

## 基于支持向量机的指纹识别

姓 名： 王宇轩 学 号： 08121205

完成日期： 2024. 6. 7 审阅教授： 杜松林

# 目 录

一， 算法简介与应用背景 .....	1
二， 算法描述 .....	1
三， 数据集描述 .....	2
四， 图像处理与特征提取 .....	3
五， 实验结果与分析 .....	6
六， 创新描述 .....	10
七， 实验总结与心得体会 .....	11
八， 参考文献 .....	11

## 一，算法简介与应用背景

支持向量机 (SVM) 是一种典型的二分类模型，它将实例的特征向量映射到多维空间中，寻求一条线（二位特征）来划分两类特征以实现分类的目的。

SVM 最早是由 Vladimir N. Vapnik 和 Alexey Ya. Chervonenkis 在 1963 年提出，目前的版本 (soft margin) 是由 Corinna Cortes 和 Vapnik 在 1993 年提出，并在 1995 年发表。深度学习 (2012) 出现之前，SVM 被认为机器学习中近十几年来最成功，表现最好的算法。SVM 具有良好的鲁棒性，对未知数据拥有很强的泛化能力，特别是在数据量较少的情况下，相较于其他传统机器学习算法具有更优的性能。

指纹 (Fingerprint) 识别是将识别对象的指纹进行分类比对从而进行判别。指纹识别技术是众多生物特征识别技术中的一种，所谓生物特征识别技术 (biometrics)，系指利用人体所固有的生理特征或行为特征来进行个人身份鉴定，由于生物识别所具有的便捷与安全等优点使得生物识别技术在身份认证识别和网络安全领域拥有广阔的应用前景，可用的生物特征识别技术有指纹、人脸、声纹、虹膜等，指纹是其中应用最为广泛的一种。现如今指纹识别技术已经走进生活，从指纹门锁到智能手机，指纹识别是如今最常用的身份认证识别技术之一。

## 二，算法描述

我们首先设定一个特征空间上的训练集为：

$$T = \{(x_1, y_1), (x_2, y_2), \dots, (x_N, y_N)\}$$

其中  $x_i \in R^n$  (实数)， $y_i$  为训练集的标签。 $N$  表示有  $N$  个训练样本。

为了更好的将训练数据分割开来，以便于将特征值映射到更高维度，svm 引入了核函数的概念，即某种确定的特征空间转换函数  $\Phi(x)$ 。

这样，支持向量机的数学模型便可以由以下的方程表示：

$$y(x) = \omega^T \Phi(x) + b$$

$\omega$  表示超平面的法向量， $b$  表示超平面相对于坐标原点的偏移量。因此样本空间中任意点到超平面的距离可以写成：

$$r = \frac{|\omega^T \Phi(x) + b|}{\|\omega\|}$$

而支持向量机分类的标准即为  $f(x) = \text{sign}(\omega^T \Phi(x) + b)$ ，简单而言便是在平面上方的为一类，在平面下方的为另外一类。

上述为支持向量机进行二分类方法时的原理，那么如何获取超平面的法向量 $\omega$ 和偏移量 $b$ 便是 svm 优化的核心问题。

评价支持向量机划分质量的一个因素便是样本空间内各向量点相对于超平面的距离，距离越大证明分割效果越好。但是在调整参数的同时需保证向量点的分割不会产生错误，即：

$$\begin{cases} \omega^T \Phi(x_i) + b \geq 0, y_i = 1 \\ \omega^T \Phi(x_i) + b \leq 0, y_i = -1 \end{cases}$$

两个异类支持向量到超平面的距离之和为：

$$y = \frac{2}{\|\omega\|}$$

因此迭代的条件便是：求得最大的 $\frac{2}{\|\omega\|}$ ，使得 $y_i(\omega^T \Phi(x_i) + b) \geq 0$  满足。

而对于多分类问题，我们可以采用以下两种方式进行分类：

#### 1) 一对多法 (one-versus-rest 简称 OVR SVMs)

训练时依次把某个类别的样本归为一类，其他剩余的样本归为另一类，这样  $k$  个类别的样本就构造出了  $k$  个 SVM。分类时将未知样本分类为具有最大分类函数值的那类。但是这种方法存在缺陷，训练数据集不平衡，因而不是很实用，本次作业拟采用 OVO 方法进行测试

#### 2) 一对一法 (one-versus-one 简称 OVO SVMs)

其做法是在任意两类样本之间设计一个 SVM，因此  $k$  个类别的样本就需要设计  $k(k-1)/2$  个 SVM。当对一个未知样本进行分类时，最后得票最多的类别即为该未知样本的类别，这种策略称为“投票法”。libsvm 中的多类分类就是根据这个方法实现的。

假设有四类 A, B, C, D 四类。在训练的时候我们选择(A, B)、(A, C)、(A, D)、(B, C)、(B, D)、(C, D)所对应的向量作为训练集，然后得到六个训练结果，在测试的时候，把对应的向量分别对六个结果进行测试，然后采取投票形式，最后得到一组结果。投票是这样的：

```
A=B=C=D=0
(A, B)-classifier: if A win, then A = A + 1
(A, C)-classifier: if A win, then A = A + 1
...
(C, D)-classifier: if C win, then C = C + 1
The decision is the MAX(A, B, C, D)
```

## 三，数据集描述

测试所采用的数据集为 FVC2004 指纹数据集中的 DB3B，即使用 Atmel 的热扫描传感器

“FingerChip FCD4B14CB”采集到的数据。数据集中共有 11 位志愿者提供的总计 88 张指纹图片信息，每个人提供 8 张相同手指、不同按压力度、不同按压角度以及手指被干燥（印痕 1 和印痕 2）和湿润（印痕 3 和印痕 4）的情况下的采集数据。

**FVC2004**

Home

Background

Categories

Protocol

**Databases**

Perf. evaluation

Participants

Results

Download

**Past editions**

FVC2000

FVC2002

**Participant area**

Register

Login

**Databases**

Four different databases (DB1, DB2, DB3 and DB4) were collected by using the following sensors/technologies:

- DB1: optical sensor "V300" by CrossMatch
- DB2: optical sensor "U.are.U 4000" by Digital Persona
- DB3: thermal sweeping sensor "FingerChip FCD4B14CB" by Atmel
- DB4: [synthetic fingerprint generation](#)

NOTE: FVC2004 databases are markedly more difficult than FVC2002 and FVC2000 ones, due to the perturbations deliberately introduced (see below). Therefore one should neither compare error rates among different FVC competitions, nor conclude that the state-of-the-art in fingerprint matching is not improving.

Students (24 years old on the average) enrolled in the Computer Science degree program at the University of Bologna kindly agreed to act as volunteers for providing fingerprints:

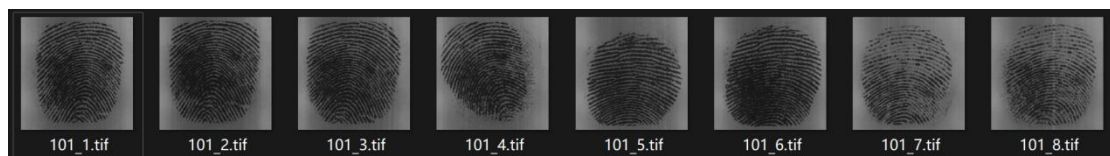
- volunteers were randomly partitioned into three groups of 30 persons; each group was associated to a DB and therefore to a different fingerprint scanner (different subjects in each database, except for a small overlap of five volunteers present in two databases);
- each volunteer was invited to present him/herself at the collection place in three distinct sessions, with at least two weeks time separating each session;
- forefinger and middle finger of both the hands (four fingers total) of each volunteer were acquired by interleaving the acquisition of the different fingers to maximize differences in finger placement;
- no efforts were made to control image quality and the sensor platens were not systematically cleaned;
- at each session, four impressions were acquired of each of the four fingers of each volunteer;
- during the first sessions, individuals were asked to put the finger at a slightly different vertical position (in impressions 1 and 2) and to alternate low and high pressure against the sensor surface (impressions 3 and 4);
- during the second session, individuals were requested to exaggerate skin distortion (impressions 1 and 2) and rotation (3 and 4) of the finger;
- during the third session, fingers were dried (impressions 1 and 2) and moistened (3 and 4).

At the end of the data collection, for each database a total of 120 fingers and 12 impressions per finger (1440 impressions) were gathered. As in previous editions, the size of each database to be used in the test was established as 110 fingers wide (w) and 8 impressions per finger deep (d) (880 fingerprints in all); collecting some additional data gave a margin in case of collection/labeling errors.

Fingers from 101 to 110 (set B) have been made available to the participants to allow parameter tuning before the submission of the algorithms; the benchmark is then constituted by fingers numbered from 1 to 100 (set A).

	Sensor Type	Image Size	Set A (wxd)	Set B (wxd)	Resolution
DB1	Optical Sensor	640x480 (307 Kpixels)	100x8	10x8	500 dpi
DB2	Optical Sensor	328x364 (119 Kpixels)	100x8	10x8	500 dpi
DB3	Thermal sweeping Sensor	300x480 (144 Kpixels)	100x8	10x8	512 dpi
DB4	SFinGe v3.0	288x384 (108 Kpixels)	100x8	10x8	about 500 dpi

其中相同指纹的采集集合如下：



## 四，图像处理与特征提取

采用局部特征和全局特征相结合的方法提取指纹图像的特征值。提取局部特征时需要将指纹图像进行处理，具体处理方法利用了图像增强和指纹脊线细化。

图像增强是将原本的 rgb 图像转化为二值图，从而消除图像中的噪声点。细化则是将脊线细化为 2-3 像素点粗细的线条，便于提取局部特征点。图像增强前后对比图如下：



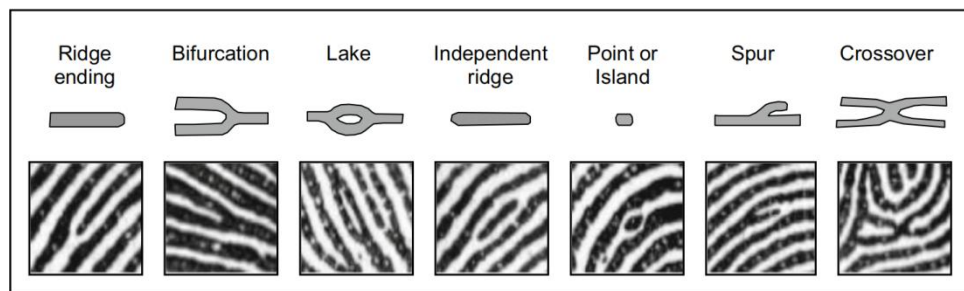
从左到右分别为原图、增强图和细化图。将图像处理至上述细化效果即可进行下一步局部特征提取操作。

指纹的特征可以分为从粗到细的三个级别。第一级：脊线方向场和频率图（奇异点是方

向场的特殊点)；第二级：脊线骨架图(细节点是脊线的特殊点)；第三级：脊线的内外轮廓(汗孔即内轮廓)。这些特征都是围绕脊线定义的，是有解剖学意义的。

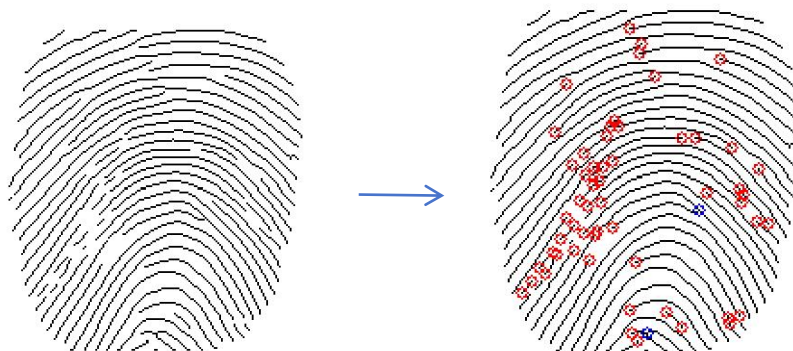
第二级特征记录具体脊线的位置，重点是它结束和分叉的地方，但是并不关注脊线的内外轮廓等细节(脊线不是数学中的线)。脊线突然结束和分叉的地方，称为细节点(minutia)。细节点是人工和自动指纹比对最常用的特征。Galton(1892)最早定义了指纹细节点并观察到它们终生不变。在比较完整的指纹中(如滚动滚动)，细节点的数量可以超过100个。实际中，仅需少量(例如，12-15个)匹配细节点就足以推断两个指纹来自同一根手指。

下面是指纹第二级特征的分类图：

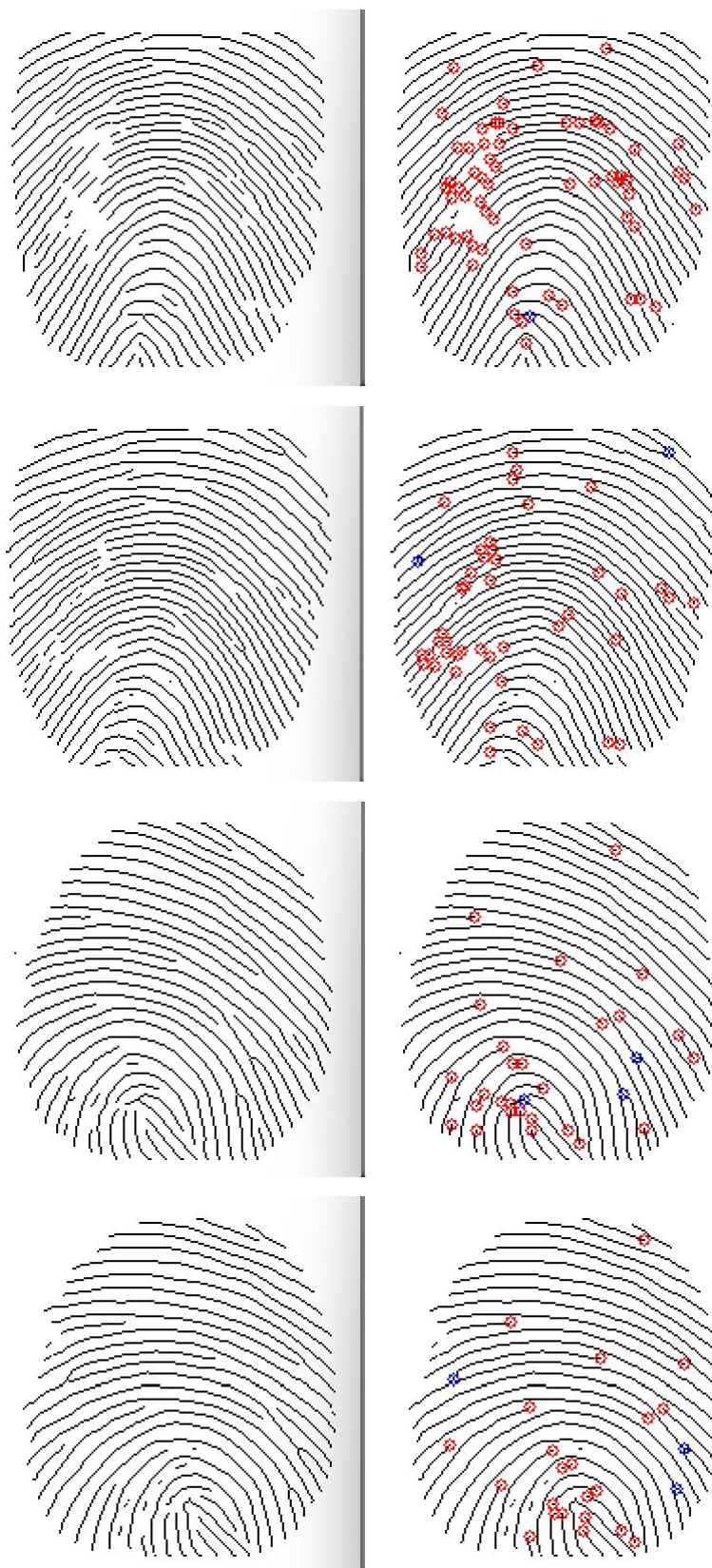


这里仅提取“Ridge ending”和“Bifurcation”作为描述指纹细节特征值。

选取 './DB3B/101\_1.tif' 图片文件检测得到以下特征点：



图中红色点代表检测得到的“Ridge ending”特征点，蓝色代表检测得到的“Bifurcation”特征点。可见基本能将指纹图像中的局部特征点检测得出。下面是上述特征采集方法在更大数据集上的应用：





前后两张图片分别为同一组指纹。他们分割得到的“Ridge ending”和“Bifurcation”的特征点统计数目：

Num	End_Num	Bifur_Num	End_Direction	End_Var	Bifur_Direction	Bifur_Var
1	64	1	-0.0406	3.0431	34.0527	0.0
2	44	2	-0.2896	3.0339	34.0086	0.0818
3	27	3	0.2345	4.1131	33.6946	0.0504
4	22	3	-0.1915	4.4650	33.8145	0.0543

上述局部特征值向量表中，End\_Num 表示“Ridge ending”的数量；Bifur\_Num 表示“Bifurcation”的数量；End\_Direction 和 End\_Var 表示“Ridge ending”特征点的全局方向向量的均值和方差；Bifur\_Direction 和 Bifur\_Var 表示“Bifurcation”特征点的全局方向向量和方差。

观察不同特征点分布可以看出，由于按压力度和角度的不同，指纹特征点的识别也会出现些许不同，但是由于个体间的特征点分布相差大，且往往特征点上的法向量分布存在差异，因此可以结合上述局部特征点进行分类。

下面进行指纹图像全局特征数的提取。

由于整张图片中灰度值的均值、方差能够反应指纹脊线的分布密度和脊线的平均宽度，另外使用图像的能量、熵以及对比度来丰富指纹图像特征值的描述性。

将上述得到的局部特征值和全局特征值组合成为描述指纹图像特征的特征向量，并将其送至支持向量机中进行监督学习。特征向量为：

$$\text{FeatureMat} = [\text{End\_Num}, \text{Bifur\_Num}, \text{End\_Direction}, \text{End\_Var}, \\ \text{Bifur\_Direction}, \text{Bifur\_Var}, \text{Gray\_Mean}, \text{Gray\_Var}, \text{Energy}, \text{Entropy}, \text{Contrast}]$$

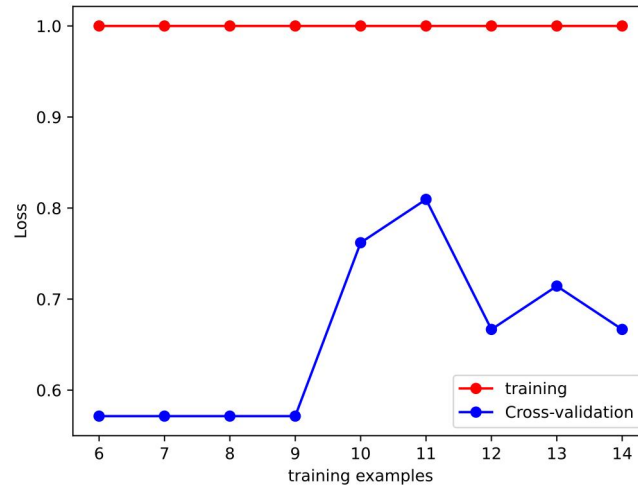
## 五，实验结果与分析

处理所有图片并将特征值写入文件中，编写 dataLoader 函数来写入和读出数据。首先导入 3 组不同指纹，每组 7 张图片作为训练集写入 './DB3\_B/features3.txt'，1 张图片作为测试集写入 './DB3\_B/features3\_test.txt'。声明 SVM 分类器为：



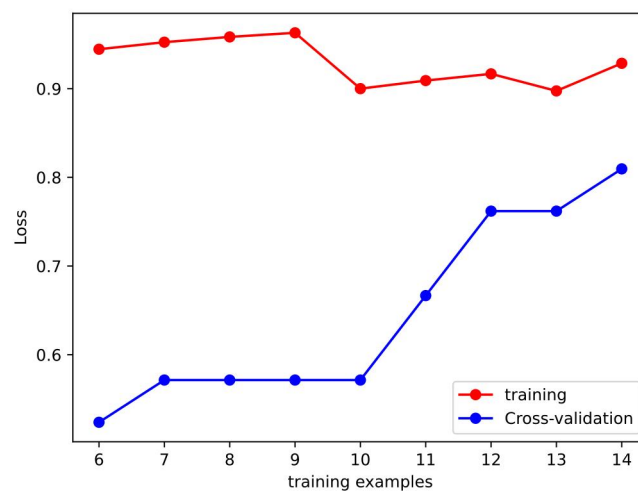
```
clf = svm.SVC(C=100, gamma=0.01, kernel='linear',  
              max_iter=40000, decision_function_shape='ovo')
```

使用交叉验证来观察模型是否训练过拟合或欠拟合，交叉验证结果如下：



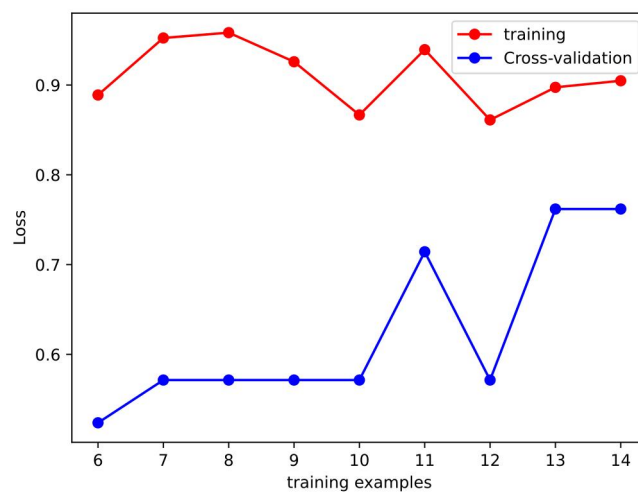
观察图像可知，模型在训练集上的训练效果很好，但是测试集上的效果不理想，模型发生过拟合，直接原因是 SVM 的参数 C 设置过大，即错误项的惩罚系数过大，SVM 分类过于趋近训练集数据，产生了硬间隔，不利于对于指纹特征浮动变化较大的情况，修正参数如下继续测试，修正参数为：

```
clf = svm.SVC(C=0.1, gamma=0.01, kernel='linear',  
              max_iter=40000, decision_function_shape='ovo')
```



可见上述参数较为有效的缓解了过拟合的问题，继续修正参数：

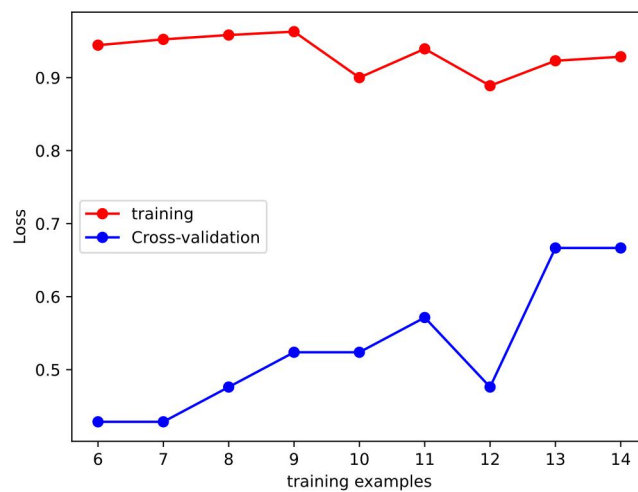
```
clf = svm.SVC(C=0.07, gamma=0.01, kernel='linear',  
              max_iter=40000, decision_function_shape='ovo')
```



可见效果没有改善太多因此使用 `linear` 核函数在  $C=0.1$  处能够取得较好的结果。

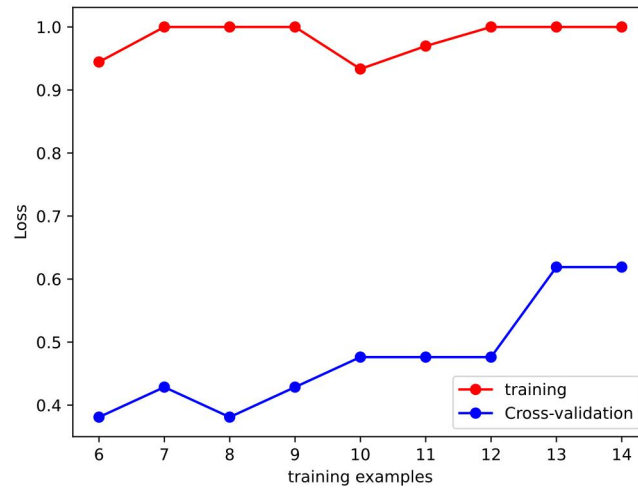
尝试修改核函数 `kernel` 为 `'rbf'` 后:

```
clf = svm.SVC(C=1, gamma=0.05, kernel='rbf',  
              max_iter=40000, decision_function_shape='ovo')
```



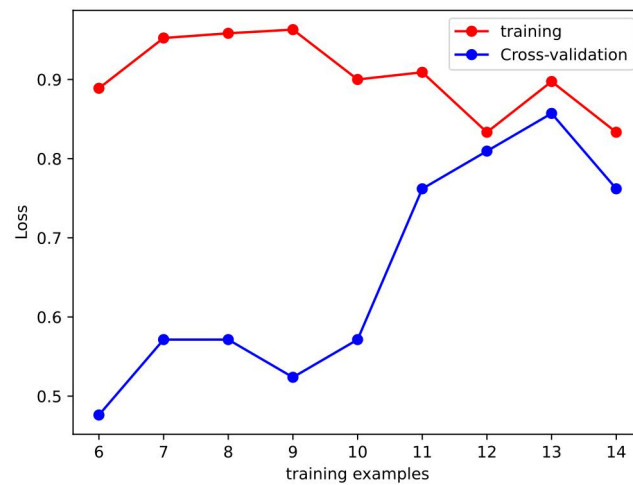
修改参数以后：

```
clf = svm.SVC(C=0.8, gamma=0.2, kernel='rbf',  
              max_iter=40000, decision_function_shape='ovo')
```



采用核函数'rbf'并未改善训练结果，同时还存在更大的过拟合，实际不采用。

尝试使用其他核函数：（sigmoid）



可见采用 sigmoid 核函数能够较好的反应特征值之间的差异，使得训练数据减小欠拟合和过拟合产生的误差。sigmoid 参数设置为：

```
clf = svm.SVC(C=0.8, gamma=0.2, kernel='sigmoid',  
              max_iter=40000, decision_function_shape='ovo')
```

选取上述交叉验证效果较好的参数设置及训练参数导出，用于测试一开始导出的测试集数据，测试准确率如下：

```
Linear(C_0.1) :Test precision: 0.6666666666666666  
Sigmoid(C_0.8G_0.2):Test precision: 0.3333333333333333
```

现将 DB3\_B 数据集中的所有指纹数据提取特征点导入 svm 进行分类，，共计 9 组，每组 7 张训练集图片，1 张测试集图片，采用 Linear(C\_0.1)的 SVM 参数设置，分类结果及准确率如下：

```
Training precision: 0.9841269841269841  
ExpectOut : [1. 2. 3. 4. 5. 6. 7. 8. 9.]  
RealOut : [5. 2. 3. 4. 2. 8. 9. 8. 9.]  
Test precision: 0.5555555555555556
```

在训练数据量较小(每组 7 张),测试数据组过多的情况下,分类器的识别准确率达到 55%。初步完成了指纹识别匹配的工作。

## 六，创新描述

本次实验在特征提取方面采用不同于以往只专注于局部特征点进行特征匹配的方法，抑或是全局计算指纹向量场分布来进行模板匹配。上述两种方式前者关注指纹细节特征往往计算量大，指纹图像特征提取算法复杂，不适合现在越来越小型化的指纹识别机构；后者只关注指纹的全局特征，如向量场等特征量，往往会受限于指纹的相似性，从而识别准确率较低。本实验采用局部特征结合全局特征的方式，通过统计方法将局部特征刻画为一些可描述局部特征位置及法向分布的向量，在降低计算量的同时保证了一定的指纹识别准确性。后续改进时需要提高准确率只需要添加特征点识别的种类即可，少量特征点种类的增加不会增加 svm 分类器的计算量。

在本次实验初期，我首先设想的是将提取到的特征点按照像素位置保存到 200\*200 大小的矩阵中，将 200\*200=40000 维的数据直接投入到 svm 分类器中进行训练，但是由于特征维度显著大于 label 维度，导致训练识别的效果不佳，在导入 3 组数据进行识别时准确率只有 33%（提取到的特征向量保存在 './DB3\_B/features.txt' 和 './DB3\_B/features\_test.txt' 当中）后续考虑到 PCA 主成分分析对上述万维数据进行降维，将 40000 维数据降维至 41 维，但是测试准确率没有明显改善。后续观察数据分布看出可以通过统计特征点数据来描述特征，才有上述特征描述方法。此种方法是图像深度学习的常用方法，我后续拟采用聚类的方法计算得到指纹特征点聚集的区域，然后通过卷积+池化+全连接层嵌套的方式再次尝试分类指纹。

对于参数选取方面，将深度学习中的验证集引入到 svm 训练当中用来测试训练是否产

生过拟合或欠拟合，从而方便调参，显著提高了调参效率以及最终正确率。

## 七，实验总结与心得体会

本次实验我首次尝试从图像处理、数据处理、写入读出数据、编写分类器等方面进行模式识别知识的应用学习。图像处理的算法是采用经典指纹书“Handbook of Fingerprint Recognition”中提到的提取特征算法加以改进得到，数据处理和写入读出数据开始对我很艰难，因为 python 相较于 C++ 缺少数据类型的声明，往往数据经过一次赋值，数据类型便会发生变化，给我的应用带来了不小的麻烦。

DataLoader 的编写不论对于传统分类器还是深度学习而言都是至关重要的，因为任何大规模的数据提取都需要将数据提前保存下来待读取使用，我从编写 DataLoader 的过程中体会到数据存储的格式规范往往会影响读取时应用，因此编写一个对当前项目数据具有一定鲁棒性 DataLoader 格外重要，在花费一定时间编写完写入读出函数以后，不论我怎么修改特征向量的格式，后续的特征提取与分类器读入特征向量时一直没遇到与数据读入读出相关问题。

关于支持向量机 SVM 的使用，一开始我尝试自己编写 SVM 的相关代码，并尝试根据 SVM 源码编写规范的 SVM 类，但是由于时间原因没有完成全部的功能，最终还是调用 sklearn 库中的 SVM 类进行训练和测试。但是实际编写部分 SVM 源码帮助我更好的理解了 SVM 类中的各类参数含义以及在 fit 过程当中充当的作用，帮助我更好的理解 SVM 的运行、优化方法，使我受益匪浅。

对于指纹识别方面，我选题的原因在于我看到一部电影当中 20 年前的警察在匹配嫌疑人的指纹时常常是带着眼睛一点一点的人眼匹配，因为指纹不同于一般的图像，在大范围内表现得特性几乎相同，其类间的相似性常常会导致一般的图像识别算法失效，因此需要传统视觉的特征提取算法提取特征点进行识别，当选题中出现基于支持向量机的指纹识别时，我突然意识到这是一次考验传统图像处理和经典分类算法的一道选题，对我来说显得十分新颖。

## 八，参考文献

- [1]. Maltoni D, Maio D, Jain A K, et al. Handbook of fingerprint recognition[M]. London: springer, 2009.
- [2]. Cortes C, Vapnik V. Support vector machine[J]. Machine learning, 1995, 20(3): 273-297.

- 
- [3]. Galton F. Finger prints[M]. Cosimo Classics, 1892.
- [4]. Champod C, Lennard C J, Margot P, et al. Fingerprints and other ridge skin impressions[M]. CRC press, 2004.
- [5]. Chang J H, Fan K C. Fingerprint ridge allocation in direct gray-scale domain[J]. Pattern Recognition, 2001, 34(10): 1907-1925.
- [6]. Mehtre B M, Murthy N N, Kapoor S, et al. Segmentation of fingerprint images using the directional image[J]. Pattern recognition, 1987, 20(4): 429-435.
- [7]. Mei Y, Sun H, Xia D. A gradient-based combined method for the computation of fingerprints' orientation field[J]. Image and Vision Computing, 2009, 27(8): 1169-1177.