

本科毕业论文

题 目：基于深度学习的农业病虫

害检测算法

院 （部）： 计算机科学与技术学院

专 业： 计算机科学与技术

班 级： 计科143

姓 名： 朱忠栋

学 号： 20141111110

指导教师： 李盛恩

完成日期： 2018年5月30日

目 录

[摘 要 IV](#_Toc515698411)

[ABSTRACT V](#_Toc515698412)

[1绪 论](#_Toc515698413)

[1.1 研究背景 - 1 -](#_Toc515698414)

[1.1.1 病虫害防治中的问题 - 1 -](#_Toc515698415)

[1.1.2 深度学习的复兴 - 1 -](#_Toc515698416)

[1.2研究意义 - 1 -](#_Toc515698417)

[1.3研究现状 - 2 -](#_Toc515698418)

[1.3.1数据集现状 - 2 -](#_Toc515698419)

[1.3.2目标检测算法 - 2 -](#_Toc515698420)

[1.4研究内容 - 2 -](#_Toc515698421)

[1.5论文的组织结构 - 3 -](#_Toc515698422)

[2神经网络与深度学习](#_Toc515698423)

[2.1神经网络 - 4 -](#_Toc515698424)

[2.1.1感知器 - 4 -](#_Toc515698425)

[2.1.2分类器 - 4 -](#_Toc515698426)

[2.1.3激活函数 - 5 -](#_Toc515698427)

[2.1.4损失函数 - 7 -](#_Toc515698428)

[2.1.5梯度下降 - 8 -](#_Toc515698429)

[2.1.6反向传播 - 9 -](#_Toc515698430)

[2.1.7正则化 - 10 -](#_Toc515698431)

[2.2卷积神经网络 - 11 -](#_Toc515698432)

[2.2.1卷积层 - 11 -](#_Toc515698433)

[2.2.2池化层 - 12 -](#_Toc515698434)

[2.2.3全连接层 - 13 -](#_Toc515698435)

[2.3 基于卷积神经网络的目标检测算法 - 13 -](#_Toc515698436)

[2.3.1传统目标检测算法 - 14 -](#_Toc515698437)

[2.3.2 RCNN - 14 -](#_Toc515698438)

[3算法分析与设计- 16 -](#_Toc515698439)

[3.1 问题描述 - 16 -](#_Toc515698440)

[3.1.1 问题定义 - 16 -](#_Toc515698441)

[3.1.2采用模型 - 16 -](#_Toc515698442)

[3.1.3评测标准 - 16 -](#_Toc515698443)

[3.2实验数据集 - 16 -](#_Toc515698444)

[3.2.1数据集采集 - 16 -](#_Toc515698445)

[3.2.2数据集划分 - 17 -](#_Toc515698446)

[3.3算法实现 - 18 -](#_Toc515698447)

[3.3.1特征抽取网络 - 18 -](#_Toc515698448)

[3.3.2激活函数选择 - 20 -](#_Toc515698449)

[3.3.3参数更新 - 20 -](#_Toc515698450)

[3.3.4随机丢弃层 - 22 -](#_Toc515698451)

[3.3.5零填充 - 23 -](#_Toc515698452)

[3.3.6数据增强技术 - 23 -](#_Toc515698453)

[3.3.7参数初始设置 - 24 -](#_Toc515698454)

[3.3.8数据集预处理 - 24 -](#_Toc515698455)

[3.3.9候选区域缩放方式 - 25 -](#_Toc515698456)

[4实验测试与评价](#_Toc515698457)

[4.1实验环境 - 27 -](#_Toc515698458)

[4.2训练参数 - 27 -](#_Toc515698459)

[4.3实验结果 - 28 -](#_Toc515698460)

[4.4实验结果评价 - 31 -](#_Toc515698461)

[4.3.1损失函数和准确率 - 31 -](#_Toc515698462)

[4.3.3卷积核展示 - 31 -](#_Toc515698463)

[5实验总结与展望 - 33 -](#_Toc515698464)

[5.1实验总结 - 33 -](#_Toc515698465)

[5.2展望 - 33 -](#_Toc515698466)

[谢 辞 - 35 -](#_Toc515698467)

[参考文献 - 36 -](#_Toc515698468)

摘 要

在农作物病例检测中，许多检测技术被应用于此领域。但由于传统机器学习方法检测准确率太低，仍然存在农作物病虫害防治不及时的问题，严重影响农业生产的质量和产量。近些年深度学习的快速发展，基于卷积神经网络的目标检测算法，在检测效率和准确率上有很大的提升。

本论文研究了基于深度学习的目标检测算法（RCNN）算法在农业病虫害识别上的应用。首先总结我国在农业病虫害检测方面的现状，然后介绍了深度神经网络和RCNN算法的原理。按照RCNN的思想，分析柑桔疮痂病的特征，设计检测网络的结构。并且讨论了检测网络训练过程中出现的问题。结合卷积神经网络的最新优化技术，对算法进行改进。测试表明在该农作物病例数据集上取得了突出的成果。最后对实验结果进行分析，总结算法存在的问题和以后努力的方向。

关键词：深度学习；农业病虫害；目标检测；卷积神经网络

**Detection algorithm of agricultural diseases and insect pests based on deep learning**

**ABSTRACT**

At present, many detection technologies are applied to identify agricultural diseases and insect pests. However, the accuracy rate of traditional machine learning methods is too low, there are still problems of treating crop disease not in time, which have a significant effect on yield and quality of crops. In recent years, with the rapid development of deep learning, the objection detection algorithm based on convolution neural network has greatly improved the detection efficiency and accuracy.

This paper studies the application of objection detection algorithm (RCNN) based on deep learning in agricultural pest and disease recognition. First, the previous research work in the field of identifying agricultural diseases was summarized. Then the principles of deep neural network and RCNN algorithm are introduced. Analyzing the characteristics of citrus scab and according to the idea of RCNN, the structure of detection network is designed. It also discussed the problems in detection network training. Combined with the latest optimization technology of training neural network, this paper made a lot of improvement for the algorithm. These methods have proved to be useful and achieve good scores on data set of the crop. Finally, the results of experiments is evaluated. Further study is needed in this field hoping that more progress can be made.

**Key Words:** Deep learning；Agricultural diseases and insect pests；Object detection；Convolutional Neural Networks

1绪 论

1.1 研究背景

1.1.1 病虫害防治中的问题

中国作为传统的农业大国，农业在我国经济中占有举足轻重的作用，同时也是经济发展的基础。目前这个基础产业却存在很多问题，例如农产品质量不高，难以保障现代农业的发展等。在农业种植中，病虫害一直是阻碍农作物生长的重要问题，直接导致了农产品的质量和产量的下降。据统计，我国农作物病虫害呈多发重发态势，每年发生面积近70亿亩次，因防控能力不足每年造成粮食损失近500亿斤、经济作物损失350多亿斤。由于我国农业的若干特点，如农民文化素质差，农业领域的专家和科技人员紧缺，和农民普遍利用自己长时间的传统种植经验来防治病虫害，这种基于经验的检测方法具有很大的局限性，造成了滥用农药，防治不准确，不及时的问题，尤其在面对各种新型病虫害时束手无策，直接造成重大的农业损失。过渡使用农药还会使病虫害产生抗药性，进一步加剧了农业病虫害的防治难度[[[[1]](#endnote-1)]]。

1.1.2 深度学习的复兴

卷积神经网络在上世纪80年代就已经出现，起初用于识别邮政系统中的手写数字，但是碍于神经网络中的几处计算细节和计算机硬件水平，导致神经网络的计算速度非常慢，因而发展进入瓶颈。2012年图像识别大赛的冠军模型AlexNet[[[2]](#endnote-2)]，作者仅仅做了几处细节的修改，如使用更为线性的ReLU函数作为激活函数等，并且使用GPU加速运算，它出色的表现又引起了深度学习和计算机视觉的复兴，同时证明了深度学习的巨大潜力。之后，关于深度学习的研究开始爆发性的增长，神经网络的深度不断增加。如今，深度学习已经在图像、语音、自然语言处理、CTR预估、大数据特征提取等方面获得广泛的应用。面向图像分类的深度学习研究蓬勃发展，已经在该领域取得了很多重大突破，甚至超越了人类的表现。应用于许多商业化的图像分类问题中，甚至成为了很多新兴科技的基础。在医疗、金融、无人驾驶行业表现出革命性的成果。深度学习作为一种应用技术，在各行业中的应用已经成为目前研究的主要方向。

1.2研究意义

信息技术正在改变人们的生活，农业想要进一步发展，必然要与信息技术相结合。为了解决我国农业病虫害防止困难的现状，设计一个用于检测农业病虫害的算法可以很好的解决病虫害防治困难的问题，提高病虫害诊断的准确度和及时性。基于深度学习的RCNN算法在目标检测上有巨大的优势，农作物病虫害的识别可以视为目标检测的任务，RCNN算法在农业病虫害识别上的应用有很大的潜力。

1.3研究现状

1.3.1数据集现状

深度学习在农业领域应用的研究者非常少，没有发现公开的病虫害数据集。病虫害往往具有明显的区域性，所以建立一种病例的数据集，可能只适用于一个区域，也导致病虫害数据集的建立的不完善。然而神经网络需要大量的数据支持，完善的数据集是算法准确度的基础，这也是现阶段检测算法准确度不高的主要原因。

1.3.2目标检测算法

传统的目标检测方法往往需要手动设计特征。对于病虫害图像分析而言，需要对农业病例的症状非常熟悉，来定义病斑的形状、大小、颜色特征，这也是制约计算机技术在农业领域应用的关进因素。研究者们经常使用梯度直方图（HoG）[[[3]](#endnote-3)]、颜色直方图。词袋、局部二值特征、SIFT特征[[[4]](#endnote-4)]、haar特征[[[5]](#endnote-5)]等方法来抽取图像的特征，然后将这些特征作为支持向量机（SVM）[[[6]](#endnote-6)]的输入训练分类器，用这个模型去预测患病的概率。但传统检测方法存在很多问题，如特征的定义往往有很强的主观性，误差很大，且检测效果的准确率不高，鲁棒性很差。这是传统检测算法没有走向实践的根本原因。

由于信息技术在农业领域应用探索比较缓慢，利用深度学习技术检测农业病虫害研究还没有论文发表。 深度学习完全以数据作为驱动，可以在数据中学到不同层次的信息，挖掘出超越人工设计的特征，可以更强的表达出数据所包含的内容，消除了主观性的影响。

1.4研究内容

本文对基于卷积神经网络的RCNN目标检测算法在农业病虫害识别问题上的作用进行了探究。针对柑桔疮痂病，收集病例图片，以及对数据集预处理，建立该病例的数据集。探究RCNN算法的原理，按照具体应用场景，经过大量实验，设计网络结构。使用数据集训练网络，评价测试结果，记录准确度变化的过程，不断调节超参数，对网络进行调优。且在训练过程中，根据所发现的问题，借鉴最新神经网络训练优化技术，改造网络结构，进一步提升模型的准确度。

1.5论文的组织结构

本论文共有5章，每一章的内容安排如下。

第一章是绪论，介绍了利用深度学习检测农业病虫害研究的背景，意义，研究现状，并对研究的大体内容作了说明。

第二章是相关知识，本章主要介绍了与RCNN算法的基本知识，包括神经网络，卷积神经网络和目标检测算法。

第三章是算法设计与分析，依据一个具体的农作物病例，分析研究所要解决的主要问题。并且按照RCNN原理，详细描述了网络结构设计过程以及原因。

第四章是算法测试与评价，描述了网络训练的过程和数据集预处理等，并结合最新在神经网络训练中卓有成效的一些技术，改进算法，调节网络结构。最后给出了在病例数据集上检测的准确度，和对结果的评价。

第五章，总结所做的工作，以及未来需要努力的方向。

2神经网络与深度学习

2.1神经网络

2.1.1感知器

神经网络的基础是感知器，神经网络由许多感知器组合而成。这是一种人工神经元模型，模仿神经元做出反应的过程。感知器原理可以被看作依据权重来做出决策的设备。一个感知器模型（如图2.1.1）接受几个二进制的输入x1,x2,x3…,并产生一个二进制的输出。权重可以理解为相对于输出，输入的重要程度。

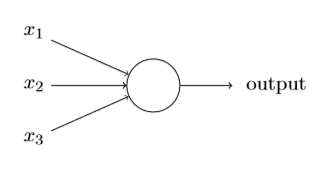


图2.1.1 感知机模型

感知器的代数描述如下，输出由输入乘以权重相加起来的总和与阈值的关系所决定。

(2.1.1)

多层感知器：第一层的输出作为第二层感知器的输入，第二层中的感知器比第一层中的感知器作更复杂的决策，第n层同理，这便是神经网络的雏形。

2.1.2分类器

分类器根据数据的特征用来决定数据的分类，用于神经网络的分类器主要有KNN，SVM，Softmax。

KNN分类器是K近邻分类器，对于被预测数据所属的种类，由K个最靠近他的数据的种类所决定。它判断数据是否属于某一类别的依据是当前数据与种类的距离，具体到图像分类上，有如下两种方法

L1距离（Manhattan distance） (2.1.2a)

L2距离（Euclidean distance） (2.1.2b)

这两种方法可以根据所面对的问题灵活选择。

KNN是比较简单的分类器，在训练阶段只是负责记忆，在预测阶段将待测数据与记忆的所有数据进行比较。因此KNN分类器在训练阶段的时间为，而在预测阶段的时间是。这显然是个问题，我们希望分类器将时间花在训练上，而不是预测上。因为在预测阶段花费时间长意味着我们不能在实践中实时做出决策。

SVM和Softmax分类器属于另一种分类器,它们将时间花在训练上而不是预测上。这种方法由两部分组成：一个是评分函数（score function），它是原始图像数据到类别分值的映射。另一个是损失函数（loss function），它用来量化预测分类标签的得分与真实标签之间一致性。该方法可转化为一个最优化问题，在最优化过程中，将通过更新评分函数的参数来最小化损失函数值。

2.1.3激活函数

我们想要让神经元学习变为可能，就需要任何权重或偏置的微小变化可以引起输出的微小变化。感知器的缺点是：网络中单个感知器上一个权重或偏置的微小变动有时会引起输出的完全翻转。激活函数可以解决这个问题，可以使神经网络拟合任意给定的函数。如果不使用激活函数，那么即使再多层的神经网络和一层也没有区别，只是在作矩阵相乘。

经常在神经网络中使用的激活函数如下：

Sigmoid： (2.1.3a)

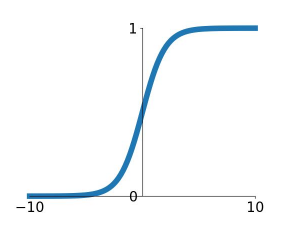


图2.1.3a Sigmoid函数形状

Tanh：tanh(x) (2.1.3b)

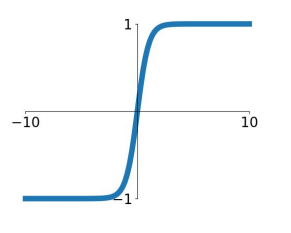


图2.1.3b Tanh函数形状

ReLU： (2.1.3c)

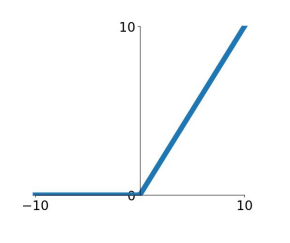


图2.1.3c ReLU函数形状

Leaky ReLU： (2.1.4d)

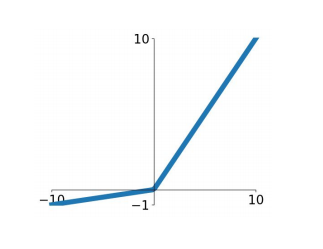


图2.1.3d Leaky ReLU函数形状

ELU： (2.1.3e)

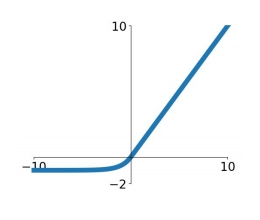


图2.1.3e ELU函数形状

2.1.4损失函数

为了找到合适的权重和偏置，我们需要有一个评价函数，这就是损失函数。

神经网络的损失函数：

(2.1.4a)

这个公式中符号表示的含义为：y(x)表示期望输出，w表示所有网络中权重的集合，b是所有的偏置，n为训练的个数，a表示输入为x时输出的向量。a取决于x、w、b。

我们感兴趣的是能正确分类的图像的数量，为什么不直接最大化这个数量，而是最小化一个类似二次代价的函数来间接评价呢？

因为被正确分类的数量关于权重并不是一个平滑的函数。大多数情况下对权重和偏置的微小变化不会明显的影响被正确分类图像的数量。我们就会陷入一个泥潭，很难改变权重和偏置做出微小的变化来提升性能。

Multiclass SVM Loss：

(2.1.4b)

(2.1.4c)

这是SVM分类器的损失函数，也被称为合页损失函数（Hinge loss）。随着的升高，损失函数不断下降，只到超过某一阈值。这个公式里的1被称为安全边际值（dela）。只要分类器得出的结果中正确分类的结果大于不正确分类结果的，并且超过安全边际值，则认为分类器良好没有损失，否则将计算损失。这里有一个有用的调试技巧，当使用一些很小的值来初始化w，你的分数结果在训练的初期倾向于呈现较小的均与分布的值，所有的得分都近乎为0并且差不多相等，损失函数预计会是C – 1。若果不是程序中就有bug。理解算法原理，使用不同的输入对程序输出结果做出预测是验证程序是否正确的重要方式。

Softmax Loss：

相对于SVM分类器，SVM只关心最后得分相差多少，Softmax给得分赋予新的含义，使用每个类的得分去计算相应的概率。具体损失函数（交叉熵损失函数）如下：

(2.1.5d)

每个概率都介于0和1之间，所有概率和为1.

(2.1.5e)

真实类别的概率再取负值。加负号因为损失函数是度量坏的程度而不是好的程度，加上负号更自然。在这公式中，因为有指数运算，容易造成运算溢出，所以在实际编程中会在类别得分中，选择一个最大的得分，然后所有的得分都减去这个最大值。

在实际应用中，这两种分类器的表现是相似的，根据测试的结果来选择表现更好的分类器。SVM分类器中，正确分类相较于不正确分类，已经得到了比边界值还要高的分数，那么就认为损失值是0，不再去优化。Softmax分类器中，对于分数是永远不会满意的，正确分类总能得到更高的可能性，错误分类总能得到更低的可能性。由于Softmax的特性，它可以获得更加精细的分数。而SVM的特性也可以被看作优势，可以提高分类器的鲁棒性。

2.1.5梯度下降

有了损失函数来评价神经网络的好坏，那么如何找到损失函数最小值点呢。直接找到是不可能。有两种方法：第一种是一个很自然的想法就是随机搜索，但在高维空间中搜索的范围太大了，这种方法在实践中不具有可行性。

可以将函数想象成一个山谷，假设有一个小球从山坡的斜坡上滚落下来，日常经验告诉我们这个球最终滚到谷底。我们可以用这一想法找到最小值。一开始小球随机选择一个起始位置，然后模拟球体滚到谷底的运动。计算函数的导数可以告诉我们山谷中局部的形状，由此我们知道球体怎么滚落。所以另外一种方法就是计算斜率。计算斜率又有两种方法，第一种是有限差分法，根据导数定义，保持一个变量变化，其他参数不变，来计算变化前后的输出差值。这个方法我们每对一个参数求偏导，就要两次在神经网络中的前向传导。在动辄几千万的神经网络面前，时间花费巨大，也是不可行的。但可以使用来检验自己程序中梯度计算是否正确，设置一组具体的参数，改变某个参数的值，来计算变化前后的差值，观察程序输出的斜率是否一致。另一种方法是利用微积分的公式，直接推导梯度。

我们只要告诉球体“现在向下”，沿梯度的相反的方向是函数在该点下降速度最快的方向。因此我们每次按照梯度相反的方向计算小球移动的位置。这个过程被称为梯度下降。更新规则可以写为：

(2.1.5a)

公示中的，被称为学习速率，他决定小球一步要走多远，这是一个超参数，神经网络训练中首先要经过大量测试，来确定这个值，它要保证损失函数可以平滑的降到最低点，太大会导致小球越过最小值点并且来回震荡，太小又会导致神经网络训练时间的加长，很难快速的到达最小值点。反复使用这个公式，来计算下一次移动位置，直到C到一个最小值。

总结一下，梯度下降算法工作方式就是重复计算梯度，然后沿着梯度相反的反向移动。

2.1.6反向传播

梯度下降算法中最重要的就是计算梯度，我们已经知道可以用微积分公式推导梯度的计算公式，但是在正向推导中，会发现很明显的问题，就是很多节点的梯度值被重复计算，造成了极大的计算浪费，解决方法就是从神经网络的最后一个节点开始计算，反向推到，前一个节点只需知道上游传过来的上游有梯度，与自己本身节点的梯度相乘，然后继续向第一层几点的方向传播，这样反向遍历一次神经网络就可以将所有节点的梯度计算出来。具体实例如图2.1.6a所示。

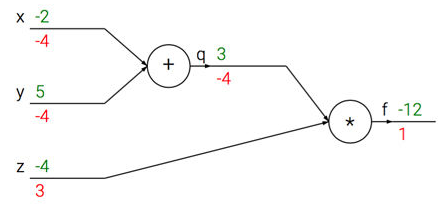


图2.1.6a 神经网络前向传导与后向传播模型（图片源于Stanford University CS231n lecture3）

反向传播是链式法则的递归调用，反向传播是一个优美的局部过程。在整个计算线路图中，每个门单元都会得到一些输入并立即计算两个东西（如图2.1.6b）：这个门的输出值和其输出值关于输入值的局部梯度。门单元完成这两件事是完全独立的，它不需要知道计算线路中的其他细节。然而，一旦前向传播完毕，在反向传播的过程中，门单元门将最终获得整个网络的最终输出值在自己的输出值上的梯度。链式法则指出，门单元应该将回传的梯度乘以它对其的输入的局部梯度，从而得到整个网络的输出对该门单元的每个输入值的梯度。

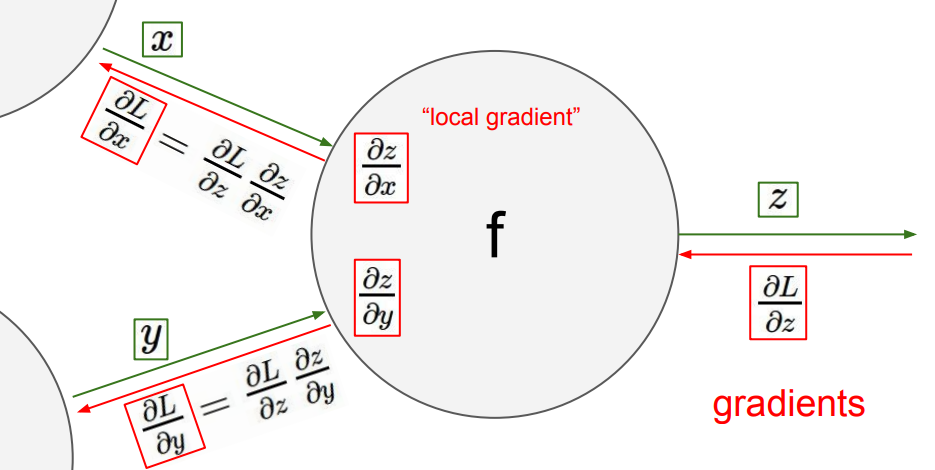


图2.1.6b 后向传播计算过程图（图片源于Stanford University CS231n lecture3）

2.1.7正则化

正则化是帮助减轻神经网络过拟合的技术。假设有一个数据集和一个权重集，能够正确的分类每个数据。问题在于这个权重集的个数可能并不唯一。我们希望模块能够以某种方式选择更简单的权重，以此来消除模糊性，迫使线性回归朝着简单多项式演进。我们可以在损失函数上添加一个正则惩罚项，最常用的是正则惩罚范式是L2范式：

(2.1.7a)

加入正则化的损失函数为：

(2.1.7b)

直觉上看，规范化的效果是让网络倾向于学习小一点的权重，因为我们希望网络输出的结果不是由某一参数决定，而是由整体的结果决定的，正则化可以抑制大的参数，是参数更加的平均化，提高神经网络的鲁棒性。它可以被理解为一种寻找小的权重和最小原始代价函数的折中。公式2.1.7b中的，是一个超参数，用来控制前后两部分的重要性，越小就越偏向最小化原始代价函数，反之，倾向于小的权重。

小的权重在某种程度上来说意味着更低的复杂性，也就是对数据给出了一种简单却更加强大的解释。不去规范化偏置的原因是有一个大的偏置并不会像权重那样使神经网络对输入变得太多敏感。允许大的偏置能够让网络更加的灵活，大的偏置可以让神经元更加容易饱和。

2.2卷积神经网络

卷积神经网络是目前在图像分类领域最流行的神经网络。它由神经网络发展而来，与神经网络相似，也是由神经元构成，具有学习能力的权重和偏置。整个网络依旧是一个可导的评分函数，在网络的最后一层往往是一个全连接层，也有一个和神经网络一样的损失函数。卷积神经网络的结构基于一个假设，即输入数据是图像，基于该假设，我们想在网络结构中添加一些特有的性质，使得前向传播函数实现起来更高效。它在特征采集上的结构上变了，大幅减少了网络中参数的数量。卷积神经网络的结构如图2.2。

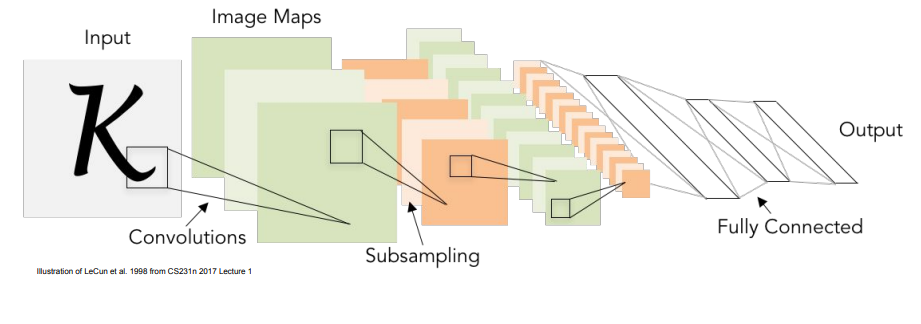


图2.2 卷积神经网络实例结构模型（图片源于Stanford University CS231n lecture5）

2.2.1卷积层

全连接神经网络有一个缺陷，没有考虑像素的空间结构，在完全相同的基础上对待相距很远和相聚很近的输入像素。生物神经元具有局部连接性，不关注于输入图片的全部，关注图像空间的一个局部区域。卷积层的设计依据这种思想，可以很好的保留空间结构，并且适应图像的平移不变性。卷积层采用了两种基本概念：局部连接和共享权重。

局部连接就是让每一个神经元与输入数据的一个局部区域连接，该连接的空间大小叫做神经元的感受野，如图2.2.1中，感受野为5\*5大小。感受野的尺寸是一个超参数。在深度方向上，连接的大小总是和输入量的深度相等。卷积层的输出大小由3个超参数控制：深度，步长，零填充。输出数据大小的计算公式为：

(2.2.1a)

共享权重是对于每个神经元使用相同的权重和偏置。这大大减少了神经网络的参数。意味着每个神经元检测相同的特征。卷积核按步长扫描输入数据，输出一张激活表。卷积核的数目对应输出激活表的数目。

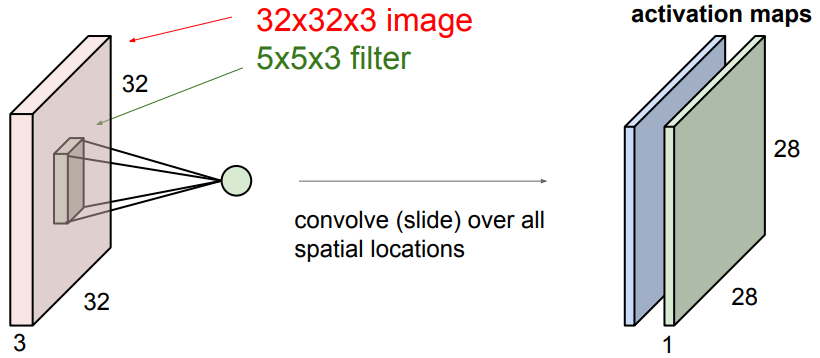


图2.2.1 卷积层模型（图片源于Stanford University CS231n lecture5）

2.2.2池化层

池化层通常跟在卷积层之后使用。它要做的是简化从卷积层输出的信息。池化层的操作类似于卷积层操作，也包含池化范围，步长，零填充等，不过操作不再是卷积而是在池化范围按照池化规则对激活值进行操作，例如最大值池化（如图2.2.2）是在池化范围内选出最大的激活值，平均值池化是在池化范围内对所有的激活值求平均值。对于图像处理，最大值池化要好于平均值池化，最大值池化可以保留更明显的激活特征。池化操作一般不在深度方向上池化处理，而是在平面上。所以池化层输出的深度与输入是一样的，池化操作的步长一般是不重叠的。总结一下，池化层的目标是尽力用一个数值来表示整个区域，目的为了降采样处理，经过池化层将减少75%的参数。

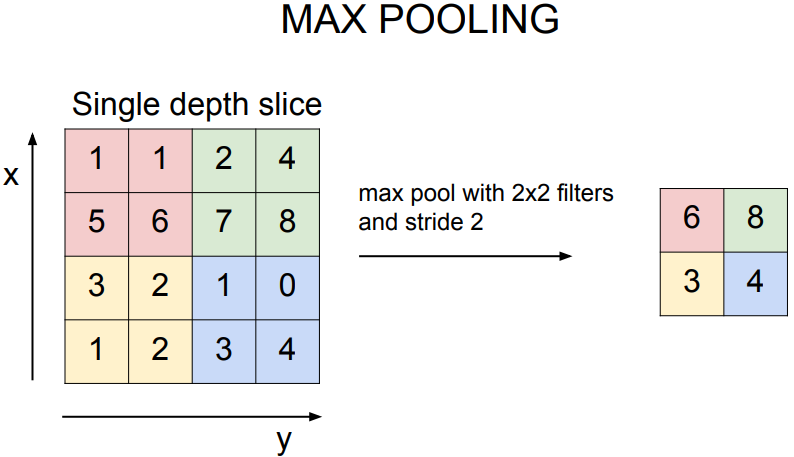


图2.2.2 最大值池化操作（图片源于Stanford University CS231n lecture5）

2.2.3全连接层

全连接层通常在卷积神经网络的最后，在图2.2的最后3层就是全连接层，它用来将最后的输出映射到线性可分的空间。这一层是传统的神经网络，起到输出得分的作用。确切的说，这一层将每一个输入数据连接到一个神经元，将卷积层提取的特征汇聚起来，得到相应类别的分数。全连接层将卷积层产生的特征图映射成一个固定长度(一般为输入图像数据集中的图像类别数)的特征向量。这个特征向量包含了输入图像所有特征的组合信息，虽然丢失了图像的位置信息，但是该向量将图像中含有最具有特点的图像特征保留了下来以此完成图像分类任务。

2.3 基于卷积神经网络的目标检测算法

目标检测与识别（如图2.3）作为计算机视觉领域的基石，也越来越受到重视。在实际生活中应用也越来越广泛，例如目标跟踪，视频监控，信息安全，自动驾驶，图像检索，医学图像分析，网络数据挖掘，无人机导航，遥感图像分析，国防系统等。目标检测算法的任务是在指定的一张图片中找出要识别的目标。所以不仅仅是要识别出图像中的物体是什么，还要把物体的范围标识出来。由于给定的图像没有所包含物体的数量不确定，任务难点在于待测区域的提取与识别。

任务具体的过程有：

1. 从场景中选出待测区域。
2. 识别待测区域的目标。
3. 精调分类模型，有效候选框的位置精修。

如今目标检测与识别的研究方法主要由两大类：基于传统图像处理和机器学习算法的目标检测与识别方法；基于深度学习的目标检测与识别方法。

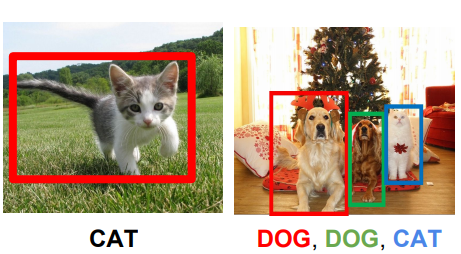


图2.3 目标检测任务（图片源于Stanford University CS231n lecture11）

2.3.1 传统目标检测算法

传统的目标检测与识别方法主要可以表示为：候选区域确定、目标特征提取、目标识别、目标定位。首先按照大小固定的窗口在图像水平和水平方向平移以遍历图像，然后提取特征。这里所用到的特征都是人为设计的，例如SIFT (尺度不变特征变换匹配算法Scale Invariant Feature Transform)，HOG(方向梯度直方图特征Histogram of Oriented Gradient),，SURF( 加速稳健特征Speeded Up Robust Features)等。通过这些特征对目标进行识别，用到SVM等分类器，然后再结合相应的策略对目标进行定位。但传统算法存在的问题是：滑动窗口虽然简单，但产生大量无效窗口，严重影响检测的速度。其次特征主观性大，鲁棒性低，导致准确率不高。

2.3.2 RCNN

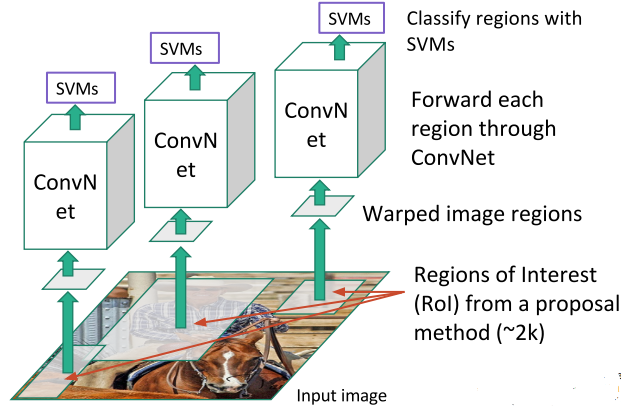


图2.3.2 RCNN结构模型（图片源于Stanford University CS231n lecture11）

RCNN(Regions with CNN features)是将CNN方法应用到目标检测问题上的一个里程碑，是利用深度学习进行目标检测的开山之作。这个算法借助卷积神经网络强大的特征提取和分类的性能，通过使用Region Proposal方法实现将目标检测任务转化为分类问题。

算法可分为三步（如图2.3.2）：

1. 候选区域的选择

选择搜索算法，通过算法利用图像中的色彩，梯度，边缘等信息，选出有可能存在目标的区域，Selective Search方法从一张图像生成约2千个候选区域。候选区域生成和后续步骤相对独立，实践中可以利用任意合适的算法进行。

1. CNN特征提取

首先预处理候选区域，因为卷积神经网络接受的区域大小固定，所以候选区域归一化为统一的大小，后续分类中使用AlexNet模型，RCNN中大小固定为227\*227\*3。然后将归一化后的候选区域送入卷积神经网络得到固定长度的特征。

1. 分类与边界回归

类别判断，对于每一类，使用一个SVM二分类器进行判别。将上一步输出的特征向量输入SVM分类器，最后边界回归（bouning-box regression）得到精确的目标区域。

目标检测的标注数据很少，如何用少量的标注数据训练高质量的模型，可以使用迁移学习。迁移学习是使用已经训练好并且表现出良好的网络模型，冻结网络结构中的特征提取阶段的参数，根据具体任务可以替换卷积神经网络最后的分类器，只训练特征分类阶段的参数。RCNN中先使用ILSVRC2012数据集训练一个1000类的分类器模型，然后使用该模型来初始化我们的CNN模型参数，使用我们的训练和验证数据集合进行微调。

因为CNN是有监督的学习，我们需要对候选区域的类别进行标注。我们要先标记出目标的真实边框和类别，再计算候选区域与真实边框的交集和并集，在计算交集和并集的比例，如果这个比例大于0.5，则将这个区域标记为正样本，否则就是负样本。在SVM训练时，这个比例小于0.3的区域为负样本。其他的区域忽略，真实边框作为正样本。这样做是因为，CNN训练时做了比较宽松的标注，候选区域只包含物体的一部分，那么我也把它标注为正样本，可以防止CNN过拟合。而SVM训练的时候，SVM适用于少量样本训练，对于训练样本数据比较严格，只有当候选区域将整个目标包含进去才把它标记为正样本。

RCNN在PASCAL VOC2007数据集上的检测结果由历史上的32.3%提升到66%，证明了深度学习在目标识别上的巨大潜力。

3算法分析与设计

3.1 问题描述

3.1.1 问题定义

为了识别病虫害，我们需要定期对农作物生长情况进行拍摄取样，然后检测图像中目标农作物的生长状况。一张图像中存在多个目标，病虫害的检测可以看成一个目标检测的任务。我们的任务是在采样图像中检测出所有的患病目标，图像中其他的内容可能是树叶，枝干，土地等，与待测目标有明显的区分，在检测算法中当作背景样本。目标是在多边的背景中准确快速的找出所有患病的目标是任务的关键。

柑桔在我国种植广泛，栽培历史悠久，种类繁多，我国是世界上最大的柑桔产地。柑桔疮痂病（图3.1.1）是柑桔主要的真菌病害之一，在我国各个柑桔种植区均有发生。主要危害幼叶、幼果，严重影响柑桔的质量和产量。且柑桔疮痂病病状明显，患病果实与正常果实有明显区别。因此选用柑桔疮痂病适合作为病虫害检测算法的测试病例。



图3.1.1 柑桔疮痂病

3.1.2采用模型

由于采用深度卷积网络的RCNN算法在目标检测中实现了突破性的进展，对于病虫害检测任务，RCNN算法理论上可以表现出很好的效果。在实际测试中，基于面对的特定的农业病虫害检测，重新设计网络结构，提出了自己的改进。

3.1.3评测标准

算法的评测标准是在给定的检测场景中可以准确找出被检测目标，并识别目标是否患病。计算目标识别的准确率来判断算法的优劣。

3.2实验数据集

3.2.1数据集采集

因为RCNN算法中，应用迁移学习的思想，需要进行预训练。预训练是用现有的大量数据集提前训练特征提取阶段的参数，解决目标检测标注数据很少的问题。但预训练有一个前提，预训练使用的数据集必须和检测数据集有很高的相似性。不然两个数据集中物体种类都不相同是不可能进行迁移学习的。如果两个数据集相似，可以根据待测数据集的大小进行微调。检测数据集非常小时，可以修改预训练网络的顶层，替换为一个线性分类器，用小数据集去训练这个分类器。检测数据体大一点，为了追求更高的准确率可以微调更多层的参数。但是如果两个数据集中的种类完全不同，那基本上不可能使用迁移学习。这就如同用个飞机分类器去识别青蛙的种类。

由于农业病虫害检测还处于起步阶段，并没有发现公开的病虫害数据集，包含多种农作物病例数据集没有，只含有柑桔疮痂病的数据集也没有发现。只能根据现有的柑桔病例图片创建数据集。创建柑桔疮痂病数据集要搜集患病柑桔和正常柑橘的图片，并为他们标上是否患病的标签。图片要求只能包含一个目标，需要将收集来的图片，进行切割和标注。鉴于公开的柑桔疮痂病的图片有限，正常柑桔的图片有很多，但是为了数据集的代表性，数据集中正常柑桔和患病柑桔的数量应该各占一半，并且他们的顺序不能前边一半是患病柑桔后一半是正常柑桔，需要将他们的顺序随机打乱。所以找收集到患病柑桔的图片只有45张，正常柑桔图片45张，共90张的数据集。图3.2.1是数据集中部分数据的展示。由图可以看出柑桔患病可能出现在柑桔的任何位置，且疮痂病的病状较小，分布比较随机，患病柑桔的形态各异，加之数据量很小，所以这是一项很有挑战性的任务。



图3.2.1 柑桔疮痂病

3.2.2数据集划分

在训练神经网络中，我们经常会把数据集划分为两类，训练集和测试集，训练集用来训练神经网络，测试集用来检验神经网络的准确率。但对于存在超参数的训练集往往是不够的，如果用测试集来检验网络中超参数的好坏，那么就相当于选择超参数是对测试集的拟合，这导致我们在实际部署的时候，神经网络的性能可能远远低于我们的预期。所以我们应该在训练集中再划分出一部分当作验证集，在验证集上对超参数进行调优。这样当训练好的神经网络在测试集上测试就相当于完全新的环境，这时得出准确率才有说服力。综上，我们将柑桔数据集划分为包含80张图片的训练集，和用于超参数调优的包含10张图片的验证集。

3.3算法实现

3.3.1特征抽取网络

RCNN算法中使用的AlexNet网络（图3.3.1），这个网络在2012年的图像识别大赛取得冠军，领先第二名很高的准确率。这是一个有8层的深度卷积神经网络，包含5个卷积层和3个全连接层，接受227\*227\*3的图像输入。第一层中有96个卷积核大小为11\*11\*3的卷积核，步长为4，操作依次顺序是卷积，激活、池化、归一化，最后输出27\*27\*96的激活图。第二层与第一层的结构一样。第三层和第四层一样，和前边的区别是只有卷积和激活的操作。第五层在第四层的底层上多了池化操作。剩下是3层是全连接的网络层用于汇聚提取的特征并得出类别得分。整个网络共有5千万多的参数，并用两个GPU进行并行运算。这是一个巨大的神经网络对计算机的硬件要求非常高。它能区分1000多个类别。由于实验计算硬件的限制和柑桔疮痂病的数据集的较小和我们只想对一种具体的病例进行区分，深度小一点的卷积网络完全可以胜任。RCNN原理中3个操作的主要步骤都是独立的，我们可以根据具体面对的环境进行选择合适的网络结构。

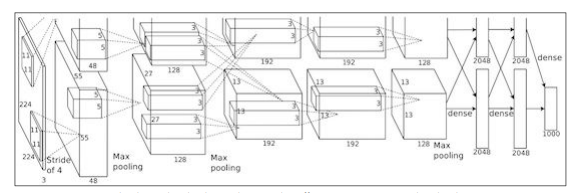


图3.3.1 AlexNet网络结构

针对柑桔疮痂病的，我们不是要检测这是一个柑桔，而是要检测柑桔是否患病，而病症通常是比较小的斑点，为了我们的卷积核能够提取到这些病斑，我们必须尽可能的保留输入数据的像素，鉴于数据集的像素太小和计算机硬件的运算速度，采用100\*100\*3的原始数据作为神经网络的输入。采用3层的卷积神经网络，第一层是卷积层，包含卷积、激活和池化操作，对原始数据进行特征提取。第二层是全连接层汇聚特征，最后一层连接一个两分类的SVM分类器的出柑桔患病和和正常的得分。检测网络结构如下：

CONV- RELU - MAXPOOL - AFFINE - RELU - AFFINE – SVM。

下列代码展示网络中各层前向传播，反向传导和损失计算的过程。

def loss(self, X, y=None):

"""

Evaluate loss and gradient for the three-layer convolutional network.

Input / output: Same API as TwoLayerNet in fc\_net.py.

"""

W1, b1 = self.params['W1'], self.params['b1']

W2, b2 = self.params['W2'], self.params['b2']

W3, b3 = self.params['W3'], self.params['b3']

# pass conv\_param to the forward pass for the convolutional layer

filter\_size = W1.shape[2]

conv\_param = {'stride': 1, 'pad': (filter\_size - 1) // 2}

# pass pool\_param to the forward pass for the max-pooling layer

pool\_param = {'pool\_height': 2, 'pool\_width': 2, 'stride': 2}

scores = None

out, con\_cachei = conv\_relu\_pool\_forward(X, W1, b1, conv\_param, pool\_param)

out, aff1\_relu\_cache=affine\_relu\_forward(out, W2, b2)

out, aff2\_cache=affine\_forward(out,W3, b3)

scores=out

if y is None:

return scores

loss, grads = 0, {}

# data\_loss, dscores = softmax\_loss(scores, y)

data\_loss, dscores = svm\_loss(scores, y)

da2, dW3, db3 = affine\_backward(dscores, aff2\_cache)

da1, dW2, db2 = affine\_relu\_backward(da2, aff1\_relu\_cache)

dX, dW1, db1 = conv\_relu\_pool\_backward(da1, con\_cachei)

# Add regularization

dW1 += self.reg \* W1

dW2 += self.reg \* W2

dW3 += self.reg \* W3

reg\_loss = 0.5 \* self.reg \* sum(np.sum(W \* W) for W in [W1, W2, W3])

loss = data\_loss + reg\_loss

grads = {'W1': dW1, 'b1': db1, 'W2': dW2, 'b2': db2, 'W3': dW3, 'b3': db3}

return loss, grads

下面详细介绍检测网络每层的具体操作和使用的原因。

3.3.2激活函数选择

设计神经网络结构中很重要的一步就是选择合适的激活函数。Sigmoid激活函数这是一种近似于神经元放电率的函数。但它有显著的缺点。Sigmoid函数的正无穷和负无穷方向导数都趋近于0，造成了神经网络的梯度消失的问题。如果输入激活层的数据较大会导致神经元饱和进而在本层的梯度为0，后层传导过来的梯度经过这层时会变为0，这样神经网络在训练的时候，权重得不到更新，神经网络完全不能训练。Sigmoid函数是非零中心函数，它的值域是(0,1)，这就导致传入下一层的数据完全为正，下一层的导数也是正的。这将导致神经网络的学习非常缓慢。因为假设权重是二维的情况，那么函数只能按照第一象限和第三象限的反向前进，如果最小值点在第二或者第四象限，那么函数只能沿折线曲折的到达最值点。Tanh激活函数的值域是(-1,1)解决了非零中心的问题，但是依然存在过饱和的现象导致梯度消失。ReLU激活函数被发现更接近于生物的神经元的性质，而且计算简单，只是比较输入与零的大小，对于零的数据不变，小于零的数据为零。它解决过饱和问题，但是没有解决非零中心的问题，同时出现了新的问题，会造成dead ReLU的现象，本质也是造成梯度消失问题，也就是说这个激活函数只解决了在正半轴的梯度消失问题。但是在我们实践中，ReLU由于运算简单比其它激活函数快6倍，而且dead ReLU现象可以通过设置较小的正偏置初始化ReLU以增加它在初始化时被激活的可能性。其他两种激活函数也很常用，可以通过限制输入让梯度集中在激活函数的线性区域来防止梯度消失的现象。但为了在小数据集上快速的收敛，在监测网络中选择ReLU作为激活函数是非常有效的，并减少梯度消失带来的危害性。

3.3.3参数更新

参数更新可以使用梯度下降法。在随机梯度下降中，计算一次梯度，需要对所有的数据输入网络，求出平均梯度，但在数据量很高的数据集中是会花费非常高的时间。这里选择使用随机梯度下降法，用小批量数据的去近似全部数据的平均梯度，这会损失导致下降缓慢，但可以通过增加迭代次数来解决。由于梯度下降存在的问题可能会卡在损失函数的鞍点处，或在鞍点处非常缓慢的下降。原因是我们没有考虑历史梯度的影响，比如小球在山坡滚落，速度会不断增加，即使有鞍点也能快速通过。基于这一思想的参数更新方法是动量（Momentum）更新，具体方法是为：

现在梯度只是影响速度，速度再来影响位置。

为了让损失函数更为准确的收敛到最小值，保持学习速率衰减是一个很好的方法。在训练的开始，我们希望能够快速下降，在接近最小值点的时候，学习速率可以慢慢下降以接近最小值点。图3.3.3a展示了动态学习速率使损失函数可以更好的收敛。

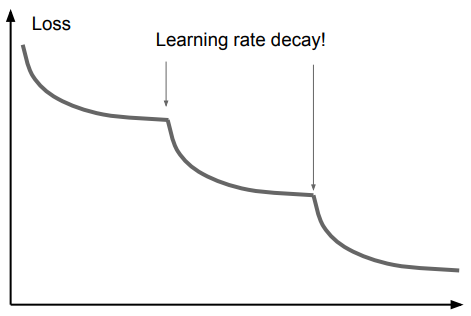


图3.3.3a 动态学习速率

Adagrad是一个由Duchi等提出的适应性学习率算法。

公式中的eps是为了防止分母为0，一般设在到之间。高梯度值的权重更新被减弱，低梯度值的权重更新效果将会增强。但有一个令人担忧的地方，单调递减的学习速率可能过于激进而过早的停止学习。Adam是最近才提出的一种更新方法，它结合了Adagrad和Momentum的思想，既可以适应学习速率又有历史梯度的影响，可以快速到达损失函数的最值点。

由图3.3.3可以看出Adam具有其他两种算法的优点，在相对快速的情况下，同时更新参数不会导致损失函数的波动，因此在设计网络模型中选择Adam参数更新的方式。

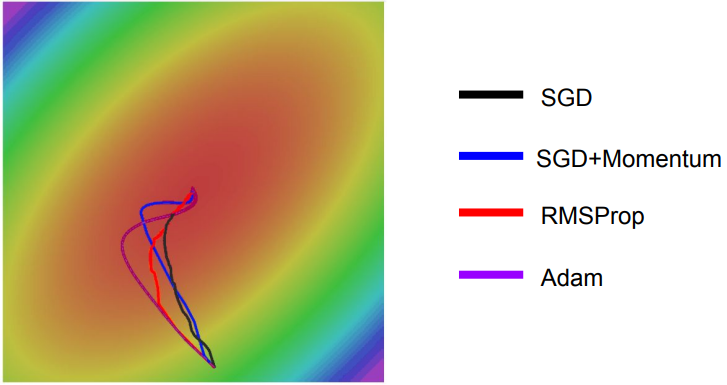


图3.3.3b四种参数更新方式比较（图片源于Stanford University CS231n lecture7）

3.3.4随机丢弃层

在小数据集的情况在需要防止网络过渡拟合，除了选择增加正则惩罚项以外，还可以选择最近提出的神经网络过拟合的弃权算法。这是一种很激进的技术。与正则化不同，弃权技术不会对损失函数进行修改，而是对网络结构的修改。弃权的工作方式随机地删除网络中一部分的隐藏神经元，每次更新完权重后在随机丢弃一部分神经元。图3.3.4形象的展示了工作的流程。弃权技术起作用的原理是，每次更新舍弃一部分神经元，就像同时训练多个神经网络，各个神经网络对同一输入的结果是不同的，通过平均或投票的方式来决定那一个输出。它是基于生物神经元不会依赖同层次其他神经元而存在的思想。本质就是减轻同层次神经元之间的相互适应。实践证明它在提升神经网络的性能应用的相当成功。在检测算法中使用弃权技术可以提升在验证集上的准确度，有效防止过拟合小的数据集。

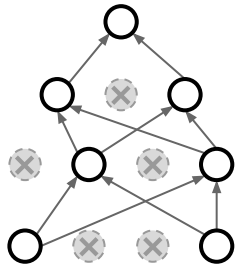


图3.3.4弃权技术

3.3.5零填充

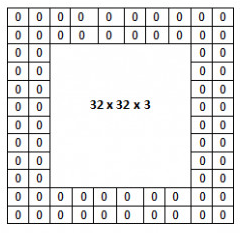


图3.3.4零填充

零填充可以用来调节卷积核的大小，并且可以防止信息过快的减小。最重要的是它可以较好的保留图像的边缘信息。填充方式就是在输入数据的周围补零。补零的边距根据卷积核和输出数据的大小决定。图3.3.4是在输入数据上做了两层的零填充，零填充可以控制输出数据的空间尺寸，当步长为1时，可以保证输出和输入有相同的空间尺寸。由于柑桔疮痂病病状位置分布不均匀，图片的边缘也可能存在病状，所以在检测网络中采用填充3层零填充尽可能保留所有位置的病状特征。

3.3.6数据增强技术

为了提高监测网络在小数据集训练下的准确度，除了调整网络网络结构和损失函数外，还可以使用数据增强技术（Data Augmentation），它是通过图像处理技术增加数据集的数据量。比如对一张图片左移一个一个像素对人类来说没有多大区别，对于计算机来说整个矩阵完全变了，但是图片的标签不变。类似的方法还有，灰度转换，水平翻转，色值跳变，旋转，随机裁剪和缩放等，来获得数倍于原数据的训练样本。图3.3.6是数据增强技术对正常柑桔图片水旋转处理。

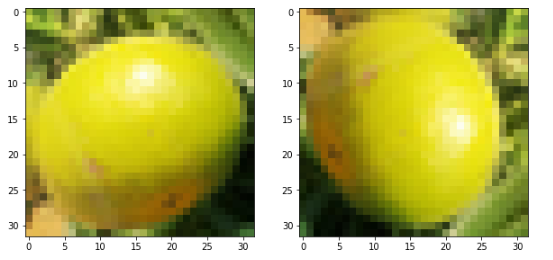


图3.3.6数据增强技术

3.3.7参数初始设置

在训练神经网络时，首先面对的是权重初始化。基于我们上边讨论的，初始化权重必须谨慎。如果权重全部初始化为零，虽然并不会让神经元消亡，神经元能处在任何的状态，这取决于神经元初始的偏置。直觉上来讲，神经元不会获得任何的输出，神经元之间相互独立。在你的输入数据上有相同的操作，输出相同的数据值，得到相同的梯度，神经网络会使用相同的方式进行更新，得到完全相同的神经元。使用相同的参数进行初始化不会打破参数对称的问题。如果使用过小的权重，会使神经网络每一层的输入数据非常小，导致每一层的梯度非常的小，神经网络就会更新缓慢。可以通过尝试增加权重来解决。但是过大的权重会导致神经网络爆炸，梯度会变得的非常的大，损失函数的更新会来回波动，进而导致神经网络无法更新。好的方法是使用Xavier initlization技术，这是一种根据输入数据动态调节初始化权重的技术，对于大的输入数据会使用小的权重，对于小的输入数据会使用大的权重。原理就是指定输入的方差等于输出的方差。

对于神经网络中的超参数，我们选择宽泛策略，简化问题的架构表面上可能会降低你的效率。实际上这样能够将进度加快，更快的找到传达有意义信号的网络。希望更快更频繁地监控网络的验证准确率来获得反馈，选择使用数据集中的一小部分可以快速达到这个效果。学习速率是训练网络的最重要的超参数，它决定网络的收敛速度，第一步确定学习速率是非常明智的选择。合适的学习速率是可以让损失函数能够平滑的下降。对于每一个要确定的超参数列一个大体范围的列表，用程序使用随机网格搜索确定参数表现优秀的粗略范围，重复这一步来确定大致范围。最后在验证集上，细致的确定超参数的最优选择。

3.3.8数据集预处理

设计网络模型所接受的输入为100\*100\*3，这就要求必须将收集到的图片，缩放成神经网络可以接受的大小。为了提高神经网络的鲁棒性，数据零中心化也是必要的处理。数据零中心化可以让线性分类器更好划分数据，提高分类器对噪点的抗干扰能力。零中心化的另一个重要作用就是减少神经网络训练中梯度消失的现象，均匀分布在原点处的数据可以让神经网络更快的收敛。图3.3.6a展示的是零中心化处理的效果。

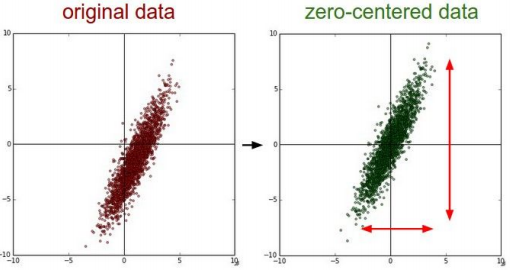


图3.3.6a零中心化处理

零中心化的具体操作是，对一个训练集求平均值，训练集和验证集在被送入神经网络之前都要减去这个平均值。图3.3.6b是柑桔疮痂病训练集上得出的均值图像，从图像中我们可以明显看出一个黄绿色的圆形柑桔。

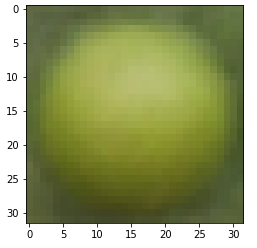


图3.3.6b柑桔数据集均值图像

3.3.9候选区域缩放方式

卷积神经网络只能接受大小固定的输入图像，我们需要将候选区域缩放成大小固定的形状。缩放方式有两种各项异性缩放方式和各项同性缩放。各项异性缩放不管长宽比直接缩放成要求的形状。各项同性缩放是保持长宽比的缩放，但保持长宽比会不符合要求的大小，会产生空隙。有两种各项同性缩放的方式。一种是我们可以采用背景颜色进行填充空隙。另一种方式是对于选定框在原位置上拓展为要求的大小。采用长宽中短的长度进行缩放裁剪虽然可以去除空隙，但是会造成目标部分丢失，加大识别难度。RCNN原理中是采用的是各项异性的缩放，在实践中各项异性缩放效果比同性缩放更好一点。针对柑桔疮痂病病状，检测算法判断是否患病的依据是病斑的形状。由于每个柑桔的形状大小不同，各项异性的拉伸病斑的形状会使用病斑发生严重的变形，增加了神经网络的识别的难度。所以选择同性缩放对于检测小的病斑很重要。在检测算法中选择各项同性的缩放病虫害的是很有必要的，同时柑桔的形状近似于圆形，选择候选区域中长宽中短的那个进行裁剪缩放也可以完整的保留柑桔的形状。

4实验测试与评价

4.1实验环境

本实验的平台为一台装有4核CPU，12GB内存，Nvidia GT755M显卡，240GB固态硬盘的个人PC，操作系统为Windows10，采用编程语言为Python3，编译环境为Anaconda 5.1.0。

由于训练中必须快速将数据集读入内存，这就要求硬盘读取速度快和内存容量大，限于硬件水平可以缩小数据集图片的尺寸来进行训练，但会影响神经网络提取的精度。神经网络中计算大多是矩阵的运算，使用可以进行大量并行计算的GPU加速训练。使用GPU比在CPU上速度提高70倍左右。图4.1为几种常见深度网络分别在CPU和GPU上的运行速度对比。

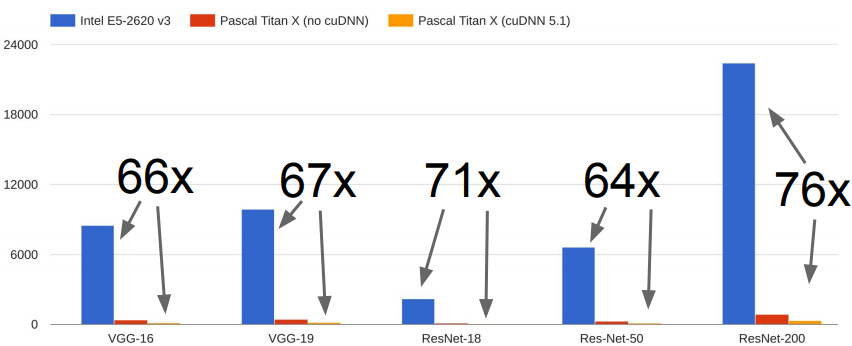


图4.1 GPU与CPU运行速度对比。

选用Python作为深度学习的语言，是因为Python类似伪代码的语法，可以在一行代码中完成向量化的操作，大幅减少代码行数，让人感觉如同在写英文文章。在深度学习中，研究人员就可以不用纠结于具体的语言语法，而把主要精力放到算法本身的逻辑上。Python拥有大量易用的库，如用于矩阵运算的Numpy，和MATLAB功能一样的matplotlib等。Python不需要编译。是一个解释性的语言，速度可能没有编译语言快速，但是可以方便的查看每一行代码的运行效果，灵活的调节参数，而不用对整个程序进行编译，这对神经网络的调参是非常重要的。Python与其他语言有良好的兼容性，Python中的一些库是用C语言写的，加速了算法的执行速度。

4.2训练参数

实验中网络所使用的参数如下：初始化权重的标准差为，偏置初始化为零，正则化参数为0.3，学习速率为，迭代周期为150次，参数更新规则Adam，激活函数为ReLU，学习速率衰减为0.95。

第一层32个大小7\*7\*3的卷积核，步长为1，零填充边距为3，输入数据的尺寸为32\*32\*3。池化层池化区域2\*2，步长为2。输出为32\*32\*32的激活图。

第二层的神经单元为200个。输出长度为200的特征向量。

第三层是二分类的SVM。输出为患病得分和正常得分，两个得分选中最大的作为目标的标签。

测试中网络初始化代码如下：

model = ThreeLayerConvNet(input\_dim=(3, 100, 100),weight\_scale = 1e-2, hidden\_dim=200, num\_classes = 2, reg = 0.3)

solver = Solver(model, small\_data,

num\_epochs = 60, batch\_size = 20,

update\_rule='adam',

optim\_config={

'learning\_rate': 1e-4,

},

verbose=True, print\_every = 5)

solver.train()

4.3实验结果

图4.3a为网络模型的训练过程，训练周期120次，每一个迭代周期输出损失函数的值，和网络在训练集和验证集上的准确率。

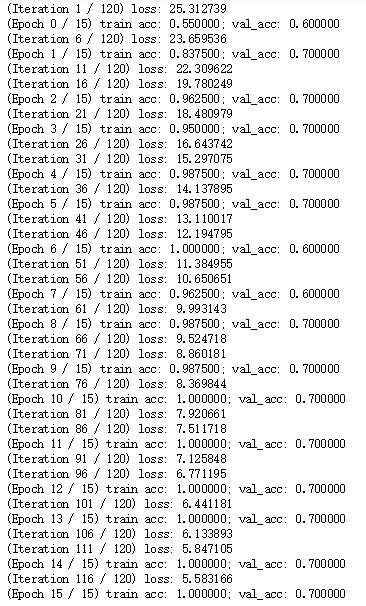


图4.3a 神经网络训练过程

根据神经网络训练的结果，神经网络理论上在可以对柑桔疮痂病的识别达到70%的准确率。

下面是检测算法的整体验证流程。

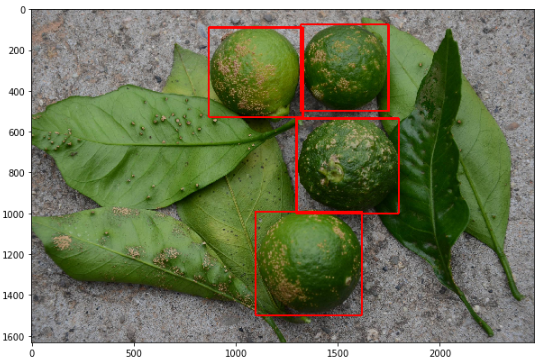


图4.3b 测试图片

第一步选择测试图片，提取可能包含目标的候选区域，并将候选区域选出并标注在图上，如图4.3b所示。

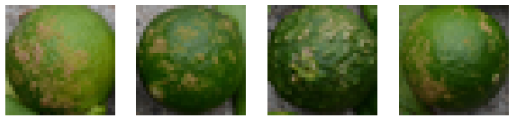


图4.3c 缩放待测目标

第二步是将候选区域的尺寸缩放成神经网络可以接受的大小，在图4.3c中可以观察到使用各项同性缩放可以完整的保留柑桔的形状。

第三步将归一化后的候选区域送入神经网络进行特征提取和分类。最后将识别标签测试标注测试图像上。

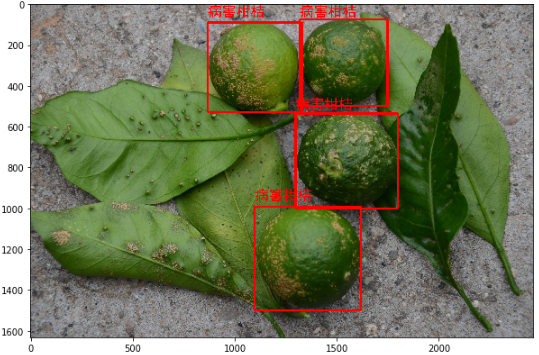
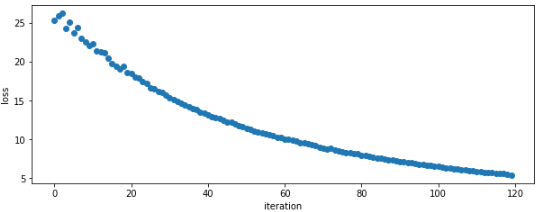


图4.3c 患病柑桔标注

根据图4.3c中展示出的结果，所有患病的柑桔全部被标记。

4.4实验结果评价

4.3.1损失函数和准确率



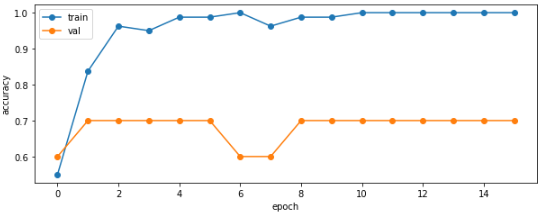


图4.3.1 损失函数和准确率在训练中的变化

图4.3.1上半部分反应了损失函数在训练过程变化的情况，可以看出损失函数下降的非常平缓，符合预期的期望，超参数学习速率的选择合适。图4.3.1的下半部分反应了准确率在验证集和训练集中的变化，可以看出训练集中的准确率在神经网络的训练中快速上升，很快达到准确率100%。验证集中的准确率上升到70%基本上稳定不变。这表明网络发生了过拟合现象，必须进一步扩大数据集来有效的减轻过拟合。虽然验证集的准确率不再发生改变，但是损失函数却不断在下降，说明测试网络的权重正在向着更小更平均的方向变动，符合网络设计的预期。

检测算法的测试过程最终准确率稳定在70%，测试图片中的全部患病图片全部被标注。算法基本达到基本上达到了预期目标，距离实际应用还有可以提升的空间。

4.3.3卷积核展示

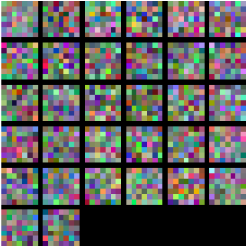


图4.3.1 可视化卷积核

图4.3.1展示了卷积核的内容。因为神经元在学习的时候本质是学习图像模版。我们可以将权重可视化为模版，来观察卷积核。这些卷积核的样子看起来有点奇怪。出现了紫色和蓝色的点，这是因为提取的是零中心化后的图像，零中心化后的图片的颜色不再是绿色和黄色。这些特征的大小为7\*7，一种可能的解释是网络提取到了人类无法观察到的微小特征。这些特征可能是柑桔病变处颜色的分布某种规律。

5实验总结与展望

5.1实验总结

为了解决我国农业病虫害防治不准确，不及时的问题，缓解农作物病虫害从业者匮乏的情况，早日实现计算机辅助诊断农业病虫害。本论文探索了深度学习技术在农业农作物病虫害的应用的效果，针对常见的、危害性严重的柑桔疮痂病的检测做了具体的分析研究。本文的主要成果有：

1. 将目标检测领域的先进算法应用到农业病虫害检测领域。

（2） 针对柑桔疮痂病病状检测的具体研究，根据疮痂病状的特点设计了检测网络的结构。并且融合当前最先进的神经网络训练技术调整模型，如权重初始化方法、弃权技术、参数更新技术等多种方法从多个角度减轻网络在较小数据集上的过拟合问题，与提高检测模型的对于小病状识别的准确度。最终在识别柑桔疮痂病上达到较好的结果。

5.2展望

目前，论文对目标检测在农业病害识别的领域只做了很小一部分工作。只是探索了对于柑桔疮痂病一种常见农业病虫害的探索。计算机辅助诊断对于促进我国农业的现代化转变有重大的意义，这是人工智能应用在农业领域的具体方式。但由于作者实验环境、研究水平、时间不足等因素，目前这个课题仍然还有很多不足，有待进一步的研究解决，主要包括以下几个方面：

1. 网络的层次不够深。

多层的神经网络是一个分级的结构，尤其适用于学习分级的知识。底层网络能够理解简单的像素，深层网络可以理解越来越复杂的概念。比如从检测边缘信息到简单的集合形状。网络层次不够深是导致检测精度不高重要因素，但越深的神经网络对实验环境要求越高。

1. 数据集太少

农业病虫害领域几乎找不到公开的数据集，对比医学影像庞大且种类繁多的数据集有明显的差距。这是因为农业领域不是人工智能研究中急需应用的领域，归根结底是对农业现代化发展不够重视，农业领域吸引不到研究人员的兴趣。作者根据网上现有的低像素的柑桔疮痂病病例图片，所收集的数据集太小且清晰度不高、可信度不足，是导致检测网络不高的根本原因。所以应该鼓励农作物种植者广泛的收集农业病例图像，建立完善丰富的病虫害数据集。

1. 检测算法的不足

作者对于农业病害知识的匮乏，只是在疮痂病病例的图像做出的直观认识，不能更深入了解柑桔疮痂病的原理，科学的设计网络模型，最大化的提高检测精度。

以上是未来需要努力的方向。除了改进已有的算法，探索无监督学习或半监督学习在目标检测上的应用是非常重要的，因为现实中大量的数据是未标注的，如果只是使用已标记的数据进行有监督的学习，可用的数据量将受到极大的限制。

谢 辞

本文是在指导教师的悉心指导下完成的，从论文选题、课题调研、试验指导、理论分析到论文撰写，无不倾注了老师的心血和汗水。向所有曾经关心和帮助过我的老师、同学和朋友致以诚挚的谢意！

参考文献

1. [1] 李加成.我国农业病虫害防治现状及建议分析[J].种子科技,2017,35(06):106+108. [↑](#endnote-ref-1)
2. [2] Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In Advances in Neural Information Processing Systems 25, 2012. [↑](#endnote-ref-2)
3. [3] N Dalal, B Triggs. Histograms of Oriented Gradients for Human Detection. IEEE Computer Society Conference on Computer Vision & Pattern Recognition, 2005, 1(12):886-893. [↑](#endnote-ref-3)
4. [4] D. G. Lowe. Distinctive image features from scale-invariant key points. International journal of computer vision, 60(2), 91-110, 2014. [↑](#endnote-ref-4)
5. [5] P Viola, MJ Jones. Robust Real-time Object Detection. International Journal of Computer Vision, 2001, 57(2):87. [↑](#endnote-ref-5)
6. [6] P. H. Chen, C. J. Lin, and B. Schölkopf, A tutorial on ν-support vector machines, Appl. Stoch. Models. Bus. Ind. 2005, 21, 111-136.

   [7] 侯欣.中国柑橘疮痂病菌的种类和变异研究[D].浙江大学,2013.

   [8] （美）lan Goodfellow Yoshua Bengio Aaron Courville著,‎ 赵申剑 黎彧君 符天凡 ‎ 李凯等译.深度学习[M].北京：人民邮电出版社，2017.

   [9]Ross Girshick, Jeff Donahue, Trevor Darrell, et al. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation[J]. 2013:580-587.

   [10]He K, Zhang X, Ren S, et al. Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition[J]. IEEE Transactions on Pattern Analysis & Machine Intelligence, 2015, 37(9):1904.

   [11] Girshick R. Fast R-CNN[C]// IEEE International Conference on Computer Vision. IEEE Computer Society, 2015:1440-1448.

   [12] Ren S, He K, Girshick R, et al. Faster R-CNN: towards real-time object detection with region proposal networks[C]// International Conference on Neural Information Processing Systems. MIT Press, 2015:91-99. [↑](#endnote-ref-6)