**问题1 黑白图像灰度扫描**

实现思路：

采用cv2.imread函数以灰度图像方式读取图像路径。当图像的行数或列数为奇数时，直接取中心行和中心列；当行数或列数为偶数时，取中间2行或2列的灰度平均值作为中心灰度值。

extract(path)函数读取图像路径path，并通过调用scanLine4e函数提取灰度图像中心行和中心列的灰度值，最后返回灰度值矢量并可视化输出。

图表, 直方图

描述已自动生成

图1：图像cameraman.tif中心行和中心列灰度值

图形用户界面, 图表, 直方图

描述已自动生成

图2：图像einstein.tif中心行和中心列灰度值

**问题 2 彩色图像转换为黑白图像**

墙上有涂鸦

中度可信度描述已自动生成动物的眼睛

描述已自动生成动物的眼睛

描述已自动生成

图片包含 人, 照片, 女人, 站

描述已自动生成女人戴着帽子

描述已自动生成女人戴着帽子

描述已自动生成

图3：average、NTSC与库函数转换结果对比

由图3可知，对于两种将彩色RGB图像转换成单色灰度图像的方法，average方法对R、G、B三个通道取均值，转换后的灰度图像并不符合人类视觉习惯，与opencv库函数转换结果相差较大；NTSC方法对RGB三通道进行加权，转换后的灰度图像和库函数结果相近，但并不完全相同，说明opencv自身采用的并不完全是NTSC方法。

**问题 3 图像二维卷积函数**

首先注意到，卷积操作需要对卷积核w进行上下左右翻转。

根据卷积后图像尺寸的计算公式



在为奇数且的前提下，若卷积后尺寸不变，则填充大小满足



在填充时，将待填充区域分为8个部分，即上、下、左、右、左上、左下、右上和右下，分别取和最近的像素值相等。

在计算卷积时，应先设定数据格式为float，全部计算完成后，再转为np.uint8数据格式输出结果。

**问题 4 归一化二维高斯滤波核函数**

首先需要确定高斯滤波器的大小



对以下几种情况展开讨论：

①m没有给定：取高斯滤波核大小为；

②m给定但：程序给出Warning，提示高斯滤波核太小；

③m给定且：最终高斯滤波核大小为m。

高斯核函数的表达形式为



其中为归一化因子。

**问题 5 灰度图像的高斯滤波**

调用gaussKernel和twodConv函数，采用0填充方式对图像进行高斯滤波，并和opencv官方高斯滤波函数cv2.GaussianBlur进行对比，结果如图4所示。

穿着西装笔挺的男子与配字黑白照

低可信度描述已自动生成穿着西装笔挺的男子与配字黑白照

低可信度描述已自动生成形状

低可信度描述已自动生成

图片包含 人, 站, 男人, 看着

描述已自动生成图片包含 人, 站, 男人, 看着

描述已自动生成![形状

低可信度描述已自动生成]()

猴子的脸

描述已自动生成猴子的脸

描述已自动生成![形状

低可信度描述已自动生成]()

女人戴着帽子

描述已自动生成女人戴着帽子

描述已自动生成![形状

低可信度描述已自动生成]()

图4：gaussKernel+twodConv滤波、cv2.GaussianBlur滤波和差值图像

通过计算可知在自定义gaussKernel+twodConv滤波和cv2.GaussianBlur滤波结果中，灰度值相差最大为1，因此可以认为两者实现了相同的功能。为可视化差异点，将相差为1的灰度映射为255的灰度值，差异可视化结果如下所示。

图片包含 雨, 游戏机, 飞行

描述已自动生成 背景图案

描述已自动生成

(a)cameraman.tif (b) einstein.tif

图片包含 雨, 游戏机, 烟花

描述已自动生成 背景图案

描述已自动生成

(c)mandril\_color.tif (d) lena512color.tiff

图5：可视化差量图像

下面比较边界填充方式为0填充和replicate填充的区别，选择cameraman.tif作为比较对象。

 水中的人的黑白照片

中度可信度描述已自动生成

(a) 0填充的滤波图像 (b)replicate填充的滤波图像

图6：0填充和replicate填充滤波图像

对比图6(a)和图6(b) 可知，0填充的图像边缘有1圈黑色边框，而replicate填充不存在黑色边框。这是由于边缘填充的灰度值计算更大，0填充在计算得出的边缘灰度值较小，由此导致了边框的区别。