SAFL: A Self-Attention Scene Text Recognizer with Focal Loss

Bao Hieu Tran *, Thanh Le-Cong *, Huu Manh Nguyen, Duc Anh Le, Thanh Hung Nguyen , Phi Le Nguyen†
School of Information and Communication Technology
Hanoi University of Science and Technology
Hanoi, Vietnam
{hieu.tb167182, thanh.ld164834, manh.nh166428, anh.nd160126}@sis.hust.edu.vn, {lenp, hungnt}@soict.hust.edu.vn

***Abstract*—In the last decades, scene text recognition has gained worldwide attention from both the academic community and actual users due to its importance in a wide range of applications. Despite achievements in optical character recognition, scene text recognition remains challenging due to inherent problems such as distortions or irregular layout. Most of the existing approaches mainly leverage recurrence or convolution-based neural networks. However, while recurrent neural networks (RNNs) usually suffer from slow training speed due to sequential computation and encounter problems as vanishing gradient or bottleneck, CNN endures a trade-off between complexity and performance. In this paper, we introduce SAFL, a self-attention-based neural network model with the focal loss for scene text recognition, to overcome the limitation of the existing approaches. The use of focal loss instead of negative log-likelihood helps the model focus more on low-frequency samples training. Moreover, to deal with the distortions and irregular texts, we exploit Spatial TransformerNetwork (STN) to rectify text before passing to the recognition network. We perform experiments to compare the performance of the proposed model with seven benchmarks. The numerical results show that our model achieves the best performance.**

***Index Terms*—Scene Text Recognition, Self-attention, Focal loss**

I. Introduction

In recent years, text recognition has attracted the attention of both academia and actual users due to its application on various domains such as translation in mixed reality, autonomous driving, or assistive technology for the blind. Text recognition can be classified into two main categories: scanned document recognition and scene text recognition. While the former has achieved significant advancements, the latter remains challenging due to scene texts' inherent characteristics such as the distortion and irregular shapes of the texts. Recent methods in scene text recognition are inspired by the success of deep learning-based recognition models. Generally, these methods can be classified in two approaches: recurrent neural networks (RNN) based and convolutional neural networks (CNN) based. RNN-based models have shown their effectiveness, thanks to capturing contextual information and dependencies between different patches. However, RNNs typically compute along with the symbol positions of the input and output sequences, which cannot be performed in parallel fashion, thus leads to high training time. Furthermore, RNNs also encounter problems such as vanishing gradient [1] or bottleneck [2]. CNN-based approach, which allows computing the hidden representation parallelly, have been proposed to speed up the training procedure. However, to capture the dependencies between distant patches in long input sequences, CNN models require stacking more convolutional layers, which significantly increases the network's complexity. Therefore, CNN-based methods suffer the trade-off between complexity and accuracy. To remedy these limitations, in natural language processing (NLP) fields, a self-attention based mechanism named transformer [3] has been proposed. In the transformer, dependencies between different input and output positions are captured using a self-attention mechanism instead of sequential procedures in RNN. This mechanism allows more computation parallelization with higher performance. In the computer vision domain, some research have leveraged the transformer architecture and showed the effectiveness of some problems [4] [5]

Inspired by the transformer network, in this paper, we propose a self-attention based scene text recognizer with focal loss, namely as SAFL. Moreover, to tackle irregular shapes of scene texts, we also exploit a text rectification named Spatial Transformer Network (STN) to enhance the quality of text before passing to the recognition network. SAFL, as depicted in Figure 1, contains three components: rectification, feature extraction, and recognition. First, given an input image, the rectification network, built based on the Spatial Transformer Network (STN) [6], transforms the image to rectify its text. Then, the features of the rectified image are extracted using a convolutional neural network. Finally, a self-attention based recognition network is applied to predict the output character sequence. Specifically, the recognition network is an encoder-decoder model, where the encoder utilizes multi-head self-attention to transform input sequence to hidden feature representation, then the decoder applies another multi-head self-attention to output character sequence. To balance the training data for improving the prediction accuracy, we exploit focal loss instead of negative log-likelihood as in most recent works [7] [8].

To evaluate our proposed model's performance, we train SAFL with two synthetic datasets: Synth90k [9] and SynthText [10], and compare its accuracy with standard benchmarks,

* Authors contribute equally
† Corresponding author

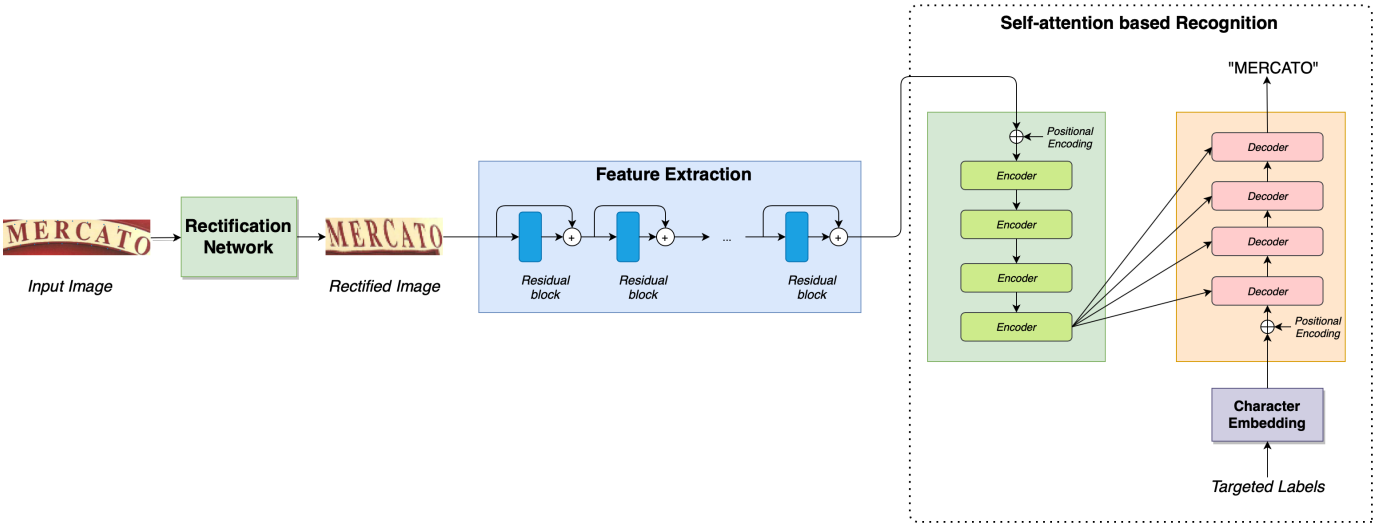

Fig. 1. Overview of SAFL

on both regular and irregular datasets. The experiment results show that our method outperforms the state-of-the-art on all datasets. Furthermore, we also perform experiments to study the effectiveness of focal loss. The numerical results show the superiority of focal loss over the negative log-likelihood loss on all datasets.

The remainder of the paper is organized as follows. Section II introduces related works. We describe the details of the proposed model in Section III and present the evaluation results in Section IV. Finally, we conclude the paper and discuss the future works in Section V.

II. Related Work

Scene text recognition has attracted great interest over the past few years. Comprehensive surveys for scene text recognition may be found in [11] [12] [13]. As categorized by previous works [8] [14] [15], scene text may be divided into two categories: regular and irregular text. The regular text usually has a nearly horizontal shape, while the irregular text has an arbitrary shape, which may be distorted.

A. Regular text recognition

Early work mainly focused on regular text and used a bottom-up scheme, which first detects individual characters using a sliding window, then recognizing the characters using dynamic programming or lexicon search [16] [17] [18]. However, these methods have an inherent limitation, which is ignoring contextual dependencies between characters. Shi et al. [19] and He et al. [20] typically regard text recognition as a sequence-to-sequence problem. Input images and output texts are typically represented as patch sequences and character sequences, respectively. This technique allows leveraging deep learning techniques such as RNNs or CNNs to capture contextual dependencies between characters [7] [19] [20], lead to significant improvements in accuracy on standard benchmarks. Therefore, recent work has shifted focus to the irregular text, a more challenging problem of scene text recognition.

B. Irregular text recognition

Irregular text is a recent challenging problem of scene text recognition, which refers to texts with perspective distortions and arbitrary shape. The early works correct perspective distortions by using hand-craft features. However, these approaches require correct tunning by expert knowledge for achieving the best results because of a large variety of hyperparameters. Recently, Yang et al. [21] proposed an auxiliary dense character detection model and an alignment loss to effectively solve irregular text problems. Liu et al. [22] introduced a Character-Aware Neural Network (Char-Net) to detect and rectify individual characters. Shi et al. [7] [8] addressed irregular text problems with a rectification network based on Spatial Transformer Network (STN), which transform input image for better recognition. Zhan et al. [23] proposed a rectification network employing a novel line-fitting transformation and an iterative rectification pipeline for correction of perspective and curvature distortions of irregular texts.

III. Proposed Model

Figure 1 shows the structure of SAFL, which is comprised of three main components: rectification, feature extraction, and recognition. The rectification module is a Spatial Transformer Network (STN) [6], which receives the original image and rectifies the text to enhance the quality. The feature extraction module is a convolution neural network that extracts the information of the rectified image and represents it into a vector sequence. The final module, i.e., recognition, is based on the self-attention mechanism and the transformer network architecture [3], to predict character sequence from the feature sequence. In the following, we first present the details of the three components in Section III-A, III-B and III-C, respectively. Then, we describe the training strategy using focal loss in Section III-D.

A. Rectification

In this module, we leverage a Thin Plate Spline (TPS) transformation [8], a variant of STN, to construct a rectification network. Given the input image I with an arbitrary size, the rectification module first resizes I into a predefined fixed size. Then the module detects several control points along the top and bottom of the text's bounding. Finally, TPS applies a smooth spline interpolation between a set of control points to rectify the predicted region to obtain a fixed-size image.

B. Feature Extraction

We exploit the convolution neural network (CNN) to extract the features of the rectified image (obtained from a rectification network) into a sequence of vectors. Specifically, the input image is passed through convolution layers (ConvNet) to produce a feature map. Then, the model separates the feature map by rows. The output received after separating the feature map are feature vectors arranged in sequences. The scene text recognition problem then becomes a sequence-to-sequence problem whose input is a sequence of characteristic vectors, and whose output is a sequence of characters predicted. Based on the proposal in [3], we further improve information about the position of the text in the input image by using positional encoding. Each position pos is represented by a vector whose value of the i^{th} dimension, i.e., $PE_{(pos,i)}$, is defined as

$$PE_{(pos,i)} = \begin{cases} \sin \frac{pos}{10000^{\frac{2i}{d_{model}}}}, & \text{if } 0 \leq i \leq \frac{d_{model}}{2} \\ \cos \frac{pos}{10000^{\frac{2i}{d_{model}}}}, & \text{if } \frac{d_{model}}{2} \leq i \leq d_{model}, \end{cases} \tag{1}$$

where d_{model} is the vector size. The position information is added into the encoding vectors.

C. Self-attention based recognition network

The architecture of the recognition network follows the encoder-decoder model. Both encoder blocks and decoder blocks are built based on the self-attention mechanism. We will briefly review this mechanism before describing each network's details.

1) Self-attention mechanism: Self-attention is a mechanism that extracts the correlation between different positions of a single sequence to compute a representation of the sequence. In this paper, we utilize the scaled dot-product attention proposed in [3]. This mechanism consists of queries and keys of dimension d_k, and values of dimension d_v. Each query performs the dot product of all keys to obtain their correlation. Then, we obtain the weights on the values by using the softmax function. In practice, the keys, values, and queries are also packed together into matrices K, V and Q. The matrix of the outputs is computed as follow:

$$\textit{Attention}(Q, K, V) = \textit{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{2}$$

The dot product is scaled by $\frac{1}{\sqrt{d_k}}$ to alleviate the small softmax values which lead to extremely small gradients with large values of d_k. [3].

2) Encoder: Encoder is a stack of N_e blocks. Each block consists of two main layers. The first layer is a multi-head attention layer, and the second layer is a fully-connected feed-forward layer. The multi-head attention layer is the combination of multiple outputs of the scale dot product attention. Each scale-dot product attention returns a matrix representing the feature sequences, which is called head attention. The combination of multiple head attentions to the multi-head attention allows our model to learn more representations of feature sequences, thereby increasing the diversity of the extracted information, and thereby enhance the performance. Multi-head attention can be formulated as follows:

$$MultiHead(Q, K, V) = Concat\left(head_1, \ldots, head_h\right) W^O \tag{3}$$

where $head_i = Attention\left(QW_i^Q, KW_i^K, VW_i^V\right)$, h is the number of heads, $W_i^Q \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^K \in \mathbb{R}^{d_{\text{model}} \times d_k}$, $W_i^V \in \mathbb{R}^{d_{\text{model}} \times d_v}$, $W^O \in \mathbb{R}^{hd_v \times d_{\text{model}}}$ are weight matrices. d_k, d_v and d_{model} are set to the same value. Layer normalization [24] and residual connection [25] are added into each main layer (i.e., multi-head attention layer and fully-connected layer) to improve the training effect. Specifically, the residual connections helps to decrease the loss of information in the backpropagation process, while the normalization makes the training process more stable. Consequently, the output of each main layer with the input x can be represented as $LayerNorm(x + \text{Layer}(x))$, where $Layer(x)$ is the function implemented by the layer itself, and $LayerNorm()$ represents the normalization operation. The blocks of the encoder are stacked sequentially, i.e., the output of the previous block is the input of the following block.

3) Decoder: The decoding process predicts the words in a sentence from left to right, starting with the $\langle start \rangle$ tag until encountering the $\langle end \rangle$ tag. The decoder is comprised of N_d decoder blocks. Each block is also built based on multi-head attention and a fully connected layer. The multi-head attention in the decoder does not consider words that have not been predicted by weighting these positions with $-\infty$. Furthermore, the decoder uses additional multi-head attention that receives keys and values from the encoder and queries from the decoder. Finally, the decoder's output is converted into a probability distribution through a linear transformation and softmax function.

D. Training

Figure 2 shows that the lexicon of training datasets suffers from an unbalanced sample distribution. The unbalance may lead to severe overfitting for high-frequency samples and underfitting for low-frequency samples. To this end, we propose to use focal loss [26] instead of negative log-likelihood as in most of recent methods [7] [8]. By exploiting focal loss, the model will not encounter the phenomenon of ignoring to train low-frequency samples.

Focal loss is known as an effective loss function to address the unbalance of datasets. By reshaping the standard cross-entropy loss, focal loss reduces the impacts of high-frequency

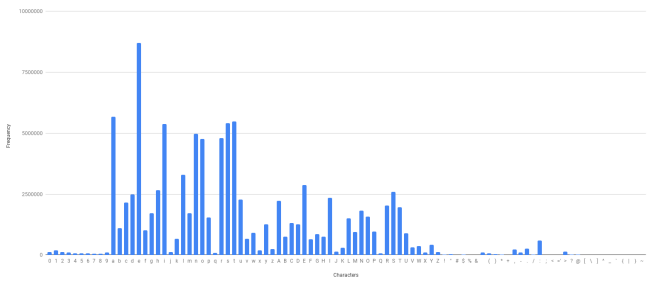

Fig. 2. Frenquency of characters in training lexicon

samples and thus focus training on low-frequency ones [26]. The focal loss is defined as follows:

$$FL\left(p_t\right) = -\alpha_t \left(1-p_t\right)^{\gamma} \log\left(p_t\right), \tag{4}$$

where, p_t is the probability of the predicted value, computed using softmax function, α and γ are tunable hyperparameters used to balance the loss. Intuitively, focal loss is obtained by multiplying cross entropy by $\alpha_t \left(1-p_t\right)^{\gamma}$. Note that the weight $\alpha_t \left(1-p_t\right)^{\gamma}$ is inversely proportional with p_t, thus the focal loss helps to reduce the impact of high-frequency samples (whose value of p_t is usually high) and focus more on low-frequency ones (which usually have low value of p_t).

Based on focal loss, we define our training objective as follows:

$$L = -\sum_{t=1}^{T} \left(\alpha_t \left(1-p\left(y_t \mid I\right)\right)^{\gamma} \log p\left(y_t \mid I\right)\right) \tag{5}$$

where y_t are the predicted characters, T is the length of the predicted sequence, and I is the input image.

IV. Performance Evaluation

In this section, we conduct experiments to demonstrate the effectiveness of our proposed model. We first briefly introduce datasets used for training and testing, then we describe our implementation details. Next, we analyze the effect of focal loss on our model. Finally, we compare our model against state-of-the-art techniques on seven public benchmark datasets, including regular and irregular text.

A. Datasets

The training datasets contains two datasets: **Synth90k** and **SynthText**. Synth90k is a synthetic dataset introduced in [9]. This dataset contains 9 million images created by combining 90.000 common English words and random variations and effects. SynthText is a synthetic dataset introduced in [10], which contains 7 million samples by the same generation process as Synth90k [9]. However, SynthText is targeted for text detection so that an image may contain several words. All experiments are evaluated on seven well-known public benchmarks described, which can be divided into two categories: regular text and irregular text. Regular text datasets include IIIT5K, SVT, ICDAR03, ICDAR13.

- IIIT5K [27] contains 3000 test images collected from Google image searches.
- ICDAR03 [28] contains 860 word-box cropped images.
- ICDAR13 [29] contains 1015 word-box cropped images.
- SVT contains 647 testing word-box collected from Google Street View.

Irregular text datasets include ICDAR15, SVT-P, CUTE.

- ICDAR15 [30] contains 1811 testing word-box cropped images collected from Google Glass without careful positioning and focusing.
- SVT-P [31] contains 645 testing word-box cropped images collected from Google Street View. Most of them are heavily distorted by the non-frontal view angle.
- CUTE [32] contains 288 word-box cropped images, which are curved text images.

B. Configurations

1) Implementation Detail: We implement the proposed model by Pytorch library and Python programming language. The model is trained and tested on an NDIVIA RTX 2080 Ti GPU with 12 GB memory. We train the model from scratch using Adam optimizer with the learning rate of 0.00002. To evaluate the trained model, we use dataset III5K. The pretrained model and code are available at [33]

2) Rectification Network: All input images are resized to 64×256 before applying the rectification network. The rectification network consists of three components: a localization network, a thin plate spline (TPS) transformation, and a sampler. The localization network consists of 6 convolutional layers with the kernel size of 3×3 and two fully-connected (FCN) layers. Each FCN is followed by a 2×2 max-pooling layer. The number of the output filters is 32, 64, 128, 256, and 256. The number of output units of FCN is 512 and 2K, respectively, where K is the number of the control points. In all experiments, we set K to 20, as suggested by [8]. The sampler generates the rectified image with a size of 32×100. The size of the rectified image is also the input size of the feature extraction module.

3) Feature Extraction: We construct the feature extraction module based on Resnet architecture [25]. The configurations of the feature extraction network are listed in Table I. Our feature extraction network contains five blocks of 45 residual layers. Each residual unit consists of a 1×1 convolutional layer, followed by a 3×3 convolution layer. In the first two blocks, we use 2×2 stride to reduce the feature map dimension. In the next blocks, we use 2×1 stride to down-sampled feature maps. The 2×1 stride also allows us to retain more information horizontally to distinguish neighbor characters effectively.

4) Recognition: The number of blocks in the encoder and the decoder are set both to 4. In each block of the encoder and the decoder, the dimension of the feed forward vector and the ouput vector are set to 2048 and 512, respectively. The number of head attention layers is set to 8. The decoder recognizes 94 different characters, including numbers, alphabet characters, uppercase, lowercase, and 32 punctuation in ASCII.

TABLE I
FEATURE EXTRACTION NETWORK CONFIGURATIONS.EACH BLOCK IS A RESIDUAL NETWORK BLOCK. "S" STANDS FOR STRIDE OF THE FIRST CONVOLUTIONAL LAYER IN A BLOCK.

	Layer	Feature map size	Configuration
Encoder	Block 0	32×100	3×3 conv, s(1×1)
	Block 1	16×50	$\begin{bmatrix} 1 \times 1 \text{conv}, 32 \\ 3 \times 3 \text{conv}, 32 \end{bmatrix} \times 3$, s$(2 \times 2)$
	Block 2	8×25	$\begin{bmatrix} 1 \times 1 \text{conv}, 64 \\ 3 \times 3 \text{conv}, 32 \end{bmatrix} \times 3$, s$(2 \times 2)$
	Block 3	4×25	$\begin{bmatrix} 1 \times 1 \text{conv}, 128 \\ 3 \times 3 \text{conv}, 32 \end{bmatrix} \times 3$, s$(2 \times 1)$
	Block 4	2×25	$\begin{bmatrix} 1 \times 1 \text{conv}, 256 \\ 3 \times 3 \text{conv}, 32 \end{bmatrix} \times 3$, s$(2 \times 1)$
	Block 5	1×25	$\begin{bmatrix} 1 \times 1 \text{conv}, 512 \\ 3 \times 3 \text{conv}, 32 \end{bmatrix} \times 3$, s$(2 \times 1)$

C. Result and Discussion

1) Impact of focal loss: To analyze the effect of focal loss, we study two variants of the proposed model. The first variant uses negative log-likelihood, and the second one leverages focal loss.

TABLE II
RECOGNITION ACCURACIES WITH NEGATIVE LOG-LIKELIHOOD AND FOCAL LOSS

Variant	Negative log-likelihood	Focal Loss
IIIT5K	92.6	**93.9**
SVT	85.8	**88.6**
ICDAR03	94.1	**95**
ICDAR13	92	**92.8**
ICDAR15	76.1	**77.5**
SVT-P	79.4	**81.7**
CUTE	80.6	**85.4**
Avarage	86.9	**88.2**

As shown in Table II, the model with focal loss outperforms the one with log-likelihood on all datasets. Notably, on average, focal loss improves the accuracy by 2.3 % compared to log-likelihood. For the best case, i.e., CUTE, the performance gap between the two variants is 4.8 %

2) Impact of rectification network: In this section, we study the effect of text rectification by comparing SAFL and a variant which does not include the rectification module.

TABLE III
RECOGNITION ACCURACIES WITH AND WITHOUT RECTIFICATION

Variant	SAFL w/o text rectification	SAFL
IIIT5K	90.7	**93.9**
SVT	83.3	**88.6**
ICDAR03	93	**95**
ICDAR13	90.7	**92.8**
ICDAR15	72.9	**77.5**
SVT-P	71.6	**81.7**
CUTE	77.4	**85.4**
Avarage	84.1	**88.2**

Table III depicts the recognition accuracies of the two models over seven datasets. It can be observed that the rectification module increases the accuracy significantly. Specifically, the performance gap between SAFL and the one without the rectification module is 4.1% on average. In the best cases, SAFL improves the accuracy by 10.1% and 7% compared to the other on the datasets SVT-P and CUTE, respectively. The reason is that both SVT-P and CUTE contains many both irregular texts such as perspective texts or curved texts.

3) Comparison with State-of-the-art: In this section, we compare the performance of SAFL with the latest approaches in scene text recognition. The evaluation results are shown in Table IV. In each column, the best value is bolded. the "Avarage" column is the weighted average over all the data sets. Concerning the irregular text, it can be observed that SAFL achieves the best performance on 3 data sets. Particularly, SAFL outperforms the current state-of-the-art, ESIR [23], by a margin of 1.2% on average, particulary on CUTE $(+2.1\%)$ and SVT-P $(+2.1\%)$. Concerning the regular datasets, SAFL outperforms the other methods on two datasets IIIT5K and ICDAR03. Moreover, SAFL also shows the highest average accuracy over all the regular text datasets. To summarize, SAFL achieves the best performance on 5 of 7 datasets and the highest average accuracy on both irregular and regular texts.

V. CONCLUSION

In this paper, we proposed SAFL, a deep learning model for scene text recognition, which exploits self-attention mechanism and focal loss. The experiment results showed that SAFL achieves the highest average accuracy on both the regular datasets and irregular datasets. Moreover, SAFL outperforms the state-of-the-art on CUTE dataset by a margin of 2.1%. Summary, SAFL shows superior performance on 5 out of 7 benchmarks, including IIIT5k, ICDAR 2003, ICDAR 2015, SVT-P and CUTE.

REFERENCES

[1] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE transactions on neural networks*, vol. 5, no. 2, pp. 157–166, 1994.

[2] K. Cho, B. Van Merriënboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," *arXiv preprint arXiv:1409.1259*, 2014.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[4] N. Carion, F. Massa, G. Synnaeve, N. Usunier, A. Kirillov, and S. Zagoruyko, "End-to-end object detection with transformers," *arXiv preprint arXiv:2005.12872*, 2020.

[5] N. Parmar, A. Vaswani, J. Uszkoreit, Ł. Kaiser, N. Shazeer, A. Ku, and D. Tran, "Image transformer," *arXiv preprint arXiv:1802.05751*, 2018.

[6] M. Jaderberg, K. Simonyan, A. Zisserman *et al.*, "Spatial transformer networks," in *Advances in neural information processing systems*, 2015, pp. 2017–2025.

[7] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 4168–4176.

[8] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "Aster: An attentional scene text recognizer with flexible rectification," *IEEE transactions on pattern analysis and machine intelligence*, vol. 41, no. 9, pp. 2035–2048, 2018.

[9] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," *arXiv preprint arXiv:1406.2227*, 2014.

[10] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 2315–2324.

TABLE IV
SCENE TEXT ACCURANCIES (%) OVER SEVEN PUBLIC BENCHMARK TEST DATASETS.

Method	Regular test dataset					Irregular test dataset			
	IIIT5k	SVT	ICDAR03	ICDAR13	Average	ICDAR15	SVT-P	CUTE	Average
Jaderberg et al. [34]	-	80.7	93.1	90.8	-	-	-	-	-
CRNN [19]	78.2	80.8	89.4	86.7	81.8	-	-	-	-
RARE [7]	81.9	81.9	90.1	88.6	85.3	-	71.8	59.2	-
Lee et al.	78.4	80.7	88.7	90.0	82.4	-	-	-	-
Yang et al. [21]	-	75.8	-	-	-	-	75.8	69.3	-
FAN [35]	87.4	85.9	94.2	93.3	89.4	70.6	-	-	-
Shi et al. [7]	81.2	82.7	91.9	89.6	84.6	-	-	-	-
Yang et al. [21]	-	-	-	-	-	-	75.8	69.3	-
Char-Net [22]	83.6	84.4	91.5	90.8	86.2	60.0	73.5	-	-
AON [36]	87.0	82.8	91.5	-	-	68.2	73.0	76.8	70.0
EP [37]	88.3	87.5	94.6	**94.4**	90.3	73.9	-	-	-
Liao et al. [38]	91.9	86.4	-	86.4	-	-	-	79.9	-
Baek et al. [14]	87.9	87.5	94.9	92.3	89.8	71.8	79.2	74.0	73.6
ASTER [8]	93.4	89.5	94.5	91.8	92.8	76.1	78.5	79.5	76.9
SAR [39]	91.5	84.5	-	91.0	-	69.2	76.4	83.3	72.1
ESIR [23]	93.3	**90.2**	-	91.3	-	76.9	79.6	83.3	78.1
SAFL	**93.9**	88.6	**95**	92.8	**93.3**	**77.5**	**81.7**	**85.4**	**79.3**

[11] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[12] Y. Zhu, C. Yao, and X. Bai, "Scene text detection and recognition: Recent advances and future trends," *Frontiers of Computer Science*, vol. 10, no. 1, pp. 19–36, 2016.

[13] Q. Ye and D. Doermann, "Text detection and recognition in imagery: A survey," *IEEE transactions on pattern analysis and machine intelligence*, vol. 37, no. 7, pp. 1480–1500, 2014.

[14] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 4715–4723.

[15] P. Wang, L. Yang, H. Li, Y. Deng, C. Shen, and Y. Zhang, "A simple and robust convolutional-attention network for irregular text recognition," *arXiv preprint arXiv:1904.01375*, vol. 6, 2019.

[16] K. Wang, B. Babenko, and S. Belongie, "End-to-end scene text recognition," in *2011 International Conference on Computer Vision*. IEEE, 2011, pp. 1457–1464.

[17] C. Yao, X. Bai, B. Shi, and W. Liu, "Strokelets: A learned multi-scale representation for scene text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4042–4049.

[18] K. Wang and S. Belongie, "Word spotting in the wild," in *European Conference on Computer Vision*. Springer, 2010, pp. 591–604.

[19] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[20] P. He, W. Huang, Y. Qiao, C. C. Loy, and X. Tang, "Reading scene text in deep convolutional sequences," in *Thirtieth AAAI conference on artificial intelligence*, 2016.

[21] X. Yang, D. He, Z. Zhou, D. Kifer, and C. L. Giles, "Learning to read irregular text with attention mechanisms." in *IJCAI*, vol. 1, no. 2, 2017, p. 3.

[22] W. Liu, C. Chen, and K.-Y. K. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[23] F. Zhan and S. Lu, "Esir: End-to-end scene text recognition via iterative image rectification," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2059–2068.

[24] J. L. Ba, J. R. Kiros, and G. E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[25] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[26] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980–2988.

[27] A. Mishra, K. Alahari, and C. Jawahar, "Top-down and bottom-up cues for scene text recognition," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012, pp. 2687–2694.

[28] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto *et al.*, "Icdar 2003 robust reading competitions: entries, results, and future directions," *International Journal of Document Analysis and Recognition (IJDAR)*, vol. 7, no. 2-3, pp. 105–122, 2005.

[29] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. A. Almazan, and L. P. De Las Heras, "Icdar 2013 robust reading competition," in *2013 12th International Conference on Document Analysis and Recognition*. IEEE, 2013, pp. 1484–1493.

[30] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu *et al.*, "Icdar 2015 competition on robust reading," in *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2015, pp. 1156–1160.

[31] T. Quy Phan, P. Shivakumara, S. Tian, and C. Lim Tan, "Recognizing text with perspective distortion in natural scenes," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 569–576.

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

[33] https://github.com/ICMLA-SAFL/SAFL_pytorch.

[34] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman, "Deep structured output learning for unconstrained text recognition," *arXiv preprint arXiv:1412.5903*, 2014.

[35] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 5076–5084.

[36] Z. Cheng, Y. Xu, F. Bai, Y. Niu, S. Pu, and S. Zhou, "Aon: Towards arbitrarily-oriented text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 5571–5579.

[37] F. Bai, Z. Cheng, Y. Niu, S. Pu, and S. Zhou, "Edit probability for scene text recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1508–1516.

[38] M. Liao, J. Zhang, Z. Wan, F. Xie, J. Liang, P. Lyu, C. Yao, and X. Bai, "Scene text recognition from two-dimensional perspective," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8714–8721.

[39] H. Li, P. Wang, C. Shen, and G. Zhang, "Show, attend and read: A simple and strong baseline for irregular text recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 33, 2019, pp. 8610–8617.