# 验证码识别报告

为保护隐私，涂去作者信息。


## 摘要

*本项目要求设计一个算法，基于给定的 3000 张 6 字符训练图片（训练集），识别给定的 300 张 4 字符测试验证码图片（测试集），并给出识别准确率。上述验证码均由验证码生成器生成，且每一张验证码的颜色、旋转角度、噪点密度等均可能不同。本文设计并训练了一个 6 层卷积神经网络，该网络含有 2 个卷积层、2 个最大池化层、1 个全连接层和一个 softmax 层。该卷积神经网络是利用 tensorflow [1] 搭建的，使用一台 GTX1080TI 经 2000 步训练，用时 45 秒训练而成。最终测试集单字符识别成功率为 95.58%（1147/1200）。并可通过 GUI 界面实现交互式识别操作。经统计，在识别错误的 53 幅单字符图片中，绝大部分是 0 与 O、Q 与 O、1 与 I、2 与 Z 等一类即使是由人类肉眼识别也容易出现错误的字符组合。*


## 1.研究现状及主要研究方法

对于字符的识别，有很多人工设计的基于启发式的方法，如结构模式识别等，这一类方法高度依赖于人工设计的特征识别方法，且鲁棒性不强，一些常见的旋转、尺度变换、噪声等干扰就会大大地影响识别的成功率。另外，也有一些基于统计模式识别的方法，如模版匹配，这种方法并不需要特征提取过程，直接使用字符的图像作为特征，与字典中的模版进行比较，相似度最高的模版类即作为识别结果，这种方法简单易行，可以并行处理，但是一个模版只能识别同样大小、同种字体的字符，对于旋转、笔画粗细等适应能力不强。另一种常用的统计模式识别方法是先使用特征变换方法，将图像变换为某种鲁棒性强的特征（如 SIFT[2]），然后以特征构成码本，给定一副输入图像，先计算其特征，然后使用基于特征的模版匹配方法得出识别结果，这一类方法要计算速度快，且类似 SIFT 的特征有很强的鲁棒性，但是当类别数目增大时，存储复杂度和计算复杂度也大大地提高。

随着计算能力的提高，深度学习技术迅猛发展。现在越来越多的视觉任务，都使用深度学习的方法进行处理。卷积神经网络（CNN）可以说是一种为视觉任务而生的深度网络结构，最早使用 CNN 进行字符识别的是 Y. LeCun 等人 [3]，2006 年，有“神经网络之父”之称 Hinton 使用多层堆叠的玻尔兹曼机进行无监督地预训练并使用标签进行微调后，在 MNIST [4]数据集的测试集上达到 1.25%的错误率，虽然 MNIST 数据集是一个非常简单的数据集，但这种方法开启了深度学习的研究热潮，目前，在 MNIST 数据集识别错误率最低的是 Wan 等人在 ICML 的一篇文章 [5]提出的，他们在训练网络的时候使用了 dropout [6]的技巧，将识别错误率降低至 0.21%，考虑到 MNIST 数据集中的一些人类都难以识别的图片，深度神经网络的学习能力之强可见一斑。

## 2.本文研究方法

### 2.1 数据处理

#### 2.1.1 滤波及去黑边

由于给定的验证码图片有黑色边界和一些随机噪点，为防止这些无关因素对网络的影响，故本文采取一种启发式的滤波和去黑边方法。对于黑边，多字符图片

左右两侧的黑边在图片分割时可以直接分割掉，上下两条黑边则以各自相邻的行像素条代替；至于噪声，经观察，大部分噪声点的像素值集中在 210~220 之间，受此启发，将像素值在此区间的像素点的像素值改为各通道的背景像素值，RGB 通道分别为 240、248、255，Figure 1 (a),(b)分别为滤波和去黑边前后的两幅图片，由此可见，滤波效果很好。

(a) (b)

Figure 1 :图片的滤波操作(a)滤波前;(b)滤波后

### 2.1.2 字符分割

本项目所给的验证码图片分为两部分，一部分是 6 字符的彩色验证码图片，即训练集，另一部分是 4 字符的彩色验证码图片，即测试集。为完成识别任务，首先需要将多字符图片分割为单字符图片，即把每幅图片分割为 28*18*3 的尺寸，经多次尝试，对于训练集和测试集的字符分割如 Table 1 所示。

Table 1 字符分割

| 字符图片顺序 | 训练集 | 测试集 |
|---|---|---|
| 1 | 7:24 | 7:24 |
| 2 | 25:42 | 25:42 |
| 3 | 43:60 | 43:60 |
| 4 | 64:81 | 64:81 |
| 5 | 82:99 | - |
| 6 | 100:117 | - |

表 1 中训练集和测试集两栏下的数字表示当前编号图片在原图片中的像素起始和终止边界。Figure 2、Figure 3 分别是 6 字符、4 字符的一个分割示例，可见，分割效果较好。

Figure 2 : 4 字符图片分割效果图

Figure 3 : 6 字符图片分割效果图

### 2.1.3 二值化

另外，如前文所述，这些图片的颜色变化较多，为简化网络结构，本文最终选择了使用二值化图片进行训练和测试，首先选择图片的第三个通道，然后选择一个阈值(threshold=181），将像素值大于这个阈值的像素点置为 0，其他置为 1，二值化后的结果如 Figure 4 所示。

Figure 4 : 6 字符图片分割效果图

## 2.2 卷积神经网络

### 2.2.1 网络结构

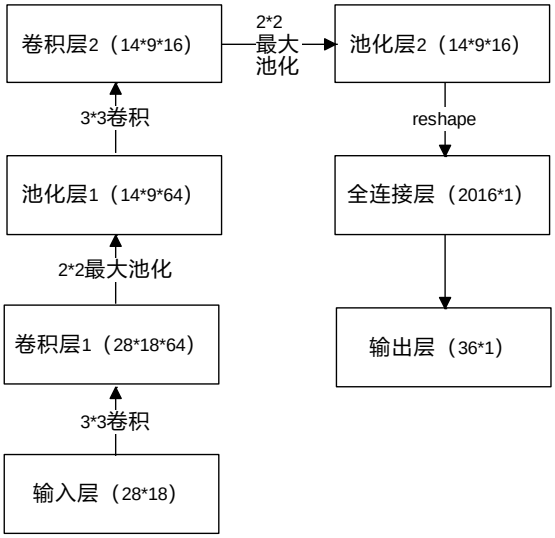


Figure 5： 卷积神经网络结构

本文中使用的卷积神经网络结构如 Figure 5 所示，该网络是一个 6 层的卷积神经网络。首先，输入二值化后的图片，然后进行 3*3 补零卷积，步长为 1，卷积核数目为 64，故卷积层 1 大小为 28*18*64，卷积操作后，使用 2*2 最大补零池化，步长为 2，故池化层 1 的尺寸为 14*9*64，后续的卷积层 2、池化层 2 类似。池化层 2 的输出被拉成一个 2016 维的列向量，即全连接层，

全连接层之后接一个 softmax 层用于输出各类别的概率。

### 2.2.2 网络训练

本文的卷积神经网络使用 tensorflow 搭建和训练，整个训练过程中需要调节的参数有学习率、训练步数、批尺寸（batch_size）等。在实际训练过程中，为防止过拟合，将训练集人为地划分为训练集和验证集，取训练集的后 3000 张图片作为验证集。以提高验证集和训练集的识别准确率为目标，进行调参，最终选择批尺寸为 50，训练步数为 2000，初始学习率为 0.001，并使用学习率指数衰减 [7]，衰减系数为 0.96，衰减间隔为 100 轮，即每训练 100 步学习率乘于 0.96。在优化算法方面，选择 Adam [8]算法，该算法同时具有 AdaGrad 和 RMSProp 算法的高效性和空间复杂度低等优点。

## 3. 实验结果

本文最终训练确定的网络，在给定的测试集上，单字符识别准确率达到 95.58% 。Figure 6 是训练好的网络在 1200 幅单字符识别测试中误识别的 53 幅图片。由 Figure 6 不难看出，这些识别错误的字符中，有很大一部分是 I 与 1，Q 与 O，0 与 O 之类的错误。如第一行最后两个字符，即使是人工识别也有可能犯错。为说明这些错误，将部分错误图片作为输入，进行前向运算并将 softmax 层输出排序，并取概率最大的两个结果列于 Table 2 中。由 Table 2 不难看出，这一类错误识别的字符，其 Top2 概率非常接近（表中几个相等的概率值是由于精度不够，实际运算时小数点后有 15 位），且对于这些误识别的结果，其第二大概率对应字符即其正确结果。全部的 53 个字符的 Top2 概率见附件 static.txt。

Table 2 概率 Top2 及其对应字符

| 原字符 | 识别结果 | 概率 | 次最大结果 | 概率 |
|---|---|---|---|---|
| 0 | O | 0.9997 | 0 | 0.9996 |
| S | 5 | 0.9993 | S | 0.9613 |
| 1 | I | 0.9999 | 1 | 0.9999 |
| S | 5 | 1.0000 | S | 1.0000 |
| O | 0 | 1.0000 | O | 0.9999 |
| 2 | Z | 0.9774 | 2 | 0.9562 |
| 1 | I | 1.0000 | 1 | 1.0000 |
| I | 1 | 1.0000 | I | 1.0000 |
| N | M | 1.0000 | N | 1.0000 |
| Z | 2 | 1.0000 | Z | 1.0000 |

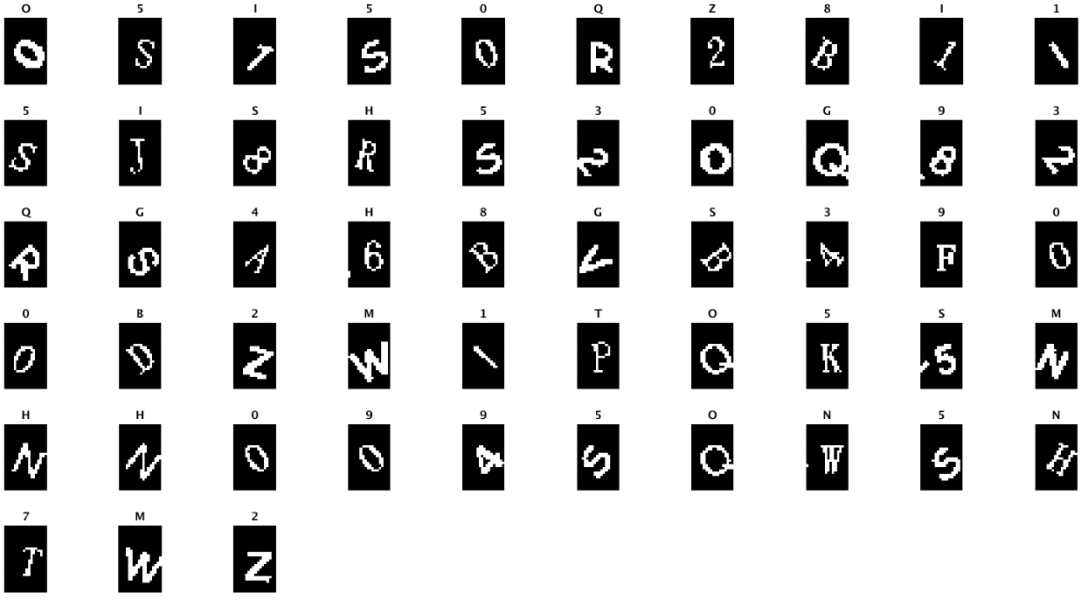

Figure 6：识别错误的 53 个字符

为说明本文训练所得网络泛化能力强、具有实用性，利用所给验证码生成器生成了 500 张 6 字符验证码图片，旋转角度从 25-75 度不等，保存至 test_image6 文件夹中，并使用本文设计的 GUI 测试其识别效果，如


Figure 7 所示，可见，识别效果较好。

Figure 7： 新生成 6 字符验证码识别结果

## 4. 总结

本文设计的 CNN 网络在单字符识别正确率上达到了一个较高的水平，且网络结构简单，识别速度较快。另外，相对于传统的基于特征匹配的方法，此算法不需要特征提取，且存储复杂度低。但是，深度神经网络有其通病，那就是调参复杂且繁琐，另外本文设计的网络只是最传统的 CNN 结构，在训练时也没有使用 dropout、BatchNormalization 等可能优化结果的技巧，所以在识别错误率上没有达到 MNIST 集的 0.23%那种程度。虽然 MNIST 集的难度远低于本任务，但那些技巧值得去尝试。

## 5. 参考文献


[1]. Martín Abadi, Ashish Agarwal, TensorFlow: Large-scale machine learning on heterogeneous systems,2015. Software available from tensorflow.org.

[2]. D. Lowe, Object recognition from local scale-invariant features, in: ICCV, 1999.

[3]. Y. LeCun and Y. Bengio: Convolutional Networks for Images, Speech, and Time-Series, in Arbib, M. A. (Eds), The Handbook of Brain Theory and Neural Networks, MIT Press, 1995

[4]. Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.

[5]. Wan, L., Zeiler, M., Zhang, S., LeCun, Y., and Fergus, R. (2013). Regularization of neural networks using dropconnect. In Proc. International Conference on Machine learning (ICML'13).

[6]. G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov. Improving neural networks by preventing co-adaptation of feature detectors. CoRR, abs/1207.0580, 2012

[7]. TensorFlow:实战Google深度学习框架/才云科技Caicloud，郑泽宇，顾思宇著.—北京：电子工业出版社，2017.3

[8]. Kingma, Diederik and Ba, Jimmy. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980, 2014.