

摘要：第二次采集三叶青块根图像，训练代码的优化

## 1. 实验准备

与第一次拍摄的任务安排一样，由队员对每个产地的块根进行不同角度的拍照（非纯白背景），每个产地拍 250 张左右。

## 2. 实验步骤

### 2.1. 第二次采集三叶青块根照片

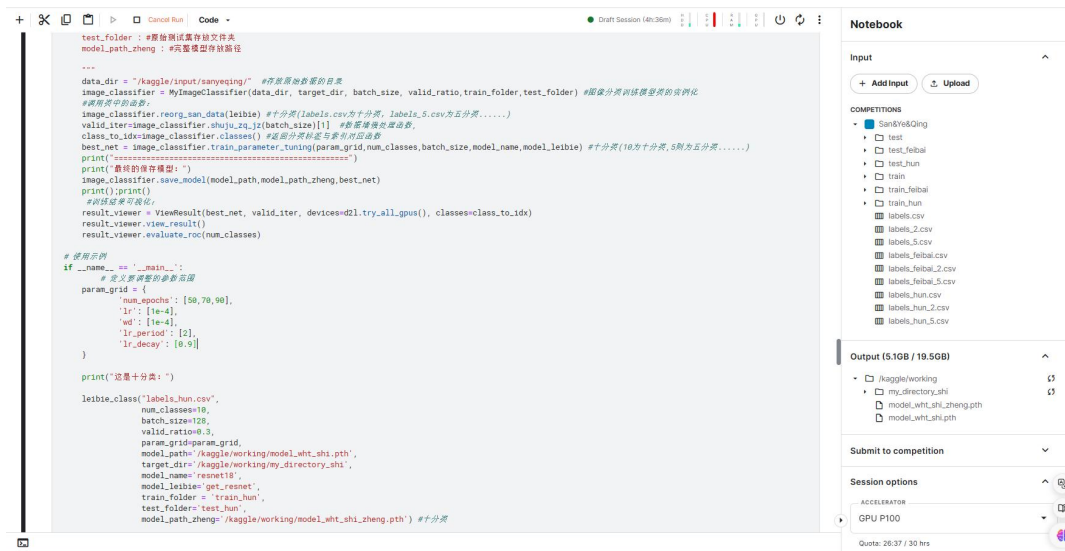
与第一次拍摄背景照片为纯白不同，这次拍摄的照片背景非纯白，而是放置在不同背景下：草地上，水泥路上、瓷砖上、交界处、手上...。（如下图所示）



### 2.2. 对原有的训练代码进行改善

对 kaggle 上的训练代码进行优化，方便训练不同模型以及其余成员的训练，添加了测试集预测代码

优化后的训练代码：



## 测试集预测代码运行：

```
classes_2 = {0: 'not_cu', 1: 'cu'}

#整体测试准确率：（主要修改处，将前面设置，实际修改）
predict_acc_all(labels_path="/kaggle/input/sanyeqing/labels.csv",
                 image_dir="/kaggle/working/my_directory_shi/train_valid_test/test/unknown/",
                 num_classes=10,
                 model_name='resnet152',
                 model_weight_ptn="/kaggle/working/model_whit_shi.pth",
                 classes=classes_10,
                 model_leibie = "get_resnet")

#单张图片的预测概率（可选看）
def show_image(img, title=None):
    """显示图像"""
    plt.imshow(img)
    if title:
        plt.title(title)
    # plt.axis('off') # 不显示坐标轴
    plt.show()

# 在predict函数调用之前，添加如下代码来显示图像
img = Image.open("/kaggle/working/my_directory_shi/train_valid_test/test/unknown/12.png")
show_image(img)
predicted_class, max_prob, class_probs = predict(img, num_classes=10,
                                                model_name='resnet152',
                                                model_weight_ptn="/kaggle/working/model_whit_shi.pth",
                                                classes=classes_10,
                                                model_leibie = 'get_resnet'
                                                )

print(f"最可能的类别: {predicted_class}")
print(f"最大分类概率: {max_prob}")
print(class_probs)

/opt/conda/lib/python3.10/site-packages/torchvision/models/_utils.py:208: UserWarning: The parameter 'pretrained' is deprecated since 0.13 and may be removed in the future, please use 'weights' instead.
  warnings.warn(
/opt/conda/lib/python3.10/site-packages/torchvision/models/_utils.py:223: UserWarning: Arguments other than a weight enum or 'None' for 'weights' are deprecated since 0.13 and may be removed in the future. The current behavior is equivalent to passing 'weights=ResNet152_Weights.IMAGENET1K_V1'. You can also use 'weights=ResNet152_Weights.DEFAULT' to get the most up-to-date weights.
  warnings.warn(msg)
真实: ZJHZ 预测: ZJHZ 最大概率: 0.99902 所有概率: {'0': 0.00056, '1': 0.0, '2': 1e-05, '3': 8e-05, '4': 0.00031, '5': 1e-05, '6': 0.0, '7': 0.0, '8': 0.0, '9': 0.99902}
真实: SXOA 预测: SXOA 最大概率: 0.88904 所有概率: {'0': 0.0, '1': 0.00062, '2': 0.00414, '3': 0.0, '4': 0.00507, '5': 0.88904, '6': 0.0, '7': 1e-05, '8': 2e-05, '9': 0.0111}
真实: ZJZT 预测: ZJZT 最大概率: 0.99766 所有概率: {'0': 0.0, '1': 0.00229, '2': 0.0, '3': 0.0, '4': 5e-05, '5': 0.0, '6': 0.0, '7': 0.0, '8': 0.99766, '9': 0.0}
真实: GXVL 预测: GXVL 最大概率: 0.99344 所有概率: {'0': 0.0, '1': 8e-05, '2': 0.99344, '3': 0.0, '4': 1e-05, '5': 0.00647, '6': 0.0, '7': 0.0, '8': 0.0, '9': 0.0}
真实: YKXSN 预测: YKXSN 最大概率: 0.99267 所有概率: {'0': 4e-05, '1': 8e-05, '2': 0.00572, '3': 5e-05, '4': 0.0, '5': 0.0013, '6': 1e-05, '7': 0.99267, '8': 2e-05, '9': 0.0011}
真实: GXVL 预测: GXVL 最大概率: 0.95701 所有概率: {'0': 0.0, '1': 4e-05, '2': 0.95701, '3': 0.0, '4': 5e-05, '5': 0.04289, '6': 0.0, '7': 1e-05, '8': 0.0, '9': 0.0}
真实: ZJZT 预测: ZJZT 最大概率: 0.98992 所有概率: {'0': 1e-05, '1': 0.00754, '2': 0.0, '3': 0.0, '4': 0.00042, '5': 1e-05, '6': 0.00042, '7': 0.00167, '8': 0.98992, '9': 1e-05}
真实: GZD 预测: GZD 最大概率: 0.99963 所有概率: {'0': 1e-05, '1': 2e-05, '2': 0.00011, '3': 0.99963, '4': 2e-05, '5': 1e-05, '6': 2e-05, '7': 0.00017, '8': 0.0, '9': 1e-05}
真实: SXOA 预测: SXOA 最大概率: 0.90005 所有概率: {'0': 0.0, '1': 0.00011, '2': 0.0097, '3': 0.0, '4': 0.0, '5': 0.90005, '6': 0.0, '7': 0.0, '8': 0.0, '9': 0.00014}
```

## 2.3. 训练结果记录

模型的训练是一个漫长的过程，要训练不同模型，每个模型的参数要调整以达到最高准确率。而且一个模型的好坏在正式生产应用中时要考虑更多，如模型大小、模型推理图片的速度等，这时候就要综合考虑实际中应该挑选哪个模型，而不仅仅考虑准确率。

故做一个表格记录训练结果：

文件上传

U U M U

Files

- train - 白背景--三叶青训练集
- test - 白背景--三叶青测试集
- labels.csv --白背景-十分类（即十个产地）的标签文件
- labels\_5.csv --白背景-五分类（即五个省份）的标签文件
- labels\_2.csv --白背景-二分类（即是否为新产）的标签文件
- train\_feibai - 非白背景--三叶青训练集
- test\_feibai - 非白背景--三叶青测试集
- labels\_feibai.csv --非白背景-十分类（即十个产地）的标签文件
- labels\_feibai\_5.csv --非白背景-五分类（即五个省份）的标签文件
- labels\_feibai\_2.csv --非白背景-二分类（即是否为新产）的标签文件
- train\_hun - 第一、二次照片混合--三叶青训练集
- test\_hun - 第一、二次照片混合--三叶青测试集
- labels\_hun.csv --第一、二次照片混合-十分类（即十个产地）的标签文件
- labels\_hun\_5.csv --第一、二次照片混合-五分类（即五个省份）的标签文件
- labels\_hun\_2.csv --第一、二次照片混合-二分类（即是否为新产）的标签文件

+ Code + Markdown

汇总记录

用于记录汇总队伍中的队员们的模型训练信息，用于直观展示项目的图像分类模型的训练效果

表格形式：注意：is\_best 标记为@代表你目前valid\_acc或test\_acc最高的一行，要实时更新

示例：（请根据实际请款填写表格！！）

时间	模型	loss_acc	valid_acc	test_acc	photo_time	num_epochs	lr	wd	lr_period	lr_decay	num_classes	bath_size	valid_ratio	model_path	model_size	shuju_leixing	is_best
2024/4/30	efficientnet-b0																@
	efficientnet-b1																
2024/5/1	efficientnet-b2																
	efficientnet-b3																

3. 实验结果

近几天来，通过不断训练，获得了一些训练结果，并记录在表格中

记录汇总队伍中的队员们的模型训练信息，用于直观展示项目的图像分类模型的训练效果

是：is\_best 标记为@代表你目前valid\_acc或test\_acc最高的一行，要实时更新

请实际请款填写表格！！）

时间	模型	loss_acc	valid_acc	test_acc	photo_time	num_epochs	lr	wd	lr_period	lr_decay	num_classes	bath_size	valid_ratio	model_path	model_size	shuju_leixing	is_best
2024/4/30	efficientnet-b0																@
	efficientnet-b1																
2024/5/1	efficientnet-b2																
	efficientnet-b3																

的训练结果

	模型	loss_acc	valid_acc	test_acc	photo_time	num_epochs	lr	wd	lr_period	lr_decay	num_classes	bath_size	valid_ratio	model_path	model_size	shuju_leixing	is_best
/30	resnet101	train loss 0.149,train acc 0.954, valid acc 0.979,valid loss 0.073	0.979	96.80%	1.213456秒	30	0.0001	0.0001	2	0.9	10	128	0.3	resnet101_sanyeqing_10.pth	168.692KB	bai	
/5	resnet18	train loss 0.171,train acc 0.942, valid acc 0.964,valid loss 0.103	0.964	95.20%	0.319273秒	30	0.0001	0.0001	2	0.9	10	128	0.3	resnet18_sanyeqing_10_1.pth	44.260KB	bai	
	resnet18	train loss 0.452,train acc 0.845, valid acc 0.859,valid loss 0.405	0.859	90.40%	0.315380秒	70	0.0001	0.0001	2	0.9	10	128	0.3	resnet18_sanyeqing_feibai_10_1.pth	44.260KB	feibai	
	resnet34	train loss 0.197,train acc 0.928, valid acc 0.962,valid loss 0.129	0.962	96.00%	0.580408秒	30	0.0001	0.0001	2	0.9	10	128	0.3	resnet34_sanyeqing_10_2.pth	83.806KB	bai	
	resnet152	train loss 0.145,train acc 0.956, valid acc 0.984,valid loss 0.063	0.984	96.80%	1.711858秒	100	0.0001	0.0001	2	0.9	10	128	0.3	resnet152_sanyeqing_10_4.pth	230.080KB	bai	@
	resnet101	train loss 0.333,train acc 0.883, valid acc 0.903,valid loss 0.317	0.903	90.80%	1.233189秒	110	0.0001	0.0001	2	0.9	10	128	0.3	resnet101_sanyeqing_feibai_10_2.pth	168.692KB	feibai	
	resnet101	train loss 0.351,train acc 0.883, valid acc 0.900,valid loss 0.318	0.900	91.60%	1.264539秒	130	0.0001	0.0001	2	0.9	10	128	0.3	resnet101_sanyeqing_feibai_10_3.pth	168.692KB	feibai	
/6	resnet152	train loss 0.142,train acc 0.954, valid acc 0.979,valid loss 0.066	0.979	96.00%	1.705440秒	70	0.0001	0.0001	2	0.9	10	128	0.3	resnet152_sanyeqing_10_3.pth	230.080KB	bai	

实验总结