

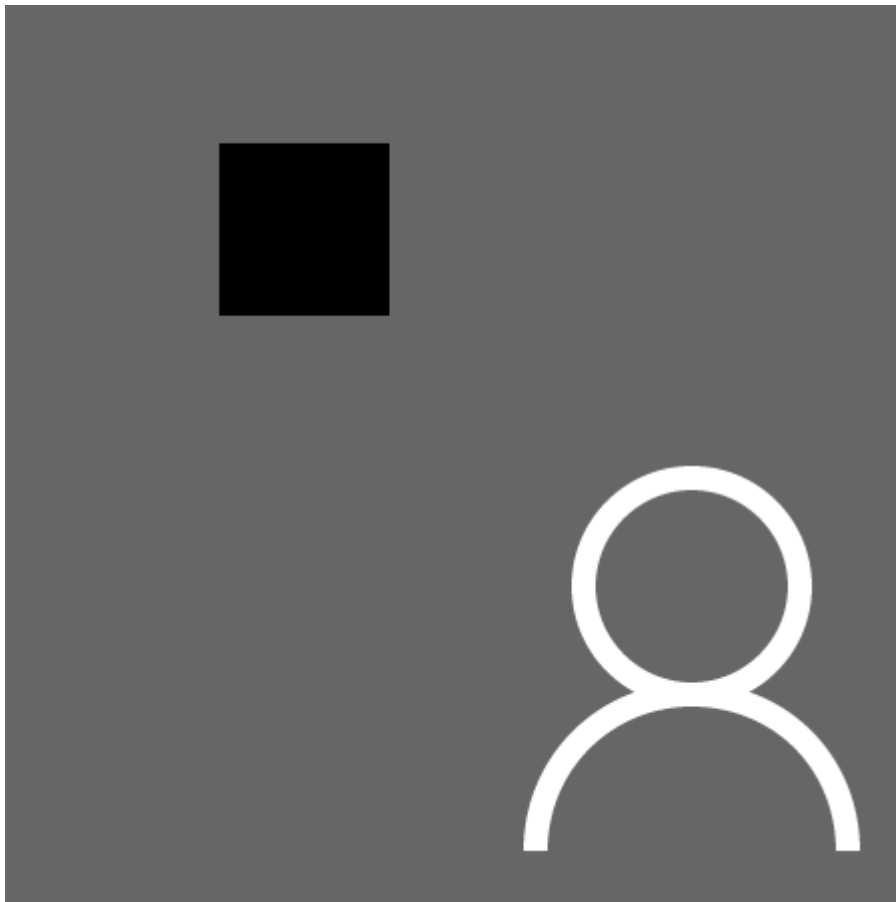
计算机视觉和机器学习 Ex1 测试文档

1. 测试环境

Windows 10 + VS2015

2. 测试数据

课程提供的图片 1.bmp, 如下：



处理要求如下：将图片中的白色区域变成红色，黑色区域变成绿色，同时绘制两个圆形区域，分别是圆心坐标为(50,50)、半径为30、填充蓝色和圆心坐标为(50,50)、半径为3、填充颜色为黄色。

3. 测试结果

本次实验使用 CImg 库进行处理，使用 load_bmp 方法进行图片

的加载，使用 draw_circle()方法进行圆的绘制，同时遍历所有像素点，进行颜色的调整，具体代码如下：

```
int main()
{
    //读取图像
    CImg<unsigned char> SrcImg;
    SrcImg.load_bmp("1.bmp");
    int w = SrcImg._width;
    int h = SrcImg._height;

    //显示图像
    SrcImg.display();

    //将白色区域变成红色,黑色区域变绿色
    CImg<unsigned char> Tar_Img("1.bmp");

    cimg_forXY(SrcImg, x, y) {
        if (SrcImg(x, y, 0) == 255 && SrcImg(x, y, 1) == 255 && SrcImg(x, y, 2) == 255) {
            Tar_Img(x, y, 0) = 255;
            Tar_Img(x, y, 1) = 0;
            Tar_Img(x, y, 2) = 0;
        } else if (SrcImg(x, y, 0) == 0 && SrcImg(x, y, 1) == 0 && SrcImg(x, y, 2) == 0) {
            Tar_Img(x, y, 0) = 0;
            Tar_Img(x, y, 1) = 255;
            Tar_Img(x, y, 2) = 0;
        }
    }

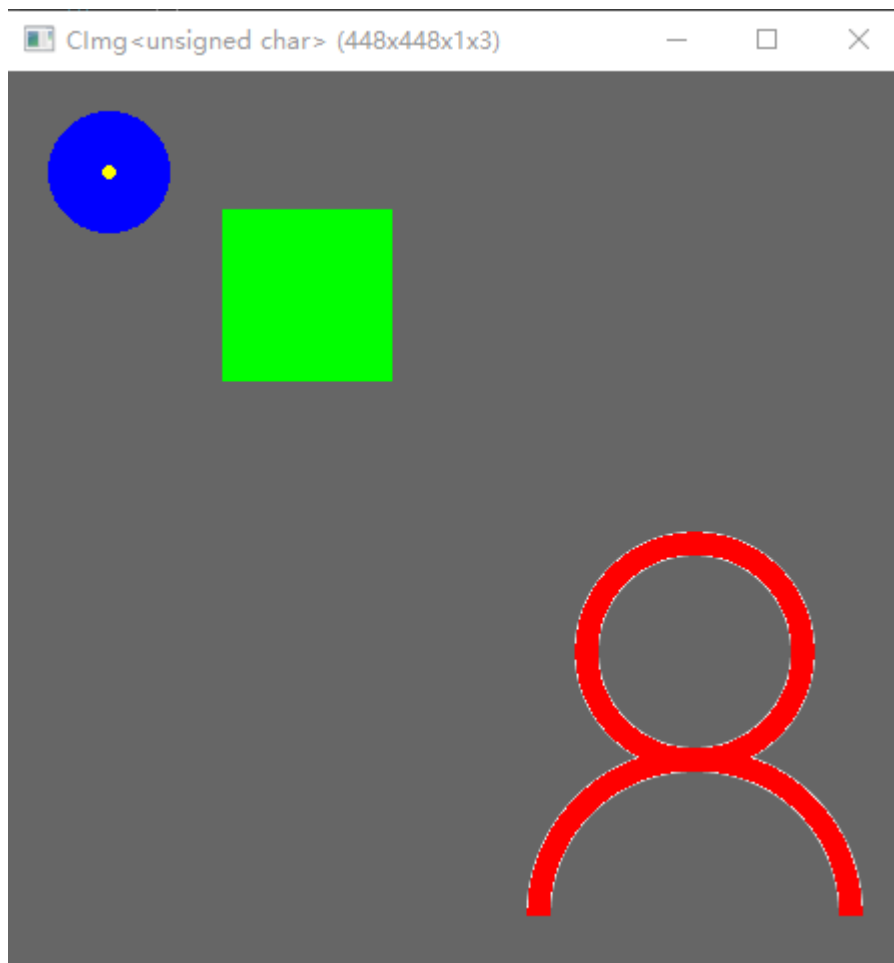
    //绘制圆形，坐标为（50,50），半径为30，填充蓝色。半径为3，填充黄色。
    unsigned char blue[] = {0, 0, 255};
    unsigned char yellow[] = { 255, 255, 0 };
    Tar_Img.draw_circle(50, 50, 30, blue, 1);
    Tar_Img.draw_circle(50, 50, 3, yellow, 1);

    //Tar_Img.blur_median(3);

    Tar_Img.display();
    Tar_Img.save("result.bmp");

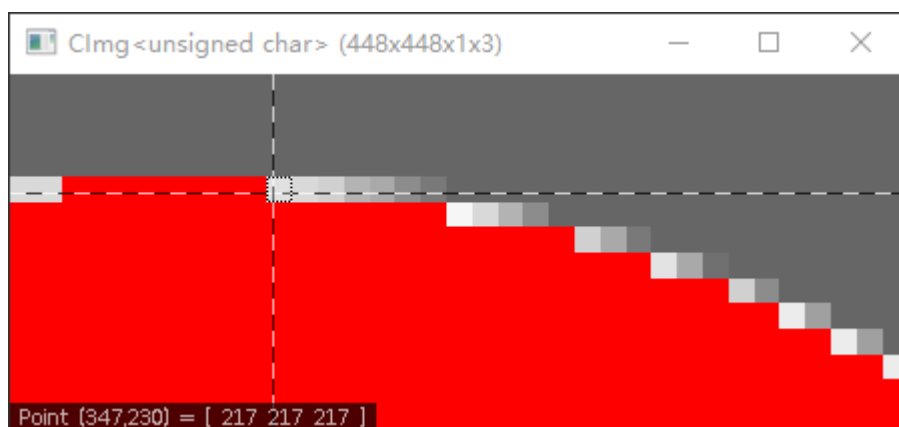
    return 0;
}
```

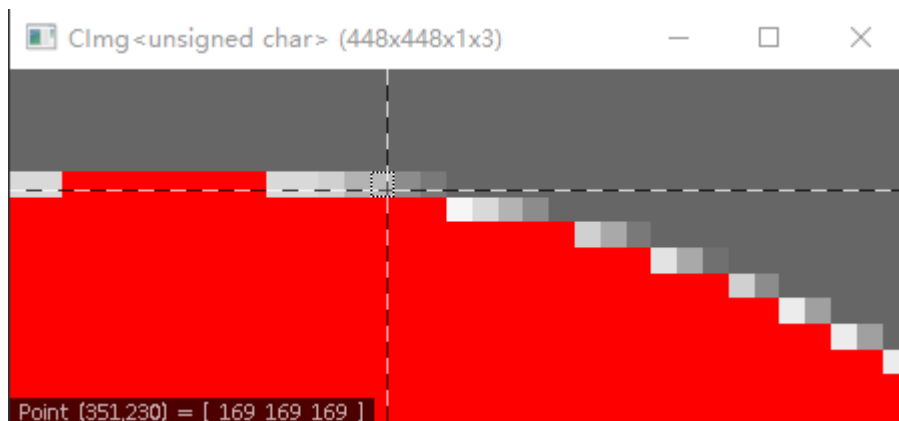
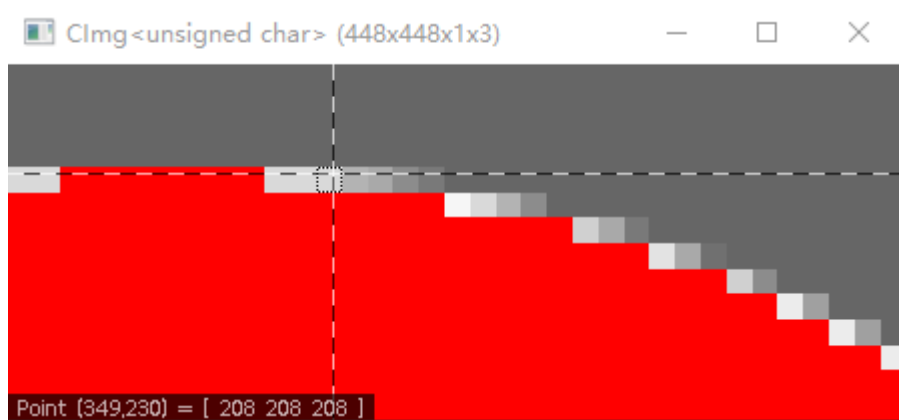
效果如下：



4. 结果分析

从上图可以看出，将白色区域变成红色区域的部分效果并不好，放大查看可以得到如下：

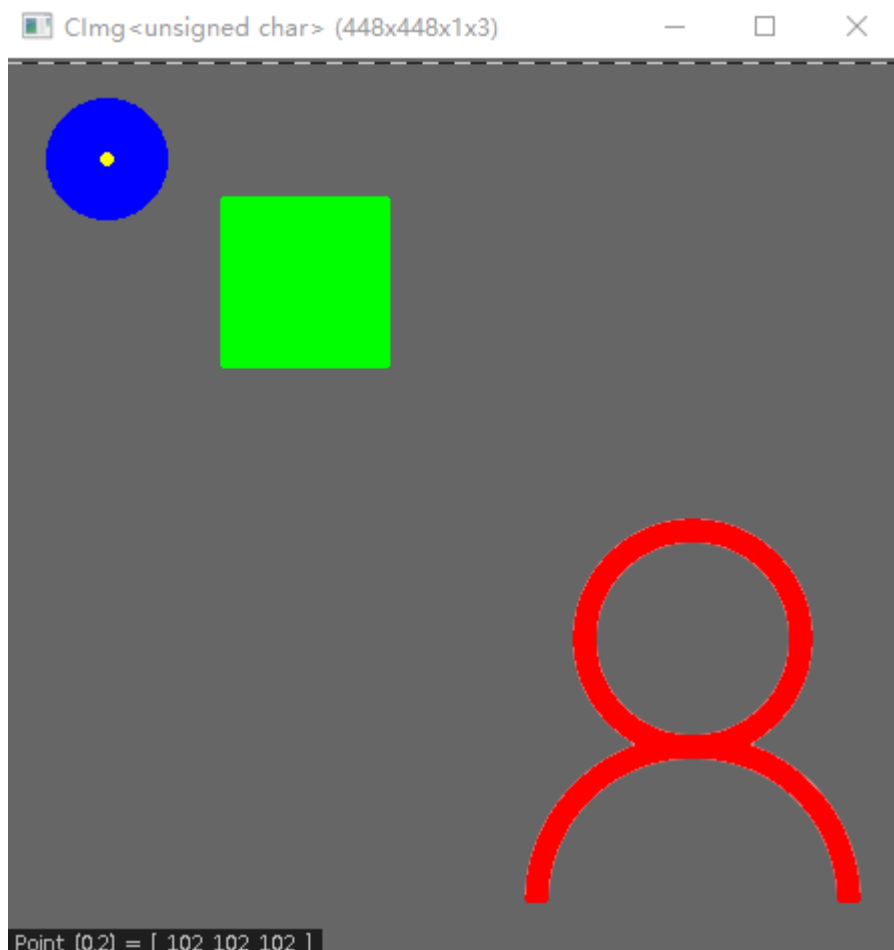




可以看出白色区域中的白色并不是都是全部 RGB 的值为 (255,255,255)，因此一些点并没有被改成颜色，为了使效果更好，这里使用中值滤波进行处理，即：

```
Tar_Img.blur_median(3);
```

效果如下：



对比上面没有经过中值滤波处理过的结果，这个显然更为自然。

5. 思考题：

为什么第四步绘制的圆形区域形状效果不好？

原因是第四步要求绘制的圆形半径太小了，因此在光栅化的过程中所用像素很少，就导致了方块感明显，边缘不平滑，边缘有锯齿等，对比与第三步绘制的圆形，第三步绘制的圆形半径大，所用的像素较多，因此在边缘较为平滑。