

青康科技

基于卷积神经网络（CNN）的垃圾分类模型 研究

作

者 王天泽

摘要

随着城市化进程的加快，垃圾问题日益严重。传统的垃圾分类方法存在效率低下、准确率不高等问题。本文提出了一种基于卷积神经网络（CNN）的垃圾分类模型，该模型能够自动识别并分类不同类型的垃圾。实验表明，该模型在垃圾分类任务中取得了较高的准确率，为垃圾处理提供了有效的技术支持。

关键词：卷积神经网络；垃圾分类；深度学习；图像识别

目录

摘要.....	- 1 -
一、 引言.....	- 3 -
(一) 背景.....	- 3 -
(二) 框架选取.....	- 3 -
二、数据集选取.....	- 3 -
三、数据集处理.....	- 7 -
四、模型训练.....	- 9 -
(一) 模型选取.....	- 9 -
(二) 优化器选取.....	- 9 -
(三) 损失函数选取.....	- 9 -
(四) 训练脚本编写.....	- 10 -
(五) 训练结果.....	- 12 -
(六) 模型测试.....	- 13 -
五、模型使用.....	- 13 -
六、结论与展望.....	- 15 -
参考文献.....	- 16 -

一、引言

（一）背景

随着人们生活水平的提高，城市垃圾产生量不断增加，垃圾分类已成为城市管理的重要任务。然而，传统的垃圾分类方法主要依赖于人工操作，存在劳动强度大、分类效率低、易受主观因素影响等问题。近年来，深度学习技术的快速发展为垃圾分类提供了新的解决方案。其中，卷积神经网络（CNN）因其强大的图像处理能力在图像识别领域取得了显著成果^[1]。因此，本文提出了一种基于 CNN 的垃圾分类模型，旨在提高垃圾分类的效率和准确率。

（二）框架选取

本文将采用由百度飞桨研发的 PaddlePaddle 深度学习框架（下文简称 Paddle 框架）。该框架支持传统深度学习框架的基本功能，如模型组网、显卡训练等。此外，该框架提供了高级 API，可以更加方便快捷地完成训练脚本的编写。

二、数据集选取

本文选取了飞桨 AI Studio 的“垃圾分类 40 种”数据集，如图 1、图 2 所示。该数据集的截取对象更偏向于“垃圾”而非“生活物品”，更符合垃圾分类模型的应用场景。



图 1 该数据集的“一次性快餐盒”数据



图 2 该数据集的“食用油桶”数据

并且，该数据集基本覆盖了常见的生活垃圾，其 40 个类别如表 1 所示。

表 1 该数据集的分类

类别
其他垃圾/一次性快餐盒
其他垃圾/污损塑料
厨余垃圾/茶叶渣
厨余垃圾/菜叶菜根
厨余垃圾/蛋壳
厨余垃圾/鱼骨
可回收物/充电宝
可回收物/包
可回收物/化妆品瓶
可回收物/塑料玩具
可回收物/塑料碗盆
可回收物/塑料衣架
其他垃圾/烟蒂
可回收物/快递纸袋

类别
可回收物/插头电线
可回收物/旧衣服
可回收物/易拉罐
可回收物/枕头
可回收物/毛绒玩具
可回收物/洗发水瓶
可回收物/玻璃杯
可回收物/皮鞋
可回收物/砧板
其他垃圾/牙签
可回收物/纸板箱
可回收物/调料瓶
可回收物/酒瓶
可回收物/金属食品罐
可回收物/锅
可回收物/食用油桶
可回收物/饮料瓶
有害垃圾/干电池
有害垃圾/软膏
有害垃圾/过期药物
其他垃圾/破碎花盆及碟碗
其他垃圾/竹筷
厨余垃圾/剩饭剩菜
厨余垃圾/大骨头
厨余垃圾/水果果皮
厨余垃圾/水果果肉

其数据总量较多，每个类别分配较平均，其每个类别的数量如表 2 所示。

表 2 该数据集的分类数量情况

类别	数量
其他垃圾/一次性快餐盒	232
其他垃圾/污损塑料	360
厨余垃圾/茶叶渣	377
厨余垃圾/菜叶菜根	726
厨余垃圾/蛋壳	321
厨余垃圾/鱼骨	399
可回收物/充电宝	347
可回收物/包	409
可回收物/化妆品瓶	342
可回收物/塑料玩具	299
可回收物/塑料碗盆	352
可回收物/塑料衣架	302
其他垃圾/烟蒂	269
可回收物/快递纸袋	216
可回收物/插头电线	647
可回收物/旧衣服	365
可回收物/易拉罐	299
可回收物/枕头	308
可回收物/毛绒玩具	540
可回收物/洗发水瓶	341
可回收物/玻璃杯	526
可回收物/皮鞋	372
可回收物/砧板	406
其他垃圾/牙签	75
可回收物/纸板箱	311
可回收物/调料瓶	436
可回收物/酒瓶	270

类别	数量
可回收物/金属食品罐	312
可回收物/锅	385
可回收物/食用油桶	341
可回收物/饮料瓶	255
有害垃圾/干电池	312
有害垃圾/软膏	381
有害垃圾/过期药物	427
其他垃圾/破碎花盆及碟碗	377
其他垃圾/竹筷	279
厨余垃圾/剩饭剩菜	385
厨余垃圾/大骨头	352
厨余垃圾/水果果皮	370
厨余垃圾/水果果肉	379
总和	14402

三、数据集处理

本文采用 Paddle 框架搭建数据集。在数据集初始化时，程序会顺序执行以下操作：

1. 获取数据集文件夹下的所有子文件夹
2. 对每个子文件夹，遍历每个图片，顺序进行如下操作：
 - a) 使用 PIL 打开图片
 - b) 重设图片大小为 256x256
 - c) 将通道维度置于最前，以适应 Paddle 框架。即将图片维度从(256, 256, 3)转为(3, 256, 256)
 - d) 将图片和标签存入对应列表

当外界需要获取指定索引的数据时，程序会按照对应索引，从列表中读取并返回数据。

数据集脚本的具体代码如下：

```
import paddle
import numpy as np
from PIL import Image
import os

def getAllPath(path):
    return [os.path.join(path, f) for f in os.listdir(path)]

class Dataset(paddle.io.Dataset):

    def __init__(self, mode='train'):
        super().__init__()
        self.data = []
        self.label = []
        for type_dir in getAllPath(f'./dataset/{mode}'):
            print('正在读取： ', os.path.basename(type_dir))
            for path in getAllPath(type_dir):
                img = Image.open(path)
                img = img.resize((256, 256))
                img = np.array(img, dtype=np.float32).transpose((2, 0, 1))
                self.data.append(img)
                self.label.append(int(os.path.basename(type_dir)))
        self.data = np.array(self.data, dtype=np.float32)
```

```
def __getitem__(self, idx):
    return self.data[idx], self.label[idx]

def __len__(self):
    return len(self.data)
```

四、模型训练

（一）模型选取

本文采用 ResNet (Residual Neural Network) 作为分类模型。ResNet 是深度学习中一种非常有效的卷积神经网络(CNN)结构,通过引入残差块(Residual Block)解决了深度神经网络训练中的梯度消失和模型退化问题。ResNet 的提出使得训练深度达到几百甚至上千层的神经网络成为可能,提高了模型的准确性,并在多项计算机视觉任务中取得了显著的性能提升。[2]

由于垃圾分类的任务相对简单,不需要过于复杂的模型,因此本文采用 ResNet-18(18 层的 ResNet 模型)进行分类任务,达到较好的性能与准确的平衡。

（二）优化器选取

在训练神经网络时,优化器的选择对模型的性能具有重要影响。优化器用于更新和计算影响模型训练和模型参数的梯度,以最小化损失函数。本文采用 Adam (Adaptive Moment Estimation) 作为优化器。

Adam 优化器将在梯度大时加速训练,梯度小时减速训练,而不是以固定的学习率训练。这将会大大加速训练过程,并且减少因训练集梯度问题而产生的训练问题,增加准确率。[3]

（三）损失函数选取

在机器学习和深度学习中,损失函数用于量化模型预测与真实标签之间的差异。选择一个合适的损失函数对于训练一个高效且准确的模型至关重要。本文采

用交叉熵损失函数（Cross-Entropy Loss Function）进行训练。

在多分类问题中，通常有多个类别（假设为 C 个），模型的输出是一个 C 维的向量，每个元素表示样本属于对应类别的概率，其表达式为：

$$L = - \sum_{i=1}^C y_i \log(p_i)$$

其中， C 表示样本个数， y_i 是第 i 个样本的标签（0 或 1）， p_i 是模型输出的 C 维向量的第 i 维。

交叉熵损失函数常用于分类问题，衡量预测概率分布与真实概率分布之间的差异。而垃圾分类是典型的分类问题，适用于交叉熵损失函数。

使用交叉熵损失函数的好处之一是它能够有效地处理类别不平衡的问题，因为它关心的是预测概率与真实概率之间的差异，而不是仅仅关注是否预测正确。此外，交叉熵损失函数在梯度下降过程中表现良好，能够提供稳定的梯度更新，有助于模型的快速收敛。^[4]

（四）训练脚本编写

Paddle 框架提供的高级 API 极大简化了训练脚本的编写。因此，该脚本只需要设置优化器、损失函数和评价指标（准确率）并调用 `model.fit` 方法即可进行训练。另外，该脚本还通过回调函数，实现损失和准确率的记录，并写到一个 `csv` 文件中。

该脚本的代码如下：

```
import paddle
import csv
from paddle.metric import Accuracy
from paddle.callbacks import ProgBarLogger, Callback
from dataset import Dataset

train_dataset = Dataset()

net = paddle.vision.models.resnet18(num_classes=40)
```

```
model = paddle.Model(net)

# 自定义一个回调函数来记录 loss 和 acc
class CSVLogger(Callback):
    def __init__(self, csv_file):
        super().__init__()
        self.csv_file = csv_file
        self.file = None
        self.csv_writer = None

    def on_train_begin(self, logs=None):
        self.file = open(self.csv_file, 'w', newline='')
        fieldnames = ['epoch', 'loss', 'acc']
        self.csv_writer = csv.DictWriter(self.file, fieldnames=fieldnames)
        self.csv_writer.writeheader()

    def on_epoch_end(self, epoch, logs=None):
        row_dict = {'epoch': epoch + 1, 'loss': logs.get('loss')[0], 'acc':
logs.get('acc')}
        self.csv_writer.writerow(row_dict)

    def on_train_end(self, logs=None):
        self.file.close()

# 实例化 CSVLogger
csv_logger = CSVLogger('training_history.csv')
```

```
# 准备模型训练
model.prepare(paddle.optimizer.Adam(parameters=model.parameters()),
              paddle.nn.CrossEntropyLoss(),
              Accuracy())

# 使用 fit 方法并传入自定义的 CSVLogger
model.fit(train_dataset,          epochs=50,          batch_size=64,
          callbacks=[ProgBarLogger(verbose=1), csv_logger])

# 模型保存
model.save('./output/model')
```

（五）训练结果

经过 50 个 epoch 的训练后，模型的准确率稳定在 97%左右，损失也逐渐趋于平稳，如图 3 和图 4 所示。

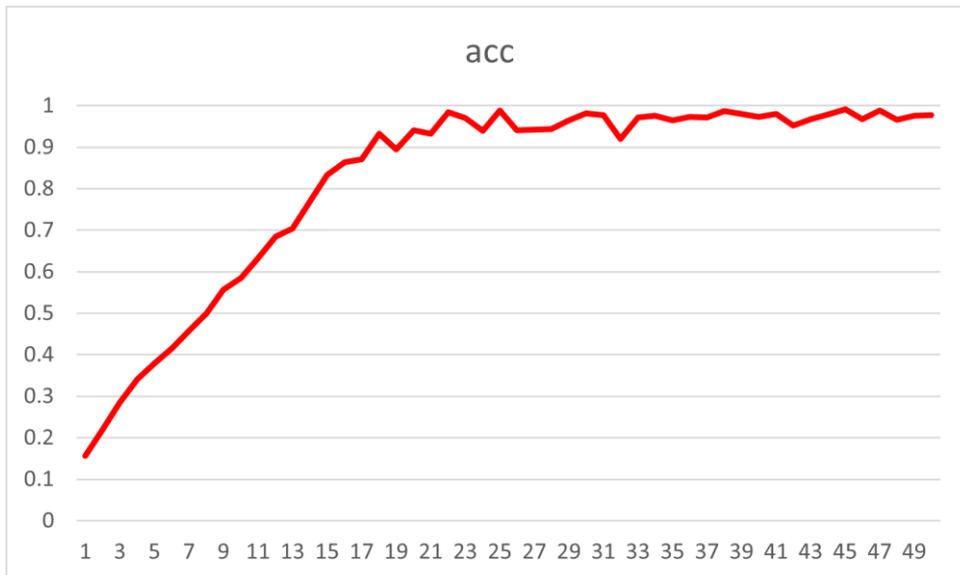


图 3 模型准确率随 epoch 的变化图

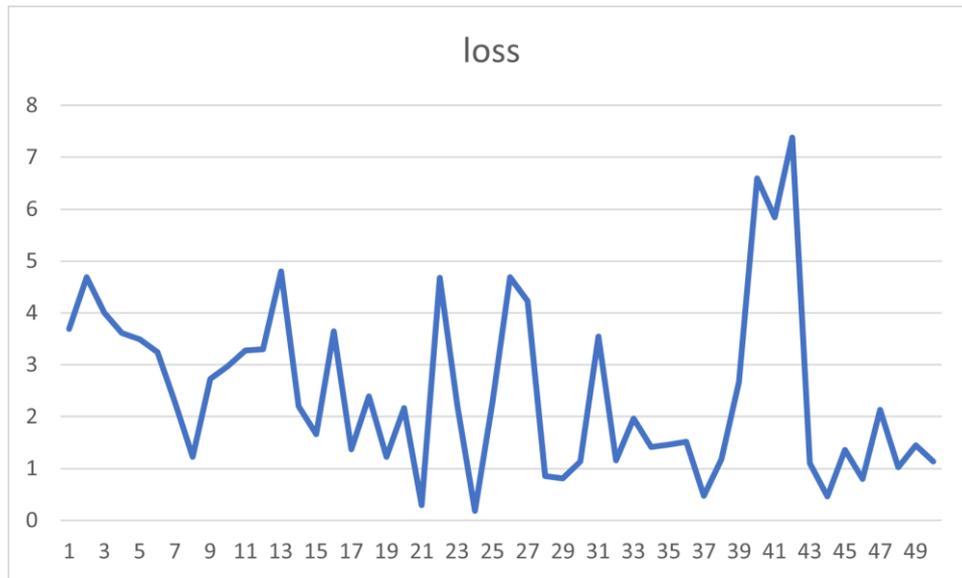


图 4 模型损失随 epoch 的变化图

(六) 模型测试

该模型在测试集上的测试结果也非常好，其损失达到了 9.6559×10^{-6} ，准确率达到 99.25%。可以看到，这个模型的分类是比较准确的，拟合性较高。

五、模型使用

本文提供了简单的模型使用程序，将会读取指定文件，并输出分类结果。其代码如下所示。

```
import paddle
from PIL import Image
import numpy as np
import json

# 初始化模型
net = paddle.vision.models.resnet18(num_classes=40)
model = paddle.Model(net)
```

```
# 加载模型
model.load('output/model')

# 加载键值对
with open('dataset/garbage_dict.json', 'r', encoding='utf-8') as f:
    mapping = json.load(f)

# 加载图片
path = 't3.jpeg'
img = Image.open(path)
img = img.resize((256, 256))
img = np.array(img, dtype=np.float32).transpose((2, 0, 1))[np.newaxis, :]

# 预测
pred = model.predict_batch(img)[0]
label = int(pred.argmax())
print(mapping[str(label)])
```

本文使用模型给出了三个网络图片的分类，如图 5、图 6、图 7 所示。



图 5 预测结果：厨余垃圾/水果果皮



图 6 预测结果：有害垃圾/过期药物



图 7 预测结果：其他垃圾/竹筷

可以看到，本模型准确地给出了这些图片所对应的垃圾分类。

六、结论与展望

本文提出了一种基于 CNN 的垃圾分类模型，并通过实验验证了其有效性。该模型能够自动识别并分类不同类型的垃圾，为垃圾处理提供了有效的技术支持。实验表明，该模型在垃圾分类任务中取得了较高的准确率，为垃圾处理提供了有效的技术支持。在应用方向上，可以将此模型应用于智能分类垃圾桶、垃圾分拣装置等，实现准确高效的垃圾分类。

参考文献

- [1] 黄琳轩. 基于卷积神经网络的垃圾自动分类算法[J]. 科技与创新, 2024, (10):70-72. DOI:10.15913/j.cnki.kjycx.2024.10.016.
- [2] 孙毅, 吴斯曼, 方伟, 等. 基于 ResNet 的安全监控目标检测[J/OL]. 集成技术 :1-10[2024-06-06]. <http://kns.cnki.net/kcms/detail/44.1691.t.20240527.1512.004.html>.
- [3] 张波, 肖杰. 深度学习模型训练的优化器实验设计[J]. 电子制作, 2024, 32(02):114-117. DOI:10.16589/j.cnki.cn11-3571/tn.2024.02.023.
- [4] 黄辉城, 李建新. 基于深度视觉的智能卸垛机快递箱边界检测系统[J]. 光学技术, 2024, 50(02):220-227. DOI:10.13741/j.cnki.11-1879/o4.2024.02.018.