

## 高级图像处理与分析课程实验报告

学号：SA23225226      姓名：郭浩天      日期：2023.10.20

实验名称	图像灰度变换
实验内容	<p>1、利用 <code>Opencv</code> 读取图像 具体内容:用打开 <code>OpenCV</code> 打开图像，并在窗口中显示。</p> <p>2、灰度图像二值化处理 具体内容:设置并调整阈值对图像进行二值化处理。</p> <p>3、灰度图像的对数变换 具体内容:设置并调整 <code>r</code> 值对图像进行对数变换。</p> <p>4、灰度图像的伽马变换 具体内容:设置并调整 <code>γ</code> 值对图像进行伽马变换。</p> <p>5、彩色图像的补色变换 具体内容:对彩色图像进行补色变换。</p>
实验完成情况  (包括完成的实验内容及每个实验的完成程度。注意要贴出每个实验的核心代码)	<pre>int showImage() {    //显示图像     Mat image = imread("mm.png"); //载入图像     imshow("彩色图像", image);     waitKey(0);     return 0; }</pre>

```

int Binarization() {
    Mat src = imread("mm.png", IMREAD_GRAYSCALE);

    Mat res;
    //用于二值化的图像res
    res = src.clone();
    int rows = res.rows;
    int cols = res.cols;

    //二值化
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            //res为mat型变量, 用了at函数对图片上固定位置的像素点进行操作
            auto gray = res.at<uchar>(i, j);
            //根据阈值对像素点的灰度值划分, 二值化
            if (gray > 180)    gray = 255;    //纯白色
            else    gray = 0;    //纯黑色
            //saturate_cast函数的作用即是: 当运算完之后, 结果为负, 则转为0, 结果超出255, 则为255
            res.at<uchar>(i, j) = saturate_cast<uchar>(gray);
        }
    }

    imshow("灰色图像", src);
    imshow("二值化后的图片", res);
    waitKey(0);

    return 0;
}

int logTrans() {
    Mat src = imread("mm.png", IMREAD_GRAYSCALE);
    Mat res;
    res = src.clone();
    int rows = res.rows;
    int cols = res.cols;

    //对数变换
    for (int i = 0; i < rows; i++) {
        for (int j = 0; j < cols; j++) {
            //gray为double型变量, 将像素点的灰度值强制转换为double型, 方便进行对数运算
            auto gray = (double)res.at<uchar>(i, j);
            //进行对数变换, 2为对数
            gray = 2 * log((double)(1 + gray));
            //限制输入输出范围
            res.at<uchar>(i, j) = saturate_cast<uchar>(gray);
        }
    }

    //对对数变换后的值进行归一化处理
    normalize(res, res, 0, 255, NORM_MINMAX);
    convertScaleAbs(res, res);

    imshow("灰色图像", src);
    imshow("变换后的图像", res);
    waitKey(0);
    return 0;
}

```

```

int gamaTrans() {
    Mat src = imread("mm.png");
    Mat gray;
    cvtColor(src, gray, COLOR_BGR2GRAY);
    Mat img_gamma;
    gray.convertTo(img_gamma, CV_32F);
    img_gamma = img_gamma / 255.0; //像素归一化0-1
    pow(img_gamma, 2.2, img_gamma); //伽马变换, 伽马值2.2

    // 归一化到0-255
    normalize(img_gamma, img_gamma, 0, 255, NORM_MINMAX);
    convertScaleAbs(img_gamma, img_gamma);

    imshow("灰色图像", gray);
    imshow("伽马变换", img_gamma);
    waitKey();

    return 0;
}

int CplColorTrans() {
    Mat img = imread("mm.png");

    // 补色变换
    Mat img_complement;
    img_complement = Scalar(255, 255, 255) - img; //C=255-R;M=255-G;Y=255-B

    imshow("彩色图像", img);
    imshow("补色变换", img_complement);
    waitKey();

    return 0;
}

```

## 实验中的问题

(包括在实验中遇到的问题, 以及解决问题的方法)

通过对原图的每个像素按照行列依次处理。二值化处理, 根据阈值对像素点的灰度值划分, 将图像上的像素点的灰度值设置为 0 或 255。对数变换, 伽马变换和补色变换按照公式处理, 其中伽马变换先将像素归一化, 然后根据伽马值进行伽马变换。

## 实验结果

(实验完成后的  
源码和打包文  
件的说明)





