An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition

Baoguang Shi, Xiang Bai and Cong Yao
School of Electronic Information and Communications
Huazhong University of Science and Technology, Wuhan, China
{shibaoguang,xbai}@hust.edu.cn, yaocong2010@gmail.com

论文地址：https://arxiv.org/abs/1507.05717

开源代码：https://github.com/bgshih/crnn

1. 摘要

基于图片的序列识别是计算机视觉领域一个长期研究的课题，其中非常重要且有挑战性的就是场景文本识别问题。本文针对这个问题，提出了一个新的神经网络结构，即 CRNN，它将特征提取、序列模型化以及转录都集成到一个统一的框架中。对比之前的工作，它有以下四个特点：

1. CRNN 是一个**端到端可训练**的网络，而之前的大部分算法都是分开训练和调试的；
2. 可以**处理任意长度的序列**，不需要进行字符分割或者水平尺度归一化；
3. 它不受任何预定义词汇的限制，在无词汇和基于词汇的场景文字识别任务中都取得了出色的表现；
4. CRNN 是一个有效且较小的模型，非常适用现实生活里的应用场景。

实验的数据集包括了 IIIT-5K，Street View Text 以及 ICDAR，在这些数据集上 CRNN 的性能都超过了之前的工作。

2. 简介

基于图片的序列识别问题有这几个特点：

1. 对比普通的目标检测问题，序列识别一般需要识别出一堆的物体标签，而不是仅仅单个物体的；

2. 目标序列的长度是不固定的，可变的，而DCNN（Deep Convolutional Neural Network）只能处理固定长度的输入和输出，所以没办法解决这个问题；

目前也有一些工作将 DCNN 应用于这个问题，主要是两个方面的做法：

1. 第一种是先训练一个检测器来检测每个字符，然后用 DCNN 来识别。但需要一个性能很好的检测器并且分割每个字符出来；
2. 第二种是将这个问题转换为图片分类问题，给每个英文单词分配一个类标签 （总共有 90k 个单词）。结果就是训练得到一个类别数量很大的模型，而且很难应用到其他类型的序列识别问题，比如中文、音乐分数等，这些问题的序列数量更加的多。

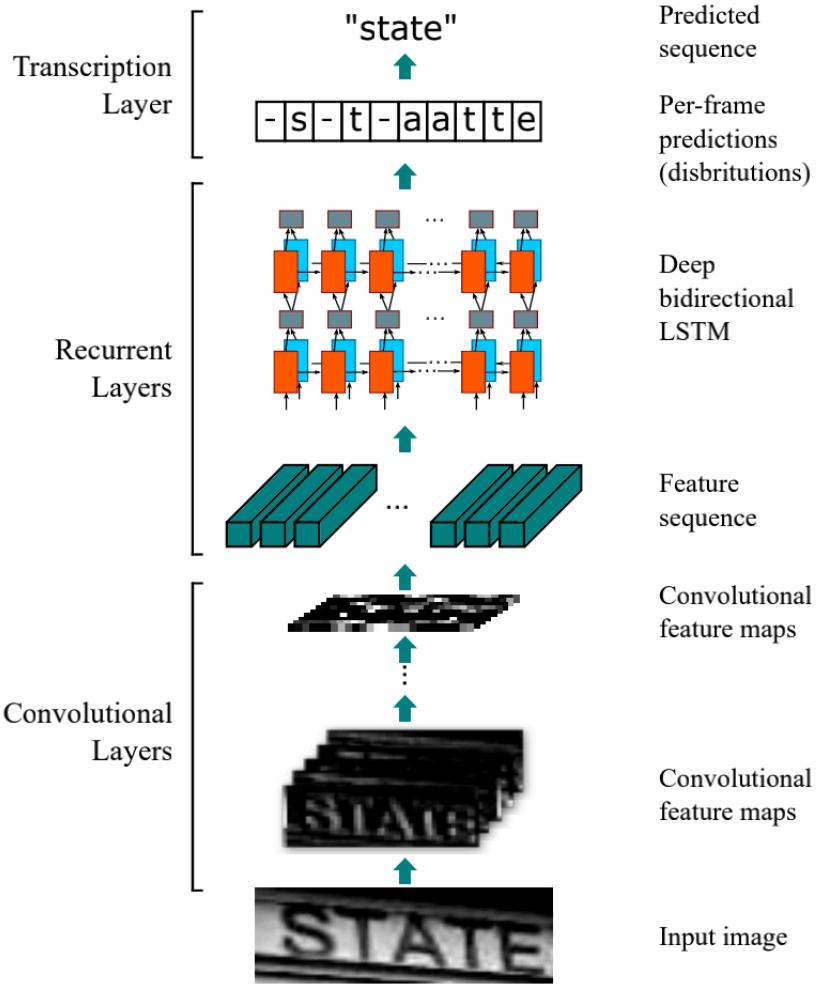
RNN也是常用于处理序列的算法，它的一个优点是对于一张序列物体图片，在训练和测试中不需要每个元素的位置。但需要做一个关键的预处理，将输入的物体图片转成图片特征的序列。这个预处理操作一般是独立于训练流程中，即这使得采用 RNN 的方法没办法做到端到端的训练和优化。

本文的主要贡献就是提出一个专门设计用于序列物体识别的新网络模型--CRNN，它结合了 CNN 和 RNN 两个算法。其优点有这几个：

1. 可以直接学习序列标签（如单词），并不需要更详细的标注（如字符）；
2. 具有和 DCNN 一样的属性，即可以直接从图片数据中学习信息表示(informative representations)，不需要手动设计特征，或者如二值化/分割、组件定位等预处理步骤；
3. 具有和 RNN 一样的属性，可以直接生成标签序列；
4. 不受物体长度的限制，训练和测试阶段只要求对高度的归一化；
5. 在场景文本问题中比之前的工作取得更好的性能表现；
6. 参数量比标准的 DCNN 模型更少，可以节省存储空间。

3. 方法

CRNN 的网络结构如下所示：



主要是分为三个组件，由下至上，分别是CNN网络、RNN 网络以及转录层。工作流程是这样的：

1. 输入图片先经过卷积层，提取得到一个特征序列；
2. 接着 RNN 对特征序列的每一帧进行预测；
3. 最后是转录层，对 RNN 的每帧预测结果进行翻译，得到最终的一个标签的序列，也就是将 RNN 预测的每个字符组合得到一个完整的单词。

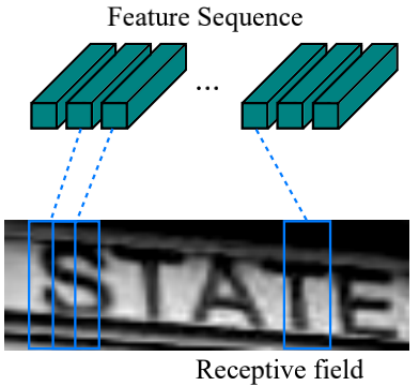
整个 CRNN 网络虽然包括不同的网络结构，比如 CNN 和 RNN，但是可以通过一个损失函数来联合训练。

3.1 特征序列提取

第一部采用的是移除了全连接层的 CNN 网络，也就是主要是卷积层和最大池化层。由于之前采用 CNN 来处理场景文本识别问题的工作，其实都无法解决目标物体的长度可变问题，所以 CRNN 的做法是将深度特征转换为序列表示，从而让其不受序列类物体长度变化的影响。

在输入图片之前，首先需要做的一个预处理步骤是将所有图片的高度都缩放为相同的高度，因为是对一行文本的识别，所以高度需要统一，而宽度不限制。

接着就是提取图片的特征，得到一个特征序列，需要注意的是，这个特征序列如下图所示，是从左到右按列生成的，即每一列特征对应原图中的一个矩形框区域，也可以说是感受野(receptive field)。



特征序列的宽度，在本文的设定中是固定为 1，即输出是 $h \times 1$，

3.2 序列标签

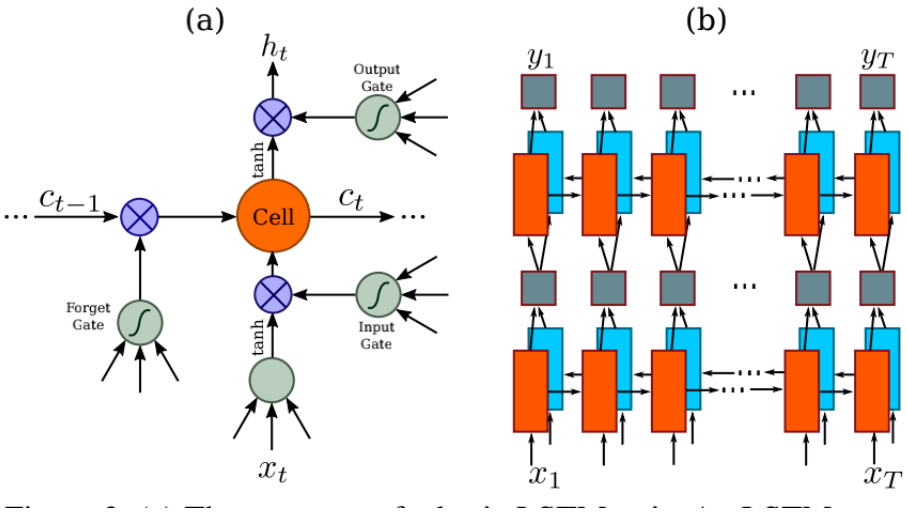
在 CNN 网络中对输入图片提取特征，并得到特征序列后，接下来就是 RNN 网络部分。RNN 网络部分将对特征序列里的每一帧 x_t 进行预测，输出得到对应的标签分布 y_t。

这里使用 RNN 网络的优点有这三个：

1. 首先，**RNN 有很强的捕获一个序列中上下文信息的能力**。对于基于图片的序列识别问题，采用上下文信息是比单独处理每个字符更稳定和更有效的方法，特别是处理一些有歧义的字符，比如 “il”；
2. 其次，**RNN 可以将误差微分反向传播到输入端**，也就是可以共同训练 CNN 和 RNN 网络；
3. 最后，**RNN 可以从头到尾对任意长度的序列进行操作**。

传统的 RNN 是可以利用到上一个状态的信息来进行预测的，但是存在一个梯度消失的问题，这会限制 RNN 可以存储的上下文长度，并且也对训练增加了负担。为了解决这个问题，所以就有了 LSTM 算法，LSTM 如下图(a)所示，一个 LSTM 包含了一个记忆单元和三个乘法门，分别是输入、输出和遗忘门。LSTM 的特别设计是可以捕获在基于图片的序列中经常出现的长期依赖性。

不过 LSTM 只是单向的，只能使用到过去的上下文信息，但在基于图片的系列中，过去和未来的上下文都是很有帮助的，因此本文采用的是双向的 LSTM，并且将多个双向 LSTM 进行堆积在一起，组成了一个深度双向 LSTM，如下图(b)所示。



在实际应用中，这里还设计了一个网络层，叫做"Map-to-Sequence"层，在 CNN 和 RNN 之间，主要是在反向传播中,让 RNN 的误差微分序列从特征图转换为特征序列的形式，然后传回到 CNN 中。

3.3 转录层

最后转录层是将 RNN 输出的每帧预测转换为一个标签序列，从数学是来说就是找到概率最大的一个标签序列。一般来说分为两种情况，带有字典和没有字典的情况：

- 如果测试集带有字典，那么输出结果就是计算出所有字典的概率，选择最大的作为最终预测的结果；
- 如果测试集没有字典，也就是测试集没有给出测试集包含哪些字符，那么就选择概率最大的作为最终的预测结果。

3.3.1 标签序列的概率

CRNN 采用在论文《 Connectionist temporal classification: labelling unseg mented sequence data with recurrent neural networks》中提出的 CTC （ Connectionist Temporal Classification） 层定义的条件概率。

这个条件概率是在每帧预测为 y, 标签序列为 l 的情况下定义的，它省略了每个标签在 l 中的位置信息，因此可以采用负对数似然函数作为目标函数来训练网络，然后只需要图片以及对应的标签序列，而不需要知道每个字符的位置的标注信息。

这里简单介绍这个条件概率的定义。输入是一个标签序列：$y = y_1, \cdots, y_T$，其中 T 是序列的长度。这里每个标签 $y_t \in R^{|L'|}$，其中 $L' = L \cup \{blank\}$ 表示任务中所有的标签（比如所有的英文字符）加上一个空白字符。另外定义一个序列转换函数 B，它负责将一个序列 π 转换为最终的标签序列 l。

这个转换操作，首先是移除 π 里重复的字符以及空白字符，比如将一个序列 “--hh-e-l-ll-oo--”（这里用 - 表示空白字符）转换为 "hello"，注意这里重复的字符之间如果有空白字符'-'，那么是不合并的，即 'l-la' 转换得到 'lla'，而 'll-a'转换得到的是 'la' 。

然后条件概率的定义就是如下所示，即将序列 π 的字符都转换为 l 的概率之和。

$$p(l|y) = \sum_{\pi:B(\pi)=l} p(\pi|y) \tag{1}$$

$$p(\pi|y) = \prod_{t=1}^{T} y_{\pi_t}^t$$

其中 $y_{\pi_t}^t$ 是在时间戳 t 的时候拥有标签 π_t 的概率。

上述(1)的等式直接计算看起来是不可行的，但是它可以按照 CTC 论文中提出的**“向前—向后”(forward-backward)**算法高效的计算。

更详细的介绍可以查看这篇文章：

https://zhuanlan.zhihu.com/p/43534801

总结一下，CTC是一种Loss计算方法，用 CTC 代替 Softmax Loss，**训练样本无需对齐**。CTC 的特点：

- 引入blank字符，解决有些位置没有字符的问题
- 通过递推，快速计算梯度

3.3.2 没有字典的转换

没有字典的情况下，就是选择概率最大的标签序列 l 作为输出结果。也就是说：

$$l \approx B(argmax_\pi p(\pi|y))$$

简单说在初始序列 π 的阶段，选择每个时间 t 里概率最大的标签 π_t ，然后再通过映射函数 B 转换得到最终的标签序列 l。

3.3.3 带字典的转换

如果是有带有字典 D的情况，那么预测的结果就是在字典 D 中根据公式(1) 计算得到的条件概率最大的标签序列。即：

$$l = argmax_{l \in D} p(l|y)$$

但这种做法的问题就是如果字典非常的大，比如带有 5 万个单词的字典中，逐个计算条件概率是一件非常耗时的做法。

为了解决这个问题，参考在没有字典的情况下预测的结果，可以发现采用编辑距离度量的结果非常接近于真实标签结果。所以，这里可以通过使用最近邻候选集来限制搜索的范围，定义最近邻候选集 $N_\delta(l^{'})$，其中 δ 是最大的编辑距离，$l^{'}$ 是在按照没有字典情况下由 y 转换得到的标签序列，所以最终的标签序列可以按如下所示得到：

$$l = argmax_{l \in N_\delta(l^{'})} p(l|y) \tag{2}$$

这里候选集 N 可以通过《 Some approaches to best match fifile searching》中提出的 BK 树结构高效得到，这是一种度量树，特别适合离散的度量空间，其搜索时间复杂度是 $O(log|D|)$，这里 $|D|$ 表示字典的大小。

在本文中，BK树通常是先离线构建好，这样就可以实现快速的在线搜索。

3.4 网络训练

定义数据集 $X = I_i, l_i$，其中I_i 表示训练图片，而 l_i 是真实的标签序列。目标函数是最小化负对数似然函数，如下所示：

$$O = -\sum_{I_i, l_i \in X} logp(l_i|y_i) \tag{3}$$

4. 实验

4.1 数据集

本文是先在一个包含 800 万训练图片的数据集上进行训练，然后在其他常用的数据集的测试集进行测试，并对比其他算法的结果。

训练集采用的是 Jaderberg 公开的一个合成数据集 Synth（论文：Synthetic data and artifificial neural networks for natural scene text recognition）。

采用的四个常用的基准场景文本识别数据集分别是：

1. ICDAR 2003
2. ICDAR2013
3. IIIT5K
4. SVT(Street View Text)

4.2 实现细节

CRNN 的网络参数如下所示：

Table 1. Network configuration summary. The first row is the top layer. 'k', 's' and 'p' stand for kernel size, stride and padding size respectively

Type	Configurations
Transcription	-
Bidirectional-LSTM	#hidden units:256
Bidirectional-LSTM	#hidden units:256
Map-to-Sequence	-
Convolution	#maps:512, k:2×2, s:1, p:0
MaxPooling	Window:1×2, s:2
BatchNormalization	-
Convolution	#maps:512, k:3×3, s:1, p:1
BatchNormalization	-
Convolution	#maps:512, k:3×3, s:1, p:1
MaxPooling	Window:1×2, s:2
Convolution	#maps:256, k:3×3, s:1, p:1
Convolution	#maps:256, k:3×3, s:1, p:1
MaxPooling	Window:2×2, s:2
Convolution	#maps:128, k:3×3, s:1, p:1
MaxPooling	Window:2×2, s:2
Convolution	#maps:64, k:3×3, s:1, p:1
Input	$W \times 32$ gray-scale image

可以看到 CNN 部分是采用 VGG 网络模型，但做出了一些修改：

1. 对第三个和第四个最大池化层，采用的是 1×2 大小的窗口，而不是 2×2;
2. 为了加快网络训练，在第五层和第六层后都加入了 BN 层；

对于第一个修改，这里需要说明一下，首先因为输入图片是进行调整到 $W \times 32$, 即高度统一为 32，那么假设一张包含 10 个字符的图片，大小是 100×32，经过上述的 CNN 后得到的特征尺度就是 25×1，刚好就符合了在 CNN 部分说的每一列特征对应原图上的一个矩形框区域。

第二点是因为尽管 CNN 部分只有 7 层，相对不是非常深，但是因为后面接着 RNN，RNN 并不好训练，所以这里是加入了两个 batch normalization 层来加快训练的速度。

训练阶段，所有图片都统一调整为 100×32 大小，而测试时候的图片，只是将高度统一调整为 32，宽度则是保持长宽比的情况下进行调整，但至少是 100。

4.3 对比结果

对比实验结果如下所示，CRNN 基本是在所有数据集上都做到了性能最好的情况，只有 3 种情况下性能不是最好的，分别是在 IC03 数据集是全集以及不使用字典，还有 IC13 测试集但不提供字典的情况。

Table 2. Recognition accuracies (%) on four datasets. In the second row, "50", "1k", "50k" and "Full" denote the lexicon used, and "None" denotes recognition without a lexicon. (*[22] is not lexicon-free in the strict sense, as its outputs are constrained to a 90k dictionary.

	IIIT5k			SVT		IC03				IC13
	50	1k	None	50	None	50	Full	50k	None	None
ABBYY [34]	24.3	-	-	35.0	-	56.0	55.0	-	-	-
Wang *et al.* [34]	-	-	-	57.0	-	76.0	62.0	-	-	-
Mishra *et al.* [28]	64.1	57.5	-	73.2	-	81.8	67.8	-	-	-
Wang *et al.* [35]	-	-	-	70.0	-	90.0	84.0	-	-	-
Goel *et al.* [13]	-	-	-	77.3	-	89.7	-	-	-	-
Bissacco *et al.* [8]	-	-	-	90.4	78.0	-	-	-	-	87.6
Alsharif and Pineau [6]	-	-	-	74.3	-	93.1	88.6	85.1	-	-
Almazán *et al.* [5]	91.2	82.1	-	89.2	-	-	-	-	-	-
Yao *et al.* [36]	80.2	69.3	-	75.9	-	88.5	80.3	-	-	-
Rodrguez-Serrano *et al.* [30]	76.1	57.4	-	70.0	-	-	-	-	-	-
Jaderberg *et al.* [23]	-	-	-	86.1	-	96.2	91.5	-	-	-
Su and Lu [33]	-	-	-	83.0	-	92.0	82.0	-	-	-
Gordo [14]	93.3	86.6	-	91.8	-	-	-	-	-	-
Jaderberg *et al.* [22]	97.1	92.7	-	95.4	80.7*	**98.7**	**98.6**	93.3	**93.1***	**90.8***
Jaderberg *et al.* [21]	95.5	89.6	-	93.2	71.7	97.8	97.0	93.4	89.6	81.8
CRNN	**97.6**	**94.4**	**78.2**	**96.4**	**80.8**	**98.7**	97.6	**95.5**	89.4	86.7

另外，文章还设置了这几个对比的角度：

- E2E Train：是否端到端训练的形式；
- Conv Ftrs：采用 CNN 的特征还是手动设计特征；
- CharGT-Free：是否提供了字符级别的标注；
- Unconstrained：是否只限制在特定的字典，能否处理不在字典上的单词
- Model Size：模型的大小

	E2E Train	Conv Ftrs	CharGT-Free	Unconstrained	Model Size
Wang *et al.* [34]	✗	✗	✗	✔	-
Mishra *et al.* [28]	✗	✗	✗	✗	-
Wang *et al.* [35]	✗	✔	✗	✔	-
Goel *et al.* [13]	✗	✗	✔	✗	-
Bissacco *et al.* [8]	✗	✗	✗	✔	-
Alsharif and Pineau [6]	✗	✔	✗	✔	-
Almazán *et al.* [5]	✗	✗	✔	✗	-
Yao *et al.* [36]	✗	✗	✗	✔	-
Rodrguez-Serrano *et al.* [30]	✗	✗	✔	✗	-
Jaderberg *et al.* [23]	✗	✔	✗	✔	-
Su and Lu [33]	✗	✗	✔	✔	-
Gordo [14]	✗	✗	✗	✗	-
Jaderberg *et al.* [22]	✔	✔	✔	✗	490M
Jaderberg *et al.* [21]	✔	✔	✔	✔	304M
CRNN	✔	✔	✔	✔	**8.3M**

5. 总结

CRNN 是结合了 CNN +LSTM+CTC 的优势：

- 首先用 CNN 提取图像的卷积特征，不需要手动设计特征；
- 接着用 LSTM 进一步提取图像卷积特征中的序列特征
- 最后引入 CTC 解决训练时候字符无法对齐的问题。

可以在只需要基本的单词级别的标签和输入图片就可以实现端到端的训练。

CRNN 是一个通用的框架，可以处理多种基于图片的序列识别问题，不仅是英文单词、音乐分数，还有中文的识别问题。

参考

1. CRNN算法详解
2. 一文读懂CRNN+CTC文字识别