《人工智能与模式识别》

实 验 报 告 书

班级： 计科2002

学号： 202801108

姓名： 李蔚

指导教师： 翟婷婷

2022-2023 学年 第 二 学期

**实验名称： 第一个模式识别算法—模板匹配算法**

**实验时间： 2023 年 03 月 17 日 第 5 周 星期 五**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 题目 | 二 | 三 | | | | 总分 |
| 1 | 2(a) | 2(b) | 2(c) |
| 得分 |  |  |  |  |  |  |

一、实验目的

1．理解最简单的一个模式识别算法—模板匹配算法。

2. 掌握模式识别的整个流程：模式采集、预处理、特征提取、分类器训练、分类器应用。

3. 在Matlab中编程实现模板匹配算法，并用于车牌数字的识别。

二、实验预习（预备知识的问题及回答）

1、 简述模板匹配算法的原理（用课件中的内容回答）。(1分)

**解答：**模板匹配法的基本原理是：为每个类别建立一个或多个标准模板，分类决策时将待识别的模式与每个类别的模板进行比对，根据与模板的匹配程度将模式划分到与其最相似的类别中。

三、实验内容（包含实验所用命令或相关程序源代码）

1.利用matlab的help命令查明以下函数的功能，**写明输入参数和输出参数的含义**。(每个0.5分，共4分)

A = imread(FILENAME,FMT)

B = imresize(A, [NUMROWS NUMCOLS])：

B = im2double(A)

v = norm(x)

[Y,I] = min(x)

NEWSTR = split(STR,DELIMITER)

T = num2str(X)

X = str2double(S)

**解答：**

（1）A = imread(FILENAME,FMT)：该函数用于读取图像文件并将其加载为一个矩阵，参数FILENAME是包含图像文件名的字符串，FMT是文件格式。输出参数A是一个矩阵，其中每个元素都对应于图像的像素值。

（2）B = imresize(A, [NUMROWS NUMCOLS])：该函数用于调整图像大小，参数A是输入图像矩阵，[NUMROWS NUMCOLS]是一个包含目标图像行数和列数的向量。输出参数B是一个与A大小不同的矩阵，它包含由输入图像缩放得到的像素值。

（3）B = im2double(A)：该函数用于将图像数据转换为双精度类型。输入参数A是要转换的图像数据。输出参数B是一个包含转换后像素值的矩阵。

（4）v = norm(x)：该函数返回输入向量x的范数，即向量的长度。输出参数v是一个标量。

（5）[Y,I] = min(x)：该函数返回输入向量x中的最小值及其索引。输出参数Y是最小值，I是最小值的位置。

（6）NEWSTR = split(STR,DELIMITER)：该函数将输入字符串STR根据DELIMITER分隔符进行拆分，并将结果存储在一个字符串数组中。输出参数NEWSTR是一个包含拆分后的字符串的字符串数组。

（7）T = num2str(X)：该函数将输入参数X转换为字符串类型，并将结果存储在输出变量T中。

（8）X = str2double(S)：该函数将输入字符串S转换为双精度浮点数类型。输出参数X是一个包含转换后值的变量。

2. 给定一个车牌数字的模板库，存放在car\_plates下的template文件夹中，要求利用模板匹配法对存放在car\_plates/test\_pic中的14张车牌数字图片进行识别，并用disp函数显示每张图片被识别为哪个数字以及该图片真正为哪个数字。

a. 依次读取模板库中的每张图片，将其归一化为相同的尺寸32\*32维的，然后将归一化后图片的unit8类型的像素矩阵转换为double类型的矩阵，将转换后的矩阵拉伸为1\*1024维的行向量，作为该图片的特征向量，并存储该图片的类标。对所有模板处理完后，最终会得到一个10\*1024维的特征矩阵template\_features和一个10\*1维的类标向量template\_labels。

**matlab代码： （4分）**

% 读取模板库中的图片并进行特征提取和标记

template\_dir = 'D:\car\_plates\car\_plates\templates';

template\_files = dir(fullfile(template\_dir, '\*.bmp'));

num\_templates = length(template\_files);

template\_features = zeros(num\_templates, 1024);

template\_labels = zeros(num\_templates, 1);

for i = 1:num\_templates

img = imread(fullfile(template\_dir, template\_files(i).name));

img = imresize(img, [32, 32]); % 将图像缩放为32\*32大小

img\_gray = im2gray(img); % 转换为灰度图像

img\_double = im2double(img\_gray); % 将像素值归一化到[0, 1]之间

img\_vector = img\_double(:)'; % 将矩阵拉伸为行向量

template\_features(i, :) = img\_vector; % 存储特征向量

template\_labels(i) = str2double(template\_files(i).name(1)); % 存储类标

end

b. 依次读取每一张待识别的图片，按照与处理模板图片相同的方法处理待识别图片，将待识别图片转换一个1\*1024维的特征向量x，然后开始进行模板匹配，即计算x与template\_features矩阵中每一个行向量的欧式距离，存储这10个距离值，找到其中最小的距离对应的模板，记为idx，则将待识别的图片预测为第idx个模板对应的类标。为了得到待识别图片的真实类标，可以对其文件名使用split函数进行分割，第一”.”前的字符对应的数字即为其真实类标。用disp函数输出每张图片的预测类标和真实类标，并统计正确分类的图片数，也用disp函数显示。

**matlab代码：（5分）**

% 读取待识别的图片并进行特征提取和预测

test\_dir = 'D:\car\_plates\car\_plates\test\_pic';

test\_files = dir(fullfile(test\_dir, '\*.bmp'));

num\_test = length(test\_files);

num\_correct = 0; % 统计正确分类的图片数

for i = 1:num\_test

img = imread(fullfile(test\_dir, test\_files(i).name));

img = imresize(img, [32, 32]); % 将图像缩放为32\*32大小

img\_gray = im2gray(img);% 转换为灰度图像

img\_double = im2double(img\_gray); % 将像素值归一化到[0, 1]之间

img\_vector = img\_double(:)'; % 将矩阵拉伸为行向量

% 计算欧式距离

distances = sqrt(sum((template\_features - repmat(img\_vector, num\_templates, 1)).^2, 2));

[~, idx] = min(distances); % 找到距离最小的模板

% 输出预测结果和真实结果

predict\_label = template\_labels(idx);

true\_label = str2double(test\_files(i).name(1));

disp(['图片', test\_files(i).name, '被识别为', num2str(predict\_label), '，真实值为', num2str(true\_label)]);

% 判断分类是否正确

if predict\_label == true\_label

num\_correct = num\_correct + 1;

end

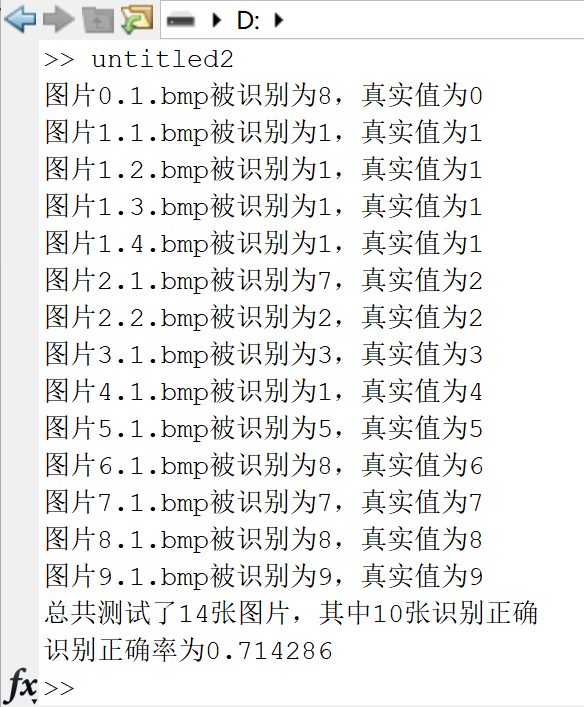
end

% 输出正确分类的图片数

disp(['总共测试了', num2str(num\_test), '张图片，其中', num2str(num\_correct), '张识别正确']);

disp(['识别正确率为', num2str(num\_correct/num\_test)]);

c. 前面两段脚本代码放在一起运行后的结果的截图。**（1分）**

****