基于深度学习的玉米异常果穗智能筛分方法

梁国际　刘兰华

南宁学院人工智能学院，南宁，530200

摘　要　玉米作为重要粮食作物，种子质量是影响玉米产量的关键因素，提高玉米制种筛分的速率和效率，具有重要意义。针对玉米制种穗选过程中异常果穗的表型识别问题，以玉米果穗图像数据为研究对象，基于深度学习图像处理技术，构建多种经典的卷积神经网络模型，调用 PaddlePaddle 的图像增强 API 进行数据增强处理，实现对玉米果穗的高通量智能筛分。实验结果表明，该方法针对数据集相对较小的情况，可以有效提升玉米异常果穗识别的准确率。

关键字　深度学习, 卷积神经网络, 玉米异常果穗, 智能筛分

The Intelligent Screening Method for Abnormal Ear of Maize based on Deep Learning

Liang Guoji　Liu Lanhua

School of Artificial Intelligence
Nanjing University
Nanning 530200, China;
gjliang@iflytek.com

Abstract—As an important grain crop, maize seed quality is the key factor affecting maize yield. It is of great significance to improve the rate and efficiency of maize seed screening. In order to solve the problem of phenotypic identification of abnormal ear in the process of corn seed production and ear selection, a variety of classical convolutional neural network models were constructed based on deep learning image processing technology. The image enhancement API of PaddlePaddle was called for data enhancement processing to realize high-throughput intelligent screening of corn ears. The experimental results show that this method can effectively improve the accuracy of corn abnormal ear recognition when the data set is relatively small.

Key words—Deep learning, Convolutional neural network, Corn abnormal ear, Intelligent screening

1　引　言

玉米是全国重要的粮食作物和重要的饲料来源，但玉米种植还受到许多因素的影响，导致玉米的形状层次不齐，其产量、质量也往往达不到目标。选用优质玉米种子即玉米制种穗选是重要环节，筛分出异常果穗（机械损伤、虫蛀、败育、病害等），有利于提高玉米种子的纯度和发芽率。

随着深度学习视觉技术的高速发展，弥补了传统检测方法特征构建能力差、特征学习过程需人为干预等问题，促进了图像识别、分类等技术在各个领域的应用。其中，深度学习中的卷积神经网络可以从大量的玉米图像数据中自动学习出异常果穗背后的特征，能够降低在识别领域上的操作门槛，并且可以适应玉米图像拍摄不规范所产生的目标平移、目标缩放、目标形变以及目标所处环境的变化，凭借其识别特征能力强、特征学习的过程中无需过多的人为干预、检测速度快和精度高等优势，迅速成为图像领域的主流方法，提高对玉米病害的识别准确率。

基于深度学习的玉米异常果穗智能筛分关键技术，主要包括数据集、图像数据预处理、卷积神经网络模型构建、模型训练、模型部署。下面介绍本文对这些关键技术的设计考虑。

2　图像数据集

2．1　异常果穗的定义

机械损伤果穗：大型机械设备收获果穗较为新鲜，新鲜果穗含水量较大（30%左右），易造成果穗表皮损伤，露出白色胚乳；受伤胚乳易霉变，对玉米发芽率影响较大，影响产量。

虫蛀果穗：受种植环境影响果穗发生病虫害，或存储过程中害虫对穗部叮咬，使得玉米果穗失去制种价值。

霉变果穗：玉米果穗在田间或存储中，由于环境水分含量过高引起穗部发生霉变，籽粒呈现出各种霉斑颜色，失去制种价值。

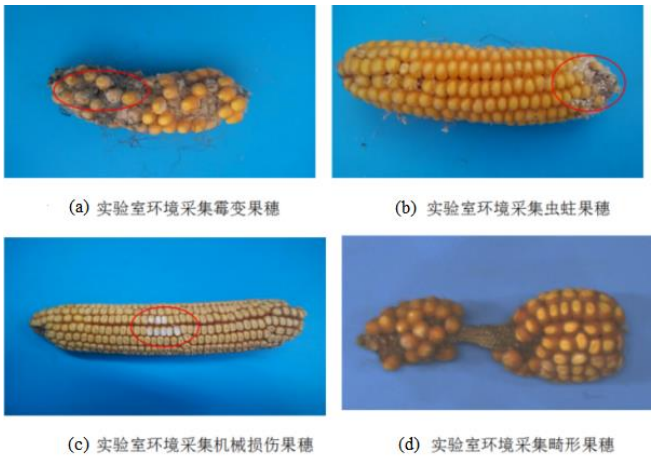
畸形果穗：玉米果穗两头大且有籽粒，中间部分小，没有结实籽粒；或果穗只有单侧面有籽粒，或只有很少籽粒散落分布，或果穗顶部秃尖过大，几乎可占果穗 1/2 以上。

异种果穗：与所选的目标品种不同的果穗，这部分果穗若混入目标品种果穗中会影响后期制种的纯度。

败育果穗：由于品种缺陷等原因导致玉米果穗顶部未发育或发育不全，表现为上部无籽粒或籽粒干瘪，严重秃尖可占果穗的 1/2 以上。

2．2　数据集合

本文所使用到的数据集来自百度飞桨网站（https://aistudio.baidu.com/）公开的数据集，图像数据集包含正常、霉变、虫蛀、机械损伤和带苞叶等几个类别果穗。如图 1 所示。



(a) 实验室环境采集霉变果穗　(b) 实验室环境采集虫蛀果穗

(c) 实验室环境采集机械损伤果穗　(d) 实验室环境采集畸形果穗

图 1　异常玉米果穗样例

3　图像数据预处理

3．1　图像裁剪

本文研究所使用的数据集图像，其主要区域集中于中间部分，裁剪其四边无用的特征，可以进一步优化模型的识别精度。为此，我们在已知图像的基础上，按照 224×224 的图像大小缩小图像大小，在精简数据的同时，保持主要信息的不变。

3．2　数据集划分

本文实验以7：2：1比例将数据集随机划分为训练集、测试集和验证集。

训练集是用来训练模型的。为了减少泛化误差，需要通过训练集不断的训练来使得模型能够更好的接近真实数据。

测试集是用来测试模型的准确性，为此，将测试集应用于训练好的模型，会得到一个模型的得分，例如混淆矩阵、recall、f1等得分。

在数据建模中，不仅需要作模型与模型之间类的比较，对于某一类模型内部，也要不断进行筛选，涉及到模型自身的评估以及超参数的调整，查看模型训练的效果是否朝着坏的方向进行，就需要从训练集中再次划分出验证集。

3．3　数据增强

针对制种果穗类别样本数据不均衡的问题，也可采用镜像、旋转任意角度、改变亮度等技术实现样本数据的扩增，如图2所示。

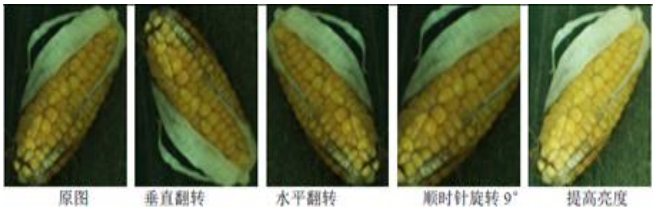


图 2 图片数据增强

3．4　数据归一化

对图像归一化可以帮助网络更快速的收敛其训练过程，防止因数据倾斜造成过拟合或者欠拟合。

数据预处理关键逻辑代码：

```
class Dataset(Dataset):
    """
    步骤一：继承io.Dataset类
    """
    def __init__(self, mode='train'):
    """
```

步骤二：实现构造函数，定义数据读取方式，划分训练和测试数据集

```
    """
    super(Dataset, self).__init__()
    train_image_dir = train_parameters['train_image_ dir']
    eval_image_dir= train_parameters['eval_image_dir']
    test_image_dir = train_parameters['test_image_dir']
    transform_train = Compose([Resize(size=(224, 224)),
Transpose(), Normalize()])
    transform_eval = Compose([Resize(size=(224, 224)),
Transpose(), Normalize()])
    train_data_folder = DatasetFolder(train_image_dir,
transform=transform_train)
    eval_data_folder = DatasetFolder(eval_image_dir,
transform=transform_eval)
    test_data_folder = DatasetFolder(test_image_dir,
transform=transform_eval)
    self.mode = mode
    if self.mode == 'train':
        self.data = train_data_folder
```

```
elif self.mode == 'eval':
    self.data = eval_data_folder
elif self.mode == 'test':
    self.data = test_data_folder
def __getitem__(self, index):
"""
```

步骤三：实现__getitem__方法，定义指定index时如何获取数据，并返回单条数据（训练数据，对应的标签）

```
  """
data = np.array(self.data[index][0]).astype('float32')
label = np.array([self.data[index][1]]).astype('int64')
return data, label
def __len__(self):
"""
```

步骤四：实现__len__方法，返回数据集总数目

```
"""
return len(self.data)
train_dataset = Dataset(mode='train')
val_dataset = Dataset(mode='eval')
test_dataset = Dataset(mode='test')
```

4 卷积神经网络

4．1 卷积神经网络结构

卷积神经网络主要包括：输入层、卷积层、池化层、线性整流层以及全连接层。

(1) 输入层主要是 n×m×3 RGB 图像，这不同于人工神经网络，人工神经网络的输入是 n×1 维的矢量。

(2) 卷积层，计算输入图像的区域和滤波器的权重矩阵之间的点积，并将其结果作为该层的输出。滤波器将滑过整个图像，重复相同的点积运算。这里注意两件事：滤波器必须具有与输入图像相同数量的通道；网络越深，使用的滤波器就越多；拥有的滤波器越多，获得的边缘和特征检测就越多。

(3) 池化层：池化即下采样（downsamples），目的是为了减少特征图大小，通过减少网络的参数来减小计算量，并且能够在一定程度上控制过拟合。池化层进行的运算一般有两种： 最大池化和平均池化。

(4) 线性整流层，一般使用 ReLU 激活函数，把卷积层输出结果做非线性映射，它的特点是收敛快，求梯度简单。

(5) 在全连接层中，我们将最后一个卷积层的输出展平，并将当前层的每个节点与下一层的另一个节点连接起来。全连接层和卷积层可以相互转换。

本文一共使用 5 种卷积网络模型：AlexNet、VGGNet、ResNet、MobileNet V1、MobileNeV2。

4．2 AlexNet 网络

使用 ReLU 作为 CNN 的激活函数，验证了其效果在较深的网络中超过了 Sigmoid，成功解决了 Sigmoid 在网络较深时的梯度弥散问题。

训练时使用 Dropout 随机忽略一部分神经元，以避免模型过拟合，一般在全连接层使用，在预测的时候是不使用 Dropout。

在 CNN 中使用重叠的最大池化(步长小于卷积核)。此前 CNN 中普遍使用平均池化，使用最大池化可以避免平均池化的模糊效果。

同时重叠效果可以提升特征的丰富性。 提出 LRN（Local Response Normalization，即局部响应归一化）层，对局部神经元的活动创建竞争机制，使得其中响应比较大的值变得相对更大，并抑制其他反馈较小的神经元，增强了模型的泛化能力。

4．3 VGGNet 网络

对比 AlexNet，加深网络层次，达到了 19 层的网络结构，且 VGG 网络的结构非常一致，从头到尾全部使用的是 3x3 的卷积和 2x2 的 max pooling，从而能够拥有更好的模型泛化能力，最终实现更小的分类错误率。

4．4 ResNet 残差网络

随着网络层数的增加，网络发生了退化（degradation）的现象：随着网络层数的增多，训练集 loss 逐渐下降，然后趋于饱和，当再增加网络深度的话，训练集 loss 反而会增大。ResNet 残差网络主要是通过残差块组成的，解决深度网络中的退化问题。与普通网络的串行结构相比，残差单元增加了跳跃映射，将输入与输出直接进行相加，补充卷积过程中损失的特征信息。

4．5 MobileNetV1 网络

MobileNet 为轻量化网络模型，若使用手机作为玉米异常果穗识别的终端，更容易用于实际应用推广。

标准卷积（CNN）的作用是对所有输入通道用相同的卷积核得到不同通道特征，将不同通道特征组合起来使得输出特征包含每个输入的特征。可以将标准卷积分为以下两个步骤： 深度卷积+逐点卷积，即为深度可分离卷积，大大减少了参数数量和计算量。

（1）深度卷积（Depthwise convolution）

对每一个输入通道应用一个单独的卷积核（与标准卷积不同，标准卷积对每一个输入通道应用同一个卷积核）得到特征图，此时，每张特征图仅与一个输入通道和其对应卷积核相关，各个通道之间特征图不关联。

（2）逐点卷积（Pointwise concolution）

逐点卷积（1x1 卷积）将深度卷积的输出特征图进行线性组合后再输出，使得最后的输出特征都包含每个输入特征，即将深度卷积输出的不关联的特征图关联起来。

4．6 MobileNetV2 网络

针对 MobileNetV1 进行改进，引入残差结构，先升维再降维，增强梯度的传播，显著减少推理期间所需的内存占用。去掉 Narrow layer（low dimension or depth） 后的 ReLU，保留特征多样性，增强网络的表达能力。网络为全卷积的，使得模型可以适应不同尺寸的图像；使用 RELU6（最高输出为 6）激活函数，使得模型在低精度计算下具有更强的鲁棒性。

4．7 PaddlePaddle 实现网络模型

关键实现代码如下：

```
class AlexNet(paddle.nn.Layer):
  def __init__(self, class_dim):
    super(AlexNet, self).__init__()
    self.conv1 = paddle.nn.Conv2D(3, 96, 11, stride=4, padding=2)
    self.pool1 = paddle.nn.MaxPool2D(3, stride=2, padding=0)
    self.conv2 = paddle.nn.Conv2D(96, 256, 5, stride=1, padding=2)
    self.pool2 = paddle.nn.MaxPool2D(3, stride=2, padding=0)
    self.conv3 = paddle.nn.Conv2D(256, 384, 3, stride=1, padding=1)
    self.conv4 = paddle.nn.Conv2D(384, 384, 3, stride=1, padding=1)
    self.conv5 = paddle.nn.Conv2D(384, 256, 3, stride=1, padding=1)
    self.pool3 = paddle.nn.MaxPool2D(3, stride=2, padding=0)
    self.fc1 = paddle.nn.Linear(9216, 4096)
    self.fc2 = paddle.nn.Linear(4096, 4096)
    self.fc3 = paddle.nn.Linear(4096, class_dim)
    self.relu = paddle.nn.ReLU()
    self.dropout = paddle.nn.Dropout(p=0.5)
    self.softmax = paddle.nn.Softmax()
    def forward(self, x):
      x = self.conv1(x)
      x = self.relu(x)
      x = self.pool1(x)
      x = self.conv2(x)
      x = self.relu(x)
      x = self.pool2(x)
      x = self.conv3(x)
      x = self.relu(x)
      x = self.conv4(x)
      x = self.relu(x)
      x = self.conv5(x)
      x = self.relu(x)
      x = self.pool3(x)
      x = paddle.flatten(x, start_axis=1, stop_axis=3)
      x = self.fc1(x)
      x = self.relu(x)
      x = self.fc2(x)
      x = self.relu(x)
      x = self.fc3(x)
      x = self.softmax(x)
    return x

  model_alexnet = AlexNet(class_dim=train_parameters["class_dim"])
  # 定义 ResNet 网络
  model_resnet = paddle.vision.models.resnet50(pretrained=True,
  num_classes=train_parameters["class_dim"])
  # 定义 VGG 网络
  model_vgg = paddle.vision.models.vgg16(pretrained=True,
  batch_norm=False,
num_classes=train_parameters["class_dim"])
  # 定义 MobileNet v1 网络
  model_mobilenet_v1 = paddle.vision.models.mobilenet_v1(pretrained=True,
  num_classes=train_parameters["class_dim"])
  # 定义 MobileNet v2 网络
  model_mobilenet_v2= paddle.vision.models.mobilenet_v2(pretrained=True,
  num_classes=train_parameters["class_dim"])
  # 切换使用的网络
  model = paddle.Model(model_mobilenet_v1)
  test_dataset = Dataset(mode='test')
```

5 模型训练

5．1 训练方式一

模型中的fit方法，指定训练集、训练迭代次数、数据批量等参数：

```
model.fit(train_dataset,
          epochs=5,
          batch_size=64,
        verbose=1)
model.evaluate(test_dataset, verbose=1)
```

5．2 训练方式二

通过paddle.io.DataLoader构建训练集数据加载器、测试集数据加载器：

```
train_loader = paddle.io.DataLoader(train_dataset, batch_size=64, shuffle=True)
test_loader = paddle.io.DataLoader(test_dataset, batch_size=64, shuffle=True)
model.fit(train_loader,
      eval_data = test_loader,
      epochs=5,
      verbose=1,
  )
model.evaluate(test_dataset, verbose=1)
```

5．3 训练方式三

模型微调，给定预训练模型（Pre_trained model），基于模型进行微调（Fine Tune）。相对于从头开始训练(Training a model from scatch)，微调省去大量计算资源和计算时间，提高了计算效率,甚至提高准确率。

微调是指将其顶部的几层“解冻”，并将这解冻的几层和新增加的部分联合训练 冻结（freeze）一个或多个层是指在训练过程中保持其权重不变网络中低

层部分被冻结，仅将顶层解冻，再根据需要设计全连接层。

表 1 MobileNet具体训练参数设置

参数	优化器	Batch-size	损失函数	学习率
大小	Adam	8	CrossEntropyLoss	0.001

在实验的过程中，网络在训练集和测试集上均循环了50次，具体反馈的识别率见图3所示。

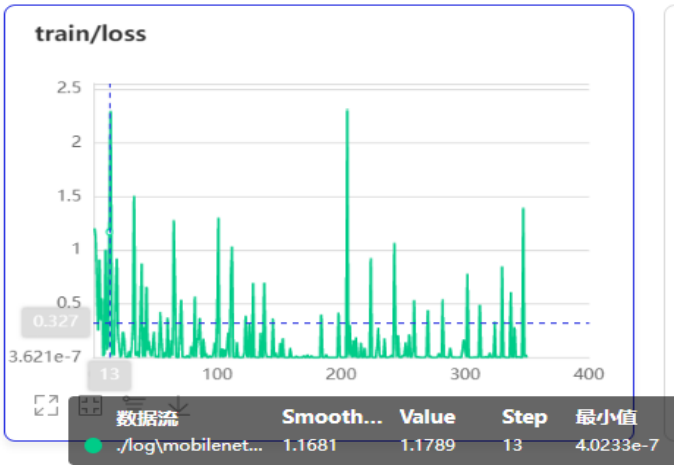


图 3 MobileNet模型训练损失率

在4.7节中，vgg、resnet、mobilenet等模型就是采用模型微调的方式进行训练。

除此之外，还需进行超参数的设置，包括Epoch、Batchsize、Iterations、optimizer、loss、metrics等参数。设置方式如下：

```
model.prepare(optimizer=paddle.optimizer.Adam(learning_rate=0.001,
 parameters=model.parameters()),
 loss=paddle.nn.CrossEntropyLoss(),
metrics=paddle.metric.Accuracy())
```

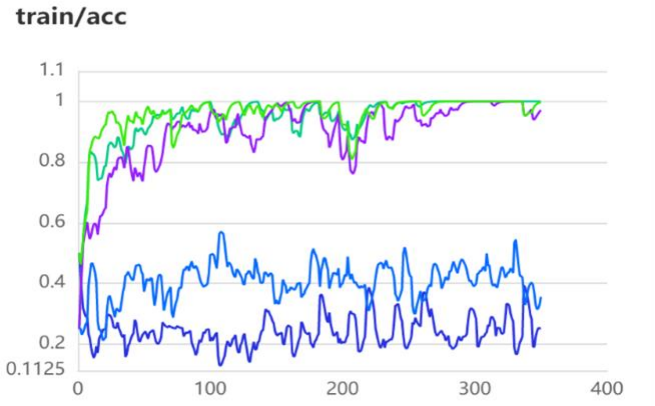


图 4 模型训练集准确率

5．4 实验的结果

MobileNet网络结构的实验参数如表1所示。

各模型在相同数据集下具有不同的性能，如图4和图5所示。从实验结果来看，经过50轮迭代训练后，ResNet与MobileNet等模型在玉米果穗分类任务中取得的效果优于AlexNet和VGG等模型的效果。

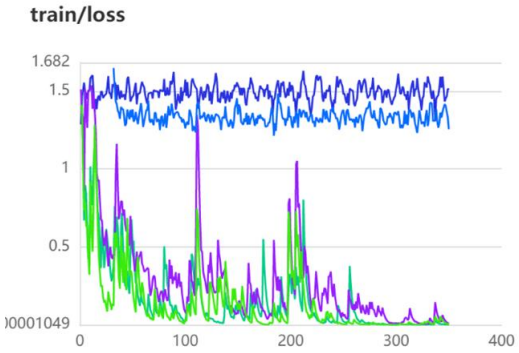


图 5 模型训练集损失率

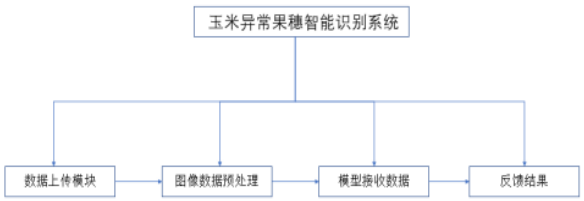


图 6 识别流程图

6 模型训练

一个简洁、易用的操作界面，可以方便操作人员直接操作，即便是没有相关知识的从业人员，也无需了解过多的原理，即可点击上传图片进行识别，并且能够快速的查看到反馈的结果。基于需求设计。主要由四个模块组成，分别为：数据的上传、数据预处理、模型接受数据、结果输出。具体流程方案如图6所示，操作界面如图7所示。

图 7 系统界面示意图

7 结束语

本文针对玉米异常果穗智能筛分这一农业领域实际生产问题，分别构造了AlexNet、VGG、ResNet、MobileNet等多种卷积神经网络进行图像分类识别。针对数据集相对较小的情况，选择较小的Batchsize，并调用PaddlePaddle的图像增强API进行数据增强处理，可以有效提升玉米异常果穗识别的准确率。

目前，系统的开发还存在许多问题还未被发现。今后将会根据应用需求对系统进行升级，以进一步提升模型准确率，提高提升用户使用率。

参 考 文 献

[1] 陈川,陈柘,丁双惠.深度学习发展形势下计算机视觉教学内容革新[J].计算机与现代化,2020,(06):107-113

[2] Dyrmanna M.,Karstoftb H., Midtibya H. S.. Plant species classification using deep convolutional neural network[J]. Biosystems Engineering,2016,151:72-80.

[3] Fuentes A., Yoon S., Kim S.,C., et al. A Robust Deep-Learning-Based Detector for Real-Time Tomato Plant Diseases and Pests Recognition. Sensors 2017, 17: 2022.

[4] Mohanty S P, Hughes D P, Salathé M. Using deep learning for image-based plant disease detection[J]. Frontiers in plant science, 2016, 7: 1419.

[5] Türkoğlu M, Hanbay D. Plant disease and pest detection using deep learning-based features[J]. Turkish Journal of Electrical Engineering & Computer Sciences, 2019, 27(3): 1636-1651.

[6] 李鑫然. 基于深度学习的苹果叶片病害检测方法研究[D]. 榆林：西北农林科技大学,2021.

[7] 李昊. 基于深度学习的柑橘病害智能在线监测系统研究[D].成都：电子科技大学,2021.

[8] 秦嘉奇.基于 Mobilenet 的农作物叶片病害识别方法[J].信息与电脑(理论版),2021,33(18):181-184.

[9] 孙红,乔金博,李松,李民赞,宋迪,乔浪.基于深度学习的玉米拔节期冠层识别[J].农业工程学报,2021,37(21):53-61.

[10] 邓国强,王君婵,杨俊,刘涛,李冬双,孙成明.基于图像和改进U-net 模型的小麦赤霉病穗识别[J].麦类作物学报,2021,41(11):1432-1440.

[11] 宋成龙. 基于深度学习的番茄病害目标检测算法的研究与应用[D].保定：河北科技大学,2020.