The Thirty-Second AAAI Conference
on Artificial Intelligence (AAAI-18)

Char-Net: A Character-Aware Neural Network for Distorted Scene Text Recognition

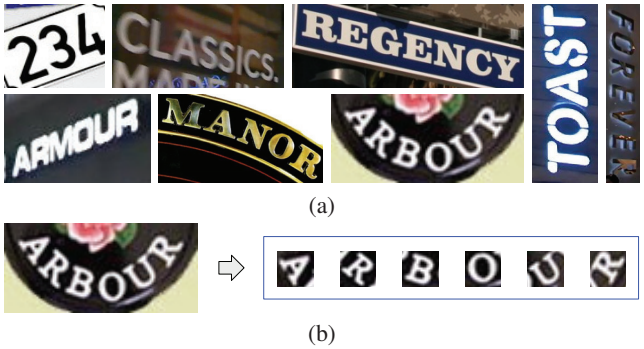
Wei Liu, Chaofeng Chen, Kwan-Yee K. Wong
Department of Computer Science, The University of Hong Kong
{wliu, cfchen, kykwong}@cs.hku.hk

Abstract

In this paper, we present a Character-Aware Neural Network (Char-Net) for recognizing distorted scene text. Our Char-Net is composed of a word-level encoder, a character-level encoder, and a LSTM-based decoder. Unlike previous work which employed a global spatial transformer network to rectify the entire distorted text image, we take an approach of detecting and rectifying individual characters. To this end, we introduce a novel hierarchical attention mechanism (HAM) which consists of a recurrent RoIWarp layer and a character-level attention layer. The recurrent RoIWarp layer sequentially extracts a feature region corresponding to a character from the feature map produced by the word-level encoder, and feeds it to the character-level encoder which removes the distortion of the character through a simple spatial transformer and further encodes the character region. The character-level attention layer then attends to the most relevant features of the feature map produced by the character-level encoder and composes a context vector, which is finally fed to the LSTM-based decoder for decoding. This approach of adopting a simple local transformation to model the distortion of individual characters not only results in an improved efficiency, but can also handle different types of distortion that are hard, if not impossible, to be modelled by a single global transformation. Experiments have been conducted on six public benchmark datasets. Our results show that Char-Net can achieve state-of-the-art performance on all the benchmarks, especially on the IC-IST which contains scene text with large distortion. Code will be made available.

Introduction

Recently, scene text recognition has been receiving much attention as it is fundamental in extracting textual information embedded in natural scenes. With the increased popularity of wearable cameras such as GoPro, more and more images are captured under arbitrary poses. This inevitably introduces different kinds of distortion in the text appearing in these images (see Figure 1a). Although remarkable results have been reported in recognizing undistorted scene text (Wang et al. 2012; Jaderberg et al. 2014; 2015a; He et al. 2016b; Shi, Bai, and Yao 2016; Lee and Osindero 2016), it remains a challenge to build a robust text recognizer that can handle highly distorted scene text effectively and efficiently.



(a)

(b)

Figure 1: (a) Some examples of scene text suffering from different kinds of distortion. (b) A text along a curve can be considered a sequence of individually rotated characters.

To the best of our knowledge, there exist two methods (Shi et al. 2016; Liu et al. 2016) that employ global spatial transformer networks (Jaderberg et al. 2015b) to rectify the distorted text. They both adopt a complicated thin-plate spline (TPS) transformation to model different types of distortion in the scene text. As their spatial transformer networks are only optimized with the weak supervision of the recognition loss, they have difficulty in precisely locating the fiducial points which tightly bound the text region. This leads to error in the estimation of the TPS transformation, and hence the deformation of the scene text, from the fiducial points.

In this work, instead of estimating a global transformation and rectifying the entire scene text, we take the approach of detecting and rectifying individual characters in the distorted text. We observe that by considering characters in the distorted text separately, the distortion modelled by a complicated transformation can actually be described by some much simpler local transformations of the individual characters. For instance, a text along a curve can be regarded as a sequence of individually rotated characters (see Figure 1b). Hence, we can remove the distortion of the scene text by rectifying each individual character through a simple local transformation (e.g., rotation). This approach of adopting a simple local transformation to model the distortion of indi-

vidual characters not only results in an improved efficiency, but can also handle different types of distortion that are hard, if not impossible, to be modelled by a single global transformation.

Driven by the above observation, we present a Character-Aware Neural Network (Char-Net) for recognizing distorted scene text. Our Char-Net is composed of a word-level encoder, a character-level encoder, and a LSTM-based decoder. We introduce a novel hierarchical attention mechanism (HAM) that bridges the word-level encoder with the character-level encoder, and the character-level encoder with the LSTM-based decoder. The newly proposed HAM consists of two layers, namely the recurrent RoIWarp layer and the character-level attention layer. The recurrent RoIWarp layer sequentially extracts a feature region corresponding to a character from the feature map produced by the word-level encoder, and feeds it to the character-level encoder which removes the distortion of the character through a simple spatial transformer and further encodes the character region. The character-level attention layer then attends to the most relevant features of the feature map produced by the character-level encoder and composes a context vector, which is finally fed to the LSTM-based decoder for decoding. Equipped with the HAM, our Char-Net is capable of handling complicated distortion exhibited in scene text both efficiently and effectively, by rectifying individual characters through simple local transformations.

In summary, the key contributions of this work are:

- A simple and efficient Character-Aware Neural Network (Char-Net) for distorted scene text recognition. The whole network can be trained in an end-to-end fashion using only text images and their corresponding character labels. Experimental results on six public benchmarks not only show that our Char-Net can achieve state-of-the-art performance, but also demonstrate the effectiveness of each of its components.
- A novel hierarchical attention mechanism that facilitates the rectification of individual characters and attends to the most relevant features for recognizing individual characters.
- A character-level encoder that removes distortion of individual characters using a simple local spatial transformer, and enables our Char-Net to handle different types of deformation exhibited in the scene text.

Related Work

Scene Text Recognition Scene text recognition has made significant progress in recent years due to the great successes in deep neural networks. For lexicon constrained methods, Wang et al. (Wang et al. 2012) and Jaderberg et al. (Jaderberg, Vedaldi, and Zisserman 2014) performed character recognition by CNNs, and they grouped the predicted characters in a left-to-right manner to output the final word predictions. Instead of character based recognition, (Jaderberg et al. 2014) and (Jaderberg et al. 2016) directly extracted CNN features from the entire word image to do a 90k-word classification (90k being the size of a pre-defined dictionary). For unconstrained scene text recognition, Jaderberg et al. (Jaderberg et al. 2015a) employed an architecture of two CNNs with one CNN to predict characters and the other to detect N-grams contained in the word image. As recurrent neural networks (RNNs) become popular in sequence recognition, recent scene text recognizers (Shi, Bai, and Yao 2016; Shi et al. 2016; Lee and Osindero 2016; Liu et al. 2016) use both CNNs and RNNs to encode features of word images. Furthermore, inspired by the successes (Bahdanau, Cho, and Bengio 2014) in Neural Machine translation, Shi et al. (Shi et al. 2016) and Lee et al. (Lee and Osindero 2016) both introduced a learnable attention mechanism in their RNNs to automatically select the most relevant features for recognizing individual characters. The literature is relatively sparse when it comes to recognizing distorted scene text. Phan et al. (Phan et al. 2013) employed a SIFT descriptor matching to handle perspective distortion exhibited in the scene text. In order to handle a more general distortion, a spatial transformer network (Jaderberg et al. 2015b) was introduced in (Shi et al. 2016; Liu et al. 2016) to rectify the entire text. Different from these two methods, our Char-Net is capable of rectifying individual characters and can therefore handle more complicated forms of distortion that cannot be modelled by a single global transformation easily.

Network Architecture

In this section, we describe the architecture of our Character-Aware Neural Network (Char-Net) for distorted scene text recognition. As illustrated in Figure 2, our Char-Net is composed a word-level encoder, a recurrent RoIWarp layer, a character-level encoder, a character-level attention layer, and a LSTM-based decoder. Here, the recurrent RoIWarp layer and the character-level attention layer form the core of our hierarchical attention mechanism (HAM). We describe the details of each of these components in the following subsections. Throughout this paper, we denote the ground truth labelling of a text image as $\mathbf{y} = \{y^1, y^2, ..., y^T, y^{T+1}\}$, where T is the length of the text, and y^{T+1} is the end-of-string (eos) token representing the end of the labelling. We refer to the process of predicting one character from the text image as being one time step/decoding step.

Word-Level Encoder We first employ a word-level encoder (WLE) to encode the entire text image. The proposed WLE takes the form of a CNN. It takes a single gray image $\mathbf{I} \in \mathbb{R}^{W \times H}$ as input, where W and H denote the width and height of the image respectively, and produces a three-dimensional feature map

$$\mathbf{F} = \text{WLE}(\mathbf{I}), \tag{1}$$

with a dimension of $W_f \times H_f \times C_f$, where W_f, H_f and C_f represent the width, height and number of channels respectively. To recognize the label y^t of the character at time step t, the recurrent RoIWarp layer of our HAM extracts a small feature region $\mathbf{F}_c^t$ from $\mathbf{F}$ that corresponds to the character being recognized, and feeds it to the character-level encoder. This process requires the feature map produced by WLE contains not only semantic information but also spatial

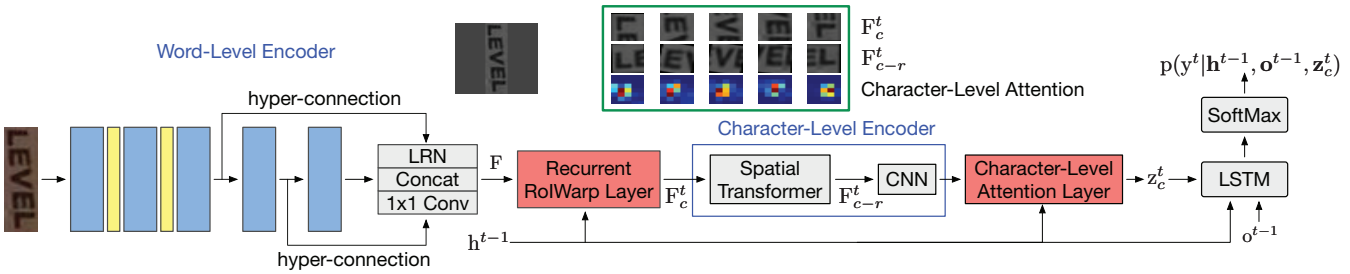


Figure 2: Overall architecture of the Char-Net. Blue and yellow rectangles in the word-level encoder represent the convolutional blocks and the max-pooling layers, respectively. Three convolutional feature maps from different levels of the CNN are normalised (local response normalization), concatenated and dimension-reduced (1×1 convolution) to produce the feature map $\mathbf{F}$. Two red rectangles denote the two layers of the hierarchical attention mechanism. The corresponding image patches of $\mathbf{F}_c^t$ and $\mathbf{F}_{c-r}^t$ together with the character-level attentions are shown in the green rectangle.

information of each character. As features in the final layer of a deep CNN are semantically strong but spatially coarse, our WLE stacks convolutional feature maps from different levels by several hyper-connections (Kong et al. 2016; He et al. 2017) to increase the spatial information encoded in the outputted feature map (refer to the word-level encoder depicted in Figure 2).

Character-Level Encoder The character-level encoder (CLE) consists of a local spatial transformer and a CNN. It takes the small feature region $\mathbf{F}_c^t$ extracted by the recurrent RoIWarp layer as input. The spatial transformer of the CLE enables our Char-Net to handle complicated distortion exhibited in the scene text. Unlike previous work (Shi et al. 2016; Liu et al. 2016) which employed a global spatial transformer network (STN) to rectify the entire text image, the local spatial transformer of the CLE targets at removing the distortion of an individual character

$$\mathbf{F}_{c\text{-}r}^t = \text{STN}(\mathbf{F}_c^t). \tag{2}$$

Here $\mathbf{F}_{c\text{-}r}^t$ is the rectified feature map having the same dimension as $\mathbf{F}_c^t$. Comparing with those global STNs which adopted a complicated thin-plate spline transformation (Bookstein 1989) to model different types of distortion, the local STN of the CLE can effectively model the complicated distortion of the scene text by simply predicting the rotation of each individual character (refer to the visualisation of the rectified character images in Figure 2). A small CNN is employed to further encode the rectified feature map $\mathbf{F}_{c\text{-}r}^t$ so as to extract more semantic features for the decoding step.

Hierarchical Attention Mechanism As mentioned previously, our hierarchical attention mechanism (HAM) consists of two layers, namely the recurrent RoIWarp layer and the character-level attention layer. The recurrent RoIWarp layer connects the word-level encoder to the character-level encoder. It is responsible for extracting a small feature region $\mathbf{F}_c^t$ that corresponds to the current character being recognized from the entire feature map $\mathbf{F}$ produced by the word-level encoder, and feeding it to the character-level encoder for further processing. The character-level attention layer connects the character-level encoder to the LSTM-based decoder. It is responsible for attending to the most relevant features of the feature map produced by the character-level encoder and computing a context vector $\mathbf{z}_c^t$ for the LSTM-based decoder. We describe the details of these layers in the next section.

LSTM-based Decoder The LSTM-based decoder recurrently predicts the ground truth labelling $\mathbf{y}$ using a Long Short-Term Memory (LSTM) layer. Let L denote the set of labels. At the decoding step t, the LSTM layer defines a probability distribution over L as

$$\mathbf{h}^t = \text{LSTM}(\mathbf{h}^{t-1}, \mathbf{o}^{t-1}, \mathbf{z}_c^t) \tag{3}$$

$$p(y^t|\mathbf{h}^{t-1}, \mathbf{o}^{t-1}, \mathbf{z}_c^t) = \text{SoftMax}(\mathbf{W}_y \mathbf{h}^t), \tag{4}$$

where $\mathbf{h}^{t-1}$ and $\mathbf{h}^t$ denote the previous and current hidden states respectively, $\mathbf{W}_y$ is the parameter matrix, y^t is the label of the current predicted character, and $\mathbf{o}^{t-1}$ is the one-hot encoding of the previously predicted character. Note that the implementation of LSTM follows the one in (Graves 2013), and $\mathbf{o}^{t-1}$ implicitly introduces a (learned) language model to assist the prediction of each character. The probability of the sequential labelling is then given by the product of the probability of each label, i.e.,

$$p(\mathbf{y}|\mathbf{I}) = \prod_{t=1}^{T+1} p(y^t|\mathbf{h}^{t-1}, \mathbf{o}^{t-1}, \mathbf{z}_c^t). \tag{5}$$

In the training process, we minimize the sum of the negative log-likelihood of Eq. (5) over the whole training dataset. During testing, we directly pick the label with the highest probability in Eq. (4) as the output in each decoding step.

Hierarchical Attention Mechanism

Given the feature map $\mathbf{F}$ of the text image produced by the word-level encoder, our hierarchical attention mechanism (HAM) aims at producing a context vector $\mathbf{z}_c^t$ for predicting the label y^t of the character being considered at time step t. As mentioned previously, HAM consists of two layers, namely the recurrent RoIWarp layer and the character-level attention layer (refer to the red rectangles in Figure 2). We describe the details of these two layers in the following subsections.

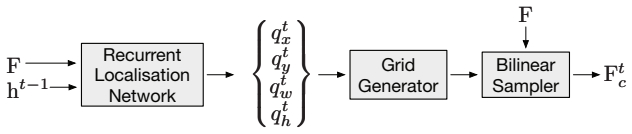


Figure 3: The detailed structure of the recurrent RoIWarp layer.

Preliminary: Traditional Attention Mechanism

Before introducing each layer of HAM, we first briefly review the traditional attention mechanism (Bahdanau, Cho, and Bengio 2014). In tradition attention mechanism, a context vector $\mathbf{z}_c^t$ is computed as a weighted sum of all the feature points in a feature map $\mathbf{F}$

$$\mathbf{z}_c^t = \sum_{i=1}^{W_f}\sum_{j=1}^{H_f} \boldsymbol{\alpha}_{ij}^t \mathbf{F}_{ij}, \tag{6}$$

where $\mathbf{F}_{ij}$ denotes a C_f-dimensional feature vector at (i, j), and $\boldsymbol{\alpha}^t$ denotes a set of weights for the feature vectors. Given the previous hidden state $\mathbf{h}^{t-1}$ of the LSTM-based decoder and the feature map $\mathbf{F}$ of the text image, the weight set $\boldsymbol{\alpha}^t$ at time t can be generated by

$$\mathbf{s}_{ij}^t = \mathbf{w}^\intercal \mathrm{Tanh}(\mathbf{M}\mathbf{h}^{t-1} + \mathbf{V}\mathbf{F}_{ij}), \tag{7}$$

$$\boldsymbol{\alpha}^t = \mathrm{SoftMax}(\mathbf{s}^t), \tag{8}$$

where $\mathbf{M}$ and $\mathbf{V}$ are the parameter matrices, $\mathbf{w}$ is a parameter vector and $\mathbf{s}^t$ is the score map for all the feature points.

Recurrent RoIWarp Layer

The recurrent RoIWarp layer of HAM aims at sequentially attending to a region of the feature map that corresponds to a character being considered at each time step. At time step t, the recurrent RoIWarp layer automatically extracts a small feature region $\mathbf{F}_c^t$ based on the predicted location of the t-th character

$$\mathbf{F}_c^t = \mathrm{RRoIWarp}(\mathbf{F}, \mathbf{h}^{t-1}). \tag{9}$$

This greatly narrows down the range of attention for computing the context vector $\mathbf{z}_c^t$. As illustrated in Figure 3, the recurrent RoIWarp layer is composed of three components, namely the recurrent localization network, the grid generator and the bilinear sampler. All these three components are differentiable and can be optimized using a gradient descent algorithm.

Recurrent Localization Network The recurrent localization network (RLN) is responsible for recurrently locating each character region of interest. Inspired by the traditional attention mechanism, we directly use the score map $\mathbf{s}^t$ computed by Equation (7) to predict the spatial information of each region-of-interest

$$(q_x^t, q_y^t, q_w^t, q_h^t) = \mathrm{MLP}_l(\mathbf{s}^t), \tag{10}$$

where MLP_l is a multilayer perceptron, (q_x^t, q_y^t) are the coordinates of the predicted center of the character region, and q_w^t and q_h^t are the predicted width and height of the character region respectively. Following (Jaderberg et al. 2015b), we use normalized coordinates for the prediction of $(q_x^t, q_y^t, q_w^t, q_h^t)$ so that $-1 \le q_x^t, q_y^t, q_w^t, q_h^t \le 1$. In the final layer of MLP, we add an extra Tanh activation layer to ensure that each predicted center point is within the convolutional feature map.

Note that we do not have any direct supervision over the location and size of each character of interest. During the whole training procedure, we only use the recognition objective function to update all the parameters of the whole network. It is therefore very difficult to optimize the recurrent localization network from scratch. Hence, we pre-train a variant of the traditional attention mechanism to ease the difficulty in training the recurrent localization network. At each decoding step, the character features of interest are restricted to a small continuous region of the feature map $\mathbf{F}$. The generated weight set $\boldsymbol{\alpha}^t$ in Equation (8) can be interpreted as an attention distribution over all the feature points in the convolutional feature map. Instead of generating an 'unconstrained' distribution by normalizing the relevancy score map $\mathbf{s}^t$, we model the attention distribution as a 2-D Gaussian distribution and calculate its parameters by

$$(\mu_x^t, \mu_y^t, \sigma_x^t, \sigma_y^t) = \mathrm{MLP}_g(\mathbf{s}^t), \tag{11}$$

where MLP_g is a multilayer perceptron for predicting the Gaussian distribution, and (μ_x^t, μ_y^t) and (σ_x^t, σ_y^t) are center and standard deviations of the distribution respectively. Similar to the traditional attention mechanism, this Gaussian attention mechanism can be easily optimized in an end-to-end manner. We then directly use the parameters of the Gaussian attention mechanism to initialize our recurrent localization network. More implementation details can be found in the experiment section.

Grid Generator and Bilinear Sampler The grid generator and bilinear sampler target at cropping out the character of interest at each time step and warping it into a fixed size $W_c \times H_c \times C_f$, where $W_c \times H_c$ gives the spatial resolution of each outputted feature map $\mathbf{F}_c^t$. Given the predicted parameters $(q_x^t, q_y^t, q_w^t, q_h^t)$ for each region of interest at each time step, the grid generator computes the sampling location (u', v') in the original convolutional feature map $\mathbf{F}$ for every point (u, v) in the outputted $\mathbf{F}_c^t$ by

$$\begin{aligned} u' &= q_x^t + (u - \frac{W_c}{2} - \frac{1}{2})\delta_x, \quad u = 1, 2, ..., W_c \\ v' &= q_y^t + (v - \frac{H_c}{2} - \frac{1}{2})\delta_y, \quad v = 1, 2, ..., H_c \end{aligned} \tag{12}$$

where $\delta_x = \frac{q_w^t - 1}{W_c - 1}$ and $\delta_y = \frac{q_h^t - 1}{H_c - 1}$. Each sampled feature point can then be calculated using the bilinear sampler

$$\mathbf{F}_c^t(u, v) = \sum_{h=1}^{H}\sum_{w=1}^{W} \mathbf{F}(u', v')\mathrm{K}(u', w)\mathrm{K}(v', h). \tag{13}$$

where $\mathrm{K}(a, b) = \max(0, 1 - |a - b|)$ is the kernel for bilinear interpolation.

Character-Level Attention Layer

The character-level attention layer (CLA) takes the responsibility of selecting the most relevant features from the rectified character feature map produced by the character-level

encoder to generate the context vector $\mathbf{z}_c^t$. It takes the form of a traditional attention mechanism. Note that CLA is essential as it is difficult for the recurrent RoIWarp layer to precisely crop out a small feature region that contains features only from the corresponding character. Even though our recurrent localization network can perfectly predict the bounding box for each character region, the distortion exhibited in the scene text would cause the warped feature region to include also features from neighboring characters. From experiment, we find that features from neighboring characters and cluttered background would mislead the update of the parameters during the training procedure if we do not employ CLA, and this would prevent us from training our Char-Net in an end-to-end manner.

Experiment

Testing Datasets

We evaluate our Char-Net with the following public benchmarks:

- **ICDAR-2003 (IC-03)** (Lucas et al. 2005) contains 860 cropped text images for testing. Following the protocol proposed by Wang et al. (Wang, Babenko, and Belongie 2011), we recognize the images containing only alphanumeric words (0-9 and A-Z) with at least three characters.
- **ICDAR-2013 (IC-13)** (Karatzas et al. 2013) is derived from IC-03. Following (Shi et al. 2016), 857 cropped word test images without any pre-defined lexicon are filtered out using the protocol in IC-03.
- **Street View Text (SVT)** (Wang, Babenko, and Belongie 2011) contains 647 test word images collected from Google Street View.
- **IIIT5K** (Mishra, Alahari, and Jawahar 2012) contains 3,000 cropped text images for testing. These images are all collected from the Internet.
- **Street View Text Perspective (SVT-P)** (Phan et al. 2013) contains 639 cropped test images which are specially picked from the side-view angles in Google Street View. Most of them suffer from a large perspective distortion.
- **ICDAR Incidental Scene Text (IC-IST)** (Karatzas et al. 2015) contains 2077 text images for testing. All the word images are cut out from incidental scene text images captured under arbitrary poses. Hence, IC-IST contains scene text with different kinds of severe distortion.

Implementation Details

There are five convolutional blocks (refer to the blue rectangles in Figure 2) in the word-level encoder. The detailed configurations of these five convolutional blocks are $[3, 64, 1, 1] \times 3$, $[3, 128, 1, 1] \times 2$, $[3, 256, 1, 1] \times 2$, $[3, 256, 1, 1] \times 4$ and $[3, 256, 1, 1] \times 4$ respectively, where the numbers in the brackets represent the filter size, number of channels, pad size and stride, respectively, and the number following the brackets gives the number of convolutional layers stacked. All the blocks use ReLU (Nair and Hinton 2010) as the activation function. The first and the second convolutional blocks are each followed by a 2×2 max-pooling layer with a stride of 2. To produce the final feature map of the word-level encoder, feature maps from the last three blocks are normalized using a local response normalization (Krizhevsky, Sutskever, and Hinton 2012), concatenated along the channel dimension and dimension-reduced with a 1×1 convolutional layer with 256 channels. For the character-level encoder, there are two convolutional layers with 256 channels in the local spatial transformer and three more convolutional layers with 512 channels for the CNN. The spatial transformer is employed to predict only the rotation angle of each character in the distorted scene text.

In the hierarchical attention mechanism, the recurrent RoIWarp layer uses Equation (2) for regressing the bounding box of each character region. As the decoder employs a LSTM layer with 256 hidden states and $\mathbf{F}_{ij}$ in the recurrent RoIWarp layer is a 256-dimensional feature vector, the corresponding dimensions of $\mathbf{M}$, $\mathbf{V}$ and $\mathbf{w}$ are 256×256, 256×256 and 1×256 respectively. Besides, MLP_l and MLP_g both take the form of a fully-connected layer with 4 hidden states. The width and height of $\mathbf{F}_c^t$ are both set to 5. For the character-level attention layer, it again employs Equation (2) to attend to the most relevant features of the character feature region, in which each feature point is a 512-dimensional feature vector after the character-level encoder. Consequently, the dimension of $\mathbf{V}$ in the character-level encoder becomes 256×512 while the others remain the same as those in the recurrent RoIWarp layer.

Comparison with Previous Methods

To make a fair comparison with previous methods (Jaderberg et al. 2015a; Lee and Osindero 2016; Shi, Bai, and Yao 2016; Shi et al. 2016), we first trained our Char-Net following their experimental settings. In this section, we trained our Char-Net to classify 37 classes (26 case-insensitive characters + 10 digits + eos) using the 8-million synthetic dataset (SynthR) generated by Jaderberg et al. (Jaderberg et al. 2014). The training batch size was set to 64. All the input images were gray-scale and resized to 100×32 in both training and testing. As mentioned in the recurrent localization network, a pre-training Gaussian attention mechanism was employed to successfully optimize our Char-Net. We first trained this Gaussian attention mechanism with a word-level encoder and a LSTM-based decoder using Adadelta (Zeiler 2012). We then initialized our Char-Net using the parameters of the pre-trained model and optimized it by Adam (Kingma and Ba 2014). The learning rate was set to 10^{-4}. We denote our Char-Net trained under this experimental setting as Char-Net[P].

Results The recognition accuracies of our Char-Net[P] are reported in Table 1. We mainly focus on the 'unconstrained' scene text recognition without any pre-defined lexicon. As the results in (Jaderberg et al. 2016) were constrained to a 90K dictionary, we also include results using the same dictionary to post-process the predictions of our Char-Net[P] on IC-03, IC-13 and SVT. Compared with previous meth-

Table 1: Scene text recognition accuracies (%) of Char-Net[P] on public benchmarks. Note that all the outputs in (Jaderberg et al. 2016) (marked with *) were constrained to a 90K dictionary even when recognizing without a pre-defined lexicon.

Method		IC-03	IC-13	IIIT5K	SVT	SVT-P	IC-IST
Bissacco et al. (Bissacco et al. 2013)		-	87.6	-	78.0	-	-
*Jaderberg et al. (Jaderberg et al. 2016)		*93.1	*90.8	-	*80.7	-	-
Jaderberg et al. (Jaderberg et al. 2015a)		89.6	81.8	-	71.7	-	-
Lee et al. (Lee and Osindero 2016)		88.7	90.0	78.4	80.7	-	-
CRNN (Shi, Bai, and Yao 2016)		89.4	86.7	78.2	80.8	66.8	-
RARE (Shi et al. 2016)		90.1	88.6	81.9	81.9	71.8	-
STAR-Net (Liu et al. 2016)		89.9	89.1	83.3	83.6	73.5	-
Char-Net[P]	Unconstrained	91.5	90.8	**83.6**	84.4	**73.5**	**60.0**
	90K Dict	**93.3**	**93.7**	-	**87.6**	-	-

Table 2: Details of ground truth labelling of public benchmarks. Note that text images in IIIT5K (marked with *) contain different punctuations, which are not labeled in the ground truth.

	IC-03	IC-13	IIIT5K	SVT	SVT-P	IC-IST
Case-Sensitive	Yes	Yes	Yes	No	No	Yes
Punctuations	No	No	*No	No	No	Yes

ods, our Char-Net[P] can achieve state-of-art performance on all six testing datasets. In particular, we compare our Char-Net[P] against RARE (Shi et al. 2016) and STAR-Net (Liu et al. 2016), which both focus on the rectification of the entire distorted text using global spatial transformer. We find that our Char-Net is able to outperform both methods on almost every benchmark. Note that although the performance of STAR-Net on SVT-P is similar to that of our Char-Net, our Char-Net does not require extra undistorted training data as in STAR-Net. Besides, our Char-Net is much simpler than STAR-Net which employs 26 convolutional layers with the powerful residue learning (He et al. 2016a).

Experiments for General Scene Text Recognition

In this section, we further evaluate our Char-Net on more general scene text recognition. In the comparisons with previous methods, we observe that their experimental settings have three limitations that prevent the recognizer from handling text in more general scenarios. First, the process of resizing images to 100×32 damages the aspect ratio of the characters when recognizing scene text with large distortion (refer to the first three images in Figure 4). Second, all the previous work focused on 37-class scene text recognition. In general scenarios, however, scene text often contains case-sensitive characters and different punctuations. Table 2 gives the detailed information of whether public benchmarks contain ground truth labelling for case-sensitive characters or punctuations. Third, the training dataset SynthR does not contain largely distorted scene text, which can be observed from the synthetic scripts[1] released by Jaderberg et al. (Jaderberg et al. 2014). In this section, we trained our Char-Net with the following modifications to address a more general scene text recognition:

- We padded the text image to a square and then resized it to 100×100. This ensures the aspect ratio of the characters in the image remains unchanged when handling scene text with a severe distortion, especially with a large rotation.
- We trained our Char-Net to perform 96-class recognition (26 upper-case letters + 26 lower-case letters + 10 digits + 33 punctuations + eos). To compensate the lack of scene text with different punctuations in SynthR, we incorporated another recently proposed dataset (Gupta, Vedaldi, and Zisserman 2016) for scene text detection and created a new dataset SynthM[2] for training.
- We performed data augmentation to generate scene text with different kinds of rotation and perspective distortion for training.

We denote our Char-Net trained under this new experimental setting as Char-Net[N]. We used the previous Char-Net[P] as a pre-trained model to initialize our Char-Net[P]. Adam with a 10^{-4} learning rate was then employed to optimize Char-Net[N] to convergence. In order to demonstrate the effectiveness of each component in the proposed model, we also trained five variants of our Char-Net (refer to Table 3).

Results: The results of our Char-Net[N] and its variants are shown in Table 3. The proposed Char-Net, which consists of the word-level encoder with hyper-connections, the hierarchical attention mechanism and the character-level encoder, can achieve either highly competitive or state-of-the-art performance among all its variant models, especially on the IC-IST dataset which consists of many largely distorted scene text.

Hyper-Connections: Comparing the results of our Char-Net[N] and V1 in Table 3, we find that the proposed architecture of Char-Net benefits a lot from the hyper-connections in the word-level encoder. This is because these hyper-connections alleviate the problem of using only features from the final layer of a deep CNN which are semantically strong but spatially coarse. This problem is particularly severe for small objects. Although we keep the aspect ratio of the characters by padding the text image to a square, the process of resizing the square image to 100×100 inevitably

[1] https://bitbucket.org/jaderberg/text-renderer.

[2] http://www.visionlab.cs.hku.hk/datasets/wliu/synthm

Table 3: Scene text recognition accuracies of Char-Net[N] and its variants on public benchmarks. 'Hyper-Connect', 'TAM', 'HAM' and 'CLE' stands for the hyper-connections of the word-level encoder, the traditional attention mechanism, the proposed hierarchical attention mechanism and the character-level encoder respectively.

Model	Hyper-Connect	TAM	HAM	CLE		IC-03		IC-13		SVT	IIIT5K		SVT-P	IC-IST	
				STN	CNN	sen	in-sen	sen	in-sen	in-sen	sen	in-sen	in-sen	sen	in-sen
Char-Net[N]	✓		✓	✓	✓	**89.4**	**92.0**	**88.3**	**91.1**	85.5	**88.7**	**92.0**	**78.9**	**71.6**	**74.2**
V1			✓	✓	✓	88.6	90.9	87.2	90.6	84.9	87.6	91.3	75.7	70.1	72.6
V2	✓		✓	✓		89.3	91.4	87.3	90.2	83.3	87.9	91	74.9	69.4	71.8
V3	✓		✓		✓	**89.4**	91.7	88.0	90.7	85	87.9	91.5	76.1	70.3	72.8
V4	✓		✓			89	91.7	87.9	90.6	**85.6**	88.5	91.7	75.5	68.8	70.8
V5	✓	✓				89.1	91.7	87.2	89.5	84.9	88	91.2	75.2	68.2	70.1

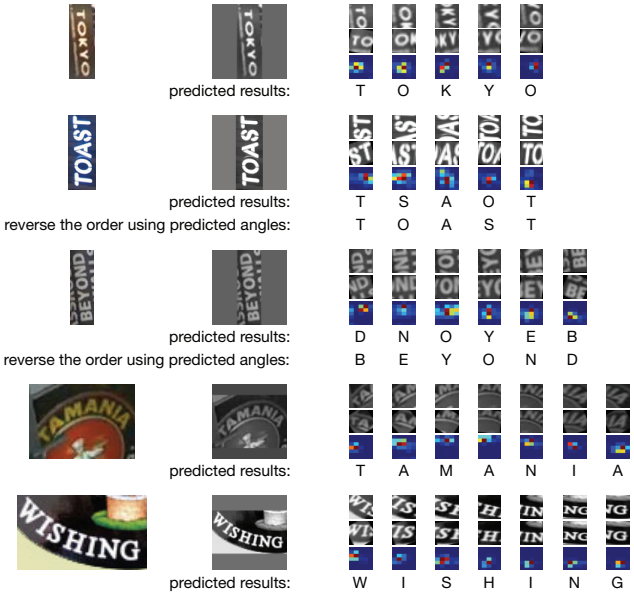


Figure 4: Some examples of largely distorted scene text being recognized by our Char-Net. The first and second columns are the original images and the square images for training, respectively. The corresponding image patches of the attended and rectified character feature regions together with the character-level attentions are shown in the third column. Note that in order to get the correct predictions of the second and third images, we reverse the order of the predicted characters according to their rotation angles.

causes the scale of the characters to vary. In the six testing datasets, the minimum and maximum lengths of scene text are 1 and 22 respectively. The size of the characters in the images ranges from 100×100 to 5×5. With the word-level encoder becomes deeper and deeper, the spatial information of each character preserved in the final feature map decreases drastically, especially for those small characters. The hyper-connections can improve the spatial resolution of characters in the final feature map by merging convolutional features from different levels. This preserves spatial information as much as possible for the following recurrent RoI-Warp layer and the STN in the character-level encoder. This allows them to better locate each feature region of interest

Figure 5: Some failure cases of Char-Net. Blue and black words denote the recognition results and ground truth labellings, respectively.

and predict its spatial distortion.

Hierarchical Attention Mechanism: From Table 3, we observe that the scene text recognized only with the proposed HAM (V4) can already achieve a slightly better performance than that with the traditional attention mechanism (V5) employed by previous work (Shi et al. 2016; Lee and Osindero 2016). However, the most important advantage of the proposed HAM is its flexible architecture, which enables our Char-Net to further employ a character-level encoder to handle the distortion of each individual character. By employing a local spatial transformer network in the character-level encoder, our Char-Net can outperform V5 on almost all the benchmarks by a large margin. Especially, the proposed Char-Net is able to handle scene text with a large distortion in the IC-IST dataset, which are illustrated in Figure 4.

Effectiveness of Each Component: We notice that the performances of our Char-Net, V2, V3 and V4 are quite similar on IC-03, IC-13, SVT and IIIT5k, which are four commonly used benchmarks,. This is mainly because most of the scene text in these datasets are tightly-bounded, horizontal and frontal. When it comes to SVT-P and IC-IST in which most of the images suffer from a large distortion, our Char-Net consistently outperforms its variants with one or more omitted components.

Weakness: Our Char-Net can achieve the state-of-the art performance for recognising largely distorted text. However, it fails when handling very blury text images. Some failure cases from IC-IST are shown in Figure 5.

Conclusion

In this paper, we present a novel Character-Aware Neural Network (Char-Net) for distorted scene text recognition. Thanks to our newly proposed hierarchical attention mechanism, our Char-Net can efficiently and effectively handle complicated forms of distortion exhibited in the scene text by attending to and rectifying individual character regions through a simple local transformer network. Experiments on six public benchmark datasets demonstrate our Char-Net can achieve state-of-the-art performance.

Acknowledgments

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research.

References

Bahdanau, D.; Cho, K.; and Bengio, Y. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Bissacco, A.; Cummins, M.; Netzer, Y.; and Neven, H. 2013. Photoocr: Reading text in uncontrolled conditions. In *Proceedings of the IEEE International Conference on Computer Vision*, 785–792.

Bookstein, F. L. 1989. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (6):567–585.

Graves, A. 2013. Generating sequences with recurrent neural networks. *arXiv preprint arXiv:1308.0850*.

Gupta, A.; Vedaldi, A.; and Zisserman, A. 2016. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2315–2324.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, P.; Huang, W.; Qiao, Y.; Loy, C. C.; and Tang, X. 2016b. Reading scene text in deep convolutional sequences. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 3501–3508.

He, P.; Huang, W.; He, T.; Zhu, Q.; Qiao, Y.; and Li, X. 2017. Single shot text detector with regional attention. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3047–3055.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. In *Workshop on Deep Learning, Advances in Neural Information Processing Systems*.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2015a. Deep structured output learning for unconstrained text recognition. In *International Conference on Learning Representations*.

Jaderberg, M.; Simonyan, K.; Zisserman, A.; et al. 2015b. Spatial transformer networks. In *Advances in Neural Information Processing Systems*, 2017–2025.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2016. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision* 116(1):1–20.

Jaderberg, M.; Vedaldi, A.; and Zisserman, A. 2014. Deep features for text spotting. In *European Conference on Computer Vision*. 512–528.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; Gomez i Bigorda, L.; Robles Mestre, S.; Mas, J.; Fernandez Mota, D.; Almazan Almazan, J.; and de las Heras, L.-P. 2013. Icdar 2013 robust reading competition. In *Proceedings of the IEEE International Conference on Document Analysis and Recognition*, 1484–1493.

Karatzas, D.; Gomez-Bigorda, L.; Nicolaou, A.; Ghosh, S.; Bagdanov, A.; Iwamura, M.; Matas, J.; Neumann, L.; Chandrasekhar, V. R.; Lu, S.; et al. 2015. Icdar 2015 competition on robust reading. In *Proceedings of the IEEE International Conference on Document Analysis and Recognition*, 1156–1160.

Kingma, D., and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Kong, T.; Yao, A.; Chen, Y.; and Sun, F. 2016. Hypernet: Towards accurate region proposal generation and joint object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 845–853.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 1097–1105.

Lee, C.-Y., and Osindero, S. 2016. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2231–2239.

Liu, W.; Chen, C.; Wong, K. K.; Su, Z.; and Han, J. 2016. Star-net: A spatial attention residue network for scene text recognition. In *Proceedings of the British Machine Vision Conference*.

Lucas, S. M.; Panaretos, A.; Sosa, L.; Tang, A.; Wong, S.; Young, R.; Ashida, K.; Nagai, H.; Okamoto, M.; Yamamoto, H.; et al. 2005. Icdar 2003 robust reading competitions: entries, results, and future directions. *International Journal of Document Analysis and Recognition* 7(2-3):105–122.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012. Scene text recognition using higher order language priors. In *Proceedings of the British Machine Vision Conference*.

Nair, V., and Hinton, G. E. 2010. Rectified linear units improve restricted boltzmann machines. In *International Conference on Machine Learning*, 807–814.

Phan, T.; Shivakumara, P.; Tian, S.; and Tan, C. 2013. Recognizing text with perspective distortion in natural scenes. In *Proceedings of the IEEE International Conference on Computer Vision*, 569–576.

Shi, B.; Bai, X.; and Yao, C. 2016. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Shi, B.; Wang, X.; Lyu, P.; Yao, C.; and Bai, X. 2016. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 4168–4176.

Wang, K.; Babenko, B.; and Belongie, S. 2011. End-to-end scene text recognition. In *Proceedings of the IEEE International Conference on Computer Vision*, 1457–1464.

Wang, T.; Wu, D. J.; Coates, A.; and Ng, A. Y. 2012. End-to-end text recognition with convolutional neural networks. In *Proceedings of the IEEE International Conference on Pattern Recognition*, 3304–3308.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.