MASTER: Multi-Aspect Non-local Network for Scene Text Recognition[⋆]

Ning Lu[a,1], Wenwen Yu[a,b,1,*], Xianbiao Qi[a], Yihao Chen[a], Ping Gong[b], Rong Xiao[a], Xiang Bai[c]

[a]*Visual Computing Group, Ping An Property and Casualty Insurance Company, Shenzhen, China*
[b]*School of Medical Imaging, Xuzhou Medical University, Xuzhou, China*
[c]*School of Electronic Information and Communications, Huazhong University of Science and Technology, Wuhan, China*

Abstract

Attention-based scene text recognizers have gained huge success, which leverages a more compact intermediate representation to learn 1d- or 2d- attention by a RNN-based encoder-decoder architecture. However, such methods suffer from **attention-drift** problem because high similarity among encoded features leads to attention confusion under the RNN-based local attention mechanism. Moreover, RNN-based methods have low efficiency due to poor parallelization. To overcome these problems, we propose the MASTER, a self-attention based scene text recognizer that (1) not only encodes the input-output attention but also learns self-attention which en-

---

[*] Corresponding author.

Email addresses: jiangxiluning@gmail.com (Ning Lu), yuwenwen62@gmail.com (Wenwen Yu), qixianbiao@gmail.com (Xianbiao Qi), o0o@o0oo0o.cc (Yihao Chen), gongping@xzhmu.edu.cn (Ping Gong), xiaorong283@pingan.com.cn (Rong Xiao), xbai@hust.edu.cn (Xiang Bai)

[1] Co-first authors.

codes feature-feature and target-target relationships inside the encoder and decoder and (2) learns a more powerful and robust intermediate representation to spatial distortion, and (3) owns a great training efficiency because of high training parallelization and a high-speed inference because of an efficient memory-cache mechanism. Extensive experiments on various benchmarks demonstrate the superior performance of our MASTER on both regular and irregular scene text. Pytorch code can be found at https://github.com/wenwenyu/MASTER-pytorch, and Tensorflow code can be found at https://github.com/jiangxiluning/MASTER-TF.

Keywords:

Scene text recognition, Transformer, Non-local network, Memory-cached mechanism

---

1. Introduction

Scene text recognition in the wild is a hot area in both industry and academia in the last two decades [1, 2, 3]. There are various application scenarios such as text identification on the signboard for autonomous driving, ID card scan for a bank, and key information extraction in Robotic Process Automation (RPA). However, constructing a high-quality scene text recognition system is a non-trivial task due to unexpected blur, strong exposure, spatial and perspective distortion, and complex background. There are two types of scene text in nature, **regular** and **irregular**, as exemplified in Figure 1.

Regular scene text recognition aims to recognize a sequence of characters from an almost straight text image. It is usually considered as an image-based sequence recognition problem. Some traditional text recognition methods [4]
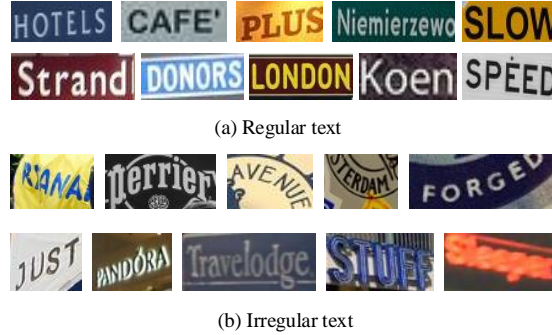
(a) Regular text

(b) Irregular text

Figure 1: Examples of regular and irregular images. (a). regular text. (b). irregular text.

use human-designed features to segment patches into small glyphs, then categorize them into corresponding characters. However, these methods are known to be vulnerable to the complicated background, diverse font types, and irregular arrangement of the characters. Connectionist temporal classification (CTC) based methods [5, 6] and attention-based methods [7, 8] are the mainstream methods for scene text recognition because they do not require character-level annotations and also show superior performance on real applications.

Irregular scene text recognition is more challenging due to various curved shapes and perspective distortions. Existing irregular scene text recognizers can be divided into three categories: rectification based, multi-direction encoding based, and attention-based approaches. Shi *et al.* [7] propose ASTER to combine a Thin-Plate Spline (TPS) [9] transformation as rectification module and an attentional BiLSTMs as recognition module. ASTER achieves excellent performance on many public benchmarks. Yang *et al.* [10] put forward a Symmetry-constrained Rectification Network (ScRN) to tackle highly

curved or distorted text instances. Liao *et al.* [11] conduct scene text recognition from a two-dimensional perspective. Xie *et al.* [12] utilize a convolutional attention networks for unconstrained scene text recognition. Fang *et al.* [13] propose to ensemble attention and language models in an attention-based architecture. Inspired by the Show-Attend-Tell [14], Li *et al.* [8] propose a Show-Attend-Read (SAR) method which employs a 2D attention in encoder-decoder architecture. Nonetheless, attention drifting remains a serious problem in these methods, especially when the text lines contain repetitive digits or characters.

Incorporating global context proves to be an effective way to alleviate the problem of attention drifting. Self-attention [15] provides an effective approach to encode global context information. Recently, self-attention attracts a lot of eyeballs and gains unprecedented success in natural language processing [15, 16, 17, 18] and computer vision [19, 20]. It can depict a long-range dependency between words and image regions. Wang *et al.* [21] proposes a Transformer-like non-local block that can be plugged in any backbone to model spatial global dependencies between objects. Its successors, GCNet, proposed in [19], found that the attention maps are almost the same for different query positions. GCNet simplifies non-local block with SE block [22] to reduce the computational complexity and enhances the representative ability of the proposed block based on a query-independent formulation.

Inspired by the effectiveness of the global context in GCNet and the huge success of the Transformer achieved in NLP and CV, we propose a **M**ulti-**A**spect non-local network for irregular **S**cene **TE**xt **R**ecognition (MASTER) to target an efficient and accurate scene text recognition for both regular and

irregular text. Our main contributions are highlighted as follows:

- We propose a novel multi-aspect non-local block and fuse it into the conventional CNN backbone, which enables the feature extracter to model a global context. The proposed multi-aspect non-local block can learn different aspects of spatial 2D attention, which can be viewed as a multi-head self-attention module. Different types of attention focus on different aspects of spatial feature dependencies, which is another form of different syntactic dependency types.

- In the inference stage, we introduce a memory-cache based decoding strategy to speed up the decoding procedure. The primary means are to remove unnecessary computation and cache some intermediate results of previous decoding times.

- Besides of its high efficiency, our method achieves the state of the art performance on both regular and irregular scene text benchmarks. Especially, our method achieves the best case-sensitive performance on the COCO-Text dataset.

2. Related Works

In academia, scene text recognition can be divided into two categories: regular and irregular texts. In this section, we will give a brief review of related works in both areas. A more detailed review for scene text detection and recognition can be found in [23, 24, 25, 26].

Regular text recognition attracts most of the early research attention. Mishra *et al.* [27] use a traditional sliding window-based method to describe

bottom-up cues and use vocabulary prior to model top-down cues. These cues are combined to minimize the character combination's energy. Shi *et al.* [5] propose an end-to-end trainable character annotation-free network, called CRNN. CRNN extracts a 1D feature sequence using CNN and then encodes the sequence encoding using RNN. Finally, a CTC loss is calculated. CTC loss only needs word-level annotation instead of character-level annotation. Su *et al.*[28] also proposed a method performing word-level recognition without character segmentation using a recurrent neural network. Bigorda *et al.* [29] design a text-specific selective search algorithm to generates a hierarchy of word hypotheses for word spotting in the wild. Gao *et al.* [30] integrates an attention module into the residual block to amplify the response of the foreground and suppress the response of the background. However, the attention module cannot encode global dependencies between pixels. Cheng *et al.* [31] observe that attention may drift due to the complex scenes or low-quality images, which is a weakness of the vanilla 2D-attention network. To address the misalignment between the input sequence and the target, Bai *et al.* [6] employs an attention-based encoder-decoder architecture, and estimate the edit probability of a text conditioned on the output sequence. Edit probability is to target the issue of character missing and superfluous. Zhang *et al.* [32] adopts an unsupervised fixed-length domain adaptation methodology to a variable-length scene text recognition area and the model is also based on attentional encoder-decoder architecture.

Irregular text recognition is more challenging than regular text recognition, nevertheless, it appeals to most of researchers' endeavour. Shi *et al.* [33, 7] attempt to address the multi-type irregular text recognition prob-

lem in one framework via Spatial Transformer Network (STN) [34]. In [35], Zhan *et al.* propose to iteratively rectify text images to be fronto-parallel to further improve the recognition performance. The proposed line-fitting transformation estimates the pose of the text line by learning a middle line of the text line and L line segments that are required by Thin-Plate Spline. However, the rectification-based methods are often constrained by the characters' geometric features and the background noise could be exaggerated unexpectedly. To overcome this, Luo *et al.* [36] propose a multi-object rectified attention network which is more flexible than direct affine transformation estimation. Unlike the rectification-based approaches, Show-Attend-Read (SAR) proposed by Li [8] uses a 2D-attention mechanism to guide the encoder-decoder recognition module to focus on the corresponding character region. This method is free to complex spatial transformation.

While 2D attention can represent the relationship between target output and input image feature, the global context between pixels and the latent dependency between characters is ignored. In [20], Hu *et al.* proposes an object relation module to simultaneously model a set of object relations through their visual features. After the success of Transformer [15], Wang *et al.* [21] incorporate a self-attention block into non-local network. Cao *et al.* [19] further simplify and improve the non-local network, and propose a novel global context network (GCNet). Recently, Sheng *et al.* [37] propose a purely Transformer-based scene text recognizer that can learn the self-attention of encoder and decoder. It extracts a 1D sequence feature using a simple CNN module and inputs it into a Transformer to decode target outputs. Nevertheless, the self-attention module of the Transformer consists of multiple fully

connected layers, which largely increases the number of parameters. Lee *et al.* [38], use the self-attention mechanism to capture two-dimensional (2D) spatial dependencies of characters. A locality-aware feedforward layer is introduced in their encoder. Wang *et al.* [39] directly abandon the encoder of the original Transformer and only retain the CNN feature extractor and decoder to conduct an irregular scene text recognizer. However, it cannot encode the global context of pixels in the feature map. The network proposed in this paper learns not only the 2D attention between the input feature and output target but also the self-attention inside the feature extractor and decoder. The multi-aspect non-local block can encode different types of spatial feature dependencies with lower computational cost and a compact model.

3. Methodology

MASTER model, as shown in Figure 2c, consists of two key modules, a Multi-Aspect Global Context Attention (GCAttention) based encoder and a Transformer based decoder. In MASTER, an image with fixed size is input into the network, and the output is a sequence of predicted characters.

3.1. Encoder

Encoder, in our MASTER model, encodes an input image into a feature tensor. For instance, we can obtain a $6 \times 40 \times 512$ tensor when inputting a $48 \times 160 \times 1$ image into the encoder of MASTER. One of our key contribution in this paper is that we introduce a Multi-Aspect Global Context Attention (GCAttention) in the encoder part. In this subsection, we will review the definition of the Global Context Block [19], and then introduce the proposed
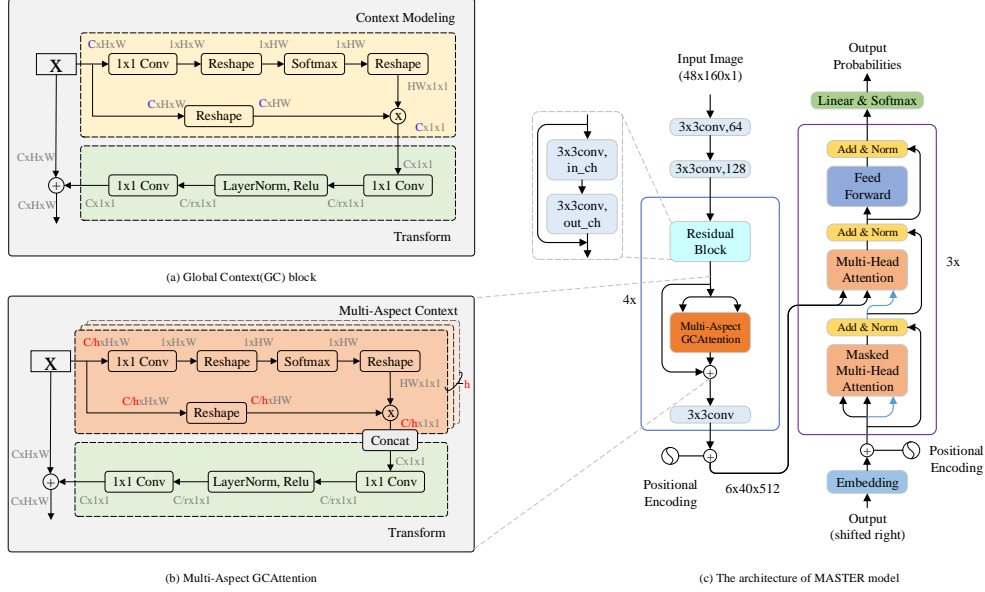
8

Figure 2: (a) representing the architecture of a standard Global Context(GC) block. (b) representing the proposed Multi-Aspect GCAttention. (c) representing the whole architecture of MASTER model, consisting of two main parts: a Multi-Aspect Global Context Attention(GCAttention) based encoder for feature representation and a transformer based decoder model. $C \times H \times W$ denotes a feature map with channel number C, height H and width W. h, r, and C/r denotes the number of Multi-Aspect Context, bottleneck ratio and hidden representation dimension of the bottleneck, respectively. $\otimes$ denotes matrix multiplication, $\oplus$ denotes broadcast element-wise addition. in_ch/out_ch donates input/output dimensions.

9

Multi-Aspect Global Context Attention (GCAttention), and finally describe the architecture of the encoder in detail.

3.1.1. Global Context Block

A standard global context block was firstly introduced in [19]. The module structure is shown as Figure 2a. From Figure 2a, the input feature map of global context block is $\mathbf{x} = \{\mathbf{x}_i\}_{i=1}^{H \times W} \in \mathcal{R}^{C \times H \times W}$ ($C = d_{model}$), where C, W, and H indicate the number of channel, width and height of the feature map individually. d_{model} indicates the dimension of the output of the encoder. In global context block, three operations are performed on the feature map $\mathbf{x}$, including (a) global attention pooling for **context modeling**, (b) bottleneck **transform** to capture channel-wise dependencies, and (c) broadcasting element-wise **addition** for feature fusion. The global context block can be expressed as

$$\mathbf{y}_i = \mathbf{x}_i + \mathbf{w}_{v2}\text{ReLU}\left(\text{LN}\left(\mathbf{w}_{v1}\sum_{\forall j}\frac{e^{\mathbf{w}_k\mathbf{x}_j}}{\sum_{\forall m}e^{\mathbf{w}_k\mathbf{x}_m}}\mathbf{x}_j\right)\right), \qquad (1)$$

where $\mathbf{x}$ and $\mathbf{y}$ denote the input and the output of the global context block, respectively. They have the same dimensions. i is the index of query positions, j and m enumerates positions of all pixels. $\mathbf{w}_{v1}$, $\mathbf{w}_{v2}$ and $\mathbf{w}_k$ denote linear transformations to be learned via a 1×1 convolution. $\text{LN}(\cdot)$ denotes layer normalization as [40]. For simplification, we denote $\alpha_j = \frac{e^{\mathbf{w}_k\mathbf{x}_j}}{\sum_m e^{\mathbf{w}_k\mathbf{x}_m}}$ as the weight for **context modeling**, and $\delta(\cdot) = \mathbf{w}_{v2}\text{ReLU}(\text{LN}(\mathbf{w}_{v1}(\cdot)))$ as the bottleneck **transform**. "+" operation denotes broadcast element-wise **addition**.

In the Global Context Block, as shown in Figure 2a, the softmax operation follows behind a 1x1 Conv and flatten operation, in which the feature map

will be converted from (C, H, W) to (1, H*W). The generated vector (1, H*W) is a channel-agnostic feature and it captures spatial information of feature map. Besides, the softmax operation depicts a long-range dependency between pixels in the feature map.

3.1.2. Multi-Aspect GCAttention

Instead of performing a single attention function in original global context block, we found it beneficial to multiple attention function. Here, we call it as Multi-Apsect GCAttention (MAGC). The structure of the MAGC is illustrated in Figure 2b, and we can formulate MAGC as

$$
\begin{cases}
\mathbf{y} = \mathbf{x} + \delta(MAGC(\mathbf{x})), \\
MAGC(\mathbf{x}) = Concat(gc_1, gc_2, \ldots, gc_h), \\
gc_i = \sum_{j=1}^{L} \alpha_j \mathbf{x}_j, \\
\boldsymbol{\alpha} = softmax\left(\dfrac{\mathbf{w}_k \mathbf{x}_1}{\sqrt{d_h}}, \dfrac{\mathbf{w}_k \mathbf{x}_2}{\sqrt{d_h}}, \ldots, \dfrac{\mathbf{w}_k \mathbf{x}_L}{\sqrt{d_h}}\right),
\end{cases}
\tag{2}
$$

where h is the number of Multi-Aspect Context, gc_i denotes the i-th global context, L is the number of positions of all pixels in the feature map ($L = W \times H$), $Concat(\cdot)$ is a concatenation function. $MAGC(\cdot)$ denotes multi-aspect global context attention operation. $\sqrt{d_h}$ is a scale factor to counteract the effect of different variance in MAGC. It can be calculated as $d_h = \frac{d_{model}}{h}$.

3.1.3. Encoder Structure

The detailed architecture of Multi-Aspect GCAttention based Encoder is shown in the left half of Figure 2c. The backbone of the encoder, following the design of ResNet31 [41] and the setting protocol in [8], is presented in Table 1. The encoder has four fundamental blocks shown in blue color in

11

Figure 2c, each fundamental block consists of a residual block, a MAGC, and a convolution block, and max pooling that is not included in the last two fundamental blocks. In the residual block, if the input and output dimensions are different we use the projection shortcut, otherwise, we use the identity shortcut. After the residual block, a Multi-Aspect GCAttention is plugged into network architectures to learn new feature representation from multi-aspect. All the convolutional kernel size is 3×3. Besides two 2×2 max-pooling layers, we also use a 1×2 max-pooling layer, which reserves more information along the horizontal axis and benefits the recognition of narrow shaped characters.

3.2. Decoder

As shown in the right halves of Figure 2c, the decoder contains a stack of $N = 3$ fundamental blocks as shown in purple color. Each fundamental block contains three core modules, a Masked Multi-Head Attention, a Multi-Head Attention, and a Feed-Forward Network (FFN). In the following, we introduce these three key modules in detail, then discuss the loss function used in this paper, and finally introduce memory-cache based inference mechanism.

3.2.1. Scaled Multi-Head Dot-Product Attention

A scaled multi-head dot product attention is firstly introduced in [10]. The inputs of the scaled dot-product attention consist of a query $\mathbf{q}_i^T \in \mathcal{R}^d, i \in [1, t']$, (where $d = d_{model}$ is the dimension of embedding output and t' is the number of queries), and a set of key-value pairs of d-dimensional vectors $\{(\mathbf{k}_j, \mathbf{v}_j)\}_{j \in [1,t]}$, $\mathbf{k}_j^T \in \mathcal{R}^d$, $\mathbf{v}_j^T \in \mathcal{R}^d$ (where t is the number of key-value pairs). The formulation of scaled dot-product attention can be expressed as

follows

$$
\begin{cases}
Atten(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathbf{a}_1, \mathbf{a}_2, \dots, \mathbf{a}_{t'}]^T \in \mathcal{R}^{t' \times d}, \\
\mathbf{a}_i = Atten(\mathbf{q_i}, \mathbf{K}, \mathbf{V}), \\
Atten(\mathbf{q_i}, \mathbf{K}, \mathbf{V}) = \sum_{j=1}^{t} \alpha_j \mathbf{v}_j^T \in \mathcal{R}^d, \\
\boldsymbol{\alpha} = softmax\left(\dfrac{\left\langle \mathbf{q_i}, \mathbf{k}_1^T \right\rangle}{\sqrt{d}}, \dfrac{\left\langle \mathbf{q_i}, \mathbf{k}_2^T \right\rangle}{\sqrt{d}}, \dots, \dfrac{\left\langle \mathbf{q_i}, \mathbf{k}_t^T \right\rangle}{\sqrt{d}} \right),
\end{cases}
\tag{3}
$$

where $\boldsymbol{\alpha}$ is the attention weights, and $\mathbf{K} = [\mathbf{k}_1; \mathbf{k}_2; \dots; \mathbf{k}_t] \in \mathcal{R}^{t \times d}$, $\mathbf{V} = [\mathbf{v}_1; \mathbf{v}_2; \dots; \mathbf{v}_t] \in \mathcal{R}^{t \times d}$. $\mathbf{Q} = [\mathbf{q}_1; \mathbf{q}_2; \dots; \mathbf{q}_{t'}] \in \mathcal{R}^{t' \times d}$ is a set of queries.

The above scaled dot-product attention can be repeated multiple times (multi-head) with different linear transformations on $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$, followed by a concatenation and linear transformation operation:

$$
MHA(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathbf{head}_1, \dots, \mathbf{head}_H] \mathbf{W}^o \in \mathcal{R}^{t' \times d}, \tag{4}
$$

where $\mathbf{head}_i = Atten\left(\mathbf{Q}\mathbf{W}_i^q, \mathbf{K}\mathbf{W}_i^k, \mathbf{V}\mathbf{W}_i^v \right) \in \mathcal{R}^{t' \times \frac{d}{H}}$, $MHA(\cdot)$ denotes multi-head attention operation. The parameters are $\mathbf{W}_i^q \in \mathcal{R}^{d \times \frac{d}{H}}, \mathbf{W}_i^k \in \mathcal{R}^{d \times \frac{d}{H}}, \mathbf{W}_i^v \in \mathcal{R}^{d \times \frac{d}{H}}$ and $\mathbf{W}^o \in \mathcal{R}^{d \times d}$. H denotes the number of multi-head attention.

3.2.2. Masked Multi-Head Attention

This module is identical to the decoder of Transformer [15]. Masked multi-head attention is an effective mechanism to promise that, in the decoder, the prediction of one time step t can only access the output information of its previous time steps. In the training stage, by creating a lower triangle mask matrix, the decoder can output predictions for all time steps simultaneously instead of one by one sequentially. This makes the training process highly parallel.

3.2.3. Position-wise Feed-Forward Network

Point-wise Feed-Forward Network (FFN) consists of two fully connected layers. Between these two layers, there is one ReLU activation function. FFN is defined

as

$$FFN(\mathbf{x}) = \max\left(\mathbf{0}, \mathbf{x}\mathbf{W_1} + \mathbf{b_1}\right)\mathbf{W_2} + \mathbf{b_2}, \tag{5}$$

where the weights are $\mathbf{W}_1 \in \mathcal{R}^{d \times d_{ff}}$, and $\mathbf{W}_2 \in \mathcal{R}^{d_{ff} \times d}$, and the bias are $\mathbf{b}_1 \in \mathcal{R}^{d_{ff}}$ and $\mathbf{b}_2 \in \mathcal{R}^d$, d_{ff} is the inner-dimension of the two linear transformations. The aim of this module is to bring in more non-linearity to the network.

3.2.4. Loss Function

A linear transformation followed by a softmax function is used to compute the prediction probability over all classes. Then, we use the standard cross-entropy to calculate the loss between the predicted probabilities w.r.t. the ground truth, at each decoding position. In this paper, we use 66 symbol classes except for COCO-Text which uses 104 symbol classes. These 66 symbols are 10 digits, 52 case-sensitive letters, and 4 special punctuation characters. These 4 special punctuation characters are "<SOS>", "<EOS>", "<PAD>", and "<UNK>" which indicate the start of the sequence, the end of the sequence, padding symbol and unknown characters (that are neither digit nor character), respectively. The parameters of the classification layer are shared over all decoding positions.

3.3. Memory-Cache based Inference Mechanism

The inference stage is different from the training stage. In the training stage, by constructing a lower triangular mask matrix, the decoder can predict out all-time steps simultaneously. This process is highly parallel and efficient, where parallel means the batch mechanism. However, the decoder in the inference stage can only predict each character one by one sequentially until the decoder predicts out the "EOS" token or the length of the decoder sequence reaches to the maximum length. In the inference stage, the output of the later step is dependent on the outputs of its previous time steps because the outputs of its previous time steps will be used as part of the input to decode itself.

Algorithm 1: Memory-Cache based Inference. B is the number of blocks. F is the addition of the CNN feature and the position embedding feature. T is the max decoder length. M and W are the parameters of the masked multi-head and multi-head attention. X_k^b, X_v^b, $keys\_memory$, $values\_memory$ are the cached variables.

Input	:	CNN feature: F
Output	:	$outputs$

1 **for** b in $range(B)$ **do**

2 $\quad$ $X_k^b, X_v^b = W_k^b * F, W_v^b * F$;

3 $\quad$ $keys\_memory[b], values\_memory[b] = [\,], [\,]$;

4 **end**

5 $t \leftarrow 0$;

6 $outputs = [\,]$;

7 $p_t \leftarrow$<SOS>;

8 **while** $p_t \neq$<EOS> and $t \leq T$ **do**

9 $\quad$ $q = $ Embedding $(p_t) + $ PositionEmbedding (t);

10 $\quad$ **for** b in $range(B)$ **do**

11 $\quad\quad$ $keys\_memory[b].append(M_k^b * q)$;

12 $\quad\quad$ $values\_memory[b].append(M_v^b * q)$;

13 $\quad\quad$ $q \leftarrow$ MaskedMHA$(M_q^b * q, keys\_memory[b], values\_memory[b])$;

14 $\quad\quad$ $q \leftarrow$ MHA$(W_q^b * q,\ X_k^b, X_v^b)$;

15 $\quad\quad$ $q \leftarrow$ FeedForward(q);

16 $\quad$ **end**

17 $\quad$ $t \leftarrow t + 1$;

18 $\quad$ $p_t \leftarrow$ Argmax(LinearSoftmax(q));

19 $\quad$ $outputs.append(p_t)$

20 **end**

To speed up the decoding process, we introduce a new decoding mechanism named memory-cache based decoding inspired by XLNet [17]. The memory-cache based decoding strategy is described in Algorithm 1 in pseudo-code. The primary approaches are to cache some intermediate results of previous decoding times in Lines 2 and 11-12, and to remove unnecessary computation in Lines 13-14 of Algorithm 1. In each decoding step, q is always a 1D vector instead of a 2D matrix in traditional decoding framework.

4. Experiments

We conduct extensive experiments on several benchmarks to verify the effectiveness of our method and compare it with the state-of-the-art methods. In Section 4.1, we give an introduction to the used training and testing datasets. Then in Section 4.2, we present our implementation details. In Section 4.3, we make a detailed comparison between our method and the state-of-the-art methods. Finally, we conduct an ablation study in Section 4.4.

4.1. Datasets

In this paper, we train our MASTER model only on three synthetic datasets without any finetuning on other real datasets. We evaluate our model on eight standard benchmarks that contain four regular scene text datasets and four irregular scene text datasets.

The training datasets consist of the following datasets.

Synth90k (MJSynth) is the synthetic text dataset proposed in [42]. The dataset has 9 million images generated from a set of 90k common English words. Every image in Synth90k is annotated with a word-level ground-truth. All of the images in this dataset are used for training.

Table 1: A ResNet-based CNN network architecture for robust text feature representation. Residual blocks are shown in brackets, and Multi-Aspect GCAttention is highlighted with gray background. "$3 \times 3, 1 \times 1, 1 \times 1, 128$" denotes the kernel size, the stride, the padding, and the output channel of a convolution layer respectively. The "Output" column means the spatial shape $height \times width$ of the output.

Layer	Configuration	Output
conv1_x	$3 \times 3, 1 \times 1, 1 \times 1, 64$	48×160
	$3 \times 3, 1 \times 1, 1 \times 1, 128$	48×160
	max_pool: $2 \times 2, 2 \times 2, 0 \times 0$	24×80
conv2_x	$\begin{bmatrix} 3 \times 3, 256 \\ 3 \times 3, 256 \end{bmatrix} \times 1$	24×80
	multi-aspect gcattention	24×80
	$3 \times 3, 1 \times 1, 1 \times 1, 256$	24×80
	max_pool: $2 \times 2, 2 \times 2, 0 \times 0$	12×40
conv3_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 2$	12×40
	multi-aspect gcattention	12×40
	$3 \times 3, 1 \times 1, 1 \times 1, 512$	12×40
	max_pool: $2 \times 1, 2 \times 1, 0 \times 0$	6×40
conv4_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 5$	6×40
	multi-aspect gcattention	6×40
	$3 \times 3, 1 \times 1, 1 \times 1, 512$	6×40
conv5_x	$\begin{bmatrix} 3 \times 3, 512 \\ 3 \times 3, 512 \end{bmatrix} \times 3$	6×40
	multi-aspect gcattention	6×40
	$3 \times 3, 1 \times 1, 1 \times 1, 512$	6×40

SynthText [43] is a synthetic text dataset originally introduced for text detection. The generating procedure is similar to [42], but different from [42], words are rendered onto a full image with a large resolution instead of a text line. 800 thousand full images are used as background images, and usually, each rendered image contains around 10 text lines. Recently, It is also widely used for scene text recognition. We obtain 7 millions of text lines from this dataset for training.

SynthAdd is the synthetic text dataset proposed in [8]. The dataset contains 1.6 million word images using the synthetic engine proposed by [42] to compensate for the lack of special characters like punctuations. All of the images in this dataset are used for training.

The test datasets consist of the following datasets.

IIIT5K-Words (IIIT5K) [44] has 3,000 test images collected from the web. Each image contains a short, 50-word lexicon and a long, 1,000-word lexicon. A lexicon includes the ground truth word and other stochastic words.

Street View Text (SVT) [45] is collected from the Google Street View. The test set includes 647 images of cropped words. Many images in SVT are severely corrupted by noise and blur or have low resolution. Each image contains a 50-word lexicon.

ICDAR 2003 (IC03) [46] contains 866 images of the cropped word because we discard images that contain non-alphanumeric characters or have less than three characters for a fair comparison. Each image contains a 50-word lexicon defined.

ICDAR 2013 (IC13) [47] contains 1,095 images for evaluation and 848 cropped image patches for training. We filter words that contain non-alphanumeric characters for a fair comparison, which results in 1,015 test words. No lexicon is provided.

ICDAR 2015 (IC15) has 4,468 cropped words for training and 2,077 cropped words for evaluation, which are capture by Google Glasses without careful posi-

tioning and focusing. The dataset contains many of irregular text.

SVT-Perspective (SVTP) consists of 645 cropped images for testing [48]. Images are generated from side-view angle snapshots in Google Street View. Therefore, most images are perspective distorted. Each image contains a 50-word lexicon and a full lexicon.

CUTE80 (CUTE) contains 288 images [49]. It is a challenging dataset since there are plenty of images with curved text. No lexicon is provided.

COCO-Text (COCO-T) was firstly introduced in the Robust Reading Challenge of ICDAR 2017. It contains 62,351 image patches cropped from the famous Microsoft COCO dataset. The COCO-T dataset is extremely challenging because the text lines are mixed up with printed, scanned, and handwritten texts, and the shapes of text lines vary a lot. For this dataset, 42,618, 9,896, and 9,837 images are used for training, validation, and testing individually.

4.2. Network Structure and Implementation Details

4.2.1. Networks

The network structure of the Encoder part is listed in Table 1. The input size of our model is 48×160. When the ratio between width and height is larger than $\frac{160}{48}$, we directly resize the input image into 48×160, otherwise, we resize the height to 48 while keeping the aspect ratio and then pad the resized image into to 48×160. In MASTER, the embedded dimension d is 512, the dimension of the output of the encoder d_{model} is 512 too, and the number H of the multi-head attention is 8. d_{ff} in the feed-forward module is set to be 2048, and the identical layers N is 3. We use 0.2 dropout on the embedding module, feed-forward module, and the output layer of the linear transformation in the decoder part. The number h of Multi-Aspect Context is 8 and the bottleneck ratio r is 16.

19

Table 2: Performance of our model and other state-of-the-art methods on public datasets. All values are reported as a percentage (%). "None" means no lexicon. * indicates using both word-level and character-level annotations to train the model. ** denotes the performance of SAR trained only on the synthetic text datasets. In each column, the best performance result is shown in **bold** font, and the second-best result is shown with an underline. Our model achieves competitive performance on most of the public datasets, and the distance between us and the first place [50] is very small on IIIT5k and SVT datasets.

Method	IIIT5K	SVT	IC03	IC13	IC15	SVTP	CUTE
	None	None	None	None	None	None	None
Jaderberg *et al.* [34]	-	80.7	93.1	90.8	-	-	-
Shi *et al.* [33]	81.9	81.9	90.1	88.6	-	71.8	59.2
STAR-Net [51]	83.3	83.6	-	89.1	-	73.5	-
Wang and Hu [52]	80.8	81.5	-	-	-	-	-
CRNN [5]	81.2	82.7	91.9	89.6	-	-	-
Focusing Attention [31]*	87.4	85.9	94.2	93.3	70.6	-	-
SqueezedText [53]*	87.0	-	-	92.9	-	-	-
Char-Net [54]*	92.0	85.5	-	91.1	74.2	78.9	-
Edit Probability [6]*	88.3	87.5	94.6	94.4	73.9	-	-
ASTER [7]	93.4	89.5	94.5	91.8	76.1	78.5	79.5
NRTR [37]	86.5	88.3	<u>95.4</u>	<u>94.7</u>	-	-	-
SAR** [8]	91.5	84.5	-	91.0	69.2	76.4	83.3
ESIR [35]	93.3	90.2	-	91.3	76.9	79.6	83.3
MORAN [36]	91.2	88.3	95.0	92.4	68.8	76.1	77.4
Wang *et al.* [39]	93.3	88.1	-	91.3	74.0	80.2	85.1
Mask TextSpotter [50]*	**95.3**	**91.8**	95.2	**95.3**	<u>78.2</u>	<u>83.6</u>	**88.5**
MASTER (Ours)	<u>95.0</u>	<u>90.6</u>	**96.4**	**95.3**	**79.4**	**84.5**	<u>87.5</u>

4.2.2. Implementation Details

Our model is only trained on three synthetic datasets without any finetune on any real data except for COCO-T dataset. These three synthetic datasets are SynText [43] with 7 millions of text images, Synth90K [42] with 9 millions of text images and SynthAdd [8] with 1.6 millions of text images.

Our model is implemented in PyTorch. The model is trained on four NVIDIA Tesla V100 GPUs with 16×4 GB memory. We train the model from scratch using Adam [55] optimizer and cross-entropy loss with a batch size of 128×4. The learning rate is set to be 4×10^{-4} over the whole training phase. We observe that the learning rate should be associated with the number of GPUs. For one GPU, 1×10^{-4} is a good choice. Our model is trained for 12 epochs, each epoch takes about 3 hours.

Only for COCO-Text, we further finetune the above model with around 9K real images collected from IC13, IC15, and IIIT5K, and the training and validation images of COCO-Text. At the test stage, for the image with its height larger than width, we rotate the images 90 degrees clockwise and anti-clockwise. We feed the original image and two rotated images into the model and choose the output result with the maximum output probability. No lexicon is used in this paper. Different from SAR [8], ASTER [7], and NRTR [37], we do not use beam search.

4.3. Comparisons with State-of-the-arts

In this section, we measure the proposed method on several regular and irregular text benchmarks and analyze the performance with other state-of-the-art methods. We also report results on the online COCO-Text datasets test server[2] to show the performance of our model.

---

[2]`https://rrc.cvc.uab.es/?ch=5&com=evaluation&task=2`

Table 3: Leaderboard of various methods on the online COCO-Text test server. In each column, **Bold** represent the best performance.

Method	Case Sensitive		Case Insensitive	
	Total Edit Distance	Correctly Recognised Words (%)	Total Edit Distance	Correctly Recognised Words (%)
SogouMM	3,496.3121	44.64	**1,037.2197**	**77.97**
SenseTime-CKD	4,054.8236	41.52	824.6449	77.22
HIK_OCR	3,661.5785	41.72	899.1009	76.11
Tencent-DPPR Team	4,022.1224	36.91	1,233.4609	70.83
CLOVA-AI [56]	3,594.4842	47.35	1,583.7724	69.27
SAR [8]	4,002.3563	41.27	1,528.7396	66.85
HKU-VisionLab [54]	3,921.9388	40.17	1,903.3725	59.29
MASTER (single model)	3,527.3165	45.96	1,528.7526	67.41
MASTER (Ours)	**3,272.0810**	**49.09**	1,203.4201	71.33

As shown in Table 2, our method achieves superior performance on both regular and irregular datasets compared to the state-of-the-art methods. On the regular datasets including IIIT-5K, IC03, and IC13, our approach largely improves SAR [8] which is based on LSTM with 2D attention and ASTER [7] which is based on Seq2Seq model with attention after a text rectification module. Specifically, our approach improves SAR by 3.5% and 6.1% on IIIT-5K and SVT individually. On the irregular datasets, our method achieves the best performance on SVTP and IC15 datasets. This fully demonstrates the multi-aspect mechanism used in MASTER is highly effective in irregular scene text. Note that all these results are not with lexicon and beam search. The method in [50] uses extra character-level data.

Furthermore, seen from Table 3, we also use online evaluation tools on COCO-Text datasets to verify our competitive performance. As we can see, our model outperforms the compared method by a large margin in case sensitive metrics,

demonstrating the powerful network. Specifically, our model gets correctly recognised word accuracy increases of 1.74% (from 47.35% to 49.09%) under case sensitive conditions. In the case of case-insensitive metrics, our model also gets the fourth place on the leaderboard and the performance is much better than SAR. Note that, the first place method of case-insensitive uses a tailored 2D-attention module and the second and third place method of case-insensitive leaderboard use model ensemble. Our results are based on ensemble of four models obtained in different time steps of the same round of training process. The prediction with the maximum probability in four models is selected as the final prediction.

Seen from Figure 3, Our method possesses more robust performance on scene text recognition than SAR [8], although the input image quality is blurry and the shape is curved or the text is badly distorted. The reason is that our model not only learns the input-output attention but also learns self-attention which encodes feature-feature and target-target relationships inside the encoder and decoder. This makes the intermediate representations more robust to spatial distortion. Besides, in our approach, the problem of attention drifting is significantly eased. As shown in Figure 3, the attention driftings lead to errors ("FOOTBALL" and "TIMMS" are misrecognized as "FOOTBAL" and "TIMMMS" individually.) in SAR, but MASTER successfully recognizes these words.

4.4. Ablation Studies

4.4.1. Influence of Key Hyperparameters

we perform a series of ablation studies to analyze the impact of different hyperparameters on the recognition performance. All models are trained from scratch on three synthetic datasets (Synth90K, SynthText, and SynthAdd). Results are reported on seven standard benchmarks without using a lexicon. Here, we study two key hyperparameters, the number h of Multi-Aspect Context in the encoder
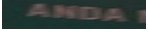
Input Images	Ours	By SAR [8]	GT
	ANDA	AMDA	ANDA
	GOOD	GCOD	GOOD
	wacom	waccom	wacom
	BONNIE	BONIE	BONNIE
	SERV	LEAD	SERV
	actaea	actara	actaea
	FOOTBALL	FOOTBAL	FOOTBALL
	Timms	Timmms	Timms

Figure 3: Samples of recognition results of our MASTER and SAR. Green characters mean correct predictions and red characters mean wrong predictions.

part, and the number N of fundamental blocks in the decoder part. The results are shown in Table 4.

There are two groups of experimental comparisons in Table 4. Fix $N = 3$, we vary h ranging in $[0, 1, 2, 4, 8, 16]$, where $h = 0$ means no MAGC is used in the model. We observe that using the MAGC module consistently improves the performance compared to that without using MAGC ($h = 0$). Compared to $h = 0$, $h = 8$ obtains performance improvement on all datasets, especially significant improvement on CUTE, IC15, and SVTP. These three datasets are difficult and irregular. We believe this phenomenon is due to the introduced MAGC module that can well capture different aspects of spatial 2D attention which is very important for irregular and hard text images. We also evaluate different settings $N = [1, 3, 6]$ of the number of fundamental blocks in the decoder part. $N = 3$ gets the best performance, and the performance of N=6 decreases a lot compared to $N = 3$. We reckon that too deep decoder layers may bring in convergence problems. Therefore, in our experiment, we use $N = 3$, $h = 8$ in default.

Table 4: Under different parameter settings our model recognition accuracy: h, N denotes the numbers of Multi-Aspect Context in the encoder and identical layers in the decoder, respectively. Standard Setting uses $h = 8$ and $N = 3$. When h or N changes, all other parameters keep the same as the Standard Setting. All values are reported as a percentage(%).

Methods	IIIT5k	SVT	CUTE	IC03	IC13	IC15	SVTP
Standard Setting: $h = 8$, $N = 3$	95.0	90.6	87.5	96.4	95.3	79.4	**84.5**
$h = 0$	94.6	90.1	86.2	95.9	95.0	78.4	82.3
$h = 1$	94.9	**91.5**	87.6	**96.9**	**95.7**	79.4	83.8
$h = 2$	94.93	90.7	**88.54**	96.6	95.4	79.5	84.0
$h = 4$	94.7	90.9	86.8	96.1	95.1	**79.6**	83.7
$h = 16$	**95.1**	91.3	85.4	96.0	95.3	79.4	84.1
$N = 1$	94.3	90.4	85.4	95.3	94.1	78.9	83.1
$N = 6$	91.3	87.4	76.7	94.3	91.6	72.9	75.7

Table 5: Speed comparison between MASTER (Ours) and SAR. MASTER is faster and more accurate than the SAR method. All timing information is on an NVIDIA Tesla V100 GPU.

Method	Input	Accuracy	Inference Time (ms)	Training Time (h)
SAR [8]	48×160	91.5	16.1	51
MASTER (original)	48×160	95.0	9.2	36
MASTER (improved)	48×160	95.0	4.3	36

4.4.2. Comparison of Evaluation Speed

We conduct a comparison of test speed on a server using an NVIDIA Tesla V100 GPU with Intel Xeon Gold 6130@ 2.10 GHz CPU. The results are averaged on 3,000 test images from IIIT-5K, the input image size is 48×160. The results of SAR is based on our implementation in PyTorch with the same setting as [8].

We observe from Table 5 that, MASTER not only achieves better performance but also runs faster than SAR. By stacking multiple test images together and

25

inputting the stacked batch in one time, we can obtain a speedup. The test time speed of our MASTER is **9.2 ms per image** compared to 16.1 ms of SAR. By using a new memory-cache based inference mechanism, the decoder can speed up to 4.3 ms from 9.2 ms. Besides, we also compare the training speed between MASTER and SAR. As shown in the last column of Table 5. The results show that MASTER has faster training speed because of the parallel training.

4.4.3. Model stability

We show the evaluation accuracies of MASTER and SAR along with training steps in Figure 4. We find that from Figure 4, the MASTER model achieves more stable recognition performance than SAR although SAR converges faster. We reckon the reason is the MASTER requires calculating global attention which is slower but SAR only needs to compute local attention. We can see that the performance of the MASTER model is very stable when it hits the best performance, it will not decrease a lot. However, the performance of SAR often decreases a little more when it reaches the best performance.
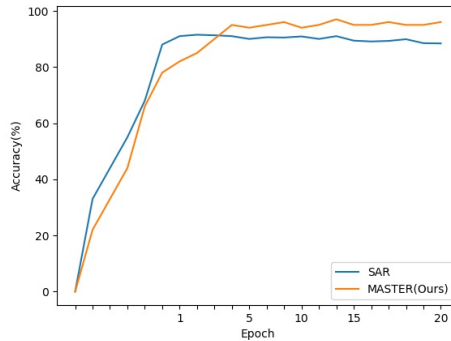
Figure 4: The model stability comparison between MASTER (Ours) and SAR [8].

5. Conclusions

In this work, we propose a novel approach MASTER: multi-aspect non-local network for scene text recognition. The MASTER consists of a Multi-Aspect Global Context Attention (GCAttention) based encoder module and a transformer-based decoder module. The proposed MASTER owns three advantages: (1) The model can both learn input-output attention and self-attention which encodes feature-feature and target-target relationships inside the encoder and the decoder. (2) Experiments demonstrate that the proposed method is more robust to spatial distortions. (3) The training process of the proposed method is highly parallel and efficient, and the inference speed is fast because of the proposed novel memory-cached decoding mechanism. Experiments on standard benchmarks demonstrate it can achieve state-of-the-art performances regarding both efficiency and recognition accuracy.

References

[1] J. Gu, Z. Wang, J. Kuen, L. Ma, A. Shahroudy, B. Shuai, T. Liu, X. Wang, G. Wang, J. Cai, T. Chen, Recent advances in convolutional neural networks, Pattern Recognit. 77 (2015) 354–377.

[2] C.-L. Liu, G. A. Fink, V. Govindaraju, L. Jin, Special issue on deep learning for document analysis and recognition, International Journal on Document Analysis and Recognition 21 (2018) 159–160.

[3] D. N. Van, S. Lu, S. Tian, N. Ouarti, M. Mokhtari, A pooling based scene text proposal technique for scene text reading in the wild, Pattern Recognit. 87 (2019) 118–129.

[4] P. Shivakumara, S. Bhowmick, B. Su, C. L. Tan, U. Pal, A New Gradient Based Character Segmentation Method for Video Text Recognition, in: Proceedings of International Conference on Document Analysis and Recognition, 2011, pp. 126–130.

[5] B. Shi, X. Bai, 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. 39 (2015) 2298–2304.

[6] F. Bai, Z. Cheng, Y. Niu, S. Pu, S. Zhou, Edit probability for scene text recognition, in: Proceedings of Computer Vision and Pattern Recognition, 2018, pp. 1508–1516.

[7] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, X. Bai, ASTER: An attentional scene text recognizer with flexible rectification, IEEE Trans. Pattern Anal. Mach. Intell. 41 (2018) 2035–2048.

[8] H. Li, P. Wang, C. Shen, G. Zhang, Show, Attend and Read: A simple and strong baseline for irregular text recognition, in: Proceedings of Association for the Advancement of Artificial Intelligence, 2019, pp. 8610–8617.

[9] F. L. Bookstein, Principal warps: thin-plate splines and the decomposition of deformations, IEEE Trans. Pattern Anal. Mach. Intell. 11 (6) (1989) 567–585.

[10] M. Yang, Y. Guan, M. Liao, X. He, K. Bian, S. Bai, C. Yao, X. Bai, Symmetry-constrained rectification network for scene text recognition, in: 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 9146–9155.

[11] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, C. Yao, X. Bai, Scene text recognition from two-dimensional perspective, in: AAAI, 2019.

[12] H. Xie, S. Fang, Z. Zha, Y. Yang, Y. Li, Y. Zhang, Convolutional attention networks for scene text recognition, ACM Trans. Multim. Comput. Commun. Appl. 15 (2019) 3:1–3:17.

[13] S. Fang, H. Xie, Z. Zha, N. Sun, J. Tan, Y. Zhang, Attention and language ensemble for scene text recognition with convolutional sequence modeling, Proceedings of the 26th ACM international conference on Multimedia (2018).

[14] K. Xu, J. Ba, R. Kiros, K. Cho, A. C. Courville, R. Salakhutdinov, R. S. Zemel, Y. Bengio, Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, in: Proceedings of International Conference on Machine Learning, 2015, pp. 2048–2057.

[15] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, I. Polosukhin, Attention is all you need, in: Proceedings of Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[16] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, in: Proceedings of The North American Chapter of the Association for Computational Linguistics, 2019.

[17] Z. Yang, Z. Dai, Y. Yang, J. G. Carbonell, R. Salakhutdinov, Q. V. Le, Xlnet: Generalized autoregressive pretraining for language understanding, in: Proceedings of Advances in Neural Information Processing Systems, 2019.

[18] J. Gehring, M. Auli, D. Grangier, D. Yarats, Y. N. Dauphin, Convolutional sequence to sequence learning, in: Proceedings of International Conference on Machine Learning, 2017, pp. 1243–1252.

[19] Y. Cao, J. Xu, S. Lin, F. Wei, H. Hu, GCNet: Non-local networks meet squeeze-excitation networks and beyond, in: Proceedings of International Conference on Computer Vision Workshop, 2019, pp. 1971–1980.

[20] H. Hu, J. Gu, Z. Zhang, J. Dai, Y. Wei, Relation networks for object detection, in: Proceedings of Computer Vision and Pattern Recognition, 2017, pp. 3588–3597.

[21] X. Wang, R. Girshick, A. Gupta, K. He, Non-local neural networks, in: Proceedings of Computer Vision and Pattern Recognition, 2018, pp. 7794–7803.

[22] J. Hu, L. Shen, G. Sun, Squeeze-and-excitation networks, in: Proceedings of Computer Vision and Pattern Recognition, 2017, pp. 7132–7141.

[23] Q. Ye, D. Doermann, Text detection and recognition in imagery: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 37 (7) (2015) 1480–1500.

[24] Y. Zhu, C. Yao, X. Bai, Scene text detection and recognition: recent advances and future trends, Frontiers of Computer Science 10 (1) (2016) 19–36.

[25] X.-Y. Zhang, Y. Bengio, C.-L. Liu, Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark, Pattern Recognit. 61 (2017) 348–360.

[26] A. P. Giotis, G. Sfikas, B. Gatos, C. Nikou, A survey of document image word spotting techniques, Pattern Recognit. 68 (2017) 310–332.

[27] A. Mishra, K. Alahari, C. Jawahar, Enhancing energy minimization framework for scene text recognition with top-down cues, Computer Vision and Image Understanding 145 (2016) 30–42.

[28] B. Su, S. Lu, Accurate recognition of words in scenes without character segmentation using recurrent neural network, Pattern Recognit. 63 (2017) 397–405.

[29] L. G. i Bigorda, D. Karatzas, Textproposals: A text-specific selective search algorithm for word spotting in the wild, Pattern Recognit. 70 (2016) 60–74.

[30] Y. Gao, Y. Chen, J. Wang, M. Tang, H. Lu, Reading scene text with fully convolutional sequence modeling, Neurocomputing 339 (2019) 161–170.

[31] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, S. Zhou, Focusing Attention: Towards accurate text recognition in natural images, in: Proceedings of International Conference on Computer Vision, 2017, pp. 5086–5094.

[32] Y. Zhang, S. Nie, W. Liu, X. Xu, D. Zhang, H. T. Shen, Sequence-to-sequence domain adaptation network for robust text image recognition, in: Proceedings of Computer Vision and Pattern Recognition, 2019, pp. 2740–2749.

[33] B. Shi, X. Wang, P. Lyu, C. Yao, X. Bai, Robust scene text recognition with automatic rectification, in: Proceedings of Computer Vision and Pattern Recognition, 2016, pp. 4168–4176.

[34] M. Jaderberg, K. Simonyan, A. Zisserman, K. Kavukcuoglu, Spatial transformer networks, in: Proceedings of Advances in Neural Information Processing Systems, 2015, pp. 2017–2025.

[35] F. Zhan, S. Lu, ESIR: End-to-end scene text recognition via iterative image rectification, in: Proceedings of Computer Vision and Pattern Recognition, 2019, pp. 2059–2068.

[36] C. Luo, L. Jin, Z. Sun, MORAN: A Multi-Object Rectified Attention Network for scene text recognition, Pattern Recognit. 90 (2019) 109–118.

[37] F. Sheng, Z. Chen, B. Xu, NRTR: A no-recurrence sequence-to-sequence model for scene text recognition, in: Proceedings of International Conference on Document Analysis and Recognition, 2018, pp. 781–786.

[38] J. Lee, S. Park, J. Baek, S. J. Oh, S. Kim, H. Lee, On recognizing texts of arbitrary shapes with 2d self-attention, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020, pp. 2326–2335.

[39] L. Yang, P. Wang, H. Li, Z. Li, Y. Zhang, A holistic representation guided attention network for scene text recognition, Neurocomputing (2020).

[40] J. Ba, J. R. Kiros, G. E. Hinton, Layer normalization, in: Advances in neural information processing systems (NIPS), 2016.

[41] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of Computer Vision and Pattern Recognition, 2015, pp. 770–778.

[42] M. Jaderberg, K. Simonyan, A. Vedaldi, A. Zisserman, Synthetic data and artificial neural networks for natural scene text recognition, in: Proceedings of Advances in Neural Information Processing Systems Workshop, 2014.

[43] A. Gupta, A. Vedaldi, A. Zisserman, Synthetic data for text localisation in natural images, in: Proceedings of Computer Vision and Pattern Recognition, 2016.

[44] A. Mishra, K. Alahari, C. V. Jawahar, Top-down and bottom-up cues for scene text recognition, in: Proceedings of Computer Vision and Pattern Recognition, 2012, pp. 2687–2694.

[45] K. Wang, B. Babenko, S. J. Belongie, End-to-end scene text recognition, in: Proceedings of International Conference on Computer Vision, 2011, pp. 1457–1464.

[46] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J.-M. Jolion, L. Todoran, M. Worring, X. Lin, ICDAR 2003 robust reading competitions: entries, results, and future directions, International Journal of Document Analysis and Recognition 7 (2004) 105–122.

[47] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. M. Romeu, D. F. Mota, J. Almazán, L.-P. de las Heras, ICDAR 2013 Robust Reading Competition, in: Proceedings of International Conference on Document Analysis and Recognition, 2013, pp. 1484–1493.

[48] T. Q. Phan, P. Shivakumara, S. Tian, C. L. Tan, Recognizing text with perspective distortion in natural scenes, in: Proceedings of International Conference on Computer Vision, 2013, pp. 569–576.

[49] A. Risnumawan, P. Shivakumara, C. S. Chan, C. L. Tan, A robust arbitrary text detection system for natural scene images, Expert Syst. Appl. 41 (2014) 8027–8048.

[50] M. Liao, P. Lyu, M. He, C. Yao, W. Wu, X. Bai, Mask TextSpotter: An end-to-end trainable neural network for spotting text with arbitrary shapes, IEEE Trans. Pattern Anal. Mach. Intell. (2019).

[51] W. Liu, C. Chen, K.-Y. K. Wong, Z. Su, J. Han, STAR-Net: A spatial attention residue network for scene text recognition, in: Proceedings of British Machine Vision Conference, 2016.

[52] J. Wang, X. Hu, Gated recurrent convolution neural network for ocr, in: Proceedings of Advances in Neural Information Processing Systems, 2017, pp. 335–344.

[53] Z. Liu, Y. Li, F. Ren, W. L. Goh, H. Yu, SqueezedText: A real-time scene text recognition by binary convolutional encoder-decoder network, in: Proceedings of Association for the Advancement of Artificial Intelligence, 2018.

[54] W. Liu, C. Chen, K.-Y. K. Wong, Char-Net: A character-aware neural network for distorted scene text recognition, in: Proceedings of Association for the Advancement of Artificial Intelligence, 2018.

[55] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, in: International Conference for Learning Representations, 2015.

[56] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, H. Lee, What is wrong with scene text recognition model comparisons? dataset and model analysis, in: Proceedings of International Conference on Computer Vision, 2019, pp. 4714–4722.