CRNN论文翻译——中英文对照

CRNN论文翻译——中英文对照

文章作者：Tyan

博客：noahsnail.com | CSDN | 简书

翻译论文汇总：https://github.com/SnailTyan/deep-learning-papers-translation

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

Abstract

Image-based sequence recognition has been a long-standing research topic in computer vision. In this paper, we investigate the problem of scene text recognition, which is among the most important and challenging tasks in image-based sequence recognition. A novel neural network architecture, which integrates feature extraction, sequence modeling and transcription into a unified framework, is proposed. Compared with previous systems for scene text recognition, the proposed architecture possesses four distinctive properties: (1) It is end-to-end trainable, in contrast to most of the existing algorithms whose components are separately trained and tuned. (2) It naturally handles sequences in arbitrary lengths, involving no character segmentation or horizontal scale normalization. (3) It is not confined to any predefined lexicon and achieves remarkable performances in both lexicon-free and lexicon-based scene text recognition tasks. (4) It generates an effective yet much smaller model, which is more practical for real-world application scenarios. The experiments on standard benchmarks, including the IIIT-5K, Street View Text and ICDAR datasets, demonstrate the superiority of the proposed algorithm over the prior arts. Moreover, the proposed algorithm performs well in the task of image-based music score recognition, which evidently verifies the generality of it.

摘要

基于图像的序列识别一直是计算机视觉中长期存在的研究课题。在本文中，我们研究了场景文本识别的问题，这是基于图像的序列识别中最重要和最具挑战性的任务之一。提出了一种将特征提取，序列建模和转录整合到

统一框架中的新型神经网络架构。与以前的场景文本识别系统相比，所提出的架构具有四个不同的特性：（1）与大多数现有的组件需要单独训练和协调的算法相比，它是端对端训练的。（2）它自然地处理任意长度的序列，不涉及字符分割或水平尺度归一化。（3）它不仅限于任何预定义的词汇，并且在无词典和基于词典的场景文本识别任务中都取得了显著的表现。（4）它产生了一个有效而小得多的模型，这对于现实世界的应用场景更为实用。在包括IIIT-5K，Street View Text和ICDAR数据集在内的标准基准数据集上的实验证明了提出的算法比现有技术的更有优势。此外，提出的算法在基于图像的音乐配乐识别任务中表现良好，这显然证实了它的泛化性。

1. Introduction

Recently, the community has seen a strong revival of neural networks, which is mainly stimulated by the great success of deep neural network models, specifically Deep Convolutional Neural Networks (DCNN), in various vision tasks. However, majority of the recent works related to deep neural networks have devoted to detection or classification of object categories [12, 25]. In this paper, we are concerned with a classic problem in computer vision: image-based sequence recognition. In real world, a stable of visual objects, such as scene text, handwriting and musical score, tend to occur in the form of sequence, not in isolation. Unlike general object recognition, recognizing such sequence-like objects often requires the system to predict a series of object labels, instead of a single label. Therefore, recognition of such objects can be naturally cast as a sequence recognition problem. Another unique property of sequence-like objects is that their lengths may vary drastically. For instance, English words can either consist of 2 characters such as “OK” or 15 characters such as “congratulations”. Consequently, the most popular deep models like DCNN [25, 26] cannot be directly applied to sequence prediction, since DCNN models often operate on inputs and outputs with fixed dimensions, and thus are incapable of producing a variable-length label sequence.

1. 引言

最近，社区已经看到神经网络的强大复兴，这主要受到深度神经网络模型，特别是深度卷积神经网络（DCNN）在各种视觉任务中的巨大成功的推动。然而，最近大多数与深度神经网络相关的工作主要致力于检测或分类对象类别[12,25]。在本文中，我们关注计算机视觉中的一个经典问题：基于图像的序列识别。在现实世界中，稳定的视觉对象，如场景文字，手写字符和乐谱，往往以序列的形式出现，而不是孤立地出现。与一般的对象识别不同，识别这样的类序列对象通常需要系统预测一系列对象标签，而不是单个标签。因此，可以自然地将这样的对象的识别作为序列识别问题。类序列对象的另一个独特之处在于它们的长度可能会有很大变化。例如，英文单词可以由2个字符组成，如“OK”，或由15个字符组成，如“congratulations”。因此，最流行的深度模型像DCNN[25,26]不能直接应用于序列预测，因为DCNN模型通常对具有固定维度的输入和输出进行操作，因此不能产生可变长度的标签序列。

Some attempts have been made to address this problem for a specific sequence-like object (e.g. scene text). For example, the algorithms in [35, 8] firstly detect individual characters and then recognize these detected

characters with DCNN models, which are trained using labeled character images. Such methods often require training a strong character detector for accurately detecting and cropping each character out from the original word image. Some other approaches (such as [22]) treat scene text recognition as an image classification problem, and assign a class label to each English word (90K words in total). It turns out a large trained model with a huge number of classes, which is difficult to be generalized to other types of sequence-like objects, such as Chinese texts, musical scores, etc., because the numbers of basic combinations of such kind of sequences can be greater than 1 million. In summary, current systems based on DCNN can not be directly used for image-based sequence recognition.

已经针对特定的类似序列的对象（例如场景文本）进行了一些尝试来解决该问题。例如，[35,8]中的算法首先检测单个字符，然后用DCNN模型识别这些检测到的字符，并使用标注的字符图像进行训练。这些方法通常需要训练强字符检测器，以便从原始单词图像中准确地检测和裁剪每个字符。一些其他方法（如[22]）将场景文本识别视为图像分类问题，并为每个英文单词（总共9万个词）分配一个类标签。结果是一个大的训练模型中有很多类，这很难泛化到其它类型的类序列对象，如中文文本，音乐配乐等，因为这种序列的基本组合数目可能大于100万。总之，目前基于DCNN的系统不能直接用于基于图像的序列识别。

Recurrent neural networks (RNN) models, another important branch of the deep neural networks family, were mainly designed for handling sequences. One of the advantages of RNN is that it does not need the position of each element in a sequence object image in both training and testing. However, a preprocessing step that converts an input object image into a sequence of image features, is usually essential. For example, Graves et al. [16] extract a set of geometrical or image features from handwritten texts, while Su and Lu [33] convert word images into sequential HOG features. The preprocessing step is independent of the subsequent components in the pipeline, thus the existing systems based on RNN can not be trained and optimized in an end-to-end fashion.

循环神经网络（RNN）模型是深度神经网络家族中的另一个重要分支，主要是设计来处理序列。RNN的优点之一是在训练和测试中不需要序列目标图像中每个元素的位置。然而，将输入目标图像转换成图像特征序列的预处理步骤通常是必需的。例如，Graves等[16]从手写文本中提取一系列几何或图像特征，而Su和Lu[33]将字符图像转换为序列HOG特征。预处理步骤独立于流程中的后续组件，因此基于RNN的现有系统不能以端到端的方式进行训练和优化。

Several conventional scene text recognition methods that are not based on neural networks also brought insightful ideas and novel representations into this field. For example, Almazan et al. [5] and Rodriguez-Serrano et al. [30] proposed to embed word images and text strings in a common vectorial subspace, and word recognition is converted into a retrieval problem. Yao et al. [36] and Gordo et al. [14] used mid-level features for scene text recognition. Though achieved promising performance on standard benchmarks, these methods are generally outperformed by previous algorithms based on neural networks [8, 22], as well as the approach proposed in this paper.

一些不是基于神经网络的传统场景文本识别方法也为这一领域带来了有见地的想法和新颖的表现。例如，Almazan等人[5]和Rodriguez-Serrano等人[30]提出将单词图像和文本字符串嵌入到公共向量子空间中，并将词识别转换为检索问题。Yao等人[36]和Gordo等人[14]使用中层特征进行场景文本识别。虽然在标准基准数据集上取得了有效的性能，但是前面的基于神经网络的算法[8,22]以及本文提出的方法通常都优于这些方法。

The main contribution of this paper is a novel neural network model, whose network architecture is specifically designed for recognizing sequence-like objects in images. The proposed neural network model is named as Convolutional Recurrent Neural Network (CRNN), since it is a combination of DCNN and RNN. For sequence-like objects, CRNN possesses several distinctive advantages over conventional neural network models: 1) It can be directly learned from sequence labels (for instance, words), requiring no detailed annotations (for instance, characters); 2) It has the same property of DCNN on learning informative representations directly from image data, requiring neither hand-craft features nor preprocessing steps, including binarization/segmentation, component localization, etc.; 3) It has the same property of RNN, being able to produce a sequence of labels; 4) It is unconstrained to the lengths of sequence-like objects, requiring only height normalization in both training and testing phases; 5) It achieves better or highly competitive performance on scene texts (word recognition) than the prior arts [23, 8]; 6) It contains much less parameters than a standard DCNN model, consuming less storage space.

本文的主要贡献是一种新颖的神经网络模型，其网络架构设计专门用于识别图像中的类序列对象。所提出的神经网络模型被称为卷积循环神经网络（CRNN），因为它是DCNN和RNN的组合。对于类序列对象，CRNN与传统神经网络模型相比具有一些独特的优点：1）可以直接从序列标签（例如单词）学习，不需要详细的标注（例如字符）；2）直接从图像数据学习信息表示时具有与DCNN相同的性质，既不需要手工特征也不需要预处理步骤，包括二值化/分割，组件定位等；3）具有与RNN相同的性质，能够产生一系列标签；4）对类序列对象的长度无约束，只需要在训练阶段和测试阶段对高度进行归一化；5）与现有技术相比，它在场景文本（字识别）上获得更好或更具竞争力的表现[23,8]。6）它比标准DCNN模型包含的参数要少得多，占用更少的存储空间。

2. The Proposed Network Architecture

The network architecture of CRNN, as shown in Fig. 1, consists of three components, including the convolutional layers, the recurrent layers, and a transcription layer, from bottom to top.

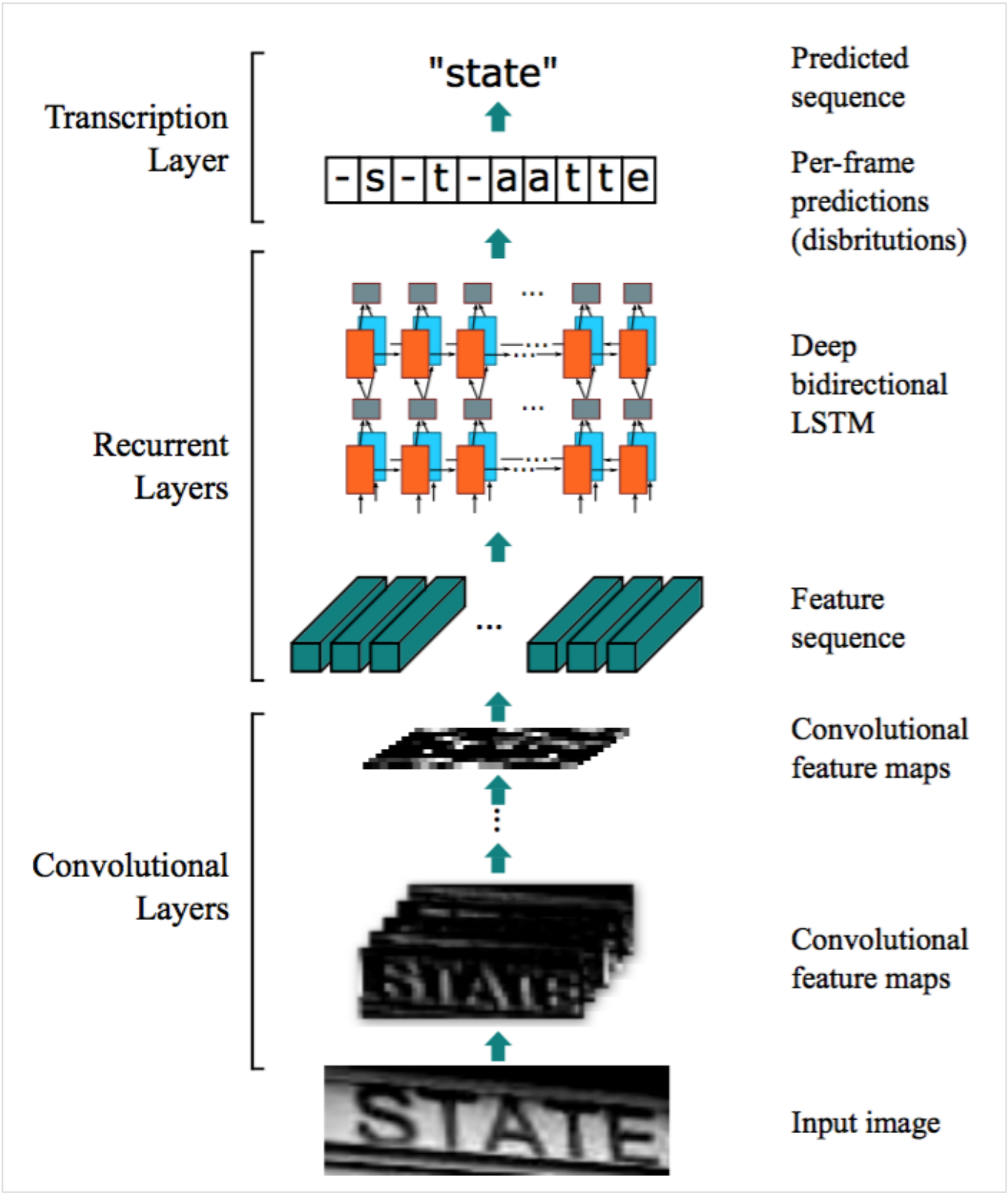

Figure 1. The network architecture. The architecture consists of three parts: 1) convolutional layers, which extract a feature sequence from the input image; 2) recurrent layers, which predict a label distribution for each frame; 3) transcription layer, which translates the per-frame predictions into the final label sequence.

2. 提出的网络架构

如图1所示，CRNN的网络架构由三部分组成，包括卷积层，循环层和转录层，从底向上。

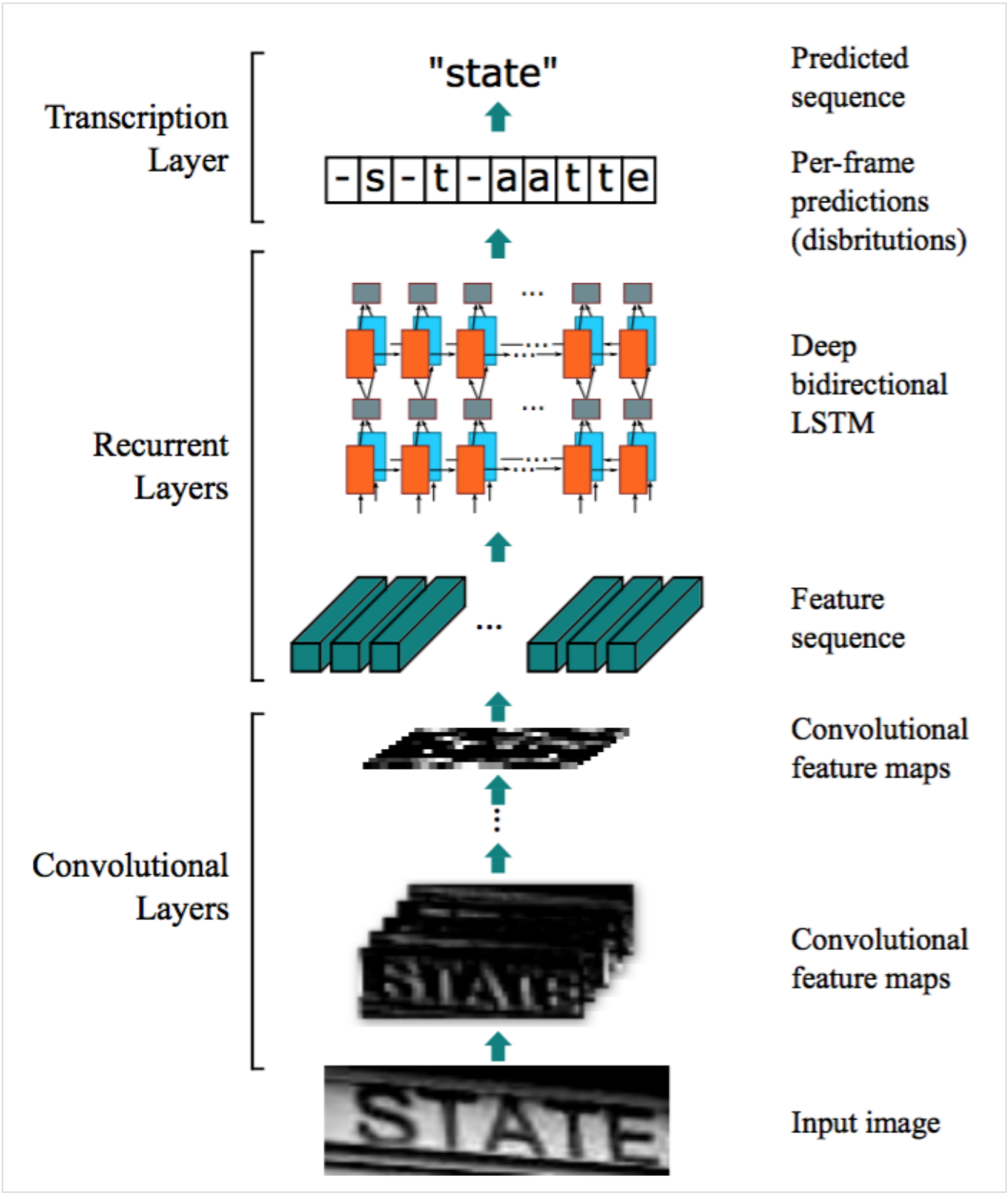

图1。网络架构。架构包括三部分：1) 卷积层，从输入图像中提取特征序列；2) 循环层，预测每一帧的标签分布；3) 转录层，将每一帧的预测变为最终的标签序列。

At the bottom of CRNN, the convolutional layers automatically extract a feature sequence from each input image. On top of the convolutional network, a recurrent network is built for making prediction for each frame of the feature sequence, outputted by the convolutional layers. The transcription layer at the top of CRNN is adopted to translate the per-frame predictions by the recurrent layers into a label sequence. Though CRNN is

composed of different kinds of network architectures (eg. CNN and RNN), it can be jointly trained with one loss function.

在CRNN的底部，卷积层自动从每个输入图像中提取特征序列。在卷积网络之上，构建了一个循环网络，用于对卷积层输出的特征序列的每一帧进行预测。采用CRNN顶部的转录层将循环层的每帧预测转化为标签序列。虽然CRNN由不同类型的网络架构（如CNN和RNN）组成，但可以通过一个损失函数进行联合训练。

2.1. Feature Sequence Extraction

In CRNN model, the component of convolutional layers is constructed by taking the convolutional and max-pooling layers from a standard CNN model (fully-connected layers are removed). Such component is used to extract a sequential feature representation from an input image. Before being fed into the network, all the images need to be scaled to the same height. Then a sequence of feature vectors is extracted from the feature maps produced by the component of convolutional layers, which is the input for the recurrent layers. Specifically, each feature vector of a feature sequence is generated from left to right on the feature maps by column. This means the i-th feature vector is the concatenation of the i-th columns of all the maps. The width of each column in our settings is fixed to single pixel.

2.1. 特征序列提取

在CRNN模型中，通过采用标准CNN模型（去除全连接层）中的卷积层和最大池化层来构造卷积层的组件。这样的组件用于从输入图像中提取序列特征表示。在进入网络之前，所有的图像需要缩放到相同的高度。然后从卷积层组件产生的特征图中提取特征向量序列，这些特征向量序列作为循环层的输入。具体地，特征序列的每一个特征向量在特征图上按列从左到右生成。这意味着第i个特征向量是所有特征图第i列的连接。在我们的设置中每列的宽度固定为单个像素。

As the layers of convolution, max-pooling, and element-wise activation function operate on local regions, they are translation invariant. Therefore, each column of the feature maps corresponds to a rectangle region of the original image (termed the receptive field), and such rectangle regions are in the same order to their corresponding columns on the feature maps from left to right. As illustrated in Fig. 2, each vector in the feature sequence is associated with a receptive field, and can be considered as the image descriptor for that region.

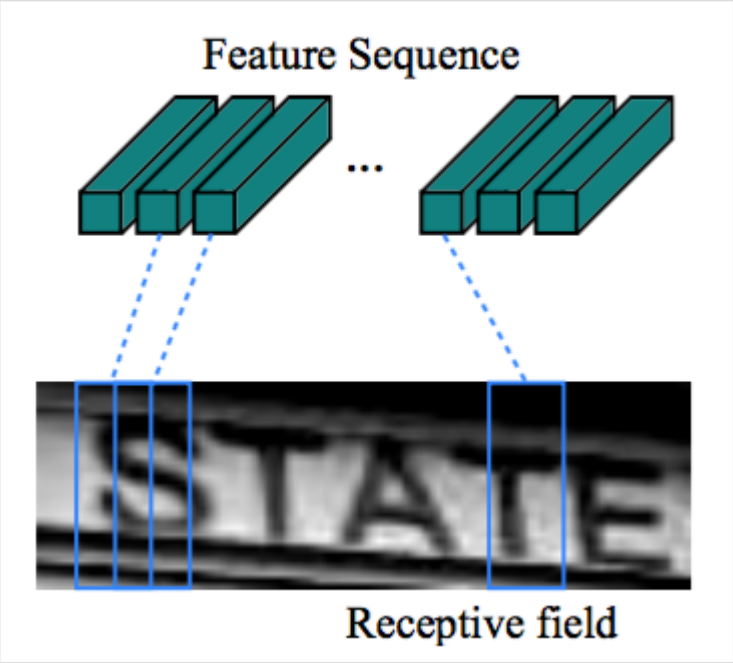

Figure 2. The receptive field. Each vector in the extracted feature sequence is associated with a receptive field on the input image, and can be considered as the feature vector of that field.

由于卷积层，最大池化层和元素激活函数在局部区域上执行，因此它们是平移不变的。因此，特征图的每列对应于原始图像的一个矩形区域（称为感受野），并且这些矩形区域与特征图上从左到右的相应列具有相同的顺序。如图2所示，特征序列中的每个向量关联一个感受野，并且可以被认为是该区域的图像描述符。

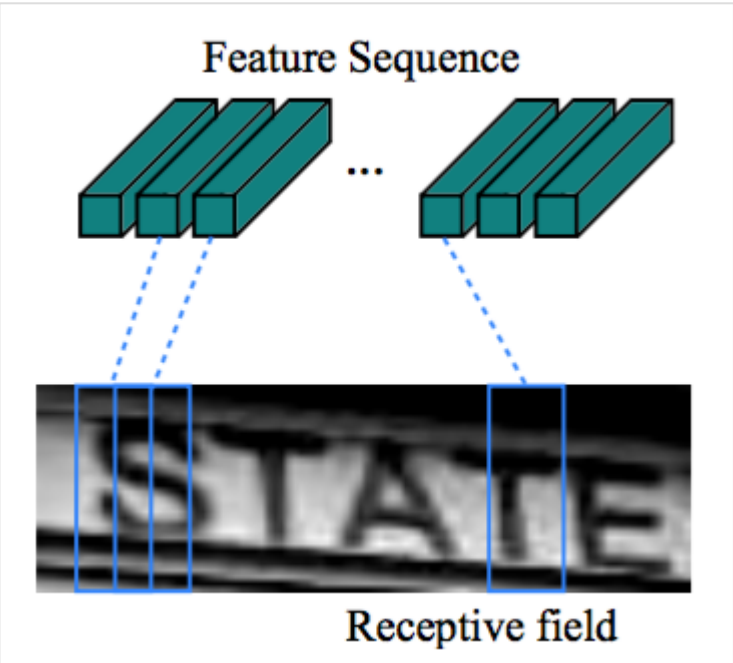

图2。感受野。提取的特征序列中的每一个向量关联输入图像的一个感受野，可认为是该区域的特征向量。

Being robust, rich and trainable, deep convolutional features have been widely adopted for different kinds of visual recognition tasks [25, 12]. Some previous approaches have employed CNN to learn a robust representation for sequence-like objects such as scene text [22]. However, these approaches usually extract

holistic representation of the whole image by CNN, then the local deep features are collected for recognizing each component of a sequence-like object. Since CNN requires the input images to be scaled to a fixed size in order to satisfy with its fixed input dimension, it is not appropriate for sequence-like objects due to their large length variation. In CRNN, we convey deep features into sequential representations in order to be invariant to the length variation of sequence-like objects.

鲁棒的，丰富的和可训练的深度卷积特征已被广泛应用于各种视觉识别任务[25,12]。一些以前的方法已经使用CNN来学习诸如场景文本之类的类序列对象的鲁棒表示[22]。然而，这些方法通常通过CNN提取整个图像的整体表示，然后收集局部深度特征来识别类序列对象的每个分量。由于CNN要求将输入图像缩放到固定尺寸，以满足其固定的输入尺寸，因为它们的长度变化很大，因此不适合类序列对象。在CRNN中，我们将深度特征传递到序列表示中，以便对类序列对象的长度变化保持不变。

2.2. Sequence Labeling

A deep bidirectional Recurrent Neural Network is built on the top of the convolutional layers, as the recurrent layers. The recurrent layers predict a label distribution y_t for each frame x_t in the feature sequence $x = x_1, \ldots, x_T$. The advantages of the recurrent layers are three-fold. Firstly, RNN has a strong capability of capturing contextual information within a sequence. Using contextual cues for image-based sequence recognition is more stable and helpful than treating each symbol independently. Taking scene text recognition as an example, wide characters may require several successive frames to fully describe (refer to Fig. 2). Besides, some ambiguous characters are easier to distinguish when observing their contexts, e.g. it is easier to recognize "il" by contrasting the character heights than by recognizing each of them separately. Secondly, RNN can back-propagates error differentials to its input, i.e. the convolutional layer, allowing us to jointly train the recurrent layers and the convolutional layers in a unified network. Thirdly, RNN is able to operate on sequences of arbitrary lengths, traversing from starts to ends.

2.2. 序列标注

一个深度双向循环神经网络是建立在卷积层的顶部，作为循环层。循环层预测特征序列$x = x_1, \ldots, x_T$中每一帧x_t的标签分布y_t。循环层的优点是三重的。首先，RNN具有很强的捕获序列内上下文信息的能力。对于基于图像的序列识别使用上下文提示比独立处理每个符号更稳定且更有帮助。以场景文本识别为例，宽字符可能需要一些连续的帧来完全描述（参见图2）。此外，一些模糊的字符在观察其上下文时更容易区分，例如，通过对比字符高度更容易识别"il"而不是分别识别它们中的每一个。其次，RNN可以将误差差值反向传播到其输入，即卷积层，从而允许我们在统一的网络中共同训练循环层和卷积层。第三，RNN能够从头到尾对任意长度的序列进行操作。

A traditional RNN unit has a self-connected hidden layer between its input and output layers. Each time it receives a frame x_t in the sequence, it updates its internal state h_t with a non-linear function that takes both

current input x_t and past state h_{t-1} as its inputs: $h_t = g(x_t, h_{t-1})$. Then the prediction y_t is made based on h_t. In this way, past contexts $\{x_{t\prime}\}_{t\prime<t}$ are captured and utilized for prediction. Traditional RNN unit, however, suffers from the vanishing gradient problem [7], which limits the range of context it can store, and adds burden to the training process. Long-Short Term Memory 18, 11 is a type of RNN unit that is specially designed to address this problem. An LSTM (illustrated in Fig. 3) consists of a memory cell and three multiplicative gates, namely the input, output and forget gates. Conceptually, the memory cell stores the past contexts, and the input and output gates allow the cell to store contexts for a long period of time. Meanwhile, the memory in the cell can be cleared by the forget gate. The special design of LSTM allows it to capture long-range dependencies, which often occur in image-based sequences.

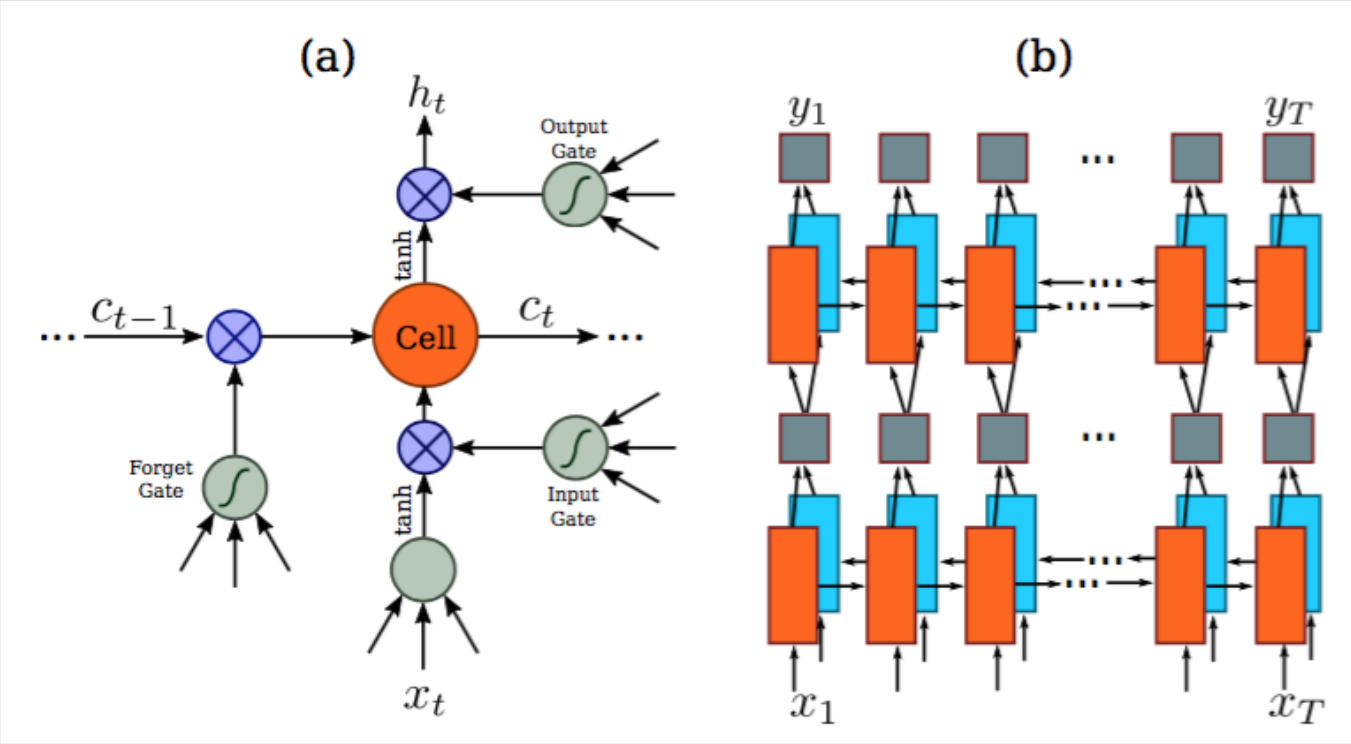

Figure 3. (a) The structure of a basic LSTM unit. An LSTM consists of a cell module and three gates, namely the input gate, the output gate and the forget gate. (b) The structure of deep bidirectional LSTM we use in our paper. Combining a forward (left to right) and a backward (right to left) LSTMs results in a bidirectional LSTM. Stacking multiple bidirectional LSTM results in a deep bidirectional LSTM.

传统的RNN单元在其输入和输出层之间具有自连接的隐藏层。每次接收到序列中的帧x_t时，它将使用非线性函数来更新其内部状态h_t，该非线性函数同时接收当前输入x_t和过去状态h_{t-1}作为其输入：$h_t = g(x_t, h_{t-1})$。那么预测y_t是基于h_t的。以这种方式，过去的上下文$\{\{x_{t\prime}\}_{t\prime<t}$被捕获并用于预测。然而，传统的RNN单元有梯度消失的问题[7]，这限制了其可以存储的上下文范围，并给训练过程增加了负担。长短时记忆[18,11]（LSTM）是一种专门设计用于解决这个问题的RNN单元。LSTM（图3所示）由一个存储单元和三个多重门组成，即输入，输出和遗忘门。在概念上，存储单元存储过去的上下文，并且输入和输出门允许单元长时间地存储上下文。同时，单元中的存储可以被遗忘门清除。LSTM的特殊设计允许它捕获长距离依赖，这经常发生在基于图像的序列中。

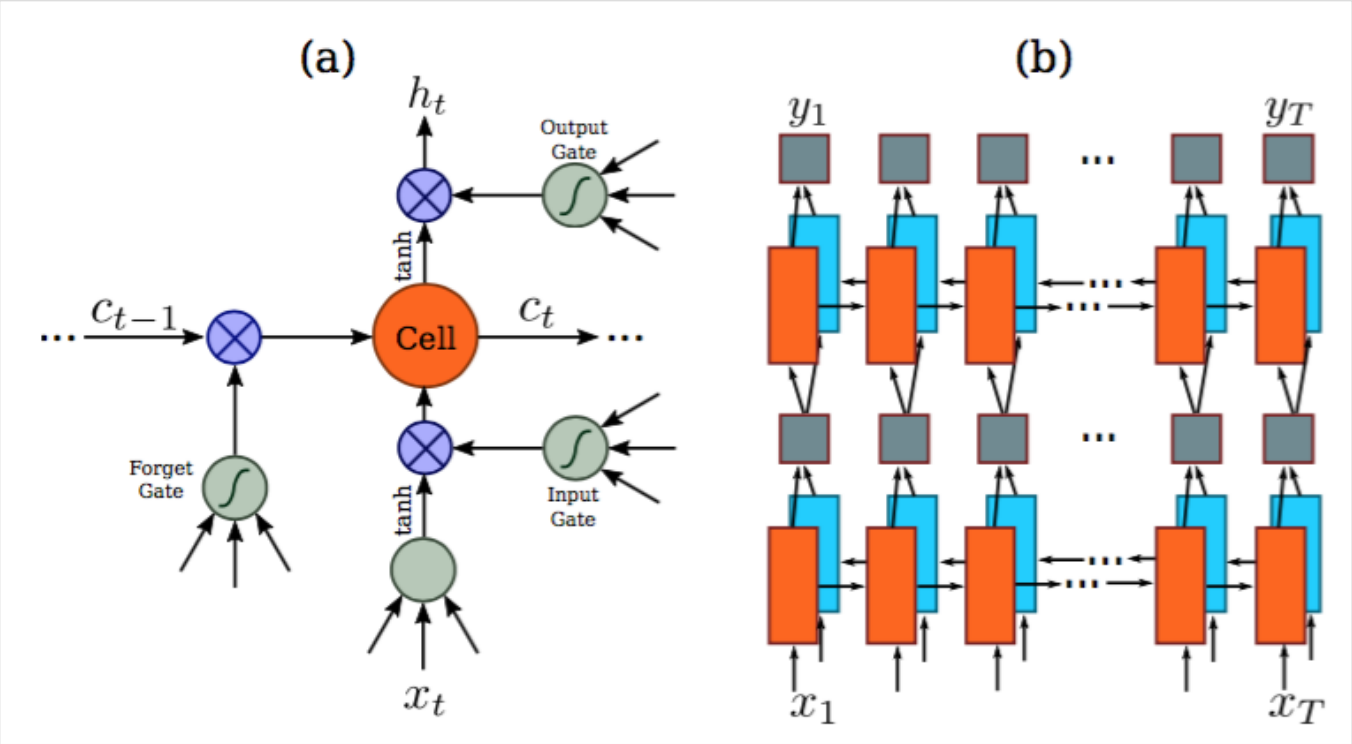

图3。(a) 基本的LSTM单元的结构。LSTM包括单元模块和三个门，即输入门，输出门和遗忘门。（b）我们论文中使用的深度双向LSTM结构。合并前向（从左到右）和后向（从右到左）LSTM的结果到双向LSTM中。在深度双向LSTM中堆叠多个双向LSTM结果。

LSTM is directional, it only uses past contexts. However, in image-based sequences, contexts from both directions are useful and complementary to each other. Therefore, we follow [17] and combine two LSTMs, one forward and one backward, into a bidirectional LSTM. Furthermore, multiple bidirectional LSTMs can be stacked, resulting in a deep bidirectional LSTM as illustrated in Fig. 3.b. The deep structure allows higher level of abstractions than a shallow one, and has achieved significant performance improvements in the task of speech recognition [17].

LSTM是定向的，它只使用过去的上下文。然而，在基于图像的序列中，两个方向的上下文是相互有用且互补的。因此，我们遵循[17]，将两个LSTM，一个向前和一个向后组合到一个双向LSTM中。此外，可以堆叠多个双向LSTM，得到如图3.b所示的深双向LSTM。深层结构允许比浅层抽象更高层次的抽象，并且在语音识别任务中取得了显著的性能改进[17]。

In recurrent layers, error differentials are propagated in the opposite directions of the arrows shown in Fig. 3.b, i.e. Back-Propagation Through Time (BPTT). At the bottom of the recurrent layers, the sequence of propagated differentials are concatenated into maps, inverting the operation of converting feature maps into feature sequences, and fed back to the convolutional layers. In practice, we create a custom network layer, called “Map-to-Sequence”, as the bridge between convolutional layers and recurrent layers.

在循环层中，误差在图3.b所示箭头的相反方向传播，即反向传播时间（BPTT）。在循环层的底部，传播差异的序列被连接成映射，将特征映射转换为特征序列的操作进行反转并反馈到卷积层。实际上，我们创建一个称为“Map-to-Sequence”的自定义网络层，作为卷积层和循环层之间的桥梁。

2.3. Transcription

Transcription is the process of converting the per-frame predictions made by RNN into a label sequence. Mathematically, transcription is to find the label sequence with the highest probability conditioned on the per-frame predictions. In practice, there exists two modes of transcription, namely the lexicon-free and lexicon-based transcriptions. A lexicon is a set of label sequences that prediction is constraint to, e.g. a spell checking dictionary. In lexicon-free mode, predictions are made without any lexicon. In lexicon-based mode, predictions are made by choosing the label sequence that has the highest probability.

2.3. 转录

转录是将RNN所做的每帧预测转换成标签序列的过程。数学上，转录是根据每帧预测找到具有最高概率的标签序列。在实践中，存在两种转录模式，即无词典转录和基于词典的转录。词典是一组标签序列，预测受拼写检查字典约束。在无词典模式中，预测时没有任何词典。在基于词典的模式中，通过选择具有最高概率的标签序列进行预测。

2.3.1 Probability of label sequence

We adopt the conditional probability defined in the Connectionist Temporal Classification (CTC) layer proposed by Graves et al. [15]. The probability is defined for label sequence l conditioned on the per-frame predictions $y = y_1, \ldots, y_T$, and it ignores the position where each label in l is located. Consequently, when we use the negative log-likelihood of this probability as the objective to train the network, we only need images and their corresponding label sequences, avoiding the labor of labeling positions of individual characters.

2.3.1 标签序列的概率

我们采用Graves等人[15]提出的联接时间分类（CTC）层中定义的条件概率。按照每帧预测$y = y_1, \ldots, y_T$对标签序列l定义概率，并忽略l中每个标签所在的位置。因此，当我们使用这种概率的负对数似然作为训练网络的目标函数时，我们只需要图像及其相应的标签序列，避免了标注单个字符位置的劳动。

The formulation of the conditional probability is briefly described as follows: The input is a sequence $y = y_1, \ldots, y_T$ where T is the sequence length. Here, each $y_t \in \Re^{|\mathcal{L}'|}$ is a probability distribution over the set $\mathcal{L}' = \mathcal{L} \cup$, where $\mathcal{L}$ contains all labels in the task (e.g. all English characters), as well as a 'blank' label denoted by `-`. A sequence-to-sequence mapping function $\mathcal{B}$ is defined on sequence $\boldsymbol{\pi} \in \mathcal{L}'^T$, where T is the length. $\mathcal{B}$ maps $\boldsymbol{\pi}$ onto $\mathbf{l}$ by firstly removing the repeated labels, then removing the `blank`s. For example, B maps "−hh-e-l-

ll-oo–" ('-' represents 'blank') onto "hello". Then, the conditional probability is defined as the sum of probabilities of all $\boldsymbol{\pi}$ that are mapped by $\mathcal{B}$ onto $\mathbf{l}$:

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

where the probability of $\boldsymbol{\pi}$ is defined as $p(\boldsymbol{\pi}|\mathbf{y})=\prod_{t=1}^{T}y_{\pi_t}^{t}$, $y_{\pi_t}^{t}$ $is the probability of having label \pi_t at time stamp t$. Directly computing Eq.1 would be computationally infeasible due to the exponentially large number of summation items. However, Eq.1 can be efficiently computed using the forward-backward algorithm described in [15].

条件概率的公式简要描述如下：输入是序列$y = y_1, \ldots, y_T$，其中T是序列长度。这里，每个$y_t \in \Re^{|\mathcal{L}'|}$是在集合$\mathcal{L}' = \mathcal{L} \cup$上的概率分布，其中$\mathcal{L}$包含了任务中的所有标签（例如，所有英文字符），以及由`-`表示的"空白"标签。序列到序列的映射函数$\mathcal{B}$定义在序列$\boldsymbol{\pi} \in \mathcal{L}'^T$上，其中$T$是长度。$\mathcal{B}$将$\boldsymbol{\pi}$映射到$\mathbf{l}$上，首先删除重复的标签，然后删除`blank`。例如，$\mathcal{B}$将"–hh-e-l-ll-oo–"（`-`表示`blank`）映射到"hello"。然后，条件概率被定义为由$\mathcal{B}$映射到$\mathbf{l}$上的所有$\boldsymbol{\pi}$的概率之和：

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

$\boldsymbol{\pi}$的概率定义为$p(\boldsymbol{\pi}|\mathbf{y})=\prod_{t=1}^{T}y_{\pi_t}^{t}$，$y_{\pi_t}^{t}$是时刻$t$时有标签$\pi_t$的概率。由于存在指数级数量的求和项，直接计算方程1在计算上是不可行的。然而，使用[15]中描述的前向算法可以有效计算方程1。

2.3.2 Lexicon-free transcription

In this mode, the sequence $\mathbf{l}^*$ that has the highest probability as defined in Eq.1 is taken as the prediction. Since there exists no tractable algorithm to precisely find the solution, we use the strategy adopted in [15]. The sequence $\mathbf{l}^*$ is approximately found by $\mathbf{l}^{*}\approx\mathcal{B}(\arg\max_{\boldsymbol{\pi}} p(\boldsymbol{\pi}|\mathbf{y}))$, $i.e. taking the most probable label \pi_t$ $at each time stamp t$, $and map the resulted sequence onto \mathbf{l}^{*}$.

2.3.2 无字典转录

在这种模式下，将具有方程1中定义的最高概率的序列$\mathbf{l}^*$作为预测。由于不存在用于精确找到解的可行方法，我们采用[15]中的策略。序列$\mathbf{l}^*$通过$\mathbf{l}^{*}\approx\mathcal{B}(\arg\max_{\boldsymbol{\pi}} p(\boldsymbol{\pi}|\mathbf{y}))$近似发现，即在每个时间戳$t$采用最大概率的标签$\pi_t$，并将结果序列映射到$\mathbf{l}^{*}$。

2.3.3 Lexicon-based transcription

In lexicon-based mode, each test sample is associated with a lexicon $\mathcal{D}$. Basically, the label sequence is recognized by choosing the sequence in the lexicon that has highest conditional probability defined in Eq.1, i.e. $\mathbf{l}^* = \arg\max_{\mathbf{l}\in\mathcal{D}} p(\mathbf{l}|\mathbf{y})$. However, for large lexicons, e.g. the 50k-words Hunspell spell-checking dictionary [1], it would be very time-consuming to perform an exhaustive search over the lexicon, i.e. to compute Equation.1 for all sequences in the lexicon and choose the one with the highest probability. To solve this problem, we observe that the label sequences predicted via lexicon-free transcription, described in 2.3.2, are often close to the ground-truth under the edit distance metric. This indicates that we can limit our search to the nearest-neighbor candidates $\mathcal{N}_\delta(\mathbf{l}')$, where δ is the maximal edit distance and $\mathbf{l}'$ is the sequence transcribed from $\mathbf{y}$ in lexicon-free mode:

$$\mathbf{l}^* = \arg\max_{\mathbf{l}\in\mathcal{N}_\delta(\mathbf{l}')} p(\mathbf{l}|\mathbf{y}). \tag{2}$$

2.3.3 基于词典的转录

在基于字典的模式中，每个测试采样与词典$\mathcal{D}$相关联。基本上，通过选择词典中具有方程1中定义的最高条件概率的序列来识别标签序列，即$\mathbf{l}^* = \arg\max_{\mathbf{l}\in\mathcal{D}} p(\mathbf{l}|\mathbf{y})$。然而，对于大型词典，例如5万个词的Hunspell拼写检查词典[1]，对词典进行详尽的搜索是非常耗时的，即对词典中的所有序列计算方程1，并选择概率最高的一个。为了解决这个问题，我们观察到，2.3.2中描述的通过无词典转录预测的标签序列通常在编辑距离度量下接近于实际结果。这表示我们可以将搜索限制在最近邻候选目标$\mathcal{N}_\delta(\mathbf{l}')$，其中$\delta$是最大编辑距离，$\mathbf{l}'$是在无词典模式下从$\mathbf{y}$转录的序列：

$$\mathbf{l}^* = \arg\max_{\mathbf{l}\in\mathcal{N}_\delta(\mathbf{l}')} p(\mathbf{l}|\mathbf{y}). \tag{2}$$

The candidates $\mathcal{N}_\delta(\mathbf{l}')$ can be found efficiently with the BK-tree data structure[9], which is a metric tree specifically adapted to discrete metric spaces. The search time complexity of BK-tree is $O(\log|\mathcal{D}|)$, where $|\mathcal{D}|$ is the lexicon size. Therefore this scheme readily extends to very large lexicons. In our approach, a BK-tree is constructed offline for a lexicon. Then we perform fast online search with the tree, by finding sequences that have less or equal to δ edit distance to the query sequence.

可以使用BK树数据结构[9]有效地找到候选目标$\mathcal{N}_\delta(\mathbf{l}')$，这是一种专门适用于离散度量空间的度量树。BK树的搜索时间复杂度为$O(\log|\mathcal{D}|)$，其中$|\mathcal{D}|$是词典大小。因此，这个方案很容易扩展到非常大的词典。在我们的方法中，一个词典离线构造一个BK树。然后，我们使用树执行快速在线搜索，通过查找具有小于或等于δ编辑距离来查询序列。

2.4. Network Training

Denote the training dataset by $\mathcal{X} = \{I_i, \mathbf{l}_i\}_i$, where I_i is the training image and $\mathbf{l}_i$ is the ground truth label sequence. The objective is to minimize the negative log-likelihood of conditional probability of ground truth:

$$\mathcal{O} = -\sum_{I_i, \mathbf{l}_i \in \mathcal{X}} \log p(\mathbf{l}_i | \mathbf{y}_i), \tag{3}$$

where $\mathbf{y}{*i*}*is the sequence produced by the recurrent and convolutional layers from I*{i}$. This objective function calculates a cost value directly from an image and its ground truth label sequence. Therefore, the network can be end-to-end trained on pairs of images and sequences, eliminating the procedure of manually labeling all individual components in training images.

2.4. 网络训练

$\mathcal{X} = \{I_i, \mathbf{l}_i\}_i$表示训练集，$I_i$是训练图像，$\mathbf{l}_i$是真实的标签序列。目标是最小化真实条件概率的负对数似然：

$$\mathcal{O} = -\sum_{I_i, \mathbf{l}_i \in \mathcal{X}} \log p(\mathbf{l}_i | \mathbf{y}_i), \tag{3}$$

$\mathbf{y}{*i*}是循环层和卷积层从*I*{i}$生成的序列。目标函数直接从图像和它的真实标签序列计算代价值。因此，网络可以在成对的图像和序列上进行端对端训练，去除了在训练图像中手动标记所有单独组件的过程。

The network is trained with stochastic gradient descent (SGD). Gradients are calculated by the back-propagation algorithm. In particular, in the transcription layer, error differentials are back-propagated with the forward-backward algorithm, as described in [15]. In the recurrent layers, the Back-Propagation Through Time (BPTT) is applied to calculate the error differentials.

网络使用随机梯度下降（SGD）进行训练。梯度由反向传播算法计算。特别地，在转录层中，如[15]所述，误差使用前向算法进行反向传播。在循环层中，应用随时间反向传播（BPTT）来计算误差。

For optimization, we use the ADADELTA [37] to automatically calculate per-dimension learning rates. Compared with the conventional momentum [31] method, ADADELTA requires no manual setting of a learning rate. More importantly, we find that optimization using ADADELTA converges faster than the momentum method.

为了优化，我们使用ADADELTA[37]自动计算每维的学习率。与传统的动量[31]方法相比，ADADELTA不需要手动设置学习率。更重要的是，我们发现使用ADADELTA的优化收敛速度比动量方法快。

3. Experiments

To evaluate the effectiveness of the proposed CRNN model, we conducted experiments on standard benchmarks for scene text recognition and musical score recognition, which are both challenging vision tasks. The datasets and setting for training and testing are given in Sec. 3.1, the detailed settings of CRNN for scene text images is provided in Sec. 3.2, and the results with the comprehensive comparisons are reported in Sec. 3.3.

To further demonstrate the generality of CRNN, we verify the proposed algorithm on a music score recognition task in Sec. 3.4.

3. 实验

为了评估提出的CRNN模型的有效性，我们在场景文本识别和乐谱识别的标准基准数据集上进行了实验，这些都是具有挑战性的视觉任务。数据集和训练测试的设置见3.1小节，场景文本图像中CRNN的详细设置见3.2小节，综合比较的结果在3.3小节报告。为了进一步证明CRNN的泛化性，在3.4小节我们在乐谱识别任务上验证了提出的算法。

3.1. Datasets

For all the experiments for scene text recognition, we use the synthetic dataset (Synth) released by Jaderberg et al. [20] as the training data. The dataset contains 8 millions training images and their corresponding ground truth words. Such images are generated by a synthetic text engine and are highly realistic. Our network is trained on the synthetic data once, and tested on all other real-world test datasets without any fine-tuning on their training data. Even though the CRNN model is purely trained with synthetic text data, it works well on real images from standard text recognition benchmarks.

3.1. 数据集

对于场景文本识别的所有实验，我们使用Jaderberg等人[20]发布的合成数据集（Synth）作为训练数据。数据集包含8百万训练图像及其对应的实际单词。这样的图像由合成文本引擎生成并且是非常现实的。我们的网络在合成数据上进行了一次训练，并在所有其它现实世界的测试数据集上进行了测试，而没有在其训练数据上进行任何微调。即使CRNN模型是在纯合成文本数据上训练，但它在标准文本识别基准数据集的真实图像上工作良好。

Four popular benchmarks for scene text recognition are used for performance evaluation, namely ICDAR 2003 (IC03), ICDAR 2013 (IC13), IIIT 5k-word (IIIT5k), and Street View Text (SVT).

有四个流行的基准数据集用于场景文本识别的性能评估，即ICDAR 2003（IC03），ICDAR 2013（IC13），IIIT 5k-word（IIIT5k）和Street View Text (SVT)。

IC03 [27] test dataset contains 251 scene images with labeled text bounding boxes. Following Wang et al. [34], we ignore images that either contain non-alphanumeric characters or have less than three characters, and get a test set with 860 cropped text images. Each test image is associated with a 50-words lexicon which is defined by Wang et al. [34]. A full lexicon is built by combining all the per-image lexicons. In addition, we use a 50k words lexicon consisting of the words in the Hunspell spell-checking dictionary [1].

IC03[27]测试数据集包含251个具有标记文本边界框的场景图像。王等人[34]，我们忽略包含非字母数字字符或少于三个字符的图像，并获得具有860个裁剪的文本图像的测试集。每张测试图像与由Wang等人[34]定义的50词的词典相关联。通过组合所有的每张图像词汇构建完整的词典。此外，我们使用由Hunspell拼写检查字典[1]中的单词组成的5万个词的词典。

IC13 [24] test dataset inherits most of its data from IC03. It contains 1,015 ground truths cropped word images.

IC13[24]测试数据集继承了IC03中的大部分数据。它包含1015个实际的裁剪单词图像。

IIIT5k [28] contains 3,000 cropped word test images collected from the Internet. Each image has been associated to a 50-words lexicon and a 1k-words lexicon.

IIIT5k[28]包含从互联网收集的3000张裁剪的词测试图像。每张图像关联一个50词的词典和一个1000词的词典。

SVT [34] test dataset consists of 249 street view images collected from Google Street View. From them 647 word images are cropped. Each word image has a 50 words lexicon defined by Wang et al. [34].

SVT[34]测试数据集由从Google街景视图收集的249张街景图像组成。从它们中裁剪出了647张词图像。每张单词图像都有一个由Wang等人[34]定义的50个词的词典。

3.2. Implementation Details

The network configuration we use in our experiments is summarized in Table 1. The architecture of the convolutional layers is based on the VGG-VeryDeep architectures [32]. A tweak is made in order to make it suitable for recognizing English texts. In the 3rd and the 4th max-pooling layers, we adopt 1×2 sized rectangular pooling windows instead of the conventional squared ones. This tweak yields feature maps with larger width, hence longer feature sequence. For example, an image containing 10 characters is typically of size 100×32, from which a feature sequence 25 frames can be generated. This length exceeds the lengths of most English words. On top of that, the rectangular pooling windows yield rectangular receptive fields (illustrated in Fig. 2), which are beneficial for recognizing some characters that have narrow shapes, such as 'i' and 'l'.

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 × 32 gray-scale image

3.2. 实现细节

在实验中我们使用的网络配置总结在表1中。卷积层的架构是基于VGG-VeryDeep的架构[32]。为了使其适用于识别英文文本，对其进行了调整。在第3和第4个最大池化层中，我们采用1×2大小的矩形池化窗口而不是传统的平方形。这种调整产生宽度较大的特征图，因此具有更长的特征序列。例如，包含10个字符的图像通常为大小为100×32，可以从其生成25帧的特征序列。这个长度超过了大多数英文单词的长度。最重要的是，矩形池窗口产生矩形感受野（如图2所示），这有助于识别一些具有窄形状的字符，例如`i`和`l`。

表1。网络配置总结。第一行是顶层。`k`，`s`，`p`分别表示核大小，步长和填充大小。

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

The network not only has deep convolutional layers, but also has recurrent layers. Both are known to be hard to train. We find that the batch normalization [19] technique is extremely useful for training network of such depth. Two batch normalization layers are inserted after the 5th and 6th convolutional layers respectively. With the batch normalization layers, the training process is greatly accelerated.

网络不仅有深度卷积层，而且还有循环层。众所周知两者都难以训练。我们发现批归一化[19]技术对于训练这种深度网络非常有用。分别在第5和第6卷积层之后插入两个批归一化层。使用批归一化层训练过程大大加快。

We implement the network within the Torch7 [10] framework, with custom implementations for the LSTM units (in Torch7/CUDA), the transcription layer (in C++) and the BK-tree data structure (in C++). Experiments are carried out on a workstation with a 2.50 GHz Intel(R) Xeon(R) E5-2609 CPU, 64GB RAM and an NVIDIA(R) Tesla(TM) K40 GPU. Networks are trained with ADADELTA, setting the parameter ρ to 0.9. During training, all images are scaled to 100 × 32 in order to accelerate the training process. The training process takes about 50 hours to reach convergence. Testing images are scaled to have height 32. Widths are proportionally scaled with heights, but at least 100 pixels. The average testing time is 0.16s/sample, as measured on IC03 without a

lexicon. The approximate lexicon search is applied to the 50k lexicon of IC03, with the parameter δ set to 3. Testing each sample takes 0.53s on average.

我们在Torch7[10]框架内实现了网络，使用定制实现的LSTM单元（Torch7/CUDA），转录层（C++）和BK树数据结构（C++）。实验在具有2.50 GHz Intel（R）Xeon E5-2609 CPU，64GB RAM和NVIDIA（R）Tesla(TM) K40 GPU的工作站上进行。网络用ADADELTA训练，将参数ρ设置为0.9。在训练期间，所有图像都被缩放为100×32，以加快训练过程。训练过程大约需要50个小时才能达到收敛。测试图像缩放的高度为32。宽度与高度成比例地缩放，但至少为100像素。平均测试时间为0.16s/样本，在IC03上测得的，没有词典。近似词典搜索应用于IC03的50k词典，参数δ设置为3。测试每个样本平均花费0.53s。

3.3. Comparative Evaluation

All the recognition accuracies on the above four public datasets, obtained by the proposed CRNN model and the recent state-of-the-arts techniques including the approaches based on deep models [23, 22, 21], are shown in Table 2.

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

3.3. 比较评估

提出的CRNN模型在上述四个公共数据集上获得的所有识别精度以及最近的最新技术，包括基于深度模型[23,22,21]的方法如表2所示。

表2。四个数据集上识别准确率(%)。在第二行，“50”，“1k”，“50k”和“Full”表示使用的字典，“None”表示识别没有字典。*[22]严格意义上讲不是无字典的，因为它的输出限制在90K的字典。

	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

In the constrained lexicon cases, our method consistently outperforms most state-of-the-arts approaches, and in average beats the best text reader proposed in [22]. Specifically, we obtain superior performance on IIIT5k, and SVT compared to [22], only achieved lower performance on IC03 with the “Full” lexicon. Note that the model in[22] is trained on a specific dictionary, namely that each word is associated to a class label. Unlike [22], CRNN is not limited to recognize a word in a known dictionary, and able to handle random strings (e.g. telephone numbers), sentences or other scripts like Chinese words. Therefore, the results of CRNN are competitive on all the testing datasets.

在有约束词典的情况中，我们的方法始终优于大多数最新的方法，并且平均打败了[22]中提出的最佳文本阅读器。具体来说，与[22]相比，我们在IIIT5k和SVT上获得了卓越的性能，仅在IC03上通过“Full”词典实现了较低性能。请注意，[22]中的模型是在特定字典上训练的，即每个单词都与一个类标签相关联。与[22]不同，CRNN不限于识别已知字典中的单词，并且能够处理随机字符串（例如电话号码），句子或其他诸如中文单词的脚本。 因此，CRNN的结果在所有测试数据集上都具有竞争力。

In the unconstrained lexicon cases, our method achieves the best performance on SVT, yet, is still behind some approaches [8, 22] on IC03 and IC13. Note that the blanks in the “none” columns of Table 2 denote that such approaches are unable to be applied to recognition without lexicon or did not report the recognition accuracies in the unconstrained cases. Our method uses only synthetic text with word level labels as the training data, very different to PhotoOCR [8] which used 7.9 millions of real word images with character-level annotations for training. The best persformance is reported by [22] in the unconstrained lexicon cases, benefiting from its large

dictionary, however, it is not a model strictly unconstrained to a lexicon as mentioned before. In this sense, our results in the unconstrained lexicon case are still promising.

在无约束词典的情况下，我们的方法在SVT上仍取得了最佳性能，但在IC03和IC13上仍然落后于一些方法[8,22]。注意，表2的“none”列中的空白表示这种方法不能应用于没有词典的识别，或者在无约束的情况下不能报告识别精度。我们的方法只使用具有单词级标签的合成文本作为训练数据，与PhotoOCR[8]非常不同，后者使用790万个具有字符级标注的真实单词图像进行训练。[22]中报告的最佳性能是在无约束词典的情况下，受益于它的大字典，然而，它不是前面提到的严格的无约束词典模型。在这个意义上，我们在无限制词典表中的结果仍然是有前途的。

For further understanding the advantages of the proposed algorithm over other text recognition approaches, we provide a comprehensive comparison on several properties named E2E Train, Conv Ftrs, CharGT-Free, Unconstrained, and Model Size, as summarized in Table 3.

Table 3. Comparison among various methods. Attributes for comparison include: 1) being end-to-end trainable (E2E Train); 2) using convolutional features that are directly learned from images rather than using hand-crafted ones (Conv Ftrs); 3) requiring no ground truth bounding boxes for characters during training (CharGT-Free); 4) not confined to a pre-defined dictionary (Unconstrained); 5) the model size (if an end-to-end trainable model is used), measured by the number of model parameters (Model Size, M stands for millions).

	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**

为了进一步了解与其它文本识别方法相比，所提出算法的优点，我们提供了在一些特性上的综合比较，这些特性名称为E2E Train，Conv Ftrs，CharGT-Free，Unconstrained和Model Size，如表3所示。

表3。各种方法的对比。比较的属性包括：1)端到端训练(E2E Train)；2)从图像中直接学习卷积特征而不是使用手动设计的特征(Conv Ftrs)；3)训练期间不需要字符的实际边界框(CharGT-Free)；4)不受限于预定义字典(Unconstrained)；5)模型大小（如果使用端到端模型），通过模型参数数量来衡量(Model Size, M表示百万)。

	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**

E2E Train: This column is to show whether a certain text reading model is end-to-end trainable, without any preprocess or through several separated steps, which indicates such approaches are elegant and clean for training. As can be observed from Table 3, only the models based on deep neural networks including [22, 21] as well as CRNN have this property.

E2E Train：这一列是为了显示某种文字阅读模型是否可以进行端到端的训练，无需任何预处理或经过几个分离的步骤，这表明这种方法对于训练是优雅且干净的。从表3可以看出，只有基于深度神经网络的模型，包括[22,21]以及CRNN具有这种性质。

Conv Ftrs: This column is to indicate whether an approach uses the convolutional features learned from training images directly or handcraft features as the basic representations.

Conv Ftrs：这一列用来表明一个方法是否使用从训练图像直接学习到的卷积特征或手动特征作为基本的表示。

CharGT-Free: This column is to indicate whether the character-level annotations are essential for training the model. As the input and output labels of CRNN can be a sequence, character-level annotations are not necessary.

CharGT-Free：这一列用来表明字符级标注对于训练模型是否是必要的。由于CRNN的输入和输出标签是序列，因此字符级标注是不必要的。

Unconstrained: This column is to indicate whether the trained model is constrained to a specific dictionary, unable to handling out-of-dictionary words or random sequences. Notice that though the recent models learned by label embedding [5, 14] and incremental learning [22] achieved highly competitive performance, they are constrained to a specific dictionary.

Unconstrained：这一列用来表明训练模型是否受限于一个特定的字典，是否不能处理字典之外的单词或随机序列。注意尽管最近通过标签嵌入[5, 14]和增强学习[22]学习到的模型取得了非常有竞争力的性能，但它们受限于一个特定的字典。

Model Size: This column is to report the storage space of the learned model. In CRNN, all layers have weight-sharing connections, and the fully-connected layers are not needed. Consequently, the number of parameters of CRNN is much less than the models learned on the variants of CNN [22, 21], resulting in a much smaller model compared with [22, 21]. Our model has 8.3 million parameters, taking only 33MB RAM (using 4-bytes single-precision float for each parameter), thus it can be easily ported to mobile devices.

Model Size：这一列报告了学习模型的存储空间。在CRNN中，所有的层有权重共享连接，不需要全连接层。因此，CRNN的参数数量远小于CNN变体[22,21]所得到的模型，导致与[22,21]相比，模型要小得多。我们的模型有830万个参数，只有33MB RAM（每个参数使用4字节单精度浮点数），因此可以轻松地移植到移动设备上。

Table 3 clearly shows the differences among different approaches in details, and fully demonstrates the advantages of CRNN over other competing methods.

表3详细列出了不同方法之间的差异，充分展示了CRNN与其它竞争方法的优势。

In addition, to test the impact of parameter δ, we experiment different values of δ in Eq.2. In Fig.4 we plot the recognition accuracy as a function of δ. Larger δ results in more candidates, thus more accurate lexicon-based transcription. On the other hand, the computational cost grows with larger δ, due to longer BK-tree search time, as well as larger number of candidate sequences for testing. In practice, we choose $\delta = 3$ as a tradeoff between accuracy and speed.

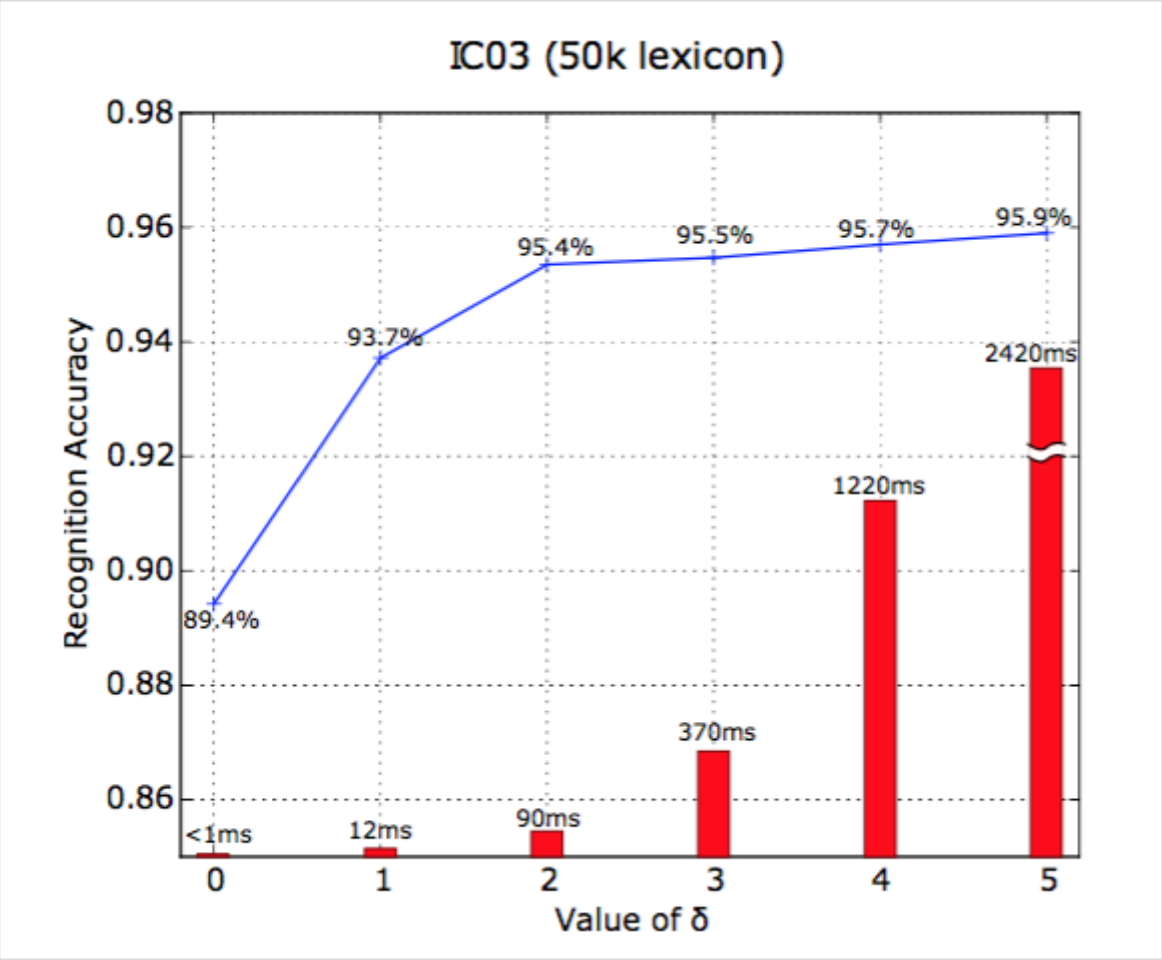

Figure 4. Blue line graph: recognition accuracy as a function parameter δ. Red bars: lexicon search time per sample. Tested on the IC03 dataset with the 50k lexicon.

另外，为了测试参数δ的影响，我们在方程2中实验了δ的不同值。在图4中，我们将识别精度绘制为δ的函数。更大的δ导致更多的候选目标，从而基于词典的转录更准确。另一方面，由于更长的BK树搜索时间，以及更大数量的候选序列用于测试，计算成本随着δ的增大而增加。实际上，我们选择$\delta = 3$作为精度和速度之间的折衷。

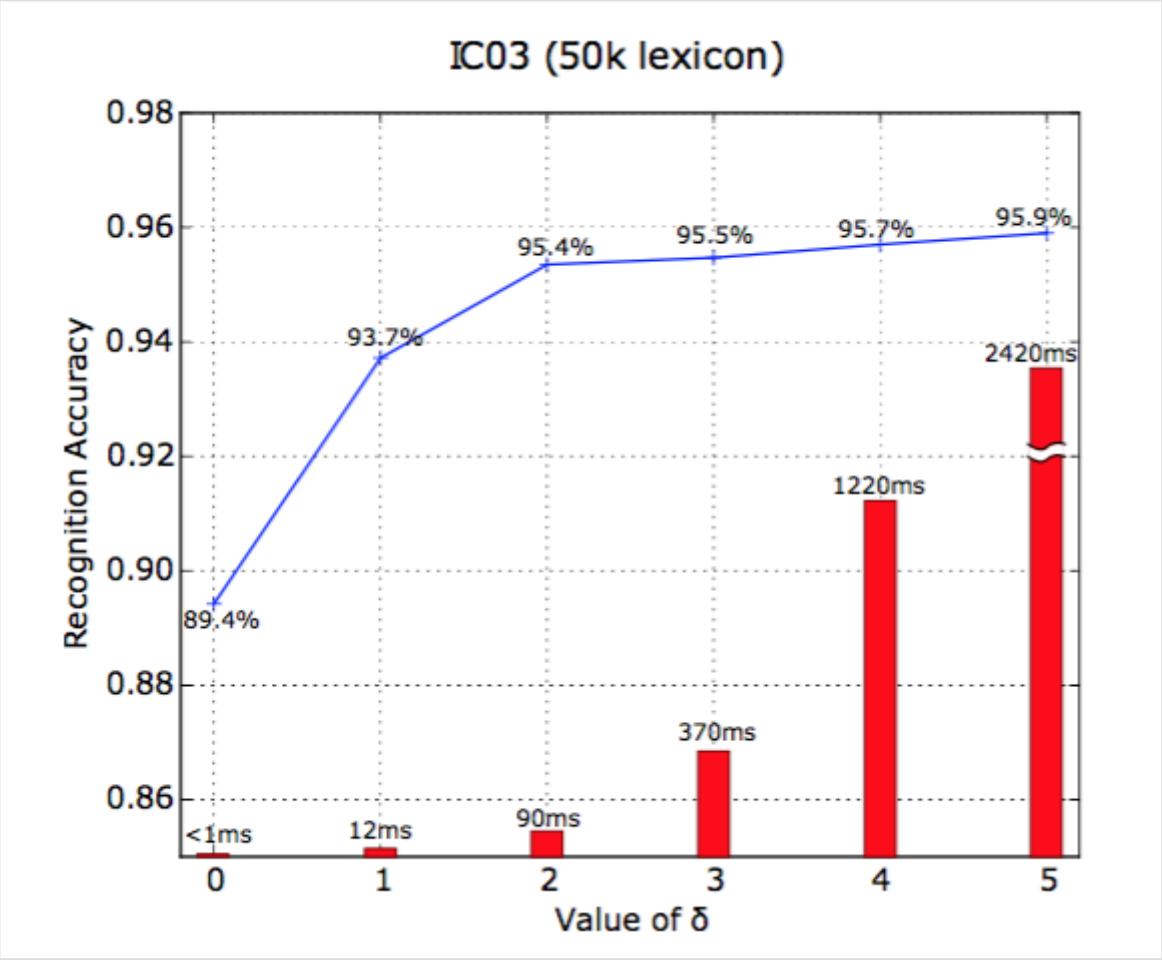

图4。蓝线图：识别准确率作为δ的函数。红条：每个样本的词典搜索时间。在IC03数据集上使用50k词典进行的测试。

3.4. Musical Score Recognition

A musical score typically consists of sequences of musical notes arranged on staff lines. Recognizing musical scores in images is known as the Optical Music Recognition (OMR) problem. Previous methods often requires image preprocessing (mostly binirization), staff lines detection and individual notes recognition [29]. We cast the OMR as a sequence recognition problem, and predict a sequence of musical notes directly from the image with CRNN. For simplicity, we recognize pitches only, ignore all chords and assume the same major scales (C major) for all scores.

3.4. 乐谱识别

乐谱通常由排列在五线谱的音符序列组成。识别图像中的乐谱被称为光学音乐识别（OMR）问题。以前的方法通常需要图像预处理（主要是二值化），五线谱检测和单个音符识别[29]。我们将OMR作为序列识别问题，直接用CRNN从图像中预测音符的序列。为了简单起见，我们仅认识音调，忽略所有和弦，并假定所有乐谱具有相同的大调音阶（C大调）。

To the best of our knowledge, there exists no public datasets for evaluating algorithms on pitch recognition. To prepare the training data needed by CRNN, we collect 2650 images from [2]. Each image contains a fragment of score containing 3 to 20 notes. We manually label the ground truth label sequences (sequences of not ezpitches) for all the images. The collected images are augmented to 265k training samples by being rotated, scaled and corrupted with noise, and by replacing their backgrounds with natural images. For testing, we create three datasets: 1) “Clean”, which contains 260 images collected from [2]. Examples are shown in Fig. 5.a; 2) “Synthesized”, which is created from “Clean”, using the augmentation strategy mentioned above. It contains 200 samples, some of which are shown in Fig. 5.b; 3) “Real-World”, which contains 200 images of score fragments taken from music books with a phone camera. Examples are shown in Fig. 5.c.

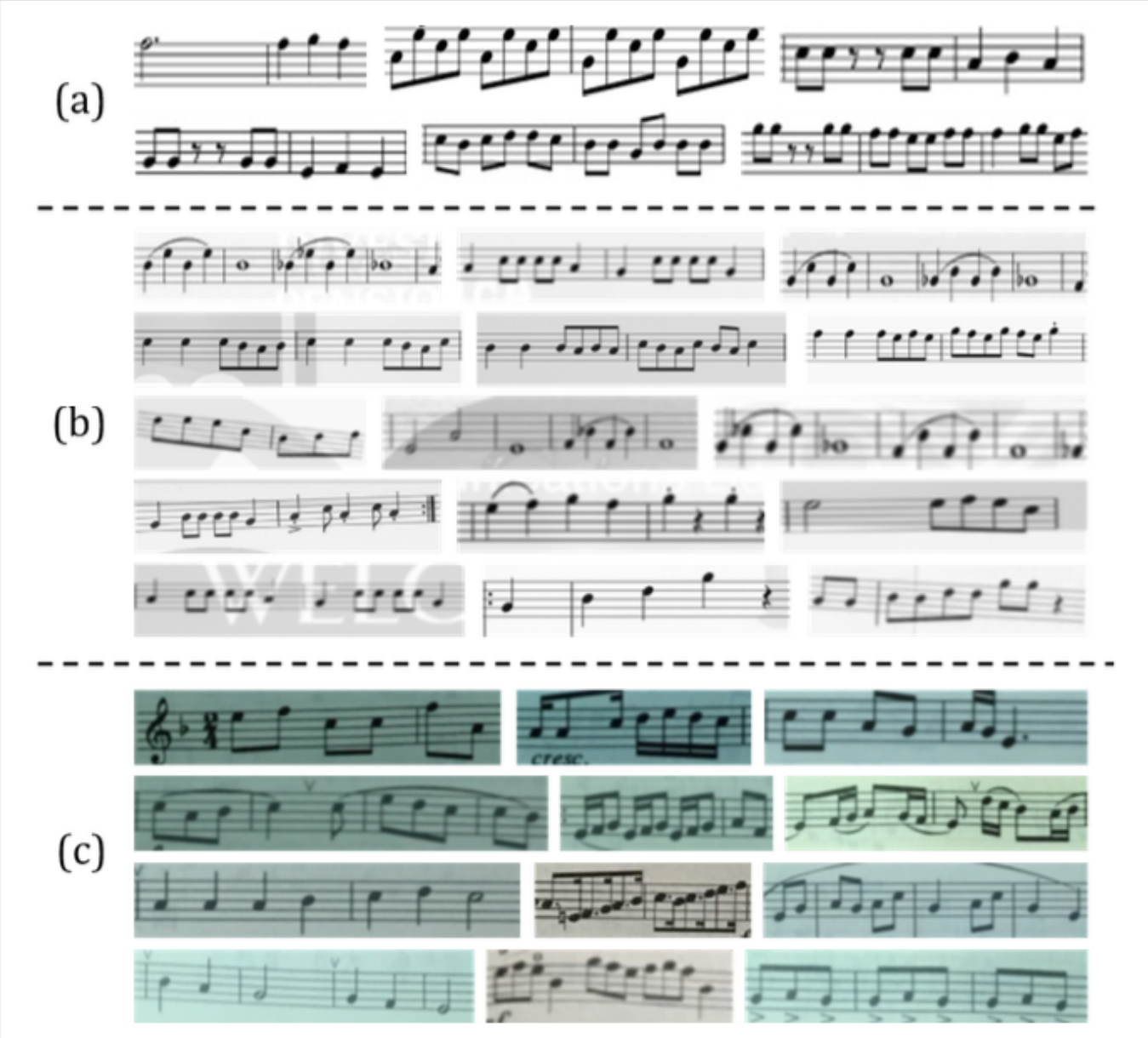

Figure 5. (a) Clean musical scores images collected from [2] (b) Synthesized musical score images. (c) Real-world score images taken with a mobile phone camera.

据我们所知，没有用于评估音调识别算法的公共数据集。为了准备CRNN所需的训练数据，我们从[2]中收集了2650张图像。每个图像中有一个包含3到20个音符的乐谱片段。我们手动标记所有图像的真实标签序列（不是的音调序列）。收集到的图像通过旋转，缩放和用噪声损坏增强到了265k个训练样本，并用自然图像替换它们的背景。对于测试，我们创建了三个数据集：1）“纯净的”，其中包含从[2]收集的260张图像。实例如图5.a所示；2）“合成的”，使用“纯净的”创建的，使用了上述的增强策略。它包含200个样本，其中一些如图5.b所示；3）“现实世界”，其中包含用手机相机拍摄的音乐书籍中的200张图像。例子如图5.c所示。

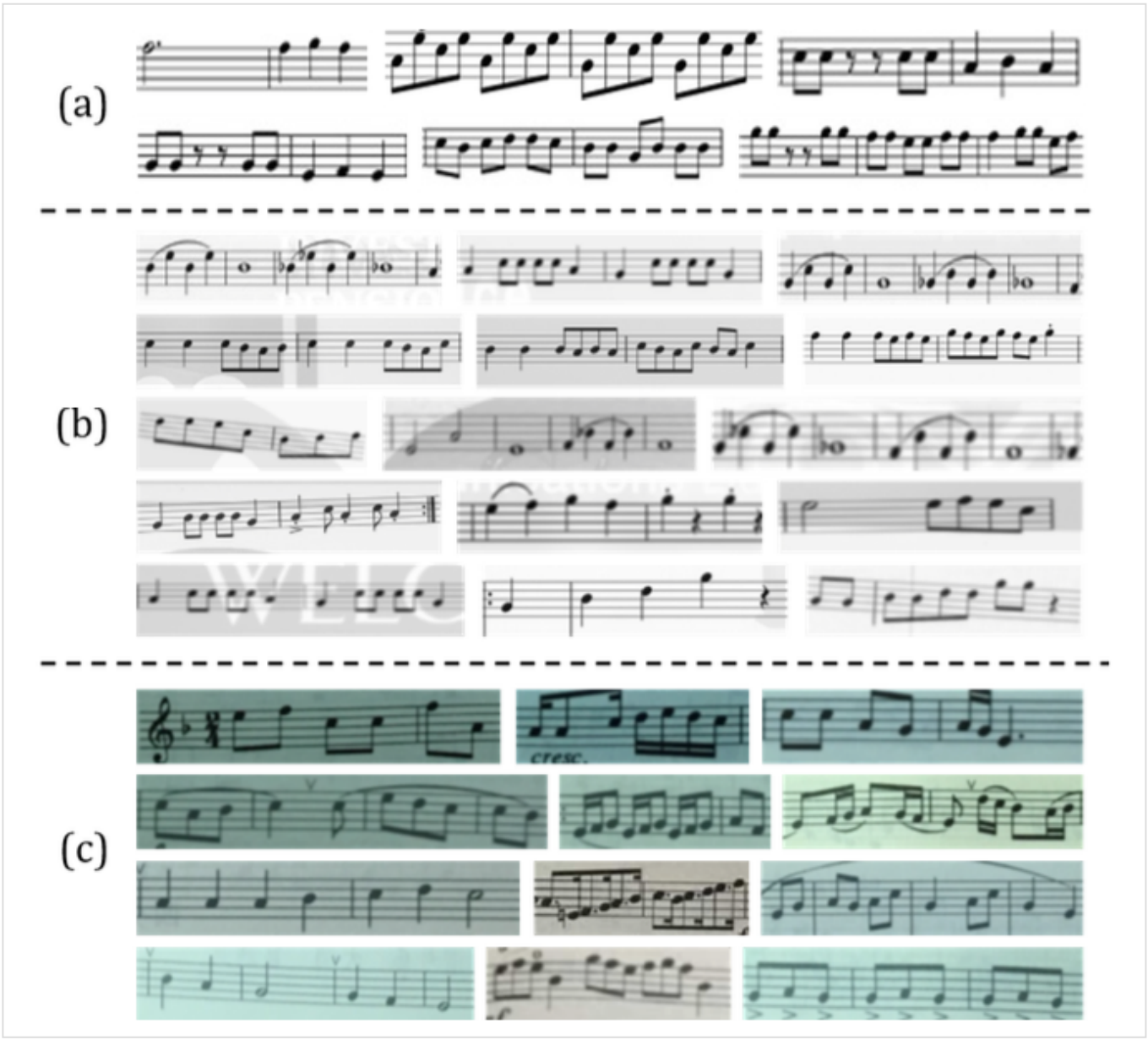

图5。(a)从[2]中收集的干净的乐谱图像。(b)合成的乐谱图像。(c)用手机相机拍摄的现实世界的乐谱图像。

Since we have limited training data, we use a simplified CRNN configuration in order to reduce model capacity. Different from the configuration specified in Tab. 1, the 4th and 6th convolution layers are removed, and the 2-layer bidirectional LSTM is replaced by a 2-layer single directional LSTM. The network is trained on the pairs of images and corresponding label sequences. Two measures are used for evaluating the recognition performance: 1) fragment accuracy, i.e. the percentage of score fragments correctly recognized; 2) average edit distance, i.e.

the average edit distance between predicted pitch sequences and the ground truths. For comparison, we evaluate two commercial OMR engines, namely the Capella Scan [3] and the PhotoScore [4].

由于我们的训练数据有限，因此我们使用简化的CRNN配置来减少模型容量。与表1中指定的配置不同，我们移除了第4和第6卷积层，将2层双向LSTM替换为2层单向LSTM。网络对图像对和对应的标签序列进行训练。使用两种方法来评估识别性能：1）片段准确度，即正确识别的乐谱片段的百分比；2）平均编辑距离，即预测音调序列与真实值之间的平均编辑距离。为了比较，我们评估了两种商用OMR引擎，即Capella Scan[3]和PhotoScore[4]。

Tab. 4 summarizes the results. The CRNN outperforms the two commercial systems by a large margin. The Capella Scan and PhotoScore systems perform reasonably well on the Clean dataset, but their performances drop significantly on synthesized and real-world data. The main reason is that they rely on robust binarization to detect staff lines and notes, but the binarization step often fails on synthesized and real-world data due to bad lighting condition, noise corruption and cluttered background. The CRNN, on the other hand, uses convolutional features that are highly robust to noises and distortions. Besides, recurrent layers in CRNN can utilize contextual information in the score. Each note is recognized not only itself, but also by the nearby notes. Consequently, some notes can be recognized by comparing them with the nearby notes, e.g. contrasting their vertical positions.

Table 4. Comparison of pitch recognition accuracies, among CRNN and two commercial OMR systems, on the three datasets we have collected. Performances are evaluated by fragment accuracies and average edit distance ("fragment accuracy/average edit distance").

	Clean	Synthesized	Real-World
Capella Scan [3]	51.9%/1.75	20.0%/2.31	43.5%/3.05
PhotoScore [4]	55.0%/2.34	28.0%/1.85	20.4%/3.00
CRNN	**74.6%/0.37**	**81.5%/0.30**	**84.0%/0.30**

表4总结了结果。CRNN大大优于两个商业系统。Capella Scan和PhotoScore系统在干净的数据集上表现相当不错，但是它们的性能在合成和现实世界数据方面显著下降。主要原因是它们依赖于强大的二值化来检五线谱和音符，但是由于光线不良，噪音破坏和杂乱的背景，二值化步骤经常会在合成数据和现实数据上失败。另一方面，CRNN使用对噪声和扭曲具有鲁棒性的卷积特征。此外，CRNN中的循环层可以利用乐谱中的上下文信息。每个音符不仅自身被识别，而且被附近的音符识别。因此，通过将一些音符与附近的音符进行比较可以识别它们，例如对比他们的垂直位置。

	Clean	Synthesized	Real-World
Capella Scan [3]	51.9%/1.75	20.0%/2.31	43.5%/3.05
PhotoScore [4]	55.0%/2.34	28.0%/1.85	20.4%/3.00
CRNN	**74.6%/0.37**	**81.5%/0.30**	**84.0%/0.30**

表4。在我们收集的数据集上，CRNN和两个商业OMR系统对音调识别准确率的对比。通过片段准确率和平均编辑距离(“片段准确率/平均编辑距离”)来评估性能。

The results have shown the generality of CRNN, in that it can be readily applied to other image-based sequence recognition problems, requiring minimal domain knowledge. Compared with Capella Scan and PhotoScore, our CRNN-based system is still preliminary and misses many functionalities. But it provides a new scheme for OMR, and has shown promising capabilities in pitch recognition.

结果显示了CRNN的泛化性，因为它可以很容易地应用于其它的基于图像的序列识别问题，需要极少的领域知识。与Capella Scan和PhotoScore相比，我们的基于CRNN的系统仍然是初步的，并且缺少许多功能。但它为OMR提供了一个新的方案，并且在音高识别方面表现出有前途的能力。

4. Conclusion

In this paper, we have presented a novel neural network architecture, called Convolutional Recurrent Neural Network (CRNN), which integrates the advantages of both Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN). CRNN is able to take input images of varying dimensions and produces predictions with different lengths. It directly runs on coarse level labels (e.g. words), requiring no detailed annotations for each individual element (e.g. characters) in the training phase. Moreover, as CRNN abandons fully connected layers used in conventional neural networks, it results in a much more compact and efficient model. All these properties make CRNN an excellent approach for image-based sequence recognition.

4. 总结

在本文中，我们提出了一种新颖的神经网络架构，称为卷积循环神经网络（CRNN），其集成了卷积神经网络（CNN）和循环神经网络（RNN）的优点。CRNN能够获取不同尺寸的输入图像，并产生不同长度的预测。它直接在粗粒度的标签（例如单词）上运行，在训练阶段不需要详细标注每一个单独的元素（例如字符）。此外，由于CRNN放弃了传统神经网络中使用的全连接层，因此得到了更加紧凑和高效的模型。所有这些属性使得CRNN成为一种基于图像序列识别的极好方法。

The experiments on the scene text recognition benchmarks demonstrate that CRNN achieves superior or highly competitive performance, compared with conventional methods as well as other CNN and RNN based algorithms. This confirms the advantages of the proposed algorithm. In addition, CRNN significantly outperforms other competitors on a benchmark for Optical Music Recognition (OMR), which verifies the generality of CRNN.

在场景文本识别基准数据集上的实验表明，与传统方法以及其它基于CNN和RNN的算法相比，CRNN实现了优异或极具竞争力的性能。这证实了所提出的算法的优点。此外，CRNN在光学音乐识别（OMR）的基准数据集上显著优于其它的竞争者，这验证了CRNN的泛化性。

Actually, CRNN is a general framework, thus it can be applied to other domains and problems (such as Chinese character recognition), which involve sequence prediction in images. To further speed up CRNN and make it more practical in real-world applications is another direction that is worthy of exploration in the future.

实际上，CRNN是一个通用框架，因此可以应用于其它的涉及图像序列预测的领域和问题（如汉字识别）。进一步加快CRNN，使其在现实应用中更加实用，是未来值得探索的另一个方向。

Acknowledgement

This work was primarily supported by National Natural Science Foundation of China (NSFC) (No. 61222308).

致谢

这项工作主要是由中国国家自然科学基金(NSFC)支持 (No. 61222308)。

References

[1] http://hunspell.sourceforge.net/. 4, 5

[2] https://musescore.com/sheetmusic. 7, 8

[3] http://www.capella.de/us/index.cfm/products/capella-scan/info-capella-scan/. 8

[4] http://www.sibelius.com/products/photoscore/ultimate.html. 8

[5] J. Almazá n, A. Gordo, A. Forné s, and E. Valveny. Word spotting and recognition with embedded attributes. PAMI, 36(12):2552–2566, 2014. 2, 6, 7

[6] O. Alsharif and J. Pineau. End-to-end text recognition with hybrid HMM maxout models. ICLR, 2014. 6, 7

[7] Y. Bengio, P. Y. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. NN, 5(2):157–166, 1994. 3

[8] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven. Photoocr: Reading text in uncontrolled conditions. In ICCV, 2013. 1, 2, 6, 7

[9] W. A. Burkhard and R. M. Keller. Some approaches to best-match file searching. Commun. ACM, 16(4):230–236, 1973.4

[10] R. Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In BigLearn, NIPS Workshop, 2011. 6

[11] F. A. Gers, N. N. Schraudolph, and J. Schmidhuber. Learning precise timing with LSTM recurrent networks. JMLR, 3:115–143, 2002. 3

[12] R. B. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In CVPR, 2014. 1, 3

[13] V. Goel, A. Mishra, K. Alahari, and C. V. Jawahar. Whole is greater than sum of parts: Recognizing scene text words. In ICDAR, 2013. 6, 7

[14] A. Gordo. Supervised mid-level features for word image representation. In CVPR, 2015. 2, 6, 7

[15] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In ICML, 2006. 4, 5

[16] A. Graves, M. Liwicki, S. Fernandez, R. Bertolami, H. Bunke, and J. Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. PAMI, 31(5):855–868, 2009. 2

[17] A. Graves, A. Mohamed, and G. E. Hinton. Speech recognition with deep recurrent neural networks. In ICASSP, 2013. 3

[18] S. Hochreiter and J. Schmidhuber. Long short-term memory. Neural Computation, 9(8):1735–1780, 1997. 3

[19] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In ICML, 2015. 6

[20] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. NIPS Deep Learning Workshop, 2014. 5

[21] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Deep structured output learning for unconstrained text recog- nition. In ICLR, 2015. 6, 7

[22] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. IJCV (Accepted), 2015. 1, 2, 3, 6, 7

[23] M. Jaderberg, A. Vedaldi, and A. Zisserman. Deep features for text spotting. In ECCV, 2014. 2, 6, 7

[24] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. Almazá n, and L. de las Heras. ICDAR 2013 robust reading competition. In ICDAR, 2013. 5

[25] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012. 1, 3

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. Proceed- ings of the IEEE, 86(11):2278–2324, 1998. 1

[27] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J. Jolion, L. Todoran, M. Worring, and X. Lin. ICDAR 2003 robust reading competitions: entries, results, and future directions. IJDAR, 7(2-3):105–122, 2005. 5

[28] A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In BMVC, 2012. 5, 6, 7

[29] A. Rebelo, I. Fujinaga, F. Paszkiewicz, A. R. S. Març al, C. Guedes, and J. S. Cardoso. Optical music recognition: state-of-the-art and open issues. IJMIR, 1(3):173–190, 2012. 7

[30] J. A. Rodŕ ıguez-Serrano, A. Gordo, and F. Perronnin. Label embedding: A frugal baseline for text recognition. IJCV, 113(3):193–207, 2015. 2, 6, 7

[31] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Neurocomputing: Foundations of research. chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, 1988. 5

[32] K. Simonyan and A. Zisserman. Very deep convolu- tional networks for large-scale image recognition. CoRR, abs/1409.1556, 2014. 5

[33] B. Su and S. Lu. Accurate scene text recognition based on recurrent neural network. In ACCV, 2014. 2, 6, 7

[34] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In ICCV, 2011. 5, 6, 7

[35] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In ICPR, 2012. 1, 6, 7

[36] C. Yao, X. Bai, B. Shi, and W. Liu. Strokelets: A learned multi-scale representation for scene text recognition. In CVPR, 2014. 2, 6, 7

[37] M. D. Zeiler. ADADELTA: anadaptive learning rate method. CoRR, abs/1212.5701, 2012. 5

Deep Learning

‹ CRNN论文翻译——中文版

Batch Normalization论文翻译——中文版 ›