# 项目背景

**数据集介绍**

102 Category Flower Dataset 是一个用于图像分类任务的数据集，由英国牛津大学的研究人员创建。该数据集包含102种不同种类的花卉，每种花卉的图片数量在40到258张之间，总共包含8189张jpg格式的图片。这些图片主要采集自英国的花卉，具有较高的多样性和复杂性，适合用于图像分类和计算机视觉的研究。

**项目目标**

本项目的主要目标是使用卷积神经网络（CNN）对102种花卉进行分类。具体任务包括：

数据预处理：加载和处理图像数据及其对应的标签。

模型构建：设计并训练一个卷积神经网络模型。

模型评估：评估模型在测试集上的性能。

结果可视化：展示分类正确和错误的图片实例，并绘制训练过程中的准确率曲线。

**项目步骤**

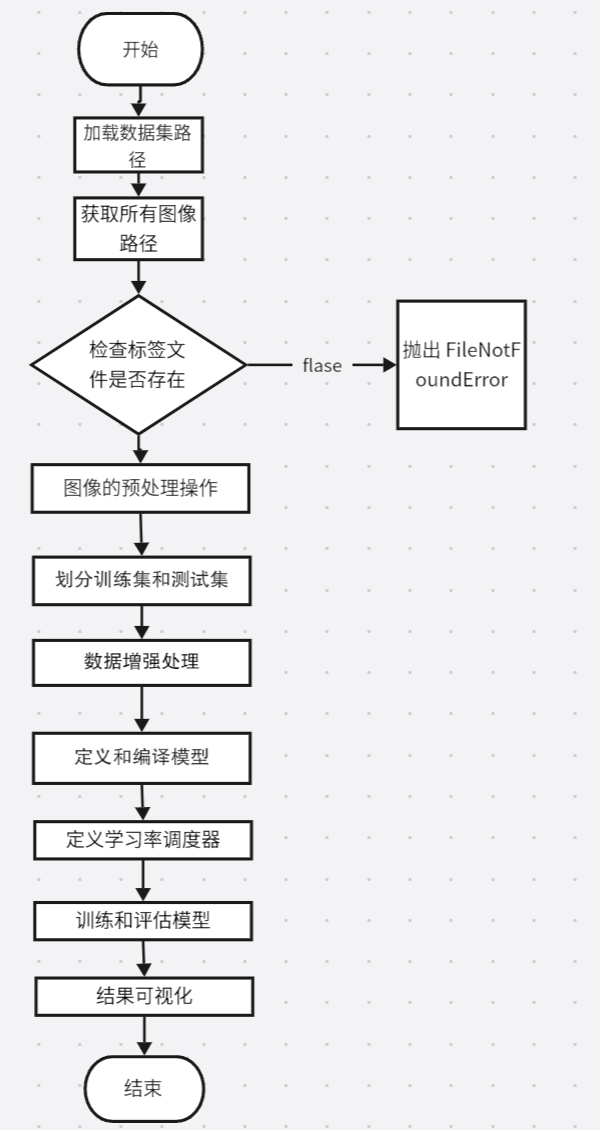
数据加载与预处理：从指定路径加载图像数据，并调整图像大小以适应模型输入。读取标签文件，并将标签转换为适合模型训练的格式（如One-Hot编码）。

将数据集划分为训练集和测试集，并进行归一化处理。

模型构建与训练：构建一个包含多个卷积层、池化层、全连接层和输出层的卷积神经网络模型。使用数据增强技术（如旋转、平移、缩放等）来扩充训练数据，提高模型的泛化能力。训练模型，并在训练过程中监控模型在验证集上的表现。测试比例为数据集的20%。

模型评估与结果可视化：评估模型在测试集上的准确率和损失。绘制训练过程中的准确率和损失曲线，帮助分析模型的训练情况。展示分类正确和错误的图片实例，直观地了解模型的分类效果。

具体流程图：



**项目意义**

通过本项目，我深入了解了卷积神经网络在图像分类任务中的应用，掌握数据预处理、模型构建、训练和评估的完整流程。同时，102 Category Flower Dataset 提供了一个具有挑战性的数据集，有助于我以后在实际应用中提升图像分类模型的性能。

# 技术路线

我使用的模型是一个卷积神经网络（CNN），具体来说是一个由多个卷积层、池化层、全连接层和一个输出层组成的tf.keras中的Sequential模型。以下是模型的详细结构：

输入层：接受形状为 (128, 128, 3) 的图像数据。

卷积层1：包含32个3x3的卷积核，激活函数为ReLU。

最大池化层1：2x2的池化窗口。

卷积层2：包含64个3x3的卷积核，激活函数为ReLU。

最大池化层2：2x2的池化窗口。

卷积层3：包含128个3x3的卷积核，激活函数为ReLU。

最大池化层3：2x2的池化窗口。

展平层：将多维数据展平成一维。

全连接层：包含512个神经元，激活函数为ReLU。

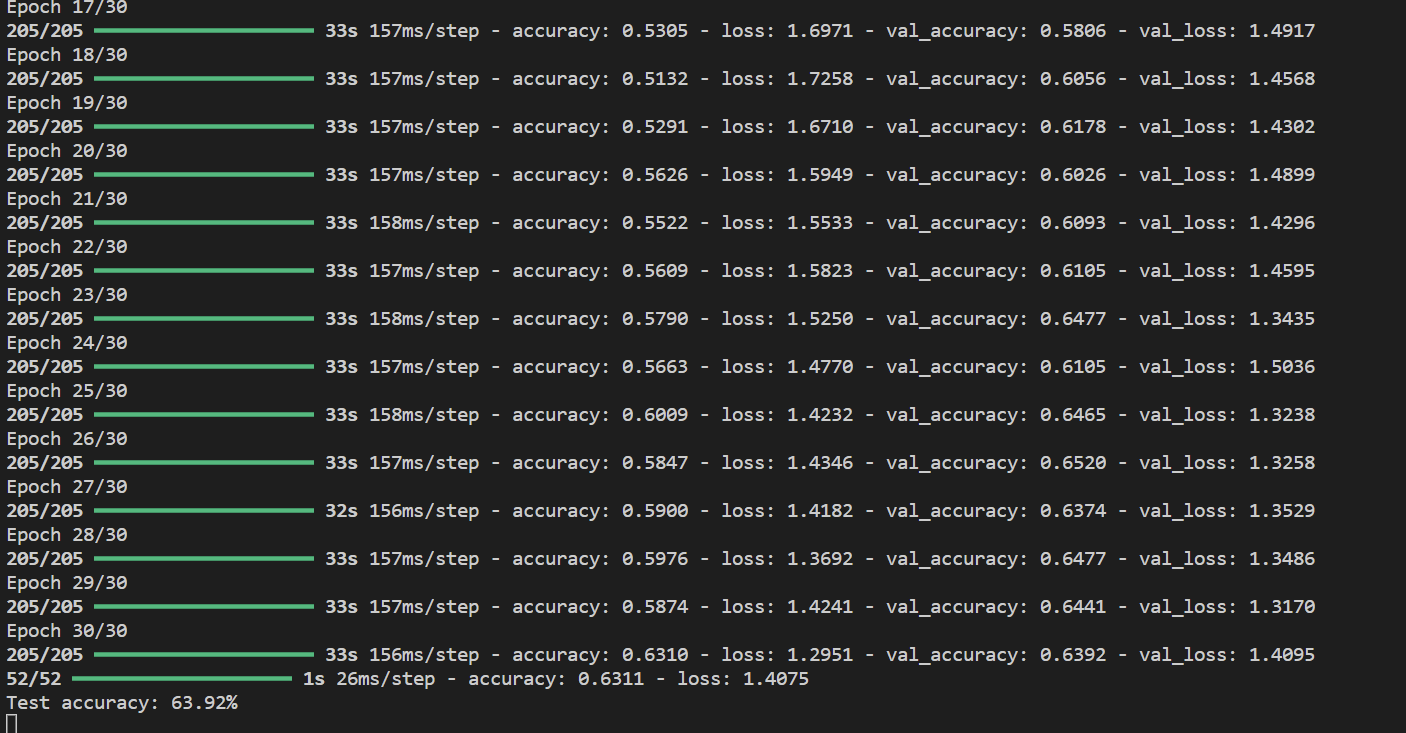
Dropout层：丢弃50%的神经元以防止过拟合。

输出层：包含102个神经元，激活函数为Softmax，用于多分类任务

# 我的优化流程

由于我运用的是已经建立好的cnn模型，所以我在如何提高准确度方面优先考虑的是数据的预处理优化，超参数的调整还有改进训练策略。

初始代码的结果如下：

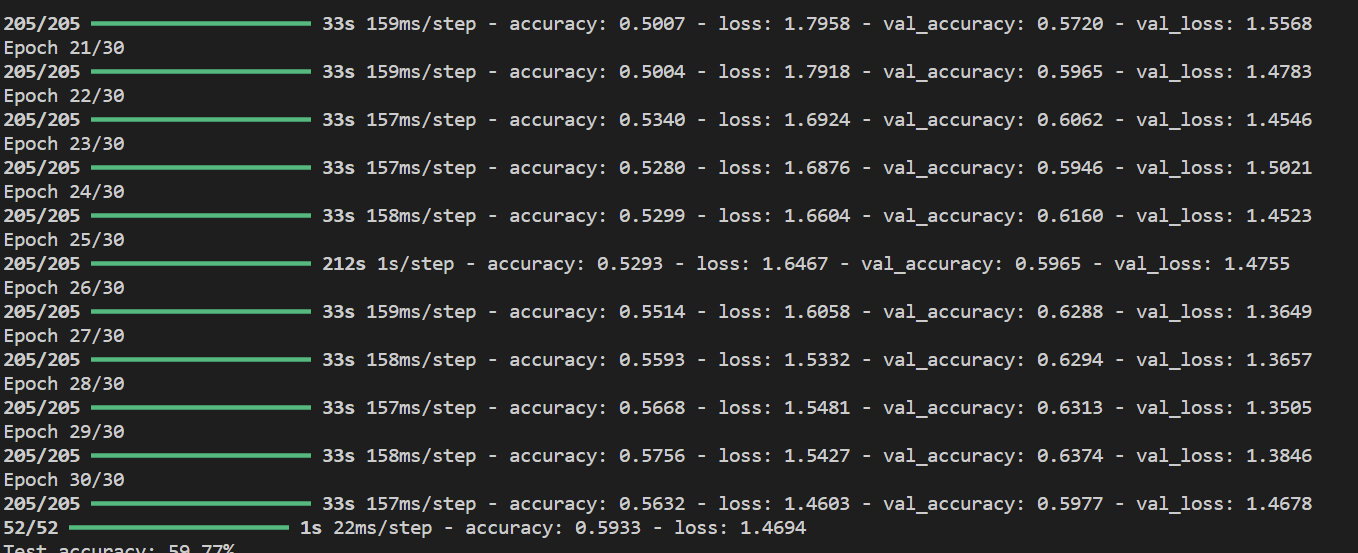


**数据预处理的优化**：使用ImageDataGenerator进行数据增强，增加数据的多样性。

代码如下：

图片包含 图形用户界面

描述已自动生成



准确率反而下降了？chatgpt给出了如下几个可能1. 数据增强参数设置不当2. 模型复杂度不足3. 训练时间不足4. 数据增强方法不适合。观察可知**loss**值一直在下降，**val\_loss**也在下降所以说明网络任在学习，说明原因3所占比例极大，我的精力有限所以我对原因3进行了着重研究。由此引发了我的代码的二次优化:超参数的调整.

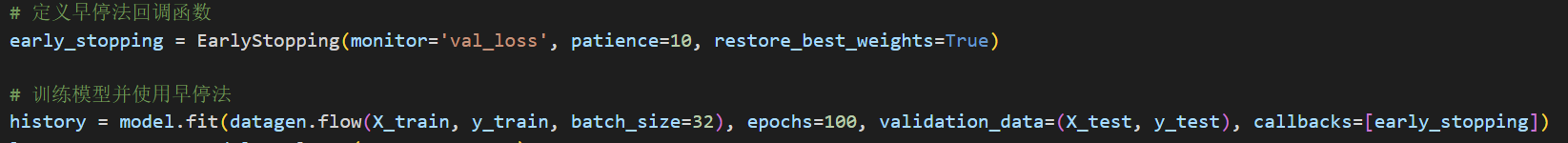
**超参数的调整:**

我将epoch值从30上调到50，结果如下，可见准确率显著增加，但是运行时间也大大增加了，而且**loss**值在不断下降，**val\_loss**趋于不变，说明可能存在过拟合了。

电脑屏幕截图

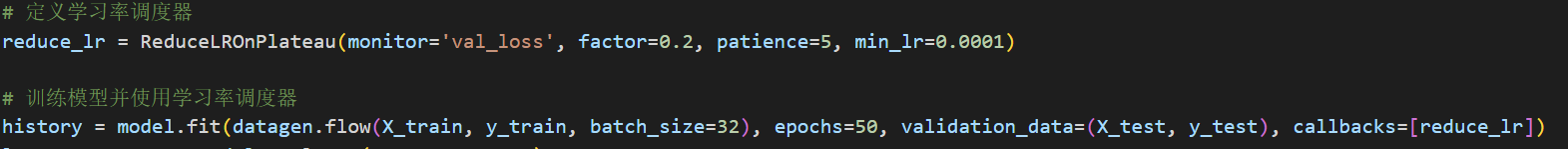
描述已自动生成

经过调查有一个方法叫做“早停法”。可以使用早停法来自动选择最佳的Epoch数量。早停法会在验证集性能不再提升时停止训练，从而避免过拟合。



最后发现该法法用处不大，因为epochs很大的话运行时间太久了，早停法确实能防止过拟合但到达过拟合的epochs值很大，与我的初衷相违背了，而且既然过拟合值epochs值很大那么我们也就不需要可以去考虑过拟合了。

经过调查学习调度器也能防止过拟合，并且动态调整学习率可以帮助模型更好地收敛



**改进训练策略：**

改进训练策略最直接的办法就是增加cnn模型的复杂程度我将原本的三层加至四层，最后的运行结果

电脑屏幕截图

中度可信度描述已自动生成

结果与开始大差不差，究其原因还是训练次数少的原因，epochs值继续增大测试准确率还会继续上升，因为loss值和val\_loss值还在不断下降中，说明说明网络仍在学习，加大epochs值之后差异会越来越明显。但是运行一次时间太长，就不再运行和测试了。

# 结果可视化

训练集和测试集准确率可视化

图表, 折线图

描述已自动生成

分类正确和错误的花卉图片示例：

图形用户界面, 应用程序, 网站

描述已自动生成

数据集地址链接：https://pan.baidu.com/s/1GN4rl84qNuS9cZo5sAHZdA

提取码：2003

将数据集下载解压放入工程文件夹里即可。

我所导入的所有库和包：

文本

描述已自动生成