**总结报告**

姓 名：张硕

学 号：20203473

所在班级：自动化2003

**目 录**

[1. 课设情况简述 1](#_Toc138372732)

[2. 石头剪刀布识别 1](#_Toc138372733)

[2.1 基本框架 1](#_Toc138372734)

[2.2 数据预处理 2](#_Toc138372735)

[2.2.1 自采集数据 2](#_Toc138372736)

[2.2.2 测试集与训练集的划分 2](#_Toc138372737)

[2.2.3 数据增强与数据归一化 2](#_Toc138372738)

[2.3 CNN网络模型的搭建 3](#_Toc138372739)

[2.4 模型的训练效果展示 3](#_Toc138372740)

[2.5 应用界面搭建 4](#_Toc138372741)

[3. 火腿肠外观检测 5](#_Toc138372742)

[3.1 基本框架 5](#_Toc138372743)

[3.2 数据增强与预处理 5](#_Toc138372744)

[3.3 图像特征提取 6](#_Toc138372745)

[3.4 SVM模型搭建（二分类、传统机器学习方法） 6](#_Toc138372746)

[3.5 K-Means聚类模型搭建（缺陷程度分析） 7](#_Toc138372747)

[3.6 ResNet-18模型搭建（深度学习方法） 8](#_Toc138372748)

[3.7 可视化界面的搭建 9](#_Toc138372749)

[4. 项目研发启示 9](#_Toc138372750)

[5. 附录 10](#_Toc138372751)

人工智能课程设计报告

自动化2003班 张硕 20203473

# 课设情况简述

我的选题为：石头剪刀布识别与火腿肠外观检测。我的课程设计完全由Python编程语言实现，程序源码由18个“.py”文件保存，均加入详尽注释，关键变量与关键公式有额外标注，所调用各种库均注有版本说明。

附作业完成环境如下表1：

|  |  |  |
| --- | --- | --- |
| 编译环境 | | Python 3.8 |
| 硬件条件 | CPU | Intel(R) Core(TM)i7-10750H CPU @ 2.60GHz |
| GPU | NVIDIA GeForce GTX 1650 Ti |

表1 作业完成环境

# 石头剪刀布识别

## 基本框架

在本题我完成了“数据自采集”、“训练集与测试集划分”“数据归一化”、“数据增强”、“CNN网络模型搭建”、“模型训练与调参”、“可视化界面搭建”等工作，调用了Python中常见的机器学习库函数，最终模型在测试集、验收集、自采集数据上的准确率达100%，程序步骤如下：

* **Step1 数据自采集**：我采集了自己手心手背、各种拍摄角度以及不同光线的照片，并标记出了自己的姓名信息，为保证数据的单一性与偶然性，我也在采集数据时加入了别的同学采集的数据；
* **Step2 训练集与测试集的划分**：根据数据样本所给编号，按每十个样本中取第八、九号数据作为测试集，其余数据作为训练集；
* **Step3 数据归一化与数据增强**：对给定数据做缩放、归一化、RGB化，得到预处理图像的numpy数组，并对得到的数据进行随机旋转、随机翻转、随机裁剪等操作，用以扩充数据集；
* **Step4 CNN网络模型搭建**：搭建了一个三层卷积层、两层最大池化层、两层全连接层的简单七层CNN神经网络模型，其中，最后一层全连接层的输出为石头剪刀布三种类别标签；
* **Step5 模型的训练与调参**：将处理好的数组送给CNN网络模型，根据训练的表现调整图像数量batch、训练步数epoch及学习率lr，并采用训练成熟的模型对测试集进行测试；
* **Step6 用户界面搭建：**使用Python中的Tkinter库来搭建图形用户界面（GUI），预测处理好的数组文件的标签类型。

## 数据预处理

### 自采集数据

首先我拍摄三种手势的照片各十张，分别加入到原始数据的各个文件夹中。为避免采集数据的单一性与偶然性，我分别采集了自己手心手背、不同拍摄角度、不同光线的照片，并收集其他同学采集的数据，共同加入到原始数据集中。部分自采集数据与原始数据文件划分展示如图1所示。

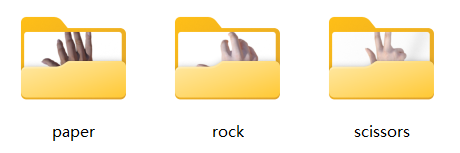
  

图1 自采集数据部分展示

### 测试集与训练集的划分

根据对给定样本的观察，样本的命名格式如下例：“paper-02-012”，paper代表标签名称，02代表该标签的第二个子类，012代表该子类的第12个样本，为保证样本训练集与测试集划分的独立性与均衡性，我按每十个数据取其中第8、9个数据作为测试集，其余数据作为训练集。

### 数据增强与数据归一化

而后我对给定数据做了数据增强处理，即对原图进行随机旋转、随机裁剪、随机翻转等操作，既扩充了数据集，也减少了模型过拟合的风险，使模型的泛化能力得到提升。而后，我也对得到的数据按比例进行缩放、RGB化、归一化转化为numpy数组，便于后续的神经网络模型的搭建，具体过程如下图2所示。



图2 数据增强具体过程

## CNN网络模型的搭建

近年来，随着残差等解决梯度下降的理论不断出现，卷积神经网络模型（CNN）的理论不断得到完善，加之计算机硬件性能的爆炸性提升，卷积神经网络的算法已经相当成熟。由于该算法避免了对图像的复杂前期预处理，且可以提取到人观察不到、更为抽象的特征，因而在图像处理领域大放异彩，得到了广泛使用。

在本次问题中，我搭建了一个简单的七层神经网络模型如下图3所示。该模型通过三个卷积层和三个池化层提取图像的不同特征，每个卷积层从输入图像中提取特定的特征，随后每个池化层对卷积层输出的特征图进行下采样，以减少每个特征图的大小，并且对检测到的特征进行平均或最大值池化，目的是减少模型中需要处理的参数数量，同时防止模型过拟合。最后，模型使用全局平均池化层将特征图压缩成一个向量，并将其传递到全连接层中，用于进行石头剪刀布的分类任务。

在该模型中，选择 CategoricalCrossentropy 做为损失函数，选择Adam优化器作为提升器，选择分类问题中常用的Accuracy作为评估指标，最后加入加入0.01正则化系数的L2正则化来减小过拟合。

## 模型的训练效果展示

经过对我自己笔记本算力的评估以及多轮超参数调整，最终确定训练时批处理的图像数量batch为64，训练步数epoch为15，模型的学习率learning rate为0.01，经过最终训练模型在训练集上的准确率达99%，在测试集数据上的准确率达100%，模型在训练过程中的loss与准确率如下图4所示，训练成熟的模型在验收集、部分自采集数据上的具体预测结果如下图5所示。

|  |  |
| --- | --- |
| 图3 CNN网络模型 | 图4 CNN模型accuracy与loss |
| 图5 验收集与自采集数据具体预测结果 |

## 应用界面搭建

为方便课堂效果展示，我调用了Python中的Tkinter库搭建了一个简单的图像用户界面（GUI），点击“Select file”选择处理好的数组文件，点击“Predict”预测，界面中就会显示出预测结果，展示效果如下图6所示：

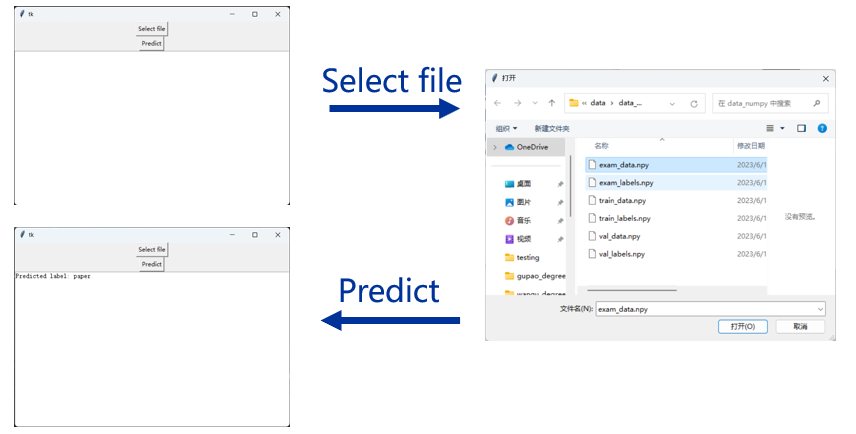


图6 应用界面展示

# 火腿肠外观检测

## 基本框架

在本题中我完成了“数据增强与预处理”、“图像特征提取”、“SVM模型搭建”、“模型训练与调参”、“K-Means聚类算法搭建”、“ResNet-18模型搭建”、“可视化界面搭建”等工作，最终模型在测试集上的准确率达86.36%

* **Step1 数据增强与预处理**：对给定数据按比例缩放、随机裁剪、随机翻转、随机旋转等操作，扩充数据集，便于后续的ResNet-18神经网络模型与SVM模型的使用
* **Step2 图像特征提取**：择了火腿肠的面积、周长、轮廓方向三个特征，对原图做灰度化、二值化，提取上述三种特征以供后续SVM模型训练；
* **Step3 SVM模型搭建（传统机器学习算法）**：拼接所有图像特征，采用C-SVC模型，设置S核函数为高斯径向基函数，训练模型最优化分类超平面，并对测试集进行验证；
* **Step4 K-Means聚类算法搭建(缺陷程度分析)**：根据Step2提取到的三个特征，分别对弯曲缺陷与鼓泡缺陷的样本，采用K-Means算法进行聚类，将缺陷程度相近的样本归为一类；
* **Step5 ResNet-18模型搭建（深度学习方法）**：将预处理好的数据送给ResNet-18模型，根据训练的表现调整图像数量batch、训练步数epoch及学习率learning rate，并与传统机器学习模型作比较；
* **Step6 用户界面搭建：**使用Python中的Tkinter库来搭建图形用户界面（GUI），预测处理好的数组文件的标签类型。

## 数据增强与预处理

按照上述框架，我首先对给定数据做了数据增强的处理，即对原图进行按比例缩放、随机裁剪、随机翻转、随机旋转等操作，具体效果如下图7扩充了数据集，也减少了过拟合风险，提高模型的泛化能力，这里与问题一中数据增强处理步骤基本一致，这里不再过多赘述。



图7 火腿肠外观检测数据增强

## 图像特征提取

观察火腿肠可以发现，在鼓泡缺陷发生时，火腿肠的整体面积与周长相较正常的火腿肠是变大的，且轮廓方向发生改变；在弯曲缺陷发生时，火腿肠的轮廓方式也会发生较明显变化，因此这里我选择提取图像的周长、面积、轮廓方向三个特征，作为传统机器学习算法的分类依据。

这里我首先将给定的火腿肠照片灰度化、二值化，统计每个二值化图像的轮廓边缘点的个数作为周长，统计轮廓边缘点包围区域作为面积特征，采用cv2.HuMoments()函数度量轮廓形状作为轮廓方向特征，简单处理过程如下图8：

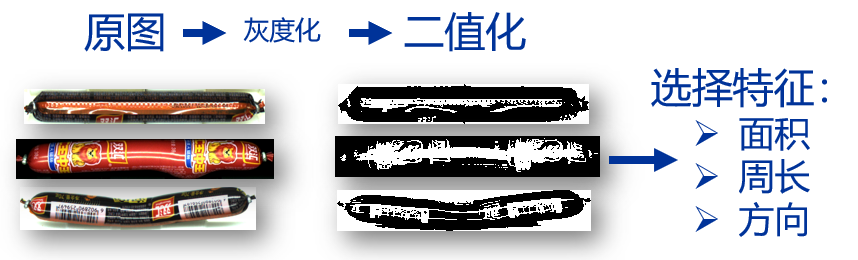


图8 火腿肠图像特征提取

## SVM模型搭建（二分类、传统机器学习方法）

SVM模型又名支持向量机，是我们课堂上学习过的典型二分类模型（也可用于多分类），我的理解中，它将特征向量映射为空间中的一些点，找到最合适的超平面，以 “最好地” 区分这些点，以至如果以后有了新的点，超平面也能做出很好的分类。SVM 十分适合中小型数据样本、非线性、高维的分类问题，被誉为深度学习前表现最优秀最成功的算法。

这里我将所有提取特征拼接为长度一致的向量，指定C-SVC模型，并设置高斯径向基函数（RBF）作为S核函数，进行训练并不断调整模型的参数，最终模型在训练集上的准确率为69.8%，在测试集上的准确率为66.7%，模型的混淆矩阵如下图9所示，可以看出该模型的准确率相对并不高。

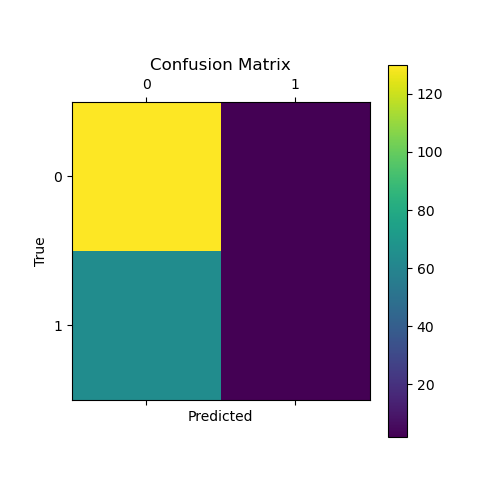
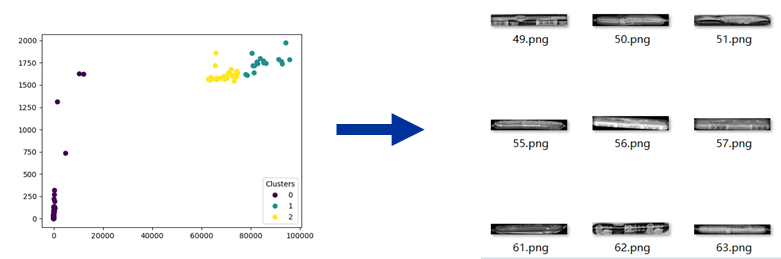


图9 SVM模型混淆矩阵

## K-Means聚类模型搭建（缺陷程度分析）

观察样本数据可知，火腿肠缺陷程度并没有给出明确的标签，因此该问题明显属于无监督学习，由此我选择了K-Means算法，是一种典型的基于划分的聚类算法，也是一种常用的无监督学习算法。

聚类算法同样需要手动提取特征，这里我采用了上一步中SVM模型中提取到的面积、周长、轮廓方向三种特征，采用手肘法分别判断两种缺陷的最佳分类簇数，结果展示如下图10所示：



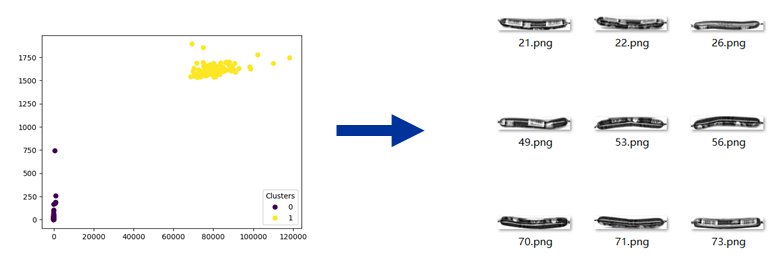


图10 K-Means聚类效果展示

根据分类后的结果可以看出，同一类之间的弯曲、鼓泡程度相近，区分的效果较为优良。

## ResNet-18模型搭建（深度学习方法）

ResNet-18是一个深度卷积神经网络模型，它由Microsoft Research在2015年发布。该模型具有18层深度和残差连接，图像分类性能非常强大。由于其相对较浅的深度和较低的复杂度，广泛应用于计算机视觉领域。

通过调用PyTorch提供的nn.Module和resnet18等组件，可以很容易实现一个简单而高效的ResNet-18图像分类模型，该模型的概述如图11所示。

这里我将原模型的全连接层替换成了一个新的全连接层，其输入尺寸与ResNet-18的全连接层的输入尺寸相同，但输出尺寸改为3，以此来适合我们需要分类的类别数量。

|  |  |
| --- | --- |
| https://ask.qcloudimg.com/http-save/yehe-8223537/f1ad11d26ff34b88ad8d705fd32d036d.png?imageView2/2/w/1200  图11 ResNet-18模型概述图 | 图12 ResNet-18训练过程 |
| 图13 ResNet-18模型混淆矩阵 |

通过不断地超参数调整，最终确定了模型地学习率ρ为0.013，训练过程与最终混淆矩阵如上图12、13所示。最终模型在测试集上的准确率达86.36%，在验收集上的准确率达100%，可以看出训练效果明显优于上述SVM算法。

## 可视化界面的搭建

这里我采用了与石头剪刀布识别相近的方式，调用了Python中的Tkinter库搭建了火腿肠外观检测模型的应用检测程序如下图14所示，点击Select image可以选择待检测的图片，点击Predict就可以输出待预测标签，可以看出该应用程序简洁高效，效果较优，加以包装并完备训练，是可以投入到实际生产中的。

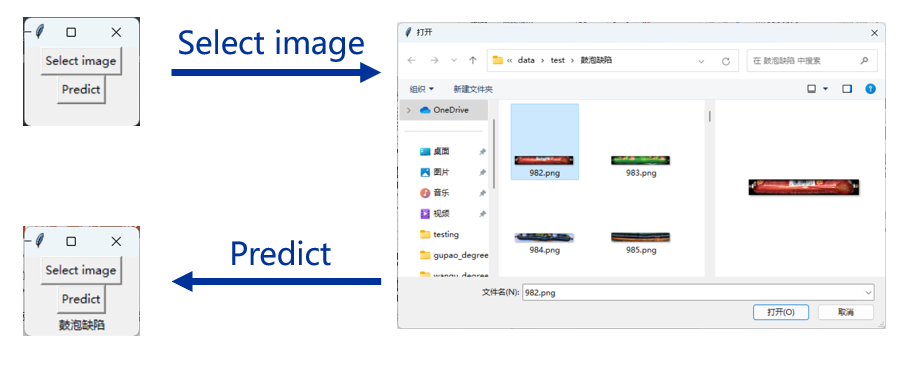


图14 火腿肠外观检测用户界面展示

# 项目研发启示

本次人工智能课程设计，让我从头到尾地经历了一遍机器学习算法应用的过程，使我人工智能的理解又高了一个层次。以我本次选择地火腿肠外观检测问题举例，传统机器学习方法与深度学习算法的区别清楚展现在眼前：SVM模型虽然训练速度相当快，但严重依赖人工特征提取；而ResNet-18模型虽然训练速度较慢，但无需人工特征提取。这也是该问题中我搭建的SVM模型准确率较为低下的部分原因——我并未选择到最合适的特征作为分类依据。与之相比，深度学习算法不仅无需繁琐的特征提取工作，也能够提取到人类或许无法理解的，更为抽象的特征作为分类依据，同时这也是深度学习模型可解释性弱的直观展示。

另一个直观感受是计算机硬件的算力问题，在火腿肠外观检测中，最开始我使用的CPU进行训练，每100个Epoch就需要6-7个小时，这对我的耐心考验极大，也非常影响我参数调整的效率，之后我改用了Python中GPU相关的库进行训练，效率得到大大提升，每三十分钟就能得到训练结果。而这尚且发生在我这个简单的课程设计，如果大到国家级别的工程，那对算力的要求又该有多么之高？这也让我更直观的理解了国家对于超级计算机的投入开发。

很可惜的是由于课程时间有限，我的参数调整与模型选择也无法做到最合适，且应用界面的制作也略显简陋，希望以后能够从数据采集、到用户界面搭建，开发一套更为完善的、可用于工业界的的方案。

# 附录

最后附上源代码文件的名称以及用途如下表2所示。

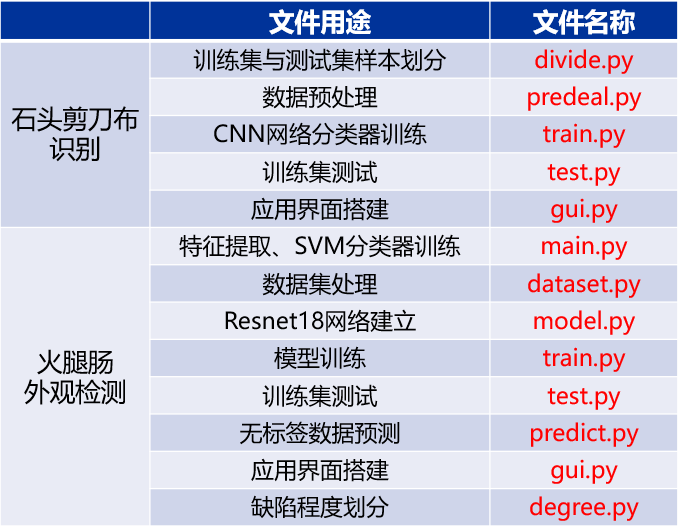


表2 源代码文件速览