Computer Vision: Representation and Recognition
Assignment 2

211220075, 孟旷宇, 1665113825@qq.com

2024 年 5 月 25 日

1 Edge Detection

1.1 Description of any design choices and parameters

在基于梯度的边界检测函数设计上，处理步骤为：

1. 使用高斯滤波器平滑图像；
2. 为每个像素计算它每个通道上沿 x 方向和沿 y 方向的梯度，并计算两个方向梯度的欧氏距离，由此得到每个像素在三个维度上的**梯度大小**；
3. 取每个像素在三个通道上梯度大小的欧氏距离作为总的梯度大小，并以梯度大小最大的那个通道的梯度方向作为总的**梯度方向**。

参数选择方面：向 gradientMagnitude(im, sigma) 传入 sigma = 2.1，并选择使用 edge(mag, 'Canny') 进行非极大值抑制。

在基于方向滤波器的边界检测函数设计上，处理步骤为：

1. 分别求出高斯滤波器关于 x 与 y 的一阶偏导，并使用 0°、45°、90° 以及 135° 四个方向计算滤波器组；
2. 将需要检测边缘的图像转换为灰度图像，并用选择的滤波器组处理（卷积）；
3. 将处理后得到的四个矩阵归一化，并选择绝对值最大的数值以及对应方向作为每个像素的**边界大小以及边界方向**。

参数选择方面，滤波器组在构建时选择 sigma = 2.6，并模仿 matlab 官方将 $2*ceil(2*sigma)+1$ 作为滤波器的 ksize，极大值抑制的函数同样选择 edge(mag, 'Canny') 而非中间插值法。

1.2 The bank of filters used for part (b)

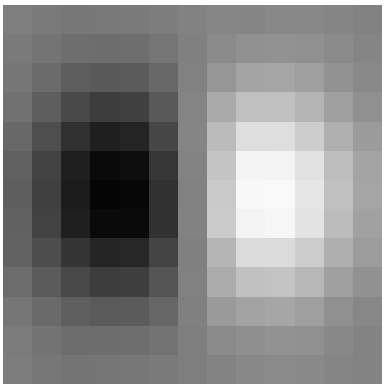

图 1: 0° 高斯滤波器

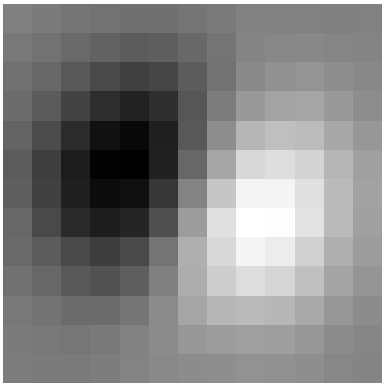

图 2: 45° 高斯滤波器

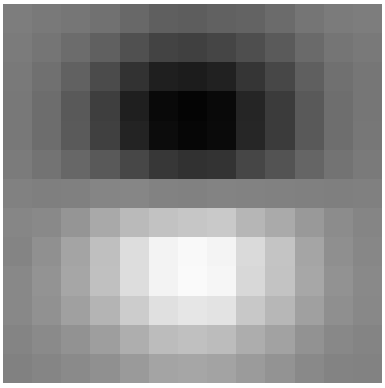

图 3: 90° 高斯滤波器

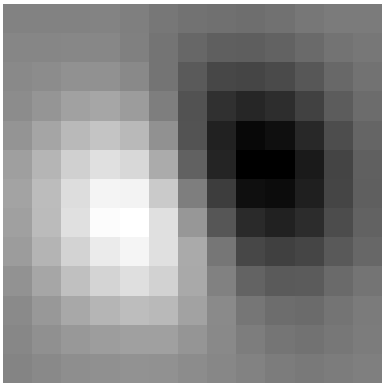

图 4: 135° 高斯滤波器

1.3 Qualitative results: choose two example images; show input images and outputs of each edge detector

example 1

图 5: input image

图 6: gradient

图 7: oriented

example 2

图 8: input image

图 9: gradient

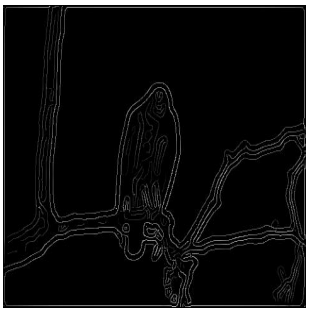

图 10: oriented

1.4 Quantitative results: precision-recall plots and tables showing the overall F-score and the average F-score

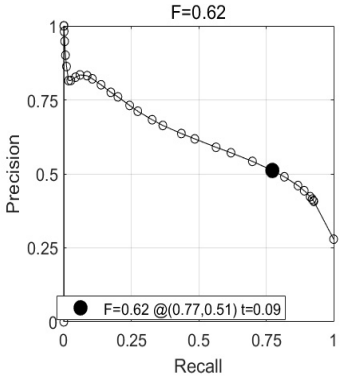

图 11: gradient-p-r 曲线

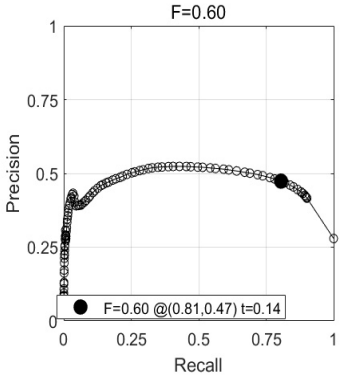

图 12: oriented-p-r 曲线

average F-score:

Method gradient: 0.652;
Method oriented: 0.615.

1.5 Ideas for improvement

1.5.1

在设计**基于方向滤波器的边界检测函数**时，我是直接将三维图像转换为灰度图像进行卷积与处理的，并没有有效使用到三个通道的数据，我认为这里同样可以模仿**基于梯度的边界检测函数**的设计：针对每个通道进行卷积，并求出不同方向滤波器下、三个通道结果的欧氏距离作为更好的**边界大小**；边界方向也一样，可以把一个像素在同一方向滤波器下、不同通道上的边界分数进行加权或粗暴地求欧氏距离，并以此来求 argmax 作为边界方向。

1.5.2

通过调参的形式确定最适合当前数据集的滤波器组以及传给高斯（方向）滤波器的 sigma 值。

2 Question 2

2.1 Choose parameter values (k, numRegions, winSize, etc) that yield a reasonable looking segmentations for each feature type and display results for the provided images

经过调参测试，我认为 k = 6，numColorRegions = 6，numTextureRegions = 6，winSize = 5 并且在创建 texton codebook 时每幅图片随机取样 num = 100 个像素点比较合适，下面是分割结果的展示：

图 13: original picture

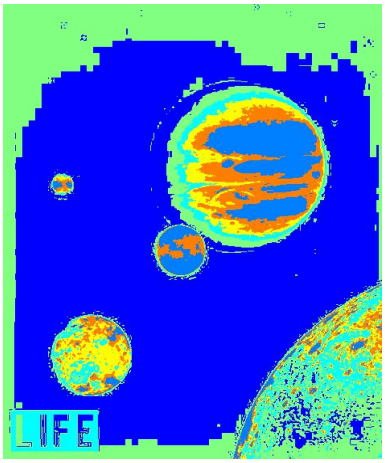

图 14: color segment

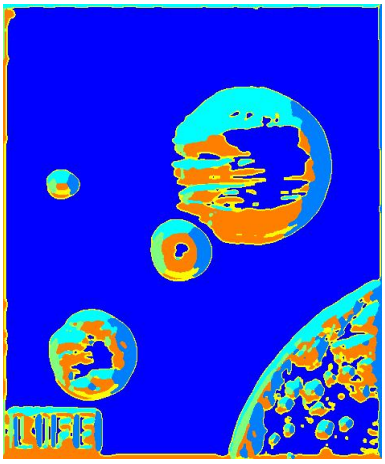

图 15: texture segment

2.2 Consider two window sizes for the texture representation, a smaller and larger one, with sizes chosen to illustrate some visible and explainable tradeoff on one of the example images

我选择的 winSize 分别为 5 与 15，不同 winSize 下纹理分割的结果如下，可以看到，小的窗口会使分割结果中的细节更加突出，这就使得蛇的鳞片与周围沙子的颜色十分接近；而大的窗口会让蛇的轮廓更加清晰，使得蛇的部分与周围的沙子更加分离与突出。

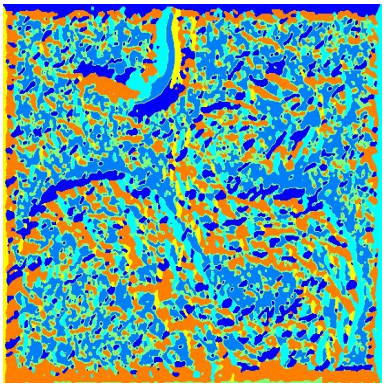

图 16: small winSize

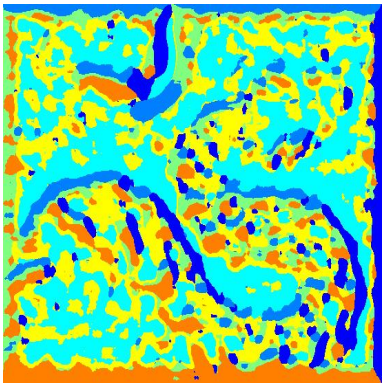

图 17: large winSize

2.3 Run the texture segmentation results with two different filter banks

我选择使用滤波器组的前六个作为子集进行处理，得到结果如下：（其中，图 18 为使用所有滤波器的纹理分割结果，图 19 为使用滤波器子集的纹理分割结果），可以看出：图 19 注重对形状、边界等细节进行追踪，将背景分割成许多类别而不能很好地统一为一个整体，并且作为前景的人的衣服也因为上面的条纹被分割成多个类；与之相对的就是图 18 中作为背景的草地基本被划分为一个大的类别，作为前景的人的衣服虽然有条纹存在，也基本被勾勒成一个整体。

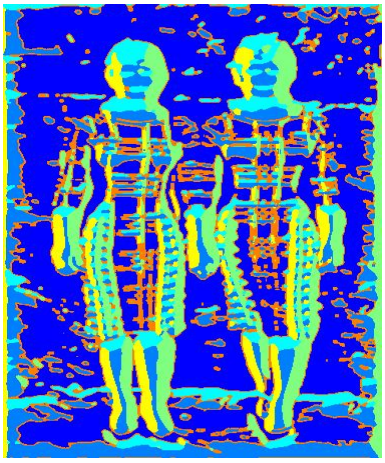

图 18: all filters

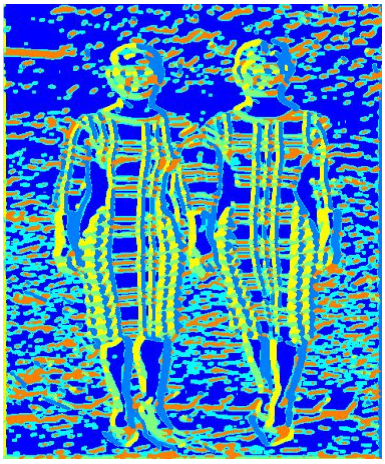

图 19: subset of filters

reference