

基于端叉特征融合的指纹识别算法

李昊翔,陈玉明*,吴克寿

(厦门理工学院 计算机与信息工程学院,福建 厦门 361000)

摘要:生物特征识别有多种多样,例如人脸识别、指纹识别、DNA基因测序等,以往各种文献中,大多数采用复杂算法或是深度学习的方法,其实时性不强。本文针对指纹纹路繁杂的特点,去繁取简抽取纹路的端叉关键点特征,利用图像处理技术和最新的OpenCV4实现指纹识别算法,避免了复杂算法和深度学习中计算耗时的问题。首先,利用图像处理的相关技术对采集的图像进行预处理,包括剪裁,旋转,多种滤波;其次,构建指纹端叉融合的特征提取算法确定相应的类型和角度,并绘制出待识别指纹特征融合关键点个数1670个;最后对指纹识别功能进行测试得出结论。本算法特征提取的平均耗时为47.0 ms,平均匹配时间约为7.7 ms,同时准确率在不同特征提取算法中最高,为93.8%。由此得出该算法能够快速准确地对指纹库的指纹进行识别与比对,有效提高了指纹识别的精度与效率。

关键词:指纹识别;端叉融合;图像处理;直方图均衡化;OpenCV

中图分类号:O436 文献标志码:A 文章编号:0253-2395(2024)01-0009-09

Fingerprint Recognition Algorithm Based on Feature Fusion of Endpoints and Bifurcation Points

LI Haoxiang, CHEN Yuming*, WU Keshou

(School of Computer and Information Engineering, Xiamen University of Technology, Xiamen 361000, China)

Abstract: There are various types of biometric recognition, such as face recognition, fingerprint recognition, DNA gene sequencing, etc. In literature, most of the methods used are complex algorithms or deep learning methods, which are insufficient in real-time effectiveness. In this paper, aiming at the characteristics of complex fingerprint patterns, we use image processing technology and the latest OpenCV4 to implement fingerprint recognition algorithms, avoiding complex algorithms and time-consuming computation in deep learning. Firstly, the collected images are preprocessed using image processing related technologies, including clipping, rotation, and multiple filtering; Secondly, a feature extraction algorithm for fingerprint cross fusion is constructed to determine the corresponding types and angles, and 1670 key points for fingerprint feature fusion to be identified are plotted; Finally, the fingerprint recognition functionality is tested, and the following conclusions are drawn from the experiments. The average time consumed by the feature extraction of this algorithm is 47.0 ms, and the average matching time is approximately 7.7 ms. Additionally, the accuracy is the highest among various feature extraction algorithms, reaching 93.8%. Therefore, it can be concluded that this algorithm enables fast and accurate identification and matching of fingerprints in the fingerprint database, effectively improving the precision and efficiency of fingerprint recognition.

Key words: fingerprint identification; end fork fusion; image processing; histogram equalization; OpenCV

收稿日期:2023-06-06;接受日期:2023-08-10

基金项目:国家自然科学基金(61976183)

作者简介:李昊翔(1999-),男,江西上饶人,硕士研究生,主要研究方向为机器学习、图像处理。E-mail:lihaoxiang13@foxmail.com

*通信作者:陈玉明(CHEN Yuming),E-mail:ymchen@xmut.edu.cn

引文格式:李昊翔,陈玉明,吴克寿.基于端叉特征融合的指纹识别算法[J].山西大学学报(自然科学版),2024,47(1):9-17. DOI:10.13451/j.sxu.ns.2023136

0 引言

随着计算机技术、人工智能的飞速发展,计算机视觉、机器学习、深度学习得到了人们更深层次的了解与研究,与之而来的则是人体生物特征,如指纹、虹膜、人脸等得到了广泛的应用。

邱潇利用奇异点追踪算法提取指纹特征^[1]。Cao等利用获取的指纹方向场和脊线流为输入特征向量,再利用支持向量机和K近邻融合分类器,完成指纹分类^[2]。Khodadoust等使用指纹方向来索引、检索其细节的方法完成指纹的识别^[3]。Hong等以Gabor滤波器响应、奇异点和脊线流为输入特征向量,以支持向量机为分类器,完成指纹分类识别^[4]。Aravindan等利用基于小波SIFT(尺度不变特征变换,Scale invariant Feature Transform)描述符的稳健部分完成指纹识别^[5]。陈斯宇提出一种改进的OPTA(One-Pass Thinning Algorithm)指纹细化算法,以Gabor滤波器增强指纹图像、修改保留模板和消除模板,完成指纹图像的细化^[6]。Kawagoe和Tojo提出Poincare索引算法,来完成指纹识别^[7],但这一方法易受各种因素影响,较容易检测出错误的奇异点^[8]。因此,王超岚利用改进Poincare索引、提取奇异点的方法实现了指纹的识别^[9]。除此之外还有Win等^[10]提出的CM(Correlation Matching)相似度匹配方法,以及Chen等^[11]提出的LTSM(Local Topological Structure Matching)局部特征匹配方法等等。

近年来,也有很多国内外的学者将深度学习引入到指纹的分类和识别上来,例如许丹丹利用CNN(Convolutional Neural Networks)、VG-GNet-19(Visual Geometry Group 19)进行特征提取与融合,再采用三层全连接网络进行指纹识别,最后通过计算相似度,得到最佳匹配的指纹^[12]。Maio等利用ResNet50模型训练FVC2004数据集(该数据集广泛用于指纹识别算法的研究和性能评估。它提供了DB1、DB2、DB3、DB4四个库。每个库中分两个数据集,数据集A中有来自100个不同的指纹图像,每个指纹存有8幅不同次按压的图像,共计800张;数据集B中有来自10个不同次按压的图像,共

计80张;因此每个数据库共计 $880 \times 4 = 3\,520$ 张图像,涵盖了不同的指纹图像特点和变化情况,以便更全面地评估算法的鲁棒性和准确性。),将训练好的模型用于 150×150 大小的指纹灰度图像的分类中^[13]。汤春阳利用深度学习中射频的方法实现了从物理层识别用户身份指纹^[14]。Pandya等^[15]提出了一种带有3层卷积层和2层全连接层的CNN网络,输入 350×233 大小的自己采集的指纹数据集。吴震东等^[16]提出利用CBF-FFPF(Central Block Fingerprint and Fuzzy Feature Points Fingerprint)算法提取指纹中心点分块图像及特征点模糊化图,再合并输入CNN网络对其进行识别。钟瑞^[17]提出了一种基于深度学习和多特征融合的小面积指纹识别方法算法。

上述研究者们提出的算法虽然对指纹识别有良好的效果,但都没兼顾效率、模型、数据量三者之间的平衡关系。本文提出端叉点特征融合的指纹识别算法,先对图像进行预处理,接着对关键点类型和角度进行计算,得出端叉特征并进行融合、绘制,然后利用绘制的待识别指纹的关键点与指纹库进行逐一匹配,最后根据匹配到的关键点个数最大原则选取相应的指纹作为识别的结果。该算法不仅简洁明了而且更加便于实际应用,例如公安取证后的指纹比对,学生上课签到时指纹比对等等。

1 图像预处理技术和OpenCV算法框架

数字图像最早应用之一是在报纸业,当时图片首次通过海底电缆从伦敦送到了纽约。目前的图像处理技术几乎已经延伸至各个技术领域,例如核医学,天文学,遥感、气象观测等^[18]。相比于获取图像,应更加注重于对图像信息进行处理,在大量复杂的图像中找出所需要的信息。因此图像信息处理在某种意义上讲,比获得图像更为重要^[19]。现如今人们对图像信息的处理提出了更高的要求,以便获得更加快速、准确、有用的信息。

直方图均衡化后的指纹图像作为后续实验的输入图片,因为对于指纹图像来说,在理想的状态下的像素点分布应该是这样的:在脊线区域中,应该有较小的灰度值;在谷线区域中,

应该有一定的灰度值,并且在反映灰度直方图上应该有两个较明显的峰值,而且这两个峰值之间应有足够的距离^[20]。对于对比度不是很好的图像我们就可以利用直方图均衡化对 L 个灰度级灰度值进行动态拉伸,从而达到图像增强的效果。具体的操作步骤如下所示:

设指纹的灰度值是连续的, r 为待处理的图像的灰度,其值域在 $[0, L-1]$ 。 $r=0$ 表示黑, $r=1$ 表示白。得出数字图像中出现的灰度级 r_k 的概率公式近似为^[21]:

$$p_r(r_k) = \frac{n_k}{MN}, \quad (1)$$

MN 为像素总数, n_k 表示灰度值为 r_k 的像素数。

由此得出像素变换函数公式为^[22]:

$$S_k = (L-1) \sum_{j=0}^k p_r(r_j), \quad (2)$$

S_k 为 r_k 的每个像素映射输出后所对应的像素值, $K \in (0, 1, 2, \dots, L-1)$ 。举个简单的例子说明直方图的均衡变化具体的实现步骤:假设指纹的大小为 $64 \text{ px} \times 64 \text{ px}$,具体灰度分布及直方图的值如表1所示。

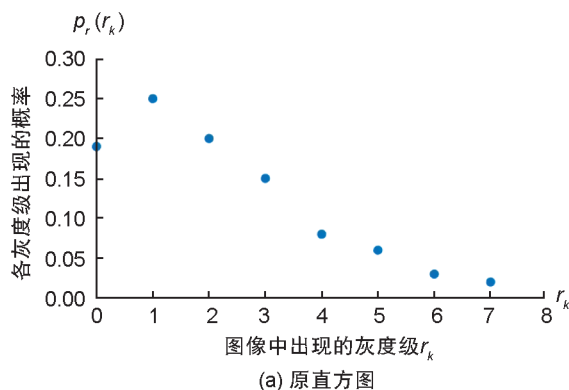
表1 具体灰度及像素数的值

Table 1 Specific grayscale and pixel number values

| r_k | n_k | $p_r(r_k)$ | r_k | n_k | $p_r(r_k)$ |
|---------|-------|------------|---------|-------|------------|
| $r_0=0$ | 796 | 0.19 | $r_4=4$ | 332 | 0.08 |
| $r_1=1$ | 1028 | 0.25 | $r_5=5$ | 268 | 0.06 |
| $r_2=2$ | 821 | 0.20 | $r_6=6$ | 134 | 0.03 |
| $r_3=3$ | 634 | 0.15 | $r_7=7$ | 86 | 0.02 |

具体步骤:

(1) 先利用公式(1)得到灰度级 r_0 的概率近似 $p_r(r_0)=0.19$ 。



$$p_r(r_0) = \frac{n_k}{MN} = \frac{790}{64 \times 64} = 0.19。$$

(2) 再由公式(2)得出变换后的 S_0 像素值为1.33。

$$S_0 = (8-1) \sum_{j=0}^0 p_r(r_0) = 1.33。$$

(3) 因为图像为数字图像,灰度值必须为整,因此需要四舍五入进行取整^[23],所以得出的 S_0 的值就为1。

依次循环步骤(1)–(3),得出 $S_1=3, S_2=4, S_3=6, S_4=6, S_5=7, S_6=7, S_7=7$,由此可得出 S_k 的值为1, 3, 4, 6, 7。同时将对应的 $p_r(r_k)$ 合并,记作 $p_s(s_k)$ 。绘制原直方图及均衡化后的直方图,如图1所示。

虽然原直方图可以通过拉伸变换使其均衡化,但像素的变化必须满足以下两个条件:1. 原始图像各灰度级在经过直方图均衡化后仍要保持原来的排列顺序。2. 变换前后的总体灰度范围保持不变。

OpenCV (Open Source Computer Vision) 是一个开源程序库,是由2 500多个优化算法组合而成,虽然它由C++语言编写并经过深度优化,但也提供了易于使用的计算机视觉接口,便于建立精巧的视觉应用。开发者们可以根据其识别准确度较高、占用系统资源较少,具有响应时间快等特点设计出高效的人工智能系统。在OpenCV提供的开源并且优化过的代码基础上继续开发,使其具有很强的可读性和移植性^[24]。

2 指纹图像处理与端叉特征融合算法

2.1 灰度化及色彩空间的转化

在处理彩色照片的时候,第一步通常都需

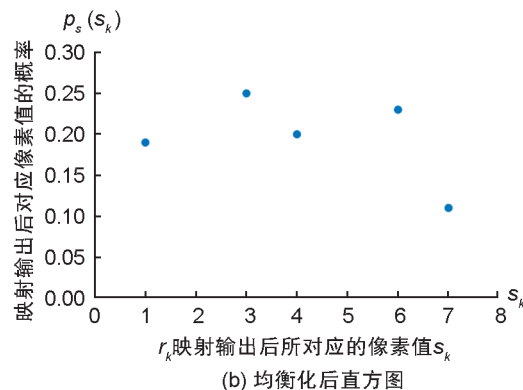


图1 8个不同灰度级直方图均衡示例

Fig. 1 Eight examples of histogram equalization with different grayscale levels

要将彩色图像中含有的大量颜色信息进行灰度化处理。那是因为如果直接对原始的彩色图像进行处理,既浪费了计算机资源且效率低下又对实验没有帮助。在将彩色图像转为灰度图像后,则需要将灰度图像转为HSV模式以便于机器更好地识别图像。而HSV是一种将RGB色彩空间中的点用倒圆锥体的方式表示。HSV即色相(Hue, H)、饱和度(Saturation, S)、亮度(Value, V)(色相(H):用角度度量的方式表示颜色;饱和度(S)表示颜色接近光谱色的程度。亮度(V)表示颜色明亮的程度,对于光源色,明度值与发光体的光亮度有关^[25])。

2.2 图像去噪

图像去噪方法有均值滤波、高斯滤波、中值滤波等。不同的滤波实现的效果各有差异,本文针对各种滤波和图像增强方法对指纹(如图2(a)所示)分别作出实验。均值滤波是典型的线性滤波算法,是指用当前像素点周围 $n \times n$ 个像素值的均值来代替当前像素值^[26]。本文采用的是 5×5 大小的卷积核对图像的每一个像素点进行遍历处理,得到效果图如图2(b)所

示。高斯滤波是针对一个模板(或称卷积、掩模)扫描图像中的每一个像素,用模板确定邻域内像素的加权平均灰度值去替代模板中心像素点的值^[27],得到效果图如图2(c)所示。而直方图均衡化是一种简单、快速且有效的图像增强算法,它通过对原始图像按照累积分布函数进行变换,将分布不均的直方图拉伸均匀分布,从而增强图像对比度,使图像更清晰^[28]。如图3(a)所示为未进行均衡化的指纹直方图,可以看出指纹的脊线和谷线对比度很不明显。经过公式(1)和(2)处理后,得到如图3(b)所示经过均衡化后的直方图,可以看出均衡化后明显提高了指纹脊线和谷线的对比度。图2(d)为均衡化后的指纹图。由于均值滤波及高斯滤波对指纹灰度值和脊、谷线的作用并不明显,对本次实验效果并未有很大的帮助,所以综合以上各种滤波和直方图均衡化后的指纹图效果,本文最终采用直方图均衡化后的指纹作为处理后的指纹图片,并进行后面识别匹配的过程。

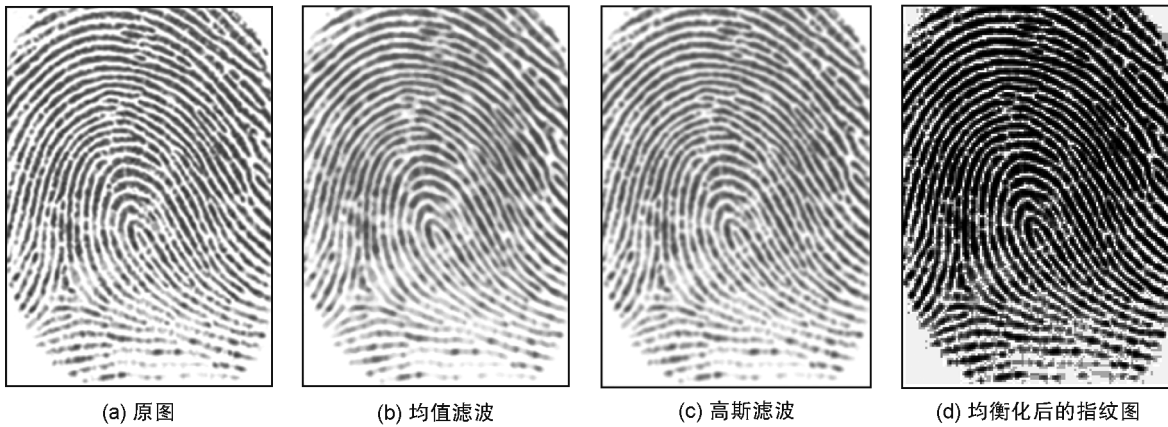


图2 各种去噪效果对比图

Fig. 2 Comparison of various noise removal effects

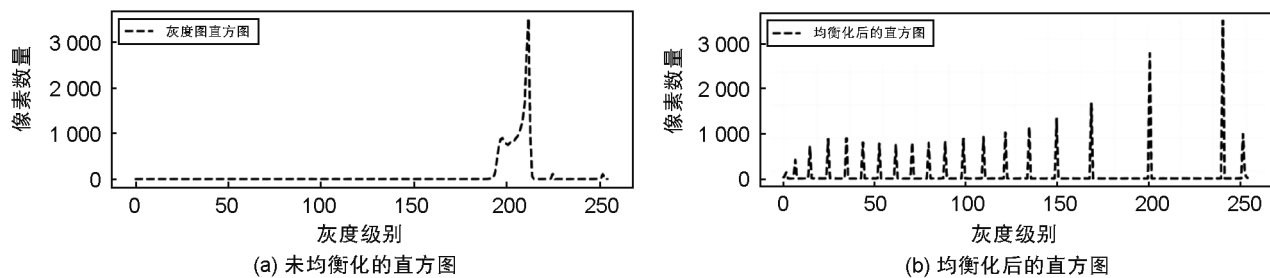


图3 直方图均衡化效果对比图

Fig. 3 Comparison diagram of histogram equalization effect

2.3 端叉融合算法原理

人体指纹的特征可以反映人类群体中来自不同个体不同手指的指纹相似程度^[29]。指纹的细节特征主要指的是纹线端点(Ridge Ending)和纹线分叉点(Ridge Bifurcation)^[30]。纹线端点指的是纹线突然结束的位置,而纹线分叉点则是指纹线一分为二的位置。大量的统计结果表明,使用这两类特征点就足以描述指纹的唯一性。具体实现方法为:

(1) 基于处理后的指纹图像,默认黑色背景(有指纹像素的)=1,白色背景(没有指纹像素点的)=0,指纹图像长为 M 个像素,宽有 N 个像素。

(2) 根据目标点 P_0 周围的8邻域分布(例如如图4所示),计算目标点 P_0 的交叉数(crossing number,简称为CN),并将交叉数定义为8个邻域中相邻像素点之间灰度差异总和的一半^[31]。原公式如公式(3)所示:

$$CN(P_0) = \frac{1}{2} \times \frac{1}{255} \sum_{i=1}^8 |g(p_{(i+1) \bmod 8}) - g(p_i)| \quad (3)$$

但由于该公式中的每个像素都要乘以 $\frac{1}{255}$ 进行归一化处理,所以造成了计算的复杂化,基于上述问题,作出优化:因为在2.1节已对图像进行了预处理,将色彩图像转为灰度图像并归一化,所以该图像每个像素点除了0就是1,就没必要再进行乘以 $\frac{1}{255}$ 的操作,所以可将公式简化为如式(4)所示:

$$CN(P_0) = \frac{1}{2} \sum_{i=1}^8 |P_i - P_{i+1}| (P_9 = P_1) \quad (4)$$

(3) 按照公式(4)计算图4(a)的目标点邻域像素分布图1,得到目标点 P_0 的交叉数 $CN(P_0)=1$ 。

同理计算得出分布图2目标点 P_0 的交叉数 $CN(P_0)=3$ 。

(4) 根据计算得到的像素点分布结果,如果得到的 $CN(P_0)=1$ 就为端点, $CN(P_0)=3$ 就为分叉点,如果得到的 $CN(P_0)$ 既不等于1,又不等于3,那就是普通的像素点。

(5) 依次向右平滑一个像素单位,得到下一个像素点领域的像素分布图,往复(2)—(4)的操作步骤,最后将所得的计算结果按照计算顺序相加融合在一起,就得到了整个指纹图像的端叉点分布情况。

在得到整个指纹图像的端叉点分布情况后,则需要根据关键点角度来确定指纹端叉点方向。计算时从关键点开始,沿着脊线,离关键点大约15个像素处的点A,在关键点到A画一条水平的切线,该直线与水平坐标轴(X)的夹角就是关键点分叉的角度。如图5所示。

2.4 指纹端叉关键点的绘制

指纹的关键点由不同颜色表示:蓝色点表示端点,绿色点表示叉点,红色点表示中心点。绘制指纹的关键点伪代码如下:
`sift = SIFT_create()`, `kp, des = sift.detectAndCompute(img, 掩膜(None))`, `kp`(关键点列表,每一个元素是一个关键点):关键点;
`des`(128维向量组成的列表):关键点描述符。关键点包含的信息:
`pt`:关键点的坐标;
`size`:描述关键点的区域;
`angle`:角度,表示关键点的方向;
`response`:响应程度,代表该关键点特征的独特性,越高越好;
`octave`:表示从金字塔哪一层提取;
`class_id`:当要对图像进行分类时,可以通过该参数对每个关键点进行区分,未设定时默认为-1;最后用

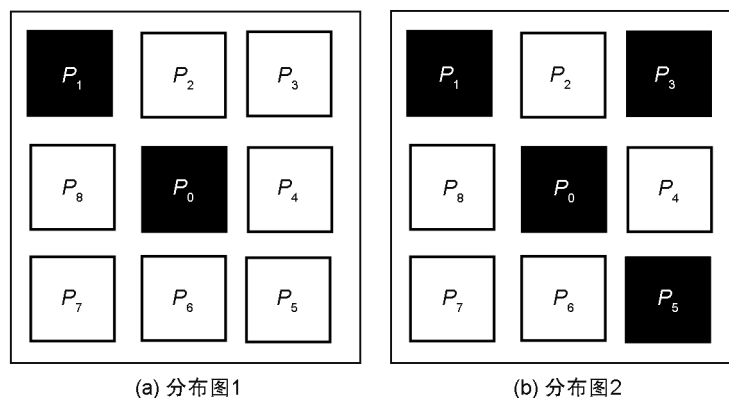


图4 目标点 P_0 邻域像素分布示意图

Fig. 4 Schematic diagram of pixel distribution in the target point P_0 domain

`cv2.drawKeypoints(src_img, kp, output_img)` 函数对该指纹的关键点进行绘制, 得到图 6。

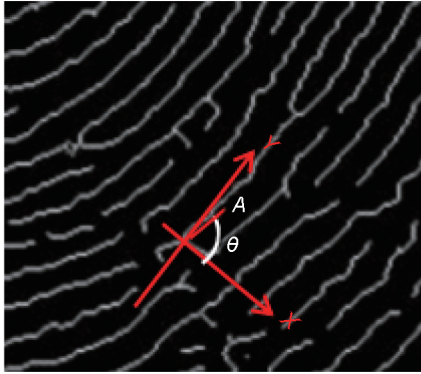


图 5 确定指纹关键点角度示意图

Fig. 5 Schematic diagram of determining the angle of fingerprint key points

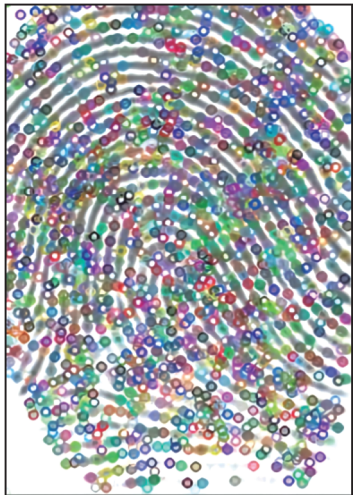


图 6 指纹关键点效果图

Fig. 6 Fingerprint key point rendering

在 FVC2004 的数据集中, 我们选取了同为根据指纹特征且性能较为优越的模型进行比较。本算法根据提取端点及分叉点融合特征的方法, 绘制每个指纹的关键点, 所用平均时间为 47 ms, Aravindan 等^[5]利用小波 SIFT 来提取指纹稳健部分的特征, 平均耗时约为 105 ms。

利用奇异点追踪算法提取指纹特征, 平均耗时约为 32 ms, Khodadoust 等^[3]使用指纹方向来索引、检索其细节的方法, 平均耗时约为 20 ms。具体耗时如表 2。

2.5 指纹端叉融合算法

算法 1 端叉融合算法

输入: 指纹数据集 $U = \{X_1, X_1, \dots, X_k, \dots, X_n\}$, 其中 X_k 为 RGB 三通道的图像。

表 2 不同特征提取算法运行时间对比

Table 2 Comparison of running times of different feature extraction algorithms

| 算法 | 算法名称 | 特征提取时间/ms |
|-------------|------------|-----------|
| 本文 | 端叉特征融合 | 47 |
| Aravindan 等 | 小波 SIFT 描述 | 105 |
| 邱潇 | 奇异点追踪 | 34 |
| Khodadoust | 纹路方向场索引 | 20 |

输出: 端叉点特征融合的指纹灰度图像 P_k , 像素值范围为 $[0, 1]$ 。

(1) 从指纹数据集 U 中随机抽取样本指纹图像 X_k 。

(2) 对抽取的指纹图像 X_k 进行预处理, 预处理包括旋转、剪裁、灰度化、归一化、直方图均衡化, 得到归一化后的灰度图像 X_k' , 其像素值范围为 $[0, 1]$ 。

(3) 根据图像 X_k' 构造二维矩阵 G , 因图像 X_k' 长有 M 个像素点, 宽有 N 个像素点, 得出矩阵 $G(M \times N)$ 。

(4) 再构造一个矩阵 $G_1(3 \times 3)$, 中心点 P_0 坐标设为 $(1, 1)$, 由此得出 $P_1 - P_9$ 的坐标, 为了便于计算交叉数 CN , 可使 $P_1 = P_9$ 。

(5) 使 $G_1(3 \times 3)$ 的 $(0, 0)$ 点与 $G(M \times N)$ 的 $(0, 0)$ 点对齐, 并计算 G_1 中目标点领域像素分布。

(6) 依次向右移动一个像素点, 并计算:

for i from 1 to $N-2$ do

for j from 1 to $M-2$ do

for k from 1 to 8 do

$$(5.1) CN += \frac{1}{2} \times (\text{abs}(P_k - P_{k+1}))$$

(5.2) if $CN == 1$ 为端点

else

if $CN == 3$ 为叉点

else 为普通像素点

(7) end for

(8) end for

(9) end for

(10) 将指纹图像的端叉点特征进行相加融合。

(11) 确定 D_k 指纹纹路方向。

(12) 绘制 D_k 端叉关键点得到 P_k 。

3 指纹的匹配及实现过程

3.1 匹配指纹库

先根据上述方法绘制其端叉点特征,并计算特征融合后关键点个数,然后通过函数 $bf = cv2.BFMatcher()$ 、 $bf.knnMatch()$ (特征描述符集合 A , 特征描述符集合 B , k); (k 为最匹配描述符个数, $k = 2$) 和 $drawMatcherKnn()$ (图像 a , a 关键点集合, 图像 b , b 关键点集合, 匹配点, 映射集), 在指纹库内逐一匹配, 最后将关键点个数匹配最多的那枚指纹作为最佳的指纹, 输出其在指纹库中对应的名字及序号。

3.2 实现过程

综合前几节的概述, 可以将指纹匹配归纳为图 7 所示的几个步骤。(1) 收集指纹; (2) 将收集到的指纹进行剪裁旋转; (3) 对指纹进行色彩空间的转化; (4) 实验判断各种滤波及直方图均衡化的处理效果, 得到如图 2(d) 所示的均衡化后的指纹; (5) 通过计算指纹的关键点类型及角度, 确定指纹中每个端点和分支点, 并进行融合; (6) 将指纹中融合的关键点特征进行绘制, 得到图 6 的效果图; (7) 通过绘制得到的指纹关键点个数与数据库中的指纹进行比较, 得到表 3 的匹配结果, 平均匹配时间约为 7.7 ms 与其他算法的平均时间及准确率对比如表 4。如表 3 所示, 文件名为 101_5.tif 的图片的匹配关键点个数为 1 670 个远远高于其他图片所匹配的关键点个数, 所以将文件名为 101_5.tif 所对应的人名 Kobe 输出, 作为此实验最佳的匹配指纹。

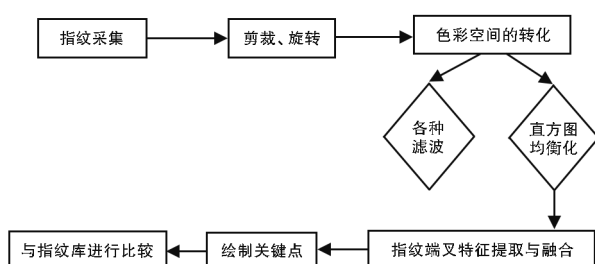


图7 具体操作步骤流程示意图

Fig. 7 Schematic diagram of specific operation steps and processes

4 结论

本算法是基于融合指纹端点与叉点特征的

表3 根据关键点个数得出最佳匹配结果

Table 3 Obtaining the best matching result based on the number of key points

| 文件名 | 匹配关键点个数 | 对应人名 |
|------------|---------|---------|
| 101_1.tif | 97 | Jack |
| 101_2.tif | 43 | Durant |
| 101_3.tif | 99 | Harden |
| 101_4.tif | 65 | James |
| 101_5.tif | 1 670 | Kobe |
| 101_6.tif | 102 | Stephen |
| ⋮ | | |
| 输出结果: Kobe | | |

表4 不同特征提取算法的指纹匹配时间及准确率对比

Table 4 Comparison of fingerprint matching time and accuracy between different feature extraction algorithms

| 算法 | 平均匹配时间/ms | 准确率/% |
|----------------------|-----------|-------|
| 本文(端叉特征融合) | 7.7 | 93.8 |
| Aravindan等(小波SIFT描述) | 4.1 | 93.1 |
| 邱潇(奇异点追踪) | 18.9 | 91.2 |
| Khodadoust等(纹路方向场索引) | 4.3 | 92.6 |

指纹识别方法。在本算法中, 先用数字图像处理技术对指纹进行预处理, 再根据指纹端叉点特征融合的算法确定该指纹端叉点的角度, 并进行特征的融合及绘制, 接着利用绘制后的融合特征进行指纹匹配。由表 2 和表 4 得出虽然本算法在提取端叉融合特征上的平均时间为 47 ms 略微比利用奇异点追踪算法提取指纹特征以及 Khodadoust 等等使用指纹方向来索引、检索其细节的方法要慢, 但在匹配时间上要优于利用奇异点追踪算法, 并在识别准确率上要略微高于其他三种算法。实验证明本算法操作便捷、运行速度快、识别效果好, 因此具备高实用性。

参考文献:

- [1] 邱潇. 低质量残缺指纹的快速识别方法研究[D]. 重庆: 重庆大学, 2017.
QIU X. Research on Fast Identification Method of Low Quality Damaged Fingerprint[D]. Chongqing: Chongqing University, 2017.
- [2] CAO K, PANG L, LIANG J, et al. Fingerprint Classification by a Hierarchical Classifier[J]. *Pattern Recognit*, 2013, 46(12): 3186-3197. DOI: 10.1016/j.patcog.2013.05.008.
- [3] KHODADOUST J, KHODADOUST A M. Partial Fin-

- gerprint Identification for Large Databases[J]. *Pattern Anal Applic*, 2018, **21**(1): 19-34. DOI: 10.1007/s10044-017-0665-0.
- [4] HONG J H, MIN J K, CHO U K, *et al.* Fingerprint Classification Using One-vs-all Support Vector Machines Dynamically Ordered with Naïve Bayes Classifiers[J]. *Pattern Recognit*, 2008, **41**(2): 662-671. DOI: 10.1016/j.patcog.2007.07.004.
- [5] ARAVINDAN A, ANZAR S M. Robust Partial Fingerprint Recognition Using Wavelet SIFT Descriptors[J]. *Pattern Anal Applic*, 2017, **20**(4): 963-979. DOI: 10.1007/s10044-017-0615-x.
- [6] 陈斯宇. 指纹识别图像预处理系统的设计和实现[D]. 长春: 吉林大学, 2015.
CHEN S Y. Design and Implement of Fingerprint Image Preprocessing System Based[D]. Changchun: Jilin University, 2015.
- [7] KAWAGOE M, TOJO A. Fingerprint Pattern Classification[J]. *Pattern Recognit*, 1984, **17**(3): 295-303. DOI: 10.1016/0031-3203(84)90079-7.
- [8] ZHU E, GUO X F, YIN J P. Walking to Singular Points of Fingerprints[J]. *Pattern Recognit*, 2016, **56**: 116-128. DOI: 10.1016/j.patcog.2016.02.015.
- [9] 王超岚. 指纹识别算法的研究[D]. 北京: 北京林业大学, 2019. DOI: 10.26949/d.cnki.gblyu.2019.000615.
WANG C L. Research on Fingerprint Recognition Algorithms[D]. Beijing: Beijing Forestry University, 2019. DOI: 10.26949/d.cnki.gblyu.2019.000615.
- [10] WIN Z M, SEIN M M. Fingerprint recognition system for low quality images[C]//Sice Conference. IEEE Xplore, 2011: 1133-1137.
- [11] CHEN X J, TIAN J, YANG X. A New Algorithm for Distorted Fingerprints Matching Based on Normalized Fuzzy Similarity Measure[J]. *IEEE Trans Image Process*, 2006, **15**(3): 767-776. DOI: 10.1109/TIP.2005.860597.
- [12] 许丹丹. 基于深度学习的多特征指纹图像识别[D]. 阜阳: 阜阳师范大学, 2021. DOI: 10.27846/d.cnki.gfysf.2021.000121.
XU D D. Multi-feature Fingerprint Image Recognition Based on Deep Learning[D]. Fuyang: Fuyang Normal University, 2004. DOI: 10.27846/d.cnki.gfysf.2021.000121.
- [13] MAIO D, MALTONI D, CAPPELLI R, *et al.* FVC2004: Third Fingerprint Verification Competition [M]//Biometric Authentication. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004: 1-7. DOI: 10.1007/978-3-540-25948-0_1.
- [14] 汤春阳. 基于深度学习的射频指纹识别研究[D]. 兰州: 兰州交通大学, 2022. DOI: 10.27205/d.cnki.gltcc.2022.001052.
- TANG C Y. Research on Radio Frequency Fingerprint Recognition Based on Deep Learning[D]. Lanzhou: Lanzhou Jiaotong University, 2022. DOI: 10.27205/d.cnki.gltcc.2022.001052.
- [15] PANDYA B, COSMA G, ALANI A A, *et al.* Fingerprint Classification Using a Deep Convolutional Neural Network[C]//2018 4th International Conference on Information Management (ICIM). 2018: 86-91. DOI: 10.1109/INFOMAN.2018.8392815.
- [16] 吴震东, 王雅妮, 章坚武. 基于深度学习的污损指纹识别研究[J]. 电子与信息学报, 2017, **39**(7): 1585-1591. DOI: 10.11999/JEIT161121.
WU Z D, WANG Y N, ZHANG J W, Fouling and Damaged Fingerprint Recognition Based on Deep Learning [J]. *J Electron Info Technol* 2017, **39**(7): 1585-1591. DOI: 10.11999/JEIT161121.
- [17] 钟瑞. 基于深度学习和多特征融合的小面积指纹识别算法研究[D]. 西安: 西安电子科技大学, 2018.
ZHONG R. Research on Small Area Fingerprint Recognition Algorithm Based on Deep Learning and Multi feature Fusion[D]. Xi'an: Xidian University, 2018.
- [18] 张铮, 徐超, 任淑霞, 等. 数字图像处理与机器视觉 [M]. 北京: 人民邮电出版社, 2014.
ZHANG Z, XU C, REN S X, *et al.* Digital Image Processing and Machine Vision[M]. Beijing: Posts and Telecommunications Press, 2014.
- [19] 梁原. 基于 MATLAB 的数字图像处理系统研究[D]. 长春: 长春理工大学, 2008.
LIANG Y. The Research of Digital Image Preprocessing Based on MATLAB[D]. Changchun: Changchun University of Science and Technology, 2008.
- [20] 简稳. 高效鲁棒指纹识别系统的研究与实现[D]. 上海: 上海交通大学, 2021. DOI: 10.27307/d.cnki.gsjtu.2021.000126.
JIAN W. Research and Implementation of an Efficient Robust Fingerprint Recognition System[D]. Shanghai: Shanghai Jiao Tong University, 2021. DOI: 10.27307/d.cnki.gsjtu.2021.000126.
- [21] MEHTRE B M, MURTHY N N, KAPOOR S, *et al.* Segmentation of Fingerprint Images Using the Directional Image[J]. *Pattern Recognit*, 1987, **20**(4): 429-435. DOI: 10.1016/0031-3203(87)90069-0.
- [22] KIM B G, PARK D J. Adaptive Image Normalisation Based on Block Processing for Enhancement of Fingerprint Image[J]. *Electron Lett*, 2002, **38**(14): 696. DOI: 10.1049/el: 20020507.
- [23] 蔡超峰. 局部直方图均衡化算法研究及其应用[D]. 郑

- 州: 郑州大学, 2005.
- CAI C F. Research and Application of Local Histogram equalization Algorithm[D]. Zhengzhou: Zhengzhou University, 2005.
- [24] 胡明. 基于OpenCV的地铁隧道衬砌裂缝检测系统的设计与实现[D]. 扬州: 扬州大学, 2021. DOI:10.27441/d.cnki.gyzdu.2021.001467.
- HU M. Design and Implementation of Crack Detection of Subway Tunnel System Based on OpenCV[D]. Yangzhou: Yangzhou University, 2021. DOI:10.27441/d.cnki.gyzdu.2021.001467.
- [25] 赵洋, 闵升锋, 李大舟. 基于HSV空间颜色和纹理特征的车牌定位算法研究[J]. 沈阳化工大学学报, 2022, **36**(1):90-95. DOI: 10.3969/j.issn.2095-2198.2022.01.015.
- ZHAO Y, MIN S F, LI D Z. Research on Vehicle License Plate Location Algorithm Based on HSV Spatial Color and Texture Features[J]. *J Shenyang Inst Chem Technol*, 2022, **36**(1):90-95. DOI: 10.3969/j.issn.2095-2198.2022.01.015.
- [26] 陈家益, 黄楠, 熊刚强, 等. 基于置信区间的自适应加权均值滤波算法[J]. 南京理工大学学报, 2017, **41**(3): 307-312. DOI:10.14177/j.cnki.32-1397n.2017.41.03.006.
- CHEN J Y, HUANG N, XIONG G Q, *et al.* Adaptive Weighted Mean Filtering Algorithm Based on Confidence Interval[J]. *J Nanjing Univ Sci Technol*, 2017, **41**(3): 307-312. DOI: 10.14177/j.cnki.32-1397n.2017.41.03.006.
- [27] 赵婉月. 基于YOLOv5的目标检测算法研究[D]. 西安: 西安电子科技大学, 2021. DOI:10.27389/d.cnki.gxadu.2021.002918.
- ZHAO W Y. Research on target detection algorithm based on YOLOv5[D]. Xi'an: Xidian University, 2021. DOI:10.27389/d.cnki.gxadu.2021.002918.
- [28] 黄从珊. 直方图均衡化图像增强算法的改进及其OpenCL并行实现[D]. 广州: 暨南大学, 2014.
- HUANG C S. Improvement of Histogram Equalization Image Enhancement Algorithm and Implementation in OpenCL[D]. Guangzhou: Jinan University, 2014.
- [29] 邬伟杰. 基于支持向量机的指纹识别技术与实现方法研究[D]. 广州: 华东理工大学, 2015.
- WU W J. Fingerprint Identification Techniques Based on Support Vector Machines and Realization[D]. Guangzhou: East China University of Science and Technology, 2015.
- [30] ARCELLI C, DI BAJA G S. A Width-independent Fast Thinning Algorithm[J]. *IEEE Trans Pattern Anal Mach Intell*, 1985, PAMI-7(4): 463-474. DOI: 10.1109/TPAMI.1985.4767685.
- [31] GAO M T, SUN H, TANG Y Q, *et al.* A Fingerprint Secondary Feature Detection Method Based on Improved YOLOv5[J]. *Laser Optoelectronics P*, 2023, **60** (10): 1010006-1010006-11. DOI: 10.3390/diagnostics12112591.