**程序报告**

学号：2013536 姓名：汤清云

1. **问题重述**

（简单描述对问题的理解，从问题中抓住主干，必填）

====================================================================

1. 建立深度学习模型，检测图中的人是否戴了口罩，调整参数优化模型。
2. 学习经典的模型 MTCNN 和 MobileNet 的结构。
3. 用已有的人脸检测的模型，再训练一个识别口罩的模型，从而提高训练的开支、增强模型的准确率。
4. **设计思想**

（所采用的方法，有无对方法加以改进，该方法有哪些优化方向（参数调整，框架调整，或者指出方法的局限性和常见问题），伪代码，理论结果验证等… **思考题，非必填**）

====================================================================

采用方法：调整预训练模型的权重，修改优化器、最佳损失率等。

1. **代码内容**

（能体现解题思路的主要代码，有多个文件或模块可用多个"===="隔开，必填）

====================================================================

# 加载 MobileNet 的预训练模型权

device = torch.device("cuda:0") if torch.cuda.is\_available() else torch.device("cpu")

train\_data\_loader, valid\_data\_loader = processing\_data(data\_path=data\_path, height=160, width=160, batch\_size=32)

modify\_x, modify\_y = torch.ones((32, 3, 160, 160)), torch.ones((32))

epochs = 2

model = MobileNetV1(classes=2).to(device)

optimizer = optim.Adam(model.parameters(), lr=4e-5) # 优化器

print('加载完成...')

====================================================================

# 学习率下降的方式，acc三次不下降就下降学习率继续训练，衰减学习率

scheduler = optim.lr\_scheduler.ReduceLROnPlateau(optimizer,

'max',

factor=0.5,

patience=4)

# 损失函数

criterion = nn.CrossEntropyLoss()

====================================================================

best\_loss = 3e8

best\_model\_weights = copy.deepcopy(model.state\_dict())

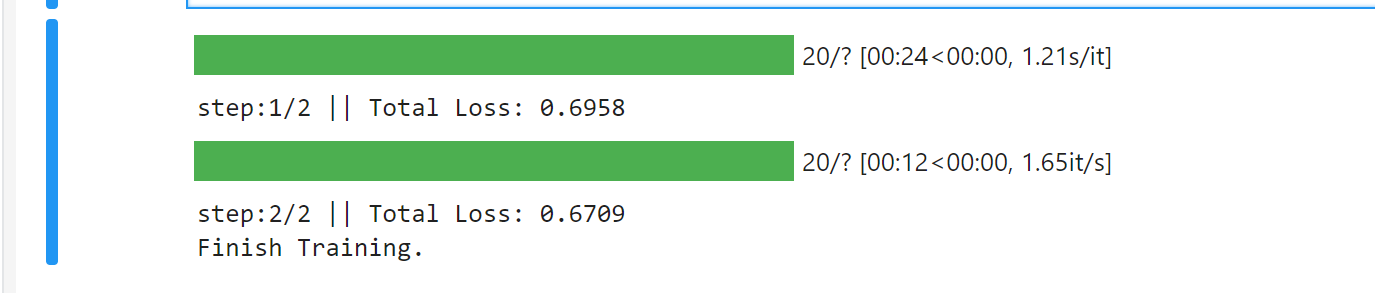
loss\_list = [] # 存储损失函数值

1. **实验结果**

（实验结果，必填）

====================================================================

判别结果如下：







1. **总结**

（自评分析（是否达到目标预期，可能改进的方向，实现过程中遇到的困难，从哪些方面可以提升性能，模型的超参数和框架搜索是否合理等），**思考题，非必填**）

====================================================================

1. 达到目标预期
2. 模型训练过程中存在随机初始化，不同的初始化对最后的结果有影响。
3. 可以根据选择的框架（torch）学习怎么固定随机数种子，使得每次实验结果能够复现，不再是随机生成。