

基于深度学习的果园病害检测

South China Agricultural University

答辩人: 曹丹伶 指导老师: 徐兴

CONTENT

Part 01 研究内容

Part 02 实验过程

Part 03 总结展望



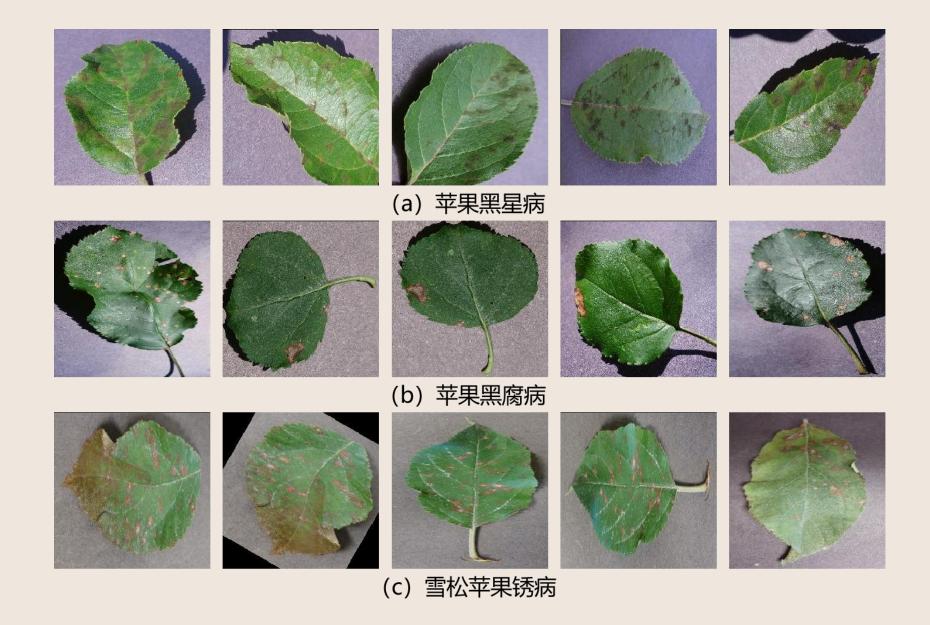


● 研究内容

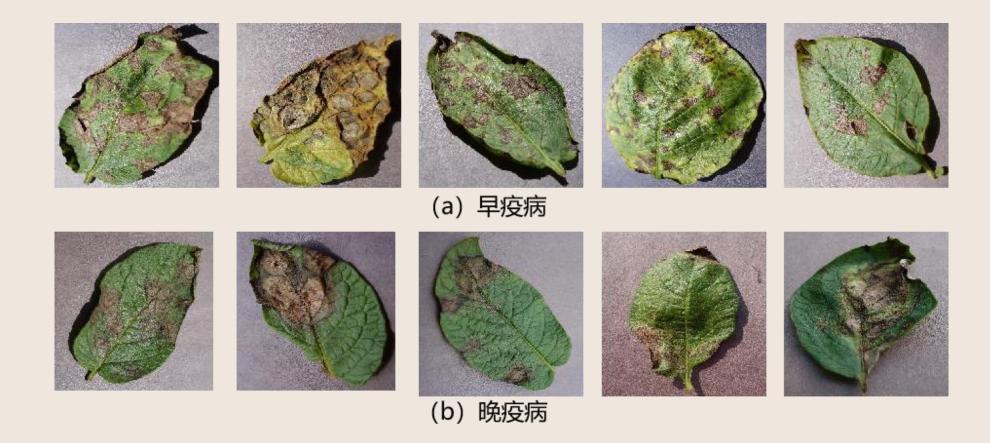
名称	修改日期	类型	大小
AppleApple_scab	2022/4/29 9:10	文件夹	
AppleBlack_rot	2022/4/29 9:10	文件夹	
AppleCedar_apple_rust	2022/4/29 9:11	文件夹	
Applehealthy	2022/4/29 9:11	文件夹	
Corn_(maize)Cercospora_leaf_spot Gray_I	2022/4/29 9:12	文件夹	
Corn_(maize)Common_rust_	2022/4/29 9:12	文件夹	
Corn_(maize)healthy	2022/4/29 9:12	文件夹	
Corn_(maize)Northern_Leaf_Blight	2022/4/29 9:12	文件夹	
PotatoEarly_blight	2022/4/29 9:15	文件夹	
Potatohealthy	2022/4/29 9:15	文件夹	
PotatoLate_blight	2022/4/29 9:16	文件夹	
TomatoBacterial_spot	2022/4/29 9:17	文件夹	
TomatoEarly_blight	2022/4/29 9:17	文件夹	
Tomatohealthy	2022/4/29 9:18	文件夹	
TomatoLate_blight	2022/4/29 9:18	文件夹	
TomatoLeaf_Mold	2022/4/29 9:18	文件夹	
TomatoSeptoria_leaf_spot	2022/4/29 9:54	文件夹	
TomatoSpider_mites Two-spotted_spider	2022/4/29 9:55	文件夹	
TomatoTarget_Spot	2022/4/29 9:55	文件夹	
TomatoTomato_mosaic_virus	2022/4/29 9:56	文件夹	
TomatoTomato_Yellow_Leaf_Curl_Virus	2022/4/29 9:56	文件夹	

将苹果、土豆、西红柿、玉米四个集群的数据集分别以**4:1的比例分为训练集,和测试集。**我们对该数据集进行基于深度学习架构的卷积神经网络模型的**纹理特征进行提取并分析**,从而对四个数据集群共21种叶片病虫害的识别分类研究。最后**对CNN网络模型得出的分类准确率进行比较,得出性能最佳的模型**,运用于果园的病害虫检测中。

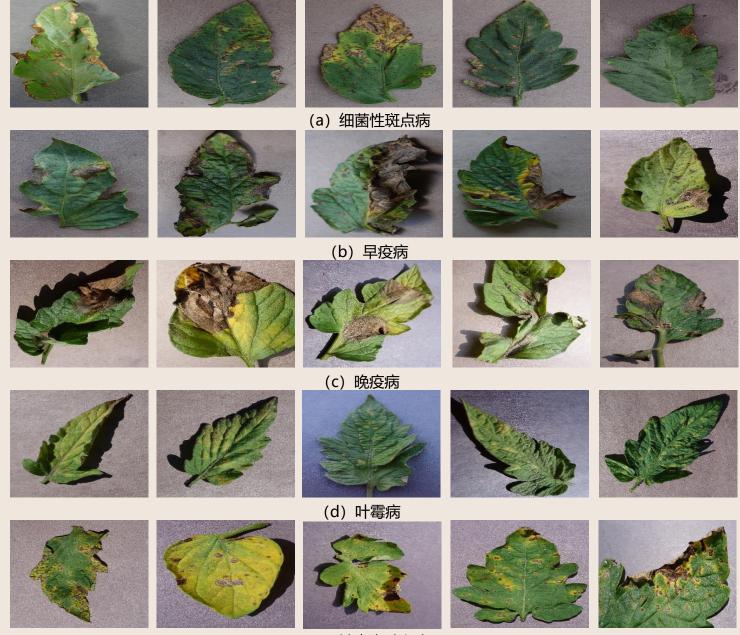
● 研究内容——苹果叶片病害种类示例



● 研究内容——土豆叶片病害种类示例



● 研究内容——番茄叶片病害种类示例

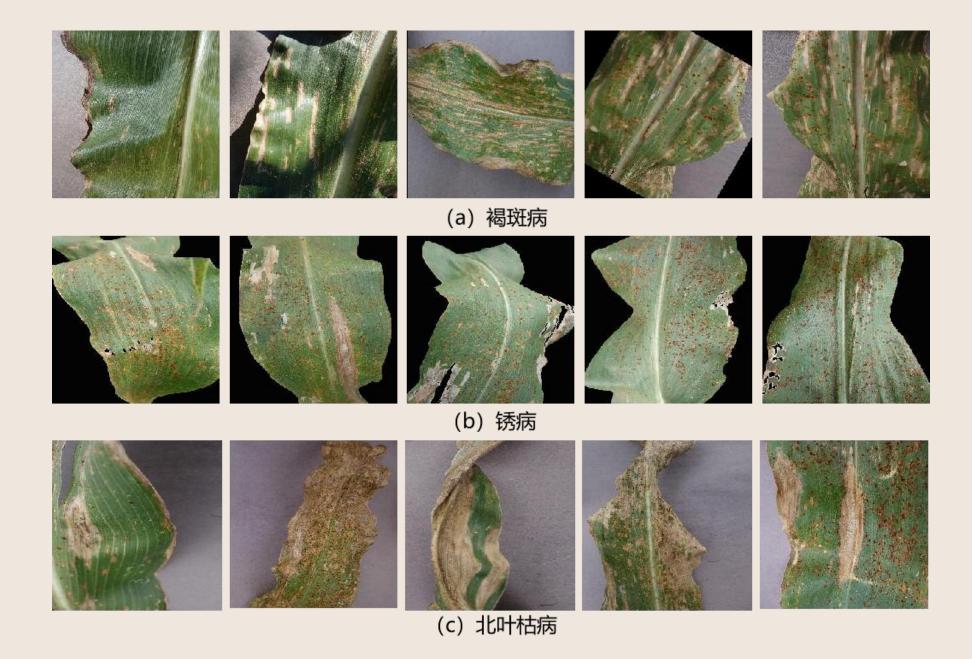


(e) 针壳孢叶斑病

● 研究内容——番茄叶片病害种类示例



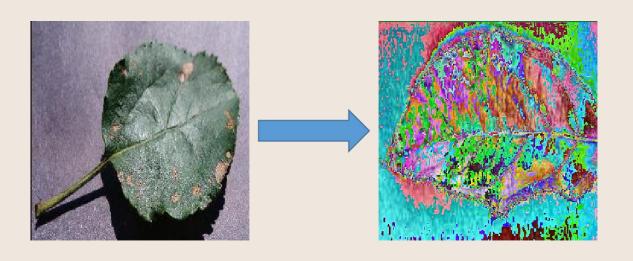
● 研究内容——玉米叶片病害种类示例





● 实验过程——数据预处理

```
from torchvision.transforms import transforms
data_transform = transforms.Compose([
    transforms.Resize(224), # 缩放图片(Image), 保持长宽比不变, 最短边为32像素
    # transforms.CenterCrop(32), # 从图片中间切出32*32的图片
    transforms.ToTensor(), # 将图片(Image)转成Tensor, 归一化至[0, 1]
    transforms.Normalize(mean=[0.3867699, 0.4292917, 0.39402938], std=[0.18648247, 0.16961637, 0.21869354]) # 标准化至[-1, 1], 规定均值和标准差
])
```



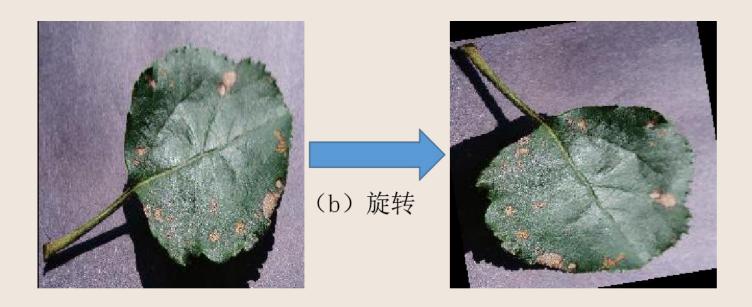
数据归一化思路:将图片3个通道中的数据都输出为[-1,1]之间,收敛公式:

output = (input - mean) / std

● 实验过程——数据预处理



Transforms.CenterCrop()



Transforms.RandomRotate()

● 实验过程——评价指标

准确率: 分类正确的样本数量占总样本数量的比例

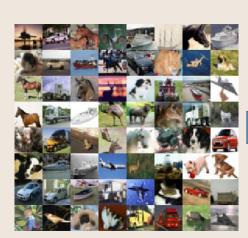
Accuracy
$$= \frac{TP + TN}{TP + FP + TN + FN}$$

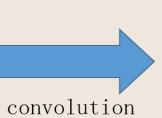
损失函数: 用来表现预测与实际数据的差距程度

```
loss_fn = nn.CrossEntropyLoss()
loss = loss_fn(output,labels)
running_loss += loss.item()
```

● 实验过程——卷积神经网络

卷积神经网络是多隐藏层的深度学习模型。由小型网络组成的网络架构,由具有非线性、权重参数和偏置项的神经元构成。神经元输出一个损失值,整个网络基于该损失值通过反向传播进行重排列。简单的全连接层有更多的权重,因为它们的存储信息用于以权重的形式处理一切。卷积神经网络的基本构造包括输入层、卷积层、池化层、激活函数、全连接层和输出层组成。





卷积层: 提取图像中的局部特征

self.conv1=Conv2d(in_channels=3,out _channels=6,kernel_size=3,stride=1,pa dding=0)

● 实验过程——卷积神经网络







池化层:对提取到的特征再进一步筛选

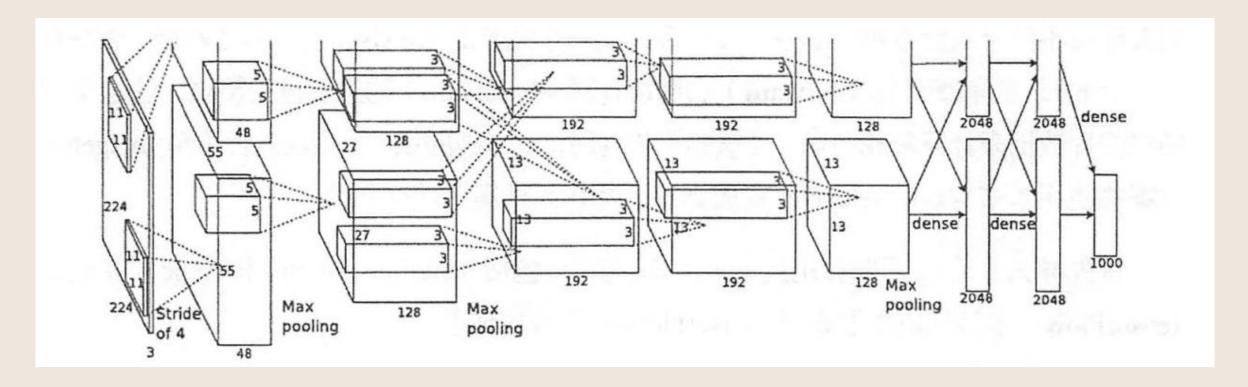
self.maxpool1=MaxPool2d(kernel size=3)



非线性处理

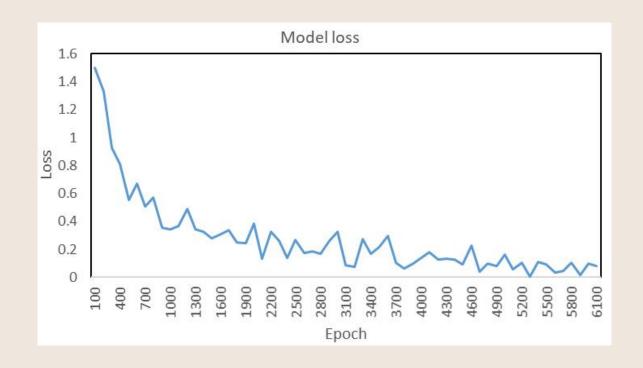


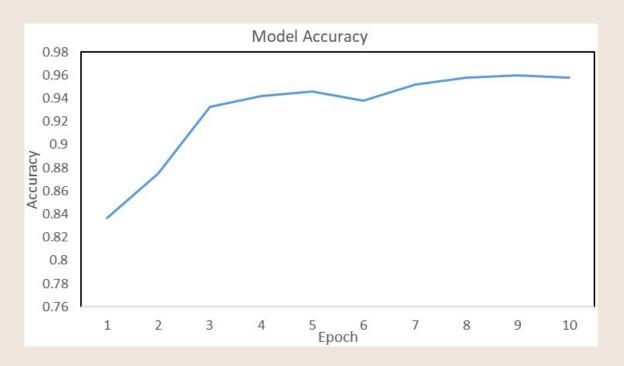
激活函数: 非线性因素,将经过 卷积池化操作得到的特征图进行非线 性变换,便于解决于非线性问题模型。



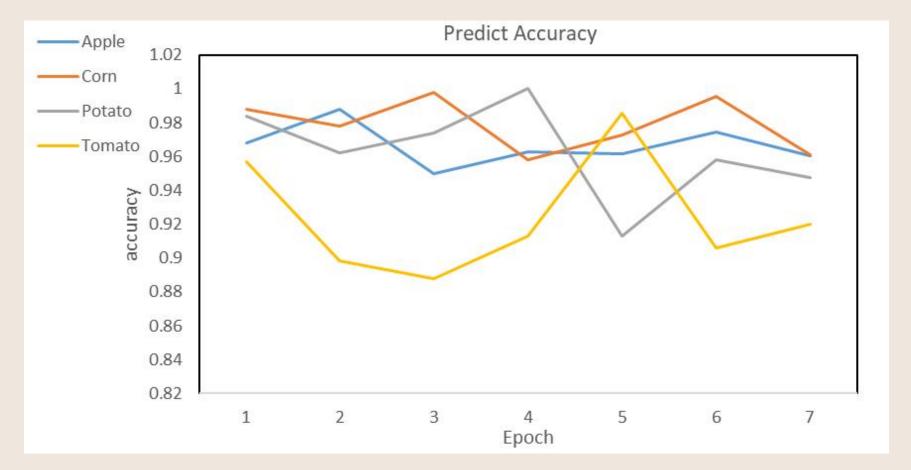
AlexNet网络有5层卷积层,3层全连接层,共计深度为8层,参数个数共60M个,有650k神经元个数,可分类1000数目的分类量。ReLu函数作为激活函数,ReLu函数在梯度衰减方面比其他函数快。池化处理方式采用池化窗口的大小大于步长的overlapping pooling,这样每次池化都会有重叠的部分,便可以避免过拟合发生。在AlexNet网络结构的搭建中,加入了dropout函数,这样在训练神经网络模型时,可以以一定概率p随机丢失一些神经元以及神经结点,这样能提高神经网络的泛华能力,减少过拟合现象。

```
义为提取特征的层结构
           nn.Conv2d(3, 48, kernel_size=11, stride=4, padding=2),
           # 第一层 # input[3, 224, 224] output[48, 55, 55],他只用了一半的卷积核(padding=(1, 2),计
算后是小数,就又一样了)
           nn.ReLU(inplace=True), # inplace是pytorch通过一种操作增加计算量减少内存占用
           nn.MaxPool2d(kernel_size=3, stride=2), # 卷积核大小是3, 步距是2
                                                                                   #
output[48, 27, 27]
           nn.Conv2d(48, 128, kernel size=5, padding=2), # output[128, 27, 27]
           nn.ReLU(inplace=True),
           nn.MaxPool2d(kernel_size=3, stride=2), # output[128, 13, 13]
           nn.Conv2d(128, 192, kernel_size=3, padding=1), # output[192, 13, 13]
           nn.ReLU(inplace=True),
           nn.Conv2d(192, 192, kernel_size=3, padding=1), # output[192, 13, 13]
           nn.ReLU(inplace=True),
           nn.Conv2d(192, 128, kernel_size=3, padding=1), # output[128, 13, 13]
           nn.ReLU(inplace=True),
           nn.MaxPool2d(kernel size=3, stride=2), # output[128, 6, 6]
       self.classifier = nn.Sequential( # 分类器
           nn.Dropout(p=0.5), # dropout的方法上全连接层随机失活(一般放在全裂阶层之间)p值随即失火的比例
           nn.Linear(128 * 6 * 6, 2048), # linear是全连接层
                                                                              A *, U = *
           nn.ReLU(inplace=True), # 激活函数
           nn.Dropout(p=0.5),
           nn.Linear(2048, 2048),
```





从图中看出训练数据集的Loss在训练**1500次之后开始趋于稳定**,曲线在**0至500区间下降幅度较大,最终收敛于0.15左右**。在验证数据集中的准确率Accuracy在10轮训练后都比前一次高,可以看到曲线总体呈现上升趋势,最终accuracy最高达到95.89%。



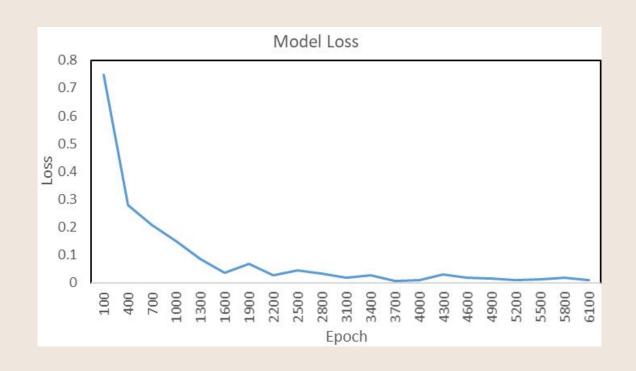
从图中看出该模型在番茄病害数据集群的预测表现不佳,分析与番茄病害数据种类最多有关,番茄数据集种类约占总种类数量的一半,为10种。 玉米病害集群识别准确率均在95%以上。该模型最终**获得了94.25%的平均** 识别准确率。

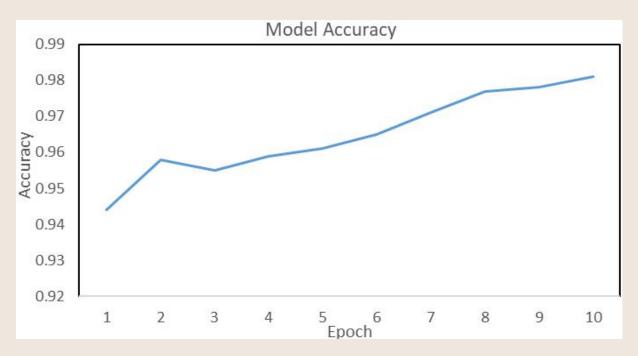
output size	18-layer	34-layer	50-layer	101-layer	152-layer	
112×112	7×7, 64, stride 2					
	3×3 max pool, stride 2					
56×56	$\left[\begin{array}{c} 3 \times 3, 64 \\ 3 \times 3, 64 \end{array}\right] \times 2$	$\left[\begin{array}{c} 3\times3, 64\\ 3\times3, 64 \end{array}\right]\times3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 3$	
28×28	$\left[\begin{array}{c}3\times3,128\\3\times3,128\end{array}\right]\times2$	$\left[\begin{array}{c} 3 \times 3, 128 \\ 3 \times 3, 128 \end{array}\right] \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 4$	$ \left[\begin{array}{c} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{array}\right] \times 8 $	
14×14	$\left[\begin{array}{c}3\times3,256\\3\times3,256\end{array}\right]\times2$	$\left[\begin{array}{c} 3 \times 3, 256 \\ 3 \times 3, 256 \end{array}\right] \times 6$	$ \begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 6 $	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 23$	$ \left[\begin{array}{c} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{array}\right] \times 36 $	
7×7	$\left[\begin{array}{c}3\times3,512\\3\times3,512\end{array}\right]\times2$	$\left[\begin{array}{c} 3\times3,512\\ 3\times3,512 \end{array}\right]\times3$	$ \begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3 $	$ \begin{bmatrix} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{bmatrix} \times 3 $	$ \left[\begin{array}{c} 1 \times 1, 512 \\ 3 \times 3, 512 \\ 1 \times 1, 2048 \end{array}\right] \times 3 $	
	112×112 56×56 28×28	$ \begin{array}{c c} 112 \times 112 \\ \hline 56 \times 56 \\ \hline \begin{bmatrix} 3 \times 3, 64 \\ 3 \times 3, 64 \end{bmatrix} \times 2 \\ 28 \times 28 \\ \hline \begin{bmatrix} 3 \times 3, 128 \\ 3 \times 3, 128 \end{bmatrix} \times 2 \\ \hline 14 \times 14 \\ \hline \begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 2 \end{array} $	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	$ \begin{array}{c ccccccccccccccccccccccccccccccccccc$	

残差神经网络(ResNet)可以用来解决网络的深度问题残差神经网络模型有18个卷积模型,有2个残差块,有1个最大池化层,有1个全连接层,有1个Dropout层以及一个分类层。

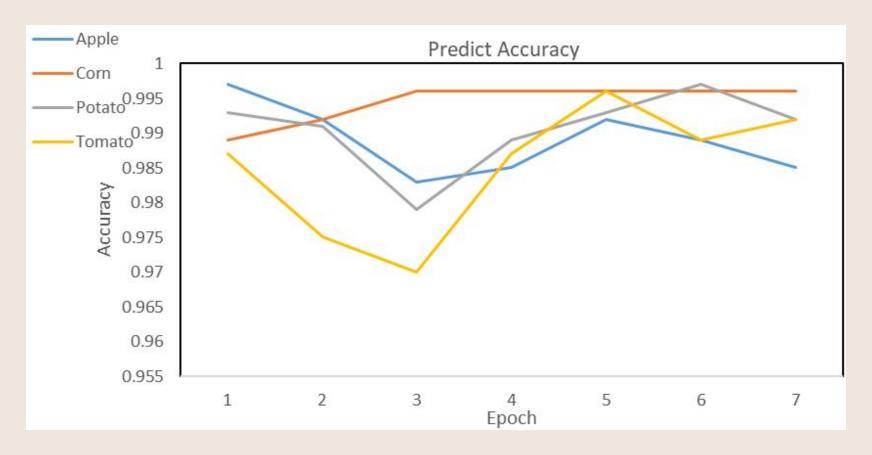
```
def Resnet34(num_classes=1000, include_top=True):
    # https://download.pytorch.org/models/resnet34-333f7ec4.pth
    return ResNet(BasicBlock, [3, 4, 6, 3], num_classes=num_classes, include_top=include_top)
```

```
# 下载预处理权重
model weight path = "./resnet34-pre.pth"
#载入模型权重
resnet.load_state_dict(torch.load(model_weight_path, map_location='cpu'))
# for param in net.parameters():
     param.requires grad = False
# 输入特征矩阵深度
in_channel = resnet.fc.in_features
resnet.fc = nn.Linear(in_channel, 21)
resnet.to(device)
```





从图中看出训练数据集的Loss在训练1500次之后开始趋于稳定,曲线在0至400区间下降幅度最大,最终收敛于0.05左右。在验证数据集中的准确率Accuracy在10轮训练过程中,每次训练后都比前一次高,可以看到曲线总体呈现上升趋势,最终accuracy最高达到98.12%。



从图中看出基于迁移学习的残差卷积神经网络模型在各个数据集预测实验中**准确率表现均为90%以上**。同样在番茄病害数据集群的预测表现稍差,分析与番茄病害数据种类最多有关,番茄数据集种类约占总种类数量的一半,为10种。最终获得了**98.04%的平均识别准确率**。



●总结展望

本文使用构建的基于迁移学习的残差神经网络模型对玉米、苹果、番茄、土豆四个数据集群共21种常见病虫害识别取得了**较高的识别准确率,说明该模型性能较高,**但若要再进一步提高并运用于实际生产中,还需在一下两个方面改进:

- (1) **可诊断的病虫害种类需要增加**:目前可识别苹果病虫害种类4种,玉米病虫害种类4种,土豆病虫害种类3种,番茄病虫害种类10种。未来还需在实地获取更多果作物病虫害种类,使本文提出的模型能对更多种类数据集进行训练,识别更多种类果作物病害。
- (2)对本文提出的分类识别网络模型加深网络模型层数:本文使用的是34层深度的ResNet网络模型。在未来的研究中,可以加深网络模型层数,深化网络结构,图像特征提取更为丰富,预测识别结果也能更加准确可信。



请各位老师批评指正

South China Agricultural University