



学生姓名： 丁晓琪

学生学号： 22336057

专业名称： 计科

一：Otsu方法的理论

- 计算归一化的直方图: $p_i = \frac{n_i}{MN}$ $\sum_{i=0}^{L-1} p_i = 1$
- 假设以像素值k为两个类别的分界, 计算类别C1的概率累积和: $P_1(k) = \sum_{i=0}^k (p_i)$
- 计算类别的累积加权均值: $m(k) = \sum_{i=0}^k i p_i$
- 计算全局灰度加权均值: $m_G = \sum_{i=0}^{L-1} i p_i$
- 计算类间方差: $\sigma_B^2(k) = \frac{[m_G P_1(k) - m(k)]^2}{P_1(k)[1 - P_1(k)]}$
- 取出使得 $\sigma_B(k)$ 最大的k值为阈值
- 比阈值低的像素值都为0, 比阈值高的像素灰度都为255

二：实现

1. 先计算归一化后的直方图 `hist, bin_edges`
2. 画出直方图, 不需要平滑处理, 有明显的两个波峰和一个深的波谷
3. 计算类别1的概率累积和 `P_1_K`
4. 计算类别的累计加权均值 `m_k` 和全局灰度加权均值 `m_G`
5. 计算类间方差: `sigma_B_sq`
6. 找到最佳阈值: `k_star`
7. 画出阈值处理后的图像

```
1 def ostu_threshold(image):
2     # 计算图像直方图 hist存储出现次数, bin_edges存储范围
3     hist, bin_edges = np.histogram(image, bins=256, range=(0, 256))
4     # 归一化直方图
5     hist = hist.astype(float) / hist.sum()
6     hist_print(hist, bin_edges)
7     # 类间方差计算
8     P_1_k = np.cumsum(hist)
9     m_k = np.cumsum(hist * np.arange(256))
10    m_G = m_k[-1]
11    sigma_B_sq = (m_G * P_1_k - m_k) ** 2 / ((P_1_k + 1e-10) * (1 - P_1_k + 1e-10))
12    # 找到最佳阈值
```

```
13 k_star=np.argmax(sigma_B_sq)
14 print(P_1_k)
15 print(k_star)
16 # 得到阈值后的图像
17 binary_image = (image > k_star).astype(np.uint8) * 255
18 return binary_image
```

三：实验结果

找到的最佳阈值为126，可看见在直方图中126能够很好地将两个波分离，从而阈值后的图像前景和背景能很好的分离（前景为255，背景为0）

