《图像分析与理解》课程——模糊方法实验

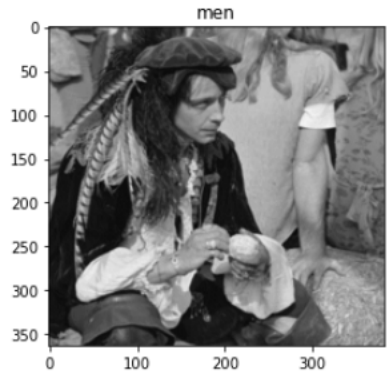
212138 卓旭

Part 1 – 模糊边界提取

1. 读取 men.txt

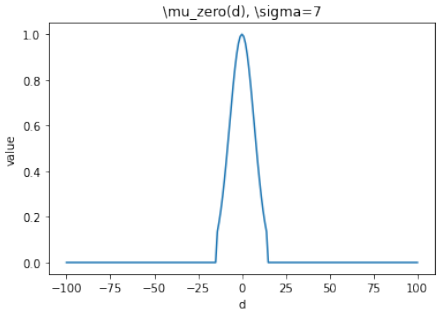
按行扫描读取即可，尺寸正确，显示如下：

```
(363, 381)
```



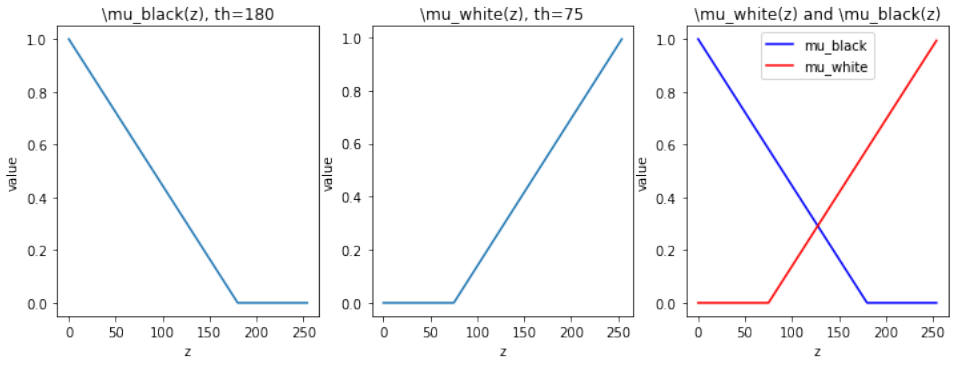
2. $\mu_{zero}(d)$的定义

按照所给公式计算即可，得到曲线：



3. $\mu_{black}(z)$、$\mu_{white}(z)$的定义

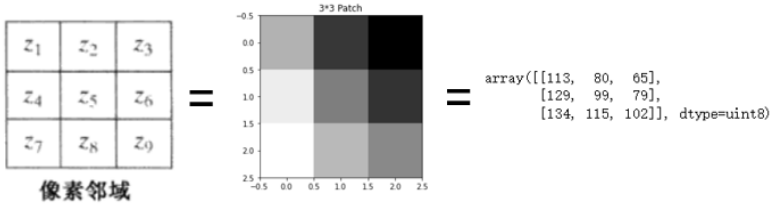
同样按照所给公式计算即可，得到曲线：



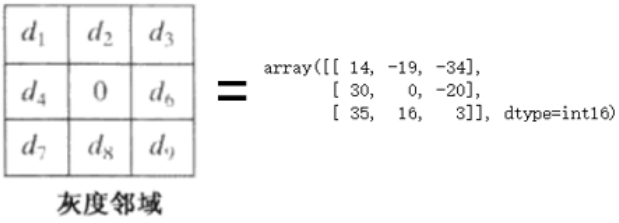
第三子图是将两条隶属度曲线画在一起的结果，可以看到中间部分形成了类似于软阈值的效果。

4. 模糊集操作

a) 考察第一个 3×3 邻域如下：

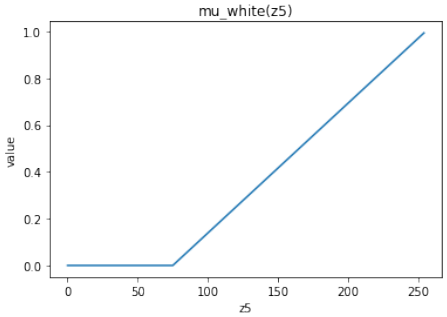


b) 考察像素差如下：

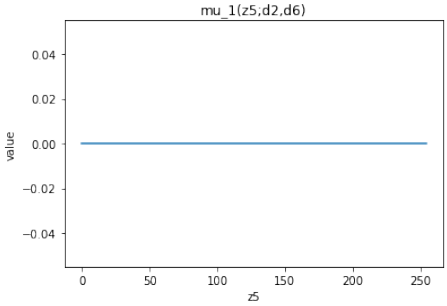


c) 程序求得$\mu_{zero}(d_2)=0$、$\mu_{zero}(d_6)=0$，验证正确，因为-19 和-20 的绝对值均超出$2\sigma=14$。

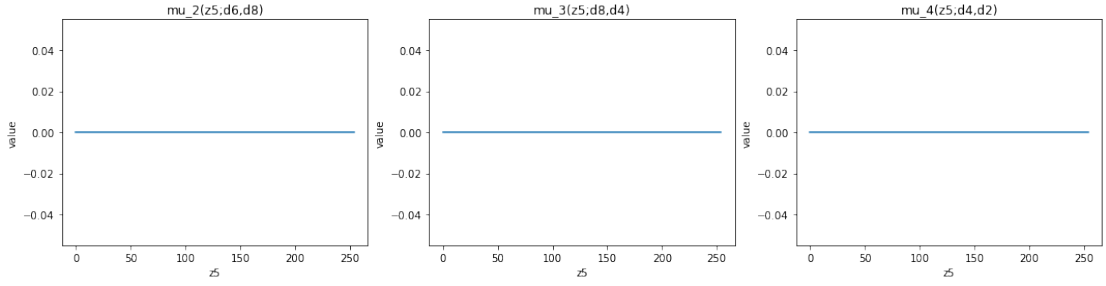
进一步，计算$\mu_{white}(z_5)$，得到变化曲线如下：



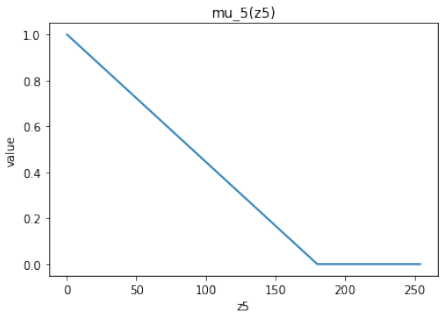
进一步求$\mu_1(z_5;d_2,d_6)=\min\left(\mu_{zero}(d_2),\mu_{zero}(d_6),\mu_{white}(z_5)\right)$得到曲线如下：



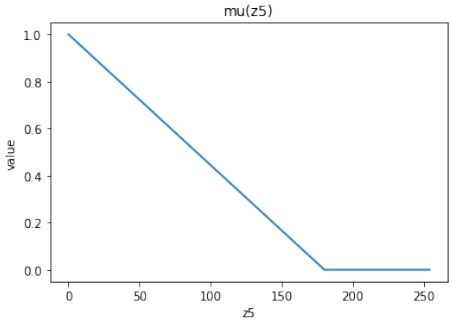
d) 同理得到$\mu_2(z_5;d_6,d_8),\mu_3(z_5;d_8,d_4),\mu_4(z_5;d_4,d_2)$：



e) 第五条规则曲线如下，与前文 3.步骤的结果相同



f) 总的隶属度函数$\mu(z_5) = \max(\mu_1(z_5), \mu_2(z_5), \mu_3(z_5), \mu_4(z_5), \mu_5(z_5))$曲线如下：



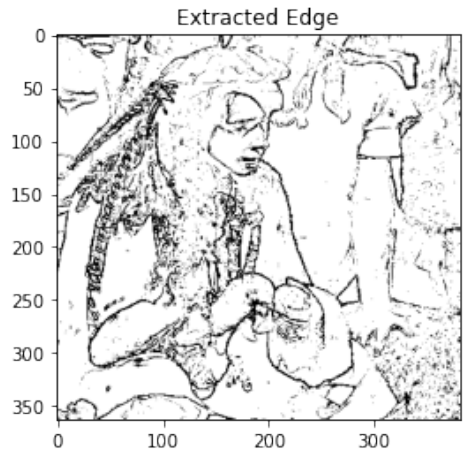
g) 使用重心规则来 defuzzify，求$z_5 = \frac{\sum_{z_5=0}^{255} z_5 \mu(z_5)}{\sum_{z_5=0}^{255} \mu(z_5)}$后得到 59.67，考虑到隶属度类似于概率分布的含义，使用四舍五入来取整更合适，得到 60。普遍地，最终结果应四舍五入后裁剪到[0, 255]并转回 uint8 型。

5. 全图提取边界

将上述步骤整理为一个函数，删去无用的中间输出部分，从而可对每个像素（每个邻域）进行处理，函数在 helper.py 的 fuzzyEdgeExtract3x3Py。

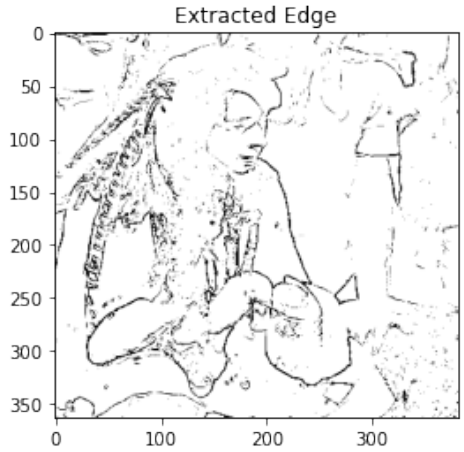
考虑到并行计算的可能性，使用 CUDA 编写了 GPU 并行加速版的相应函数，在 cuFuzzy.cu，调用接口在 cuFuzzy.py 的 fuzzyEdgeExtract3x3。未加速版本运行时间 3 分 20 秒，CUDA 版本运行时间 0.03 秒，加速约 6666 倍。

最终的边界提取结果如下：

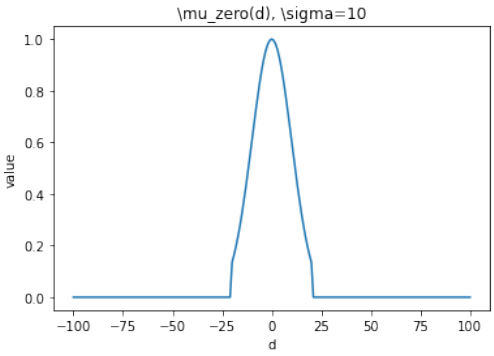


6. 调整 $\sigma = 10$ 后重复

前文使用 $\sigma = 7$，本节调整 $\sigma = 10$，其余步骤保持一致，得到边界提取结果如下：



简要解释如下。此时关于“是不是 0”的隶属度函数$\mu_{zero}(d)$曲线变为：



相比前文的曲线，钟形放宽了。这意味着专家知识中，“d_i是 0 AND d_j是 0”的四条规则更容易满足了，那么z_5是白色的情况就变多了，反映到提取的边界中时就是黑色像素变少了。更进一步地解释，是“像素属于边界”的条件变得更严格了，即需要更大的差异d_i才能满足 2σ 的限定，所以提取到的边界更少了，留下的更多属于“很明显、很强”的边界。

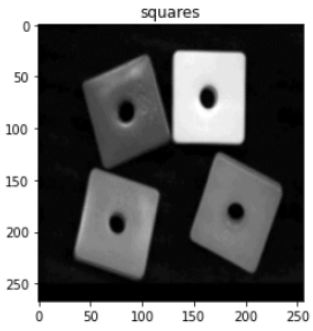
IF d_2是 0 AND d_6是 0 THEN z_5是白色
IF d_6是 0 AND d_8是 0 THEN z_5是白色
IF d_8是 0 AND d_4是 0 THEN z_5是白色
IF d_4是 0 AND d_2是 0 THEN z_5是白色
ELSE z_5是黑色

Part 2 – 模糊阈值分割

1. 读取 squares.txt

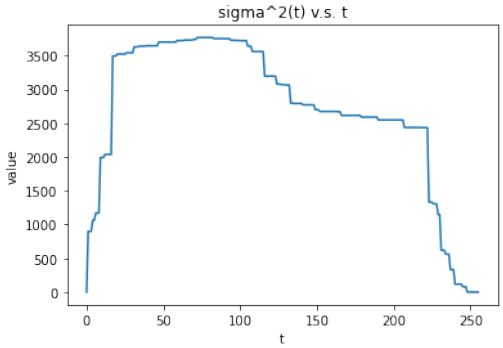
按行扫描读取即可，尺寸正确，显示如下：

```
(267, 256)
```

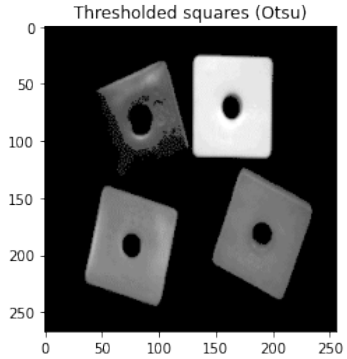


2. 直接 Otsu 阈值分割

按描述步骤实现 Otsu 阈值分割算法，求取$\sigma^2(t)$的函数为 calcOtsuSigma2t，试验 0~255 阈值得到变化曲线如下：



取 argmax 位置对应灰度作为阈值，为 78。将 squares 小于阈值的像素全部置 0（黑），得到分割后结果图如下：

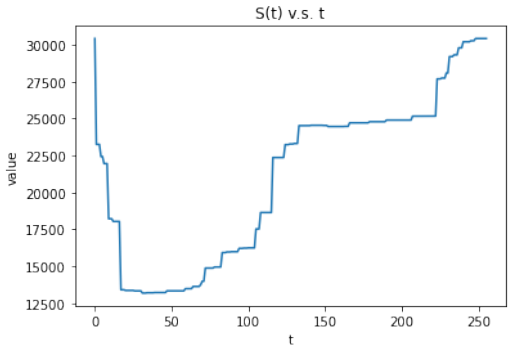


可以看到左上角的方块分割效果不好，出现残缺。

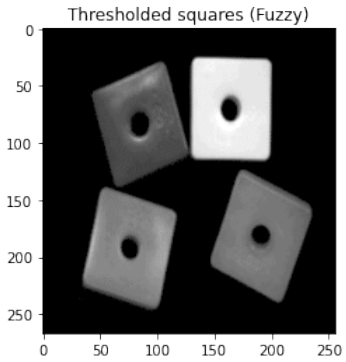
3. 模糊阈值分割

同样按描述步骤实现模糊阈值分割算法，求取全体像素熵和的函数为 calcFuzzyEntropy。注意到熵的公式$S(X) = -\mu_X(X)\,ln\,\mu_X(X) - (1-\mu_X(X))\,ln(1-\mu_X(X))$可能在数值上出现$0\log 0$导致计算报错，由于$0\log 0 = 0$，只需让结果受前项支配，可在log的操作数中加一小量解决问题。

得到熵和随阈值变化曲线如下：



按最小熵原则取 argmin 对应 t 为 31。同样将 squares 小于阈值的像素全部置 0（黑），得到分割后结果图如下：



可以看到左上角的方块分割效果变好了，没有残缺。