机器学习

Project:Classify-leaves

杨亿酬 3230105697

2025-12-15 ~ 2025-12-19

1.实验目的

- train.csv: training set(18353 sets, 176 unique values)
- test.csv: testing set(8800 sets)
- images/: the folder contains all images

基于18353个训练图像与标签预测8800个树叶类别，评估指标为分类准确度

2.实验过程

Take a glance

`glance.py`展示数据的基本信息，包括数据量，类别数量，每个类别的样本数和占比
同时输出图像的大小颜色等信息，并计算归一化后图像各通道的平均像素值与方差（用于后续数据预处理），
最后通过train_valid_split分割训练集与验证集（90%训练集，10%验证集，期待模型学习更多输入信息）

```
=== Data Information ===
train data num:18353
test data num:8800
          image            label
0  images/0.jpg  maclura_pomifera
1  images/1.jpg  maclura_pomifera
2  images/2.jpg  maclura_pomifera
3  images/3.jpg  maclura_pomifera
4  images/4.jpg  maclura_pomifera
             image
0  images/18353.jpg
1  images/18354.jpg
2  images/18355.jpg
3  images/18356.jpg
4  images/18357.jpg

=== Label Check ===
unique label num:176

=== Label Distribution ===
Label                                Count      Percentage
------------------------------------------------------------
maclura_pomifera                     353        1.92%
ulmus_rubra                          235        1.28%
prunus_virginiana                    223        1.22%
acer_rubrum                          217        1.18%
broussonettia_papyrifera             214        1.17%
```

```
Image checking...

=== Image size statistics ===
  (224, 224): 27153 (100.0%)
✓ all images share the same size

=== color mode statistics ===
  RGB: 27153 (100.0%)

=== train_valid split ===
original train data size: 18353
validation percentage: 10.0%
train size after split: 16517
valid size after split: 1836
Load train images succeed.
Load valid images succeed.
=== image normalization ===
mean (R, G, B): [0.485 0.456 0.406]
variance (R, G, B): [0.229 0.224 0.225]
100%|| 27153/27153 [02:00<00:
00, 226.26it/s]
Image normalized complete.
Normalized data saved to: ./../classify-leaves/normalized
  - train_images.npy: (16517, 224, 224, 3)
  - valid_images.npy: (1836, 224, 224, 3)
  - test_images.npy: (8800, 224, 224, 3)

=== label extracting ===
num_classes: 176
label saved to ./../classify-leaves/normalized/
```

preprocess

`preprocess.py`进行数据的预处理，主要包括数据的归一化normalization和标签的提取，生成可以输入模型的.npy文件

```
def normalize_single(self, image_paths):
    with Image.open(image_path) as img:
        img_array = np.array(img, dtype=np.float32) / 255.0
        normalized = (img_array - self.mean) / self.std
    return normalized
```

通过除以255.0归一化，减去均值mean再除以方差std得到标准化的输入图像

base_model

首先尝试基本的多层卷积神经网络：输入224*224的像素数组，经过多层Conv2d→BatchNorm→ReLU和MaxPool，最后通过一个全连接层输出一个分类结果
选用最基本的优化器SGDmomentum和调度器ReduceLROnPlateau进行5-8次epoches的迭代 得到50%+的准确率

model

ResNet50Pretrained Model

使用预训练的`resnet50`模型，修改最后的全连接层输出预测类别

```
        self.backbone = models.resnet50(weights=ResNet50_Weights.IMAGENET1K_V2)

        # last layer
        num_features = self.backbone.fc.in_features
        self.backbone.fc = nn.Sequential(
            nn.Dropout(0.5),
            nn.Linear(num_features, 512),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(512, num_classes)
        )
```

Data Augmentation

在输入数据进入模型之前对数据进行增强，包括resize,水平与垂直裁切，旋转，视角扭曲，弹性形变，仿射变换，颜色变换，转化为张量后进行随机擦除和高斯滤波
对验证集只进行大小变换与裁切
对测试集也需要先进行变换再输入

optimizer & schduler

```
optimizer = {SGD, AdamW}
scheduler = {ReduceLROnPlateau, CosineAnnealingLR,CosineAnnealingWarmRestarts,
OneCycleLR}
```

确认SGDmomentum与CosineAnnealingWarmRestarts收敛效果最佳
在验证集上获得超过90%的准确率

```
------------------------------------------------------------
Epoch [3/50], Batch [100/517], Loss: 0.2024, Acc: 93.34%, LR: 0.000342
Epoch [3/50], Batch [200/517], Loss: 0.1809, Acc: 94.28%, LR: 0.000342
Epoch [3/50], Batch [300/517], Loss: 0.1877, Acc: 93.22%, LR: 0.000342
Epoch [3/50], Batch [400/517], Loss: 0.1746, Acc: 94.00%, LR: 0.000342
Epoch [3/50], Batch [500/517], Loss: 0.1659, Acc: 94.12%, LR: 0.000342
✅ new best, acc: 91.34%
model saved to: ./../checkpoints/best_model1.pth
Epoch [3/50] summary:
  training loss: 0.1684, training acc: 94.14%
  validation loss: 0.2734, validation_acc: 91.34%
  learning_rate: 0.000161
------------------------------------------------------------
Epoch [4/50], Batch [100/517], Loss: 0.1611, Acc: 94.00%, LR: 0.000161
Epoch [4/50], Batch [200/517], Loss: 0.1494, Acc: 94.44%, LR: 0.000161
Epoch [4/50], Batch [300/517], Loss: 0.1528, Acc: 94.81%, LR: 0.000161
Epoch [4/50], Batch [400/517], Loss: 0.1628, Acc: 94.53%, LR: 0.000161
Epoch [4/50], Batch [500/517], Loss: 0.1595, Acc: 94.16%, LR: 0.000161
✅ new best, acc: 91.67%
model saved to: ./../checkpoints/best_model1.pth
Epoch [4/50] summary:
  training loss: 0.1563, training acc: 94.48%
  validation loss: 0.2641, validation_acc: 91.67%
  learning_rate: 0.010000
------------------------------------------------------------
```

predict

首先获取索引与标签的映射
将测试数据进行变换后输入模型进行预测，将各输入对应概率最高的输出类别作为结果输出到final_submission.csv

3.实验结果

在test.csv数据集上获得超过90%的准确率，预测文件final_submission.csv