成为文本。

敦煌古籍文档图像首先被预处理、二值化等操作，然后进行字符切割来提取每个汉字。提取的汉字被缩放到 65×65 大小。本文只考虑字符识别，因此文档图像的预处理和字符切割过程被忽略。实验中，所有输入的字符图像是来自于这些文档图像的二值图像（见图4.1）。这些字符图像总数近60,000，涵盖了5,401类中的3,000类以上。实验中用了9个数据集，每个数据集中的字符数和类别数的统计如表1.1所示。经过提取加权方向编码直方图（WDCH）特征，每个字符图像变成了392维的特征向量（详见附录A）。

根据1.1.2节所提到的难点，用这套数据库难以训练合适的分类器，因此需要用迁移学习的方法来帮助训练分类器。

4.2.2 中科院手写汉字数据库介绍

CASIA OLHWDB1.1是联机手写汉字数据库^[8]，由300个书写者书写。数据库由GB2312-80编码的一级 3,755类汉字组成。为了方便起见，直接采用提取



图 4.1 敦煌古籍汉字“目标数据集”的获取过程

好512维的特征向量作为输入，这些特征向量可以直接从主页下载到^①。特征向量被分为训练集和测试集。每个文件的书写者数和字符数见表4.1

表 4.1 中科院手写汉字数据库CASIA OLHWDB1.1

数据库	类别数	特征维度	书写者人数		样本数	
			训练	测试	训练	测试
OLHWDB1.1	3,755	512	240	60	898,573	224,559

这个数据库识别过程中的主要困难是不同书写者具有的不同的书写风格，所以需要一个和风格无关的分类器来克服这个困难。

4.3 实验设置

在GP-STM模型中，核函数式（3-13）的参数 $\theta = (\theta_0, \theta_1, \theta_2)^T$ 。其中 θ_0 是允许的最大方差， θ_1 控制 s_i 和 s_j 的距离， θ_1 越大则二者越相关， θ_1 越小则二者越不相关。 θ_2 是噪声水平，能够让核 \mathbf{K} 避免成为奇异矩阵。参数 θ 结合优化算法和实验而设定。在本文的模型中，在优化算法的基础上，通过实验方法设定参数 θ 为 $(1, 3, 0.01)^T$ 。

在敦煌古籍数据数据库的实验中，将部分测试集作为STM训练集，其余部分作为STM测试集；在手写体汉字数据库的实验中，将测试集随机分为相等的两半，一半是STM训练集，另一半是STM测试集。需要注意的是，由于手写体汉字数据库的每个测试集都是一个类别对应一个样本，因此可以看作 STM训练集和STM测试集不重叠。为了让实验数据更具有说服力，每个实验重复5次随机抽取50%的STM训练集，得到识别结果后再取平均数。

^① CASIA OLHWDB1.1网址：<http://www.nlpr.ia.ac.cn/databases/handwriting/Home.html>

4.4 与传统方法对比

为了和STM方法做比较，将STM中的参数按照文献[56]设为 $\beta = 0.03$ 、 $\gamma = 0.01$ 。因为本文的研究是监督学习，所以置信度 $f_i(i = 1, \dots, n)$ 置为 $\frac{1}{n}$ 。

敦煌古籍数据库的实验结果如图4.2所示。实验中，有9套测试样本，采用了4种方法进行对比：MQDF、STM、GP-STM和最小欧氏距离。

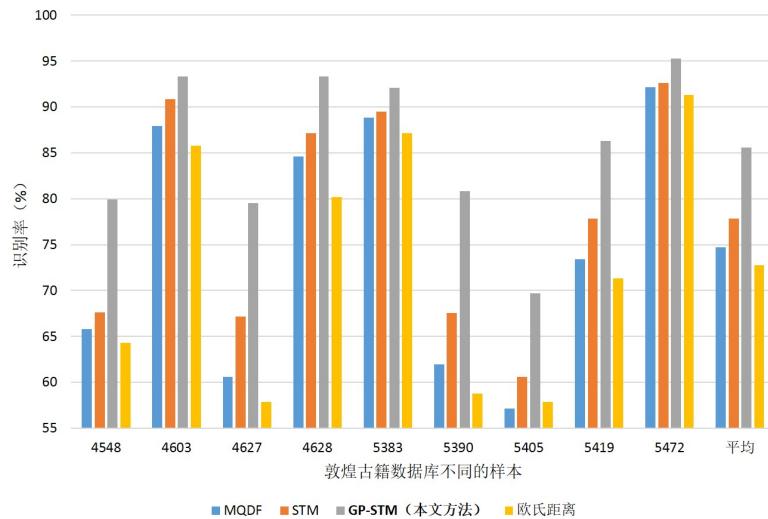


图 4.2 GP-STM 模型在敦煌古籍数据库的实验结果

从图4.2中可以看出本文提出的GP-STM模型可以达到最好的识别率。STM的识别率高于MQDF方法，最小欧氏距离方法的识别率最低。

由于图中各个方法的识别率相近，为了使GP-STM、STM和MQDF方法的区别更加明显，可以使用相对数据，比如相对最小欧氏距离错误率的减少率：

$$\text{错误减少率}_{\text{新方法}} = \frac{\text{错误率}_{Euclid} - \text{错误率}_{\text{新方法}}}{\text{错误率}_{Euclid}} = \frac{\text{识别率}_{\text{新方法}} - \text{识别率}_{Euclid}}{1 - \text{识别率}_{Euclid}} \quad (4-1)$$

这样图4.2可以变为图4.3。更大的错误减少率表明更好的识别性能，从图中可以明显看到GP-STM模型的识别结果优于其他模型。

GP-STM模型在手写汉字数据库CASIA OLHWDB1.1上的实验结果如图4.4所示。在实验中，共测试了60套样本，将4种方法加入对比：MQDF, STM, GP-STM(本文的方法)和最小欧氏距离。

从图4.4可以看出，虽然偶尔有差异，但GP-STM模型仍然能普遍达到最好的识别率，STM、MQDF和最小欧氏距离方法的识别率依次下降。同样采用式(4-1)可得到图4.5。纵轴是相比于最小欧氏距离方法的错误率下降的百分比。数值越大说明识别效果越好。

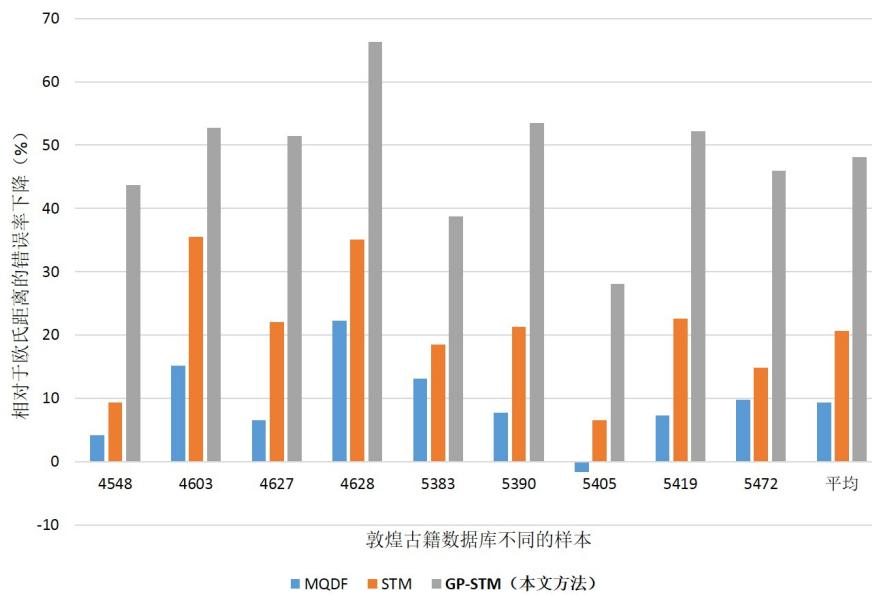


图 4.3 相对于欧氏距离方法的错误率下降的百分比

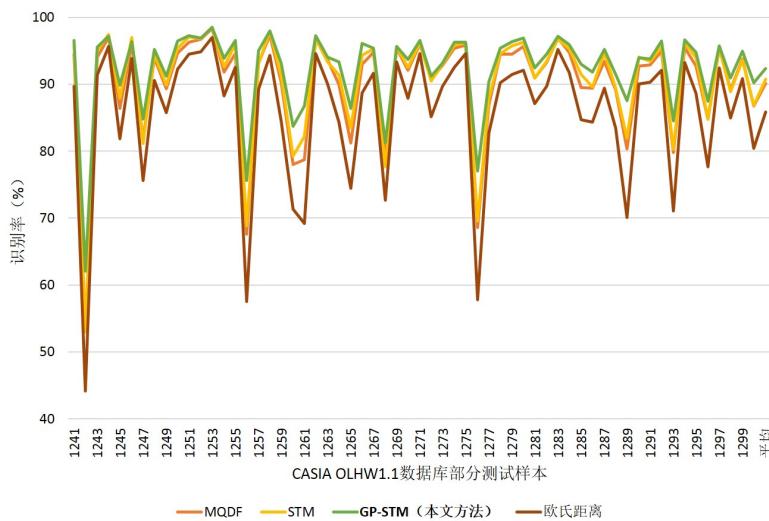


图 4.4 GP-STM模型在手写汉字数据库CASIAOLHWDB1.1上的实验结果

4.5 STM训练集比例与识别率的关系

在4.4节中，STM训练集的大小是50%。为了考察STM训练集对识别率的影响，本节中将以5%开始，每次增加5%，做十组实验，来看识别率的变化趋势。

古籍汉字数据库的识别结果如图4.6(a)所示。手写体汉字数据库取1241-1270共30套样本，其识别结果如图4.6(b)所示。

从图4.6(a)和图4.6(b)中均可看出，本文的GP-STM模型均高于其他方法，并且明显随着STM训练集的比例增加而增加。线性STM方法与GP-STM方法在原理上相似，所以也可以随着STM训练集的比例增加而增加。最小欧氏距离和MQDF方法与STM训练集的比例无关，呈水平状。

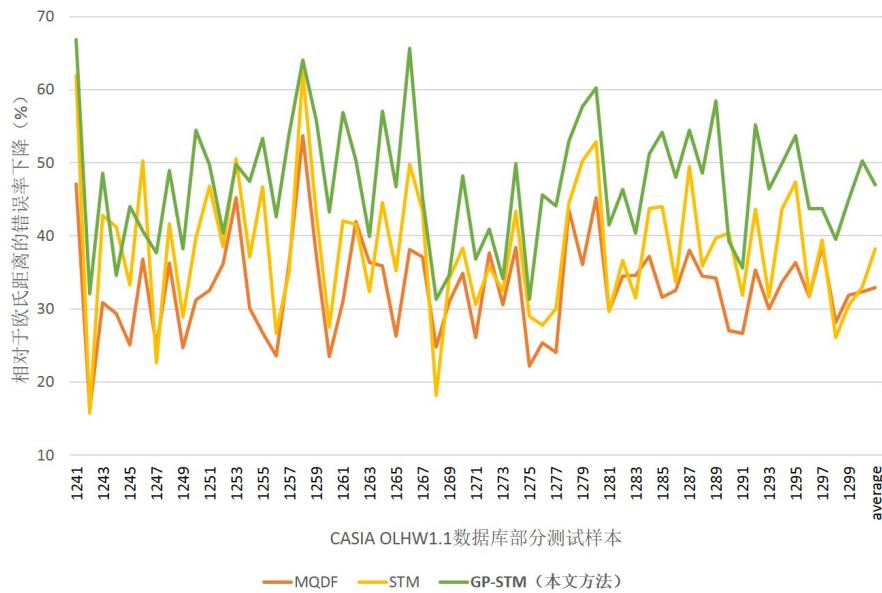


图 4.5 手写汉字数据库CASIA OLHWDB1.1的实验结果

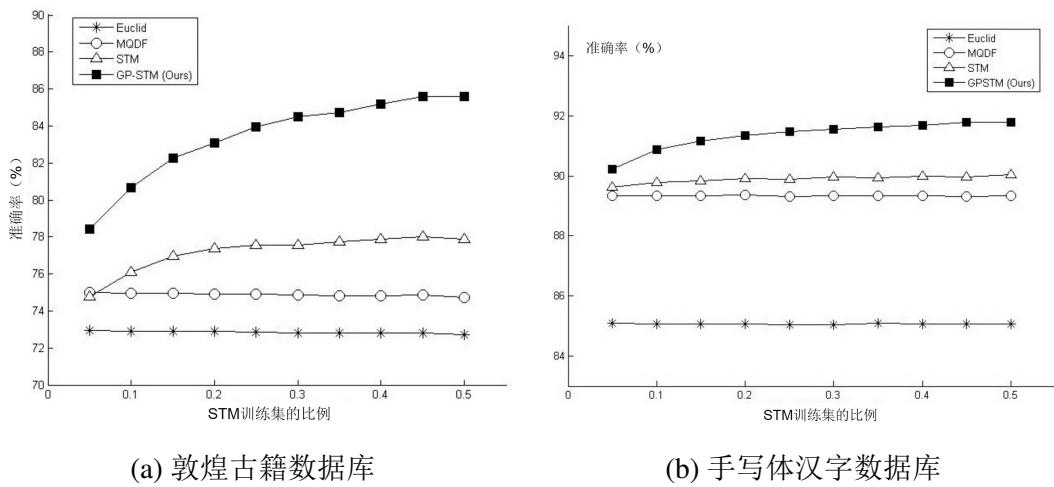


图 4.6 识别率随STM训练集百分比的实验结果

4.6 核函数选择

4.6.1 核函数介绍

除了式 (3-13) 以外的核函数外，还有很多其他的函数可供选择，比如 Ornstein-Uhlenbeck 过程在描述布朗运动时采用的指数核，表达式如下：

$$k(\mathbf{t}_i, \mathbf{t}_j) = \theta_0 \exp(-\theta_1 \|\mathbf{t}_i - \mathbf{t}_j\|) + \theta_2 \delta(\mathbf{t}_i, \mathbf{t}_j), \quad (4-2)$$

指数核的参数与高斯核式 (3-13) 的参数的意义类似。通过在 [0,1] 的范围内实验调试，选得一个合适的 $\theta = [1, 0.0011, 0]$.

如果想加入更远的相关项，可以采用双高斯核：

$$k(\mathbf{t}_i, \mathbf{t}_j) = \theta_0 \exp\left[\frac{-\|\mathbf{t}_i - \mathbf{t}_j\|^2}{2\theta_1^2}\right] + \theta_3 \exp\left[\frac{-\|\mathbf{t}_i - \mathbf{t}_j\|^2}{2\theta_4^2}\right] + \theta_2 \delta(\mathbf{t}_i, \mathbf{t}_j), \quad (4-3)$$

其中 $\theta_3 \approx 6\theta_1$ 。双高斯核的参数 $\boldsymbol{\theta} = (\theta_0, \theta_1, \theta_2, \theta_3, \theta_4)^T$ 中，新加入了两个参数 θ_3 和 θ_4 。第二项中的 θ_4 应该比 θ_1 更大，意味着考虑进去更远的相关性。 θ_1 和 θ_3 控制远、近相关性的权重。本文中令 $\theta_3 = 0.1 \times \theta_1$, $\theta_4 = 6 \times \theta_2$.

实际上，对于核函数仅有的限制是协方差矩阵 \mathbf{K} 对于任何 \mathbf{t}_i 和 \mathbf{t}_j 是半正定的^[58]，因此构造核函数有很大的灵活性。

4.6.2 不同核函数的对比

为了对比不同核函数的效果，在本文的研究中分别用式 (3-13)、式 (4-3) 和式 (4-2) 来计算GP-STM模型的协方差矩阵。为提高速度，训练时每个类别用10组样本（这样会导致识别率下降）。实验的结果如图4.7所示。

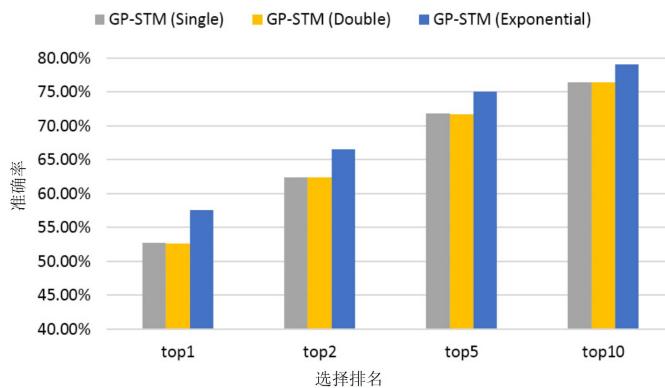


图 4.7 不同核对识别率的影响

在图4.7中，top1、top2、top5和top10分别表示首选、前2选、前5选和前10选的识别结果。可以看出，指数形式的核函数比其他两个核函数表现都好。单高斯核和双高斯核差异不大。

4.7 GP-STM模型理解

为了探索GP-STM模型的运行原理，本文从协方差矩阵拟合、可视化和实验结果分析等角度进行探究。

4.7.1 协方差矩阵拟合

根据3.2.1节，在线性回归中有

$$\mathbf{s} = \mathbf{b} + \sum_{k=1}^d a_k \mathbf{t}_k + \boldsymbol{\epsilon}, \quad (4-4)$$

其中 $\boldsymbol{\epsilon}$ 是一个噪声系数。

假定系数 \mathbf{b} 和 a_k , ($k = 1, \dots, d$)的均值是0、方差是 σ_b^2 和 σ_{ak}^2 ，那么源向量的协方差的估计值为

$$\begin{aligned} Cov_{Linear}(\mathbf{s}_i, \mathbf{s}_j) &= E \left[\left(\mathbf{b} + \sum_{k=1}^d a_k t_{ik} + \boldsymbol{\epsilon}_i \right) \left(\mathbf{b} + \sum_{k=1}^d a_k t_{jk} + \boldsymbol{\epsilon}_j \right) \right] \\ &= \sigma_b^2 + \sum_{k=1}^d \sigma_{ak}^2 t_{ik} t_{jk} + \delta_{ij} \sigma_\epsilon. \end{aligned} \quad (4-5)$$

其中 δ_{ij} 只有在 $i = j$ 的情况下为1，其他均为0。

而在高斯过程中，根据GP-STM模型的假设条件式 (3-13)，可以得到源向量的协方差估计值：

$$Cov_{gp}(\mathbf{s}_i, \mathbf{s}_j) = k(\mathbf{t}_i, \mathbf{t}_j) = \theta_0 \exp \left[\frac{-\|\mathbf{t}_i - \mathbf{t}_j\|^2}{2\theta_1^2} \right] + \theta_2 \delta(\mathbf{t}_i, \mathbf{t}_j). \quad (4-6)$$

作为参照，源向量的协方差应该是：

$$Cov(\mathbf{s}_i, \mathbf{s}_j) = E(\mathbf{s}_i, \mathbf{s}_j) = \frac{1}{d} \sum_{k=1}^d s_{ik} s_{jk}. \quad (4-7)$$

为了从协方差这个角度来对比线性回归和高斯过程，本文在研究中做了如下实验。取STM训练集的 \mathbf{s}_i 、 \mathbf{t}_i ，分别计算 $Cov_{gp}(\mathbf{s}_i, \mathbf{s}_j)$ 和 $Cov_{Linear}(\mathbf{s}_i, \mathbf{s}_j)$ ，比较它们与 $Cov(\mathbf{s}_i, \mathbf{s}_j)$ 的拟合情况。实验中，取某汉字 $i = i_0$ ，分别与其他汉字计算协方差，然后对 $Cov(\mathbf{s}_{i_0}, \mathbf{s}_i)$, $i = 1, \dots, N$ 进行排序，并用方差归一化，选最大的前50个字符。实验结果如图4.8所示。图中横轴代表字符，纵轴代表协方差。因为协方差已经被排序，所以横轴上1点的纵坐标一定是1，代表方差（最大）。从图4.8中可以看出，高斯过程在协方差上的拟合比线性回归更平滑、更为切合。

4.7.2 可视化展示

为了让STM和GP-STM变换过程可视化，我们直接使用字符的像素特征。字符图片首先被归一化到 30×30 ，然后拉伸为 900×1 的向量。根据式 (3-5) 和式 (3-22)，STM和GP-STM估计被计算，然后重新归一化到 30×30 。可视化的结果如图4.9所示。

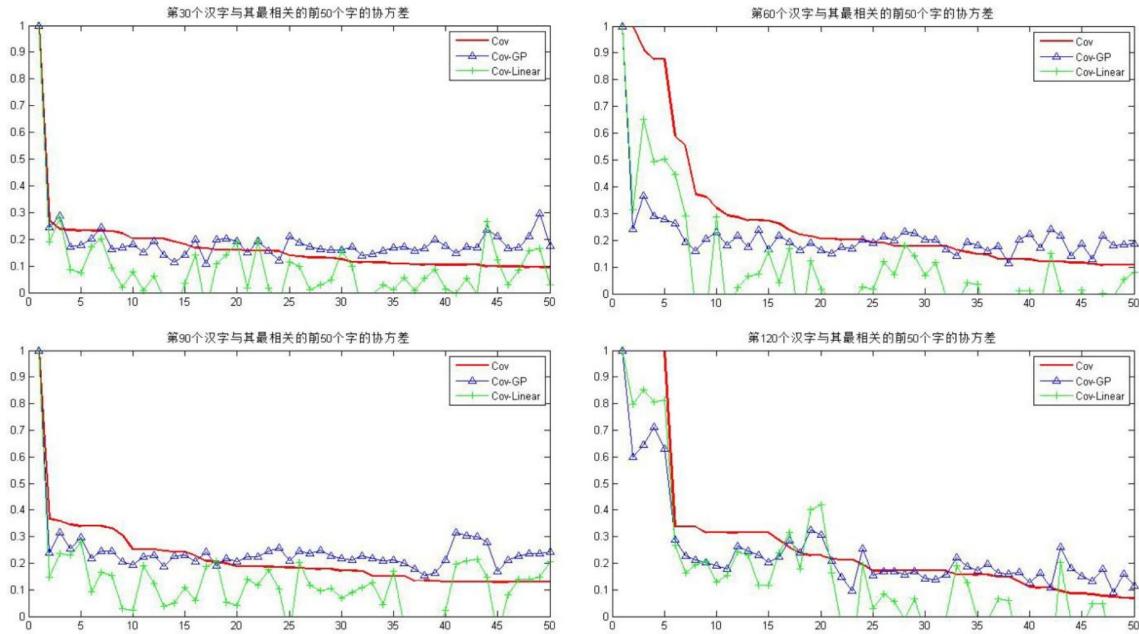


图 4.8 线性回归和高斯过程在协方差拟合上的对比

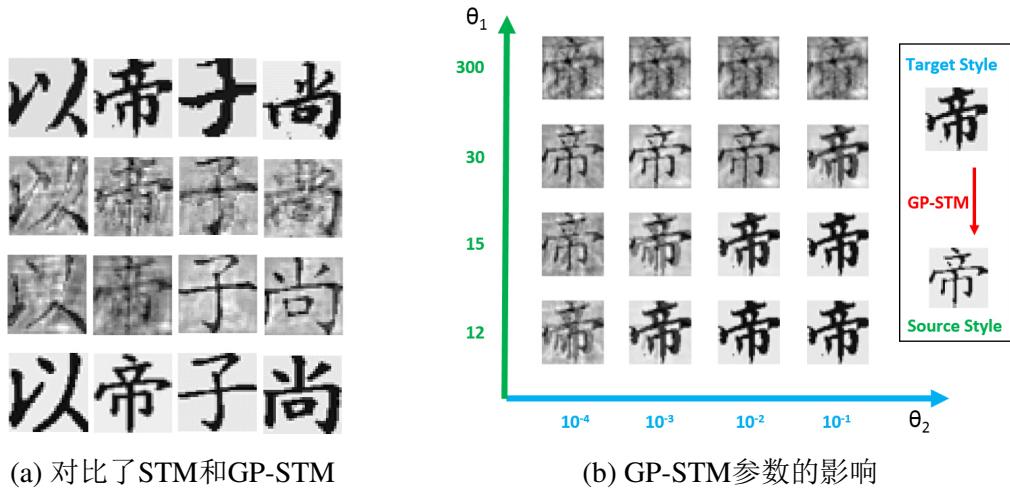


图 4.9 STM和GP-STM的可视化

在图4.9(a)中，第一行来自敦煌古籍。第二行是用STM估计的字符。第三行是用GP-STM估计的字符。第四行是中文印刷体。注意到左边两列来自于STM测试集，右边两列来自于STM训练集。可以看到，在STM训练集中，GP-STM估计的字符更像印刷体。在STM测试集中，GP-STM估计的字符被模糊的更严重，同时在细节上更像印刷体。总之，GP-STM比STM有更强的迁移能力，因此有更高的识别率。

从图4.9(b)中可以看出，正如在4.3节中的讨论，更大的 θ_1 能引起更大的变换，有时会引起模糊。相反，更大的 θ_2 让向量保持原来的样子。合适的参数能让目标风格的向量成功转为源风格，比如 $\theta_1 = 30$ 和 $\theta_2 = 10^{-3}$ 。这一过程正如用相机对焦，

合适的焦距能让景色更清晰。

4.7.3 识别结果分析

在采用了GP-STM模块之后，普通的MQDF分类器获得了更高的识别性能。这一小节将会展现哪些字符的识别得以改善。

取敦煌古籍数据库的4548.pnt数据集作为测试集，从中随机抽选出5%的汉字作为STM训练集，剩下的95%作为STM测试集。

实验统计显示，4548.pnt共有11085个汉字属于BIG5编码的前5401类中，用MQDF分类器直接测试，共有7302个汉字识别正确，准确率65.87%；增加GP-STM模块后，共有7725个汉字识别正确，准确率69.69%；共有922个汉字因为GP-STM模块，由识别错误变为识别正确；同时有499个汉字因为GP-STM模块，由识别正确变为识别错误。表4.2显示了部分由于GP-STM模块的加入而识别正确的字。其中的阴影部分表示该字也包含在STM训练集中。所有字按照BIG5编码顺序排列，也即按照笔画顺序排列。这样排列的好处是相似字的笔画相差不远，更容易看到STM训练集对相似字的影响。

表 4.2 部分因GP-STM模块的增加而新识别正确的字。

乃	九	人	十	又	三	下	丈	上	也	乞	士	大
子	山	川	已	才	不	中	之	云	井	五	今	元
六	公	及	太	少	尤	尺	文	日	曰	月	比	父
牛	王	乎	以	代	令	兄	出	北	卯	四	奴	市
平	幼	旦	正	母	永	田	白	交	亦	光	先	列
匡	名	合	后	地	如	字	宇	守	安	州	托	有
此	而	自	至	色	位	何	兵	即	壯	岑	彤	扶

从表4.2中可以看出，增加GP-STM模块后，有一些新增识别正确的字在字形上与STM训练集相似，比如STM训练集中的“令”和“日”，使得相似的“今”和“曰”识别正确；还有一些新增识别正确的字在字形上与STM训练集中的字并不相似，这可能是因为在提取了特征向量后，相似的字在特征空间差异会比较大，而不相似的字也许会在特征空间有较大相似度。

4.8 本章小结

本章用GP-STM模型做了一系列实验，来验证GP-STM模型的性能和内在原理。从实验可以看出，GP-STM模型相比于线性模型具有更强的映射能力，当然也需要更多的耗时，复杂度也更大。

第5章 基于张量分解的字符识别

5.1 本章引论

在进行迁移学习的研究工作之前，本文也尝试用其他方法研究手写体古籍汉字识别，基于张量分解的字符识别方法是其中之一。

近年来，张量在机器学习和模式识别领域引起了研究者的注意^[26]。Lathauwer等人推广了矩阵奇异值分解（SVD），提出张量高阶奇异值分解（HOSVD）^[69] 及其快速算法高阶正交迭代（HOOI）^[70]，之后Sheehan等人阐明了HOSVD、HOOI、主成分分析（PCA）及矩阵低秩估计（GLRAM）之间的联系^[71]，这些工作为张量在识别算法中的应用打下了理论基础。Savas等人首次把张量分解的方法用于识别手写数字^[72]，但因为其直接取图像像素作为特征，没能很好地提取字符的局部纹理，致使其在MNIST数据库上识别率并不太高；Liu Huchuan等人注意到了这一问题，用HOOI改进张量分解算法，同时提取LBP特征做人脸识别，不足之处是求解原始的最小二乘问题得到识别结果，这样计算量大而且耗时；周丙寅使用张量分解识别动态纹理^[73]，在识别率和计算量上都有较大改进。

基于上述文献中提到各种方法，本章将张量分解应用到手写数字和古籍汉字的识别中，先提取字符局部纹理特征，然后用HOOI来加速张量分解，最后使用一种改进的方法求解最小二乘问题，使得计算量更小，以适应手写体古籍汉字集类别多的问题。实验结果显示其具有良好的识别效果。

5.2 张量简介

5.2.1 张量运算

数学中的张量可以由数量表示，比如一维的向量、二维的矩阵。本文中采用三阶张量^[72]（以后文中提到的“张量”如不加说明均指三阶张量）

$$A \in \mathbb{R}^{I \times J \times K}. \quad (5-1)$$

可以在张量上定义内积与范数：

$$\langle A, B \rangle = \sum_i^I \sum_j^J \sum_k^K a_{ijk} b_{ijk}, \quad (5-2)$$

$$\|A\| = \sqrt{\langle A, A \rangle}. \quad (5-3)$$

张量独特的运算有n-模式矩阵展开

$$\begin{cases} A_{(1)} : a_{ijk} = a_{iv}^{(1)}, v = j + (k-1)K, \\ A_{(2)} : a_{ijk} = a_{jv}^{(1)}, v = k + (i-1)I, \\ A_{(3)} : a_{ijk} = a_{kv}^{(1)}, v = i + (j-1)J. \end{cases} \quad (5-4)$$

另一个常用的运算是n-模式张量矩阵乘积。设张量 $A \in \mathbb{R}^{I_1 \times I_2 \times \dots \times I_N}$, 它与矩阵 $F \in \mathbb{R}^{J_n \times I_n}$ 的乘积:

$$(A \times_n F)(i_1, \dots, i_{n-1}, j_n, i_{n+1}, \dots, i_N) = \sum_{i_n=1}^{I_n} A(i_1, \dots, i_n)F(j_n, i_n). \quad (5-5)$$

5.2.2 张量的CP分解

张量有两种主要的分解方法, 均来源于矩阵的奇异值分解(SVD)的拓展。一种是CP分解(Candecomp/Parafac decomposition), 保证分解后核的对角性, 但是其他因子不具有正交性。CP分解的表达式和示意图如下^[73]

$$A \approx \sum_{r=1}^R \lambda_r x_r^1 \circ x_r^2 \circ x_r^3. \quad (5-6)$$

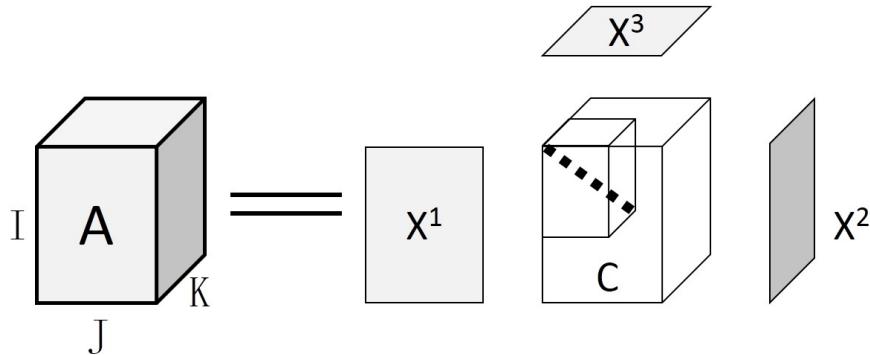


图 5.1 3阶张量的CP分解示意图

5.2.3 张量的Tucker分解

另一种是Tucker分解, 它保留了因子的正交性, 放弃了核的对角性。虽然核没有了对角性, 但其正交性为识别提供了可靠的基础。Tucker分解的表达式和示意图如下^[73]

$$A \approx C \times_1 X^{(1)} \times_2 X^{(2)} \times_3 X^{(3)}. \quad (5-7)$$

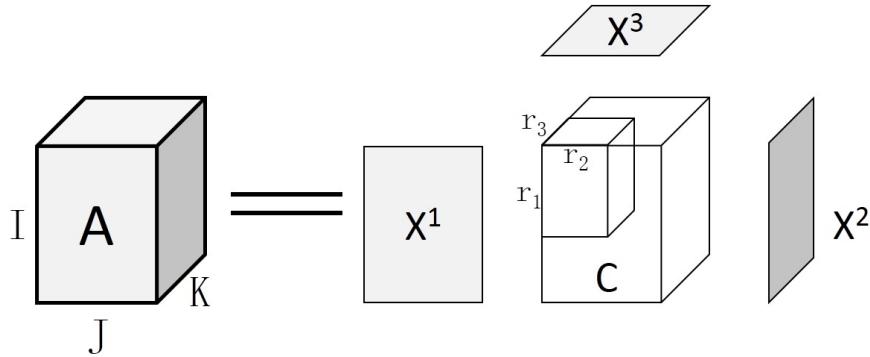


图 5.2 3阶张量的Tucker分解示意图

Tucker分解有一种低秩快速分解算法：高阶正交迭代（HOOI，High Order Orthogonal Iteration）算法^[70]。HOOI算法用迭代的方式，依次求出 $X^{(1)}$ ， $X^{(2)}$ 和 $X^{(3)}$ ，过程如下算法2所示^[71]。

算法2	张量的HOOI分解算法
输入：	张量 $A \in \mathbb{R}^{I \times J \times K}$, 截断数 r_1, r_2, r_3 , 临时张量B
输出：	张量 $C \in \mathbb{R}^{r_1 \times r_2 \times r_3}$, 矩阵 $X^{(1)}$, $X^{(2)}$ 和 $X^{(3)}$
初始化：	选取初始的 $X^{(2)} \in \mathbb{R}^{I_2 \times r_2}$, $X^{(3)} \in \mathbb{R}^{I_3 \times r_3}$
迭代至收敛：	$B = A \times_2 (X^{(2)})^T \times_3 (X^{(3)})^T; \quad \textcircled{1}$ $X^{(1)} = SVD(B_{(1)}, r_1);$ $B = A \times_1 (X^{(1)})^T \times_3 (X^{(3)})^T;$ $X^{(2)} = SVD(B_{(2)}, r_2);$ $B = A \times_1 (X^{(1)})^T \times_2 (X^{(2)})^T;$ $X^{(3)} = SVD(B_{(3)}, r_3);$
结果：	$C = B \times_3 (X^{(3)})^T$

① $SVD(X, r)$ 表示对X做奇异值分解，并截取r个特征向量

5.3 识别过程

5.3.1 预处理

用张量分解的方法识别字符图像，需要有一部分作为训练集。训练集由是每个字符图像的特征向量组成，可以排列为一个张量如图5.3所示，图中张量的大小为特征 \times 样本数 \times 类别，即 $n_{feature} \times n_{samples} \times n_{classes}$ 。

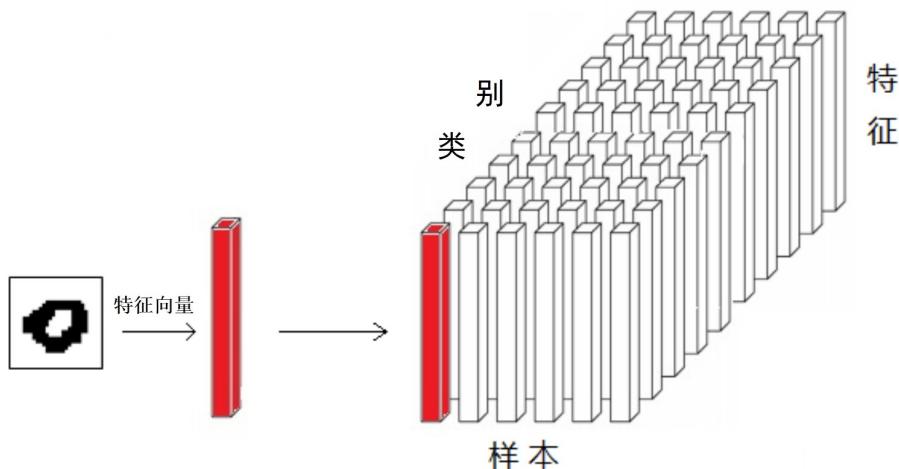


图 5.3 特征向量组成的张量

5.3.2 训练

训练的主要过程是张量的HOOI分解和特征矩阵的提取，得到如下结果：

$$A \approx C \times_1 U_{feature} \times_2 U_{samples} \times_3 U_{classes}. \quad (5-8)$$

其中核张量C反映了各个因子之间的关系， $U_{feature}$ ($n_{feature} \times n_{feature}$) 是在特征维度上的特征矩阵、 $U_{samples}$ ($n_{samples} \times n_{samples}$) 是在样本维度上的特征矩阵、 $U_{classes}$ ($n_{classes} \times n_{classes}$) 是在类别维度上的特征矩阵。为了减小数据量，本文对特征和样本维度均做了截断，分别取p和q，这样得到： $U_{feature}$ ($n_{feature} \times p$)， $U_{samples}$ ($n_{samples} \times q$)。张量分解的一个优点在于，识别过程中能忽略同一个类别因样本变化引起的差异。张量 $B = A \times_1 U_{feature}^T \times_2 U_{samples}^T$ 是一个映射张量，将对识别有着重要作用。

5.3.3 识别

由Tucker分解容易得到 $A = B \times_3 U_{classes}$ 。可以看出，之前分解得到的模式矩阵 $U_{classes}$ 的行向量 c_n^T 是每个类别n在映射张量B下独特的系数。固定B的“样本”维度指标s，得到一个子张量 B_s ，大小是 $p \times 1 \times n_{classes}$ ，或者看作一个 $p \times n_{classes}$ 的矩阵。任何一个训练图片的特征向量d先经过压缩

$$d^* = U_{feature}^T d \quad (5-9)$$

然后在样本s下的系数表示可以写成

$$c_s = B_s^T d^*. \quad (5-10)$$

识别时，给定一个未知的字符图像的特征向量 I ，可以在 B_s 下映射到系数 c_s ，识别方法就是对于任意的 s ，取可以使得 $\|c_s - c_n\|$ 值最小的 c_n ，则识别的类别即是 n 。可以写成下面的表达式

$$n^* = \arg \min_{s,n} \|c_s - c_n\|. \quad (5-11)$$

5.3.4 改进的识别方法

识别时采用式(5-14)求解原始最小二乘问题会使计算量非常大，因为需要遍历整个训练样本和类别。下面可以有一种更为快速的方法^[74]：令 $B^\mu = B(:, \mu)$ 是指每个类别的特征矩阵，每一列代表一个基向量，其中 μ 是 $1 \sim n_{class}$ 之间的类别，现把 B^μ 进行SVD分解并取其前 k 个最重要的特征向量，即：

$$B^\mu = D^\mu \Sigma^\mu Q^\mu, \mu = 1, 2, \dots, n_{class}. \quad (5-12)$$

取 D^μ 的前 k 列基向量构成一个映射矩阵 E^μ ，这样每个数字 μ 都有一个映射矩阵。给定一个未知的字符图像的特征向量 I ，可以通过计算

$$\mu^* = \arg \min_\mu \|I - E^\mu (E^\mu)^T U_{feature}^T I\| \quad (5-13)$$

得到最终的识别结果，这样只须遍历类别即可求解，大大加速了识别过程^[74]。

5.4 实验结果分析

在验证模型时，分别用手写数字数据库和古籍汉字数据库进行测试。

5.4.1 手写数字MNIST数据库

MNIST数据库是一套手写数字字符库，由60,000多个训练样本和10,000多个测试样本组成，每个样本是 28×28 的灰度图像^①。在实验中，均采用928维的局部二值模式（Local Binary Pattern, LBP）特征^[75]。

5.4.1.1 识别率与压缩率

在本文的研究中，对张量分解采用了近似截断，特征和样本维度的截断数分别是 p 和 q ，数据压缩率可以定义为：

$$\text{数据压缩率} = \left(1 - \frac{p \times q}{n_{feature} \times n_{samples}}\right) \times 100\%. \quad (5-14)$$

^① MNIST数据库主页：<http://yann.lecun.com/exdb/mnist/>

对于MNIST数据库来说，如果样本数 $n_{samples}$ 取1000个，当p=32，q=32时数据压缩率为99.89%。当p和q取其他值时，识别率和数据压缩率如表5.1所示：

表 5.1 识别错误率%（数据压缩率%）随不同截断数p, q的变化

$p \setminus q$	32	48	64	80
32	5.65 (99.89)	4.79 (99.83)	4.54 (99.78)	4.40 (99.72)
48	4.23 (99.83)	3.55 (99.75)	3.56 (99.67)	3.36 (99.59)
64	3.68 (99.78)	3.07 (99.67)	3.07 (99.56)	2.87 (99.45)
80	3.44 (99.72)	2.86 (99.59)	2.77 (99.45)	2.55 (99.31)

从表中可以看出，p、q越大，识别率越高，数据压缩率减小。在错误率小于2.6%的情况下能达到99.31%的数据压缩率。经实验发现，取训练样本 $n_{samples} = 5000$ ，在p=320，q=96时能达到最低的错误率1.99%，此时数据压缩率为99.34%。

5.4.1.2 不同特征的对比

本文与文献[72]中的方法做了对比。文献[72]中，有两种算法对手写数字进行识别。第一种算法是对每个数字构建一个张量，第二种方法与本文类似，而特征维度单纯使用像素。与LBP特征类似，还有一种特征，方向梯度直方图（Histogram of Oriented Gradient, HOG）也广泛用于图像局部特征提取和识别中。表5.2是本文的HOOI-LBP方法与其他方法的对比。

表 5.2 不同方法在MNIST数据库上的识别率

方法	HOSVD-Alg1	HOSVD-Alg2	HOOI-HOG	HOOI-LBP(本文)
识别率 (%)	93.88	94.83	96.44	97.45

通过对上述四种方法采用不同的k值，得到的识别率如图5.4所示（纵轴表示错误率）。

由图5.4可以发现，在和文献[72]同样的参数p和q的情况下，当k的值取12时本文中的方法（HOOI - LBP）达到了最佳识别效果，错误率为2.55%。相比文献[72]有了很大的改进。

5.4.1.3 结果分析

在测试MNIST库时，每个数字取400个样本作为训练，400个样本用作测试，阶段数k=12，经过实验得到正确率92.4%。图5.5是每个数字识别错误的个数统计

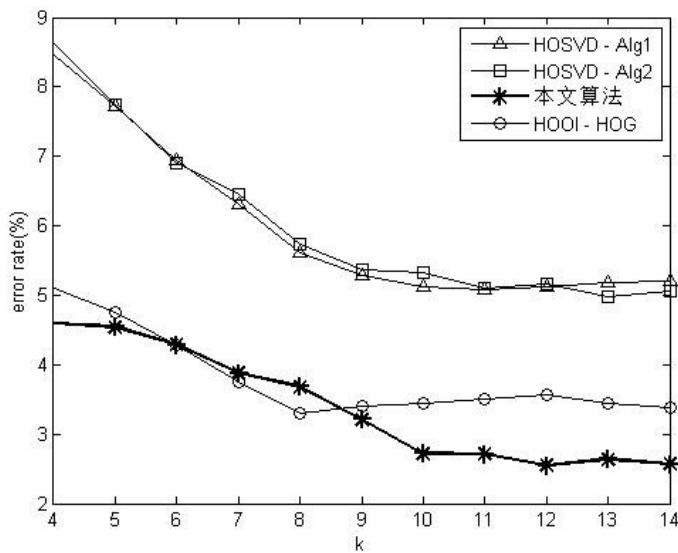
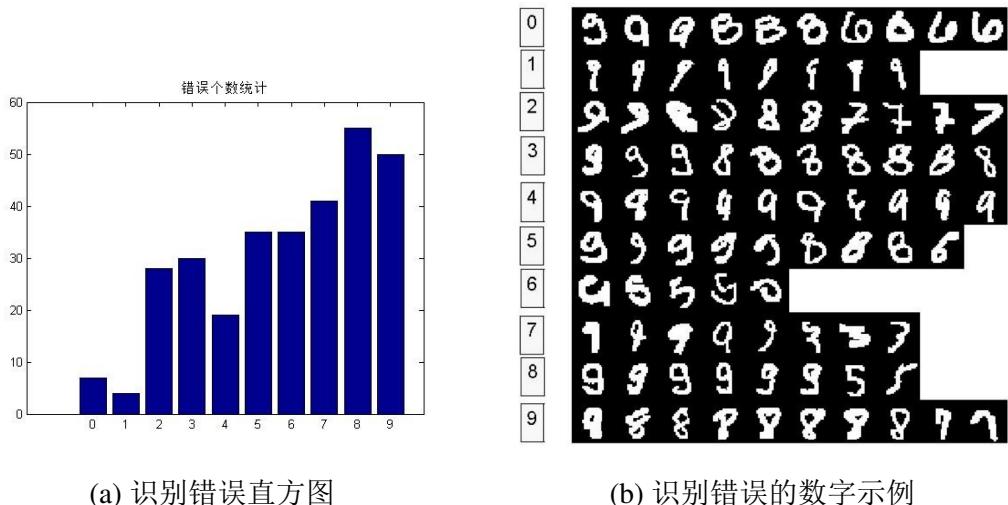


图 5.4 不同算法的错误率(%)随k值的变化



(a) 识别错误直方图

(b) 识别错误的数字示例

图 5.5 识别错误的数字分析

从图5.5可以看出，MNIST数据库中确实存在一些书写不规范的数字图片，在本识别程序中被较好地分离出来。但是同时，还有一些规范的数字也被识别错了，这是今后进一步研究需要改进的地方。

5.4.2 敦煌古籍数据库

本文在敦煌古籍数据库上做了测试，选取1000类、每个类别20个印刷体汉字的方向线索特征作为训练样本，然后比较最小欧氏距离、MQDF和张量分解方法。因为欧氏距离和MQDF方法都有LDA降维环节，所以再加“张量+LDA”一项，即用LDA降维后的特征向量作为训练时待分解的张量。总识别率如表5.3所示。

表 5.3 不同方法在古籍数据上的识别率比较

方法	欧氏距离	MQDF	张量分解法	张量分解法+LDA
识别率 (%)	76.14	78.90	80.08	77.64

表5.3中“张量分解法”的参数p（特征截断数）为200、参数q（训练样本的截断数）为10，可以与MQDF相比要稍微低一点。增加LDA后识别率更低，其实张量分解法本来是一种降维过程，不需要再用LDA降维处理。

同样为了探究古籍汉字识别率随p和q的变化，设定不同的p和q来看识别率的变化，如表5.4所示。

表 5.4 敦煌古籍汉字识别率 (%) 随不同截断数p, q的变化

p\q	5	10	15	20
50	71.87	72.61	72.61	72.87
100	76.83	78.30	78.34	78.35
150	78.13	79.80	79.71	79.73
200	78.33	80.08	80.05	80.08

从表5.4看出，一般来说，p和q越大，识别率越高，在p=200、q=10的时候张量分解法可以达到平均80.08%的识别率。

5.5 本章小结

本章介绍了张量的概念及其分解方法，同时介绍了基于张量分解的字符识别方法，同时在MNIST手写数字数据库和古籍汉字库上做了实验。实验证明张量分解法在训练时具有较快的速度，而且识别率随着截断数而变化。该方法计算量较大，耗时较长，适用于小字符集的分类，在大字符集上的识别处理还有待进一步改进。

第6章 在线古籍识别系统

6.1 本章引论

随着计算机、智能移动设备的普及和网络技术的快速发展，互联网已经渗透到方方面面，成为人们生活中获取信息资源的重要渠道。为了让古籍处理和识别技术为更多研究者提供服务，本文研发了一款在线古籍识别系统的原型，将本文中的研究成果以及常见汉字识别处理方法集成到该系统中。

6.2 在线OCR平台介绍

OCR技术越来越成熟，人们对OCR技术服务的需求也增多（比如图像文件转文本文件），在线OCR平台因其免安装、跨平台和易学易用的优点，被一些技术公司和组织采用。一般来讲，在线OCR平台是一个网站，用户可以提交一个含有文字的图像文件，然后该网站可以返回识别结果。

常见的在线OCR平台比如文通TH-OCR、OnlineOCR和RP-OCR。文通TH-OCR是北京文通科技有限公司的产品^①，其中TH-OCR资料数字化系统是专门应用在图书、报纸、杂志等书籍的数字化加工中，系统分管理端、加工端和数据库端。管理端由用户管理、角色管理、日志管理等构成，客户端由预处理、版面分析、字符切割、OCR、校对等模块组成。基于公司自主研发的高性能文字识别引擎，TH-OCR能识别多种文字，包括简体汉字、繁体汉字、英文、日文、韩文以及一些少数民族文字。

OnlineOCR是一款免费的基于网络的OCR软件^②，能够把PDF文件、传真、照片或者数字图像转为可编辑的文本。它支持46种语言（包括英语、汉语和西班牙语等）。在没注册的情况下，它可以每小时免费为用户转换15张图片。对于网络开发者，OnlineOCR还提供了网络服务的API，帮助网络开发者操作各种图像格式、识别图像文档、转换各种文本格式（Microsoft Word、PDF、TXT等）。

RP-OCR是云南瑞攀科技有限公司的产品^③，它能提供在线的OCR服务，支持常规图片格式，对影印和扫描的文件具有很好的支持。在识别中，它能将汉语、法语、日语和英语等多种文字混识别。它也能支持多种识别结果导出。

① 文通公司主页：<http://www.wintone.com.cn/a/default.aspx>

② OnlineOCR产品主页：<http://www.onlineocr.net>

③ RP-OCR产品主页：<http://www.rpocr.net>

然而，在调研的过程中发现现有的在线OCR平台都不支持古籍文档识别。

6.3 系统应用方案选择

搭建在线的应用系统有多种技术方案可供选择，该系统采用B/S结构，基于微软的.NET Framework 4.0，由C#编写网站后台程序。

6.3.1 B/S和C/S结构

常见的网络平台结构分为C/S和B/S。C/S结构是指“客户端/服务器（Client/Server）”结构，在这种结构下，普通用户的电脑上和服务器都有相应的软件，通过它可以充分利用两端的计算资源，同时可以在网络上传输精简的指令，降低了系统的通讯成本。常见的C/S结构比如智能手机的APP，通过不同的APP，用户可以享受不同的服务。B/S结构是指“浏览器/服务器（Browser/Server）”结构，在这种结构下，普通用户完全通过浏览器来浏览网页，不需要在用户的电脑上安装任何支持的软件，方便用户使用；网页的界面、网页后台运行和数据处理全部都在服务器端。这样做节约了成本，能够使不同平台的用户享受服务器端的服务。

C/S的不足之处在于开发者需要开发客户端和服务器端的两套程序，不利于程序维护；同时，由于客户端硬件设备限制和平台差异，客户端程序往往会遇到兼容性问题。B/S的不足之处在于所有页面数据都需要传递，网络通讯开销较大。考虑到古籍识别的用户应该是不同平台的研究者，因此在系统的构建时选择B/S的结构，方便研究者使用。

6.3.2 .NET Framework平台

B/S结构的服务器端常用平台是Java和.NET Framework。Java Web是一种跨平台的、技术标准开源的网站编程技术，通常由JSP（Java服务器页面，Java Server Pages）做页面表示，Servlet做任务处理。在开发时，需要安装JDK（Java Development Kit，Java开发工具），可以用eclipse和NetBeans等集成开发环境编写；开发完成后在任何一台装有JRE（Java Runtime Environment，Java运行时环境）的服务器上运行。很多大公司将自己的产品基于Java的技术规范，比如Oracle、IBM和JBoss等。

.NET Framework是微软公司提出的一种面向网站后台编程的开发平台。它能提供多语言的编程环境（C#、VB、ASP、VC++.NET等）。.NET Framework包括CLR（公共语言运行时，Common Language Runtime）和.NET Framework类库。

CLR是.NET Framework的基础，在代码执行的时候实时管理代码的运行，帮助回收内存。托管代码可以被编译为运行时能识别的代码比如Visual C++.NET，否则叫非托管代码，拥有自己的运行时，比如C++。类库是为方便开发应用程序（Windows窗体、命令行和ASP.NET网站等）提供的库函数^①。.NET Framework的结构介绍如图6.1所示。

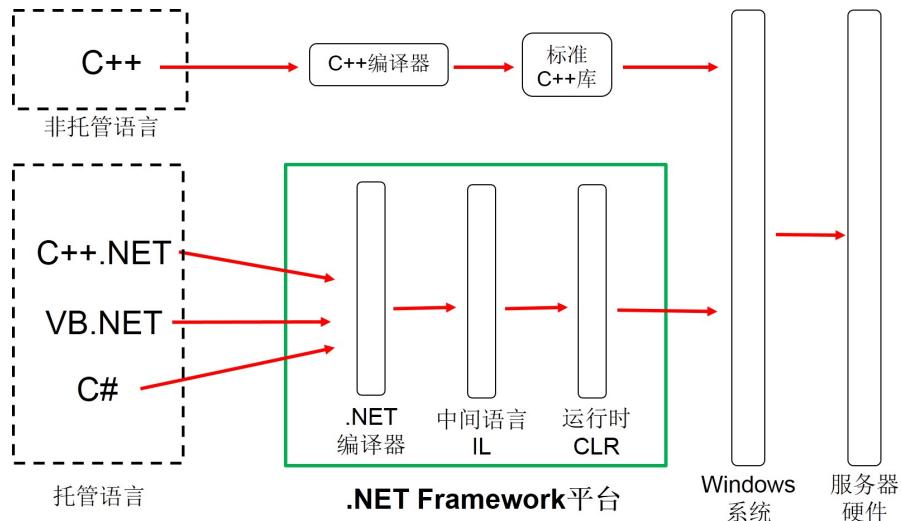


图 6.1 .NET Framework的结构介绍

.NET Framework具有编程效率高、上手快、开源资料丰富的特点，能够充分调用Windows的资源。它的缺点是目前只能运行在Windows系统中。不过在2014年11月12日，微软宣布将.NET堆栈基于MIT协议开源，托管在Github上^②，相信不久的将来，.NET Framework也能高效地运行在诸如Linux、Mac OS等其他操作系统中。

6.3.3 C#图像编程

C#是一款微软公司发布的高级程序设计语言，运行在.NET Framework环境中。它综合了VB和Delphi的可视化操作特点、C/C++较高的运行效率和Java的语法和编译方法，非常容易上手。

C#可以调用GDI+库来显示和处理数字图像。GDI+（Graphics Device Interface plus，图形硬件接口升级版）是一组应用程序与各种图形设备交互的库函数，是微软Windows系统的核心部件。具体操作时先用Bitmap对象读入图片，然后用Graphics.FromImage方法在Bitmap对象的基础上新建一个Graphics类，

① .NET官方介绍：<https://msdn.microsoft.com/zh-cn/library/zw4w595w.aspx>

② .NET开源主页：<https://github.com/dotnet>

对Graphics对象进行各种操作（裁剪、旋转、绘图）后释放，即可对原来的Bitmap做出相应处理。

6.4 系统构成与实现

在线古籍图像识别系统（V1.0）能够上传单张古籍图片，然后按照使用者的意图进行预处理、字符切分和识别。在识别中，可以选择BIG5或者GB2312的编码方式。系统的结构如图6.2所示。

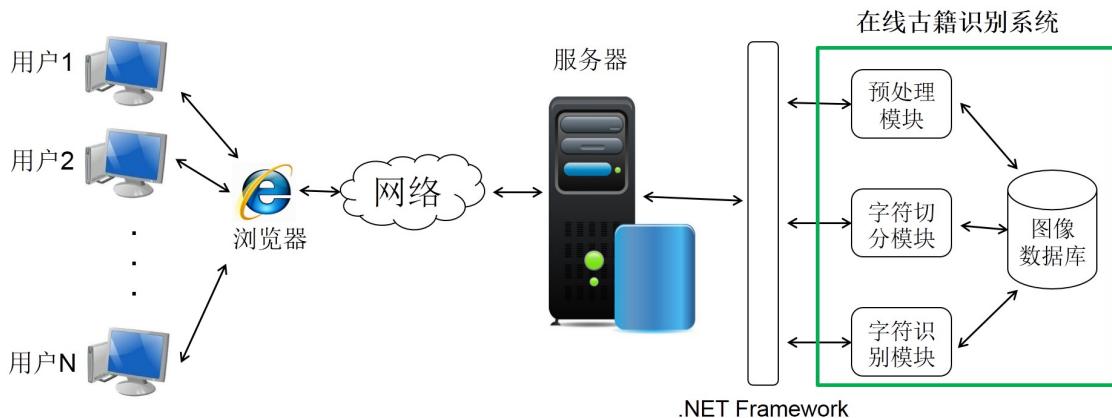


图 6.2 在线古籍识别系统的结构图

该系统主要分为四个部分：预处理、字符切分、字符识别和图像数据库。接下来将详细介绍各部分的实现原理。在预处理模块中，采用C#的GDI+图形库，将彩色的古籍图像转为灰度图像，然后再用阈值法对灰度图像进行二值化处理；生成的二值化图片显示在网页中，同时将结果保存在图像数据库中，方便下一阶段的处理。在字符切分模块中，由于现有字符切分的程序为MATLAB语言，故采用Matlab BuilderTM NE^①，将MATLAB函数编译生成.NET组件，然后在C#中调用（详见附录C）。切割完成后，在网页中显示切割结果，同时把切割好的每个字符图像储存在图像数据库中，方便下一阶段的处理。在字符识别模块中，先调用C++编译的提取方向线索特征的EXE程序，将每个切割好的字符图像提取特征，然后根据用户选择的方法（最小欧氏距离、MQDF、STM和GP-STM）进行识别。识别返回结果显示在网页中。

识别的具体技术细节参见第3章、第5章和附录A。其中识别需要的系数（比如均值向量和变换矩阵等）都事先训练好，保存在图像数据库中，在识别的时候加载相应的数据即可。

^① MATLAB编译介绍：<http://cn.mathworks.com/products/matlab-compiler-sdk/index.html>

6.5 系统使用演示

打开网页，可以看到在线古籍识别系统（V1.1）的主页如图6.3所示。



图 6.3 在线古籍识别系统初始界面

点击上方按钮“浏览...”，可以选择待识别的古籍图像，然后点击“古籍源文件”下面的“上传原图”按钮，可以看到“古籍源文件”下方的图片框已经成功加载原图，如图6.4所示。

点击“预处理”，鼠标变为等待的图标，等待系统预处理，然后将预处理后的图像显示在“预处理后图片”中。同样再依次点击“字符切分”和“识别”，即可在下方的“识别结果”文本框中看到识别的文本，如图6.5所示。需要注意的是，在识别前需要选择编码方式和识别方法。

在本文的实验中，古籍文档图像来自《昌平山水记（卷上）》。从图中的结果可以看出，该系统在字符分割中能自动检测字符区域，滤除列分割线，得到较好的字符切分结果，能按照古籍文档图片自动分段，达到了较高的识别率。

6.6 总结

本章中介绍了在线古籍识别系统，并且介绍了系统的搭建、系统组成以及使用演示。目前该系统尚处于初级版本，功能还不太完善，但是已经具备了在线古籍识别系统的原型框架，为今后的工作打下了扎实的基础。



图 6.4 在线古籍识别系统上传原图



图 6.5 在线古籍识别系统识别结果

第7章 结论

7.1 研究结论

本文在研究中，就手写体古籍汉字的识别方法进行了较为深入的探究，具体分为以下几个方面。

首先，本文在文献调研的基础上，介绍了古籍数字化的进展和常用研究方法，以国际敦煌项目IDP和欧洲IMPACT古籍识别项目为例介绍了典型的古籍数字化项目；较为详细地介绍了传统汉字识别中的特征提取、降维和分类器设计等方法；引入迁移学习的概念，介绍了线性迁移学习方法及其在古籍汉字识别中的应用；介绍了高斯过程的原理和应用领域。从文献调研来看，本文的研究基于迁移学习的思想，采用较为成熟的高斯过程回归模型，在已有的线性迁移学习方法的基础上做了非线性推广，对古籍数字化项目中的古籍全文数字化过程具有较大的帮助。

其次，本文重点介绍高斯过程风格迁移映射（GP-STM）模型。根据古籍汉字的特点，本文提出三条假设：回归假设、高斯过程假设和同协方差矩阵假设；在假设条件的基础上，本文通过数学推导将线性的迁移学习映射（STM）推广到非线性的GP-STM，给出了模型的变换公式，并讨论了模型参数的一种优化方法；为了验证GP-STM模型的有效性，本文采用敦煌古籍汉字和中科院手写汉字作为实验样本，在识别率上与传统方法做了对比，探究了STM训练集的比例与识别率的关系，对比了各种核函数的作用，同时做了GP-STM模型中变换过程的可视化展示。从实验结果可以看出，本文提出的GP-STM模型相比线性STM模型具有更强的映射能力，通过使用高斯过程的协方差矩阵，可以在风格变换中将相关度较高的汉字赋以更高的权重，增强风格变换的效果，较大地提升古籍汉字的识别率，这种基于迁移学习的方法也能较为有效地解决古籍汉字训练样本少的问题。当然，在实验中发现采用GP-STM比传统的线性STM需要更多的运行时间和内存，计算复杂度也更大。

再次，在进行迁移学习的研究工作之前，本文也尝试用其他方法研究手写体古籍汉字识别，基于张量分解的字符识别方法是其中之一。本文简单介绍了张量的概念、运算以及两种分解方式：CP分解和Tucker分解；用数学公式推导了基于Tucker分解的字符识别方法；在手写数字MNIST数据库和古籍汉字数据库上做了实验，与传统的方法进行对比，同时分析了实验结果。从实验结果可以看出，

本文提出的基于张量分解的识别方法将训练样本的特征向量由特征、样本和类别三个维度组成一个张量，然后通过张量的Tucker分解保留对分类有用的信息，在古籍汉字类别数稍小（1000类）时，识别率要优于传统的方法，训练速度快。当然，在实验中发现这种方法需要的内存非常大，而且测试速度较慢。

最后，本文开发了在线古籍识别系统的原型。本文简单介绍了现有的在线字符识别（OCR）平台，发现它们都不支持古籍文档识别；然后通过调研一般的网站编程技术，选定了本文中的系统的应用方案，即基于B/S结构，通过C#语言在.NET Framework平台上搭建在线古籍图像识别系统。该系统整合了现有的文档预处理、字符切割和汉字识别技术，让用户自主选择处理过程，预期为中文古籍数字化项目的研究者提供跨平台、免安装、高性能的古籍文档图像识别服务。

7.2 需要进一步开展的工作

在后续的工作中，还有较多可以进一步开展的工作，部分如下：

首先是迁移学习理论方面的工作，可以寻找更好的非线性映射，或者换一种迁移学习的思路，通过对源域、目标域的样本分析，动态修正分类器，使得源域的分类器能够适应目标域；同时尝试方向线索之外的其他特征，在特征提取的层面就不同域之间的转换做深入研究，制定符合特征的识别策略；

其次是高斯过程理论方面的工作，如果不采用同协方差矩阵假设，高斯过程回归模型将会变得非常复杂^[53]，所以可以研究GP-STM的原理，探究其起作用的深层机制，为寻找下一个更加具有通用型的模型；同时尝试将GP-STM模型用在其他领域的识别问题上，比如人脸识别、人脸验证等等；

然后是探究张量分解在大字符集识别中的应用，探索张量CP分解在字符识别中的应用，同时寻找Tucker分解较少占用内存的方法；探索新的快速识别算法，使之能够达到初步的实用效果；

最后是在线古籍图像识别系统的后续改进，目前的系统在系统管理、界面、人机交互、识别准确率方面都存在一些问题，字符切分算法还需要更加自动化，其他识别方法也有待进一步加入。同时，当前系统只能识别在繁体印刷体汉字字符集中的古籍汉字，对于超出字符集的古籍汉字，未来可以考虑用部件组合的方法识别，即首先识别出其每个部件（部首和笔画等），然后将各个部分合并求解。此外，当系统的使用人数增多后，现有的后台文件系统须改为数据库系统，并增加多线程计算，保证并发任务的顺利运行。

参考文献

- [1] 李璐. 古籍全文数据库建设的技术与实践. 图书馆学研究, 2004, 11:22–25.
- [2] Kimura F, Wakabayashi T, Tsuruoka S, et al. Improvement of handwritten Japanese character recognition using weighted direction code histogram. Pattern recognition, 1997, 30(8):1329–1337.
- [3] Ojala T, Pietikainen M, Maenpaa T. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24(7):971–987.
- [4] Dalal N, Triggs B. Histograms of oriented gradients for human detection. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 1, 2005. 886–893.
- [5] 张学工. 模式识别（第三版）. 清华大学出版社, 2010.
- [6] Zhang H, Guo J, Chen G, et al. HCL2000-A large-scale handwritten Chinese character database for handwritten character recognition. 10th International Conference on Document Analysis and Recognition, 2009. 286–290.
- [7] Su T, Zhang T, Guan D. Corpus-based HIT-MW database for offline recognition of general-purpose Chinese handwritten text. International Journal of Document Analysis and Recognition, 2007, 10(1):27–38.
- [8] Liu C L, Yin F, Wang D H, et al. Online and offline handwritten Chinese character recognition: benchmarking on new databases. Pattern Recognition, 2013, 46(1):155–162.
- [9] Caruana R. Multitask learning. Springer, 1998.
- [10] Tur G. Multitask learning for spoken language understanding. IEEE International Conference on Acoustics, Speech and Signal Processing, volume 1, 2006. I–I.
- [11] Pan S J, Yang Q. A survey on transfer learning. IEEE Transactions on Knowledge and Data Engineering, 2010, 22(10):1345–1359.
- [12] Raina R, Battle A, Lee H, et al. Self-taught learning: transfer learning from unlabeled data. Proc. the 24th international conference on Machine learning, 2007. 759–766.
- [13] 毛建军. 古籍数字化的概念与内涵. 图书馆理论与实践, 2007, 4:82–84.
- [14] 周迪, 宋登汉. 中文古籍数字化开发研究综述. 图书情报知识, 2010, 6:40–49.
- [15] 师文. 海峡两岸中国古籍整理研究现代化技术研讨会在京举行. 语文建设, 1993, 12.
- [16] 王桂平. 我国古籍数字化的现状及展望. 图书情报知识, 2000, 4:50–51.
- [17] Peng L, Xiu P, Ding X. Design and development of an ancient Chinese document recognition system. Proc. SPIE 5296, Document Recognition and Retrieval XI, 2003. 166–173.
- [18] Zhang X, Nagy G. The CADAL calligraphic database. Proc. the 2011 Workshop on Historical Document Imaging and Processing, 2011. 37–42.
- [19] 王婷婷, 董超俊. 图像增强技术在古籍图书电子化中的应用. 五邑大学学报（自然科学版）, 2015, 29(1):26–29.

- [20] 姜哲, 马少平, 夏莹. 大型中文古籍《四库全书》自动版面分析系统. 中文信息学报, 2000, 14(2):14–20.
- [21] 朱雷. 古籍手写汉字图像分割算法研究[D]. 重庆大学, 2011.
- [22] 衡中青. 地方志知识组织及内容挖掘研究[D]. 南京农业大学, 2007.
- [23] 贾雪莎. 基于对称区域的古籍汉字图像检索[D]. 河北大学, 2014.
- [24] 张彩录, 郭宝兰, 张宇桐, et al. 一个实用的古籍印刷汉字识别系统. 中文信息学报, 1996, 10(3):43–49.
- [25] 郑惠珍. 敦煌古籍流失及其整理的研究. 茂名学院学报, 2002, 12(2):32–34.
- [26] 赵继印, 郑蕊蕊, 吴宝春, et al. 脱机手写体汉字识别综述. 电子学报, 2010, 38:405–415.
- [27] 邵洁, 成瑜. 关于手写汉字切分方法的思考. 计算机技术与发展, 2006, 16(6):184–190.
- [28] 丁晓青. 汉字识别研究的回顾. 电子学报, 2002, 30(9):1364–1368.
- [29] 周昌乐, 张雄伟. 一种基于段化的手写汉字特征点提取方法及其实现. 电子学报, 1997, 25(5):57–60.
- [30] Tseng Y H, Lee H J. Recognition-based handwritten Chinese character segmentation using a probabilistic Viterbi algorithm. Pattern Recognition Letters, 1999, 20(8):791–806.
- [31] Ji J, Peng L, Li B. Graph model optimization based historical chinese character segmentation method. Document Analysis Systems (DAS), 2014 11th IAPR International Workshop on, 2014. 282–286.
- [32] Sun X, Peng L, Ding X. Touching character segmentation method for chinese historical documents, 2010. <http://dx.doi.org/10.11117/12.840251>.
- [33] 周昌乐, 张雄伟. 一种基于段化的手写汉字特征点提取方法及其实现. 电子学报, 1997, 25(5):57–60.
- [34] 耿强, 马珏. 手写体汉字识别笔画提取方法的研究. 江苏广播电视台学报, 2006, 1(17):41–43.
- [35] Cao R, Tan C L. A model of stroke extraction from chinese character images. Proceedings of 15th International Conference on Pattern Recognition, 2000. 4368–4371.
- [36] 吴佑寿. 教电脑识字:浅谈汉字识别. 北京: 清华大学出版社, 广州: 暨南大学出版社, 2000.
- [37] Shi D, Damper R I, Gunn S R. Offline handwritten chinese character recognition by radical decomposition. ACM Transactions on Asian language information processing (TALIP), 2003, 2(1):27–48.
- [38] 何浩智, 朱宁波, 刘伟. 基于霍夫变换和弹性网格的手写汉字识别方法. 计算机仿真, 2008, 25(1):240–243.
- [39] 陈光, 张洪刚, 郭军. 一种新的加权动态网格汉字特征抽取方法. 中文信息学报, 2007, 21(2):89–93.
- [40] 杨玲, 毛以芳, 吴天爱. 基于弹性网格和方向线素特征的脱机手写汉字识别. 辽宁省交通高等专科学校学报, 2008, 10(1):38–39.
- [41] 马少平, 夏莹, 朱小燕. 基于模糊方向线素特征的手写体汉字识别. 清华大学学报(自然科学版), 1997, 37(3):42–45.
- [42] 王学文, 丁晓青, 刘长松. 基于Gabor 变换的高鲁棒汉字识别新方法. 电子学报, 2002, 30(9):1317 – 1322.

- [43] 王先梅, 杨扬, 颉斌. 基于Krawtchouk矩与HMM的脱机手写汉字识别技术. The Sixth World Congress on Intelligent Control and Automation (WCICA 2006), 2006. 10068 – 10072.
- [44] Fu Q, Ding X, Liu C. Cascade MQDF classifier for handwritten character recognition. Journal of Tsinghua University (Science and Technology), 2008, 48(10):1065–1068.
- [45] 王建平, 张丽萍. 脱机手写体汉字识别的支持向量机方法研究. 计算机与数字工程, 2008, 36(1):146 – 150.
- [46] Lu D, Chen Q, Pu W, et al. Study on pre-classification for handwritten Chinese character based on neural net and fuzzy matching algorithm. 2007 IEEE International Conference on Robotics and Biomimetic (ROBIO), 2007. 1344 –1349.
- [47] 赵巍, 刘家锋, 唐降龙. 基于部件HMM级联的联机手写体汉字识别方法. 哈尔滨工业大学学报, 2004, 36(5):570 – 573.
- [48] 刘健, 李会方, 牛新伟. 基于MHMM模型的手写体汉字识别算法. 信息安全与通信保密, 2007, 36:75–77.
- [49] 高彦宇, 杨杨. 脱机手写体汉字识别研究综述. 计算机工程与应用, 2004, 40(7):74 – 77.
- [50] 陈友斌, 丁晓青, 吴佑寿. 非特定人脱机手写汉字识别. 中国计算机报, 1997. <Http://media.ccidnet.com/media/ciw/663/01350001.htm>.
- [51] Jiang J, Zhai C. Instance weighting for domain adaptation in NLP. In ACL 2007, 2007. 264–271.
- [52] Dai W, Xue G R, Yang Q, et al. Co-clustering based classification for out-of-domain documents. Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, 2007.
- [53] Bonilla E V, Ming K, Chai A, et al. Multi-task Gaussian process prediction. Nips, 2008..
- [54] Wu P, Dietterich T G. Improving svm accuracy by training on auxiliary data sources. In ICML, 2004. 871–878.
- [55] Pan S J, Kwok J T, Yang Q, et al. Adaptive localization in a dynamic Wi-Fi environment through multi-view learning. Proc. 22nd AAAI Conf. Artificial Intelligence (AAAI 07), AAAI Press, 2007, 2007, 2:1108–1113.
- [56] Zhang X Y, Liu C L. Writer adaptation with style transfer mapping. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2013, 35(7):1773–1787.
- [57] Li B, Peng L, Ji J. Historical Chinese character recognition method based on style transfer mapping. 11th IAPR International Workshop on Document Analysis Systems, 2014. 96–100.
- [58] Bishop C M, et al. Pattern recognition and machine learning. Springer, 2006.
- [59] Rasmussen C E. Gaussian processes for machine learning. MIT Press, 2006.
- [60] Ebden M. Gaussian processes for regression: A quick introduction. Technical report, 2008. <Http://www.robots.ox.ac.uk/~mebden/reports/GPtutorial.pdf>.
- [61] Boyle P. Gaussian processes for regression and optimisation. Victoria University of Wellington, 2007..
- [62] O'Hagan A, Kingman J. Curve fitting and optimal design for prediction. Journal of the Royal Statistical Society. Series B (Methodological), 1978. 1–42.

- [63] Rumelhart, E D, Hinton, et al. Learning representations by back-propagating errors. *Nature*, 1986, 323(6088):533–536.
- [64] Neal R M. Bayesian learning for neural networks[D]. Springer New York, 1996.
- [65] Rasmussen C, Ghahramani Z. Infinite mixtures of Gaussian process experts. In *Advances in Neural Information Processing Systems 14*, 2001. 881–888.
- [66] Paciorek C. Nonstationary Gaussian processes for regression and spatial modelling[D]. Carnegie Mellon University, 2003.
- [67] Lawrence N D, Platt J C. Learning to learn with the informative vector machine. *Proc. the twenty-first international conference on Machine learning*, 2004. 65–73.
- [68] Schwaighofer A, Tresp V, Yu K. Learning Gaussian process kernels via hierarchical Bayes. *Advances in Neural Information Processing Systems*, 2004. 1209–1216.
- [69] De Lathauwer L, De Moor B, Vandewalle J. A multilinear singular value decomposition. *SIAM journal on Matrix Analysis and Applications*, 2000, 21(4):1253–1278.
- [70] Lathauwer L D, Moor B D, Vandewalle J. On the best rank-1 and rank-(r 1,r 2,...,r n) approximation of higher-order tensors. *SIAM journal on matrix analysis and applications*, 2000, 21(4):1324–1342.
- [71] Sheehan B, Saad Y. Higher order orthogonal iteration of tensors (HOOI) and its relation to PCA and GLRAM. *SIAM international conference on data mining*, 2007..
- [72] Savas B, Elden L. Handwritten digit classification using higher order singular value decomposition. *Pattern recognition*, 2007, 40(3):993–1003.
- [73] 周丙寅. 张量分解及其在动态纹理中的应用[D]. 河北师范大学, 2012.
- [74] Elden L. Matrix methods in data mining and pattern recognition. SIAM, 2009..
- [75] Vedaldi A, Fulkerson B. VLFeat: An open and portable library of computer vision algorithms. <http://www.vlfeat.org/>, 2008.
- [76] Yamada H, Yamamoto K, Saito T. A nonlinear normalization method for handprinted kanji character recognition—line density equalization. *Pattern Recognition*, 1990, 23(9):1023–1029.
- [77] Belhumeur P N, Hespanha J P, Kriegman D. Eigenfaces vs. Fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1997, 19(7):711–720.
- [78] Kimura F, Takashina K, Tsuruoka S, et al. Modified quadratic discriminant functions and the application to Chinese character recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1987, 9(1):149–153.

致 谢

在硕士学习期间，我得到了许多帮助。衷心感谢导师彭良瑞副教授对本人的精心指导，她严谨的治学态度将使我终生受益。感谢实验室的老师们和同学们的热情帮助和支持，感谢无研136班的同学们，我们曾经一起为冲刺校级优秀集体奋力拼搏。在研究过程中，感谢李鹏超同学在理论知识方面的帮助，感谢李博晗同学在STM方面的帮助，感谢梁亦聪师兄对高斯过程的介绍以及对分类器的指正，感谢叶浩师兄在读取字符文件方面的支持，感谢胡晓灵同学在编辑 L^AT_EX语法时的支持，感谢王彦伟师兄在样本库方面的支持。感谢国家图书馆提供的敦煌古籍样本。感谢 THU^AESIS，它的存在让我能充分利用 L^AT_EX将版面编辑得漂亮。

在学术论文的投稿中，感谢DRR 2015的匿名评审员的建议，感谢Marquette大学的Michael T. Johnson教授，他曾逐行地帮我修改英语论文表达。

还要感谢父母对我的养育之恩，感谢亲人和朋友们在我求学道路上的关怀和鼓励，感谢女友卉卉的陪伴和支持。

论文研究工作得到国家自然科学基金委员会（NSFC）与法国国家科研署（Agence Nationale de la Recherche, ANR）共同资助的中法合作项目“手写体中文古籍识别”（项目编号：61261130590）的支持。论文相关研究工作还得到了973项目“面向三元空间的互联网中文信息处理理论与方法”（子课题编号：2014CB340506）的部分资助，特此致谢。

声 明

本人郑重声明：所呈交的学位论文，是本人在导师指导下，独立进行研究工作所取得的成果。尽我所知，除文中已经注明引用的内容外，本学位论文的研究成果不包含任何他人享有著作权的内容。对本论文所涉及的研究工作做出贡献的其他个人和集体，均已在文中以明确方式标明。

签 名： _____ 日 期： _____

附录 A 汉字识别常用技术

A.1 预处理

在预处理过程中，本研究用到了二值化、裁剪空白区域、重心居中、字符线密度均衡等技术手段。二值化过程采用简单的阈值法，对灰度图像进行量化；在裁剪了空白的背景区域后，找出字符图像的重心，将图像分为4块（左上角到重心、右上角到重心、左下角到重心、右下角到重心），然后把每块缩放为大小相同的四小块，然后进行对应拼接。下面详细介绍字符线密度均衡^[76]。

首先对线密度进行定义。设图像为 $f(i, j), i = 1, 2, \dots, H; j = 1, 2, \dots, W$ ，其中 (i, j) 是第*i*行*j*列的像素。图像的前景（字符）像素值为1，背景像素值为0。假定在x方向的笔画右边缘为 L_1 和 L_2 ，笔画的左边缘为 L_3 和 L_4 ，同时 L_1 和 L_3 在像素点 (i, j) 的左边， L_2 和 L_4 在像素点 (i, j) 的右边，同时满足

$$\begin{aligned} L_1 &= \arg \max_{i'} \{i' < i, f(i', j) \cdot \overline{f(i'+1, j)} = 1\} \\ L_2 &= \arg \min_{i'} \{i' \geq i, f(i', j) \cdot \overline{f(i'+1, j)} = 1\} \\ L_3 &= \arg \max_{i'} \{i' < i, \overline{f(i'-1, j)} \cdot f(i', j) = 1\} \\ L_4 &= \arg \min_{i'} \{i' \geq i, \overline{f(i'-1, j)} \cdot f(i', j) = 1\} \end{aligned} \quad (\text{A-1})$$

在一些情形下， L_1, L_2, L_3, L_4 之中会有一些不存在，比如在行方向几种情况

$$L_x = \begin{cases} 2W, & \text{仅 } L_1, L_3 \text{ 不存在} \\ 2W, & \text{仅 } L_2, L_4 \text{ 不存在} \\ 2W, & \text{仅 } L_1, L_4 \text{ 不存在} \\ L_4 - L_3, & \text{仅 } L_1 \text{ 不存在} \\ L_2 - L_1, & \text{仅 } L_4 \text{ 不存在} \\ (L_2 - L_1 + L_4 - L_3)/2, & L_1, L_2, L_3 \text{ 和 } L_4 \text{ 都存在} \\ 4W, & \text{其他情况} \end{cases} \quad (\text{A-2})$$

同样，在列方向也可以计算得到 L_y ，然后 $\min(L_x, L_y)$ 可被粗略视作每个点的内切

圆的直径。在这里，线密度 ρ 定义为直径的倒数，即

$$\rho(i, j) = \begin{cases} \max(W/L_X, W/L_Y), & \text{如果 } L_X + L_Y < 6W \\ 0, & \text{如果 } L_X + L_Y \geq 6W \end{cases} \quad (\text{A-3})$$

现在，把线密度函数投影到x轴或y轴，可以得到投影函数 $h_X(i)$ 和 $h_Y(j)$ ，以及它们的累积和 C_X 和 C_Y ：

$$\begin{cases} h_X(i) = \sum_{j=1}^W \rho(i, j), C_X = \sum_{i=1}^H h_X(i) \\ h_Y(j) = \sum_{i=1}^H \rho(i, j), C_Y = \sum_{j=1}^W h_Y(j) \end{cases} \quad (\text{A-4})$$

通过实验可知，大多数字字符图像的 $h_X(i)$ 和 $h_Y(j)$ 函数是不均匀的曲线，为了达到均衡的目的，需要把“密集”的地方放大，把“稀疏”的地方缩小，即下面的变换

$$\begin{cases} i' = \min_j \left\{ \sum_{k=1}^{i'} h_X(k) \geq i \cdot C_X / H \right\} \\ j' = \min_i \left\{ \sum_{k=1}^{j'} h_Y(k) \geq j \cdot C_Y / W \right\} \end{cases} \quad (\text{A-5})$$

线密度的示意图如图A.1所示。

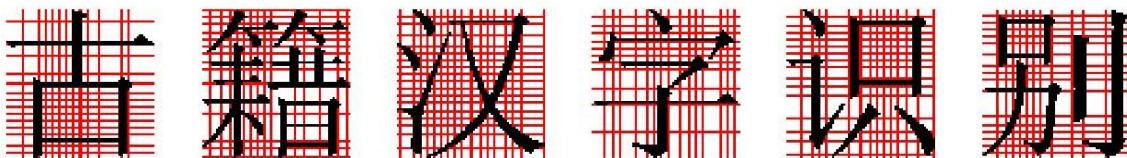


图 A.1 线密度的示意图

A.1.1 字符切分

针对敦煌古籍文档的特点，本文采用较容易操作的投影法进行切割。具体过程是如算法3所示。

实验的过程如图A.2所示。

A.2 字符的特征提取和降维

A.2.1 特征提取

给字符图像提取特征向量、降维，是识别的准备步骤。这项工作的前提是字符图像已经完成了预处理工作（比如二值化、大小归一化、笔画均匀化等等）。本文的研究只用到了方向线索特征、Fisher线性鉴别分析降维，因此在这里着重介绍这两个方法。

算法3 古籍图像的投影法切割**目标：**给古籍图像，切分出其中的每个汉字

- 具体步骤：**
1. 对文档图像进行局部二值化和缩放操作，便于后续工作，减小计算量
 2. 统计水平方向的像素直方图，根据直方图峰值找出古籍的文本区域
 3. 在文本区域中统计垂直方向的像素直方图，据直方图峰值找出古籍的列区域
 4. 对每一列单独统计水平方向的像素直方图，找出每个字的位置
 5. 在每一列中，根据字符的垂直高度适当合并过切分字符

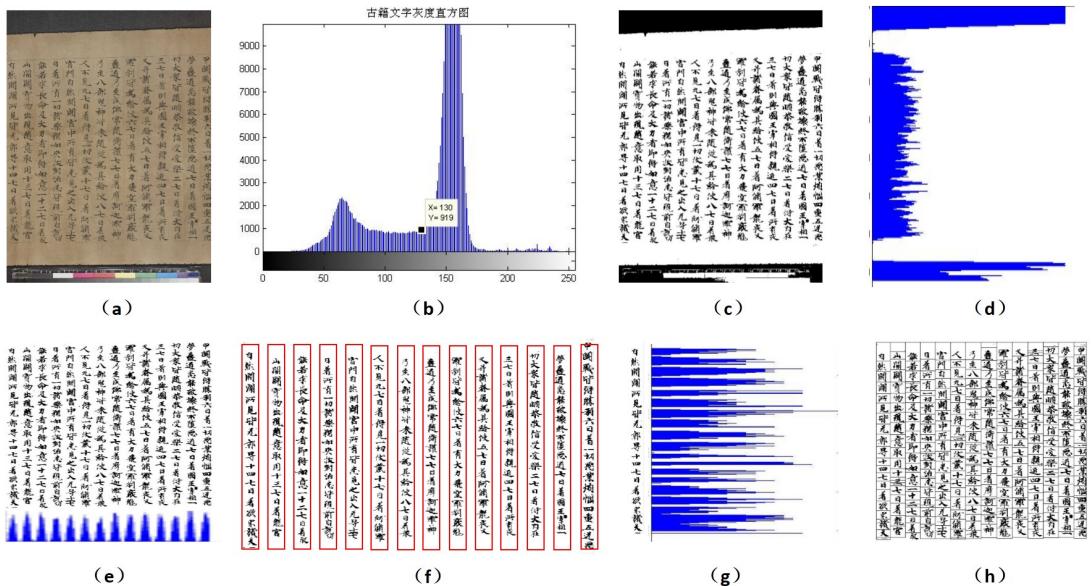


图 A.2 敦煌古籍样本的切割过程 (a) 是古籍图像的原始样本; (b) 是古籍图像的灰度直方图, 从图中选择灰度的分界点; (c) 是古籍二值化以后的图像; (d) 是古籍图像在垂直方向上的投影; (e) 是选取的文档内容, 并做了水平方向投影; (f) 是古籍图像列切割的结果 (列为自动切开, 红线是为了直观手动标注); (g) 是左起第一列的垂直投影; (h) 是文档图像字符切割的处理结果。

方向线索特征又称加权方向编码直方图 (Weighted Direction Code Histogram)^[2], 缩写为 WDH 特征。它的提取过程分为边缘提取、链式方向编码和采样。其中边缘提取和链式方向编码可以采用查表的方式一齐进行。具体方式是, 对于输入的二值图像每个像素点为中心取一个 3×3 的小窗 (图像四周的像素点设为 0), 如果该像素点为前景色 (字符), 则根据该像素点周围的 8 个像素值, 自动查找对应的直方图。具体的方式如下图A.3所示。

图像块				
	1	2	3	4
图像块				
	5	8	7	6

图 A.3 8方向模板示意图

在本文的研究中，采用 65×65 的二值字符图像，在提取边缘、得到方向编码后，将图像分为13个不重叠的小块（每个小块是 5×5 ），然后在小块中统计直方图。比如“古”字的字符图有如图 A.4 的提取过程

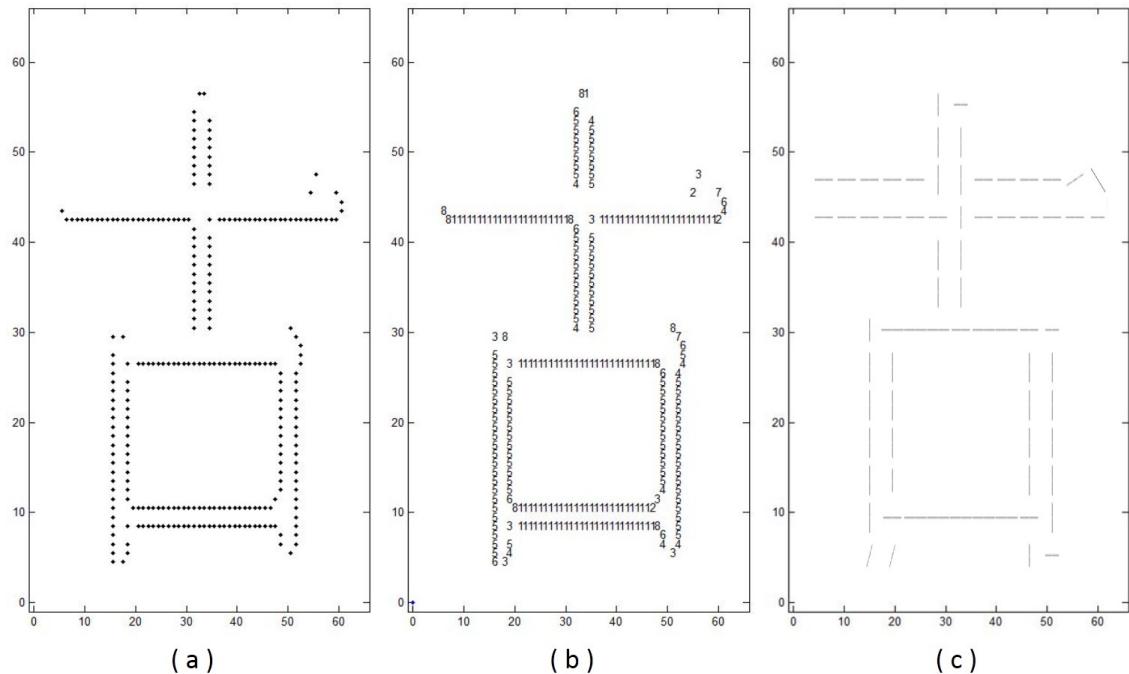


图 A.4 用WDH特征提取汉字“古”的示意图。其中(a)是图像的边缘，(b)是每个边缘像素点的方向编码，(c)是分块统计的方向直方图。

这样可以得到一个 $13 \times 13 \times 8$ 的原始特征，可以先简单进行采样处理。常用的做法是，用一个 5×5 的高斯滤波器每两个块采样，将 13×13 降为 7×7 ，这样成了一个 392 维的特征向量。为了让特征向量更像高斯分布，特征向量的每一维都取平方根，得到最终的向量。

A.2.2 特征降维

在汉字识别前，还需要进一步对特征向量降维。本文的研究中用到了Fisher线性判别分析（Linear Discriminant Analysis, LDA）^[77]。LDA希望能找到一个投影方向，使得投影后类内距离小，类间距离大，更便于分类。假定全部N个特征向量 $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_N$ 的维数是d维，全部的类别为c类 C_1, C_2, \dots, C_c 。通过映射矩阵W，d维的特征向量被映射到d' ($d' < d$) 维，新的特征向量 $\mathbf{x}^* \in \mathbb{R}^{d'}$ 由线性映射定义：

$$\mathbf{x}_k^* = W^T \mathbf{x}_k, k = 1, 2, \dots, N \quad (\text{A-6})$$

这里 $W \in \mathbb{R}^{d \times d'}$ 。设 $\mu \in \mathbb{R}^d$ 为全部特征向量的均值， $\mu_i \in \mathbb{R}^d$ 为第i类的均值，同时定义全体散度矩阵 S_T 、类内散度矩阵 S_W 和类间散度矩阵 S_B 为：

$$\begin{aligned} S_T &= \frac{1}{N} \sum_{i=1}^N (\mathbf{x}_k - \mu)(\mathbf{x}_k - \mu)^T, \\ S_W &= \frac{1}{N} \sum_{i=1}^c \sum_{\mathbf{x}_k \in C_i} (\mathbf{x}_k - \mu_i)(\mathbf{x}_k - \mu_i)^T, \\ S_B &= \frac{1}{N} \sum_{i=1}^c N_i (\mu_i - \mu)(\mu_i - \mu)^T. \end{aligned} \quad (\text{A-7})$$

其中 N_i 是第i类特征向量的个数，容易得知 $S_T = S_W + S_B$ 。如果 S_W 是非奇异的，那么W满足

$$W_{opt} = \arg \max_W \frac{\|W^T S_B W\|}{\|W^T S_W W\|}, \quad (\text{A-8})$$

矩阵 W_{opt} 的每一列 $\{\mathbf{w}_i \in \mathbb{R}^{d'} | i = 1, 2, \dots, d'\}$ 是 S_B 和 S_W 的前m个最大的广义特征值 $\{\lambda_i | i = 1, 2, \dots, d'\}$ 对应的特征向量：

$$S_B \mathbf{w}_i = \lambda_i S_W \mathbf{w}_i, i = 1, 2, \dots, d'. \quad (\text{A-9})$$

但是，如果 S_W 是奇异的，需要给 S_W 先做PCA（Principal Components Analysis，主成分分析）降维，具体的方法是

$$\begin{aligned} W_{pca} &= \arg \max_W \|W^T S_T W\|, \\ W_{fld} &= \arg \max_W \frac{\|W^T W_{pca}^T S_B W_{pca} W\|}{\|W^T W_{pca}^T S_W W_{pca} W\|}, \\ W_{opt}^T &= W_{fld}^T W_{pca}^T. \end{aligned} \quad (\text{A-10})$$

A.3 分类器设计

在模式识别中，常用的分类器有许多种，比如最简单的最小欧氏距离

法，就是用待识别字符的特征向量和每个类别的均值进行比较，选出距离最近的类别。在汉字识别领域目前效果最好的是改进二次判别函数（Modified Quadratic Discriminant Function, MQDF）^[78]。对于MQDF来说，需要有训练集（Training Set）用来估计参数。训练集是一组已知类别标签的特征向量，设特征向量 \mathbf{x}_i ($i = 1, \dots, N_{tr}$)的类别标签为 y_i ($i = 1, \dots, N_{tr}$)，这里 N_{tr} 是训练集大小。下面先介绍二次判别函数，再介绍MQDF。

A.3.1 二次判别函数

假定每一类的特征向量都是从高斯分布抽样得到，即

$$p(\mathbf{x}|C_k) = \frac{1}{(2\pi)^{\frac{d}{2}} \sqrt{|\Sigma_k|}} e^{-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}-\boldsymbol{\mu}_k)} \quad (\text{A-11})$$

其中 d 是 \mathbf{x} 的维数， Σ_k 是第 k 类的协方差矩阵，

$$\Sigma_k = \frac{\sum_{\mathbf{x}_i \in C_k} (\mathbf{x}_i - \boldsymbol{\mu}_i)(\mathbf{x}_i - \boldsymbol{\mu}_i)^T}{N_k - 1}. \quad (\text{A-12})$$

要判别 \mathbf{x}_i 属于哪一类，需要找到 k 使得 $p(C_k|\mathbf{x}_i)$ 最大。由Bayes定理

$$p(C_k|\mathbf{x}_i) = \frac{p(\mathbf{x}_i|C_k)p(C_k)}{\sum_{i=1}^c p(\mathbf{x}_i|C_k)p(C_k)}, \quad k = 1, 2, \dots, c \quad (\text{A-13})$$

这里 $p(C_k)$ 是第 k 类的先验概率，可以通过 $p(C_k) = N_k/N$ 简单求得。

根据最大后验概率（Maximum a Posteriori, MAP）原理， \mathbf{x}_i 所属类别为

$$\begin{aligned} y_i &= \arg \max_k p(C_k|\mathbf{x}_i) \\ &= \arg \max_k p(\mathbf{x}_i|C_k)p(C_k) \\ &= \arg \max_k \log p(\mathbf{x}_i|C_k)p(C_k) \\ &= \arg \max_k \left[-\log((2\pi)^{\frac{d}{2}} \sqrt{|\Sigma_k|}) - \frac{1}{2}(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) + \log(p(C_k)) \right] \\ &= \arg \min_k \left[(\mathbf{x}_i - \boldsymbol{\mu}_k)^T \Sigma_k^{-1} (\mathbf{x}_i - \boldsymbol{\mu}_k) + \log(|\Sigma_k|) - 2 \log(p(C_k)) \right] \\ &= \arg \min_k Q_k(\mathbf{x}_i), \end{aligned} \quad (\text{A-14})$$

这里 $Q_k(\mathbf{x}_i)$ 就是二次判别函数（Quadratic Discriminant Function, QDF）， \mathbf{x}_i 所属类别是使得 $Q_k(\mathbf{x}_i)$ 最大的 k 。

A.3.2 MQDF分类器

上一节中的二次判别函数在高斯分布的假设下，充分考虑了样本的均值和类内方差，具有较好的分类能力。但是在实际应用中存在以下几个问题：

- 对参数估计敏感，随着训练样本数量 N_t 增大，错误率先下降后上升；
- 需要 $O(d^2)$ 的计算量和存储量，当d较大时非常耗时；
- 特征向量不一定服从高斯分布，这样精确的建模会导致偏差。

MQDF (Modified QDF) 旨在用近似的方法减小计算量，同时让模型对参数的敏感度降低，从而更加具有鲁棒性。为了说明近似的原理，首先需要改写式(A-14)。

根据矩阵的特征值分解，协方差矩阵 Σ_k 可以写为

$$\Sigma_k = \sum_{i=1}^d \lambda_i \varphi_i \varphi_i^T, \quad (\text{A-15})$$

这里 d 是特征向量维数， $\lambda_i (\lambda_i \geq \lambda_{i+1})$ 是特征值， φ_i 是对应的特征向量。然后QDF的表达式可以改写为

$$\begin{aligned} Q_j(\mathbf{x}) &= (\mathbf{x} - \boldsymbol{\mu}_j)^T \Sigma_j^{-1} (\mathbf{x} - \boldsymbol{\mu}_j) + \log(|\Sigma_j|) - 2 \log(p(C_j)) \\ &= \sum_{i=1}^d \frac{1}{\lambda_i} \left\{ \varphi_i^T (\mathbf{x} - \boldsymbol{\mu}_j) \right\}^2 + \log \prod_{i=1}^d \lambda_i, \end{aligned} \quad (\text{A-16})$$

为了便于讨论，假设所有类别都具有相同的先验概率，所以常数项 $\log(p(C_k))$ 被忽略了。

在近似时，MQDF选取前 $k (1 \leq k < d)$ 个特征值和特征向量保持和原来一致，在 $k+1$ 以后用一个公共的特征值 h^2 代替，所以式 (A-16) 被修改为

$$\begin{aligned} Q_j(\mathbf{x}) &= \sum_{i=1}^k \frac{1}{\lambda_i} \left\{ \varphi_i^T (\mathbf{x} - \boldsymbol{\mu}_j) \right\}^2 + \sum_{i=k+1}^d \frac{1}{h^2} \left\{ \varphi_i^T (\mathbf{x} - \boldsymbol{\mu}_j) \right\}^2 + \log \left(h^{2(d-k)} \prod_{i=1}^k \lambda_i \right) \\ &= \frac{1}{h^2} \left[\|\mathbf{x} - \boldsymbol{\mu}_j\|^2 - \sum_{i=1}^k \left(1 - \frac{h^2}{\lambda_i} \right) \left\{ \varphi_i^T (\mathbf{x} - \boldsymbol{\mu}_j) \right\}^2 \right] + \log \left(h^{2(d-k)} \prod_{i=1}^k \lambda_i \right), \end{aligned} \quad (\text{A-17})$$

其中第二步的变形时因为有

$$\sum_{i=1}^d \left\{ \varphi_i^T (\mathbf{x} - \boldsymbol{\mu}_j) \right\}^2 = \|\mathbf{x} - \boldsymbol{\mu}_j\|^2. \quad (\text{A-18})$$

在本文的研究中， h^2 用所有类的 λ_{k+1} 的均值代替。

附录 B 汉字BIG5编码以及存储格式

B.1 BIG5编码

现代汉语的简体字常用GB2312编码方式，而古籍汉字的编码是BIG5，它是一种繁体中文的编码方式，共收录13,060个汉字，广泛用于台湾、香港和澳门等地的电脑中。编码中用两个字节（二进制16位）来表示汉字，第一个字节叫“高位字节”，第二个叫“低位字节”，常用字部分从“一”（AA40）起，按笔画和部首顺序排序。高位字节是连续的，但是每个高位字节里的低位字节的范围只有0x40-0x7E和0xA1-0xFE。

B.2 PNT格式介绍

PNT格式是一种汉字样本保存格式。在本文的研究中，训练源分类器的繁体印刷体汉字即用这种格式保存。它可以存储灰度和二值两种字符图像，每个文件循环存有不同数量的字符，每个字符的格式如表B.1所示。

表 B.1 PNT的字符图像格式

含义	块大小	编码	高	宽	字符像素块
类型	uint16	uint16	uint8	uint8	uint8
长度	2 Bytes	2 Bytes	1 Byte	1 Byte	(wb ^① *高) Bytes

① 对于灰度格式的字符图像，wb=宽；对于二值格式，wb=[宽/8]

附录 C MATLAB编写.NET组件概述

MATLAB是一款数值计算软件，具有强大的计算能力、简洁的脚本语言和方便的调试工具，在算法设计中非常容易上手。本文中的在线古籍识别系统在搭建时，将字符切割和字符识别模块用MATLAB的脚本语言来描述，因此采用了C#和MATLAB混编的结构。下面将介绍具体过程。

首先打开MATLAB（本文中是MATLAB 2010b），在MATLAB的命令窗口中输入“deploytool”，弹出组件部署窗口（如图C.1(a)所示），然后填写项目名称和路径。在C#中，项目名称默认是命名空间，可以自己添加类及写好的.m文件作为类的方法（如图C.1(c)所示）。需要注意的是，在设置中实现选好.NET版本，如图C.1(b)所示。设置好后点击编译（如图C.1(d)所示），编译成功后会在目录下生成一个C#的dll文件（本文中是MatlabRecogTool.dll）。

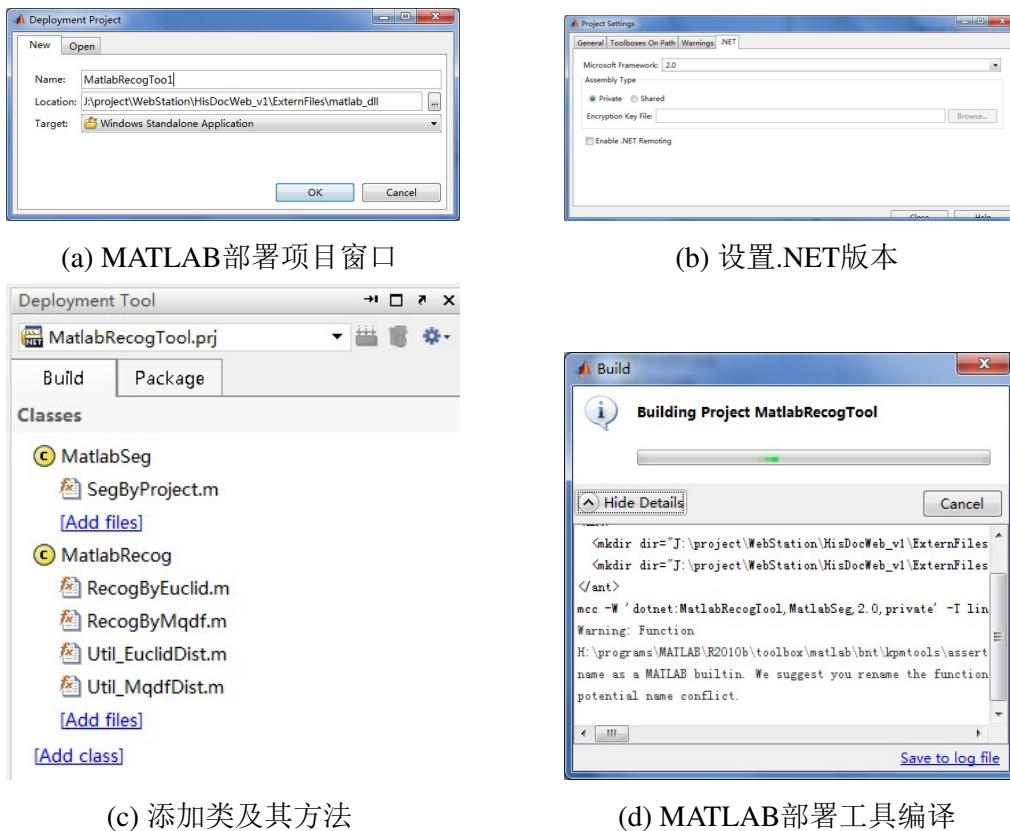


图 C.1 MATLAB .NET组件编译过程

其次打开C#编译环境（本文中用VS2012），在加载了网站的.cs源文件后，还需要添加引用，添加MATLAB软件中的MWArray.dll，再添加MATLAB编译生成

的MatlabRecogTool.dll（如图C.2所示）。

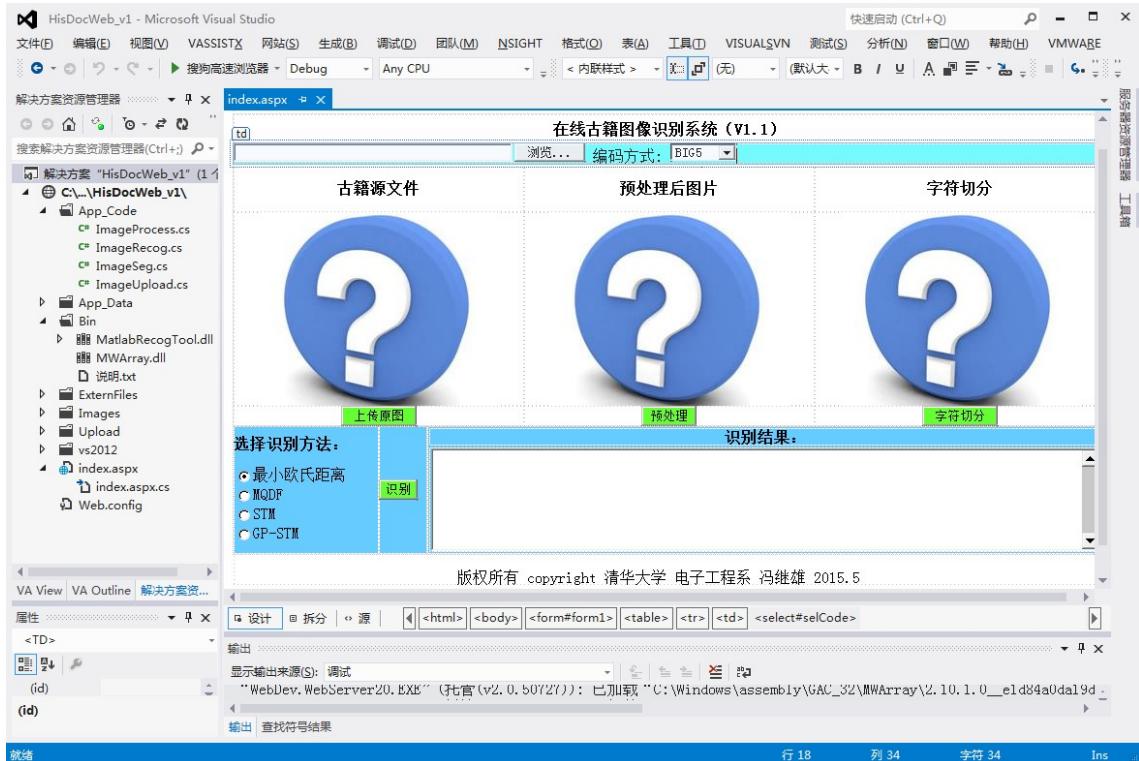


图 C.2 Visual Studio 2012项目界面

最后在需要调用的.cs文件中，使用

```
using MathWorks.MATLAB.NET.Arrays;
using MathWorks.MATLAB.NET.Utility;
using MatlabRecogTool;
```

这样就可以在文件中调用dll中的类了。需要注意的是，MATLAB编译生成的函数的输入和输出全都是MWArray类型，需要将C#中的int、double、string等做类型转换，具体可以参考《MATLAB Builder NE User's Guide》。

个人简历、在学期间发表的学术论文与研究成果

个人简历

1990年10月28日出生于陕西省榆林市榆阳区。

2009年9月考入浙江大学，在信息与电子工程学系学习信息与通信工程专业，2013年7月本科毕业并获得工学学士学位。

2013年9月免试进入清华大学电子工程系信息与通信工程专业，攻读硕士学位至今。

发表的学术论文

- [1] Feng J, Peng L, Lebourgeois F. Gaussian process style transfer mapping for historical Chinese character recognition[C]. IS&T/SPIE Electronic Imaging. International Society for Optics and Photonics, 2015: 94020D-94020D-12. (EI检索, doi:10.1117/12.2076119; <http://dx.doi.org/10.1117/12.2076119>)

获得奖项

- [1] 2010年10月 国家奖学金
- [2] 2011年10月 国家奖学金
- [3] 2012年10月 国家奖学金
- [4] 2013年6月 浙江省优秀毕业生，浙江大学优秀毕业生
- [5] 2014年10月 清华大学综合三等奖学金
- [6] 2015年2月 第22届国际文档识别与检索会议（DRR 2015）最佳学生论文奖