



中国地质大学（武汉）

模式识别与机器学习实验报告

学 院： 自动化学院

课 程： 模式识别与机器学习

指导老师： 陆承达

学 号： 20201000128

班 级： 231202

姓 名： 刘瑾瑾

2023 年 11 月 9 日

目录

第一章 引言	1
1.1 实验目的	1
1.2 实验意义	1
第二章 实验内容	2
2.1 所选题目	2
2.2 实验内容	2
2.3 实验要求	2
第三章 实验方法	3
3.1 实验原理	3
3.2 实验流程	4
第四章 实验结果与分析	13
4.1 实验结果	13
4.2 分析与评价	15
第五章 总结	16
5.1 收获与体会	16
5.2 遇到的问题及解决方案	16

第一章 引言

1.1 实验目的

- 1、使学生进一步巩固模式识别基本的概念。
- 2、掌握相关实验的工具箱，能熟练使用工具箱中的各种功能。
- 3、增强学生应用 Matlab 编写程序及分析、解决实际问题的能力。

1.2 实验意义

模式识别可以帮助我们从大量的数据中发现规律和模式，从而提取有用的信息和知识。它可以应用于多个领域，如计算机视觉、语音识别、自然语言处理、生物信息学等。模式识别的研究不仅可以帮助我们理解和解释现象，还可以用于预测和决策。本实验主要包含人眼识别定位和手写数字识别两个部分：

研究人眼识别在理论和技术上都有重要的意义：一是可以推进对人类视觉系统本身的认识；二是可以满足人工智能应用的需要。采用人眼识别技术，建立自动人眼识别系统，用计算机实现对面脸图像的自动识别有着广阔的应用领域和诱人的应用前景。人眼识别可以识别和定位人类脸部、眼睛和其他生物特征，在生活中用于各种场景，如人脸识别登录、刷脸支付、监控系统和安全身份验证等。

研究手写体数字识别技术,将对于今后的信息发展有着重要的作用,在光学字符识别领域中,手写字符识别是一个非常重要的分支,在现实生活中有着广泛的应用.例如在银行票据识别中,根据手写体进行识别数字;在邮件分拣过程中,根据邮编进行识别实现自动分拣,省事又省力.此外,手写数字识别技术的发展对于推动人工智能技术的发展、提高数字化文档的自动化程度、增强人机交互的智能化水平等方面都具有重要的意义。

模式识别的研究意义和应用广泛，它可以帮助我们从大量的数据中发现规律和模式，从而提取有用的信息和知识。随着计算能力的提升和算法的改进，模式识别技术在各个领域都取得了重大突破，但仍然面临着一些挑战和问题。未来，我们可以进一步研究和改进模式识别算法，提高其准确性和鲁棒性，以更好地应用于实际场景中。

第二章 实验内容

2.1 所选题目

- 1、人眼识别定位
- 2、手写数字识别

2.2 实验内容

- 1、人眼识别定位
 - 1) 设计和建立基于机器视觉的模式识别系统。
 - 2) 基于基本的图像处理方法分割出包含人眼区域, 根据特征确定。
 - 3) 实现眼睛的定位识别。
 - 4) 输入一张图像, 能定位图像中的眼睛。
- 2、手写数字识别
 - 1) 设计和建立模式识别系统。(确定编程环境, 构建实验平台框架)
 - 2) 准备数据集。
 - 3) 实现数字 0~9 识别, 并统计正确率。

2.3 实验要求

- 1、人眼识别定位
 - 1) 利用 Matlab/Python 进行编程, 实现图像预处理。
 - 2) 了解熟悉图像处理相关函数, 对各个模块功能有一定的了解
 - 3) 通过实验加深对《模式识别与机器学习》理论课堂所学知识的理解。
 - 4) 实验前做好相关的准备(环境配置)
- 2、手写数字识别
 - 1) 利用 Matlab/Python 进行编程, 实现数字识别。
 - 2) 介绍所使用的算法特点。
 - 3) 了解算法实现的过程, 熟悉训练数据和测试数据的功能。

第三章 实验方法

3.1 实验原理

1、人眼识别定位

1) 人脸检测原理

经典的人脸检测算法流程是这样的：用大量的人脸和非人脸样本图像进行训练，得到一个解决 2 类分类问题的分类器，也称为人脸检测模板。这个分类器接受固定大小的输入图片，判断这个输入图片是否为人脸，即解决是与否的问题。

由于人脸可能出现在图像的任何位置，在检测时用固定大小的窗口对图像从上到下、从左到右扫描，判断窗口里的子图像是否为人脸，这称为滑动窗口技术（sliding window）。为了检测不同大小的人脸，还需要对图像进行放大或者缩小构造图像金字塔，对每张缩放后的图像都用上面的方法进行扫描。由于采用了滑动窗口扫描技术，并且要对图像进行反复缩放然后扫描，因此整个检测过程会非常耗时。由于一个人脸附件可能会检测出多个候选位置框，还需要将检测结果进行合并去重，这称为非极大值抑制（NMS）。

2) 人眼识别原理

人眼定位原理是通过视网膜上的视觉感受器（视杆细胞和视锥细胞）接收光信号，将其转化为神经信号并传递到大脑的视觉皮层进行解析和处理。在此过程中，光信号的强度和颜色被分析，并与周围环境进行比较，以确定物体在视野中的位置和方向。

在 Matlab 中，可以通过图像处理和计算机视觉技术模拟人眼定位的原理。具体来说，可以使用 Matlab 中的图像处理函数对图像进行预处理、滤波和特征提取，然后使用计算机视觉算法进行目标检测、跟踪和定位。例如，可以使用 Matlab 中的视觉工具箱中的函数来实现基于特征点匹配的物体定位和跟踪，或者使用深度学习技术来训练神经网络实现图像分类和目标检测。

2、手写数字识别

SVM 的工作原理是将输入数据映射到高维空间中，找到一个最优的超平面来分割不同类别的数据点。最优的超平面是指能够最大化不同类别之间的间隔，同时最小化分类错误的数量。SVM 通过对训练数据进行支持向量的选择来确定最优的超平面，这些支持向量是距离最优超平面最近的数据点。在测试数据上，SVM 根据输入数据的特征向量与最优超平面的位置关系，预测输入数据所属的类别。

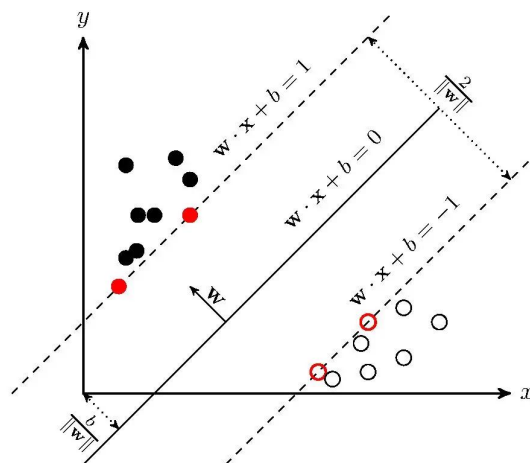


图 1 SVM 原理图

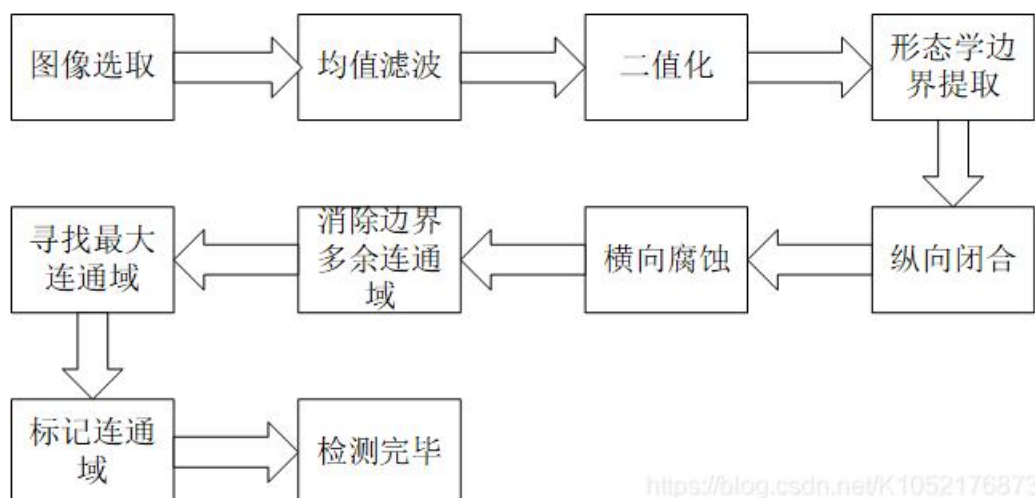
SVM(支持向量机)是一种经典的机器学习算法，它可以用于分类和回归任务。在手写数字识别中，SVM 可以用于分类任务，将输入的手写数字图像分为不同的数字类别。在 MATLAB 中，可以使用 SVM 工具箱中的函数来实现手写数字识别。具体步骤如下：

- 1) 数据预处理:将手写数字图像转换为向量形式，并进行归一化处理。
- 2) 特征提取:使用 HOG(方向梯度直方图)算法提取手写数字图像的特征向量。
- 3) 训练模型:使用 SVM 训练函数训练分类器模型。
- 4) 测试模型:使用测试数据对训练好的模型进行测试，计算预测准确率。

3.2 实验流程

1、人眼识别定位

1) 人脸检测流程



<https://blog.csdn.net/K1052176873>

代码如下：

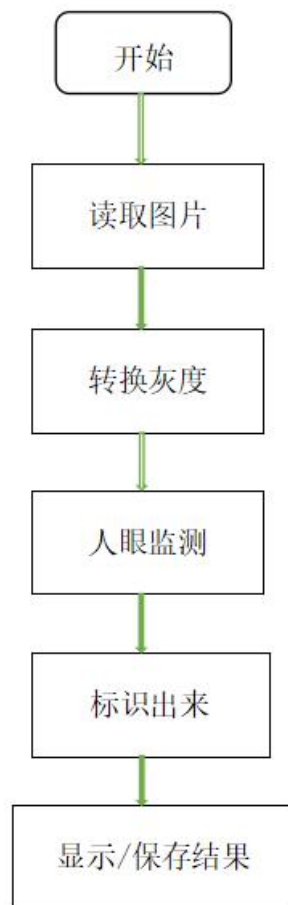
```
clc;
clear;
close all;
rgb = imread('liu.jpg');
I = rgb2gray(rgb);
[n1,n2] = size(I);
%灰度图
tic
h = ones(9)/81;
I = uint8(conv2(I,h));%线性均值滤波
BW = imbinarize(I);%二值化
B = ones(21);%结构元素
BW = -imerode(BW,B) + BW;%边界提取
BW = bwmorph(BW,'thicken');%加粗边界
BW = not(bwareaopen(not(BW), 300));%填补空洞
%进行形态学运算
B = strel('line',50,90);
BW = imdilate(BW,B);%扩展灰度图
BW = imerode(BW,B);%侵蚀图像
B = strel('line',10,0);
BW = imerode(BW,B);
%最小化背景
%细分
div = 10;
r = floor(n1/div);%分成10块行
c = floor(n2/div);%分成10块列
x1 = 1;x2 = r;%对应行初始化
s = r*c;%块面积
%判断人脸是否处于图片四周，如果不是就全部弄黑
%figure
for i=1:div
    y1 = 1;y2 = c;%对应列初始化
    for j=1:div
```

```
loc = find(BW(x1:x2,y1:y2)==0); %统计这一块黑色像素的位置
num = length(loc);
rate = num*100/s; %统计黑色像素占比
if (y2<=0.2*div*c||y2>=0.8*div*c)|| (x1<=r||x2>=r*div)
    if rate <=100
        BW(x1:x2,y1:y2) = 0;
    end
    %imshow(BW)
else
    if rate <=25
        BW(x1:x2,y1:y2) = 1;
    end
    %imshow(BW)
end %下一列
y1 = y1 + c;
y2 = y2 + c;
end %下一行
x1 = x1 + r;
x2 = x2 + r;
end
%figure
%subplot(1,2,1)
%imshow(BW)
%title('最终处理')
L = bwlabel(BW,8); %利用 belabel 函数对 8 连通域区间进行标号
BB = regionprops(L, 'BoundingBox'); %得到矩形框，框柱每一个连通域
BB = cell2mat(struct2cell(BB));
[s1,s2] = size(BB);
BB = reshape(BB,4,s1*s2/4)';
pickshape = BB(:,3)./BB(:,4); %
shapeind = BB(0.3<pickshape&pickshape<3,:); %筛选掉尺寸比例不合格
[~,arealind] = max(shapeind(:,3).*shapeind(:,4));
%subplot(1,2,2)
imshow(rgb)
```



```
hold on
rectangle('Position',[shapeind(arealind,1),shapeind(arealind,2),s
hapeind(arealind,3),shapeind(arealind,3)],...
'EdgeColor','g','Linewidth',2)
title('人脸检测')
toc
```

2) 人眼识别流程



代码如下：

```
detector=vision.CascadeObjectDetector('LeftEyeCART');
input_image=imread('liu.jpg');
figure();imshow(input_image);title('原图');
gauss_image=imgaussfilt(input_image,1);
eyes=step(detector,gauss_image);
size_eyes = size(eyes);%眼睛
hold on;
```

```
for i =1:size_eyes(1)
eyes_final=eyes(i,1:2);%x,y
image = imcrop(gauss_image,eyes(i,:));
rectangle('Position',eyes(i,:), 'edgeColor', 'b');

image = rgb2gray(image);
image = im2double(image);
[grad_x,grad_y] = gradient(image);%计算梯度

g = grad_x.*grad_x + grad_y.*grad_y;
g = sqrt(g);%眼睛区域的梯度图

std_g = std(g(:));%计算标准差
mean_g = mean(g(:));%计算平均值
%其实这一步取得是< mean_g - 0.5*std_g 和> mean_g + 0.5*std_g 部分
g_1 = g;
g_1(g_1 < mean_g + 0.5*std_g) = 0;%梯度图中小于均值+0.5*标准差的设置
为0
g_2 = g;
g_2(g_2 > mean_g - 0.5*std_g) = 0;%梯度图中大于均值-0.5 标准差的设置
为0
g = g_1 + g_2;
temp_g = g;
w = 1-image;
temp_g(temp_g ~= 0) = 1;%将不等于 0 的设置为 1 这该图只有 0 和 1 两个值
grad_x = grad_x.*temp_g;%与梯度相成
grad_y = grad_y.*temp_g;

size_img = size(image);
Dx = repmat(1:size_img(2),size_img(1),1);%每一行都是 1-44
Dy = repmat(transpose(1:size_img(1)),1,size_img(2));%转置
C_max = 0;
X = 0;
Y = 0;
```

%为了得出眼睛的中心位置

```
for x = 1:size_img(2)
    for y = 1:size_img(1)
        D_x = Dx - x;
        D_y = Dy - y;

        D_t = D_x.*D_x + D_y.*D_y;
        E = sqrt(D_t);

        D_x = D_x./E;
        D_y = D_y./E;

        C_x = D_x.*grad_x;
        C_y = D_y.*grad_y;

        C = w(y,x)*(C_x + C_y);
        C(C < 0) = 0;
        ind = find(isnan(C));
        C(ind)=0;

        total_C = sum(sum(C));

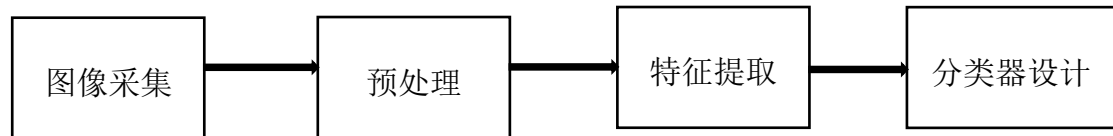
        if(total_C > C_max)
            C_max = total_C;
            X = x;
            Y = y;
        end
    end
end

plot(X+eyes_final(1),Y+eyes_final(2),'g+', 'MarkerSize',10);
p(i,1)=X+eyes_final(1);
p(i,2)=Y+eyes_final(2);
end
```

2、手写数字识别

划分数据集，70%作为训练集，30%作为测试集，进行 SVM 分类模型训练：

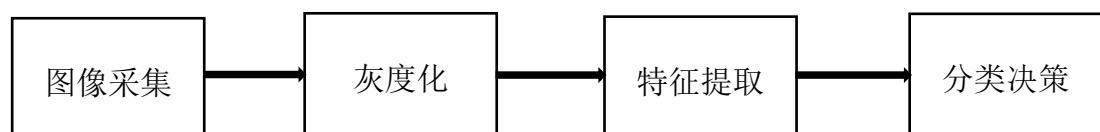
1)训练过程



代码如下：

```
clear,clc,close all;
k=1;
for m=0:9
    for n=1:11
        f=imread(strcat('nums\ ',int2str(m),'\'',int2str(n),'.bmp'));
        f=picPretreatment(f);
        xx=double(f(:));
        P(k,:)=xx';
        T(k,1)=m;
        k=k+1;
    end
end
t=templateSVM('Standardize',1);
net=fitcecoc(P,T,'Learners',t);%多类分类器
save mynet net;
function x=picPretreatment(im)
bw=im2bw(im);%二值图像
[r,c]=find(bw);
x=bw(min(r):max(r),min(c):max(c));%找到图像区域
x=imresize(x,[16,16]);%调整图像长度和宽度
```

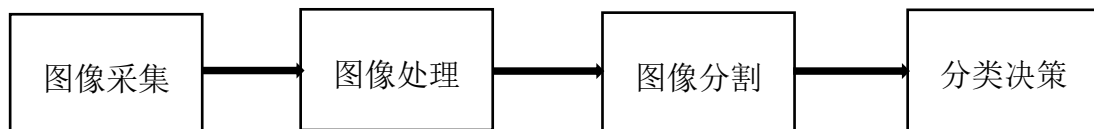
2)测试过程：



代码如下：

```
k=0;j=0;
for m=0:9
    for n=12:15
        f=imread(strcat('nums\ ',int2str(m),'\'',int2str(n),'.bmp'));
        f=picPretreatment(f);
        fx=double(f(:));
        wo=predict(net,fx');
        if wo(1,1)==m
            k=k+1;
        end
        j=j+1;
    end
end
acc='准确率为: '
accuracy=k/j
```

3) 应用过程:



```
f=imread('num1.jpg');
bw=~im2bw(f);%二进制取反图像
se=strel('line',200,0);
bw_dilate=imdilate(bw,se);%扩展灰度图像
bw_dilate_erode=imerode(bw_dilate,strel('line',1000,0));%侵蚀图像
bw_re=imreconstruct(bw_dilate_erode,bw_dilate);%形态学重构
result=imreconstruct(bw_re,bw);[r,c]=find(result);
nextresult=result(min(r):max(r),min(c):max(c));
[h,w]=size(nextresult);
hs=sum(nextresult);
a=1;b=1;i=1;
splitfs={};points=[];
```

```
figure
while (a<w)
    while (hs(a)==0&&a<w)
        a=a+1;
    end
    b=a;
    while (hs(b)>0&&b<w)
        b=b+1;
    end
    if (b-a>2)
        hresult=nextresult(:,a:b);
        [r,c]=find(hresult);
        result2=hresult(min(r):max(r),:);
        m=min(r);n=max(r);
        xi=hresult(m:n,:);
        splitfs{i}=xi;
        points=[points;m,n,a,b];
        subplot(3,6,i),imshow(result2);
        i=i+1;
    end
    a=b;
end
xs=[splitfs,points];
fonts='0123456789';
picSize=[20,50];
endresult='手写数字为: ';
load mynet;
for m_18=1:size(xs,2)-1
    g=xs{m_18};
    g=double(picPretreatment(g));
    lastresult=predict(net,g(:)');
    endresult=strcat(endresult,string(lastresult));
end
endresult
```

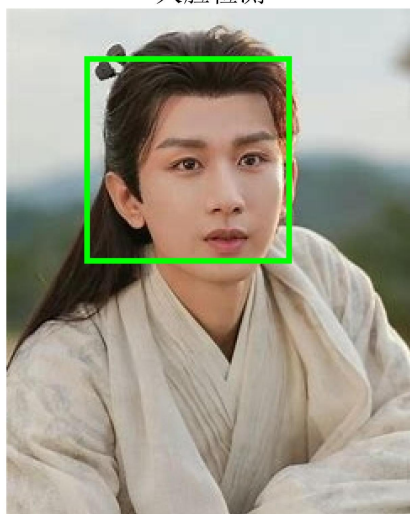
第四章 实验结果与分析

4.1 实验结果

1、人眼识别定位

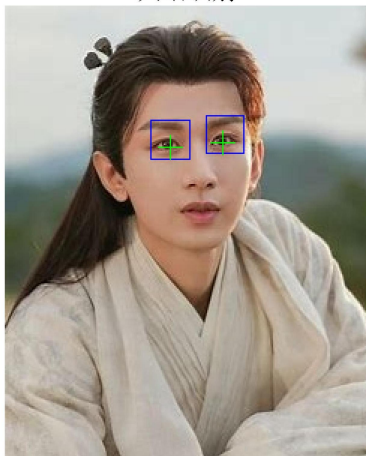
1) 人脸检测定位

人脸检测

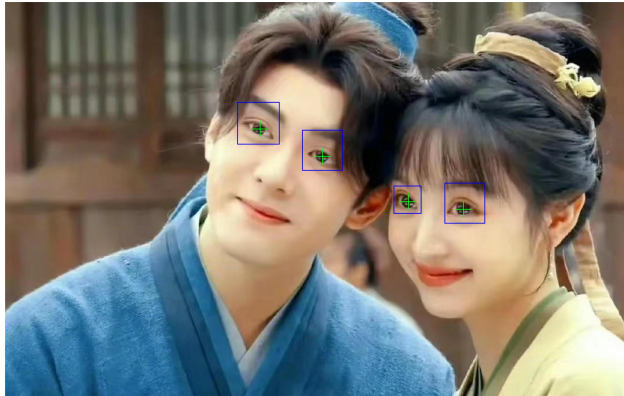


2) 人眼识别定位（单人、多人）

人眼识别



人眼识别



2、手写数字识别

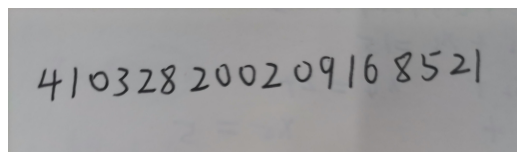
1) 准确率计算

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

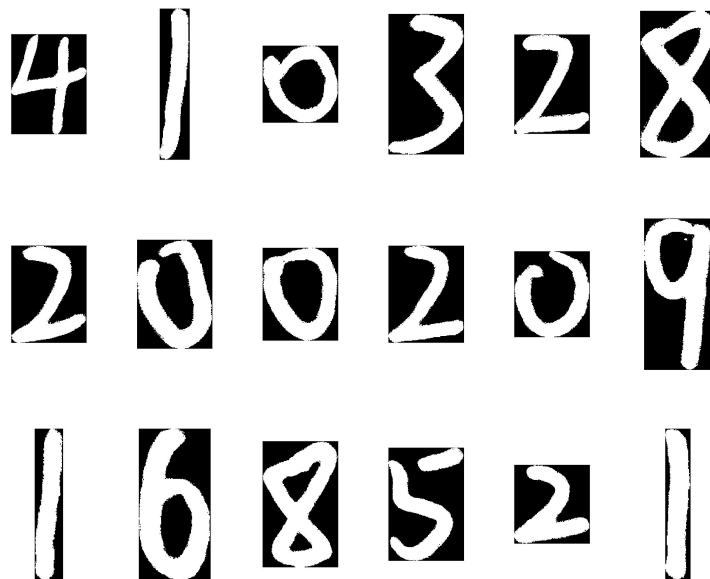
导入测试集计算，得到准确率如下：

```
accuracy =  
  
0.8250
```

2) 导入手写数字图片



图像处理分割结果如下：



识别结果如下：

```
endresult =  
  
"手写数字为:410328260209868521"
```


4.2 分析与评价

1、人眼识别定位

1) 人脸检测定位

通过使用本次实验的代码，当输入一张较为清晰的图片时，可以明确地判断出照片中的人脸。但代码也有不足的地方，当人脸在图片中范围较小或者图片其他部分亮度较高，就会出现识别识别出错的情况，后续还可以使用其他算法进一步优化。

2) 人眼识别定位

借助于 **Matlab** 图像处理函数实现人眼定位可以达到较高的准确率，但是对于有遮挡（如眼镜、头发）的图像仍然有较大的误差，鲁棒性不高，可以对算法进行进一步的优化，实现眼睛和眼镜的双向识别，后续也可以借助于深度学习，张量补全等方法解决人眼遮挡的问题。对于多张人脸的人眼识别，由于照片中有一部分人眼存在角度倾斜，容易识别不出，此后可以进一步优化。

2、手写数字识别

通过 **SVM** 分类算法可以有效实现手写数字的识别，但是由于训练数据较少，并且个人的书写习惯不同，所以准确率仅达 82.5%，后续可以加大训练数据，修改模型参数并且加入个人习惯的手写数字图像，进一步提升优化。

第五章 总结

5.1 收获与体会

通过本次实验，我学习到了模式识别在生活中的一些具体运用，并且通过人眼识别定位和手写数字识别的具体实验操作，学习到了关于图像识别的一些基础方法，如灰度图转换、图像二值化、图像分割、分类器模型的设定等，加深了对课堂知识的理解和体会，对于模式识别在图像识别领域的具体应用有了进一步的了解。

5.2 遇到的问题及解决方案

1、灰度图等级设置问题

灰度值指的是将灰度对象转变成`rgb`格式的时候所对应的每一个对象的颜色值数。灰度的等级则是指的将白色和黑色按照对数的关系来进行若干的分级，通过黑白灰度的扫描仪来测试生成的图像，通常都是以灰度来显示的。灰度级越高代表图片的色彩更加鲜艳，看上去更加丰富真实。

刚开始时，本次实验所用的代码对照片中的人眼识别度较低且混乱，查阅资料发现是因为灰度图等级较低所致，提高灰度图等级后再次实验，发现人眼识别的精度有了明显的提高。

2、人眼定位问题

在人眼识别时，容易出现识别不出的问题，发现是照片中可以更换人眼范围更大的图片进行识别。对于多张人脸的人眼识别，由于照片中有一部分人眼是侧面照，容易识别不出，更换人眼正面照即可