



中国科学院大学

University of Chinese Academy of Sciences

博士学位论文

指纹奇异点提取及小面积指纹识别算法研究

作者姓名: 秦进

指导教师: 郭田德 教授 中国科学院大学数学科学学院

韩丛英 副教授 中国科学院大学数学科学学院

学位类别: 理学博士

学科专业: 运筹学与控制论

培养单位: 中国科学院大学数学科学学院

2018 年 6 月

Research on Fingerprint Singular Point Extraction and Small-area

Fingerprint Recognition Algorithm

**A thesis submitted to the
University of Chinese Academy of Sciences
in partial fulfillment of the requirement
for the degree of
Doctor of Natural Science
in Operational Research and Cybernetics**

By

Qin Jin

Supervisor: Professor Guo Tiande

**School of Mathematical Sciences
University of Chinese Academy of Sciences**

June, 2018

中国科学院大学 学位论文原创性声明

本人郑重声明：所呈交的学位论文是本人在导师的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，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对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或致谢。本人完全意识到本声明的法律结果由本人承担。

作者签名： 
日 期： 2018.5.23

中国科学院大学 学位论文授权使用声明

本人完全了解并同意遵守中国科学院大学有关保存和使用学位论文的规定，即中国科学院大学有权保留送交学位论文的副本，允许该论文被查阅，可以按照学术研究公开原则和保护知识产权的原则公布该论文的全部或部分内容，可以采用影印、缩印或其他复制手段保存、汇编本学位论文。

涉密及延迟公开的学位论文在解密或延迟期后适用本声明。

作者签名： 	导师签名： 
日 期： 2018.5.23	日 期： 2018.5.23

摘 要

指纹作为一种重要的生物特征，在身份认证等领域一直发挥着特别重要的作用。本文针对指纹自动识别系统的关键技术进行了研究。

第二章提出了基于深度学习的奇异点检测方法，该方法通过变尺度的全卷积神经网络确定奇异点区域，建立概率估计模型确定了奇异点位置。该方法不依赖于指纹方向场，适应性优于基于方向场的方法。实验结果表明该方法在质量差的指纹上取得更优异的准确率。

第三章针对警用指纹设备和民用指纹设备不同，分析对比了两种设备上的指纹识别算法。针对民用指纹采集设备的特点分析了小面积指纹识别算法存在的问题，设计的难度。并在第四、五章中给出了完整的小面积指纹识别算法。

第四章提出了基于脊线特征与改进的广义哈夫变换的小面积指纹识别算法。该方法将脊线作为小面积指纹的主要特征，给出了脊线的表示方法。在提取出脊线特征的基础之上，设计并使用了基于改进的广义哈夫变换算法进行两幅小面积指纹的粗对齐。改进的广义哈夫变换相对于传统的广义哈夫变换在速度上提高了近 20 倍，同时内存消耗也仅为后者的 5.68%，这表明本文的算法可以满足现在实际应用中的实时性要求。在对齐的基础上本文提出通过机器学习的方法进行综合评估，给出最终的匹配分数，从而确定是否为同一枚指纹。

第五章提出了基于改进的极坐标傅立叶变换和深度卷积网络的小面积指纹识别算法。首先，通过改进的极坐标傅立叶变换给出两幅小面积指纹图像的多种可能的对齐候选；然后，在这些候选的基础上，提取出两幅小面积指纹图像的重合区域；最后，使用深度卷积网络对重合区域进行匹配相似度估计，选择匹配的最高分作为最终得分。该方法不需要提取任何人工经验的特征，对指纹质量无任何要求，识别性能优异。

关键词： 指纹奇异点，深度学习，卷积神经网络，小面积指纹，指纹识别，机器学习，广义哈夫变换，极坐标傅立叶变换

Abstract

As an important biological feature, fingerprinting has been playing a particularly important role in identity authentication and other fields. This paper studies the key technologies of fingerprint automatic identification system.

The second chapter put forward the singular points detection method based on the deep learning. Our method calculates the probability of singular point values on each pixel by the convolution neural network, and then the singular point position is determined through a probability estimation model. This method has a more extensive adaptability since it does not depend on the direction field. Experimental results show that the accuracy of this method is better than before.

The third chapter analyses the fingerprint identification algorithms on two kinds of devices for the differences between police fingerprint devices and civilian fingerprint devices. According to the characteristics of civilian fingerprint acquisition equipment, the problems of the small area fingerprint identification algorithm and the difficulty of design are analysed. And in the fourth and fifth chapters, a complete small area fingerprint identification algorithm is given.

The fourth chapter put forward a partial fingerprint identification algorithm based on ridge line feature and the modified generalized Hough Transform. A new concept has been put forward that places ridge as the main feature of the partial fingerprint and we obtain the representation method of ridge. After extracted the characteristic of ridge, we use the generalized Hough Transform algorithm to make coarse alignment of two partial fingerprints. Compared with the traditional Hough Transform, the efficiency is raised to 20 times, at the same time memory consumption is only 5.68% of the latter. It shows that this algorithm can meet the real-time requirements of practical application. In this paper, the final matching scores are obtained on the basis of comprehensive evaluation by using the method of machine learning, to determine whether it is the same fingerprints.

The fifth chapter propose an algorithm based on improved polar Fourier transform and deep convolutional neural network. First, we coarsely align two fingerprints by algorithm based on phase-only correlation. Then, the overlap region will be extracted.

Finally, we employ a deep convolutional neural network (CNN) with spatial pyramid pooling to verify whether these two fingerprint regions come from the same finger. Our method is independent from any particular hand-crafted features and fingerprint quality and has better performance.

Keywords: Fingerprint Singular Points, Deep Learning, Convolutional Neural Network, Small-size Fingerprint, Fingerprint Recognition, Machine Learning, Generalised Hough Transform, Polar Fourier Transform

目 录

第 1 章 绪论	1
1.1 研究背景	1
1.2 自动指纹识别系统	4
1.3 指纹图像获取	5
1.4 指纹图像的压缩存储	7
1.5 指纹的特征提取	9
1.5.1 指纹的特征	9
1.5.2 指纹图像的前背景分割	13
1.5.3 指纹图像的方向场估计	15
1.5.4 指纹图像的奇异点提取	17
1.5.5 指纹图像的增强	19
1.5.6 二值化与细化	21
1.5.7 指纹细节节点的提取	23
1.6 指纹的特征匹配	23
1.7 本文的主要研究内容	26
第 2 章 基于多尺度全卷积神经网络的指纹奇异点检测算法 ...	29
2.1 引言	29
2.2 深度学习中的卷积神经网络介绍	30
2.2.1 神经网络的基本原理	30
2.2.2 卷积神经网络的基本原理	31
2.2.3 全卷积神经网络	34
2.3 检测奇异点区域的多尺度全卷积神经网络算法	36
2.3.1 卷积神经网络的构建	37
2.3.2 奇异点区域的检测	39
2.3.3 多尺度	40
2.4 定位奇异点位置的概率预测模型	42
2.4.1 步骤 1: 尽可能地去掉虚假预测	42
2.4.2 步骤 2: 确定三角点的位置	43
2.4.3 步骤 3: 确定中心点的位置	43
2.5 实验结果及分析	44
2.6 本章结论	46

第 3 章 小面积指纹识别系统	47
3.1 引言	47
3.2 小面积指纹的匹配算法	51
3.3 小面积指纹识别系统	53
3.3.1 注册策略	54
3.3.2 自学习策略	57
3.4 可能遇到的问题及解决方案	60
3.5 本章结论	61
第 4 章 基于脊线特征与改进的广义哈夫变换的小面积指纹识别 算法	63
4.1 引言	63
4.2 基于脊线与广义哈夫变换的指纹匹配算法	63
4.2.1 指纹脊线提取	64
4.2.2 基于广义哈夫变换的对齐	65
4.2.3 相似度计算	74
4.3 实验结果及分析	76
4.4 本章结论	79
第 5 章 基于改进的极坐标傅立叶变换和深度卷积网络的小面积 指纹识别算法	81
5.1 引言	81
5.2 基于极坐标傅立叶变换的指纹对齐	81
5.3 基于深度卷积网络的指纹相似度估计	84
5.3.1 网络结构	86
5.3.2 训练样本	86
5.3.3 损失函数	87
5.4 实验结果及分析	87
5.5 本章结论	91
第 6 章 总结与展望	93
参考文献	97
作者简历及攻读学位期间发表的学术论文与研究成果	105
致谢	107

图形列表

1.1 几种常见的生物特征	2
1.2 2020 年全球生物特征识别市场规模预测 (亿美元)	3
1.3 2015 年全球生物特征识别市场份额	4
1.4 NIST-14 油墨纳印获得的指纹	5
1.5 (a) 光学传感器采集的指纹图像, (b) 电容式传感器采集的指纹图像	6
1.6 WSQ 压缩算法的效果示例图: (a) 原图; (b) 10 倍; (c) 20 倍; (d) 30 倍	8
1.7 指纹的脊线和谷线	9
1.8 指纹奇异点: 圆为中心点; 三角形为三角点	10
1.9 指纹的五种纹型示例图: (a) 斗; (b) 斗 (双旋); (c) 弓; (d) 帐弓; (e) 左旋; (f) 右旋	10
1.10 指纹细节点: (a) 分叉点; (b) 端点	11
1.11 同一根手指的同一个区域的按压两次采集得到的指纹图像, 在 (a) 中, 该位置为分叉点, 但是 (b) 中为端点	12
1.12 指纹的毛孔特征	12
1.13 指纹分割示例图: (a) 原图, (b) 分割得到的前景区域	13
1.14 文献 (Zhang et al., 2013) 方法对于现场指纹的分割结果: (a) 原图, (b) 分割结果	14
1.15 指纹方向场示例图, 图片来自 (Shao et al., 2014)	15
1.16 难以求取方向场的指纹图像	17
1.17 各种质量的指纹图像: (a) 高质量的指纹图像, 脊线和谷线区别明显; (b) 脊线不清晰, 连续性差; (c) 脊线和谷线分界线不明显, 有多个糊成一片的区域; (d) 低质量指纹, 噪声大, 脊线不清晰	20
1.18 Gabor 滤波对指纹图像增强的结果: (a) 原图; (b) 增强后的结果	21
1.19 (a): 原图; (b): 对 (a) 做指纹增强后的结果; (c): 对 (b) 做二值化的结果; (d): 对 (c) 做细化的结果	22
1.20 8 邻域局部模式示例, 正方块为中心细节点, 圆点为选取得到的邻点细节点, 叉点为其余的细节点	25
1.21 指纹图像的所用 8 邻域局部模式组成的星状图: (a) 原图; (b) 细节点 8 邻域局部模式星状图	25
2.1 一个 5 层的神经网络示例图	30
2.2 LeNet-5 模型 (LeCun et al., 1998)	31

2.3	全连接层与卷积层的对比	32
2.4	一个简单的最大值池化事例	33
2.5	常见的激活函数 (a) sigmoid 函数 (b) tanh 函数 (c) ReLU 函数 (d) ELU 函数 (e) PReLU 函数	35
2.6	全卷积神经网络用于像素级语义分割, 图片来至于文献 (Long et al., 2015)	36
2.7	(a) 来至于 NIST-4 中的两幅指纹图像, 其奇异点区域的纹线信息 缺失。(b) 使用 FCN-SingleScale80 对 (a) 中两幅指纹图像的奇异 点检测结果。(c) 使用 FCN-SingleScale160 对 (a) 中两幅指纹图像 的奇异点检测结果。	38
2.8	CNN-C 的网络结构示例图	39
2.9	使用全卷积神经网络对指纹图像中的像素点进行是否为奇异点的 判定, 如果判定该像素为奇异点, 则对该像素赋值 0 (即标为黑色)。	40
2.10	使用全卷积神经网络对指纹图像中的像素点进行是否为奇异点的 判定, 如果判定该像素为奇异点, 则对该像素赋值 0 (即标为黑色)	41
2.11	以从上到下的方向来画出 CP 直径上所有点的概率值 (a) 存在两个 中心点的指纹图像画出的曲线 (b) 存在一个中心点的指纹图像 画出的曲线	44
3.1	小面积指纹识别系统的实际应用案例: (a) 智能手机; (b) 指纹 U 盘; (c) 指纹门锁; (d) 指纹箱包	47
3.2	在 500 Dpi 下各种尺寸的指纹图像: (a) 512×512 (b) 300×300 (c) 160×160 (d) 96×96	49
3.3	一对 96×96 的小面积指纹图像匹配示意图, 其中红色点为重合区 域的细节点, 蓝色点为非重合区域细节点。(a) 原图; (b) 特征提 取结果; (c) 正确匹配结果	50
3.4	脊线表示点示例图, 其中圆点代表细节点, 方块点代表细节点生成 的脊线表示点	51
3.5	汗孔, 短棒等三级特征示例图	52
3.6	传统指纹识别系统注册流程示例图	54
3.7	小面积指纹识别系统注册流程示例图	55
3.8	用一根手指按压多次采集的小面积指纹图像	56
3.9	拼接注册过程中录入的小面积指纹示例图	57
3.10	自学习策略流程示例图	59
4.1	本章算法的一个框架示例图	63
4.2	本章脊线特征提取流程的示例图	64

4.3	在模版坐标系中, 广义哈夫变换所用的几何信息示例图	71
4.4	小面积指纹脊线点集错周期匹配事例	73
4.5	PFDB1690 中的 6 幅指纹图像, 第一行的 3 幅来源于一根手指, 第二行的 3 幅来源于另一根手指	76
4.6	不同参数设置下的 MGHT 在小面积指纹图像库 PFDB1690 上的 ROC 曲线	78
4.7	各算法在小面积指纹图像库 PFDB1690 上的 ROC 曲线	78
4.8	使用机器学习打分的 MGHT 与使用启发式策略打分的 MGHT 在小面积指纹图像库 PFDB1690 上的 ROC 曲线	79
5.1	基于改进的极坐标傅立叶变换的小面积指纹对齐算法	83
5.2	难以区分的假匹配和真匹配实例	85
5.3	重叠区域相似度估计示例图	85
5.4	深度卷积网络的结构示意图	86
5.5	训练样本实例	89
5.6	各算法在测试库上的 ROC 曲线	90

表格列表

1.1	常用生物识别技术的比较.....	3
1.2	常用生物识别技术的比较.....	7
2.1	各算法在 NIST-4 上的测试结果	45
2.2	各算法在 FVC2002 DB1 上的测试结果	46
4.1	广义哈夫变换所使用的 R 表	72
4.2	12 维的匹配结果信息.....	74
4.3	各算法在小面积指纹图像库 PFDB1690 上的测试结果	77
5.1	各算法在测试集上的 EER	88

第1章 绪论

1.1 研究背景

近年来,中国移动互联网发展迅速,各行各业都深受其影响。随着社会信息化的不断发展,个人身份认证和识别的需求日益剧增。准确高效地鉴别和认证每个社会成员的身份可以带来巨大的社会价值,例如现在的手机指纹支付、机场海关的人脸识别等大大降低了人力成本的同时又提升了工作效率,促进了社会的发展。但是,当前的生物识别也存在着诸多挑战,例如2018年开年国内有某指纹厂商爆料,宣称用“透明胶带+导电笔,秒破安卓和苹果的指纹识别,甚至于橘子皮都能打开”(今日头条,2018),这个问题正是由于不规范、不成熟的生物识别技术的使用导致的。错误的身份识别结果将会导致财产损失、隐私泄漏等严重的后果。如何准确的进行身份认证的同时保障人们的合法权益,并有效地应对各种违法犯罪的攻击,已经是成为一个重要的研究课题。

传统的身份认证主要是指基于物件(如钥匙、U盾、磁卡)或基于知识(如密码、口令、暗号)的认证方式。基于物件的认证方式存在着物件必须随身携带,还容易丢失、被盗、伪造等缺点。例如在一些重要的场所,犯罪分子通过伪造证件即可进入并进行犯罪活动。基于知识的认证方式存在着知识容易忘记、泄漏等缺点。例如一个人在ATM机上取款时,一旁的人只要观察他输入密码的按压动作,就能轻易获得其密码,从而可进行盗窃犯罪。因此,传统的身份认证方法存在着许多安全问题,给人们的生活带来了许多不必要的麻烦。寻找一种更加方便、更加安全的身份认证方式成为一个需要解决的重要问题。

基于生物特征的身份认证方法的出现解决了传统身份认证方法的诸多问题。基于生物特征的身份识别技术以生物特征为基础。生物特征分为生理特征和行为特征。每种生物都有其生物特征,这些特征安全性高、不会丢失、不会忘记,也不容易伪造和盗用,且使用方便。生理特征是先天的特征。对人而言,是指对人体某部分进行测量所获得的数据,例如掌纹、指纹、声纹、人脸、DNA、静脉和虹膜等,图1.1中展示几种常见的生物特征;行为特征是后天形成的特征,对人而言,行为特征是对个人习惯的度量,例如签名、走路的步态等。文献(Jain et al., 2009)中给出了一种生物特征用于身份认证的必要条件:(1)普遍性(Universality),指每个人都是否具有这种特征;(2)唯一性(Distinctiveness),指不存在两个人拥有相同的这种生物特征;(3)永久性(Permanence),指这种生物特征应该是终身

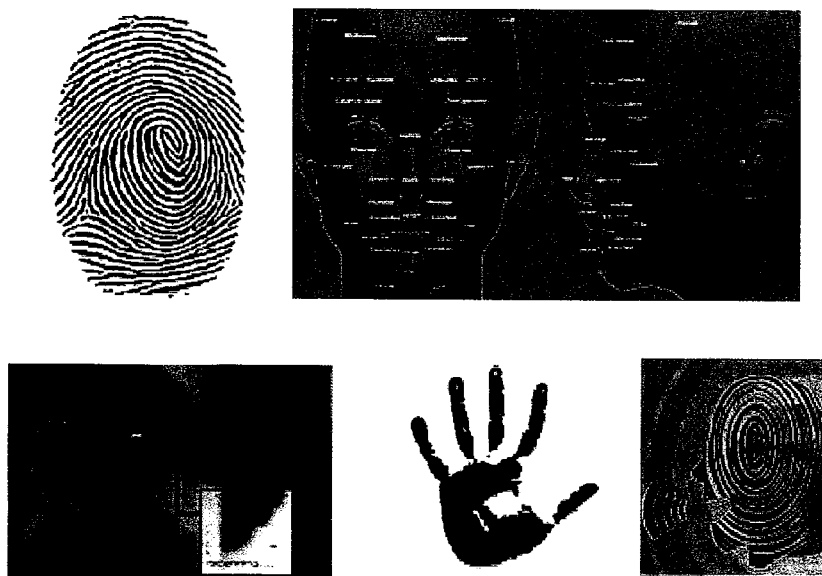


图 1.1 几种常见的生物特征

保持不变的；(4) 可采集性 (Collectability)，指这种生物特征是否便于采集获取；(5) 性能 (Performance)，指用这种生物特征进行识别时的效果如何；(6) 可接受性 (Acceptability)，指使用者是否愿意接受用这种生物特征进行身份认证；(7) 安全性 (Circumvention)，指这种生物特征在用于身份认证时是否有很强的防伪功能。表1.1列出了几种常用的生物特征在这七个评价指标下的比较。从中可以看出，指纹是整体性能最好的生物特征之一，是用于身份认证比较理想的生物特征。图1.2据国际生物组织 (IBG(International Biometric Group)) 统计，在 2015 年对未来 2020 年生物识别技术的全球市场预测。基于生物特征的身份认证方式的市场收入在未来几年还会大幅度的上升。图 1.3是 IBG 在 2015 年对各种生物特征用于身份认证的市场份额统计报告，其中指纹的市场占有率高达 58%，可见指纹识别技术已经有越来越广泛的应用。目前，中国新一代的身份证已经添加了指纹信息，后续关于如何使用身份证上的指纹信息势必会将指纹识别的应用推向一个新的高度。

表 1.1 常用生物识别技术的比较

生物特征	普遍性	唯一性	永久性	可采集性	性能	可接受性	安全性
指纹	中	高	高	中	高	中	中
DNA	高	高	高	低	高	低	高
虹膜	高	高	高	中	高	低	低
签名	低	低	低	高	低	高	高
人脸	高	低	中	高	低	高	中
声音	中	低	低	中	低	高	中
手静脉	中	中	中	中	中	中	低

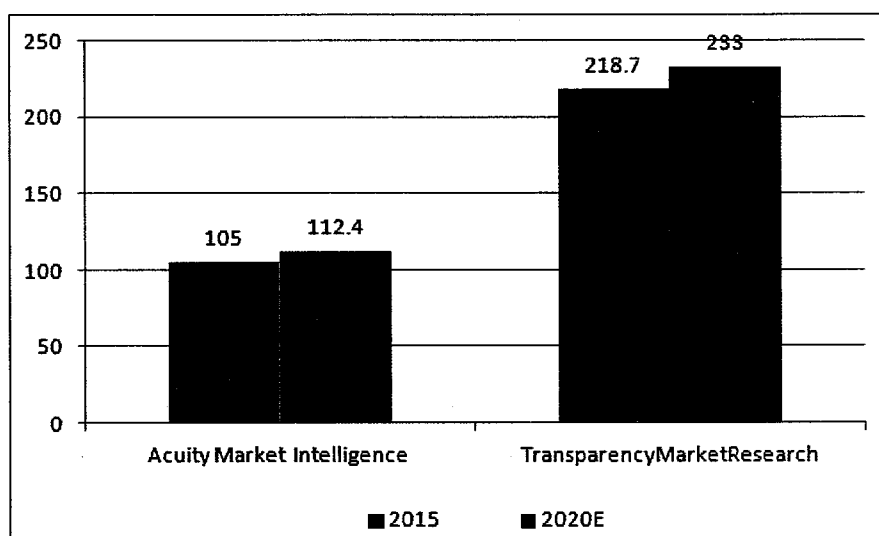


图 1.2 2020 年全球生物特征识别市场规模预测（亿美元）

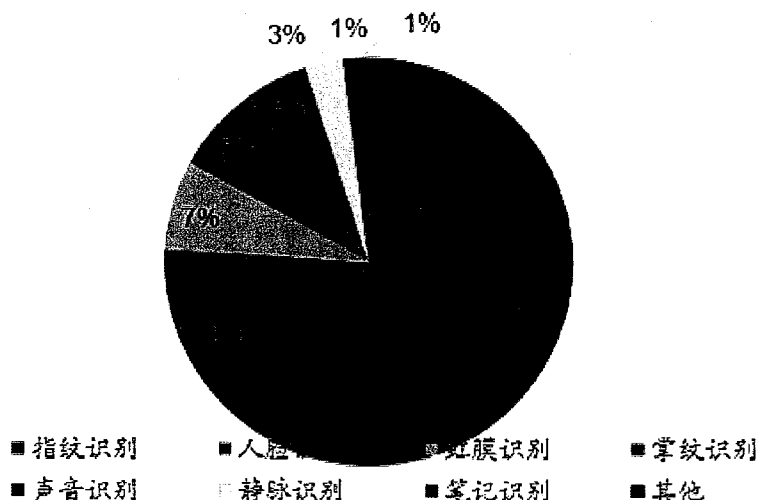


图 1.3 2015 年全球生物特征识别市场份额

1.2 自动指纹识别系统

指纹识别系统是一个典型的模式识别系统，其使用指纹信息进行个人认证和不同人之间的区分，可以避开密码、信用卡丢失和盗用的风险，并且方便安全。自动指纹识别系统有三大模块：图像采集，特征提取和指纹匹配。图像采集是用扫描的方式将指纹的凸/凹槽结构转换成数字图像；特征提取是指从数字化的指纹图像中获得指纹特征用于匹配；指纹匹配是将两幅已获得指纹特征的指纹图像进行比较，从而判定两幅图像是否来自同根手指。

自动指纹识别系统通常分为两个阶段：注册阶段和识别阶段。在注册阶段，用户将自己的用户名和指纹图像输入到系统中，在系统获得用户指纹图像的同时，自动指纹识别系统将自动从指纹图像中提取得到指纹特征，并将指纹特征与用户名关联在一起一并保存到系统数据库中。在识别阶段，指纹特征作为模板被提取出来，与输入指纹进行匹配。指纹识别分为身份认证（1:1 匹配）和身份识别（1:N 匹配）两种。对于身份认证匹配，需要用户同时输入用户名和指纹，系统的特征提取模块将从输入的指纹中提取出指纹特征，然后从数据库中找到在注册阶段该用户存储在数据库中的指纹，最后将输入指纹与该指纹进行匹配，如果匹配结果返回通过，说明当前认证通过，反之认证被拒；对于身份识别匹配，用户则只需要输入指纹，系统的特征提取模块从输入指纹提取指纹特征，然后系统将根据某些策略，不断地从指纹库中提取特征与输入指纹进行匹配，如果在数据库中找到与输入指纹相匹配的指纹，则系统的识别通过，并输出该指纹的用户名，否则系统识别为不通过。

如何评价一套自动指纹识别系统的好坏主要看三个评价指标：正确性指标、时间指标和空间指标。正确性指标包括错误接受率 (FAR, False Accept Rate)、错误拒绝率 (FRR, False Reject Rate) 和相等错误率 (EER, Equal Error Rate)。错误接受率是指纹识别系统将两幅来自不同手指的指纹图像判断为来自同一跟手指的概率；错误拒绝率是指纹识别系统将来自同一手指的两幅指纹图像判断为来自不同手指的概率；相等错误率表示当错误拒绝率同错误接受率相等时，错误拒绝率或错误接受率所取的值。时间指标是对指纹识别系统的识别速度的评价，包括注册时间和匹配时间。注册时间是指在注册阶段注册一枚指纹进入模板数据库所需的时间；匹配时间是指匹配两枚指纹所需的时间。空间指标包括分配内存大小和模板特征大小，分配内存大小是指自动指纹识别系统正常工作时所分配分配的内存大小，模板特征大小是指获得的指纹特征所占的磁盘空间大小。

1.3 指纹图像获取

当前指纹图像的采集方式主要分为两种：脱机扫描 (off-line) 和活体扫描 (live-scan)。

脱机扫描（油墨纳印技术）在法律和刑事侦察领域应用广泛，是指对油墨纳印在白纸上的指纹进行采集。油墨纳印时，将黑墨水均匀地涂抹在手指上，然后将手指按压在指纹卡上以获得指纹，接着用扫描仪将指纹卡上的指纹扫描转化为数字图像。相比于将手指平按在扫描仪上获得指纹，油墨纳印获得的指纹将包含更多的信息，但是如果油墨过重，将导致部分区域呈模糊状，而如果油墨过轻，则会丢失指纹某些区域的信息。图1.4所示为 NIST-14 油墨纳印获得的指纹 (Watson, 1993)。



图 1.4 NIST-14 油墨纳印获得的指纹

活体扫描是指将手指直接放在传感器采集仪上进行扫描并得到指纹图像,因此,对于活体扫描来讲,最重要的部分为传感器。按照工作机制的不同,传感器可分为光学传感器、电容式传感器和超声波传感器。光学传感器的发展历史悠久,已经有 30 年的历史。实现的原理也最简单。它主要利用光的折摄和反射原理,将手指放在光学镜片上,手指在内置光源照射下,光从底部射向三棱镜,并经棱镜射出,射出的光线在手指表面指纹凹凸不平的线纹上折射的角度及反射回去的光线明暗就会不一样。用棱镜将其投射在电荷耦合器件 CMOS 或者 CCD 上,进而形成脊线(指纹图像中具有一定宽度和走向的纹线)呈黑色、谷线(纹线之间的凹陷部分)呈白色的数字化的、可被指纹设备算法处理的多灰度指纹图像。电容式传感器是目前手机及便携式终端设备使用最多的传感器。其原理是将电容感测器整合于一块芯片中,当指纹按压芯片表面时,内部电容感测器会根据指纹波峰与波谷而产生电荷差,从而形成指纹图像。超声波传感器的原理与探测海底物质的声纳类似,靠反射波的时间差探知脊和谷的距离差,根据这个距离差绘制出指纹图像。超声波传感器成像质量最好,但由于成本太高,其实际应用面临很大挑战。表1.2对以上三种指纹活体扫描传感器的性能进行了比较。图1.5展示了光学和电容式传感器采集得到的指纹图像。



图 1.5 (a) 光学传感器采集的指纹图像, (b) 电容式传感器采集的指纹图像

指纹识别研究需要大量的指纹图像,而无论是脱机扫描还是活体扫描都需要大量的人力成本和时间成本,为了解决这个问题,一些学者尝试用数学模型自动生成指纹图像。提出了一些基于奇异点的指纹图像自动生成方法 (Cappelli et al., 2000, 2002, 2004)。

表 1.2 常用生物识别技术的比较

传感器类型	成像质量	体积	成本	耗电	耐用程度
光学传感器	干手指成像差；汗多的手指成像模糊	大	低	多	高
电容式传感器	干手指成像好；汗多的手指难以成像	小	高	少	低
超声波传感器	非常好	中	很高	多	中

1.4 指纹图像的压缩存储

随着自动指纹识别系统的应用发展，指纹系统数据库的规模也越来越大。早在 1995 年，FBI 的指纹卡存档就已经达到了 2 亿张 (Jain et al., 2009)，并且每天还以 3 万到 5 万张的速度在不断增长。在当时，一幅指纹图像的分辨率大小是 768×768 ，灰度级为 256。存储 2 亿的指纹数据就需要 1072TB 的空间。为了降低存储空间，就必须对指纹图像进行压缩。传统的图像压缩分为有损压缩和无损压缩。无损压缩能保留图像的完整信息，但压缩比率有限。为了提高压缩效率，大幅度降低存储空间，指纹图像采用了有损压缩的方法。在保证高压缩比的情况下，指纹有损压缩还不能丢失过多的指纹信息，因此，指纹有损压缩的研究也成为一个重要课题。目前指纹压缩最常用的方法是基于小波尺度量化的方法 (WSQ, Wavelet Scalar Quantization)(Hopper, 1994; Hopper et al., 1993; Kasaei et al., 1997; Deriche et al., 1999)。WSQ 是美国联邦调查局 (FBI) 提出的对分辨率为 500dpi 指纹图像的标准压缩算法，如图 1.6 所示，是 WSQ 算法的压缩示例图。除此之外，文献 (Shao et al., 2014) 提出的基于稀疏表示的指纹图像压缩方法也有很好的压缩效果，尤其是在高倍数压缩情况下，该方法能得到较高的峰值信噪比。

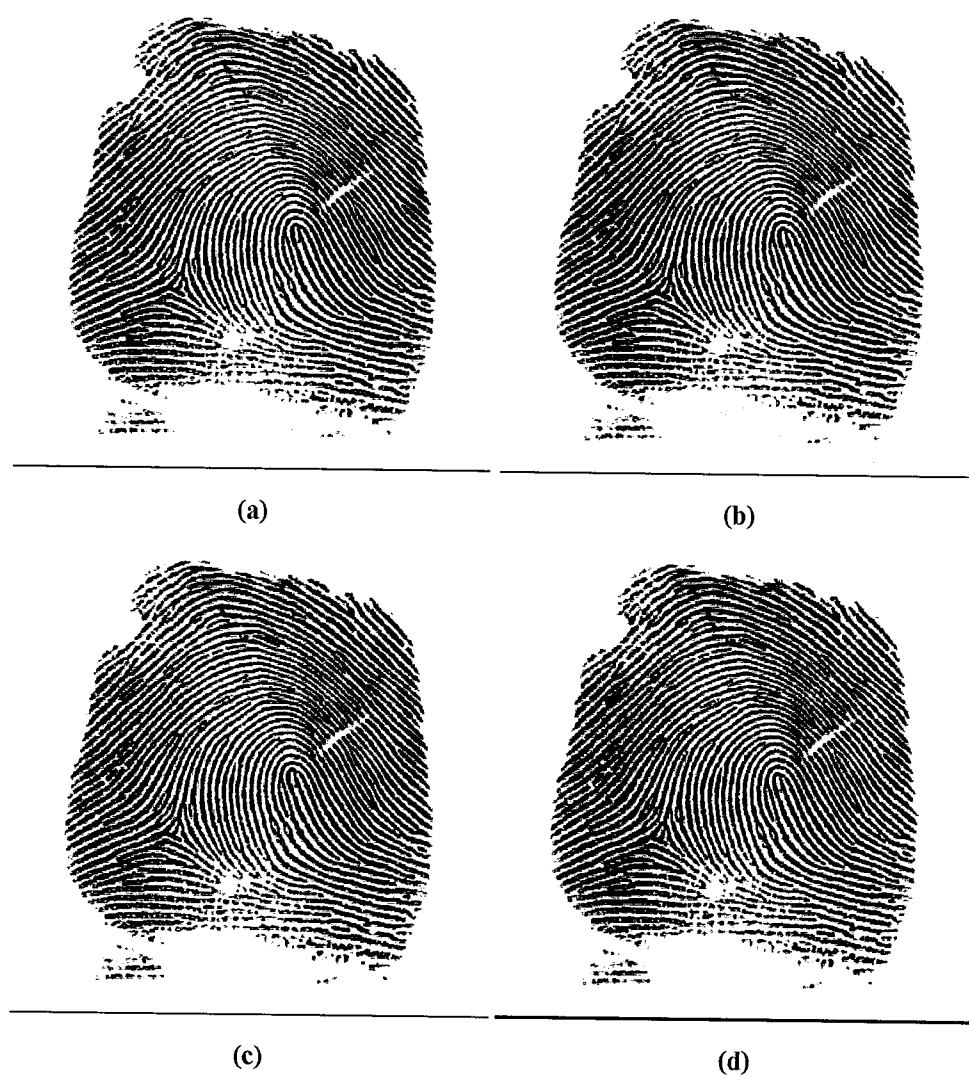


图 1.6 WSQ 压缩算法的效果示例图：(a) 原图；(b) 10 倍；(c) 20 倍；(d) 30 倍

1.5 指纹的特征提取

1.5.1 指纹的特征

指纹是人手指末端指腹上由凹凸的皮肤形成的纹路，指纹能使手在接触物体时增加摩擦力，从而更容易发力及抓紧物件，它是人类进化过程中自然形成的。指纹图像是指纹的复制，一般由交替出现的脊线和谷线组成，如图1.7所示，脊线是黑色部分，谷线是白色部分。脊线的宽度一般在 100um 到 300um 之间，据统计，脊线/谷线的周期大约为 500um(Stosz et al., 1994)。指纹脊线之间的拓扑结构保持着终生不变性，它是由真皮层的结构直接决定的，因而当手指表面磨损、烧伤之后，经过一段时间长出来的新皮肤依然能维持原来的拓扑结构。

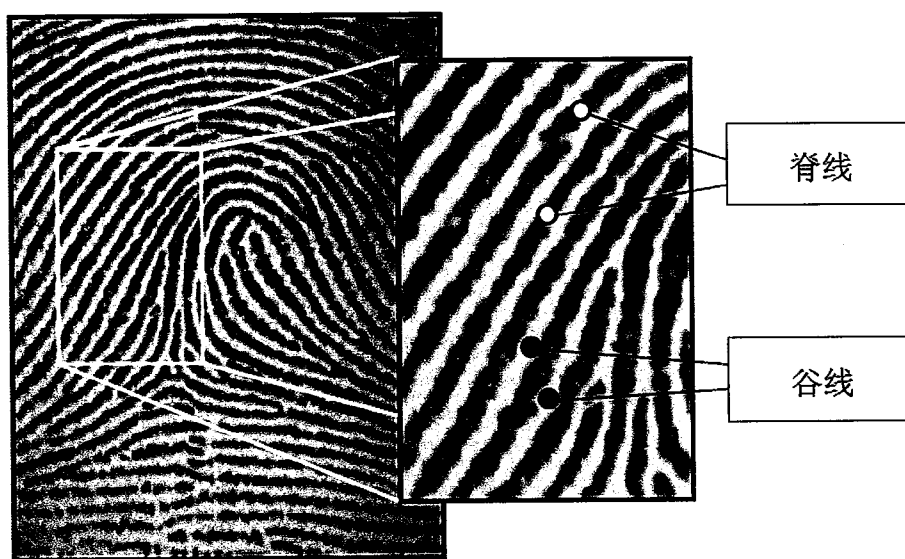


图 1.7 指纹的脊线和谷线

指纹的有些区域与其他区域有着明显的不同，这些区域有着非常高的曲率，脊线的方向变化很大。这些区域被称为奇异点区域，方向变化最剧烈的那一点被称为奇异点。奇异点因其模式的不同分为中心点 (Core) 和三角点 (Delta) 两种，如图1.8所示。中心点被认为是指纹最里端脊线最顶端的点 (Henry, 1905)，但是这一说法有争议；三角点被认为是指纹图像中三角形纹路区域的中间点，离该点最近的三条指纹脊线形成一个近似的三角形。奇异点可用于指纹分类、和作为指纹匹配的参考点 (Ratha et al., 1996; Zhang et al., 2004; Chan et al., 2004b)，从而使得系统大大缩减了在指纹大库中的搜索范围，大大降低了匹配时间。通过奇异点的个数、位置以及类别，可以将指纹分为五种纹型：弓、帐弓、左旋、右旋和斗，据统计，它们在人群中分别占的比例为 3.7%，2.9%，33.8%，31.7%，27.9%(Wilson

et al., 1994)。图中给出了 5 种纹型的示例图。



图 1.8 指纹奇异点：圆为中心点；三角形为三角点

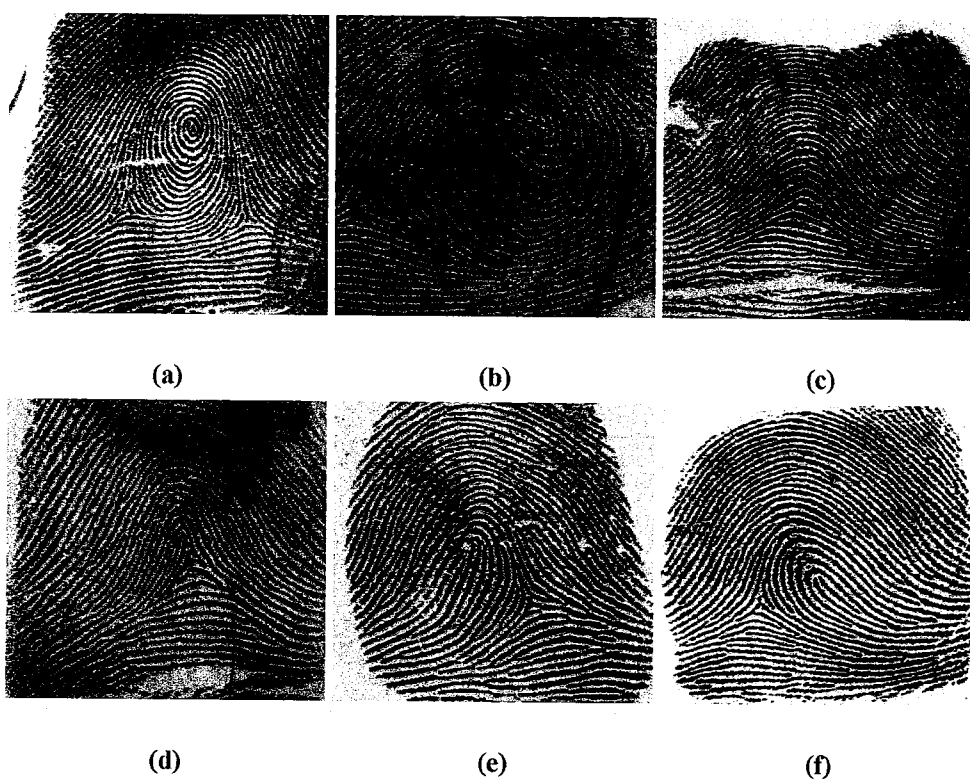


图 1.9 指纹的五种纹型示例图：(a) 斗；(b) 斗（双旋）；(c) 弓；(d) 帐弓；(e) 左旋；(f) 右旋

奇异点可看作是指纹的一种全局特征点，除此之外，指纹还有一种非常重要的局部特征点，称为细节点。指纹的脊线并不是连续不断的，在一些地方可能会

终止，这样的点称为端点，在另外一些地方，两条指纹脊线会合并成一条脊线，这样的点称为分叉点。端点和分叉点统称为细节点，图1.10给出了细节点的示例图。在有些情况，一根手指经过几次采集得到的指纹图像在同一个位置有可能出现细节点类型的不同，这通常是由于手指按压力度和角度的不同引起的。如图1.11所示，端点在某些情况可能会以分叉点的形式出现，反之分叉点也有可能以端点的形式出现，这被称为指纹细节点的二元性。由于细节点的二元性，通常在指纹匹配中不将细节点的类型作为一个参考指标，目前绝大多数指纹自动识别系统只记录细节点的位置坐标，以及细节点的方向（不同的算法对细节点的方向定义不同，一般为细节点所在脊线处的沿脊线方向）。

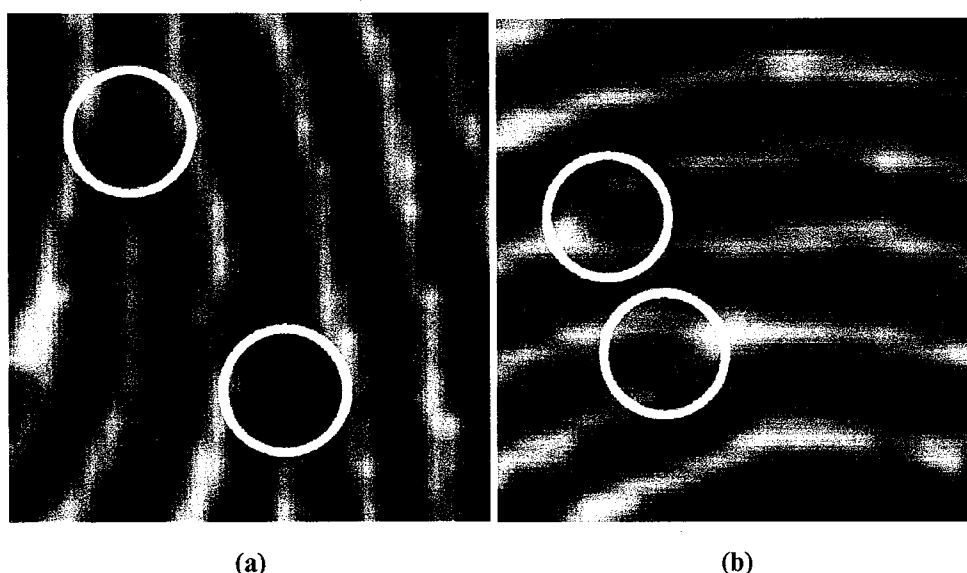


图 1.10 指纹细节点：(a) 分叉点；(b) 端点

在高分辨率（大于 1000dpi 以上）的指纹图像中，可以清楚的看到指纹上的毛孔，如图1.12所示。毛孔的半径大小不等，一般在 60um 到 250um 之间。虽然毛孔的数量较多，位置明显，且具有不变性，但是却并不实用。目前为止，主流的指纹识别系统几乎都没有使用毛孔信息来进行指纹识别。这是因为高分辨率的指纹图像太难获取，犯罪现场一般得到指纹图像质量很差，而民用指纹市场中如果使用高分辨率的指纹采集设备将大大提高成本，得不偿失。因此无论是警用系统还是民用系统目前都没有考虑指纹的毛孔信息。但是不少学者还是将毛孔作为指纹特征来进行指纹识别的研究 (Chen et al., 2007; Zhao et al., 2010a,b; Malathi et al., 2011; Conti et al., 2013)。

将指纹的特征按照所在尺度分级可分为三个级别的特征，奇异点属于一级特征，细节点属于二级特征，指纹的毛孔属于三级特征。指纹的这三级特征在人

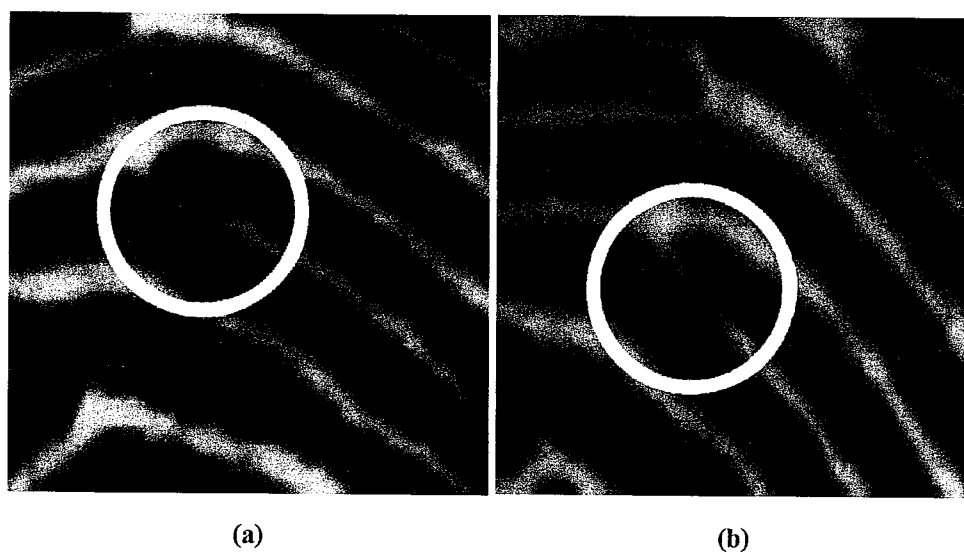


图 1.11 同一根手指的同一个区域的按压两次采集得到的指纹图像，在 (a) 中，该位置为分叉点，但是 (b) 中为端点

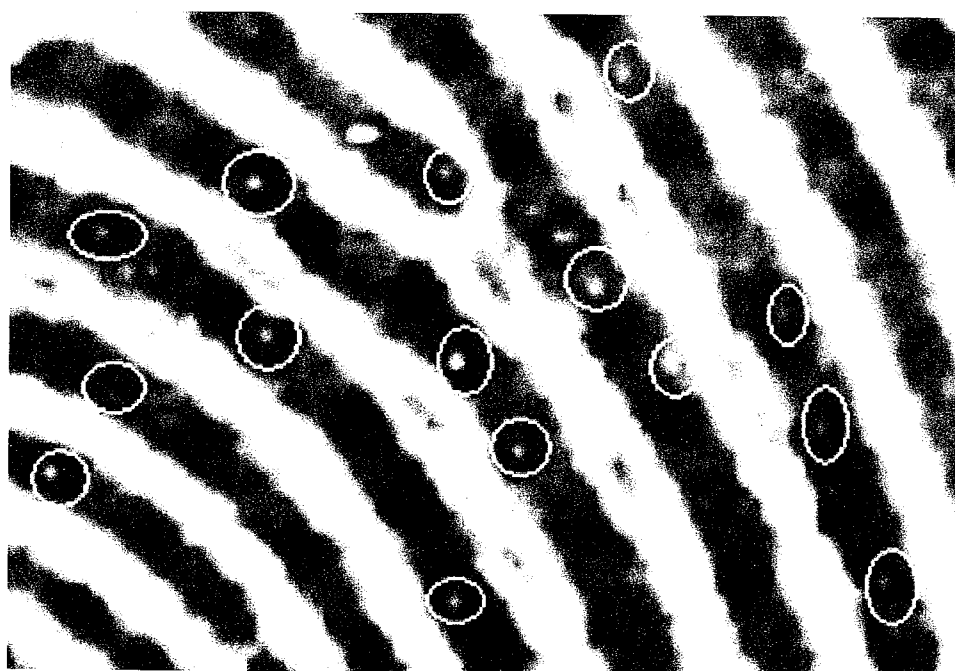


图 1.12 指纹的毛孔特征

的一生当中都不会改变 (Jain et al., 2009), 因此使用指纹的这三级特征作为身份识别的依据非常适合。

指纹的特征提取就是从指纹图像中获取指纹识别所需要的特征。目前大部分的指纹自动识别系统主要用的特征是指纹的奇异点以及指纹的细节点。这些系统的特征提取算法一般都是一个基本的流程: 指纹图像的前背景分割、指纹图像的方向场估计、指纹图像的奇异点提取与分类、指纹图像的增强、二值化与细化、指纹细节点的提取。下面将简要的介绍一下这些步骤。

1.5.2 指纹图像的前背景分割

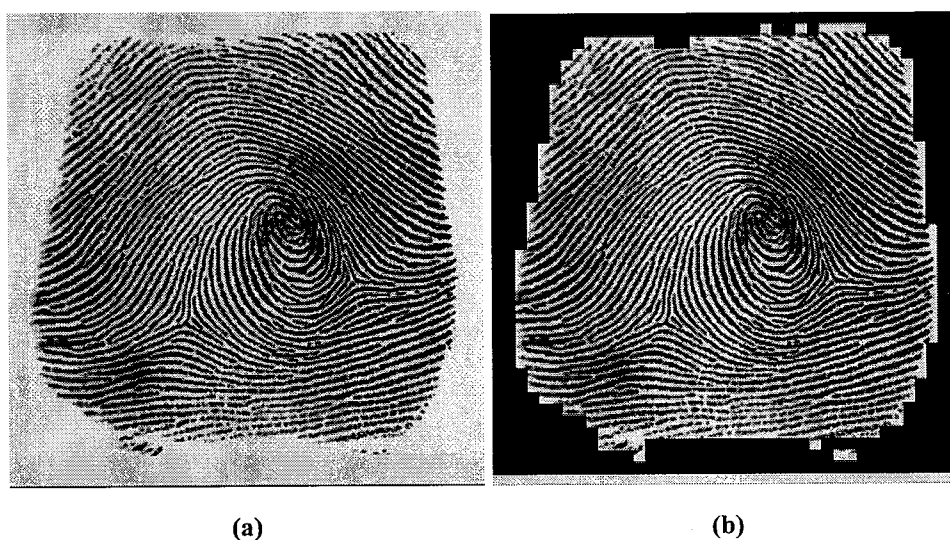


图 1.13 指纹分割示例图: (a) 原图, (b) 分割得到的前景区域

指纹图像的前背景分割是将指纹图像中指纹区域与非指纹区域分割开来, 保留指纹区域而去除非指纹区域。如图1.13所示, 分割后确定出前景区域, 这样后续的处理只需在前景区域上进行操作, 从而在提高效率的同时又可避免非指纹区域带来的干扰。在民用指纹识别中, 所处理的指纹一般由传感器采集得到, 因此背景区域多为电磁噪声。而在犯罪现场采集得到的现场指纹图像的背景要复杂许多, 对前背景的分割也要求更高。

对于背景噪声大、质量中等或差的指纹图像来说, 指纹图像的前背景分割准确率对指纹的特征提取结果有着重要影响。如果前背景分割的准确, 那么后续的处理可以避免背景噪声区域的影响, 提取的特征就比较可靠, 反之, 如果将背景噪声区域判定为指纹区域, 那么后续处理将无法改正这个错误, 从而有较大的可能在背景区域提取得到错误的特征。

目前,大部分的指纹分割方法都是基于图像的均值、方差以及纹线方向的一致性信息,例如文献 (Mehtre et al., 1989) 使用了基于纹线方向一致性的方法,该方法首先将指纹图像分块,然后分别计算每一个指纹图像块的梯度方向直方图。对于一个指纹区域的块来说,它的梯度直方图有明显的峰值,而如果一个杂乱无章的背景噪声块,它的梯度直方图就会比较扁平而没有明显的峰值。文献 (Ratha et al., 1995) 使用了基于方差的方法,该方法计算了指纹图像每一个块在不同方向上的灰度值方差,对于指纹区域块,沿指纹脊线的方向上该方差值小,而垂直方向上的方差值大。对与背景噪声区域,在各个方向上它的方差值都很小。相对于前两种使用单一信息的方法,文献 (Shi et al., 2004) 综合了灰度的均值,方差,偏心率以及对比度来分析每一个图像块是否是指纹区域,该方法在低质量图像的分割中能更加有效。

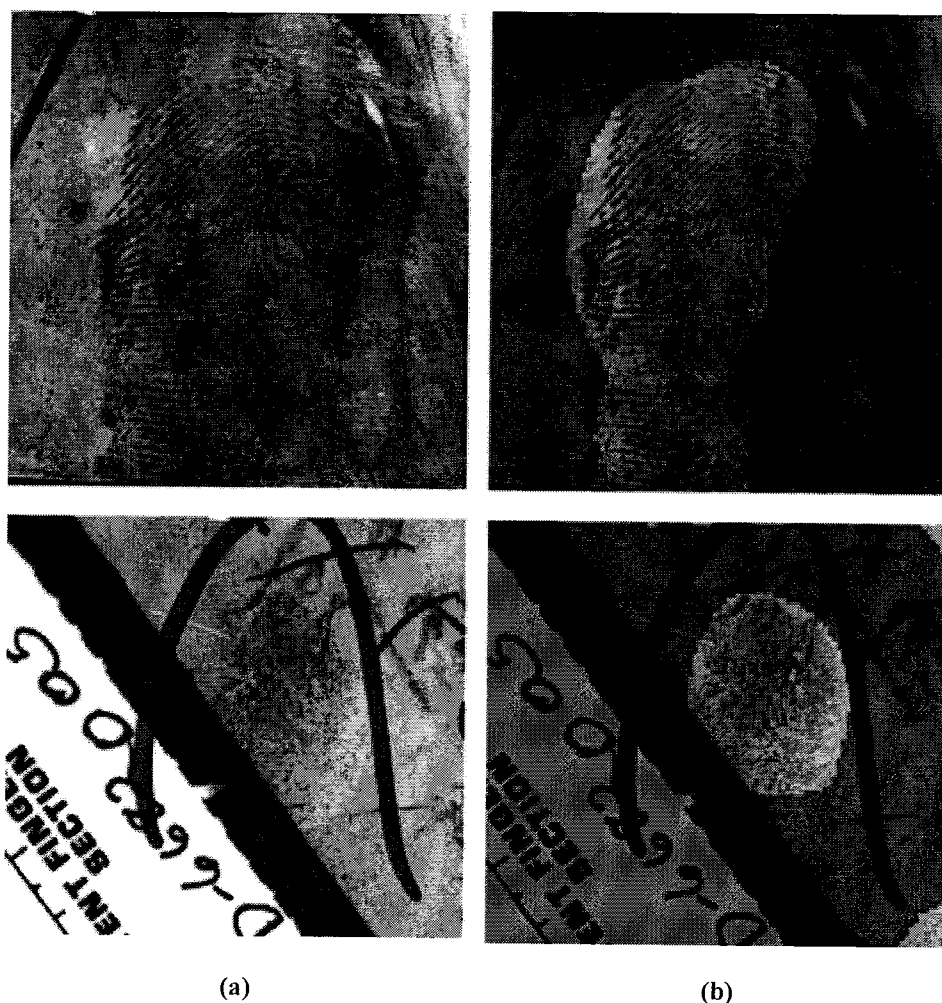


图 1.14 文献 (Zhang et al., 2013) 方法对于现场指纹的分割结果: (a) 原图, (b) 分割结果

上面所提到的方法都是专家先验知识驱动的方法。近几年来,模型驱动和

数据驱动的模式分类方法得到迅速发展，这些方法也都成功应用到指纹分割中。文献 (Zhang et al., 2013) 提出了一种自适应方向全变差模型，该模型能将现场指纹分解为纹理和卡通两部分，卡通部分即为非指纹信息部分，而纹理部分即为指纹信息部分。图1.14为该方法的结果，可以看到其对于复杂的背景区域和较差的前景区域有较好的处理效果。文献 (Dai et al., 2017) 使用了最新的机器学习方法，将深度卷积神经网络用于指纹分割中，也得到了不错的效果。

1.5.3 指纹图像的方向场估计

指纹脊线的方向是指纹的一个重要特征，指纹的方向场就是指指纹脊线方向的一种表示。方向场是一个二维矩阵，矩阵的大小与所需的精细度有关。最高的精细度下，方向场与指纹图像大小一致。方向场上位置 (x, y) 的值 θ_{xy} 表示指纹图像 (x, y) 像素位置的方向，该方向定义为纹线在 (x, y) 的切线与水平轴形成的夹角，由于指纹纹线方向没有极性，因此 θ_{xy} 的取值范围为 $[0, 180^\circ]$ ，如图1.15所示。

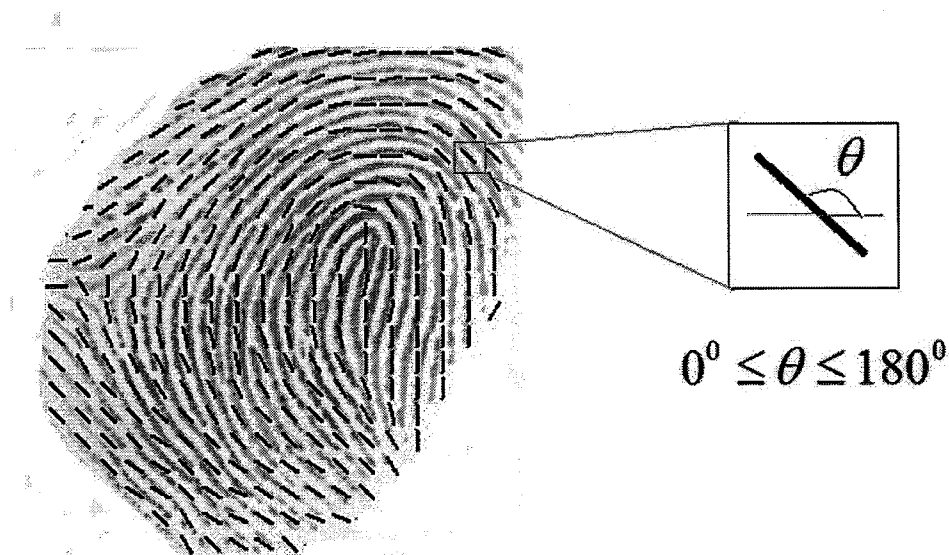


图 1.15 指纹方向场示例图，图片来自 (Shao et al., 2014)

通常情况下，方向场的大小并不等于指纹图像的大小，方向场的大小一般是通过人工经验来设定。方向场上每一点的值代表着指纹图像上对应小块的整体方向。例如在图1.15中，方向场中的每一个方向代表着原图中一个 16×16 小块的方向。这种策略将精度的损失控制在一定范围内的同时使得向场的计算量大大降低。

指纹的方向场对指纹特征提取的效果起着至关重要的作用。在后续的指纹

增强、奇异点提取步骤都将使用其结果，而指纹增强的结果直接影响着最终细节点的提取。因此，好的方向场估计方法能够显著地提高指纹识别系统的性能。指纹方向场的估计是指纹研究中的一个重要研究方向，目前已经存在大量的指纹方向场计算方法，下面将简要介绍几种最主要的方法。

文献 (Jain et al., 2009) 给了最简单使用最广泛的梯度法，对于指纹图像上一点 (x, y) ，通过梯度算子求得该位置的梯度 (D_x, D_y) ，最后该点的局部方向 θ_{xy} 即为 $\arctan(D_y/D_x)$ 。这个方法计算简单，速度快，但是由于仅仅只用到局部信息，因而很容易受到噪声的影响，在低质量的指纹图像中效果不理想，特别是在褶皱大的指纹区域。虽然梯度法存在着各种不足，但在此基础上提出的各种改进增强方法都较好的解决了梯度法遇到的各种问题。

人们注意到使用局部信息虽然计算简单有效，但是它对于噪声太敏感，鲁棒性较差。因此开始引入其他信息来弥补局部信息的不足。其中最常用的方法就是光滑处理和全局优化。文献 (Ji et al., 2008; Oliveira et al., 2008) 将纹线方向 $[0, 180^\circ]$ 量化为几个离散方向，每个方向对应一个模版，对于待求方向的一个指纹块，计算这个小块与每个方向模版的响应值，根据响应值的大小来确定该块的方向。这类算法不仅能给出一个主方向，还能根据响应值给出其他的候选方向，在后续的局部平滑处理和全局处理中通过使用这些候选方向可使得求出的方向场更加的鲁棒，容错性更强。但是由于量化了角度，并且引入一些复杂的平滑和挑选策略，使得这类算法的计算速度比梯度法要慢，精度也较差。

尽管上面的方法对于中等以上质量的指纹图像可以得到较好的方向场结果，但是对于一些质量差、残缺、褶皱剧烈的指纹区域，上面的方法依旧无能为力。如图1.16所示，对于这些情况，仅仅使用现有信息已经无法得到更好的结果，需要用更加全局的信息来估计本身无法求取方向的低质量指纹区域。文献 (Sherlock et al., 1993) 提出了一种数学模型来描述指纹的方向场，叫作零极点模型。该模型将指纹的中心点和三角点看作是复平面上的零点和极点，当零点和极点的位置给定之后，通过模型公式即可计算出指纹图像上每一个位置处的方向。后续人们对这个模型做了许多的改进 (Gu et al., 2004; Zhou et al., 2004b,a)，使得该模型效果不断提升。但是，这个模型需要以指纹奇异点作为已知条件，而对于方向场难以求解的低质量指纹来说，求取奇异点本身就很困难。因此其他学者希望找到其他不需要奇异点等较难获得的信息的模型来表示方向场。文献 (Wang et al., 2007b) 使用三角级数来表示方向场，文献 (Ram et al., 2010) 使用多项式拟合的方法来估计方向场。这些方法虽然不再需要奇异点，但是只是一般的拟合方法照搬

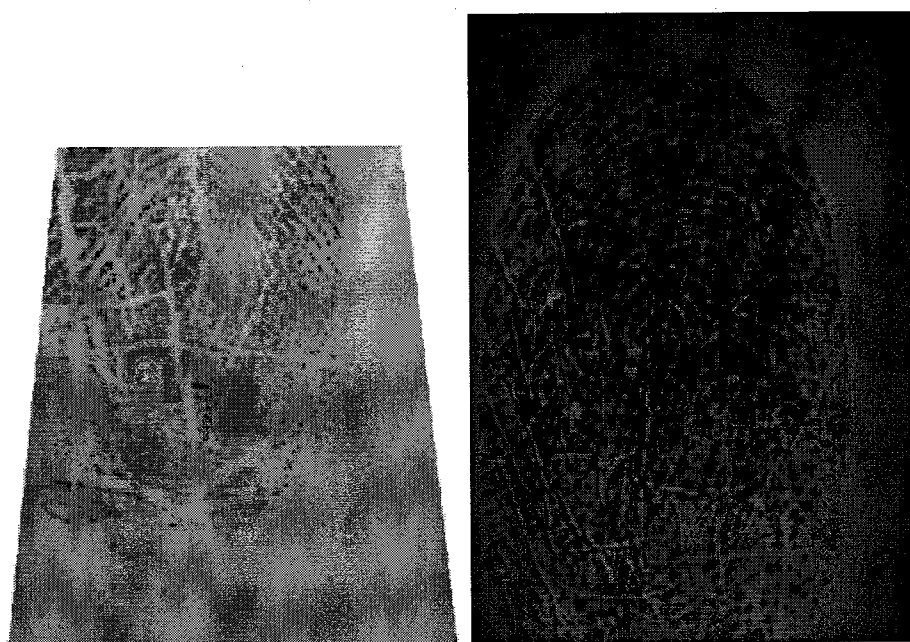


图 1.16 难以求取方向场的指纹图像

到这个问题上，没有使用任何指纹方向场的先验知识，因此效果不好。

近年来，基于机器学习和人工智能的方法在其他领域取得了显著的效果，一些学者受到启发，将这些方法引入到指纹方向场的计算之中。文献 (Shao et al., 2012) 提出了一种基于非负矩阵分解的指纹图像方向场估计算法。该方法通过学习的方法对各个方向的指纹图像块建立基矩阵，然后通过图像块在基矩阵上的响应来判断方向。文献 (Feng et al., 2013) 提出了基于指纹已有知识结构的计算方法。该方法建立了一个指纹图像的方向场库，将指纹方向场的计算转化为在知识库中寻找。当所求指纹图像的方向场模式在知识库中时，该方法可以取得较好的效果，但是所求指纹图像的方向场模式不在知识库中时，效果骤降。并且该方法的计算量相对于简单方法来说开销巨大。文献 (Cao et al., 2015) 将计算每一个指纹块中的方向看作是一个模式分类问题，并使用深度卷积神经网络来求解这问题。在现场指纹的方向场估计上取得了不错的效果。但是这类方法需要大量的样本进行神经网络的训练，且模型复杂度高。

1.5.4 指纹图像的奇异点提取

在前面的1.5.1部分介绍了指纹图像中的奇异点特征。在自动指纹识别系统中，奇异点起着非常重要的作用。通过奇异点的数量、位置和类型，可以将指纹进行纹型的分类，在匹配过程中只需要进行同类型纹型的指纹匹配即可，从而可

以大大降低在大库指纹系统中的识别时间。除此之外，在同类型纹型的指纹匹配中，奇异点还可以当作参考点，通过充分利用此信息能大大提升指纹验证的速度。因此奇异点提取算法的好坏对自动指纹识别系统影响很大，好的奇异点提取算法可以获得更加可靠、准确的奇异点，从而在保证精度的同时可以大大提升指纹系统的匹配速度。

文献 (Kawagoe et al., 1984) 首先提出了基于 Poincare 指数的指纹奇异点检测算法。这也是目前最简单且使用最广泛的一种方法。该方法通过计算指纹图像上每一点的 Poincare 指数来判断其是否是奇异点。对于指纹图像上的一点 (x, y) ，Poincare 指数定义为围绕该点的任意一条封闭曲线在方向场上逆时针旋转一周，方向场取值上改变量的总和。假设指纹方向场为 O ，则 (x, y) 处的 Poincare 指数通过如下公式计算：

$$Pincare(x, y) = \lim_{r \rightarrow 0^+} \frac{1}{2\pi} \int_0^{2\pi} \frac{\partial}{\partial \theta} O(x + r \cos \theta, y + r \sin \theta) d\theta \quad (1.1)$$

对于指纹图像上的一点 (x, y) ，获得 Poincare 指数之后，再通过公式1.2即可判断其是否是奇异点以及奇异点类型。

$$Pincare(x, y) = \begin{cases} 180^\circ & \text{中心点} \\ -180^\circ & \text{三角点} \\ 0 & \text{其他} \end{cases} \quad (1.2)$$

基于 Poincare 指数的算法因其简单实用性一直是目前各种指纹系统所广泛使用的奇异点检测算法 (Kawagoe et al., 1984; Zhang et al., 2004; Bazen et al., 2002)。但是最传统的 Poincare 指数方法非常依赖于方向场的准确性，对于质量不好的指纹图像，由于方向场不准确，会检测出大量虚假的奇异点而丢失真实的奇异点。在 Poincare 指数方法的基础上，许多学者对其进行了改进 (Zhang et al., 2004; Wang et al., 2007a; Zhou et al., 2007)。文献 (Zhou et al., 2009) 在使用传统的 Poincare 指数之后，通过提出的一种 DORIC 特征来去除 Poincare 给出的大量虚假奇异点，然后再通过模型来挑选奇异点使得最终得到的奇异点与方向场达到模型的最优值。

文献 (Srinivasan et al., 1992) 通过计算指纹图像每一个小块中的局部方向直方图来判断小块中是否存在奇异点，由于奇异点区域的曲率较大，方向变化剧烈，不存在主方向。因此如果直方图没有明显峰值，则认为该区域为奇异点区域。

思想类似，文献 (Koo et al., 2001; Zheng et al., 2006) 则是直接计算局部区域的曲率，通过得到的曲率值来判断是否存在奇异点，高曲率区域被认为是奇异点区域。文献 (Fan et al., 2008) 提出一个基于零极点模型和哈夫变换的奇异点检测算法。零极点模型是通过奇异点位置和类型来表示指纹方向场，在该方法中，其假设指纹方向场已知来求奇异点的位置和类型，相当于零极点模型的逆问题求解。但该方法找不到解析解，并且是一个 NP 难的问题。作者通过巧妙的利用哈夫变换对该问题进行了近似求解，并取得了不错的效果。但是该方法依然依赖于方向场的结果，当方向场结果不准时该方法性能骤降。

1.5.5 指纹图像的增强

经过上面介绍的步骤，奇异点特征已经提取得到。特征提取剩下的目的是为了准确的提取得到稳定的指纹细节点特征。在1.5.1部分我们介绍了指纹的细节点特征，细节点是指纹脊线拓扑结构的重要表示，它所包含的信息是后续指纹匹配的根基，细节点提取结果的准确性与稳定性直接影响着指纹系统的最终识别精度。为了得到精准的细节点结果，就需要获得精准的脊线骨架图，这便是指纹图像增强的目的。

对于质量好的指纹图像，脊线和谷线之间区别明显，可以很容易获得清晰的脊线结构信息。但是对于质量差的指纹图像来说，其对比度一般较低，脊线谷线边界很难区分。如图1.17所示为各种质量的指纹图像。对于这些不同质量的指纹图像来说，获得清晰脊线结构的难易程度有很大区别。指纹增强的主要作用是对中低质量的指纹图像进行一些处理，使得脊线和谷线之间的差异明显，从而得到稳定准确的脊线结构信息。

最常用的指纹图像增强方法是基于 Gabor 滤波器的方法 (Hong et al., 1998)，Gabor 滤波的模版与指纹图像平坦的局部区域非常相似，都是黑白相间的条纹状，其数学表达式如下：

$$g(x, y, \theta, f) = e^{-\frac{1}{2}(\frac{x_{\theta}^2}{\sigma_x^2} + \frac{y_{\theta}^2}{\sigma_y^2})} \cos(2\pi f x_{\theta}) \quad (1.3)$$

其中 x, y 为指纹图像的横纵坐标， θ 为滤波器的方向， f 为滤波器的频率， σ_x 和 σ_y 分别为高斯模版在横纵方向上的标准差， (x_{θ}, y_{θ}) 为坐标 (x, y) 进行方向 $(90 - \theta)$ 度旋转后的坐标。

Gabor 滤波器中的方向和频率两个参数和指纹局部区域求得的方向场和频率所表示含义一致。因此，在得到方向场和频率场之后，可以设置一簇 Gabor 滤波

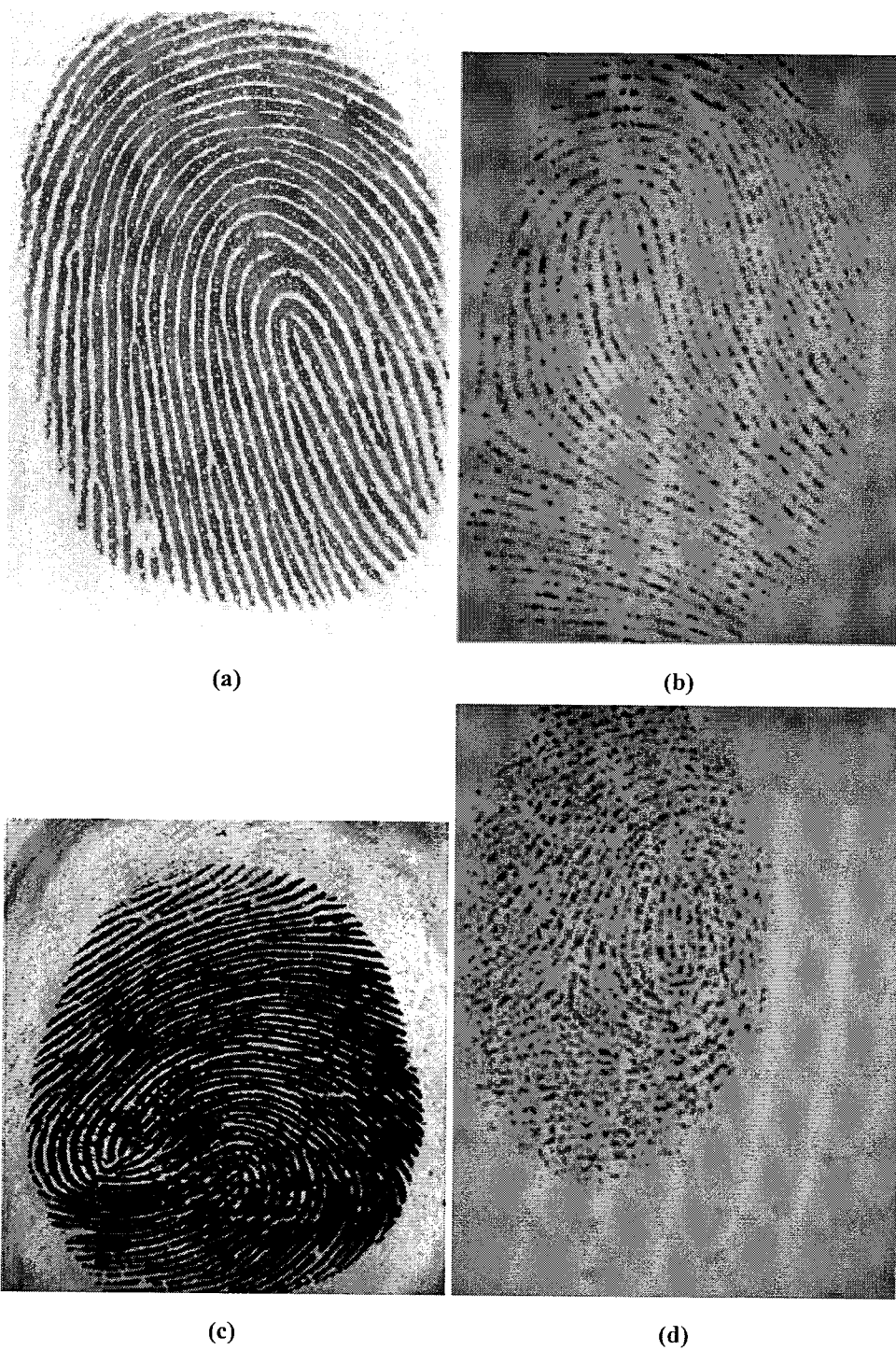


图 1.17 各种质量的指纹图像：(a) 高质量的指纹图像，脊线和谷线区别明显；(b) 脊线不清晰，连续性差；(c) 脊线和谷线分界线不明显，有多个糊成一片的区域；(d) 低质量指纹，噪声大，脊线不清晰

器对相应区域进行滤波，从而使得图像脊线和谷线的对比度得到增强。图1.18给出了一幅指纹图像使用 Gabor 滤波增强之后的结果。

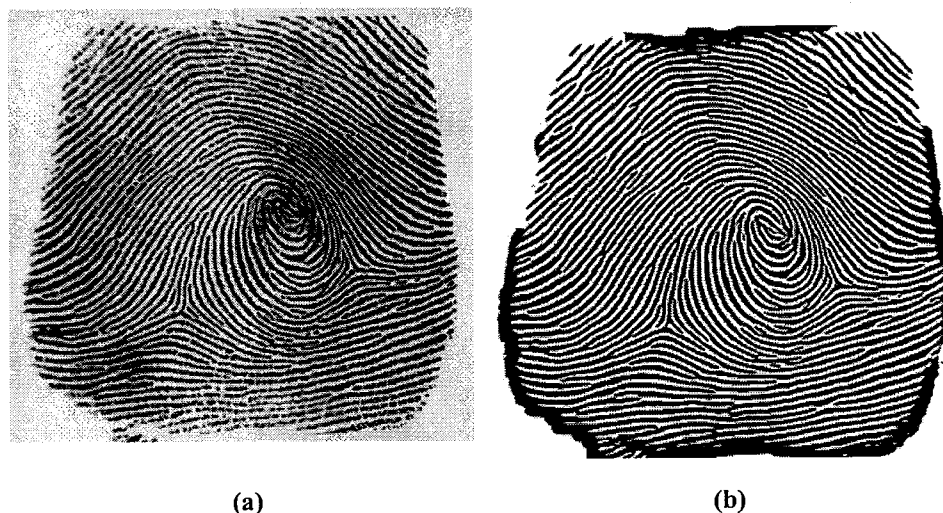


图 1.18 Gabor 滤波对指纹图像增强的结果：(a) 原图；(b) 增强后的结果

1.5.6 二值化与细化

如图1.18b所示，增强之后的指纹图像中脊线和谷线区别明显，图像对比度强，在此基础上通过二值化与细化操作即可得到指纹的骨架图。

顾名思义，二值化操作是将图像转为二值图像。在图像处理领域，图像二值化是一个重要研究方向，已经被广泛研究。文献 (Trier et al., 1995) 给出了一个最简单的方法，通过一个人工经验的阈值来对图像上的每一个像素点逐一进行二值化。对考虑的像素点的值高于该阈值时，像素值重新取值为 1，反之取值为 0。该方法虽然简单，但是没有考虑到指纹图像各个区域的差异性，仅凭单一的阈值常常无法取得较好的结果。文献 (Zhang et al., 2006b) 针对这个缺陷提出了改进的方法，考虑了指纹各个区域的灰度差异性，对每个区域自适应地选取一个阈值来进行局部二值化操作。

细化操作 (Naccache et al., 1984; Arcelli et al., 1985) 的目的是将二值图中的脊线宽度缩减至一个像素，得到的结果被称为指纹的骨架图。指纹的骨架图简化了指纹图像，保留了指纹最重要的脊线拓扑信息，在此基础上进行细节点提取是目前的指纹识别系统普遍采用的方法。图1.19给出了一幅指纹图像进行特征提取的各个步骤直到得到骨架图的结果。

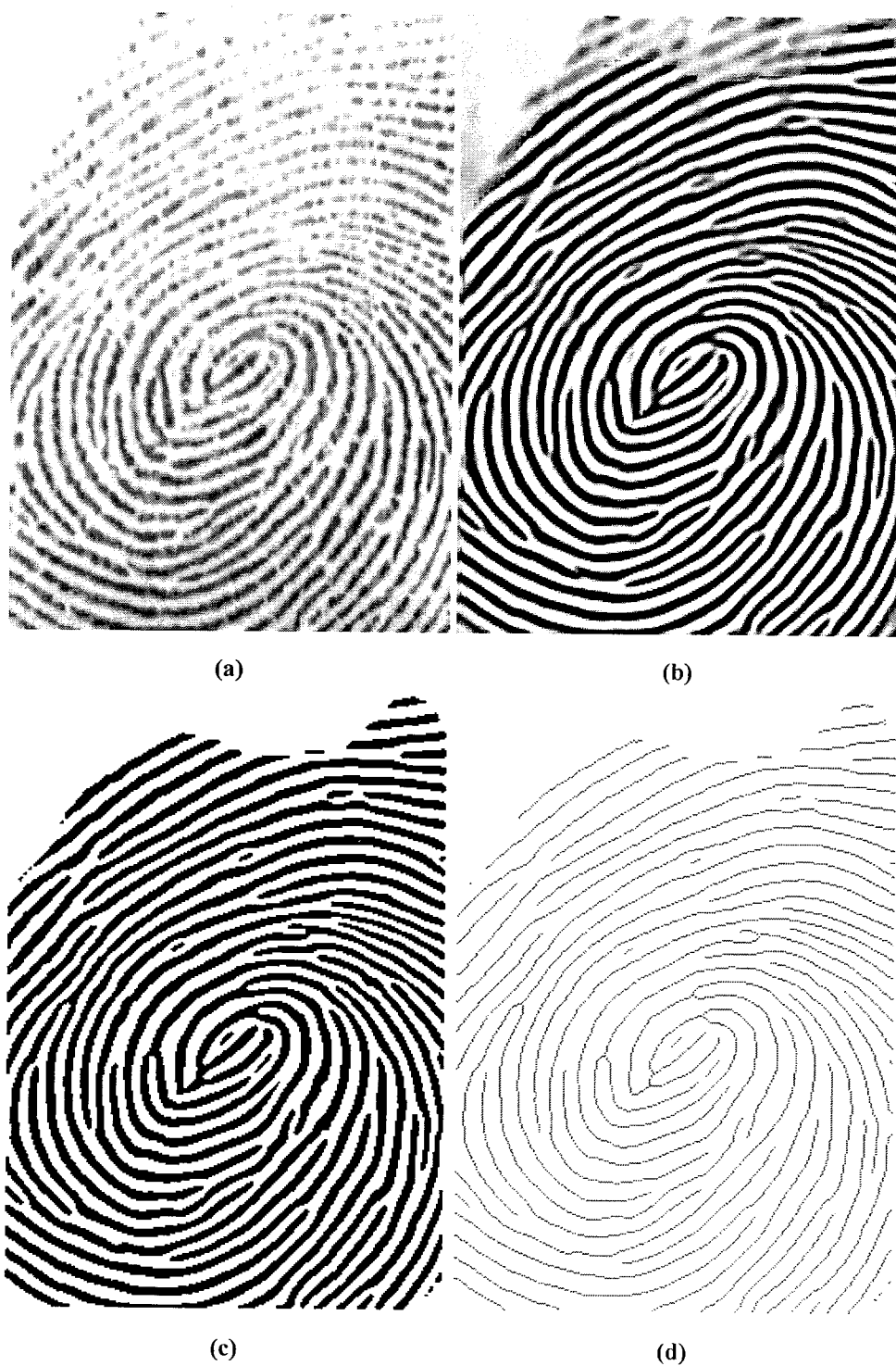


图 1.19 (a): 原图; (b): 对 (a) 做指纹增强后的结果; (c): 对 (b) 做二值化的结果; (d): 对 (c) 做细化的结果

1.5.7 指纹细节点的提取

对骨架图上的每一个脊线点,检测其8邻域中的脊线点个数,如果只有1个点,则说明该点为细节点,并且类型为端点;如果有2个点,则说明该点是脊线上的点而并非细节点;如果有3个或4个点,则说明该点是细节点,并且类型为分叉点;由细化操作带来的特性,使得脊线点周围8邻域的脊线点个数不会大于4个。

由于特征提取的各个步骤都无法达到绝对准确,同时低质量区域还会带来许多负面效果,因此经过上面检测步骤得到的指纹图像的细节点中存在许多伪细节点。对于如何判断细节点是否真伪,一般是基于专家先验知识的方法和基于机器学习的方法。文献(Rao, 1976; Hung, 1993; Farina et al., 1999)通过人工经验的总结给出了一些去除伪细节点的策略,例如去除脊线孔洞、马刺、桥等带来的虚假细节点。文献(Maio et al., 1998; Prabhakar et al., 2003)使用了机器学习的方法来判断每个细节点的真伪,这类方法需要准备充足的学习样本才能得到较好的结果。

1.6 指纹的特征匹配

经过上一节指纹的特征提取之后,指纹图像的匹配转化为指纹的特征匹配。在大多数民用的指纹识别系统中,指纹图像一般不会保存在数据库中,对于每一个需验证身份的用户,系统保存的是其指纹图像提取得到的指纹特征。通常指纹特征占用的空间要比指纹图像的小很多。

通常,指纹匹配与指纹的特征提取关系密切,匹配算法本身对自动指纹识别系统的性能有直接影响,但特征提取的优劣对匹配算法也起着至关重要的作用。文献(Jain et al., 2009)对传统的指纹匹配算法进行了总结,根据所使用的特征,将目前的指纹识别算法分为三类:(1)基于像素相关性的匹配;(2)基于细节点的匹配;(3)基于纹线特征的匹配。

基于像素相关性的匹配来源于传统的图像匹配,思想简单,不需要进行特征提取,直接使用指纹图像本身进行匹配。通过枚举所有可能的对应关系,计算每个对应关系下图像之间像素的相关性获得匹配的相似度。该方法没有用到太多指纹本身的特性,并且没有考虑指纹的非线性形变问题,因而性能较差,在实际的自动指纹识别系统中很少使用。

基于纹线的匹配主要是为了弥补当细节点数量不足的情况。在遇到部分指纹匹配或是小面积指纹匹配时,细节点由于数量不足而难以提供足够的区分能

力, 此时就需要引入其他额外的信息。文献 (Ross et al., 2006) 使用薄板样条模型 (TPS) 处理指纹的非线性形变问题, 使得细节点与脊线点能得到更加准确的对应结果。文献 (Feng et al., 2006) 通过脊线上的点和细节点构建了子结构, 同时考虑细节点与脊线的对应关系。文献 (Marana et al., 2005) 使用哈夫变换来求解脊线点集的对应关系, 将脊线匹配与细节点匹配进行融合后获得了更好的结果。文献 (Fang et al., 2007) 对细节点所在的脊线进行跟踪采样, 得到一些脊线上的点来表示整条脊线, 这些点被称为脊线表示点 (RRP, Representative Ridge Point), 然后将脊线表示点加入到细节点集来弥补细节点的不足。

尽管现在已经提出了许多指纹匹配算法, 但真正能在目前的自动指纹识别系统得到应用的主要还是基于细节点的匹配算法。在目前的中国指纹身份证标准 (中华人民共和国, 2012) 中规定了指纹的特征仅为细节点。即各指纹识别厂商提供的匹配算法只能使用细节点。因此本章的指纹特征提取主要围绕细节点提取进行介绍, 下面主要介绍基于细节点的匹配算法。

基于细节点的匹配是目前实际的自动指纹识别系统广泛采用的方法。该方法将指纹图像的匹配转化为了指纹的细节点集匹配, 而细节点集的匹配可看作是一般的点集匹配问题 (Point Pattern Matching), 而点集匹配问题是模式识别领域中一个重要核心问题, 对其研究非常广泛 (Ranade et al., 1980; Chang et al., 1997; Goshtasby et al., 1985)。文献 (Jain et al., 1997) 给出一个完整的细节点匹配算法, 在实际系统中成功应用。文献 (Kovacs-Vajna, 2000) 在细节点集上做了三角剖分, 将问题转化为三角形匹配, 并针对细节点的弹性形变做了相应处理。文献 (Chikkerur et al., 2006) 提出了一种基于 K 邻域局部模式的细节点匹配算法, 该方法的本质思想是通过充分利用细节点集的结构信息快速找到真实对应关系。

在基于细节点的匹配算法当中, 基于局部模式的细节点匹配 (Chikkerur et al., 2006) 是目前使用最广泛的。该算法可简单的分为四个阶段: (1) 建立局部模式; (2) 计算两细节点集间的匹配关系; (3) 删除虚假匹配关系; (4) 计算匹配相似度。对每个细节点来说, 局部模式是其周围细节点分布的一种表示。每个细节点 m_i 的局部模式建立过程分为两个步骤: (1) 构建局部坐标系, 以细节点 m_i 为坐标原点, 以其方向为 X 轴的正方向; (2) 在该坐标系的每个象限中, 取距离坐标原点最近的 $K/4$ 个点, 作为该局部模式的邻点, 构成 K 邻域局部模式。在目前的实际应用中, K 一般取值为 8 或 12。对于每个局部模式, 其中心点 m_i 和每个邻点 m_j 都会得到一组旋转平移无关量 $(\theta_j^i, \phi_j^i, Dis_j^i)$, $(\theta_j^i, \phi_j^i, Dis_j^i)$ 的定义详见文献 (Chikkerur et al., 2006)。对于一个 K 邻域局部模式,

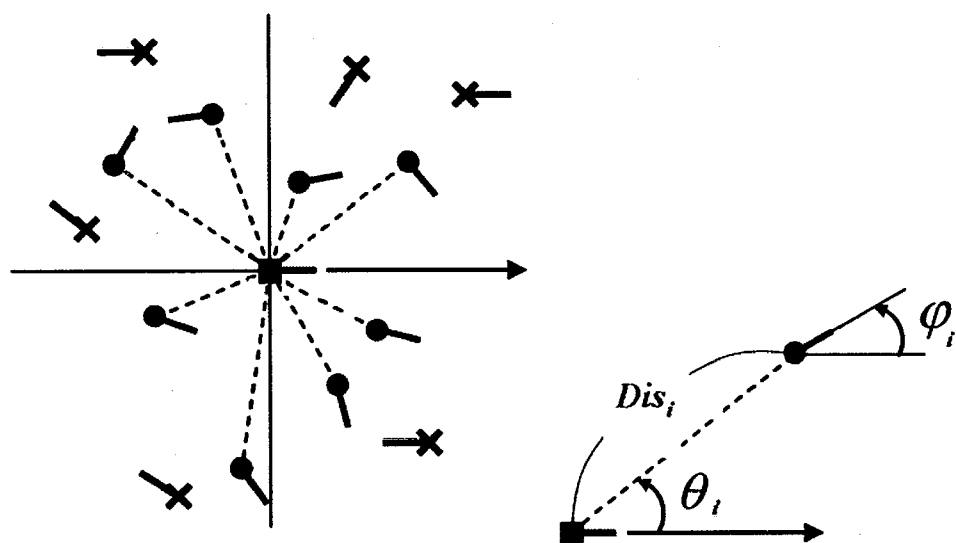


图 1.20 8 邻域局部模式示例，正方块为中心细节点，圆点为选取得到的邻点细节点，叉点为其余的细节点

其 K 个邻点分别与中心点构成一组旋转平移无关量，因此每个局部模式可表示为： $LP_i = (A_1, A_2, \dots, A_K)$, $A_j (j = 1, 2, \dots, K) = (\theta_j^i, \varphi_j^i, Dis_j^i)$ 。图1.20给出了一个 8 邻域局部模式示例。图1.21是整个指纹图像建立的局部模式，被称为星状图。

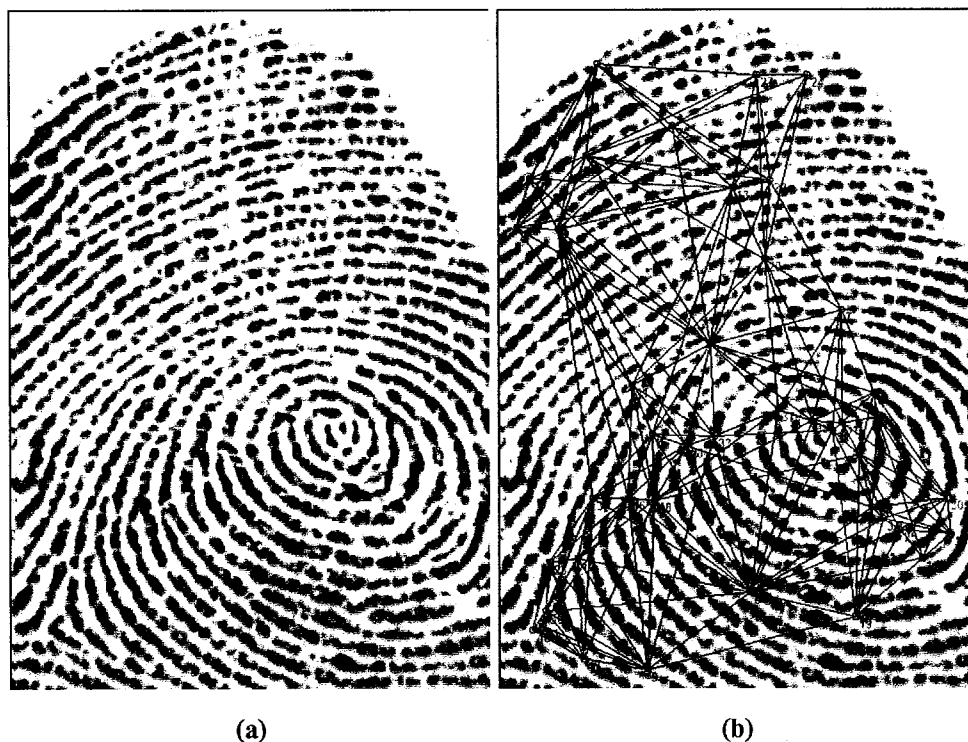


图 1.21 指纹图像的所用 8 邻域局部模式组成的星状图：(a) 原图；(b) 细节点 8 邻域局部模式星状图

在局部模式建立完之后，即可根据局部模式计算细节点之间的相似度。文献 (Chikkerur et al., 2006) 中使用动态规划的方法来计算两个局部模式之间的相似度。经过局部模式的匹配之后，模版指纹中的任一个细节点与查询指纹的所用细节点均可获得一个相似度分数，其中可能有多个高相似度匹配结果，但是一个细节点最多只能与一个细节点匹配。因此从众多候选匹配中确定正确的一对一匹配是一个重要环节。文献 (Jea et al., 2005) 提出使用最小费用最大流的模型来求解该问题。文献 (Wen et al., 2009) 根据正确匹配关系的一致性对局部模式的匹配结果进行增强，然后直接用贪心的方法挑出最优的匹配。

在获得一对一的匹配关系之后，还需对这些匹配关系的变换一致性进行检测，这里的变换可以是刚性变换或仿射变换等。具体选择与算法整体设计的复杂度有关，一般来说，由于指纹匹配中存在着弹性形变，符合该假设的变换设计能获得更好的精度，但也会导致算法整体复杂度变高从而降低计算效率。

最终的匹配打分计算将获得两幅指纹图像匹配的最终相似度，直接关系到算法的性能。目前大多数文献 (Jain et al., 1997; Jea et al., 2005) 给出的都是通过匹配上的细节点数目与质量等进行的简单评估。而在实际应用的自动指纹识别系统中，最终的匹配打分计算一般涉及很多因素，包括匹配细节点数目和质量、重合面积、重合区域未匹配细节点数目和质量、重合区域质量等。

1.7 本文的主要研究内容

近年来，民用指纹识别应用发展迅猛，并取得了巨大成功。然而目前指纹识别的成熟度与人们所理解的不同，指纹识别在一些方面仍然存在着亟待解决的问题。本文主要研究了指纹识别中的两个主要问题，一是在大面积指纹中如何更有效地提取指纹奇异点，从而提升大面积指纹的匹配效率。二是目前尚处在发展阶段的小面积指纹识别算法。本文共分为六章，第一章为绪论；第二章为指纹奇异点提取算法的研究；第三章到第五章为小面积指纹识别的研究；第六章为结论与展望。

第一章绪论介绍了指纹识别的相关知识和研究现状，其中着重介绍了目前使用最广泛的指纹细节点特征的提取过程。

第二章提出了基于深度学习的指纹奇异点检测算法。该算法将奇异点检测问题转化为模式分类问题，充分利用了深度学习方法和指纹大数据，通过全卷积神经网络直接在原图上识别出奇异点可能存在的区域。并在此基础上，设计了概率模型和启发式策略来确定奇异点的最终位置、数目和类型。与传统的奇异点检

测算法不同,该算法完全不依赖于指纹图像的预处理以及指纹的方向场。此外,多尺度策略的运用使得该算法对奇异点周围纹线缺失、奇异点周围噪声较大等情况具有更好的鲁棒性。实验结果表明,该算法的提取精度优于传统算法。

第三章研究了小面积指纹识别系统。小面积指纹识别系统的核心匹配算法、注册与验证等都与传统的指纹识别系统存在着较大差异。虽然目前提出的小面积指纹匹配算法有很多,但是其中并没有一个算法相比其他算法来说具有明显优势,各个匹配算法均有自己的优缺点,并且这些算法都还没有达到让人满意的程度。因此,目前的小面积指纹识别系统与核心匹配算法都还处于发展阶段。特别值得关注的是,2018年初出现智能手机中的小面积指纹识别系统被攻破一事,这一事件受到了社会媒体的极大关注。在本章中,我们对这一问题出现的原因进行了深入分析,并给出了相应的解决方案。

第四章提出了基于指纹脊线特征与改进的广义哈夫变换的小面积指纹匹配算法。该算法通过类似细节点特征提取的方法来提取得到精确可靠的脊线特征,针对小面积指纹脊线点集匹配问题的特点,改进了传统的广义哈夫变换,并用其来处理该问题。有效地改进使得该算法内存空间消耗极低且匹配速度满足实时要求。通过使用基于机器学习的匹配打分方法,使得算法性能又进一步提升。实验结果表明,该算法的匹配性能与目前其他主流算法相比也具有明显的优势。

第五章提出了传统图像匹配算法与当前主流的深度学习相结合的小面积指纹匹配算法。首先改进了传统的极坐标傅立叶变换算法,并将其用来进行小面积指纹的粗对齐,在此粗对齐的基础上,设计了一个深度卷积神经网络来进行相似度评估。该算法不依赖于任何人为定义的特征,即使指纹图像质量较差但仍然能够进行匹配。此外,该算法可以进行多种尺寸的指纹匹配的同时训练,得到的深度卷积网络模型可以适应任意尺寸的输入。实验结果表明,该算法性能优异且具有较好的鲁棒性和稳定性。

第六章总结了博士期间的研究工作,并对后续研究做了展望。

第2章 基于多尺度全卷积神经网络的指纹奇异点检测算法

2.1 引言

作为指纹的全局特征，奇异点在自动指纹识别系统中扮演着重要角色。由于具有旋转平移不变性，奇异点被绝大多数指纹识别算法用作参考点，从而可以大大提升指纹匹配的速度。其次，指纹纹型可根据奇异点来确定，而通过指纹纹型可进行指纹的分类，进而可大大降低在大规模指纹库上的指纹查询时间 (Jain et al., 1999; Chan et al., 2004a)。

目前，学者们已经提出了许多奇异点的检测方法。最传统的奇异点检测方法是文献 (Kawagoe et al., 1984) 中提出的基于 Poincare 指数的方法。该方法通过计算以方向场中一点为中心的一条逆时针曲线上方向变化的累加和来判断该中心点是否是奇异点。基于 Poincare 指数方法的基础上，人们又做了许多改进 (J. Zhou, 2009; Bazen et al., 2002)。但是几乎所有的基于 Poincare 指数的方法都使用局部信息来检测奇异点，而这样的特性使得算法对噪声敏感，不鲁棒。此外，基于 Poincare 指数的方法是一种非常依赖于指纹方向场的方法，对于低质量指纹或是奇异点周围存在褶皱、疤痕等影响的指纹时，方向场通常较难获得一个准确的结果，从而导致奇异点检测也失效。文献 (L. Fan et al., 2008) 提出了一种融合了零极点模型 (Zero-pole Model) 和哈夫变换 (Hough Transform) 来检测奇异点的方法。与使用零极点模型来生成方向场相反，该方法通过已知的方向场来求解零极点模型中奇异点的数目、位置以及类型。通过哈夫变换使用全局的方向场信息来弥补局部信息的不足。最终其使用 Poincare 指数的方法来对哈夫变换求出的奇异点位置进行一个更加精细的修正。该方法虽然相对于传统的 Poincare 指数来说考虑了全局的信息。但其最终检测结果的准确度还是依赖于方向场以及 Poincare 指数。到目前为止，大部分主要的奇异点检测方法都需要方向场的估计，此外，奇异点检测结果的精度也完全依赖于方向场的精确程度。在本章中，我们将提出一种完全不依赖方向场的奇异点检测方法。

在后面的小节中，我们将详细题提出的奇异点检测算法。首先，我们在第二节简要介绍本章所用到的深度学习知识，在第三节我们构建了针对指纹图像的多尺度全卷积神经网络，并使用其检测奇异点区域。第四节将详细介绍估计最终奇异点位置的概率模型。第五节是我们的算法在 NIST-4 和 FVC2002 上的实验结果。第六节给出本章的结论。

2.2 深度学习中的卷积神经网络介绍

深度学习的概念源于人工神经网络，其本质上是多层神经网络。根据其中所使用的层的传播操作不同，可以将多层神经网络分为很多类，卷积神经网络便是其中的一类，且是目前为止最为成功的一类。卷积神经网络中的主要层操作是卷积运算，该操作降低了输入对平移旋转和缩放的敏感度，并提取出有效的空间特征。此外，该操作具有权值共享、稀疏连接等优点，大大降低了网络的参数量，使用简单的随机梯度法即可将网络训练得很好。第一个具有代表性的卷积神经网络模型是 LeCun 提出的 LeNet-5 模型 (LeCun et al., 1998)。在这之后，研究者们进行了各种改进，使得卷积神经网络模型的深度和宽度不断增大，表示能力不断增强。其中主要的有分类网络 AlexNet (Krizhevsky et al., 2012)、目标检测网络 R-CNN (Girshick et al., 2014)、VGG (Simonyan et al., 2014)、GoogleNet (Szegedy et al., 2015)、SPP-Net (He et al., 2015) 等。

2.2.1 神经网络的基本原理

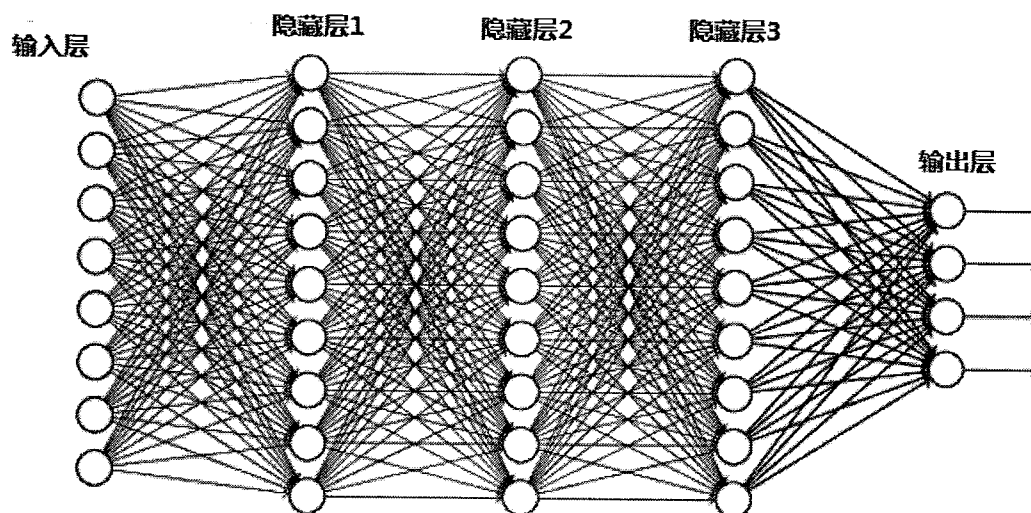


图 2.1 一个 5 层的神经网络示例图

在介绍卷积神经网络之前，我们首先介绍更为简单基础的普通神经网络。图 2.1 给出了一个 5 层的神经网络示例图。一个神经网络最简单的结构包括输入层、隐藏层和输出层，每一层网络有多个神经元，上一层的神经元通过激活函数映射到下一层神经元，每个神经元之间有相对应的权值，输出即为分类类别。一般的神经网络又叫作全连接神经网络，其前一层的每一个神经元都会连接到后一层的每一个神经元，如果前一层有 m 个节点，后一层有 n 个节点，那么总共有

$m \times n$ 条边连接，每一条边具有一个权重参数，因此这两层之间就有 $m \times n$ 个参数需要学习。对于整个网络来说，所需要训练的参数非常多，参数一多，模型就变复杂，越是复杂的模型就越是难以优化，也更容易过拟合。此外，从神经网络的反向传播过程来看，梯度在反向传播时会不断的迭代导致越来越小并趋于零，造成参数权值更新缓慢甚至停滞，模型很难收敛。因此，在图像领域，最基本的神经网络是不太合适的。

2.2.2 卷积神经网络的基本原理

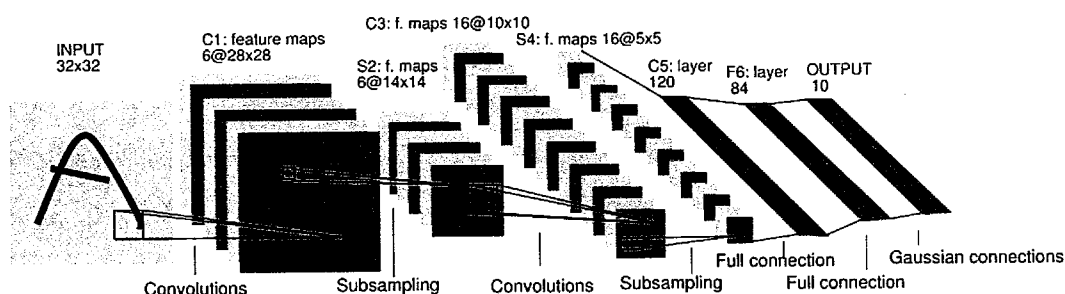


图 2.2 LeNet-5 模型 (LeCun et al., 1998)

图2.2是最具代表性的卷积神经网络 LeNet-5 模型 (LeCun et al., 1998)，该网络被用来识别手写体数字，具有非常优异的效果。可以看到，卷积神经网络沿用了普通的神经元网络的结构，是一个前馈网络。这个典型的结构可以分为五个部分：（1）输入图像层。为了减小复杂度，一般使用灰度图像。当然，也可以使用 RGB 彩色图像。输入图像一般需要归一化，如果使用 sigmoid 函数，则归一化到 $[0, 1]$ ，如果使用 tanh 函数，则归一化到 $[-1, 1]$ ；（2）卷积层。该层为核心层，上一层的输出与本层权重做卷积得到下一层的输入。（3）池化层。该层实际上是一个下采样层，采样方法有均值采样或是最大值采样等。（4）全连接层。这一层相当于普通神经网络中的隐藏层。（5）Softmax 层。这一层对应最后的分类模式输出，本质上也是一个全连接层，但是激活函数不一样。

2.2.2.1 卷积层

卷积操作是数字图像处理中常使用的一种操作。给定一幅图像 $X(i, j), 1 \leq i \leq M, 1 \leq j \leq N$ ，和滤波模版 $F(i, j), 1 \leq i \leq m, 1 \leq j \leq n$ ，通常 $m \ll M, n \ll N$ 。则卷积结果 $Y(i, j)$ 为：

$$Y(i, j) = \sum_{u=1}^m \sum_{v=1}^n F(u, v) \cdot X(i - u + 1, j - v + 1). \quad (2.1)$$

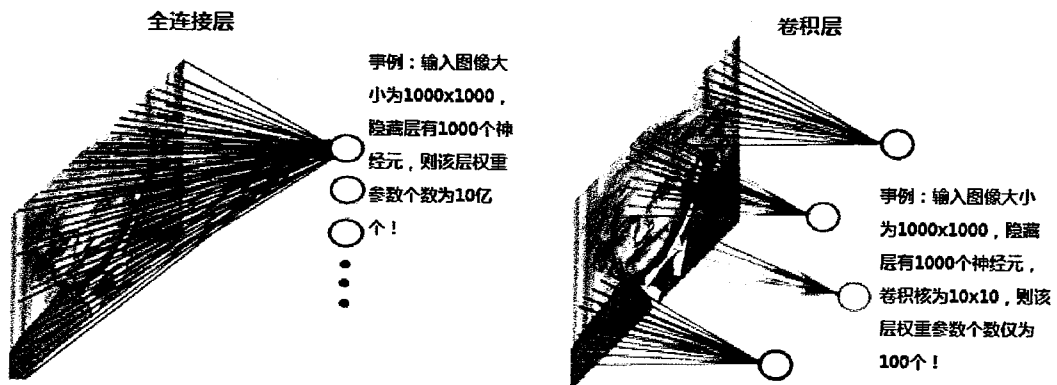


图 2.3 全连接层与卷积层的对比

卷积层的使用，使得神经网络对图像的处理能力大大提升，其中最重要的一个原因就是其大大降低了模型的复杂度，如图2.3所示，在普通的神经网络中每个神经元都要与输入图像上每个像素相连接，这样就会造成权重参数的数量巨大，从而导致网络难以训练。而在含有卷积层的神经网络中每个神经元的权重都是同一个卷积核，相当于每个神经元只与对应图像部分的像素相连接，并且这些连接都共用相同的权重参数，这样就极大的减少了权重参数的数量。

2.2.2.2 池化层

池化层本质上就是下采样层，其操作也可以看作是一种卷积操作，只不过下采样的卷积核并不是需要训练的权重参数，而是事先已经设定好的参数，等价于取对应位置的平均值、最大值等（平均池化、最大池化）。图2.4是一个简单的最大值池化实例，在图左边部分中，左上角 4×4 的矩阵中6最大，右上角 4×4 的矩阵中8最大，左下角 4×4 的矩阵中3最大，右下角 4×4 的矩阵中4最大，所以图右边部分的池化结果为：6 8 3 4。池化，简而言之，即为区域平均或最大。

虽然卷积操作大大减少了权重参数的数量，但是图像在经过卷积之后，数据量依然很大。图像中相邻位置的像素相似度很高，因此使用下采样方法可以大大降低数据维度的同时只损失少量的信息。池化层是卷积神经网络中非常重要的一个部分，其大大减少深度卷积神经网络的计算量，同时又使得深度卷积神经网络更加鲁棒。目前，使得最多的池化是最大值池化。

2.2.2.3 激活函数

激活函数是使得神经网络具有非线性表示能力的主要贡献部分。目前主要的激活函数有 sigmoid, tanh, ReLU, ELU, PReLU 等激活函数。

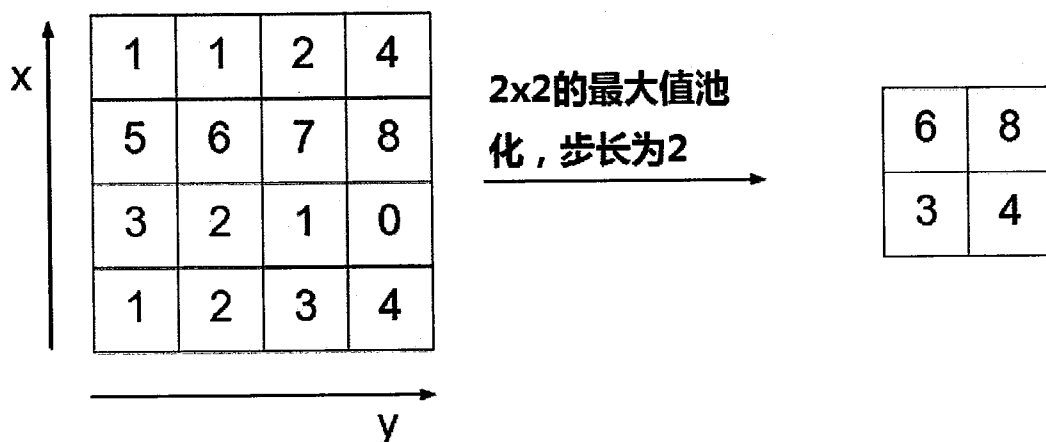


图 2.4 一个简单的最大值池化事例

sigmoid 函数公式如下:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}. \quad (2.2)$$

sigmoid 函数的值域为 $(0, 1)$, 通常会将其输出当作是神经元激活或抑制的概率。由于 sigmoid 函数与神经科学中神经元突触对信号的处理很相似, 其曾经一度非常流行。但是后来在深度学习的实际应用和发展中, 人们发现 sigmoid 函数有一个缺点, 当输入的绝对值较大时, sigmoid 函数的梯度就变得很小, 导致在反向传播时权重更新缓慢甚至停滞, 这一现象通常称为梯度消失。

tanh 函数公式如下:

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \quad (2.3)$$

tanh 函数可以看作是类 sigmoid 函数, 其好处是它的输出均值是零。与 sigmoid 函数类似, tanh 函数在早期也比较流行, 但是其具有和 sigmoid 函数一样的问题, 也会导致梯度消失。

ReLU 函数公式如下:

$$\text{ReLU}(x) = \max(0, x). \quad (2.4)$$

ReLU(Rectified Linear Unit) 函数是目前流行的一个激活函数, 相比于 sigmoid 函数和 tanh 函数, 它不存在梯度消失现象, 并且它的计算速度要快很多。它缺点则是当输入为负时, 将完全不被激活。

ELU 函数公式如下:

$$\text{ELU}(x) = \begin{cases} x & , x > 0 \\ \alpha(e^x - 1) & , x \leq 0 \end{cases}. \quad (2.5)$$

ELU 函数是 ReLU 函数的一个种改进版, 相比于 ReLU 函数, 其在输入为负的情况下依然有一定的输出, 而且这部分输出还具有一定的抗干扰能力。

PReLU 函数公式如下:

$$PReLU(x) = \max(\alpha x, x). \quad (2.6)$$

PReLU 也是针对 ReLU 的一种改进版, 在输入为负的情况下 PReLU 有一个很小的斜率, 这样也可以避免 ReLU 不被激活的问题。

图2.5给出了目前主要的激活函数。总体来看, 这些激活函数都有自己的优点和缺点, 关于激活函数的选取, 目前还没有定论, 实践中更多的还是需要结合实际情况, 参考实验结果, 考虑不同激活函数的优缺点来综合使用。

2.2.2.4 卷积神经网络的训练

整体上来看, 可以将神经网络当作是一个函数, 只不过该函数的形式非常复杂, 参数很多。参数确定之后, 神经网络即可看作是一个非线性程度很高的映射函数, 它可以将高维的输入数据映射到我们所期望的分布上。训练的目的即为选取一组最佳参数, 使得输入数据经过网络映射之后, 得到的输出尽可能的接近我们想要的预测值。

考虑一个简单的普通神经网络, 其由 n 层全连接层组成, $W_k (k = 1, 2, \dots, n)$ 表示每一层的权重, $f_k (k = 1, 2, \dots, n)$ 表示每一层的激活函数, 假设我们有 N 个输入数据 $X_i (i = 1, 2, \dots, N)$, 对应的期望输出为 $Y_i (i = 1, 2, \dots, N)$, 则我们学习的目标, 训练的目的, 即是求解如下这样一个优化问题:

$$\min_{W_k (k=1,2,\dots,n)} Loss = \sum_{i=1}^N \|Y_i - f_n(\dots(f_2(f_1(X_i W_1) W_2) \dots) W_n)\|^2. \quad (2.7)$$

可以看到, 对于深度神经网络的训练, 求解难度主要取决于神经网络的模型的复杂度, 对于深度卷积神经网络来说, 对应的上述优化问题将更加复杂。目前, 深度卷积神经网络训练主要采用最简单的梯度下降法, 并且由于计算量的限制, 只能使用小批次的随机梯度下降法。

2.2.3 全卷积神经网络

全卷积神经网络是卷积神经网络的一种改进版, 它将卷积神经网络中的全连接层全部替换成了卷积层, 并在之后又添加了反卷积层, 实现了对每个像素点

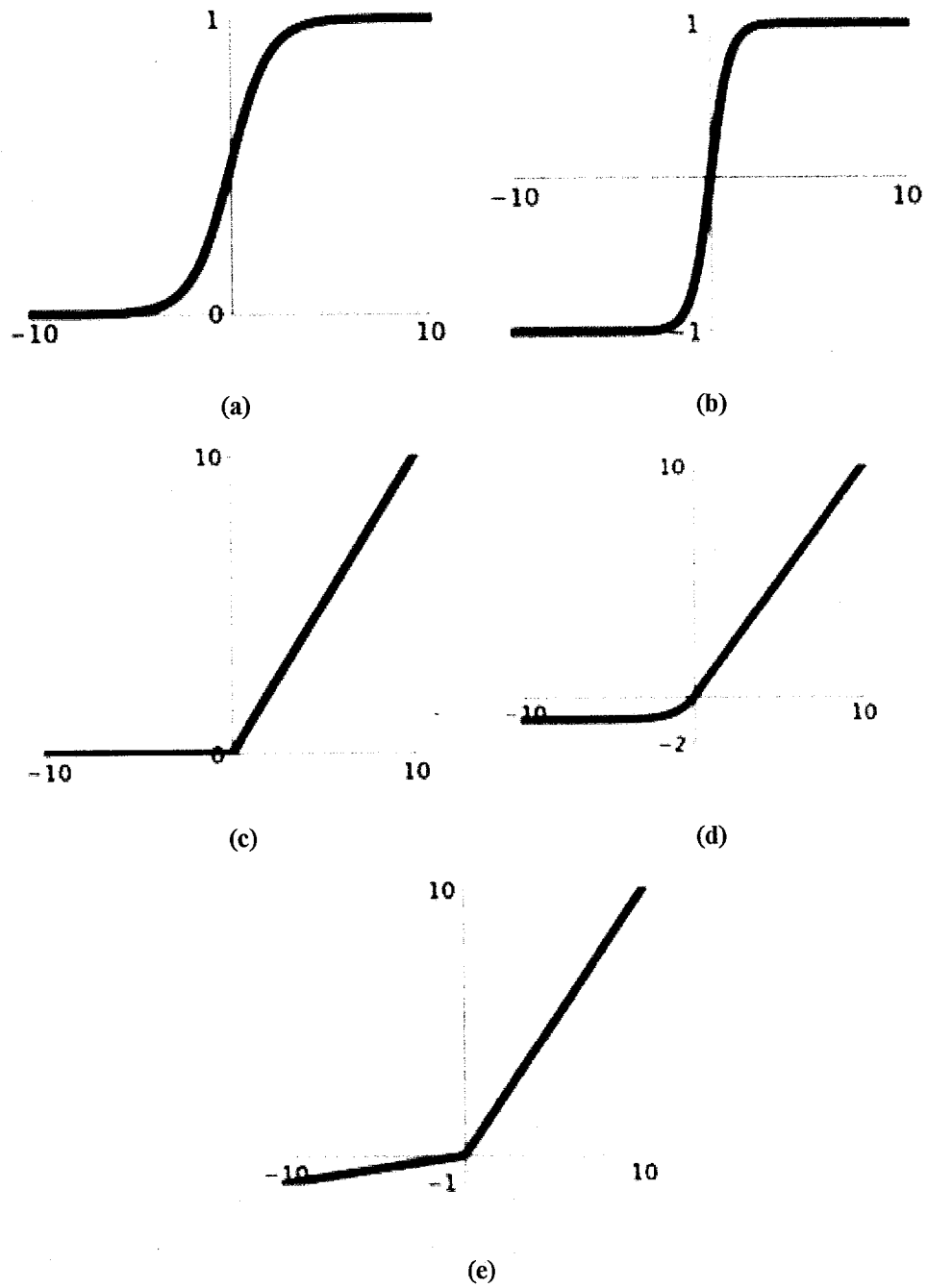


图 2.5 常见的激活函数 (a) sigmoid 函数 (b) tanh 函数 (c) ReLU 函数 (d) ELU 函数 (e) PReLU 函数

的模式预测。此外，相比于卷积神经网络，全卷积神经网络的输入可以为任意尺寸。图2.6是文献 (Long et al., 2015) 用于语义分割所采用的全卷积网络的结构示意图。

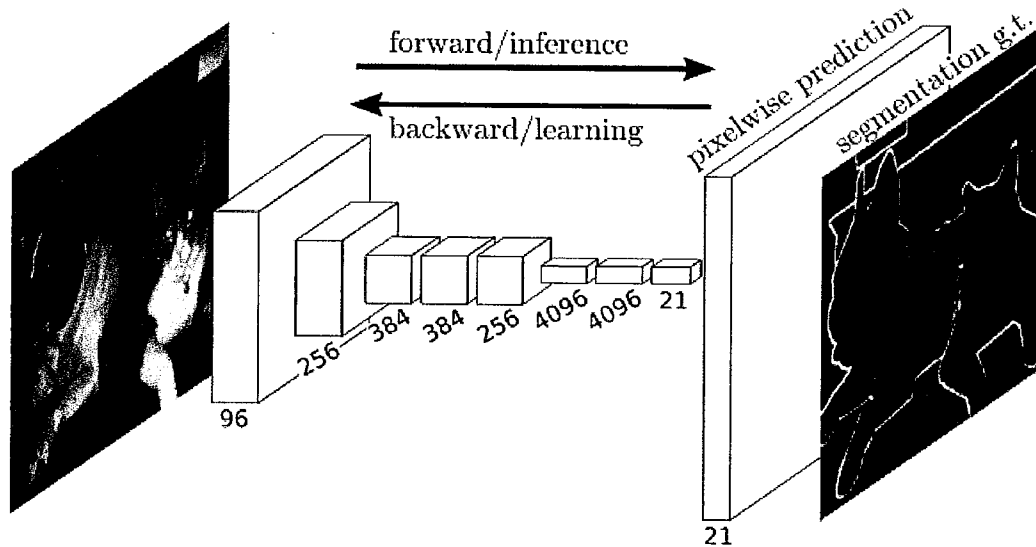


图 2.6 全卷积神经网络用于像素级语义分割，图片来自于文献 (Long et al., 2015)

全卷积神经网络中最重要的是上采样部分，即反卷积。这里需要注意的是，它并不是卷积的逆过程，确切的来说，它更应该称为转置卷积。卷积层的前向操作可以表示为输入矩阵和权重矩阵相乘，那么，我们很容易得到卷积层的反向传播就是和权重矩阵的转置相乘。在卷积神经网络中，随着层数的增加，卷积层的输出所表示的信息会越来越抽象，更高层的输出代表着更加全局的特征，较低层的输出则代表相对局部的特征。由于不同层上可以获取到不同层次的特征，全卷积神经网络中一般会将各个层上的输出进行叠加融合，在此基础上来进行上采样操作，优化输出结果。

2.3 检测奇异点区域的多尺度全卷积神经网络算法

卷积神经网络是一种非常有效的识别方法。在近几年的模式识别领域发展中，这一点已经被充分的验证。卷积神经网络在手写体识别和人脸检测与识别中都取得了显著的成果 (Hinton et al., 2012; A. Krizhevsky, 2012)，受此启发，我们提出了一种基于全卷积神经网络的奇异点检测方法。为了使用卷积神经网络，需要将奇异点检测问题转换为一个图像模式识别问题。为此，我们将指纹图像进行分块处理（块与块之间可能重合），然后对一个指纹图像块，判断其中心是否为奇异点。这是一个典型的图像分类问题。我们训练了一个卷积神经网络来处理

这个问题,然后将训练好的卷积神经网络转换为一个全卷积神经网络,并对其微调。通过该全卷积神经网络,对输入指纹图像的每一个像素点进行其是否为奇异点的判定。

2.3.1 卷积神经网络的构建

构建卷积神经网络时,我们充分考虑指纹的一些特殊情况。例如在图2.7a中,指纹图像的三角点(奇异点)附近的纹线信息缺失。在这种情况下,如果我们的指纹块不够大,那么在以这个真实的三角点为中心的指纹图像块中就几乎没有任何纹型信息,从而无法对该图像块进行正确的判定(很大可能判定为非奇异点区域、背景区域)。然而,如果我们的指纹分块处理使得指纹块过大又会使得我们的检测精度下降。这与人眼的视觉系统相类似,当我们在一幅指纹图像上寻找奇异点时,我们的视线是由宽到窄进行变化的,在这个过程中,我们对奇异点的位置估计越来越准确,最终我们视线聚焦到一点,即奇异点位置。受到人眼搜索奇异点的方式的启发,我们将多尺度的策略融入到奇异点检测中,具体的实现是通过在卷积神经网络中加入空间金字塔池化层(He et al., 2015)。

我们在实验中使用的卷积神经网络结构如图2.8所示。我们将该网络命名为CNN-C。该网络将一幅任意大小的指纹图像块作为输入,神经网络的第一层为卷积层,包含了16个大小为 11×11 的卷积核。紧接着的是一个步长为2的最大值池化层,其输出作为第二个卷积层的输入,第二个卷积层使用了大小为 5×5 的32个卷积核。后面卷积层的相似操作处理如图2.8定义。经过第四个卷积层之后,数据流将被输入到空间金字塔池化层(He et al., 2015),该层将输出大小为 10×10 的特征图,最后将这些特征图输入一个全连接Softmax层中,Softmax层的输出即为三个节点,代表判别的三种模式(中心点模式(core pattern),三角点模式(delta pattern),非奇异点模式(nonsingular point pattern))。

为了充分利用卷积神经网络强大的特征表示能力与分类能力,我们需要大量充分的样本来训练神经网络。我们的实验室拥有千万人级的数据库,这些数据是中国各个省市的公安厅所提供。我们从中随机挑选了1万5千幅指纹图像来进行此次实验,这些指纹图像包括各种不同纹型的指纹,指纹图像的质量从低到高分布均匀。所有的这些指纹图像大小均为 640×640 像素大小。我们将所有以奇异点为中心的指纹图像块(块的大小有三种尺寸: 80×80 像素大小, 120×120 像素大小, 160×160 像素大小)都切割出来作为训练神经网络所使用的正样本。要获取到这1万5千幅指纹图像中奇异点的准确信息,需要人工来对这些图像进行标签化的工作,然而这是一个耗费巨大人力和时间的工作。为了获取到足够的

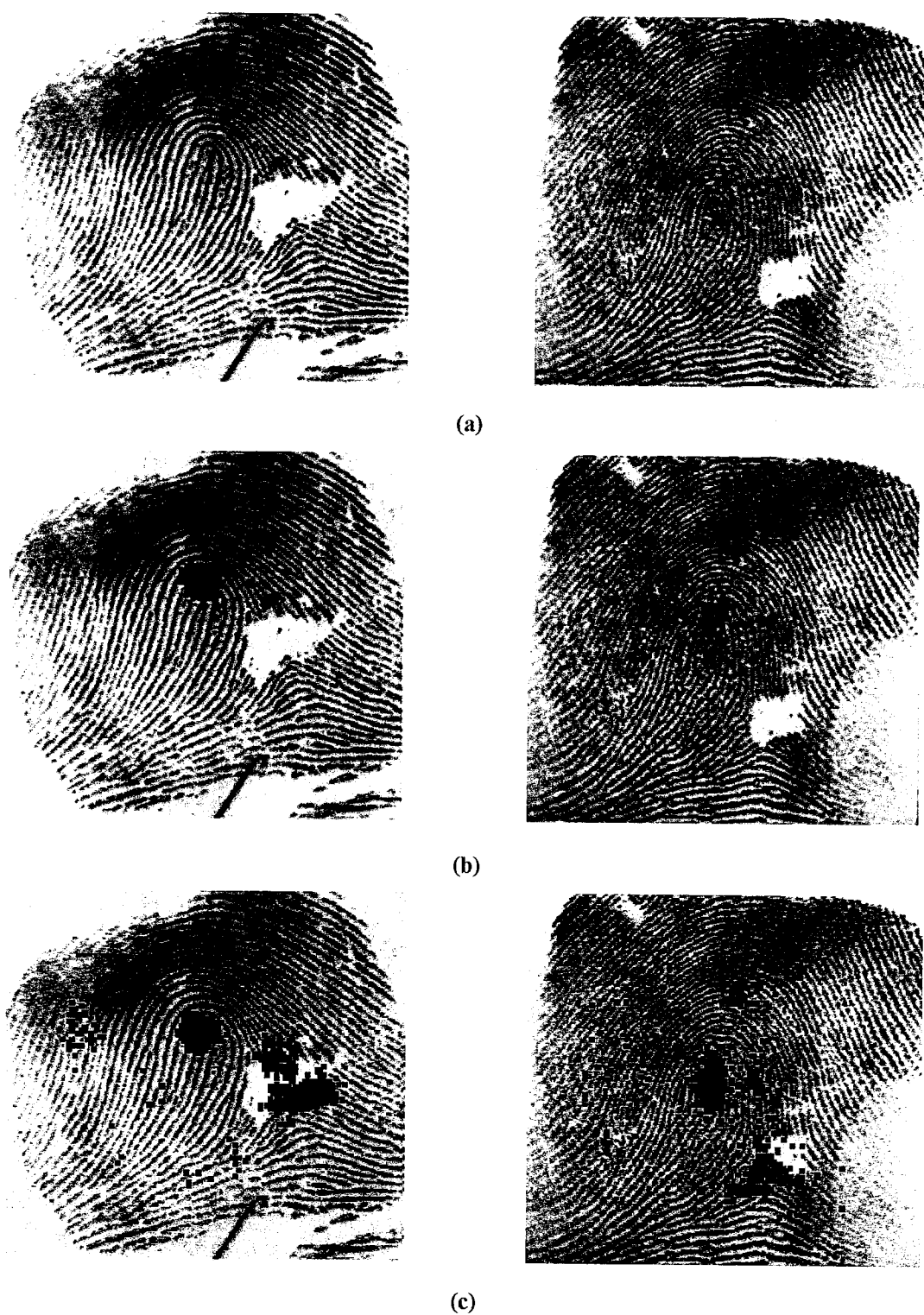


图 2.7 (a) 来至于 NIST-4 中的两幅指纹图像，其奇异点区域的纹线信息缺失。(b) 使用 FCN-SingleScale80 对 (a) 中两幅指纹图像的奇异点检测结果。(c) 使用 FCN-SingleScale160 对 (a) 中两幅指纹图像的奇异点检测结果。

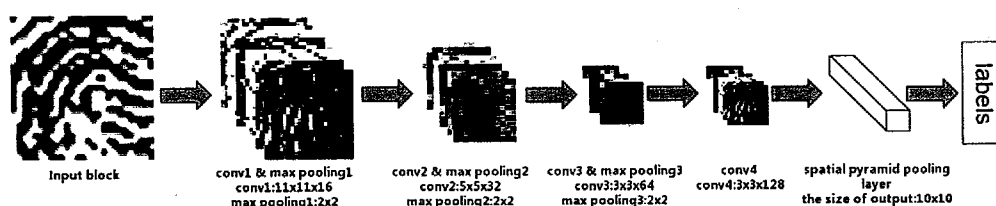


图 2.8 CNN-C 的网络结构示例图

训练样本，我们在实验中通过指纹奇异点自动提取算法 (L. Fan et al., 2008) 来获取奇异点标签数据。通过使用这样的数量充足、内容充分的训练样本库，我们的神经网络将得到充分的训练。

2.3.2 奇异点区域的检测

如果仅仅使用卷积神经网络来对一幅指纹图像中的每个像素点进行是否是奇异点的判定，则有以下诸多缺点：（1）巨大的存储开销。为了对每个像素点都进行判定，就需要切割出以每个像素点为中心的指纹图像块，并逐一将这些图像块输入到神经网络中进行判定，而这些图像块存在着大量的重复区域，使得空间开销大大增加；（2）计算效率低下，需要判定的图像块之间存在着大量的重复区域，这些重复区域在神经网络中被大量的重复计算。（3）如何选取适当的图像块尺寸（即神经网络的输入大小）是一个问题。如果选取的尺寸太小，则图像块中仅仅包含了非常局部的信息，导致无法处理局部信息不足的情况。而如果选取的尺寸太大，则图像块包含太多无用信息，距离较近的像素点生成几乎同样的图像块，导致奇异点位置的估计不精确。

因此，我们将卷积神经网络转化为全卷积神经网络，使用全卷积神经网络来一次性对全图的像素点进行预测估计，从而达到实用高效的目的。如图2.9所示为全卷积神经网络检测的示例图，将卷积神经网络转为全卷积神经网络的方法如同文献 (Long et al., 2015)。

为了提升全卷积网络检测的速度，并不需要对每个像素点都进行预测，这是由于距离很近的像素点之间具有很强的相似性。可以设置全卷积神经网络输出图的长宽为原始图像的 $1/n$ 。在本章的实验中，我们经对比实验发现 $n = 5$ 为最佳选择（速度和精度的综合）。图2.10给出了使用全卷积神经网络对一些指纹图像进行奇异点区域预测的结果。图中的黑点即为全卷积神经网络判定为奇异点的区域。在全卷积网络的输出中，对三种模式（中心点模式，三角点模式，非奇异点模式）都有一个概率预测值，三个预测值中的最大值所对应的模式即为原图

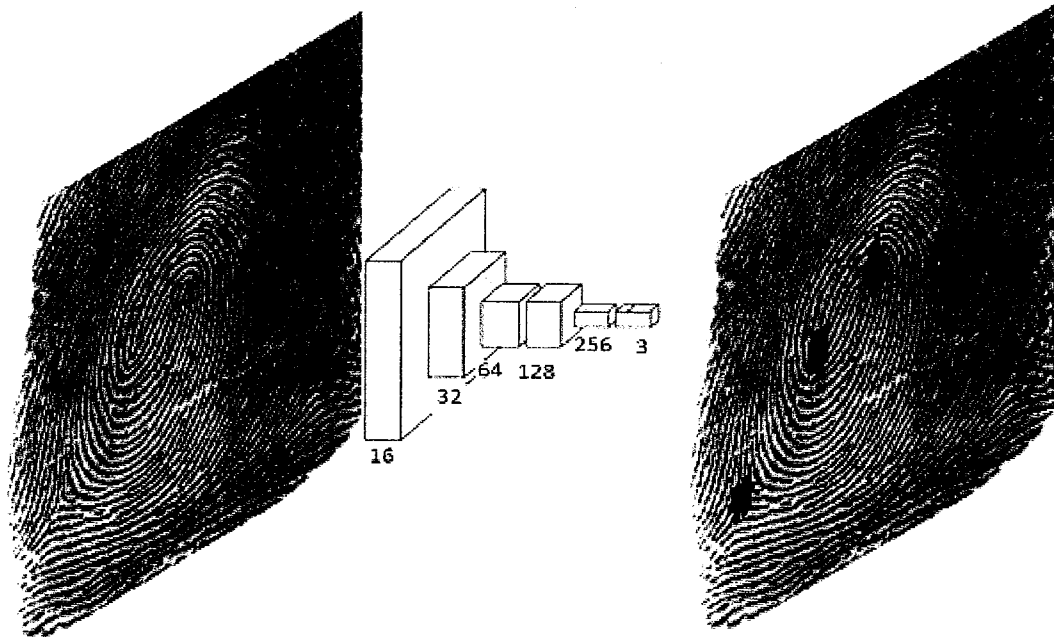


图 2.9 使用全卷积神经网络对指纹图像中的像素点进行是否为奇异点的判定，如果判定该像素为奇异点，则对该像素赋值 0（即标为黑色）。

中相应位置的预测结果。但对于奇异点模式（中心点模式和三角点模式），我们将判定该概率预测值是否大于某个阈值 w 来确定其是否为奇异点模式。图2.10中所使用的 w 为 0.95。

2.3.3 多尺度

在卷积神经网络 CNN-C 中，输入图像块的尺寸决定了神经网络中的感受野对应于指纹区域的大小。因此输入图像块的尺寸对神经网络的结果影响很大。过小的尺寸将导致神经网络只能提取到一些局部信息，从而导致分类的准确率下降。为了试验多尺度对于神经网络结果的影响，我们构造了一个对比试验。我们训练了两个卷积神经网络，CNN-CSingleScale80 和 CNN-CSingleScale160。这两个卷积神经网络的结构与 CNN-C 完全一样，但是所使用的训练样本是单一尺寸的。CNN-CSingleScale80 使用的训练样本尺寸均为 80×80 像素大小，相应地，CNN-CSingleScale160 使用的训练样本尺寸则均为 160×160 像素大小。使用相同的处理方式，我们将 CNN-CSingleScale80 和 CNN-CSingleScale160 分别转化为 FCN-SingleScale80 和 FCN-SingleScale160。

如图2.7所示，FCN-SingleScale160 能提取隐藏在信息缺失区域中的奇异点，但是由于视觉窗口也相对过大，因此其预测的精度比 FCN-SingleScale80 要低。

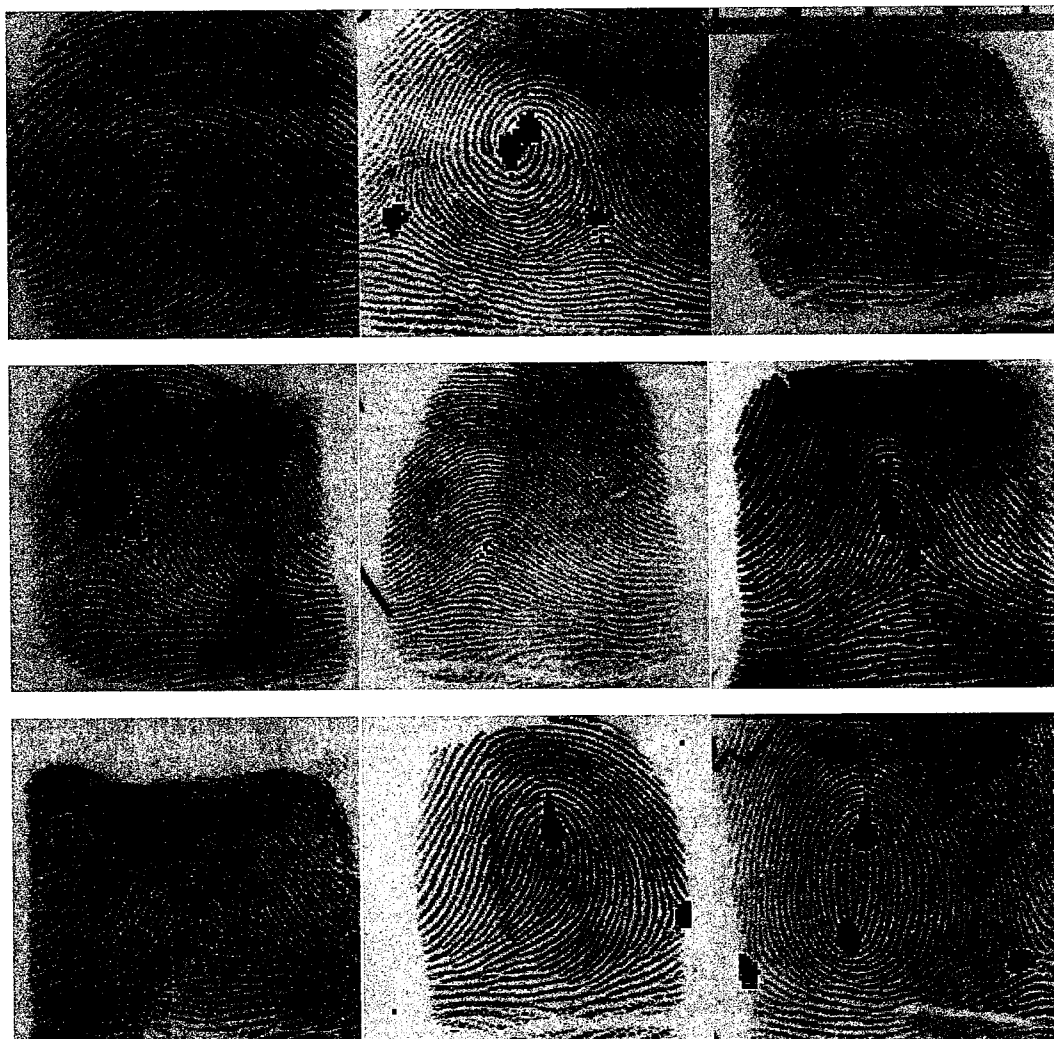


图 2.10 使用全卷积神经网络对指纹图像中的像素点进行是否为奇异点的判定，如果判定该像素为奇异点，则对该像素赋值 0（即标为黑色）

这与人类的视觉系统相类似，当我们所观察的范围很大时，范围内一些小物体看得就不是很清晰，而当我们仔细观察一个局部区域，获得清晰的结果时，我们的观察范围则只能维持在很小的范围。

2.4 定位奇异点位置的概率预测模型

如图2.10所示，经过全卷积神经网络的预测之后，我们得到了其输出的奇异点区域。在中高质量的指纹图像中，全卷积神经网络输出的错误预测区域很少并且覆盖范围都相对很小。随着所预测的指纹图像质量不断降低，错误的预测结果也相应增多如图2.10中第三行最左边的指纹图像，全卷积神经网络在其左上角错误预测出了一块奇异点区域（黑点覆盖区域），可以发现这是一块非常像中心点的区域。

奇异点检测的最终输出必须给出精准的一个位置而不能是一个区域，因此，在全卷积神经网络的预测结果基础上我们需要最终确定出有多少个奇异点，以及它们的类型和相应的像素坐标。我们通过下面三个步骤来输出最终结果。

2.4.1 步骤 1: 尽可能地去掉虚假预测

2.3.2中阈值 w 的取值对全卷积网络的奇异点区域预测结果影响很大。当阈值 w 取值较大时，预测出的奇异点区域的准确率也相对较高，但奇异点区域的漏提率也会变高。因此，选择一个合适的 w 至关重要。在我们的算法中， w 通过一个自适应策略来自动选取，其取值被定义为全图奇异点预测概率的最大值减去一个固定阈值 $\varepsilon=0.01$ 。

我们定义 SP 为全卷积神经网络预测输出为奇异点模式（中心点模式或三角点模式）的所有点的集合。 $f(P)$ 是全卷积神经网络预测 P 点为奇异点模式的概率输出，则 w 的计算公式如下：

$$w = \max\{f(P) \mid \forall P \in SP\} - \varepsilon, \quad (2.8)$$

经过全卷积神经网络的预测输出和自适应地选取 w 来作用预测结果之后，我们得到了一个更加精确的奇异点区域预测结果。对于此时的每一个预测为奇异点模式的点 P_i ，我们计算一个相应的值 NBP_i ，该值定义为以 P_i 为中心的一个窗口中，被预测为奇异点模式的点的个数，本章实验中使用的窗口大小为 5×5 。我们通过 NBP_i 的取值结果来进行进一步的筛选。对于每一个奇异点模式点 P_i ，如果 NBP_i 小于某个阈值 NBW （在本章实验中 NBW 取值为 6），则 P_i 被认为是

一个虚假预测结果并将该点去除。

2.4.2 步骤2: 确定三角点的位置

在第一步给出的奇异点区域预测结果的基础上, 这一步将计算获得当前指纹图像的三角点个数以及位置。具体操作如下:

(a) 定义 $DP := \{p \text{ 被全卷积神经网络预测为三角点模式} | \forall p \in SP\}$, 即 DP 为全卷积神经网络预测输出为三角点模式的所有点的集合。当 $n = \|DP\|$ 小于某个阈值, 即预测输出的奇异点区域很小很稀疏时, 我们判定在当前指纹图像中没有三角点。

(b) 计算点集 DP 的直径 d :

$$d = \max\{\|p_i - p_j\|_2 \mid \forall p_i, p_j \in DP\}, \quad (2.9)$$

其中 $\|p_i - p_j\|_2$ 是点 p_i 与 p_j 之间的欧氏距离。

如果该直径 d 小于某个阈值, 则我们判定当前指纹图像只有一个三角点。该三角点 $P_d(x, y)$ 位置的计算公式如下:

$$P_d(x, y) = \left(\frac{\sum_{i=1}^n x_i * v_i}{\sum_{i=1}^n v_i}, \frac{\sum_{i=1}^n y_i * v_i}{\sum_{i=1}^n v_i} \right). \quad (2.10)$$

其中 x_i 与 y_i 是集合 DP 中的点 $p_i(x_i, y_i) (i = 1 \dots n)$ 的坐标, $v_i = f(p_i)$ 是点 p_i 的预测概率值。

(c) 如果直径 d 大于阈值, 则判定当前指纹图像有两个三角点。通过 DP 直径的垂直平分线将 DP 分为两个子集合 DP_1 和 DP_2 , 由先验知识可知每个子集合仅包含一个三角点。通过 (b) 中方法即可计算得到各子集合所对应的三角点, 从而获得这两个三角点的位置信息。

2.4.3 步骤3: 确定中心点的位置

同理, 我们定义 CP 为全卷积神经网络预测输出为中心点模式的所有点的集合。当 CP 的直径大于阈值时, 我们可以判定存在两个中心点。但当 CP 的直径小于阈值时, 我们不能像判定三角点时那样判定当前指纹图像只存在一个中心点。这是由中心点与三角点的分布特性不一样导致的。在几乎所有存在两个三角点的指纹中, 两个三角点之间都存在着一定的距离。但是两个中心点之间就没有一个显著的距离间隔, 中心之间的距离在一些指纹中可以非常小, 从而导致全卷积神经网络输出的奇异点区域在某些一个中心点和两个中心点的情况下非常相似。

为了确定中心点的个数，我们对 CP 直径上所有点的概率输出进行统计观察，如图2.11所示是通过折线图来展示的结果。

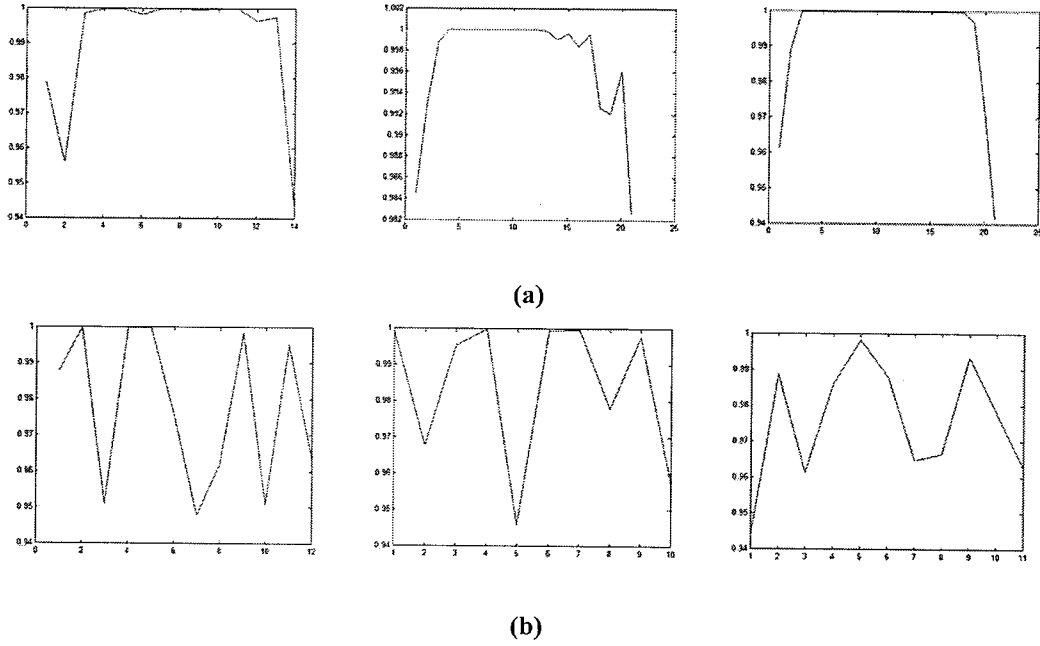


图 2.11 以从上到下的方向来画出 CP 直径上所有点的概率值 (a) 存在两个中心点的指纹图像画出的曲线 (b) 存在一个中心点的指纹图像画出的曲线

确定中心点个数的方法如下：

(a) 当 $m = \|CP\|$ 小于某个阈值，判定在当前指纹图像中没有中心点。

(b) 计算点集 CP 的直径 d ，如果直径 d 大于阈值，则判定当前指纹图像有两个中心点。中心点的位置计算方法与步骤 2 (c) 同理。

(c) 计算 $\xi = \sum_{i=1}^{m-1} |p_i - p_{i+1}|$ ，其中 p_i 是 CP 的直径上的点， p_i 到 p_{i+1} 的方向为图像坐标系中从上到下的方向。当 ξ 大于阈值时，判定只有一个中心点，否则为两个中心点。一个中心点的位置计算方法与步骤 2 (b) 同理，两个中心点的位置计算方法与步骤 2 (c) 同理。

2.5 实验结果及分析

实验中，对于不同的尺寸 (80×80 , 120×120 , 160×160)，我们均从 1 万 5 千幅指纹图像 (2.3.1) 中提取出了 4 万 8 千个指纹图像块，其中三种模式的图像块各占 1 万 8 千。其中非奇异点模式的图像块是随机挑选距离奇异点一定距离的点为中心生成的。对于每一类尺寸，我们挑选 4 万个图像块作为神经网络的训

练样本, 8 千个图像块作为神经网络的测试样本, 并保证在训练样本和测试样本中三种模式的样本数量均匀。对于整个算法在完整图像上的测试, 我们使用了 NIST-4 公开库 (Watson et al., 1992) 与 FVC2002 DB1 公开库 (Maio et al., 2002b)。NIST-4 为 2000 幅 512×512 的灰度图组成。FVC2002 DB1 则为 800 幅 388×387 的指纹图像。这两个库的指纹图像质量有高有低, 并包含了各种噪声, 以及诸如褶皱、刀疤、污迹、干燥、潮湿、模糊等情况。对这两个库的奇异点标签, 我们并不通过现有的自动特征提取算法获取, 而是通过人工标记获取, 从而保证了标签的准确性。对于每一个标签奇异点 $SP'(x', y', type')$ 和一个奇异点提取结果 $SP(x, y, type)$, 我们计算两点之间的距离 $Dis = \|SP' - SP\|_2$, 如果 $Dis < 6$ 并且 $type' = type$, 则我们称 SP' 与 SP 匹配, SP 是一个正确的提取结果, 否则, SP' 与 SP 不匹配, SP 是一个错误的提取结果。对于每个标签奇异点 SP' , 如果存在某个奇异点提取结果 SP 与其匹配, 则 SP' 被检测成功, 否则称 SP' 被漏提。

我们将本章提出的奇异点检测算法与文献 (L. Fan et al., 2008; J. Zhou, 2009; Chikkerur et al., 2005) 的方法做了对比。这些方法都是依赖于 Poincare 指数以及方向场的准确度。而由于传统的方向场计算方法在中低质量的指纹上表现不佳, 因此这些方法都受其影响而难以检测准确。由于我们的训练样本包含很多中低质量的样本, 因此经过充分训练后的神经网络在这些情况下表现相对良好很多。表2.1和表2.2给出各个算法性能的对比结果。从中可以看出, 本章提出的算法优于其他算法。无论是在奇异点提取结果的准确性还是真实奇异点的提取率上, 本章的算法都获得了更好的结果。我们认为如果使用更加准确的训练数据 (全人工标记), 则本章的奇异点提取算法将表现得更加优异。

表 2.1 各算法在 NIST-4 上的测试结果

方法	提取率 (%)		错误率 (%)	
	中心点	三角点	中心点	三角点
本章提出的算法	88.60	94.26	6.43	3.71
Fan's (L. Fan et al., 2008)	88.76	91.45	13.77	9.92
Zhou's (J. Zhou, 2009)	86.13	89.51	8.47	6.15
Chikkerur's (Chikkerur et al., 2005)	85.40	86.66	9.93	8.70

表 2.2 各算法在 FVC2002 DB1 上的测试结果

方法	提取率 (%)		错误率 (%)	
	中心点	三角点	中心点	三角点
本章提出的算法	95.39	98.26	1.03	4.10
Fan's (L. Fan et al., 2008)	95.64	96.95	1.88	10.84
Zhou's (J. Zhou, 2009)	95.78	96.98	2.27	9.97
Chikkerur' s (Chikkerur et al., 2005)	95.89	92.75	6.93	8.16

2.6 本章结论

本章提出一个全新的指纹图像奇异点提取算法,该方法融合了全卷积神经网络、概率模型以及多尺度策略,将奇异点检测问题转化为分类问题并用深度神经网络解决。训练了多尺度的卷积神经网络并将其转换为全卷积神经网络后再进行微调,最后在全卷积网络预测结果的基础上,使用了自适应和概率分布的分析策略进一步确定奇异点的个数与位置。在指纹公开数据库 NIST-4 和 FVC2002DB1 上的对比实验证明了该算法的有效性以及相对于传统算法的优越性。与传统的指纹奇异点提取算法不同,本章的方法不需要计算指纹方向场,也不需要任何图像预处理操作。此外,本章的算法还可以充分利用大规模的指纹数据库,通过内容更加丰富的数据来训练神经网络,将使得该方法的性能进一步提升。在后续的研究中,我们将分析各种尺寸的图像块对于神经网络的影响,从而更进一步提升多尺度的效果。其次,我们将寻求更高效更优异的方法来去除全卷积神经网络错误的预测结果,从而提升算法的精度。

第3章 小面积指纹识别系统

3.1 引言

小面积指纹识别算法主要应用于民用的移动设备当中，例如手机、U 盘以及笔记本电脑等，图3.1展示了一些目前市面上可见的实际产品。随着各种移动设备越来越普及，在这些设备上使用的小面积指纹识别算法得到越来越多的关注。例如苹果手机 Touch ID 技术和安卓系统中的指纹识别系统，为用户提供了安全方便又可靠的身份验证方式，在当今人们的生活中已经扮演着重要角色。大部分专业学者认为像指纹识别这样的生物特征识别方法将取代传统的密码。

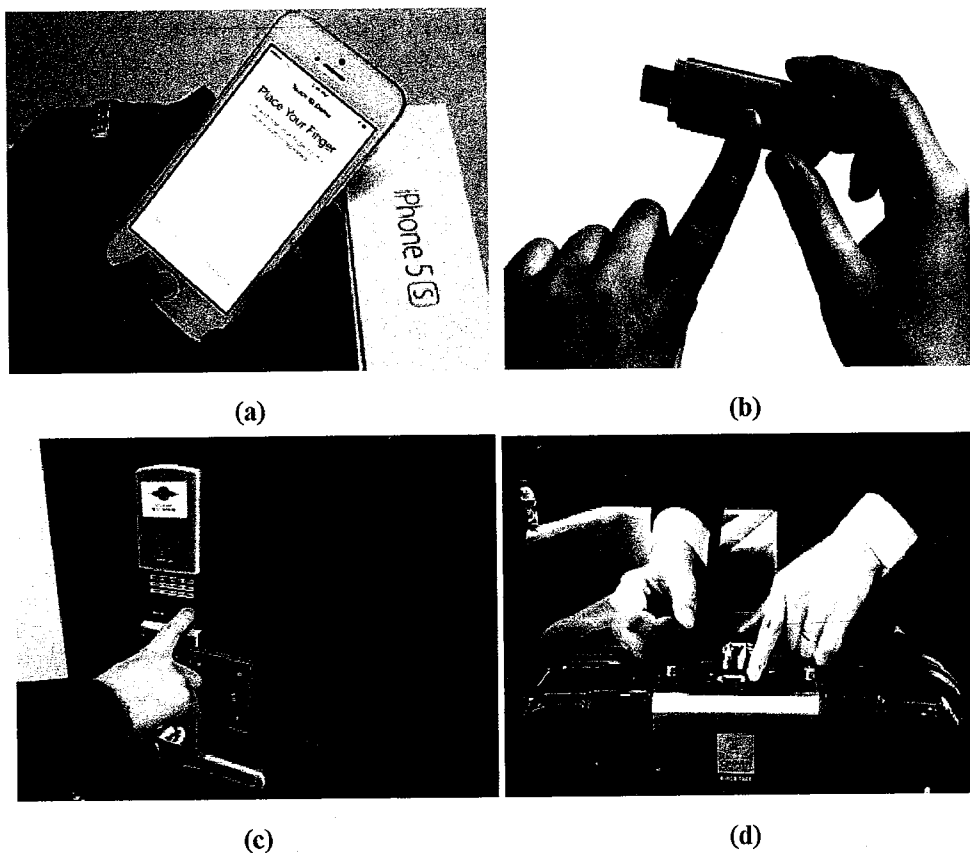


图 3.1 小面积指纹识别系统的实际应用案例：(a) 智能手机；(b) 指纹 U 盘；(c) 指纹门锁；(d) 指纹箱包

指纹识别模块的厂商为了降低成本，通常会将指纹传感器的面积尽可能地做小，在低成本的基础上，指纹传感器的性能也很一般，因此，在移动设备中的指纹传感器采集得到的指纹图像均是小面积且中等质量的指纹图像。虽然基于细节点的传统指纹识别算法已经得到了很广泛的研究并且很成熟，但是与传统

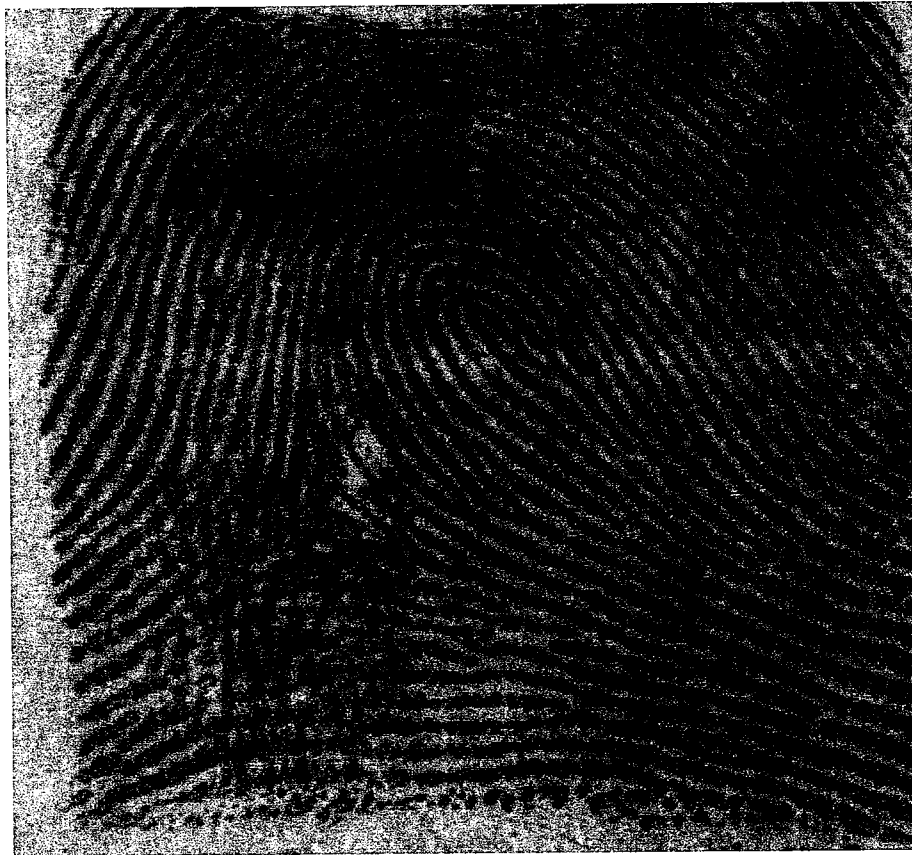
的指纹识别不同，小面积指纹识别算法无法使用基于细节节点的算法。这是由于小面积指纹图像中没有足够多的细节节点，如果强行使用基于细节节点的算法，将得到性能糟糕的结果。

图3.2给出了 500 Dpi 下各种尺寸的指纹图像。显然，随着指纹图像尺寸的降低，可使用的信息也随之减少。传统的基于细节节点的指纹匹配随着指纹图像尺寸的降低，性能骤降。事实上，当指纹图像的尺寸下降到 160×160 左右时，基于细节节点的算法性能已经难以满足需求。根据法医指南，当两枚指纹至少有 12 个细节节点匹配上时，才能将它们认定为来源于同一根手指 (Jain et al., 2009)。而当指纹图像的尺寸下降到 160×160 左右时，一幅指纹图像中的平均细节节点个数已经不到 20 个，考虑到匹配的非完全重合性，匹配上的平均细节节点个数将远达不到 12 个的判定标准。图3.3为一对 96×96 的小面积指纹图像匹配示意图，图中可以看到细节节点信息已经明显不足，在真实匹配情况下，重合区域中仅有 3 个细节节点，在这种情况下，基于细节节点的匹配方法将完全无法区分真假匹配。

目前，根据使用特征的不同，可以把已提出的小面积指纹匹配算法分为 5 类：

- (1) 基于细节节点特征的匹配算法；
- (2) 基于指纹第三级特征的匹配算法；
- (3) 基于图像像素相关的匹配算法；
- (4) 基于 SIFT 的匹配算法；
- (5) 基于纹线特征的匹配算法；

在本章中，我们将在第二节详细介绍这些目前主流的小面积指纹识别算法。在第三节中，我们将介绍小面积指纹识别系统与传统指纹识别系统的区别。并在第四节中介绍目前小面积指纹识别系统在实际运用中遇到的问题及可行解决方案。最后，我们在第 5 节中给出本章的总结。



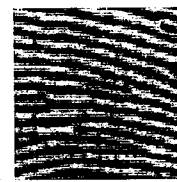
(a)



(b)



(c)



(d)

图 3.2 在 500 Dpi 下各种尺寸的指纹图像: (a) 512×512 (b) 300×300 (c) 160×160 (d) 96×96

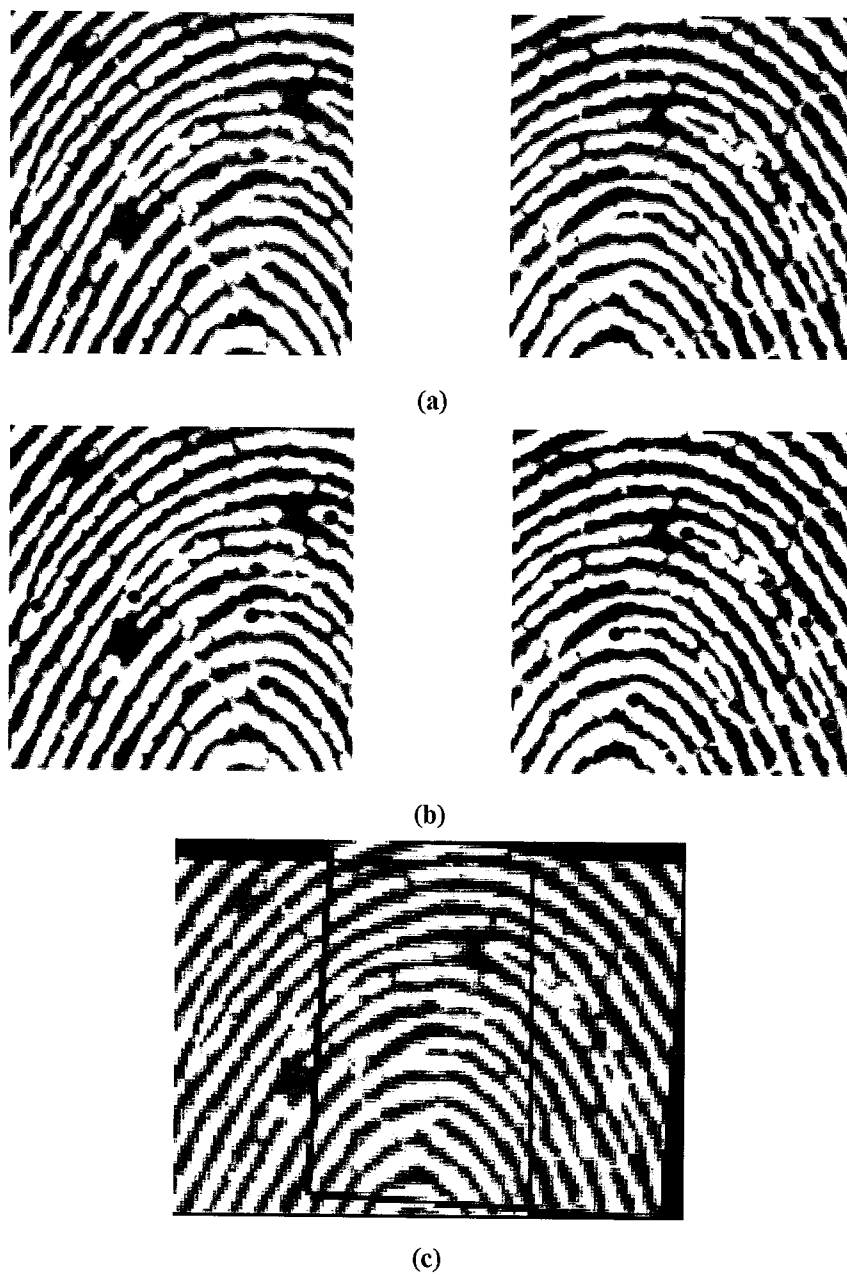


图 3.3 一对 96×96 的小面积指纹图像匹配示意图，其中红色点为重合区域的细节点，蓝色点为非重合区域细节点。(a) 原图；(b) 特征提取结果；(c) 正确匹配结果

3.2 小面积指纹的匹配算法

在本节中，我们将介绍目前主要的小面积指纹匹配算法，并对这些方法进行比较和分析。

(1) 基于细节点特征的匹配算法

仅仅只使用细节点信息已经无法提供足够的区分力，因此，一些学者尝试使用细节点周围的信息来进行补充。文献 (Jea et al., 2005) 提出使用以细节点为中心的局部二级特征来进行匹配，但其使用的信息几乎还是细节点信息。文献 (Fang et al., 2007) 提出了一种新的特征点，称为脊线表示点 (Representative Ridges Points, RRP)。该点定义为细节点沿着脊线按一定距离采样得到的点，由图3.4所示可以看出，一些细节点无法表示的区域中都含有脊线表示点，从而其提供了更多的信息来匹配。虽然该方法确实能弥补细节点信息的不足，但补充的信息量十分有限。此外，其本质上还是依赖于细节点来生成脊线表示点。在小面积指纹的匹配中，这类方法虽然可以提升传统的基于细节点的算法性能，但是最终还是无法达到满意的结果。

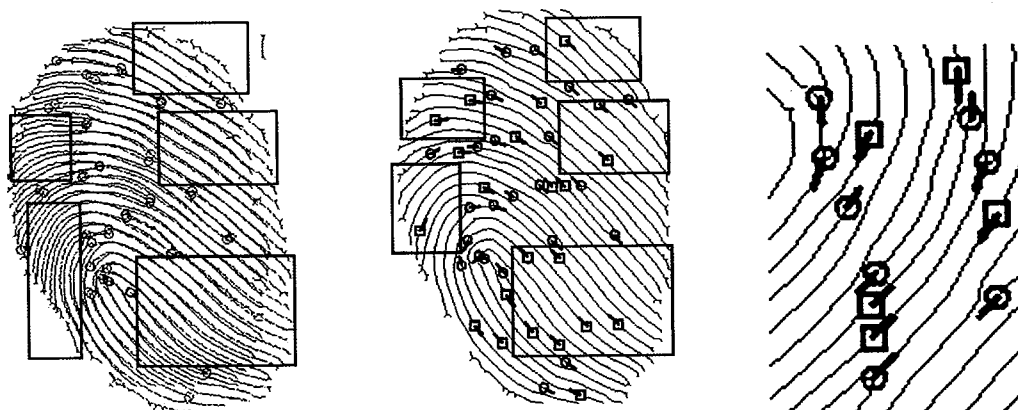


图 3.4 脊线表示点示例图，其中圆点代表细节点，方块点代表细节点生成的脊线表示点

(2) 基于指纹第三级特征的匹配算法

文献 (Chen et al., 2007) 提出使用指纹的三级特征（汗腺毛孔、短棒、突触等）来进行匹配识别。图3.5展示了指纹真皮照片和采集得到指纹图像中一些三级特征的实例。同细节点一样，三级特征也具有终身不变性，唯一性等特性。并且，三级特征相比于二级特征细节点来说，在小面积指纹中数量更多，拥有比细节点更丰富的信息量。文献 (Chen et al., 2007) 中实验结果也说明了该特征的有效性和可用性。但是，该方法需要指纹传感器能采集到高分辨率的指纹图像（至

少 1000 Dpi(Zhao et al., 2009)), 而这就需要使用成本极高的指纹传感器。在实际应用中, 使用这类方法获得的收益(匹配性能的提升)与成本(昂贵的指纹传感器)相比, 得不偿失, 因此, 到目前为止, 主流的小面积指纹识别系统中依旧没有使用该特征。

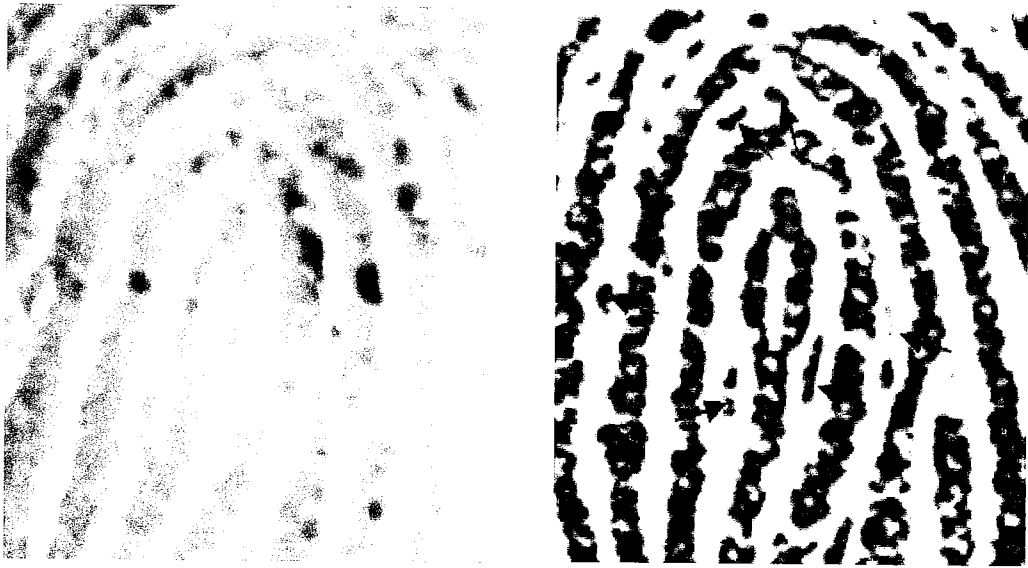


图 3.5 汗孔, 短棒等三级特征示例图

(3) 基于图像像素相关的匹配算法

这类方法来源于最早期的通用目标匹配算法, 这类方法 (Lindoso et al., 2007; Shabrina et al., 2016; Zanganeh et al., 2014) 使用最简单朴素的匹配思想, 穷举所有可能的刚性变换对齐参数, 在每一个刚性变换对齐参数下, 计算重叠区域中的像素相关性来估计当前的刚性变换参数的优劣。这类方法在空域上的实现存在着效率低下, 匹配慢的问题。后来的学者 (Zhang et al., 2006a) 提出了通过使用傅立叶变换来加速的策略, 在很大程度上提升了这类方法的匹配速度。这类方法对于无形变的物体识别来说一般具有很好的性能, 但在指纹匹配中由于非线性形变的普遍存在, 这类方法难以达到很好的性能。

(4) 基于 SIFT 的匹配算法

SIFT 特征 (Lowe, 2004) 是通用目标识别领域最知名的 2D 图像特征。该方法通过对图像的多尺度空间上检测到的极值点进行筛选, 从中选出非边缘高对比的点作为特征点。基于该特征的匹配算法已经成功应用于许多实际问题中, 并取得了很好的结果。文献 (Pankanti et al., 2008) 首先提出将 SIFT 特征和细节点特征相结合来进行指纹匹配的方法。文献 (Zhou et al., 2011) 跟进了这个方法的研究,

提出了两点改进,一是使用哈夫变换来替代原先的 SIFT 局部描述子匹配,二是在原始图像上进行了一些预处理之后再行多尺度空间的构建。两篇文献都没有在小面积指纹上进行测试。文献(Mathur S, 2016)使用了 KAZE 特征来进行小面积指纹的匹配。KAZE 特征是一种 SIFT 特征的改进版。与 SIFT 特征使用各向同性的高斯模糊生成多尺度空间不同,KAZE 特征的多尺度空间是通过各向异性的非线性模糊而构建的,KAZE 特征更加重视边缘的保持和噪声的过滤。文献(Mathur S, 2016)的实验结果表明对与指纹识别来说 KAZE 特征比 SIFT 特征更加优秀。这类算法是目前手机上的小面积指纹识别系统主要使用的算法,但在一些资源配置较低的移动设备上,这类算法不可用。这是因为这类算法需要大量的内存空间来存储图像的多尺度空间,而且由于每一个特征点都附加了一个描述子,导致了这类方法提取一幅小面积指纹图像得到的特征文件也占了很大的内存。

(5) 基于纹线特征的匹配算法

文献(Yager et al., 2004)提出了一种通过指纹方向场来进行匹配的方法,该方法能较为粗略地对齐两幅质量较差的小面积指纹图像,但该方法区分能力较差,匹配速度一般。文献(Marana et al., 2005)提出使用指纹的脊线来进行匹配的方法,其匹配方法使用哈夫变换来将脊线点匹配转换为脊线点所在直线的匹配。其实验结果表明了指纹脊线拥有可以挖掘利用的大量信息。但其匹配方法太过于简单,计算量巨大,不实用。文献(Ross et al., 2003)中使用了脊线特征图来进行匹配,其脊线特征图是在增强后的图上进行 8 个方向的 Gabor 滤波并下采样得到的 8 个特征图。在该文献中,脊线特征图只是指纹脊线信息较为宏观的一种表示,其丢失了脊线大部分的信息,在匹配过程中也还是以细节点的方法为主而以脊线特征图匹配的方法为辅。

3.3 小面积指纹识别系统

小面积指纹识别系统与传统的指纹识别系统有很多不同,其中最大的一个不同点是用户的模版特征不同。在传统的指纹识别系统中,用户注册时只需要用手指按压一次得到一幅指纹图像,通过提取该指纹图像的特征作为该用户的一个模版特征。而在小面积指纹识别系统中,用户注册时需要用同一根手指按压多次来获得多幅小指纹图像,该用户的模版特征是由这些小指纹图像的特征融合而成。这是因为一幅大面积的指纹图像即可包含一根手指指纹的大部分信息,而小面积指纹则不能。并且如果在小面积指纹识别系统注册时按压次数过少,则验证过程中待验证指纹的信息可能并不在注册模版中,从而导致较高的拒

识。图3.6与图3.7分别为两种系统中的注册流程示例图，可以看到，小面积指纹识别系统的注册流程要复杂得多，其目的是为了提升在使用过程中用户的体验（拒识低则体验好）。下面我们将详细地介绍小面积指纹系统中的注册策略。

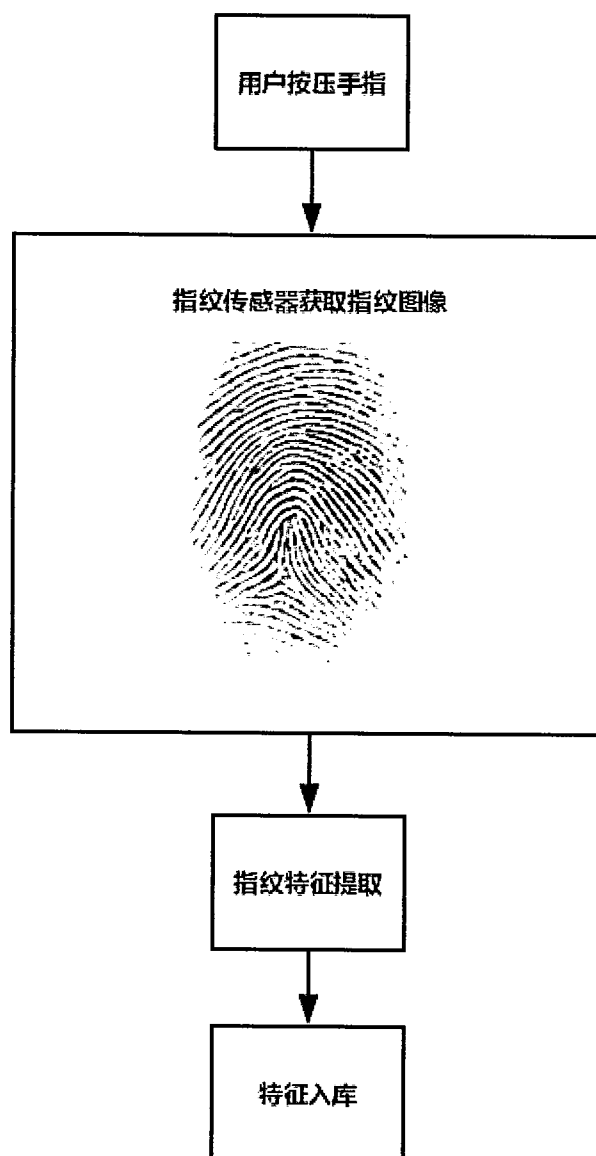


图 3.6 传统指纹识别系统注册流程示例图

3.3.1 注册策略

注册的目的是为了得到用户指纹的信息，该信息用于验证和识别阶段。如果该信息不能完整准确地表示用户指纹，则将严重影响用户在验证和识别时的体

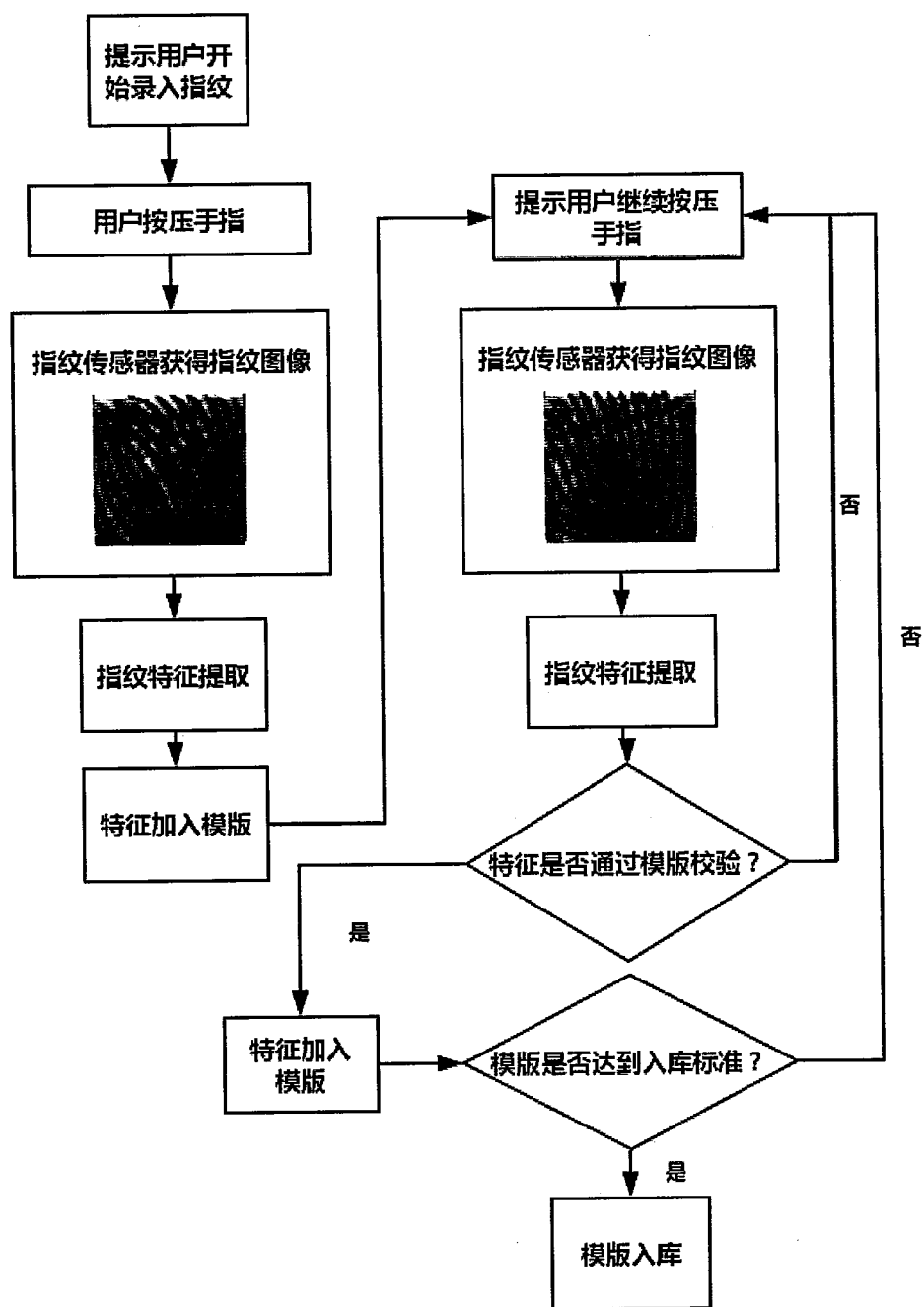


图 3.7 小面积指纹识别系统注册流程示例图

验，具体表现为已注册的用户较难通过验证。一幅小面积指纹图像只能包含指纹中的一小部分信息，因此，为了能够包含一枚指纹的所有信息，需要多幅小面积指纹图像。图3.8为一根手指按压8次采集得到的8幅小面积指纹图像，可以看到，其中的第二幅与第七幅具有很高的重叠比例，如果将它们都注册到特征模版中，就导致特征模版中包含了过多的冗余信息。这个原因是由于用户难以精确感知自己按压的部位所引起的，在多次按压采集，用户几乎不可能控制按压采集在指纹的不同部位。

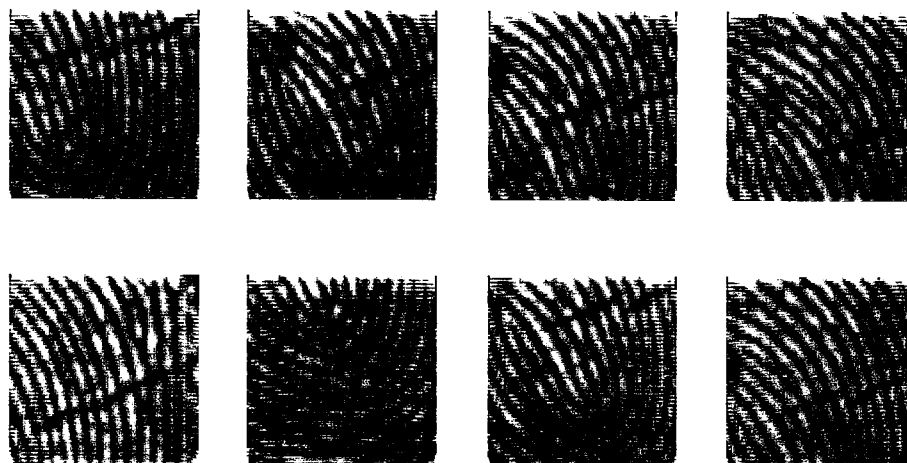


图 3.8 用一根手指按压多次采集的小面积指纹图像

小面积指纹识别系统中的注册一般需要用户手指按压 N 次，如果不添加任何交互提示和控制，则很大可能用户按压的 N 次都采集到指纹的同一个部位，这就没有达到注册的目的，导致用户后续使用体验差。如果 N 过大，使得用户需要许多次按压，又使得用户注册体验不佳，因此，如何通过注册过程中的自动判断和交互提示来使得用户尽可能按压次数少的情况下又能将手指信息充分录入是小面积指纹注册策略面临的主要问题。

图3.7为目前小面积指纹识别系统的一个注册流程图。其中最为关键的一步为：特征是否通过模版校验。这一步的目的是通过已经添加的指纹图像来判断当前的指纹图像是否有添加的价值。一个简单的策略是将当前的指纹图像与已经添加的指纹图像逐一进行匹配，判断是否存在重叠比例很高的真匹配，如果存在，意味着当前的指纹图像并没有提供足够的信息，从而没有添加的必要，然后提升用户挪动手指部位重新按压手指。使用该策略需要注意一个问题，在注册

的前几次按压中,由于已添加的信息较少,用户的按压采集图像较容易通过该校验,但当添加次数较多后,已经添加的信息已经足够多,此时可能出现用户的按压采集图像很难通过校验的情况。因此,这一步的判断中需要以再添加一个判断当前已添加的信息是否已经足够的判断条件,如果已经足够,则可通过校验。图3.9给出了一个基于 SIFT 算法的小面积指纹拼接实例,通过该方法即可获得注册过程中已添加指纹所包含的信息,通过该信息量即可判断已添加的信息是否足够,以及最终的模版是否达到入库标准。

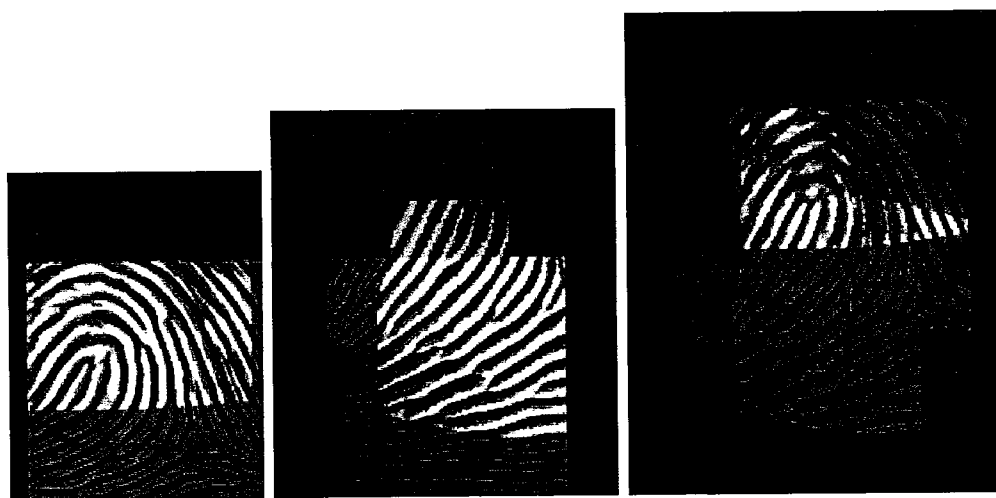


图 3.9 拼接注册过程中录入的小面积指纹示例图

目前的小面积指纹识别系统的注册流程大致相似,但由于所使用的核心匹配算法不同,其注册策略中的细节处理也会有所不同。例如一些算法有特征融合的处理,有效的特征融合可以降低最终融合模版特征的冗余信息。

3.3.2 自学习策略

在一般的指纹识别系统中,用户注册录入的指纹特征信息是一个不可修改的数据。在后续的验证与识别过程中也仅使用该数据进行匹配。这种方式存在一个问题,如果用户录入的指纹较差(手指干燥或潮湿、手指上有灰尘等),则在后续的使用中将导致糟糕的体验效果。这个问题在小面积指纹识别系统中表现的尤为明显。在小面积指纹的注册时需要录入多幅小面积指纹图像,并且需要这些小面积指纹图像尽可能地覆盖到整个手指指纹。为了使注册的信息完整可靠,就需要较为复杂的注册策略以及和用户进行人机交互,很大程度上需要用户的仔细配合。这就使得注册过程对用户要求偏高,导致用户注册体验较差。但如果仅使用简单的注册策略以及较弱的交互和用户配合,则注册的特征信息在后续

使用中又很大可能导致体验不佳。因此，在小面积指纹识别系统中，仅凭注册策略难以做到用户操作使用简单的同时又能得到很好的性能体验。

为了解决上述所说的易用性与准确性互斥的问题，目前的做法是使用一种叫作自学习的策略。该方法通过在用户后续的验证使用中动态地更新数据库中用户对应的指纹特征信息来使得该指纹特征信息包含越来越多的用户手指信息或使得信息越来越可靠。简而言之，就是越用越好用。图3.10给出了一个自学习策略的简要流程图。在学习策略中，最为关键的部分是如何更新模版特征。这与核心的匹配算法相关，基于不同特征和不同匹配方法的算法在这个步骤都要单独考虑，一个简单的做法是直接使用注册时的特征融合方法。

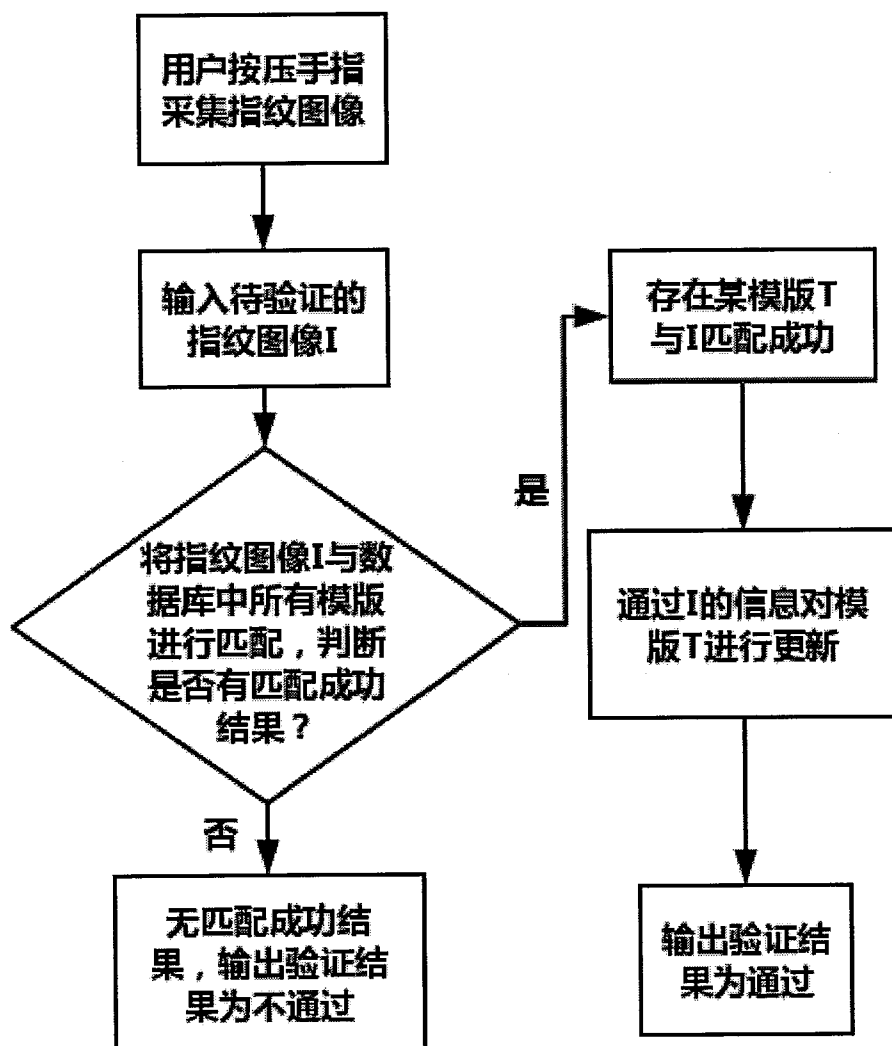


图 3.10 自学习策略流程示例图

3.4 可能遇到的问题及解决方案

自学习策略的运用使得小面积指纹识别系统的用户体验大大提升,尤其是智能手机上小面积指纹识别系统,在实测情况下达到了十万分之一的 FAR 和百分之一以下的 FRR 这样的高性能。但自学习策略也随之带来了一个问题。

2018 年初国内某指纹厂商爆料,宣称用“透明胶带+导电笔,秒破安卓和苹果的指纹识别,甚至于橘子皮都能打开”(今日头条,2018),在此之后,媒体连续报道了智能手机上的指纹识别功能设计缺陷,苹果、华为、OPPO、vivo 等主流品牌手机可通过在指纹传感器上“贴膜”实现使用未录入的手指指纹进行解锁,由于上述攻击的技术难度、攻击成本都很低,引起了广泛关注。

事实上,从算法原理上看,这个问题正是由于自学习策略与基于图像识别的匹配算法导致的。首先,除了基于细节点与指纹纹线的匹配算法,其他的小面积指纹匹配算法都不会关心所处理的图像到底是否是指纹图像,也就是说,输入的图像可以是任意带有内容的图像。其次,自学习策略使得这些非指纹的攻击图像所带有的信息有可能被学习进入用户的模版特征中。正是利用目前小面积指纹识别系统中的这两点缺陷,导致了报道中的手机指纹识别被破解这一事件,该攻击/破解流程如下:

(1) 用导电笔和胶布欺骗电容式的指纹传感器,让指纹传感器可以成功采集到胶布上的任意图案;

(2) 真实用户按压手指时,部分手指图像和注册的图像吻合,手机匹配算法就判定为同一手指,同时因为自学习机制起作用,把胶布上的图案当作该手指的一部分学进了用户模版信息中;如此重复按压几次,胶布上的图案就作为一个稳定图像/图像特征存储在模板中。

(3) 后续用其他手指,甚至于橘子皮按压,由于指纹传感器依然采集到胶布上的图案,此时胶布上的图案和已经学习进模板的胶布图案匹配成功,因此任意指纹都可以进入手机,至此整个攻击过程完成。

除了算法原理上的漏洞,导致这个问题的其他原因也有很多。一般而言,生物特征识别技术架构涵盖图像采集、图像质量判断、特征提取、存储、比对等流程。但是在目前的移动设备指纹识别实际应用中,厂商只注重用户体验和设备成本,更多关注如何保证指纹识别效率的同时压低成本,对于图像质量判断、特征提取、存储、比对等流程处理缺乏重视,技术实现不完整,流程不规范,形成安全漏洞,给恶意攻击提供了较大空间。

从算法原来角度出发,应对该问题的方案如下:

(1) 算法添加能够自动识别输入图像是否为指纹图像的功能。

(2) 自学习策略中添加检测指纹连贯性的功能，因为攻击图像一般很难做到与真实指纹纹线信息保持纹线的连贯性。

3.5 本章结论

本章介绍了目前主要的一些小面积指纹匹配算法，这些匹配算法使用的特征各不相同，各项评估指标也各有利弊，并没有一个算法相对于其他方法表现出明显的优势。因此，小面积指纹识别算法的研究正处在一个发展阶段，需要更多的研究或提出新的算法才能使得该问题得到更好的决解。其次，本章讨论了小面积指纹识别系统与传统识别系统的区别，并详细地介绍了小面积指纹识别系统的特点，其中，在智能手机上出现的“被破解”问题也在本章被详细讨论并给出了应对策略。事实上，安全是一个相对的概念，“世界上没有攻不破的盾，也没有不可抵挡的矛”，主要是以攻击成本为价值衡量的。如同传统门锁一样，世界上没有打不开的锁，只是复杂程度以及支付的成本不同而已，指纹识别也是一样，只是科技的进步让用户尽享便利，同时让攻击变得越来越难，成本越来越高。即便如此，我们依然呼吁指纹领域的从业者不断提高技术水平防范各种安全漏洞。

第4章 基于脊线特征与改进的广义哈夫变换的小面积指纹识别算法

4.1 引言

在上一个章节中，我们介绍了小面积指纹识别算法的重要意义，以及目前一些主要的小面积指纹匹配算法。本章将提出一个全新的小面积指纹识别算法，该算法在基于脊线特征的基础上，使用针对小面积指纹识别问题而改进的广义哈夫变换进行两枚小面积指纹的对齐，最后通过机器学习的方法进行综合评估并给出最终匹配分数。实验结果表明该算法在性能上表现优异，内存开销低，速度满足实时要求，是一套可以实际应用的小面积指纹匹配算法。

本章的第二节将详细介绍所提出的算法，之后，实验结果及分析在第三节中给出。第四节我们给出了基于该算法基础上的特征拼接算法，使其应用在注册与自学习策略当中。在第五节给出本章结论。

4.2 基于脊线与广义哈夫变换的指纹匹配算法

在这个小节中，我们将提出一种基于脊线特征和改进的广义哈夫变换（Modified Generalized Hough Transform, MGHT）的指纹匹配算法。该算法主要分为三个步骤：(1) 指纹脊线特征提取；(2) 基于改进的广义哈夫变换的指纹对齐；(3) 基于机器学习的相似度评估。该算法的主要目的是为了提升小面积指纹识别系统中匹配准确率，特别是在缺失细节点的情况下。图4.1是所提出算法的一个大致框架，下面我们将逐一详细介绍算法的各个步骤。

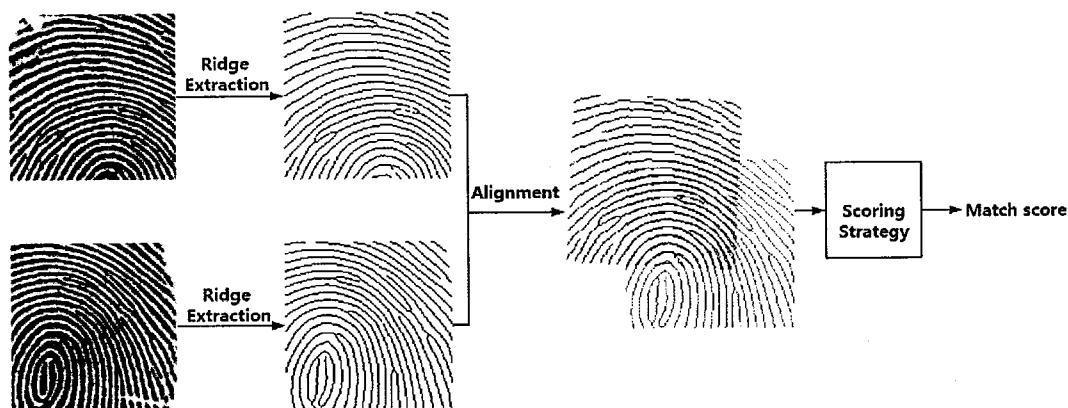


图 4.1 本章算法的一个框架示例图

4.2.1 指纹脊线提取

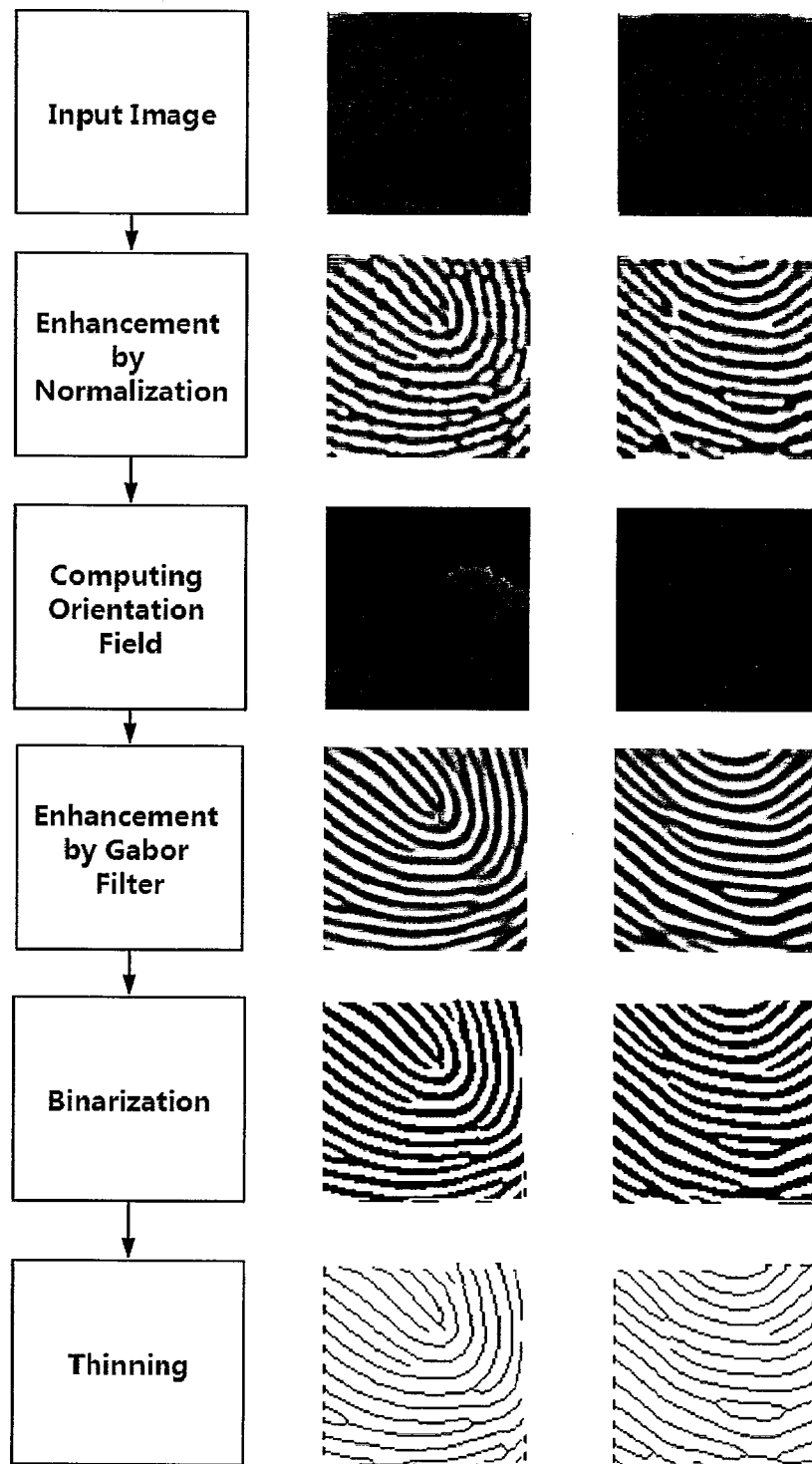


图 4.2 本章脊线特征提取流程的示例图

与细节点特征在基于细节点的匹配算法中的地位一样，指纹脊线作为本章所提的匹配算法所使用的主要特征，在整个算法体系中起着至关重要的作用。可

靠准确的脊线特征提取结果才能使最终的算法性能表现优异。本章使用的脊线特征提取方法以指纹原图作为输入,经过一些列处理之后输出该指纹图像的脊线点集。该脊线点集中的每一个点包括五个特征属性: X-轴坐标, Y-轴坐标, 该点处的脊线方向、脊线频率以及曲率。图4.2给出了该脊线特征提取流程的一个示例图。

脊线特征提取算法的主要步骤如下:

(1) 指纹图像归一化增强 (Enhancement by Normalization): 首先, 输入的指纹原图的灰度值分布千差万别, 为了便于后续处理, 需要进行一个标准化的操作使得指纹图像的灰度值都变成几乎相同的分布。如图4.2中前两行所示, 对输入指纹原图 (Input Image) 进行直方图均衡化和灰度值归一化后, 指纹图像的质量得到改进, 指纹的脊线和谷线变得更加清晰明显。

(2) 方向场估计 (Computing Orientation Field): 指纹的方向场与频率场将在这个步骤被计算出。方向场通过梯度法 (Jain et al., 1997) 计算得到。在方向场的基础上通过投影法 (Hong et al., 1998) 计算得到频率场。这一步得到的方向场与频率场将被保存到后, 用于赋值脊线点的方向和频率。

(3) Gabor 滤波增强 (Enhancement by Gabor Filter): 使用 (2) 中获得的方向场和频率场, 构造 Gabor 滤波对当前的指纹图像进行增强 (Greenberg et al., 2002)。

(4) 二值化 (Binarization): 二值化的阈值是通过频率分析的自适应取值 (Bartunek et al., 2006)。

(5) 细化 (Thinning): 本章所使用的细化方法来源于文献 (Ji et al., 2007)。在获得脊线图之后, 继续对每个脊线点计算其曲率 (Biswas et al., 2008)。

通过以上 5 个步骤之后, 我们即可获得指纹的脊线点集及其特征属性。为了充分利用可获得的信息, 我们使用文献 (Farina et al., 1999) 的细节点提取方法在 (5) 之后进行了细节点提取, 并在后续的匹配算法中使用细节点信息。虽然在本章的对齐算法使用到了细节点信息, 但细节点信息并不是算法的必要信息。事实上, 这里提取的细节点集可以是空集。

4.2.2 基于广义哈夫变换的对齐

这一小节将介绍如何使用上一小节中提取得到的特征来进行两枚指纹的匹配对齐。在对齐过程中, 我们假设两枚指纹之间存在着一个最优的刚性变换关系 (旋转和平移), 在该变换关系下, 两枚指纹之间的脊线点集和细节点集 (可以是空集) 达到最佳的一一对应关系。给定两幅小面积指纹图像, 通过4.2.1小节的特

算法 1 基于改进的广义哈夫变换的对齐匹配算法

输入: $\{p^T\} = \{(p_x^T, p_y^T, p_\theta^T, p_\rho^T, p_c^T)\} \in \mathbb{P}^T$; $\{p^Q\} = \{(p_x^Q, p_y^Q, p_\theta^Q, p_\rho^Q, p_c^Q)\} \in \mathbb{P}^Q$;
 $\{m^T\} = \{(m_x^T, m_y^T, m_\theta^T)\} \in \mathbb{M}^T$; $\{m^Q\} = \{(m_x^Q, m_y^Q, m_\theta^Q)\} \in \mathbb{M}^Q$; 角度缩放因子:
 $SC_a (= 4)$; 尺度缩放因子: $SC_s (= 2)$; 细节点权重: W_m ; 粗候选个数: CN

输出: 查询指纹映射到模版指纹的最佳刚性变换 T

- 1: 计算模版指纹的参考点 $O^T (\frac{\sum p_x^T}{Num(\mathbb{P}^T)}, \frac{\sum p_y^T}{Num(\mathbb{P}^T)}, 0)$;
- 2: 使用模版脊线点集构建第 1 级 R 表和距离场 DMT ;
- 3: 使用查询脊线点集构建距离场 DMQ , 初始化第 1 级投票空间 A_1 ;
- 4: **细节点投票**, 详见算法 2;
- 5: **脊线点集粗投票**, 详见算法 3;
- 6: 对第 1 级投票空间 A_1 进行高斯平滑;
- 7: 遍历 A_1 , 找到最大值 $MaxVotes$, 取出 A_1 中前 CN 个值最大的单元作为查
选指纹参考点 O^Q 的候选, 如果某单元的值小于 $\frac{MaxVotes}{2}$, 则排除该单元。
- 8: **脊线点集精投票**, 详见算法 4
- 9: **最优估计**, 详见算法 5。

return T .

算法 2 细节点投票

for all 可能匹配的细节点对 m^T, m^Q **do**

 计算与 O^T 对应的参考点 O^Q 的角度和位置, 并在投票空间中头票。

$$O_\theta^Q = \lfloor \frac{m_\theta^Q - m_\theta^T}{SC_a} \rfloor$$

$$\begin{pmatrix} O_x^Q \\ O_y^Q \end{pmatrix} = \begin{pmatrix} O_x^T - m_x^T \\ O_y^T - m_y^T \end{pmatrix} \cdot \begin{pmatrix} \cos(O_\theta^Q) & \sin(O_\theta^Q) \\ -\sin(O_\theta^Q) & \cos(O_\theta^Q) \end{pmatrix} + \begin{pmatrix} m_x^Q \\ m_y^Q \end{pmatrix} \quad (4.1)$$

$$A(O_x^Q, O_y^Q, O_\theta^Q) = A(O_x^Q, O_y^Q, O_\theta^Q) + W_m$$

end for m^T, m^Q

算法3 脊线点集粗投票

for each 查询指纹脊线点 $p^Q(p_x^Q, p_y^Q, p_\theta^Q, p_\rho^Q, p_c^Q)$ **do**

$$\theta^c = \lfloor \frac{p_\theta^Q}{SC_\theta} \rfloor$$

for all 可能的索引方向 $\Delta\theta$ **in** 第1级R表 **do**

for all 偏移向量 $\mathbf{r}(r_x, r_y, r_\rho, r_c)$ **in** 第1级R表 **do**

if $|r_\rho - p_\theta^Q| > \text{Maximum_period_diff}$ **or**

$|r_c - p_c^Q| > \text{Maximum_curvature_diff}$ **then**

Continue

end if

计算与 O^T 对应的参考点 O^Q 。

$$O_x^Q = \lfloor \frac{p_x^Q}{SC_x} \rfloor + r_x, O_y^Q = \lfloor \frac{p_y^Q}{SC_y} \rfloor + r_y, O_\theta^Q = \theta^c + \Delta\theta \quad (4.2)$$

if $|(O_x^Q)^2 + (O_y^Q)^2| < \text{Maximum_offset}$ **then**

在第1级投票空间 A_1 中 O^Q 所对应的单元上进行投票。

$$A(O_x^Q, O_y^Q, O_\theta^Q) = A(O_x^Q, O_y^Q, O_\theta^Q) + 1$$

end if

$$O_x^Q = \lfloor \frac{p_x^Q}{SC_x} \rfloor - r_x, O_y^Q = \lfloor \frac{p_y^Q}{SC_y} \rfloor - r_y, O_\theta^Q = \theta^c + \Delta\theta + 180^\circ \quad (4.3)$$

if $|(O_x^Q)^2 + (O_y^Q)^2| < \text{Maximum_offset}$ **then**

在第1级投票空间 A_1 中 O^Q 所对应的单元上进行投票。

$$A(O_x^Q, O_y^Q, O_\theta^Q) = A(O_x^Q, O_y^Q, O_\theta^Q) + 1$$

end if

end for r

end for $\Delta\theta$

end for p^Q

算法 4 脊线点集精投票

for each 查选指纹参考点的候选 O^Q **do**

8.1: 使用模版指纹脊线点集构建角度范围为 $(O_\theta^Q * SC_a - SC_a, O_\theta^Q * SC_a + SC_a)$ 的第 2 级 R 表;

8.2: 初始化第 2 级投票空间 A_2 , 其中心位置代表 O^Q , 大小为 $(2SC_s + 1) \times (2SC_s + 1) \times (2SC_a + 1)$;

8.3: 使用第 2 级 R 表和查选指纹脊线点集在 A_2 上进行投票, 投票方法与脊线点集粗投票同理;

8.4: 遍历 A_2 , 找到最大值对应位置, 记为 O_{fine}^Q 并保存。

end for O^Q

征提取获得脊线点集和细节点集之后, 如何求解获得最优刚性变换是本节的主要问题。我们提出一种改进的广义哈夫变换方法来对这个问题进行求解。

广义哈夫变换 (Generalized Hough Transform) 是由 Ballard (Ballard, 1987) 首先提出的。在此之前, 哈夫变换已经被广泛使用来检测图像中的直线和圆等可解析表达的图形。为了检测任意形状的图形, 广义哈夫变换应运而生。该方法通过使用图形的边界和边界上的梯度来对图形进行表示, 将这种表示称为 R 表, 在匹配时通过 R 表中的记录在变换空间中的投票, 最后找出在变换空间中得票数最高的单元, 该得票数即为匹配的相似度, 该单元对应的变换即为这个匹配下的模版与检测物体之间的变换关系。该方法在识别任意形状图形的问题中表现出非常优异的效果, 特别是其在噪声、遮挡以及多物体检测等困难复杂的情况下也具有较好的适应性。但是, 广义哈夫变换也有其缺点。广义哈夫变换的本质思想是穷举所有可能的对应关系, 当需要角度旋转、缩放等因素时, 投票空间的内存开销变得非常巨大。当模版的 R 表较大, 并且在待检测图中的边界点也较多时, 算法的计算量大大增加。因此, 在很多情况下, 广义哈夫变换是一个时间和空间都有较大开销的算法, 这使得其在很多需要满足实时性的系统中无法使用。

如果使用传统的广义哈夫变换来求解本节的问题, 将使得本章的算法成为一个耗时且耗费空间的算法, 从而很难在实际应用中使用该算法。在本章的实验中, 我们所使用的小面积指纹图像大小均为 160×160 , Dpi 均为 500。对于处理这个尺寸的图像, 传统的广义哈夫变换需要申请 14.83 MB 的内存空间来存放刚

算法 5 最优估计

初始化 $S_{Max} = 0, O_{final}^Q = (0, 0, 0)$.

for each 候选参考点 O_{fine}^Q **do**

通过 O^T 和 O_{fine}^Q 计算查询指纹与模版指纹之间的刚性变换 $T(O^T, O_{fine}^Q)$

初始化 $S_q = S_t = N_q = N_t = 0$.

for each 查询指纹脊线点 $p^Q(p_x^Q, p_y^Q, p_\theta^Q, p_\rho^Q, p_c^Q)$ **do**

计算 p^Q 经变换 $T(O^T, O_{fine}^Q)$ 映射之后, 在模版指纹中的位置 $\bar{p}^Q$.

if $\bar{p}^Q$ 在模版指纹内 **then**

$$S_q = S_q + DMT(\bar{p}_x^Q, \bar{p}_y^Q), N_q = N_q + 1$$

end if

end for p^Q

for each 模版指纹脊线点 $p^T(p_x^T, p_y^T, p_\theta^T, p_\rho^T, p_c^T)$ **do**

计算 p^T 经变换 $T(O^T, O_{fine}^Q)$ 映射之后, 在查询指纹中的位置 $\bar{p}^T$.

if $\bar{p}^T$ 在查询指纹内 **then**

$$S_t = S_t + DMQ(\bar{p}_x^T, \bar{p}_y^T), N_t = N_t + 1$$

end if

end for p^T

$$S_m = (\frac{S_t}{N_t} + \frac{S_q}{N_q})/2$$

if $S_m > S_{Max}$ **then**

$$S_{Max} = S_m, O_{final}^Q = O_{fine}^Q, T = T(O^T, O_{fine}^Q)$$

end if

end for O_{fine}^Q

return T .

性变换关系的投票空间¹，同时需要至少 4.34 MB 的内存空间来存放 R 表²。而如果使用本章提出的改进的广义哈夫变换，投票空间与 R 表仅仅分别需要 293 KB 和 180 KB。在内存空间的占用上，改进的广义哈夫变换仅仅是传统的广义哈夫变换的 5.68%³此外，在 1 对 1 匹配速度上，改进的广义哈夫变换比传统的广义哈夫变换快将近 20 倍。

在两枚指纹的匹配中，我们称用户事先注册录入的指纹为模版指纹，当前需要验证身份的输入指纹为查询指纹。假设模版指纹的脊线点集为 $\mathbb{P}^T$ ，细节点集为 $\mathbb{M}^T$ ，即 $\mathbb{P}^T$ 、 $\mathbb{M}^T$ 为模版指纹经过 4.2.1 节的特征提取之后获得的特征。同样的，假设查询指纹的脊线点集和细节点集分别为 $\mathbb{P}^Q$ 和 $\mathbb{M}^Q$ 。算法 1 为本章提出的对齐算法的详细描述。对于脊线点集 $\mathbb{P}^T$ 和 $\mathbb{P}^Q$ ，所有可能的匹配关系都将被考虑，每一种可能的匹配关系对应一个变换参数，将求解最优变换关系问题转换为在变换参数空间搜索极值的问题。如图 4.3 和表 4.1 所示，模版脊线点所提供的信息将被以角度索引的方式存储在 R 表中。给定模版指纹的参考点 O^T ，对每一个脊线点 $P_i^T(x, y, \theta)$ ，计算偏移向量 $r_i = O^T(x, y) - P_i^T(x, y)$ 。将偏移向量 r_i 存储为以角度为索引的离散函数 Φ 。其中每个 Φ 的索引 R_Φ 可以有多个偏移向量 r_i 与之对应。对每一个脊线点 P_i^T 都进行这样的操作之后，R 表就以另一种方式完整地表示了模版指纹。与模版指纹的参考点 O^T 对应的查询指纹参考点记为 O^Q ，如果在查询指纹中确定了 O^Q 的位置和方向，则等价于知道了模版指纹与查询指纹间的刚性变换关系。 O^Q 的参数投票空间 A 将被用于记录所有可能的 O^Q 所在位置和角度。一般来说， O^Q 可能所在位置和角度将被量化为离散值，这里 O^Q 的位置和角度信息通过一个三维向量描述，因此投票空间 A 是一个三维矩阵，其中前两维代表位置信息，第三维代表角度信息。投票空间 A 中的每一个单元对应着 O^Q 在查询指纹中可能的一个位置和角度。对每一个查询指纹中的脊线点 P_j^Q ，在 R 表中按照角度 $\theta_j^Q = \theta_i^T$ 索引到所有对应的偏移向量 r_i 。则脊线点 P_j^Q 与这些偏移向量可计算得到当前情况下所有可能的 O^Q ，即 $O^Q = P_j^Q + r_i$ ，对投票空间 A 中 $P_j^Q + r_i$ 所对应的单元投票，即将该单元内的计数加 1。对所有的查询脊线点 P_j^Q 都进行这样的操作之后， O^Q 所有可能的情况即都被考虑到。最后在投

¹ $2 \times (160 + 20) \times (160 + 20) \times 240 \approx 14.83 \text{ Mb}$ ，每个投票单元需要 2 个字节，位置维度需要向外延伸 20 个单元，本章算法中将 0 到 360 度量化为 0 到 240。

² $3000 \times 240 \times 6 + 240 \times 240 \times 4 \approx 4.34 \text{ MB}$ ，3000 是 160×160 的指纹图像脊线点的平均个数，每个 R 表单元需要 6 个字节，每个角度的索引单元需要 4 个字节（其中 2 个字节对应 R 表中单元存储位置的开销，2 个字节对应个数）。

³ 在改进的广义哈夫变换中，粗投票占用内存最大，但也仅仅使用 $2 \times \frac{160+20}{2} \times \frac{160+20}{2} \times \frac{240}{4} + \frac{3000}{2} \times \frac{240}{4} \times 4 + \frac{240}{4} \times \frac{240}{4} \times 4 \approx 1.08 \text{ Mb}$ 的内存空间，在精投票阶段使用的内存相对来说可以忽略不计。

票空间 A 中找到票数最多的单元，该单元对应的位置和角度即为最优 O^Q 。这里需要注意，与细节点不同，脊线点没有极性（角度为 0 到 180 度而不是 0 到 360 度），因此，每一对脊线点之间进行匹配投票时，将在两个不同的位置分别投出一票，如同等式 4.2 和等式 4.3 所描述。

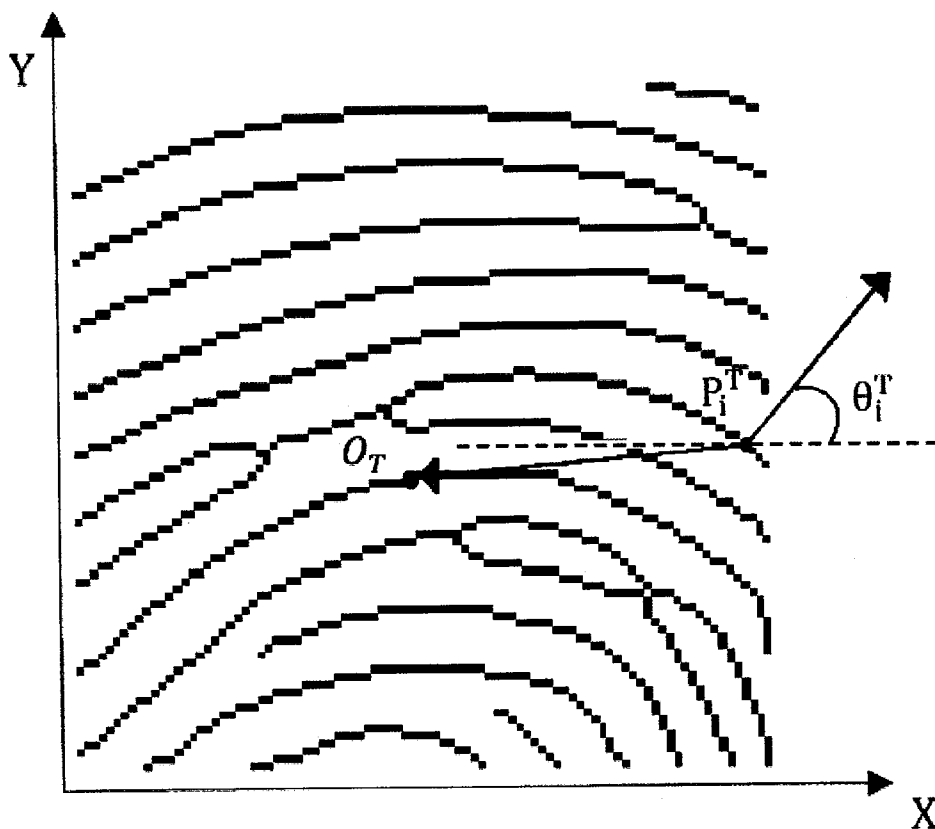


图 4.3 在模版坐标系中，广义哈夫变换所用的几何信息示例图

在小面积指纹图像中，细节点个数稀少，而脊线点数量相对大很多，例如在本章实验中使用的 160×160 的小面积指纹图像库中，一幅图像得到的脊线点集的平均个数为 3000 左右。大量的脊线点导致了大量的虚假投票，不仅使得计算量变得庞大，还容易将真实结果掩埋在错误结果中。为了使投票过程（即点集匹配过程）变得高效和准确，我们使用了脊线频率和脊线曲率来增加脊线点的区分能力，这两个脊线点属性的获取方式已在特征提取部分 4.2.1 中给出，这里不再赘述。在投票过程中，如果当前匹配点之间的脊线频率差和脊线曲率差大于固定阈值，则这样的匹配对应情况将不被考虑。此外，由于两个临近的脊线点之间具有很高的相似性（位置、角度、脊线频率、脊线曲率都几乎相同），因此我们在原始的脊线点集基础上对每根脊线沿着脊线进行了两倍的下采样，从而使得脊线点集的平均个数降为 1500 左右，大大降低匹配负担的同时，结果的准确率几乎

表 4.1 广义哈夫变换所使用的 R 表

k	Φ_k (index)	R_{Φ_k}
1	0	$\{r_i \theta_i^T = 0\}$
2	$\Delta\phi$	$\{r_i \theta_i^T = \Delta\phi\}$
3	$2\Delta\phi$	$\{r_i \theta_i^T = 2\Delta\phi\}$
...	...	...

不降。

投票空间（即查询指纹参考点的参数空间）在整个算法中占用大量的内存，而每一次投票都需要在该空间上进行操作，因此，对这个部分的优化是提升算法速度的关键。我们使用了一种多级搜索策略 (Ulrich et al., 2003) 来优化这个投票过程。整个投票过程被分别两个阶段：粗投票和精投票。同时，R 表和投票空间也分为两级。在粗投票阶段中使用第 1 级 R 表和第 1 级投票空间，在精投票阶段中使用第 2 级 R 表和第 2 级投票空间。具体的投票过程见算法 3 和算法 5。

通常情况下，对于小面积指纹脊线点集之间的匹配问题，在最后的投票空间中票数最多的点并非最优选择。这其中有两个主要原因，一是小面积指纹脊线点集之间实际的变换关系还是非线性的弹性变换关系，使用刚性变换的原因是为了简化求解方法，使得算法实际可以。二是广义哈夫变换寻求的得票最多的参考点并不等价于小指纹匹配的真实对应参考点，事实上，由于指纹脊线的周期性和自相似性，以及非完全重合匹配的特性，广义哈夫变换可能会出现“错周期”的错误匹配结果，在该匹配情况下，两匹配脊线点集间的重合点数最多（得票数最多），但并不是真实对应关系。如图 4.4 所示，广义哈夫变换给出的最优结果与真实的变换关系刚好差了一个纹线周期。

如上所述，由于该问题的特殊性，我们不能简单的通过广义哈夫变换的得票数来选取最优参考点，因此，在算法框架中，我们首先通过改进的广义哈夫变换来获取一些参考点作为候选，然后使用其他估计方法来从中选取最优参考点。经过算法 1 中的步骤 8 之后，得到了一些疑似最优查询指纹参考点的候选，如何从中选出一个参考点作为最终结果是算法 4 的目的。对于每一个参考点候选，算法都将计算一个相似度分数来衡量在该参考点情况下模版指纹与查询指纹重合区域的相似性。其中，算法使用了脊线点集生成的距离场，传统的距离场中每一点定义为距离该点最近的前景点（这里即为脊线点）到该点的距离。与传统的距离

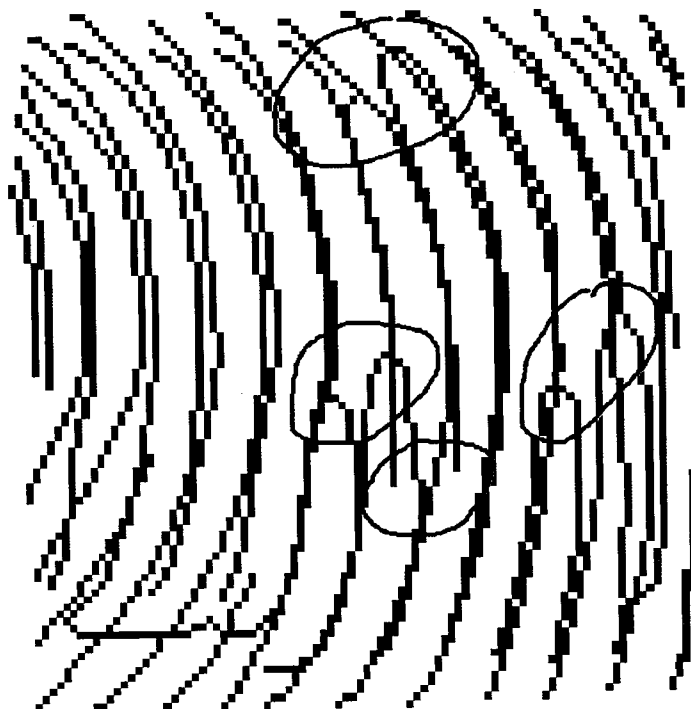


图 4.4 小面积指纹脊线点集错周期匹配事例

场不同，本章所使用的距离场中每一点的取值公式如下：

$$DM(x, y) = k_1 - k_2 * e^{k_3 * D(x, y)} \quad (4.4)$$

其中 $D(x, y)$ 表示传统距离场的取值， k_1, k_2, k_3 是三个人工设定的参数，在本章的实验中 $(k_1, k_2, k_3) = (27, 6, 0.5)$ 。与传统的距离场不同，本章使用的距离场为距离越近取值越大，即得分越大。传统的距离场可以通过动态规划的方式实现，因此相对于整个匹配算法运行的时间，距离场的计算时间几乎可以忽略不计。最优估计的详细描述见算法 4。

本小节详细介绍了所提出的改进的广义哈夫变换算法。对于小面积指纹脊线点集匹配问题，如果使用传统的广义哈夫变换进行匹配，则匹配算法是一个内存开销巨大且耗时的算法。而如果使用本小节所提出的改进的广义哈夫变换，则匹配算法是一个内存消耗低且快速高效的算法，同时匹配性能优于传统的广义哈夫变换。下面是本小节对传统的广义哈夫变换所做改进的总结：

1. 提取并使用脊线点的频率信息和曲率信息。在进行投票之前使用这两个信息进行校验，从而大幅度降低虚假投票；
2. 引入分级投票策略，大幅度降低内存空间消耗并提升投票速度。
3. 引入多候选和基于距离图的评价策略。相比于使用投票数的评价策略，可

大幅度提升正确性。

4. 对细节点投票（有极性）和脊线点投票（无极性）进行区分，添加权重。
5. 对脊线点集进行跟线采样，大幅度降低没有必要的投票。

4.2.3 相似度计算

对齐过程中的最优评估给出的相似度是用于区分当前两枚小指纹的多个候选结果，而匹配结果的最终相似度是用于区分所有小指纹之间的最优匹配结果。如果使用最优评估给出的相似度作为最终相似度，则其性能表现不佳。对于小面积指纹匹配来说，对齐结果的最终相似度评估是一个复杂的问题。这个问题需要考虑的许多因素，例如脊线点集的对齐情况、细节点集的对齐情况（细节点可缺失）、重叠面积、重叠区域的形变情况、重叠区域中未匹配的脊线点和细节点等。我们对匹配结果信息进行了人工专家经验的分析，从中挑选出了 12 个量化值来表示匹配结果的情况，具体定义见表 4.2 我们提出了一个启发式的打分策略（算法 6），但是在实际测试中我们发现，这个方法还是在一些特殊情况下表现不佳。这是因为很难确定每个需要考虑的因素所占的权重。

表 4.2 12 维的匹配结果信息

1	S_{Max}	算法 4 中获得的最优相似度
2	N_{mm}	匹配上的细节点个数
3	N_{ummt}	重叠区域中没有匹配上的模版脊线点个数
4	N_{ummq}	重叠区域中没有匹配上的查询脊线点个数
5	N_{mpt}	匹配上的模版脊线点个数
6	N_{mpq}	匹配上的查询脊线点个数
7	N_{ot}	重叠区域中的模版脊线点个数
8	N_{oq}	重叠区域中的查询脊线点个数
9	N_{mnq}	$Min\{Num(\mathbb{P}^T), Num(\mathbb{P}^Q)\}$
10	$\frac{N_{mpt} + N_{mpq}}{N_{ot} + N_{oq} + 1}$	
11	$\frac{N_{mpt} + N_{mpq}}{2 * N_{mnq} + 1}$	
12	$\frac{2 * N_{mm}}{N_{ummt} + N_{ummq} + 1}$	

事实上，最终结果的判定可以看作是一个二分类问题（接受/拒接），于是我们使用逻辑回归模型来解决这个问题。该模型获得的概率预测分布与 log 型函数相似，可以用其来作为最终匹配结果的相似度。逻辑回归模型是一种统计分析

算法 6 The Heuristic rule-based method**输入：** 匹配结果信息表.**输出：** 匹配结果相似度 S .

```

if  $N_{mm} \geq 3$  then  $Threshold = N_{mnp}/4$  end if
if  $N_{mm} = 2$  then  $Threshold = N_{mnp}/3$  end if
if  $N_{mm} = 1$  then  $Threshold = N_{mnp}/2$  end if
if  $N_{mm} = 0$  then  $Threshold = 3 * N_{mnp}/4$  end if
if  $N_{ot} + N_{oq} < 2 * Threshold$  then
     $S = 0$ 
else
     $S = (\frac{S_{Max}}{20} + \frac{N_{mpt} + N_{mpq}}{N_{ot} + N_{oq} + 1}) * 10000 + (N_{mm} - N_{ummt} - N_{ummq}) * 300$ 
     $S = \min\{9999, \max\{0, S\}\}$ 
end if
return  $S$ .

```

方法，模型的参数通过样本来学习获得。logistic 分布函数是其模型的决策函数，其公式如下：

$$p(x) = \frac{1}{1 + e^{-(w^T x + b)}} \quad (4.5)$$

其中， $p(x)$ 是一个输入为 n -维向量的概率密度函数， $w = (w_1, w_2, \dots, w_n)^T$ 为权重因子， b 是截断因子。在本章的实验中，输入 x 为一个 12 维的匹配结果信息的向量， $p(x)$ 则表示当前匹配是真匹配的概率， $(1 - p(x))$ 表示为假匹配的概率。 x 的详细描述见表 4.2。

为了获取得到模型的参数 w 和 b ，需要最小化损失函数：

$$\min_{w, b} Loss = \frac{1}{2} w^T w + \lambda \sum_{i=1}^M \text{Ln}(e^{-y^i(w^T x^i + b)} + 1) \quad (4.6)$$

其中 λ 是惩罚因子， M 是训练样本的个数， y^i 是样本 x^i 的真实标签。每个 $|w_j|$ 表示 x 的第 j 维信息的重要程度。一个大的 $|w_j|$ 表明其对应维度的信息具有很强的区分能力，而一个小的 $|w_j|$ 则表明其对应维度的信息对真假匹配结果的预测没有什么贡献。实验中，我们使用随机梯度下降法 (Schmidt et al., 2016) 来训练 w 和 b 。

4.3 实验结果及分析

由于目前并没有类似 NIST 和 FVC 这样的小面积指纹公开测试库，因此本章的匹配算法是在实验室内数据（小面积指纹库 PFDB1690）上进行评估。为了方便同行学者的研究，我们公开了该测试库⁴。PFDB1690 由 169 根手指按压采集得到的 1690 幅指纹图像组成。每跟手指按压采集了 10 次，且这 10 幅指纹图像经过人工确认包含该枚手指的同一个区域。指纹图像的大小均为 160×160 , 500dpi。如图所示为一些该库中的小面积指纹图像。实验中，我们使用了 690 幅指纹图像去训练 4.2.3 节中的逻辑回归模型。在整个 PFDB1690 上进行了本章所提出的匹配算法的测试。



图 4.5 PFDB1690 中的 6 幅指纹图像，第一行的 3 幅来源于一根手指，第二行的 3 幅来源于另一根手指

为了方便叙述，我们将本章所提出的基于脊线与广义哈夫变换的指纹匹配算法称为 MGHT。我们在整个 PFDB1690 库上进行了评估，其中一共有 1690×9 个真匹配⁵，以及 $1690 \times 1681 = 2840890$ 个假匹配。在本章的整个算法中存在着很多参数阈值，其中的一些参数对算法的性能起着重要影响。 W_m 是细节点投票的权重，如果将 W_m 设置为 0，则等价于在匹配中不使用细节点信息，反之，如果 W_m 取值很大，则意味着匹配的结果完全依赖于细节点的匹配。 CN 是所求查询指纹参考点的候选个数，对于匹配精度来说， CN 取值越大，即候选越多，则获得准确的对应结果的可能性就越大，从而匹配精度就越高。但是， CN 取值越大

⁴小面积指纹库 PFDB1690 的下载地址为 <https://pan.baidu.com/s/134XjS38nbkQSxyXRE1kVXw>，提取码为 jk68。

⁵指纹 A 比对指纹 B 与指纹 B 比对指纹 A 通常被认为是两个不同的匹配，因为模版指纹和查询指纹对换过后在大多匹配算法中并不等价，结果有时甚至会区别很大。

算法	EER(%)	FRR(%) (at 0.1% FAR)	FRR(%) (at 0.01% FAR)	FRR(%) (at 0.002% FAR)	1:1 匹配时间
0_1_Rule	4.32	9.44	13.02	15.73	25.44 ms
20_5_Rule	0.82	1.64	3.56	5.85	29.39 ms
20_5_ML	0.50	0.66	1.00	1.59	29.41 ms
20_20_ML	0.32	0.41	0.72	1.30	46.17 ms
NBIS	—	48.06	60.79	76.19	—
Fourier-Mellin	17.08	30.97	45.36	53.92	27.08 ms
SIFT	1.01	2.68	6.88	11.63	50.43 ms

表 4.3 各算法在小面积指纹图像库 PFDB1690 上的测试结果

的同时也意味着要进行更多的判定，从而增加算法的计算量，导致匹配速度降低。因此， CN 的取值依赖于实际应用中对精度需求和速度需求的综合考虑。在最终的匹配相似度计算中，我们提出了两种方法，一种是基于人工专家经验的启发式算法，另一种是基于机器学习的算法，两种打分方法都在实验中进行了测试。我们使用 X_Y_Z 的表示形式来表示参数选择为 $\{W_m = X, CN = Y, SS = Z\}$ 的 MGHT 算法，例如，20_5_ML 表示参数选择为 $\{W_m = 20, CN = 5, \text{打分策略} = \text{机器学习方法}\}$ 的 MGHT，而 0_1_Rule 表示参数选择为 $\{W_m = 0, CN = 1, \text{打分策略} = \text{启发式方法}\}$ 的 MGHT。

表4.3给出了各种算法在 PFDB1690 上的测试结果。图4.6给出了不同参数设置下的 MGHT 在 PFDB1690 上的 ROC 曲线。尽管我们的算法使用到了细节点，但实际上当细节点信息缺失时，MGHT 依旧能达到很好的性能。如图4.7所示，0_5_ML 表示细节点权重为 0，即不使用细节点信息的 MGHT，该结果与 20_5_ML 的结果几乎没有差别。其中算法 20_5_ML_WithoutRPCV 表示在整个算法框架中跳过脊线点集粗投票（算法 3）步骤。该算法结果表现糟糕，表明 MGHT 的主要匹配能力是在脊线点集投票步骤，即脊线点集的匹配是该算法的核心。图4.8表明机器学习方法在处理综合打分问题时具有更加强的鲁棒性和适应性。实验中，我们不仅对比了不同参数下的 MGHT 的结果，还与其他小面积指纹算法也进行了对比。其中包括基于细节点的匹配算法 (NIST, 2015; Watson et al., 2007)，基于像素相关的匹配算法 (Zhang et al., 2006a) 以及基于图像匹配的

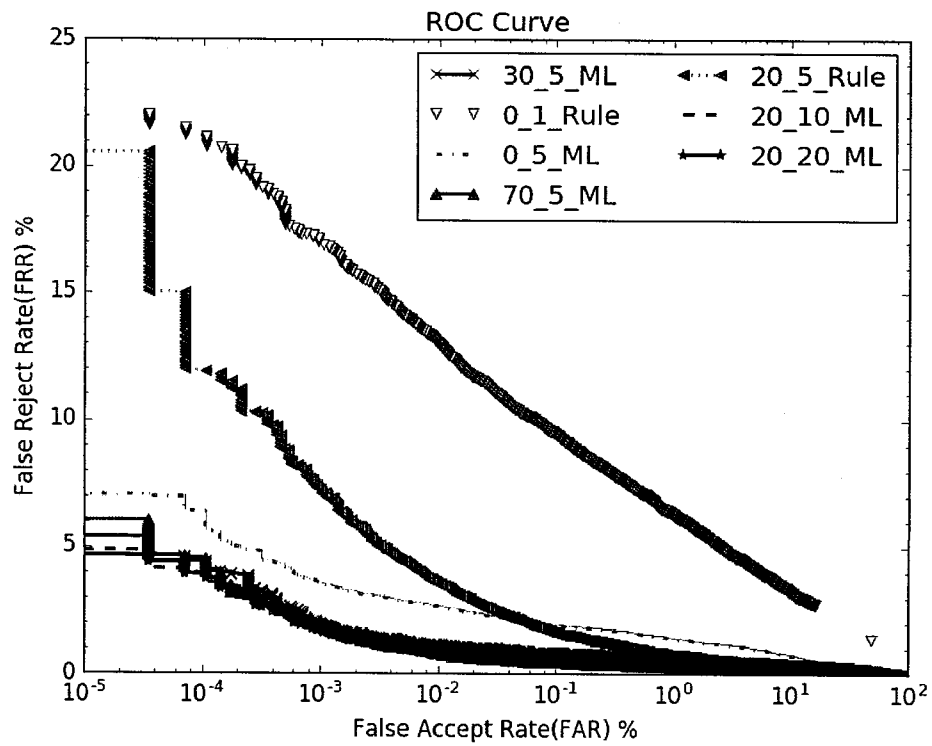


图 4.6 不同参数设置下的 MGHT 在小面积指纹图像库 PFDB1690 上的 ROC 曲线

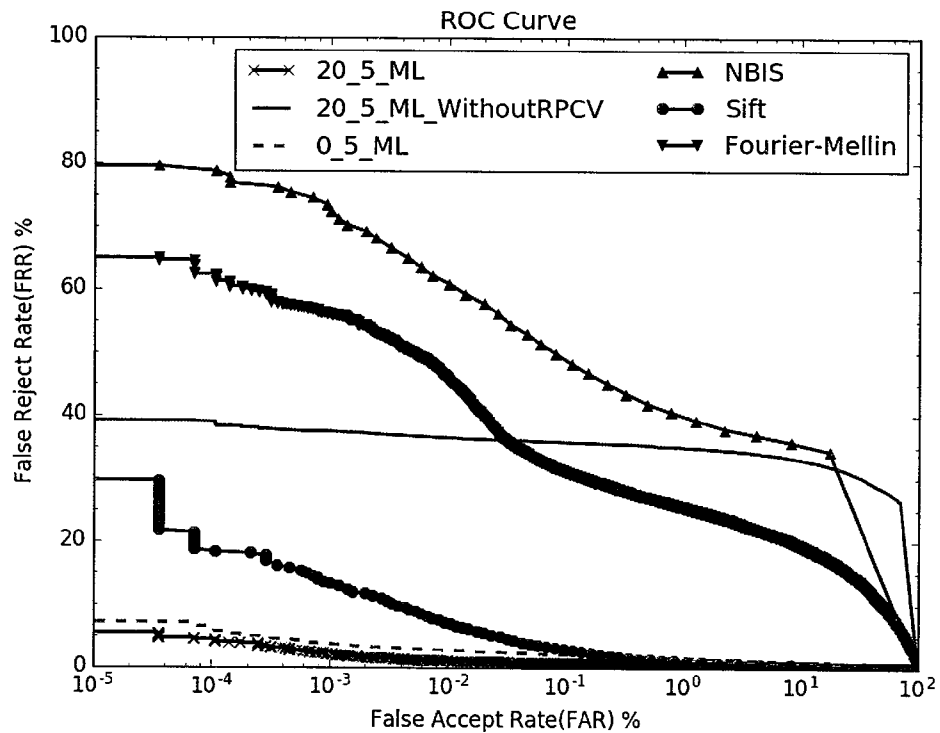


图 4.7 各算法在小面积指纹图像库 PFDB1690 上的 ROC 曲线

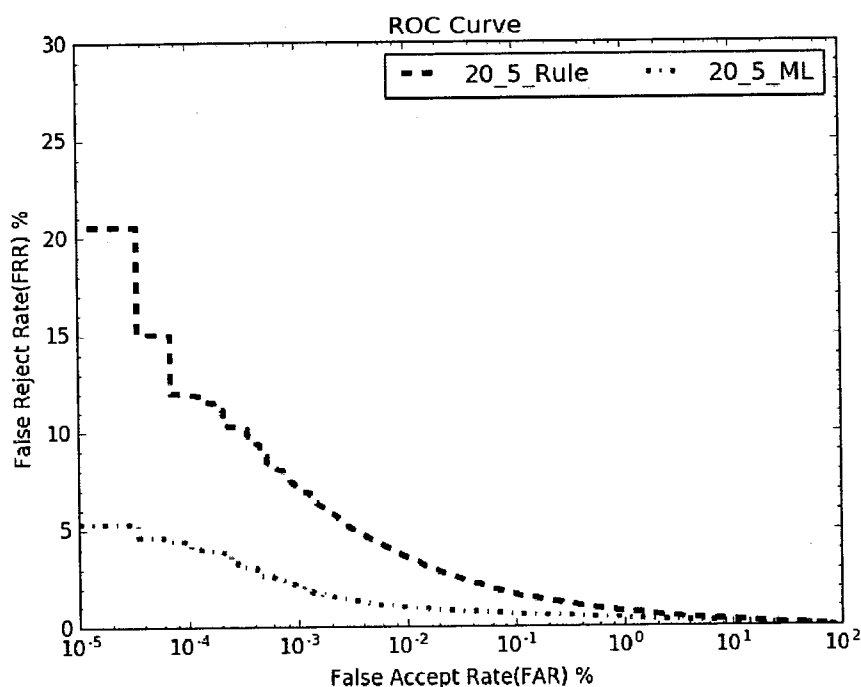


图 4.8 使用机器学习打分的 MGHT 与使用启发式策略打分的 MGHT 在小面积指纹图像库 PFDB1690 上的 ROC 曲线

算法 (Yamazaki et al., 2015; Zhou et al., 2011; Pankanti et al., 2008; Mathur S, 2016)。在本章的实验中, SIFT 的方法是通过 OpenCV2.4.9 与文献 (Yamazaki et al., 2015; Zhou et al., 2011; Pankanti et al., 2008; Mathur S, 2016) 中的打分策略相结合实现的。本章的 MGHT 算法通过 C++ 实现, 1 幅 160×160 的小面积指纹图像的特征提取时间为 30 毫秒左右, 使用参数 20_5_ML 的 MGHT 的 1 对 1 匹配时间为 29 毫秒。本章实验环境配置为: 操作系统 (Windows 7), CPU (Core i7-4710MQ 2.5Ghz), RAM (DDR3L 1600MHz 8GB)。

4.4 本章结论

目前, 小面积指纹识别还未得到很好满意的识别效果, 这是一个重要的实际应用问题, 具有重要的研究意义。本章提出了一套完整的基于脊线特征和改进的广义哈夫变换的小面积指纹识别算法, 该算法使用指纹特有的脊线作为特征, 可以有效防止目前其他算法存在的被非指纹图像攻击的问题。针对性的改进使得广义哈夫变换在处理匹配对齐时变得精准高效。在匹配相似度估计中, 我们使用机器学习的方法来应对这一综合性的复杂问题, 避免了人工调参等费时费力问题的同时还取得了更加优异的性能。许多优化策略被应用到算法当中, 使得本章的算法能够满足实际应用中的高精度、低消耗、实时等要求。

第5章 基于改进的极坐标傅立叶变换和深度卷积网络的小面积指纹识别算法

5.1 引言

小面积指纹识别中最关键的一个问题是如何充分利用指纹图像所包含的信息。上一章提出的小面积指纹匹配算法使用了指纹的脊线信息和细节点信息,虽然在大部分情况下这两个信息已经提供了足够的区分信息,但是在一些质量差的指纹图像中,脊线信息和细节点信息都较难准确地提取,此时基于这些信息的匹配算法就会失效。为了有效地应对这种情况,本章提出了一种基于改进的极坐标傅立叶变换和深度卷积网络的小面积指纹匹配算法。该算法通过改进的极坐标傅立叶变换来进行粗对准,然后提取出匹配的重合区域,最终使用深度卷积网络来对重合区域进行相似度估计。依赖于特定特征的算法存在某些情况下特征可能无法稳定提取的缺点,例如 SIFT 中的关键点特征、指纹的奇异点、细节点以及脊线特征等,都可能在某些低质量指纹图像中难以提取或提取到虚假不稳定的特征。与其他的基于特定特征的算法不同,本章所提出的算法不依赖于任何特定的特征,因此有更广泛的适应性。实验结果表明本章所提出的匹配算法具有优异的性能。

本章的第二节将详细介绍匹配过程中的对准算法,在第三节中,我们使用了一个深度卷积神经网络来提取重叠区域的特征向量,并使用该特征向量来计算两幅小面积指纹图像之间的相似度。第四节为本章的实验部分,本章的算法与其他主流的小面积指纹识别算法将在这个部分进行充分的对比测试。最后,我们在第五节给出本章的结论。

5.2 基于极坐标傅立叶变换的指纹对齐

近年来,深度学习在图像处理和模式识别领域掀起了一番热潮,在通用目标检测和识别问题以及人脸识别问题上,深度学习都取得了巨大突破。在人脸识别中,两幅待匹配的人脸图像首先需要进行预对齐,这一步是通过人脸中标志点,诸如两眼各自的中心点、鼻子的顶点、嘴唇的两个角点等来进行配准的。而与人脸识别不同,小面积指纹中不存在这样的参考点。也难以找到类似的稳定点。因此基于深度学习的人脸识别方法无法直接应用于小面积指纹识别。

基于深度学习的识别方法不需要任何的特点特征,如果我们使用类似 SIFT

关键点或是上一章中提出的脊线特征来做对齐，则将使得算法的适应性被该特征所拖累（难以提取都该特征的情况下就无法对齐，也就无法进行匹配识别）。为了保持对齐中使用的信息与相似度估计中使用的信息一致，我们利用了基于图像匹配的算法来进行对齐。

图5.1为对齐算法的一个详细流程图。该算法包括三个步骤：（1）旋转角度估计；（2）平移估计；（3）提取重叠区域。

相位相关（Phase-Only Correlation, POC）是数字图像处理中广泛使用的一种方法。其通过计算两幅图像像素的相关性来判断是否相似。给定两幅尺寸为 $M \times N$ 的图像 $f(x, y)$ 和 $g(x, y)$ 。假设 $F(u, v)$ 和 $G(u, v)$ 为这两幅图像对应的 2 维离散傅立叶变换。我们用 $\mathcal{F}$ 来表示 2 维离散傅立叶变换，则 $F(u, v)$ 和 $G(u, v)$ 为：

$$F(u, v) = \mathcal{F}(f(x, y)) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})}. \quad (5.1)$$

$$G(u, v) = \mathcal{F}(g(x, y)) = \frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} g(x, y) e^{-i2\pi(\frac{ux}{M} + \frac{vy}{N})}. \quad (5.2)$$

定义两幅图像之间的归一化交叉谱（Normalized Cross-Spectrum）为：

$$K(u, v) = \frac{F(u, v) \cdot G^*(u, v)}{|F(u, v) \cdot G^*(u, v)|}. \quad (5.3)$$

其中，“ $\cdot$ ”表示哈达玛内积，“ $*$ ”表示取复共轭操作。

$$POC_{f,g}(x, y) = \mathcal{F}^{-1}\left(\frac{F \cdot G^*}{|F \cdot G^*|}\right) = \sum_{u=0}^{M-1} \sum_{v=0}^{N-1} K(u, v) e^{i2\pi(\frac{ux}{M} + \frac{vy}{N})}. \quad (5.4)$$

如果 $g(x, y)$ 是 $f(x, y)$ 平移之后得到的图像，即 $g(x, y) = f(x + \Delta x, y + \Delta y)$ ，其中 $(\Delta x, \Delta y)$ 为平移向量。则 $f(x, y)$ 与 $g(x, y)$ 之间的 POC 函数 $POC_{f,g}(x, y)$ 将会在 $(\Delta x, \Delta y)$ 位置处取得最大值。通过计算两幅图像之间的 POC 函数并检测其极值点，我们即可估计出两幅图像之间的平移关系，极值点的取值代表平移之后两幅图像的相似度。

指纹的脊线宽度一般为 100 微米到 300 微米之间，如一些女性和小孩的指纹一般较细，脊线宽度为 100 微米左右，300 微米宽的指纹脊线一般为男性或老人。大部分情况下，指纹脊线和谷线的周期为 500 微米左右。由于指纹脊线的周期是在一定范围内的这个特性，我们使用了带通滤波来去除噪声并增强指纹图

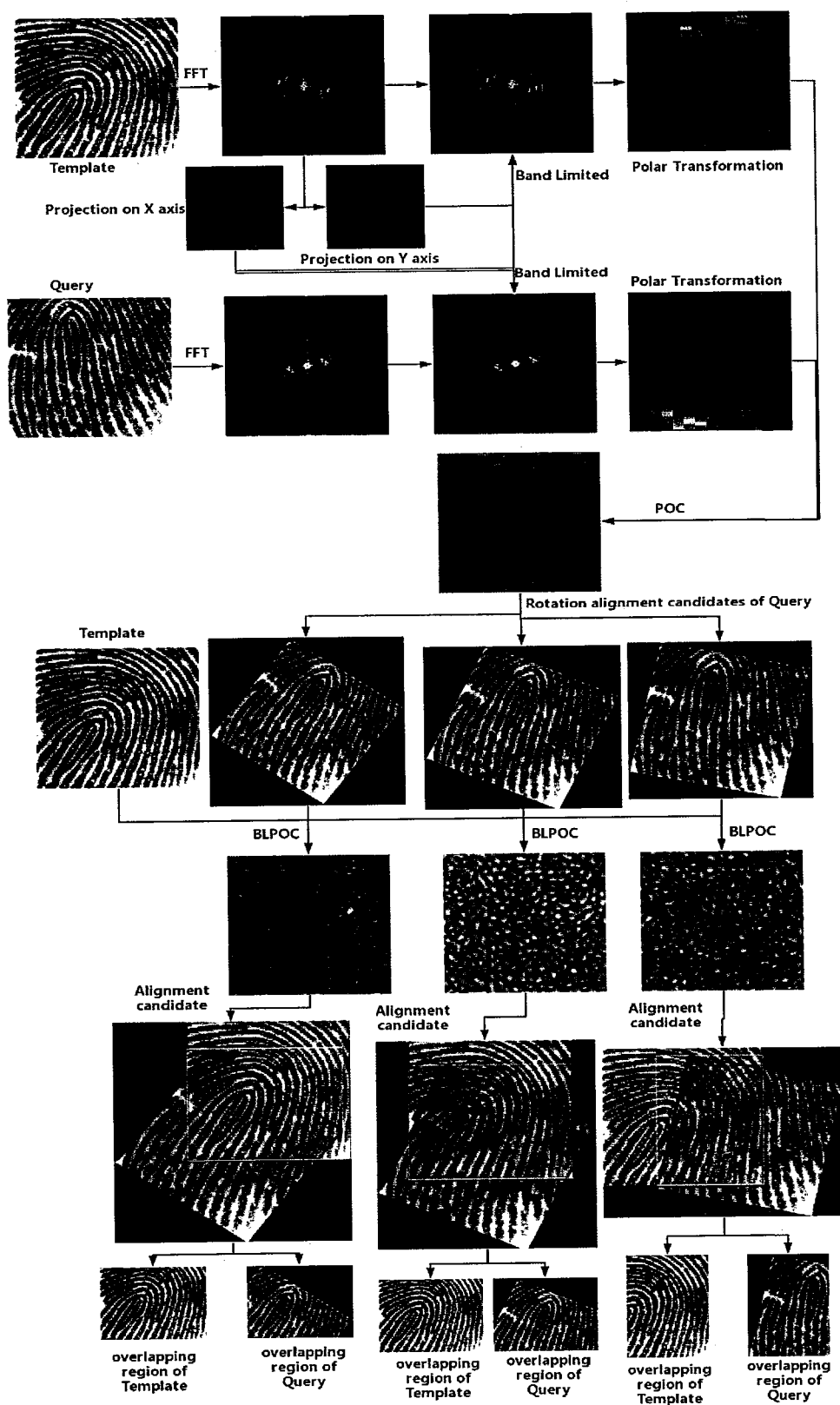


图 5.1 基于改进的极坐标傅立叶变换的小面积指纹对齐算法

像,实验结果表明这个策略可以显著提高算法的性能。实验中,我们使用一个环状的带通滤波去除图像的高频部分和低频部分,然后保留指纹脊线频率所在的中频部分。高频的阈值通过文献 (Ito et al., 2004) 中的方法来获取。低频的阈值为一个人工经验阈值。

$$\begin{aligned} g(x, y) &= f(x \cos(\theta) + y \sin(\theta), -x \sin(\theta) + y \cos(\theta)) . \\ \Downarrow \\ G(u, v) &= F(u \cos(\theta) + v \sin(\theta), -u \sin(\theta) + v \cos(\theta)) . \end{aligned} \quad (5.5)$$

傅立叶变换具有旋转不变性,如公式5.5所示,在原图旋转一定的角度之后执行傅立叶变换等价于先傅立叶变换然后再旋转。由此特性,我们即可通过估计两幅图像在傅立叶域上的旋转角度差来获得两幅图像之间的角度差。为了使用计算平移的 POC 函数来计算旋转角度,极坐标变换在此处被运用。通过极坐标变换,旋转角度和放缩因子等估计等价于平移估计问题,从而使用同一种方法,即可把旋转角度和平移参数都估计出来,这个方法通常称为极坐标傅立叶变换方法。为了提升算法的性能,我们取出了 POC 函数的前 N 个最大值所对应的参数作为参数估计结果的候选。对齐算法的详细流程图如5.1所示,我们首先估计了两幅图像之间的旋转角度,此处将得到 N_A 个旋转角度的候选,对于每一个可能的旋转角度候选,我们都将进行对应的平移参数估计,同样地,每一个旋转角度候选将对应 N_D 个平移参数候选。图5.1为一个 $N_A = 3, N_D = 1$ 的算法框架示例。在本章的实验中,我们设置 $N_A = 6, N_D = 3$ 。因此,通过该对齐算法,我们将获得 $N = 6 \times 3 = 18$ 个对应候选,其中每一个对齐候选都将进行下一步的重叠区域相似度计算,最后将相似度最大的结果作为最终结果输出。对于整个匹配算法, $N = N_A \times N_D$ 的取值非常关键, N 值越大,候选越多,真实的匹配情况越有可能被找到,因此,性能越好,但随之而来的是计算量也越大。 N 值选取是匹配性能与匹配速度之间的一个权衡问题。

5.3 基于深度卷积网络的指纹相似度估计

在传统的基于相位相关的匹配算法中,通常使用 POC 函数的最大值来作为最终的相似度估计。这种相似度估计方法虽然简单但区分力很差,特别是在小面积指纹识别这个问题上,不同指纹之间的相似度可能很高,而由于形变,同一根手指的多次按压可能区别很大。图5.2给出了我们在实验中使用 POC 函数得到的错误事例,从中可以看到,由于大部分区域的相似性,假匹配可能会得到非常高

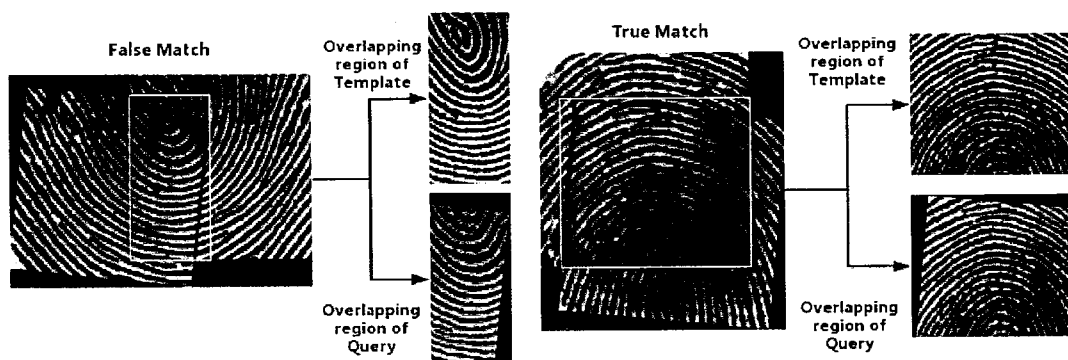


图 5.2 难以区分的假匹配和真匹配实例

的相关值，而因为整个区域的形变，导致真匹配之间的相关值非常低。此外，对齐结果也可能存在一定的偏差（事实上，要获得一个精确的对齐结果是一个非常难的问题），这就使得简单的相位相关值更加不能反应真实的相似度。

指纹匹配中的重叠区域相似度估计问题一直是一个没有得到很好解决的问题。该问题是一个综合因素估计的问题。例如在图5.2的假匹配中，虽然整体很相似，但是我们通过很小的局部脊线交错就可以进行一票否决。而在图5.2的真匹配中，虽然整体的像素相关值不高，但其实是因为形变的原因，脊线之间整体的拓扑结构都是一致，因此可以认定为是同一根手指。受到文献 (Hu et al., 2014) 的启发，我们使用卷积神经网络来处理这个问题。图5.3给出了该相似度估计算法的一个清晰示例图，深度卷积神经网络在两幅指纹图像的重叠区域分别提取到一个固定长度的特征向量，最终的相似度通过计算两特征向量之间的欧氏距离来获得。该方法使用了重叠区域中的所有信息，并通过有效的学习方法来弥补对齐精度的不足。

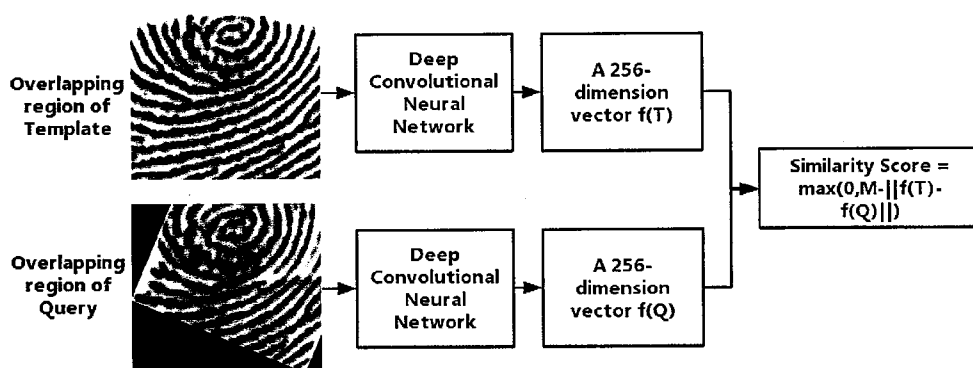


图 5.3 重叠区域相似度估计示例图

5.3.1 网络结构

算法中所使用的深度卷积神经网络结构如图5.4所示。假设 $f(x)$ 表示在某个重叠区域 x (x 为一幅图像) 上经过该深度卷积网络提取得到的特征向量。在这里需要注意, 大部分情况下, 两幅指纹图像匹配的重叠区域是一个不规则的区域, 为了方便处理, 我们将提取重叠区域外包围框所包含的区域作为实际处理中的重叠区域 (如图5.1和图5.2中黄色框), 对于无图像信息的区域, 我们将填充 0。此外, 对于不同的对齐结果, 重叠区域 x 具有不同的尺寸, 因此需要该深度卷积神经网络能够处理任意尺寸的输入。在本章的深度神经网络设计中, 我们通过使用金字塔池化层 (He et al., 2015) 来克服这一问题。金字塔池化层对于任意尺寸的输入将得到一个固定长度的输出, 该输出即为最终的特征向量。另外, 金字塔池化层对物体的形变还具有一定的鲁棒性。在本章的算法框架中, 金字塔池化层局有 3 个等级, 分别为 3×3 , 2×2 和 1×1 。为了提高深度神经网络的性能, 目前最优的残差模版 (He et al., 2016) 也应用于其中。

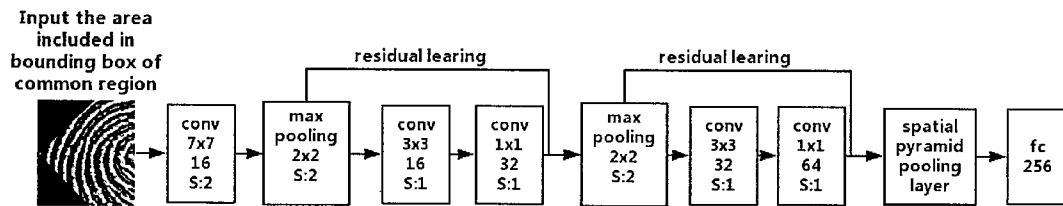


图 5.4 深度卷积神经网络的结构示意图

5.3.2 训练样本

在对齐阶段得到的两幅指纹图像的重叠区域将被用来作为训练样本。如果两重叠区域来自于同一根手指的同一个区域, 则这一对重叠区域为正样本, 否则为负样本。显然, 我们可以很容易的获得许多负样本, 只要使用两幅不来自于同一根手指的指纹图像进行对齐操作, 得到的对齐结果均为负样本。但是, 如果不加任何的筛选策略, 则得到的负样本大多都是非常容易区分的负样本, 这些大量的简单负样本对模型区分能力的提升几乎没有帮助。此外, 很难得到精确的正样本, 这是因为很难通过自动算法去确定两重叠区域来自于同一根手指的同一个区域, 而如果通过人工去确定, 则需要很大的工作量。因此, 如何得到大量可靠的训练样本是一个困难但关键的问题。在本章中, 我们使用如下的策略去生成训练样本:

- (1) 对每一对对齐的重叠区域, 使用文献 (Ito et al., 2004) 中的相似度计算方

法来进行估计，得到一个相似度分数 S_c 。

(2) 如果 S_c 大于某个阈值，则将该重叠区域对选为训练样本，如果 S_c 小于阈值，则丢弃。

(3) 如果将该重叠区域对来自于同一根手指，则标记为正样本，否则，标记为负样本。

通过上面的步骤，我们即可简单的获得一批较为准确且有代表性的训练样本。

5.3.3 损失函数

在本章的深度神经网络训练中，我们将 L2 损失 (5.6) 与正则化逻辑损失 (5.7) 相结合得到损失函数 (5.8)，并通过该损失函数来进行训练。

$$L_{L_2}(T_i, Q_i, y_i) = \begin{cases} \frac{1}{2} \|f(T_i) - f(Q_i)\|_2^2 & \text{if } y_i = 1 \\ \frac{1}{2} \max(0, m - \|f(T_i) - f(Q_i)\|_2)^2 & \text{if } y_i = -1 \end{cases} \quad (5.6)$$

其中 y_i 是第 i 个训练样本的标签， $y_i = 1$ 表示 (T_i, Q_i) 是一个正样本（真匹配）， $y_i = -1$ 表示 (T_i, Q_i) 是一个负样本（假匹配）。

$$L_{\text{logistic}}(T_i, Q_i, y_i) = \frac{1}{2} g(1 - y_i(m - \|f(T_i) - f(Q_i)\|_2)^2) \quad (5.7)$$

其中 $g(x) = \frac{1}{\beta} \log(1 + e^{\beta x})$ 是正则化逻辑损失函数 (Hu et al., 2014)。

$$Loss = \frac{1}{N} \sum_i^N (L_{L_2}(T_i, Q_i, y_i) + \lambda L_{\text{logistic}}(T_i, Q_i, y_i)) \quad (5.8)$$

其中 λ 为权重参数。

5.4 实验结果及分析

本章所提出的算法框架具有一个很明显的优势，那就是可以使用多种不同尺寸的指纹图像来训练同一个模型，使得最终模型可以适应任意尺寸的小面积指纹匹配。因此，该算法可以充分利用已有的指纹数据库。在本章的实验中，我们使用的数据库如下：(1) FVC2000 (Maio et al., 2002a), (2) FVC2002 (Maio et al., 2002b), (3) FVC2004 (Maio et al., 2004), (4) FingerPass DB7 (Jia et al., 2012), (5) 实验室内部小面积指纹数据库 PFDB1690。

FingerPass DB7 (Jia et al., 2012) 数据库包含 8640 幅大小为 144×144 (500 Dpi) 的指纹图像, 这些图像来自于 720 根不同的手指, 每根手指按压采集得到了 12 幅指纹图像。实验室内部小面积指纹数据库 PFDB1690 由 169 根手指按压采集得到的 1690 幅大小为 160×160 的指纹图像组成。每跟手指按压采集了 10 次, 这 10 幅指纹图像经过人工确认包含该枚手指的同一个区域。为了方便同行学者的研究, 我们公开了该测试库¹。为了获得尽可能多得用小面积指纹来训练神经网络, 我们利用了传统的指纹公开库 FVC(Maio et al., 2002a,b, 2004) 来模拟生成小面积指纹。模拟生成小面积指纹的方法来自于文献 (Roy et al., 2017), 该方法获得的小面积指纹大小为 150×150 。由于该模拟生成的方法不能确保来自于同一幅原始指纹图像的两幅小面积指纹图像包含有某个相同的区域, 因此我们在测试中不使用该生成方法得到的小面积指纹图像。我们使用所有生成方法得到全部的小面积指纹图像, 一半的 FingerPass DB7 以及一半的内部小指纹数据库来训练神经网络。对任意两幅指纹图像进行匹配, 对齐算法可以获得 18 个对齐结果 (见 5.2, $N = 6 \times 3 = 18$), 因此最多可以获得 18 个训练样本。图 5.5 展示了一些实验中的训练样本。

本章所提出的小面积指纹匹配算法主要通过 C++ 进行实现, 其中使用到的工具包括: OpenCV 2.4.10, FFTW(Frigo et al., 1998), Caffe(Jia et al., 2014)。

表 5.1 各算法在测试集上的 EER

Method	FingerPass DB7	PFDB1690
本章算法	4.01%	1.32%
A-KAZE(Mathur S, 2016)	15.55%	4.60%
Minutiae-based(Jea et al., 2005)	30.67%	11.41%

本章实验中, 我们与基于图像匹配的算法 (Mathur S, 2016) 和基于细节点的算法 (Jea et al., 2005) 做了比较, 测试集为 FingerPass DB7 和实验室内部小面积指纹数据库 PFDB1690。表 5.1 为各算法在测试集上的 ERR 结果, 图 5.6 为 ROC 曲线。实验结果表明本章所提出的小面积指纹匹配算法性能优异, 具有很强的适应性。针对一些错误实例, 我们进行了分析。FingerPass DB7 中每一幅指纹图像都包含一个黑色边框, 使得其实际的前景区域仅为 120×120 左右, 经过自动细

¹小面积指纹库 PFDB1690 的下载地址为 <https://pan.baidu.com/s/134XjS38nbkQSxyXRE1kVXw>, 提取码为 jk68。

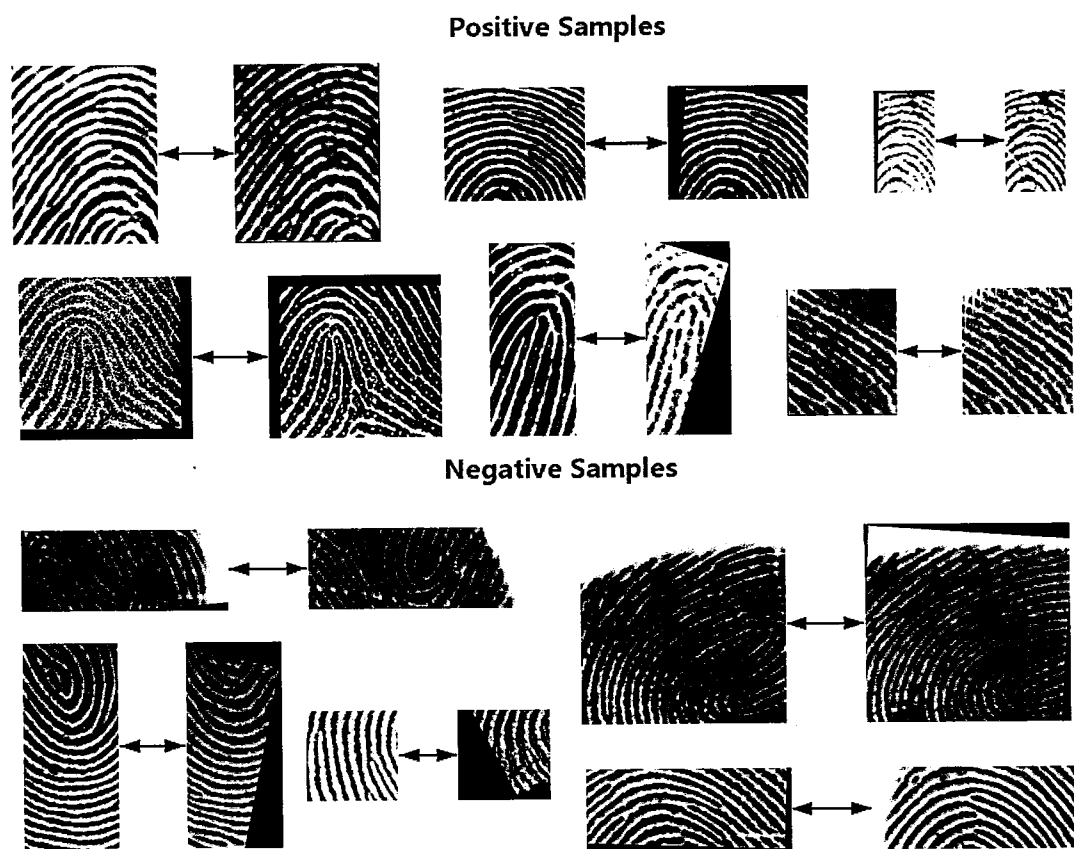


图 5.5 训练样本实例

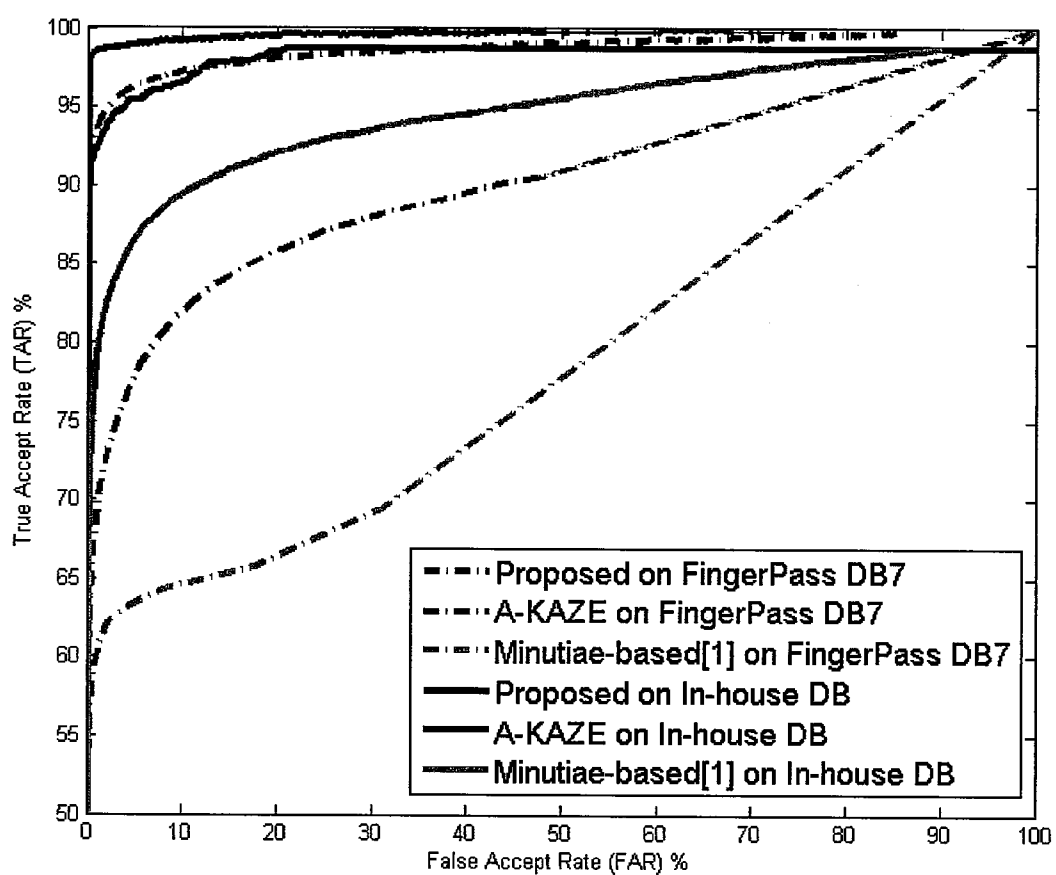


图 5.6 各算法在测试库上的 ROC 曲线

节点提取之后的统计,发现其细节点个数平均仅为5个左右,而PFDB1690中的细节点个数平均为12个,因此基于细节点的算法在两个库上差异明显。A-KAZE算法难以处理干湿手指的情况,此外该算法需要较大的重叠面积来区分真假匹配。本章算法的错误实例一般由错误的对齐结果导致,尽管我们设置了足够多的对齐候选,但在某些剧烈形变或严重噪声干扰的情况下,正确的对齐结果依旧难以进入对齐候选之中,事实上,在这些情况下,其他算法也无法处理。

5.5 本章结论

本章提出了一种新的小面积指纹匹配算法,该算法使用改进的极坐标傅立叶变换来获得两幅指纹图像的多种对齐可能,在对齐的基础上使用深度卷积神经网络来解决相似度估计这一复杂问题。整个算法将传统算法与当前主流的深度学习方法相结合,在实验中性能表现优异,表明了该算法的可行性。在实验结果分析中,我们发现该算法的主要瓶颈还是在传统方法部分,即对齐阶段。后续,我们将设计新的深度网络框架,使其将对齐过程也加入到其中,形成一个完整的指纹图像匹配网络。

第6章 总结与展望

生物特征识别产业在近年来发展迅猛,尤其是人脸识别与指纹识别的应用,在人们的日常生活中随处可见。指纹识别作为生物特征识别中使用最广泛的技术,已经发展了几十年,取得了长足的进步。但是由于所处时代的不同,当前指纹识别也面临着新的挑战。大数据和移动互联网时代对指纹识别技术提出了更高的要求。大数据时代下,指纹数据库的规模也不断增大,在千万级甚至亿级的指纹数据库上进行高效查询是目前指纹识别技术面临的一个难题。移动互联网时代下,各种移动终端随处可见,在这些资源配置较差的移动设备上进行高效、可靠、安全的身份认证是当前指纹识别技术面临的另一个难题。

奇异点作为指纹的一种全局特征,在指纹的匹配中扮演着重要角色。通过充分利用可靠精准的奇异点,可以使得指纹的精匹配速度大大提升。其次,奇异点作为指纹的形态学特征,可以用来对指纹进行分类,从而使得指纹粗匹配速度大大提升。在指纹索引中,一些学者使用奇异点作为参考点来提取其周围区域中的信息进行特定编码,从而对指纹建立更精细的分类索引,大大提升指纹查询的速度。因此,稳定可靠的奇异点提取结果对指纹的高效查询起着重要作用。本文对这一关键问题进行了深入研究,传统的奇异点检测算法大都是基于 Poincare 指数,且非常依赖于指纹方向场的精确度,在一些方向场难以计算的指纹图像中,这些方法都将失效。作者在第二章中提出了一种基于深度学习的指纹奇异点检测算法。深度学习方法与大数据紧密结合,可以充分利用大数据带来的优势。所谓见多识广,通过大量的指纹奇异点模式样本来充分训练深度卷积神经网络,使得该奇异点检测算法具有更强鲁棒性。此外,该算法不依赖于指纹方向场,相比于传统算法具有更好的适应性。

传统的指纹识别系统一般用于刑侦、海关、银行等领域,其所处理的指纹识别关系着刑事案件、海关的身份认证以及经济活动等重要事件,服务对象也均为政府军队等。因此,指纹厂商和政府主要关注的是指纹识别系统的安全性、高效性以及准确性,通常不计较指纹识别系统的成本。而与传统的指纹识别系统不同,目前的移动设备或是小型终端上的指纹识别系统均是民用产品。一方面,民用指纹识别系统对安全性、可靠性等需求就比警用军用的低很多。另一方面,当前人们在关心安全性的同时,同样关心产品的实用性和价格。因此,在市场竞争的结果下,更便宜且好用好看的指纹识别系统更受人们的青睐。正是由于这些原

因,小面积指纹识别系统成为了当前民用指纹识别系统的代表。小面积指纹识别系统所使用的指纹传感器均是小尺寸,不仅价格低廉,还更容易设计嵌入到小型移动设备中,并且外观更好看。但是小面积指纹识别系统所能获取到的指纹图像均是小面积指纹图像。量变导致质变,由于小面积指纹图像所包含的信息急剧减少,传统的基于细节点的指纹识别算法已无法在小面积指纹图像上使用。因此,各个基于不同特征的不同算法相继被提出,这些算法各有利弊,都还无法达到让人满意的效果。目前智能手机上使用的小面积指纹识别算法多为基于通用图像匹配的指纹识别算法,该方法的优势是适应性强,因此用户体验更好,但存在着易被非指纹图像攻击的缺点。2018年初媒体报道几乎所有的智能手机都可通过贴模来破解指纹识别,这一事件引起了社会的广泛关注,而这一问题的本质正是由于基于通用图像匹配的指纹识别算法所造成的。

本文对小面积指纹识别进行了深入的研究,在第三章中对目前小面积指纹识别的发展进行了详细介绍,对其所面临的问题进行了详细的分析。针对当前小面积指纹识别所面临的资源配置低、实时响应、以及较高安全性需求等问题,我们在第三章和第四章分别提出了两种不同的小面积指纹识别算法。第三章中的匹配算法使用指纹脊线这一指纹特有的特征,使得攻击图像难以模仿,从而具有更高的安全性。第四章中的匹配算法主要针对老人小孩等指纹质量较差的情况,在这些情况下,依赖于特定特征的算法很可能会失效,因此我们提出了一种不依赖于任何特定特征的匹配算法,该算法中同样使用到了深度学习方法 and 大量指纹数据,从而使得该算法具有更强的鲁棒性。

虽然做了以上工作,但是要解决指纹识别当前面临的两个难题还远远不够。如何充分利用奇异点信息来进行海量指纹数据库中的高效查询是今后工作的一个重点。此外,深度学习在通用目标识别领域和人脸识别领域都占据着主要地位,但是在指纹识别领域,基于深度学习方法的研究工作还非常少,是否能使用深度学习方法完全替代传统的指纹识别方法,得到更高效更精确的识别算法也是接下来的研究重点。

在小面积指纹识别方面,还有存在着一个重要的研究方向我们没有涉及,即如何在资源配置极低的嵌入式环境中进行小面积指纹识别,这是一个极具挑战性但重要的问题。其实际应用案例也处在迅猛发展阶段,例如指纹门锁、指纹箱包等。在这些传统物件中添加嵌入式的指纹识别系统也已经成为一个发展趋势。而在这些情况下,指纹算法所使用的计算资源、内存资源以及存储空间都将被压缩到极致,在这种情况下是否还能进行安全、可靠、高效的指纹识别,如果存

在,那么是否还能继续减少资源。这些问题都是小面积指纹识别算法即将面临的重要问题,对这些问题的研究,也是小面积指纹识别走向成熟的必要研究。