程序报告

学号：2211290　　姓名：姚知言

一、问题重述

建立深度学习模型，检测图中的人是否佩戴口罩，并通过学习经典的 MTCNN 人脸识别和 Mobilenet 口罩识别，不断进行参数调优使得模型达到更好状态，得到训练模型。

二、设计思想

该实验的总体框架为预处理及数据增强-创建数据集-MCTNN 人脸识别-MobileNet 口罩识别-得到最终模型。

首先对未进行参数调优的测试组进行一次测试，得到结果为 keras 组 77.5 分，touch 组 56.67 分，mindspore 组 75.0 分，以便后续对比。

针对 keras 组进行优化，修改 epochs=10/100 后，分数都没有变化，为权衡运行时间，选用 epoch=20。针对 patience=3，batch_size = 16/32 进行尝试后，未能取得更理想的分数。

先对 touch 组进行优化，修改 patience=3，并修改 epochs=10，增加学习次数，得分为 90.0 分。在此之后，我尝试了更多的参数组合，最终取得了较好的模型实现。

三、代码内容

```
#touch 组代码
epochs = 9
model = MobileNetV1(classes=2).to(device)
optimizer = optim.Adam(model.parameters(), lr=1e-3)  # 优化器
print('加载完成...')
scheduler = optim.lr_scheduler.ReduceLROnPlateau(optimizer,
                                                 'max',
                                                 factor=0.5,
                                                 patience=3)
# 损失函数
criterion = nn.CrossEntropyLoss()
best_loss = 1e9
best_model_weights = copy.deepcopy(model.state_dict())
loss_list = []  # 存储损失函数值
for epoch in range(epochs):
    model.train()
    for batch_idx, (x, y) in tqdm(enumerate(train_data_loader, 1)):
        x = x.to(device)
        y = y.to(device)
        pred_y = model(x)
        loss = criterion(pred_y, y)
        optimizer.zero_grad()
```

```
            loss.backward()
            optimizer.step()
            if loss < best_loss:
                best_model_weights = copy.deepcopy(model.state_dict())
                best_loss = loss
            loss_list.append(loss)
        print('step:' + str(epoch + 1) + '/' + str(epochs) + ' || Total Loss: %.4f' % (loss))
torch.save(model.state_dict(), './results/temp.pth')
print('Finish Training.')
==========================================================================
#torch 组提交结果
from torch_py.Utils import plot_image
from torch_py.MTCNN.detector import FaceDetector
from torch_py.MobileNetV1 import MobileNetV1
from torch_py.FaceRec import Recognition
from torch_py.FaceRec import Recognition
from PIL import Image
import cv2
model_path = 'results/temp.pth'
def predict(img):
    if isinstance(img, np.ndarray):
        img = Image.fromarray(cv2.cvtColor(img,cv2.COLOR_BGR2RGB))

    recognize = Recognition(model_path)
    img, all_num, mask_num = recognize.mask_recognize(img)
    return all_num,mask_num
==========================================================================
#kera 组
reduce_lr = ReduceLROnPlateau(
                                monitor='accuracy',   # 检测的指标
                                factor=0.5,        # 当 acc 不下降时将学习率下调的比例
                                patience=3,        # 检测轮数是每隔两轮
                                verbose=2           # 信息展示模式
                            )
early_stopping = EarlyStopping(
                                    monitor='val_loss',   # 检测的指标
                                    min_delta=0,             # 增大或减小的阈值
                                    patience=10,             # 检测的轮数频率
                                    verbose=1                 # 信息展示的模式
                                )
# 一次的训练集大小
batch_size = 32
# 图片数据路径
data_path = basic_path + 'image'
```

```
# 图片处理
train_generator,test_generator = processing_data(data_path, height=160, width=160, batch_size=batch_size, test_split=0.1)
# 编译模型
model.compile(loss='binary_crossentropy',  # 二分类损失函数
              optimizer=Adam(lr=0.1),              # 优化器
              metrics=['accuracy'])          # 优化目标
# 训练模型
history = model.fit(train_generator,
                    epochs=20, # epochs: 整数，数据的迭代总轮数。
                    # 一个 epoch 包含的步数,通常应该等于你的数据集的样本数量除以批量大小。
                    steps_per_epoch=637 // batch_size,
                    validation_data=test_generator,
                    validation_steps=70 // batch_size,
                    initial_epoch=0, # 整数。开始训练的轮次（有助于恢复之前的训练）。
                    callbacks=[checkpoint_period, reduce_lr])
# 保存模型
model.save_weights(model_dir + 'temp.h5')
```

四、实验结果

五、总结

在本次实验中，通过对 MTCNN 人脸识别和 Mobilenet 口罩识别构建，了解了深度学习的基本原理，对深度学习有了基础的认识，也对这些库函数有了一定的理解。随着实验的进行，我对于训练模型的开销有了认识，也认识到一些细微的参数对于模型整体的影响。因此，对模型的细节认知理解是非常重要的。