

**2024年《机器学习》**

**工程报告**

**（个人版）**



**课 程：** 机器学习

**姓 名：** 朱炳旭

**学 号：** 2022217593

**完成时间：** 2024/05/17

|  |  |  |  |
| --- | --- | --- | --- |
| **“机器学习-大作业”评分细则** | | | |
| **成绩等级** | **具体表现** | **教师评分** | |
| 优秀（100-90] | 报告撰写优秀，题目本身难度大，工作量饱满；Notebook代码清晰准确，实验与可视化优秀 | □ |  |
| 良好（90-80] | 报告撰写良好，题目本身难度较大，工作量较多；Notebook代码较好，实验与可视化良好 | □ |  |
| 中等（80-70] | 报告撰写中等，题目本身难度一般，工作量达标；Notebook代码一般，实验与可视化达标 | □ |  |
| 及格（70-60] | 能完成基本要求，工作量较少，非Notebook代码 | □ |  |
| 不及格（<60） | 未达最低要求或抄袭线上线下资源 | □ |  |

教师签名：

一． 自己所作工作的简介

通过PyTorch框架中的ResNet18模型，对血细胞进行分类任务。ResNet18是一个深度学习模型，常用于图像分类任务。

数据集包含 12,500 张血细胞增强图像 （JPEG） 以及随附的细胞类型标签 （CSV）。4 种不同细胞类型中的每一种大约有 3,000 张图像，分为 4 个不同的文件夹（根据细胞类型）。细胞类型为嗜酸性粒细胞、淋巴细胞、单核细胞和中性粒细胞。该数据集附带一个附加数据集，其中包含原始 410 个图像（预增强）以及两个附加子类型标签（WBC 与 WBC），以及这 410 个图像中每个单元格的边界框（JPEG + XML 元数据）。更具体地说，文件夹“dataset-master”包含 410 张带有子类型标签和边界框 （JPEG + XML） 的血细胞图像，而文件夹“dataset2-master”包含 2,500 张增强图像以及 4 个附加子类型标签 （JPEG + CSV）。4 个类中每个类大约有 3,000 个增强图像，而文件夹“dataset-master”中每个类有 88、33、21 和 207 个图像。

数据集链接：[Blood Cell Images (kaggle.com)](https://www.kaggle.com/datasets/paultimothymooney/blood-cells/data)

二、研究背景与意义

血液疾病的诊断通常涉及识别和表征患者血液样本。检测和分类血细胞亚型的自动化方法具有重要的医学应用。

通过计算机技术实现的血液疾病诊断方法通常是基于图像处理和机器学习技术。这些方法利用数字图像分析来识别和分类血细胞，从而帮助医生进行诊断。这个项目主要涉及医学图像处理和机器学习领域。在这个领域，主要研究方法包括图像分割、特征提取、特征选择、分类器设计等。医学图像处理技术用于处理血液样本的数字图像，提取其中的有用信息；机器学习算法则用于训练分类模型，实现自动化的血液细胞分类和诊断。

在这种方法中，首先需要获取患者血液样本的数字图像，可以通过显微镜或数字相机等设备获取。然后，利用图像处理技术对血细胞进行分割和特征提取，以提取血细胞的形态特征、颜色特征等信息。

利用机器学习算法对提取的特征进行训练和分类，从而建立血细胞分类模型。这些模型可以根据不同的特征将血细胞分为不同的类别，如红细胞、白细胞、血小板等，甚至可以进一步分类不同的白细胞亚型等等。

最后，医生可以通过这些模型对患者血液样本的图像进行自动化分类和诊断，从而快速准确地获取患者的血液疾病信息。这种基于计算机技术的方法可以提高诊断效率，减少人为误差，并为医生提供更多的辅助信息，帮助他们做出更准确的诊断和治疗方案。

其中可能出现的问题：

1. 数据标注困难：对于大量的血液细胞图像数据，需要进行准确的标注和分类，但这一过程需要耗费大量的人力和时间。
2. 模型泛化能力差：由于血液细胞的形态和颜色特征变化较大，训练的分类模型在未知数据上的泛化能力可能较差。
3. 实时性要求高：在临床诊断中，需要快速准确地对血液样本进行分析和诊断，因此算法的实时性要求较高。

解决方案：

1. 数据增强技术：通过数据增强技术，可以扩充有限的标注数据集，提高模型的泛化能力。
2. 迁移学习：利用已有的大规模数据集进行预训练，再通过微调等方法对血液细胞数据进行训练，提高模型的性能。

三、模型方法

使用ResNet18模型和权重的神经网络模型。首先导入了必要的库和ResNet18\_Weights。然后定义了一个名为Net的类，继承自nn.Module。在初始化函数中，首先实例化了一个ResNet18模型，并根据参数决定是否冻结模型。接着获取了模型最后一层全连接层的输入特征数，并将全连接层替换为一个输出维度为4的线性层。接着定义了一个冻结模型参数的方法freeze\_resnet，通过将模型参数的requires\_grad属性设置为False来实现。最后定义了forward方法，将输入x传入ResNet模型中并返回输出。

四、系统设计

**模块简单介绍**

**数据转换模块**：

定义了一个函数transform\_data()，根据不同的数据集类型（“TRAIN”、“TEST\_SIMPLE"或"TEST”），使用transforms.Compose()函数创建了一系列图像转换操作，例如调整大小、随机水平翻转、随机旋转、转换为张量和归一化操作。

**数据加载模块：**

使用ImageFolder类和DataLoader类加载数据集。根据给定的数据目录和转换操作，创建了ImageFolder对象，并使用DataLoader对象对数据进行批处理和洗牌。

**图像可视化模块：**

使用tensor\_to\_array()函数将张量转换为数组，并使用show\_grid()函数显示一批图像。遍历数据加载器的键，获取下一个批次的图像和标签，将图像转换为网格，并使用show\_grid()函数显示。

**模型定义模块：**

使用ResNet18\_Weights.DEFAULT加载预训练的ResNet-18模型的权重。定义了一个自定义的模型Net，其中包含了预训练的ResNet-18模型和一个线性层作为分类器。还定义了一个freeze\_resnet()方法用于冻结ResNet-18模型的参数。

**训练循环模块：**

定义了一个训练循环函数train()，用于训练模型。在每个训练周期中，计算训练损失、验证损失和分类指标，并保存它们。如果验证损失较小，则保存模型的权重。

**训练结果可视化模块：**

使用plt.subplots()函数创建一个包含两个子图的画布，分别用于显示训练损失和验证损失的变化情况，以及准确率和F1-macro分数的变化情况。使用ax[0].plot()和ax[1].plot()函数绘制折线图。最后使用plt.tight\_layout()函数调整图像布局，并使用plt.show()函数显示图像。

**评估循环模块：**

定义了一个评估循环函数evaluate()，用于评估模型在测试集上的性能。计算测试集上的损失、准确率和F1-macro分数，并返回这些指标。

**混淆矩阵可视化模块：**

使用confusion\_matrix()函数计算混淆矩阵，并使用heatmap函数绘制混淆矩阵的热力图。

**核心代码介绍：**

**数据可视化:**  
展示图像，并绘制柱状图。

**图像展示部分：**

通过随机选择每个类别的一个图像来展示。使用plt.subplots()函数创建一个包含4个子图的画布。对于每个子图，使用axs[i].imshow()函数显示图像，axs[i].set\_title()设置子图的标题，axs[i].axis('off')关闭子图的坐标轴显示。最后使用plt.tight\_layout()函数调整子图布局，使其更加美观。

**柱状图部分：**

使用plt.bar()函数绘制柱状图。横轴为类别名称，纵轴为对应的图像数量。使用plt.text()函数在每个柱状图上显示对应的数值。最后使用plt.xlabel()、plt.ylabel()和plt.title()函数设置横轴、纵轴和标题的标签，使用plt.tight\_layout()函数调整图像布局。

**训练：**

**训练循环：**

初始化一个空字典callbackers，用于保存训练过程中的损失和指标。

循环执行指定次数的训练轮次（num\_epochs）。

在每个轮次中，首先将训练损失（train\_loss）和验证损失（valid\_loss）初始化为0。

将模型设置为训练模式（model.train()）。

遍历训练数据加载器（train\_loader），将数据传输到指定设备（device），并进行前向传播计算损失。

优化器（optimizer）清零梯度，反向传播计算梯度，更新模型参数。

计算累积的训练损失。

将模型设置为评估模式（model.eval()）。

在验证数据加载器（valid\_loader）上进行模型评估，计算验证损失，并将预测结果和真实标签转换为numpy格式。

计算准确率（acc）和F1分数（f1\_macro）。

打印当前轮次的训练损失、验证损失、准确率和F1分数。

将训练过程中的损失和指标保存到callbackers字典中。

如果当前轮次的验证损失低于之前记录的最小验证损失（valid\_loss\_min），则保存模型参数到文件"ResNet.pt"。

返回保存的训练过程指标callbackers。

**可视化训练过程中的统计数据：**

创建了一个包含两个子图的画布，分别用于显示训练损失和验证损失，并显示分类准确率和F1-macro分数。

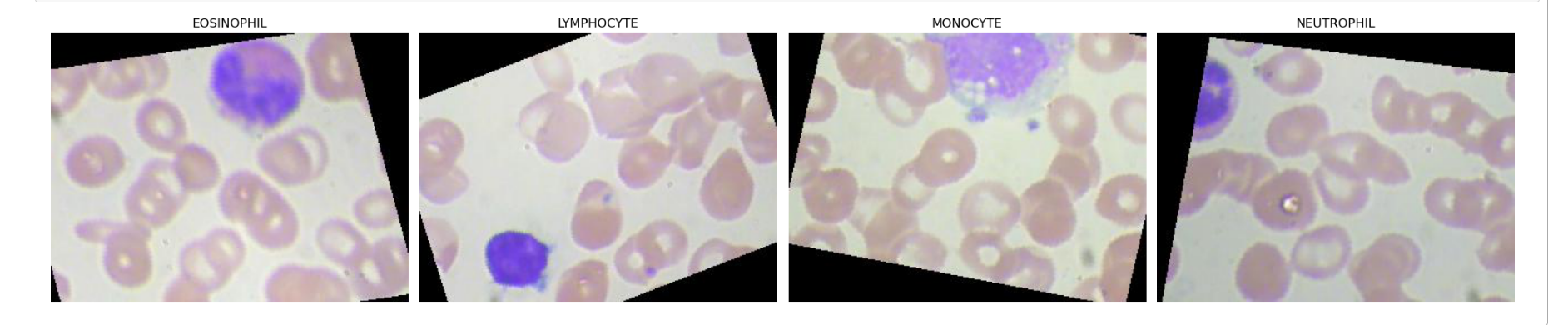
在第一个子图中，通过使用ax[0].plot()函数，将训练损失和验证损失的数据绘制成折线图。train\_stats['train\_loss']和train\_stats['valid\_loss']是存储了训练过程中损失数据的列表。ax[0].set\_title()函数用于设置子图的标题，ax[0].legend()函数用于添加图例。

在第二个子图中，通过使用ax[1].plot()函数，将准确率和F1-macro分数的数据绘制成折线图。train\_stats['accuracy']和train\_stats['f1\_macro']是存储了训练过程中准确率和F1-macro分数的列表。同样地，ax[1].set\_title()和ax[1].legend()函数用于设置标题和图例。

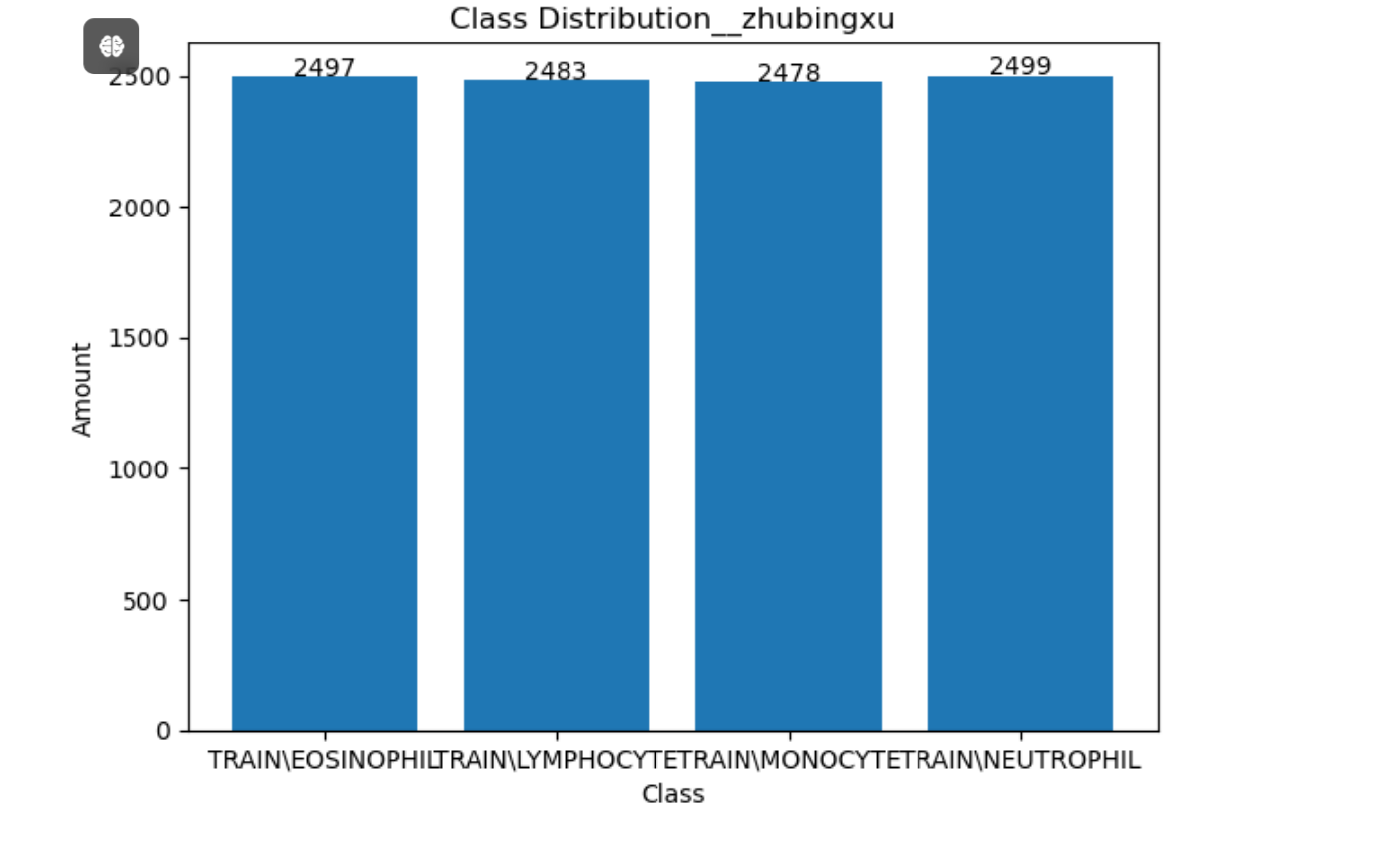
plt.tight\_layout()函数用于自动调整子图的布局，使其更加美观。plt.show()函数用于显示图形。

五．实验结果分析、对比和讨论

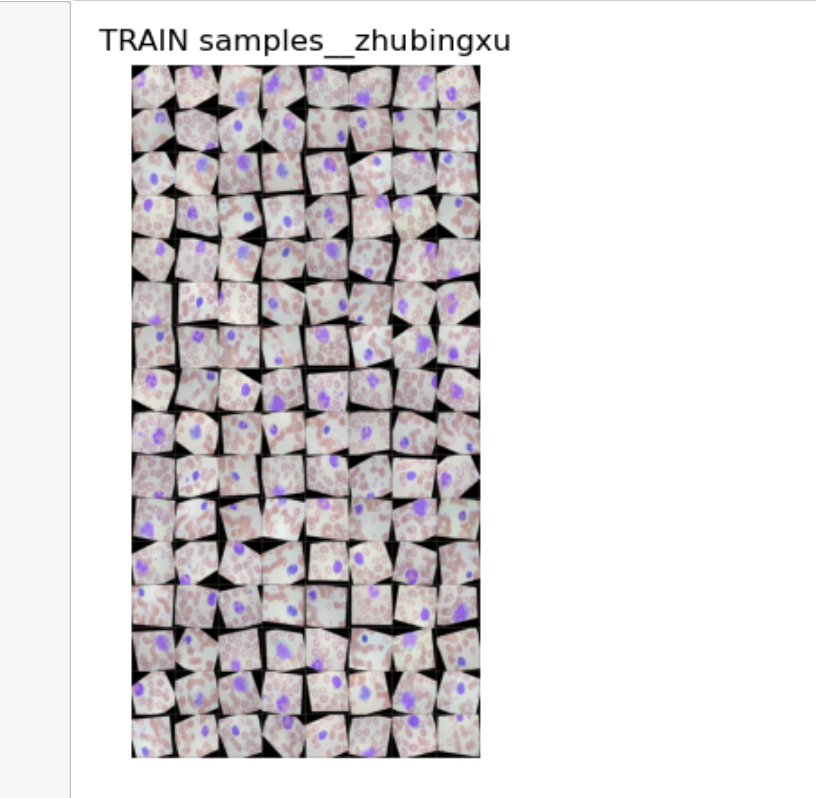
图像展示



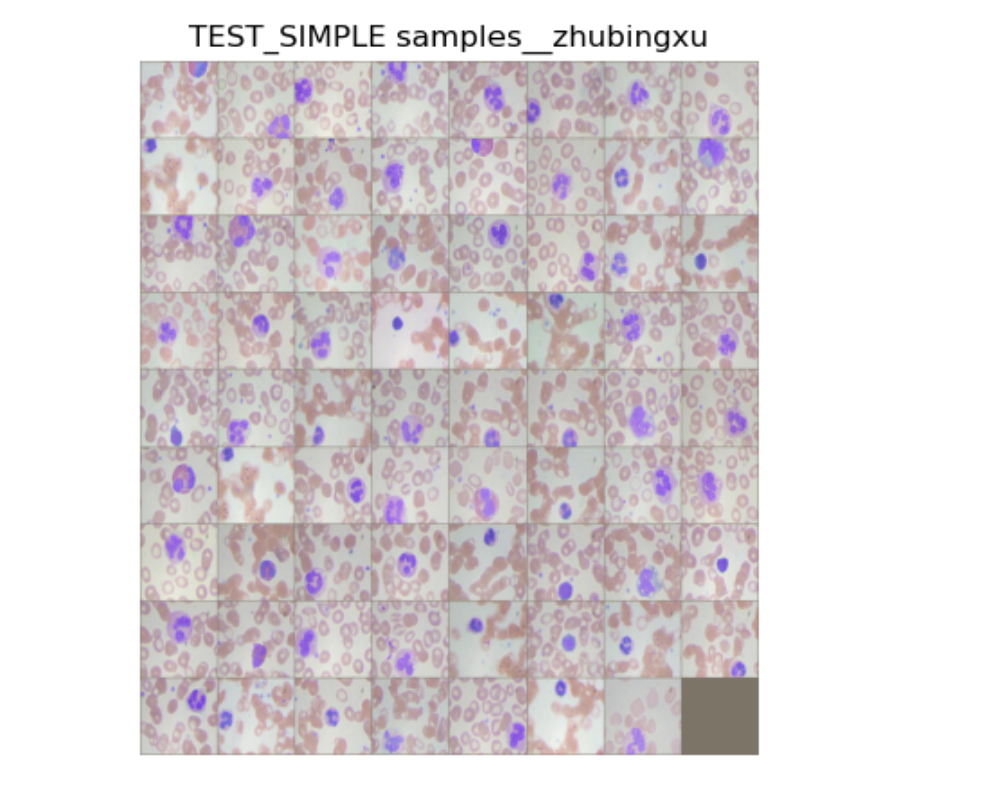
类分布



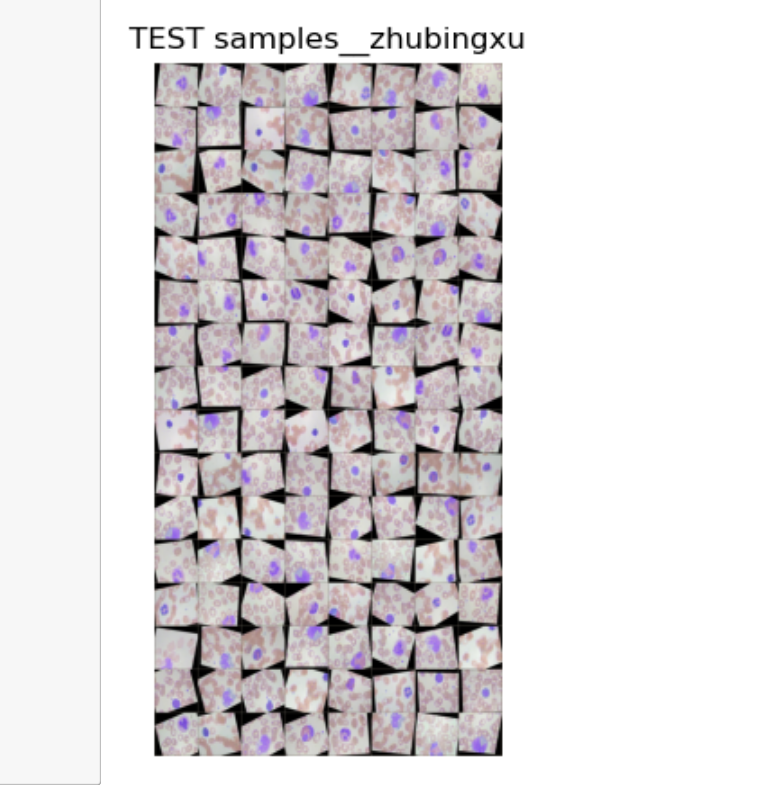
训练样本集合



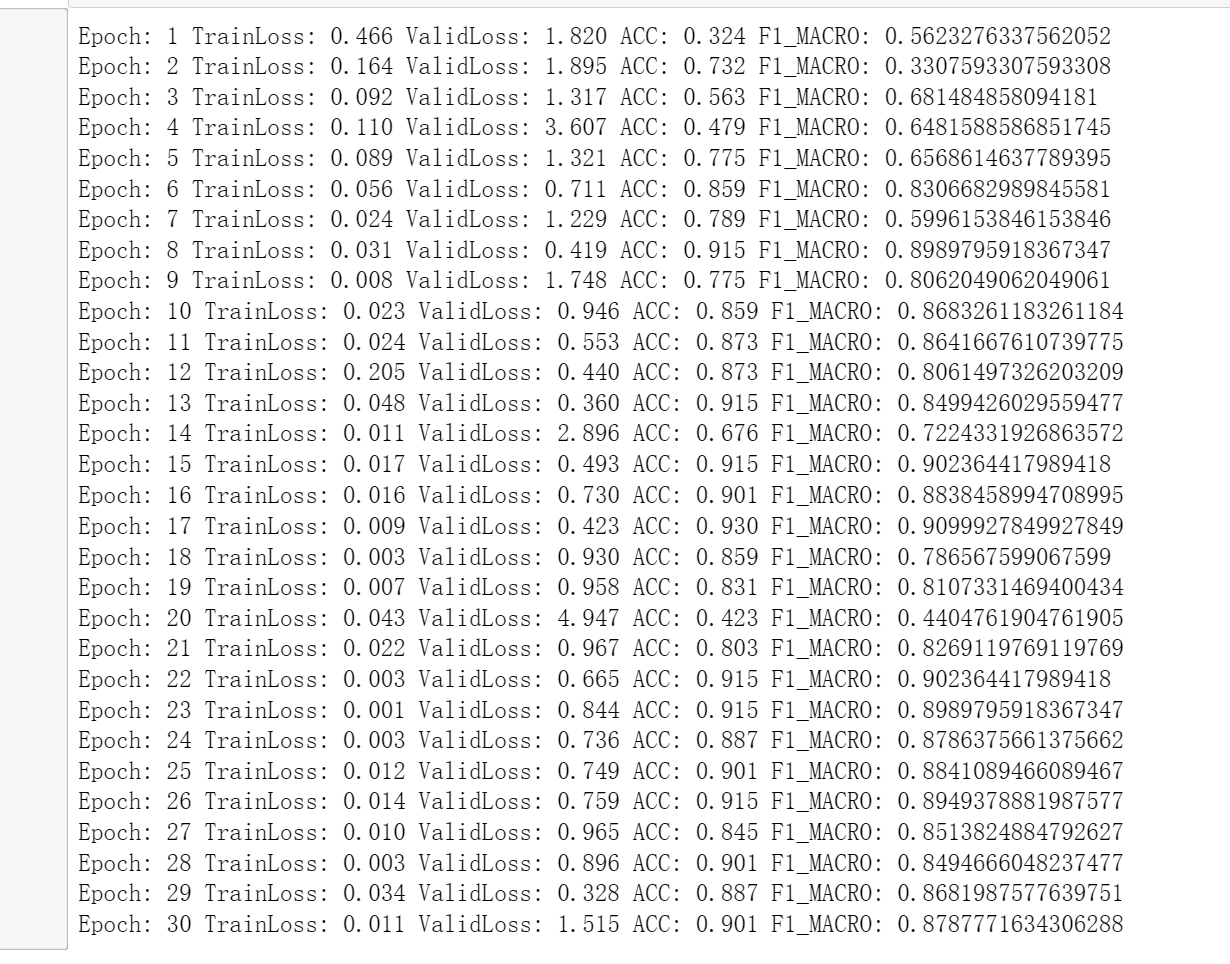
简单测试样本集合



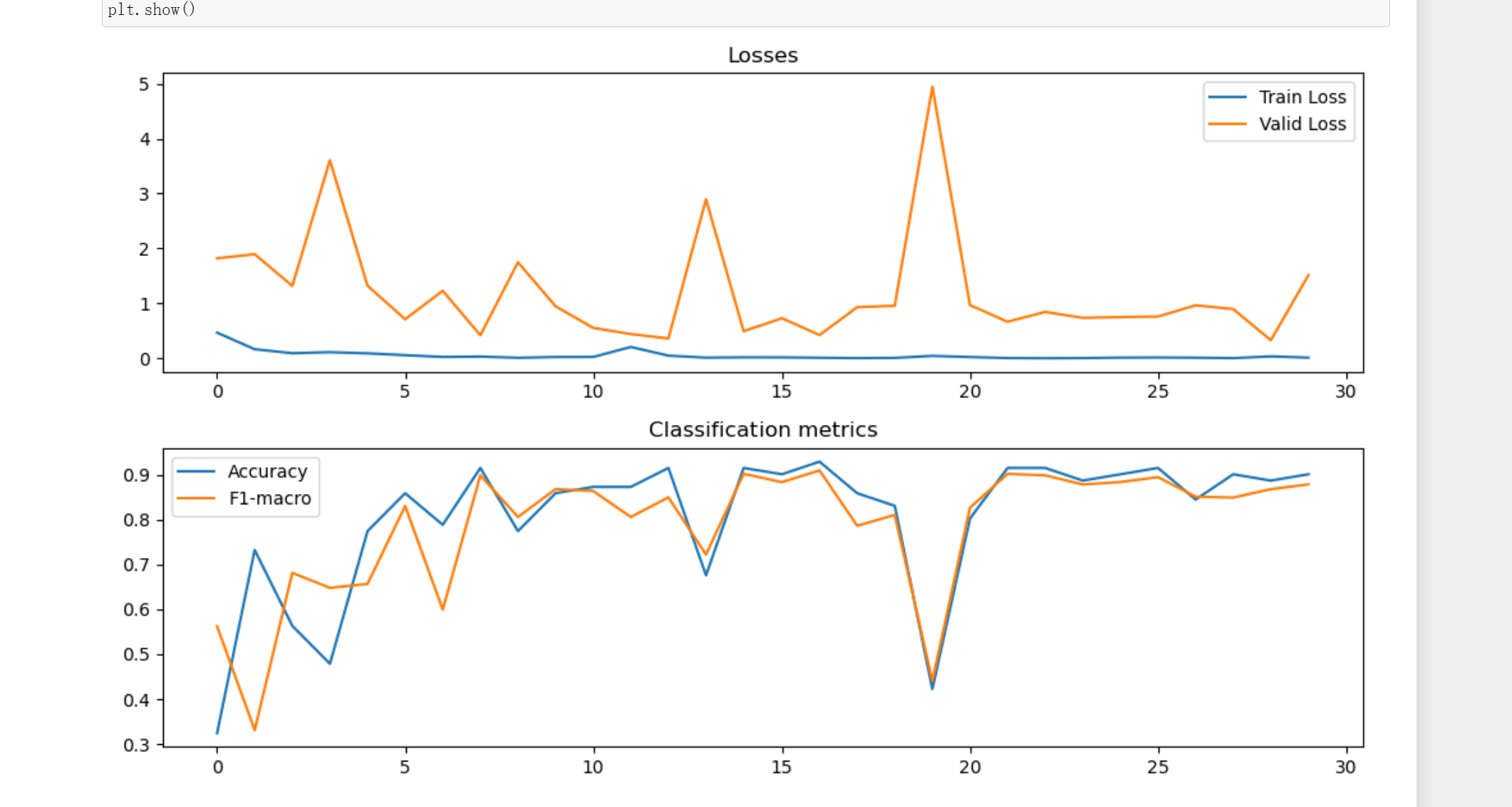
测试样本集合



训练损失 验证损失 准确率 宏平均



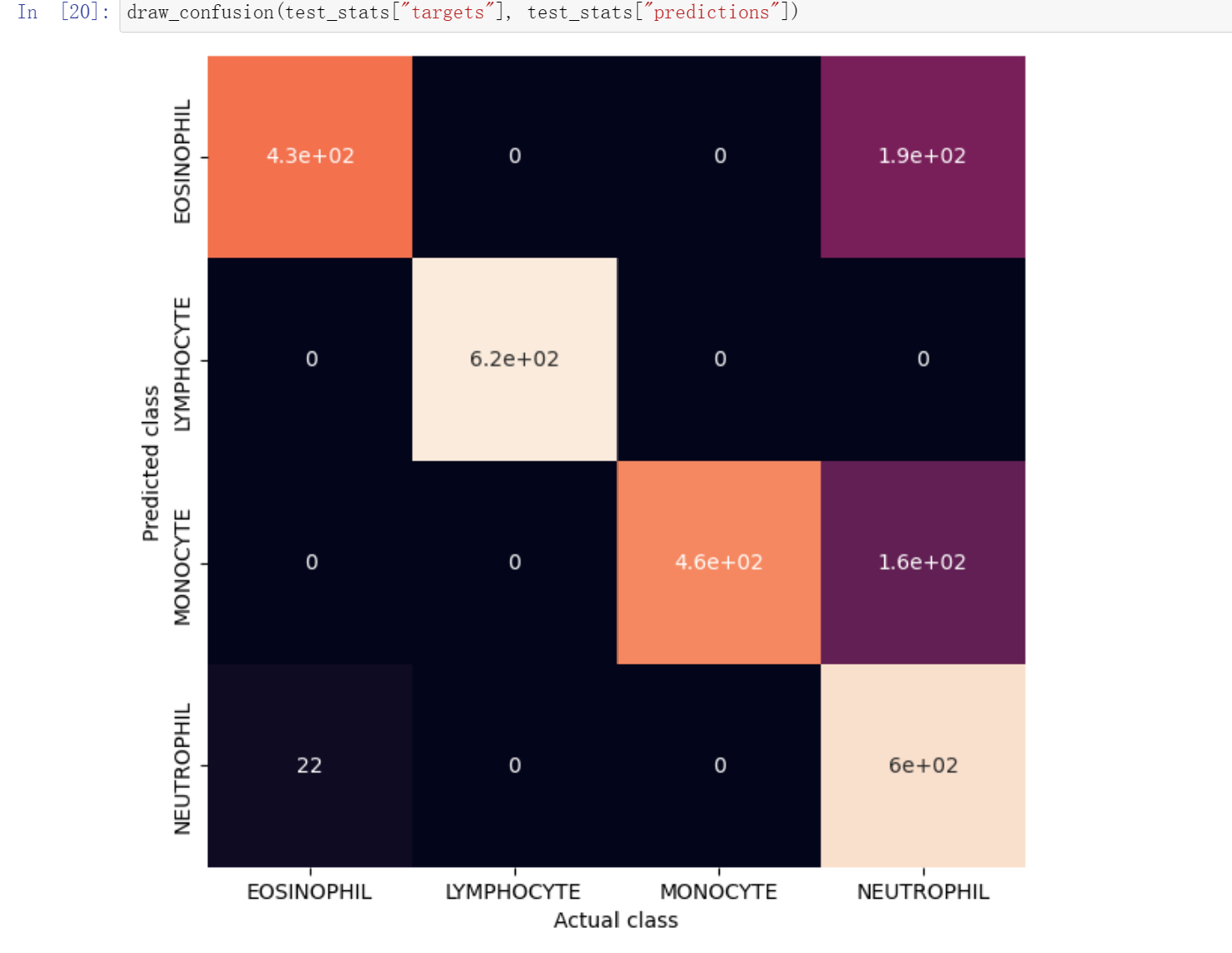
损失 分类指标



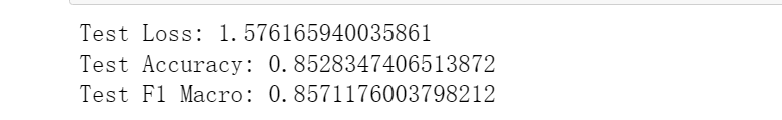
实际类

纵轴：嗜酸性粒细胞 淋巴细胞 单核细胞 中性粒细胞

横轴：嗜酸性粒细胞 淋巴细胞 单核细胞 中性粒细胞



测试损失 测试准确率 测试F1宏平均



结果分析：

测试损失（Test Loss）为1.341，准确率（Test Accuracy）为0.876，F1 Macro分数（Test F1 Macro）为0.879。这表示模型在测试集上的性能相对较好，但仍有改进的空间。

通过混淆矩阵可以看出，模型在某些类别上的预测效果较好，而在其他类别上的预测效果较差。这可能是由于数据集中某些类别的样本数量较少，导致模型在这些类别上的学习效果不佳。

改进方向：

数据增强：

可以尝试使用更多的数据增强技术来增加数据集的多样性，例如随机裁剪、颜色变换等。这有助于提高模型的泛化能力。

模型调优：

可以尝试调整模型的超参数，例如学习率、优化器、损失函数等，以提高模型的性能。也可以尝试使用更大的模型或其他预训练的模型来提高模型的表达能力。

类别不平衡处理：

如果数据集中某些类别的样本数量较少，可以使用类别平衡技术，例如过采样、欠采样或类别加权，以平衡各类别的样本数量，从而提高模型在少数类别上的学习效果。

更多的训练和调试：

可以增加训练周期，尝试不同的超参数组合，并进行更多的实验和调试，以找到最佳的模型配置。

六．对本门课的感想、意见和建议

**感想：**

机器学习是一个非常有趣和前沿的领域，可以帮助我理解和利用数据来做出预测和决策。

学习机器学习需要一定的数学、统计和编程基础，通过实践和项目可以加深理解和掌握技能。

机器学习领域不断发展和演变，需要持续学习和跟进最新的技术和算法。

**建议：**

  课程内容应该既包括机器学习的理论知识，也要注重实践操作，让学生能够真正掌握算法原理并应用到实际问题中。

  引入项目驱动学习的方法，让学生通过实际项目的完成来深入理解机器学习算法和技术，提升实践能力。

提供专门的实践课程或实验室课程，让学生有机会动手实践，独立完成机器学习项目，培养他们的实际操作能力。

更新课程内容，关注最新的机器学习技术和趋势，让学生了解最前沿的发展，保持教育内容的时效性。