

上海交通大学

SHANGHAI JIAO TONG UNIVERSITY



数据挖掘大作业

Project 3

指导老师：王磊

系 别：工业工程与管理系

小组成员：陈梦月 1140209139

陈 震 1140209140

邓业雯 1140209142

刘 晔 1140209144

徐扬斌 1140209147

一、问题介绍

1.1 目的要求

- Propose the framework and methods to mark the choice questions in the images of test papers.
- ✓ (a) Propose the methods to recognize the answers, i.e., the letters for single choice questions, combinations of letters for multiple choice questions in the images, and the questions numbers and brackets (Hint: You can learn and call character recognition codes developed by others).
- ✓ (b) Propose a cognitive model to recognize the various patterns of the answers in the images.

提出一种可以给给定试卷图像中选择题评分的框架和方法，具体如下：

- 1、提出识别答案的方法，如图片中单选题的字母，多选题的字母组合以及题号和括号；
- 2、提出一个认知模型来识别图片中不同模式的答案。

1.2 研究思路

为了很好地识别图片中的字母，我们首先要将图片进行预处理，将其中的字母分割出来，即要将图片切割成单个字母，然后将切割出的字母与手写字母图片模板进行匹配识别，识别出的结果与正确答案匹配评分，得出相应的分数。

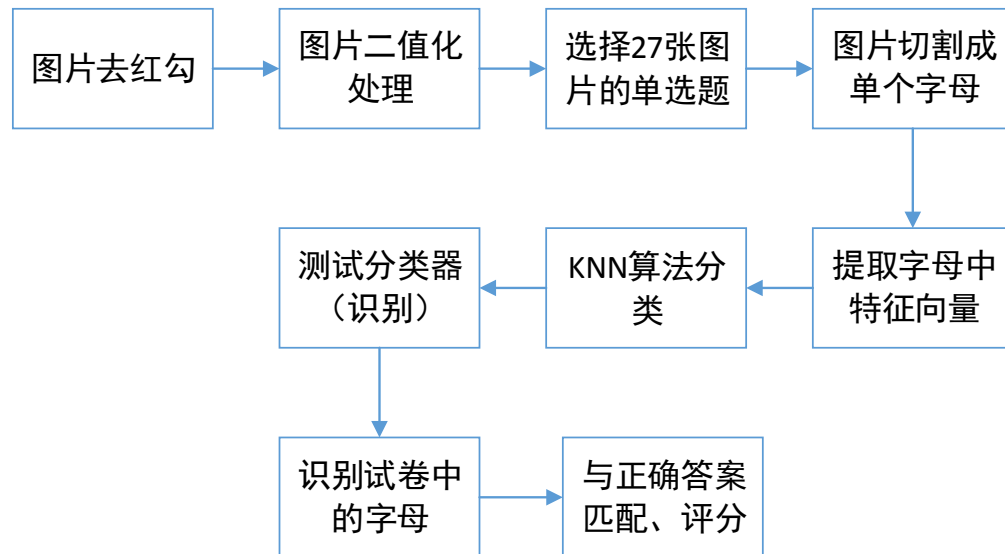
1.3 研究步骤

1. 图片切割成单个字母
 - 1) 图片去红勾
 - 2) 图片二值化，得到黑白图片
 - 3) 图片切割，得到单个字母
2. 图片识别

- 1) 提取字母中的特征向量
- 2) KNN 算法分类
- 3) 测试分类器（识别测试集）

3. 与正确答案匹配评分

- 1) 识别试卷中的字母
- 2) 与正确答案匹配评分



二、方法及应用

2.1 图片切割

2.1.1 图片去红勾

由于给定的试卷答案的图片中有红色的打分笔迹（如图 1），为了便于识别图中选项答案中的字母，因此第一步是先取出图片中的红色的对勾和分数，得到去红勾后的图片如图 2 所示。图片去红勾的 MATLAB 程序如图 3.

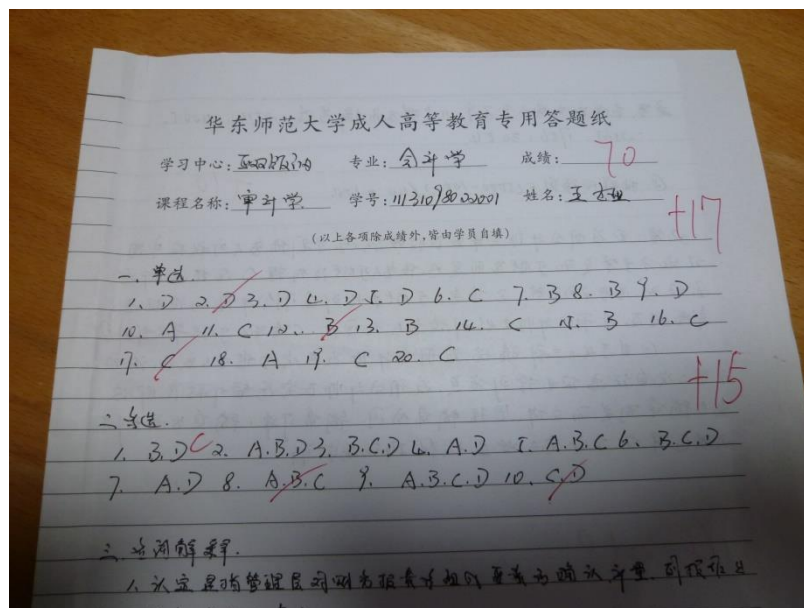


图 1.试卷原图

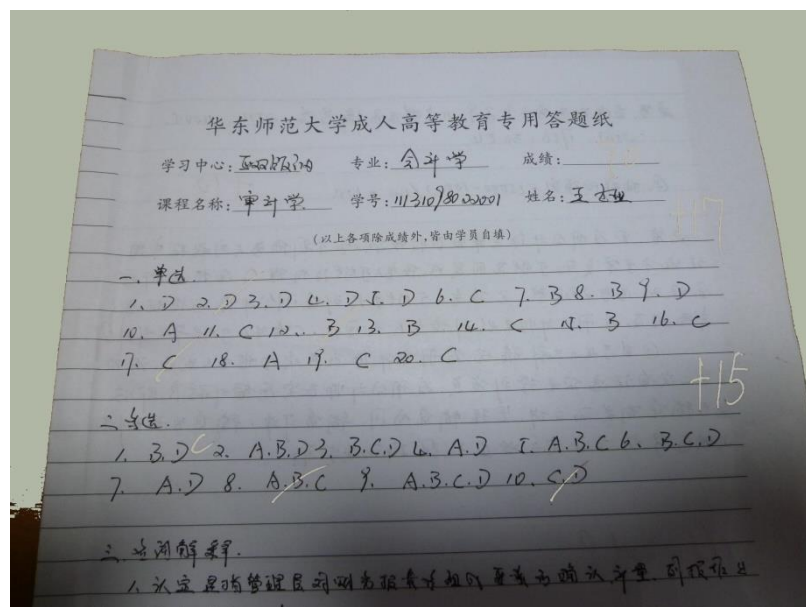


图 2.去红勾后的图片

```

1 % 去除红色的勾
2 file_path = 'H:\Data\'; % 图像文件夹路径
3 img_path_list = dir(strcat(file_path, '*.jpg')); % 获取该文件夹中所有jpg格式的图像
4 img_num = length(img_path_list); % 获取图像总数量
5 for k = 1:img_num % 逐一读取图像
6     image_name = img_path_list(k).name; % 图像名
7     I = imread(strcat(file_path, image_name));
8     [x,y,z]=size(I);
9     for i = 1:x
10        for j = 1:y
11            if I(i,j,1) - I(i,j,2) > 25 && I(i,j,1) - I(i,j,3) > 5
12                I(i,j,1)=176;
13                I(i,j,2)=183;
14                I(i,j,3)=165;
15            end
16        end
17    end
18    imwrite(I, strcat('H:\Img_1\', image_name));
19 end

```

图 3. 图片去红勾的 MATLAB 程序

2.1.2 二值化，得到黑白图片

对图 2 中去红勾后的图片进行二值化处理。图像的二值化，就是将图像上的像素点的灰度值设置为 0 或 1，也就是将整个图像呈现出明显的只有黑和白的视觉效果。得到二值化后的图像如图 4 所示。图片二值化的 MATLAB 程序如图 5 所示。

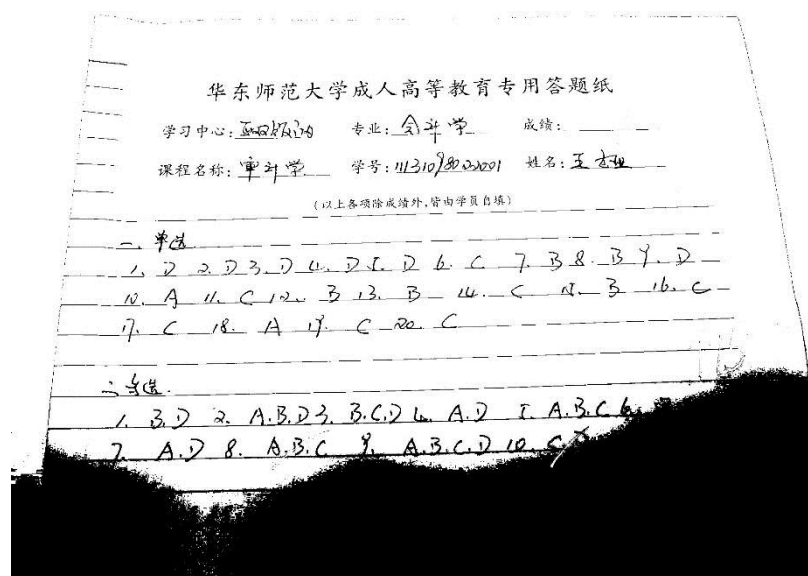


图 4. 二值化后的图片

```

1      %图片的二值化
2      I=imread('1.jpg');
3      I=rgb2gray(I);%先将图片转化为灰度图
4      thresh=graythresh(I);%自动确定二值化阈值
5      I2=im2bw(I,thresh);%对图像二值化，0代表黑色，1代表白色
6      [x,y]=size(I2);
7      for i=1:x
8          for j=1:y
9              I2(i,j)=1;
10         end
11     end

```

图 5.二值化的 MATLAB 程序

2.1.3 选取 27 张图片的单选题，切割成单个字母

如图 4 所示，上述得到的二值化后的图像存在较大的阴影面积，进行切割和识别比较困难，因此选择效果比较好的图片进行识别；由于多选题部分多被阴影覆盖，故针对单选题进行分割和识别操作。本文选择题目给出的 27 张图片中的单选题进行分析处理。

首先，对于选定的二值化处理后的 27 张图片，以图 4 为例，分割出单选题部分，得到如图 6 所示图片。然后，对图 6 图片切割图片得到单个字母，结果如图 7 所示。切割单个字母的 MATLAB 程序如图 8 所示。

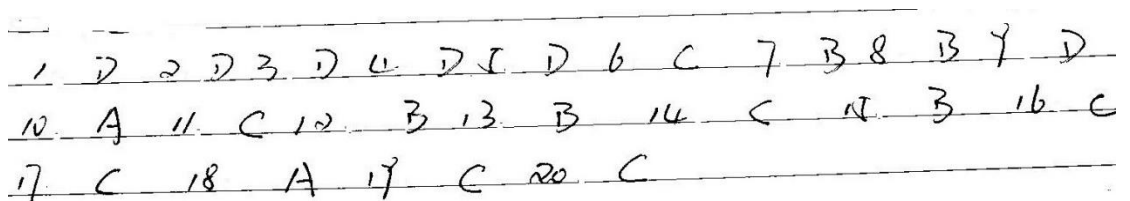


图 6.选择单选题部分



图 7.切割字母结果图

```

1      %%图片切割成单个字母或数字
2      clc;clear;
3      file_path='H:\handwritten\Entire_D\';%图片文件夹路径
4      Pic_list=dir(strcat(file_path,'*.jpg'));%获取该文件夹中所有jpg格式的图像
5      Pic_name=Pic_list(4).name;%获取第一张图片的图片名
6      I=imread(strcat(file_path,Pic_name));%读取图片
7      I=rgb2gray(I);%将图片转成灰度图
8      thresh=graythresh(I);%自动确定二值化阈值
9      I=im2bw(I,thresh);%对图像二值化，0代表黑色，1代表白色
10     [h0,w0]=size(I);%获取图片大小
11     name_net=Pic_name(1:end-4);|
12
13     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
14     %%先将图片切割成行
15     %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
16
17     %计算每一行黑色像素点的个数%%
18     counter_1=[];
19     for i=1:1:h0
20         counter_1(i)=0;
21         for j=1:1:w0
22             if I(i,j)==0
23                 counter_1(i)=counter_1(i)+1;
24             end
25         end
26     end
27     %找出空白行作为行切割的边界%%
28     row=[];
29     j=1;
30     row(1)=1;

```

```

31 - for i=1:1:h0
32 -     if counter_1(i)<3%定义黑色像素点的个数小于3，则为行边界，即空白行
33 -         j=j+1;
34 -         row(j)=i;%记录空白行的行数
35 -     end
36 - end
37 - row(j+1)=h0;
38 - %行切割%%
39 - num_1= 0;
40 - for k=1:1:j
41 -     if row(k+1)-row(k)>40 %按照空白行边界切割图片，间距大于40则为有效边界
42 -         num_1=num_1+1;
43 -         a=row(k);
44 -         b=row(k+1);
45 -         eval(['R_',num2str(num_1),'=I(a:b,1:w0)']);%保存切割出来的每一行
46 -     end
47 - end
48 -
49 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
50 - %将生成的每一行图片切割成列，即生成单个字母或数字
51 - %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
52 - for m=1:1:num_1
53 -     R=eval(['R_',num2str(m),' :']);
54 -     [h1,w1]=size(R);%获取行图片的大小
55 -     counter_2=[];
56 -     %计算每一列的黑色像素点个数%%
57 -     for y=1:1:w1
58 -         counter_2(y)=0;
59 -         for x=1:1:h1
60 -             if R(x,y)==0
61 -                 counter_2(y)=counter_2(y)+1;
62 -             end
63 -         end
64 -     end
65 -     %找出空白列作为列切割的边界%%
66 -     col=[];
67 -     n=1;
68 -     for y=1:1:w1
69 -         if counter_2(y)==0
70 -             col(n)=y;%记录空白列的列数
71 -             n=n+1;
72 -         end
73 -     end
74 -     col(n)=w1;
75 -     %列切割%%
76 -     num_2=0;
77 -     for s=2:1:n
78 -         if col(s)-col(s-1)>14 %按照空白列切割行图片，间距大于15则为有效边界
79 -             num_2=num_2+1;
80 -             a1=col(s-1);
81 -             b1=col(s);
82 -             str=[name_net,'_',int2str(m),'_',int2str(num_2)];
83 -             imwrite(imresize(R(1:h1,a1:b1),[40,40]),strcat('H:\handwritten\Single_D\'',name_net,'\'',str,'.jpg'));
84 -             %储存图片，大小为40*40
85 -         end
86 -     end
87 - end

```

图 8.切割单个字母的 MATLAB 程序

2.2 手写字母识别

2.2.1 KNN 算法介绍

k-近邻算法（KNN）采用不同特征值之间的距离方法进行分类，其工作原理是存在一个样本数据集合，也称作训练样本集，并且样本集中每个数据都存在标签，即我们知道样本集中每一数据与所属分类的对应关系。输入没有标签的新数据后，将新数据的每个特征与样本集中数据对应的特征进行比较，然后算法提取样本集中特征最相似数据（最近邻）的分类标签。一般来说，我们只选择样本数据集中前 k 个最相似的数据，这就是 k-近邻算法中 k 的出处。最后，选择 k 个最相似数据中出现次数最多的分类，作为新数据的分类。

k-近邻算法的一般流程：

- (1) 收集数据：可以使用任何方法。
- (2) 准备数据：距离计算所需要的数值，最好是结构化的数据格式。
- (3) 分析数据：可以使用任何方法。
- (4) 训练算法：此步骤不适于 k-近邻算法。
- (5) 测试算法：计算错误率。
- (6) 使用算法：首先需要输入样本数据和结构化的输出结果，然后运行 k-近邻算法判定输入数据属于哪个分类，最后应用对计算出的分类执行后续的处理。

2.2.2 提取特征，训练集→切割→2000 个字模

首先，准备训练集。为选用合适的数据结构存储训练数据和测试元组，我们分别选择不同的实验者在空白纸上写下 A、B、C、D 四个字符，通过对每张纸进行分割处理，一共得到 2000 个字模，我们将这 2000 字模作为初步的训练及测试集。

对于我们制作的字模，进行切割操作，得到切割后的图片，如图 9 所示。每个切割后的图片包含 $40 \times 40 = 1600$ 个像素点，我们以 $4 \times 4 = 16$ 个像素点大小的图片为一个基本单位，将每张图片划分为 $10 \times 10 = 100$ 个单位图片。根据每

单位图片中黑色像素点所占的比例得到一个 10×10 的矩阵，将该矩阵从第一行到第十行中的元素排列得到一个 1×100 的向量，且向量中的元素值均处于 0-1 之间，该向量可以衡量不同字符（A/B/C/D）的特征。在该向量的最后增加一位，即我们将该向量中第 101 个数值定义为 ABCD 的类别，并分别用 1234 表示，即 1 代表 A，2 代表 B，3 代表 C，4 代表 D，由此得到的 1×101 的的向量表示不同字模的特征。特征采集的 MATLAB 程序如图 10 所示。





图 9.切割字模

```

1      %特征采集
2      function feature=feature_pic(img, img_label)
3      feature=[];
4      for i=1:1:10
5          for j=1:1:10
6              Atemp=sum(img((i*4-3):(i*4), (j*4-3):(j*4)));
7              %分割成4x4小方格，计算小方格内黑色像素点占有的比率
8              feature((i-1)*10+j)=(16-sum(Atemp))/16;
9          end
10     end
11     feature(101)=img_label;
12 end

1      %字母标签的提取, 字母A的标签为1, B的标签为2, C的标签为3, D的标签为4
2      function Category=label(x)
3      if x=='A'
4          Category=1;
5      elseif x=='B'
6          Category=2;
7      elseif x=='C'
8          Category=3;
9      else
10         Category=4;
11     end
12 end

```

图 10.特征提取的 MATLAB 程序

2.2.3 KNN 算法分类

KNN 分类算法的 MATLAB 程序如图 11 所示。

实施 KNN 算法：

对未知类别属性的数据集中的每个点依次执行以下操作：

- (1) 计算已知类别数据集中的点与当前点之间的距离；
- (2) 按照距离递增次序排序；
- (3) 选取与当前点距离最小的 k 个点；
- (4) 确定前 k 个点所在类别的出现频率；
- (5) 返回前 k 个点出现频率最高的类别，作为当前点的预测分类。

本实验中，我们使用的距离是欧氏距离，即两个向量点 x_A 与 x_B 之间的距离是

$$d = \sqrt{(x_{A_0} - x_{B_0})^2 + (x_{A_1} - x_{B_1})^2}$$

```
1      %KNN-K邻近算法分类
2      function Class_label=KNN_classify(test_ex,train_set,k)
3      [m,n]=size(train_set);
4      distances=[];
5      for i=1:1:m
6          diff=test_ex(1:100)-train_set(i,1:100);
7          distances(i)=sqrt(sum((diff.^2)));
8      end
9      [A,B]=sort(distances);
10     count=zeros(1,4);
11     subscript=B(1:k);
12     for j=1:1:k
13         r=subscript(j);
14         train_label=train_set(r,101);
15         if train_label==1
16             count(1)=count(1)+1;
17         elseif train_label==2
18             count(2)=count(2)+1;
19         elseif train_label==3
20             count(3)=count(3)+1;
21         else
22             count(4)=count(4)+1;
23         end
24     end
25     [C,D]=sort(count);
26     Class_label=D(4);
27     end
```

图 11. KNN 算法分类

2.2.4 测试分类器（测试集识别分类）

为检验该分类器的识别的准确率，我们从 2000 字模中选出 200（ABCD 分别为 50 个）字模作为测试集，其余 1800 个字模作为训练集。对每个测试集中的字模，我们分别计算其特征向量与 1800 个训练集中的特征向量之间的距离，并将距离按从小到大的顺序排序，经测试，计算前 100 个点中 ABCD 出现的频率，以出现频率最高的字符作为输出对象。最后得到分类准确率为 96%，证明该分类器是可用的。

手写字母识别的 MATLAB 程序如图 12 所示。

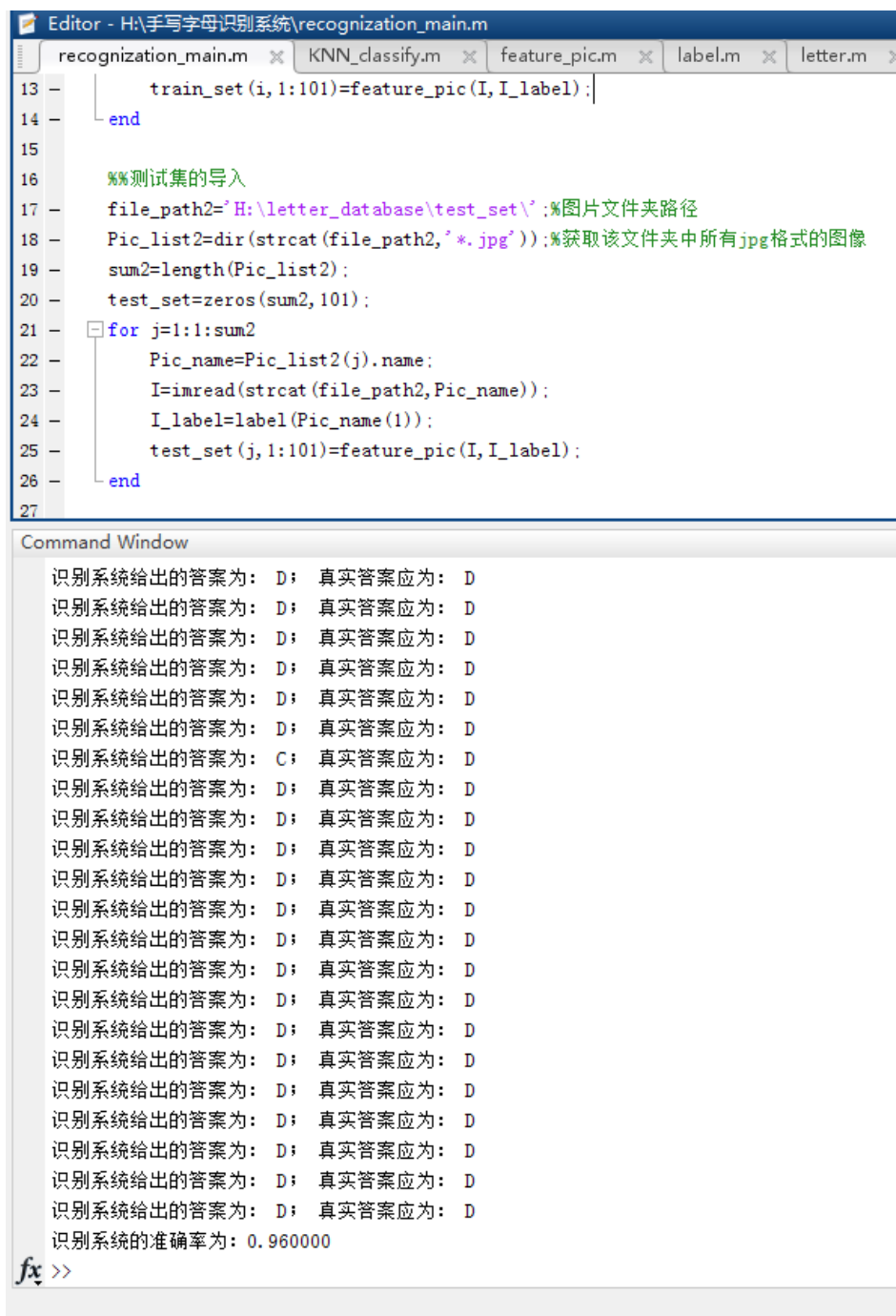
```
1      %手写字母识别系统
2 -    clc;clear;
3
4      %%训练集的导入
5 -    file_path1='H:\letter_database\train_set2\';%图片文件夹路径
6 -    Pic_list1=dir(strcat(file_path1,'*.jpg'));%获取该文件夹中所有jpg格式的图像
7 -    sum1=length(Pic_list1);
8 -    train_set=zeros(sum1,101);
9 -    for i=1:1:sum1
10 -        Pic_name=Pic_list1(i).name;%获取第i张图片的图片名
11 -        I=imread(strcat(file_path1,Pic_name));%读取图片
12 -        I_label=label(Pic_name(1));
13 -        train_set(i,1:101)=feature_pic(I,I_label);
14 -    end
15
16     %%测试集的导入
17 -    file_path2='H:\letter_database\Formal_Test\';%图片文件夹路径
18 -    Pic_list2=dir(strcat(file_path2,'*.jpg'));%获取该文件夹中所有jpg格式的图像
19 -    sum2=length(Pic_list2);
20 -    test_set=zeros(sum2,101);
21 -    for j=1:1:sum2
22 -        Pic_name=Pic_list2(j).name;
23 -        I=imread(strcat(file_path2,Pic_name));
24 -        I_label=label(Pic_name(1));
25 -        test_set(j,1:101)=feature_pic(I,I_label);
26 -    end
```

```

27
28 %%手写字母识别和准确率计算
29 - k=200;
30 - amount=0;
31 - for r=1:1:sum2
32 -     true_label=test_set(r,101);
33 -     true_letter=letter(true_label);
34 -     test_ex=test_set(r,:);
35 -     test_label=KNN_classify(test_ex,train_set,20);
36 -     test_letter=letter(test_label);
37 -     if true_label==test_label
38 -         amount=amount+1;
39 -     end
40 -     fprintf(' 识别系统给出的答案为: ');
41 -     fprintf(test_letter);
42 -     fprintf(' ; 真实答案应为: ');
43 -     fprintf(true_letter);
44 -     fprintf('\n');
45 - end
46
47 %%识别系统的准确率
48 - Accuracy=amount/sum2;
49 - fprintf(' 识别系统的准确率为: %f\n',Accuracy);

```

图 12. 手写字母识别的 MATLAB 程序



The image shows a MATLAB environment with the Editor window displaying the script 'recognition_main.m'. The script includes a loop for training and a loop for testing. The Command Window shows the results of the testing process, listing the predicted and true answers for 27 samples, and the final accuracy rate.

```
13 - train_set(i,1:101)=feature_pic(I,I_label);
14 - end
15
16 %%测试集的导入
17 - file_path2='H:\letter_database\test_set\';%图片文件夹路径
18 - Pic_list2=dir(strcat(file_path2,'*.jpg'));%获取该文件夹中所有jpg格式的图像
19 - sum2=length(Pic_list2);
20 - test_set=zeros(sum2,101);
21 - for j=1:sum2
22 -     Pic_name=Pic_list2(j).name;
23 -     I=imread(strcat(file_path2,Pic_name));
24 -     I_label=label(Pic_name(1));
25 -     test_set(j,1:101)=feature_pic(I,I_label);
26 - end
27
```

Command Window

识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: C; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统的准确率为: 0.960000

fx >>

图 13. 手写字母识别测试结果-准确率

2.3 与正确答案匹配及评分

2.3.1 识别试卷中的字母

对选出的 27 份试卷中的单选题，通过去红勾、二值化处理，应用 kNN 算法

进行分类后，得到识别的准确度为 74.4%。



The image shows a MATLAB environment with a script editor and a command window. The script editor displays the following code:

```
1 %手写字母识别系统
2 clc;clear;
3
4 %%训练集的导入
5 file_path1='H:\letter_database\train_set2\';%图片文件夹路径
6 Pic_list1=dir(strcat(file_path1, '*.jpg'));%获取该文件夹中所有jpg格式的图像
7 sum1=length(Pic_list1);
8 train_set=zeros(sum1,101);
9 for i=1:sum1
10     Pic_name=Pic_list1(i).name;%获取第i张图片的图片名
11     I=imread(strcat(file_path1,Pic_name));%读取图片
12     I_label=label(Pic_name(1));
13     train_set(i,1:101)=feature_pic(I,I_label);
14 end
15
16 %%测试集的导入
17 file_path2='H:\letter_database\Formal_Test\';%图片文件夹路径
18 Pic_list2=dir(strcat(file_path2, '*.jpg'));%获取该文件夹中所有jpg格式的图像
19 sum2=length(Pic_list2);
20 test_set=zeros(sum2,101);
```

The Command Window displays the following output:

```
识别系统给出的答案为: C; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: C; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: C; 真实答案应为: D
识别系统给出的答案为: C; 真实答案应为: D
识别系统给出的答案为: C; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统给出的答案为: D; 真实答案应为: D
识别系统的准确率为: 0.744086
```

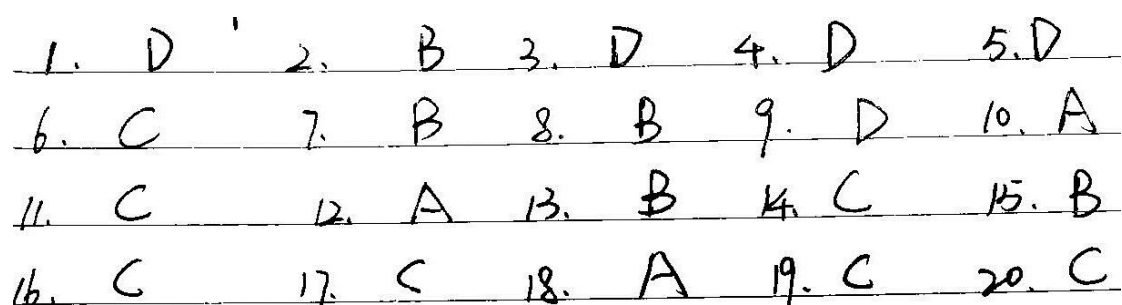
The Command Window ends with the prompt `fx >>`.

图 14. 试卷识别结果

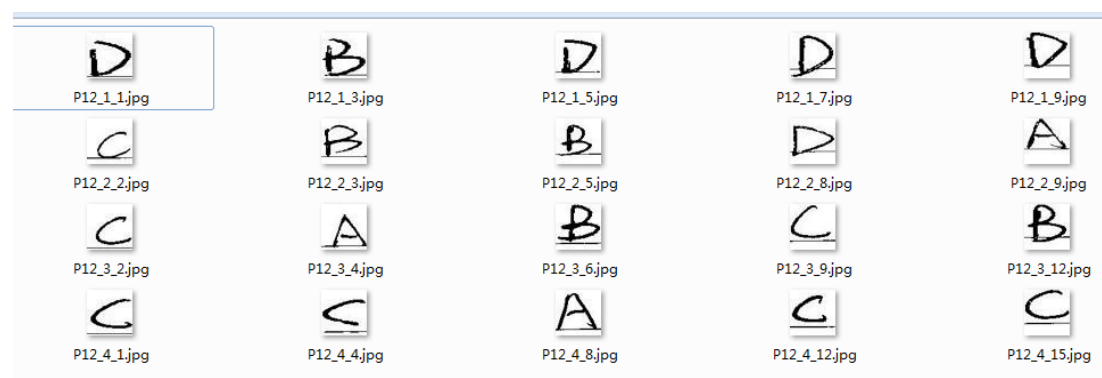
2.3.2 与正确答案匹配并评分

最后利用上述的手写字母识别分类器来对试卷进行打分。我们仍然选择单选题样张来作为范例，进行演示。如下图所示，选择一位同学的试卷，将其单选题

部分切割出来，并进行去红勾、二值化、切割单个字母等处理。



处理后的学生答案所呈现的单个字母如下。



老师给出的单选题标准答案为：

DBDDD CBBDA CABCB CBACC

通过手动判卷，该学生单选题的得分应为 19 分。

下面我们通过手写字母识别分类器来对试卷进行打分，所调用的程序与结果如下：

```
Scoring.m
1 %对试卷的单选题部分进行打分评卷
2 %手写字母识别系统
3 clc;clear;
4
5 %%训练集的导入
6 file_path1='H:\letter_database\Formal_Itrain\';%图片文件夹路径
7 Pic_list1=dir(strcat(file_path1,'*.jpg'));%获取该文件夹中所有jpg格式的图像
8 sum1=length(Pic_list1);
9 train_set=zeros(sum1,101);
10 for i=1:1:sum1
11     Pic_name=Pic_list1(i).name;%获取第i张图片的图片名
12     I=imread(strcat(file_path1,Pic_name));%读取图片
13     I_label=label(Pic_name(1));
14     train_set(i,1:101)=feature_pic(I,I_label);
15 end
```

```

Editor - H:\手写字母识别系统\Scoring.m
Scoring.m  x  +
16
17 %%样本试卷的学生答案导入
18 file_path2='H:\letter_database\Scoring\';%图片文件夹路径
19 Pic_list2=dir(strcat(file_path2,'*.jpg'));%获取该文件夹中所有jpg格式的图像
20 sum2=length(Pic_list2);
21 test_set=zeros(sum2,101);
22 question_num=[];%用来储存题号
23 for j=1:1:sum2
24     Pic_name=Pic_list2(j).name;
25     net_name=Pic_name(1:end-4);
26     question_num(j)=str2num(net_name);
27     I=imread(strcat(file_path2,Pic_name));
28     I_label=0;%由于学生试卷的答案在识别前未知，所以其标签统一置为0
29     test_set(j,1:101)=feature_pic(I,I_label);
30 end
31
32 %%试卷单选题标准答案标签的输入,1代表A, 2代表B, 3代表C, 4代表D
33 standard_label=[4, 2, 4, 4, 4, 3, 2, 2, 4, 1, 3, 1, 2, 3, 2, 3, 2, 1, 3, 3];
34
35 %%手写字母识别
36 k=200;
37 amount=0;
38 for r=1:1:sum2
39     test_ex=test_set(r,:);
40     test_label=KNN_classify(test_ex,train_set,20);
41     test_letter=letter(test_label);
42     standard_letter=letter(standard_label(question_num(r)));
43     if test_label==standard_label(question_num(r))
44         amount=amount+1;
45     end
46     fprintf(' 识别系统读出的学生答案为: ');
47     fprintf(test_letter);
48     fprintf('； 评卷的标准答案为: ');
49     fprintf(standard_letter);
50     fprintf('\n');
51 end

```

```

52
53      %%给出判卷所得的单选题分数
54      score=amount;
55      fprintf('该学生单选题得分为: %d\n', score);
56

```

Command Window

```

识别系统读出的学生答案为: D; 评卷的标准答案为: D
识别系统读出的学生答案为: C; 评卷的标准答案为: A
识别系统读出的学生答案为: C; 评卷的标准答案为: C
识别系统读出的学生答案为: D; 评卷的标准答案为: A
识别系统读出的学生答案为: B; 评卷的标准答案为: B
识别系统读出的学生答案为: C; 评卷的标准答案为: C
识别系统读出的学生答案为: B; 评卷的标准答案为: B
识别系统读出的学生答案为: C; 评卷的标准答案为: C
识别系统读出的学生答案为: C; 评卷的标准答案为: B
识别系统读出的学生答案为: A; 评卷的标准答案为: A
识别系统读出的学生答案为: C; 评卷的标准答案为: C
识别系统读出的学生答案为: B; 评卷的标准答案为: B
识别系统读出的学生答案为: C; 评卷的标准答案为: C
识别系统读出的学生答案为: D; 评卷的标准答案为: D
识别系统读出的学生答案为: D; 评卷的标准答案为: D
识别系统读出的学生答案为: D; 评卷的标准答案为: D
识别系统读出的学生答案为: C; 评卷的标准答案为: C
识别系统读出的学生答案为: B; 评卷的标准答案为: B
识别系统读出的学生答案为: A; 评卷的标准答案为: B
识别系统读出的学生答案为: D; 评卷的标准答案为: D
该学生单选题得分为: 16

```

fx >>

我们的手写字母识别系统给出的判卷评分为 16 分，与真实成绩 19 分存在偏差，这说明该方法还有待于改进的地方。

三、总结及后续问题的解决思路

3.1 总结

前面详细具体的阐述了我们小组的手写字母识别系统的原理、程序以及运用结果，不过最后的结果存在一定的偏差，因此我们就造成的原因进行了一定的分析。

1. 由于样本试卷上所做答案的随意性，以及阴影与横线等因素的干扰，在切割成单个字母块时，便存在着样本量的不准确问题。在用收集到的无干扰的测试字模进行分类器测试时，准确率能达到 96%；而在用试卷上干扰因素较大的测试集，进行测试时，准确率便大幅度下降为 74.4%。原本手写字母的识别难度就比打印体难，再加上干扰条件，导致偏差略大。因此测试样本的随意性会对结果产生较大影响。
2. 分类算法本身的局限性。KNN 算法在提取图片特征时，仅仅是通过计算每个方格内的黑色像素点所占的概率值，没有极大程度的利用和收集图片信息。因此算法的局限性也限制了准确率的进一步提升。
3. 对于切割的方法，可能需要沿着所有字母的边线切割出来，所得到的字模会更利于判断。

3.2 对于图片阴影和多选题的思考

1. 由于所给试卷的阴影比较明显，会影响到识别的判断。可以通过对阴影区域进行直方图统计分析，分别获得阴影在 H、S、V 通道下的各自的颜色特征，再根据以上得出的特征在相应的三个通道上使用阴影样本训练模型参数建立高斯阴影模型，在此基础上采用合适的算法进行阴影消除。
2. 针对多选题，在切割时需要边切割边判断。即在编辑出一个手写数字的识别程序，在每切割出一个单个字母或数字时，进行识别：若为数字，则跳转到下一题；若为字母，则是相同题号下的，多个答案。通过实时的判别分析，将每道多选题的多个答案归在一起，再通过字母识别进行评分判断。
3. 而算法准确率的问题，可以考虑采用 BP 神经网络或者卷积神经网络，以充分获取图片的信息，最大化的利用训练集，从而提高判断的准确率。