

# COMP9517: 计算机视觉

## 2023年第二学期

### 作业规范 可达到的最高分值：

**10**

这项作业的分值占课程总分的10%。

作业文件应在网上提交。

提交说明将在接近截止日期时公布。

**提交的截止日期是第4周，2023年6月19日星期一，18:00:00。**

**可交付的成果：**你必须提交以下项目：

1. 一份报告，解释你在每项任务中所采取的方法。报告可以用Word、LaTeX或类似的文本处理器编写，但必须以PDF格式提交，并在第一页包括你的姓名和zID。它还必须显示输入图像的样本以及获得的中间和最终输出图像。不提供模板，但报告必须不超过5页（A4），而且必须是单栏，使用11点Times New Roman或类似字体，并有2.5厘米的边距。
2. 源代码和Jupyter笔记本（.ipynb）形式的结果，包括所有输出（见下面的编码要求）。

**提交：**以一个压缩文件的形式提交所有可交付成果。提交说明将在接近截止日期时发布。为了避免导师/评委的混淆，他们可能会处理许多提交的文件，请在所有的文件名中加上你的zID（例如，**zID\_Report.pdf**代表报告，**zID\_Notebook.ipynb**代表笔记本）。

**预备工作：**如课程大纲所述，我们假定您熟悉Python编程或愿意独立学习。您不需要成为专家，因为无论如何您会在课程中进一步发展您的技能，但您至少应该知道基础知识。如果您还不知道Python，我们假设您至少熟悉一种其他编程语言，如C语言，在

这种情况下，学习Python应该还是比较容易的。

要学习或提高你的Python技能，请看本文末尾列出的几个免费在线资源。在来参加第一次导师咨询会议（见下一节）之前，查看（其中一些）是最有帮助的。特别是如果你已经知道C语言或类似的语言，就没有必要详细查看所有的链接资源。只需快速学习语言的语法和主要特征。其余的会随着你的学习而跟进。

为了实现和测试计算机视觉算法，我们在本课程中使用OpenCV。OpenCV是一个主要用于实时计算机视觉的编程函数库。该库是跨平台的，在Apache许可证2下被授权为免费和开源的软件。它还支持机器/深度学习的模型执行。它最初是用C语言编写的，新的算法是用C++开发的，它有Python和Java等语言的封装器。如上所述，在本课程中，我们将着重于用Python编程。请看下面的链接，了解OpenCV的教程和文档。

**软件：**你需要使用OpenCV 3+与Python 3+，并以Jupyter笔记本的形式提交你的代码（见上面的交付物和下面的编码要求）。在2023年6月9日星期五下午2-3点的第一次导师咨询会上，你的导师将对将要使用的软件进行演示，你可以提出任何问题。

**目标：**这项作业是为了熟悉基本的图像处理方法。它还向你介绍了使用OpenCV的常见图像处理和分析任务。

**学习成果：**完成这项任务后，你将学会如何：

1. 打开并读取图像文件。
2. 显示和写入图像文件。
3. 对图像进行基本的图像处理操作。
4. 实现并应用各种自动阈值技术于图像。

**描述：**在许多计算机视觉应用中，要对图像中感兴趣的物体（或区域）进行定量分析，关键的第一步是确定哪些像素属于该物体（相关像素），哪些属于背景（不相关像素）。这项任务被称为二元图像分割。

进行二元图像分割的最简单的技术是强度（灰度）阈值处理。虽然每幅图像的最佳阈值可以由用户手动选择，但在需要完全自动化的应用中这是不可取的。幸运的是，存在几种自动阈值处理技术，正如第一周讲座中所讨论的那样。

这项任务的目标是编写图像处理算法，能够打开数字图像并使用各种自动阈值技术进行二进制分割。

**任务：**这项任务由以下四项任务组成。

### **任务1 (2.5分)： 大津阈值法**

第一种技术被称为大津阈值法（以发明者的名字命名）。它将最佳阈值定义为使类内方差最小的阈值，或者说，使两个类中的像素值的类间方差最大的阈值（对象与目标）。

背景)。更多技术细节见讲座的幻灯片。

编写一个算法，可以接受任何大小的输入图像，并对其进行大津阈值处理，产生一个二进制（分割）图像。请注意，OpenCV或其他Python库可能已经有这方面的现有函数，但你不会从使用它们中学到任何东西。这里的任务是要从头开始写你自己的大津阈值算法。特别是，在像素上写出你自己的循环，以计算像素等级、标准偏差和变异，并寻找最佳阈值。

### **任务2 (2.5分)：等值数据阈值化**

第二种技术被称为等数据阈值法。它通过从一个随机阈值开始，计算产生的两类像素的平均强度（灰度），将两个平均值作为新的阈值，并重复这个过程直到收敛，从而找到最佳阈值。更多技术细节见讲座幻灯片。

编写一个算法，可以接受任何大小的输入图像，并对其进行等值数据阈值处理。在此，请不要使用现有的库函数，这些函数可能只需调用一个函数就能完成这一任务，而是要从头开始编写自己的算法。

### **任务3 (2.5分)：三角形阈值处理**

第三种技术被称为三角阈值法。它从图像的强度（灰度级）直方图中计算出最佳阈值。更具体地说，它从直方图的峰值到直方图的极值（最高灰度点）计算一条直线，并找到直方图与该直线偏差最大的灰度点。更多技术细节见讲座幻灯片。

对于某些图像，可能会出现直方图的峰值与最高灰度等级相对应的情况。在这种情况下，直方图的形状可能与讲座幻灯片中的例子相比发生了翻转。也就是说，大多数像素的灰度等级较高（而不是较低），而直方图的尾部是朝向较低（而不是较高）的灰度等级。在这种情况下，算法应该是"倒置"的，也就是说，线应该从直方图的峰值画到最低灰度点（而不是最高灰度点）。

编写一个可以接受任何尺寸的输入图像并对其进行三角形阈值处理的算法。这里同样不要使用现有的库函数，这些函数可能只需调用一次就能完成这一任务，而要从头开始写自己的算法。你可以使用一个现有的函数来计算图像的直方图。

#### **任务4（2.5分）：比较阈值处理技术**

将你的Otsu、isodata和三角形阈值算法应用于本作业提供的五张图像，并在报告中展示结果。以表格形式展示结果，以

允许对所有图像进行简单的视觉比较。例如，每个表行显示输入图像、其直方图和三个阈值处理结果。

在你的报告中，讨论结果的差异，并提供解释（根据直方图或其他方式），为什么对某些图像来说，一种阈值技术可能比其他技术更好，而对其他图像来说，可能是相反的情况。提出一些一般准则，说明哪种阈值技术最适合哪类图像。

### **编码要求**

对于每项任务，请自己实现该算法，不要使用现有的库函数（来自OpenCV或任何其他软件包）来直接执行该任务。使用这些函数而不是你自己的实现将导致被扣分。

在未来的实验和小组项目中，你可以使用现有的库函数，但在这个任务中（就像在第一个实验中），目标是学习基本的图像处理操作如何在像素级工作，并获得自己实现它们的经验。当然，你可以使用现有的函数来验证你自己实现的结果。

对于这项作业（与实验室相同），确保在你的Jupyter笔记本中，输入的图像可以从作为参数指定的位置读取，所有的输出图像和其他要求的结果都显示在笔记本环境中。你的笔记本中的所有单元都应该已经执行，这样导师/标记者就不必再执行笔记本来查看结果。

### **免费的在线Python资源：**

W3Schools Python Tutorial

<https://www.w3schools.com/python/>

LearnPython.Org

<https://www.learnpython.org/>

Python初学者<https://www.python.org/about/gettingstarted/>

哈佛的Python编程入门 <https://cs50.harvard.edu/python/>

谷歌的Python类

<https://developers.google.com/edu/python/>

FreeCodeCamp的Python在4小时内完成了YouTube上的课程(40M浏览量)

<https://www.youtube.com/watch?v=rfscVS0vtbw>

**免费的OpenCV资源：**



关于OpenCV <https://opencv.org/about/>

OpenCV教程[https://docs.opencv.org/4.x/d9/df8/tutorial\\_root.html](https://docs.opencv.org/4.x/d9/df8/tutorial_root.html)

OpenCV Wiki  
<https://github.com/opencv/opencv/wiki>

OpenCV 文档 <https://docs.opencv.org/>

**版权所有：**新南威尔士大学CSE COMP9517团队。复制、出版、张贴、分发或翻译本作业是对版权的侵犯，并将被提交给新南威尔士大学学生行为与诚信处处理。

**发布：**2023年6月5日