

# 基于 PyTorch 和 CNN 的图片验证码识别

## Image Verification Code Recognition Based on PyTorch and CNN

何金涛<sup>1</sup> 何金波<sup>2</sup> 鲁智谦<sup>3</sup>

Jintao He<sup>1</sup> Jinbo He<sup>2</sup> Zhiqian Lu<sup>3</sup>

1.河海大学常州校区 物联网工程学院  
中国·江苏 常州 213022;

2.南昌航空大学 航空制造工程学院  
中国·江西 南昌 330063;

3.河海大学常州校区 机电工程学院  
中国·江苏 常州 213022

1. Thing Networking Engineering College  
Changzhou Campus of Hohai University,  
Changzhou, Jiangsu, 213022, China;

2. College of Aeronautical Manufacturing  
Engineering Nanchang Hangkong University,  
Nanchang, Jiangxi, 330063, China;

3. Mechanic and Electronic Engineering College  
Changzhou Campus of Hohai University,  
Changzhou, Jiangsu, 213022, China

**【摘要】**论文提出了基于深度学习框架 PyTorch 和卷积神经网络 (Convolutional Neural Networks, CNN) 的图片验证码识别方法, 先对验证码图片进行简单的预处理后, 搭建了四组卷积神经网络结构, 对网站上收集到的验证码图片进行训练。

**【Abstract】**This paper proposes a picture verification code recognition method based on deep learning framework PyTorch and Convolutional Neural Networks (CNN). After simple preprocessing of the verification code images, four groups of convolutional neural network structures are built to train the verification code pictures collected on the website.

**【关键词】**验证码; PyTorch; 卷积神经网络

**【Keywords】**verification code; PyTorch; convolutional neural network

**【DOI】**10.36012/etr.v2i2.1157

## 1 引言

本文利用新兴的深度学习框架 PyTorch, 并结合卷积神经网络 (CNN) 搭建的模型进行训练, 对于收集到的由数字和字母 (区分大小写) 组成的复杂的图片验证码的识别取得了较为良好的效果<sup>[1]</sup>。

## 2 PyTorch 简介

PyTorch 是 Torch 在 Python 上的衍生, 与之前大多数采用静态计算图的开源框架比如 TensorFlow、Caffe 不同, PyTorch 采用了动态计算图结构构建神经网络<sup>[2,3]</sup>。

## 3 CNN 模型搭建

卷积神经网络 (CNN) 的一般结构如下: ①输入层, 用于数据的输入。②卷积层: 使用适当大小的卷积核进行特征提取与特征映射。③激励层: 用于非线性映射, 这里使用 torch.nn 中的 ReLU 函数实现。④池化层: 对前面得到的特征映射图进行降维操作减小拟合, 本实验中采用的是最大池化层。⑤全连

接层: 对特征进行重新拟合, 减少信息量的缺失。⑥输出层: 目标结果输出<sup>[4]</sup>。在本实验中, 在池化层之后另外加上了批规范化层, 对数据进行泛化, 用于加快模型的收敛速度。此模型结构代码如下:

```
self.comv = nn.Sequential(  
    nn.Conv2d(3, 16, 3, padding=(1, 1), #num_batch*3*120*40  
    nn.MaxPool2d(2, 2)  
    nn.BatchNorm2d(16),  
    nn.ReLU(),  
)
```

其中 nn.Sequential() 可以看作是神经网络模块的一个有序容器, 这样有利于快速搭建神经网络结构。num\_batch 表示每次放入网络的数据个数即图片数量, 本实验网络的输入为长为 120 像素, 宽为 40 像素, RGB 三通道的验证码图片, 通过第一个卷积层, 保持输出的图片大小不变, 输入通道数为 3, 输出通道数为 16, 卷积核大小此处设为 3×3, 随后通过一个最大池化层, 对应的窗口大小为 2×2, 步长为 3, 然后在通

过一个批规范化层以及一个激励层,这样便得到了第一层输出为 16×60×20 的张量。同样的,经过上述类似的 4 组结构的卷积后,通过一个全连接层便为模型的输出。

通过全连接层的映射成为一个[num\_batch,num\_class\*num\_char]的张量输出,这里的 num\_class 和 num\_char 分别表示字符的种类和一张图片中的字符数量,分别为 62 和 4。

## 4 加载数据集

### 4.1 数据集

本实验所收集的数据集来自 2019 年全国高校计算机能力挑战赛官方提供的 5000 张验证码图片,如图 1 所示。



图 2 验证码 4Jy3

### 4.2 Dataset

PyTorch 里的 Dataset 是数据集的封装,可以高效方便地读取数据,代码所示:

```
source += [chr(i) for i in range(97, 97+26)]
source += [chr(i) for i in range (65,65+26)]
alphabet = ''.join(source)
class CaptchaData(Dataset):
    def __init__(self, data_path, num_class=62, num_char=4,
                transform=None, target_transform=None, alphabet=alphabet):#62,4)
        super(Dataset, self).__init__()
```

## 5 网络训练流程

神经网络训练的流程可以分为如下 4 个主要流程:①定义网络,本实验则采用卷积神经网络;②定义优化器和损失函数,此处使用 torch 库里的 torch.optim.Adam () 和 nn.MultiLabelSoftMarginLoss()分别作为优化器和损失函数;③遍历数据集 dataloader,每次取一个 num\_batch 放在 GPU 或 CPU 上训练,计算损失函数 loss,本实验中, batch\_size 取 128, epoch 取 200,学习效率取 0.001。④训练集跑完一个 epoch,计算测试集的准确率,用来评估模型。这里需要自己定义计算准确率的函数作为模型评估的指标,代码如下所示:

```
def test_acc(output, target):
    output, target = output.view(-1, 62), target.view(-1, 62)
    output = nn.functional.softmax(output, dim=1)
    output = torch.argmax(output, dim=1)
    target = torch.argmax(target, dim=1)
    output, target = output.view(-1, 4), target.view(-1, 4)
    correct_list = []
    for i, j in zip(target, output):
        if torch.equal(i, j):
            correct_list.append(1)
        else:
            correct_list.append(0)
    accuracy = sum(correct_list) / len(correct_list)
    return accuracy
```

## 6 实验结果及分析

训练完参数后,对新的 5000 张验证码图片进行测试,测试部分结果如图 2 所示。

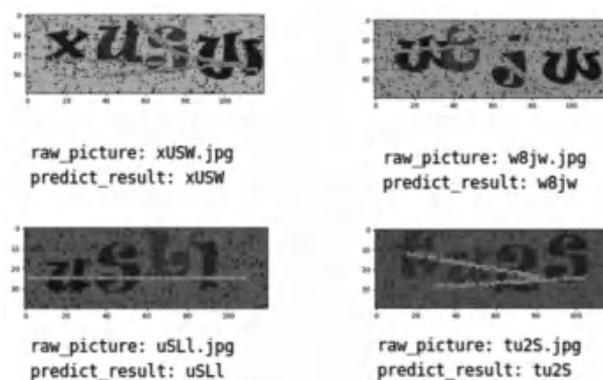


图 2 验证码识别效果

最后的结果表明,用此模型进行该类型的验证码进行识别,能达到 96%左右的正确率。

## 7 结语

本文针对 2019 年全国高校计算机能力挑战赛官网提供的验证码进行分析,提出了基于 PyTorch 和 CNN 的验证码识别算法,由于该类型的验证码图片中的字符间存在粘连,而且相同字符写法还不同,图片背景噪声与字符相融合,所以人眼很难识别,故本实验利用 PyTorch 深度学习框架,结合 CNN 卷积神经网络对数据集进行训练,结果表明能达到较高的识别正确率,测试集的正确率能够达到 96%,当然,本实验也有需要改进的地方,在图片预处理方面,本实验并未做过多操作,如果在本套验证码图片上进行旋转,倒置,达到扩充数据集的目的,预计结果会更好。

### 参考文献

[1]晋大鹏,张天心,刘涛.基于 Python 和 CNN 的验证码识别[J].软件工程,2019,22(6):1-4.  
[2]张涛,张乐乐.基于卷积神经网络的图片验证码识别[J].电子测量技术,2018,41(14):83-87.  
[3]连晓岩.基于图像识别和神经网络的验证码识别[C]//中国自动化学会智能自动化专业委员会.2011 年中国智能自动化学术会议论文集(第一分册).中国自动化学会智能自动化专业委员会:中国自动化学会智能自动化专业委员会,2011.  
[4]徐星,宋小鹏,杜春晖.基于深度学习的验证码图像识别[J].测试技术学报,2019,33(2):138-142.