Character-Aware Sampling and Rectification for Scene Text Recognition

Ming Li, Bin Fu, Zhengfu Zhang, and Yu Qiao, *Senior Member, IEEE*

***Abstract*—Curved scene text recognition is a challenging task in multimedia society due to large shape and texture variance. Previous methods address this challenge by extracting and rectifying text line with equidistantly sampling, which ignore character level information and lead to distorted characters. To address this issue, this paper proposes a Character-Aware Sampling and Rectification (CASR) module, which rectifies irregular text instance according to the location and orientation information of each individual character. Specifically, CASR regards each character as a basic unit and predicts the character-level attributes for sampling and rectification. Our module not only exploits detailed character information to obtain better rectification of text line, but also employs character-level supervision in training process. In addition, CASR provides a plug-and-play module which can be easily incorporated to existing text recognition pipeline. Extensive experiments on several benchmarks demonstrate that our method obtains more accurate rectified text instances and achieves promising performance. We will release our code and models in the future.**

***Index Terms*—Scene text recognition, scene optical character recognition, deep learning.**

I. Introduction

READING text in the wild is a fundamental and challenging task for multimedia society, which aims at translating images of text instance into a sequence of machine-readable symbols [1]. As an important image analysis technique, this task has been widely used in various real-world applications such as autonomous driving, human computer interaction and visual auxiliaries. In recent years, scene text recognition has witnessed a significant improvement due to the success of deep learning technology. The standard philosophy of these recognition models is to employ a encoding network [2] to extract visual context information and then employ a decoding model [3] to translate feature vectors into the target sequence. The decoding module is mainly based on one-dimensional sequence-to-sequence model [3], [4] which has an inherent limitation to address irregular text instances, especially for perspective texts and curved texts, which brings new challenge for converting text images to texts symbols.

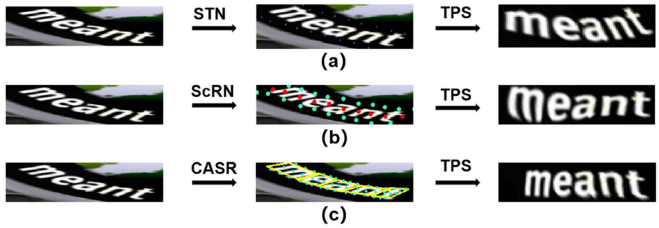

Fig. 1. The rectification pipelines of (a) ASTER [5], (b) ScRN [8] and (c) CASR (ours).

To address irregular text recognition issue, several methods [5]–[10] have been put forward in recent years which can be roughly divided into two categories: two-dimensional (2D) attention-based approaches and rectification-based approaches. For the 2D attention mechanisms [9], 2D attention maps are generated for feature alignment and sequence decoding. However, the recognition performance is limited due to the inaccurate attention maps. For the rectification-based methods, specific sampling mechanisms are designed to generate control points for rectifying irregular text instances to the canonical ones by thin-plate spline (TPS) [11]. For example, ASTER [5] employs a spatial transform network (STN) [12] to directly predict the control points on boundaries of text instance in a weakly supervised manner and ScRN [8] extends this work by adding an extra supervision in the rectification module. The rectification pipelines of ASTER and ScRN are shown in Fig. 1(a) and (b), respectively.

Although several local attributes have been employed to obtain accurate text lines, the text-level equidistantly sampling process largely ignores the character level information and might lead to distorted characters. Since characters are basic units of text instances, the essential for text recognition is to translate every individual character correctly. For perspective and curved texts, the recognition performance highly depends on whether individual characters are accurately rectified into canonical form. Moreover, since TPS strictly transforms the selected

Manuscript received 11 April 2021; revised 28 September 2021; accepted 4 November 2021. Date of publication 22 November 2021; date of current version 7 February 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 6163302, in part by Shenzhen Research Program (JSGG20191129141212311), and in part by the Shanghai Committee of Science, and Technology, China under Grant 20DZ1100800. The Associate Editor coordinating the review of this manuscript and approving it for publication was Dr. Palaiahnakote Shivakumara. *(Ming Li and Bin Fu contributed equally to this work) (Corresponding author: Yu Qiao.)*

Ming Li, Bin Fu, and Zhengfu Zhang are with the ShenZhen Key Lab of Computer Vision and Pattern Recognition, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China (e-mail: ming.li3@siat.ac.cn; bin.fu@siat.ac.cn; Zhang,zf.zhang@siat.ac.cn).

Yu Qiao is with the ShenZhen Key Lab of Computer Vision and Pattern Recognition, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 518055, China, and also with the Shanghai AI Laboratory, Shanghai 200030, China (e-mail: yu.qiao@siat.ac.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TMM.2021.3129651.

Digital Object Identifier 10.1109/TMM.2021.3129651

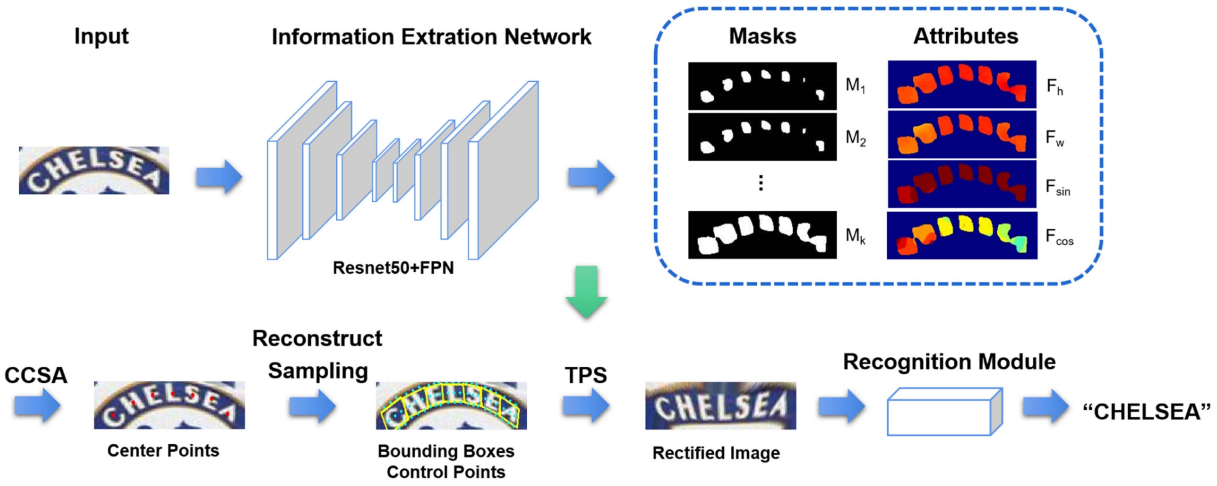

Fig. 2. The overall pipeline of our CASR-Net. The Information Extraction Network is Resnet50 with Feature Pyramid Network structure (the detailed structure is shown in Fig. 4). The output are segmentation maps of $k+4$ channels. M_1 to M_k represent the character-level masks of different scales, where M_1 is the smallest one and M_k is the largest one. k is the number of masks we used. F_h, F_w, F_{sin} and F_{cos} represents the segmentation map containing each character's height, width, sine and cosine value of character's angle. Connected Component Selecting Algorithm (CCSA) is proposed to obtain the center of each individual character by finding its connected component. Given the character information mentioned above, bounding boxes can be reconstructed and control points can be sampled. After the control points are generated, Thin Plate Spline (TPS) can be implemented to get the rectified image. Finally, the recognition module will recognize text on this image.

control points to the corresponding pre-defined positions, the rectification results of control points and their surroundings are more accurate. Based on above observations, we assume that rectifying irregular text instances based on character-level sampling strategy can obtain better result since it will incorporate more character-specific guidance in rectification process.

Motivated by above discussion, we propose a novel Character-Aware Sampling and Rectification (CASR) module which rectifies irregular text instance according to the location and orientation information of each individual character. Specifically, CASR regards each character as a basic unit and employs a segmentation-based network to predict character-specific geometrical attributes, including the mask, height, width and angle of each character. Since characters in text instance are pretty closed to each other, causing the challenge of separating, our module predict a set of character-level masks with different sizes to avoid the adherence. Then a Connected Components Selecting Algorithm (CCSA) is employed to obtain the center of each individual character across different masks. Based on these geometrical attributes, the character level text instance can be reconstructed and accurate control points can be generated for image rectification as shown in Fig. 1(c). Therefore, character-based sampling process can ensure each character to keep near-horizontal after rectification and then the recognition module can translate text instance into a sequence easily.

We incorporate our proposed CASR into commonly used attention-based recognition platform [3], [4] to construct a Character-Aware Sampling and Rectification Network (CASR-Net), the overall pipeline is shown in Fig. 2. Extensive experiments have been conducted to demonstrate the effectiveness of our CASR-Net on various public benchmarks. Our CASR-Net achieves promising performance on a number of text recognition datasets especially on irregular text datasets, such as ICDAR2015, SVTP and CUTE80. The main contributions of this paper are summarized as follows:

1. We notice that character-specific sampling is a reasonable and promising approach to obtain more accurate rectified text instances and propose a Character-Aware Sampling and Rectification (CASR) module to perform character-level rectification.

2. CASR is a plug-and-play module and we incorporate this module into attention-based recognition platform to construct a Character-Aware Sampling and Rectification Network (CASR-Net).

3. Experimental results verify the effectiveness of our CASR-Net,which achieves promising performance on a number of public datasets, such as SVT, ICDAR2013, ICDAR2015, SVTP and CUTE80 datasets.

II. Related Work

Scene text detection and recognition forms a sequential pipeline to localize text area and recognize text instances, which are vital for machines to read text from scene images. Detection models [13]–[15] are used to locate the text regions on the natural images while recognition models are further employed to translate text instances into sequences of symbols, both of which are important in this information translation process and this paper focuses on scene text recognition.

A. Scene Text Recognition

Reading text in natural scene is a challenging task and has widely received attention from industry and academia. Various methods have been put forward in recent years which can be categorized into two approaches: bottom-up-based methods and top-down-based methods. In this section we will give a brief

introduction of several state-of-the-art models and a comprehensive study for the development of scene text recognition has been given in [16].

Bottom-up approaches [17]–[20] firstly generate the character-level predictions and then connect each character into the corresponding sequences. Text information is extracted by hand crafted feature extraction module such as strokelet generation [21] and semi-markov conditional random field [22], then a classifier is employed to predict each character in text instance. Recently, several deep learning based methods significant improve the performance by replacing hand crafted feature extraction methods with neural networks such as [23], [24]. For example, LCSegNet [25] utilizes segmentation model to generate pixel-wise prediction for each characters and employs conditional random field to smooth label assignments, which achieves promising performance in several public benchmarks.

The top-down fashion employs another philosophy to recognize text instance which directly reading entire text instances without any predictions of individual characters. Inspired by image classification task, Jaderberg *et al.* [26] design a classification network with 90 k categories to recognize 90 k words. However, this method cannot be widely used due to the out-of-vocabulary words and the huge categories in classification network. To reading text instance with arbitrary length, the sequence methods have been put forward which can be roughly divided into two categories, Connectionist Temporal Classification (CTC) based and Attention-based methods. The CTC-based approaches usually employ a deep network to encode visual context and sequence information, and then employ CTC [27] to obtain conditional probability for arbitrary-length text, such as [28], [29]. In recent years, attention mechanism has been widely used in recognition models which generates a focusing map for each character position in text regions to improve recognition performance, such as [30], [31].

B. Recognition of Irregular Text Instance

Due to the success of CTC-based and attention-based models, reading text in regular and near-horizontal text has achieved acceptable performance and has been widely used in various real-world applications. However, due to inherent limitation of sequence model, the recognition for irregular text instance is still a challenging task, especially for perspective text and curved text. Recently, several methods have been put forward and they can be divided into two different categories. The first approaches generalize one-dimensional (1D) sequence model into two-dimensional (2D) version by employing 2D attention model to align features and decode the corresponding sequence, such as [6], [9], [32]. For this approach, the recognition performance is struggled with the inaccuracy attention map generated by deep network, and thus needs to be further improved. The second approaches [2], [5], [7], [8], [33] employs a rectification module to convert irregular text instances to regular and near-horizontal ones. ASTER [5] employs a spatial transform network (STN) [12] to directly predict the control points on the input image. This unsupervised method fails to predict the control points precisely enough due to the lack of guidance. ScRN [8] extends ASTER by adding an extra supervision in the rectification module. It predicts the text center line (TCL) with several geometrical attributes and generates the text-level control points by equidistantly sampling on TCL. However, this approach ignores the character level information which can provide more accurate information for TPS transformation. [34] extends ASTER by proposing a progressive rectification network to recognize irregular text instances in an iterative manner. The main differences of [34] and our model are twofold: 1. [34] is a multi-step method, it will estimate the transformation parameters at each step to boost recognition performance while our method is a single-step method. 2. Our method introduces geometric supervision for each character to predict more accurate character information and then generate transformation parameters based on each individual character.

Motivated that characters are the basic unit of text instances, we propose a novel sampling module which perform character-level rectification on irregular text instances.

III. Method

A detailed description of our proposed method will be given in this section. As shown in Fig. 2, our CASR-Net employs a novel character-level sampling and rectification module to rectify irregular text instances. Firstly a segmentation-based deep network is utilized to predict a set of character-specific attributes to reconstruct each individual character. Secondly, the rectification is performed to obtain canonical text instances based on character-aware information. Finally, the rectified instances are recognized by a commonly used recognition network.

A. Image Rectification Module

As discussed in previous section, since characters are basic units of text instances, the recognition performance will be greatly improved by rectifying each individual character accurately. Thus we firstly discuss the needed geometric attributes of each individual character in our model. After carefully analysing the text instance in existing irregular text recognition datasets, we find the individual character can be well represented by parallelogram bounding box. Therefore, we select the character center, angle, height and width to construct parallelogram bounding box. In this paper, for each individual character j, we define character center C_j as the central point of the given bounding box. The angle of character is defined as the angle between the character direction and the horizon, which is presented by its sine value $x_{j,sin}$ and cosine value $x_{j,cos}$. The height $x_{j,h}$ and width $x_{j,w}$ of character are obtained by calculating the average distance between corresponding points.

In this section, we propose a Character-Aware Sampling and Rectification module (CASR) to incorporate character-specific guidance in rectification process, which includes a deep network to extract character-level information and a character-wise constructing method to sample control points.

1) Character-Level Information Extraction: In our model, obtaining accurate character-specific information is fundamental, thus a segmentation-based deep network is employed to predict a set of geometric attributes, including the masks, height,

Fig. 3. The visualization of segmentation masks. (a) is the input image and (b)(c)(d) are masks of different character sizes. Specifically, (b) represents masks with same size as original character. (c) represents masks of smallest size. (d) represents masks of a medium size.

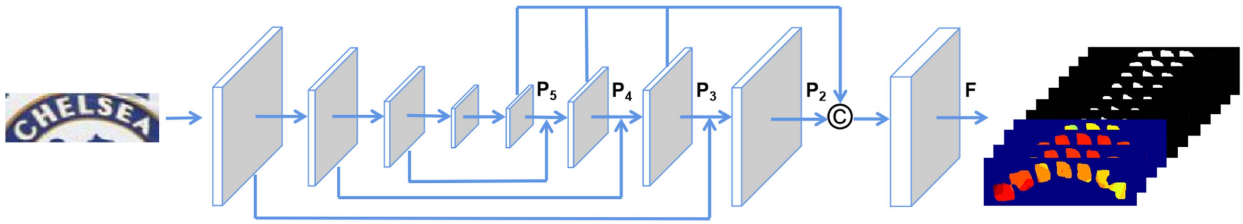

Fig. 4. The detailed backbone of our feature extractor network. The backbone is Resnet50 modified by FPN [35]. The symbol C is denotes concatenate operation. P_2, P_3, P_4 and P_5 are feature maps gained in different stages of FPN. F is the feature map fused by P_2, P_3, P_4 and P_5. The outputs are segmentation maps of the same resolution as the input image with $k+4$ channels, including k multi-scale character masks and 4 feature maps contained geometry information (F_h, F_w, $F_{\sin}$, $F_{\cos}$).

width and angle of each individual character. Among above attributes, generating accurate character/non-character mask is the most challenging due to the adhesion and missing problems of generated masks. Since characters in text instance are closed to each other, the corresponding masks would adhere to each other, causing the difficulty of character separation as shown in Fig. 3(b). A feasible solution is to expand boundaries between adjacent characters by shrinking the character-level masks. However, compared with larger mask, employing smaller ones means only high-probability regions would be predicted as character regions. As a consequence, character regions with relatively low probability will be ignored, causing the missing of masks (eg: the mask of first and last letter are not predicted in Fig. 3(c)). Therefore, to solve adhesion and missing problems, we regress a set of segmentation masks with different character sizes.

As shown in Fig. 4, we modify Feature Pyramid Network (FPN) [35] as our backbone to extract character-level information. A simple yet efficient segmentation head is utilized to perform dense prediction for multi-scale masks and geometric attributes. Specifically, a 3×3 convolution is employed to reduce the number of channels from 1024 to 256 after fusing feature maps P_2, P_3, P_4 and P_5. A 1×1 convolution is employed to generate the final segmentation maps of the same resolution as input image with $k+4$ channels. The k channels are multi-size character-level segmentation masks (noted as M_1 to M_k) representing the position of characters, while the four extra channels are utilized to perform dense prediction for segmentation map of height F_h, width F_w and angle ($F_{\sin}$ and $F_{\cos}$). Specifically, F_h, F_w, $F_{\sin}$ and $F_{\cos}$ represents segmentation map containing these geometry information, while $x_{j,h}$, $x_{j,w}$, $x_{j,sin}$ and $x_{j,cos}$ represents the specific geometry value of jth character.

To obtain accurate character-level masks, a Connected Component Selecting Algorithm (CCSA) is developed by selecting connected component of each character across different masks. CCSA searches the connected components from the smallest mask M_1 to the largest mask M_l and records each connected component(CC) that is non-overlapping with others in previous mask. l is the hyper-parameter represents how many masks are utilized in the algorithm which can be set from 1 to k.

As shown in Fig. 5, the top images of (b), (c) and (d) represent the minimal mask (M_1), 3 rd (M_3) and 5th mask (M_5), respectively. In (b), we only employ the minimal mask to detect characters, and thus miss the first character "U" and the last one "Y". The final rectified image is thus distorted. Then in (c) and (d), we implement CCSA with the hyper-parameter $l=3$ and $l=5$ respectively, and thus we can recover those two missing characters. The details of this algorithm are summarized in Algorithm 1.

With the accurate connected component of jth character SC_j, the center point can be represented by the circumcenter of SC_j and the character-wise geometrical attributes can be calculated from

$$x_{j,\theta} = \frac{\sum_{x,y}(SC_{j,x,y} * F_{\theta,x,y})}{\sum_{x,y}(SC_{j,x,y})} \tag{1}$$

where θ denotes character level information including the height, width and angle (sin and cos). $SC_{j,x,y}$ and $F_{\theta,x,y}$ denote to the value of pixel (x,y) in selected connected component mask SC_j and segmentation result F_θ.

Since the character-level annotations are provided in Synth-Text (ST) dataset [36], the geometric attributes can be calculated by character-level labels. According to the provided bounding boxes and geometric values, we generate the ground truth of the height G_h, width G_w, angle ($G_{\sin}$ and $G_{\cos}$) by filling the corresponding values to the character region. Moreover, we shrink the bounding box with several different predefined sizes as the ground truth of character/non-character masks noted as G_1 to

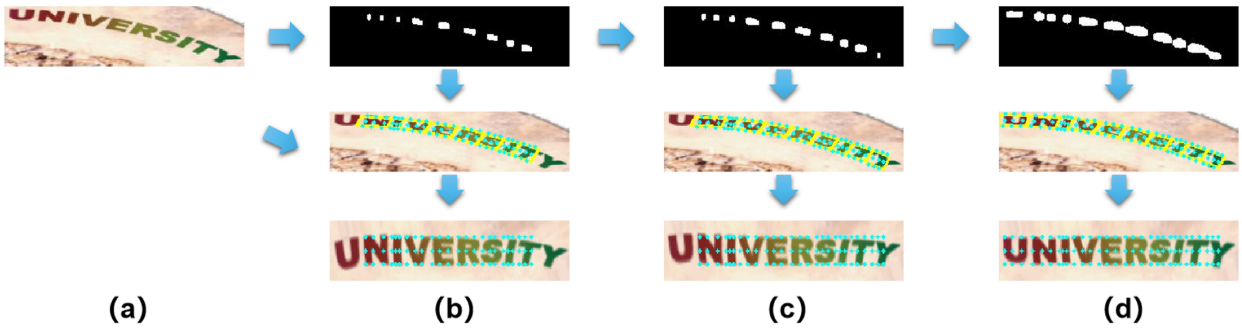

Fig. 5. Visualization of CCSA. (a) represents the input image of our CASR. The first row of (b)(c)(d) represents the segmentation mask of different layer, which represents the minimal (M_1), third (M_3) and fifth masks (M_5), respectively. The second row represents the reconstruct and sampling result printed in original image. The yellow line is the reconstructed bounding box and the blue points are control points. Third row shows the rectification results.

Algorithm 1: Connected Components Selecting Algorithm.

Require: $Masks : M$
Ensure: $Selected\ Connected\ Components : SC$
  find $\{CC_{1,1}, CC_{1,2}, \ldots, CC_{1,n}\}$ in M_1
  Enqueue$(SC, \{CC_{1,1}, CC_{1,2}, \ldots, CC_{1,n}\})$
  for each $M_i (i > 1)$ in M **do**
    find $\{CC_{i,1}, CC_{i,2}, \ldots, CC_{i,n}\}$ in M_i
    for each $CC_{i,j}$ in M_i **do**
      if sum$(CC_{i,j} * M_{i-1}) = 0$ **then**
        Enqueue$(SC, CC_{i,j})$
      end if
    end for
  end for
  return SC

G_k. With above annotations, the loss function can be formulated as

$$L = L_{mask} + L_{geo}, \tag{2}$$

where L_{mask} and L_{geo} represent the loss functions of scaled masks and geometry attributes, respectively.

For the loss function L_{mask}, since the background (non-character) pixels are dominated in image, especially for images with small characters, the imbalance of training samples will become a serious problem in training process. The Dice coefficient $D(M_i, G_i)$ [37] is employed to release this issue which can be expressed as

$$D(M_i, G_i) = \frac{2 * \sum_{x,y}(M_{i,x,y} * G_{i,x,y})}{\sum_{x,y}(M^2_{i,x,y}) + \sum_{x,y}(G^2_{i,x,y})} \tag{3}$$

where $M_{i,x,y}$ and $G_{i,x,y}$ represent the value of pixel (x, y) in ith mask and the corresponding ground truth, respectively. Moreover, we implement Online Hard Example Mining (OHEM) [38] to further boost the performance of mask prediction. Therefore, the final loss function for mask prediction can be expressed as

$$L_{mask} = \sum_{i=1}^{k}(1 - D(M_i \times O, G_i \times O)), \tag{4}$$

where k represents the number of scaled masks we used and the O represents the training mask given by OHEM.

The smooth L_1 loss is implemented to obtain accurate predictions for geometrical attributes in character regions, which can be calculated by

$$L_{geo} = \sum_{\theta} \mathrm{Smooth}L_1(G_\theta, M_{max} \times F_\theta), \tag{5}$$

where M_{max} represents the largest mask and θ denotes character level information including the height, width and angle (sin and cos).

2) Character Reconstruct and Sampling: Inevitably, the segmentation head might predict some non-character regions as positive due to the complexity of background. As shown in Fig. 6(b), irrelevant connected components are predicted because of the extra small text line in the background. Fig. 6(c) contains all the center points generated after CCSA and it is impossible to reconstruct characters based on these chaotic points. Thus a quadratic function is employed to fit all the points and greedily eliminate fluctuated points when the residual error larger than the predefined threshold. Specifically, after gaining the chaotic point set, a quadratic curved is employed to fit all the points and a residual value is generated. If this value is greater than predefined threshold, the point that cause the largest increase of residual value will be eliminated until the overall residual value is small. The resulted center points is presented on Fig. 6(d). After above refinement, the reconstruction result in Fig. 6(e) and rectification result in Fig. 6(f) both satisfactory.

The text instances then can be reconstructed based on the predicted geometrical attributes. For curved and perspective text instances, rectangular bounding boxes are not suitable, thus we reconstruct each individual character with a general parallelogram. The detailed process is shown in Fig. 7, where C $_x$ is the center point of x th character. We firstly find the demarcation point D^1 on the line segment C_2C_3 where $C_2D^1/C_2C_3 = W2/(W2 + W3)$. $(W2, W3$ represent the predicted widths of character $C2$ and $C3)$. This step further regularizes the predicted width of each character. Then draw a line segment pass through this point D^1 with the provided angle and height to form a bounding box boundary A^1A^4 in Fig. 7(a). The segment A^2A^3 is generated by the same process. Finally we connect A^1A^2 and A^3A^4 to form the parallelogram bounding box shown in Fig. 7(b).

After construction, the thin-plate spline (TPS) transformation is employed to rectify text instance, which is determined by

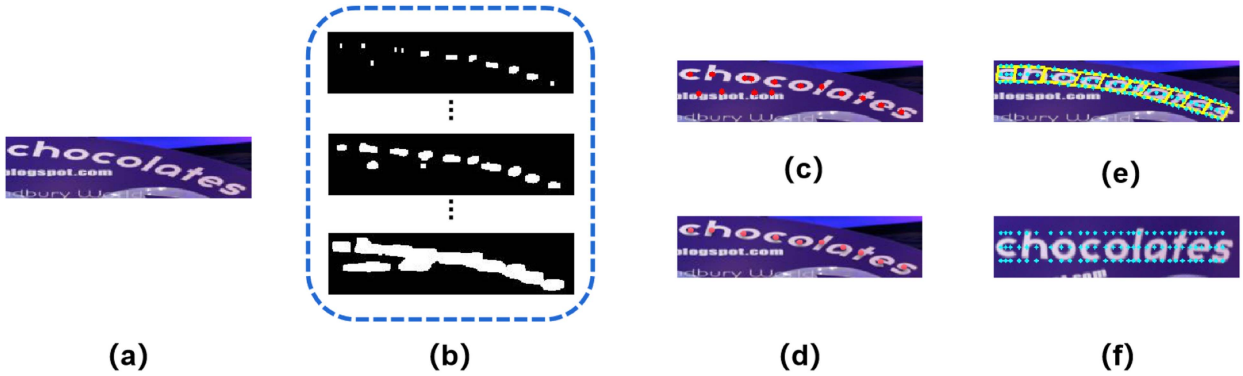

Fig. 6. Effect of Refinement. (a) Represents the input image. (b) Represents the set of masks we predict. (c) Shows the central points we can get from our network. (d) Visualize the center of characters after refinement. (e) Shows how we sample and (f) is our rectification result.

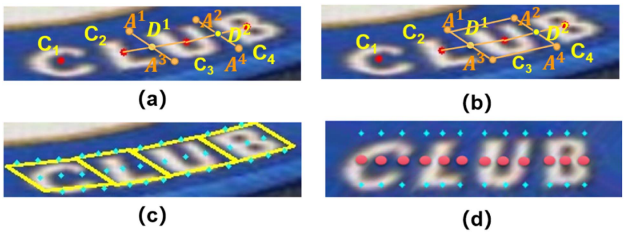

Fig. 7. C $_j$ is the center point of j th character. (a) and (b) represent process of reconstructing the character "U". D^1 and D^2 are demarcation points of line segment C_2C_3 and C_3C_4. A^1, A^1, A^1 and A^1 are corner points of the bounding box. The yellow line in (c) is the constructed bounding box and the blue points are control points in set P, which is the control point before TPS. (d) is the rectification result and the points P' are corresponding control points after TPS.

a pair of points sets $P = \{p_1, p_2, \ldots, p_{9n}\}$ (n represents the number of character) and P', where P and P' denote the control points before and after TPS, respectively. With the parallelogram bounding boxes, we sample 3 points each on the upper and lower bounding box boundary together with 3 points on the character's center line shown in Fig. 7(c). Therefore, totally 9 control points for each character are generated as P.

Similar to P, $P' = \{p'_1, p'_2, \ldots, p'_{9n}\}$ are generated by the same geometry attributes. We firstly determine the position of the characters' center points, whose y coordinates lie on the center line in rectified image and x coordinates is the same with the corresponding center points in original image. Based on these points, we generate the left, right points according to the width of each character and these 3 points of each character are represented by the red points in Fig. 7(d). Finally we move these points upward and downward according to the character's height to form all the corresponding points in P' shown in Fig. 7(d).

B. Text Recognition Module

To verify the effectiveness and portability of our CASR, we implemented both ASTER and DRNet as our recognition module. ASTER and DRNet are both 1D attention based recognizer, in which rectification module affects performance severely. We replace the original rectification modules of these two recognizers with our CASR and keep other configurations unchanged, the detailed network configurations are shown in Table I.

TABLE I
TEXT RECOGNITION NETWORK CONFIGURATIONS OF DRNET (TA+NC VERSION). "S" STANDS FOR STRIDE OF THE FIRST CONVOLUTION LAYER IN EACH STAGE. "OUT SIZE" IS FEATURE MAP SIZE OUTPUT FROM EACH STAGE (CHANNEL × HEIGHT × WIDTH). "-" MEANS NO THIS MODULE. "ATT. LSTM" STANDS FOR ATTENTIONAL LSTM DECODER [5]

	Stages	ASTER	DRNet
Encoder	Stage 0	3×3 conv, s 1×1	3×3 conv, s 1×1
	Stage 1	$\begin{bmatrix} 1\times1 \text{ conv}, \\ 3\times3 \text{ conv}, \end{bmatrix} \times 3$, s 2×2	$[\ 1 \times 1$ conv, 3×3 conv $] \times 1$, s 2×2 $\begin{bmatrix} 3\times3 \text{ TA}, \\ 16\times3 \text{ conv}, \quad BiLSTM \end{bmatrix} \times 2$
	Stage 2	$\begin{bmatrix} 1\times1 \text{ conv}, \\ 3\times3 \text{ conv}, \end{bmatrix} \times 4$, s 2×2	$[\ 1 \times 1$ conv, 3×3 conv $] \times 1$, s 2×2 $\begin{bmatrix} 3\times3 \text{ TA}, \\ 8\times3 \text{ conv}, \quad BiLSTM \end{bmatrix} \times 3$
	Stage 3	$\begin{bmatrix} 1\times1 \text{ conv}, \\ 3\times3 \text{ conv}, \end{bmatrix} \times 6$, s 2×2	$[\ 1 \times 1$ conv, 3×3 conv $] \times 1$, s 2×1 $\begin{bmatrix} 3\times3 \text{ TA}, \\ 4\times3 \text{ conv}, \quad BiLSTM \end{bmatrix} \times 5$
	Stage 4	$\begin{bmatrix} 1\times1 \text{ conv}, \\ 3\times3 \text{ conv}, \end{bmatrix} \times 6$, s 2×2	$[\ 1 \times 1$ conv, 3×3 conv $] \times 1$, s 2×1 $\begin{bmatrix} 3\times3 \text{ TA}, \\ 2\times3 \text{ conv}, \quad BiLSTM \end{bmatrix} \times 5$
	Stage 5	$\begin{bmatrix} 1\times1 \text{ conv}, \\ 3\times3 \text{ conv}, \end{bmatrix} \times 3$, s 2×2	$[\ 1 \times 1$ conv, 3×3 conv $] \times 1$, s 2×1 $\begin{bmatrix} 3\times3 \text{ TA}, \\ 1\times3 \text{ conv}, \quad BiLSTM \end{bmatrix} \times 2$
	Context	BiLSTM (256 hidden units)	-
Decoder	Att. LSTM	256 attention units 256 hidden units	256 attention units 256 hidden units

1) ASTER: An attention-based sequence-to-sequence recognition model [3]–[5] is employed to predict character sequence in rectified text instances. The rectified images are re-sized to 32×100 and then a deep residual network is employed to encode visual information into the corresponding feature sequence. In first two residual blocks, a stride 2×2 convolution is utilized to down-sample feature maps and enlarge receptive field. For the following blocks, the stride becomes 2×1 to reserve more local details in horizontal direction. A two-layer Bidirectional long short-term memory (BiLSTM) is employed to further encode feature maps into sequence. Finally, an attentional decoder is utilized to iteratively predict character symbol y_t at step t, resulting sequence $y = \{y_1, y_2, \ldots, y_T\}$, where T is the number of characters. The loss function can be formulated as

$$L = -\frac{1}{T}\sum_{t=1}^{T} \log\ p(y_t|I), \tag{6}$$

where I represents the input image.

2) DRNet: DRNet is another attention-based sequence-to-sequence recognition model with a similar structure. The normal

residual blocks in ASTER are replaced by DR-blocks which extract local visual and long-range contextual relation simultaneously from low level features. In each DR-block, a 3×3 convolution is employed to extract local features while a $h \times 3$ (h is the height of output feature in each stage) convolution followed by an Bidirectional LSTM is deployed to model long-range contextual relation. The overall stage configurations are kept the same as ASTER including the down sample blocks and stride in each stage.

IV. Experiment

A. Datasets

We train our model on two synthetic datasets, **Synth90 K** (Sy90) [39] and **SynthText**(ST) [36]. Specifically, the recognition module is trained on two datasets while the rectification module is only trained on SynthText dataset with the character-wise annotations. No extra data is used.

SynthText(ST) [36] is a synthetic text image dataset which is originally used for scene text detection. Not only text level labels, character level annotations are provided in this dataset as well. Scene text recognition inputs are generated by by cropping the text instance according to the given bounding boxes. Our CASR module is trained merely on this dataset for its character wise annotations.

Synth90 K(Sy90) [39] is a synthetic dataset containing 9 million word box images. It is generated by rendering the words to background with some hand-crafted disturbance and is widely used for the training text recognition model.

To demonstrate the effectiveness of our CASR-Net, seven scene text recognition datasets are employed to evaluate our method which can be devided into regular datasets or irregular datasets.

Regular datasets include:

IIIT5K-Words (IIIT) [40] includes 3000 testing images collected from Google searches.

Street View Text (SVT) [17] contains 647 outdoor street images collected by Google Street View.

ICDAR 2003 (IC03) [41] contains 860 images after filtering the those contain non-alphanumeric characters and few characters.

ICDAR 2013 (IC13) [42] contains 1015 scene text images mostly inherited from IC03.

Irregular datasets include:

ICDAR 2015 (IC15) [43] contains 2077 texts images captured by Google Glasses. Images in this dataset are blurry

SVT-Perspective (SVTP) [44] contains 645 testing images. Images here are severely distorted by non-frontal view angle.

CUTE80 [45] contains 288 word images which focus on curved texts.

B. Implement Details

The proposed method is implemented on Pytorch platform. For the image rectification module, the training samples are resized to 64×256 as the input images. The number of masks is a hyper-parameter and we set $k = 7$ in our method by following PSENet [46]. The PSENet utilizes 7 masks for each text center line to locate text instance, while we employ 7 masks for each character to separate each individual character. The $k = 7$ is the recommended configuration in PSENet and thus we follow this setting in our model. We train the module from scratch with batch size 512 for 6 epochs. The initial learning rate is set as $\mathrm{lr}_{\mathrm{initial}} = 0.1$ and will be divided by 10 at 2, 4 and 5 epochs. The loss of masks and geometry attributes are equally weighted and the negative-positive ratio of OHEM [38] is set to 3. The network is optimized by stochastic gradient descent (SGD) with the momentum of 0.99 and weight decay of 5×10^{-4}.

Recognition module is trained on ST [36] and Sy90 [39] datasets. Along with the original ST and Sy90, we rectify all text instances in two datasets by our CASR as our overall training data.

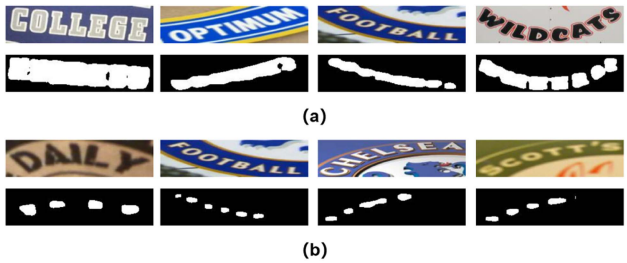

Fig. 8. Visualization on the problem of mask adhesion (a) and mask missing (b).

C. Visualization of Mask Adhesion and Mask Missing

To better understand the necessity of generating character masks with different scales, we present more visualization on images that suffer from the the mask adhesion and mask missing. As shown in the Fig. 8(a), the mask of each character adheres to each other and they form a connected mask of entire text line. When mask adhesion appears, the central point of each individual character can not be obtained, and thus it is impossible to reconstruct characters and to sample control points in the character-wise manner, which is the reason we generate small-scale masks. However, when the scale is small, character masks are easy to be lost. Fig. 8(b) shows the problem of mask missing, where we generate masks fewer than characters appeared in the image. In order to solve the above problem, we generate masks with different scales and propose the CCSA algorithm.

D. Ablation Study for Connected Components Selecting Algorithm

In this section, we perform a set of extensive experiments to discuss the effect of our CCSA in details. We generate character-level masks by selecting connected components with different number of layers. The experimental results are shown in Table II, where the hyper-parameter l represents the number of layers we use, which is set from 1 to k. The baseline is CASR-Net-l_1 which directly employs the minimal segmentation mask to get the center of each character. From the TABLE II, we

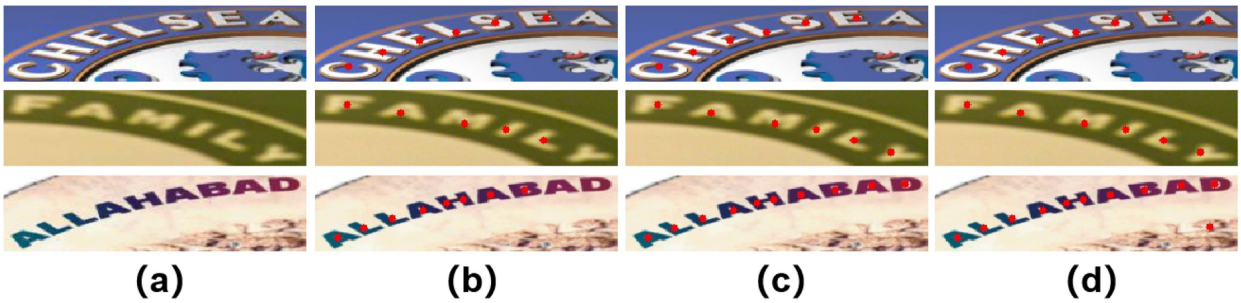

Fig. 9. Visualization results of CCSA. (a) represents the input images. (b)(c)(d) is the visualization of the center points obtained from CCSA with the the hyper-parameter l set as 1, 4 and 7, respectively. The red dots in the images are central points.

TABLE II
COMPARISON OF RECOGNITION ACCURACY BETWEEN MODELS USING CCSA WITH DIFFERENT NUMBERS OF LAYERS

Variants	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE80
CASR-Net-l_1	93.8	89.5	93.3	94.6	**79.5**	80.5	89.2
CASR-Net-l_2	93.9	90.0	92.9	94.3	79.3	80.8	87.5
CASR-Net-l_3	93.8	90.6	**93.7**	95.0	79.0	**82.5**	89.6
CASR-Net-l_4	93.9	**90.7**	93.6	94.6	79.1	82.2	89.2
CASR-Net-l_5	**94.0**	89.6	93.5	95.0	79.3	80.3	**90.3**
CASR-Net-l_6	93.7	90.1	93.4	**95.1**	**79.5**	80.3	89.9
CASR-Net-l_7	**94.0**	90.0	93.4	94.9	79.3	80.0	89.6

TABLE III
THE RECOGNITION ACCURACY OF DIFFERENT SAMPLE STRATEGIES ON CUTE80 DATASET. $ctl = k$ REPRESENTS THE NUMBER OF CONTROL POINTS IS k

Strategy	$ctl = 2$	$ctl = 3$	$ctl = 4$	$ctl = 5$	$ctl = 6$	$ctl = 7$	$ctl = 9$
Acc	87.5	88.2	88.6	89.2	89.2	89.6	89.6

Fig. 10. Visualization on the sampling strategies. The blue points in each text image is the control points. To better show the differences of control points, we use the same text image for comparison.

can draw two conclusions: (1). CCSA with multi-scale segmentation masks can significant improves recognition performance about 1.2%, 2.0% and 1.1% for SVT, SVTP and CUTE80, respectively. (2). The performance shows non-linear correlation with the value of l. The overall performance will reach the highest when $l = 3$, and a higher or lower value of l will lead to a drop of the average accuracy. This phenomenon is reasonable since CCSA gradually append the the character area from highest possibility to lower, and the medium value reaches a balance.

To better demonstrate the improvement, we visualize the results of CCSA in Fig. 9. As shown in Fig. 9(b), without CCSA, some characters cannot be predicted correctly, causing the rectification unsatisfactory. Fig. 9(c) shows that the missing characters will be recovered after implementing CCSA. However, including larger masks will lead to the wrong prediction of non-character region shown in the lowest image in Fig. 9(d) which will make the image difficult to be rectified.

E. Ablation Study for Number of Control Points

In this section, we conduct extensive experiments to study the recognition accuracy on curved texts with respect to the number of control points. Fig. 10 shows the visualization of different sampling strategies in this ablation study, where the blue points are control points. As shown in TABLE III, with the growth of control points, the recognition accuracy firstly grows accordingly, but it becomes steady when the number of control points reaches 7. Based on above observation, we think 9 control points in our method are sufficient.

F. Ablation Study for Re-Construction Strategies

Once we obtain the accurate character-specific attributes, text instances can be reconstructed by the corresponding character-level geometrical information. Normally, two different kinds of bounding boxes are widely implemented to construct characters, parallelogram-based bounding boxes (Para-bb) and rectangular-based bounding boxes (Rect-bb). For comparison, we visualize the reconstruction results of Para-bb and Rect-bb in Fig. 11. From Fig. 11, we can obtain the following conclusions: (1). As shown in Fig. 11(b), characters represented by Rect-bb is enough for regular text instance, and the rectification results of two approaches are almost the same. (2). For the extremely perspective-shifted or distorted images, reconstructing with Rect-bb will lead to a disaster as shown in (d) while Para-bb performs well in (c). Thus, in order to obtain more robust and satisfied results, we reconstruct irregular text instances by Para-bb.

G. Ablation Study for Character-Aware Sampling

Rectifying irregular text instance based on character-specific sampling method is an important contribution in this work. We

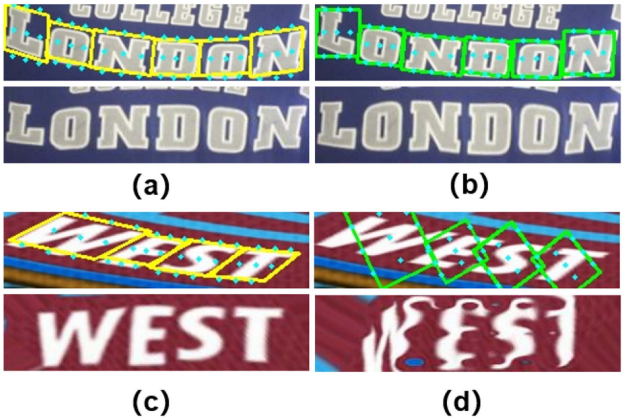

Fig. 11. Visualization of Rect-bb and Para-bb. The first row is the visualization of reconstruting and sampling and the second is the rectification result. (a)(c) are reconstructed in Para-bb and (b)(d) in Rect-bb. From this visualization, using rectangle bounding box might lead to sever distortion of image.

TABLE IV
THE COMPARISON OF RECOGNITION ACCURACY BETWEEN EQUIDISTANTLY SAMPLING AND CHARACTER-AWARE SAMPLING. IN CASR-EVENLY, THE NUMBER IN BRACKET DENOTES THE NUMBER OF CONTROL POINT PAIRS IN THIS EXPERIMENT. THE LAST COLUMN IS THE AVERAGE ACCURACY ON BENCHMARKS

Variants	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE80	Ave
CASR-para	**93.8**	**90.6**	**93.7**	**95.0**	**79.0**	**82.5**	**89.6**	**89.2**
CASR-evenly(5)	93.3	89.0	93.3	94.4	78.7	78.8	86.1	87.7
CASR-evenly(10)	93.5	89.3	93.4	94.3	78.9	79.4	88.9	88.2
CASR-evenly(15)	93.5	89.5	93.3	94.7	78.6	79.3	88.5	88.2
CASR-evenly(20)	93.3	89.3	93.0	94.4	78.1	79.4	86.5	87.7

conduct ablation experiments to verify the effectiveness of our character-aware sampling approach and the experimental results are show in TABLE IV. The number in bracket represents the pair number of control points being used. The last column is averaging acuracy on these seven benchmarks. Compared with the equidistantly sampling method, our method improves recognition performance on all of the seven benchmarks no matter how many pairs of evenly-sampled control points are used. This clearly demonstrates that character-based sampling approach is superior than equidistantly sampling approach since more character-aware information is incorporated.

H. Visualization of Rectification Results

In this section we visualize the rectification results of our CASR in Fig. 12. Compare with ASTER [5], our CASR can accurately rectify the text instances according to the character-aware geometrical attributes. For those perspective images, ASTER might bend the text lines slightly as shown in Fig. 12(a)(b) while ours can rectify them smoothly. For those heavily curved ones, ASTER is incapable of rectifying them as Fig. 12(c)(d). Moreover, it sometimes crops some useful regions and cause the recognition module unable to recognize as shown in Fig. 12(e)(f) while ours can transfer them into near-horizontal texts. Therefore, our model can obtain a better recognition performance based on the accurately rectified images. More visualization results will be presented in appendix.

TABLE V
THE COMPARISON OF RECOGNITION ACCURACY BETWEEN ASTER, CASR-ASTER, DRNET AND CASR-DRNET

Variants	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE80
ASTER	93.4	89.5	94.5	91.8	76.1	78.5	79.5
CASR-ASTER	93.8	90.6	93.7	95.0	79.0	82.5	89.6
DRNet	93.6	89.6	94	94.6	81.6	83.4	82.3
CASR-DRNet	95.1	92.3	94.9	95.3	83	85	89.2

I. Comparison With Baseline

Though our model improves the performance compared with ASTER in a large margin, the gap between ASTER and recent state-of-the-art models makes our improvement inconspicuous. Thus to demonstrate the effect and portability of our CASR, we implemented DRNet as our another recognizer along with ASTER. DRNet is an 1D attention based recognizer which needs a rectification module to normalize input images. It extracts local visual and long-range contextual information simultaneously from low level features and improves the recognizing performance. Similar to CASR-ASTER, we just replace the rectification module of DRNet with CASR and denote it as CASR-DRNet.

As shown in the TABLE V, our CASR is a plug-and-play module which can be used in any 1D text recognize models. Both CASR-ASTER and CASR-DRNet have steady improvement compared with baseline especially for the three irregular datasets. CASR improves ASTER and improves DRNet to a high performance.

J. Comparison With State of the Art

To demonstrate the effectiveness of our CASR, we compare our method with state-of-the-art models in this section. As shown in TABLE VI, our method achieves promising performance in all irregular datasets (IC15, SVTP and CUTE80) which demonstrate the superior performance of our CASR-Net for recognizing perspective and curved text instances. Compared with other rectified-based STR models, including ASTER, ESIR and ScRN, our model achieves the best performance. The improvement shows that rectifying irregular text instances based on character-aware sampling method can significantly improves recognition performance. Moreover, due to the fluctuation of character-level attribute prediction, the character-specific rectification module may introduce some noise, but our CASR-Net still achieves superior performance on regular datasets, especially for SVT dataset.

Finally, the inference time of overall model is 0.026 seconds per image, which is acceptable and can be used in real-time applications.

K. Synthetic Data for Testing

Although the CUTE80 [45] dataset have been widely used for evaluating the performance on curved texts, it lacks sever-curved texts. Therefore, to verify the recognition performance of our model on complex fluctuation texts, we followed [36] and generated more curved texts and visualized the rectification result in these extra-synthetic data as shown in Fig. 13. In each

Fig. 12. Visualization of rectification results. The first row shows the original input images. Second row shows the rectification result of ASTER. Third row presents our rectification results.

TABLE VI
RECOGNITION ACCURACY ACROSS A NUMBER OF METHODS AND DATASETS

Methods	IIIT5k	SVT	IC03	IC13	IC15	SVTP	CUTE80
Mishra *et al.* [18]	64.1	73.2	81.8	-	-	-	-
Almazan *et al.* [47]	91.2	89.2	-	-	-	-	-
Yao *et al.* [20]	80.2	75.9	88.5	-	-	-	-
Gordo [48]	93.3	91.8	-	-	-	-	-
Jaderberg *et al.* [26]	-	80.7	93.1	90.8	-	-	-
Jaderberg *et al.* [23]	-	71.7	89.6	81.8	-	-	-
Shi *et al.* [28]	81.2	82.7	91.9	89.6	-	-	-
Shi *et al.* [2]	81.9	81.9	90.1	88.6	-	71.8	59.2
Lee *et al.* [30]	78.4	80.7	88.7	90.0	-	-	-
Yang *et al.* [49]	-	-	-	-	-	75.8	69.3
Cheng *et al.* [31]	87.4	85.9	94.2	93.3	70.6	-	-
Cheng *et al.* [6]	87.0	82.8	91.5	-	68.2	73.0	76.8
Liu *et al.* [50]	92.0	85.5	92.0	91.1	74.2	78.9	-
Bai *et al.* [51]	88.3	87.5	94.6	94.4	73.9	-	-
Liu *et al.* [52]	87.0	-	93.1	92.9	-	-	-
Liu *et al.* [53]	89.4	87.1	94.7	94.0	-	73.9	62.5
Liao *et al.* [32]	91.9	86.4	-	91.5	-	-	79.9
Shi *et al.* [5] (ASTER)	93.4	89.5	94.5	91.8	76.1	78.5	79.5
Zhan *et al.* [7] (ESIR)	93.3	90.2	-	91.3	76.9	79.6	83.3
Yang *et al.* [8] (ScRN)	94.4	88.9	95.0	93.9	78.7	80.8	87.5
Wang *et al.* [9]	94.3	89.2	95.2	94.2	74.5	80.0	84.4
Wang *et al.* [54]	91.5	84.5	-	91.0	69.2	76.4	83.3
Yue *et al.*(RobustScanner) [55]	**95.3**	88.1	-	94.8	77.1	79.5	**90.3**
Qiao *et al.*(SEED) [56]	93.8	89.6	-	92.8	80.0	81.4	83.6
Yu *et al.*(SRN) [57]	94.8	91.5	-	**95.5**	82.7	**85.1**	87.8
Litman *et al.*(SCATTER) [58]	92.9	89.2	**96.5**	93.8	81.8	84.5	85.1
Zhang *et al.*(AutoSTR) [59]	94.7	90.9	93.3	94.2	81.8	81.7	-
Gao *et al.*(PRN) [34]	94.3	88.7	94.0	93.3	76.8	81.2	88.2
CASR-ASTER	93.8	90.6	93.7	95.0	79.0	82.5	89.6
CASR-DRNet	95.1	**92.3**	94.9	95.3	**83.0**	85.0	89.2

The highest accuracy is in bold and the second highest accuracy is underlined.

pair, the upper image is the original input, and the below is the rectified image.

The performance of our model on this dataset reaches 90.7% of recognition accuracy while the baseline model, Aster [5], reaches 67.1%. Our model outperforms the baseline about 23.6%, which demonstrates the effectiveness of our model.

The detailed synthetic process described as follows: We firstly analyze the texts and their frequencies appeared in the benchmarks, and then select the top 500 texts as our corpus. Given a background image, color segmentation method [60] is utilized to separate the image into several smooth areas. Then a text is randomly chosen from corpus and a random font is used to put

Fig. 13. Examples in our synthetic data. In each pair, the upper image is the original input, and the below is the rectified image.

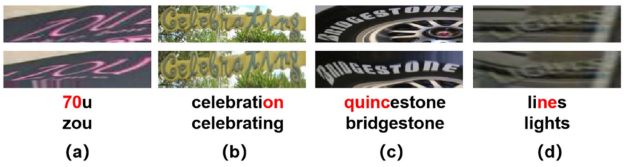

Fig. 14. Visualization of bad cases. From top to bottom, the rows represent original images, rectified images, the recognition results and the ground truth, respectively. The red character represents the false prediction.

the characters into the smooth area of image. Finally, the text is blended into the scene using Poisson image editing [61].

L. Comparison With Related Works

Since we employ segmentation model to predict the position of each character, mask adhesion is a serious problem and we utilize multi-scale masks to separate different characters. [62] deals with the similar problem utilizing text instance segmentation to locate irregular texts, which develops the concept of weighted text border to separate the adjacent text instances. This method [62] employs text center line to locate the position of each text and text border line to separate adjacent text instances. Since the short edges tend to be undetectable, the authors develop the weighted text border to generate a reliable border prediction. Compared with this work, our feature extraction module only employs multi-scale character masks to separate each individual character without the border of each character, which is simple and easy to implement than [62].

M. Limitations

Though achieving the best rectification results, we visualize some bad cases to analyse the limitation of our method. We classify bad cases into two categories shown in Fig. 14 and Fig. 15, the former is caused by the poor rectification while the latter is caused by the imperfect recognition module.

As shown in Fig. 14, due to the complex background and the blurry texts, our CASR is not able to generate precise geometrical attributes, which lead to the relatively bad rectification results.

In this paper, we find that most wrong predicted images belong to the second category as shown in Fig. 15. In this case, our CASR rectifies the text images to canonical ones but the

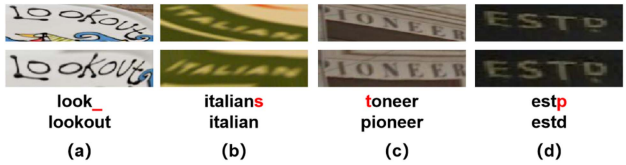

Fig. 15. Visualization of bad cases. From top to bottom, the rows represent original images, rectified images, the recognition results and the ground truth, respectively. The red character represents the false prediction and the underline represents the missing of character.

recognition module fails to recognize them. Though the result of Fig. 15(a) is clear enough for human readers, it's difficult for the recognition module. For the result of Fig. 15(b), the image is too blurry to recognize. Moreover, some text images in testing set are imperfect to be recognized. For example, the first letter in Fig. 15(c) is covered and causes the false prediction while the last letter "d" in Fig. 15(d) is prone to be predicted as "p" due to the stain beneath.

N. Discussion

In this method, we do not consider other attributes, such as the deformation of each single character. The reasons are threefold: (1) Attributes including center, angle and size of characters are enough for reconstruct bounding boxes to represent each character. (2) The deformation of character is not very common in existing scene text recognition datasets and parallelogram bounding boxes are enough for most of the situation. (3) Predicting more attributes will introduce more computation and make the pipeline redundant.

V. Conclusion

In this paper, we propose a Character-Aware Sampling and Rectification Network (CASR-Net) to rectify and recognize the irregular texts by character-wise geometrical attributes. CASR-Net employs a well-designed segmentation-based network and a Connected Components Selecting Algorithm (CCSA) to obtain accurate character-level information. Based on predicted geometrical attributes, the character-level text instance can be reconstructed and accurate control points can be generated for image rectification. Extensive experiments have been performed to demonstrate the effectiveness of our CASR-Net on various public benchmarks and our CASR-Net achieves promising performance on a number of text recognition datasets.

References

[1] 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, Nov. 2014.

[2] 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.*, 2016, pp. 4168–4176.

[3] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," in *Proc. Int. Conf. Learn. Representations*, 2015.

[4] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. 28th Int. Conf. Neural Informat. Process. Syst.*, vol. 1, MIT Press, 2015, pp. 577–585.

[5] B. Shi *et al.*, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, Sep. 2019.

[6] 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 Recognit.*, 2018, pp. 5571–5579.

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

[8] M. Yang *et al.*, "Symmetry-constrained rectification network for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9147–9156.

[9] T. Wang *et al.*, "Decoupled attention network for text recognition," in *Proc. AAAI Conf. Artif. Intell.*, vol. 34, no. 07, 2020, pp. 12 216–12 224.

[10] P. Dai, H. Zhang, and X. Cao, "Deep multi-scale context aware feature aggregation for curved scene text detection," *IEEE Trans. Multimedia*, vol. 22, no. 8, pp. 1969–1984, Aug. 2020.

[11] 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, Jun. 1989.

[12] M. Jaderberg, K. Simonyan, A. Zisserman, and K. kavukcuoglu, "Spatial transformer networks," in *Adv. Neural Inf. Process. Syst.*, Curran Associates, Inc., vol. 28, pp. 2017–2025, 2015.

[13] S. Zhang, Y. Liu, L. Jin, Z. Wei, and C. Shen, "OPMP: An omni-directional pyramid mask proposal network for arbitrary-shape scene text detection," *IEEE Trans. Multimedia*, vol. 23, pp. 454–467, 2021, doi: 10.1109/TMM.2020.2978630.

[14] M. Xue *et al.*, "Arbitrarily-oriented text detection in low light natural scene images," *IEEE Trans. Multimedia*, vol. 23, pp. 2706–2720, 2021, doi: 10.1109/TMM.2020.3015037.

[15] Y. Tang and X. Wu, "Scene text detection using superpixel-based stroke feature transform and deep learning based region classification," *IEEE Trans. Multimedia*, vol. 20, no. 9, pp. 2276–2288, Sep. 2018.

[16] 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.

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

[18] 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.*, 2012, pp. 2687–2694.

[19] T. Novikova, O. Barinova, P. Kohli, and V. Lempitsky, "Large-lexicon attribute-consistent text recognition in natural images," in *Proc. Eur. Conf. Comput. Vis.*, Springer, 2012, pp. 752–765.

[20] 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 Recognit.*, 2014, pp. 4042–4049.

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

[22] J.-H. Seok and J. H. Kim, "Scene text recognition using a Hough forest implicit shape model and semi-Markov conditional random fields," *Pattern Recognit.*, vol. 48, no. 11, pp. 3584–3599, 2015.

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

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

[25] X. Wu *et al.*, "LCSegNet: An efficient semantic segmentation network for large-scale complex chinese character recognition," *IEEE Trans. Multimedia*, vol. 23, pp. 3427–3440, 2021, doi: 10.1109/TMM.2020.3025696.

[26] 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.

[27] 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.

[28] 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. 2016.

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

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

[31] Z. Cheng *et al.*, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5076–5084.

[32] M. Liao *et al.*, "Scene text recognition from two-dimensional perspective," in *Proc. AAAI Conf. Artif. Intell.*, vol. 33, no. 1, 2019, pp. 8714–8721.

[33] 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.

[34] Y. Gao, Y. Chen, J. Wang, and H. Lu, "Progressive rectification network for irregular text recognition," *Sci. China Inf. Sci.*, vol. 63, no. 2, pp. 1–14, 2020.

[35] T.-Y. Lin *et al.*, "Feature pyramid networks for object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2117–2125.

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

[37] F. Milletari, N. Navab, and S.-A. Ahmadi, "V-Net: Fully convolutional neural networks for volumetric medical image segmentation," in *Proc. 4th Int. Conf. 3D Vis.*, 2016, pp. 565–571.

[38] A. Shrivastava, A. Gupta, and R. Girshick, "Training region-based object detectors with online hard example mining," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 761–769.

[39] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *Workshop Deep Learn.*, NIPS, 2014.

[40] A. Mishra, K. Alahari, and C. V. Jawahar, "Scene text recognition using higher order language priors," in *BMVC*, pp. 127.1–127.11, 2012.

[41] S. M. Lucas *et al.*, "ICDAR 2003 robust reading competitions: entries, results, and future directions," in *Proc. Int. Conf. Document Anal. Recognit.*, vol. 7, no. 2-3, Springer, 2005, pp. 105–122.

[42] D. Karatzas *et al.*, "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recognit.*, 2013, pp. 1484–1493.

[43] D. Karatzas *et al.*, "ICDAR 2015 competition on robust reading," in *Proc. Int. Conf. Document Anal. Recognit.*, 2015, pp. 1156–1160.

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

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

[46] W. Wang *et al.*, "Shape robust text detection with progressive scale expansion network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9336–9345.

[47] 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.

[48] A. Gordo, "Supervised mid-level features for word image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 2956–2964.

[49] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to read irregular text with attention mechanisms," in *Proc. Twenty-Sixth Int. Joint Conf. Artif. Intell., IJCAI-17*, 2017, pp. 3280–3286.

[50] W. Liu, C. Chen, and K.-Y. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7154–7161.

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

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

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

[54] 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.*, vol. 33, no. 1, 2019, pp. 8610–8617.

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

[56] Z. Qiao, Y. Zhou, D. Yang, Y. Zhou, and W. Wang, "SEED: Semantics enhanced encoder-decoder framework for scene text recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13528–13537.

[57] D. Yu *et al.*, "Towards accurate scene text recognition with semantic reasoning networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12113–12122.
[58] R. Litman *et al.*, "Scatter: Selective context attentional scene text recognizer," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11962–11972.
[59] H. Zhang, Q. Yao, M. Yang, Y. Xu, and X. Bai, "AutoSTR: Efficient backbone search for scene text recognition," in *Proc. Comput. Vis.-ECCV 2020: 16th Eur. Conf.*, Glasgow, U.K., Aug. 23-28, 2020, *Proceedings, Part XXIV 16*. Springer, 2020, pp. 751–767.
[60] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011.
[61] P. Pérez, M. Gangnet, and A. Blake, "Poisson image editing," in *ACM SIGGRAPH 2003 Papers*. Association for Computing Machinery, 2003, pp. 313–318.
[62] J. Chen, Z. Lian, Y. Wang, Y. Tang, and J. Xiao, "Irregular scene text detection via attention guided border labeling," *Sci. China Inf. Sci.*, vol. 62, no. 12, pp. 1–11, 2019.

Ming Li received the B.S. degree from Xi'an Jiaotong University, Xi'an, China, in 2020 and was a Research Assistant with the ShenZhen Key Laboratory of Computer Vision and Pattern Recognition, Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Beijing, China, from 2019 to 2021. He is currently a Master Student with Texas A&M university, College Station, TX, USA.

Bin Fu received the B.E. degree from Lanzhou University, Lanzhou, China, in 2014, the Ph.D. degree from the University of Hong Kong, Hong Kong, in 2006. He is currently an Assistant Research Fellow with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences Beijing, China. His research interests include semantic segmentation and scene text recognition.

Zhengfu Zhang received the B.S. degree from the Dalian University of Technology, Dalian, China, in 2016 and the M.S. degree from the Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Beijing, China, in 2020. He is currently working with SenseTime.

Yu Qiao (Senior Member, IEEE) is a Professor with the Shenzhen Institutes of Advanced Technology, the Chinese Academy of Science, Beijing, China and Shanghai AI Laboratory. He has authored or coauthored more than 240 papers in international journals and conferences, including T-PAMI, IJCV, T-IP, T-SP, CVPR, ICCV etc. His research interests include computer vision, deep learning, and bioinformation. His H-index is 67, with 28,000 citations in Google Scholar. He was the recipient of the Distinguished Paper Award in AAAI 2021. His Group achieved the first runner-up at the ImageNet Large Scale Visual Recognition Challenge 2015 in scene recognition, and the winner at the ActivityNet Large Scale Activity Recognition Challenge 2016 in video classification. He was the Program Chair of IEEE ICIST 2014.