

分类号 _____ 密级 _____

UDC _____



本科毕业设计

棉花图像自动分级系统

学生姓名 张雪冰 学号 15020032030

指导教师 董军宇

院、系、中心 信息科学与工程学院

专业年级 保密管理 2015 级

文答辩日期 2019 年 6 月 6 日

中国海洋大学

棉花图像自动分级系统

摘 要

棉花图像自动分级系统的实质是分类器，本设计使用深度学习技术模拟人类视觉系统，训练卷积神经网络 MobileNet 模型，并将模型移植到 android 移动端，使用 android-studio 开发工具构建棉花图像自动分级系统。

深度学习通过组合简单、非线性模块自动提取不同层次的特征信息，解决了传统机器学习手工构造特征、耗费人力、物力的难题，其中 CNN 卷积神经网络算法在图像识别领域表现优异，因此成为深度学习代表算法之一。

MobileNet 是 google 为适用移动端与嵌入端而提出的小型卷积神经网络，其深度和复杂度相比其他卷积神经网络大幅下降，由于复杂度与参数的大幅度减小，计算速度因此大幅上升，且分类精度可以满足多数场景的要求。

迁移学习即将在一个问题上训练好的模型保留一定的层数，冻结为瓶颈层，瓶颈层的输出节点可以被作为任何一个图像抽象的、凝练性表达的特征向量，输入此特征向量，重新调整训练剩余网络层以适用新问题。本设计考虑到棉花数据集规模较小，重训练模型准确率损失较大，因此使用迁移学习训练模型并对比分析冻结不同网络层的分类结果，获得分类精度最高的模型。

关键字：深度学习；MobileNet；迁移学习

The automatic grading system of cotton image

Abstract

The essence of cotton image automatic grading system is classifier. This design uses deep learning technology to simulate human visual system, which extracts cotton image feature for training and learning through convolution network to construct cotton image automatic grading system.

Deep learning trains neural networks by automatically extracting different levels of feature information, and solves the bottleneck of traditional machine learning manual construction features, labor and material resources. The improvements of convolutional neural networks have promoted the breakthrough of image recognition in the field of artificial intelligence, and thus become one of the deep learning representative algorithms.

MobileNet is a small convolutional neural network proposed by Google for mobile and embedded. Its depth and complexity has been greatly reduced compared with other convolutional neural networks. Due to the reduction of parameters, the calculation speed is greatly increased, and the classification accuracy can also satisfy most scenarios requirements.

Migration learning will retain a certain number of layers on a problem-trained model, and freeze it as a bottleneck layer. The output node of the bottleneck layer can be used as a feature vector for any image concise expression. Entering this feature vector and re-adjusting the remaining layers of the training to apply new problem. The Graduation design considers that the scale of cotton dataset is small, so that the precision loss of the heavy training model can be higher. Therefore, this design uses the migration learning to freeze different network layers and analyze classification accuracy, which can obtain the highest accuracy model.

Keywords: Deep learning; MobileNet; Migration learning

目 录

摘 要.....	I
Abstract.....	II
1 绪论.....	1
1.1 研究背景及意义.....	1
1.2 国内外研究现状.....	2
1.2.1 棉花品级研究现状.....	2
1.2.2 深度学习发展现状.....	3
1.3 研究内容.....	4
1.4 创新.....	4
2 理论基础.....	5
2.1 神经网络概述.....	5
2.1.1 前向传播.....	5
2.1.2 损失函数.....	5
2.1.3 优化算法.....	5
2.1.4 训练过程.....	7
2.2 卷积神经网络结构.....	8
2.2.1 网络结构对比.....	8
2.2.2 卷积层.....	9
2.2.3 池化层.....	11
2.3 迁移学习.....	12
3 MobileNet	13
3.1 深度可分离卷积介绍.....	13
3.2 网络结构.....	15
3.2.1 标准卷积与深度可分离卷积结构对比.....	15
3.2.2 MobileNet 完整结构	15
3.2.3 MobileNet 瘦身	16

4	棉花自动分级系统实验.....	17
4.1	模型训练.....	17
4.1.1	数据处理.....	17
4.1.2	模型配置.....	21
4.2	移动端配置.....	24
4.2.1	开发工具.....	24
4.2.2	ckpt 文件转换 pb 文件.....	24
4.2.3	实时检测.....	25
4.2.4	图片分类.....	26
4.2.5	Bitmap 图像转换.....	28
4.2.6	Tensorflow Mobile 接口.....	28
4.3	实验结果与分析.....	30
5	总结.....	33
5.1	完成的工作.....	33
5.2	设计的不足.....	33
	参考文献.....	34
	致谢.....	36

1 绪论

1.1 研究背景及意义

战略物资一旦因贸易摩擦、资源匮乏、环境污染等原因缺乏时，无法得到及时供给或无其他替代物资，将会严重危害国家经济安全。近年来，我国实现经济高速的发展同时逐渐转型为绿色环保型可持续发展方式，这一举动将战略物资储备推向新高度^[1]。棉花无可替代的经济价值决定其成为我国进出口贸易的重要商品，棉花的品级直接影响交易定价，与国家利益密切相关，棉花也因此成为国家实施宏观调控的重要参考。

现阶段中国的棉花品级检验主要以“感官检验”为主，辅助“仪器质检”。“感官检验”要求在符合要求的光照条件下，检验人员凭借经验，使用触觉、视觉甚至嗅觉综合考察棉花的成熟程度、色泽特征和轧工质量评定棉花品级。“感官检验”中影响棉花品级的因素各种各样，例如光照影响、检验人员状态、检验人员的经验等，这都会给棉花品级评定造成不同程度的误差；“仪器质检”主要依托于国外研发的 HVI、红外线检测设备等评测棉花的颜色、纹理特征等各种内在质量指标^[2]，但该设备要求检测环境必须达到标准的温度和湿度，且设备对棉花色特征检测结果与人工定级分类结果存在差异，行内争议较大。因此，如何制定有效的棉花定级策略一直是行业研究重点。

随着计算机技术交叉学科的逐渐开展，计算机在相关研究领域取得一定研究成果，并初步适用于指导农作物种植、收获，例如通过分析农作物生长的监测画面，不断调整其种植环境，提高作物收成与质量，这些技术的不断应用与积累为从图像角度研究棉花定级奠定了基础^[2]。深度学习由于其出色的表现成为人工智能领域强大的工具，而卷积神经网络作为代表算法之一，其高效性则推动了深度学习在图像应用领域取得突出成就。与此同时，为了解决卷积神经网络深度增加，效率逐渐下降的问题，让卷积神经网络走出实验室，更广泛的应用于移动端，轻量化模型设计又逐渐走进人们的视野，进一步推动了深度学习在移动端的应用。

本设计基于深度学习技术，训练神经网络实现自动化棉花分级，并将模型部署在移动终端，最终实现棉花图像分级错误率缩小在 2% 以内，且本方法经济便捷，为神经网络适用到棉花定级提供了有效参考。

1.2 国内外研究现状

1.2.1 棉花品级研究现状

棉纤维测试技术以瑞士乌斯特技术公司的 HVI 单元为代表,美国 Motion 公司与印度 Premier 公司相继崛起,逐渐成为其竞争对手,成为棉纤维测试市场的三大巨头^[4]。自 1991 年以来,美国已规定应使用最先进的 HVI 设备来测试美国棉花的质量,近年又开发了近红外设备测试等其他指标,其中 AFIS 通过测量出单根纤维的成熟度和直径,使得棉花和棉纺织厂以此参考调整和优化每台机器的工作条件,经济地生产出高质量的棉纺织品^[5]。美国棉花分级体系完备,本设计分级标准即参考美国陆地棉中的白棉的七个等级: gm(Good Middling)、sm(Strict Middling)、m(Middling)、slm(Strict Low Middling)、lm(Low Middling)、sgo(Strict Good Ordinary)、go(Good Ordinary)。

现阶段,计算机图像处理技术在人工智能领域的广泛应用为从图像角度出发研究棉花品级分类奠定了基础。美国研究人员曾将神经网络应用到棉花分类,最终将 HVI 设备与人工定级的误差由 54.08%降到了 16.35%^[6]; 山东出入境检验检疫局通过人工构造棉花图像特征,结合距离度量学习、机器学习技术训练分类器,误差率最小达到 1.32%^[7]; 北京工业大学信息部基于深度学习,结合图像分割、分布检测、目标检测、AlexNet 图像分类等多种方法学习不同时期棉花的图像特征,建立了棉花发育期自动识别系统,其在 2016 与 2017 年份的测试集上的准确率分别达到 80.52%和 75.48%,为探究气象与棉花发育期之间的联系提供参考数据^[8]; 国外学者结合神经网络与模糊逻辑推理系统,用 HVI 光纤测试结果训练神经模糊推理系统,系统的预测能力优于神经网络^[9]。

综上所述,从最初的质检仪器到神经网络的应用,棉花分级一直是各个国家的行业研究重点。随着图像识别在 AI 人工智能领域的不断发展,国内外逐渐探索使用现代技术进行棉花分级,均取得较好的研究成果。以上研究成果表明,从图像角度利用神经网络进行棉花定级具有可实施性,同时表明深度学习与图像处理技术结合将会在农业领域发挥巨大作用。

1.2.2 深度学习发展现状

特征提取是促进人工智能飞速发展的核心。深度学习通过组合多层简单但非线性的模块逐一提取输入特征，解决了传统的机器学习算法需要人工构造特征，阻碍技术发展的瓶颈问题。深度学习通过特征提取，将输入转换为更高、更抽象的层面，表示出数据的分布特征并用于设计分类功能，从而被越来越多的程序应用到图像识别、语音转录、模式识别等领域。图 1-1 显示了传统机器学习与深度学习流程区别。

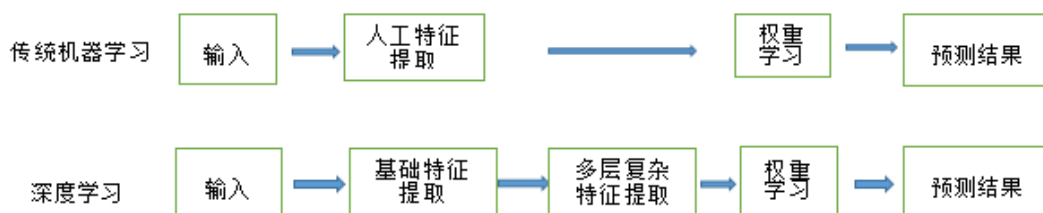


图 1-1 传统机器学习与深度学习流程对比

笔者在“中国知网”论文查询平台输入关键字“卷积神经网络”，共找到 9043 条结果，例如：基于多层特征深度融合的 AlexNet 卷积神经网络人脸识别，网络模型识别率达到 98.7%^[10]；基于 VGGNet 深度卷积神经网络将语义分割精度提高至 71.3% mIOU^[11]。检索结果表明卷积神经网络在多种领域表现优异并且取得突出成就。

深度学习包括多种学习工具。其中，TensorFlow 作为一款开源工具，内部实现多种机器学习算法，包括前向传播、模型优化等算法，并且不因平台系统差异改变内部实现原理，对外提供执行算法接口，方便开发者灵活调用。资料表明，TensorFlow 已被广泛用于无人驾驶、智能家居、机器人工厂作业、医院药物分类与检索等十几个领域^[12]，在各大网络交流平台与谷歌内部表现突出，是人工智能领域工程师使用的重点工具^[13]。笔者在搜索引擎上输入“深度学习工具”，TensorFlow 占据榜首，各种项目资源也多使用 TensorFlow 部署深度学习项目。

事实证明，深度学习非常适合处理维度高、结构复杂、特征提取困难的数据问题，适用但不限于人文、地理、自然、科学领域，刷新了传统机器学习方法在图像识别和语音识别中的记录，并在医学、物理、基因研究、人体神经网络等交叉领域取得优异成绩，为计算机交叉学科的发展奠定基础^[14]。

1.3 研究内容

本设计参考美国陆地棉白棉的分级标准，训练卷积神经网络 MobileNet，并探讨训练卷积神经网络不同瓶颈层、数据集均衡分布等因素对模型分类准确度的影响，经过多次对比分析获取准确率最高的模型，并用该模型进行 android 移动端的部署配置，设计棉花图像自动分级系统。

第一章 介绍该课题的背景、研究内容及创新点，并从棉花品级分类与深度学习发展现状两个角度探讨该课题的国内外研究现状，为该设计的实施提供理论基础与技术支持。

第二章 介绍神经网络与 CNN 卷积网络理论知识，并对 CNN 基本架构、实现原理详细阐述，最后对本设计得以实施的重要技术“迁移学习”进行介绍。

第三章 介绍 MobileNet 的核心技术深度可分离卷积，着重对比深度可分离卷积与传统标准卷积的差异与优势，最后介绍 MobileNet 实现精简化的两个重要参数。

第四章 记录棉花自动分级系统实现过程，包括数据集的处理、模型的训练参数配置及特征图的输出、移动开发工具介绍、模型文件的适配等过程，最后对比分析不同条件下训练所得模型分级准确率，寻找分级效果最好的模型。

第五章 总结本设计实现的主要工作与完成过程存在的不足。

1.4 创新

1. 本文使用卷积神经网络，解决了传统的感官检验方法受多种因素影响，误差较大的难题，并将分级错误率控制在 2% 以内，具有一定的实施价值。
2. HVI 使用色度仪来测评棉花的反射率、黄度属性，本文仅从图像角度出发，给出评定棉花品级的参考方法，更加经济便捷。
3. 传统的方法耗费人力物力，不易实施，本文提供的方法仅需一部移动设备与相机便可完成。

2 理论基础

2.1 神经网络概述

2.1.1 前向传播

定义网络结构，将原始数据经过层层推导转换为具有分布特征的输出向量，即神经网络的前向传播过程。该过程在已知神经网络的输入、结构、各层结构之间的权重值的前提下即可计算出神经网络的前向传播结果。

2.1.2 损失函数

分类问题中常用交叉熵作为损失函数，用于描述神经网络计算输出和理想值的接近程度。对于 N 分类问题，输入数据经过神经网络计算，得到 N 维数组的输出向量，数组中每一个节点数值代表此图像属于该类的概率，在理想情况下输出类别所对应节点的值为 1，其余节点的值为 0。例如，对于 5 分类问题，输出结果越接近 $[0,0,0,1,0]$ 、 $[1,0,0,0,0]$ 、 $[0,1,0,0,0]$ 、 $[0,0,1,0,0]$ 、 $[0,0,0,0,1]$ 之一，则表明该模型的分类效果越好。神经网络在训练过程中通过更新参数，逐渐降低交叉熵损失值，提高模型的准确率。

2.1.3 优化算法

1. 反向传播算法

反向传播过程通过更新神经网络模型的参数，逐渐降低训练损失、逼近待分类数据特征，提高模型识别准确率。反向传播更新参数使用梯度下降算法降低损失值，假设 a 为神经网络中的参数， $F(a)$ 表示该参数的前向传播损失值，优化过程即为寻找合适的 a 值，使得 $F(a)$ 最小。图 2-1 为梯度下降示意图。

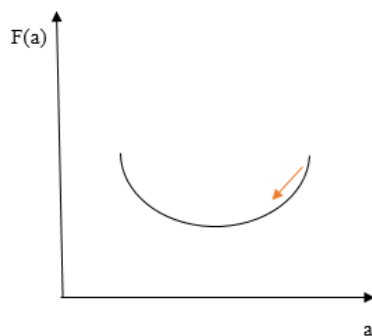


图 2-1 梯度下降示意图

图中 x 轴表示参数 a 的取值范围，y 轴表示 F(a) 的取值，梯度下降算法即将 a 沿着 x 轴向左侧移动，从而使 F(a) 沿着图中箭头的方向逐渐下降到最低点。参数的梯度即为求偏导的过程，a 参数的梯度即为 $F'(a)$ 。若 m 代表下降幅度，则参数更新的公式为：

$$a_{n+1} = a_n - mF'(a)$$

神经网络的优化过程的第一阶段是得出计算值，得到计算值和真实值的差距；第二阶段为更新参数，降低损失。实际训练过程，一方面需要考虑训练速度，另一方面受电脑内存限制，因此不可能使用所有训练数据的偏差值计算损失函数。通常每次只计算被称为一个 batch 的训练数据上的损失函数，并通过多轮迭代，逐渐优化，这样不仅可以加快训练速度，最终的收敛状态也会非常接近优化完整训练数据上的损失函数的效果。

2. 学习率的设置

学习率决定模型迭代更新的速度，如果学习率过大，可能会造成模型的损失值震荡无法收敛；设置过小，参数则更新缓慢，降低模型的训练速度。为了避免上述问题，通常在训练过程中使用指数衰减法迭代更新学习率，训练初期先使用较大学习率使得损失函数快速收敛到较优解，随着训练轮数增加，逐步降低学习率，减小参数更新幅度，缓慢寻找最优解，避免参数在极优值附近徘徊。当前使用的学习率由初始设置学习率、衰减系数、训练轮数、衰减速度决定，其中衰减速度代表完整使用一遍训练数据需要的迭代轮数，即数据总数/batch。

指数衰减法公式为：

$$\text{Decay_learning_rat} = \text{Learning_rate} * \text{Decay_rate}^{(\text{Global_step}/\text{Decay_steps})}$$

3. 过拟合问题

过拟合问题即网络参数过多而导致模型过于复杂，此时模型能很好的记忆训练数据的分布特征，从而表现为训练数据识别准确度高，测试数据准确度异常低，失去对未知数据的判断能力。为了防止此类问题，通常在损失函数中添加刻画模型复杂度的正则化损失，增大训练数据中的噪音。

4. 滑动平均模型

滑动平均模型通过使用变量的平均值控制模型更新的速度，提高模型在测试数据上的健壮性。

2.1.4 训练过程

模型训练时随机选择 batch 数据训练神经网络，优化损失函数提高模型的准确率，根据训练目标、迭代轮数决定是否结束训练。图 2-2 显示了神经网络完整的训练过程。

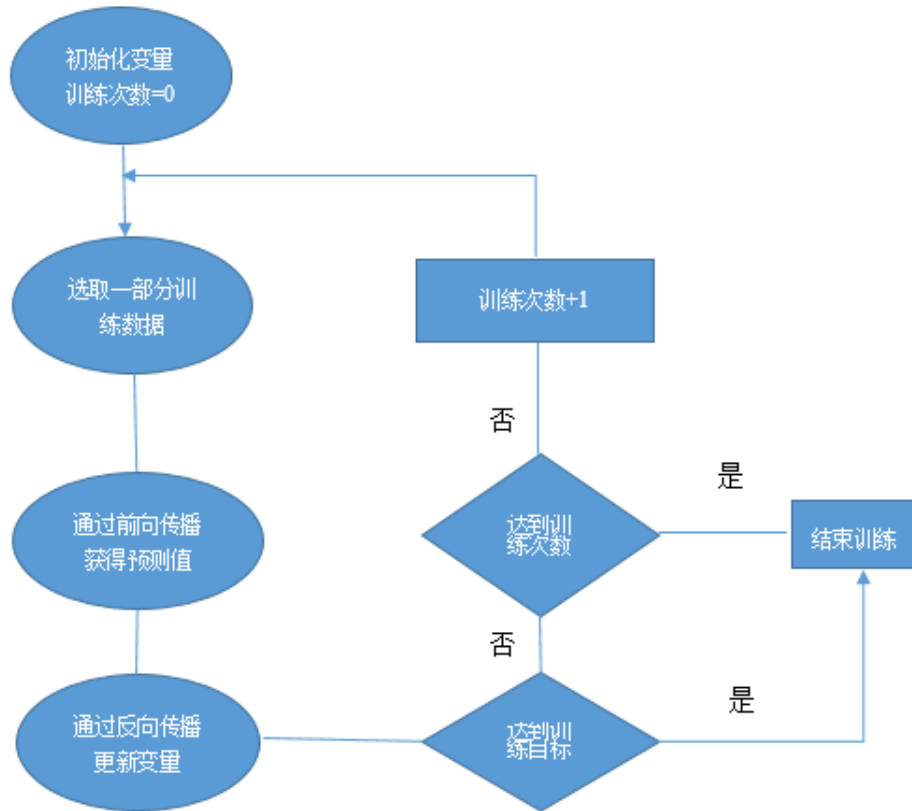


图 2-2 神经网络训练过程

2.2 卷积神经网络结构

卷积神经网络结构包括输入层、卷积层、池化层，全连接层、softmax 输出层。图像识别网络输入层是用于表示图像数据三维矩阵，矩阵三个维度大小分别代表图像的长、宽、色彩通道，例如图像矩阵大小为 $500 \times 500 \times 3$ ，表明该图像长、宽为 500，色彩通道数为 3；卷积层提取图像的特征，增加图像深度，经过多层卷积，图像的底层信息将被提取，形成更加抽象、凝练性图像表达；池化层的作用是通过降低图像的分辨率来减少最后全连接层参数的大小；全连接层利用提取后的特征向量完成分类任务；softmax 层则将全连接层输出转换为概率分布，用于计算交叉熵损失值^[15]。

2.2.1 网络结构对比

神经网络结构由输入层，若干隐藏层，softmax 输出层组成，输入节点经过一系列隐藏层的 ReLu 非线性变化，提取出更加抽象、凝练的表达作为输入的特征向量，并经输出层表示数据的分布特征，训练分类器。传统的神经网络结构任意一层的节点均与下一层的所有节点连接，通常称为全连接神经网络，而卷积神经网络的任意两层之间只有部分节点连接。为了便于显示连接结构，一般将每一层的神经元组成一列显示。图 2-3 与 2-4 显示两种结构的差异。

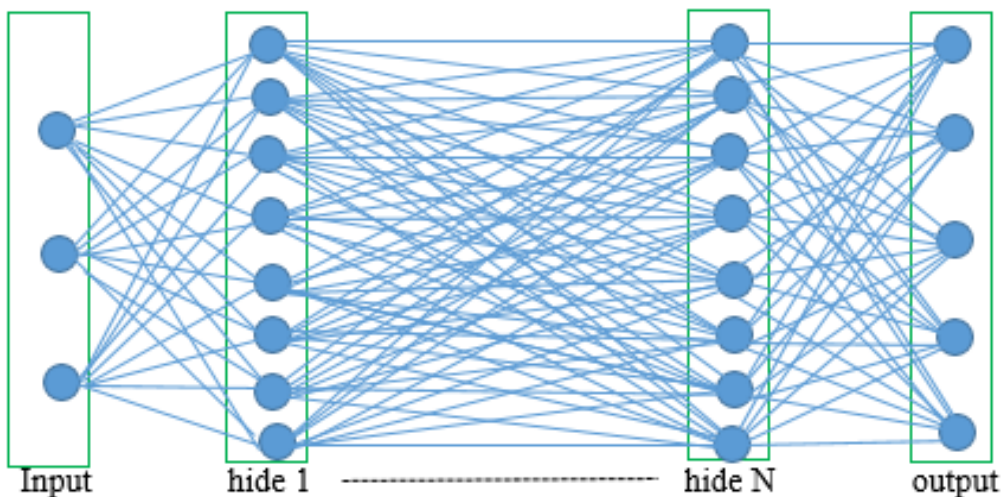


图 2-3 传统全连接层网络结构

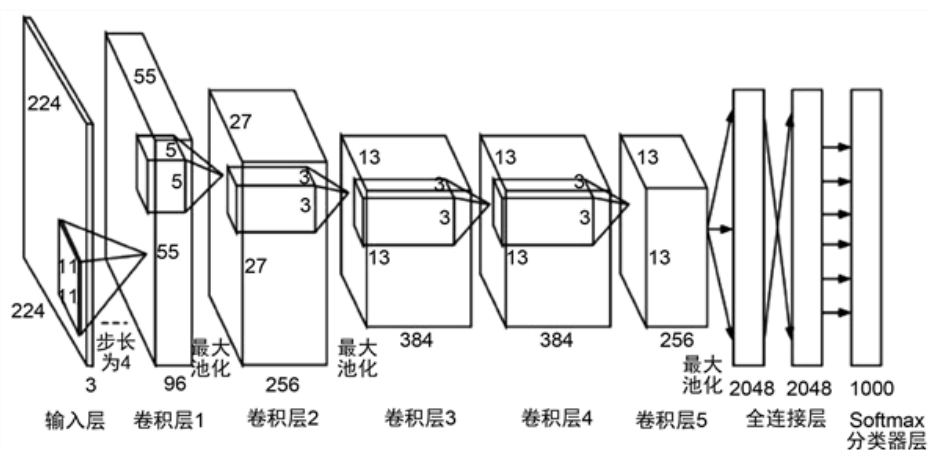


图 2-4 卷积神经网络结构

由对比图可知，传统神经网络任意两层节点处处相连，每条边分别代表可更新权重，形成全连接层。在输入节点较少的前提下，此时网络参数已经较多，然而训练图像数据大小节点多至几千，这将会直接降低模型的计算速度。卷积神经网络任意两层相邻边的数量大大减少，提高模型计算速度的同时，也可以有效防止模型过拟合训练数据而无法预测未知数据。

卷积神经网络相比传统网络，参数规模虽然大大减小，但依旧遵循神经网络功能部署整体架构，基于此优势，卷积神经网络在图像识别问题上表现更为突出。

2.2.2 卷积层

卷积层通常被称为过滤器或者内核，本文将卷积层使用的矩阵称作卷积核。卷积核将待处理图像的子节点转换为深度不限、长宽均为 1 的单位节点矩阵。常将卷积核的尺寸描述为 $\text{length} \times \text{width} \times \text{depth}$ ， $\text{length} \times \text{width}$ 为卷积核将要处理子节点矩阵的大小，通常为 3×3 或 5×5 ， depth 是卷积核将子节点经过卷积后输出节点矩阵的深度，即该层卷积核的数量，每个卷积核的通道数与待处理节点矩阵通道数保持一致，不加特殊指定。图 2-5 为 $32 \times 32 \times 3$ 图像经过 $3 \times 3 \times 5$ 卷积示意图：

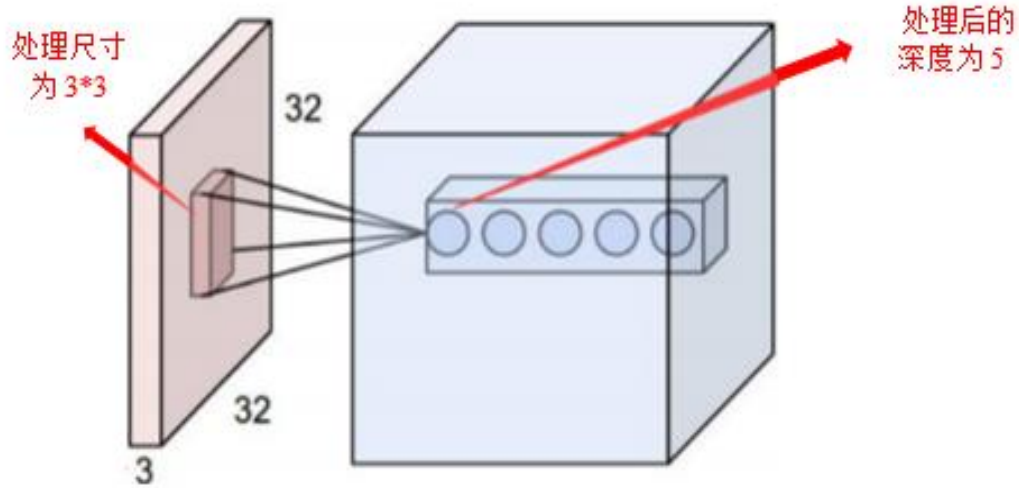


图 2-5 卷积过程

卷积核将一系列子节点进行卷积计算，将左侧矩阵转换为右侧矩阵。本文将结合具体样例，将 $4 \times 4 \times 3$ 矩阵经过 $3 \times 3 \times 5$ 的卷积核转变为 $2 \times 2 \times 5$ 矩阵。图 2-6 为不同通道的卷积过程。

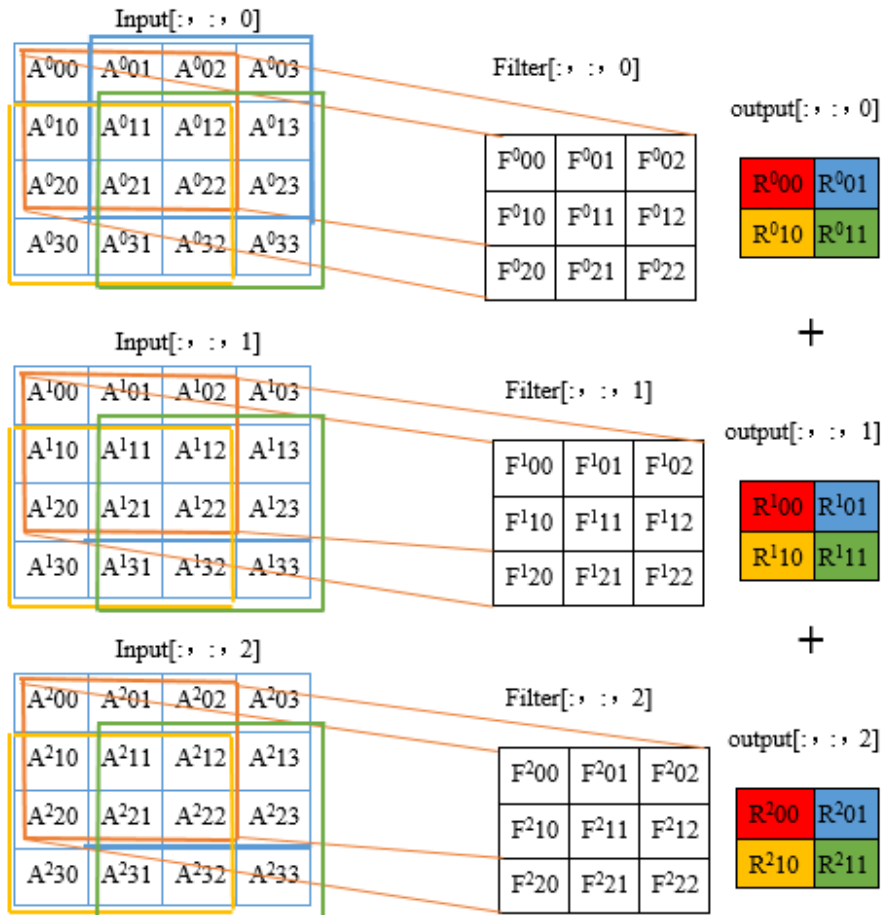


图 2-6 不同通道的卷积

图像三个通道上的卷积示意图中，每个颜色代表将要处理的子节点矩阵，卷积核遍历当前矩阵，旋转 180 度分别与子节点矩阵对应像素点相乘并相加得到运算结果，最后将所有通道上的输出矩阵相加得到深度为 1 的节点矩阵。剩余四个卷积核按照同样的操作对当前矩阵进行卷积操作，则得到深度为 5 的单位矩阵。 $4*4*3$ 矩阵经过卷积前向传播过程变为 $2*2*5$ 的矩阵，尺寸缩小一倍，实际应用中可以通过设置卷积核移动步长或者在当前矩阵的边界进行全 0 填充避免处理节点尺寸变化过大。

2.2.3 池化层

卷积层处理后节点矩阵经过池化层缩小尺寸，降低全连接层的参数。与卷积层中卷积核的设置类似，池化层也需要人工设定尺寸、是否使用全 0 填充、移动的步长等。池化层通常采用最大值或平均值运算。图 2-7、2-8 为池化层示意图。

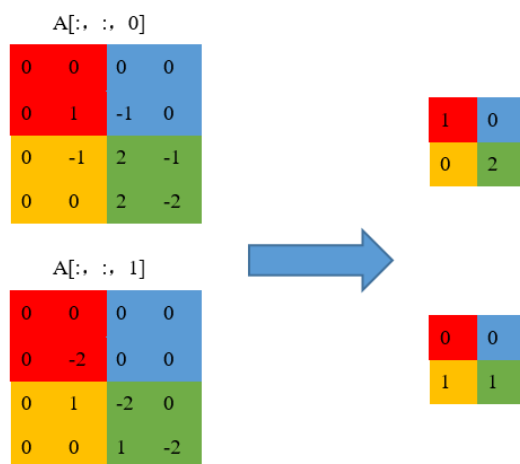


图 2-7 $3*3*2$ 节点矩阵经过全 0 填充且步长为 2 的最大池化层过程

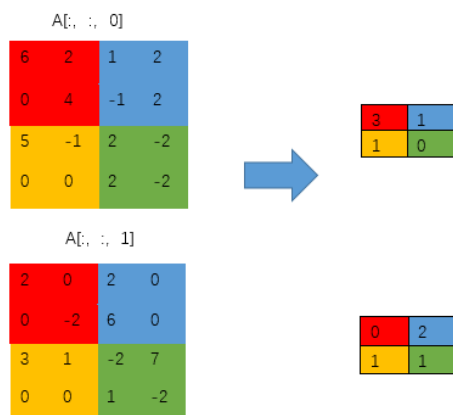


图 2-8 $4*4*2$ 节点矩阵经过步长为 2 的平均池化层过程

2.3 迁移学习

当训练样本数据规模较小时,可以使用迁移学习将在一个问题上训练好的模型保留一定的层数上的参数值,冻结为瓶颈层,瓶颈层的输出可以被作为任何一个图像底层信息特征,输入此特征向量,重新调整训练剩余层上的参数。相比重新训练完整模型,使用迁移学习大幅度减少训练时间的同时,模型精度损失较小,可以满足大多数场景的需求。

3 MobileNet

卷积神经网络自推广以来，多次在计算机视觉领域挑战成功，主流趋势推动卷积网络沿着更加深入、复杂的方向发展以获得更高的准确率，网络规模逐渐增加的同时，速度与效率逐渐降低。然而，一些例如人脸识别、目标检测等应用场景需要实时响应计算结果，相关学者基于此现状提出建立小型、低延迟模型轻松满足移动和嵌入式视觉应用设计要求^[15]的网络架构-----MobileNet。MobileNet 核心技术为深度可分离卷积，本文将对此种卷积模式做出具体介绍。

3.1 深度可分离卷积介绍

深度可分离卷积将标准卷积分解为深度卷积与逐点卷积两种方式。本文以输入图像(A_I, A_I, D)，卷积核(A_F, A_F, D, N)，输出特征为(A_O, A_O, N)为例， A_I 表示输入图像的维度， D 表示图像通道数目， A_F 表示卷积核的维度， N 为输出图像深度， A_O 为输出特征的维度。图 3-1 为标准卷积核，

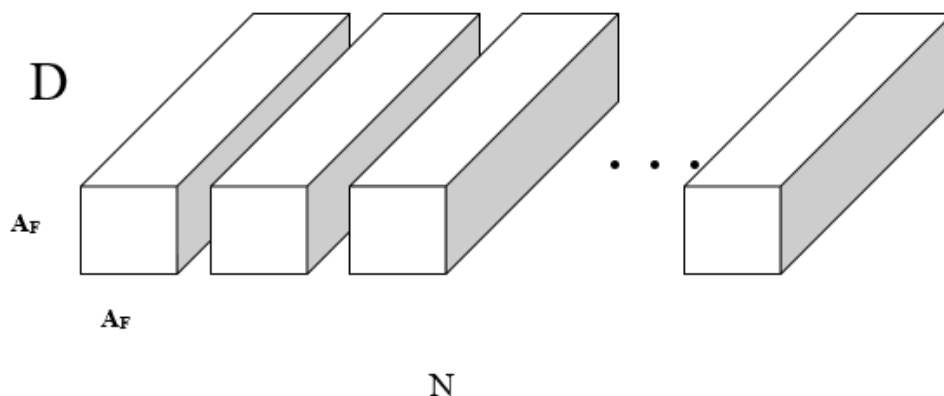


图 3-1 标准卷积核

对于输入通道数为 D ，经过深度为 N 标准卷积核的参数量为： $D*N*A_F*A_F$ 。

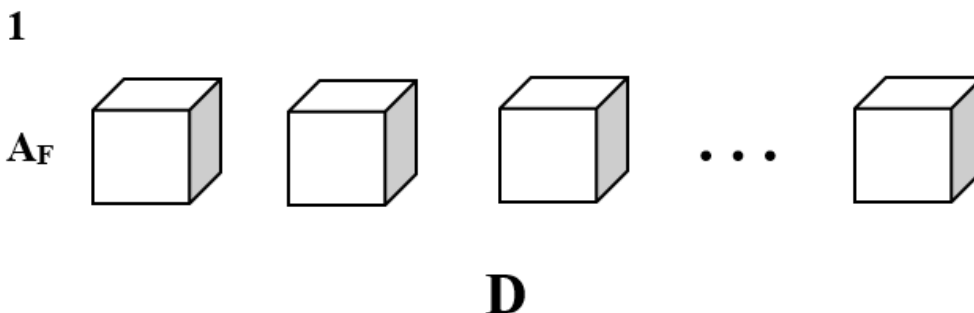


图 3-2 深度卷积

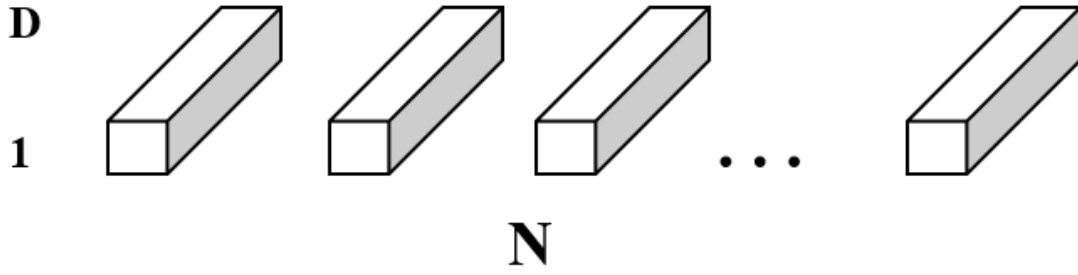


图 3-3 逐点卷积

深度卷积核的尺寸为 $(A_F, A_F, 1, D)$ ，负责提取特征，如图 3-2，参数量为 $1 * D * A_F * A_F$ ；逐点卷积的尺寸为 $(1, 1, D, N)$ ，负责转换通道，如图 3-3，参数量为 $1 * 1 * D * N$ 。其中标准卷积中将输入的第 d 通道对应的第 n 卷积核的 d 通道，并将通道计算结果相加，计算出第 n 卷积核的输出结果；而深度卷积中将输入的第 d 通道作用第 d 卷积核，产生输出特征的第 d 通道。深度可分离卷积的参数量变为 $1 * D * A_F * A_F + 1 * 1 * D * N$ 。

深度可分离卷积与标准卷积的参数量比例为：

$$\frac{1 * D * A_F * A_F + 1 * 1 * D * N}{D * N * A_F * A_F} = \frac{1}{N} + \frac{1}{A_F * A_F}$$

例如 $N=5$ ， $A_F=4$ 时，参数之比为 $\frac{1}{5} + \frac{1}{4 * 4} = \frac{21}{80}$ ； $N=100$ ， $A_F=4$ 时，参数之比为 $\frac{1}{100} + \frac{1}{4 * 4} = \frac{29}{400}$ 。公式可知，当卷积核尺寸、输出深度增大时，深度可分离卷积相比标准卷积参数量大幅度下降，模型计算速度大幅上升。

3.2 网络结构

3.2.1 标准卷积与深度可分离卷积结构对比

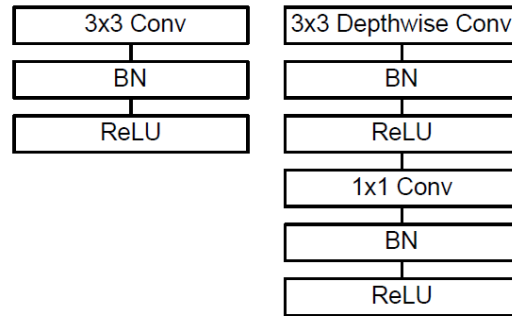


图 3-4 标准卷积与深度可分离卷积

图 3-4 左代表完整的标准卷积结构，图 3-4 右表示深度可分离卷积结构。BN 表示 Batch Normalization 层用于优化模型的训练，ReLU 表示激活层处理应用中的非线性问题。

3.2.2 MobileNet 完整结构

Type / Stride	Filter Shape	Input Size
Conv / s2	$3 \times 3 \times 3 \times 32$	$224 \times 224 \times 3$
Conv dw / s1	$3 \times 3 \times 32 \text{ dw}$	$112 \times 112 \times 32$
Conv / s1	$1 \times 1 \times 32 \times 64$	$112 \times 112 \times 32$
Conv dw / s2	$3 \times 3 \times 64 \text{ dw}$	$112 \times 112 \times 64$
Conv / s1	$1 \times 1 \times 64 \times 128$	$56 \times 56 \times 64$
Conv dw / s1	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 128$	$56 \times 56 \times 128$
Conv dw / s2	$3 \times 3 \times 128 \text{ dw}$	$56 \times 56 \times 128$
Conv / s1	$1 \times 1 \times 128 \times 256$	$28 \times 28 \times 128$
Conv dw / s1	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 256$	$28 \times 28 \times 256$
Conv dw / s2	$3 \times 3 \times 256 \text{ dw}$	$28 \times 28 \times 256$
Conv / s1	$1 \times 1 \times 256 \times 512$	$14 \times 14 \times 256$
5×	Conv dw / s1	$3 \times 3 \times 512 \text{ dw}$
	Conv / s1	$1 \times 1 \times 512 \times 512$
	Conv dw / s2	$3 \times 3 \times 512 \text{ dw}$
	Conv / s1	$1 \times 1 \times 512 \times 1024$
	Conv dw / s2	$3 \times 3 \times 1024 \text{ dw}$
	Conv / s1	$1 \times 1 \times 1024 \times 1024$
	Avg Pool / s1	Pool 7×7
	FC / s1	1024×1000
	Softmax / s1	Classifier

图 3-5 MobileNet 网络结构

图 3-6 MobileNet 用于解决 1000 分类问题，对于其他分类问题只需替换 classifier 的分类数量即可。结构中 dw 表示深度卷积，Sn 表示卷积核移动的步长 stride，原始图像经一层 $3 \times 3 \times 3 \times 32$ 的标准卷积与若干数层深度可分离卷积层提取特征后，采用 average pooling 平均值操作降维，输入到全连接层 FC，经 Softmax 激活层输出预测值。分析其结构可知，mobileNet 的参数和计算量主要集中在卷积核数量较多的 1×1 逐点卷积层，其次是任意两层均有边相连的全连接层。

表 3-1 mobileNet 计算量分布

Type	Mult-Adds	Parameters
Conv 1×1	94.86%	74.59%
Conv DW 3×3	3.06%	1.06%
Conv 3×3	1.19%	0.02%
Fully Connected	0.18%	24.33%

由于底层优化， 1×1 卷积运算可以迅速实现。因此，相对其他卷积神经网络，MobileNet 准确率虽然略微下降，但其计算量和参数占据绝对优势。这种在准确率与延迟率之间做了折中的网络，是要求进行实时响应计算结果的应用场合的首选。

3.2.3 MobileNet 瘦身

MobileNet 的标准模型可以适用大部分实时性要求较高的应用场景，但对某些要求更高的场景，便有必要对 MobileNet 进一步精简化。

1. width multiplier

width multiplier，记为 a ，用于比例压缩网络，取值范围为 $(0,1]$ ，压缩网络后的深度可分离卷积的计算量变为 $A_I \cdot A_I \cdot a \cdot D \cdot A_F \cdot A_F + a \cdot D \cdot a \cdot N \cdot A_F \cdot A_F$ 。

2. resolution multiplier

resolution multiplier，记为 b ，用于按照比例缩小图像的分辨率，取值范围为 $(0,1]$ ，减少分辨率后的深度可分离卷积的计算量为 $b A_I \cdot b A_I \cdot a \cdot D \cdot A_F \cdot A_F + a \cdot D \cdot a \cdot N \cdot b A_I \cdot b A_I$ 。该因子的引用仅会影响模型的计算量，并不会降低参数量。

根据应用场合的要求，可以在准确率、计算速度、延迟率、模型大小之间进行折中，选择使用或者不使用两个参数因子。

4 棉花自动分级系统实验

4.1 模型训练

4.1.1 数据处理

1. 格式转换

本文的分类标准参考美国陆地棉中的白棉的七个等级：gm、go、lm、m、sgo、slm、sm。棉花原始数据集以 tif 格式保存，每个类别包含 108 张 tif 格式图片，且部分图片含有多个图层。如图 4-1、4-2 所示。



图 4-1 原始图片格式

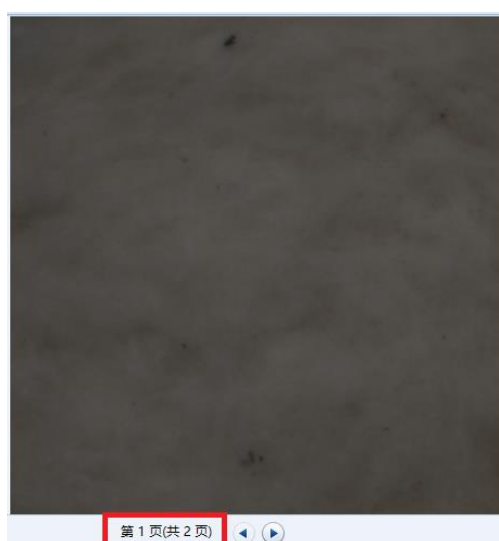


图 4-2 多图层

然而 tensorflow 框架训练模型只能处理 jpg、jpeg、png 格式图片，因此需要转换图片格式。本文使用 matlab 将图片转换为 png 格式，图 4-3 以 sm 级别为例进行操作。

```
path = 'G:\cotton_photos\sm\';
savepath = 'G:\cotton_png\sm\';
cd(path); %切换到该文件夹下
allfigs = struct2cell(dir('*.tif')); %只处理tif文件
[w,img_num]=size(allfigs); %获得tif文件的个数
for pn=1:img_num %逐次取出图片
    image_name = allfigs{1,pn};
    info = imfinfo(image_name);
    slice = size(info,1);
    for i = 1:slice % 依次读取每个图层
        image = imread(image_name,i);
        new = image_name(1:end-4);
        imwrite(image,[savepath,new,'_',num2str(i,'%01d'),'png']);
    end
end
```

图 4-3 格式转换函数

转换后的每个类别棉花包含 126 张图片，增加了 18 张原始图片的第二图层，为避免此图像对模型训练产生影响，将其剔除。如图 4-4 所示。

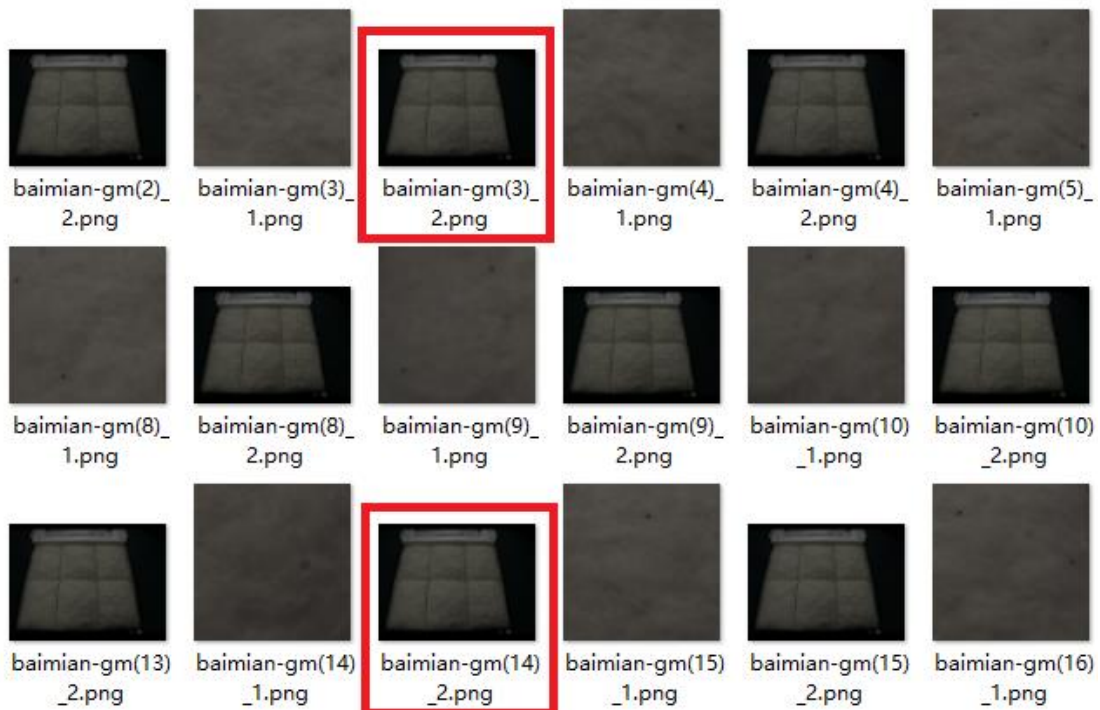


图 4-4 png 棉花图片

2. 数量扩充

格式转换后的棉花图片每一类别仅有 108 张, 考虑到数据规模对模型训练影响较大的同时, 棉花图像的杂质点分布并不均匀, 平均裁剪图片扩大数据集规模并不可行。综合考虑, 本实验采用上下翻转、左右翻转、对角线翻转、随机调整亮度、随机调整对比度、随机调整色调、随机调整饱和度扩充图片的数量。扩充后的每一类别图片包含 $108 \times 8 = 864$ 张。图 4-5 为图片扩充核心函数、图 4-6 为扩充效果图。

```
def expand_Image(img, selector):  
    if selector==0:  
        newImage = tf.image.random_flip_up_down(img)  
    elif selector==1:  
        newImage = tf.image.random_flip_left_right(img)  
    elif selector==2:  
        newImage = tf.image.transpose_image(img)  
    elif selector==3:  
        newImage = tf.image.random_brightness(img, max_delta=0.2)  
    elif selector==4:  
        newImage = tf.image.random_contrast(img, lower=0.5, upper=1.5)  
    elif selector==5:  
        newImage = tf.image.random_hue(img, max_delta=0.2)  
    else:  
        newImage = tf.image.random_saturation(img, lower=0.5, upper=1.5)  
    newImage = tf.image.convert_image_dtype(newImage, dtype=tf.uint16)  
    encoded_image = tf.image.encode_png(newImage)  
    return encoded_image
```

图 4-5 图片扩充核心函数

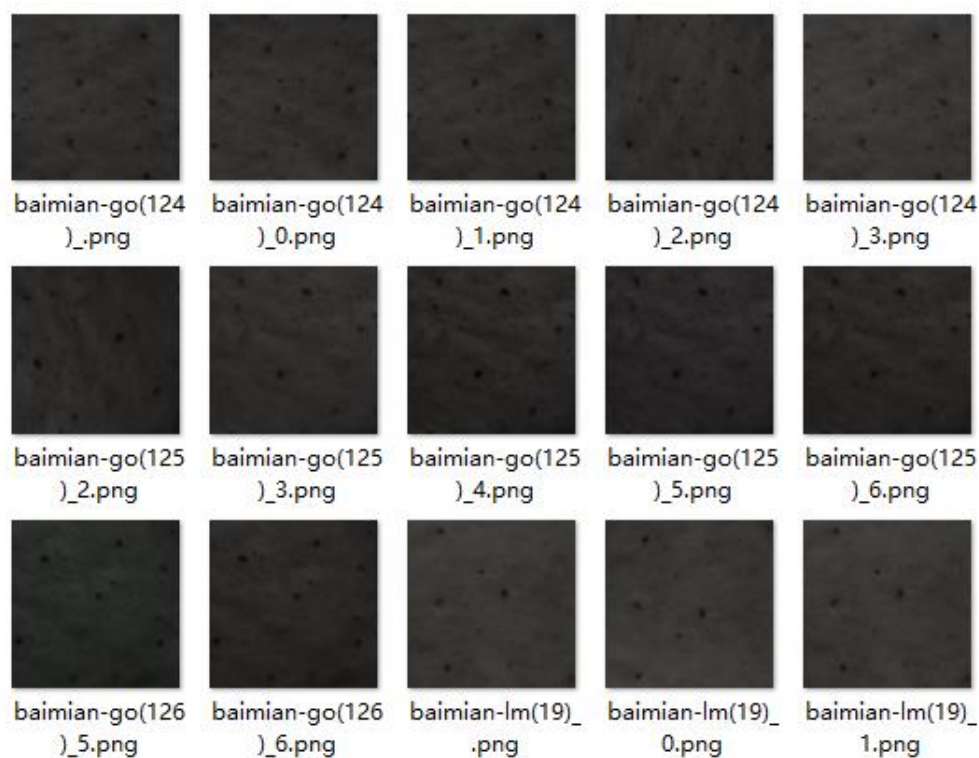


图 4-6 单张图片扩充效果图

3. tfRecord 格式转换

单张图片散列在磁盘上不仅占据大量空间，而且训练时读取图片繁琐。本文按照 train:validation=9:1 比例进行数据集划分组织成 tfRecord 格式文件，输入神经网络训练，加快模型读取图片速度，提高内存利用率。图 4-7 为转换后的 tfRecord 文件。

cottons_train_00001-of-00009.tfrecord	2019/4/8 11:15	TFRECORD 文件	146,737 KB
cottons_train_00002-of-00009.tfrecord	2019/4/8 11:17	TFRECORD 文件	146,690 KB
cottons_train_00003-of-00009.tfrecord	2019/4/8 11:21	TFRECORD 文件	145,524 KB
cottons_train_00004-of-00009.tfrecord	2019/4/8 11:26	TFRECORD 文件	147,345 KB
cottons_train_00005-of-00009.tfrecord	2019/4/8 11:31	TFRECORD 文件	145,648 KB
cottons_train_00006-of-00009.tfrecord	2019/4/8 11:38	TFRECORD 文件	145,431 KB
cottons_train_00007-of-00009.tfrecord	2019/4/8 11:47	TFRECORD 文件	144,073 KB
cottons_train_00008-of-00009.tfrecord	2019/4/8 11:56	TFRECORD 文件	147,275 KB
cottons_validation_00000-of-00009.tfr...	2019/4/8 11:56	TFRECORD 文件	17,149 KB
cottons_validation_00001-of-00009.tfr...	2019/4/8 11:57	TFRECORD 文件	16,686 KB
cottons_validation_00002-of-00009.tfr...	2019/4/8 11:57	TFRECORD 文件	17,140 KB
cottons_validation_00003-of-00009.tfr...	2019/4/8 11:57	TFRECORD 文件	16,674 KB
cottons_validation_00004-of-00009.tfr...	2019/4/8 11:57	TFRECORD 文件	15,651 KB
cottons_validation_00005-of-00009.tfr...	2019/4/8 11:57	TFRECORD 文件	16,263 KB
cottons_validation_00006-of-00009.tfr...	2019/4/8 11:57	TFRECORD 文件	17,232 KB
cottons_validation_00007-of-00009.tfr...	2019/4/8 11:58	TFRECORD 文件	17,315 KB
cottons_validation_00008-of-00009.tfr...	2019/4/8 11:58	TFRECORD 文件	13,827 KB

图 4-7 tfrecord 文件

4.1.2 模型配置

1. 模型参数配置

由于棉花的原始数据集较小，再次进行 MobileNet 网络的精简，可能会造成模型的分类精度不高，从而使该自动分类系统不具备使用价值。因此，该网络的 width multiplier 与 resolution multiplier 均设置为 1，即处理的图像尺寸为默认最大 224*224，不进行网络的压缩。

2. 学习率、batch、滑动平均衰减系数、迭代轮数

本实验选择指数衰减方法更新学习率，初始学习率初始设置较大为 0.005，加快模型参数更新速度，提高模型训练速度，每过 2 个 epoch 的训练数据后更新学习率，学习率衰减到 0.00001 后停止更新，这样做可以避免后期学习率过小，参数更新缓慢，模型训练精度提升不明显。

过小的 batch 可能会导致损失值曲线震荡幅度过大，使模型无法收敛；过大的 batch 受电脑内存限制，则会降低计算速度，甚至会出现程序崩溃，不利于模型的训练，经过训练探索，本实验 batch 设置为 60。

滑动平均衰减系数设置为 0.99，本实验初始迭代轮数为 40000，并随着模型精度随时决定模型训练是否继续进行。

3. 优化算法

反向传播使用梯度下降法更新参数，根据计算梯度方式划分为不同类型梯度下降算法。常用的有 BGD 计算整个数据集参数的梯度、SGD 计算单个样本参数的梯度、MBGD 计算一小批样本参数的梯度、RMSprop 为各个参数自适应分配不同学习率。由于本实验采用指数衰减更新学习率，且实验后期学习率下降过快，模型的收敛速度可能会很慢，而 RMSProp 很好的考虑到这一问题，因此本文采用 RMSprop 优化算法。

4. 训练特征输出

由于不同卷积层的卷积核的数量不同，为了全面体现某卷积层的卷积结果，本实验进行卷积层切割，取得每一个卷积核上的特征，按照卷积核的平方数组合成一个正方形矩阵，按照类似过程处理网络全连接层输出的 7 个分类结果特征图。图 4-9 显示了单张图片经过网络中不同卷积层所提取的特征图，由于篇幅限

制，本文仅截取少部分卷积层提取的特征图，图 4-10 显示单张图片不同训练轮数后，经卷积神经网络计算输出的分类结果特征图。

```
def _concat_features(conv_output):
    num_or_size_splits = conv_output.get_shape().as_list()[-1]
    each_convs = tf.split(conv_output, num_or_size_splits=num_or_size_splits,
axis=3)
    concat_size = int(math.sqrt(num_or_size_splits) / 1)
    all_concat = None
    for i in range(concat_size):
        row_concat = each_convs[i * concat_size]
        for j in range(concat_size - 1):
            row_concat = tf.concat([row_concat, each_convs[i * concat_size +
j + 1]], 1)
        if i == 0:
            all_concat = row_concat
        else:
            all_concat = tf.concat([all_concat, row_concat], 2)
    return all_concat
```

图 4-8 特征图拼接函数

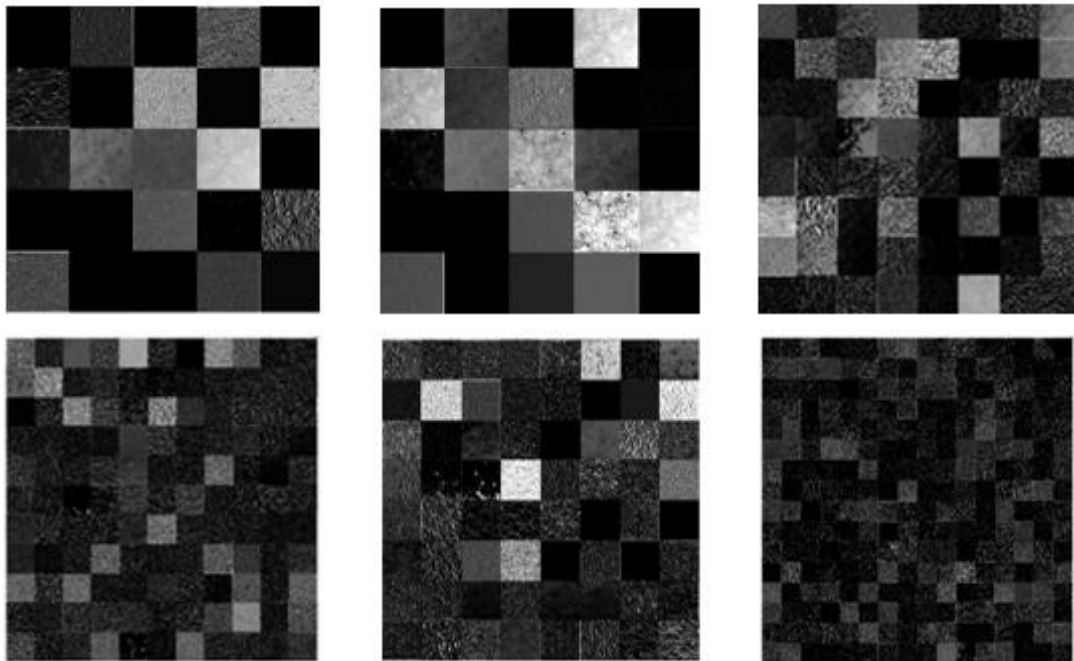


图 4-9 特征图

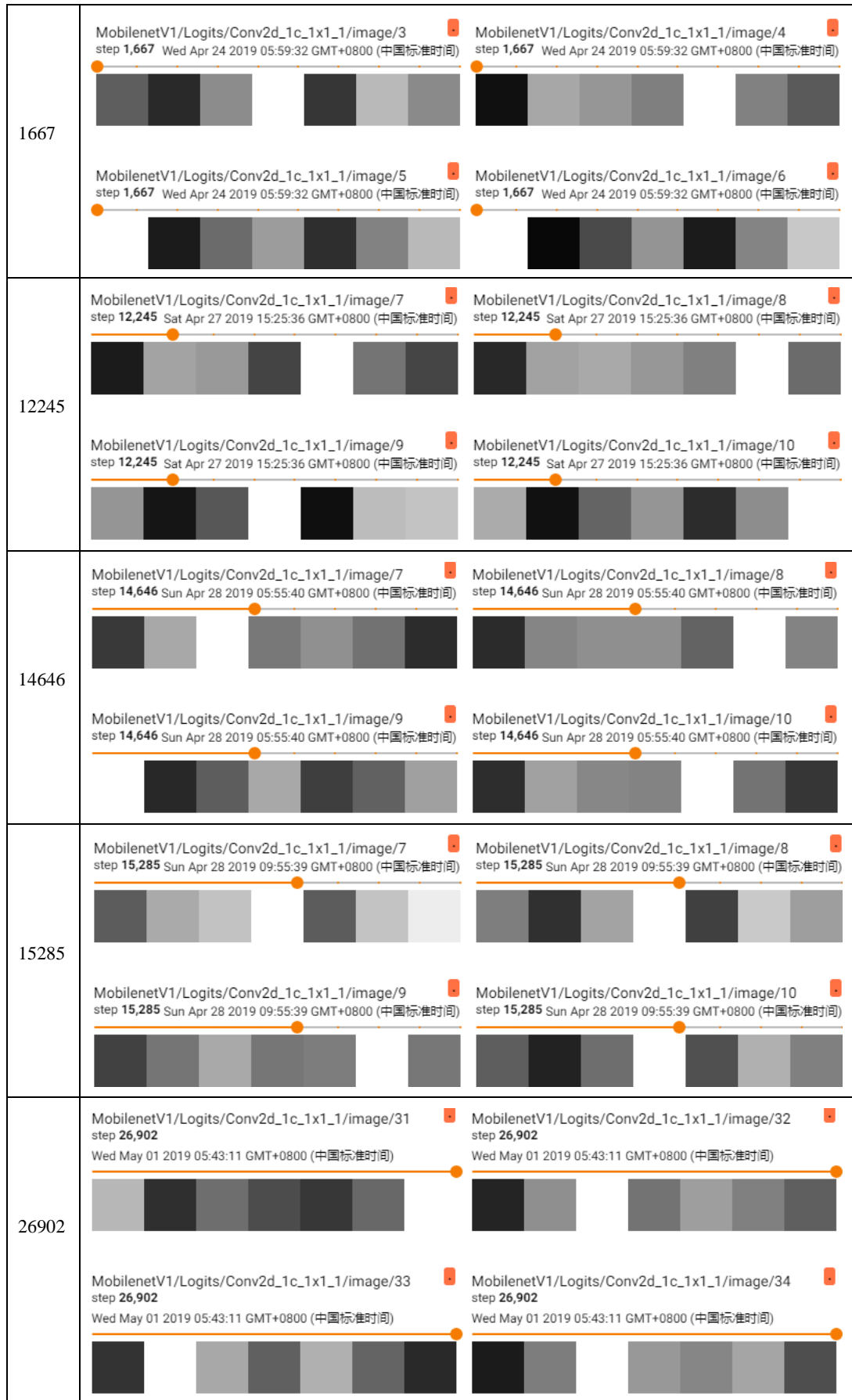


图 4-10 分类结果特征图

4.2 移动端配置

4.2.1 开发工具

棉花图像自动分类系统使用 android studio、SDK 进行移动端部署。android studio 工具基于 gradle 进行移动端的配置，具有支持多个工程构建的强大能力，同时提供功能强大的布局编辑器，方便开发者拖动 UI 控件进行效果预览；SDK，即 software development kit 提供兼容 android 系统各版本的软件包，辅助 android studio 进行移动端开发。

4.2.2 ckpt 文件转换 pb 文件

TensorFlow 保存的训练模型，称作 ckpt 文件。ckpt 文件记录了模型的结构、数据流、网络权重等完整训练信息，防止因断电、误操作等突发状况造成训练停止，因信息丢失重新训练，也方便后期对模型进行微调、部署、测试。但该结构的文件因保存大量信息，不易进行移植，为了便于移动端移植，需要将模型进行固化，即固定变量，不再进行反馈调节更新参数，形成 pb 文件。图为训练输出的 ckpt 文件，图 4-12 为 ckpt 转换成 pb 文件的核心函数。




	mobilenet_v1_1.0_224.ckpt.data-0000...	2018/8/3 9:38	DATA-00000-OF...	66,312 KB
	mobilenet_v1_1.0_224.ckpt.index	2018/8/3 9:38	INDEX 文件	20 KB
	mobilenet_v1_1.0_224.ckpt.meta	2018/8/3 9:38	META 文件	3,308 KB

图 4-11 ckpt 文件

重 构 模 型	<pre>def build_model(inputs): with slim.arg_scope(mobilenet_v1_arg_scope(is_training=False)): logits, end_points = mobilenet_v1(inputs, is_training=False, depth_multiplier=1.0, num_classes=7) scores = end_points['Predictions'] print(scores) output = tf.nn.top_k(scores, k=3, sorted=True) return output.indices, output.values</pre>
加 载 模 型	<pre>def load_model(sess): loader = tf.train.Saver() loader.restore(sess, CKPT)</pre>

固 化 模 型	<pre> inputs = tf.placeholder(dtype=tf.float32, shape=(1, 224, 224, 3), name='input') classes_tf, scores_tf = build_model(inputs) classes = tf.identity(classes_tf, name='classes') scores = tf.identity(scores_tf, name='scores') with tf.Session() as sess: load_model(sess) graph = tf.get_default_graph() output_graph_def = tf.graph_util.convert_variables_to_constants(sess, graph.as_graph_def(), [classes.op.name, scores.op.name]) tf.train.write_graph(output_graph_def, 'E:/model/test/15576', frozen.pb', as_text=False) </pre>
------------------	---

图 4-12 ckpt-pb 文件转换核心函数

4.2.3 实时检测

实时检测即实时获取屏幕图像，输入模型计算棉花类别，该功能基于 SDK 提供的管理屏幕内容的 surface 工具包实现，该工具包含 surfaceHolder 与 surfaceView 两个基类，surfaceHolder 在系统获取用户的摄像头权限后，监听 surface 的创建、改变、销毁，通过回调函数对当前屏幕像素进行操作；surfaceView 将当前窗体数据输出显示到屏幕预览。需要注意的是 Android 摄像头横置，为了方便用户预览，另外需要将捕捉到的屏幕内容垂直翻转。图 4-13、4-14、4-15 分别为该操作的主要函数。

```

public void surfaceCreated(SurfaceHolder holder) {}

public void surfaceDestroyed(SurfaceHolder holder) {}

public void surfaceChanged(SurfaceHolder holder, int format, int w, int h) {}

```

图 4-13 屏幕活动

```

public void onPreviewFrame(byte[] bytes, Camera camera) {
    Bitmap bm = byte2bitmap(bytes,camera);
    Map out = utils.run(bm);
    String[] labels=(String[]) out.get("classes");
    float[] scores=(float[]) out.get("scores");
    String s="";
    for(int i=0;i<3;i++){
        s=s+"类别"+labels[i]+", 识别概率"+scores[i]+"n";
    }
    outputTv.setText(s);
}

```

图 4-14 相机回调函数

```

private Bitmap byte2bitmap(byte[] bytes, Camera camera) {
    Bitmap bitmap = null;
    Camera.Size size = camera.getParameters().getPreviewSize(); // 获取预览大小
    final int w = size.width;
    final int h = size.height;
    final YuvImage image = new YuvImage(bytes, ImageFormat.NV21, w, h, null);
    ByteArrayOutputStream os = new ByteArrayOutputStream(bytes.length);
    if (!image.compressToJpeg(new Rect(0, 0, w, h), 100, os)) {
        return null;
    }
    byte[] tmp = os.toByteArray();
    bitmap = BitmapFactory.decodeByteArray(tmp, 0, tmp.length);
    Matrix matrix = new Matrix();
    matrix.setRotate(-90);
    bitmap = Bitmap.createBitmap(bitmap, 0, 0, bitmap.getWidth(), bitmap.getHeight(), matrix, true);
    return bitmap;
}

```

图 4-15 像素解码及转置

4.2.4 图片分类

系统成功获取相册权限，在回调函数中获取用户选择的棉花图片路径，读取图片显示在屏幕，并输入模型计算棉花类别。图 4-16、4-17、4-18、4-19 为该操作的主要函数。

```

public void openAlbum(){}

@Override

public void onRequestPermissionsResult(int requestCode, String[] permissions, int[] grantResults) {}

@RequiresApi(api = Build.VERSION_CODES.KITKAT)

@Override

public void onActivityResult(int requestCode, int resultCode, Intent data){}

```

图 4-16 相册回调函数

```
private String getImagePath(Uri uri,String selection){
    String path = null;
    Cursor cursor  = getResolver().query(uri,null,selection,null, null);
    if(cursor!=null){
        if(cursor.moveToFirst())
path=cursor.getString(cursor.getColumnIndex(MediaStore.Images.Media.DATA));
        cursor.close();}
    return path;
}
```

图 4-17 获取图片路径

```
private void displayImage(String imagePath){
    if(imagePath!= null){
        Bitmap bitmap = BitmapFactory.decodeFile(imagePath);
        picture.setImageBitmap(bitmap);
        Log.d("bitmap", "bitmap is"+bitmap);
        getResult(bitmap);
        if (bitmap != null && !bitmap.isRecycled())
        {
            bitmap=null;
        }
    }
    else
        Toast.makeText(this,"failed to get image", Toast.LENGTH_SHORT).show();
}
```

图 4-18 展示图片

```
private void getResult(Bitmap bitmap){
    Map out = utils.run(bitmap);
    String[] labels=(String[]) out.get("classes");
    float[] scores=(float[]) out.get("scores");
    String s="";
    for(int i=0;i<3;i++){
        s=s+"类别"+labels[i]+"， 识别概率"+scores[i]+"n";
    }
    resultTv.setText(s);
}
```

图 4-19 获取分类结果

4.2.5 Bitmap 图像转换

屏幕获取与相册读取的图片格式均为 bitmap 格式，但模型输入为 224*224 的 float 型数据，因此输入模型进行分类前需进行图片尺寸的调整与格式的转换。

图 4-20、4-21 分别为图像尺寸调整函数，格式转换函数。

```
public Bitmap getResizedBitmap(Bitmap bm, int newWidth, int newHeight) {
    int width = bm.getWidth();
    int height = bm.getHeight();
    float scaleWidth = ((float) newWidth) / width;
    float scaleHeight = ((float) newHeight) / height;
    Matrix matrix = new Matrix();
    matrix.postScale(scaleWidth, scaleHeight);
    Bitmap resizedBitmap = Bitmap.createBitmap( bm, 0, 0, width, height, matrix,
false);
    return resizedBitmap;
}
```

图 4-20 图像尺寸调整

```
private float[] getFloatImage(Bitmap bitmap){
    Bitmap bm = getResizedBitmap(bitmap,inputWH,inputWH);
    bm.getPixels(inputIntData, 0, bm.getWidth(), 0, 0, bm.getWidth(),
bm.getHeight());
    for (int i = 0; i < inputIntData.length; ++i) {
        final int val = inputIntData[i];
        inputFloatData[i * 3 + 0] =(float) (((val >> 16) & 0xFF)/255.0-0.5)*2;
        inputFloatData[i * 3 + 1] = (float)(((val >> 8) & 0xFF)/255.0-0.5)*2;
        inputFloatData[i * 3 + 2] = (float)(( val & 0xFF)/255.0-0.5)*2 ;
    }
    return inputFloatData;
}
```

图 4-21 格式转换

4.2.6 Tensorflow Mobile 接口

android studio 中使用 Tensorflow Mobile 库进行模型移植，通过调用库中的 `org.tensorflow.contrib.android.TensorFlowInferenceInterface` 封装类调用模型完成分类预测等任务，图 4-22 为该类中的四个主要函数：

```
inferenceInterface= new TensorFlowInferenceInterface(assetMgr, model);
inferenceInterface.feed(inputName, inputData, 1, inputWH, inputWH, 3);
inferenceInterface.run( outputNames );
inferenceInterface.fetch(outputNames, dst);
```

图 4-22 接口函数

TensorFlowInferenceInterface()进行初始化,参数 assetManager 是资源管理器,用于获取 android 工程 asset 目录下模型、分类标签等文件,参数 model 表示 pb 模型文件名;feed()函数组织数据输入模型,参数 inputname 表示模型输入节点名称,src 与 dims 表示源数据、shape,如该模型输入数据 shape=[1,224,224,3];run()函数执行模型从输入到输出节点的计算过程,参数 outputNames 表示定义的模型输出节点名称;fetch()函数用于获得指定的数据,参数 outputName 表示输出节点,dst 表示指定的输出数据。本设计将该接口封装成 util 工具类,并对外提供 run()函数完成对模型的调用。图 4-23 为对外提供的接口。

```
public Map<String,Object> run(Bitmap bitmap){
    float[] inputData = getFloatImage(bitmap);
    inferenceInterface.feed(inputName, inputData, 1, inputWH, inputWH, 3);
    inferenceInterface.run( outputNames );
    int[] classes=new int[3];
    float[] scores=new float[3];
    inferenceInterface.fetch(outputNames[0], classes);
    inferenceInterface.fetch(outputNames[1], scores);
    Map<String,Object> results=new HashMap<>();
    results.put("scores",scores);
    String[] classesLabel = new String[3];
    for(int i =0;i<3;i++){
        int idx=classes[i];
        classesLabel[i]=dict.get(idx);
    }
    results.put("classes",classesLabel);
    return results;
}
```

图 4-23 对外接口

4.3 实验结果与分析

1. 重训练 logits 输出层

本实验在进行数据集规模扩充时，由于随机调整亮度、随机调整对比度、随机调整色调、随机调整饱和度产生部分图片与原始数据外观差异较大。实验前期直接将该部分图片剔除，此时棉花各类别的数目不等，gm 类别 847 张、go 类别 821 张、lm 类别 833 张、m 类别 847 张、sgo 类别 835 张、slm 类别 848 张、sm 类别 855 张，模型准确率为 0.941667；考虑到数据不均衡可能会对模型精度造成影响，实验后期进行再处理扩充后的棉花数据集，使棉花各类别数量相等，各个类别均为 850 张，重新训练，模型准确率为 0.96。图 4-24 为剔除图片。

表 4-1 数据均衡分布对模型准确率的影响

数据是否均衡分布	测试集准确率
是	96%
否	94.1667%

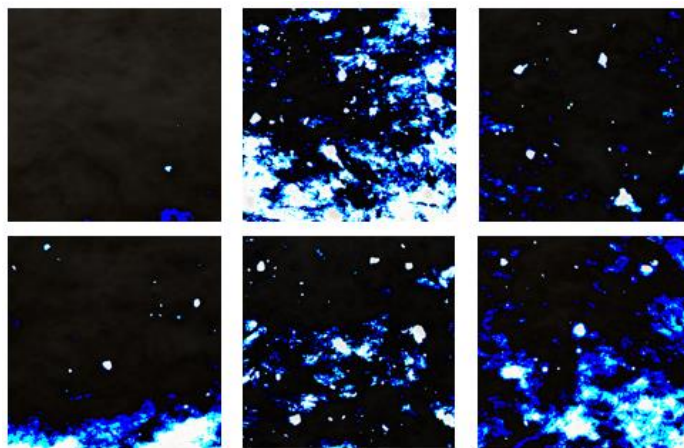


图 4-24 剔除图片

从表 4-1 结果可知，数据非均衡分布时，模型识别准确率略有下降，由此可知，数据是否均衡分布对模型的准确率有一定影响，当数据均衡分布，模型的识别也会随之提高。

2. 重训练不同卷积层

本实验在数据均衡分布条件下继续探讨训练不同层数对模型准确率的影响，其中 **depthwise** 与 **pointwise** 为一个深度可分离卷积层，模型训练时以一个深度可分离卷积层为单位，不单独拆开训练，每次增加三个深度可分离卷积层。

表 4-2 训练不同网络层对模型准确率的影响

网络层名称	测试集识别准确率
logits 层	96%
11-13 卷积层、logits 层	97.5%
8-13 卷积层、logits 层	97.5%
5-13 卷积层、logits 层	97.33%
完整网络	95.33%

由表 4-2 数据可知，随着训练层数的增加，棉花分类准确率呈上升增长趋势，随后准确率开始出现下降，说明逐渐增加训练层数并不能保证分类准确率直线上升，本实验将在准确率最高的训练层数附近，即 8-13 层之间，继续寻找训练该模型最合适的训练层数。

3. 寻找最优瓶颈层模型

表 4-3 寻找最优瓶颈层模型

网络层名称	测试集识别准确率
12-13 卷积层、logits 层	96.67%
10-13 卷积层、logits 层	98.33%
9-13 卷积层、logits 层	96.33%

由表 4-3 呈现的准确率变化趋势，可知从第 10 层深度可分离卷积层重新训练模型，相比重新训练其他层，准确率达到最高值 98.33%。

4. 移动端使用界面

android 移动端使用界面如下图 4-25，分别显示分级前三类别的识别结果与概率。由系统运行结果可知，读取相册进行分级，准确率与模型在测试集表现并无差异，准确率较高，“实时检测”受屏幕亮度以及光照等条件影响，分类准确率变动较大。

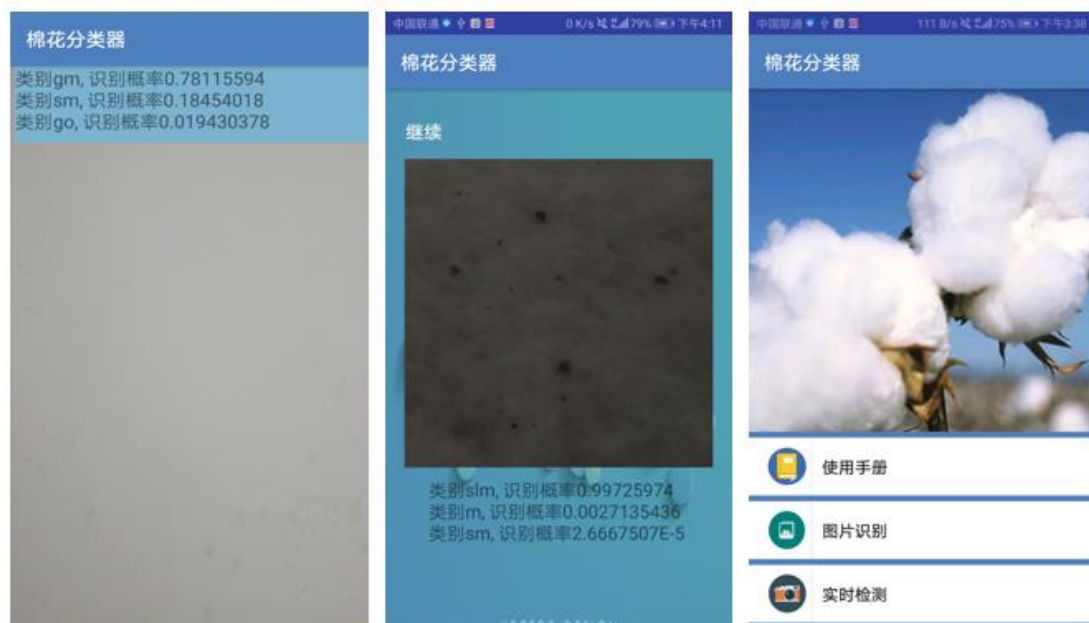


图 4-25 移动端使用截图

5 总结

5.1 完成的工作

本设计基于 MobileNet 卷积神经网络，使用迁移学习训练调整模型，解决棉花图像分级问题，并使用 android studio、SDK 开发工具，将模型部署在 android 移动端。实验对数据集进行扩充、调整，最终模型在测试集上的错误率减小到 2% 以内，相比依赖机器检测与人工感官检验传统分级方法，本文提出的方法受外界因素影响较小，只需要满足一定的光照条件，一台 android 手机即可进行棉花分级，更加方便快捷，分级结果也具有较高的准确度，因此，本系统具有较高的使用价值。

5.2 设计的不足

1. 实时检测分类结果变化较大

本文在进行数据集拍摄时，对硬件环境要求较高，拍摄地点、相机与棉花的相对位置均须固定，且模型的训练数据集为灰度图片，因此使用该数据集训练的模型进行“实时检测”时，受屏幕亮度、光照等条件影响，对分级准确率影响较大。

2. 只适用于 Android 移动端

本文只进行 Android 移动端棉花图像自动分类系统的部署，并未考虑 ios 系统的实现。

3. 算法优化

MobileNet 网络结构相对其他卷积神经网络结构简单，比较适用数据集规模较小的情况。本文基于该网络，使用迁移学习对比不同瓶颈层分类效果，最终达到棉花分类模型准确度最高 98.33%，限于时间与精力，本文并未考虑通过自定义损失函数等方法优化模型结构，提高模型的准确率。

参考文献

- [1] <https://zhidao.baidu.com/question/135296505021323005.html>. 百度知道
- [2] 陈昌江,李成山. 浅谈我国棉花分级系统的演变及认识[J]. 中国纤检, 2012(11):63-65.
- [3] 李静. 计算机图像技术在农业工程中的应用研究[J]. 电子技术与软件工程, 2014(2):123-123.
- [4] T. Schneider, 张国莲, 陈廷. HVI 和 ART——未来的质量检验仪器[J]. 国际纺织导报, 2005(1):25-26.
- [5] 林起.美国棉检技术进展概述[J]. 中国棉花, 1996(4):2-5.
- [6] 雷雷.神经网络在棉花色征级检验中的研究应用进展[J].中国纤检,2008(05):60-63.
- [7] 张婷,高颖,王东,吕炎,董军宇,亓琳,陈鹏.基于距离度量学习的棉花品级分类方法研究[J]. 中国棉花,2018,45(08):12-19.
- [8] 毋立芳,汪敏贵,付亨,简萌.深度目标检测与图像分类相结合的棉花发育期自动识别方法[J].中国科技论文,2018,13(20):2309-2316.
- [9] Majumdar,P. K. Majumdar,B. Sarkar. Application of an adaptive neuro-fuzzy system for the prediction of cotton yarn strength from HVI fibre properties[J]. Journal of the Textile Institute,2010,7(08):55-60.
- [10]徐亚伟,杨会成.基于多层特征深度融合的卷积神经网络人脸识别方法[J].平顶山学院学报,2019,34(02):53-58.
- [11]马冬梅,贺三三,杨彩锋,严春满.特征融合型卷积神经网络的语义分割[J/OL].计算机工程与应用:1-10.
- [12]Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo. TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems[J]. Duker University: Preliminary White Paper, 2016.
- [13]郑泽宇,顾思宇. TensorFlow 实战 Google 深度学习框架. 第一版. 北京: 电子工业出版社, 2017. 283
- [14]Yann LeCun, Yoshua Bengio, Geoffrey Hinton. Deep learning[J]. Nature, 2015,5(10):436-444
- [15]Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks[J]. Communications of the ACM, 2017,60(6):84-90

- [16] Anrew G.Howard, Menglong Zhu, Bo Chen. MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications[J].Computer Vision and Pattern Recognition, 2017. arXiv:1704.04861

致谢

大学生活即将结束，在此，我要感谢所有教导我的老师和陪伴我一齐成长的同学，他们在我的大学生涯给予了很大的帮助。回首整个毕业设计工作，通过查阅、筛选、收集资料，我从最初面对如此复杂工程而茫然不知所措，逐渐掌握实施毕业设计的关键技术，最后顺利完成毕业设计工作。持续将近四个月的设计过程，我不仅学会了如何在浩如烟海的资料中收集到目标资料，提高个人信息检索能力，在一次次的调试配置过程中，提高了定位、分析、解决问题与编码的能力，而这些能力的提高对我个人职业生涯将起到巨大的积极作用。

在此，万分感谢我的指导教师董军宇教授于百忙之中，仍旧积极引导我如何思考寻找问题的关键，“授人以鱼不如授人以渔”，这句话在董教授指导方式中得到了充分体现。同时，感谢高颖学姐耐心回答我关于毕业设计过程中出现的所有问题，在我疑惑、遇到瓶颈时，安抚我的情绪，帮我找到顺利进行工作的突破口。董教授严谨的治学态度与高颖学姐的热情帮助，给我精神带来了巨大的触动，无论是未来的求学或是职业生涯，他们的态度与精神将是我人生路上的指明灯，再次对两位表示衷心的感谢！