

计算机视觉 课程实验报告

学号：201800130086	姓名：徐鹏博	
-----------------	--------	--

实验题目：图像基本操作

实验过程中遇到和解决的问题：

（记录实验过程中遇到的问题，以及解决过程和实验结果。可以适当配以关键代码辅助说明，但不要大段贴代码。）

实验 1.1：图像加载、显示

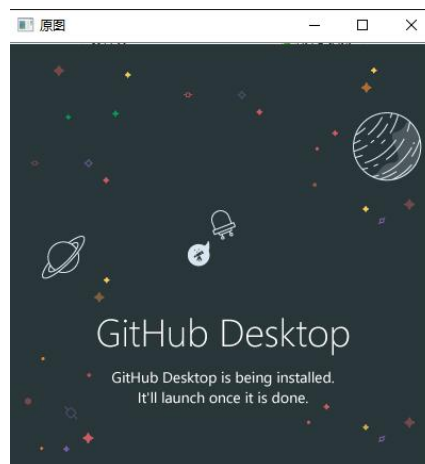
配置环境后图像无法正常加载，发现是链接文件配置错误，链接文件需要对照对应的编译模式。

显示部分的代码：

```
Mat img = imread("B://CV//exp//images//10111.png");
if (img.empty()) {
    cout << "图像不存在" << endl;
    return -1;
}
imshow("原图", img);
```

成功加载图像：

Cat.jpg , icon.png



实验 1.2：图像通道分离

三通道分离出单色通道的时候仍然需要设置为 U8C3, 如果设置为 U8C1 时只有灰度图。

实现通道获取函数：

//遍历每个像素, 把需要分离出来的通道数据保留到新 Mat 的 data 里, 把剩下的通道都设为 0。

```
void getChannel(const uchar* input, int width, int height, int inStep, int inChannels,
               uchar * output, int outStep, int channelToGet) {
    for (int i = 0; i < height; ++i) {
        for (int j = 0; j < width; ++j) {
            for (int k = 0; k < inChannels; ++k) {
```

```

        if (k == channelToGet)
            output[i*outStep+j*inChannels+k] = input[i*inStep+j*inChannels+k];
        else output[i*outStep+j*inChannels+k]=0;
    }
}
}
}
}

```

实现结果:

原图和三个子通道的图



实验 1.3 图像子区域操作

只需要以 x,y 为起点，遍历设置的矩形长度和宽度参数范围内的像素，将其分离出来即可。

```

Mat localGetChannel(const Mat input, Rect R, int channelToGet) {
    int x = R.x, int y = R.y, int w = R.width, int h = R.height;
    Mat Output; Output.create(h, w, CV_8UC3);
    uchar* inM = input.data;
    int wi = input.rows, he = input.cols, inStep = input.step, inChannels = input.channels();

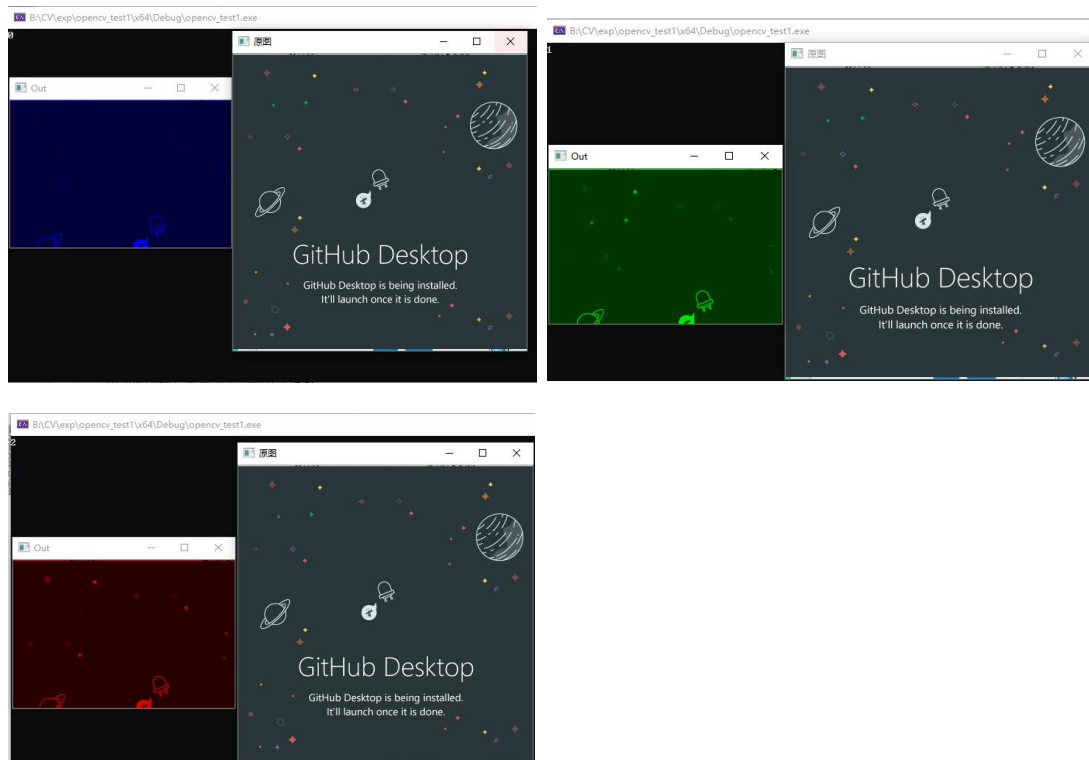
```

```

uchar* OutM = Output.data;
int outStep = Output.step;
for (int i = 0; i < h; ++i) {
    for (int j = 0; j < w; ++j) {
        for (int k = 0; k < 3; ++k) {
            if (k == channelToGet)
                OutM[i*outStep+j*inChannels+k] = inM[(x+i)*inStep + (y+j)*inChannels+k];
            else OutM[i*outStep + j*inChannels + k] = 0;
        }
    }
}
return Output;
}

```

实验结果:



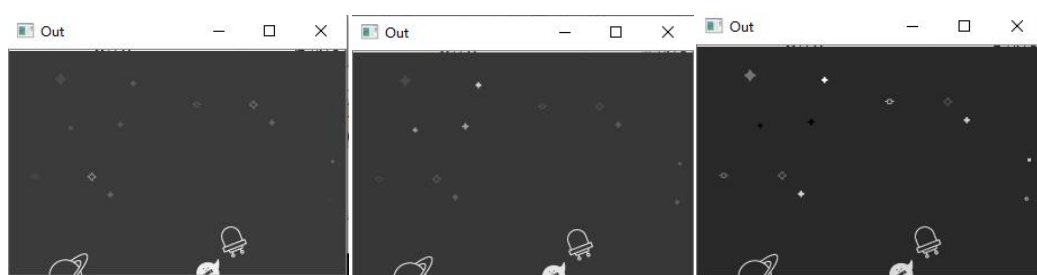
如果 `O` 是一个数据类型为 `CV_8UC1` 的图像, `getChannel` 可以直接输出到 `O` 的某一个子区域中, 但是最后会是一个对应的 `8UC3` 子通道图的灰度图。

如:

原图:



三个通道分别输出的 8UC1 图像:



结果分析与体会:

图像在计算机中逻辑上以二维数组存储, 当图像是 BGR 交叉存储时, 以 8 位三通道的图像, 每个像素就占有三个通道数据, 需要分离出单通道的时候, 就将其中需要的子通道数据提取到新的 Mat 对象对应的数组中存入, 最后在桌面显示出来。