《人工智能导论》大作业

任务名称：不良内容检测与识别

完成组号：10

小组人员：彭远翔、孟想、王博文、林瑞洲

完成时间：2024/6/20

1．任务目标

基于暴力图像检测数据集，构建一个二分类模型，实现对于数据集进行不良内容分类与检测，在合理的运行时间下达到较高的分类准确率。

2．具体内容

（1）实施方案

基于所提供的 dataset.py, model.py 与 train.py 三个子程序实现接口类子程序 classify.py，以实现一个使用预训练的暴力检测分类器进行图像分类和测试准确率的功能。

（2）核心代码分析

以下对提供代码作简要分析

1）dataset.py:

1. CustomDataset 类用于加载图像数据。__init__方法加载对应的训练、验证或测试数据集。; __len__方法返回图像的数量; __getitem__方法加载图像并返回转换后的图像数据和对应的标签。
2. CustomDataModule 类用于组织训练、验证和测试数据加载器。__init__方法初始化批次大小和工作进程数量；setup 方法根据需要加载不同数据集；train_dataloader、val_dataloader 和 test_dataloader 方法分别返回用于训练、验证和测试的 DataLoader 对象。

2）model.py

ViolenceClassifier 类用于暴力检测分类。在__init__方法中，加载了预训练的模型并进行初始化；forward 方法传递输入 x，并得到模型的输出；configure_optimizers 方法定义了优化器，并设置了学习率；training_step 方法、validation_step 方法和 test_step 方法分别对训练过程、验证过程与测试过程进行了定义。

3）train.py

该程序设置了训练参数、数据模块、模型检查点与日志记录器，并调用 trainer.fit()开始训练。

以下将基于我们小组所实现代码进行分析。

1）classify.py

1. ViolenceDataset 类用于加载图像数据并返回图像及对应标签。__init__方法用于初始化，并接收文件。四个属性 folder_path, transfor, images, labels 分别用于存储数据集文件夹路径、数据预处理的变换函数、加载的图像数据与图像对应的标签，最后调用了 _load_data 来加载图像与标签；_load_data 方法遍历并加载文件夹下的图像文件。先根据文件名结尾将所需的标签提取出来，并将其存储在 self.labels 列表中，再打开图像文件，将图像转换为 RGB 格式并储存在 self.image 列表中；__len__方法返回图像的数量；__getitem__方法获取指定索引 idx 处的图像和对应标签。先从 self.images 和 self.labels 中取出索引处的图像和标签，使用可

能定义的 self.transform 变换函数对图像进行变换操作，最后返回二者。

2. ViolenceClass 类用于加载预训练的暴力分类器模型并实现图像分类和测试准确率功能。__init__方法进行了初始化，并接收一个模型路径 model_path。先指定设备为 CPU 并禁用 GPU 来确保模型在 CPU 上运行，然后使用 ViolenceClassifier.load_from_checkpoint 加载预训练的暴力分类器模型，并将其移到 CPU 设备上，最后将模型设置为评估模式；classify 方法对输入的图像张量分类。先接收图像张量 imgs 并移动到指定设备，再使用 torch.no_grad()禁止在推理阶段所不必须的梯度计算，然后使用加载的模型 self.VioCL 进行前向传播并获取模型输出，将该输出通过 sigmoid 函数转换为概率，并根据阈值 (0.5) 进行二分类预测，最后将预测结果和概率转换为 CPU 上的 python 列表并返回；classify_folder 方法对指定文件夹中的图像进行分类。先定义图像预处理的转换操作，包括调整大小、转换为张量和归一化，然后将转换后的张量存储到 imgs 列表中，并记录文件名到 img_names 列表中，再将所有的图像张量堆叠成一个张量 imgs_tensor，然后调用 classify 方法对图像进行分类，最后返回分类结果和概率，另外，对于读取图像错误和未加载任何成功图像两种特殊情况，会将错误信息打印并返回空值；test_accuracy 方法评估模型在指定文件夹中的图像上的分类准确率。先定义与 classify_folder 方法中相同的图像预处理转换操作，再创建 ViolenceDataset 对象，传入文件夹路径和预处理转换对象并加载数据集，不打乱其顺序并设置适当的批次大小，然后遍历数据加载器，将每个批次的图像和标签推送到设备上，并使用 classify 方法获取预测结果，最后统计预测正确的样本数，计算并打印测试集的准确率。
3. 最后，在文件末给出了示例用法。

3. 工作总结

（1）收获、心得

本次实验我们小组实现了对于数据集进行不良内容分类与检测，让我们更加深入地了解了人工智能在实际中的具体应用以及初步实现。

（2）遇到问题、解决思路及测试结果

在老师给的模型基础上，我们经过修改 0 和 1 的权重比反复测试得到最佳预测率为 99.28 下的模型，此时 0、1 权重比为 10：1，具体见 model.py 和 train.py

```
(ai) PS E:\2024-1\Intro_to_AI\ai_img_srt> python classify.py
Reading images from folder: violence_224/test_val
Folder image predictions: [0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0
, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0,
, 0, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1
, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]
Accuracy on the test set: 99.28%
```

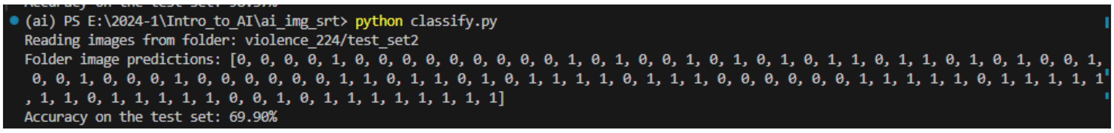
在已训练模型的基础上，为了提高模型对于 **AIGC** 风格变化、图像噪声、对抗样本等的鲁棒性，我们尝试做了以下修改：

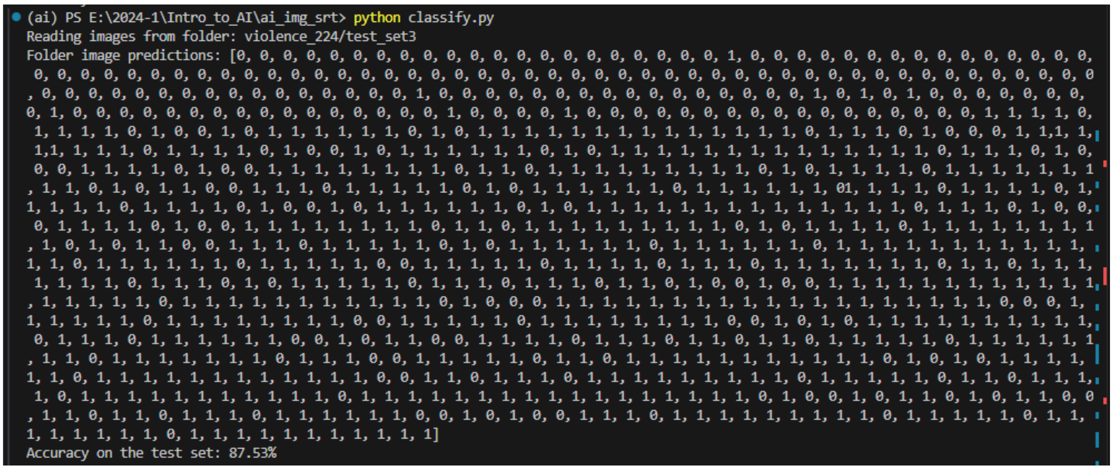
进行数据增强：

我们在 CustomDataModule 中添加使用 torchvision 提供的各种数据增强方法，如随机旋转、缩放、水平翻转等。

```python
from torchvision import transforms

# 在 CustomDataModule 中添加数据增强
class CustomDataModule(...):
    def __init__(self, batch_size):
        self.transform = transforms.Compose([
            transforms.RandomRotation(degrees=15),
            transforms.RandomResizedCrop(size=(224, 224), scale=(0.8,
1.0)),
            transforms.RandomHorizontalFlip(),
            transforms.ToTensor(),
            transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229,
0.224, 0.225]),
        ])
        ...

    def train_dataloader(self):
        return DataLoader(self.train_dataset, batch_size=self.batch_size,
shuffle=True, num_workers=4, pin_memory=True, drop_last=True)

    def val_dataloader(self):
        return DataLoader(self.val_dataset, batch_size=self.batch_size,
num_workers=4, pin_memory=True)
```

然后我们在加载预训练模型的基础上再训练，得到新的模型，经过测试，新模型相较于原模型在对 AIGC 图像的预测准确率由原来的 62.14%提高至 64%至 70%不等，但对噪声图像和原有测试集的预测率则在原模型 82.57 的基础上部分上升或有所下降，进行多次参数调整后最终选取综合预测率最高的模型：对同源测试集预测率 98.37；AIGC 预测率 69.90；对噪声图像预测率 87.53。只牺牲 1%的同源测试集预测率的情况下较大幅度地提高对 AIGC 和噪声图像的预测率，基本达到我们的预期。





受时间限制，我们没有采用多样化训练集、对抗训练、模型复杂性与正则化等更多方法来提高模型的鲁棒性。

（3）小组分工占比

小组成员各 25%

4. 课程建议

没有特别的建议，期待这门课越办越好，同学们学习收获更加丰硕。