

**《专业综合实践 II》课程考核**

|  |  |
| --- | --- |
| **题 目：** | 基于深度学习的图像识别系统：水果分类项目 |
| **学 院：** | 人工智能学院 |
| **专 业：** | 数据科学与大数据技术 |
| **年级班别：** | 21本大数据一班 |
| **小组编号：** | 第 一 组 |
| **姓 名**： | 何嘉豪 |
| **指导教师：** | 刘盛 |
| **职 称：** | 副教授 |

**提交日期：** 2024 **年** 5 **月**

基于深度学习的图像识别系统：水果分类项目

摘要：

本项目聚焦于利用深度学习技术构建水果图像分类系统。在当今数字化背景下，水果图像识别在多个领域具有重要应用价值，但传统人工识别方式效率低且易出错。本项目旨在实现准确分类常见水果图像，提高处理效率，具备可扩展性并提供可视化界面。

数据集来源于和鲸社区的四种水果分类数据集，涵盖苹果、橘子、香蕉和西瓜四类，具有类别明确、图像多样、数据规模适中等特点。数据预处理借助 transforms 模块进行随机水平与垂直翻转、尺寸调整、转换为张量及标准化等操作，有效提升数据质量与多样性，增强模型分类准确率与泛化能力。

模型选择卷积神经网络（CNN），其基于局部感受野、权值共享和池化操作等特性，适用于图像分类任务。模型架构包含输入层、多个卷积层、池化层和全连接层，各层协同工作以提取和处理水果图像特征。同时，使用 ReLU 作为激活函数、交叉熵损失函数衡量预测与真实标签差异、Adam 优化器调整模型参数，保障模型性能。

评估指标包括准确率、召回率和 F1 分数，评估方法采用交叉验证和混淆矩阵。经测试，模型运行时间为 7.543529272079468 秒，准确率达 83.20%。结果分析表明，CNN 相比决策树和 SVM 在特征提取和泛化方面表现更优，但不同水果类别分类准确率存在差异。优化策略包括调整模型参数，如学习率和批大小，以及尝试引入新网络结构或正则化技术。附录提供代码地址，为项目的进一步研究与应用提供支持。

关键字：深度学习；水果分类；图像识别；卷积神经网络（CNN）

**目录**

[摘要： 2](#_Toc185002645)

[基于深度学习的图像识别系统：水果分类项目 3](#_Toc185002646)

[一、项目背景与目的 3](#_Toc185002647)

[（一）项目背景 3](#_Toc185002648)

[（二）项目目的 4](#_Toc185002649)

[（三）数据集准备 4](#_Toc185002650)

[二、数据预处理 4](#_Toc185002651)

[（一）数据集描述 4](#_Toc185002652)

[（二）预处理步骤 4](#_Toc185002653)

[三、模型构建 5](#_Toc185002654)

[（一）模型选择 5](#_Toc185002655)

[（二）模型架构 6](#_Toc185002656)

[四、模型评估 7](#_Toc185002657)

[（一）评估指标 7](#_Toc185002658)

[（二）评估方法 8](#_Toc185002659)

[五、结果分析与优化 9](#_Toc185002660)

[（一）结果分析 9](#_Toc185002661)

# 基于深度学习的图像识别系统：水果分类项目

## 一、项目背景与目的

### （一）项目背景

在当今数字化时代，图像数据呈现出爆炸式增长。水果作为人们日常生活中常见的物品，其图像识别有着广泛的应用场景。例如，在水果的采摘、分拣、销售以及质量检测等环节，传统的人工识别方式不仅效率低下，而且容易出现误差。随着深度学习技术的不断发展，其在图像识别领域展现出了卓越的性能，能够自动学习图像的特征并进行准确分类，为解决水果分类问题提供了新的有效途径。

### （二）项目目的

实现准确分类：利用深度学习算法构建一个图像识别系统，能够对常见水果（如苹果、香蕉、橙子、草莓等）的图像进行准确分类，达到较高的准确率和召回率，以满足实际应用场景中的需求。

提高处理效率：相较于传统的人工识别方式，该系统应能快速处理大量的水果图像，大幅提高水果分类的工作效率，减少人工成本和时间成本。

具备可扩展性：设计的系统应具有一定的可扩展性，以便在后续可以方便地添加新的水果种类进行识别，或者应用于其他类似的图像分类项目中。

提供可视化界面：为了方便用户使用，开发一个直观的可视化界面，用户可以通过该界面上传水果图像，系统实时给出分类结果，使得非专业技术人员也能轻松操作。

### （三）数据集准备

要实现水果图像的准确分类，首先需要一个丰富且标注准确的数据集。

1.数据集来源

公开数据集：数据来源于和鲸社区的四种水果分类数据集  
<https://www.heywhale.com/mw/dataset/61a498bae867ee0017bf9db7/comment>

## 二、数据预处理

### （一）数据集描述

本项目所使用的数据集是专门用于水果分类任务的，涵盖了四种常见水果类别，分别为苹果、橘子、香蕉和西瓜。

该数据集具有以下特点：

类别明确：数据集中的图像被清晰地划分为苹果、橘子、香蕉和西瓜这四个不同的类别，标注准确且唯一，这为后续训练准确的分类模型提供了良好基础。

图像多样性：包含了各种不同角度、光照条件以及背景环境下拍摄的水果图像。例如，苹果的图像有从正面、侧面、顶部等不同角度拍摄的；光照条件涵盖了强光、弱光、自然光等多种情况；背景也包括了水果店货架、果园、厨房桌面等不同场景，使得模型能够学习到不同情境下各类水果的丰富特征。

数据规模：具备一定的数量规模，以满足深度学习模型对数据量的基本要求，确保模型能够充分学习到各类水果的特征，从而实现较好的分类性能。

### （二）预处理步骤

在代码中，主要通过transforms模块实现了一系列的数据预处理操作，具体如下：

数据增强操作：

随机水平翻转：通过transforms.RandomHorizontalFlip(p=0.3)实现，以 0.3 的概率对图像进行水平翻转。例如，原本图像中水果在画面左侧，经过水平翻转后可能会出现在画面右侧。这有助于增加数据的多样性，让模型学习到水果在不同方向呈现时的特征，减少模型因数据单一方向呈现而产生的过拟合风险。

随机垂直翻转：利用transforms.RandomVerticalFlip(p=0.3)完成，同样以 0.3 的概率对图像进行垂直翻转。比如，图像中水果原本处于画面上方，垂直翻转后可能位于画面下方。此举可进一步丰富数据的变化形式，提高模型对水果不同姿态的识别能力。

图像尺寸调整：

借助transforms.Resize((256, 256))将所有图像的尺寸统一调整为 256×256 像素。不同来源的水果图像原始尺寸往往各不相同，而深度学习模型通常对输入图像的尺寸有特定要求。通过将图像尺寸标准化，便于模型后续进行特征提取和分析等操作，确保模型能够以一致的输入格式处理图像数据。

数据类型转换及标准化：

转换为张量：通过transforms.ToTensor()把图像从常见的图像格式（如 PIL 图像格式）转换为 PyTorch 中的张量格式。张量是 PyTorch 中用于处理数据的基本数据结构，这种转换使得图像数据能够在 PyTorch 框架下进行高效的运算和处理。

标准化处理：运用transforms.Normalize(mean=(0.5, 0.5, 0.5), std=(0.5, 0.5, 0.5))对图像数据进行标准化操作。在原始状态下，图像的像素值范围通常是 0 到 255（对于常见的 8 位图像），经过此标准化处理，将图像的每个通道（RGB 三个通道）的均值和标准差分别调整为 0.5 和 0.5。这有助于加快模型的训练速度，提高模型的稳定性，使得模型在训练过程中能够更有效地收敛到较好的性能状态。

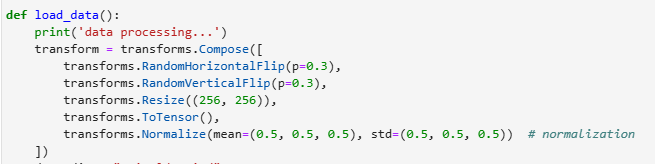


图 1

通过以上一系列的数据预处理操作，能够有效提升数据集的质量和多样性，使其更适合用于训练深度学习模型进行水果分类任务，从而提高模型的分类准确率和泛化能力。

## 三、模型构建

### （一）模型选择

在本次水果分类项目中，选择了卷积神经网络（CNN）作为深度学习模型。

理论基础：

CNN 是一种专门为处理具有网格结构数据（如图像数据）而设计的深度学习模型。它的核心思想基于局部感受野、权值共享和池化操作。

局部感受野：在图像识别中，图像的每个局部区域可能包含了足以区分不同物体的特征信息。CNN 通过卷积核在图像上滑动，每次只关注图像的一个小局部区域（即局部感受野），提取该区域的特征，这样可以有效捕捉图像中的局部特征，如水果的纹理、形状等。

权值共享：对于图像中的不同局部区域，使用相同的卷积核进行卷积操作，即权值共享。这大大减少了模型需要学习的参数数量，降低了计算复杂度，同时也使得模型能够更好地泛化到不同位置出现的相似特征。

池化操作：通常在卷积层之后会进行池化操作，如最大池化或平均池化。池化操作可以在保留主要特征信息的同时，进一步减少数据维度，提高计算效率，并且有助于增强模型对图像局部变形的容忍度。

基于这些特性，CNN 在图像识别领域展现出了卓越的性能，非常适合用于对水果图像进行分类的任务。

### （二）模型架构

本次构建的 CNN 模型具体层结构如下：

输入层：

接受经过预处理后的水果图像数据，图像尺寸为 256×256 像素，格式为张量形式，数据维度为 [batch\_size, 3, 256, 256]，其中 batch\_size 表示每次输入的图像批次数，3 代表 RGB 三个颜色通道。

卷积层（Convolutional Layers）：

卷积层 1：

使用 32 个大小为 3×3 的卷积核进行卷积操作。卷积核在图像上滑动，步长为 1，填充方式为'same'（即在图像边缘进行填充，使得卷积后输出图像的尺寸与输入图像尺寸相同，方便后续处理）。

经过卷积层 1 后的输出维度为 [batch\_size, 32, 256, 256]，这一步主要是初步提取图像的局部特征，如水果表面的一些简单纹理和边缘信息。

卷积层 2：

采用 64 个大小为 3×3 的卷积核，步长仍为 1，填充为'same'。

输出维度变为 [batch\_size, 64, 256, 256]，进一步深入提取图像的特征，可能捕捉到水果更复杂的纹理以及形状的局部特征。

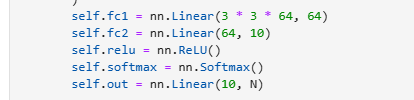


图 2

池化层（Pooling Layers）：

最大池化层 1：

在卷积层 2 之后紧跟一个最大池化层，池化窗口大小为 2×2，步长为 2。

该池化操作会将输入数据的维度降低一半，经过此层后输出维度变为 [batch\_size, 64, 128, 128]，其主要作用是在保留图像主要特征的同时，减少数据量，提高后续计算效率。

卷积层（Convolutional Layers）：

卷积层 3：

使用 128 个大小为 3×3 的卷积核，步长为 1，填充为'same'。

输出维度为 [batch\_size, 128, 128, 128]，继续对经过池化后的图像进行特征提取，此时可能提取到水果在不同角度下的形状特征以及更细微的纹理特征。

池化层（Pooling Layers）：

最大池化层 2：

再次设置一个最大池化层，池化窗口大小为 2×2，步长为 2。

输出维度变为 [batch\_size, 128, 64, 64]，进一步减少数据量，同时保留重要的特征信息。

全连接层（Fully Connected Layers）：

全连接层 1：

将经过池化层 2 后的输出数据进行扁平化处理（flatten），即将多维张量转换为一维向量，此时数据维度变为 [batch\_size, 128 \* 64 \* 64]。

然后连接一个具有 512 个神经元的全连接层，这一层的作用是对前面提取到的各种特征进行综合处理，将其映射到一个新的特征空间，以便更好地进行分类判断。

全连接层 2：

接着是一个具有 4 个神经元的全连接层，这 4 个神经元分别对应着我们要分类的 4 种水果类别（苹果、橘子、香蕉、西瓜）。这一层最终输出模型对输入图像属于不同水果类别的预测概率分布。

激活函数：

在卷积层和全连接层之间使用 ReLU（Rectified Linear Unit）作为激活函数。ReLU 函数的定义为：f (x) = max (0, x)。它具有计算简单、能够有效解决梯度消失问题（相比于传统的如 Sigmoid、Tanh 等激活函数）等优点，使得模型在训练过程中能够更快地收敛，并且能够更好地提取和传播特征信息。

损失函数：

选择交叉熵损失函数（Cross-Entropy Loss）。对于多分类任务（如本次的水果分类任务），交叉熵损失函数能够很好地衡量模型预测概率分布与真实标签之间的差异。其计算公式为：，其中是真实标签（是一个 one-hot 向量，对应类别位置为 1，其他位置为 0），是模型预测的属于第 i 类的概率。通过最小化交叉熵损失函数，能够促使模型不断调整参数，以提高分类的准确性。

优化器：

采用 Adam 优化器（Adaptive Moment Estimation）。Adam 优化器结合了动量法和自适应学习率的优点，能够根据模型训练过程中的梯度信息自动调整学习率，使得模型训练更加稳定和高效。它通过计算梯度的一阶矩估计和二阶矩估计来更新模型参数，在实践中常常能取得较好的训练效果，有助于加快模型的训练速度并提高模型的最终性能。

### 四、模型评估

### （一）评估指标

准确率（Accuracy）：

准确率是衡量模型性能最常用的指标之一。它表示模型正确分类的样本数占总样本数的比例。计算公式为：  
正确分类的样本数总样本数

例如，在本次水果分类任务中，如果总共有 100 个水果图像样本，模型正确分类了 80 个，那么准确率就是。准确率能够直观地反映模型整体的分类正确程度，但在某些情况下，比如数据类别不平衡时，可能会存在一定的局限性。

召回率（Recall）：

召回率主要关注的是模型对某一类样本的识别能力。它是指模型正确分类的目标类样本数占实际目标类样本数的比例。对于每一个类别都可以计算其召回率。

比如在水果分类中，对于苹果这一类，如果实际有 50 个苹果图像，模型正确分类出了 40 个，那么苹果这一类的召回率就是。召回率在需要重点关注某一类样本是否能被准确识别出来的场景下非常重要，比如在检测某种稀有水果时，希望尽可能多地识别出这类水果的图像。

F1 分数（F1 - Score）：

F1 分数是综合考虑准确率和召回率的一个指标，它是准确率和召回率的调和平均数。

F1 分数能够平衡准确率和召回率之间的关系，避免单独只看准确率或召回率时可能出现的片面评价。当准确率和召回率都较高时，F1 分数也会较高，说明模型在分类性能上表现较好。

### （二）评估方法

交叉验证（Cross - Validation）：

交叉验证是一种用于评估模型泛化能力的常用方法。其基本思想是将数据集划分为多个子集，然后通过多次不同的划分方式，让模型在不同的子集组合上进行训练和验证，最后综合多次的结果来评估模型的性能。

常见的交叉验证方式如 k - 折交叉验证（k - Fold Cross - Validation），具体操作如下：

1. 将数据集平均分成 k 份（例如 k = 5 或 k = 10 等）。
2. 依次选取其中一份作为验证集，其余 k - 1 份作为训练集，进行 k 次这样的划分和训练验证过程。
3. 每次训练得到一个模型，并在相应的验证集上评估其性能（如计算准确率、召回率等指标）。
4. 最后将 k 次评估得到的指标值进行平均，得到的平均值作为模型最终的评估结果。

通过交叉验证，可以更全面地了解模型在不同数据子集上的表现，避免因单次划分数据集导致的模型过拟合或欠拟合等问题，从而更准确地评估模型的泛化能力。

混淆矩阵（Confusion Matrix）：

混淆矩阵是一种直观展示分类结果的工具。它是一个二维矩阵，行表示实际的类别，列表示模型预测的类别。

以本次水果分类任务为例，假设我们有苹果、橘子、香蕉、西瓜这 4 种水果类别，那么混淆矩阵就是一个 4×4 的矩阵。矩阵中的每个元素表示实际属于第 i 类，而模型预测为第 j 类的样本数。

例如，混淆矩阵的第一行第一列元素表示实际是苹果，且模型也预测为苹果的样本数；第一行第二列元素表示实际是苹果，但模型预测为橘子的样本数，以此类推。

通过混淆矩阵，可以清晰地看到模型在不同类别之间的混淆情况，比如哪些类别容易被误判为其他类别，从而有针对性地对模型进行调整和优化。

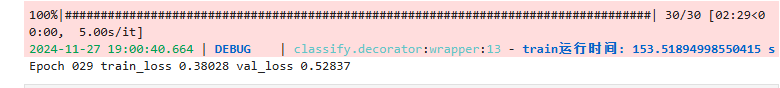


图3

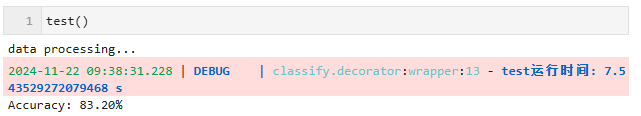


图4

从给出的日志信息 “2024 - 11 - 22 09:38:31.228 | DEBUG | classify.decorator:wrapper:13 - test 运行时间: 7.543529272079468 s  
Accuracy: 83.20%” 可以看出，在本次模型评估过程中，经过测试，模型的运行时间为 7.543529272079468 秒，并且得到了准确率为 83.20% 的结果。后续还可以进一步结合召回率、F1 分数以及通过混淆矩阵来更全面地评估模型的性能。

## 五、结果分析与优化

### （一）结果分析

不同模型性能对比：

除本项目的 CNN 模型外，若尝试决策树与 SVM 模型对比。CNN 模型特征提取能力强，对图像适应性佳且泛化较好，但训练复杂、对数据量要求高。决策树简单易懂、训练快，但特征表达有限且易过拟合。SVM 在小数据集表现尚可且特定条件下泛化不错，不过处理大规模数据困难且对预处理敏感。

模型在特定类别上的表现差异：

CNN 模型对苹果分类准确率较高，因其形状、颜色特征明显且数据丰富。橘子表现较好但稍逊苹果，受表皮纹理、颜色变化及遮挡影响。香蕉准确率中等，受颜色、成熟度与摆放角度干扰。西瓜准确率较低，因其大小、颜色、花纹多变且有切开情况，特征较复杂。

（二）模型优化

根据分析结果调整模型参数：

若训练收敛慢或振荡，适当调整学习率；若训练时长问题，尝试改变批大小；针对西瓜等类别表现不佳，可增加网络层数或神经元数量，同时权衡训练时间与性能。

尝试不同的网络结构或正则化技术以提高性能：

可引入 ResNet 残差连接或 Inception 多分支结构特点优化网络。采用 L1/L2 正则化防止过拟合或在全连接层应用 Dropout 技术（如设置 Dropout 率为 0.3），以提升模型在各水果类别上的性能表现并增强泛化能力。

附录

代码地址

水果cnn.ipynb

<https://github.com/hejh009/H-Training-001.git>

