

鸟类细粒度图像分类

组长：周雨童（模型训练调试，ppt制作，使用预训练模型预测鸟种类）

组员：李扬（图形界面、鸟类百科、PDF报告）

该文档主要记录思路，具体操作写在readme文档中

1.问题描述

细粒度图像分类问题是对大类下的子类进行识别。细粒度图像分析任务相对通用图像（General/Generic Images）任务，分类的粒度很小，细粒度图像分类非常困难，主要原因有三：

- 子类之间差异细微：只在某个局部上有细微差异，如狗的眼睛
- 子类内部差异巨大：如姿态、背景带来的差异
- 受视角、背景、遮挡等因素影响较大

2.数据集

使用CUB-200-2011数据集（http://www.vision.caltech.edu/datasets/cub_200_2011/）

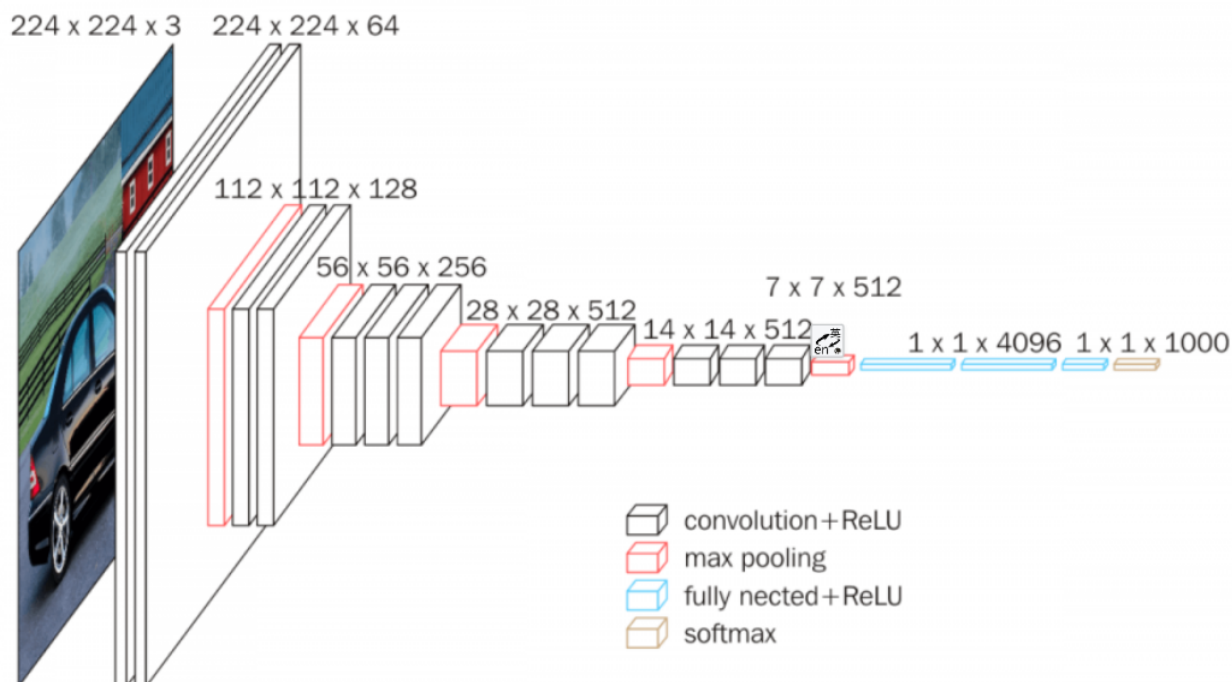
该数据集中包括200种鸟的子类，5994张训练集图像，5794张测试集图像

3.方法

我们尝试了两种方法：Vgg16与ResNet152

Vgg16:

VGG16是牛津大学的K.Simonyan和A.Zisserman提出的卷积神经网络模型^[1]，它通过用多个 3×3 内核大小的过滤器一个接一个地替换大内核大小的过滤器（第一和第二卷积层中分别为 11 和 5）来对 AlexNet 进行改进。



ResNet152:

Resnet^[2]引入了一种称为残差学习单元的结构来缓解深度神经网络的退化。该单元的结构是一个具有快捷连接的前馈网络，可将新输入添加到网络中并生成新输出。该单元的主要优点是它可以在不增加模型复杂性的情况下产生更好的分类精度,是当前图片分类任务中表现最好的神经网络之一

4.模型，尝试与改进

代码正确性与特征提取

做法：写代码从每个鸟类中抽了几张图片，组合成一个小数据集，观察模型能否在小数据集上过拟合

代码：尝试使用预训练好的vgg16提取特征，但vgg16的输出是1000维，于是我在输出的1000维后面加了一个全连接层，输入100维输出200维

```
import torch.nn.functional as F
class vgg16FineTuneModel(nn.Module):
    def __init__(self):
        super().__init__()
        self.model=models.vgg16(weights=VGG16_Weights.IMAGENET1K_V1).eval()
        self.finalFC1=nn.Linear(in_features=1000, out_features=1000)
        self.finalFC2=nn.Linear(in_features=1000, out_features=200)

    def forward(self, x):
        with torch.no_grad():
            self.model.eval()
            x=self.model(x)
        x=self.finalFC1(x)
        x=F.relu(x)
        x=self.finalFC2(x)
        return x
```



在前向传播时，冻结预训练vgg梯度，只训练两个全连接层

详见`vgg16finetuneSmall.ipynb`

结果：代码大致是正确的，预训练模型可以有效提取特征；在提取的小数据集上可以拟合

性能测试

目标：在整个训练集训练，测试性能

做法：加载训练集，使用预训练的vgg16提取特征，将原有vgg最后一层改为输出200维的全连接层，训练

阶段1:微调新加的全连接层

阶段2:调整所有参数

```

import torch.nn.functional as F
class vgg16FineTuneModel(nn.Module):
    def __init__(self):
        super().__init__()
        self.model=models.vgg16(pretrained=True)
        self.model.classifier[-1]=nn.Linear(in_features=4096, out_features=200)

    def forward(self, x):
        x=self.model(x)
        return x

    def frz(self,fz):
        if fz:
            for param in self.model.parameters():
                param.requires_grad=False
            for param in self.model.classifier[-1].parameters():
                param.requires_grad=True
        else:
            for param in self.model.parameters():
                param.requires_grad=True

```

改进2：将训练分为两个步骤：

1.微调新加的全连接层

2.对所有参数整体进行微调

分别对应了model.frz(fz=True)和
model.frz(fz=False)

详见`vgg16finetuneBig.ipynb`

结果：测试集正确率47%，训练集78%，还可以改进准确率、泛化性

提升准确率

代码：尝试使用预训练的ResNet提取特征，具体而言，尝试了resnet101和resnet152

```

import torch.nn.functional as F
class resNetFineTuneModel(nn.Module):
    def __init__(self):
        super().__init__()
        self.res=models.resnet152(pretrained=True)
        # 这一行换成如下代码，就可改变选择加载的预训练模型
        # self.res=models.resnet101(pretrained=True)
        self.res.fc=nn.Linear(2048,200)

    def forward(self, x):
        x=self.res(x)
        return x

    def frz(self,fz):
        if fz:
            for param in self.res.parameters():
                param.requires_grad = False
            for param in self.res.fc.parameters():
                param.requires_grad = True
        else:
            for param in self.res.parameters():
                param.requires_grad = True

```

详见res152.ipynb(Colab)

结果：测试集准确率66%，训练集准确率100%，泛化性有待提升

提升泛化性

尝试1：torchvision.transforms的各种数据增强方法

尝试2:改变transforms.Normalize，计算训练集中图片均值、方差，作为测试集的估计

尝试3:不同batch_size

详见res152.ipynb(Colab)

5.最终结果

- resnet101在数据增强后训练更快，训练集84%,测试集76%
- resnet152+data augmentation（根据训练集计算均值方差），训练集90%-77.7%
- resnet152仅替换最后一层，冻结其他层参数进行stage1训练，测试集65.35%,训练集73.81%，
- 继续训练stage2，最终测试集72.63%，训练集84.80%
- resnet152 stage1训练两轮就开始训练stage2，最终测试集77%，训练集87%
- batch_size太大、太小都不行。设为160时，收敛速度明显慢于50，且batch_size=160的最终准确率低于batch_size=50；batch_size太小

$$Accuracy = \frac{R_a}{R} = 78.16\%$$

6. 图形界面

详见 *GUI.py*

分类表准备

获取所有鸟类名，并获取对应的维基百科网址，手动撰写鸟类简要介绍，最终结果如图所示：

number	category	wiki	character
1	Black_footed_Albatross	https://en.wikipedia.org/wiki/Black-footed_Albatross	黑脚信天翁,黑色羽毛,喙底部和眼睛下方有白色斑纹
2	Laysan_Albatross	https://en.wikipedia.org/wiki/Laysan_Albatross	黑背信天翁
3	Sooty_Albatross	https://en.wikipedia.org/wiki/Sooty_Albatross	乌黑信天翁,煤烟色,头部两侧有较深的阴影
4	Groove_billed_Ani	https://en.wikipedia.org/wiki/Groove-billed_Ani	沟嘴美洲鸚鵡,尾巴长,喙又大又弯
5	Crested_Auklet	https://en.wikipedia.org/wiki/Crested_Auklet	凤头海雀,喙呈橙红色和黄色,虹膜呈黄白色
6	Least_Auklet	https://en.wikipedia.org/wiki/Least_Auklet	最小海雀,海雀中最小的种类
7	Parakeet_Auklet	https://en.wikipedia.org/wiki/Parakeet_Auklet	长尾小鸚鵡,短的橙色喙,嘴巴向上翘
8	Phalarope_Auklet	https://en.wikipedia.org/wiki/Phalarope_Auklet	扁嘴海雀

- 注：这里仅撰写了少量鸟类的特征用做功能介绍，剩下种类的介绍由于是重复性工作，有需要的话也可以完成。

界面设计

使用 *thinter* 库作为框架，设计了“打开图片”按钮，本地选择图片后会在下方展示，并给出分类，同时点击右上方的“Wiki”可直接跳转到对应鸟类的维基百科，适合进一步了解该鸟类，界面如图所示：



参考文献

- [1] Simonyan, K., and A. Zisserman. "Very Deep Convolutional Networks for Large-Scale Image Recognition." 3rd International Conference on Learning Representations (ICLR 2015), Computational and Biological Learning Society, 2015, pp. 1–14.
- [2] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 770-778, doi: 10.1109/CVPR.2016.90.