

# 上海交通大学

SHANGHAI JIAO TONG UNIVERSITY

## 课程任务报告

COURSEWORK REPORT



任务名称: 不良内容图像检测

完成组号: 第九组

小组成员: 胡皓文、唐子蕙、师峰杰

完成时间: 2024 年 5 月 31 日

## 1. 实验过程描述

### 1.1 任务目标

基于一个给出的暴力图像检测的数据集，通过训练大模型的方式，构建一个二分类的检测模型。这个模型要实现的主要功能是可以对数据集的图像进行不良内容进行检测与识别，该数据集的图像应当包括与训练集分布类似的图像，对于 AIGC 风格变化、图像噪声、对抗样本也应当有一定的检测能力。模型还应当有合理的运行时间。

### 1.2 实现过程

#### 1.2.1 实施方案

我们构建了一个深度学习的训练管道，在参考代码给出的数据集加载、数据预处理、模型定义、训练和验证的基础上，进行了一定的修改。

#### 1.2.2 核心代码分析

在数据集定义与预处理的 `dataset.py` 文件中，我们对所有分割的数据集进行了图像大小的转换，用于符合我们的测试环境，对其转换为张量并进行标准化处理，此外，我们还对训练分割的数据进行了随机水平翻转进行数据加强。

然后我们定义了批量处理的大小，工作线程数等 `dataloader` 的配置信息。

在模型的定义初始化上，我们使用了 ResNet50 模型，拥有更好的图像识别能力，定义了一个交叉熵损失函数，设置学习率，并初始化准确率度量工具。除了初始化外，我们自定义了前向传播，优化器，训练步骤，验证步骤，测试步骤等部分。

最后我们实例化这个数据模块，在给定的数据集上进行训练。得到一个训练结果模型。

在核心文件 `classify.py`，也就是接口类文件中，我们定义了一个名为

ViolenceClass 的类，包含两个函数，misc 与 classify。在该类的初始化过程中，我们首先在设定的设备上加载了模型，设置为评估模式，并设置了预处理的各项参数，包括图像大小，将图转为 Tensor 并归一化以及标准化；在 misc 函数中，我们接受一个图像路径列表，使用设置过的预处理方法进行操作，返回一个预处理后的图像张量，形状为 (batch\_size, channels, height, width)；在 classify 函数中，我们输入 misc 函数返回的张量，形状为 (batch\_size, channels, height, width)，在关闭梯度计算（用于节省计算资源）的条件中，使用我们训练过的模型对其进行结果输出，返回每个输入在第 1 维度（类别维度）上的最大值的索引，也就是模型预测的类别标签。

### 1.2.3 模型测试结果

项目的检测我们主要以四个方面的准确率作为指标：与训练集同源的数据，非同源数据，AIGC 生成图像的数据作为测试集，加上图像噪声、对抗噪声之后的数据。同时，我们通过使用 CNN，构造了一个简单的 5 层卷积神经网络，作为我们实验的 Baseline。得到的实验测试结果图如下图 1.2.3.4。

图1 同源训练集测试结果

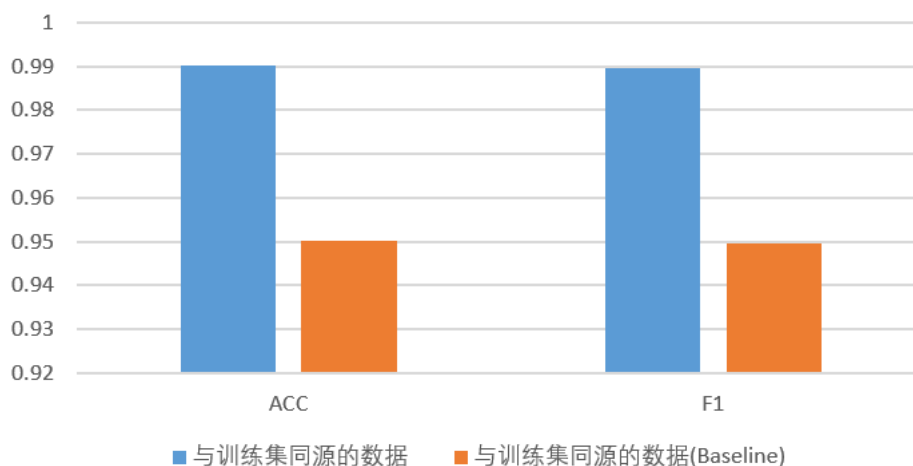


图2 非同源测试集测试结果

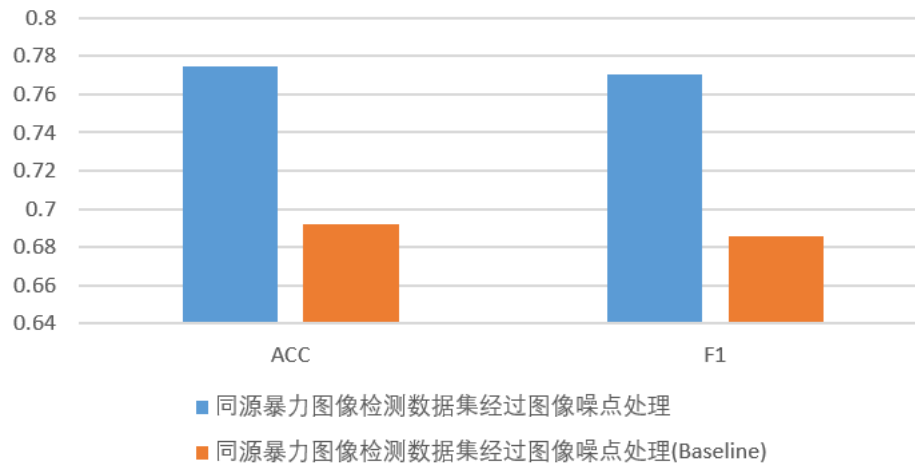


图3 噪声处理测试结果

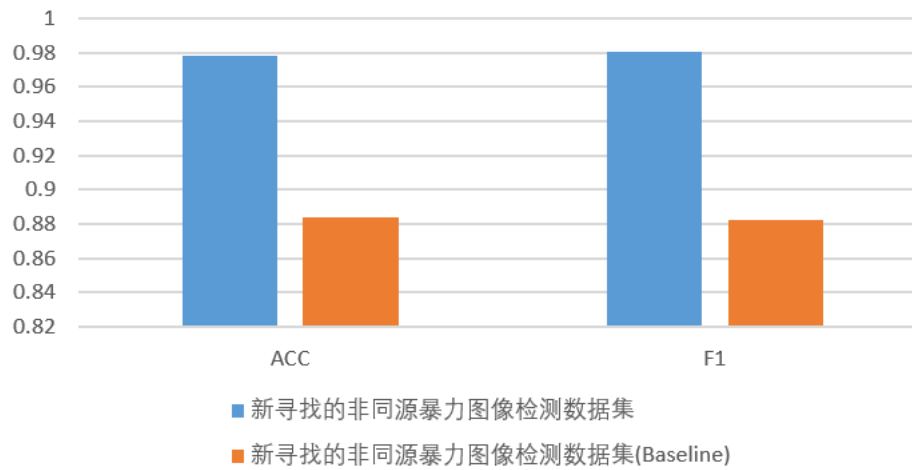
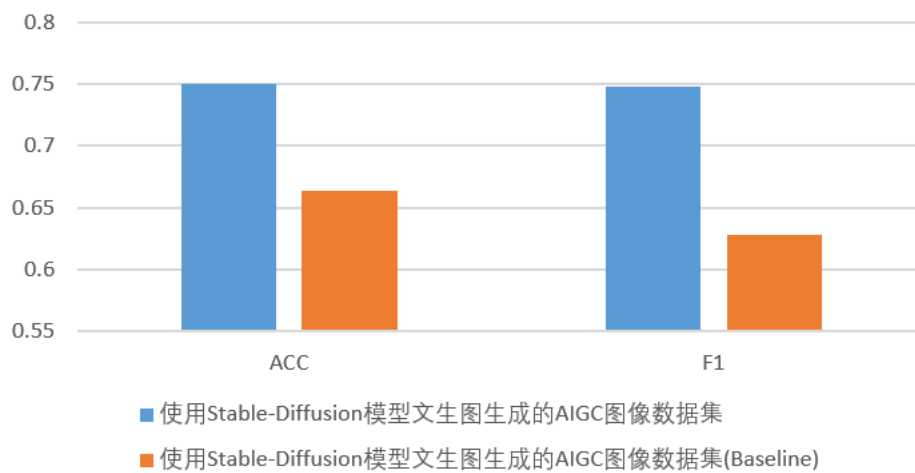


图4 AIGC图像测试结果



根据测试结果可见,我们训练的模型在准确率上,相比比较浅层的 CNN 而言,

提升较大。且我们的模型即使在面对，非同源数据，AIGC 生成图像的数据作为测试集，加上图像噪声、对抗噪声之后的数据时，依然有较好的准确率。

## 1.3 工作总结

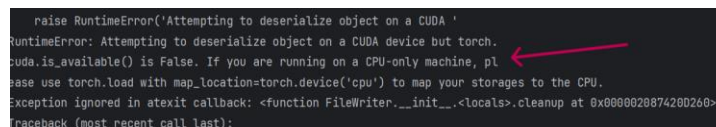
### 1.3.1 收获、心得

这次的大作业给了我一个机会去体验如何使用 CV 中比较热门的预训练模型来训练一个有一定功能的模型。整个流程感受了如何书写让我们的模型能够得到应用的接口类模型，在网络上收集了相关的非同源的自然数据集用于测试，对数据集进行对抗性处理用于测试，对数据集进行 AIGC 的图生图、文生图尝试用于生成测试集。

### 1.3.2 遇到问题及解决思路

暴力测试数据集并不容易从网上获取，大多数是已有的视频数据集，图片数据集较少，同时要增强泛化能力就需要大量 AI 生成图来作为测试集。

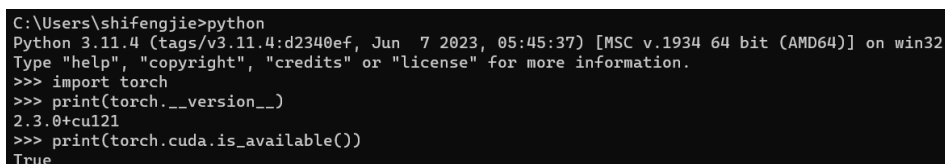
过程中尝试使用本地机训练模型，需要配置 pytorch 训练环境。遇到了安装问题。配置的 pytorch 是 CPU 版本，无法调用 GPU 训练。如图 1 所示



```
raise RuntimeError('Attempting to deserialize object on a CUDA '
RuntimeError: Attempting to deserialize object on a CUDA device but torch.
cuda.is_available() is False. If you are running on a CPU-only machine, pl
ase use torch.load with map_location=torch.device('cpu') to map your storages to the CPU.
Exception ignored in atexit callback: <function FileWriter.__init__ at 0x0000020874280260>
Traceback (most recent call last):
```

图 5 pytorch 环境问题

通过 `torch.cuda.is_available()` 返回 `false` 进一步验证上述问题。查阅相关文档<sup>1</sup>发现问题是在官网下载的 pytorch 库中没有对应主机的 python 版本，使用下载命令自动下载 CPU 版本。通过分别在不同镜像网站下载对应库实现安装。如图 2 所示。



```
C:\Users\shifengjie>python
Python 3.11.4 (tags/v3.11.4:d2340ef, Jun 7 2023, 05:45:37) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import torch
>>> print(torch.__version__)
2.3.0+cu121
>>> print(torch.cuda.is_available())
True
```

图 6 配置环境

在运行样例程序时，训练报错如下

1. RuntimeError: CUDA error: invalid device ordinal

<sup>1</sup> <https://windses.blog.csdn.net/article/details/125910538>

2. CUDA kernel errors might be asynchronously reported at some other API call, so the stacktrace below might be incorrect.
3. For debugging consider passing `CUDA_LAUNCH_BLOCKING=1`.
4. Compile with ``TORCH_USE_CUDA_DSA`` to enable device-side assertions.

按照第三行提示查看 train.py 文件发现尝试调用 GPU 的 id 为 1, 如图 3 所示。

```
195 from model import ViolenceClassifier
196 from dataset import CustomDataModule
197
198 gpu_id = [1]
199 lr = 3e-4
200 batch_size = 128
201 log_name = "resnet18_pretrain_test"
202 print("{} gpu: {}, batch size: {}, lr: {}".format(log_name, gpu_id, batch_size, lr))
203
204 data_module = CustomDataModule(batch_size=batch_size)
```

图 7 代码分析

但是本地机是有一个 GPU 核, 将图示部分 id 改为 0 后成功在本地机上训练初始模型。

## 2. 课程建议

课程内容十分充实, 较为全面的介绍了当下人工智能的主要热点领域, 勾画出了人工智能领域的基本图景。

课程内容充实, 部分非核心部分或者非关键的算法可以化简节省时间, 减少展开探讨, 比如绪论部分。有选择性的侧重于某几个模块进行深入介绍, 比如课程给定了图像分类模型和语音识别模型, 就可以就主题深入探讨, 并给定一个案例资源训练使用的具体流程, 再让学生去实现另一个项目, 有了样例依托实现会更加清晰。

AI 实践平台可以优化, 在提交后可以给出参考的答案结果, 便于及时纠错与进一步学习。

基于课程性质有许多讨论环节, 当下人工智能应用领域众多, 可以给定某一个专题应用视频统一观看或者几篇文章统一阅读为依托来展开讨论。

最后感谢马进老师和陈秀真老师一个学期以来辛苦认真的授课!