# 基于深度学习的路面纹理参数检测模型

## 1深度学习

2012年，AlexNet取得ImagNet挑战赛冠军，在Top-1分类精度上超过第二名近10个百分点，深度学习开始引起学者们的关注。2016 年，AlphaGo算法相继赢得了与李世石和柯洁的围棋比赛，超过人类围棋顶尖水平，深度学习彻底火爆起来。目前，深度学习技术已经在日常生活中不同领域得到应用。

深度学习的本质是表示学习，即对特征信息的提取和映射。目前，深度学习中发展较快的技术就是深度神经网络。深度神经网络对许多常见的数据均具有较强的处理能力。深度神经网络可以提取输入数据的特征，之后进行特征的多层映射，最后再将其映射回特定的任务解空间，这样便可以得到特定任务的解。深度神经网络中的参数都能够进行自动更新，使其参数更加接近最优解，一般是采取反向传播算法将其更新，这样网络便可以学习更加抽象和丰富的特征表示，建模输入特征和解空间的映射关系，从而实现对目标检测和自然语言处理等问题的求解。

在传统的机器学习中，需要事先根据识别目标的特点设计专门的特征提取器，特征工程的好坏直接决定了检测效果。但是，深度学习能够通过学习输入数据的特点和分布，自动地学习与识别目标相关的特征，而且学到的特征表示性能更好。其次，随着数据规模的增加，深度学习模型具有更好的泛化能力。最后，通过替换不同数据集，深度学习可以在不修改模型结构的情况下完成对新类别图像的分类任务。

### 1.1框架比选

选择一个合适的深度学习框架十分重要的，因为一个合适的深度学习框架能起到事半功倍的作用。在深度学习框架形成之前，研究者使用各种不同的方式达到他们的研究目的，需要写大量的重复代码，导致学习的门槛比较高，而且难以复现，影响了学习的积极性。为了解决这些阻碍深度学习发展的问题，越来越多的开发者就这些代码写成框架放在网上，方便研究者使用。当前最为流行的深度学习框架有PyTorch、Tensorflow、Caffe和PaddlePaddle。

1. PyTorch是一个开源的python工具库，是在2017年由Facebook人工智能院基于Torch.开发而来的深度学习框架；Torch是由卷积神经网络之父YannLeCun领导开发的框架，于2014年开源，曾是Facebook和'Twitter主推的一款开源深度学习框架，使用Python修改了很多内容，使Pytorch更加灵活，同时支持动态图，得到了更多研发人员的认可。PyTorch有利于研究开发人员快速生成模型，支持GPU，更易于搭建和训练深度神经网络。

2. Tensorflow是由谷歌开发的深度学习框架，是一个端到端平台，学者或专家均可以轻松地构建、试验和部署模型。Tensorflow提供多个高级API(Application Programming Interface，应用程序接口)，用户基于业务需求选择适用的API或者自行定制API，将TensorFlow部署在CPU、GPU的服务器中，或应用在移动设备中开展目标检测等，并支持AdaGrad和Neuron Dropout等算法优化。

3. Caffe由加州大学伯克利的博士贾扬清开发，全称是Convolutional Architecture for Fast Feature Embedding，是一个清晰而高效的开源深度学习框架，支持Python和Matlab接口，为研究者提供多种编程语言来开发算法，并由伯克利视觉中心(Berkeley V ision and Learning Center， BVLC)进行维护。

PaddlePaddle 是2016年百度研发的深度学习平台，是国内最早开源的一个功能较完备的深度学习平台。PaddlePaddle 具有全面的工业级应用模型，涵盖计算机视觉和自然语言处理等多个领域，并开放多个领先的预训练中文模型，复现了多套有价值的算法模型。目前， PaddlePaddle已经实现了API的稳定和兼容性，形成完善的中英双语使用文档，简洁高效、易学易用。

Pytorch不但具有强大的GPU加速的张量计算（如NumPy），更包含自动求导系统的深度神经网络，因此，本项目将基于pytorch搭建路面纹理参数检测模型。

### 1.2残差网络模型

随着网络的加深，其训练难度也越来越大，有些网络在训练的初期有收敛趋势，但随着训练步数的增加存在退化的风险，造成准确率较快达到峰值，错误率会变得越高。然而这种退化导致的更高错误率并不是由于过拟合引起的，而是仅仅增加了更多的层数，深度残差学习( deep residual learning)的提出可以有效的解.决了退化问题，对于更深层的网络都可以训练。

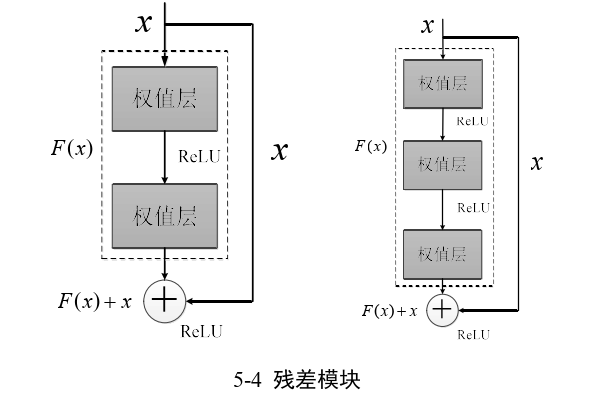


图1 残差模块

与传统的神经网络区别是残差网络在此基础上增加了跨层连接，用来构建残差模块。在一个残差模块中，跨层连接- -般只跨越2~3层， 如图1所示，但也可以跨越更多层。仅仅跨越1层的意义不大，识别结果也不理想，假设从残差模块中丢弃跨层连接，用H(x)代表对应的计算结果，在添加了跨层连接后的计算结果F(x)与H(x)之间存在下面的关系:

(1)

这意味着跨层连接块计算的是不跨连时的残差，所以称为残差模块，从整体的性能上，假如用{W}表示残差模块的全部权值，实际上残差模块算出的是下面

的输出结果:

(2)

其中，F(x，{w})叫做残差映射，这个过程是要学习的，在两个权值层的情况图，在忽略偏置的情况下，。

残差模块的计算要求F(x， {w，})和x有相同的维数，若维数不同，就需要引入一个额外的权值矩阵w，对x进行线性投影，使得维数相同，其计算结果为:

( 3)

基于残差网络，深层残差网络可以拥有非常深的网络结构，可以达到千层以上，表1给出了5个残差网络结构，层数分别为18、34、50、101、152，层名和输出大小和残差模块(用方括号表示)，另外每个残差模块右边的数字代表重复的次数。

表1 五种深层残差网络结构

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| layer name | output size | 18-layer | 34 layer | 50-layer | 101-layer | 152-layer |
| conv 1 | 112×112 | 7×7, 64, stride 2 | | | | |
| conv2\_x | 56×56 | 3x 3 max pool, stride 2 | | | | |
|  |  |  |  |  |
| conv3\_x | 28×28 |  |  |  |  |  |
| conv4\_x | 56×56 |  |  |  |  |  |
| conv5\_x | 56×56 |  |  |  |  |  |
|  | 1×1 | average pool, 1000-d fc, softmax | | | | |
| FLOPs | | 1.8×109 | 3.6×109 | 3.8×109 | 7.6×109 | 11.3×109 |

从表2可以看出，Resnet残差网络结构是由输入层、1个独立的卷积层、1个最大池化层、4种卷积残差模块、1个平均池化和1个最大输出组成。网络的复杂度是由FLOP来度量的，其值越大，网络就越复杂。路面纹理检测模型将采用较常用的 Resnet50残差网络结构，其具体结构如图2所示。

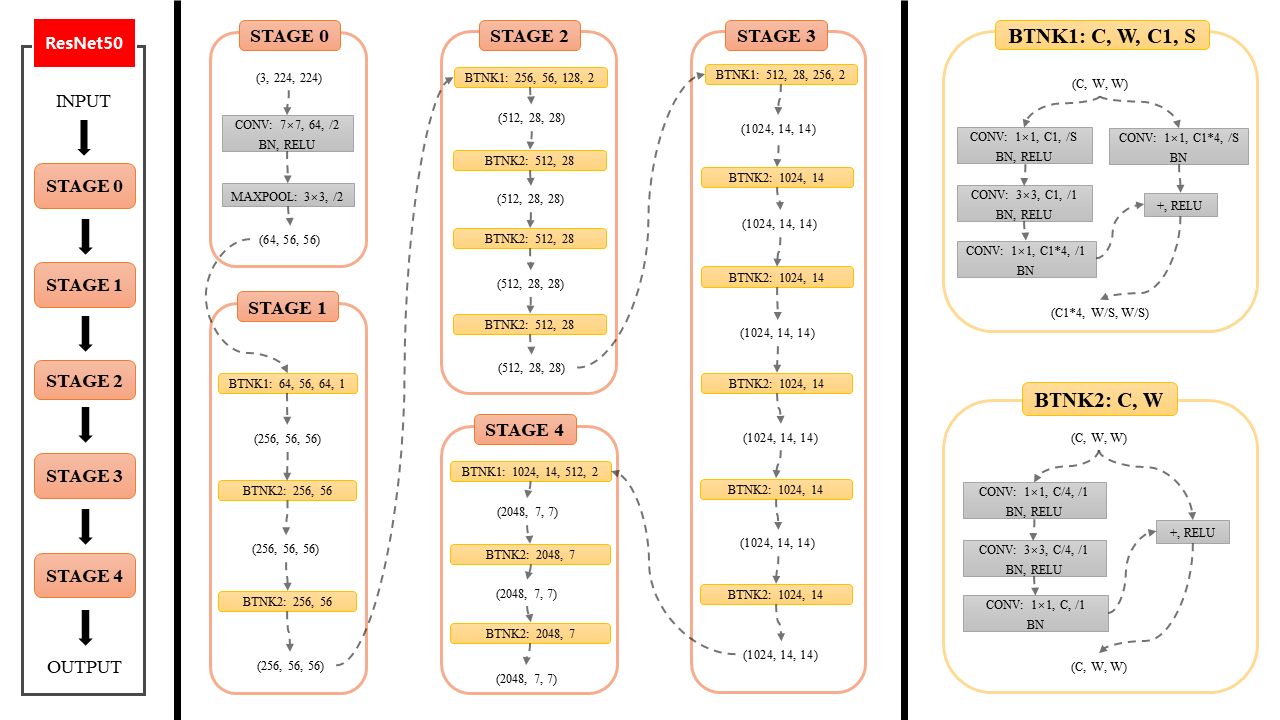


图2 Resnet50网络具体结构

### 1.3优化器及方法

使用严格的反向传播传播算法训练神经网络，需要同时考虑所有样本对梯度的贡献。若样本集比较大，那么梯度下降的每--次迭代都可能花费很长时间，这使得整个过程收敛的速度很慢。

随机梯度下降(SGD)是对梯度下降优化方法的近似。其应用条件是目标函数能够表示成一-组可微函数之和。而神经网络就满足这个条件，所以可以使用随机梯度下降。对神经网络而言，随机梯度下降有两个基本模式:在线和迷你(mini-batch)。

在线模式是先把所有样本随机打乱，再逐一计算每个样本对梯度的贡献去更新权值，即：

其中，表示网络计算样本的实际输出和期望输出之间的误差。

在线模式的缺点是梯度下降的过程不太确定、波动较大。一个折中的方法就是采用“迷你块”模式，实际上就是把全部样本随机处理后分成多个大小为m的块，并逐个计算每块对梯度的贡献去更新权值，即

为了进一步改善随机梯度下降的训练结果，通常采用权值衰减系数，可得下式:

为了进一步提高稳定性，可以引入一个动量项d及其加权系数*v*，得到SGD的基本动量模式：

Adagrad梯度下降是一种自适应的方法，参数的更新量等于梯度乘以学习率，也就是说，更新量与梯度的关系属于线性正相关。但是在运用的过程中，参数的梯度各有不同，有些参数梯度比较大，有参数梯度比较小，因此就有造成参数优化不平衡的现象。

参数优化不均衡对模型训练来说是严峻的问题，这意味着不同的参数更新适用于不同的学习率。而Adagrad 的自适应算法可以解决这样的问题。该算法目的是让不同参数的更新量能够比较均衡。对于已经更新比较多的参数，其更新量要适当衰减，而更新比较少的参数，它的更新量要尽量多一些。给定在时间步长为*t*的参数的损失函数为。

Adagrad旨在每一个时间步长*t*中，求出梯度并推导参数，从而改变学习率*lr*。虽然Adagrad优化算法能够自适应的调节学习率，但是也是存在缺点的。这类算法积累了先前的梯度值和，而且利用这个值来调整全部参数的更新量，先前参数的更新量越大，那么分母也会变大，于是未来它的更新量可能会相对小一点，对于之前更新量小的参数，分母也会相对小一些，于是未来它的更新量会相对大一些。总之就是随迭代次数的增加，更新公式分母也会变得很大，所以理论上更新量也会越来越小，这对优化十分不利。.

Rmsprop算法试图解决Adagrad算法存在的问题。在该算法中分母的梯度平方和不再随优化而递增，而是做加权平均。把Adagrad算法中的改成为。 其中是衰减系数，取值范围为[0，1]。特殊情况是当=1时就成为了Adagrad优化算法，当=0时就成为了SGD优化算法。总的来说，Rmsprop与Adagrad的表现相差不大，但是它的学习率比Adagrad要小很多，其更新的速度比Adagrad要快，这说明了分母对优化的阻碍明显变小了。

Adagrad和Rmsprop算法虽然在某种程度.上解决了“自适应确定更新量”的问题，但是在Adadelta算法中，远远不止这些。Adadelta算法的目的是控制学习率，降低学习速度，在Adagrad的基础上，进行了改进。通过采用窗口*w*，对梯度的历史窗口序列使用均值代替平方和，表达式如下:

在式5.6中，表示当前时刻t梯度的均值， r与动量项相似，都是在0.9左右，与先前SGD参数更新方式的变动如下:

其参数更新的表达式如下：

在式9中，

在式5.10中，表示梯度平方的平均值

参数更新的均方根误差如式5.13：

相比其他优化算法，Adadelta算法不需要设置学习率。

聚类是一个将数据集中在某些方面相似的数据成员进行分类组织的过程，聚类就是一种发现这种内在结构的技术，聚类技术经常被称为无监督学习。

K-means是最著名的划分聚类算法，由于简洁和效率使得他成为所有聚类算法中最广泛使用的。人为给定一个数据点集合和需要的聚类数目k，k均值算法根据某个距离函数反复把数据分入k个聚类中，如图4所示。

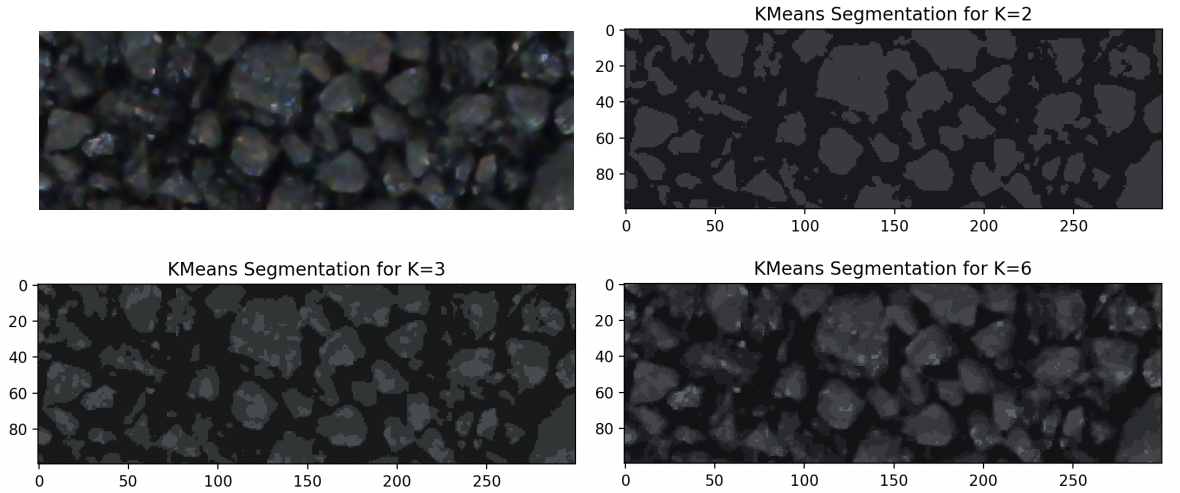


图4 K-means（原图及K=2、3、6）

## 2纹理图像数据集建立

### 2.1图像获取

通过双目相机路面纹理检测系统采集试件图像，共计500张沥青车辙板图片，其中AC级配试件图片143张，SMA级配试件图片170张，OGFC级配试件图片187张。图5为采集的部分试件图像。



图5 部分试件照片

### 2.2图像分割

如图6所示，将每张试件照片分割成6张尺寸为300×100的图像，其中1~4区域位于轮迹带上，5、6区域则靠近车辙板试件中心，共计3000张。图 为分割后的部分图像。

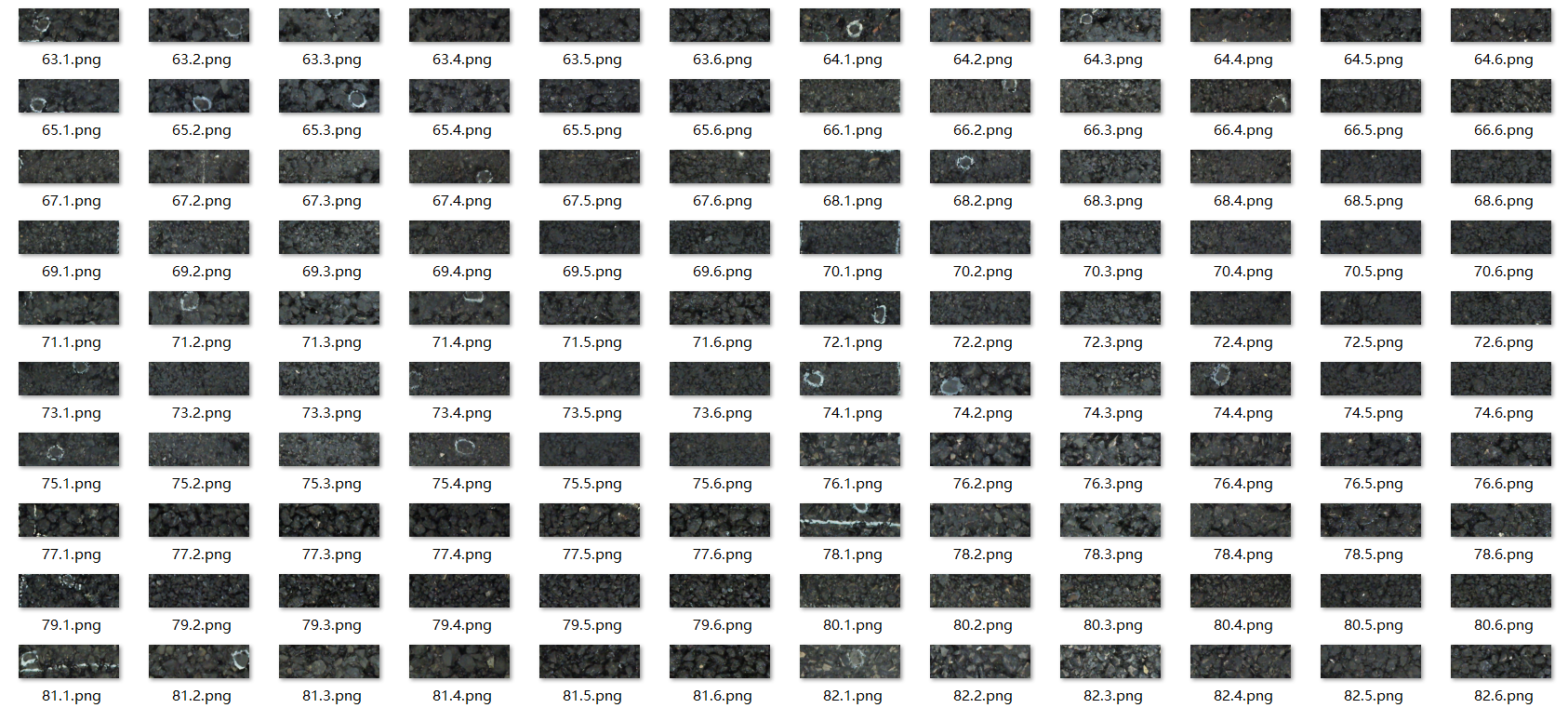
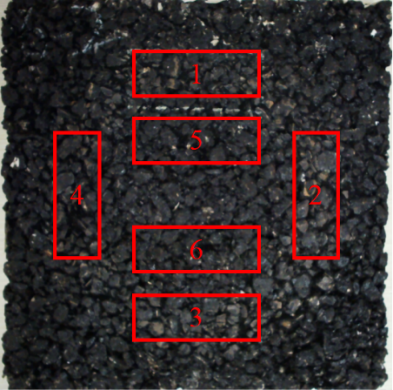


图6

利用双目相机路面纹理检测系统计算每张图像的RMSR，并按级配类型分为AC数据集，SMA数据集，OGFC数据集，ALL数据集（包含三种级配类型试件图像） 。如表2所示，为统一数据集参数值范围跨度，均在实测范围基础上选取跨度为0.8的范围，并细分为8类。最后按照VOC数据集格式构建数据集。

表2 各数据集图片数量

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 数据集 | RMSR实测范围 | 图片数量 | RMSR选取范围 | 图片数量 |
| AC | 0.2~1.1 | 858 | 0.2~1 | 822 |
| SMA | 0.3~1.3 | 1020 | 0.4~1.2 | 948 |
| OGFC | 0.6~1.8 | 1122 | 0.8~1.6 | 978 |
| ALL | 0.2~1.8 | 3000 | 0.5~1.3 | 2000\* |

注，随机选择其中2000张图片。

## 3基于pytorch搭建的路面纹理检测模型

### 3.1实验环境

实验使用 Python 3.6 开发环境，开发工具使用 Anaconda3，操作系统Windows 10。计算机硬件配置：CPU- i7 8750H、GPU-GTX1060、内存DDR4- 16G、硬盘 SA400S37-240G。软件环境：CUDA 10.1、Cudnn 10.1、pytorch-gpu 1.7.1、Spyder 3.6。

### 3.2模型评估指标

在深度学习领域，混淆矩阵（Confusion Matrix）又称为可能性矩阵或错误矩阵。混淆矩阵是可视化工具，特别用于监督学习，在无监督学习一般叫做匹配矩阵。在图像精度评价中，主要用于比较分类结果和实际测得值，可以把分类结果的精度显示在一个混淆矩阵里面。混淆矩阵的结构如图7所示。

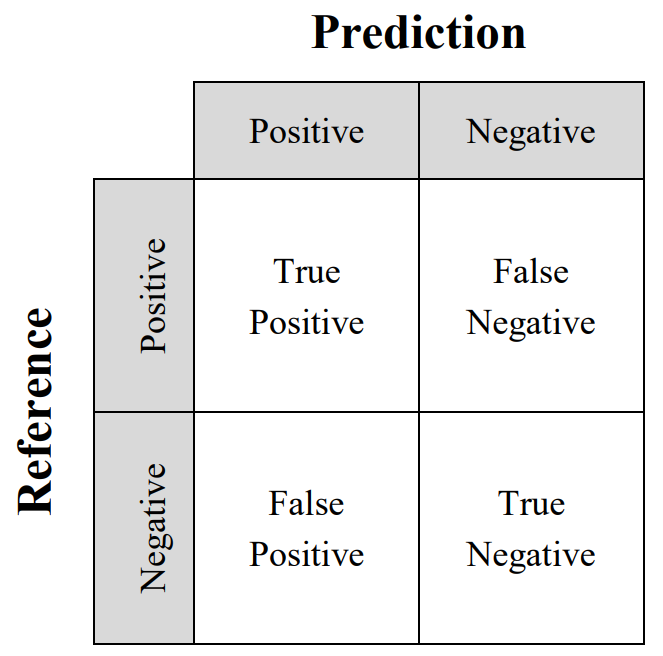


图7 混淆矩阵

混淆矩阵要表达的含义：每一列代表了预测类别，每一列的总数表示预测为该类别的数据的数目；每一行代表了数据的真实归属类别，每一行的数据总数表示该类别的数据实例的数目；每一列中的数值表示真实数据被预测为该类的数目。

True Positive（TP）：真正类。样本的真实类别是正类，并且模型识别的结果也是正类。

False Negative（FN）：假负类。样本的真实类别是正类，但是模型将其识别为负类。

False Positive（FP）：假正类。样本的真实类别是负类，但是模型将其识别为正类。

True Negative（TN）：真负类。样本的真实类别是负类，并且模型将其识别为负类。

准确率（Accuracy）：即所有的预测正确的占总的比重。

精准率（Precision）：即正确预测为正的占全部预测为正的比例。

召回率（Recall）：即正确预测为正的占全部实际为正的比例。

F1值：为准确率和召回率的调和平均数。

准确率是最常见的评价指标，就是被预测对的样本数目除以所有的样本数。在样本平衡的情况下，正确率越高，分类器性能越好。另外，在深度学习常常使用 Top 1 和 Top 5 来评价模型的性能。Top 1 指的是在预测标签中正确的类别落在分类概率最大就算对;Top 5 指的是在预测标签中，正确的类别落在前面5个最大的分类概率就算对。考虑到本实验类别较少，因此不执行 Top 5 正确率，换作执行 Top2正确率。

在Pytorch中也并没有提供现成的API来计算Top-K准确率，也只是提供了一个类似于get\_top\_k\_result功能的函数。因此Pytorch可以通过如下代码来实现Top-K准确率的计算：

import torch

logits = torch.tensor([[0.1, 0.3, 0.2, 0.4],

[0.5, 0.01, 0.9, 0.4]])

y = torch.tensor([3, 0])

def calculate\_top\_k\_accuracy(logits, targets, k=2):

values, indices = torch.topk(logits, k=k, sorted=True)

y = torch.reshape(targets, [-1, 1])

correct = (y == indices) \* 1

top\_k\_accuracy = torch.mean(correct) \* k

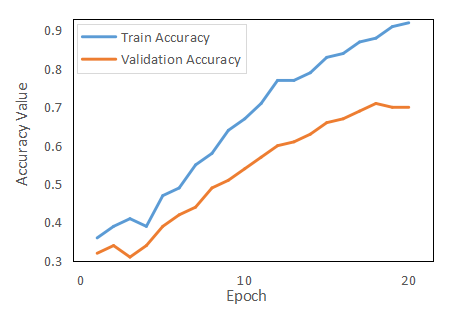
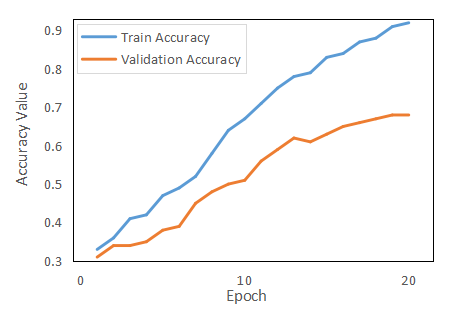
return top\_k\_accuracy

print(calculate\_top\_k\_accuracy(logits, y, k=2).item())# 1.0

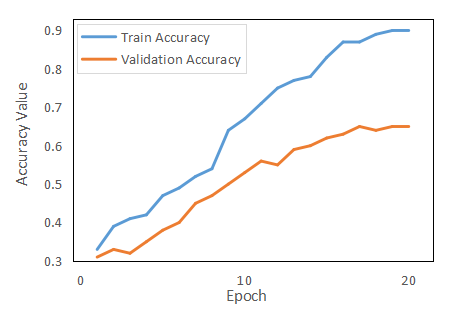
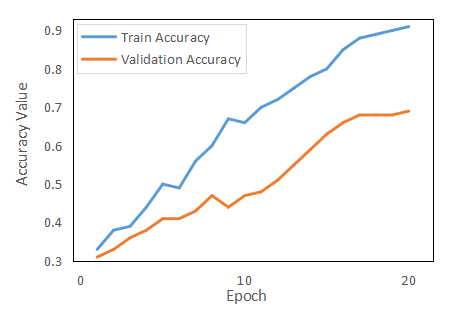
print(calculate\_top\_k\_accuracy(logits, y, k=1).item())# 0.5

### 3.3试验结果对比分析

本文通过Pytorch搭建深度学习模型实现沥青路面纹理的自动检测。在设置的初始参数相同的前提下，通过sgd、adagrad、rmosprop及adadelta优化器对模型分别进行优化。设置超参数如下：Epoch为1000，batch size为64，learning rate为0.001，weight decay为0.0001。并绘制性能曲线图，试验结果如下：

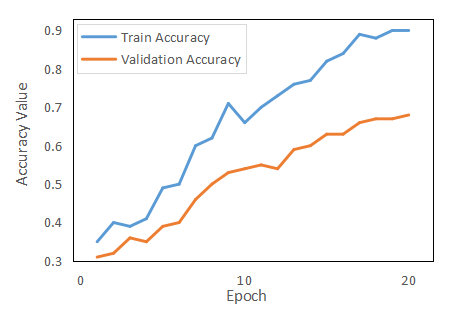
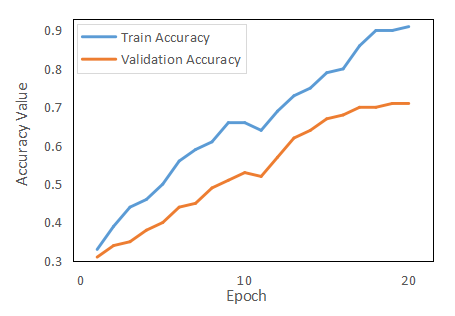
AC(SGD) AC(Adagrad)

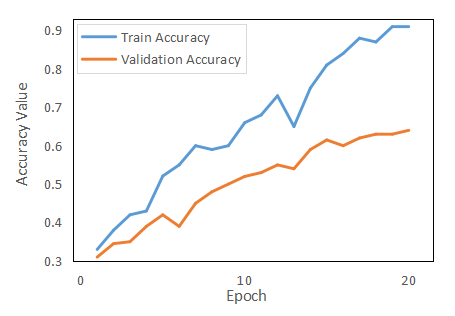
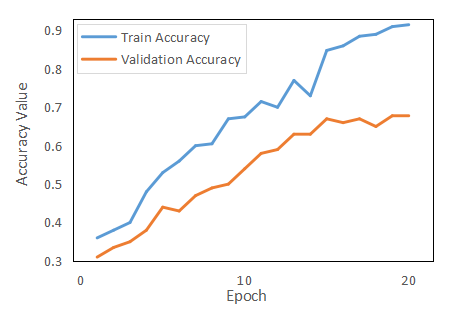
AC(Rmsprop) AC(Adadelta)

图8 模型使用各优化器下AC数据集准确率图

从图 可以看出，沥青路面纹理检测模型在训练AC数据集时，共训练了1000轮，在600轮左右达到收敛。其中，使用SGD优化器效果最佳，准确率最终稳定在0.7，F1，召回值，精准率及TOP2准确率分别达到。



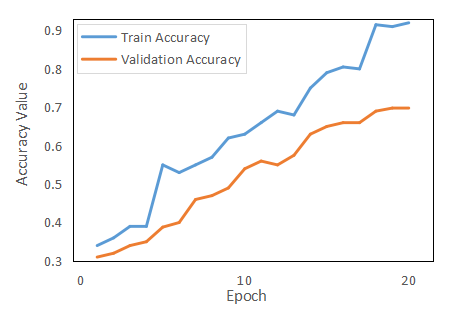
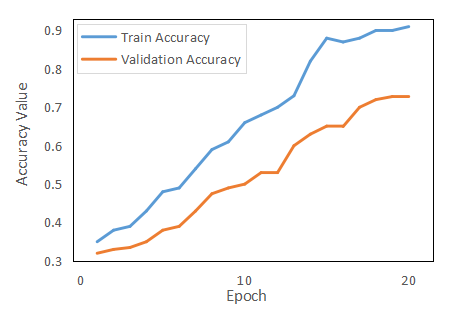
SMA(SGD) SMA (Adagrad)



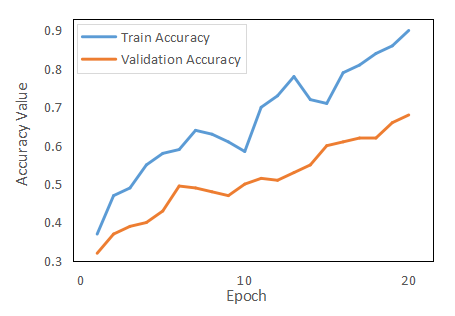
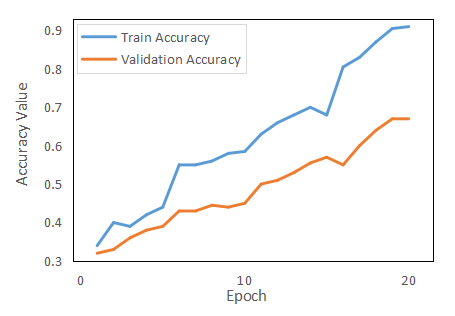
SMA (Rmsprop) SMA (Adadelta)

图9 模型使用各优化器下SMA数据集准确率图

从图 可以看出，沥青路面纹理检测模型在训练SMA数据集时，共训练了1000轮，在550轮左右达到收敛。其中，使用SGD优化器效果最佳，准确率最终稳定在0.7，F1，召回值，精准率及TOP2准确率分别达到。



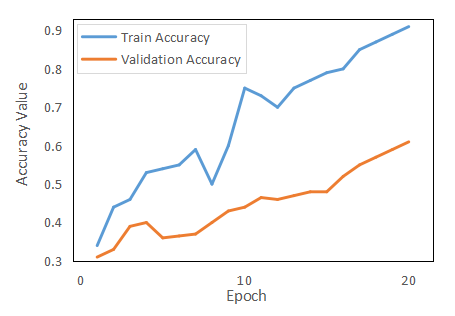
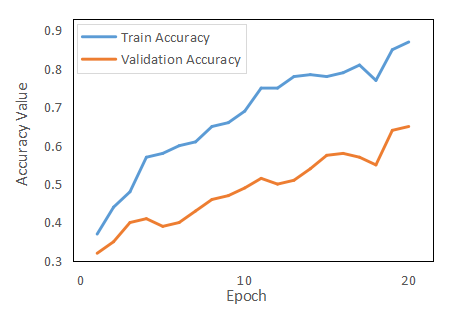
OGFC(SGD) OGFC (Adagrad)



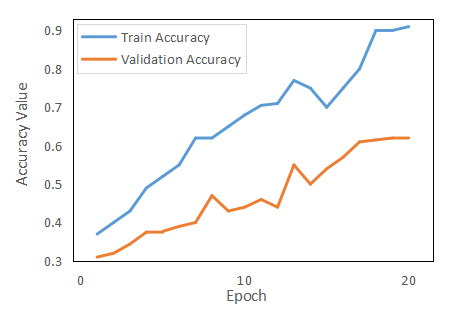
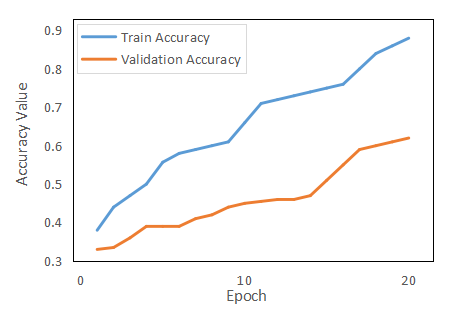
OGFC (Rmsprop) OGFC (Adadelta)

图10 模型使用各优化器下OGFC数据集准确率图

从图 可以看出，沥青路面纹理检测模型在训练OGFC数据集时，共训练了1000轮，在550轮左右达到收敛。其中，使用SGD优化器效果最佳，准确率最终稳定在0.7，F1，召回值，精准率及TOP2准确率分别达到。



ALL(SGD) ALL (Adagrad)



ALL (Rmsprop) ALL (Adadelta)

图11 模型使用各优化器下ALL数据集准确率图

从图 可以看出，沥青路面纹理检测模型在训练ALL数据集时，共训练了1000轮，在500轮左右达到收敛。其中，使用SGD优化器效果最佳，准确率最终稳定在0.7，F1，召回值，精准率及TOP2准确率分别达到。

由表3可知，AC，SMA，OGFC数据集的训练结果是完全优于ALL数据集的，主要是由相同级配沥青路面纹理图像所组成的训练集，一定程度上克服了其二维纹理图像差异较大，纹理参数值却相近的问题。其中OGFC和SMA比AC拥有更好的训练效果，原因是OGFC和SMA图像中灰度差异性更大，所以特征更为明显。模型使用SGD、Adagrad优化器的表现略优于使用Rmsprop、Adadelta优化器。

表3 四个数据集在不同优化器的模型检测结果

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 数据集名称 | 20轮训练结果(Validation Accuracy)TOP1 | | | | 20轮训练结果(Validation Accuracy)TOP2 | | | |
| SGD | Adagrad | Rmsprop | Adadelta | SGD | Adagrad | Rmsprop | Adadelta |
| AC | 0.70 | 0.68 | 0.65 | 0.69 | 0.85 | 0.84 | 0.80 | 0.83 |
| SMA | 0.71 | 0.68 | 0.68 | 0.64 | 0.88 | 0.85 | 0.87 | 0.79 |
| OGFC | 0.73 | 0.70 | 0.67 | 0.68 | 0.88 | 0.86 | 0.84 | 0.83 |
| ALL | 0.65 | 0.61 | 0.62 | 0.62 | 0.81 | 0.79 | 0.79 | 0.78 |