

深圳大学实验报告

课程名称：人工智能导论

实验项目名称：人工智能应用实践

学院：计算机与软件学院

专业：软件工程

指导教师：高 灿

报告人：黄乃建 吴欣悦

学号：2018151070 2018151042 班级：软工3班

实验时间：2019年11月25日至12月15日

实验报告提交时间：2019年12月15日

一、实验目的与要求：

1.1. 实验目的

- 了解人工智能和机器学习的地位、作用以及应用场景
- 掌握人工智能和机器学习常用方法及原理
- 掌握人工智能系统设计思想，培养学生解决实际问题的能力
- 让学生养成用实验数据说话，让结果说理的报告论文写作习惯

1.2. 内容要求

- 题目自拟，但必须与主题密切相关
- 报告结构包括：摘要、简介及相关工作、原理及方法分析、实验部分、实验分析（论证自己观点）、结论、参考文献
- 报告中实验测试数据需丰富，需有明确的实验结论，例如系统考虑准确率问题，那么通过实验分析，应给出方法的准确率如何，并在实验分析部分分析实验结果

1.3. 其它要求

- 严禁全文抄袭文献材料，报告内容必须自己总结和提炼
- 要有相关参考文献的引用和评述
- 应明确地展示论据、实验过程和结果
- 作业 语言：报告 中文（鼓励用英语写，但不做硬性要求），编程语言不限；
- 报告正文要求 1500 字以上，不超过 15 页
- 报告的格式要求遵循参考格式，并包含参考格式中标明的内容

1.4. 参考格式以及说明

报告的格式见下列格式。严格按照文档格式的要求来完成考查报告并按时提交。

报告正文

指纹识别

【摘要】

随着计算机和网络等科学技术的飞速发展，人们对身份识别的准确性、可靠性和便捷性提出了更高的要求。生物特征识别技术也逐步受到人们普遍的重视。在生物识别领域，人们发现指纹具有其独有的生物学特征：普遍性、不变性和唯一性，并已被普遍地运用于金融、网络、医疗和安全等领域。本文主要介绍了指纹识别的原理和基本算法。

【正文】

第一章 指纹识别简介及相关工作

1.1 简介

根据维基百科的介绍，指纹是灵长类手指末端指腹上由凹凸的皮肤所形成的纹路，也可指这些纹路在物体上印下的印痕，纹路的细节特征点有起点、终点、结合点和分叉点。由于每个人的指纹并不相同，同一人的不同手指的指纹也不一样，指纹识别就是通过比较这些细节特征的区别来进行鉴别。

指纹识别的过程分为指纹采集、指纹预处理和指纹特征比对和匹配。

1.2 相关工作

指纹识别的第一步是指纹图像的获取，目前已经有多种指纹图像的获取方式，但因为这次实验我们的重点放在对指纹的预处理以及指纹特征的比对和匹配上，在这里就不在赘述。

获得图像后，就要对图像进行预处理加工。对于图像的预处理，很多研究者都提出了不同的预处理方法，其中 O'Gorman 由指纹局部脊方向得出 $k \times k$ 模板系数，并给出详细的滤波器设计的讨论，还有针对指纹图像降质带来的困难，提出的基于指纹结构特征的某些合理假设，在此基础上设计相应的处理算法增强指纹图像的对比度并提取指纹有效区域，根据方向信息分割图像以及去除图像中气泡噪声的算法等等。

最后一步，指纹特征的比对和匹配。目前已知的指纹匹配方法有基于图像的方法、基于指纹纹路结构的方法、基于图匹配的方法和基于特征点模式匹配的算法。

第二章 指纹识别原理及方法分析

2.1 指纹的预处理

1. 图像灰度的规格化和指纹与背景的分割

例图：



因为各种指纹图像的对比度，图像灰度的差别，我们需要将上述做图像规格化处理，处理成在一定灰度范围内的图像。这部分的内容，我们参考了资料《指纹图像预处理算法研究.pdf》中所讲到的指纹图像的规格化的方法，对图像进行规格化：

- 图像灰度规格化的方法如下：

对图片中的每一个像素作如下的公式转换：

$$N(x, y) = \begin{cases} M_0 + \sqrt{\frac{Var_0 (G(x, y) - M^2)}{Var}}, & G(x, y) > M; \\ M_0 - \sqrt{\frac{Var_0 (G(x, y) - M^2)}{Var}}, & \text{其他}. \end{cases}$$

其中， $N(x, y)$ 是新生成的像素灰度值， $G(x, y)$ 是原像素的灰度值， M_0 和 Var_0 分别是整个图像灰度值的均值与方差的期望值，这里分别取固定值 150 和 2000，而 M 和 Var 分别是整个图像灰度值的实际均值与方差。

由于在指纹图像中，往往背景是作为不需要的部分，因此，我们需要将指纹图像与背景分割开来。

- 指纹图案与背景分割的方法如下：

将整个图像分割成若干个 20×20 的小块区域，对每块区域中的所有像素的灰度值做方差运算，如果方差小于一定阈值，就代表这块区域的灰度值区别不大，是背景，于是就将这块区域的所有像素设置成白色。

在图像的规格化和指纹与背景的分割处理后，我们就得到了下面这个图像：



可以看到，图像的灰度值已经调整成我们想要的范围了，指纹纹路的清晰度也有所增强，同时，指纹图案也与背景分隔开了来。

2. 图像的二值化

- 基于方向场的二值化

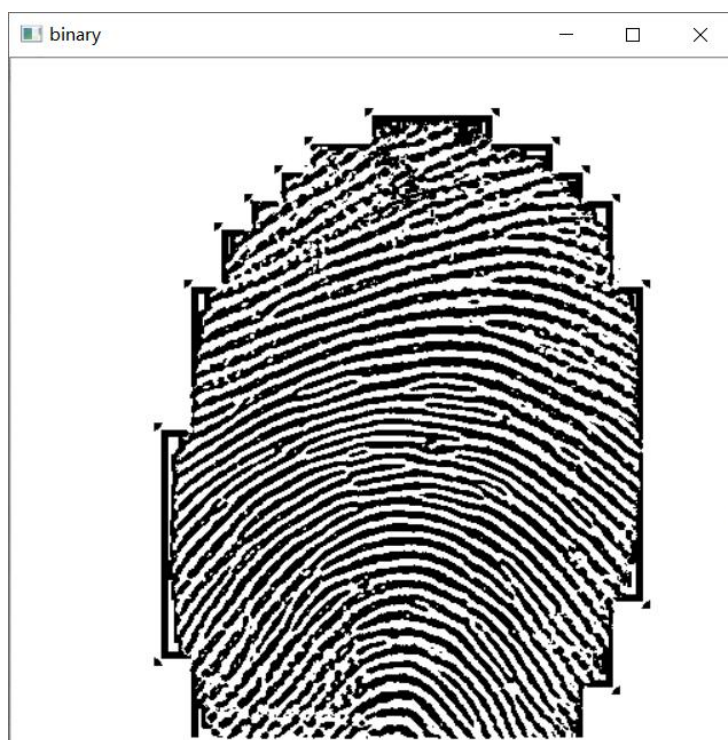
指纹图像一般都有着清晰的方向场，如果我们能够利用这种方向性，对图像进行方向滤波处理，就能得到较为清晰纹路图像。

- 基于方向场的二值化方法一：

我们首先参考了博客 https://blog.csdn.net/weixin_42121843/article/details/100124486 中

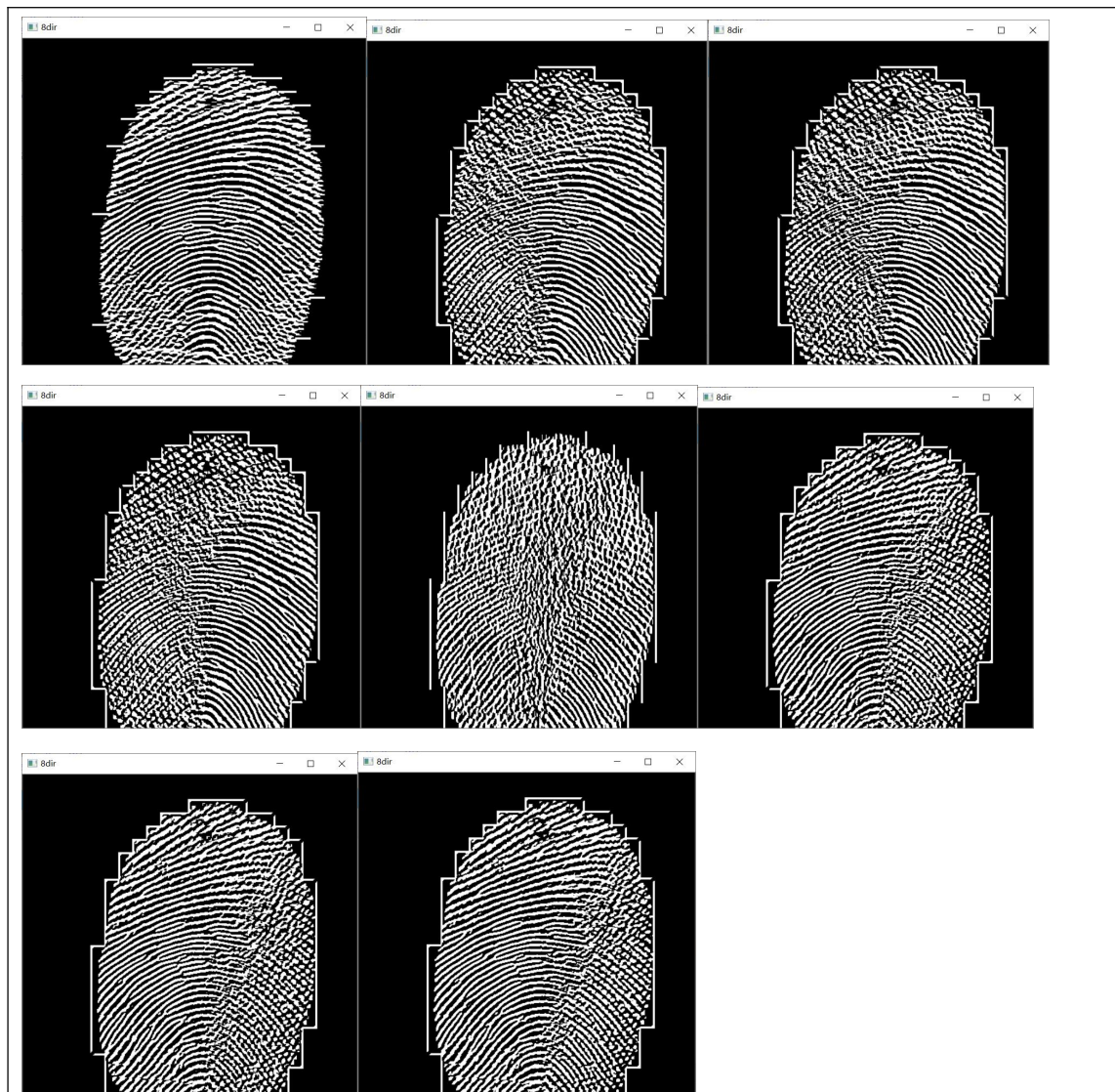
的基于方向场二值化的方法。首先，对整个图像进行 3*3 的均值滤波，然后对每个像素的八邻接块的八个方向进行灰度值的累加计算，然后去掉最大值 $summax$ 与最小值 $summin$ ，如果满足 $summax + summin + 4 * (\text{该像素的灰度值}) > (3 * sum / 8)$ ，其中 sum 为所得八个方向灰度累加之和，则该脊线方向为 $summin$ 所对应的方向，否则为 $summax$ 。

用这个方法所得指纹的二值化图像如下图所示：



- 基于方向场的二值化方法二：

除上述方法之外，我们还参考了资料《基于 8 方向脊向滤波器的指纹预处理方案.pdf》，企图对图像二值化处理作进一步优化。资料中给出了 8 个方向的卷积核作为指纹 8 个方向的滤波器，我们直接拿这 8 个方向的卷积核对图像进行了卷积运算，其结果如下图所示：

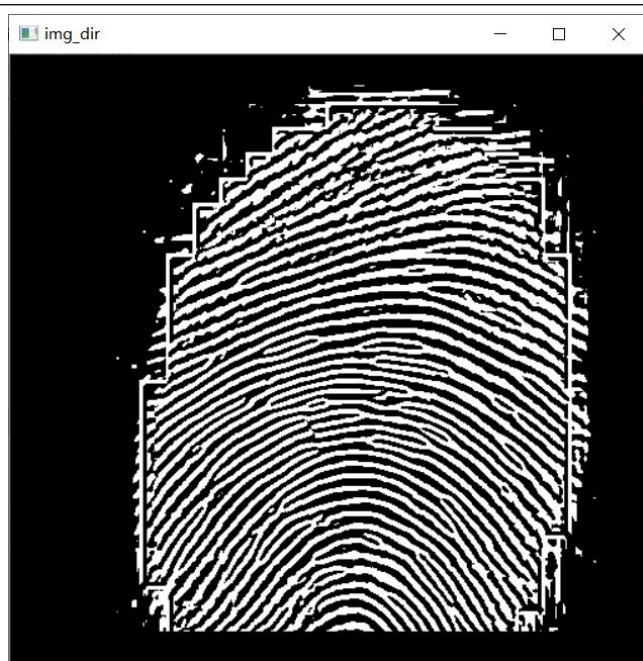


但是，如何将这 8 个方向滤波出来的图像合成，又成了个问题。

于是我们参考了资料《八方向滤波器的设计及其在图像增强中的应用.pdf》中，对多个方向滤波出来的图像的合成方法。

资料中提到，可以通过对 8 个图像中，取每个相同位置像素点的周围一定范围区域，得到该像素的邻接块，对每块中的像素进行方差运算，所得方差最大的那个邻接块所含的信息最丰富，于是在合成图像中，取其对应的那个像素点。但是，经过实践，发现这种方法虽合成效果不错，但其运行的时间较长。

以下是对这 8 个方向图的合成效果：



3、去除指纹中的空洞和毛刺

原指纹在进行二值化后，指纹上的空洞和毛刺等一些为伪特征点在抽取特征点的阶段都有可能被错误地抽取为特征点，从而影响特征的匹配。因此，去除指纹上的空洞和毛刺就变得十分必要。空洞的判断条件是：如果该点的背景在二值化后是白色，即图像矩阵上的值为 1，并且八邻域的值和为 0，即八零域的点均为黑色，这样的点就是空洞。毛刺的判断条件是：如果该像素点的值为 0（黑色），并且该像素点的上下左右的像素点的值加起来大于 3，即上下左右的像素点中有多个是白色，就是毛刺。去除空洞只要让该像素点变为黑色即可，而去除毛刺只要将该像素点变成白色即可。

效果如下图所示：



4、图像的细化

在去除了空洞和毛刺之后，有可能还存在很多小的由图像形成的斑点。同时，可能因为前面去除空洞和毛刺的过程导致很多本来连接的部分被误分成许多小块的对象。在此，采用形态学中的开闭运算进行处理。开运算的过程：先定义一个 3×3 的矩形结构元素，然后对图像的每个像素点进行腐蚀操作，即让每个结构元素和与其覆盖的二值图像的 3×3 子块进行与运算，如果结果的矩阵里的元素都为 1，则该像素点的值就为 1，否则置为 0。然后再对每个像素点进行膨胀操作，即用结构元素和与其覆盖的二值图像的 3×3 子块进行与运算，如果结构都为 0，则该像素点的值就为 0，否则置为 1。而闭运算的过程和开运算的相反，即先进行膨胀操作，再进行腐蚀操作。通过开运算，可以移除掉由噪声引起的斑点，通过闭运算，可以连接起分开的小块。

5、中心点提取

这一步求取中心点是为了消除指纹边缘的伪特征点。中心点是指纹的脊曲线曲率最大的点。先通过梯度法求取指纹图像的块方向图，然后通过 3×3 的结构元素对块方向图进行自适应降噪滤波，然后将块方向图表示为 x 方向和 y 方向的分量，

再对这两个分量进行自适应降噪滤波即可，最后在平滑后的方向图找到曲率最大的点即可。

6.特征点的选取

根据 Crossing Number (CN) 理论，取某个实心像素 P (黑色像素) 为中心，建立 3*3 的像素区块，绕边缘环绕一周分别为 P1,P2,...,P8，如下图所示：

P1	P2	P3
P8	P	P4
P7	P6	P5

根据公式：

$$CN = 0.5 \sum_{i=1}^8 |P_i - P_{i+1}|, \quad P_9 = P_1$$

算出 CN 值，其中实心像素的值为 1，空心像素的值为 0。

根据 CN 理论，当 CN 值等于 1 时，该像素 P 为端点，当 CN 值等于 3 时，该像素 P 为分叉点。按照这个理论，以指纹的中心点为中心，周围一定区域内计算出所有的特征点。

参考博客：<https://blog.csdn.net/MrCharles/article/details/79300671>

7. 特征点的筛选

因为图像本身以及处理过程不是百分之百精确的，所以，我们找出的特征点中必定有许多“伪特征点”，因此，我们要筛选这些特征点。

- 根据所得特征点去掉细化图像的毛刺：

所得特征点中，有许多端点，这些端点有可能是细化图中毛刺的端点。对每一个端点，

顺着纹路向前走若干个像素（比如 5 个），如果遇到了分叉点，那么就说明这个端点就是毛刺的端点，将其去除，同时也在顺着纹路走的同时，去除走过的黑色像素点，从而

对图像起到光滑的作用。

- 去除相距过近的特征点

相距太近的特征点也可能是图像中某个短线的所造成的为特征点，应该予以去除。

2.2 指纹图像匹配

根据所得特征点匹配：

参考了资料《基于全等三角形的指纹匹配算法的研究与实现.pdf》。资料中提到，可以利用全等三角形法寻找两图的基本点，从基本点出发，用点匹配法对两图像的特征点进行匹配。

- 具体方法：

取得两个图像的特征点序列，若其中一个序列中有特征点 P 与其最临近的两个与 P 类型不同的特征点构成一个三角形，然后，在另一序列中也能找到一个特征点 Q ，也与其最邻近的这种特征点构成一个相似的三角形，那么这两个特征点对应，可以作为两个图的基本点。

而点匹配法稍作了简化：以这两个基本点 P 、 Q 出发，在其周围找出若最邻近的若干个点，计算其中的端点个数，若端点个数两边相差小于一定的阈值，那么就判定这两个指纹图像匹配。

第三章 实验论证

3.1 验证指纹识别程序

1. 对两个相匹配的指纹进行验证：

主函数：

```

if __name__ == '__main__':
    txy1 = fingerTxy('image/022_5_7.bmp')
    txy2 = fingerTxy('image/022_5_8.bmp')
    result = pointMatch(txy1, txy2) # 匹配检测
    print(result)

```

这两个指纹图像如下图所示：



运行程序：

```

In[2]: runfile('E:/CodingWo
480 504
480 504
匹配率：90.0%
匹配成功
True

```

2. 对不同的指纹图像进行验证：

这次这两个指纹图像如下：



程序运行的结果如下：

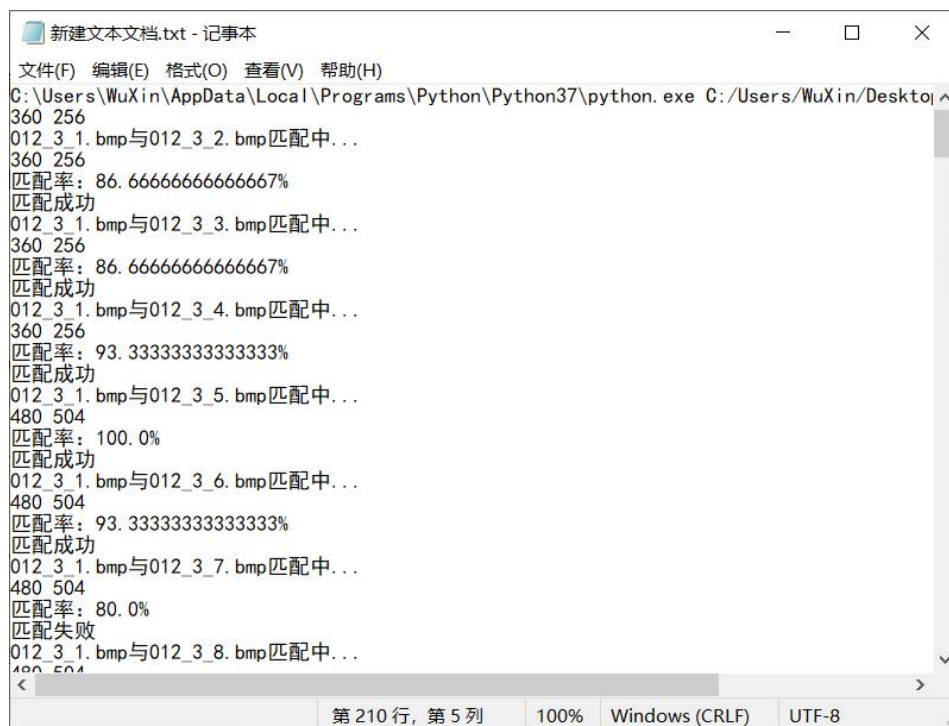
```

Python 3.7.3 (default, Mar
In[2]: runfile('E:/CodingWo
480 504
480 504
匹配失败
False

```

3.2 验证该指纹识别程序的正确率

我们取了若干对相同的指纹进行进一步比对，其部分结果如下：



```
新建文本文档.txt - 记事本
文件(F) 编辑(E) 格式(O) 查看(V) 帮助(H)
C:\Users\WuXin\AppData\Local\Programs\Python\Python37\python.exe C:/Users/WuXin/Desktop/...
360 256
012_3_1.bmp与012_3_2.bmp匹配中...
360 256
匹配率: 86.66666666666667%
匹配成功
012_3_1.bmp与012_3_3.bmp匹配中...
360 256
匹配率: 86.66666666666667%
匹配成功
012_3_1.bmp与012_3_4.bmp匹配中...
360 256
匹配率: 93.33333333333333%
匹配成功
012_3_1.bmp与012_3_5.bmp匹配中...
480 504
匹配率: 100.0%
匹配成功
012_3_1.bmp与012_3_6.bmp匹配中...
480 504
匹配率: 93.33333333333333%
匹配成功
012_3_1.bmp与012_3_7.bmp匹配中...
480 504
匹配率: 80.0%
匹配失败
012_3_1.bmp与012_3_8.bmp匹配中...
480 504
匹配率: 80.0%
匹配失败
```

具体内容将见附录。

我们发现，对于同一个指纹的匹配，其效果确实还算不错，但是当我们换成不同的指纹去做验证时，结果却不尽人意了。由于时间问题，我们并没有做进一步更详细的验证，但这个匹配算法仍待改进和优化。

【结论和体会】（谈谈自己的看法和理解）

实验结论：本程序通过实验的对指纹的规格化处理，分割，然后进行八方向滤波和二值化处理，之后对图像细化，特征点采集等预处理操作，得到一串基于特征点的指纹模型，之后根据所写的匹配算法进行匹配，所得到的结果是能大概区分出相同的指纹与不同的指纹，但精度上还有待加强。本程序所含的代码中图片的读取与展示，细化过程的轮廓提取，对卷积核的卷积操作，中值与均值滤波等调用了外部库（opencv、numpy、skimage、scipy等），其余为手写。

黄乃建：这个实验我查阅了大量的资料，也参考了许多他人的代码，同时与队友合作，最后终于把这个实验做出来了，还是很有成就感的。但唯一美中不足的是最后的结果，这种最后的匹配结果并不是那么完美，还是有些可惜。

吴欣悦：通过这个实验，我初步接触到了图像处理方面的知识，了解到了均值滤波，中值滤波，自适应滤波的原理，以及形态学方面的知识。有些遗憾的是，自己的能力还是有些欠缺，有时队友出现问题时，自己帮不上什么忙。

权重分配：黄乃建 60% 吴欣悦 40%

【参考文献】（逐个举出自己用到的参考资料）

1. https://blog.csdn.net/weixin_42121843/article/details/100124486
2. <https://blog.csdn.net/MrCharles/article/details/79292496>
3. <https://blog.csdn.net/baimafujinji/article/details/73302911>
4. https://blog.csdn.net/wsp_1138886114/article/details/82917661
5. <https://blog.csdn.net/MrCharles/article/details/79300671>



6. 基于全等三角形的
指纹匹配算法的研究



7. 八方向滤波器的设计及其在图像增强中的应用.pdf



8. 基于8方向脊向滤波器的指纹预处理方案.pdf



9. 指纹图像预处理算法研究(1).pdf

【附录】

1. 字数统计 4596 个字
2. （对这门课和课程课堂教学的看法，建议，意见均可以提出）

指导教师批阅意见：

成绩评定：

指导教师签字：高灿
2019 年 12 月 25 日

备注：