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

SqueezedText: A Real-Time Scene Text Recognition by Binary Convolutional Encoder-Decoder Network

Zichuan Liu,[1] **Yixing Li,**[2] **Fengbo Ren,**[2] **Wang Ling Goh,**[1] **Hao Yu**[3]
Nanyang Technological University, Singapore[1], Arizona State Unviversity, the USA[2]
and Southern University of Science and Technology, China[3]
{zliu016@e., ewlgoh@}ntu.edu.sg[1], {yixingli,renfengbo}@asu.edu[2] and yuh3@sustc.edu.cn[3]

Abstract

A new approach for real-time scene text recognition is proposed in this paper. A novel binary convolutional encoder-decoder network (B-CEDNet) together with a bidirectional recurrent neural network (Bi-RNN). The B-CEDNet is engaged as a visual front-end to provide elaborated character detection, and a back-end Bi-RNN performs character-level sequential correction and classification based on learned contextual knowledge. The front-end B-CEDNet can process multiple regions containing characters using a one-off forward operation, and is trained under binary constraints with significant compression. Hence it leads to both remarkable inference run-time speedup as well as memory usage reduction. With the elaborated character detection, the back-end Bi-RNN merely processes a low dimension feature sequence with category and spatial information of extracted characters for sequence correction and classification. By training with over 1,000,000 synthetic scene text images, the B-CEDNet achieves a recall rate of 0.86, precision of 0.88 and F-score of 0.87 on ICDAR-03 and ICDAR-13. With the correction and classification by Bi-RNN, the proposed real-time scene text recognition achieves state-of-the-art accuracy while only consumes less than 1-ms inference run-time. The flow processing flow is realized on GPU with a small network size of 1.01 MB for B-CEDNet and 3.23 MB for Bi-RNN, which is much faster and smaller than the existing solutions.

Introduction

The success of convolutional neural network (CNN) has resulted in a potential general machine learning engine for various computer vision applications (LeCun et al. 1998; Krizhevsky, Sutskever, and Hinton 2012), such as text detection, recognition and interpretation from images. Applications, such as Advanced Driver Assistance System (ADAS) for road signs with text, however, require a real-time processing capability that is beyond the existing approaches (Jaderberg et al. 2014; Jaderberg, Vedaldi, and Zisserman 2014) in terms of processing functionality, efficiency and latency.

For a real-time scene text recognition application, one needs a method with memory efficiency and fast processing time. In this paper, we reveal that binary features (Courbariaux and Bengio 2016) can effectively and efficiently represent the scene text image. Combining with deconvolution technique, we introduce a binary convolutional encoder-decoder network (B-CEDNet) for real-time one-shot character detection and recognition. The scene text recognition is further enhanced with a back-end character-level sequential correction and classification, based on a bidirectional recurrent neural network (Bi-RNN). Instead of detecting characters sequentially (Bissacco et al. 2013; Wang et al. 2012; Shi, Bai, and Yao 2015), our proposed method, called SqueezedText, can detect multiple characters simultaneously and extracts a length-variable character sequence with corresponding spatial information. This sequence will be subsequently fed into a Bi-RNN, which then learns the detection error characteristics from the previous stage to provides character-level correction and classification based on the spatial and contextual cues.

By training with over 1,000,000 synthetic scene text images, the proposed SqueezedText can achieve recall rate of 0.86, precision of 0.88 and F-score of 0.87 on ICDAR-03 (Lucas et al. 2003) dataset. More importantly, it achieves state-of-the-art accuracy of 93.8%, 92.7%, 94.3% 96.1% and 83.6% on ICDAR-03, ICDAR-13, IIIT5K, STV and Synthe90K datasets. SqueezedText is realized on GPU with a small network size of 1.01 MB for B-CEDNet and 3.23 MB for Bi-RNN; and consumes less than 1 ms inference run-time on average. It is up to 4× faster and 6× smaller than state-of-the-art work.

The contributions of this paper are summarized as follows:

- We propose a novel binary convolutional encoder-decoder neural network model, which acts as a visual front-end module to provide unconstrained scene text detection and recognition. It effectively detects individual character with high recall rate, realizing an extremely fast run-time speed and small memory consumption.
- We reveal that the text features can be learned and encoded in binary format without loss of discriminative information. This information can be further decoded and recovered to perform multi-character detection and recognition in parallel.
- We further design a back-end bidirectional RNN (Bi-RNN) to provide fast and robust scene text recognition with correction and classification.

Related work

It is challenging to recognize text from the natural images since the text image will suffer from noise, blur, distortion, occlusion and variation. Generally, there are two categories of methods that can be applied, character-level method and word-level method. The character-level method (Mishra, Alahari, and Jawahar 2012b; 2012a; Sawaki, Murase, and Hagita 2000; Zhou and Lopresti 1997; Zhou, Lopresti, and Lei 1997; Novikova et al. 2012) performs an individual character detection and recognition. It relies on a multi-scale sliding window strategy to localize and recognize characters. A robust word recognition relies on a strong character detector which will be run on different parts of the image for many times. The word-level methods such as (Jaderberg et al. 2014; Rodriguez-Serrano, Perronnin, and Meylan 2013) treat scene text recognition as an image classification problem, and assign a class label to each English word. (Rodriguez-Serrano, Perronnin, and Meylan 2013) proposed to embed word labels and word images into a common Euclidean space. The text recognition is equivalent to finding the closest word label in this space when given a word image. This space is learned by Structed SVM (Hare et al. 2016) by enforcing matching label-image pairs to be closer than non-matching pairs. (Jaderberg et al. 2014) presented a deep neural network model which is trained on data produced by a synthetic text generation engine. This network encodes 90,000 character sequence and achieves the state-of-the-art recognition performance.

We propose a binary convolutional encoder-decoder neural network model to provide unconstrained scene text detection and recognition, which effectively detects individual character with high recall rate, realizing an extremely fast run-time speed and small memory consumption. With the elaborated character detection by B-CEBDNet, the back-end Bi-RNN merely processes a low dimension feature sequence for sequence classification.

Approach

SqueezedText overview

The overall recognition pipeline is illustrated in Fig 1. Given a scene text image with size of $W_I \times H_I$, the proposed B-CEDNet produces C salience maps with size of $W_I \times H_I$ which can be combined into a 3D array $S \in \mathbf{R}^{W_I \times H_I \times C}$. Note that C denotes the number of characters plus a background class. A character sequence with spatial information $U = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_T]$ is extracted from S by firstly thresholding S with confidence factor F_{conf} and then performing binary morphologic filtering with kernel size of M_{mf}. Here, $\mathbf{u}_t \in \mathbf{R}^{D^u}$ denotes label vector indicating the category, position, width and height of detected character. The extracted sequence U will be fed into a Bi-RNN network (Ma and Hovy 2016) that corrects the detecting error in U by performing a contextual correction and classification and then outputs the recognition results.

B-CEDNet for character detection

Binary feature encoding and decoding for real-time character detection There exists large amounts of redundancy in real-valued feature encoding, which prohibits the deployment of traditional CNN on embedded devices for real-time scene text recognition. It has been shown that both weights and the activations can be constrained in binary format during training without a significant accuracy loss (Courbariaux and Bengio 2016). The binary weights and activations result in a large amount of memory reduction. More importantly, the convolution can be realized by bitwise XNOR followed by bit-count operation (Courbariaux and Bengio 2016), which leads to a much higher level of computing parallelism when compared with conventional CNNs.

In the conventional CNN, multiple convolutional blocks are stacked together, forming a convolutional encoder that generates discriminative features with lower dimension (LeCun et al. 1998). Then, a classification is performed by a fully-connected layer based on the output of the convolutional encoder. When the traditional CNN is applied for scene text recognition, generally an input image is divided (from left to right) into patches with equal size and stride, and the classification is performed on each patch by CNN (Shi, Bai, and Yao 2015; Jaderberg, Vedaldi, and Zisserman 2014). This approach can cause duplicated detection if one character lies in multiple patches, or meaningless detection if multiple characters lie in just one patch, requiring additional complex post-processing. The reason behind is that the traditional CNN is designed to recognize one object for one image. Although the features provided by the convolutional layers are highly correlated to a corresponding region

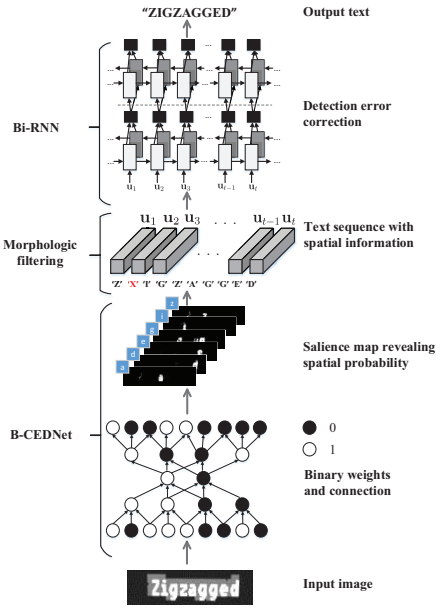

Figure 1: SqueezedText overview: The B-CEDNet produces salience maps for each character which reveal their category and spatial information. Thresholding and morphologic filtering find the position and size of character region which will be organized to a vector sequence for contextual correction and text classification provided by Bi-RNN.

in an image, this spatial information is ignored by the fully-connected layer which naively treats the multi-dimensional features as a one-dimension vector to perform the classification with loss of accuracy.

In our method, we firstly extract the features using binary convolutional encoder, and then use deconvolution technique (Kim and Hwang 2016; Badrinarayanan, Kendall, and Cipolla 2015) to reconstruct a rich set of discriminative features from the output of convolutional encoder. Note that all the features are in binary format. Combined with binary decoding operation, less discriminative information is suppressed and the highly discriminative information is boosted. More importantly, the binary weights, activation and convolution operation lead to a massive computing parallelism with a great reduction of memory usage.

B-CEDNet architecture Fig. 2 illustrates the architecture of the proposed Binary Convolutional Encoder-decoder Network (B-CEDNet). The B-CEDNet consists of three main modules, adapter module, binary encoder module and binary decoder module.

Adapter. The adapter module (block-0) contains a full-precision convolutional layer, followed by a batch-normalization (BN) layer and binarization (Binrz) layer. It transforms the input data into binary format before feeding the data into the binary encoder module.

Binary convolutional encoder. The binary encoder module consists of 4 blocks (block-1 to -4), each of which has one binary convolutional (BinConv) layer, one batch-normalization (BN) layer, one pooling layer and one binarization (Binrz) layer. The BinConv layer takes binary feature maps $a^b_{k-1} \in \{-1,+1\}^{W_{k-1}\times H_{k-1}\times D_{k-1}}$ as input and performs binary convolution operation which is illustrated as follows:

$$s_k(x,y,z) = \sum_{i=1}^{w_k}\sum_{j=1}^{h_k}\sum_{l=1}^{D_{k-1}} XNOR(w^b_k(i,j,l,z), a^b_{k-1}(i+x-1, j+y-1, l)), \tag{1}$$

where $XNOR(\cdot)$ is defined as bitwise XNOR operation, $w^b_k \in \{-1,+1\}^{w_k\times h_k\times D_{k-1}\times D_k}$ are the binary weights in k-th block and $s_k \in \mathbb{R}^{W_k\times H_k\times D_k}$ is the output of the spatial convolution. Note that the BinConv operation can be implemented on GPU by concatenating 32 binary variables into 32-bit registers and a 32× speedup can be obtained on bitwise operations (XNOR) (Courbariaux and Bengio 2016). Then s_k will be normalized by the BN layer before pooling and binarization. The output of k-th BN layer $a_k \in \mathbb{R}^{W_k\times H_k\times D_k}$ is represented by

$$a_k(x,y,z) = \frac{s_k(x,y,z)-\mu(x,y,z)}{\sqrt{\sigma^2(x,y,z)+\epsilon}}\gamma(x,y,z) + \beta(x,y,z), \tag{2}$$

where μ and σ^2 are the expectation and variance over the mini-batch, while γ and β are learnable parameters (Ioffe and Szegedy 2015) and ϵ is a small value avoiding the infinite output. The output of the BN layer is subsequently down-sampled by the pooling layer. Here we apply 2×2 max-pooling to filter out the strongest activation which will be binarized by the Binrz layer. The binarized activations a^b_k of k-th block can be represented as

$$a^b_k(x,y,z) = \begin{cases} -1, & a_k(x,y,z) \le 0 \\ +1, & a_k(x,y,z) > 0 \end{cases}. \tag{3}$$

Binary convolutional decoder. What is more for the decoder module, it transforms the compact high-level representation $a^b_5 \in \{-1,+1\}^{W_5\times H_5\times D_5}$ generated by the encoder into a set of salience maps $S \in \mathbb{R}^{W_I\times H_I\times C}$ which indicate the spatial probability distribution over category space including 26 characters and a background class. The decoder module is composed of 6 convolutional blocks (block-5 to -10). Block-5 to -8 are formed by one unpooling layer, one BinConv layer, one BN layer and one Binrz layer. Note that there exists a symmetric structure along block-1 to -8. Thus the unpooling layers (Badrinarayanan, Kendall, and Cipolla 2015) within block-5 to -8 simply assign the input pixels back to their original position according to the index generated by the corresponding max-pooling layer and pad the remains with -1. The up-sampled feature maps then go through the binary convolution, normalization and binarization. The output of block-8 $a^b_8 \in \{-1,+1\}^{W_8\times H_8\times D_8}$ will be processed by block-9 and -10 to generate spatial salience maps. Block-9 and -10 form a 2-D spatial classifier with 1×1 convolution window and softmax output. It produces the posterior probability distribution S over the category space for each pixel in the original image.

Text sequence extraction

To hunt the candidates of character regions, we first perform thresholding to S with thresholding factor F_{conf}. Then a binary image $I_{dom} \in \{0,1\}^{W_I\times H_I}$ indicating the dominated area of texts is generated by averaging S along the 3rd dimension, non-zero thresholding and binary morphologic filtering with a kernel size M_{mf}. Afterwards, we apply I_{dom} as a mask to each slide of S that removes most of the isolated false detection with low confidence value, which can be illustrated by the following equation:

$$S'(x,y,z) = S(x,y,z)\cdot I_{dom}(x,y). \tag{4}$$

where x, y and z are the index of the 3D array S. To facilitate the evaluation of the position and the size of each character, we conduct another binary morphologic filtering to each $S'(:,:,z)$, $z = 1,2,\cdots C$, and extract the character regions with their position $\mathbf{p}_c = (x_c, y_c)^T$, size $\mathbf{s}_c = (w_c, h_c)^T$ and categories $q_c \in \{1,2,\cdots,C\}$ by finding the connected component. Finally, we construct a vector sequence $U = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_T]$ with $\mathbf{u}_t = (\mathbf{p}^T_c, \mathbf{s}^T_c, q_c)^T$. Note that the elements in U are ordered from left to right and will be fed to Bi-RNN one by one for contextual correction and classification.

Bi-RNN for contextual text correction and classification

In text sequence extraction, the false detections which mostly occur near the edge of an image have been removed. However, there still exist some false detections with high

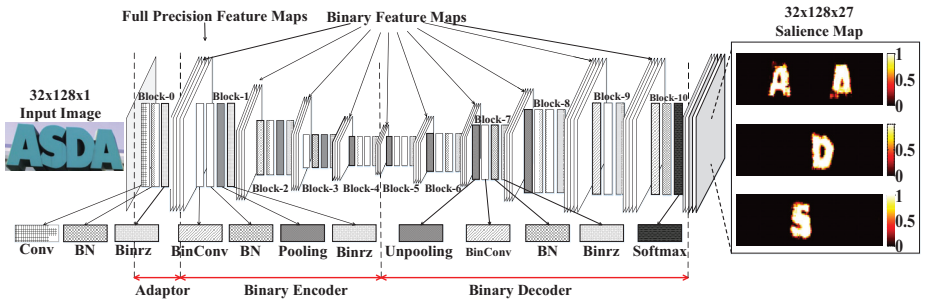

Figure 2: The architecture of Binary Convolutional Encoder-decoder Network (B-CEDNet).

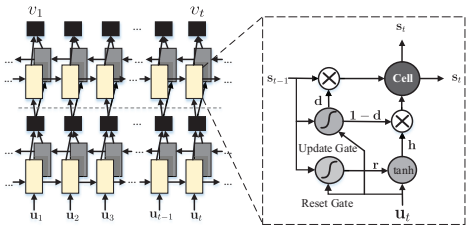

Figure 3: Bi-RNN architecture for contextual text correction and classification: The "update" gate decides which element in the sequence to be accepted to update the state $\mathbf{c}_t$ based on the category and spatial information in $\mathbf{u}_t$; and "reset" gate determines where is the end of a word.

confidence value within the dominated area of text. These false detections are due to the similar local features between characters. For example, the upper part of 'Y' is similar to the upper part of 'X', which could mislead the B-CEDNet to generate the false activation of 'X' together with the true activation 'Y'. This insertion error is highly correlated to the ground-true character and is hard to be removed by the thresholding and morphologic filtering. Another common detection error is that some true activations with small area could be removed by the morphologic filtering. It causes a deletion error in text sequence. To correct the insertion and deletion error, we apply a bidirectional RNN model (Ng et al. 2014) for character-level correction and classification.

The architecture of the RNN model for character-level sequence correction and classification is shown in Fig. 3. The model consists of an encoder and a decoder (Chan et al. 2015). The N-layer encoder maps the input sequence $U = [\mathbf{u}_1, \mathbf{u}_2, \cdots, \mathbf{u}_T]$ to a high-level representation $\mathbf{c}^N$ with bidirectional RNN architecture (Chan et al. 2015). Given an input sequence $\mathbf{u}_t$ containing character label q_c and the corresponding spatial information $\mathbf{p}_c$ and $\mathbf{s}_c$, the forward, backward, and combined activations of the jth hidden layer of the encoder are computed as:

$$\begin{aligned}\mathbf{f}_t^j &= GRU(\mathbf{f}_{t-1}^j, \mathbf{c}_t^{j-1}),\\ \mathbf{b}_t^j &= GRU(\mathbf{b}_{t+1}^j, \mathbf{c}_t^{j-1}),\\ \mathbf{h}_t^j &= \mathbf{f}_t^j + \mathbf{b}_t^j\end{aligned} \tag{5}$$

where GRU denotes the gated recurrent unit function that can be represented by

$$\begin{aligned}\mathbf{d} &= \sigma(\mathbf{c}_t^{j-1} \cdot \mathbf{U}^d + \mathbf{s}_{t-1}^j \cdot \mathbf{W}^d),\\ \mathbf{r} &= \sigma(\mathbf{c}_t^{j-1} \cdot \mathbf{U}^r + \mathbf{s}_{t-1}^j \cdot \mathbf{W}^r),\\ \mathbf{g} &= \tanh(\mathbf{c}_t^{j-1} \cdot \mathbf{U}^h + (\mathbf{s}_{t-1}^j \circ \mathbf{r}) \cdot \mathbf{W}^h)\\ \mathbf{s}_t &= (\mathbf{1} - \mathbf{d}) \circ \mathbf{g} + \mathbf{d} \circ \mathbf{s}_{t-1}\end{aligned} \tag{6}$$

In Eq. 6, $\mathbf{d}$ and $\mathbf{r}$ are "update" gate and "reset" gate, which determine how to combine the previous memory and how much of the previous memory to keep around. The input of the first layer $\mathbf{c}_t^0 = \mathbf{u}_t$ and $\mathbf{c}_t^j, j > 0$ is represented as:

$$\mathbf{c}_t = \tanh(\mathbf{W}_{pyr}^j \cdot [\mathbf{h}_{2t}^{j-1}, \mathbf{h}_{2t+1}^{j-1}]^T + \mathbf{b}_{pyr}^j) \tag{7}$$

where $\mathbf{b}_{pyr}^j$ is the bias and $\mathbf{W}_{pyr}^j$ is the output matrix of jth hidden layer. The gating units $\mathbf{d}$ and $\mathbf{r}$ allow the network to selectively reject the false detections and decide where is the end of a sequence based on the current state $\mathbf{s}_t$ and the input sequence U. The bidirectional structure considers not only the past context but also the future context. This contextual information is useful and complementary, and can improve the representation capacity and accuracy of the model. Next, the RNN decoder is an M-layer recurrent neural network that generates the output sequence character by character. It produces an output sequence based on the encoded representation $\mathbf{c}^N$ using an attention mechanism (Bahdanau, Cho, and Bengio 2014). At the jth decoder layer, the hidden activations are computed as

$$\mathbf{e}_t^j = GRU(\mathbf{e}_{t-1}^j, \mathbf{e}_t^{j-1}), \tag{8}$$

where $\mathbf{e}_t^j$ is jth hidden layer activation at time step t. Thereafter, the final hidden layer activation $\mathbf{e}_t^M$ is used as part of the contentbased attention mechanism (Bahdanau, Cho, and Bengio 2014):

$$\begin{aligned}\beta_{tk} &= \phi_1(\mathbf{e}_t^M)^T \phi_2(\mathbf{c}_k^N)\\ \alpha_{tk} &= \frac{\beta_{tk}}{\sum_j \beta_{tj}}\\ a_t &= \sum_j \alpha_{tj} c_j\end{aligned} \tag{9}$$

where ϕ_1 and ϕ_2 denote the feedforward affine transforms followed by a $\tanh$ nonlinearity. The weighted sum of the

Figure 4: Examples in the synthetic dataset for training of B-CEDNet. There are 1 million training images with pixel-wise labels.

encoded hidden states a_t is then concatenated with d_t^M, and passed through another affine transform followed by a ReLU before the final softmax output layer. The softmax output is a sequence $V = [\mathbf{v}_1, \mathbf{v}_2, \cdots, \mathbf{v}_K]$ where $\mathbf{v}_t \in \mathbb{R}^{C'}$ indicates the probability distribution over the character space at tth time step.

Training

B-CEDNet training. The B-CEDNet can be trained and optimized under binary constraints proposed in (Courbariaux and Bengio 2016), which can significantly reduce memory usage and also improve level of parallelism. In this paper, we apply cross-entropy error as the loss function by removing the Binrz layer in block-10. For our application, the prediction error J is represented as follows:

$$J(w) = -\frac{1}{N_s \cdot W_{10} \cdot H_{10}} \sum_{i=1}^{N_s} \sum_{m=1}^{W_{10}} \sum_{n=1}^{H_{10}} \sum_{c=1}^{C} [1\{Y^{(i)}(x,y) = c\} \ln \frac{e^{a_{10}(m,n,c)}}{\sum_{l=1}^{C} e^{a_{10}(m,n,l)}}], \tag{10}$$

where N_s is the number of training samples in a mini-batch, C is the number of classes (characters and background), w is the filter weights, $Y^{(i)} \in \{1, ..., C\}^{H_{10} \times W_{10}}$ is the 2-D label of i-th training image, and $a_{10} \in \mathbb{R}^{H_{10} \times W_{10} \times C}$ is the output of the BN layer in block-10. To achieve generality of trained model, it usually needs a large amount of labeled data for training. However, the existing datasets are limited to word-level annotation (Veit et al. 2016) or cannot provide enough pixel-wise labeled data (Karatzas et al. 2013). Therefore, we create a text rendering engine that generates texts with different fonts, graylevels and projective distortions. The labeled image has the same size with the corresponding text image and provides a pixel-wise labeling over the category space. This dataset contains over 1,000,000 synthesized text images. Some examples are shown in Fig. 4.

Bidirectional RNN training. To train the RNN model for character-level correction and classification, we also use the cross-entropy loss per time step summed over the output sequence V:

$$L(U, V) = -\sum_{t=1}^{K} \sum_{c=1}^{C'} 1\{c == \theta_t\} \ln \frac{v_t(c)}{\sum_{i=1}^{C'} v_t(i)} \tag{11}$$

where θ_t is the index of ground true character. Note that we need a large dataset that captures the stochastic characteristics of error in sequence extraction phase. Thus, we build another dataset with training sequence and corresponding labeled sequence. The training sequence is output of sequence extraction (U) and the label sequence is the ground-true word in synthetic dataset.

Experiments

Datasets

To evaluate the effectiveness of the proposed method, we conducted experiments on standard benchmarks for the scene text recognition. Since SqueezedText contains two neural networks, we conducted two-stage training for the whole flow. The B-CEDNet is trained on synthetic scene text dataset with 1 million training images. The Bi-RNN model is trained on a dataset constructed from the character sequence output by B-CEDNet and sequence extraction operation.

Four popular benchmarks for scene text recognition are used for performance evaluation, ICDAR-2003 (IC03), ICDAR-2013 (IC13), IIIT 5k-word (IIIT5k) and Synth90k. IC03(Lucas et al. 2003) contains 251 scene images with labeled text bounding boxes. In the experiment, we ignore images that contain either non-alphabetic characters or have less than three characters, and obtain 860 cropped text images. IC13 (Karatzas et al. 2013) inherits most of its data from IC03 and have 1015 ground truths cropped word images. IIIT5k (Mishra, Alahari, and Jawahar 2012a) contains 3,000 cropped word test images collected from the Internet. SVT (Wang, Babenko, and Belongie 2011) dataset consists of 249 street view images collected from Google Street View, from which 647 word images are cropped. Each word image corresponds to a 50 lexicon. Synth90k (Jaderberg et al. 2014) is a synthetic scene text dataset containing 8 million images with ground-true labels and we randomly select 5,000 images for performance evaluation.

Implementation details

Both the B-CEDNet model and the Bi-RNN model are built based on Tensorflow 0.9v (Abadi et al. 2016). For the B-CEDNet, we implement the C-level binary convolution, binarization, un-pooling operation and morphologic filtering with GPU support based on cuBlas library. The network architecture for B-CEDNet and Bi-RNN is built with Python interface. The experiments are carried out on Dell Precision T7500 server with Intel Xeon 5600 processor, 64GB memory and NVIDIA TITAN X GPU. The training images for B-CEDNet are in the size of 128×32. The testing images are resized to the same scale. The training data for the Bi-RNN is generated using the approach mentioned in Sec. with varying confidence thresholding and size of filtering kernel. Both networks are trained using Adam optimizer with learning rate of 0.0005, default decay rates $\beta_1 = 0.9$ and $\beta_2 = 0.999$, and a batch size of 20. The B-CEDNet is trained for up to 50 epochs and the bidirectional RNN is trained for 40 epochs before the convergence is observed.

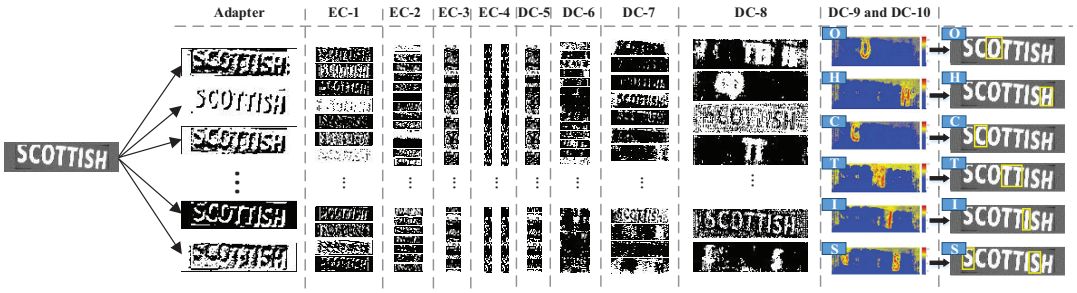

Figure 5: Visualization of binary activation of each convolutional block as well as the generated salience maps and bounding boxes.

Table 1: Accuracy comparison of existing scene text recognition approaches.

	IIIT5k			IC03				IC13	SVT
	50	1k	None	50	Full	50k	None	None	50
(Rodriguez-Serrano, Gordo, and Perronnin 2015)	76.1	57.4	-	-	-	-	-	-	-
(Jaderberg, Vedaldi, and Zisserman 2014)	-	-	-	96.2	91.5	-	-	-	86.1
(Su and Lu 2014)	-	-	-	92.0	82.0	-	-	-	-
(Gordo 2015)	93.3	86.6	-	-	-	-	-	-	-
(Jaderberg et al. 2016)	97.1	92.7	-	98.7	98.6	93.3	93.1	90.8	95.4
(Jaderberg et al. 2014)	95.5	89.6	-	97.8	97.0	93.4	89.6	81.8	-
(Shi, Bai, and Yao 2015)	97.6	94.4	78.2	98.7	97.6	**95.5**	89.4	86.7	**96.4**
(Liu and Chen 2016)	**97.7**	**94.5**	83.3	96.9	95.3	-	89.9	89.1	95.5
(Lee and Osindero 2016)	96.8	94.4	78.4	97.9	97.0	-	89.6	90.0	96.3
(He and Huang 2016)	94.0	91.5	-	97.0	93.8	-	-	-	93.5
OURS (binary)	96.9	94.3	86.6	98.4	97.9	93.8	**93.1**	92.7	96.1
OURS (full-precision)	97.0	94.1	**87.0**	**98.8**	**97.9**	93.8	**93.1**	**92.9**	95.2

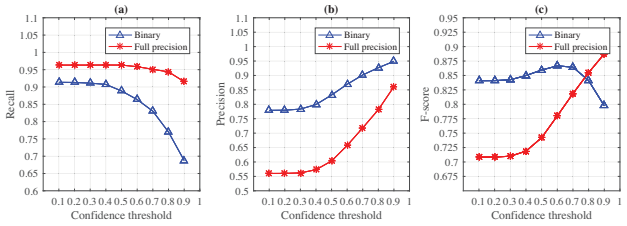

Figure 6: The trade-off between confidence threshold and character retrieval performance.

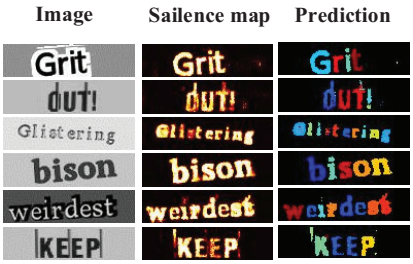

Figure 7: Test images and corresponding salience maps and predictions. In salience map, high confidence text region are rendered with red and white colors. The pxiel-wise predictions are labeled with different colors.

Comparative evaluation

Binary features and character detection Fig. 5 visualizes the binary activation of each convolutional block. The feature maps from left to right correspond to the output binary activation of Binrz layers. The left-most is the input image which is converted by the adapter block into binary images in various styles. These binary features are further encoded into high-level representations by binary convolution pooling and binarization. In the binary decoder network, the activations from the background are suppressed through propagation while the activation closely related to the target characters are retained (see DC-8 to -10). Fig. 7 shows the salience map and pixel-wise prediction produced by the B-CEDNet. The B-CEDNet can provide pixel-wise classification with prediction error lower than 10%, which indicates that the B-CEDNet can effectively capture the class-specific shape information of the character.

Character extraction In this experiment, we compare the character retrieval performance of the B-CEDNet and its full-precision version (CEDNet) on IC03 dataset. We use the extracted spatial information of characters to generate bounding box which will be compared with the ground truth. A detection is considered as successful if the predicted bounding box overlaps with ground-true bounding box. As shown in Fig. 6, the B-CEDNet maintains high recall with small confidence threshold F_{conf} and experiences a rapid drop when F_{conf} goes higher than 0.6, Fig. 6 (a). Accordingly, the precision increases with F_{conf} but the B-CEDNet shows much higher precision than the full-precision one

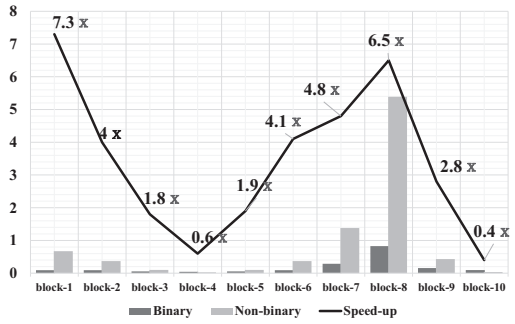

Figure 8: Run-time comparison between B-CEDNet and it full-precision version (CEDNet).

Table 2: Storage and speed comparison between B-CEDNet and existing methods.

	Network Size (MB)	Inference Time (ms)
(Jaderberg et al. 2016)	1960 MB	1000
(Jaderberg et al. 2014)	1216 MB	-
(Shi, Bai, and Yao 2016)	25.2 MB	4
Ours	4.30 MB	< 1

since the CEDNet generates a large amount of false alarms when low F_{conf} is applied, Fig. 6 (b). Different from the CEDNet that has monotone increasing F-score, there exists a trade-off between recall and precision for B-CEDNet to ensure the best retrieval performance.

Sequential classification Table. 1 shows the recognition accuracy on the aforementioned four public datasets achieved by our method (including binary version and full-precision version) and also the related works. In the lexicon case, our method achieves the state-of-the-art performance and performs best in IC03 dataset with 98.4% accuracy. In the non-lexicon scenario, our method outperforms the existing methods with a large margin. The non-lexicon recognition accuracy on IIIT5k, IC03 and IC13 is 2-8% higher than the methods in (Shi, Bai, and Yao 2015; Jaderberg, Vedaldi, and Zisserman 2014; Jaderberg et al. 2014; He and Huang 2016; Lee and Osindero 2016; Liu and Chen 2016). The accuracy gain in non-lexicon case comes from explicit spatial information in the feature sequence input to Bi-RNN. It helps Bi-RNN recognize the false character detection based on learned error characteristics of previous stages and potential language model. On the other hand, the binary version still have comparable accuracy with the full-precision version, which shows that the text features can be learned and encoded in binary format without loss of discriminative information.

Speed and memory usage Fig. 8 compares the inference time for B-CEDNet running on baseline kernel and XNOR kernel (Courbariaux and Bengio 2016). Baseline kernel is an optimized matrix multiplication kernel, while the XNOR kernel is tailored for bit-count operation in binary network. We measure the inference time with a batch of input images (size of 32) to obtain higher utilization of the GPU. Due to the bit-count operation and huge memory access reduction, the B-CEDNet achieves an average of 0.38 ms inference time and 5× speedup with XNOR kernel on TITAN X GPU compared with baseline kernel.

Table 2 reports the storage space and inference time of the SqueezedText and existing neural network based approach. In B-CEDNet, all layers have weight-sharing connections, and the fully-connected layers are replaced by the binary decoder network which has much less parameters. The binary weights and activations lead to a great amount of storage reduction. Due to the elaborated character extraction of B-CEDNet, the Bi-RNN processes feature sequence with much lower dimension when compared with the work in (Shi, Bai, and Yao 2015), leading to a fast inference. The total storage requirement of SqueezedText is only 4.24 MB which is much smaller (up to 5×) than the memory space reported in (Jaderberg, Vedaldi, and Zisserman 2014; Jaderberg et al. 2014; Shi, Bai, and Yao 2015). In terms of processing time, the proposed SqueezedText achieves 4× speedup when compared with the state-of-the-art method proposed in (Shi, Bai, and Yao 2015).

Conclusion

In this paper, we proposed a real-time scene text recognition method, called SqueezedText. Firstly, a binary convolutional encoder-decoder neural network (B-CEDNet) is developed to perform unconstrained character detection and recognition. Our study reveals that the binary representation (with deconvolution) can lead to an effective and efficient multi-character detection and recognition. Furthermore, a back-end bidirectional recurrent neural network (Bi-RNN) can be used for a character level sequential correction and classification. The proposed SqueezedText achieves the state-of-the-art performance in run-time speed, memory usage and accuracy as compared to benched results on ICDAR-03, ICDAR-13, IIIT5K, SVT and Synthe90K datasets.

Acknowledgment

The work by Arizona State University is supported by Cisco Research Center ($CG\#594589$).

References

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; et al. 2016. Tensorflow: Large-scale machine learning on heterogeneous distributed systems. *arXiv preprint arXiv:1603.04467*.

Badrinarayanan, V.; Kendall, A.; and Cipolla, R. 2015. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*.

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.

Chan, W.; Jaitly, N.; Le, Q. V.; and Vinyals, O. 2015. Listen, attend and spell. *arXiv preprint arXiv:1508.01211*.

Courbariaux, M., and Bengio, Y. 2016. Binarynet: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830*.

Gordo, A. 2015. Supervised mid-level features for word image representation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2956–2964.

Hare, S.; Golodetz, S.; Saffari, A.; Vineet, V.; Cheng, M.-M.; Hicks, S. L.; and Torr, P. H. 2016. Struck: Structured output tracking with kernels. *IEEE transactions on pattern analysis and machine intelligence* 38(10):2096–2109.

He, P., and Huang. 2016. Reading scene text in deep convolutional sequences. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Ioffe, S., and Szegedy, C. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*.

Jaderberg, M.; Simonyan, K.; Vedaldi, A.; and Zisserman, A. 2014. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*.

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. Springer.

Karatzas, D.; Shafait, F.; Uchida, S.; Iwamura, M.; i Bigorda, L. G.; Mestre, S. R.; Mas, J.; Mota, D. F.; Almazan, J. A.; and de las Heras, L. P. 2013. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, 1484–1493. IEEE.

Kim, H.-E., and Hwang, S. 2016. Deconvolutional feature stacking for weakly-supervised semantic segmentation. *arXiv preprint arXiv:1602.04984*.

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.

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

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., and Chen. 2016. Star-net: A spatial attention residue network for scene text recognition. In *British Machine Vision Conference*.

Lucas, S. M.; Panaretos, A.; Sosa, L.; Tang, A.; Wong, S.; and Young, R. 2003. Icdar 2003 robust reading competitions. In *ICDAR*, volume 2003, 682. Citeseer.

Ma, X., and Hovy, E. 2016. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012a. Scene text recognition using higher order language priors. In *BMVC 2012-23rd British Machine Vision Conference*. BMVA.

Mishra, A.; Alahari, K.; and Jawahar, C. 2012b. Top-down and bottom-up cues for scene text recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, 2687–2694. IEEE.

Ng, H. T.; Wu, S. M.; Briscoe, T.; Hadiwinoto, C.; Susanto, R. H.; and Bryant, C. 2014. The conll-2014 shared task on grammatical error correction. In *CoNLL Shared Task*, 1–14.

Novikova, T.; Barinova, O.; Kohli, P.; and Lempitsky, V. 2012. Large-lexicon attribute-consistent text recognition in natural images. In *European Conference on Computer Vision*, 752–765. Springer.

Rodriguez-Serrano, J. A.; Gordo, A.; and Perronnin, F. 2015. Label embedding: A frugal baseline for text recognition. *International Journal of Computer Vision* 113(3):193–207.

Rodriguez-Serrano, J. A.; Perronnin, F.; and Meylan, F. 2013. Label embedding for text recognition. In *Proceedings of the British Machine Vision Conference*.

Sawaki, M.; Murase, H.; and Hagita, N. 2000. Automatic acquisition of context-based images templates for degraded character recognition in scene images. In *Pattern Recognition, 2000. Proceedings. 15th International Conference on*, volume 4, 15–18. IEEE.

Shi, B.; Bai, X.; and Yao, C. 2015. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *arXiv preprint arXiv:1507.05717*.

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*.

Su, B., and Lu, S. 2014. Accurate scene text recognition based on recurrent neural network. In *Asian Conference on Computer Vision*, 35–48. Springer.

Veit, A.; Matera, T.; Neumann, L.; Matas, J.; and Belongie, S. 2016. Coco-text: Dataset and benchmark for text detection and recognition in natural images. In *arXiv preprint arXiv:1601.07140*.

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

Wang, T.; Wu, D. J.; Coates, A.; and Ng, A. Y. 2012. End-to-end text recognition with convolutional neural networks. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, 3304–3308. IEEE.

Zhou, J., and Lopresti, D. 1997. Extracting text from www images. In *Document Analysis and Recognition, 1997., Proceedings of the Fourth International Conference on*, volume 1, 248–252. IEEE.

Zhou, J.; Lopresti, D. P.; and Lei, Z. 1997. Ocr for world wide web images. In *Electronic Imaging'97*, 58–66. International Society for Optics and Photonics.