IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 17, 2018, accepted January 4, 2019, date of publication January 9, 2019, date of current version February 4, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2891767

Fully Convolutional Sequence Recognition Network for Water Meter Number Reading

FAN YANG, LIANWEN JIN, (Member, IEEE), SONGXUAN LAI, XUE GAO, AND ZHAOHAI LI
School of Electronic and Information Engineering, South China University of Technology, Guangzhou 510640, China

Corresponding author: Lianwen Jin (eelwjin@scut.edu.cn)

This work was supported in part by GD-NSF under Grant 2017A030312006, in part by the National Key Research and Development Program of China under Grant 2016YFB1001405, and in part by the NSFC under Grant 61472144, Grant 61673182, and Grant 61771199.

ABSTRACT One of the most widely used frameworks for image-based sequence recognition is the convolutional recurrent neural network, which uses a convolutional neural network (CNN) for feature extraction and a recurrent neural network (RNN) for sequence modeling. However, the RNN is computationally expensive in both training and inference, which limits its application in time-constrained systems. Some models replace the RNN with an attention mechanism for sequence modeling but still, require expensive iterative computations. In this paper, we argue that a CNN with sufficient depth can capture contextual information, eliminate the need for recurrent operations and thus be fully parallelized. We focus on the problem of water meter number reading (WNR), which is a typical sequence recognition task but has rarely been investigated. We propose a fully convolutional sequence recognition network (FCSRN) for the fast and accurate reading of water meter numbers. Furthermore, we design an augmented loss (AugLoss) function to manage the intermediate states of the digits and effectively improve performance. The experimental results demonstrate that the FCSRN has the ability to capture contextual information and eliminate the need for recurrent layers, and simultaneously requires fewer parameters and less computation. The FCSRN with AugLoss outperforms RNN-based and attention-based models. In addition, AugLoss can effectively improve the performance for RNN-based and attention-based models. Moreover, we constructed and released a dataset that contains 6000 water meter images with labels, which is available at https://github.com/HCIILAB/Water-Meter-Number-DataSet.

INDEX TERMS Image-based sequence recognition, water meter number reading, augmented loss function, water meter number dataset.

I. INTRODUCTION

Automatic water meter reading is in great demand in many practical applications, such as water charging systems and real-time monitoring of water consumption. Among the water meter number reading methods proposed in the literature, the most common are to count or measure the water flow rate using an embedded electronic device, for example, an optoelectronic device or ultrasonic wave unit. In recent years, a novel method has begun to attract the attention of researchers, which uses a camera to capture water meter images and then recognizes the numbers in the images. This method is highly flexible, easy to implement and can take full advantage of the available measuring units. For instance, a mechanical water meter is the most popular and stable measure unit in water transmission systems, and it is very unlikely that it will be replaced by electronic meters in the foreseeable future. The most important point is that, the capture of water meter images can be used to prevent cheating regarding water consumption because, currently, it is easier to cheat on electronic devices than on images. The pipeline for reading number in a water meter image consists of first detecting and cropping the number area in the image and then recognizing the water meter number in the cropped image. In this paper, we focus on the second part, that is the water meter number reading (WNR) problem, which is a typical image-based sequence recognition task.

There are several general solutions for the image-based sequence recognition task. The first category is called the segmentation-based method [2]–[9], which involves two steps: character segmentation and recognition. For example, Bissacco *et al.* [2] detected characters using bounding boxes, recognized the detected characters using a pre-trained convolutional neural network (CNN) model, and finally combined the contextual information to obtain the recognition result.

2169-3536 © 2019 IEEE. Translations and content mining are permitted for academic research only.
Personal use is also permitted, but republication/redistribution requires IEEE permission.
See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

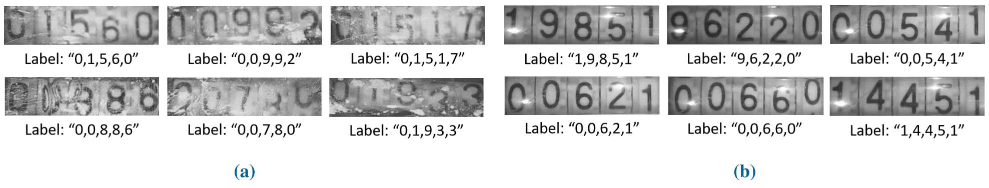

FIGURE 1. Samples in the water meter image dataset. (a) Difficult samples. (b) Easy samples.

Such methods require that each character is easily located and labeled, therefore they are not suitable for recognizing images in which characters are difficult to segment.

The second category of solution is called the segmentation-free method, which includes the holistic method, recurrent neural network (RNN)-based method and attention-based method. The holistic method [10]–[13] recognizes words in an image as a whole without recognizing each individual character. For instance, Jaderberg *et al.* [10] trained a CNN to directly classify different English words. These methods rely heavily on a predefined vocabulary and lack the ability to recognize out-of-vocabulary words. Moreover, for a water meter image with a label such as "0 0 0 0 0" (which has only five digits but each digit ranges from "0" to "19", as explained in Section II), the total number of possible combinations can be up to $20^5 = 3.2 \times 10^6$, which precludes the use of the holistic method.

The RNN-based method [1], [14]–[19] avoids the segmentation problem by iterating over the feature sequence (often derived from a CNN) and performing recognition at each time step. Most segmentation-free sequence recognition systems use an RNN with long short-term memory (LSTM) units [1], [14]–[16], which have successfully demonstrated the ability to align input and output sequences without explicit segmentation. Although the results of RNN-based systems are impressive, there exist two important drawbacks: (1) the training speed may be slow because of the iterative matrix multiplication over time steps in the recurrent layers; and (2) the optimization process may suffer from gradient vanishing/exploding problems [20], [21].

Recently, attention mechanisms have become an important part of compelling sequence modeling and transduction models in various tasks, thereby allowing the modeling of dependencies without regard to their distance in the input or output sequences [22], [23]. In most cases [24], attention mechanisms are used in conjunction with an RNN. However, generally, attention modules [25] can be used in any neural network-based models. Gehring *et al.* [26] proposed a convolutional seq2seq learning model. The representation of the input is computed by a CNN in a parallel style for the attention mechanism; the decoder state is also determined by a CNN with features that are already produced [27]. Although the results of attention-based convolutional systems are impressive, there exists one drawback: the computation of attention weights can be fully parallelized only during training, whereas at inference time the calculation of the attention weight at the current time step must depend on the output at the last time step, which cannot be easily parallelized.

WNR is similar to the text recognition task; however, its decoding process is slightly different because, in WNR, there exists some "mid-state" characters (as explained in Section II). To build a fast and accurate WNR system, we propose a fully convolutional sequence recognition network (FCSRN), which combines a fully convolutional network (FCN) [28] and CTC [29] without any intermediate recurrent connections. Moreover, by analyzing the post-processing method after decoding (in Section III-D), we propose an augmented loss (AugLoss) function to effectively improve network performance.

The experimental results demonstrate that the FCSRN has the ability to capture contextual information and eliminate the need for recurrent layers and simultaneously requires fewer parameters and less computation. The FCSRN with AugLoss outperforms RNN-based and attention-based models. Additionally, AugLoss can effectively improve performance for RNN-based and attention-based models.

The remainder of this paper is organized as follows: In Section II, we describe the constructed water meter image dataset. Then we introduce the proposed FCSRN in Section III and AugLoss in Section IV. We present the experimental results in Section V and then we conclude the paper.

II. WATER METER IMAGE DATASET

For the study of WNR, we constructed a dataset named SCUT-WMN for non-commercial use, which is available at https://github.com/HCIILAB/Water-Meter-Number-DataSet. To the best of our knowledge, this is the first public water meter image dataset.

The water meter images were captured by a camera and labeled using bounding boxes and water meter numbers. We cropped the bounding-box area to build our dataset for recognition. The dataset consists of two parts. The first part contains 5,000 difficult samples (as shown in Fig. 1a). Within the difficult samples, there is a wide range of variation caused by, for example, illumination, refraction and occlusion. The second part contains 1,000 easy samples (as shown in Fig. 1b). Both the difficult and easy samples are labeled with sequential characters such "1 2 2 5 8". Additionally, the number of each character in the dataset is shown in Table 1.

In WNR, there exist some "mid-state" characters, as shown in Fig. 2. Considering the 4th image (in row 2,

TABLE 1. Number of characters in the water meter image dataset.

character	0	1	2	3	4	5	6	7	8	9
number	6612	3234	1100	1214	517	916	754	1037	1538	920
character	10	11	12	13	14	15	16	17	18	19
number	157	210	215	340	165	184	202	221	245	219

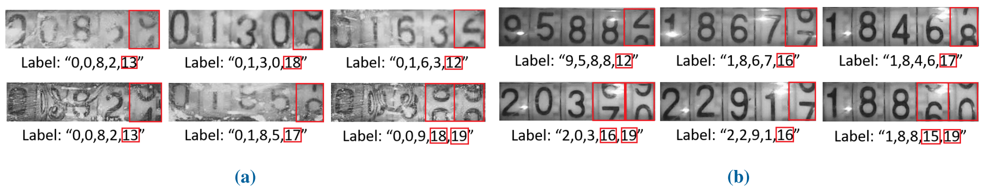

FIGURE 2. Samples in the water meter image dataset with "mid-state" characters marked in red. (a) Difficult samples. (b) Easy samples.

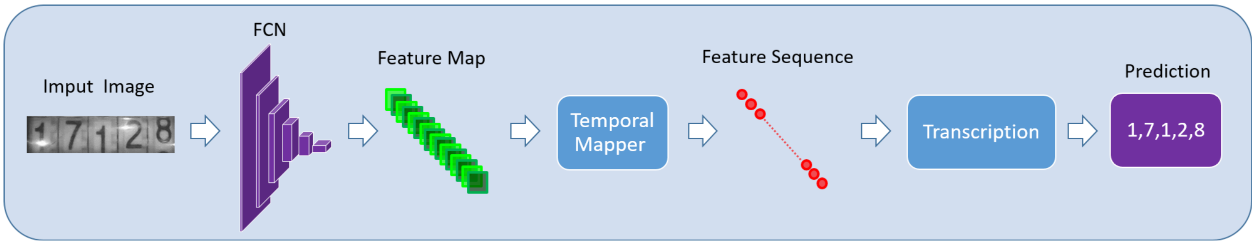

FIGURE 3. Proposed network architecture. First, the model extracts features with a fully convolutional network. Then a temporal mapper is used to transfer the two-dimensional feature map into a one-dimensional feature sequence. Finally, a transcription layer outputs the predicted label using the sequential features.

column 1) in Fig. 2b as an example, the number exceeds "20369" but does not reach "20370", so the last two characters in the number appear in the "*mid-state*". Additionally, the proper water meter number should be "20369.5". To manage "mid-state" characters, we consider them as separate classes, with labels ranging from "10" to "19", as shown in Fig. 2. Label $l \in [10, 19]$ denotes a character that exceeds "$l - 10$" (called the "lower-state") but does not reach "$l - 9$" (called the "higher-state"). As a special case, label "$l = 19$" indicates that the character is in the "mid-state" between "9" and "0" (ignoring carry). In such a setting, the label for the 4th image in Fig. 2b is "2 0 3 16 19". This label sequence could be further processed to the water meter number "20369.5" as described in Section III-D, which is more reasonable and practically useful in real-world applications.

III. FULLY CONVOLUTIONAL SEQUENCE RECOGNITION NETWORK

The proposed FCSRN consists of three components: a fully convolutional backbone network, temporal mapper and transcription layer (as shown in Fig. 3). Convolutional layers are powerful in terms of learning features from images, and we use an FCN [28] as the backbone network. On the top of FCN, a temporal mapper is used to generate one-dimensional feature sequences from the two-dimensional feature maps. Finally a transcription layer translates these sequential features into the final label sequences.

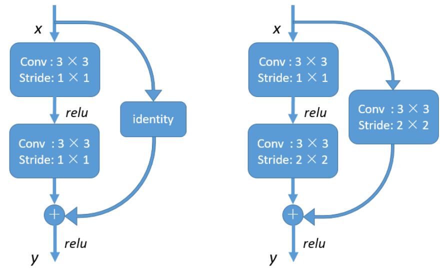

FIGURE 4. Two types of residual blocks. (Left: residual block A) The dimensions of the input (*x*) and output (*y*) are the same. (Right: residual block B) The feature map size of *y* is halved and the number of filters is doubled compared with *x*.

A. FULLY CONVOLUTIONAL NETWORK

We use the residual block proposed in [30] to build the FCN as the backbone network. Two types of residual blocks are adopted in our FCSRN, which are shown in Fig. 4. When input x and output y are of the same dimension, residual block A (shown left) is used to obtain a large receptive field from the increased model depth and avoid the notorious vanishing/exploding gradients problem [20], [21], whereas residual block B (shown right) is used to reduce the feature map and double the number of filters to increase the model capabilities and preserve model complexity [30]. A batch normalization [31] layer and ReLU [32] are used after each convolutional layer.

We construct a 16-layer FCN with residual blocks to extract features from input images, as shown in Table 2.

TABLE 2. Architecture of the fully convolutional backbone network.

layer	channel	kernel	stride	stack
conv+bn+relu	16	3×3	1×1	$\times 1$
residual block A	16	3×3	1×1	$\times 3$
residual block B	24	3×3	2×2	$\times 1$
residual block A	24	3×3	1×1	$\times 3$
residual block B	32	3×3	2×2	$\times 1$
residual block A	32	3×3	1×1	$\times 3$
residual block B	48	3×3	2×2	$\times 1$
residual block A	48	3×3	1×1	$\times 3$

We use small convolutional kernel sizes (3×3) to learn fine-grained local features, and stack multi-convolutional layers to extract long-term features with large receptive fields. As discussed in [33], locations in the response maps correspond to rectangular regions (called receptive fields) in the input images. Layer-wise formulations to calculate exact locations and receptive field sizes are provided as (1, 2):

$$r_i = (r_{i+1} - 1) \times s_i + k_i, \tag{1}$$

$$p_i = s_i \times p_{i+1} + (\frac{k_i - 1}{2} - d_i), \tag{2}$$

where r_i is the local region size of the i^{th} layer, k is the kernel size, s is the stride size, p denotes the position and d is the padding size of a particular layer [33]. We found that the high-layer features in a CNN with sufficient depth correspond to large receptive fields, and are capable of capturing long-term dependencies with contextual information.

B. TEMPORAL MAPPER

As features extracted from the FCN already contain context information, we use a temporal mapper rather than any recurrent or fully connected layers to generate feature sequences before the transcription layer. The temporal mapper consists of a convolution layer, batch normalization [31] layer, and height normalization layer. The convolutional layer outputs K (total character classes plus *blank*) channel feature maps, to generate one feature map for each corresponding category. A convolutional layer is preferred over the fully connected layer because it is more natural for the convolutional structure to enforce the correspondences between the feature maps and the categories [34]. The kernel size, stride, and padding size of the convolutional layer are set to 3×3, 1×1, and 1×1, respectively. We assume that each column of the input feature map is a time step to be predicted; therefore, we use a height normalization layer (also called the average pooling layer) to normalize the features along each column. Therefore, we can map the two-dimensional feature maps to one-dimensional sequential features, which are to be fed into the final transcription layer. As discussed in Section V-D3, the mapped sequential features contain contextual information required for transcription.

C. TRANSCRIPTION LAYER

In WNR, transcription is used to convert features into a sequence of characters that range from "0" to "19". We use connectionist temporal classification (CTC) [29] as the transcription layer. CTC allows the training of the model without any prior alignment between input sequences and labels.

Let's denote the character set as $C = \{0, 1, \cdots, 19, blank\}$, where *blank* represents no prediction at the corresponding time step. Given input feature sequence "$s^k = s_1^k \, s_2^k \cdots s_t^k$", with channels $k = 21$ and total time steps T, CTC specifies a distribution over the sequences by applying a softmax function to each time step, and provides the probability of outputting predicted characters at the corresponding time step. Each latent sequence sampled from this distribution could be transformed into an output sequence using a mapping function σ, which first merges the consecutively repeated non-blank characters into one character and then removes the blank characters. As an example, "1 2 2 5 8" can be transformed by σ from the sequence "$b\ b$ 1 1 b 2 2 2 $b\ b$ 2 $b\ b\ b$ 5 8 8 $b\ b\ b$" (where b represents *blank* and $T = 20$). The final output sequence probability is the summation over all possible sequences for the same target result after applying function σ.

D. DECODING

We use the *naive decoding algorithm*, which is also referred to as *best path decoding* in [29], to make predictions from the output of the FCSRN. We first apply a softmax function to every time step, then the predicted sequence is obtained by considering the most probable label l_t at each time step t, and finally we map the result using function σ (in Section III-C).

As described in Section II, the transcription result of the image (in Fig. 2b, in row 2, column 1) after decoding is: "2 0 3 16 19", and this prediction should be transformed to "20369.5" in real-world applications. Therefore, we introduce a post-processing method for each character c in the "mid-state":

$$c = \begin{cases} c - 9.5 & c \text{ is at the end of sequence} \\ c - 10 & \text{otherwise.} \end{cases} \tag{3}$$

If the predicted digit is at the end of the sequence and is in the "mid-state" (see Section II for details), we subtract 9.5 from the digit. For example, for a predicted result "12", which means that the corresponding number exceeds "2" but does not reach "3", we choose to predict it as "2" and append a "mid-state" flag to the result by adding 0.5 to the final number. If the "mid-state" character is not the last character, then we simply subtract 10 from the digit, thereby indicating that the "lower-state" is selected, which is intuitively more reasonable.

IV. AUGMENTED LOSS FUNCTION

A. AUGMENTED LOSS FOR CTC

For the FCSRN and RNN-based models, we use CTC [29] as the objective function. Given input sequence x with length T, vectors y_t are normalized using the softmax function, then interpreted as the probability of emitting the label (or *blank*) with index k at time step t:

$$Pr(k,t|x) = \frac{exp(y_t^k)}{\sum_{k'} exp(y_t^{k'})}. \tag{4}$$

where y_t^k is element k of y_t. A CTC alignment a is a sequence of *blank*s and labels with length T. Probability $Pr(a|x)$ of a is the product of the emission probabilities at every time step:

$$Pr(a|x) = \prod_{t=1}^{T} Pr(a_t, t|x). \tag{5}$$

For a given transcription sequence, there are as many possible alignments as different ways of separating labels with *blank*s. Additionally, the total probability of an output transcription y is equal to the sum of the probabilities of the alignments corresponding to it:

$$Pr(y|x) = \sum_{a \in \sigma^{-1}(y)} Pr(a|x). \tag{6}$$

Given target transcription y^*, the model can then be trained to minimize the CTC objective function [35]:

$$CTC(x) = -logPr(y^*|x). \tag{7}$$

As described in Section II, a character that exceeds "5" ("lower-state") but does not reach "6" ("higher-state") is defined as "mid-state" and should be labeled as "15". Intuitively, it is confusing for the model to distinguish among the "lower-state", "mid-state" and "higher-state" characters. However, for a sample with label: "1 2 12 15 8", it is tolerable to predict it as, for example, "1 12 12 15 8" or "1 12 2 15 8", because all these possible predictions can be decoded to the same result, "12258", as described in Section III-D. Therefore, we introduce an augmented loss (denoted as AugLoss), which computes the CTC loss of predictions with corresponding "lower-state" labels (denoted as y'), in addition to the normal CTC loss (denoted as CTCLoss) which is calculated using ground truth labels:

$$Aug(x) = -logPr(y'|x). \tag{8}$$

Intuitively, these two losses are both proxies for the recognition accuracy in WNR. We jointly train the network with these two losses to improve the discriminative ability:

$$\begin{aligned} loss &= CTCLoss + \alpha \times AugLoss, \quad \alpha \in [0, 1] \\ &= -logPr(y^*|x) - \alpha \times logPr(y'|x), \end{aligned} \tag{9}$$

where adjustment parameter α represents the importance of AugLoss. AugLoss can be interpreted as biasing the "mid-state" to the "lower-state" rather than the "higher-state" or "out-of-state" (e.g. predicting "1" to "7") predictions. We refer the first scenario as the "mis-state" error (MSE), and the others as the "mis-recognition" error (MRE). As shown in Section V-E, AugLoss can effectively reduce the MRE and improve performance.

B. AUGMENTED LOSS FOR THE ATTENTION-BASED MODEL

For the attention-based model, we use the softmax cross entropy function as the objective function. The probability $Pr_A(y|x)$ of output y is the product of the emission probabilities at every time step:

$$Pr_A(y|x) = \prod_{t=1}^{T} Pr(y_t, t|x). \tag{10}$$

Given target transcription y^*, the model can then be trained to minimize the following objective function:

$$Att(x) = -logPr_A(y^*|x). \tag{11}$$

As described in Section IV-A, we define the augmented loss function as follows:

$$Aug(x) = -logPr_A(y'|x). \tag{12}$$

Additionally, the loss function for attention-based model is:

$$\begin{aligned} loss_A &= AttLoss + \alpha \times AugLoss, \quad \alpha \in [0, 1] \\ &= -logPr_A(y^*|x) - \alpha \times logPr_A(y'|x). \end{aligned} \tag{13}$$

V. EXPERIMENTS

A. DATA PREPROCESSING

The SCUT-WMN dataset contains 5,000 difficult and 1,000 easy samples that encompass a wide range of image sizes and ratios. The maximum and minimum widths of all (difficult and easy) samples are 418 and 201 pixels, respectively. The maximum and minimum heights are 111 and 37 pixels, respectively. Additionally, the maximum and minimum aspect ratios (width/height) are 6.619 and 2.933, respectively.

For parallel computation, we resized all the images so that they had the same height H and width W, with necessary zero padding on the *relatively* short side of the image (by comparing the image ratio with W/H). The aim of data preprocessing is to manage different image sizes while preserving the ratios to avoid distortion. We set $W = 160$ and $H = 48$, as shown in Fig. 5.

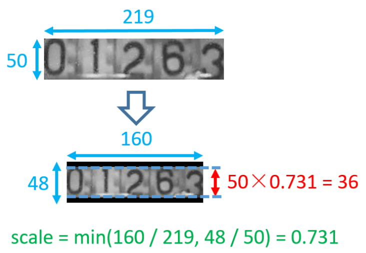

FIGURE 5. Data preprocessing. We first resize the image while maintaining the ratio, and then pad the resized image to a fixed size.

B. PERFORMANCE EVALUATION

For WNR, we used two criteria [36] to measure performance:

$$LCR = N_c/N_l, \tag{14}$$

$$AR = 1 - (D_e + S_e + I_e)/N_t, \tag{15}$$

where LCR represents the line correct rate and AR represents the accuracy rate. N_c is the number of correctly recognized samples, N_l is the number of test samples, and N_t is the number of characters in the test set. The substitution errors S_e, deletion errors D_e and insertion errors I_e are calculated by error correcting string matching using dynamic programming [36].

We also proposed another criterion, called the line partial-accuracy rate (LPR):

$$LPR = (N_c + N_c')/N_l, \tag{16}$$

where N_c' is the number of partial correctly recognized samples, which refers to samples for which the predictions contain error characters but can be decoded to the correct target numbers, as described in Section III-D. For instance, a sample with ground truth "1 2 12 15 8" and prediction "1 12 12 15 8" or "1 12 2 15 8", is a partial correctly recognized sample because both predictions can be decoded to the same water meter number: "12258". There exist several possible predictions that yield the same target number after decoding. Therefore, LPR is a more accurate measurement of the model performance. Additionally, the error rates caused by "mis-recognition" and "mis-state" (described in Section IV) are denoted as the following, respectively:

$$MRE = 1 - LPR \tag{17}$$

$$MSE = LPR - LCR \tag{18}$$

C. TRAINING DETAIL

We constructed the training set based on the **SCUT-WMN** dataset. The training set contained 4,000 difficult samples, and the test set was constructed with the remaining 1,000 difficult samples. For the data augmentation experiment, we added an additional 1,000 easy samples to the training set.

The objective function of the model was minimized using stochastic gradient descent. A batch size of 100 was used. The learning rate, momentum, and weight decay were set to 0.01, 0.9, and 0.0001, respectively. The parameters of the network were initialized using a uniform distribution with a bound range of 0.01. We trained each model for 100 epochs using the MxNet deep-learning framework [37].

D. COMPARISON WITH STATE-OF-THE-ART METHODS

We evaluated various methods for the WNR task. For the segmentation-based methods, we evaluated two convolutional neural networks (ResNet [30] and the backbone network of FCSRN which we call it CharNet). Because all the images in SCUT-WMN contained five digits, we first performed data preprocessing as described in Section V-A and then manually divided each processed image into five equal columns. Then, the task was reduced to an isolated character recognition problem. However, it is notable that in real-world applications, water meters may have various numbers of digits. For the segmentation-free methods, we evaluated the proposed FCSRN, RNN-based CRNN [1], and attention-based ConvS2S [26] models. The detailed settings are as follows:

1) ISOLATED CHARACTER RECOGNITION

We trained an 18-layer ResNet [30] (with output channels set to 20) as the baseline. For the purpose of fair comparison with the FCSRN, we trained a CNN that consisted of the same backbone network as the FCSRN, convolutional layer (with kernel size set to 3×3, stride set to 1×1, and output channels set to 20), batch normalization layer, and global pooling layer. We called this network CharNet. The objective function was the softmax cross entropy loss function.

2) RNN-BASED CRNN AND ATTENTION-BASED CONVS2S

To fairly compare the FCSRN and CRNN [1], we constructed a series of models to determine the optimal hyper parameters. The backbone networks of the CRNN models were the same as the FCN in the FCSRN. We used a multi-layer LSTM (with k layers, where each layer contains n hidden units) and two fully-connected layers (which output c and 21 channel features, respectively), before the final CTC layer on top. We trained a series of models with $k \in [1, 2, 3, 4]$, $n \in [32, 48, 64, 128]$ and $c \in [32, 48, 64, 96, 128, 256]$, and reported the best result, which was obtained at $k = 1, n = 32$, $c = 64$.

For the attention-based ConvS2S [26] model, the backbone network was the same as that of the FCSRN. We used 64 hidden units for both the encoder and decoder. All embeddings, including the output produced by the decoder before the final linear layer, had a dimensionality of 32.

3) PROPOSED FCSRN AND AUGMENTED LOSS

We evaluated the proposed FCSRN with/without AugLoss. For the FCSRN with AugLoss, two temporal mappers and two transcription layers were adopted to compute CTCLoss and AugLoss, respectively, and the same FCN was shared across the two losses. We trained a series of models to determine the optimal hyper parameter α defined in Eq. (9), and reported the best result achieved at $\alpha = 0.2$ (explained in the next section).

The experimental results of the above models are shown in Table 3. We can observe that the FCSRN performed better than the digit recognition models (ResNet18 and CharNet) and comparably with the best CRNN and the ConvS2S models. Therefore, we can conclude that the convolutional model with sufficient depth and a large context window is also able to learn contextual dependence (as discussed in Section III-A) that is required for transcription. The FCSRN took 35% (25%) less training time and involved 45% (33%) fewer parameters compared with the CRNN (ConvS2S).

TABLE 3. Comparison of the proposed methods with state-of-the-art methods.

Model	LCR (%)	AR (%)	LPR (%)	MSE (%)	MRE (%)	training time	storage
ResNet18	81.70	95.92	89.00	7.30	11.00	-	-
CharNet	84.10	96.64	92.30	8.20	7.70	-	-
CRNN [1]	86.10	97.04	92.50	**6.40**	7.50	17 min 30 s	2.2 MB
ConvS2S [26]	85.80	97.28	93.30	7.50	6.70	15 min 07 s	1.8 MB
FCSRN (ours)	85.60	96.98	93.00	7.40	7.00	**11 min 20 s**	**1.2 MB**
FCSRN + *AugLoss* (ours)	**89.60**	**97.82**	**96.80**	7.20	**3.20**	12 min 14 s	1.3 MB

TABLE 4. Effectiveness of the augmented loss function.

Model	LCR (%)	AR (%)	LPR (%)	MSE (%)	MRE (%)
CharNet	84.10	96.64	92.30	8.20	7.70
CRNN [1]	86.10	97.04	92.50	**6.40**	7.50
ConvS2S [26]	85.80	97.28	93.30	7.50	6.70
FCSRN (ours)	85.60	96.98	93.00	7.40	7.00
CharNet + *AugLoss*	87.40	97.28	96.50	9.10	4.60
CRNN [1] + *AugLoss*	89.20	97.66	**97.30**	8.10	**2.70**
ConvS2S [26] + *AugLoss*	88.20	**97.82**	96.00	9.52	4.00
FCSRN + *AugLoss* (ours)	**89.60**	**97.82**	96.80	7.20	3.20

TABLE 5. Data augmentation experimental results.

Model	LCR (%)	AR (%)	LPR (%)	MSE (%)	MRE (%)
FCSRN + *AugLoss*	89.60	**97.82**	**96.80**	7.20	**3.20**
FCSRN + *AugLoss* + Data_Aug	**90.60**	**97.82**	**96.80**	**6.20**	**3.20**

TABLE 6. Optimal hyper parameter α of augmented loss function for different models.

	α	1.0	0.9	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1	*0*
CharNet + AugLoss	$LCR(\%)$	84.40	85.20	**87.40**	85.60	86.30	86.00	85.70	85.00	85.60	87.00	*84.10*
	$AR(\%)$	96.76	96.94	**97.28**	96.84	97.04	97.06	96.94	96.70	97.02	97.24	*96.64*
	$LPR(\%)$	92.80	93.60	95.00	94.40	93.50	93.60	94.20	94.30	**96.50**	94.60	*92.30*
FCSRN + AugLoss	$LCR(\%)$	88.60	88.80	88.60	88.90	89.40	88.70	88.60	88.30	**89.60**	89.20	*85.60*
	$AR(\%)$	97.58	97.62	97.62	97.64	97.72	97.50	97.58	97.44	**97.82**	97.70	*96.98*
	$LPR(\%)$	95.70	96.40	95.90	**97.00**	96.70	96.10	96.00	95.80	96.80	96.40	*93.00*
CRNN + AugLoss	$LCR(\%)$	85.40	72.50	87.30	88.60	88.30	88.00	87.40	**89.20**	*82.20*	82.60	86.10
	$AR(\%)$	96.72	*93.66*	97.28	97.62	97.44	97.48	97.28	**97.66**	96.08	96.02	97.04
	$LPR(\%)$	92.80	*80.90*	96.20	95.80	94.90	96.20	94.90	**97.30**	88.30	89.00	92.50
ConvS2S + AugLoss	$LCR(\%)$	83.70	85.51	80.70	83.30	81.10	*78.20*	86.50	79.70	**88.20**	83.70	85.80
	$AR(\%)$	*96.00*	97.23	96.35	96.70	96.30	95.82	97.23	96.05	**97.82**	96.94	97.28
	$LPR(\%)$	91.30	93.90	90.70	91.50	89.30	95.82	95.00	88.60	**96.00**	*83.70*	93.30

The superior training speed of the FCSRN over the CRNN and ConvS2S was mainly caused by the parallel nature of convolutional operations, as computations in a recurrent model are sequential and cannot be easily parallelized, whereas the decoder in an attention mechanism requires parameters and therefore takes more training time.

The FCSRN equipped with AugLoss outperformed the isolated recognition, RNN-based and attention-based models by a large margin. We achieved a relative reduction for 54.29% on MRE compared with the FCSRN without AugLoss. The AugLoss regarded the "lower-state" and "mid-state" characters as the same class; that is, a character with label "2" predicted as "2" or "12" resulted in no error in AugLoss. However, a character with label "5" predicted as "12" yielded errors for both CTCLoss and AugLoss. Therefore, the model learned to avoid the MRE when being optimized against AugLoss and to distinguish among "different-states" to minimize CTCLoss. Thus, AugLoss effectively reduced the MRE and improved performance.

E. EFFECTIVENESS OF THE AUGMENTED LOSS

The proposed AugLoss could not only be used with the FCSRN, but also improved the performance of CharNet, CRNN, and ConvS2S. We trained these models with the proposed loss function and conducted a series of experiments to determine the optimal hyper parameter α for each model. The experimental results are shown in Table 6, and the best results are shown in Table 4 for comparison.

The optimal α^* for the FCSRN, CRNN, ConvS2S, and CharNet were 0.2, 0.3, 0.2, and 0.8, with relative MRE reductions of 54.29%, 64.00%, 40.30%, and 40.26%, respectively. We conclude that AugLoss is a general technique that effectively improves performance for the WNR task.

F. DATA AUGMENTATION

We adopted data augmentation to improve performance. The training set was constructed with 4,000 difficult and 1,000 easy samples. We enhanced the images by adding noise and randomly adjusting the brightness, saturation, hue, and contrast. The noise consisted of Gaussian-distributed additive noise, Poisson-distributed noise, salt noise, pepper noise, and a random combination of these types of noise. The experimental results (in Table 5) demonstrate that data augmentation slightly improved performance. Further improvements will be considered in future work.

VI. CONCLUSION

In this paper, we focused on the problem of WNR, which is a typical image-based sequence recognition task. We constructed a dataset called SCUT-WMN for open research, and proposed the FCSRN for fast and accurate reading. Furthermore, an augmented loss function to manage intermediate states of characters and reduce the MRE was proposed. The experimental results demonstrate that the FCSRN has the ability to capture contextual information and eliminate the need for recurrent layers, and simultaneously requires fewer parameters and less computation. The FCSRN with AugLoss outperforms RNN-based and attention-based models. Additionally, AugLoss effectively improves performance for RNN-based and attention-based models. WNR has many promising applications in the real world, and we hope that our method along with the dataset can benefit future research on this topic.

REFERENCES

[1] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, Nov. 2017.

[2] A. Bissacco, M. Cummins, Y. Netzer, and H. Neven, "PhotoOCR: Reading text in uncontrolled conditions," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 785–792.

[3] X.-D. Zhou, D.-H. Wang, F. Tian, C.-L. Liu, and M. Nakagawa, "Handwritten Chinese/Japanese text recognition using semi-Markov conditional random fields," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 10, pp. 2413–2426, Oct. 2013.

[4] X.-D. Zhou, Y.-M. Zhang, F. Tian, H.-A. Wang, and C.-L. Liu, "Minimum-risk training for semi-Markov conditional random fields with application to handwritten Chinese/Japanese text recognition," *Pattern Recognit.*, vol. 47, no. 5, pp. 1904–1916, 2014.

[5] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng, "End-to-end text recognition with convolutional neural networks," in *Proc. 21st Int. Conf. Pattern Recognit. (ICPR)*, Nov. 2012, pp. 3304–3308.

[6] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Computer Vision—ECCV*, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Cham, Switzerland: Springer, 2014, pp. 512–528.

[7] A. Mishra, K. Alahari, and C. Jawahar, "Scene text recognition using higher order language priors," in *Proc. Brit. Mach. Vis. Conf. (BMVC)*, 2012, pp. 1–12.

[8] A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2012, pp. 1–8.

[9] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, "Large-lexicon attribute-consistent text recognition in natural images," in *Computer Vision—ECCV*, A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, Eds. Berlin, Germany: Springer, 2012, pp. 752–765.

[10] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 1–20, 2016.

[11] J. A. Rodriguez-Serrano, F. Perronnin, and F. Meylan, "Label embedding for text recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2013, pp. 1–11.

[12] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. (2014). "Synthetic data and artificial neural networks for natural scene text recognition." [Online]. Available: https://arxiv.org/abs/1406.2227

[13] J. Almazán, A. Gordo, A. Fornés, and E. Valveny, "Word spotting and recognition with embedded attributes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 12, pp. 2552–2566, Dec. 2014.

[14] M. Liwicki, A. Graves, S. Fernández, H. Bunke, and J. Schmidhuber, "A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks," in *Proc. 9th Int. Conf. Document Anal. Recognit. (ICDAR)*, 2007, pp. 1–5.

[15] A. Graves, M. Liwicki, S. Fernández, R. Bertolami, H. Bunke, and J. Schmidhuber, "A novel connectionist system for unconstrained handwriting recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 5, pp. 855–868, May 2009.

[16] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Proc. AAAI*, vol. 16, 2016, pp. 3501–3508.

[17] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Computer Vision—ACCV*, D. Cremers, I. Reid, H. Saito, and M.-H. Yang, Eds. Cham, Switzerland: Springer, 2015, pp. 35–48.

[18] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4168–4176.

[19] R. Wu, S. Yang, D. Leng, Z. Luo, and Y. Wang, "Random projected convolutional feature for scene text recognition," in *Proc. 15th Int. Conf. Frontiers Handwriting Recognit. (ICFHR)*, Oct. 2016, pp. 132–137.

[20] S. Hochreiter, "Untersuchungen zu dynamischen neuronalen netzen," M.S. thesis, Josef Hochreiter Inst. Informatik, Techn. Univ. München, München, Germany, 1991, vol. 91, no. 1.

[21] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Netw.*, vol. 5, no. 2, pp. 157–166, Mar. 1994.

[22] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Represent.*, 2015, pp. 1–15.

[23] Y. Kim, C. Denton, L. Hoang, and A. M. Rush, "Structured attention networks," in *Proc. Int. Conf. Learn. Represent.*, 2017, pp. 1–21.

[24] A. P. Parikh, O. Täckström, D. Das, and J. Uszkoreit. (2016). "A decomposable attention model for natural language inference." [Online]. Available: https://arxiv.org/abs/1606.01933

[25] A. Vaswani *et al.*, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[26] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1243–1252.

[27] T. Young, D. Hazarika, S. Poria, and E. Cambria, "Recent trends in deep learning based natural language processing," *IEEE Comput. Intell. Mag.*, vol. 13, no. 3, pp. 55–75, Aug. 2018.

[28] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[29] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks," in *Proc. 23rd Int. Conf. Mach. Learn.*, 2006, pp. 369–376.

[30] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

[31] S. Ioffe and C. Szegedy. (2015). "Batch normalization: Accelerating deep network training by reducing internal covariate shift." [Online]. Available: https://arxiv.org/abs/1502.03167

[32] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Stat.*, 2011, pp. 315–323.

[33] Z. Xie, Z. Sun, L. Jin, Z. Feng, and S. Zhang, "Fully convolutional recurrent network for handwritten Chinese text recognition," in *Proc. 23rd Int. Conf. Pattern Recognit. (ICPR)*, Dec. 2016, pp. 4011–4016.

[34] J. Dong, Y. Gao, H. Li, and T. Guo, "Bilateral filtering NIN network for image classification," in *Intelligent Computing Theories and Application*, D.-S. Huang, K.-H. Jo, and J. C. Figueroa-García, Eds. Cham, Switzerland: Springer, 2017, pp. 655–665.

[35] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1764–1772.

[36] F. Yin, Q.-F. Wang, X.-Y. Zhang, and C.-L. Liu, "ICDAR 2013 Chinese handwriting recognition competition," in *Proc. 12th Int. Conf. Document Anal. Recognit. (ICDAR)*, Aug. 2013, pp. 1464–1470.

[37] T. Chen *et al.* (2015). "MXNet: A flexible and efficient machine learning library for heterogeneous distributed systems." [Online]. Available: https://arxiv.org/abs/1512.01274

FAN YANG received the B.S. degree in electronics and information engineering from the South China University of Technology, Guangzhou, China, in 2016, where he is currently pursuing the master's degree with Deep Learning and Vision Calculations Laboratory. His current research interests include deep learning, computer vision, and optical character recognition.

LIANWEN JIN (M'98) received the B.S. degree from the University of Science and Technology of China, Anhui, China, and the Ph.D. degree from the South China University of Technology, Guangzhou, China, in 1991 and 1996, respectively, where he is currently a Professor with the College of Electronic and Information Engineering. He has authored over 100 scientific papers. His research interests include handwriting analysis and recognition, image processing, machine learning, and intelligent systems. He is a member of the IEEE Computational Intelligence Society, the IEEE Signal Processing Society, and the IEEE Computer Society. He has received the New Century Excellent Talent Program of MOE Award and the Guangdong Pearl River Distinguished Professor Award.

SONGXUAN LAI received the B.S. degree in electronics and information engineering from the South China University of Technology, in 2016, where he is currently pursuing the Ph.D. degree in information and communication engineering. His research interests include machine learning, handwriting analysis, recognition signal processing, and computer vision.

XUE GAO received the Ph.D. degree in electronic science and technology from the South China University of Technology, in 2003, where he is currently an Associate Professor with the School of Electronic and Information Engineering. From 2004 to 2005, he was a Postdoctoral Researcher with the École Polytechnique de l'Université de Nantes, Nantes, France. His current research interests include document analysis and recognition, and machine learning. He is a member of the IEEE.

ZHAOHAI LI received the B.S. degree in electronics and information engineering from the South China University of Technology, in 2016, where he is currently pursuing the master's degree in information and communication engineering. His research interests include deep learning, computer vision, and optical character recognition.