

计算机视觉

——二值腐蚀与膨胀

罗宇辰 516030910101

提交内容

- ◆ 代码
bi_op.py & main.py
- ◆ 文档
- ◆ example (示例图片)
- ◆ result (处理结果)

函数和实现类介绍

- bi_op.py

二值图像的数学形态学操作实现

- ✓ Function: toBilmage

@param: img矩阵

@功能: 将一个多通道普通图像进行简单二值化处理, 返回单通道二值图像

- ✓ Class: Bilmage

- Dilation (img) 膨胀操作
- Erosion (img) 腐蚀操作
- Open (img) 开操作
- Close (img) 闭操作

- main.py

程序启动入口

可以修改参数, 设置结构元的大小、内容和中心点

具体函数实现解释

- 简易的灰度图像二值化

```
def toBiImage(img):    #translate a normal image to a binary image
    H,W,C = img.shape
    img = img.reshape(C,H,W)
    for c in range(C):
        for h in range(H):
            for w in range(W):
                if img[c,h,w]>128:
                    img[c,h,w]=255
                else:
                    img[c,h,w]=0
    return img.reshape(H,W,C)[:,:,:0]
```

简易的二值化处理

返回第一个通道的灰度值

具体函数实现解释

- 膨胀操作（腐蚀与其类似）

```
def dilation(self, img):  
    result = img.copy()  
  
    H, W = result.shape  
    HH, WW = self.SE.shape  
  
    for h in range(H-HH+1):  
        for w in range(W-WW+1):  
            area = img[h:h+HH, w:w+WW]  
  
            match = False  
            for hh in range(HH):  
                for ww in range(WW):  
                    if self.SE[hh, ww] == 255 and area[hh, ww] == self.SE[hh, ww]:  
                        match = True  
                        break  
  
            if match:  
                result[h+self.centR, w+self.centC] = 255  
            else:  
                result[h+self.centR, w+self.centC] = 0  
  
    return result
```

选中要进行匹配的区域

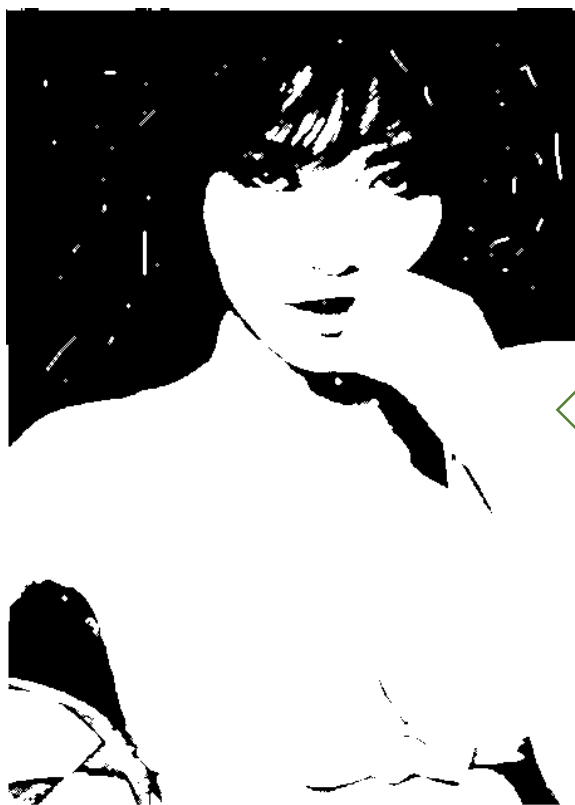
将结构元和选中区域中的元素一一比对，若至少有一个命中，则判断结构元与选中区域匹配

若匹配，则结果区域中结构元中心点位置处元素指为255，否则为0

图像处理结果

结构元:

0	255	0
255	0	255
0	255	0



膨胀



原图

腐蚀



结构元:

0	255	0
255	0	255
0	255	0



腐蚀



开



膨胀



闭



原图

结构元:

255	0
0	255

腐蚀

开

膨胀

闭

原图