

人工智能大作业2 海洋生物分类

自02 吴浩睿 2020011063

1. 对海洋生物的19分类问题

1.1设计一个卷积神经网络模型来解决该分类问题

本次使用预训练的vgg19模型，特点是使用小卷积核、小池化核。因为计算资源有限，而越大的卷积核计算量越大，所以使用小卷积核也是合理的。

1.2采用一个深度学习框架来实现你设计的模型

本次使用pytorch来实现。

1.3提出合理的指标来评价你的测试结果

①准确率：每一类的准确率=该类的数据中预测正确的数量÷该类的数据的总数量。

②混淆矩阵：混淆矩阵的每一列代表了预测类别，每一行代表了数据的真实归属类别。

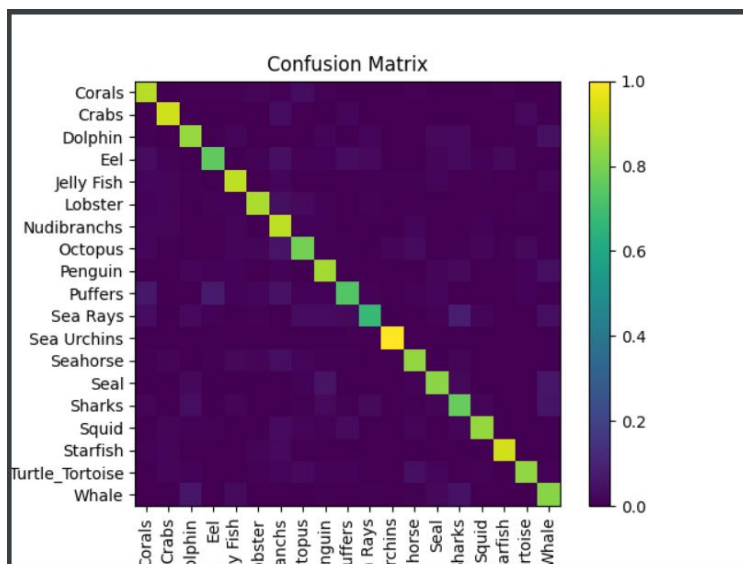
③F1： $F1=2*(precision*recall)/(precision+recall)$ ，同时兼顾了分类模型的精确率和召回率。F1分数可以看作是模型精确率和召回率的一种调和平均，它的最大值是1，最小值是0

1.4根据你的测试结果计算这些指标的值，解释其含义

①准确率：每一类的准确率=该类的数据中预测正确的数量÷该类的数据的总数量。

```
Accuracy of Corals : 87 %
Accuracy of Crabs : 93 %
Accuracy of Dolphin : 86 %
Accuracy of Eel : 74 %
Accuracy of Jelly Fish : 92 %
Accuracy of Lobster : 83 %
Accuracy of Nudibranchs : 83 %
Accuracy of Octopus : 74 %
Accuracy of Penguin : 77 %
Accuracy of Puffers : 74 %
Accuracy of Sea Rays : 67 %
Accuracy of Sea Urchins : 97 %
Accuracy of Seahorse : 82 %
Accuracy of Seal : 83 %
Accuracy of Sharks : 73 %
Accuracy of Squid : 73 %
Accuracy of Starfish : 87 %
Accuracy of Turtle_Tortoise : 87 %
Accuracy of Whale : 83 %
average: 0.8236607142857143
```

②混淆矩阵



可以看到对角线的比例接近1，其他非对角线的颜色接近0，说明模型可以较好地预测分类。

③F1分数：

19个类的F1分数和全局的平均F1分数分别是：

```
19个类的F1分数分别是 [0.82033898 0.94244604 0.82550336 0.75576037 0.9516129 0.80733945
0.81368821 0.7983871 0.82524272 0.78297872 0.84848485 0.97169811
0.86381323 0.8172043 0.82527881 0.83185841 0.95019157 0.91176471
0.82638889]
全局的F1分数是 0.8547185430463576
```

2.改进：输入数据的处理

2.1对训练数据做三种处理：

- ①水平翻转
- ②随机旋转-15到15度
- ③依概率0.1转为灰度图

```
# 准备数据集,对训练集预处理
data_transform = transforms.Compose([
    transforms.RandomResizedCrop(224), # 裁剪
    transforms.RandomHorizontalFlip(), # 翻转
    transforms.RandomRotation(15), # 随机旋转-15到15度
    transforms.RandomGrayscale(), # 依概率0.1转为灰度图
    transforms.ToTensor()
])
```

2.2训练结果

①准确率：

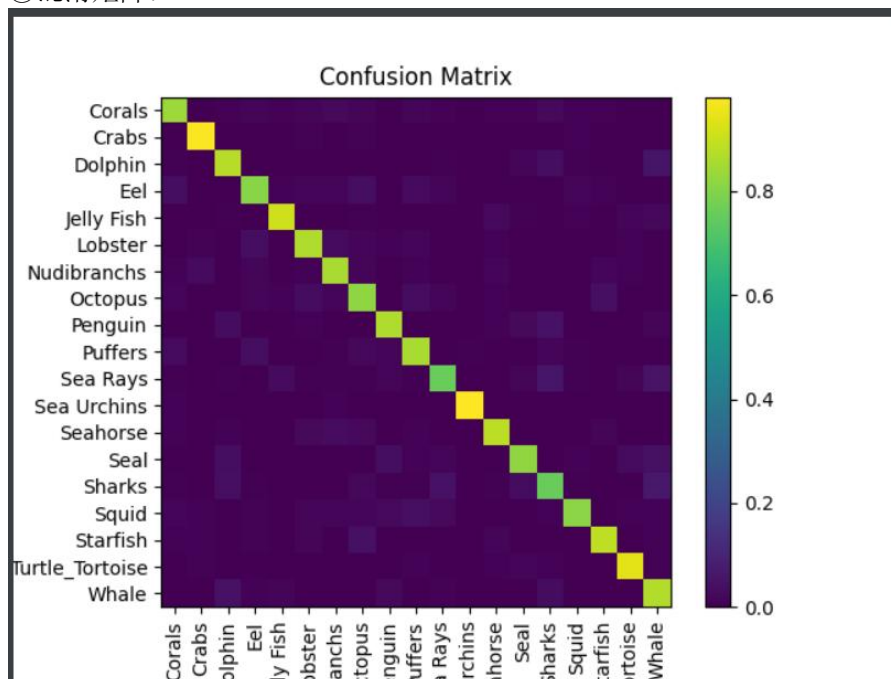
```

Accuracy of Corals : 87 %
Accuracy of Crabs : 96 %
Accuracy of Dolphin : 86 %
Accuracy of Eel : 77 %
Accuracy of Jelly Fish : 94 %
Accuracy of Lobster : 83 %
Accuracy of Nudibranchs : 89 %
Accuracy of Octopus : 82 %
Accuracy of Penguin : 86 %
Accuracy of Puffers : 80 %
Accuracy of Sea Rays : 74 %
Accuracy of Sea Urchins : 97 %
Accuracy of Seahorse : 89 %
Accuracy of Seal : 83 %
Accuracy of Sharks : 78 %
Accuracy of Squid : 81 %
Accuracy of Starfish : 94 %
Accuracy of Turtle_Tortoise : 93 %
Accuracy of Whale : 89 %
average: 0.8666801948051948

```

相比无预处理的训练结果有一定的改进

②混淆矩阵:



与无预处理的训练结果区别不明显，都是对角线的比例接近1，其他非对角线的颜色接近0。

③F1分数

```

19个类的F1分数分别是 [0.84496124 0.95934959 0.85714286 0.81481481 0.93048128 0.84821429
0.8411215 0.80152672 0.8685259 0.84644195 0.79518072 0.98604651
0.88188976 0.83333333 0.77441077 0.85355649 0.89380531 0.92741935
0.80141844]

全局的F1分数是 0.8609271523178808

```

每个类各自的F1分数相比无预处理有的增加有的变小，但全局的F1分数变大了。

总体来说，使用了水平翻转、随机旋转-15到15度、依概率0.1转为灰度图的预处理后模型的预测准确度有了一定的提高，可能是由于预处理增加了训练图片的数量，减少模型学到数据分布的可能性，提升模型的泛化能力。

3.遇到的问题和展望

3.1问题1: vgg19的输入是224*224的图片，而本次分类任务的图片不都是224*224的。

解决办法：在训练之前对图片做预处理，将图片都裁剪为224*224的大小，同时位置是随机的，也使训练后的模型更泛化。

3.2问题2: 最初的模型最后有softmax，训练不是很顺利。

解决办法：查阅材料后知道CrossEntropyLoss()函数里已经有softmax，将模型最后的softmax删去之后训练变得顺利。

3.3问题3: 在画混淆矩阵时报错提示“不能将gpu tensor 转为ndarray类型”。

解决办法：先将gpu上的tensor拷贝到cpu上再用cpu上的tensor画混淆矩阵。

3.4展望

本次使用vgg19卷积网络的预训练模型来提取图片特征，如果在算力允许的情况下，使用ResNet或DenseNet可能会取得更好的预测效果。