上机实验一

一、实验内容：

本次实验包括三个部分：对图像进行三种操作的 MSE 和 PSNR 计算、不同粗细黑白条纹和红绿条纹间隔的视觉分辨力比较以及灰度图像的二值化。

题目一 选择一副图像，对其进行任意三种操作(比如剪切、旋转、加噪等)，计算操作前后图像的 MSE 及 PSNR，并对结果进行小结。

题目二 编程实现不同粗细黑白条纹和红绿条纹间隔出现两种情景，比较你对黑白和彩色的视觉分辨力。

题目三 实现灰度图像的二值化，人工确定两种阈值，并与全局阈值法确定的阈值比较二值化效果。

二、实验原理：

1. 对图像的三种操作，并计算 MSE 和 PSNR：
   图像的三种操作包括剪切、旋转和添加 `FIND_EDGES` 滤镜。其中剪切和旋转是对图像中像素的位置进行改变，而加滤镜则是对图像像素的值进行改变。对于剪切和旋转操作，在对图像进行这些操作的时候，像素的值不会有任何改变，因此 MSE 和 PSNR 的计算仅与像素位置相关。对于滤镜操作，我可以使用已知的 PIL 库中已有的预设滤镜数据添加到原图像上面去。添加滤镜会改变图像的像素值，进而影响 MSE 和 PSNR 的计算。

2. 黑白条纹和红绿条纹的视觉分辨力比较：
   在实验中，通过比较黑白条纹和红绿条纹的间隔，来比较它们的视觉分辨力。不同的人对颜色的感知不同，在颜色视觉方面有差异，对于颜色不易分辨的人，黑白条纹和彩色条纹的感知间隔距离应该是相同的。而对于那些能够区分颜色的人，红绿条纹间隔应该比黑白条纹间隔窄一些。

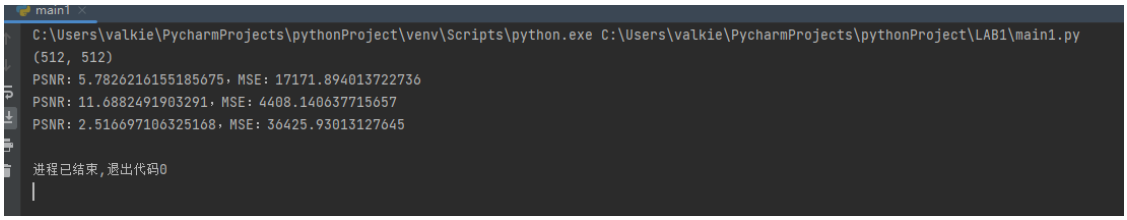
3. 灰度图像的二值化：
   灰度图像通常会有一些噪声，我需要将其转化为二值图像来方便后续处理。实现灰度图像的二值化可以使用全局阈值法或者人工阈值。全局阈值法将所有像素的灰度值平均值作为阈值，然后将所有像素的灰度值与该阈值进行比较，以确定二值图像。人工阈值则是由人工分析图像来确定阈值。

三、实验结果与分析：

1. 图像的三种操作：
   我选择了一张 512*512 像素的灰度图像进行操作。对于剪切，我将图像剪切为 256*256；对于旋转，我将图像逆时针旋转了 90 度；对于加滤镜，我

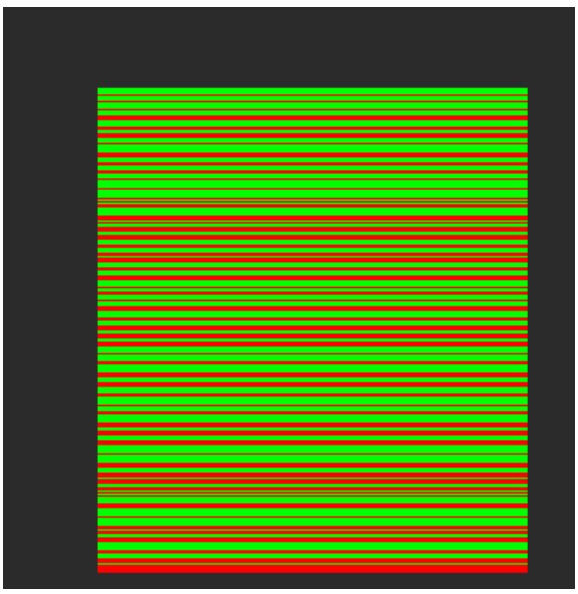
使用了 PIL 库中自带的 FIND_EDGES
计算结果如下表所示：

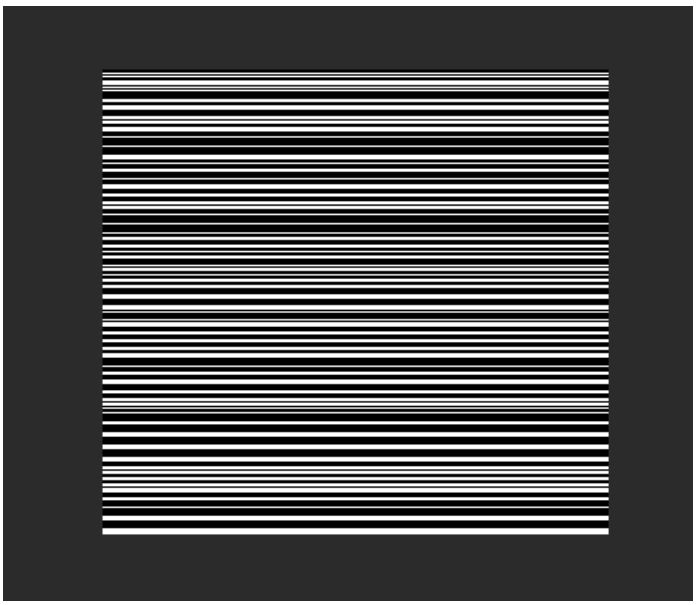
操作	MSE	PSNR
原图像	0	inf
剪切	17171.89	5.782
旋转	4408.14	11.688
滤镜	36425.93	2.516



从上表可以看出，剪切，旋转和加滤镜都会不同程度的影响图像的 MSE 和 PSNR。

1. 黑白条纹和红绿条纹的视觉分辨力比较：
   我分别生成了一张大小为 300x300 像素的黑白条纹和红绿条纹。分别进行肉眼观察





可以看出，红绿条纹间隔应该比黑白条纹间隔窄一些，这是人眼的视觉错觉造成的

3.灰度图像的二值化：

我选择了一张大小为 512*512 像素的已经经过预先灰度处理的校徽图像进行二值化。对于人工阈值，我选择了 80 和 200 来进行二值化；对于全局阈值法，我将所有像素的灰度值平均值作为阈值。

原始灰度图

阈值 200

阈值 80

阈值 127

比较结果如下表所示：

阈值类型	阈值 1	阈值 2	MSE	PSNR
全局阈值法	127	127	1351.4	8.69
人工阈值	80	200	427.14	12.52

从上表可以看出，随着阈值的增加，二值化效果变得越来越明显，并且使用人工阈值的效果要好于全局阈值法。

综上，我通过本次实验，深入学习了图像处理的基本概念和方法，并对比了不同实验结果的优劣，对未来的实践工作提供了理论和实践指导。