Read Like Humans: Autonomous, Bidirectional and Iterative Language Modeling for Scene Text Recognition

Shancheng Fang   Hongtao Xie*   Yuxin Wang   Zhendong Mao   Yongdong Zhang

University of Science and Technology of China

{fangsc, htxie, zdmao, zhyd73}@ustc.edu.cn, wangyx58@mail.ustc.edu.cn

Abstract

Linguistic knowledge is of great benefit to scene text recognition. However, how to effectively model linguistic rules in end-to-end deep networks remains a research challenge. In this paper, we argue that the limited capacity of language models comes from: 1) implicitly language modeling; 2) unidirectional feature representation; and 3) language model with noise input. Correspondingly, we propose an autonomous, bidirectional and iterative ABINet for scene text recognition. Firstly, the autonomous suggests to block gradient flow between vision and language models to enforce explicitly language modeling. Secondly, a novel bidirectional cloze network (BCN) as the language model is proposed based on bidirectional feature representation. Thirdly, we propose an execution manner of iterative correction for language model which can effectively alleviate the impact of noise input. Additionally, based on the ensemble of iterative predictions, we propose a self-training method which can learn from unlabeled images effectively. Extensive experiments indicate that ABINet has superiority on low-quality images and achieves state-of-the-art results on several mainstream benchmarks. Besides, the ABINet trained with ensemble self-training shows promising improvement in realizing human-level recognition. Code is available at https://github.com/FangShancheng/ABINet.

1. Introduction

Possessing the capability of reading text from scene images is indispensable to artificial intelligence [24, 41, 45]. To this end, early attempts regard characters as meaningless symbols and recognize the symbols by classification models [42, 15, 46]. However, when confronted with challenging environments such as occlusion, blur, noise, etc., it becomes faint due to out of visual discrimination. Fortunately, as text carries rich linguistic information, characters can be reasoned according to the context. Therefore, a bunch of methods [16, 14, 29, 48] turn their attention to language modeling and achieve undoubted improvement.

However, how to effectively model the linguistic behavior in human reading is still an open problem. From the observations of psychology, we can make three assumptions about human reading that language modeling is autonomous, bidirectional and iterative: 1) as both deaf-mute and blind people could have fully functional vision and language separately, we use the term *autonomous* to interpret the independence of learning between vision and language. The *autonomous* also implies a good interaction between vision and language that independently learned language knowledge could contribute to the recognition of characters in vision. 2) The action of reasoning character context behaves like cloze task since illegible characters can be viewed as blanks. Thus, prediction can be made using the cues of legible characters on the left side and right side of the illegible characters simultaneously, which is corresponding to the *bidirectional*. 3) The *iterative* describes that under the challenging environments, humans adopt a progressive strategy to improve prediction confidence by iteratively correcting the recognized results.

Firstly, applying the **autonomous** principle to scene text recognition (STR) means that recognition models should be decoupled into vision model (VM) and language model (LM), and the sub-models could be served as functional units independently and learned separately. Recent attention-based methods typically design LMs based on RNNs or Transformer [39], where the linguistic rules are learned *implicitly* within a coupled model [19, 36, 33] (Fig. 1a). Nevertheless, whether and how well the LMs learn character relationship is unknowable. Besides, this kind of methods is infeasible to capture rich prior knowledge by directly pre-training LM from large-scale unlabeled text.

Secondly, compared with the unidirectional LMs [38], LMs with **bidirectional** principle capture twice the amount of information. A straightforward way to construct a bidirectional model is to merge a left-to-right model and a right-to-left model [28, 5], either in probability-level [44, 36] or in feature-level [52] (Fig. 1e). However, they are strictly less powerful as their language features are unidirectional *repre-*

sentation in fact. Also, the ensemble models mean twice as expensive both in computations and parameters. A recent striking work in NLP is BERT [5], which introduces a deep bidirectional representation learned by masking text tokens. Directly applying BERT to STR requires masking all the characters within a text instance, whereas this is extremely expensive since each time only one character can be masked.

Thirdly, LMs executed with **iterative** principle can refine the prediction from visual and linguistic cues, which is not explored in current methods. The canonical way to perform an LM is auto-regression [44, 3, 47] (Fig. 1d), in which error recognition is accumulated as noise and taken as input for the following prediction. To adapt the Transformer architectures, [25, 52] give up auto-regression and adopt parallel-prediction (Fig. 1e) to improve efficiency. However, noise input still exists in parallel-prediction where errors from VM output directly harm the LM accuracy. In addition, parallel-prediction in SRN [52] suffers from unaligned-length problem that SRN is tough to infer correct characters if text length is wrongly predicted by VM.

Considering the deficiencies of current methods from the aspects of internal interaction, feature representation and execution manner, we propose ABINet guided by the principles of *Autonomous*, *Bidirectional* and *Iterative*. Firstly, we explore a decoupled method (Fig. 1b) by blocking gradient flow (BGF) between VM and LM, which enforces LM to learn linguistic rules explicitly. Besides, both VM and LM are autonomous units and could be pre-trained from images and text separately. Secondly, we design a novel bidirectional cloze network (BCN) as the LM, which eliminates the dilemma of combining two unidirectional models (Fig. 1c). The BCN is jointly conditioned on both left and right context, by specifying attention masks to control the accessing of both side characters. Also, accessing across steps is not allowed to prevent leaking information. Thirdly, we propose an execution manner of iterative correction for LM (Fig. 1b). By feeding the outputs of ABINet into LM repeatedly, predictions can be refined progressively and the unaligned-length problem could be alleviated to a certain extent. Additionally, treating the iterative predictions as an ensemble, a semi-supervised method is explored based on self-training, which exploits a new solution toward human-level recognition.

Contributions of this paper mainly include: 1) we propose autonomous, bidirectional and iterative principles to guide the design of LM in STR. Under these principles the LM is a functional unit, which is required to extract bidirectional representation and correct prediction iteratively. 2) A novel BCN is introduced, which estimates the probability distribution of characters like cloze tasks using bidirectional representation. 3) The proposed ABINet achieves state-of-the-art (SOTA) performance on mainstream benchmarks, and the ABINet trained with ensemble self-training shows promising improvement in realizing human-level recognition.
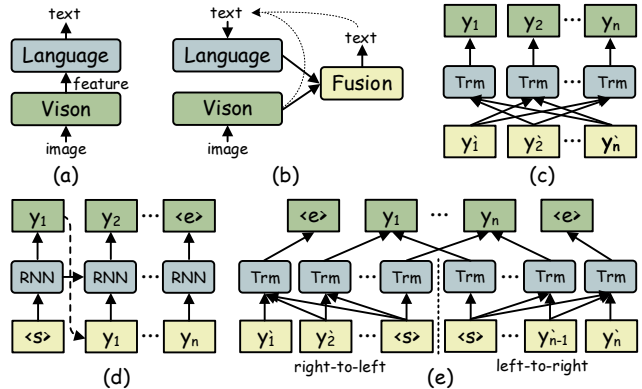

Figure 1. (a) Coupled language model. (b) Our autonomous language model with iterative correction. (c) Our bidirectional structure. (d) Unidirectional RNN in auto-regression. (e) Ensemble of two unidirectional Transformers in parallel-prediction.

2. Related Work

2.1. Language-free Methods

Language-free methods generally utilize visual features without the consideration of relationship between characters, such as CTC-based [7] and segmentation-based [21] methods. The CTC-based methods employ CNN to extract visual features and RNN to model features sequence. Then the CNN and RNN are trained end-to-end using CTC loss [34, 11, 37, 12]. The segmentation-based methods apply FCN to segment characters in pixel-level. Liao *et al*. recognize characters by grouping the segmented pixels into text regions. Wan *et al*. [40] propose an additional order segmentation map which transcripts characters in the correct order. Due to lacking of linguistic information, the language-free methods cannot resolve the recognition in low-quality images commendably.

2.2. Language-based Methods

Internal interaction between vision and language. In some early works, bags of N-grams of text string are predicted by a CNN which acts as an explicit LM [14, 16, 13]. After that the attention-based methods become popular, which implicitly models language using more powerful RNN [19, 36] or Transformer [43, 33]. The attention-based methods follow encoder-decoder architecture, where the encoder processes images and the decoder generates characters by focusing on relevant information from 1D image features [19, 35, 36, 3, 4] or 2D image features [51, 47, 23, 20]. For example, R²AM [19] employs recursive CNN as a feature extractor and LSTM as a learned LM implicitly modeling language in character-level, which avoids the use of N-grams. Further, this kind of methods is usually boosted by integrating a rectification module [36, 54, 50] for irregular images before feeding the images into networks. Different from the methods above, our method strives to build a more powerful LM by explicitly language modeling. In attempting
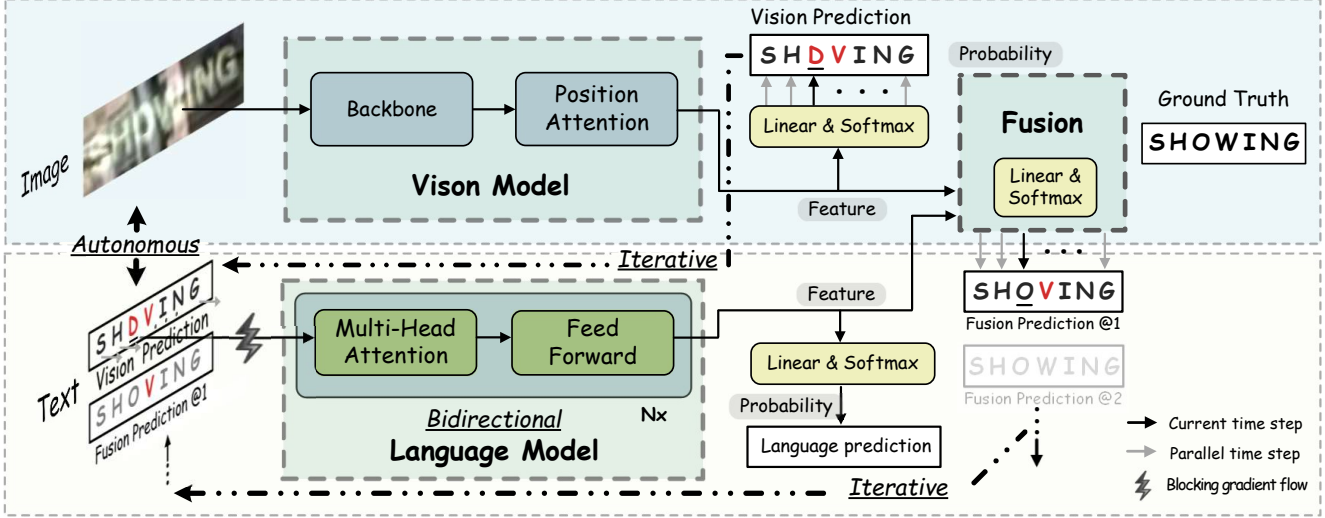
Figure 2. A schematic overview of ABINet.

to improve the language expression, some works introduce multiple losses where an additional loss comes from semantics [29, 25, 52, 6]. Among them, SEED [29] proposes to use pre-trained FastText model to guide the training of RNN, which brings extra semantic information. We deviate from this as our method directly pre-trains LM in unlabeled text, which is more feasible in practice.

Representation of language features. The character sequences in attention-based methods are generally modeled in left-to-right way [19, 35, 3, 40]. For instance, Textscanner [40] inherits the unidirectional model of attention-based methods. Differently, they employ an additional position branch to enhance positional information and mitigate misrecogniton in contextless scenarios. To utilize bidirectional information, methods like [8, 36, 44, 52] use an ensemble model of two unidirectional models. Specifically, to capture global semantic context, SRN [52] combines features from a left-to-right and a right-to-left Transformers for further prediction. We emphasize that the ensemble bidirectional model is intrinsically a unidirectional feature representation.

Execution manner of language models. Currently, the network architectures of LMs are mainly based on RNN and Transformer [39]. The RNN-based LM is usually executed in auto-regression [44, 3, 47], which takes the prediction of last character as input. Typical work such as DAN [44] obtains the visual features of each character firstly using proposed convolutional alignment module. After that GRU predicts each character by taking the prediction embedding of the last time step and the character feature of the current time step as input. The Transformer-based methods have superiority in parallel execution, where the inputs of each time step are either visual features [25] or character embedding from the prediction of visual feature [52]. Our method falls into parallel execution, but we try to alleviate the issue of noise input existing in parallel language model.
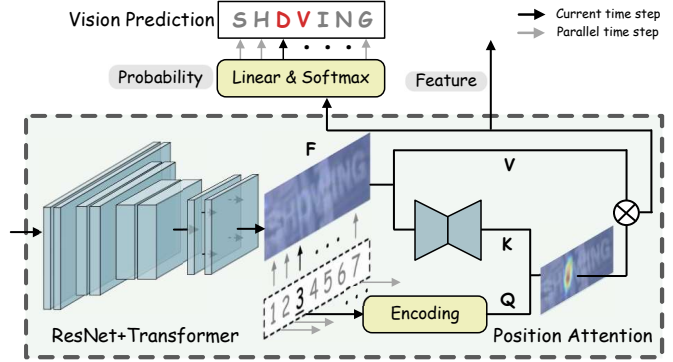

Figure 3. Architecture of vision model.

3. Proposed Method

3.1. Vision Model

The vision model consists of a backbone network and a position attention module (Fig. 3). Following the previous methods, ResNet[1] [36, 44] and Transformer units [52, 25] are employed as the feature extraction network and the sequence modeling network. For image x we have:

$$\mathbf{F}_b = \mathcal{T}(\mathcal{R}(\boldsymbol{x})) \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times C}, \tag{1}$$

where H, W are the size of $\boldsymbol{x}$ and C is feature dimension.

The module of position attention transcribes visual features into character probabilities in parallel, which is based on the query paradigm [39]:

$$\mathbf{F}_v = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^{\mathsf{T}}}{\sqrt{C}})\mathbf{V}. \tag{2}$$

Concretely, $\mathbf{Q} \in \mathbb{R}^{T \times C}$ is positional encodings [39] of character orders and T is the length of character sequence.

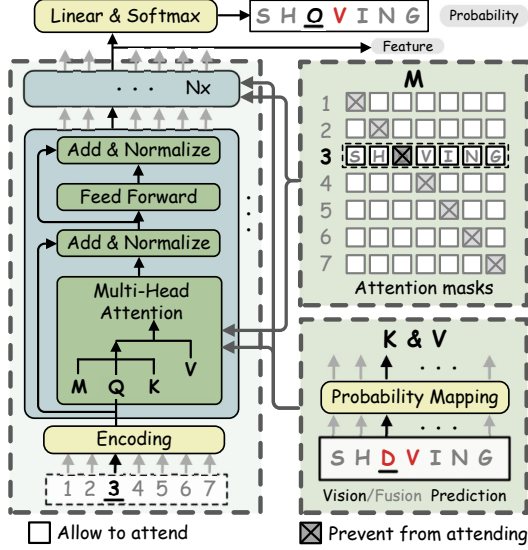[1]There are 5 residual blocks in total and down-sampling is performed after the 1st and 3nd blocks.

Figure 4. Architecture of language model (BCN).

$\mathbf{K} = \mathcal{G}(\mathbf{F}_b) \in \mathbb{R}^{\frac{HW}{16} \times C}$, where $\mathcal{G}(\cdot)$ is implemented by a mini U-Net[2] [32]. $\mathbf{V} = \mathcal{H}(\mathbf{F}_b) \in \mathbb{R}^{\frac{HW}{16} \times C}$, where $\mathcal{H}(\cdot)$ is identity mapping.

3.2. Language Model

3.2.1 Autonomous Strategy

As shown in Fig. 2, the autonomous strategy includes following characteristics: 1) the LM is regarded as an independent model of spelling correction which takes probability vectors of characters as input and outputs probability distributions of expected characters. 2) The flow of training gradient is blocked (BGF) at input vectors. 3) The LM could be trained separately from unlabeled text data.

Following the strategy of autonomous, the ABINet can be divided into interpretable units. By taking the probability as input, LM could be replaceable (*i.e.*, replaced with more powerful model directly) and flexible (*e.g.*, executed iteratively in Section 3.2.3). Besides, an important point is that BGF enforces model to learn linguistic knowledge inevitably, which is radically distinguished from implicitly modeling where what the models exactly learn is unknowable. Furthermore, the autonomous strategy allows us to directly share the advanced progresses in NLP community. For instance, pre-training the LM can be an effective way to boost the performance.

3.2.2 Bidirectional Representation

Given a text string $\boldsymbol{y} = (y_1, \ldots, y_n)$ with text length n and class number c, the conditional probability of y_i for bidirectional and unidirectional models are

---

[2]A network with 4-layer encoder, 64 channels, *add* fusion and interpolation upsample.

$P(y_i|y_n, \ldots, y_{i+1}, y_{i-1}, \ldots, y_1)$ and $P(y_i|y_{i-1}, \ldots, y_1)$, respectively. From the perspective of information theory, available entropy of a bidirectional representation can be quantified as $H_{\boldsymbol{y}} = (n-1)\log c$. However, for a unidirectional representation the information is $\frac{1}{n}\sum_{i=1}^{n}(i-1)\log c = \frac{1}{2}H_{\boldsymbol{y}}$. Our insight is that previous methods typically use an ensemble model of two unidirectional models, which essentially are unidirectional representations. The unidirectional representation basically captures $\frac{1}{2}H_{\boldsymbol{y}}$ information, resulting in limited capability of feature abstraction compared with bidirectional counterpart.

Benefitting from the autonomous design in Section 3.2.1, off-the-shelf NLP models with the ability of spelling correction can be transferred. A plausible way is utilizing the masked language model (MLM) in BERT [5] by replacing y_i with token [MASK]. However, we notice that this is unacceptable as MLM should be separately called n times for each text instance, causing extreme low efficiency. Instead of masking the input characters, we propose BCN by specifying the attention masks.

Overall, the BCN is a variant of L-layers transformer decoder. Each layer of BCN is a series of multi-head attention and feed-forward network [39] followed by residual connection [10] and layer normalization [1], as shown in Fig. 4. Different from vanilla Transformer, character vectors are fed into the multi-head attention blocks rather than the first layer of network. In addition, attention masks in multi-head attention are designed to prevent from "seeing itself". Besides, no self-attention is applied in BCN to avoid leaking information across time steps. The attention operation inside multi-head blocks can be formalized as:

$$\mathbf{M}_{ij} = \begin{cases} 0, & i \neq j \\ -\infty, & i = j \end{cases}, \tag{3}$$

$$\mathbf{K}_i = \mathbf{V}_i = P(y_i)\mathbf{W}_l, \tag{4}$$

$$\mathbf{F}_{mha} = \text{softmax}(\frac{\mathbf{QK}^\mathsf{T}}{\sqrt{C}} + \mathbf{M})\mathbf{V}, \tag{5}$$

where $\mathbf{Q} \in \mathbb{R}^{T \times C}$ is the positional encodings of character orders in the first layer and the outputs of the last layer otherwise. $\mathbf{K}, \mathbf{V} \in \mathbb{R}^{T \times C}$ are obtained from character probability $P(y_i) \in \mathbb{R}^c$, and $\mathbf{W}_l \in \mathbb{R}^{c \times C}$ is linear mapping matrix. $\mathbf{M} \in \mathbb{R}^{T \times T}$ is the matrix of attention masks which prevents from attending current character. After stacking BCN layers into deep architecture, the bidirectional representation $\mathbf{F}_l$ for text $\boldsymbol{y}$ is determined.

By specifying the attention masks in cloze fashion, BCN is able to learn more powerful bidirectional representation elegantly than the ensemble of unidirectional representation. Besides, benefitting from Transformer-like architecture, BCN can perform computation independently and parallelly. Also, it is more efficient than the ensemble models as only half of the computations and parameters are needed.

3.2.3 Iterative Correction

The parallel-prediction of Transformer takes noise inputs which are typically approximations from visual prediction [52] or visual feature [25]. Concretely, as the example shown in Fig. 2 under bidirectional representation, the desired condition for $P(\text{"O"})$ is "SH-WING". However, due to the blurred and occluded environments, the actual condition obtained from VM is "SH-VING", in which "V" becomes noise and harms the confidence of prediction. It tends to be more hostile for LM with increased error predictions in VM.

To cope with the problem of noise inputs, we propose iterative LM (illustrated in Fig. 2). The LM is executed M times repeatedly with different assignment for $\boldsymbol{y}$. For the first iteration, $\boldsymbol{y}_{i=1}$ is the probability prediction from VM. For the subsequent iterations, $\boldsymbol{y}_{i \geq 2}$ is the probability prediction from the fusion model (Section 3.3) in last iteration. By this way the LM is able to correct the vision prediction iteratively.

Another observation is that Transformer-based methods generally suffer from unaligned-length problem [52], which denotes that the Transformer is hard to correct the vision prediction if character number is unaligned with ground truth. The unaligned-length problem is caused by the inevitable implementation of padding mask which is fixed for filtering context outside text length. Our iterative LM can alleviate this problem as the visual feature and linguistic feature are fused several times, and thus the predicted text length is also refined gradually.

3.3. Fusion

Conceptually, vision model trained on image and language model trained on text come from different modalities. To align visual feature and linguistic feature, we simply use the gated mechanism [52, 53] for final decision:

$$\mathbf{G} = \sigma([\mathbf{F}_v, \mathbf{F}_l]\mathbf{W}_f), \qquad (6)$$

$$\mathbf{F}_f = \mathbf{G} \odot \mathbf{F}_v + (1 - \mathbf{G}) \odot \mathbf{F}_l, \qquad (7)$$

where $\mathbf{W}_f \in \mathbb{R}^{2C \times C}$ and $\mathbf{G} \in \mathbb{R}^{T \times C}$.

3.4. Supervised Training

ABINet is trained end-to-end using the following multi-task objectives:

$$\mathcal{L} = \lambda_v \mathcal{L}_v + \frac{\lambda_l}{M} \sum_{i=1}^{M} \mathcal{L}_l^i + \frac{1}{M} \sum_{i=1}^{M} \mathcal{L}_f^i, \qquad (8)$$

where $\mathcal{L}_v$, $\mathcal{L}_l$ and $\mathcal{L}_f$ are the cross entropy losses from $\mathbf{F}_v$, $\mathbf{F}_l$ and $\mathbf{F}_f$, respectively. Specifically, $\mathcal{L}_l^i$ and $\mathcal{L}_f^i$ are the losses at i-th iteration. λ_v and λ_l are balanced factors.

3.5. Semi-supervised Ensemble Self-training

To further explore the superiority of our iterative model, we propose a semi-supervised learning method based on

---

Algorithm 1 Ensemble Self-training

Require: Labeled images $\mathcal{X}$ with labels $\mathcal{Y}$ and unlabeled images $\mathcal{U}$
1: Train parameters θ_0 of ABINet with $(\mathcal{X}, \mathcal{Y})$ using Equation 8.
2: Use θ_0 to generate soft pseudo labels $\mathcal{V}$ for $\mathcal{U}$
3: Get $(\mathcal{U}', \mathcal{V}')$ by filtering $(\mathcal{U}, \mathcal{V})$ with $\mathcal{C} < Q$ (Equation 9)
4: **for** $i = 1, \ldots, N_{max}$ **do**
5:     **if** $i == N_{upl}$ **then**
6:        Update $\mathcal{V}$ using θ_i
7:        Get $(\mathcal{U}', \mathcal{V}')$ by filtering $(\mathcal{U}, \mathcal{V})$ with $\mathcal{C} < Q$ (Equation 9)
8:     **end if**
9:     Sample $B_l = (\mathcal{X}_b, \mathcal{Y}_b) \subsetneqq (\mathcal{X}, \mathcal{Y})$, $B_u = (\mathcal{U}'_b, \mathcal{V}'_b) \subsetneqq (\mathcal{U}', \mathcal{V}')$
10:     Update θ_i with B_l, B_u using Equation 8.
11: **end for**

---

self-training [49] with the ensemble of iterative predictions. The basic idea of self-training is first to generate pseudo labels by model itself, and then re-train the model using additional pseudo labels. Therefore, the key problem lies in constructing high-quality pseudo labels.

To filter the noise pseudo labels we propose the following methods: 1) minimum confidence of characters within a text instance is chosen as the text certainty. 2) Iterative predictions of each character are viewed as an ensemble to smooth the impact of noise labels. Therefore, we define the filtering function as follows:

$$\begin{cases} \mathcal{C} & = \min_{1 \leq t \leq T} e^{\mathbb{E}[\log P(y_t)]} \\ P(y_t) & = \max_{1 \leq m \leq M} P_m(y_t) \end{cases}, \qquad (9)$$

where $\mathcal{C}$ is the minimum *certainty* of a text instance, $P_m(y_t)$ is probability distribution of t-th character at m-th iteration. The training procedure is depicted in Algorithm 1, where Q is threshold. B_l, B_u are training batches from labeled and unlabeled data. N_{max} is the maximum number of training step and N_{upl} is the step number for updating pseudo labels.

4. Experiment

4.1. Datasets and Implementation Details

Experiments are conducted following the setup of [52] in the purpose of fair comparison. Concretely, the training datasets are two synthetic datasets MJSynth (MJ) [13, 15] and SynthText (ST) [9]. Six standard benchmarks include ICDAR 2013 (IC13) [18], ICDAR 2015 (IC15) [17], IIIT 5K-Words (IIIT) [27], Street View Text (SVT) [42], Street View Text-Perspective (SVTP) [30] and CUTE80 (CUTE) [31] are as the testing datasets. Details of these datasets can be found in the previous works [52]. In addition, Uber-Text [55] removing the labels is used as unlabeled dataset to evaluate the semi-supervised method.

The model dimension C is set to 512 throughout. There are 4 layers in BCN with 8 attention heads each layer. Balanced factors λ_v, λ_l are set to 1, 1 respectively. Images are directly resized to 32×128 with data augmentation such as geometry transformation (*i.e.*, rotation, affine and perspective), image quality deterioration and color jitter, *etc*. We use

Table 1. Ablation study of VM. Attn is the attention method and Trm Layer is the layer number of Transformer. SV, MV$_1$, MV$_2$ and LV are four VMs in different configurations.

Model Name	Attn	Trm Layer	IC13 IC15	SVT SVTP	IIIT CUTE	Avg	Params (×10^6)	Time[3] (ms)
SV (small)	parallel	2	94.2 80.6	89.6 82.3	93.7 85.1	88.8	19.6	12.5
MV$_1$ (middle)	position	2	93.6 80.8	89.3 83.1	94.2 85.4	89.0	20.4	14.9
MV$_2$ (middle)	parallel	3	94.5 81.1	89.5 83.7	94.3 **86.8**	89.4	22.8	14.8
LV (large)	position	3	**94.9** **81.7**	**90.4** **84.2**	**94.6** 86.5	**89.8**	23.5	16.7

Table 2. Ablation study of autonomous strategy. PVM is pre-training VM on MJ and ST in supervised way. PLM$_{in}$ is pre-training LM using text on MJ and ST in self-supervised way. PLM$_{out}$ is pre-training LM on WikiText-103 [26] in self-supervised way. AGF means allowing gradient flow between VM and LM.

PVM	PLM$_{in}$	PLM$_{out}$	AGF	IC13 IC15	SVT SVTP	IIIT CUTE	Avg
-	-	-	-	96.7 84.5	93.4 86.8	95.7 86.8	91.7
✓	-	-	-	97.0 85.0	93.0 88.5	96.3 89.2	92.3
-	✓	-	-	97.1 83.6	**93.8** 88.1	95.5 86.8	91.6
✓	✓	-	-	**97.2** 84.9	93.5 **89.0**	96.3 88.5	92.3
✓	-	✓	-	97.0 **85.3**	93.7 88.5	**96.5** **89.6**	**92.5**
✓	-	-	✓	96.7 83.3	92.6 86.5	95.7 88.5	91.4

4 NVIDIA 1080Ti GPUs to train our models with batch size 384. ADAM optimizer is adopted with the initial learning rate $1e^{-4}$, which is decayed to $1e^{-5}$ after 6 epochs.

4.2. Ablation Study

4.2.1 Vision Model

Firstly, we discuss the performance of VM from two aspects: feature extraction and sequence modeling. Experiment results are recorded in Tab. 1. The *parallel* attention is a popular attention method [25, 52], and the proposed *position* attention has a more powerful representation of key/value vectors. From the statistics we can conclude: 1) simply upgrading the VM will result in great gains in accuracy but at the cost of parameter and speed. 2) To upgrade the VM, we can use the position attention in feature extraction and a deeper transformer in sequence modeling.

4.2.2 Language Model

Autonomous Strategy. To analyze the autonomous models, we adopt the LV and BCN as VM and LM respectively. From the results in Tab. 2 we can observe: 1) pre-training VM is useful which boosts the accuracy about 0.6%-0.7% on average; 2) the benefit of pre-training LM on the training

Table 3. Ablation study of bidirectional representation.

Vision	Language	IC13 IC15	SVT SVTP	IIIT CUTE	Avg	Params (×10^6)	Time (ms)
SV	SRN-U	96.0 81.9	90.3 86.0	94.9 85.4	90.2	32.8	19.1
	SRN	96.3 82.6	90.9 **86.4**	95.0 87.5	90.6	45.4	24.2
	BCN	**96.7** **83.1**	**91.7** 86.2	**95.3** **88.9**	**91.0**	32.8	19.5
LV	SRN-U	96.0 84.0	91.2 86.8	96.2 87.8	91.5	36.7	22.1
	SRN	96.8 84.2	92.3 87.9	96.3 88.2	91.9	49.3	26.9
	BCN	**97.0** **85.0**	**93.0** **88.5**	96.3 **89.2**	**92.3**	36.7	22

Table 4. Top-5 accuracy of LMs in text spelling correction.

Language Model	Character Accuracy	Word Accuracy
SRN	78.3	27.6
BCN	**82.8**	**41.9**

datasets (*i.e.*, MJ and ST) is negligible; 3) while pre-training LM from an additional unlabeled dataset (*e.g.*, WikiText-103) is helpful even when the base model is in high accuracy. The above observations suggest that it is useful for STR to pre-train both VM and LM. Pre-training LM on additional unlabeled datasets is more effective than on training datasets since the limited text diversity and biased data distribution are unable to facilitate the learning of a well-performed LM. Also, pre-training LM on unlabeled datasets is cheap since additional data is available easily.

Besides, by allowing gradient flow (AGF) between VM and LM, the performance decreases 0.9% on average (Tab. 2. We also notice that the training loss of AGF reduces sharply to a lower value. This indicates that overfitting occurs in LM as the VM helps to cheat in training, which might also happen in implicitly language modeling. Therefore it is crucial to enforce LM to learn independently by BGF. We note that SRN [52] uses *argmax* operation after VM, which is intrinsically a special case of BGF since *argmax* is non-differentiable. Another advantage is that the autonomous strategy makes the models a better interpretability, since we can have a deep insight into the performance of LM (*e.g.*, Tab. 4), which is infeasible in implicitly language modeling.

Bidirectional Representation. As the BCN is a variant of Transformer, we compare BCN with its counterpart SRN. The Transformer-based SRN [52] shows superior performance which is an ensemble of unidirectional representation. For fair comparison, experiments are conducted with the same conditions except the networks. We use SV and LV as the VMs to validate the effectiveness at different accuracy levels. As depicted in Tab. 3, though BCN has similar parameters and inference speed as the unidirectional version of SRN (SRN-U), it achieves competitive advantage in accuracy under different VMs. Besides, compared with the bidirectional SRN in ensemble, BCN shows better performance especially on challenging datasets such as IC15 and CUTE. Also, ABINet equipped with BCN is about 20%-25% faster
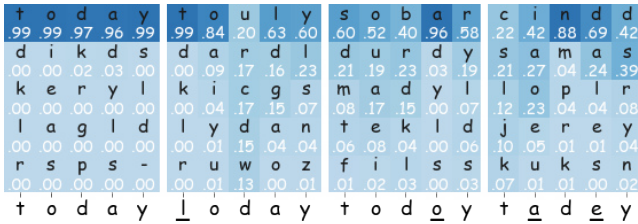
Figure 5. Visualization of top-5 probability in BCN.

Table 5. Ablation study of iterative correction.

Model	Iteration Number	IC13 IC15	SVT SVTP	IIIT CUTE	Avg	Params ($\times 10^6$)	Time (ms)
SV	1	96.7 83.1	91.7 86.2	95.3 88.9	91.0	32.8	19.5
+	2	**97.2** 83.3	91.8 86.4	**95.4** 89.2	91.2	32.8	24.5
BCN	3	97.1 **83.4**	**93.0** **86.7**	**95.4** **89.6**	**91.4**	32.8	31.6
LV	1	97.0 85.0	93.0 88.5	96.3 89.2	92.3	36.7	22
+	2	97.1 85.2	93.4 88.7	96.3 **89.6**	92.4	36.7	27.3
BCN	3	**97.3** **85.5**	**94.0** **89.1**	**96.4** 89.2	**92.6**	36.7	33.9

than SRN, which is practical for large-scale tasks.

Section 3.2.1 has argued that the LMs can be viewed as independent units to estimate the probability distribution of spelling correction, and thus we conduct experiments from this view. The training set is the text from MJ and ST. To simulate spelling errors, the testing set is 20000 items which are chosen randomly, where we add or remove a character for 20% text, replace a character for 60% text and keep the rest of the text unchangeable. From the results in Tab. 4, we can see BCN outperforms SRN by 4.5% character accuracy and 14.3% word accuracy, which indicates that BCN has a more powerful ability in character-level language modeling.

To better understand how BCN works inside ABINet, we visualize the top-5 probability in Fig. 5, which takes "today" as an example. On the one hand, as "today" is a string with semantic information, taking "-oday" and "tod-y" as inputs, BCN can predict "t" and "a" with high confidence and contribute to final fusion predictions. On the other hand, as error characters "l" and "o" are noise for the rest predictions, BCN becomes less confident and has little impact to final predictions. Besides, if there are multiple error characters, it is hard for BCN to restore correct text due to lacking of enough context.

Iterative Correction. We apply SV and LV again with BCN to demonstrate the performance of iterative correction from different levels. Experiment results are given in Tab. 5, where the iteration numbers are set to 1, 2 and 3 both in training and testing. As can be seen from the results, iterating the BCN 3 times can respectively boost the accuracy by 0.4%, 0.3% on average. Specifically, there are little gains on IIIT which is a relatively easy dataset with clear character appearance. However, when it comes to other harder datasets
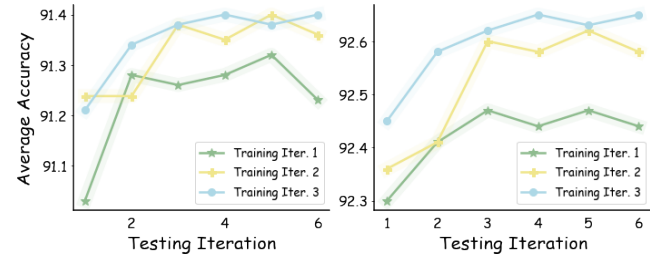

Figure 6. Accuracy of iterating BCN in training and testing.

betty bet**in** | jeanswear **is**answear | bookstore books**s**tre | earning earni_g
beti**y** betty | **ia**answear jeanswear | bookst**t**re bookstore | earnin_ earning

christmas ch**y**ustmaa | school se**h**dol | belgium bel**y**jum | available availall_
chr**u**stmaa christmas | sc**n**ool school | bel**y**ium belgium | availabl_ available

Figure 7. Successful examples using iterative correction. Text strings are ground truth, vision prediction, fusion prediction without iterative correction and with iterative correction respectively from left to right and top to bottom.

such as IC15, SVT and SVTP, the iterative correction steadily increases the accuracy and achieves up to 1.3% and 1.0% improvement on SVT for SV and LV respectively. It is also noted that the inference time increases linearly with the iteration number.

We further explore the difference of iteration between training and testing. The fluctuation of average accuracy in Fig. 6 suggests that: 1) directly applying iterative correction in testing also works well; 2) while iterating in training is beneficial since it provides additional training samples for LM; 3) the accuracy reaches a saturated state when iterating the model more than 3 times, and therefore a big iteration number is unnecessary.

To have a comprehensive cognition about iterative correction, we visualize the intermediate predictions in Fig. 7. Typically, the vision predictions can be revised approaching to ground truth while remain errors in some cases. After multiple iterations, the predictions can be corrected finally. Besides, we also observe that iterative correction is able to alleviate the unaligned-length problem, as shown in the last column in Fig. 7.

From the ablation study we can conclude: 1) the bidirectional BCN is a powerful LM which can effectively improve the performance both in accuracy and speed. 2) By further equipping BCN with iterative correction, the noise input problem can be alleviated, which is recommended to deal with challenging examples such as low-quality images at the expense of incremental computations.

4.3. Comparisons with State-of-the-Arts

Generally, it is not an easy job to fairly compare with other methods directly using the reported statistics [2], as differences might exist in backbone (*i.e.*, CNN structure and

Table 6. Accuracy comparison with other methods.

	Methods	Labeled Datasets	Unlabeled Datasets	Regular Text			Irregular Text		
				IC13	SVT	IIIT	IC15	SVTP	CUTE
SOTA methods	2019 Lyu *et al.* [25] (Parallel)	MJ+ST	-	92.7	90.1	94.0	76.3	82.3	86.8
	2019 Liao *et al.* [22] (SAM)	MJ+ST	-	95.3	90.6	93.9	77.3	82.2	87.8
	2020 Qiao *et al.* [29] (SE-ASTER)	MJ+ST	-	92.8	89.6	93.8	80.0	81.4	83.6
	2020 Wan *et al.* [40] (Textscanner)	MJ+ST	-	92.9	90.1	93.9	79.4	84.3	83.3
	2020 Wang *et al.* [44] (DAN)	MJ+ST	-	93.9	89.2	94.3	74.5	80.0	84.4
	2020 Yue *et al.* [53] (RobustScanner)	MJ+ST	-	94.8	88.1	95.3	77.1	79.5	**90.3**
	2020 Yu *et al.* [52] (SRN)	MJ+ST	-	95.5	91.5	94.8	82.7	85.1	87.8
Ours	SRN-SV (Reproduced)	MJ+ST	-	96.3	90.9	95.0	82.6	86.4	87.5
	ABINet-SV	MJ+ST	-	**96.8**	**93.2**	**95.4**	**84.0**	**87.0**	88.9
	SRN-LV (Reproduced)	MJ+ST	-	96.8	92.3	**96.3**	84.2	87.9	88.2
	ABINet-LV	MJ+ST	-	**97.4**	**93.5**	96.2	**86.0**	**89.3**	**89.2**
	ABINet-LV$_{st}$	MJ+ST	Uber-Text	97.3	94.9	96.8	**87.4**	**90.1**	93.4
	ABINet-LV$_{est}$	MJ+ST	Uber-Text	**97.7**	**95.5**	**97.2**	86.9	89.9	**94.1**

Figure 8. Hard examples successfully recognized by ABINet-LV$_{est}$.

parameters), data processing (*i.e.*, images rectification and data augmentation) and training tricks, *etc*. To strictly perform fair comparison, we reproduce the SOTA algorithm SRN which shares the same experiment configuration with ABINet, as presented in Tab. 6. The two reimplemented SRN-SV and SRN-LV are slightly different from the reported model by replacing VMs, removing the side-effect of multi-scales training, applying decayed learning rate, etc. Note that SRN-SV performs somewhat better than SRN due to the above tricks. As can be seen from the comparison, our ABINet-SV outperforms SRN-SV with 0.5%, 2.3%, 0.4%, 1.4%, 0.6%, 1.4% on IC13, SVT, IIIT, IC15, SVTP and CUTE datasets respectively. Also, the ABINet-LV with a more strong VM achieve an improvement of 0.6%, 1.2%, 1.8%, 1.4%, 1.0% on IC13, SVT, IC15, SVTP and CUTE benchmarks over its counterpart.

Compared with recent SOTA works that are trained on MJ and ST, ABINet also shows impressive performance (Tab. 6). Especially, ABINet has prominent superiority on SVT, SVTP and IC15 as these datasets contain a large amount of low-quality images such as noise and blurred images, which the VM is not able to confidently recognize. Besides, we also find that images with unusual-font and irregular text can be successfully recognized as the linguistic information acts as an important complement to visual feature. Therefore ABINet can obtain second best result on CUTE even without image rectification.

4.4. Semi-Supervised Training

To further push the boundary of accurate reading, we explore a semi-supervised method which utilizes MJ and ST as the labeled datasets and Uber-Text as the unlabeled dataset. The threshold Q in Section 3.5 is set to 0.9, and the batch size of B_l and B_u are 256 and 128 respectively. Experiment results in Tab. 6 show that the proposed self-training method ABINet-LV$_{st}$ can easily outperform ABINet-LV on all benchmark datasets. Besides, the ensemble self-training ABINet-LV$_{est}$ shows a more stable performance by improving the efficiency of data utilization. Observing the boosted results we find that hard examples with scarce fonts and blurred appearance can also be recognized frequently (Fig. 8), which suggests that exploring the semi-/unsupervised learning methods is a promising direction for scene text recognition.

5. Conclusion

In this paper, we propose ABINet which explores effective approaches for utilizing linguistic knowledge in scene text recognition. The ABINet is 1) autonomous that improves the ability of language model by enforcing learning explicitly; 2) bidirectional that learns text representation by jointly conditioning on character context at both sides; and 3) iterative that corrects the prediction progressively to alleviate the impact of noise input. Based on the ABINet we further propose an ensemble self-training method for semi-supervised learning. Experiment results on standard benchmarks demonstrate the superiority of ABINet especially on low-quality images. In addition, we also claim that exploiting unlabeled data is possible and promising for achieving human-level recognition.

Acknowledgments

References

[1] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 4

[2] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *ICCV*, pages 4715–4723, 2019. 7

[3] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *ICCV*, pages 5076–5084, 2017. 2, 3

[4] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Aon: Towards arbitrarily-oriented text recognition. In *CVPR*, pages 5571–5579, 2018. 2

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186, 2019. 1, 2, 4

[6] Shancheng Fang, Hongtao Xie, Zheng-Jun Zha, Nannan Sun, Jianlong Tan, and Yongdong Zhang. Attention and language ensemble for scene text recognition with convolutional sequence modeling. In *Proceedings of the 26th ACM international conference on Multimedia*, pages 248–256, 2018. 3

[7] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, pages 369–376, 2006. 2

[8] Alex Graves, Marcus Liwicki, Santiago Fernández, Roman Bertolami, Horst Bunke, and Jürgen Schmidhuber. A novel connectionist system for unconstrained handwriting recognition. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):855–868, 2008. 3

[9] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 5

[10] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 4

[11] Pan He, Weilin Huang, Yu Qiao, Chen Change Loy, and Xiaoou Tang. Reading scene text in deep convolutional sequences. In *AAAI*, 2016. 2

[12] Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In *AAAI*, pages 11005–11012, 2020. 2

[13] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *NIPS Deep Learning Workshop*, 2014. 2, 5

[14] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep structured output learning for unconstrained text recognition. In *ICLR*, 2015. 1, 2

[15] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional

[16] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Deep features for text spotting. In *ECCV*, pages 512–528. Springer, 2014. 1, 2

[17] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, pages 1156–1160. IEEE, 2015. 5

[18] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *ICDAR*, pages 1484–1493. IEEE, 2013. 5

[19] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *CVPR*, pages 2231–2239, 2016. 1, 2, 3

[20] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, volume 33, pages 8610–8617, 2019. 2

[21] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *CVPR*, pages 2359–2367, 2017. 2

[22] Minghui Liao, Pengyuan Lyu, Minghang He, Cong Yao, Wenhao Wu, and Xiang Bai. Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. *IEEE transactions on pattern analysis and machine intelligence*, 2019. 8

[23] Minghui Liao, Jian Zhang, Zhaoyi Wan, Fengming Xie, Jiajun Liang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Scene text recognition from two-dimensional perspective. In *AAAI*, volume 33, pages 8714–8721, 2019. 2

[24] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, pages 1–24, 2020. 1

[25] Pengyuan Lyu, Zhicheng Yang, Xinhang Leng, Xiaojun Wu, Ruiyu Li, and Xiaoyong Shen. 2d attentional irregular scene text recognizer. *arXiv preprint arXiv:1906.05708*, 2019. 2, 3, 5, 6, 8

[26] Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*, 2016. 6

[27] Anand Mishra, Karteek Alahari, and CV Jawahar. Scene text recognition using higher order language priors. In *British Machine Vision Conference (BMVC)*, 2012. 5

[28] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of NAACL-HLT*, pages 2227–2237, 2018. 1

[29] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *CVPR*, pages 13528–13537, 2020. 1, 3, 8

neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016. 1, 5

[30] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *ICCV*, pages 569–576, 2013. 5

[31] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014. 5

[32] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241, 2015. 4

[33] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *ICDAR*, pages 781–786. IEEE, 2019. 1, 2

[34] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016. 2

[35] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *CVPR*, pages 4168–4176, 2016. 2, 3

[36] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018. 1, 2, 3

[37] Bolan Su and Shijian Lu. Accurate recognition of words in scenes without character segmentation using recurrent neural network. *Pattern Recognition*, 63:397–405, 2017. 2

[38] Martin Sundermeyer, Ralf Schlüter, and Hermann Ney. Lstm neural networks for language modeling. In *Thirteenth annual conference of the international speech communication association*, 2012. 1

[39] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017. 1, 3, 4

[40] Zhaoyi Wan, Mingling He, Haoran Chen, Xiang Bai, and Cong Yao. Textscanner: Reading characters in order for robust scene text recognition. 2020. 2, 3, 8

[41] Zhaoyi Wan, Jielei Zhang, Liang Zhang, Jiebo Luo, and Cong Yao. On vocabulary reliance in scene text recognition. In *CVPR*, pages 11425–11434, 2020. 1

[42] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464. IEEE, 2011. 1, 5

[43] Peng Wang, Lu Yang, Hui Li, Yuyan Deng, Chunhua Shen, and Yanning Zhang. A simple and robust convolutional-attention network for irregular text recognition. *arXiv preprint arXiv:1904.01375*, 6, 2019. 2

[44] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. In *AAAI*, pages 12216–12224, 2020. 1, 2, 3, 8

[45] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Youliang Tian, Zilong Fu, and Yongdong Zhang. R-net: A relationship network for efficient and accurate scene text detection. *IEEE Transactions on Multimedia*, 2020. 1

[46] Yuxin Wang, Hongtao Xie, Zheng-Jun Zha, Mengting Xing, Zilong Fu, and Yongdong Zhang. Contournet: Taking a further step toward accurate arbitrary-shaped scene text detection. In *CVPR*, pages 11753–11762, 2020. 1

[47] Zbigniew Wojna, Alexander N Gorban, Dar-Shyang Lee, Kevin Murphy, Qian Yu, Yeqing Li, and Julian Ibarz. Attention-based extraction of structured information from street view imagery. In *ICDAR*, volume 1, pages 844–850. IEEE, 2017. 2, 3

[48] Hongtao Xie, Shancheng Fang, Zheng-Jun Zha, Yating Yang, Yan Li, and Yongdong Zhang. Convolutional attention networks for scene text recognition. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*, 15(1s):1–17, 2019. 1

[49] Qizhe Xie, Minh-Thang Luong, Eduard Hovy, and Quoc V Le. Self-training with noisy student improves imagenet classification. In *CVPR*, pages 10687–10698, 2020. 5

[50] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In *ICCV*, pages 9147–9156, 2019. 2

[51] Xiao Yang, Dafang He, Zihan Zhou, Daniel Kifer, and C Lee Giles. Learning to read irregular text with attention mechanisms. In *IJCAI*, volume 1, page 3, 2017. 2

[52] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, pages 12113–12122, 2020. 1, 2, 3, 5, 6, 8

[53] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *ECCV*, 2020. 5, 8

[54] Fangneng Zhan and Shijian Lu. Esir: End-to-end scene text recognition via iterative image rectification. In *CVPR*, pages 2059–2068, 2019. 2

[55] Ying Zhang, Lionel Gueguen, Ilya Zharkov, Peter Zhang, Keith Seifert, and Ben Kadlec. Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In *SUNw: Scene Understanding Workshop - CVPR 2017*, 2017. 5