**2020机器学习**

**印花疵点分类**

**组员：**

须子逸 20210240350

李帅

李锦程 20210240213

朱柏霖

郭露

# 1 任务描述

基于机器学习模型对印花疵点数据集中标签为1、2、14的数据以及标签为1、2、4、14、20的数据分别做3分类及5分类任务；

利用ModelArts平台进行模型训练任务。

# 2 数据集介绍

# 3 数据预处理

数据预处理的目标是去除数据中的冗余信息，突出数据中的重要信息，即瑕疵。在印花疵点数据集中，已经注明了瑕疵所在的位置，并且可以作为已知信息用于分类任务，因此本文首先对图片进行裁剪，去除瑕疵位置以外的像素点，从而使分类模型专注于瑕疵所在的位置。

由于Bbox的尺寸各不相同，因此在裁剪之后需要对图片进行resize使得数据的尺寸一致，本文分别尝试了缩放和padding两种resize方式，其中缩放可以减少数据为0的像素点，使得瑕疵填满整张图片，但是会因此失去瑕疵区域的尺寸信息，而padding保留了尺寸信息。根据实验结果发现用padding的进行resize后模型的准确率较高，

由于数据集中某些类别的数量过少，因此我们通过旋转、翻转、改变亮度、增加噪点等方式生成新的图片，进行数据增强，扩充了数据集。实验结果表明数据增强可以有效提高准确率。

在印花疵点数据集中，对每个数据提供了瑕疵图和原图一对图片，为了去除背景冗余信息，本文使用了图片差分的方式，通过将瑕疵图和原图进行差分，去除瑕疵图中属于背景部分的像素，只保留瑕疵信息。

最后，本文也尝试了在差分前对瑕疵图和原图进行池化从而提取特征，本文分别对最大池化和平均池化后的数据集进行训练，通过实验得到了效果最好的预处理方式。

## 3.1 裁剪

在裁剪操作中，根据标注中的Bbox信息对图片对进行裁剪，去除瑕疵区域以外的像素，如图3-1所示：

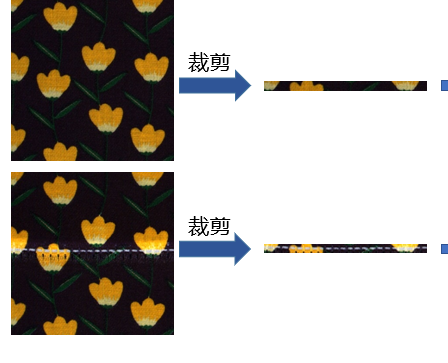


图3-1 裁剪

## 3.2 数据增强

在对图片进行裁剪之后，我们通过旋转随机角度，翻转，改变亮度等操作进行数据扩充。将每一个训练集中类别的图片扩充为600张，从而平衡样本数量。

## 3.3 池化

由于数据集中的图片对没有完全对其，我们尝试对图片进行池化后再进行差分，实验结果表明。。。

## 3.4 差分

对图片对进行差分后，去除了部分背景信息，只保留存在瑕疵的像素点，实验结果表明差分对分类准确率有明显提升。如图3-2所示：

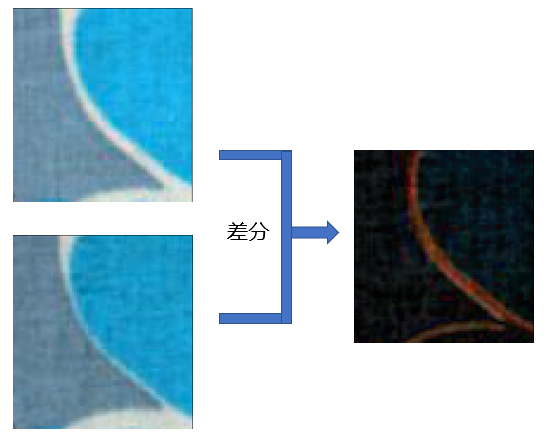


图3-2 差分

## 3.5 resize

将差分后的图片resize成统一大小有两种方法：缩放和padding，缩放通过插值将原图变形成为新的大小，该方法可以使得有效像素填充整个输入向量，但同时会失去图片原来的尺寸信息，如图3-3所示：

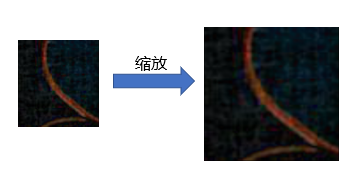


图3-3 缩放

而padding通过在周围填充黑色像素点从而适应形状，该方法保留了图片原来的尺寸信息，并且纹理不会发生变形。如图3-4所示：

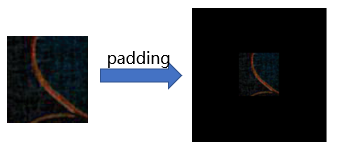


图3-4 padding

# 4 模型介绍及实验

## 4.1 决策树

# 4.2 SVM

# 4.3 逻辑回归

# 4.4 简单CNN

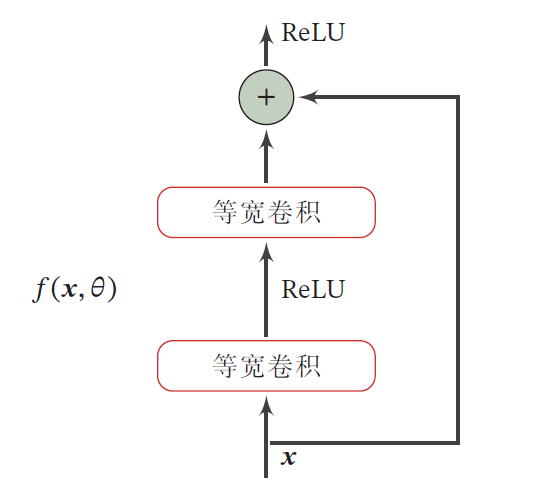
# 4.5 ResNet-18

残差网络(Residual Network,ResNet)通过给非线性的卷积层增加直连边（Shortcut Connection)的方式来提高信息的传播效率。

假设在一个深度网络中，期望一个非线性单元（一层或者多层的卷积层）去逼近一个目标函数。如果将目标函数拆分成两部分：恒等函数（Identity Function)和残差函数（Residue Function)

根据通用近似定理，一个由神经网络构成的非线性单元有足够的能力来近似逼近原始目标函数或残差函数，但实际中后者更容易学习。因此，原来的优化问题可以转换为：让非线性单元去近似残差函数

下图给出了一个典型的残差单元示例，残差单元由多个级联（等宽）卷积层和一个跨层的直连边组成，再经过ReLU激活后得到输出。



残差网络就是将很多个残差单元串联起来构成的一个非常深的网络。

# 本次实验中采用的图片尺寸为64\*64\*3，考虑到图片尺寸较小，实验中采用的残差网络模型为Resnet18模型，其网络结构如下图所示：实验中保持残差块的结构以及大小不变，对残差层之间的卷积层以及池化层结构进行调整，使其更适应数据尺寸。

为了减少计算量并保证图像信息的完整性，对第一层卷积层conv1做调整：卷积核大小缩小为3\*3，同时扩大步长为3；同时删去卷积层之后的最大池化层。

# 

**优化器选择：**实验过程中，分别选择了随机梯度下降（SGD）以及Adam优化器对网络参数进行优化，比较实验结果后发现，SGD算法迭代多次后在损失上依然波动，Adam优化器能够使曲线快速收敛，因此实验采用Adam优化器。

**超参数选择：**实验中超参数设置为，epoch\_times=10,lr(优化器学习率)=1e-3。实验以准确率作为评判分类准确率的主要标准，分别比较了6种数据处理方式下的损失函数曲线以及准确率指标。

**损失曲线分析：**实验在task1（3分类）以及task2（5分类）任务下的损失函数曲线如下图所示，可以观察到在3分类（左图），损失曲线在4次epoch时已经收敛；而在5分类（右图），损失函数在7次epoch左右时才完成收敛。

# 4.6 ResNet-50 on ModelArts

# 5 实验结果及分析

## 5.x 结果对比

三分类：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 预处理方式 | | | | | | 分类模型 | | | | | | |
| 差分 | 最大池化 | 平均池化 | 缩放 | 填充 | 数据增强 | 决策树 | 逻辑回归 | SVM | CNN | ResNet-18 | ResNet-50 |
|  |  |  |  | √ |  |  |  |  |  | 89.01% |  |
| √ |  |  | √ |  |  |  |  |  |  | 89.53% |  |
| √ |  |  |  | √ |  |  |  |  |  | 86.39% |  |
| √ |  |  |  | √ | √ |  |  |  |  | 73.30% |  |
| √ | √ |  |  | √ | √ |  |  |  |  | 60.73% |  |
| √ |  | √ |  | √ | √ |  |  |  |  | 62.83% |  |

五分类：

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 预处理方式 | | | | | | 分类模型 | | | | | | |
| 差分 | 最大池化 | 平均池化 | 缩放 | 填充 | 数据增强 | 决策树 | 逻辑回归 | SVM | CNN | ResNet-18 | ResNet-50 |
|  |  |  |  | √ |  |  |  |  |  | 81.29% |  |
| √ |  |  | √ |  |  |  |  |  |  | 80.61% |  |
| √ |  |  |  | √ |  |  |  |  |  | 79.25% |  |
| √ |  |  |  | √ | √ |  |  |  |  | 68.37% |  |
| √ | √ |  |  | √ | √ |  |  |  |  | 57.14% |  |
| √ |  | √ |  | √ | √ |  |  |  |  | 56.46% |  |

**Resnet18实验结果分析：**

Resnet-18在task1上的最高分类准确率为89.53%，在task2上的最高分类准确率为81.29%。通过对两表数据观察和比较可以发现，采用填充或者差分+缩放的数据预处理方式，随后使用Resnet18进行分类任务的分类准确率最高，采用差分+缩放以后的准确率次之。而在以上数据预处理方式的基础上加入数据增强步骤，反而会导致准确率的下滑。另一方面，观察损失函数以及测试集上变化趋势以后发现，在5次左右epoch时准确率达到最高值，之后准确率可能下滑，考虑到有可能存在训练集上过拟合问题。

优化方向：使用预训练的Resnet模型进行分类；加入Dropout、Batch Normalization等防止过拟合策略。