

电子科技大学实验报告

课程名称: 数学实验

实验地点: 229

指导教师: 原子霞

评 分:

完成实验学生信息:

选课序号	姓名	学号	贡献百分比/%	备注(主要工作)
113	朱若愚	2022150501027	25%	代码准备
112	黄思宇	2022150501024	25%	资料查找
105	高畅	2022150501009	25%	拟写报告
110	王宇阳	2022150501022	25%	结果测试

注:

1. 学生人数按照任课教师要求限定;
2. 对于“评价、改进、总结和体会”都要认真填写,和其他内容是评价实验成绩的重要参考。

实验题目名称：限速标志识别算法

一. 实验内容

在汽车行驶过程中，会有各种限速标志牌.有的行车记录仪已经具备了限速语音提示功能.请设计模型和算法识别出“图片集”中的限速速度值。

本实验提供了少了图片样例，学生可以自拟数据集。

图片数据见文件夹“限速图片集”。

二. 实验目的

利用 Matlab 实现算法，以识别图片中的限速标志牌，并从中提取出限速速度值。通过这个实验，掌握实现在图像处理领域的算法的能力，实现对限速数值的自动识别和信息提取。

三. 实验过程

问题：设计模型和算法识别出“图片集”中的限速速度值。

问题分析

对于这个实验，要先确定图片中限速数字的位置，用标记标出，并识别其中的数值，最终在结果中输出带有标记和数字数值的处理后的图片，方便阅读和检验。识别数值需要读取和预处理图像（先灰度化再二值化），便于识别，然后遍历匹配相应模板。

1. 模型假设

通过给定相应的模板，可以在二值化后的图片中寻找与模板相似的部分，以此来确定其出现的位置，并根据匹配的模板确定其数值为多少，从而实现识别的目的。

2. 变量与符号说明

变量名	符号说明
image	读取到的图像
grayImage	灰度化后的图像
bwImage	二值化图像
templates	模板图片
templateValues	模板图片对应的速度值，用行向量存储
numTemplates	模板图片数量
detectedPosition	检测到速度标志所在的位置

3. 模型建立与算法设计

3.1 读取和预处理图像

首先,代码通过 `imread` 读取了一个限速标志的图像。然后,它使用 `rgb2gray` 函数将图像从 RGB 颜色空间转换为灰度空间,因为灰度图像处理起来更简单、计算量更小。接着,代码使用 `imbinarize` 函数进行二值化,创建了一个黑白图像,这有助于突出数字和边缘,便于后续的模板匹配。

3.1.1 灰度化

灰度化是将彩色图像转换为灰阶图像的过程。在彩色图像中,每个像素的颜色由红(R)、绿(G)和蓝(B)三个颜色通道的强度值决定。灰度化的目的是去除颜色信息,仅保留亮度信息。这对于简化图像处理任务非常有用,因为它降低了处理的复杂度(从三个颜色通道减少到一个灰度通道)。

加权法:大多数灰度化方法使用加权平均法,其中绿色权重最高,其次是红色,蓝色权重最低。这是因为人眼对绿色的敏感度最高。一个常用的加权法是:

$$\text{Gray}=0.299\times R+0.587\times G+0.114\times B$$

这里, R,G,B 是彩色图像中每个像素的红色、绿色和蓝色分量的值,而 Gray 是得到的灰度值。

亮度保留:这种加权方法能够确保转换到灰度图像的亮度与彩色图像在人眼中的亮度感知相匹配。

3.1.2 二值化

二值化是将灰度图像转换为仅包含黑色和白色的图像的过程。二值图像通常用于进一步的图像分析,例如轮廓提取、文字识别和图像分割。

原理:

阈值决定:二值化涉及选择一个灰度阈值。灰度值高于或等于该阈值的像素将被设置为白色(通常对应于 255 或 1),而低于该阈值的像素将被设置为黑色(通常对应于 0)。

全局阈值:在一些简单的场景中,可以选择一个全局阈值应用于整个图像。这个阈值可以是固定的,也可以使用算法自动确定,如 Otsu 方法,该方法通过最大化类间方差自动计算最佳阈值。

自适应阈值:在更复杂的图像中,由于光照变化或图像不均匀,可能需要使用自适应阈值。这种方法会为图像的不同区域计算不同的阈值,更适应局部的亮度变化。

二值化效果:二值化后的图像会损失大量信息,但它可以突出一些特定的特征,例如边缘或形状,这在许多视觉应用中非常有用。

3.2 模板匹配

这部分代码定义了一系列模板图像的文件路径和相应的限速值。`templates` 数组包含了不同限速标志的模板图像路径, `templateValues` 数组包含了与这些模板相对应的限速值。

代码遍历每个模板图像,对每个模板进行相同的预处理步骤(灰度化和二值化),然后调整大小以确保模板可以与原图进行匹配。

3.3 执行模板匹配

`normxcorr2` 函数执行标准化的互相关运算，返回一个相关性映射 (`corrMap`)，表明模板在原图像上各个位置的匹配程度。

3.4 确定最佳匹配

代码通过比较各个模板的最大相关性分数来确定最佳匹配的模板。然后，它计算匹配位置的坐标。

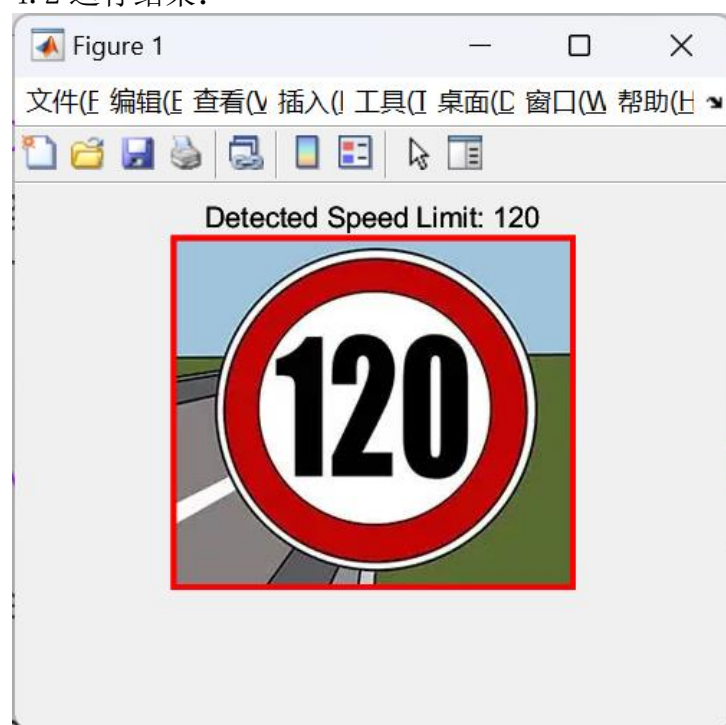
4. 运行结果及结果分析

全部运行结果在附录中，这里以图 120-1 的运行结果为例，进行分析

4.1 检测图片：



4.2 运行结果：



4.3 结果分析

结果图片中，将限速数据所在区域用红框标记，并在上方给出了监测到的限速数值，实现了实验的要求，并且识别正确，由此可以总结，我们的算法较好地实现了限速标志的识别。

四. 优缺点及改进方向

本算法的优点在于算法思想简洁，每次读取待处理的图像，进行处理后与现有模板比较、匹配，再标出限速标志的位置和限速数据。

但是，本算法也存在一些缺点和不足。对于目标的识别依赖于足够的样本模板，且字体不同时，算法也可能识别出错，导致需要的样本量大，没有对应相似模板时可能会出错。此外，对于较为复杂的图片（文字较多），可能无法匹配到正确的区域，导致结果出现误差（如附录中出错示例所示）。

对于本算法，或许可以尝试将样本模板改为单个的数字，依次进行匹配，再将图片中所有与样本匹配的区域组合起来，还原正确的数据，这样可以减小所需的模板的数量，实现算法的改进。

五. 心得体会与总结

(收获、感想；思路、方法的总结)

通过本次实验，我们小组的 MATLAB 的编程能力有了进一步的提高，真正将生活中需要的实际问题转化为数学模型，利用 MATLAB 图像处理领域的算法的能力编写程序识别图像解决问题，意识到 MATLAB 的应用范围非常广泛，同时培养了问题解决和团队合作的能力。

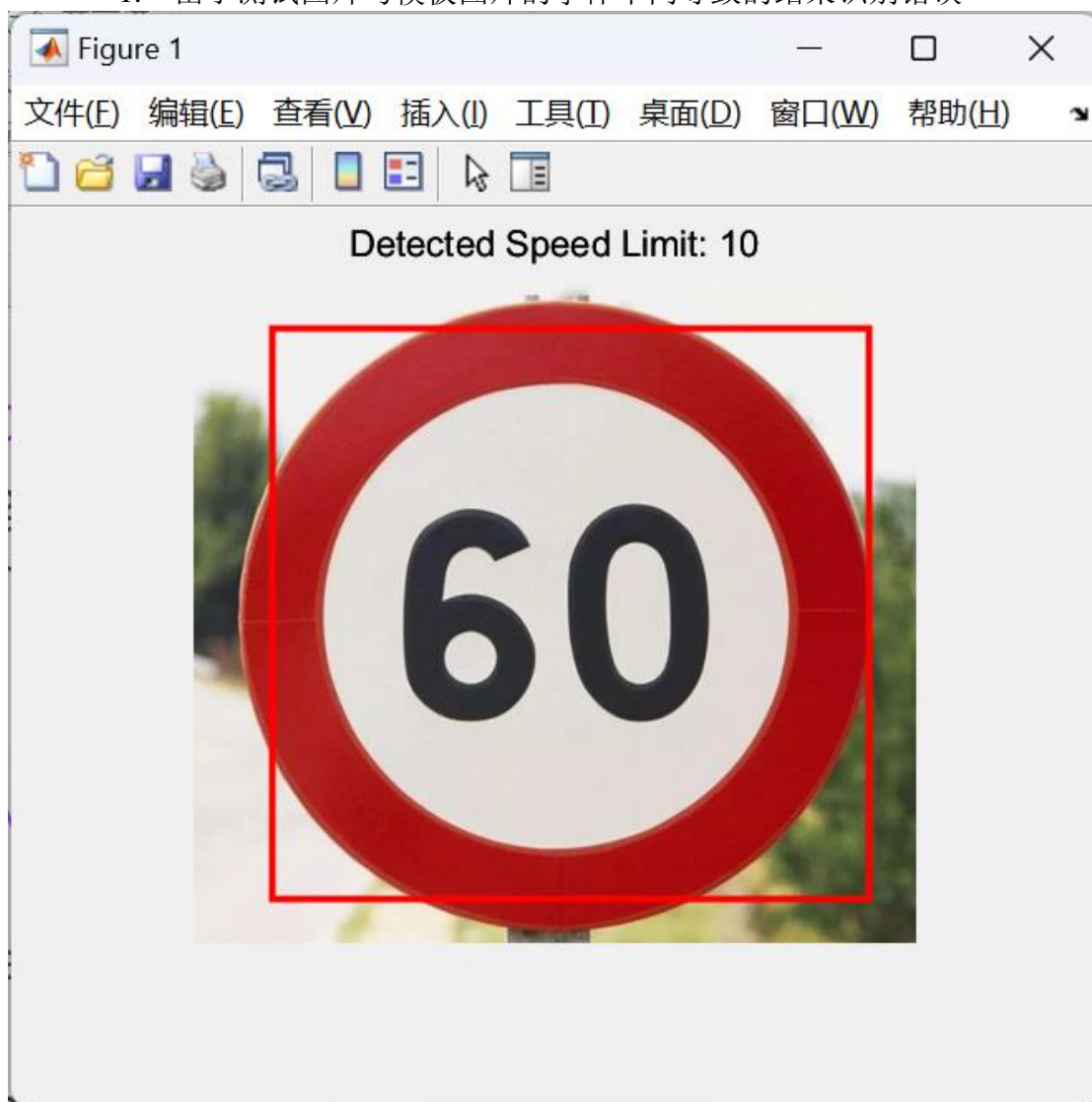
六. 对本实验问题的设计提出改进意见

(向教师反馈实验的设计，例如，修改实验任务使得更能体现能力、水平；增加实验任务等等；所有围绕实验的建议、意见。)

可以更加清晰明确实验目标，以便更好理解实验的目的和预期结果，例如给出一定的例子或者结果，这样可以帮助更好地理解实验的重要性。同时也增加实验任务问题的多样性，以便接触到不同类型的问题和挑战，更好的灵活使用程序域代码，培养解决问题的能力 and 创新思维。

但是，本代码仍然存在一些问题：

1. 由于测试图片与模板图片的字体不同导致的结果识别错误



改进方向：同一速度值增加多组模板图片并多次匹配。

2. 框选的位置不对



附件

附件 1. 速度识别 MATLAB 程序

```
% 读取图像
image = imread('D:\限速图片集\60-4.png');
grayImage = rgb2gray(image); % 转换为灰度图
bwImage = imbinarize(grayImage); % 二值化

% 准备多个模板
templates = {'D:\限速图片集\5-3.png', 'D:\限速图片集\10-1.png', 'D:\限速图片集\20-1.png', 'D:\限速图片集\40-2.png', 'D:\限速图片集\60-1.png', 'D:\限速图片集\100-2.png', 'D:\限速图片集\120-1.png'}; % 模板文件名列表
templateValues = [5, 10, 20, 40, 60, 100, 120]; % 与模板对应的速度值
numTemplates = length(templates);
highestCorr = 0;
detectedValue = 0;

% 循环处理每个模板
for i = 1:numTemplates
    % 读取并预处理模板
    template = imread(templates{i});
    template = imbinarize(rgb2gray(template)); % 转换为二值图

    % 确保模板尺寸小于原图像尺寸
    if any(size(template) > size(bwImage))
        scaleFactor = min(size(bwImage) ./ size(template));
        template = imresize(template, scaleFactor); % 调整模板尺寸
    end

    % 确保原图像大于模板图像
    if any(size(template) > size(bwImage))
        % 计算缩放因子
        scaleFactor = min(size(bwImage) ./ size(template));
        % 缩放模板图像
        template = imresize(template, scaleFactor);
    end

    % 模板匹配
    corrMap = normxcorr2(template, bwImage);
    maxCorr = max(corrMap(:));

    % 检查是否为最佳匹配
    if maxCorr > highestCorr
        highestCorr = maxCorr;
        detectedValue = templateValues(i);
        [yPeak, xPeak] = find(corrMap == maxCorr, 1); % 寻找最大相关性的位置

        % 计算模板在原图像中的位置
        yoff = yPeak - size(template, 1) + 1;
    end
end
```



```

        xoff = xPeak - size(template, 2) + 1;

        % 记录用于绘制矩形框的位置
        detectedPosition = [xoff, yoff, size(template, 2), size(template, 1)];
    end
end

% 显示检测结果
if detectedValue > 0
    figure; imshow(image); hold on;
    rectangle('Position', detectedPosition, 'EdgeColor', 'r', 'LineWidth', 2);
    title(['Detected Speed Limit: ', num2str(detectedValue)]);
else
    disp('No speed limit sign detected');
end

```

附件 2.程序运行结果整理

