计算机视觉导论（2024年春季）作业1

发布日期：2024年3月15日，截止日期：2024年3月30日晚上11:59

1 概述

该任务包括4个任务：实现卷积操作，Canny边缘检测器，Harris角点检测器，使用RANSAC进行平面拟合，总共100分，将计入您本课程的最终得分中。该任务完全涵盖了第2和第3讲的课程内容。

本任务的目标是让您熟悉在张量风格中编码和实现经典计算机视觉的基本模块和算法。这里的张量风格指的是避免使用任何for循环，而是通过矩阵或张量运算来实现所有操作，例如矩阵乘法。这与人们在PyTorch或Tensorflow中进行编码以实现并行计算的方式完全相同。我们为所有任务提供了起始代码，您需要使用Python和Numpy来实现关键函数。

2 通知

如果您在Numpy中发现了一些强大的功能，可以轻松解决一个任务，请在使用之前向我们询问是否允许。

2. 如果任务没有指明允许使用"for循环"，在你的代码中使用"for循环/while"将会受到惩罚（每使用一次扣1分）。请注意，np.vectorized也不允许使用，因为它不是并行计算，速度与"for循环"相同。附录5中包含了一些有用的Numpy函数供参考。

3. 请在pack.py中更新您的个人信息，运行pack.py脚本压缩您的代码和结果，然后将zip文件提交到course.pku.edu.cn。

请随意在讨论区发布任何问题，并鼓励大家报告可能改进此任务的潜在方法，最高可获得5分的奖励。

3 任务

1. 卷积操作（15分）：

   卷积是深度学习和图像处理中最常用的操作之一。在本节中，您需要以两种不同的方式实现二维卷积。您将能够

你需要以两种不同的方式实现2D卷积，以检查你的卷积运算符在高斯滤波器和Sobel滤波器中的表现。请完成HM1 convolve.py。

实施填充功能

您需要实现一个填充函数，该函数接受一个二维图像、填充大小（图像周围的圈数）和填充模式（零填充或复制填充）作为输入。其中，复制填充是指使用输入边界的复制来填充输入张量（详见附录2）。请注意，您不允许使用numpy.pad函数。

-

通过Toeplitz矩阵实现2D卷积

卷积可以通过单个矩阵乘法实现，这被称为Toeplitz矩阵（见附录1）。对于一个6×6的输入2D数组和一个3×3的滤波器，您需要构建双重块Toeplitz矩阵，然后使用您的Toeplitz矩阵对2D数组进行卷积。在这里，我们假设卷积使用零填充，以保持输出大小不变。

提示：如果您的输入数组没有填充，则需要构建一个36×36的Toeplitz矩阵；如果您先对输入进行填充，则需要构建一个36×64的Toeplitz矩阵。您可以避免使用for循环，因为Numpy允许您同时索引许多矩阵元素并将值一起赋给它们。

滑动窗口卷积

Toeplitz矩阵实现的卷积是完全并行的，但它要求你构建一个高度稀疏的Toeplitz矩阵（几乎是N×N）用于大小为N×N的图像。实现卷积的另一种方法是通过滑动窗口，它允许你在每个窗口内简单地计算图像值和内核之间的点积。在这个问题中，你被要求实现一个卷积运算符，它可以接受一个任意形状的2D图像作为输入，并使用k×k的内核进行卷积。在实现这个卷积运算符时，可以假设没有填充。

提示：您需要从输入图像构建一个（N − k + 1）× k矩阵来支持并行计算。为了避免使用任何循环操作，您可以使用np.meshgrid来生成索引。

高斯滤波器

通过滑动窗口卷积，我们可以简单地构建一个3×3的高斯滤波器。在代码中，我们将在a)中使用您实现的复制模式填充函数，并在c)中使用您实现的滑动窗口卷积来构建一个3×3的高斯滤波器。

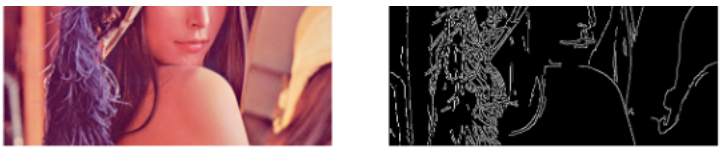
e)[0分] Sobel滤波器

在第二讲中，我们讨论了使用有限差分来近似图像梯度。更常用的方法实际上是使用Sobel滤波器（见附录3）。在代码中，我们将使用您实现的反射模式填充函数（a部分）和滑动窗口卷积函数（c部分）来构建一个Sobel滤波器。

2. Canny边缘检测器（30分）

在这个任务中，你将构建自己的Canny边缘检测器。具体来说，需要实现三个关键函数，图像梯度计算、非极大值抑制和边缘连接（使用滞后阈值）。请完成HM1 Canny.py，你将看到一个类似于图中的结果。

1.



Input (Lenna.png)　　Output

图1：带有Lenna的Canny边缘检测器

计算图像梯度。

我们将使用Sobel滤波器来近似图像的梯度– x方向导数和y方向导数。然后，我们可以得到梯度的幅度M和方向D。

$$M = \sqrt{\left(\frac{\partial I}{\partial x}\right)^2 + \left(\frac{\partial I}{\partial y}\right)^2}, \qquad D = \text{反正切}\left(\frac{\partial I}{\partial y} \cdot \frac{\partial x}{\partial I}\right). \tag{1}$$

幅度和方向提供了边缘的粗略估计，包括像素可能位于边缘的可能性（幅度）和边缘的法线方向（方向）。您将完成计算梯度幅度方向的函数。

非最大抑制（NMS）

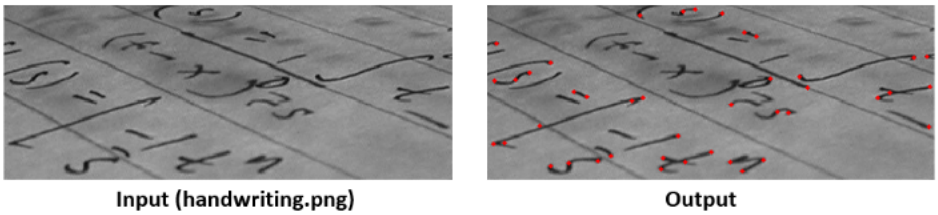
获得梯度的大小和方向后，应检查每个像素并删除可能不构成边缘的不需要的像素。实现这个目标的常见技术是非极大值抑制（NMS）。NMS沿着梯度方向找到局部最大值，这可能导致“细边缘”。请完成幻灯片中介绍的非极大值抑制函数。为了降低难度，您可以实现Lec. 3 Page 13中的简化版本。

边缘链接与滞后阈值

非极大值抑制产生的边缘通常过于细小，可能无法连接。为了连接潜在的不连续边缘，我们应该进行边缘连接。请完成滞后阈值函数（此问题允许使用"for循环"）。为了降低难度，您不必使用梯度的方向，只需在幅度大于下阈值的情况下简单地连接相邻像素。

3. 哈里斯角点检测器（30分）：

计算机视觉中的一个关键主题是找到给定图像的显著且准确的点特征。早期尝试找到这些特征的方法之一被称为Harris角点检测器。在这个问题中，您被要求实现角响应函数来研究强度的属性。



Input (handwriting.png)　　Output

图2：带有手写字的Harris角点检测器.png

在这个问题中，给定一个窗口的变化。请完成HM1 HarrisCorner.py，你将会看到一个类似于图2的结果。

提示：你只需要实现幻灯片中介绍的矩形窗口版本。

4. 平面拟合使用RANSAC算法（25个点）

在这个任务中，你需要使用RANSAC算法实现一个平面拟合算法，从一个带有噪声的三维点云中找到一个平面。具体来说，你需要执行以下步骤。

1）假设生成：对于一个假设，你需要随机选择一组种子点，然后估计平面参数。请计算出理论上能够保证至少一个假设不包含任何异常值的最小样本次数，使得概率大于99.9%。

你被期望同时生成所有假设。

验证：找到并计算所有假设的内点数量。

最后的改进：选择具有最大内点数的假设作为最佳假设，并使用最小二乘法估计其内点的最终平面参数，该方法最小化点与平面之间的垂直距离的平方和。

请完成HM1 RANSAC.py，您将会看到一个类似于图3的结果。

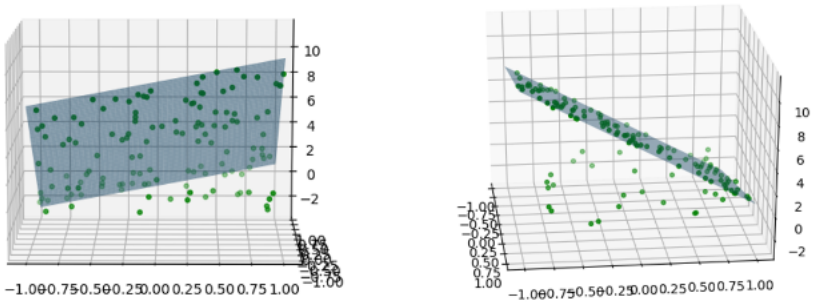


图3：RANSAC拟合的平面（蓝色）；点（绿色）。

附录

1. 1D卷积与Toeplitz矩阵。
https://zh.wikipedia.org/wiki/Toeplitz矩阵#离散卷积

2. 一个直观的视频演示，展示了使用Toeplitz矩阵进行2D卷积。

https://www.youtube.com/watch?v=4uvDVEQkzXQ

在PyTorch中的复制填充。
https://pytorch.org/docs/stable/generated/torch.nn.ReplicationPad2d.html

OpenCV Sobel导数教程。
https://docs.opencv.org/3.4/d2/d2c/tutorial sobel derivatives.html

5. 从Justin Johnson的线性层的反向传播。

http://cs231n.stanford.edu/handouts/linear-backprop.pdf

我们推荐一些方便的Numpy函数，这些函数可能有助于您的张量式编码。

- 网格，https://numpy.org/doc/stable/reference/generated/numpy.meshgrid.html
- 连接, https://numpy.org/doc/stable/reference/generated/numpy.concatenate.html
- 在哪里，https://numpy.org/doc/stable/reference/generated/numpy.where.html
- argmax，https://numpy.org/doc/stable/reference/generated/numpy.argmax.html
- linalg.svd，https://numpy.org/doc/stable/reference/generated/numpy.linalg.svd.html
- arctan2，https://numpy.org/doc/stable/reference/generated/numpy.arctan2.html