天津大学

基于 Pytorch 的 Ghost-ResNet-50 垃圾分类模型的构建与优化

姓名：武嘉闻（3019208197）

姓名：雷伦昊（3019207152）

姓名：徐德军（3019208140）

姓名：杨　昊（3019207255）

天津大学微电子学院

2022 年 1 月

摘　要

城市生活垃圾清运量逐年增加，由此带来的环境问题日益严峻。不同垃圾组分有其各自适宜的处理与利用方式。因此，有效的垃圾分类技术具有重要意义。我国现有的以人工为主的垃圾分类方式存在效率低、成本高、准确性差等弊端，而垃圾智能分类技术是解决相关瓶颈的重要手段。GhostNet 与 ResNet-50 是常用的两种模型，本文用 GhostNet 模型替换 ResNet-50 模型中一半的卷积层，采用 Pytorch 编程框架完成了这个新的模型（Ghost-ResNet-50）的构建，并按照题目要求基于华为云 ModelArts 平台完成训练，最终实现对六类生活垃圾（cardboard，glass，metal，paper，plastic，trash）的分类任务。

关键词：Pytorch；图像分类；GhostNet；ResNet-50

ABSTRACT

The volume of municipal solid waste is increasing year by year, and the environmental problems are becoming more and more serious. Different waste components have their own appropriate treatment and utilization methods. Therefore, effective garbage classification technology is of great significance. The existing garbage classification method based on human has disadvantages such as low efficiency, high cost and poor accuracy, and intelligent garbage classification technology is an important means to solve the relevant bottleneck. GhostNet and ResNET-50 are two commonly used models. In this paper, GhostNet is used to replace half of the convolution layer in ResNet-50, and the Pytorch programming framework is used to complete the new model (Ghost-ResNet-50) on Huawei Cloud ModelArts platform. To achieve sorting tasks through image classification.

Key words: Pytorch; Image Classification; GhostNet; ResNet-50;

目录

1 引言 1

2 实验准备 1

2.1 GhostNet 1

2.2 ResNet-50 3

3 Ghost-ResNet-50 模型 4

4 训练 6

4.1 卷积神经网络 6

4.2 损失函数（Loss 函数）..... 7

4.3 优化方法及初始值设置 7

4.4 数据预处理 7

4.5 训练的结果 7

5 测试结果 8

5.1 测试的各项结果 9

5.2 测试分类图片 9

5.3 错误原因分析 12

6 总结 13

参考文献 14

附录 15

1 引言

随着国家经济的发展，人们的生活水平不断提高，日常生活的垃圾产量日益增长，导致环境状况不断地日益恶化。与此同时，社会的环保意识日益增强，虽然我国不断推出了垃圾分类政策和与之相关的垃圾分类措施，以保证社会与自然的和谐发展，但由于人民群众长期缺乏垃圾分类知识，导致垃圾分类工作成效不明显，没有充分发挥其影响力。缺乏环保知识是制约大众进行有效垃圾分类的主要原因之一，而居民作为垃圾分类的主要执行者，居民能否进行有效垃圾分类直接对国家的垃圾分类成效产生影响，因此提高居民参与垃圾分类的准确程度是本文旨在解决的问题。

据世界银行 2018 年发布的《垃圾何其多 2.0》显示，2016 年中国产生近 2.2 亿吨城市生活垃圾，其中可以二次回收利用的占 60%左右。可见，垃圾回收处理有着巨大的生态收益和经济收益，而要充分利用垃圾资源或者减少垃圾的产生，前提是对垃圾进行分类。当前，我国的垃圾分类主要以人工分拣为主，存在工作量大、效率低、工作环境恶劣等问题，利用深度学习技术实现智能化分类可以有效降低人工成本，节约人力资源。垃圾分类的本质是对图像(或视频)中的目标进行检测。传统的目标检测方法主要使用 HOG、SIFT 等特征对滑动窗口进行判别，存在耗时长、鲁棒性差、准确度不高的问题；基于深度卷积网络的目标检测方法主流模型有：YOLO、Faster CNN、SSD 等，在相应的数据集领域有较好的效果。但在垃圾分类问题上存在训练耗时长、准确率不高等问题。

GhostNet 与 ResNet-50 是常用的两种模型，本文用 GhostNet 替换 ResNet-50 中一半的卷积层，并使用 Pytorch 编程框架完成了这个新的模型，并按照题目要求基于华为云 ModelArts 平台完成训练，最终实现对六类生活垃圾（cardboard，glass，metal，paper，plastic，trash）的分类任务。

2 实验准备

2.1 GhostNet

GhostNet 是华为诺亚方舟实验室于 2020 年提出的一种轻量化神经网络。作者在分析大量的输出特征图时发现，其实有些特征图是比较相似的，而部分相似的特征图可以通过简单的变换得到，如下图 1 所示。

基于此，Kai Han 等人得到启发，不是每张图都需要用到这么大的计算量去得到。我们可以通过 cheap transformation 得到这些相似的特征图。于是就诞生了 GhostNet。Ghost 就代表着与此相似的特征图，犹如另一个的幽灵，可以通过简单的线性变换得到。

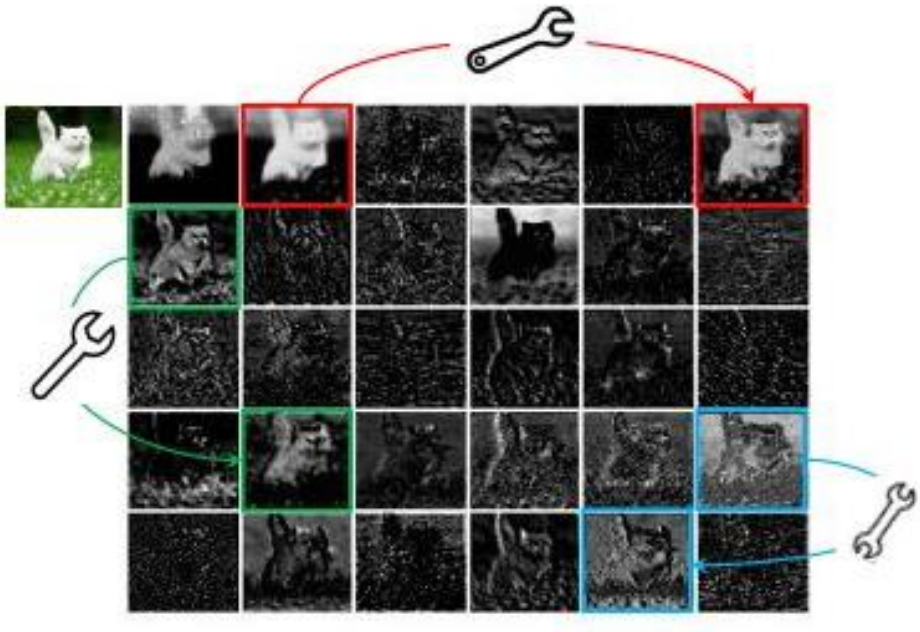

图 2.1　GhostNet 特征图

为了减少神经网络计算量，Kai Han 等人将传统的卷积分成两步进行，首先利用较少的计算量通过传统的卷积（使用 pointwise conv 效率较高）生成 channel 较小的特征图，然后在此特征图的基础上，通过 cheap operation (depth wise conv) 再进一步利用较少的计算量，生成新的特征图。最后将两组特征图拼接到一起，得到最终的 output。具体结构如下图所示。

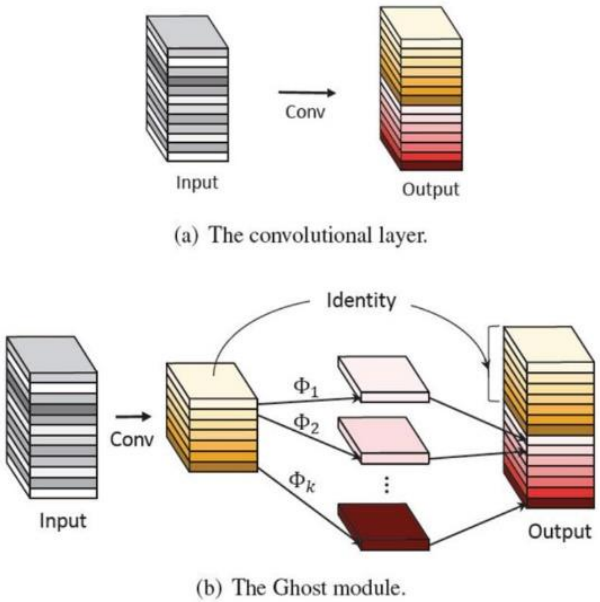

图 2.2　Ghost module 结构图

2.2 ResNet-50

ResNet-50 有 2 个基本的 block（使用残差网络结构），一个是 Identity Block，输入和输出特征向量的维度是一样的，所以可以串联多个。另外一个基本 block 是 Conv Block，输入和输出特征向量的维度是不一样的，所以不能连续串联，它的作用是为了改变特征向量的维度。

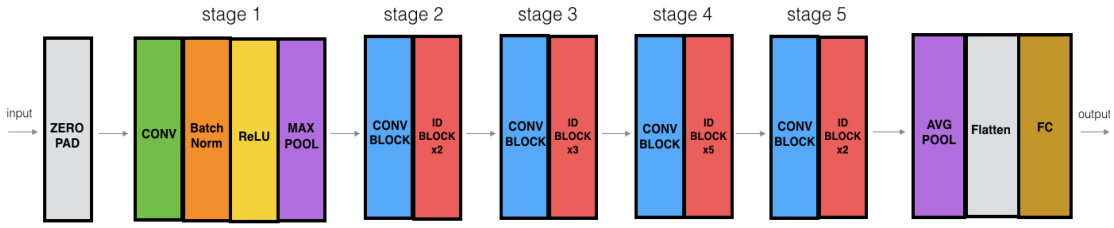

图 2.3　ResNet-50 整体结构图

因为 CNN（卷积神经网络）最后都是要把输入图像一点点的转换成很小但是 depth 很深的 feature map，一般的套路是用统一的比较小的 kernel（比如 VGG 都是用 3*3），但是随着网络深度的增加，output 的 channel 也增大（学到的东西越来越复杂），所以有必要在进入 Identity Block 之前，用 Conv Block 转换一下维度，这样后面就可以连续接 Identity Block。

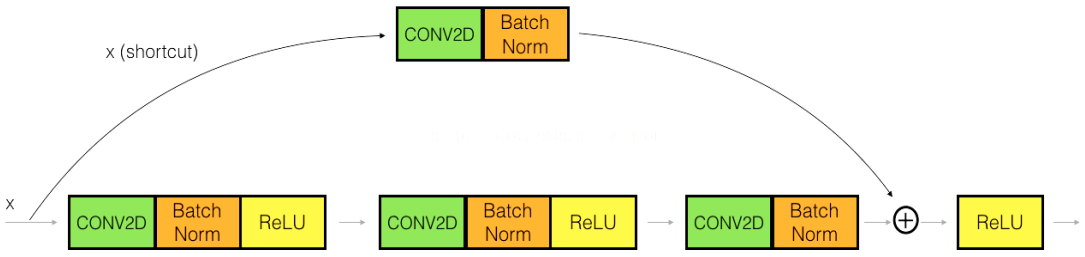

图 2.4　Conv Block 结构图

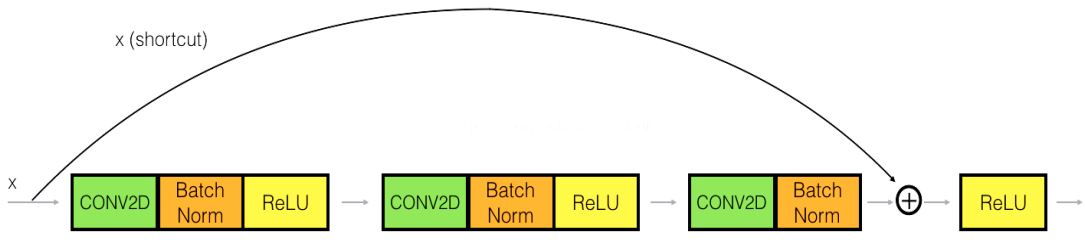

图 2.5　Identity Block 结构图

3 Ghost-ResNet-50 模型

我们通过将 GhostNet 中的 Ghost module 模块替换掉 ResNet-50 模型中的部分传统卷积层得到了我们的新模型 Ghost-ResNet-50。 新网络的整体结构依旧是 ResNet-50 的传统结构没变，变化的只是把 Conv Block 和 Identity Block 这两个残差网络结构中的部分传统卷积替换成了 Ghost module。

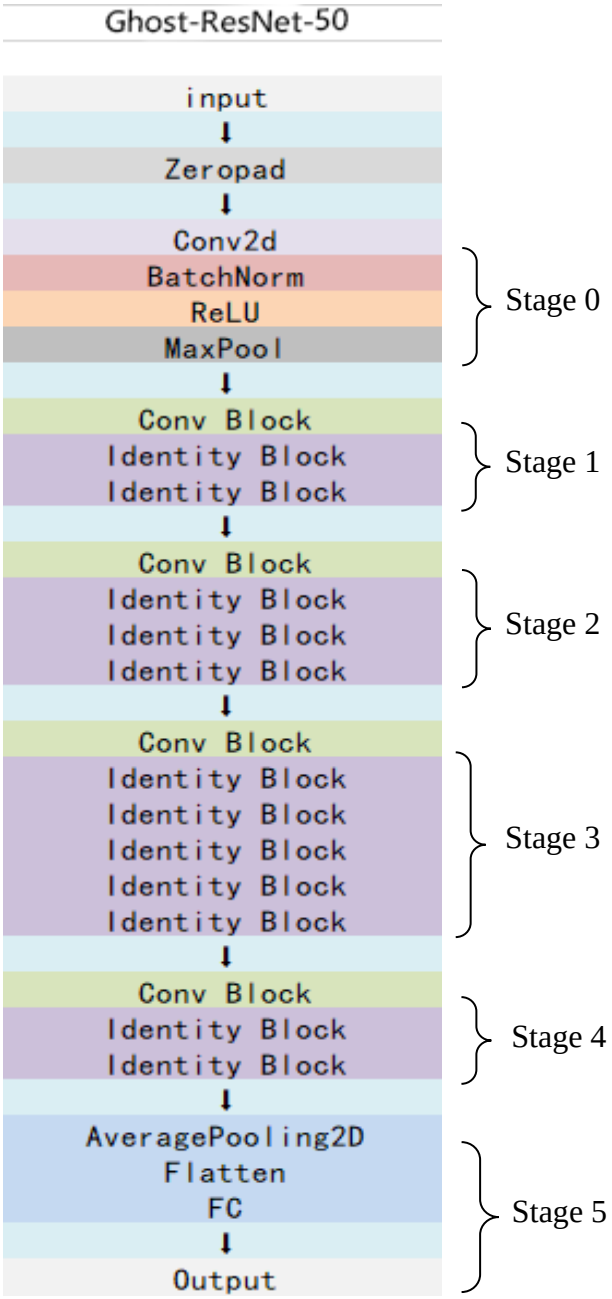

图 3.1　Ghost-ResNet-50 结构图

新模型 Ghost-ResNet-50 大体上可以分为 6 个 Stage。

1) Stage 0

1) Conv2d：Conv2d 是传统卷积层。

2) Batch Norm：即 BN 层，对数据进行批量归一化，可以加速网络收敛速度。

3) ReLU：ReLU 即 ReLU 激活函数，是人工神经网络中常用的一种激活函数。

4) MaxPool：即最大池化层，可以减小参数的数量和计算量，控制过拟合。

Stage0 部分代码如下：

```
self.conv1 = nn.Sequential(
    nn.Conv2d(3, 64, kernel_size=3, padding=1, bias=False),
    nn.BatchNorm2d(64),
    nn.ReLU(inplace=True))
#we use a different inputsize than the original paper
#so conv2_x's stride is 1
self.conv2_x = self._make_layer(block,  64, num_block[0], 1)
self.conv3_x = self._make_layer(block, 128, num_block[1], 2)
self.conv4_x = self._make_layer(block, 256, num_block[2], 2)
self.conv5_x = self._make_layer(block, 512, num_block[3], 2)
```

2) Stage 1

该 Stage 包括了 3 个 Bottleneck 结构：1 个 Conv Block 和 2 个 Identity Block。我们对 ResNet-50 中的两种 Block 结构进行了一定的修改，将 Ghost Module 替换掉了其中部分卷积层。修改后的 Identity Block 依旧不会改变通道数，Conv Block 依旧会改变输入的维度。

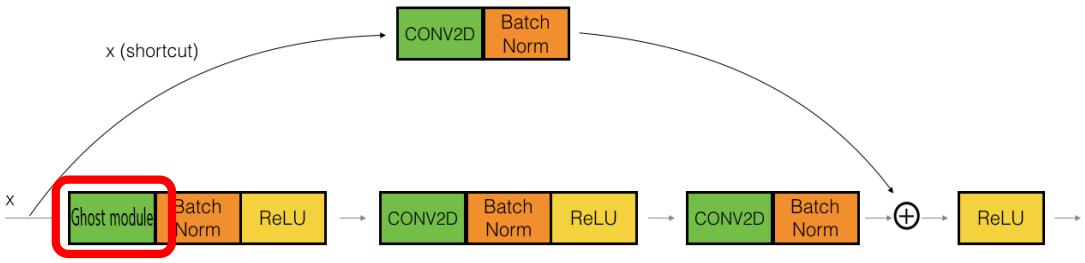

图 3.2　替换后的 Conv 结构

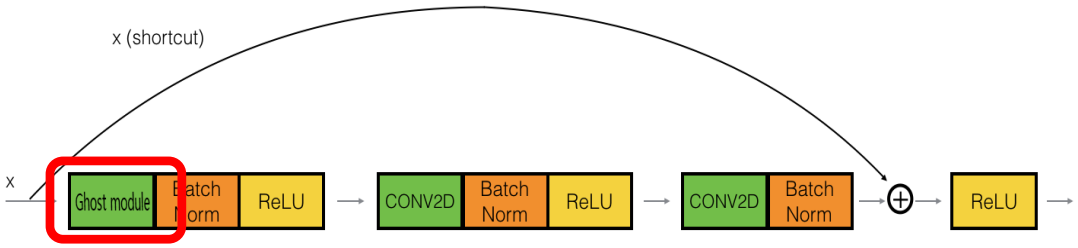

图 3.3　替换后的 Identity 结构

3) Stage 2

Stage 2 与 Stage 1 类似，不过 Bottleneck 变成了 4 个：1 个 Conv Block 和 3 个 Identity Block。

4) Stage 3

该 Stage 的 Bottleneck 增至 6 个：1 个 Conv Block 和 5 个 Identity Block。

5) Stage 4

该 Stage 的 Bottleneck 减至 3 个：1 个 Conv Block 和 2 个 Identity Block。

6) Stage 5

Stage 5 即对的 Stage 4 输出进行全局平均池化（Average Pooling）和全连接（Fully Connect）操作，然后输出最终的结果。Stage5 部分代码如下：

```
self.avg_pool = nn.AdaptiveAvgPool2d((1, 1))
self.fc = nn.Linear(512 * block.expansion, num_classes)
```

4　训练

我们的数据集选用“./GarbageClassification/GarbageClassification_train”文件夹下的 2275 张照片进行训练。

4.1 卷积神经网络

我们使用 Ghost-ResNet-50 作为我们的卷积神经网络：

```
GhostModule(in_channels, out_channels),
nn.BatchNorm2d(out_channels),
nn.ReLU(inplace=True),
nn.Conv2d(out_channels, out_channels, stride=stride,
kernel_size=3, padding=1, bias=False),
nn.BatchNorm2d(out_channels),
nn.ReLU(inplace=True),
nn.Conv2d(out_channels, out_channels * BottleNeck.expansion,
kernel_size=1, bias=False),
nn.BatchNorm2d(out_channels * BottleNeck.expansion),
```

4.2 损失函数（Loss 函数）

本模型才使用的的 Loss 函数为交叉熵损失函数，定义如下：

```
loss_function = nn.CrossEntropyLoss()
```

其中 nn.CrossEntropyLoss()是 nn.logSoftmax()和 nn.NLLLoss()整合而成的。交叉熵的计算公式如下：

$$H(p,q) = -\sum_{x}(p(x)logq(x))$$

4.3 优化方法及初始值设置

初始值和超参数的设置因计算机资源和时间有限，初始学习率设置为 0.1，训练 10 次 epoch，在 batch_size = 4 的情况下，weight_decay = 5e-4。

本模型所采用的优化方法是 torch.optim.SGD()。

SGD 是最基础的优化方法，普通的训练方法,需要重复不断的把整套数据放入神经网络中训练,这样消耗的计算资源会很大。当我们使用 SGD 会把数据拆分后再分批不断放入神经网络中计算。每次使用批数据,虽然不能反映整体数据的情况,不过却很大程度上加速了神经网络的训练过程,而且也不会丢失太多准确率。

4.4 数据预处理

图像预处理部分代码如下：

```
transform_train = transforms.Compose([
    transforms.RandomCrop(32, padding=4), # 随机裁剪
    transforms.RandomHorizontalFlip(),# 依概率 p 随机水平翻转
    transforms.RandomRotation(15), # 依概率 0.15 随机旋转
    transforms.ToTensor(), # 转换为 tensor 格式，可以直接输入进神经网络
    transforms.Normalize(mean, std) # 对像素值进行归一化处理
```

4.5 训练的结果

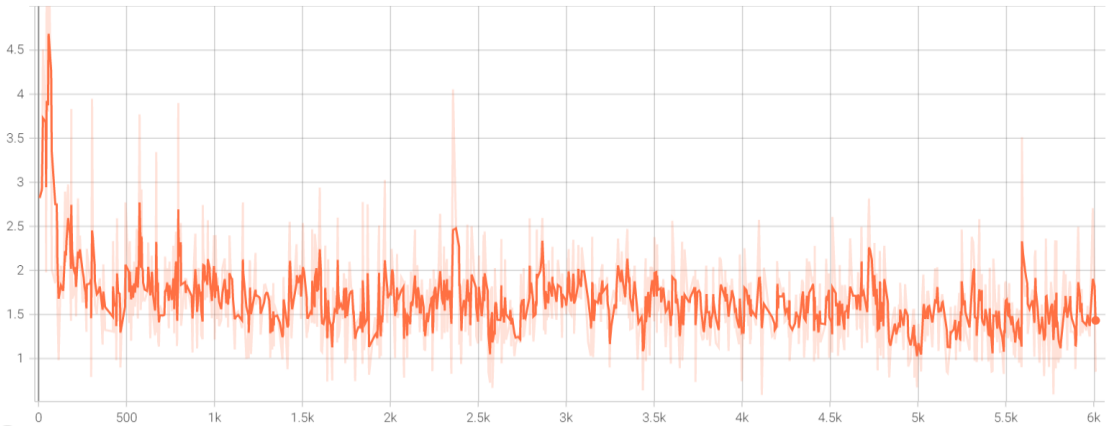

图 4.1　训练的损失函数

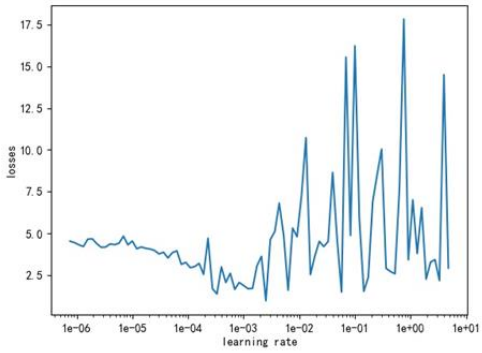

图 4.2　学习率

5　测试结果

我们的测试数据集为“./GarbageClassification/GarbageClassification_test”文件夹下的 152 张照片进行训练。

5.1 测试的各项结果

模型精度：

```
Top1 err: tensor(0.6310)
Top5 err: tensor(0.0556)
```

总参数量：21.6M

```
Parameter Numbers: 215605060
```

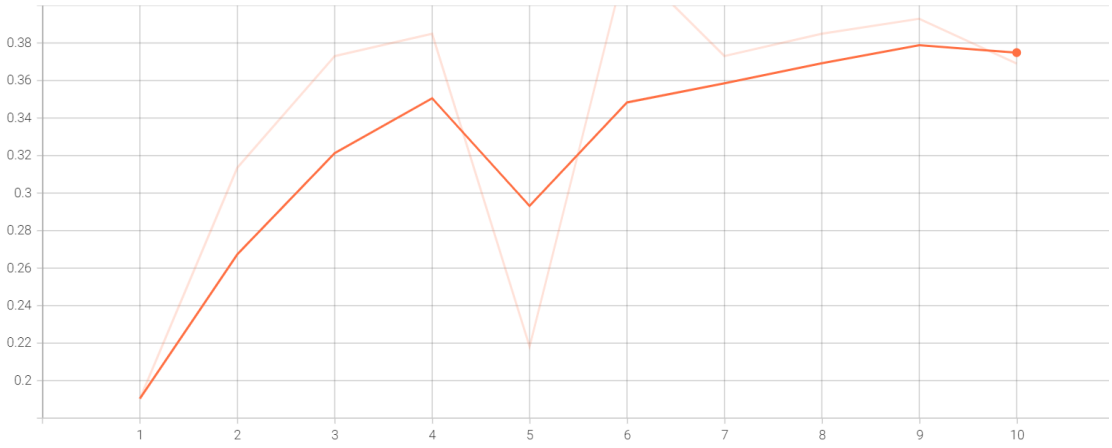

图 5.1　测试的准确性

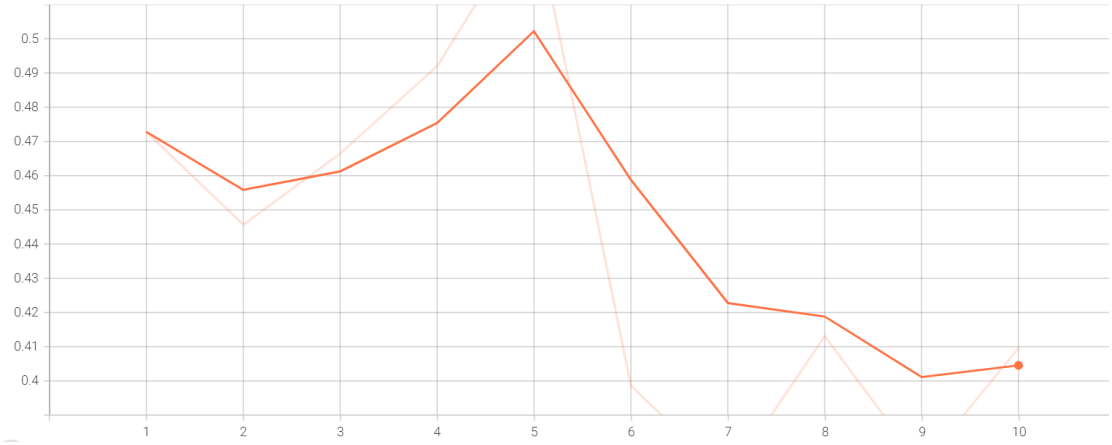

图 5.2　测试的平均损失函数

```
iteration: 1    total 16 iterations

iteration: 2    total 16 iterations
iteration: 3    total 16 iterations
iteration: 4    total 16 iterations
iteration: 5    total 16 iterations
iteration: 6    total 16 iterations
iteration: 7    total 16 iterations
iteration: 8    total 16 iterations
iteration: 9    total 16 iterations
iteration: 10   total 16 iterations
iteration: 11   total 16 iterations
iteration: 12   total 16 iterations
iteration: 13   total 16 iterations
iteration: 14   total 16 iterations
iteration: 15   total 16 iterations
iteration: 16   total 16 iterations

Top 1 err:  tensor(0.6310)
Top 5 err:  tensor(0.0556)
Parameter numbers: 21565060
```

图 5.3　模型测试输出结果

5.2　测试分类图片

以华为云 ModelArts 中的 ResNet-50 模型为例，下面展示 4 张分类正确的图片和 4 张分类错误的图片

1) 分类正确

图 5.4 原图片分类为 `metal` 类，预测结果为 `metal` 类，分类正确。

图 5.4　分类正确 1

图 5.5 原图片分类为 plastic 类，预测结果为 plastic 类，分类正确。

图 5.5　分类正确 2

图 5.6 原图片分类为 paper 类，预测结果为 paper 类，分类正确。

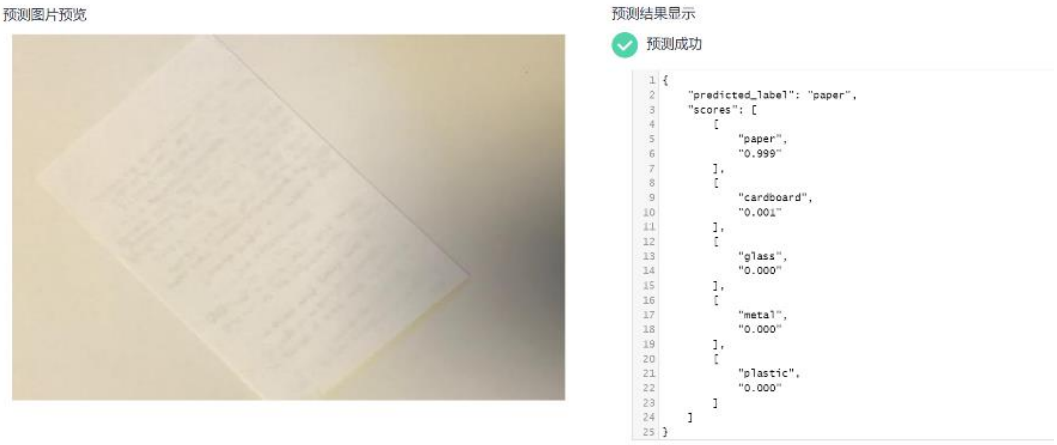

图 5.6　分类正确 3

图 5.7 原图片分类为 glass 类，预测结果为 glass 类，分类正确。

图 5.7　分类正确 4

2) 分类错误

图 5.8 原图片分类为 trash 类，预测结果为 cardboard 类，分类错误。

图 5.8 分类错误 1

图 5.9 原图片分类为 trash 类，预测结果为 plastic 类，分类错误。

图 5.9 分类错误 2

图 5.10 原图片分类为 trash 类，预测结果为 glass 类，分类错误。

图 5.10　分类错误 3

图 5.11 原图片分类为 trash 类，预测结果为 paper 类，分类错误。

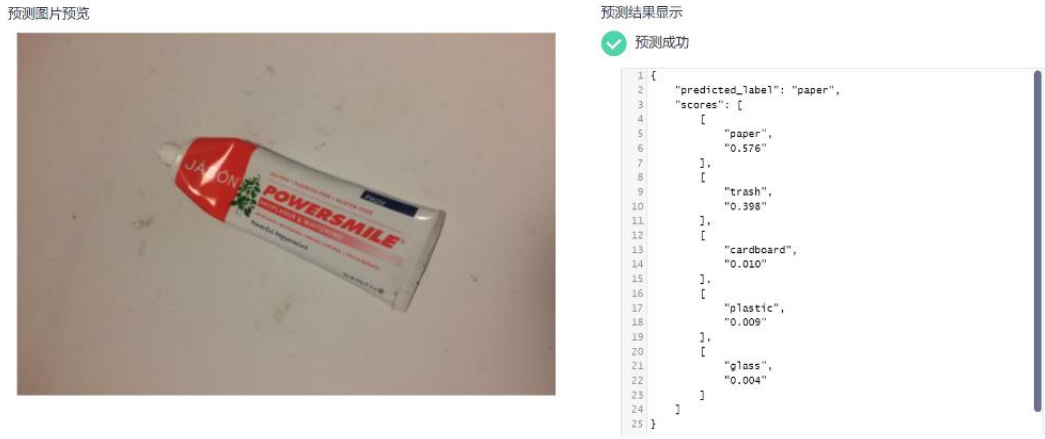

图 5.11　分类错误 4

5.3　错误原因分析

测试中部分图像出现了错误，原因可能是:

1) 训练样本量太少：考虑类别和样本量对应关系；
2) 训练样本分布和测试样本分布不一致;
3) 训练样本经过太多手动裁剪处理，和最终测试样本大小差别较大;
4) 迭代次数不够，即 epochs 太小;
5) 没有使用预训练的更好的模型。

6 总结

通过实验让我们实际动手操作了一次通过人工智能实现图像分类的全流程。数据集到训练，再到测试和调参，还有通过华为云平台的云计算，使我们了解了训练一个深度学习神经网络的全流程。

我们通过 ModelArts 平台，和 Ghost-ResNet-50 实现了对垃圾分类的图像识别。首先加载和处理测试的数据集，然后定义一个卷积神经网络和损失函数，再使用训练，最后使用测试数据集测试模型各项性能。

在计算算力和时间的限制下，可能我们的各项参数无法达到论文中的那么理想。但是我们通过自己的学习和实践，复现了论文中模型训练的全部步骤，并能做到有所创新。

参考文献

[1] 吴寅. 我国垃圾分类现状及建议[J]. 资源节约与环保，2020(9) : 25-26.

[2] 宁凯，张东波，印峰，等. 基于视觉感知的智能扫地机器人的垃圾检测与分类[J]. 中国图象图形学报，2019(8) : 1 358-1 368.

[3] Kai Han, Yunhe Wang, Qi Tan, Jianyuan Guo, Chunjing Xu, and Chang Xu. GhostNet: More Features from Cheap Operations. In CVPR, 2020.

[4] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep Residual Learning for Image Recognition. In CVPR, 2016.

[5] https://github.com/weiaicunzai/pytorch-cifar100

附录

Requirements

- python 3.7
- pytorch1.6.0 + cu101
- tensorboard

Usage

1. 进入文件夹

```
$ cd Ghost_ResNet_50_RealJW
```

2. 数据集

```
.\GarbageClassification\
```

3. 训练模型

```
# use gpu to train ghostresnet
$ python train.py -net ghostresnet -gpu
```

4. 测试模型

```
$ python test.py -net ghostresnet -weights
".\checkpoint\ghostresnet\Tuesday_18_January_2022_15h_31m_2
6s\ghostresnet-10-regular.pth"
```

5. tensorboard

```
python -m tensorboard.main --logdir
".\runs\ghostresnet\Tuesday_18_January_2022_15h_31m_26s"
```