人工智能基础
题目二：表情识别

2018010734 自 92 魏子卜

一、 算法设计

1、数据预处理

在 process_dataset.py 中，读取.csv 文件中的图片和标签，其中图片是 48*48 的数据，标签为 0-6，代表 7 个不同的表情，并划分训练集和测试集。

通过继承 torch.utils.data.Dataset 设计数据集类，重载 init/getitem/len 三个方法，对图片做一些变换。

2、网络设计

采用了 ResNet50 作为神经网络的 backbone，将原网络最后一层的输出维度，从 1000 改为 7。如下图所示。

```
  (8): AdaptiveAvgPool2d(output_size=(1, 1))
)
(linear_layer): Linear(in_features=2048, out_features=7, bias=True)
```

3、训练网络

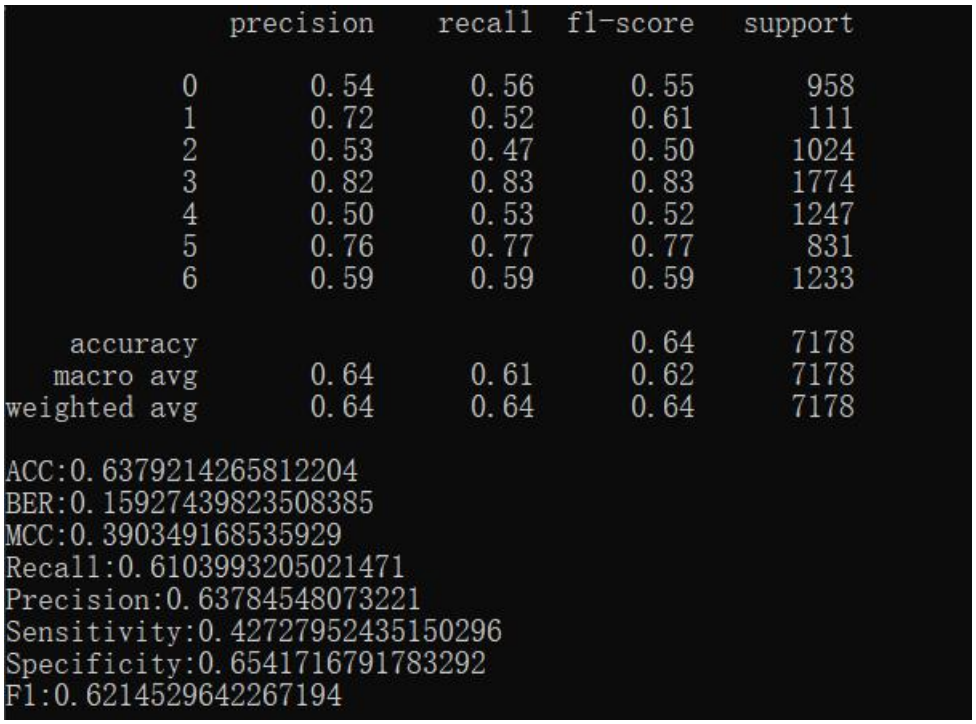
```
optimizer.zero_grad()
data = data.view(-1, 3, 48, 48)
output = net(data.float())
loss = criterion(output, label)
loss.backward()
optimizer.step()
```

将数据输入模型进行训练，进行正向传播和反向传播，使用交叉熵损失函数得到 loss，使用随机梯度下降更新参数。一共训练 50epoch，batch size 设为 64，学习率为 0.001。

4、测试网络

使用测试集验证网络的性能，在第 44 个 epoch 性能最好，达到 64%。

测得模型各指标如下图所示。



	precision	recall	f1-score	support
0	0.54	0.56	0.55	958
1	0.72	0.52	0.61	111
2	0.53	0.47	0.50	1024
3	0.82	0.83	0.83	1774
4	0.50	0.53	0.52	1247
5	0.76	0.77	0.77	831
6	0.59	0.59	0.59	1233
accuracy			0.64	7178
macro avg	0.64	0.61	0.62	7178
weighted avg	0.64	0.64	0.64	7178

ACC:0.6379214265812204
BER:0.15927439823508385
MCC:0.390349168535929
Recall:0.6103993205021471
Precision:0.63784548073221
Sensitivity:0.42727952435150296
Specificity:0.6541716791783292
F1:0.6214529642267194

二、 类别不均衡问题

在该数据集中，存在一定程度的类别不均衡问题，如 Happy 较多，而 Disgust 较少。类别不均衡会导致训练集不再是真实样本的无偏采样，因此会影响模型的性能。假设 99%的训练数据都是 Happy，那么模型只需要直接预测为 Happy 即可达到 99%的准确率，但实际上这个模型完全不对。在未进行类别不均衡改进时，Disgust 类别的准确率也相对较高，可见原有模型的性能不错。

1、数据增强（过采样）

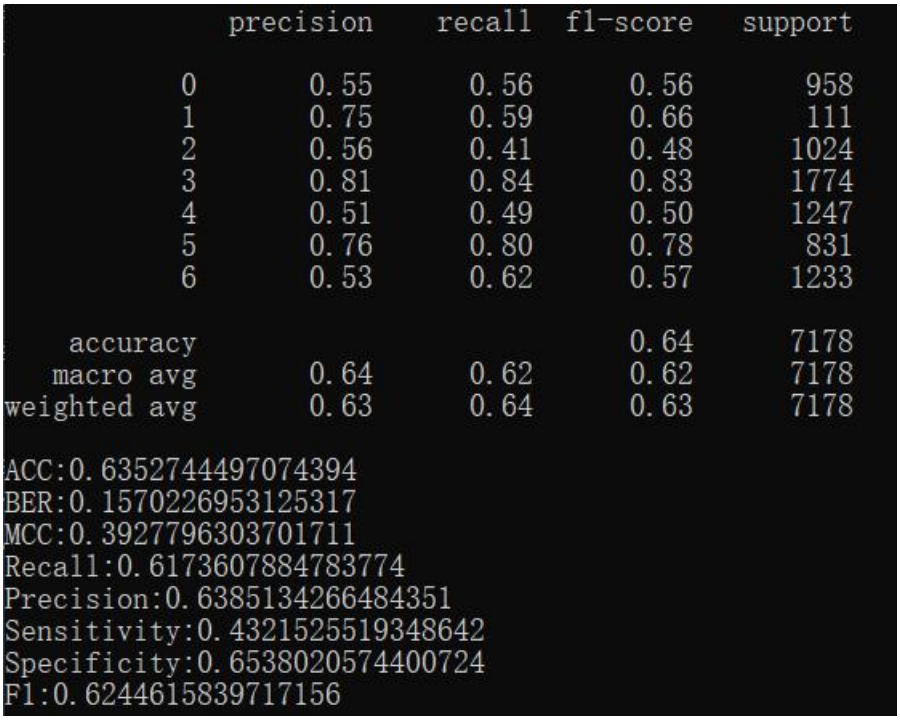
在加载数据集时，已经对图象进行了随机翻转，因此已经进行了一些增强。希望通过过采样增加 Disgust 类别的数据量，使得正反样本数量相近，还可以使用随机裁剪等方法进行数据增强。但这一类别数据还是过少，这些方法会导致模型出现过拟合，因此没有采用。

2、代价敏感

代价敏感的方法是修改损失函数，增加 Disgust 类别的权重，在本模型中，修改交叉熵损失函数为如下形式，以调和各个类别的数量。

criterion=nn.CrossEntropyLoss(weight=torch.Tensor([0.1,0.3,0.1,0.1,0.1,0.1,0.1]).to(device))

训练后，选取第 48epoch 进行检测，整体准确率基本没有变化，而 Disgust 类别的准确率略有提高，说明该方法有一点效果，各个指标如下图所示。



	precision	recall	f1-score	support
0	0.55	0.56	0.56	958
1	0.75	0.59	0.66	111
2	0.56	0.41	0.48	1024
3	0.81	0.84	0.83	1774
4	0.51	0.49	0.50	1247
5	0.76	0.80	0.78	831
6	0.53	0.62	0.57	1233
accuracy			0.64	7178
macro avg	0.64	0.62	0.62	7178
weighted avg	0.63	0.64	0.63	7178

ACC:0.6352744497074394
BER:0.1570226953125317
MCC:0.3927796303701711
Recall:0.6173607884783774
Precision:0.6385134266484351
Sensitivity:0.4321525519348642
Specificity:0.6538020574400724
F1:0.6244615839717156

3、SMOTE 算法

上网查阅相关资料，SMOTE(Synthetic Minority Oversampling Technique)算法基本思想是对少数类样本进行分析，并根据少数类样本人工合成新样本添加到数据集中，这样增加的数据可以避免过拟合的问题，但在这个问题中可能难以实现。

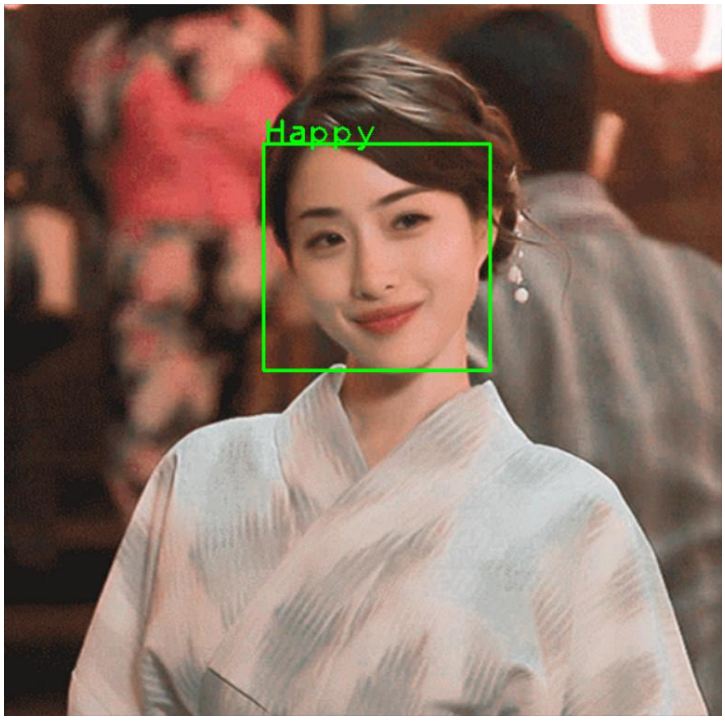
综上列举了一些可能有效的解决方案，并进行相应的研究和探讨，考虑到原有模型较好的处理了 Disgust 类别，因此没有进行过多的实验。如果将来项目中遇到数据不均衡的问题，可以使用这些方法。

三、 接口设计(选做)

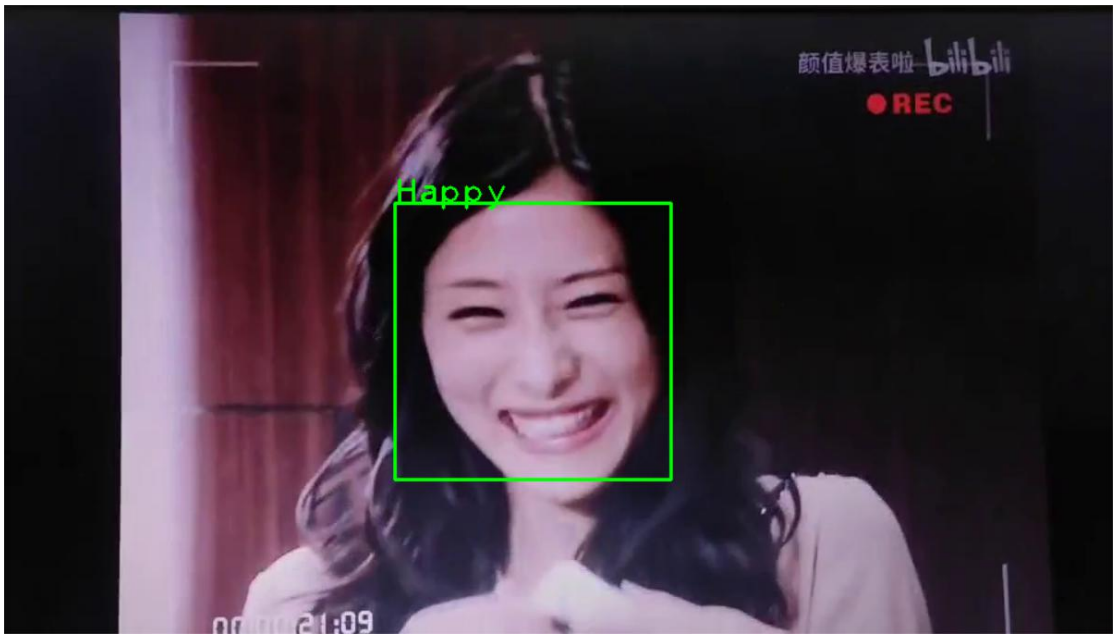
提供一个输入图象/视频的接口，自动识别出其中的人脸，利用训练的模型判断人物表情，标注于图象/视频中。首先使用 opencv 读取图片数据并转

为灰度图，再利用 cv2.CascadeClassifier('haarcascade_frontalface_default.xml') 识别出人脸的位置，提取出该区域后 resize 到 48*48，使用训练的模型得到预测的表情。对于视频，使用 cv2.VideoCapture()，之后类似图象的处理。

运行 interface.py，可以看到对图象的处理，将代码中的注释部分取消注释，再次运行，即可看到视频的效果。如下图所示。



图象



视频截图

四、 总结

这次大作业使用神经网络实现了一个简单的面部表情分类器，达到较好效果。还分析了本项目中的类别不均衡问题，以及相应的解决办法。最后调用 opencv 的函数，自动识别出图象和视频中的人脸，并判断面部表情，标注在图象和视频中，非常有趣。

这次大作业是我做的第一个深度学习小项目，综合运用了课上所学内容知识，也更加熟悉 pytorch 的使用，收获非常大。