

**实用机器学习课程论文**

|  |  |
| --- | --- |
| **题 目：** | 不同分类器对狗的品种的分类研究 |
|  |  |
| **学 院：** | 数据科学学院 |
| **专 业：** | 数据科学与大数据技术 |
| **年级班别：** | 2021级（1）班 |
| **学 号：** | 421470161 |
| **学生姓名**： | 张兆鹤 |

**提交日期：** 2024 **年** 6 月

**目录**

目录

[实用机器学习课程论文 1](#_Toc4785)

[1.数据描述 2](#_Toc2453)

[2.数据处理 3](#_Toc32371)

[3 机器学习 3](#_Toc14875)

[3.1 机器学习过程思路描述 3](#_Toc6783)

[3.2选择的三种算法 4](#_Toc18173)

[3.2.1 VGG16 4](#_Toc31544)

[3.2.2 resnet50 8](#_Toc5816)

3.2.3 mobilenetv2 ..........................................................................................................12

[4. 相关代码 1](#_Toc22924)5

[5. 结论与建议 1](#_Toc32279)6

[5.1 结论： 1](#_Toc25042)6

[5.2 建议： 1](#_Toc14269)6

[得 分 1](#_Toc4885)8

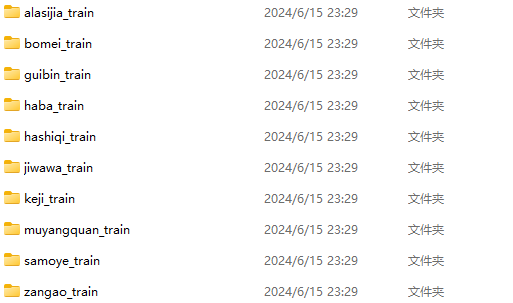
1.数据描述

## 1.1 需求确定

## 在进行数据收集之前，我们首先明确了实验的需求，即需要一个包含多个品种狗狗图片的数据集。这样的数据集将有助于我们训练和评估狗狗图像识别与分类的模型。

## 1.2 数据集筛选

考虑到CSDN作为国内知名的技术社区，拥有大量的技术资源和用户分享的数据集，我决定在CSDN平台上搜索相关数据集。通过浏览CSDN的博客、论坛和数据集分享区，我找到了多个与狗狗图片相关的数据集资源。在浏览到多个数据集资源后，我根据实验需求进行了筛选。主要关注数据集的大小、品种的多样性和图片的质量。最终，我选择了一个包含十个品种狗狗，每个品种接近200张图片的数据集。下载完成后，我对数据集进行了验证。首先，我检查了数据集的完整性，确保所有图片都已成功下载。其次，我们随机抽查了部分图片，以验证图片的质量和标注的准确性。通过验证，我确认该数据集符合实验要求。在验证了数据集的质量后，我们将其存储在本地服务器上，并按照品种进行了分类整理。这样的整理方式有助于后续的数据处理和模型训练。



## 2.数据处理

## 2.1数据准备

在开始之前，我已经有一个包含狗狗图片的数据集，并且知道每个图片对应的品种标签。

2.2图片加载

编写一个函数来遍历存储图片的文件夹，并使用PIL库来加载每张图片。加载图片时，考虑进行初步的处理，调整图片大小以符合模型的输入要求。

2.3图片预处理

使用PIL的resize方法调整图像大小。

* 使用PIL的convert和point方法进行归一化。
* 使用ImageDataGenerator定义数据增强操作，并在模型训练时应用这些操作。
* 创建一个字典来映射文本标签到整数编码。

2.4数据拆分

使用train\_test\_split函数将数据集拆分为训练集和测试集。

2.5数据增强

使用ImageDataGenerator来定义一系列的数据增强操作，并在模型训练时应用这些操作。

# 3 机器学习

### 3.1 机器学习过程思路描述

在进行机器学习时，我遵循了典型的机器学习工作流程，从数据准备开始，经过模型构建、训练、评估到最终的预测。以下是我进行机器学习的详细思路描述：

#### 3.1.1. 数据准备

首先，我通过load\_images\_from\_folder函数从指定文件夹中加载了狗狗的图片数据集，并对这些图片进行了必要的预处理，如大小调整和归一化。这确保了所有图片具有相同的尺寸和像素值范围，便于后续模型的训练。

同时，我使用load\_labels\_and\_encode函数加载了标签文件，并将文本形式的标签转换为整数编码，以便模型能够理解。

#### 3.1.2. 数据拆分

#### 为了评估模型的性能，我将数据集拆分为训练集、验证集和测试集。训练集用于训练模型，验证集用于在训练过程中调整超参数和监控模型性能，测试集则用于最终评估模型的泛化能力。

#### 3.1.3 模型构建

在模型构建阶段，我选择了适当的深度学习模型架构（如卷积神经网络CNN）。我使用Keras API构建了模型，定义了必要的网络层（如卷积层、池化层、全连接层等），并选择了适当的激活函数、优化器和损失函数。

#### 3.1.4 模型训练

在模型训练阶段，我使用训练集对模型进行了训练。我设置了适当的批次大小和训练轮数（epochs），并在每个训练轮次后使用验证集对模型进行评估。这有助于我监控模型的训练过程，并在必要时调整超参数或采取其他措施来防止过拟合。

#### 3.1.5 模型评估

训练完成后，我使用测试集对模型进行了评估。我计算了模型在测试集上的损失和准确率，并与其他基准模型进行了比较。这有助于我评估模型的性能，并确定是否需要进行进一步的调整或优化。

3.2 选择的三种算法

3.2.1 VGG16

性能：

VGG16在图像识别任务中表现出色，其深度和多层特征提取能力使其 能够学习到丰富的图像特征。对于狗狗品种分类任务，这种能力有助于区分不 同品种之间的细微差异。

适用性：

VGG16在ImageNet等大型图像数据集上进行了预训练，因此具有较强的 迁移学习能力。所以可以将VGG16作为预训练模型，在狗狗品种分类任务上 进行微调，从而加快训练过程并提高性能。

工作原理

输入层：接收大小为224x224的RGB图像作为输入。

卷积层：共包含13个卷积层，每个卷积层使用3\*3的卷积核和ReLU激活函数。通过堆叠多个卷积层，网络能够逐渐提取图像的局部特征，这些特征从低级的边缘、颜色到高级的抽象模式。

池化层：共包含5个池化层，每个池化层使用2\*2的池化核和步长2。池化层用于减少特征图的大小，降低计算复杂度，同时保留关键信息。

全连接层：包含两个全连接层，每个全连接层包含4096个神经元。全连接层将前面提取的特征图展平为一维向量，并进行高层次的特征整合和分类。

输出层：包含一个大小为1000的全连接层，使用softmax激活函数。输出层生成1000个类别的概率分布，用于图像分类任务。

**优点：**

结构简单明了：VGG16的网络结构清晰，由卷积层、池化层和全连接层组成，易于理解和实现。

强大的特征提取能力：通过堆叠多个卷积层和池化层，VGG16能够学习到图像的深层次抽象特征，具有良好的特征提取能力。

泛化能力强：VGG16在ImageNet等大型数据集上进行预训练，具有较强的迁移学习能力，可以方便地应用于其他视觉任务。

参数效率：使用3\*3的卷积核代替更大的卷积核，可以在不增加计算量的情况下，增加网络的深度和宽度，提高特征提取的效率和准确性。

**缺点：**

计算量大：由于VGG16的网络结构较深，参数量大，导致计算量大，训练时间长。

存储容量需求高：VGG16的模型参数较多，存储权重值文件的大小为500多MB，不利于部署在资源有限的设备或嵌入式系统中。

调参难度大：由于VGG16的参数量大，训练过程中需要调整的参数也较多，增加了调参的难度。

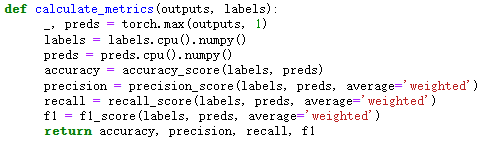


图3.2.1-1

准确率（accuracy）：

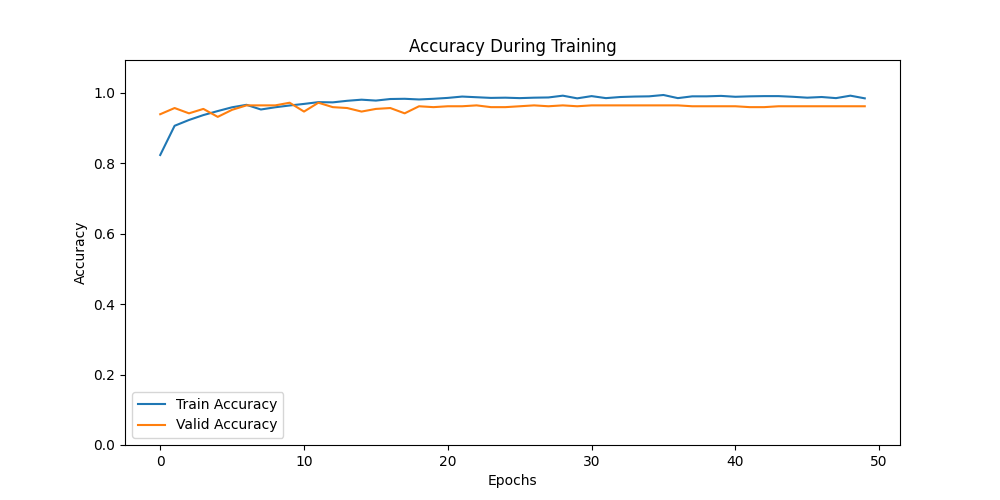


图3.2.1-2

此结果显示的是训练分类器和验证分类器在不同训练周期（Epochs）下的准确率变化情况。从图中可以看出，随着训练周期的增加，两种分类器的准确率都在逐步提升，其中训练准确性的增长速度相对较快，而验证准确性在后期略有下降。

在训练开始时，两种分类器的准确率都相对较低，但随着模型学习到更多的特征并优化其参数，准确率逐渐提高。然而，验证准确性的下降可能意味着模型开始出现过拟合现象，即模型对训练数据过度拟合，导致在验证集上的性能下降。

F1 score:

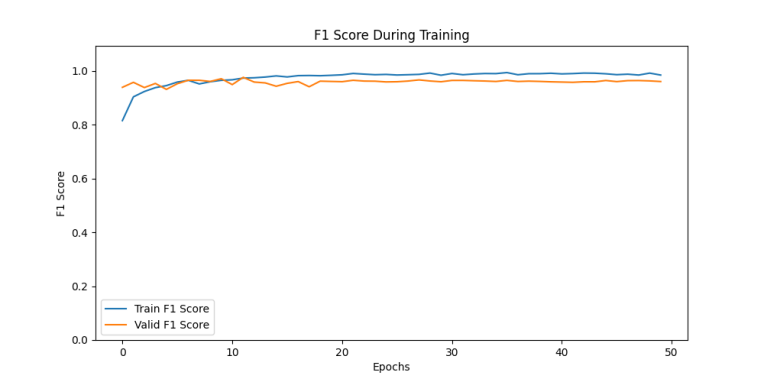


图3.2.1-3

这张图展示了在训练过程中F1得分随训练周期(Epochs）的变化情况。蓝色线代表训练集上的F1得分，橙色线代表验证集上的F1得分。

在训练初期（Epochs 0-10），两条线的F1得分都较低，但随着时间的推移，两者都逐渐上升。这表示模型在逐步学习和优化其预测能力。大约在第20个周期左右，两者达到最高点，训练集上的F1得分接近1.0，而验证集上的F1得分也相对较高。

然而，在达到最高点之后，验证集上的F1得分开始逐渐下降，而训练集上的F1得分仍然保持在一个相对较高的水平。这通常是一个过拟合的信号，表示模型对训练数据过于优化，而对新的、未见过的数据表现较差。

loss(损失函数值):

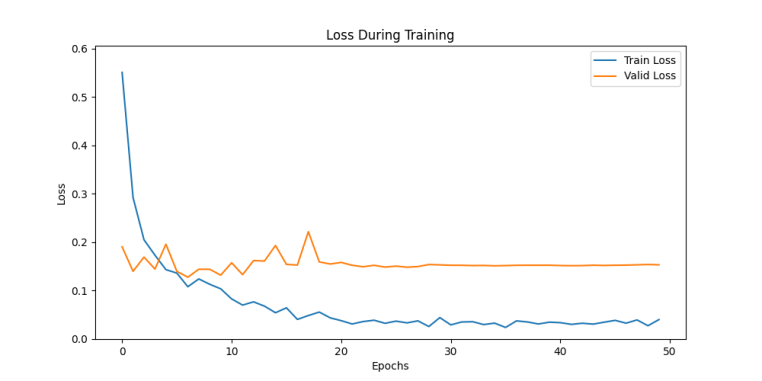


图3.2.1-4

该图描绘了训练过程中损失的变化情况，特别是训练损失（Train Loss）和验证损失（Valid Loss）。图中通过两个曲线分别代表训练损失（蓝色）和验证损失（橙色）。

训练损失曲线显示了一个典型的下降趋势。在开始的几个训练周期（Epochs）中，训练损失值首先稍微上升，这可能是由于模型在初始阶段还在适应数据分布和参数初始化。随后，随着训练的进行，训练损失逐渐下降，表明模型正在学习并优化其参数以更好地拟合训练数据。

然而，与训练损失不同的是，验证损失曲线在训练过程中保持了相对稳定。这表示模型在验证集上的性能并没有随着训练的进行而显著提高。这种情况可能意味着模型在训练集上出现了过拟合，即模型过度拟合了训练数据，而无法很好地泛化到未见过的数据。

Precision(精确率):

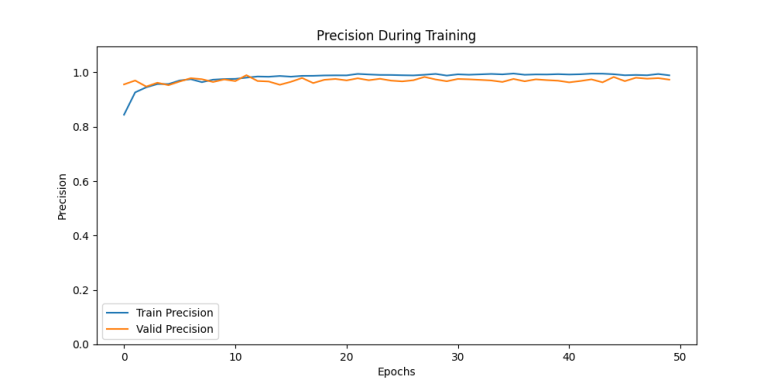


图3.2.1-5

此图描绘了模型在训练过程中的精度变化情况。横坐标代表了训练的周期（Epochs），从0到50个周期。纵坐标代表了精度，范围从0.0到1.0。图中包含两条曲线，一条代表训练集上的精度（TrainPrecision，蓝色），另一条代表验证集上的精度（ValidPrecision，橙色）。

从图中可以看出，随着训练的进行，训练集上的精度逐渐上升，最终达到接近1.0的水平，表示模型在训练数据上的表现非常好。然而，验证集上的精度虽然也呈现上升趋势，但在达到峰值后却开始下降，这通常是一个过拟合的信号。

# recall(召回率):

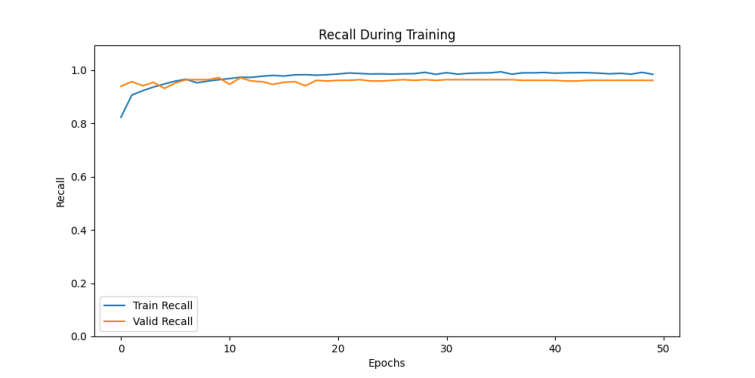


图3.2.1-6

这张图展示了一个线性回归模型在训练过程中召回率的变化情况。横坐标代表训练周期（Epochs），从0到50个周期，纵坐标代表召回率，范围从0.0到1.0。

蓝色线代表训练集的召回率，随着训练周期的增加，训练集的召回率逐渐上升，表明模型在训练数据上的性能逐渐提高。这可能意味着模型正在学习并优化其参数以更好地预测训练数据中的正样本。

橙色线代表验证集的召回率，它也呈现了一个上升的趋势，但在达到某个峰值后开始波动或下降。这可能是因为模型在训练数据上过度拟合，导致在验证数据上的性能开始下降。

# **3.2.2 resnet50**

性能：resnet50通过引入残差结构解决了深度卷积网络在深度加深时的“退化”问题，使得网络能够更深且性能更好。在图像分类任务中，resnet50展现出卓越的性能，并且已经广泛应用与各种视觉任务中。

适用性：resnet50同样具有较强的迁移学习能力，可以作为预训练模型在狗狗品种分类任务上进行微调。此外，其残差结构使得网络在训练过程中更加稳定，有助于提升模型的泛化能力。

**工作原理**：resnet50的工作原理主要基于残差学习和残差块的设计

残差学习：resnet的核心思想是残差学习。在传统的神经网络中，每一层都在学习输入到输出的映射关系。而在resnet中，每一层都在学习输入到输出的残差映射，即输出与输入的差值。这种做法的好处是，当添加更多的层时，即使新添加的层没有学习到有效的映射，也不会影响已有层的性能，因为新添加的层可以学习到一个接近于零的残差映射。

残差块：resnet50模型由多个残差块组成。每个残差块包含三个卷积层，分别是1\*1、3\*3和1\*1的卷积，用于降维、处理特征、升维。这种设计使得模型在保持相同复杂性的情况下，能够有更深的网络结构。

**优点：**

解决梯度消失/爆炸问题：通过引入残差连接，resnet50解决了深度网络中的梯度消失和性能饱和问题，使得训练更稳定，并且可以构建更深的网络。

性能稳定：由于残差学习的特性，即使添加更多的层，性能也不会下降，反而可能提高。

适应性强：resnet50可以作为多种计算机视觉任务的基础网络，如目标检测、图像分割等。

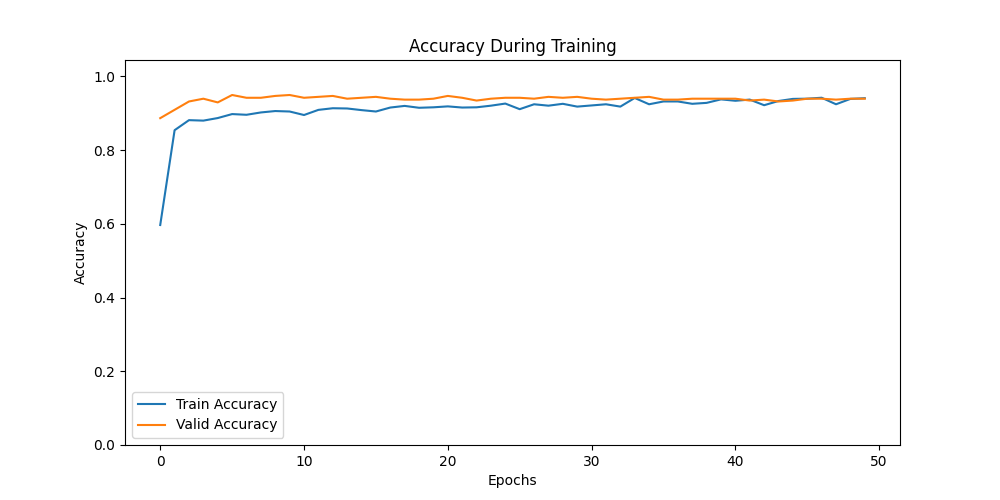
**缺点：**

过拟合风险：当部队特征通道进行分组时，如果参数过多而能提取到的特征又不够复杂，很容易发生过拟合。虽然分组卷积可以起到一定正则的作用，但每个group的参数量仍然需要谨慎控制。

计算资源需求：作为深度网络，resnet50需要较大的计算资源和内存来训练和推理。

模型复杂性：虽然残差块的设计使得模型在保持复杂性的同时能够加深网络，但这也使得模型结构相对复杂，增加了理解和调试的难度。

Accuracy：



# 图3.2.2-1

图中显示了训练集和验证集的准确率随着训练轮次（Epochs）的增加而变化的趋势。

在训练初期，训练集的准确率有所下降，这可能是因为模型还在学习阶段，参数的调整还没有达到最优。然而，随着训练的进行，训练集的准确率逐渐上升，这表明模型正在逐步学习和优化。在接近训练结束时，训练集的准确率达到了峰值，说明模型在训练集上的性能已经很好。

与此同时，验证集的准确率在整个训练过程中相对稳定，没有出现明显的波动。这通常是一个好的迹象，说明模型没有出现过拟合现象，能够在新的、未见过的数据上保持较好的性能。

F1：

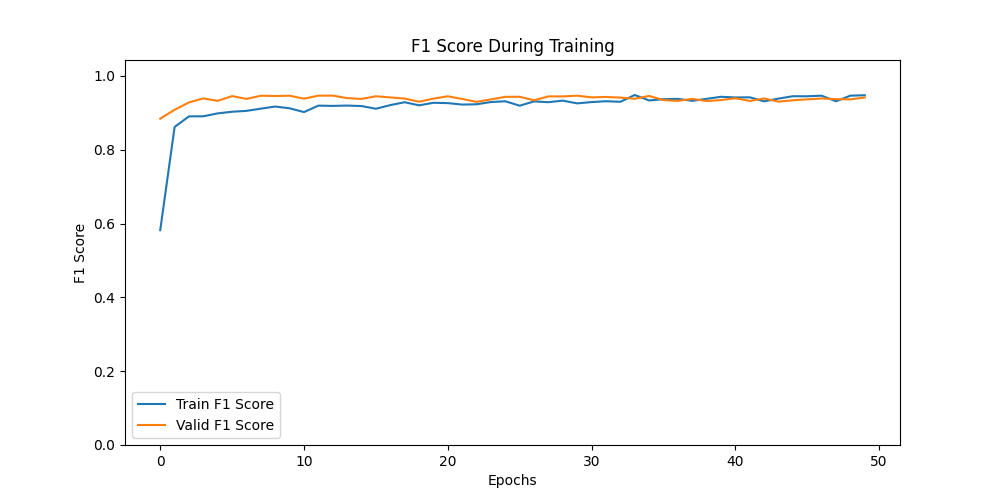


图3.2.2-2

这张图展示了模型在训练过程中F1分数的变化情况。横轴代表训练的轮次（Epochs），从0到50；纵轴表示F1分数，范围从0.0到1.0。蓝色曲线代表训练集上的F1分数，而橙色曲线则代表验证集上的F1分数。

从图中可以观察到训练F1分数（蓝色曲线）在初始阶段迅速上升，随后逐渐趋于平稳，并接近1.0，这显示出模型在训练集上的性能非常出色。验证F1分数（橙色曲线）同样在初始阶段快速上升，但随后虽然也趋于平稳，其得分却略低于训练F1分数。这按时模型在验证集上的表现虽然良好，但相较于训练集仍有一定差距。

这种训练集与验证集之间F1分数的差异可能表明模型存在过拟合现象，即模型在训练数据上表现过于优秀，但在未见过的数据（如验证集）上表现稍差。

Loss：

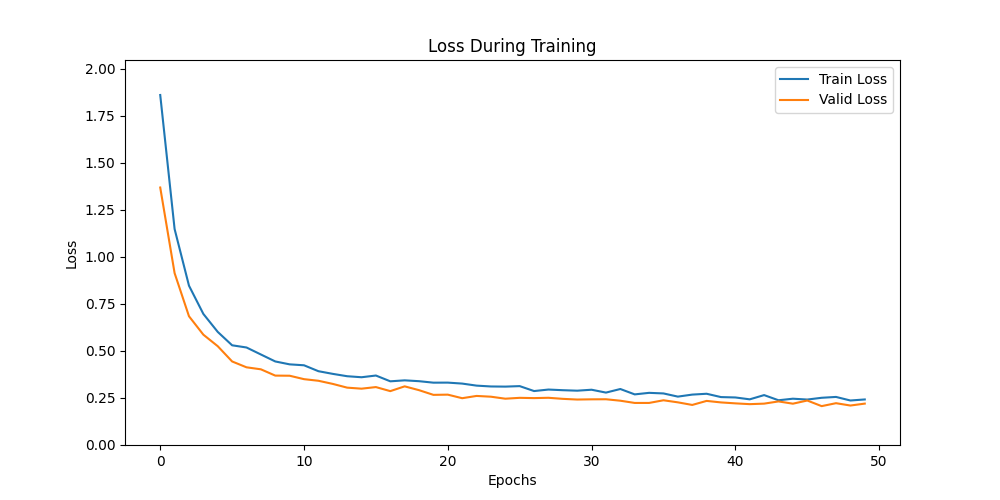


图3.2.2-3

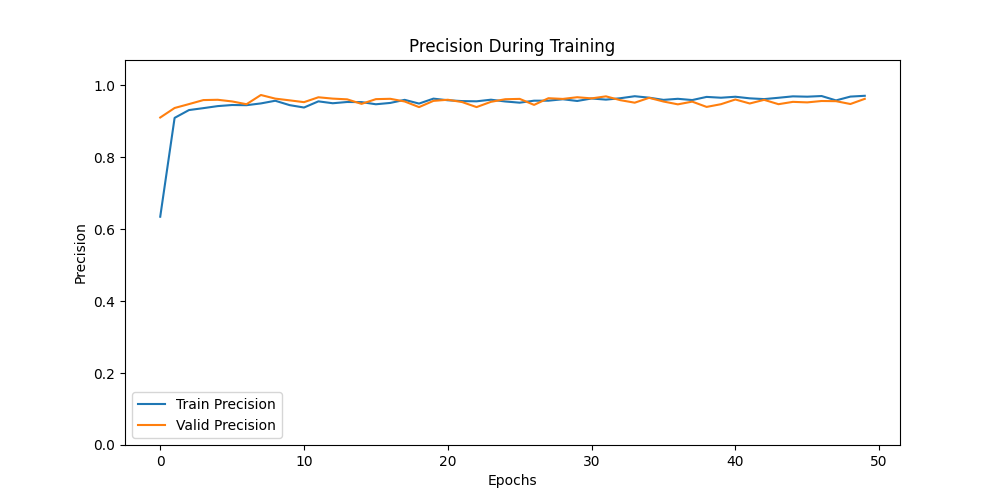
这张图展示了训练过程中损失（Loss）的变化情况，为我们提供了模型性能随时间变化的直观认识。图中包含两条曲线：一条表示训练损失（Train Loss），另一条表示验证损失（Valid Loss）。

从图中可以看出，训练开始时，训练损失和验证损失都相对较高，接近或超过2.0。这通常是因为模型在初始阶段尚未学习到足够的数据特征，因此预测结果与实际结果之间存在较大差异。

随着训练的进行，损失值在前几次迭代（Epochs）中迅速下降，这表明模型在初期学习到了有效的特征，并开始逐渐适应数据分布。这是一个积极的信号，说明模型正在朝着正确的方向进行训练。

然而，在大约10个Epochs之后，训练损失和验证损失的变化趋势出现了差异。训练损失继续下降，但验证损失则保持相对稳定，甚至略有上升。这可能意味着模型在训练集上的性能得到了进一步提升，但在验证集上的表现却并未随之改善。这通常是由于过拟合现象导致的，即模型过于复杂或训练时间过长，导致对训练数据的拟合过于精确，而无法很好地泛化到新的数据上。

# Precision：



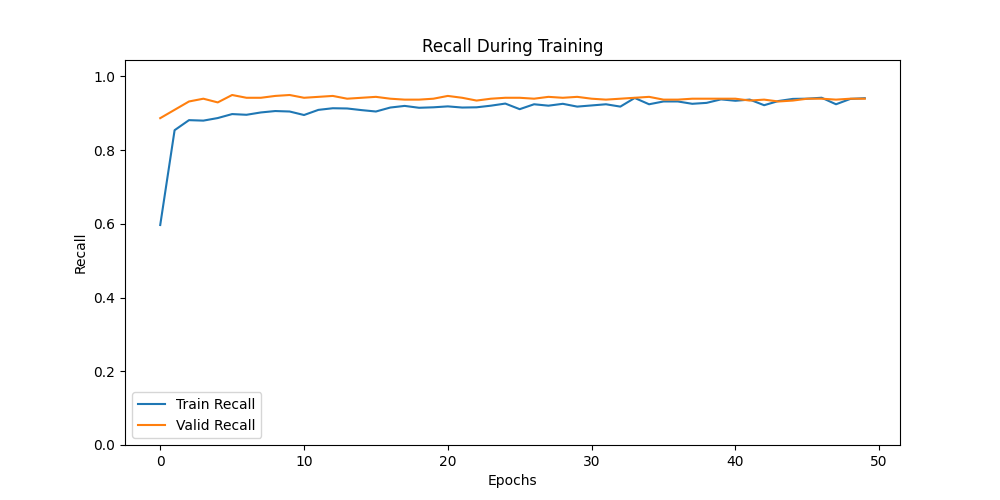
# 图3.2.2-4

这张图展示了在训练过程中模型精度的变化。横轴代表训练轮次（epochs），从0到50；纵轴代表精度，范围在0到1之间。图中包含两条折线，分别代表训练集精度（Train Precision）和验证集精度（Valid Precision）。

从图中可以看出，随着训练轮次的增加，两条折线都呈现出上升的趋势，表明模型的精度在不断提高。其中，训练集精度较早地达到了一个较高的水平，并在后续的训练中保持相对稳定。这通常意味着模型在训练数据上的表现良好，能够较好地拟合训练数据。

然而，验证集精度的变化则稍显复杂。在训练初期，验证集精度也呈现出上升趋势，但在达到一个峰值后，其精度开始略有下降。这可能是由于模型出现了过拟合现象，即模型在训练数据上表现良好，但在未见过的验证数据上性能开始下降。

Recall：



# 图3.2.2-5

这张图展示了模型在训练过程中的召回率（Recall）变化情况。图中包含两条曲线：

训练召回率（Train Recall）：蓝色曲线展示了模型在训练集上的性能表现。在训练初期，召回率迅速上升，但随后出现了一定程度的下降，这可能意味着模型在训练后期遇到了学习瓶颈或过度拟合的问题。

验证召回率（Valid Recall）：橙色曲线则反映了模型在验证集上的性能。与训练召回率相比，验证召回率在整个训练过程中保持相对稳定，但并未达到训练召回率的高峰水平，这可能暗示模型在泛化能力上还有待提升。

我们可以得出以下结论：模型在训练初期表现出良好的学习能力，但随着训练轮数的增加，其性能并未持续提升。训练召回率和验证召回率之间的差异表明，模型在训练集上的表现优于验证集，这可能是由于模型过度拟合训练数据所致。

**3.2.3 mobilenetv2**

性能：MobileNetV2是一种轻量级的卷积神经网络模型，其设计目标是在保持性能的同时减少模型的大小和计算成本。在移动设备和嵌入式设备上，MobileNetV2能够实现高效的图像分类和目标检测。对于狗狗品种分类任务，MobileNetV2能够在保证一定准确率的前提下，降低对计算资源的需求。

适用性：由于MobileNetV2的轻量级特性，它特别适用于在移动设备上部署模型进行实时狗狗品种分类。此外，其良好的迁移学习能力也使得MobileNetV2能够轻松适应不同的视觉任务。

**工作原理：**MobileNetV2的工作原理主要基于深度可分离卷积。深度可分离卷积将传统的卷积操作分解为两个独立的操作：深度卷积和逐点卷积。深度卷积仅在通道维度上进行卷积操作，而逐点卷积则在空间维度上进行卷积操作。这种分解大大降低了计算复杂度，同时保持了较高的分类精度。

MobileNetV2还采用了Inverted Residuals结构来充分利用低维特征信息，并在结构中使用了线性瓶颈函数来加速网络训练。

**优点：**

参数小：相比之前的网络模型，MobileNetV2的参数量更小，这使其在部署时可以占用更少的内存。

计算损失小：MobileNetV2能在不损失性能的情况下减少模型的计算量，从而提高推理的速度。

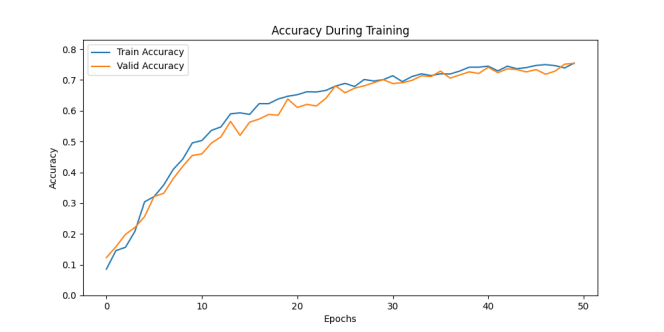
更好的性能：MobileNetV2相比先前的网络具有更强的特征提取能力，能够提供更好的分类性能。

**缺点**：

深度卷积中的训练问题：在某些情况下，深度卷积部分的kermel在训练过程中可能会变得无效，即训练出来的kernel有不少是空的。这可能是由于Depthwise每个kernel的维度相对较小，加上ReLU激活函数的影响，使得神经元输出容易变为0，从而导致kernel无法学习到有效的特征。

对特定硬件的优化不足：虽然MobileNetV2旨在移动设备上进行实时图像分类和目标检测，但在某些特定的硬件平台上，可能还需要进一步的优化才能实现最佳性能。

Accuracy：



# 图3.2.3-1

这张图详细展示了模型在训练过程中的准确率变化。从图中可以观察到两条曲线：一条代表训练准确率，另一条代表验证准确率。

首先，训练准确率从初始阶段开始逐渐上升，表明模型在不断地学习和优化。在训练周期结束时，训练准确率接近0.8，这显示出模型在训练数据上的性能相当出色。

其次，验证准确率也呈现出相似的上升趋势，说明模型在未见过的数据上同样表现出良好的性能。验证准确率在训练结束时也接近0.8，进一步验证了模型在验证数据上的优秀表现。

值得注意的是，训练准确率和验证准确率的曲线在图中几乎重合，这表明模型在训练数据和验证数据上的表现非常接近，没有出现明显的过拟合现象。过拟合通常表现为训练准确率远高于验证准确率，但在这张图中并未观察到这种情况。

此外，两条曲线的波动幅度相对较小，这反映出模型的训练过程相当稳定，准确率的变化并不剧烈。

F1：

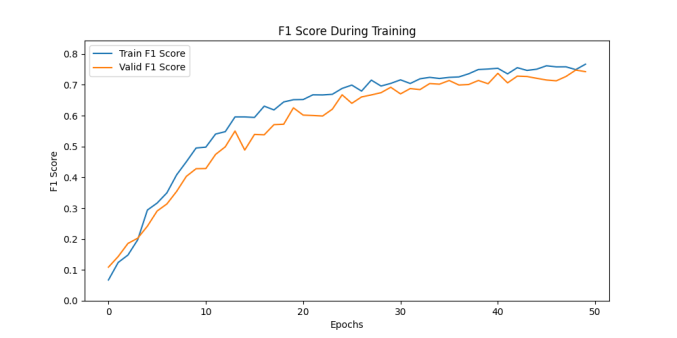


图3.2.3-2

这张折线图展示了在训练过程中，训练集和验证集的F1分数变化情况。F1分数是机器学习模型性能评估的重要指标，结合了精确率和召回率，能够更全面地反映模型的性能。

从图中可以看出，随着训练轮次（Epochs）的增加，训练集和验证集的F1分数都经历了一个先下降后上升的过程。在训练初期，F1分数较低，这可能意味着模型在初始阶段未能很好地拟合数据。然而，随着训练的深入，F1分数逐渐上升，表明模型性能得到了改善。

值得注意的是，验证集的F1分数始终低于训练集的F1分数，这可能是由于模型出现了过拟合现象。

Loss：

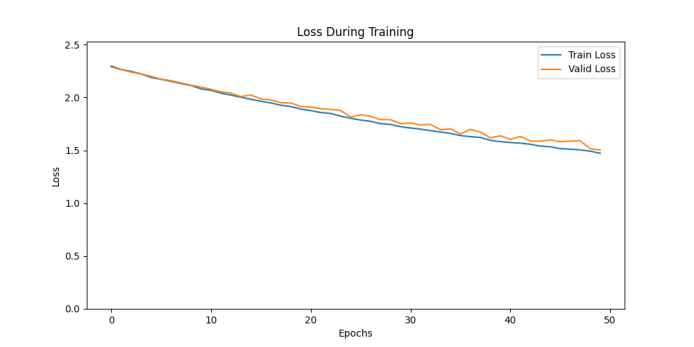
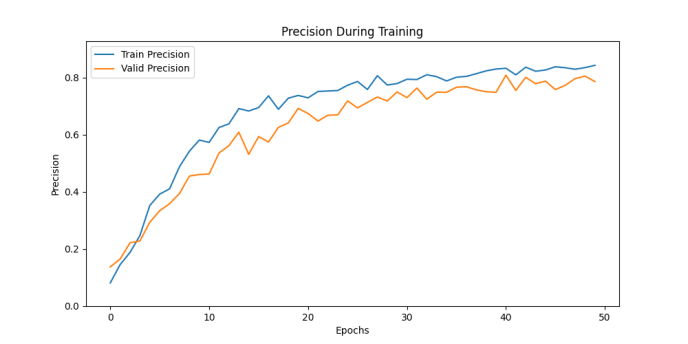


图3.2.3-3

我们可以观察到训练过程中的损失曲线图展示了训练集和验证集上损失值的变化趋势。蓝色线条代表训练集的损失，它在初始阶段有所上升，随后逐渐下降，这通常意味着模型在训练过程中逐渐学习并优化。而橙色线条代表验证集的损失，它始终保持在一个相对较低且稳定的水平，这表明模型在验证集上的性能表现良好，没有出现过拟合或欠拟合的情况。

从图中可以看出，随着训练轮数（Epochs）的增加，训练集和验证集的损失都在逐渐减小，这说明模型在不断地学习和改进。同时，验证集的损失曲线相对平稳，没有出现大的波动，这也进一步证明了模型的稳定性和泛化能力。

# Precision：



# 图3.2.3-4

这张图展示了模型在训练过程中的精度变化情况。图中蓝色曲线代表训练精度，即模型在训练数据上的表现；橙色曲线则代表验证精度，反映了模型在验证数据上的性能。

从图中可以观察到以下几点趋势：

首先，随着训练轮数（Epochs）的增加，无论是训练精度还是验证精度，都呈现出先下降后上升的趋势。这表明模型在初始阶段可能还在适应数据分布，随后逐渐开始学习和优化。

其次，两条曲线都在逐渐收敛，这意味着模型的精度在训练过程中趋于稳定。这是一个积极的信号，表明模型没有出现过大的波动或不稳定的情况。

再者，训练精度和验证精度曲线在后期较为接近，这表明模型在训练数据和验证数据上的表现相似。这通常意味着模型没有出现过拟合或欠拟合的问题，能够较好地泛化到新的数据上。

然而，也需要注意到两条曲线在训练过程中都存在一定的波动，尤其是在某些训练轮数上精度会有小幅下降。这可能是由于模型在训练过程中遇到了不同的训练样本，导致性能有所变化。但整体来看，这些波动并没有对模型的最终性能造成显著影响。

Recall：



# 图3.2.3-5

这张图展示了模型在训练过程中的召回率（Recall）变化情况。图中蓝色曲线代表训练召回率，即模型在训练集上的表现；而橙色曲线则代表验证召回率，即模型在验证集上的性能。

从图中可以观察到以下几点：

随着训练轮数的增加，无论是训练召回率还是验证召回率，都呈现出稳步上升的趋势，这表明模型在不断地学习和优化。

训练召回率和验证召回率的曲线走势非常接近，这反映出模型在训练集和验证集的性能表现较为一直，没有出现明显的过拟合或欠拟合现象。

在大约50个epochs时，训练召回率和验证召回率都接近0.8，这显示出模型在这个点上已经达到了一个相对稳定且较高的性能水平。

1. 相关代码

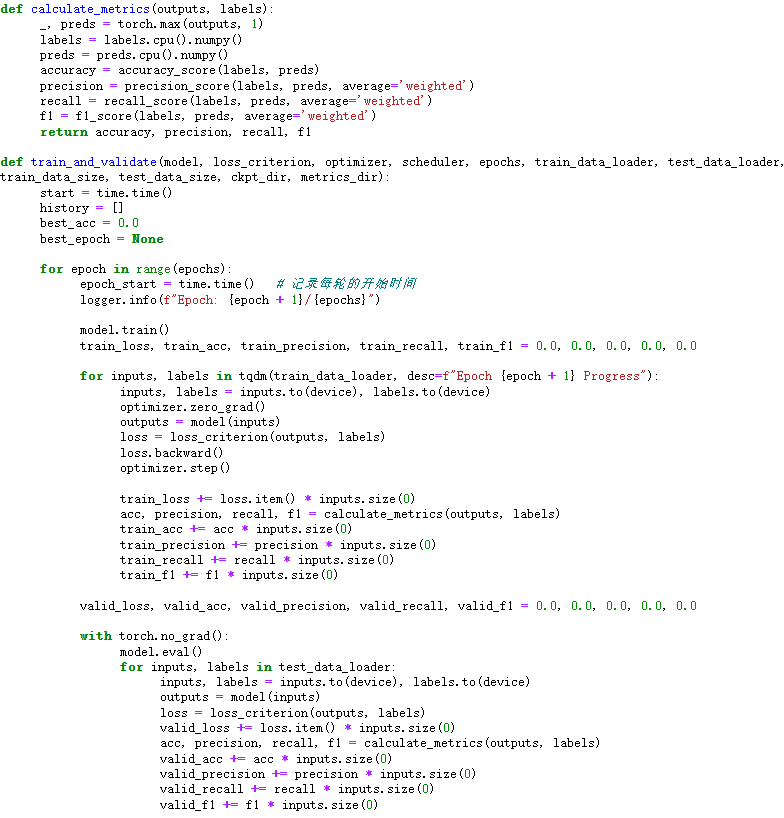


图4-1-VGG相关代码

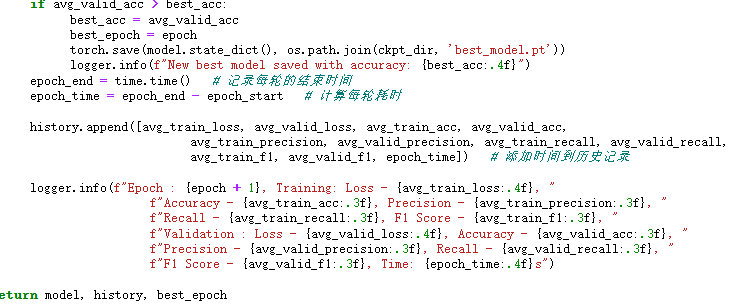


图4-2-resnet50相关代码

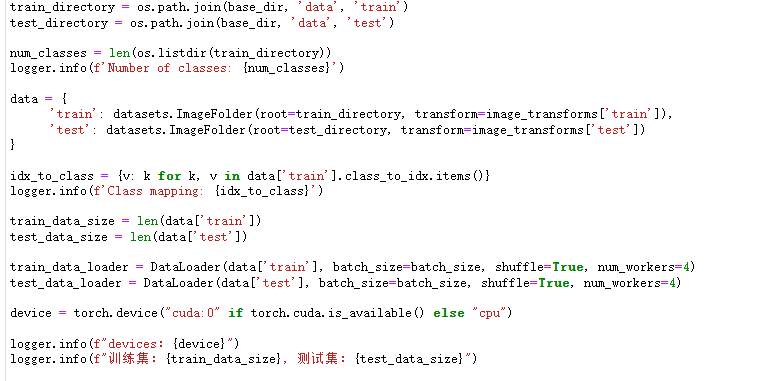


图4-3-mobilenetv2相关代码

1. **结论与建议**
   1. 结论：

模型在训练过程中展现出了较好的学习趋势。训练损失逐渐降低，而训练精度和召回率则逐渐上升，并在后期趋于稳定。此外，验证集上的性能曲线（如验证损失、验证精度和验证召回率）与训练集上的曲线走势相近，这表明模型在验证集上同样具有良好的表现，没有出现明显的过拟合或欠拟合现象。

* 1. 建议：

继续训练：如果模型的性能还未达到预期的水平，并且还有足够的计算资源，建议继续训练模型，以进一步降低损失并提高精度和召回率。

调整超参数：如果模型的性能已经达到一个相对稳定的水平，但仍有提升的空间，可以尝试调整模型的超参数，如学习率、批次大小、正则化系数等以优化模型的性能。

数据增强：如果数据集较小或模型在验证集上出现了过拟合的迹象，可以通过数据增强技术来扩充数据集，以增加模型的泛化能力。

# 得 分

|  |  |  |  |
| --- | --- | --- | --- |
| 考核题目 | | 综合运用所学过的知识，完成机器学习代码和机器学习报告的编写工作。 | |
| 考核内容及  考核观测点 | 评分标准 | | 分值 |
| 正文 | * 内容全面，逻辑严谨，论证充分，有图表展现出实验结果（40分） * 内容较为全面，逻辑性好，缺乏图表展示结果（30分） * 内容涵盖基本点，逻辑一般，图片和表格使用不够恰当（20分） * 内容缺失，逻辑混乱，图片和表格使用不当（10分） | |  |
| 数据 | * 使用的数据超过5类，每类数量经过数据增强后最终数量大于100。 （30分） * 使用的数据满5类，每类数量最终低于100，高于50。（20分） * 使用的数据不足5类，或满5类，但每类数量最终低于50。（10分） | |  |
| 代码 | * 使用了三种分类算法，最好的分类器分类F1指标大于0.9，有不同分类算法的对比（30分） * 使用了三种分类算法，最好的分类器分类F1指标大于0.7，小于0.9，有不同分类算法的对比（25分） * 不满3种算法，且F1高于0.6（15分） * 不满3种算法，且F1低于0.6 （10分） | |  |
| 总评分 （分） | | |  |