

场景分类报告

一、数据集

cvd12021-scene-classification 场景分类数据中训练集有 68059 张 100 个场景下的图片，测试集中有 9160 张图片。训练集中图片尺寸大小从 256*256 到 1000*1000，测试集中图片尺寸大小从 130*98 到 1000*1000。考虑到数据量大且种类多，选择人工提取特征来进行分类较为复杂且精度不高。因此，选用深度卷积神经网络(CNN)来完成分类任务。

二、深度卷积神经网络分类模型

卷积神经网络 CNN 由纽约大学的 Yann Lecun^[1]于 1998 年提出，其本质是一个多层感知机，CNN 采用的局部连接和权值共享的方式，减少了权值的数量使得网络易于优化；另一方面降低了模型的复杂度，也就是减小了过拟合的风险。

卷积神经网络一般由卷积层、激活函数、池化层和全连接层组成。卷积层由众多的卷积核组成，卷积核会以滑动的方式对输入范围的输入进行内积求和操作。卷积是一种局部操作，通过一定大小的卷积核作用于局部图像区域获得图像的局部信息。卷积层本质就是图像滤波器，而滤波的参数可以在训练中动态调整，除了可以学习到边缘滤波器，还可以学习到检测形状、颜色等信息。

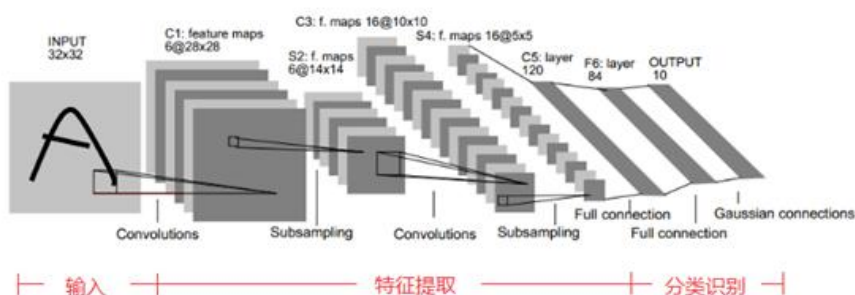


图 1 LeNet 卷积神经网络结构

卷积神经网络中的每一层传递都为矩阵相乘，导致网络的输出和输入一直为线性关系，使用激活函数引入非线性特性，使网络具有更好的特征表达能力。池化层一般分为最大池化、平均池化和随机池化。平均池化即对卷积核内的区域取平均特征值，可以更好提取背景特征。最大池化则取该区域的最大特征值，更利于留下纹理特征。随机池化按照与其值成正相关的概率值大小随机选择，在不让所有区域均取最大值的条件下可以最大化取到各个区域的最大值，泛化能力较强。全连接的核心操作是矩阵乘法，本质上是把一个特征空间线性变换到另一个特征空间，全连接层采用Softmax全连接，得到的激活值即卷积神经网络提取到的图片特征。

本次场景分类采用的方法是采用卷积神经网络来自动提取数据中的特征，然后使用全连接层对提取的特征进行分类。

三、主要工作

在 Pycharm 中使用 PyTorch 深度学习框架来实现场景分类，主要工作包括两大部分，一是数据加载、模型搭建、模型训练验证、测试集测试；二是尝试了不同模型对分类精度的影响、调整不同学习率来训练网络、数据增强对分类精度的影响。

- **网络模型**：搭建 AlexNet 卷积神经网络，使用 torchvision 中的 models 模块调用 ResNet18、ResNet50 和 efficientnet_pytorch¹ 模块中的 efficientnet-b0、efficientnet-b3 网络，修改网络模型中全连接层输出层为 100 个类。
- **数据集划分**：将训练数据划分 20% 为验证集，在训练时边训练边验证模型的训练效果；
- **数据加载**：使用 pandas 读取 csv 中的训练信息，获取训练数据的标签信息；
- **数据转换**：将读取的图片尺寸统一缩放到 (224, 224)，转化为 tensor 后做归一化处理；
- **损失函数**：选用交叉熵损失函数 CrossEntropyLoss 作为损失函数来判定实际的输出与期望的输出的接近程度；
- **优化器**：选用 Adam 优化器作为优化函数；

1. 不同的模型对精度的影响

表一 不同模型在测试集上的精度

模型	学习率	训练轮数	优化器	损失函数	测试精度
AlexNet	0.001	30	Adam	交叉熵	0.56277
ResNet18	0.001	30	Adam	交叉熵	0.73890
ResNet50	0.001	30	Adam	交叉熵	0.69432
EfficientNet-b0	0.001	30	Adam	交叉熵	0.72379
EfficientNet-b3	0.001	30	Adam	交叉熵	0.73198

2. 学习率调整

针对 ResNet18 网络尝试了不同的学习率进行训练，分别设置为 0.1、0.01、0.001 和 0.0001 四组不同的学习率。当学习率设置为 0.1 和 0.01 时，训练损失下降一点后突然增大或下降一点后不再变化；当学习率设置为 0.001 和 0.0001 时，模型精度分别为 0.001 和 0.0001，当学习率较小时，精度有所提高，但训练时间更长。学习率为 0.0001 时，ResNet18 训练损失变化如图 2 所示。

表 2 不同学习率的测试精度

模型	学习率	Epoch	测试精度
ResNet18	0.1	30	/
ResNet18	0.01	30	/
ResNet18	0.001	30	0.73890
ResNet18	0.0001	30	0.74818

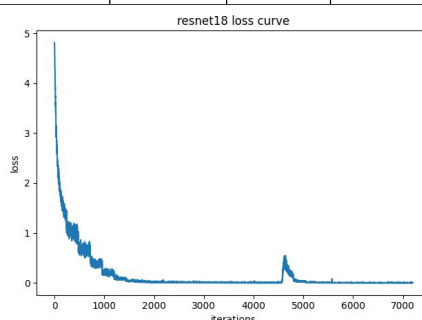


图 2 resnet18 训练损失函数

¹ <https://github.com/lukemelas/EfficientNet-PyTorch>

3. 数据增强

在数据转换 `data_transform` 中尝试使用随机裁剪、水平翻转、垂直翻转等数据增强手段，但是在训练中发现对精度提升效果不明显。

参考文献

[1] Lecun Y , Bottou L . Gradient-based learning applied to document recognition[J]. Proceedings of the IEEE, 1998, 86(11):2278-2324.