



日期： 2023 年 3 月 7 日

课程： 图像处理

实验： 手写数字识别

周次： 第 1-2 周

专业： 智能科学与技术

姓名： 周德峰

学号： 21312210

1 实验内容及目的

- 熟练利用 matlab 对图像进行处理
- 从数据库中挑选训练集和测试集
- 识别 MNIST 中的数字
- 进行准确度计算

2 实验相关原理描述

基于图像处理的手写数字识别是一种利用图像处理技术对手写数字进行预处理和特征提取，然后使用模式识别算法进行分类的方法实验步骤如下：

- 从数据集中选取合适的模板
- 对待测数据进行图像处理
- 运用相关算法识别出最匹配的模板，得到结果

3 实验过程

3.1 选定模板

任务：即从数据集中选出具有代表性的模板，从而使最后识别结果更好
那么如何选择模板呢？

3.1.1 人工选择

从 0-9 的数据中挑选出角度各异的数字，每个数字的模板大约 10 个左右

3.1.2 matlab 批量随机随机选择

考虑到训练数据较大，且机器学习的训练比一般为 7: 3，模板选择数量应较合适，所以设计一个函数来随机自动挑选模板，使得模板数量与待检测数据比为 7: 3

```
1 for i=1:length(sub_trainindir)
    if( isequal( sub_trainindir( i ).name, '.' ) ||...
3       isequal( sub_trainindir( i ).name, '..' ))
        continue;
5     end
    sub_trainindirpath=fullfile(trainindir,sub_trainindir(i).name);
7     train_data=dir(sub_trainindirpath); %具体打开某一个数字文件夹
    num_tem=300; %模板数量
9     randlist=randperm(1000); %生成不重复的序列
    %错误原因：取 1-1000 的序列，实际应取 3-1002，因为 1 是. 2 是..
11    %解决方法：直接在后面 +2
    for j =1: num_tem
13        index=randlist(j)+2;
        src= fullfile (sub_trainindirpath,train_data(index).name);
15        dst= fullfile (temdir,sub_temdir(i).name);
        movefile(src,dst);
17    end
    % template=fullfile(trainindir,)
19 end
```

3.1.3 模板预处理

利用 `img=imread()` 将选定的图片载入到工作区变量中，再利用 `cell2mat()` 函数将图片的数据格式从元胞变为基础数据格式使用 `imwrite()` 将转换数据格式后的模板图片存入下图为二值化图片与原始图片比较

3.2 图像规整

任务：即对待识别的图片的大小，数据类型进行统一化处理

具体处理包括

- 对待测图片进行数据处理（参考上一步）
- 将待测图片与模板图片大小统一（`imresize()`）
- 将待测图片灰度化（`mat2gray()`）



图 1: 二值化图片与原始图片

- 将待测图片二值化 (imbinarize() 或 im2bw())
- 检测是否需要反转 (imcomplement())

为使代码更加简便，复用性更高，已经所有操作封装成函数

```
function [img2]=imgprocess(img)
2   %图像处理函数，将图像进行一系列处理操作
   img=imresize(img,[28,28]); %调整图像大小
4   img = filter2(fspecial('sobel'),img); %此运算寻找边缘
   img_g = mat2gray(img); %灰度化
6   threshold = graythresh(img_g); %使用 Otsu 方法计算全局图像阈值
   img2 = imbinarize(img_g, threshold);
```

3.3 对比识别

任务：即将待测图片与模板图片进行对比，得到识别结果

3.3.1 损失函数

在此次实验中，我们选择差值法来作为我们识别的标准，亦即损失函数

$$\text{Error} = \sum_{i=1}^h \sum_{j=1}^w |(\text{img}(i,j) - \text{model}(i,j))| \quad (1)$$

其中 $\text{img}(i,j)$ 为图像矩阵的某个点

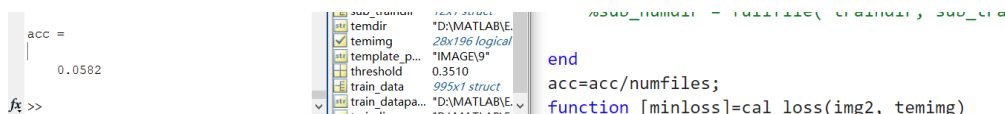
3.3.2 具体实现

由于模板是由多张图片横向拼接组成, Model 是一个很大的横型矩阵, 所以我们对 Model 的每一部分与待训练数据比较, 得到最小的 loss 值, 作为该模板与训练数据的比较 loss

4 实验结果

4.1 人工选择

人工选择时, 每个模板我选择了 10 张图片, 结果正确率为 5 %, 比较不合理



The image shows a MATLAB environment. On the left, the command window displays the following commands and outputs:

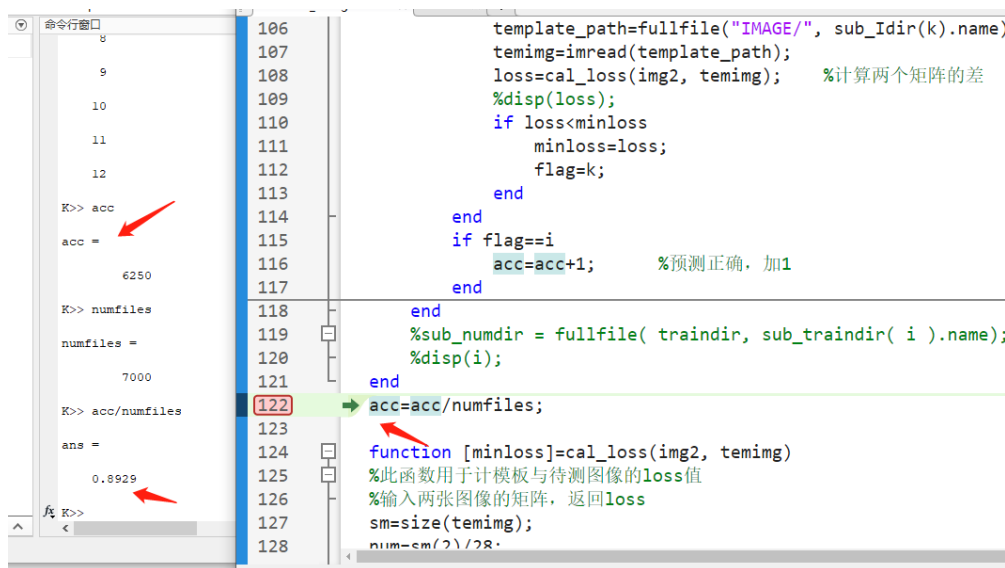
```
K>> acc =  
0.0582  
K>>
```

On the right, the script editor shows a function definition for `cal_loss` and a loop that iterates over template files. The function `cal_loss` calculates the loss between a template image and a test image. The loop iterates over the first 10 files in the `train_data` directory.

图 2: 人工选择

4.2 随机选择, 训练比 7: 3

与人工选择对比, 随机选择完美解决了模板数量不足的问题, 所以总体表现优秀, 正确率为 89 %



The image shows a MATLAB environment. On the left, the command window displays the following commands and outputs:

```
K>> acc =  
6250  
K>> numfiles  
numfiles =  
7000  
K>> acc/numfiles  
ans =  
0.8929  
K>>
```

On the right, the script editor shows a function definition for `cal_loss` and a loop that iterates over template files. The function `cal_loss` calculates the loss between a template image and a test image. The loop iterates over the first 10 files in the `train_data` directory. A red arrow points to line 122, which is `acc=acc/numfiles;`.

图 3: 随机选择

5 总结

本次实验主要在熟练运用 matlab 对图像进行预处理，灰度化，二值化等基础操作，并进一步融合各步操作，对 MNIST 数据集进行手写数字识别在此次实验中，我进行了实验对比，并得出运用随机挑选较多模板可以得到更好的效果的结论，但随机选取仍然存在模板选择不佳的可能性

补充：根据我对深度学习的了解，运用 CNN 卷积神经网络来进行手写数字识别会更好的提取到数字的特征与学习，从而使准确率更高