This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3146779, IEEE Transactions on Multimedia

A Two-level Rectification Attention Network for Scene Text Recognition

Lintai Wu, Yong Xu*, *Senior Member, IEEE,* Junhui Hou, *Senior Member, IEEE,* C. L. Philip Chen, *Fellow, IEEE,* and Chenglin Liu, *Fellow, IEEE*

***Abstract*—Scene text recognition is a challenging task in the computer vision field due to the diversity of text styles and the complexity of the image backgrounds. In recent decades, numerous text rectification and recognition methods have been proposed to solve these problems. However, most of these methods rectify texts at the geometry level or pixel level. The former is limited by geometric constraints, and the latter is prone to blurring the text. In this paper, we propose a two-level rectification attention network (TRAN) to rectify and recognize texts. This network consists of two parts: a two-level rectification network (TORN) and an attention-based recognition network (ABRN). Specifically, the TORN first rectifies texts at the geometry level and then performs a pixel-level adjustment, which not only eliminates the geometric constraints but also renders clear texts. The ABRN's role is to recognize text in the rectified images. To improve the feature extraction ability of our model, we design a new channel-wise and kernel-wise attention unit, which enables the network to handle significant variations of character size and channel interdependencies. Furthermore, we propose a skip training strategy to make our model converge smoothly. We conduct experiments on various benchmarks, including regular and irregular datasets. The experimental results show that our method achieves a state-of-the-art performance.**

***Index Terms*—scene text recognition, text rectification, spatial transformer network, optical character recognition**

I. Introduction

TEXTS are widely used in our daily life. We can find texts on documents, doorplates, road signs and other objects. Texts convey a large amount of important information to human beings. Therefore, it is extremely useful to have a robust text recognition system.

To date, the traditional optical character recognition (OCR) technique, which reads texts on scanned documents, is very mature because the optical characters are regular and image backgrounds are clean. However, unlike scanned documents, scene text images are usually taken from a wide variety of scenes, such as streets and supermarkets. Owing to the complex backgrounds and variable text appearances, recognition of scene texts is still a challenging problem. Fig. 1 shows some typical cases of irregular texts, including multioriented texts, curved texts and perspective-distorted texts[1]. Due to the application's needs and challenges, scene text recognition has attracted tremendous attention in the last decade. Numerous methods have been proposed to recognize regular (horizontally aligned) texts or irregular texts (especially curved texts). Irregular scene text recognition has been addressed through shape rectifications or attention-based recognizers[2], [3], [4], [5]. Since Jaderberg et al.[6] proposed spatial transformer networks (STNs), which can rectify irregular text images at the geometry level, an increasing number of text recognition studies have used STN[7], [8], [9], [10]. Although STN can effectively rectify some deformed texts, it performs poorly on curved texts. Subsequently, some researchers have attempted to predict the geometric attributes of texts in images and then use geometric constraints to correct the texts[9]. However, this process requires manually generating geometric labels to supervise the prediction of the corresponding attributes. In addition, the extra geometric-parameter prediction branch increases the complexity of the models. Recently, Luo et al.[11] proposed a pixel-level rectification network, which predicts the coordinate offset of each pixel and then adjusts the pixel values accordingly. However, this method performs poorly in recognizing vertical and highly curved texts.

(a) (b) (c)

Fig. 1. Examples of irregular scene texts. (a) oriented text. (b) curved text. (c) perspective distorted text.

Being aware that single-level rectification methods cannot adequately correct irregular texts, i.e., the geometry-level methods suffer from geometric constraints and the pixel-level methods are prone to blurring texts due to the large variance of the offsets (We demonstrate their drawbacks in detail in Section III-A), in this paper, we propose a Two-level Rectification Attention Network (TRAN) to effectively rectify and recognize scene texts. As illustrated in Fig. 2, the TRAN

Lintai Wu and Yong Xu are with the Bio-Computing Research Center, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, Guangdong, China. They are also with the Shenzhen Key Laboratory of Visual Object Detection and Recognition, Shenzhen 518055, Guangdong, China. And Yong Xu is also with the Peng Cheng Laboratory, Shenzhen 518055, Guangdong, China. (E-mail:wulintai@stu.hit.edu.cn; yongxu@ymail.com).

Junhui Hou is with the Department of Computer Science, City University of Hong Kong, Hong Kong, and also with the Shenzhen Research Institute, City University of Hong Kong, Shenzhen 518057, China. E-mail: jh.hou@cityu.edu.hk.

C. L. Philip Chen is with the School of Computer Science and Engineering, South China University of Technology, Guangzhou 510641, China, also with the Unmanned System Research Institute, Northwestern Polytechnical University, Xi'an 710072, China, and also with the Faculty of Science and Technology, University of Macau, Macau 99999, China. E-mail:philip.chen@ieee.org.

Chenglin Liu is with the NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, P.R. China, and the CAS Center for Excellence in Brain Science and Intelligence Technology, University of Chinese Academy of Sciences, Beijing 100049, China. E-mail: liucl@nlpr.ia.ac.cn.

*Corresponding author: Yong Xu (Email: yongxu@ymail.com)

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3146779, IEEE Transactions on Multimedia

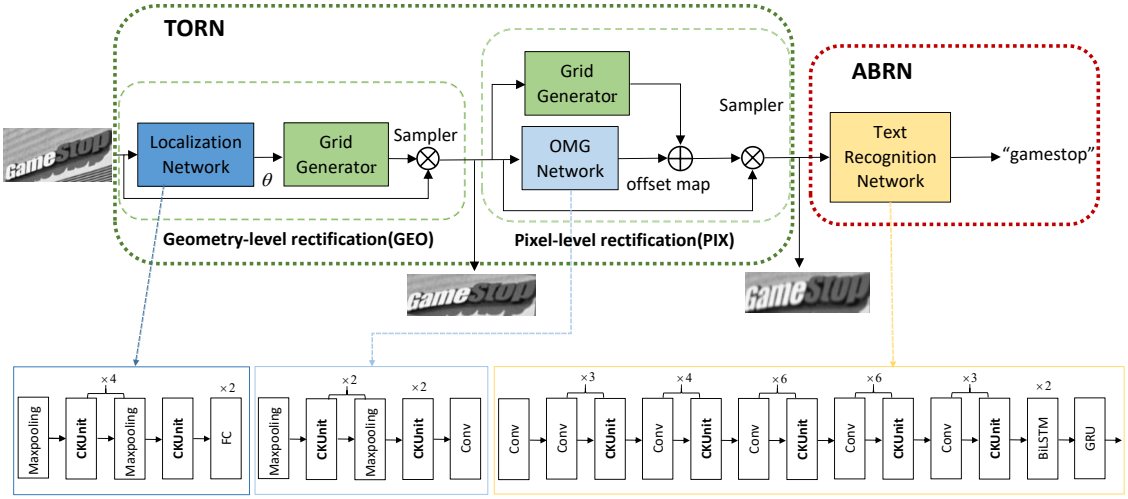

Fig. 2. The overall structure of our model. Note that θ is the parameter of the TPS transformation. The "OMG network" is the offset map generation network. The "CKUnit" is the proposed channel-wise and kernel-wise attention unit.

is composed of two parts: a two-level rectification network (TORN) and an attention-based recognition network (ABRN). Specifically, the TORN consists of a geometry-level rectification network (GEO) and a pixel-level rectification network (PIX). First, the GEO predicts a group of thin plate spline (TPS)[12] parameters and then applies a TPS transformation to the original images. In this step, texts are rectified to a relatively regular level. Next, the PIX predicts an offset map that indicates the coordinate offset of every pixel and then performs a pixelwise adjustment by adding the offsets and pixel coordinates. Note that the range of coordinate offsets to be predicted from images rectified by the GEO is smaller than that predicted from the original images because the rectified images are more similar to the regular ones (see Fig. 2). Since it is easier for a neural network to regress a value within a small range than a value within a large range[13], the PIX can predict the offsets more accurately. To the best of our knowledge, this is the first work utilizing both geometry and pixel-level rectification strategies. Compared with the single-level methods, the two-level rectification strategy can make use of complementary advantages. In addition, unlike the methods relying on the geometric labels of texts, the proposed TORN rectifies texts in a weakly supervised way that needs only text labels for supervision and no geometric attributes are required.

The recognition module named ABRN has the task of recognizing texts in the rectified images. It is a hybrid architecture composed of a convolutional neural network (CNN) and a recurrent neural network (RNN), whose decoder is an attentional sequence-to-sequence model.

The character size variance is commonly very large in text recognition datasets. However, few studies take this variance into consideration. In this paper, to address this issue, we design a new attention unit that can automatically assign weights to different sizes of kernels and thus adjust the receptive fields dynamically. In addition, as illustrated in [14], the interdependencies between the channels of the feature maps are of great importance to the representational ability of a network, and we integrate them into the unit of another mechanism to cause the network to automatically recalibrate the channel correlation. We term this channel-wise and kernel-wise attention unit, the CKUnit.

Since the model consists of three series-connection subnetworks (namely, GEO, PIX and ABRN), convergence is difficult if these subnetworks are trained simultaneously. To solve this problem, we propose a skip training strategy to make our model converge smoothly.

In summary, the major research contributions of this paper are as follows:

1) We propose an end-to-end scene text rectification and recognition network named TRAN. TRAN rectifies texts at both the geometry level and pixel level without supervision, which proves to be more effective than single-level rectification methods.

2) We propose a new channel-wise and kernel-wise attention unit to tackle the large variance of character size and channel interdependencies to enhance the feature extraction ability of our model. Experimental results show that our attention mechanism achieves a significant improvement in recognition accuracy.

3) We propose a skip training strategy, which enables our model to converge smoothly in cases where the entire network is relatively deep.

4) We conduct experiments on standard text recognition datasets, including ICDAR2003, ICDAR2013, ICDAR2015, IIIT5K, SVT, SVT-Perspective and CUTE80. The experimental results show that our method achieves state-of-the-art performance.

II. Related Work

A. Scene Text Recognition

In the early years, researchers devoted attention to recognizing texts with traditional methods, such as clustering[15], hidden Markov modeling (HMM)[16], stroke width transform (SWT)[17], [18], [19] and stable extreme regions (MSER)[20], [21]. With the great success of deep learning methods for

many computer vision tasks[22], [23], [24], [25], [26], [27], [28], [29], [30], [31], an increasing number of text recognition endeavors were based on CNNs, which dramatically improved the methods' recognition performance. Jaderberg et al.[32] viewed text recognition as a word classification problem, and they used CNNs to classify each word over 90k-class labels. Jaderberg et al.[33] developed a CNN architecture that could detect and recognize characters simultaneously.

In the past few years, since RNNs performed outstandingly on sequence processing tasks, an increasing number of researchers have treated text recognition as a sequence labeling problem[34], [35]. The decoders of text-sequence recognitions can be divided into two categories: connectivist temporal classification-based (CTC) methods[36], [10], [37], [38], [39], [40] and attention-based methods[41], [42], [43], [44], [45], [46], [47], [48], [49], [50]

The CTC-based methods divide text sequences into several frames and then predict the label of each frame. Shi et al.[10] proposed a model that used RNNs to process feature sequences and made predictions and then employed CTC to translate the framewise prediction to a label sequence. Instead of RNNs, Gao et al.[39] used CNNs to capture the contextual dependencies of feature sequences, which greatly reduced the number of parameters.

Attention-based methods classify characters in an orderly way by using the attention vectors generated from the feature sequence[51]. Shi et al.[52] introduced an attention mechanism to automatically align and read characters. Cheng et al.[41] discovered the attention drift phenomenon and proposed a focusing attention mechanism to alleviate this problem.

B. Irregular Text Recognition

Irregular texts, which are ubiquitous in the real world, are one of the most difficult texts to recognize. To address text deformations from shapes and fonts, an STN is usually employed. The STN predicts several transformation parameters to rectify texts at the geometry level. Two typical transformation methods are the TPS and affine transformation. Shi et al.[7] used TPS to rectify text images and then input the results to the sequence recognition network, resulting in a high improvement in accuracy. Liu et al.[8] proposed a model named CharNet to detect characters by a localization network and then used STN to rectify each individual character. However, it is very difficult to localize all characters accurately without a character-level annotation. Bartz et al.[53], [54] used STN to simultaneously localize and rectify texts in images in a semi-supervised way. Sun et al.[55] employed perspective transformations to enhance the rectification of perspective texts.

However, a simple STN performs well on slightly deformed texts but poorly on highly deformed texts, such as curved texts[56]. Therefore, researchers tend to rectify irregular texts with the help of the geometric attributes of the texts. Zhan et al.[9] proposed predicting the middle line of scene texts by using a polynomial regression. Then, a line-fitting transformation was designed to iteratively correct the text direction.

Although this method can effectively rectify curved texts, multiple iterations of rectification and extra geometric parameter predictions lead to more parameters and higher time consumption. In a variation from geometry-level methods, Luo et al.[11] proposed a new method named MORAN, which rectifies texts at the pixel level. MORAN utilizes CNNs to predict an offset feature map to calibrate each image pixel horizontally and vertically. Later, Luo et al. published a new version of MORAN[1], which only rectified text images vertically. The second version of MORAN achieved a better performance than its first version.

III. Methodology

The TRAN is comprised of two parts: TORN and ABRN. TORN jointly rectifies irregular texts at both the geometry level and pixel level. ABRN is an attention-based sequence recognition network. To enhance the text-feature extraction ability of our model, we propose a new channel-wise and kernel-wise attention unit. For the training stage, we propose a skip training strategy to make our model converge smoothly under weak supervision.

A. Two-level Rectification Network (TORN)

We observe that there are some limitations with single-level rectification methods. Geometry-level methods usually rectify entire texts with some transformations, such as rotation, translation and scaling[11]. However, the deformations of some texts are too complicated to be rectified by these transformations. Taking TPS as an example, as Table I shows, this transformation will make characters slant when rectifying horizontally misaligned texts. With respect to the pixel-level methods, in some cases, characters will be overrectified by this method and become highly blurred. We argue that this is because the offsets of some highly irregular texts vary in a large range, which makes accurate regressions more difficult[13].

By contrast, in our model we make use of the complementary advantages of geometry-level and pixel-level methods. Specifically, our model first employs TPS to rectify texts. After this rectification, some irregular texts are closer to regular texts in shape, and thus, the variances of the pixel offsets between them are reduced. Next, the model predicts more accurate offsets and then performs pixelwise adjustments. As a result, it not only aligns characters more effectively without slant but also keep texts at a relatively high resolution.

Fig. 2 illustrates the overall architecture of TORN, which consists of a geometry-level rectification module and a pixel-level rectification module.

1) Geometry-level rectification network: Based on the STN, we use the TPS to transform the source image I to the target image I_g at the geometry level. We generate $2K$ equidistant control points that are distributed on the top and bottom borders of the target image, with K points on each border. The most critical step of our method is to find homologous $2K$ control points on I by using a rectification network.

As depicted in Fig. 2, this rectification network consists of three parts: a localization network, a grid generator and a sampler.

[1]https://github.com/Canjie-Luo/MORAN_v2

TABLE I
THE RECTIFICATION RESULTS OF OUR METHODS AND SINGLE-LEVEL METHODS. THE GREEN LINES INDICATE THE VERTICAL DIRECTION OF THE CHARACTERS AND THE RED LINES SHOW THE HEIGHT DIFFERENCE OF TWO CHARACTERS. NOTE THAT THE GEOMETRY-LEVEL METHOD IS BASED ON A TPS TRANSFORMATION IN THIS FIGURE.

Input image	By geometry level	By ours
		
Input image	**By pixel level**	**By ours**
		

The localization network targets predicting $2K$ control points on the source image. It consists of several convolutional and pooling layers followed by two fully connected layers. Its architecture will be introduced in Section III-A3 in detail.

The grid generator generates transformation parameters to compute a sampling grid on I according to the control points. In this paper, $c = [c_1, ..., c_{2K}]$ and $c^g = [c_1^g, ..., c_{2K}^g]$ denote the coordinates of $2K$ control points on I and I_g, respectively, where $c_i = (x_i, y_i)^T$ are the x, y coordinates of the ith point of c, likewise $c_i^g = (x_i^g, y_i^g)^T$. Given a target point $p_i^g = [x_i^g, y_i^g]^T$ on I_g, the corresponding source point on I is computed as:

$$p_i = T_\theta \begin{bmatrix} 1 \\ p_i^g \\ \phi(||p_i^g - c_1^g||) \\ \vdots \\ \phi(||p_i^g - c_{2K}^g||) \end{bmatrix}, \tag{1}$$

where $\phi(r) = r^2 log(r)$ is the radial basis kernel. T_θ is the transformation matrix generated as follows[7]:

$$T_\theta = \begin{bmatrix} u & 0 & 0 & 0 \\ v & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 1^{1\times 2K} & 0 & 0 \\ c & 0 & 0 \\ \hat{c} & 1^{2K\times 1} & c^T \end{bmatrix}^{-1}, \tag{2}$$

where $u = [x_1, ..., x_{2K}]$ and $v = [y_1, ..., y_{2K}]$ are the x and y coordinates of c, respectively. $\hat{c} \in \mathbb{R}^{2K\times 2K}$ is a matrix comprised of $\hat{c}_{i,j} = \phi(||c_i - c_j||)$.

After the transformation matrix is computed, the sampling grid $p = [p_1, ..., p_{H_g \times W_g}]$can be generated as Eq. 1, where H_g and W_g are the height and width of the target image, respectively. According to the sampling grid, the target image is generated by sampling each pixel from the source image using a bilinear interpolation. Specifically, for $\forall i \in [1, ..., H_g \times W_g]$, the value at location (x_i^g, y_i^g) of the target image is as follows[6]:

$$V_i = \sum_n^H \sum_m^W U_{nm} k(x_i - m; \phi_x) k(y_i - n; \phi_y), \tag{3}$$

where U_{nm} is the value at location (n, m) of I. H and W denote the height and width of I. ϕ_x and ϕ_y are the parameters of a generic sampling kernel $k()$ defining the bilinear interpolation method.

2) Pixel-level rectification network: The pixel-level rectification network produces an offset map to adjust the value of each pixel. Note that we adjust only the y coordinate of the pixel because it was found that adding the x-coordinate correction hardly improves the result. The rectification procedure is as follows:

First, we generate a source grid of size $H_g \times W_g$, which is the same as that of the input image I_g. The source grid is composed of two channels of maps that represent the x and y coordinates of each pixel in the input image.

Next, by inputting I_g to a fully convolutional network, we generate an offset map of size $H_o \times W_o$, where $H_o < H_g, W_o < W_g$. Then, we apply the bilinear interpolation method to resize the offset map to the same size as the input image. In this way, each value of this offset map represents the offset of the y coordinate in the input image.

Then, the target grid is obtained by summing the offset map and the channel, denoting the y coordinate in the source grid as follows:

$$\begin{aligned} G_x(i,j)' &= G_x(i,j), \\ G_y(i,j)' &= G_y(i,j) + O(i,j), \end{aligned} \tag{4}$$

where $i = 1, ..., W_g; j = 1, ..., H_g$. G_x and G_y are the feature maps denoting the x and y coordinates in the source grid, respectively. O represents the offset map.

Finally, the rectified image is obtained by sampling from the input image according to the target grid. The value at position (i, j) in the rectified image is:

$$\begin{aligned} I'(i,j) &= I_g(i', j'), \\ i' &= G_x(i,j)', \\ j' &= G_y(i,j)'. \end{aligned} \tag{5}$$

Similar to the GEO module, we apply the bilinear interpolation method to sample each pixel from I_g.

3) Channel-wise and Kernel-wise Attention Unit: Most studies on the attention mechanism of text recognition focus on the RNN module, but few researchers are concerned with the attention mechanism on the feature extraction module. In fact, it is common for text recognition datasets to have texts in images that vary in size, and the fewer characters the image contains, the larger these characters are. Fixed size receptive fields cannot adapt to the variation in character size. Therefore, different sizes of receptive fields are required. Inspired by [57], we design an inception-like[24] unit that combines kernels of different sizes in parallel. Furthermore, we propose a mechanism that allows the network to learn the weights of different kernels to dynamically adjust the importance of the different sizes of kernels. However as demonstrated in [14], the interdependencies between the channels of the feature maps have a large impact on the representational abilities of a network. Therefore, we integrate into the block another mechanism that enables the network to automatically recalibrate the channel correlation.

Thus, this unit improves the text-feature extraction ability of the network by selectively emphasizing the informative channels and kernels but suppresses the less useful channels and kernels. In this paper, we term this new channel-wise and

kernel-wise attention unit the "CKUnit", and its architecture is introduced below.

As depicted in Fig. 3, given an input feature map I with a size of $c \times h \times w$, we conduct two convolution operations $I \rightarrow E_1 \in \mathbb{R}^{c \times h \times w}$ and $I \rightarrow E_2 \in \mathbb{R}^{c \times h \times w}$ with two kernels k_1 and k_2. Then, by employing the global pooling (GP) operation, we compress each channel of E_1 and E_2 and use a single value to represent the information of each channel:

$$p_n = f_{GP}(E_n) = \frac{1}{h * w} \sum_{i=1}^{h} \sum_{j=1}^{w} E_n(i,j), \ n = 1, 2. \quad (6)$$

Furthermore, to decrease the number of parameters, the channel number is reduced to $\frac{c}{z}$ by a fully connected layer:

$$d_n = f_{FC}(p) = \delta(W * p_n), \ n = 1, 2, \quad (7)$$

where δ denotes the rectified linear unit (ReLU), $W \in \mathbb{R}^{\frac{c}{z} * c}$, and z is a positive integer. Thus the size of d_n is $\frac{c}{z} \times 1 \times 1$.

Next, d_1 and d_2 are spliced and processed by the convolution operation. Then, we apply the sigmoid function to all elements of the feature map and split the result into two $c \times 1 \times 1$ feature maps v_1 and v_2, which represent the weight vectors of E_1 and E_2, respectively.

$$v_n(k) = \frac{1}{1 + e^{A * d_n}} + 1, \quad (8)$$

where $k = 1, ..., c, \ A \in \mathbb{R}^{c \times \frac{c}{z}}$.

Finally, we multiply the feature maps E_1 and E_2 by the corresponding weight vector and then combine the result by elementwise addition.

$$O(k) = v_1(k) * E_1(k) + v_2(k) * E_2(k), \quad (9)$$

where $O = [O(1), O(2), ..., O(c)], \ O(k) \in \mathbb{R}^{h \times w}$.

The CKUnit is embedded into the architecture of TORN, which is shown in Fig. 2. In our work, the sizes of kernels k_1 and k_2 are set to 1×1 and 3×3, respectively.

B. Attention-based Recognition Network

The recognition network is a CRNN-based[10] attention model whose architecture is shown in Fig. 2. It consists of an encoder and a decoder. The encoder extracts image features with a residual network[23] extended by two layers of bidirectional LSTM (BLSTM). Note that the residual network is also embedded with the CKUnit.

The decoder iteratively converts the feature sequence H produced by the encoder into a character sequence, denoted by $(y_1, y_2, ..., y_T)$. In each step, the decoder first computes a set of weights $a_t \in \mathbb{R}^n$ as an attention vector:

$$\begin{aligned} e_{t,i} &= w^T tanh(W s_{t-1} + V h_i + b), \\ a_{t,i} &= \frac{exp(e_{t,i})}{\sum_{i'=1}^{n} exp(e_{t,i'})}, \end{aligned} \quad (10)$$

where s_{t-1} and h_i denote the internal state and the ith element of the input sequence, respectively. w, W and V are weights of the model. Next, the decoder generates a glimpse based on the input sequence and attention vector:

$$g_t = \sum_{i=1}^{n} a_{t,i} h_i. \quad (11)$$

Then, the glimpse is fed into a GRU[58] cell to produce an output vector and a state vector:

$$(x_t, s_t) = GRU(s_{t-1}, g_t, f(y_{t-1})), \quad (12)$$

where $f(y_{t-1})$ is the embedding vector of the previous predicted symbol. Finally, the output value is obtained as follows:

$$y_t = Softmax(W_{out} x_t + b_{out}). \quad (13)$$

To capture more complete text information, we employ a bidirectional decoder proposed by Shi et al[7], which contains two decoders with opposite directions.

C. Skip Training

Because our model is composed of three series-connected deep networks, convergence is difficult if these subnetworks are trained simultaneously. To this end, we propose a simple yet effective skip training strategy. The general idea of this strategy is to first make some of the modules roughly converge and then use the parameters of these modules to guide the training of the other modules to make the entire network converge.

In the first stage, we skip the PIX and optimize only the GEO and ABRN. These two modules converge rapidly in this way, and the ABRN is initially capable of recognizing rectified texts.

In the second stage, we connect the PIX with the GEO and ABRN and jointly optimize these three subnetworks. The optimization of the PIX module will be smooth under the guidance of the GEO and ABRN, and the end-to-end training contributes to a more complete convergence of the model. Fig. 4 depicts the general procedure of the skip training strategy.

In addition, following [11], we take the fractional pickup method to enhance the ability of the decoder in the training stage.

The loss function of the TRAN is as follows:

$$L = -\sum_{t=1}^{T} \log \ p_1(y_{i,t}|I_i) + \log \ p_2(y_{i,t}|I_i), \quad (14)$$

where p_1 and p_2 represent the predicted distribution of the bidirectional decoders. $I_i (i = 1, 2, ..., T)$ denotes the ith element of the training set. $y_{i,t}$ is the ground truth of the t th character in I_i.

IV. Experiments

Because there are no pretrained models, we train TRAN via two synthetic datasets from scratch. Then, we use seven benchmarks to test our model, including regular texts and irregular texts. It is noteworthy that all these benchmarks contain texts of various sizes. In addition, we used word accuracy as the assessment criterion for all the methods.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3146779, IEEE Transactions on Multimedia

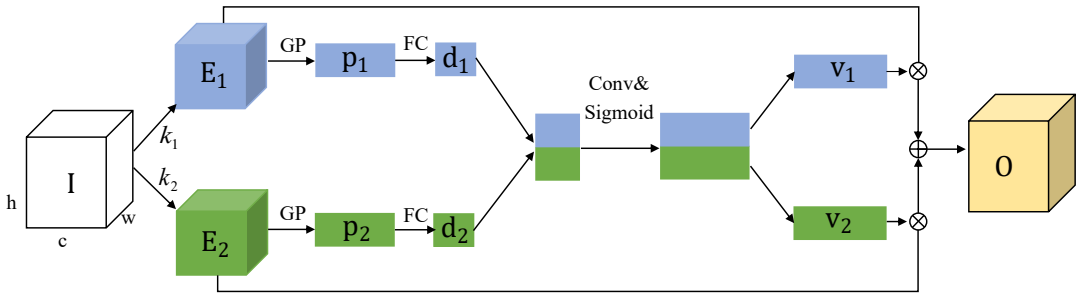

Fig. 3. The architecture of the CKUnit."k_1" and "k_2" represent two kernels with different sizes. "GP", "FC" and "Conv" denote the global pooling layer, fully connected layer and convolutional layer, respectively.

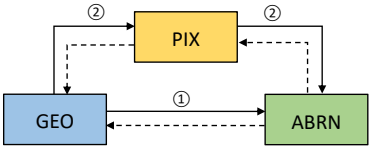

Fig. 4. The procedure of skip training. The dashed lines show the direction of gradient propagation.

A. Datasets

Synth90k[32] is a synthetic text dataset with a lexicon of 90 K. It contains 9 million text images with ground truth word annotations. All the images of Synth90k are used for training in our experiments.

SynthText[59] is the synthetic text dataset used for text detection. It is annotated with ground truth bounding boxes and word labels. Using the word bounding boxes, we crop 4 million text image patches from SynthText for training.

IIIT5k-Words (IIIT5k)[60] consists of 2000 training images and 3000 test images cropped from the street, web, etc. But only the test images are used in our experiment. IIIT5K contains many regular texts and a small quantity of irregular texts.

Street View Text (SVT)[61] is comprised of 647 word images that are captured from Google Street View. Many words in the SVT images are horizontal or heavily corrupted by noise, blur or low resolution.

ICDAR 2003 (IC03)[62] contains 867 images where most of the texts are regular. Following [61] and [11], we discard images where the number of characters is less than three or the images contain nonalphanumeric characters.

ICDAR 2013 (IC13)[63] inherits most of its data from IC03 but is extended it with new samples, resulting in 1015 images being contained in total.

ICDAR 2015 (IC15)[64] contains 2077 text images, where most of the texts suffer from various distortions in regard to orientation, perspective and curvature.

SVT-Perspective (SVTP)[1] contains 645 text images, where most of the words suffer from perspective distortion.

CUTE80 (CUTE)[65] contains 288 text images, where most words are highly curved.

B. Implementation Details

We implement our model by using the PyTorch framework. Without a data augmentation, images are transformed to grayscale and are resized to 64×200 before being fed into the network, and the input size of ABRN is set to 32×100. The number of control points for the TPS transformation is set to 16. The kernel sizes in all the CKUnits are 1×1 and 3×3, and the stride sizes are 1. The padding sizes are 1 and 0 for large and small kernels, respectively. In TORN, the kernel size, stride and padding of standard convolutional operations are 3×3, 1 and 1, respectively. Except for the CKUnits, the configurations of ABRN are in accordance with the recognition network in [11]. With the ADADELTA[66] optimizer and a batch size of 128, the model is trained for 1 and 5 epochs in two stages of the skip training strategy. The learning rate is set to 1.0 at the beginning and decreased to 0.1 and 0.01 in the fourth and fifth epochs, respectively. In addition, the model is trained on an NVidia GTX-1080Ti GPU with 11 GB memory.

C. Results Comparison

1) Comparison with State of the Art: We compare the performance of our method with a number of classic and notable algorithms. The results are indicated by the word recognition accuracy without a lexicon, which is shown in Table II. Our method achieves the 3 best results and the 5 top three results. Furthermore, the TRAN yields a competitive performance with respect to both the regular texts and most of the irregular texts. Specifically, the accuracy on the SVTP dataset exceeds the previous state-of-the-art method by 2.2%, which indicates that the TRAN is adept in recognizing perspective texts.

Moreover, compared with the classic geometry-level rectification method, ASTER[7] and pixel-level rectification method MORAN[11], TRAN achieves a large performance improvement over all the benchmarks, especially for irregular datasets.

In addition, it is worth noting that the TRAN even outperforms the methods that utilize the geometric attributes of texts[9] or the methods that use semantic information[43] for supervision. On the one hand, although the supervised rectification approaches rectify irregular texts better than our weakly supervised method, the gap of the rectification results is relatively small except on highly irregular texts. However, there are only a few highly irregular texts in the used datasets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3146779, IEEE Transactions on Multimedia

TABLE II
LEXICON-FREE RESULTS ON BENCHMARKS. **BOLD** REPRESENTS THE BEST RESULT. UNDERLINE IN THE LAST ROW REPRESENTS THE TOP-THREE RESULTS. '*' INDICATES THAT A CHARACTER-LEVEL ANNOTATION IS REQUIRED. '△' INDICATES THAT EXTRA SEMANTIC SUPERVISION IS REQUIRED.

Methods	IIIT5K	SVT	IC03	IC13	IC15	SVTP	CUTE
Jarderberg *et al.*[67](2015)	-	71.7	89.6	81.8	-	-	-
Jarderberg *et al.*[68](2016)	-	80.7	93.1	90.8	-	-	-
Shi *et al.*[52](2016)	81.9	81.9	90.1	88.6	-	71.8	59.2
Lee *et al.*[46](2016)	78.4	80.7	88.7	90.0	-	-	-
Shi *et al.*[10](2017)	81.2	82.7	91.9	89.6	-	-	-
Cheng *et al.*[41](2017)*	87.4	85.9	94.2	93.3	70.6	-	-
Cheng *et al.*[4](2018)	87.0	82.8	91.5	-	68.2	73.0	76.8
Liu *et al.*[8](2018)*	92.0	85.5	92.0	91.1	74.2	78.9	-
Bai *et al.*[69](2018)*	88.3	87.5	94.6	94.4	73.9	-	-
Liu *et al.*[70](2018)*	87.0	-	93.1	92.9	-	-	-
Liu *et al.*[71](2018)	89.4	87.1	94.7	94.0	-	73.9	62.5
Shi *et al.*[7](2018)	93.4	89.5	94.5	91.8	76.1	78.5	79.5
Liao *et al.*[72](2019)*	91.9	86.4	-	91.5	-	-	79.9
Xie *et al.*[73](2019)	-	-	-	-	68.9	70.1	82.6
Li *et al.*[5](2019)	91.5	84.5	-	91.0	69.2	76.4	83.3
Luo *et al.*[11](2019)	91.2	88.3	95.0	92.4	74.7	76.1	77.4
Zhan *et al.*[9](2019)	93.3	90.2	-	91.3	76.9	79.6	83.3
Gao *et al.*[45](2020)	**94.3**	88.7	94.0	93.3	76.8	81.2	**88.2**
Huang *et al.*[42](2020)	94.0	88.9	95.0	**94.5**	73.9	79.4	82.6
Qiao *et al.*[43](2020)△	93.8	89.6	-	92.8	**80.0**	81.4	83.6
Wang *et al.*[44](2020)	**94.3**	89.2	95.2	94.2	74.5	80.0	84.4
Zhang *et al.*[74](2021)	90.3	89.5	95.4	**96.8**	76.0	78.5	78.9
Gao *et al.*[75](2021)	91.4	88.8	**95.5**	94.9	78.6	82.8	77.5
Rowel *et al.*[76](2021)	88.4	87.7	94.3	92.4	72.6	81.8	81.3
TRAN (ours)	**94.3**	**90.7**	95.3	93.6	77.7	**85.0**	81.6

TABLE III
INFERENCE SPEED OF OUR METHOD AND SINGLE-LEVEL RECTIFICATION METHODS. 'FPS'(FRAMES PER SECOND) INDICATES THE RECOGNITION SPEED OF MODELS.

Methods	Speed (FPS)	Average Accuracy (%)	Rectification strategy
ESIR[9]	35.71	85.77	Geometry-level
MORAN[11]	29.80	85.01	Pixel-level
ours	28.40	88.31	Two-level

TABLE IV
PERFORMANCE COMPARISON WITH SINGLE-LEVEL RECTIFICATION METHODS. 'FPS'(FRAMES PER SECOND) INDICATES THE RECOGNITION SPEED OF MODELS.

Methods	Datasets							FPS
	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE	
GEO+ABRN	93.6	89.2	95.0	93.5	76.6	82.2	**82.3**	30.81
PIX+ABRN	93.3	88.9	94.9	92.9	75.7	81.7	79.9	29.13
GEO+PIX+ABRN (ours)	**94.3**	**90.7**	**95.3**	**93.6**	**77.7**	**85.0**	81.6	28.40

On the other hand, it is common that the variance of character size is very large in text recognition datasets. The proposed channel-wise and kernel-wise attention units help our model flexibly extract features of different sizes of texts.

2) Speed analysis: To evaluate the efficiency of our model, we compare the inference speed of our method with two single-level models, ESIR[9] and MORAN. Both models were tested with the same configuration as ours. As shown in Table III, although our method uses more computational time than these state-of-the-art methods, the gap is small. Moreover, our method outperforms ESIR and MORAN by a large margin in terms of the average accuracy of the seven datasets.

D. Ablation Study

1) Comparison with single-level rectification methods: To evaluate the effect of the two-level rectification strategy, we compare the recognition accuracy of our method with that of the methods that only use GEO or PIX as the rectification module. For a fair comparison, all three methods use the ABRN as the recognition module.

As Table IV shows, the recognition accuracy of our two-level rectification method exceeds two single-level rectification methods on most datasets. Particularly, the accuracy gaps are no less than 1.9% on the SVT, IC15 and SVTP datasets, which indicates that the two-level rectification method is more adept in recognizing oriented and perspective texts compared to the single-level rectification methods. Nevertheless, our method falls short on one result. The accuracy of our method on the CUTE dataset is less than the "GEO+ABRN" by 0.7%, which suggests that the effect of the two-level rectification strategy is slightly worse than the GEO with respect to highly curved texts. We can see from the last column that the recognition speed of the two-level rectification module only falls behind that of GEO or PIX by less than 2.41 FPS.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3146779, IEEE Transactions on Multimedia

TABLE V
THE RECTIFICATION RESULTS OF OUR METHODS AND TWO STATE-OF-THE-ART METHODS. THE ASTER AND THE MORAN ARE GEOMETRY-LEVEL AND PIXEL-LEVEL METHODS, RESPECTIVELY.

Input image	Rectified by ASTER[7]	Rectified by ours
		
		
Input image	Rectified by MORAN[11]	Rectified by ours
		
		

In addition, we compare the rectification results between our TRAN and two state-of-the-art single-level methods. As Table V shows, our method achieves better rectification performance compared with ASTER and MORAN, which indicates the effectiveness of the two-level rectification strategy.

2) Performance analysis of the CKUnit: To demonstrate the effect of the CKUnit, we first analyze the kernel weight in the CKUnit when the model recognizes texts of different sizes. We crop 1,2,...7 characters from an image containing 8 characters in total and resize them to 100×32. Then, we input them to the ABRN in order and calculate the average gap between the weights of 3×3 and 1×1 kernels. Table VI shows the cropped and original images and the prediction results of the ABRN. Fig. 5(a) compares the weight gap between 3×3 and 1×1 kernels of the CKUnit when the above 8 images are fed into the ABRN. It can be observed that the weight of the 3×3 kernel is larger than that of the 1×1 kernel across all images, but as the character sizes decrease, the gap between the 3×3 and 1×1 kernels decreases. This indicates that the importance of small kernels increases when the texts are small and vice versa.

Furthermore, to understand the requirement of the three modules for different sizes of kernels, we calculate the average gap between the weights of 3×3 and 1×1 kernels in the GEO, PIX and ABRN. This experiment is conducted on the IC15 dataset, which consists of 2077 images. The result is shown in Fig. 5(b). The mean weight of the 3×3 kernel is larger than that of the 1×1 kernel in the GEO and ABRN, but the opposite occurs in the PIX. We observe that the PIX conducts a pixel-level rectification and produces the offset of every pixel, so it relies more on a 1×1 kernel to extract pixel-level features.

In addition, to explore the channel selection of the CKUnit, we sample the channel weight of one CKUnit layer in the GEO, PIX and ABRN. As Fig. 5(c)(d)(e) shows, the weights of different channels vary to different extents, which demonstrates that the CKUnit is capable of calibrating the weights of channels automatically. Finally, we compare the performance across the models in which different modules are embedded with the CKUnit. As Table VII shows, whether embedding the CKUnit with the TORN or the ABRN, the performance

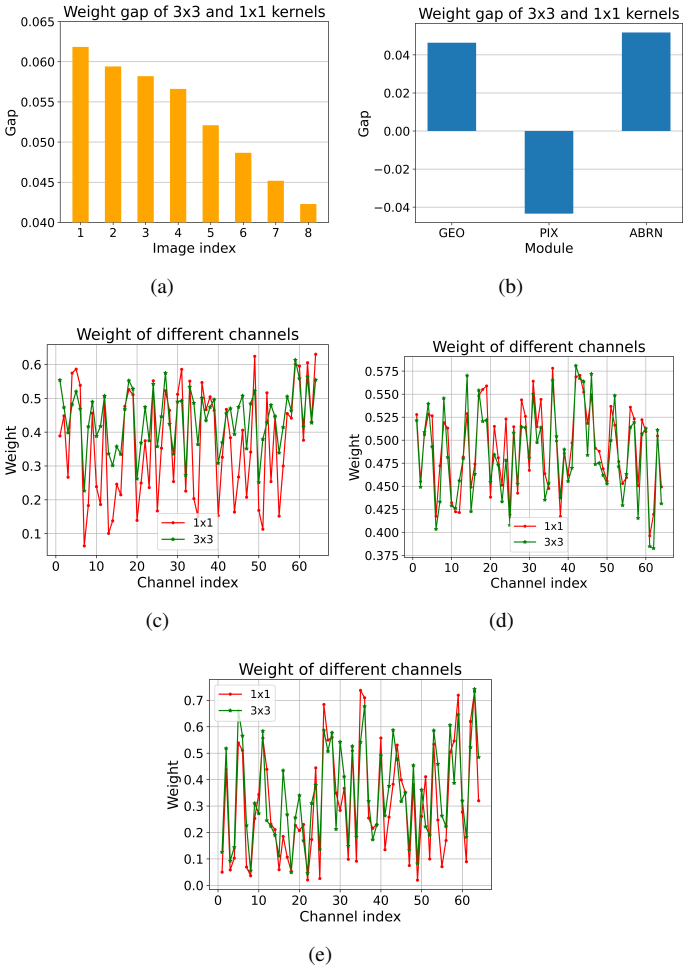

Fig. 5. (a) Mean weight gap between 3×3 and 1×1 kernels of the CKUnit in the ABRN when the images in Table VI are fed into the ABRN. (b) Mean weight gap between 3×3 and 1×1 kernels of the CKUnit in the GEO, PIX and ABRN. (c) Channel weights of the second CKUnit in the GEO. (d) Channel weights of the first CKUnit in the PIX. (e) Channel weights of the fourth CKUnit in the ABRN.

of the model is considerably improved. Specifically, compared with the raw model, the model with the CKUnit can greatly improve the recognition accuracy on irregular datasets. These results suggest that the CKUnit has positive effects on both rectification and recognition tasks. It can be concluded from the aforementioned analysis that the CKUnit can highly enhance the text-feature extraction ability of the model and thus improve the recognition accuracy.

3) Visualization of rectification results: To observe the effect of the TORN qualitatively, we visualize the rectification images of benchmarks. Some samples are shown in Table VIII. Since the input images in our experiment are grayscale with a size of 64×200, we adopt the same settings for the visualization images for authenticity.

As seen from Table VIII, the TRAN has outstanding performance on rectifying oriented and perspective texts. Even for images with complicated backgrounds, our model can still locate and rectify the texts effectively (e.g., “supermodel”, “personal” and “chocolates”). The TRAN is also capable of rectifying slightly curved texts (e.g., “close” and “chocolates”).

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TMM.2022.3146779, IEEE Transactions on Multimedia

TABLE VI
CROPPED IMAGES. NO.1-7 ARE THE CROPPED IMAGES AND NO.8 IS THE ORIGINAL IMAGE.

Image index	1	2	3	4	5	6	7	8
Image								
Ground truth	p	pe	per	pers	perso	person	persona	personal
Prediction	p	pe	per	pers	perso	person	persona	personal

TABLE VII
PERFORMANCE COMPARISON OF THE IMPACT OF THE CKUNIT ON DIFFERENT MODULES. ✓ AND ✗ REPRESENTS THE MODULE WITH OR WITHOUT THE CKUNIT RESPECTIVELY. THE MODULES WITHOUT THE CKUNIT ARE COMPOSED OF STANDARD CONVOLUTIONAL AND POOLING LAYERS.

CKUnit		Datasets						
TORN	ABRN	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE
✗	✗	93.1	88.1	94.3	92.4	73.8	81.4	79.9
✓	✗	93.5	88.7	93.9	92.6	75.7	82.2	**81.6**
✗	✓	93.6	89.6	94.8	93.4	76.7	83.1	79.5
✓	✓	**94.3**	**90.7**	**95.3**	**93.6**	**77.7**	**85.0**	**81.6**

E. Limitation of our method

A comprehensive observation of Tables II, IV and VIII shows that the TRAN performs unsatisfactorily on the CUTE80 dataset, which consists of highly curved texts. Table VIII shows that the TORN fails to flatten the curved texts. We suggest that the reasons are as follows:

Our rectification module is based on STN and pixel-level methods without predicting any geometric attributes of texts. On the one hand, the STN rectifies entire texts with some transformations, such as rotation, translation and scaling. However, the deformations of the entire curved text are too complicated to be flattened by these transformations. On the other hand, the y-axis offsets of highly curved texts vary in a larger range than slightly irregular texts, which is more difficult to accurately regress[13]. In the future, we will attempt to address this issue.

TABLE VIII
VISUALIZATION OF SOME RECTIFIED IMAGES.

Input image	Rectified image	Ground truth Prediction
		villa villa
		close close
		overseas overseas
		jewelers jewelers
		esplanade esplanade
		epicentre epicentre
		supermodel supermodel
		personal personal
		chocolates chocolates
		manchester messageid
		athletic ithletic
		salmon antor

V. CONCLUSION

In this paper, we proposed a two-level rectification attention network for scene text recognition. The two-level rectification network combined geometry-level and pixel-level methods, which made up for their respective shortcomings. To improve the ability of our model to extract text features, we designed a channel-wise and kernel-wise attention unit to tackle the large variance of character size and the channel interdependencies. This architecture was adopted by both the rectification and recognition modules. In addition, we proposed a skip training strategy to make the model converge smoothly. The model was trained end-to-end, and required only text labels as supervision. Extensive experimental results on multiple benchmarks showed its effectiveness and rationality. Therefore, this paper conclusively proves that our method is of great value for scene text recognition.

ACKNOWLEDGMENT

This work was supported in part by the National Nature Science Foundation of China (Gant No. 61876051) and in part by the Shenzhen Key Laboratory of Visual Object Detection and Recognition (Grant No. ZDSYS20190902093015527).

REFERENCES

[1] T. Quy Phan, P. Shivakumara, S. Tian, and C. Lim Tan, "Recognizing text with perspective distortion in natural scenes," in *Proc. IEEE Int. Conf. Comput. Vis*, 2013, pp. 569–576.

[2] X. Liu, D. Liang, S. Yan, D. Chen, Y. Qiao, and J. Yan, "Fots: Fast oriented text spotting with a unified network," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 5676–5685.

[3] W. Liu, C. Chen, K.-Y. K. Wong, Z. Su, and J. Han, "Star-net: A spatial attention residue network for scene text recognition," in *Proc. British Mach. Vis. Conf.*, vol. 2, 2016, p. 7.

[4] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "Aon: Towards arbitrarily-oriented text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 5571–5579.

[5] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 8610–8617.

[6] M. Jaderberg, K. Simonyan, A. Zisserman, and K. Kavukcuoglu, "Spatial transformer networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 2017–2025.

[7] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, 2018.

[8] W. Liu, C. Chen, and K.-Y. K. Wong, "Char-net: A character-aware neural network for distorted scene text recognition." in *Proc. AAAI Conf. Artif. Intell.*, 2018, p. 7154–7162.

[9] F. Zhan and S. Lu, "Esir: End-to-end scene text recognition via iterative image rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 2059–2068.

[10] 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, 2016.

[11] C. Luo, L. Jin, and Z. Sun, "Moran: A multi-object rectified attention network for scene text recognition," *Pattern Recog.*, vol. 90, pp. 109–118, 2019.

[12] F. L. Bookstein, "Principal warps: Thin-plate splines and the decomposition of deformations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 6, pp. 567–585, 1989.

[13] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 770–779.

[14] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 7132–7141.

[15] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2014, pp. 4042–4049.

[16] O. Alsharif and J. Pineau, "End-to-end text recognition with hybrid HMM maxout models," in *Proc. Int. Conf. Learn. Representations*, 2014.

[17] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2010, pp. 2963–2970.

[18] C. Yao, X. Bai, and W. Liu, "A unified framework for multioriented text detection and recognition," *IEEE Trans. Image Process.*, vol. 23, no. 11, pp. 4737–4749, 2014.

[19] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu, "Detecting texts of arbitrary orientations in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.* IEEE, 2012, pp. 1083–1090.

[20] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.

[21] L. Neumann and J. Matas, "A method for text localization and recognition in real-world images," in *Proc. Asian Conf. Comput. Vis.* Springer, 2010, pp. 770–783.

[22] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

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

[24] C. Huang, Z. Wu, J. Wen, Y. Xu, Q. Jiang, and Y. Wang, "Abnormal event detection using deep contrastive learning for intelligent video surveillance system," *IEEE Trans. Ind. Informatics*, vol. 14, no. 11, pp. 4724–4734, 2021.

[25] C. Tian, Y. Xu, W. Zuo, B. Zhang, L. Fei, and C.-W. Lin, "Coarse-to-fine cnn for image super-resolution," *IEEE Trans. Multimedia*, vol. 205, p. 106235, 2020.

[26] J. Wen, Z. Zhang, Z. Zhang, L. Fei, and M. Wang, "Generalized incomplete multiview clustering with flexible locality structure diffusion," *IEEE Trans. Cybern.*, vol. 51, no. 1, pp. 101–114, 2020.

[27] X. Yun, Y. Zhang, F. Yin, and C. Liu, "Instance gnn: A learning framework for joint symbol segmentation and recognition in online handwritten diagrams," *IEEE Trans. Multimedia*, pp. 1–1, 2021.

[28] H. Ren, W. Wang, and C. Liu, "Recognizing online handwritten chinese characters using rnns with new computing architectures," *Pattern Recog.*, vol. 93, pp. 179–192, 2019.

[29] C. Zhang, Q. Zhao, C. P. Chen, and W. Liu, "Deep compression of probabilistic graphical networks," *Pattern Recog.*, vol. 96, p. 106979, 2019.

[30] P. Dai, H. Zhang, and X. Cao, "Deep multi-scale context aware feature aggregation for curved scene text detection," *IEEE Trans. Multimedia*, vol. 22, pp. 1969–1984, 2021.

[31] X. Wu, Q. Chen, Y. Xiao, W. Li, X. Liu, and B. Hu, "Lcsegnet: An efficient semantic segmentation network for large-scale complex chinese character recognition," *IEEE Trans. Multimedia*, pp. 1–1, 2020.

[32] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *CoRR*, vol. abs/1406.2227, 2014.

[33] M. Jaderberg, A. Vedaldi, and A. Zisserman, "Deep features for text spotting," in *Proc. European Conf. Comput. Vis.* Springer, 2014, pp. 512–528.

[34] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *Int. J. Comput. Vis.*, vol. 129, no. 1, pp. 161–184, 2021.

[35] X. Chen, L. Jin, Y. Zhu, C. Luo, and T. Wang, "Text recognition in the wild: A survey," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 42:1–42:35, 2021.

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

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

[38] B. Su and S. Lu, "Accurate scene text recognition based on recurrent neural network," in *Proc. Asian Conf. Comput. Vis.* Springer, 2014, pp. 35–48.

[39] Y. Gao, Y. Chen, J. Wang, and H. Lu, "Reading scene text with attention convolutional sequence modeling," *Neurocomputing*, 2017.

[40] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, "Gtc: Guided training of ctc towards efficient and accurate scene text recognition." in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11 005–11 012.

[41] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis*, 2017, pp. 5076–5084.

[42] Y. Huang, Z. Sun, L. Jin, and C. Luo, "Epan: Effective parts attention network for scene text recognition," *Neurocomputing*, vol. 376, pp. 202–213, 2020.

[43] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "Seed: Semantics enhanced encoder-decoder framework for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 13 528–13 537.

[44] T. Wang, Y. Zhu, L. Jin, C. Luo, X. Chen, Y. Wu, Q. Wang, and M. Cai, "Decoupled attention network for text recognition." in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 12 216–12 224.

[45] Y. Gao, Y. Chen, J. Wang, and H. Lu, "Progressive rectification network for irregular text recognition," *Sci. China Inf. Sci.*, vol. 63, no. 2, p. 120101, 2020.

[46] C.-Y. Lee and S. Osindero, "Recursive recurrent nets with attention modeling for ocr in the wild," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2231–2239.

[47] C. Wang, F. Yin, and C. Liu, "Memory-augmented attention model for scene text recognition," in *Int. Conf. Frontiers Handwriting Recog.*, 2018, pp. 62–67.

[48] C. Luo, Q. Lin, Y. Liu, L. Jin, and C. Shen, "Separating content from style using adversarial learning for recognizing text in the wild," *Int. J. Comput. Vis.*, vol. 129, no. 4, pp. 960–976, 2021.

[49] D. Yu, X. Li, C. Zhang, T. Liu, J. Han, J. Liu, and E. Ding, "Towards accurate scene text recognition with semantic reasoning networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 12 110–12 119.

[50] X. Yue, Z. Kuang, C. Lin, H. Sun, and W. Zhang, "Robustscanner: Dynamically enhancing positional clues for robust text recognition," in *Proc. European Conf. Comput. Vis.* Springer, 2020, pp. 135–151.

[51] Z. Wan, J. Zhang, L. Zhang, J. Luo, and C. Yao, "On vocabulary reliance in scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2020, pp. 11 425–11 434.

[52] 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 Recog.*, 2016, pp. 4168–4176.

[53] C. Bartz, H. Yang, and C. Meinel, "See: towards semi-supervised end-to-end scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 6674–6681.

[54] ——, "Stn-ocr: A single neural network for text detection and text recognition," *CoRR*, vol. abs/1707.08831, 2017.

[55] Y. Sun, C. Zhang, Z. Huang, J. Liu, J. Han, and E. Ding, "Textnet: Irregular text reading from images with an end-to-end trainable network," in *Proc. Asian Conf. Comput. Vis.* Springer, 2018, pp. 83–99.

[56] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *Int. J. Comput. Vision*, pp. 1–24, 2020.

[57] X. Li, W. Wang, X. Hu, and J. Yang, "Selective kernel networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 510–519.

[58] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using rnn encoder-decoder for statistical machine translation," in *Proc. Empirical Methods Natural Lang. Process.*, 2014, pp. 1724–1734.

[59] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2016, pp. 2315–2324.

[60] A. Mishra, K. Alahari, and C. V. Jawahar, "Scene text recognition using higher order language priors," in *Proc. British Mach. Vis. Conf.*, 2012, pp. 1–11.

[61] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *Proc. IEEE Int. Conf. Comput. Vis*, 2011, pp. 1457–1464.

[62] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young, "Icdar 2003 robust reading competitions," in *Proc. Int. Conf. Document Anal. Recognit.*, 2003, pp. 682–687.

[63] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "Icdar 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.* IEEE, 2013, pp. 1484–1493.

[64] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *Proc. Int. Conf. Document Anal. Recognit.* IEEE, 2015, pp. 1156–1160.

[65] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8027–8048, 2014.

[66] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.

[67] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep structured output learning for unconstrained text recognition," in *Proc. Int. Conf. Learn. Representations*, 2015.

[68] ——, "Reading text in the wild with convolutional neural networks," *Int. J. Comput. Vision*, vol. 116, no. 1, pp. 1–20, 2016.

[69] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit probability for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2018, pp. 1508–1516.

[70] Z. Liu, Y. Li, F. Ren, W. L. Goh, and H. Yu, "Squeezedtext: A real-time scene text recognition by binary convolutional encoder-decoder network," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7194–7201.

[71] Y. Liu, Z. Wang, H. Jin, and I. Wassell, "Synthetically supervised feature learning for scene text recognition," in *Proc. European Conf. Comput. Vis.*, 2018, pp. 435–451.

[72] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, C. Yao, and X. Bai, "Scene text recognition from two-dimensional perspective," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, 2019, pp. 8714–8721.

[73] Z. Xie, Y. Huang, Y. Zhu, L. Jin, Y. Liu, and L. Xie, "Aggregation cross-entropy for sequence recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2019, pp. 6538–6547.

[74] M. Zhang, M. Ma, and P. Wang, "Hierarchical refined attention for scene text recognition," in *Int. Conf. Acoustics, Speech and Signal Process.*, 2021, pp. 4175–4179.

[75] H. Gao, Y. Li, J. Dai, X. Wang, J. Han, and R. Li, "Multi-granularity deep local representations for irregular scene text recognition," *IEEE Trans. Data Sci.*, vol. 2, pp. 15:1–15:18, 2021.

[76] A. Rowel, "Vision transformer for fast and efficient scene text recognition," *CoRR.*, vol. abs/2105.08582, 2021.

Lintai Wu is currently pursuing the Ph.D degree in the School of Computer Science and Technology at Harbin Institute of Technology, Shenzhen. He received his B.S. degree in Management Information System at Wuhan University of Technology in 2017. He received his M.S. degree in Computer Science and Technology at Harbin Institute of Technology, Shenzhen in 2020. His research interests include, scene text recognition, pattern recognition and deep learning.

Yong Xu (Senior Member, IEEE) received his B.S. degree, M.S. degree in 1994 and 1997, respectively. He received the Ph.D. degree in Pattern Recognition and Intelligence system at NUST (China) in 2005. Now he works at Harbin Institute of Technology, Shenzhen. His current interests include pattern recognition, deep learning, biometrics, machine learning and video analysis. He has published over 100 papers in toptier academic journals and conferences. He has served as an Co-Editors-in-Chief of the International Journal of Image and Graphics, an Associate Editor of the CAAI Transactions on Intelligence Technology, an editor of the Pattern Recognition and Artificial Intelligence. More information please refer to http://www.yongxu.org/lunwen.html.

Junhui Hou (Senior Member, IEEE) received the B.Eng. degree in information engineering (Talented Students Program) from the South China University of Technology, Guangzhou, China, in 2009, the M.Eng. degree in signal and information processing from Northwestern Polytechnical University, Xian, China, in 2012, and the Ph.D. degree in electrical and electronic engineering from the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, in 2016. He immediately joined the Department of Computer Science, City University of Hong Kong, as an Assistant Professor in Jan. 2017. His research interests fall into the general areas of visual computing, such as image/video/3D geometry data representation, processing and analysis, semi/un-supervised data modeling, and data compression.

Dr. Hou was the recipient of several prestigious awards, including the Chinese Government Award for Outstanding Students Study Abroad from China Scholarship Council in 2015 and the Early Career Award (3/381) from the Hong Kong Research Grants Council in 2018. He is an elected member of MSA-TC and VSPC-TC, IEEE CAS. He is currently an Associate Editor for IEEE Transactions on Image Processing, IEEE Transactions on Circuits and Systems for Video Technology, Signal Processing: Image Communication, and The Visual Computer. He also served as the Guest Editor for the IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing and an Area Chair of ACM MM'19/20/21, IEEE ICME'20, VCIP'20/21, and WACV'21.

C.L. Philip Chen (Fellow, IEEE) received the M.S. degree from the University of Michigan, Ann Arbor, MI, USA, in 1985, and the Ph.D. degree from Purdue University, West Lafayette, IN, USA, in 1988, all in electrical engineering. He is currently the Chair Professor and the Dean of the College of Computer Science and Engineering, South China University of Technology, Guangzhou, China. His current research interests include systems, cybernetics, and computational intelligence. Dr. Chen is a Fellow of the American Association for the Advancement of Science (AAAS), the International Association of Pattern Recognition (IAPR), the Chinese Association of Automation (CAA), and the Chinese Association of Automation (HKIE) and a member of the Academia Europaea (AE), the European Academy of Sciences and Arts (EASA), and the International Academy of Systems and Cybernetics Science (IASCYS). He received the IEEE Norbert Wiener Award in 2018 for his contribution in systems and cybernetics, and machine learning. He is the Editor-in-Chief of the IEEE TRANSACTION ON CYBERNETICS and an associate editor of several IEEE TRANSACTIONS. He is also a 2019 Highly Cited Researcher in both computer science and engineering by Clarivate Analytics.

Chenglin Liu Cheng-Lin Liu (Fellow, IEEE) received the B.S. degree in electronic engineering from Wuhan University, Wuhan, China, in 1989, the M.E. degree in electronic engineering from Beijing Polytechnic University (currently Beijing University of Technology), Beijing, China, in 1992, and the Ph.D. degree in pattern recognition and intelligent control from the Institute of Automation, Chinese Academy of Sciences, Beijing, in 1995. He was a Postdoctoral Fellow with the Korea Advanced Institute of Science and Technology (KAIST), Daejeon, South Korea, and later with the Tokyo University of Agriculture and Technology, Fuchu, Japan, from March 1996 to March 1999. From 1999 to 2004, he was a Research Staff Member and later a Senior Researcher with the Central Research Laboratory, Hitachi, Ltd., Tokyo, Japan. Since 2005, he has been a Professor with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, where he is currently the Director. He has contributed many effective methods to different aspects of handwritten document analysis, including image preprocessing, page segmentation, feature extraction, classifier design, and character string recognition. His algorithms have yielded superior performance and have been transferred to industrial applications, including mail sorting, form processing, and web document retrieval. He has published over 300 technical papers in journals and conferences. His research interests include pattern recognition, image processing, neural networks, machine learning, and especially the applications to character recognition and document analysis. Dr. Liu is a Fellow of the Chinese Association for Artificial Intelligence (CAAI) and the International Association for Pattern Recognition (IAPR). He received the IAPR/ICDAR Young Investigator Award of 2005 and the Outstanding Youth Fund of NSFC in 2008. He is also an Associate Editor-in-Chief of Pattern Recognition and an Associate Editor of Image and Vision Computing, the International Journal on Document Analysis and Recognition, and Cognitive Computation.