CF-005

ViT-CTC: Vision Transformers with CTC for Scene Text Recognition

Rina Buoy[1] Masakazu Iwamura[1] Sovila Srun[2] Koichi Kise[1]

1. Abstract

Connectionist temporal classification (CTC) is one of the popular decoders in scene text recognition (STR), thanks to its simplicity and low latency. Many CTC-based methods rely on a one-dimensional (1D) vector sequence that is an output from a recurrent neural network (RNN) encoder. Nevertheless, for curved or irregular text, 2D features are required for accurately predicting characters. In this paper, we considered a ViT-CTC STR design that uses a CTC decoder and a pretrained vision transformer (ViT) as a two-dimensional (2D) feature extractor. Since a CTC decoder expects a 1D vector sequence as input, we explored two options to collapse a 2D feature sequence from a ViT model to a 1D feature sequence by (1) vertical averaging in feature space and (2) height marginalization in class probability space. Based on the latter option, we derived association maps that associate each predicted character to regions on an input image and determine character locations without any visual-linguistic cross-attention mechanism. The association maps can be used to explain why a model predicts what it predicts. The ViT-CTC models were trained on synthetic training datasets and inoculated by finetuning on real labeled datasets. The inoculated ViT-CTC models outperform the recent CTC-based state-of-the-art (SOTA) methods on the aggregated public benchmark dataset.

2. Introduction

Scene text recognition (STR) recognizes text in a natural scene and has remained an active area of research because of complex and distorted imaging conditions [1, 2]. Many existing deep learning-based STR methods can be categorized according to how they decode extracted visual features to characters. These methods can be based on a connectionist temporal classification (CTC) decoder, an attention-based decoder, or a transformer decoder [3, 4]. The CTC-based methods have lower latency and performance because each decoded character is assumed to be conditionally independent. The attention-based and transformer decoder-based methods achieve higher latency and performance because each decoded character is conditionally dependent on previous characters [4]. CTC is a many-to-one alignment-free algorithm that assigns a total probability of an output sequence conditioned on an input sequence by marginalizing over all possible alignments [5]. CTC was first proposed to map a sequence of one-dimensional (1D) acoustic features to a sequence of characters [6]. CTC is a non-autoregressive decoder since the output at the current timestep is not conditioned on the previous outputs [7]. Since a CTC decoder expects a 1D vector sequence, existing CTC-based STR methods such as CRNN [8], Rosetta[9], GRCNN [10], STAR-Net [11], and TRBC [12] use a 1D feature extractor that maps a two-dimensional (2D) image to a 1D feature vector sequence by collapsing height dimension and increasing feature dimension, as shown in Figure 1(a).

However, for irregular or curved text and non-Latin scripts with complex structure, a 2D feature sequence, as shown in Figure 1(b), is required for accurately predicting characters distributed in a two-dimensional space [13, 14, 15]. In addition, the imposed requirement of a 1D feature vector sequence by a CTC decoder suggests that powerful pretrained 2D feature extractors such as vision transformers (ViT) cannot be utilized directly. In contrast, existing CTC-based methods use a custom 1D feature extractor, especially a modified convolutional architecture that is often not pretrained on a large-scale image data. There was an attempt by Wan et al. [16] to extend the standard CTC decoder to include the height dimension during alignment operation. Despite the improved performance by the proposed 2D-CTC, the 2D prediction problems remain unsolved [14].

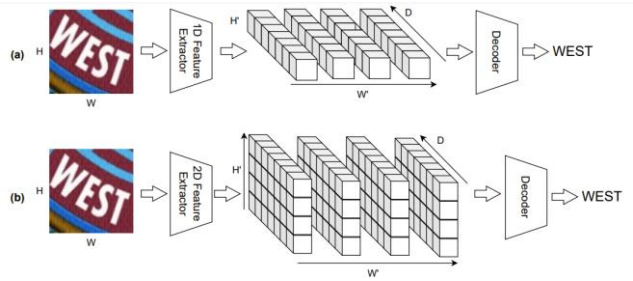

Figure 1: (a) 1D vs. (b) 2D STR Predictions

To overcome limitations and maintain low latency of a standard CTC decoder, we considered a ViT-CTC scene text recognition design that uses a CTC decoder and a pretrained vision transformer (ViT) as a 2D visual-temporal feature extractor. Since CTC operates on a 1D vector sequence, we explored two options: (1) vertical averaging in feature space and (2) height marginalization in class probability space. In the former option, a 2D feature ($H' \times W'$) sequence from a pretrained ViT model is averaged over height dimension to form a 1D feature sequence as input for a character softmax classifier and a CTC decoder. In the latter option, the character softmax classifier operates on a 2D feature sequence and produces a 2D joint distribution sequence over height (H') and characters (C). Marginalization is applied over the height dimension to obtain a 1D marginal distribution sequence over characters for a CTC decoder. While the former option is simple, the latter option provides model explainability by simultaneously predicting and locating characters through association maps shown in Figure 2. The association maps associate each predicted character to regions in an input image and thus explain why a model predicts what it predicts. We experimented with different ViT-CTC designs by using pretrained ViT models including data-

[1] Osaka Metropolitan University
[2] Royal University of Phnom Penh

efficient image transformers (DeiT-III) [17] and class-attention in image transformers (CaiT) [18]. Our ViT-CTC models were trained on a collection of synthetic training datasets and inoculated by finetuning on a collection of real labeled datasets. The experiment results show that our inoculated ViT-CTC models outperform the recent CTC-based state-of-the-art (SOTA) methods on the aggregated public benchmark dataset. Our contributions can be summarized as follows:

- We proposed a height marginalization method in class probability space as a more explainable alternative to a simple vertical feature averaging technique. With the proposed technique, we experimented with various ViT-CTC designs by using different pretrained ViT architectures and a CTC decoder.
- With the proposed method, we derived association maps that associate each predicted character to regions in an input image and thus, determine character locations without any visual-linguistic cross-attention mechanism.
- We found that the inoculated ViT-CTC models achieve competitive performance with the recent CTC-based state-of-the-art (SOTA) methods on the public benchmark datasets.

Figure 2: Association maps for model explainability and character lor localization

3. Proposed Method

The marginalization approach works in probability space instead of feature space. If $F(H', W', D)$ is the extracted feature maps from any 2D feature extractor, $F(H', W', D)$ is directly fed to a linear classifier to produce unnormalized score distributions $S(H', W', C)$. $S(H', W', C)$ is given by:

$$S(H', W', C) = LinearClassifier(F(H', W', D)) \quad (1)$$

A softmax normalization is applied to $S(H', W', C)$ along H' and C to produce $S^*(H', W', C)$ in which a slice along W' is a valid joint distribution over H' and C.

$$S^*(H', W', C) = Softmax_{H',C'}(S(H', W', C)) \quad (2)$$

Next, $S^*(H', W', C)$ is marginalized over H' to produce a sequence of valid distributions over characters, ${S^*}_M(W', C)$ that is required for a CTC decoder. ${S^*}_M(W', C)$ is given by:

$${S^*}_M(W', C) = \sum_h^{H'} S^*(h, W', C)) \quad (3)$$

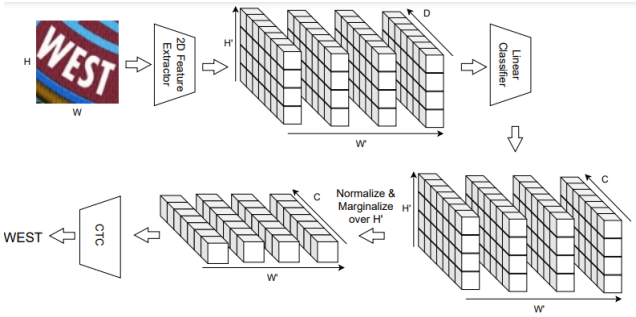

Figure 3: The marginalization-based STR method

The marginalization-based STR method architecture is shown in Figure 3.

4. Datasets

The synthetic training set comprises 8.5M images from 50% of MJSynth [19], 50% of SynthText [20], 100% of SynthAdd [13], 10% of SynthTiger [21].

The real finetuning set comprises 2.4M images from COCO-Text [22], RCTW [23], Uber-Text [24], ArT [25], LSVT [26], ReCTS [27], TextOCR [30], and OpenImmages [29]. For COCO-Text, RCTW , Uber-Text, ArT, LSVT, and ReCTS, we used the processed versions by [30] while for TextOCR and OpenImmages, we used the processed versions by [3].

Sample synthetic training and real finetuning images are given in Figure 4.

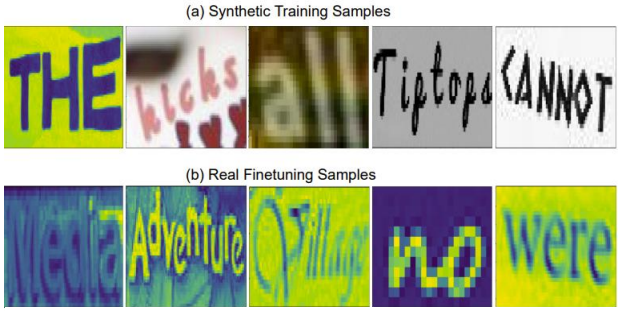

Figure 4: (a) Synthetic training samples. (b) Real finetuning samples.

The evaluation datasets include the test sets of SVT [31], IIIT5k [32], IC13 [33], IC15 [34], and the entire sets of SVTP [35] and CUTE80 [36].

5. Experiment Design

We experimented with three variants of DeiT-III [17] including DeiT-Small, DeiT-Medium, and DeiT-Base and a CaiT-Small [18]. The details of these four pretrained ViT models are shown in Table 1. For an input image of 224×224, the output feature maps are $14 \times 14 \times D$, and D is an embedding dimension.

Table 1: The specifications of pretrained ViT models. FPS:

ViT Model	Params	GFLOPs	Size	Emb. Dim,D	Acc@1 (INet-1k)	FPS
DeiT-Small	22.2M	4.6	224×224	384	81.4	65
DeiT-Medium	38.8M	8.0	224×224	512	83.0	66
DeiT-Base	86.6M	17.5	224×224	768	83.8	62
CaiT-Small	47.0M	9.4	224×224	384	83.5	25

The character set has 37 characters including case-insensitive alphabets, numbers, and a blank token, ε. The input images were

resized to 224×224 pixels. The training strategy comprised two phases: (1) training on the synthetic datasets and (2) finetuning on the real datasets. These two phases of training allow us to identify models' weaknesses or training datasets' blindspots during evaluation [37]. The training process consisted of 50 iterations. During each iteration, 300,000 images were randomly selected, and a batch of 64 images were used for training without any data augmentation. This is equivalent to around 2 epochs on the whole training data. The fine-tuning phase followed the same settings as before, but it only lasted for 30 iterations, which is approximately equivalent to 3 epochs over the entire finetuning dataset. The cyclic learning schedules between 10^{-4} and 10^{-5} and between 10^{-5} and 10^{-6}, were used for training and finetuning phase, respectively.

6. Results

For fair comparisons, we compared with only the existing CTC-based and non-autoregressive methods. The existing methods except for ViTSTRs [1] and DiG-ViTs [2] are based on a 1D feature extractor. ViTSTRs use a pretrained ViT model and a non-autoregressive decoder while DiG-ViTs use a vertical feature averaging technique and a CTC decoder. GTC [38] was trained with an attention-based decoder but performed inference with a CTC decoder. CRNN [8], STAR-Net [11], GRCNN [10], and Rosetta [9] are based on re-implementations by Baek et al. [12] that includes evaluations on all benchmark datasets. Table 2 shows that performance differences between feature averaging and marginalization in the ViT-CTC models were marginal although the marginalization-based models obtained the highest accuracy across all the public benchmark datasets except for CUTE80. In addition, with the marginalization method, we have access to $S^*(H', W', C)$, a sequence of joint distributions over H' and C. $S^*(H', W', C)$ allows us to reason why a model predicts what it predicts. It is also possible to localize a predicted characters by using $S^*(H', W', C)$ which will be discussed in the below section. In the context of attention-based or encoder-decoder STR methods, $S^*(H', W', C)$ is equivalent to cross-modality attention maps.

The finetuned ViT-CTC models outperformed the recent CTC-based SOTAs such as DiG-ViTs and GTC on the aggregated benchmark dataset, as shown in Figure 5. A major difference between the ViT-CTC and DiG-ViT models is that DiG-ViT models were pretrained without supervision on a collection of 15.77M unlabeled real data and finetuned on 2.78M labeled data. However, the ViT-CTC models were pretrained with supervision on labeled synthetic datasets.

7. Model Explainability and Character Localization

With existing CTC-based methods, it is not possible to explain why a model predicts what it predicts. Since a slice along W' in $S^*(H', W', C)$ is a valid joint distribution over H' and C, it is possible to spatially associate model prediction to regions on the feature maps or an input image. Such spatial association between a predicted character and regions on the feature maps is crucial for explaining why a model predicts what it predicts. Another benefit is that we can use the association map to localize the predicted characters. The association map (AM) is expressed below:

$$AM(H', w) = 1 \; if \; S^*(H', w, x_w) \geq \alpha \; and \; x_w \neq \in \quad (4)$$

$$AM(H', w) = 0 \; otherwise \quad (5)$$

w is a prediction timestep or frame. x_w is a CTC predicted character at w. α is a threshold between zero and one. A high α associates x_w only to high probability regions. Figure 6 shows the association maps corresponding to different values of α. The association maps demonstrate that the model manages to associate each predicted character to the correct lower regions instead of the top intruding regions.

Table 2: Model performance comparisons with the existing CTC-based. Underscore : highest for the existing methods. Bold : highest for the ViT-CTC models. FT: finetuning. FA : feature averaging. M : marginalization. Params: parameters in millions.

Method	Params	IIIT	SVT	IC13	IC15	SVTP	CUTE	Total
CRNN	8.3	82.9	81.6	89.2	69.4	70.0	65.5	78.5
STAR-Net	48.7	87.0	86.9	91.5	76.1	77.5	71.7	83.5
GRCNN	4.6	84.2	83.7	88.8	71.4	73.6	68.1	80.1
Rosetta	44.3	84.3	84.7	89.0	71.2	73.8	69.2	80.3
TRBC	48.7	87.0	86.9	91.5	76.1	77.5	71.7	83.5
GTC	-	96.0	91.8	93.2	79.5	85.6	91.3	90.1
CRNN	8.3	89.8	84.3	90.9	73.1	74.6	82.3	83.8
SeqCLR	-	80.9	-	86.3	-	-	-	-
ViTSTR-S	21.5	86.6	91.2	77.9	87.3	81.4	77.9	84.4
ViTSTR-B	85.8	88.4	92.4	78.5	87.7	81.8	81.3	85.6
PerSec-ViT (CTC)	-	85.4	86.1	92.8	70.3	73.9	69.2	81.2
DiG-ViT-T (CTC)	20.0	93.3	89.7	92.5	79.1	78.8	83.0	87.7
DiG-ViT-S (CTC)	36.0	95.5	91.8	95.0	84.1	83.9	86.5	91.0
DiG-ViT-B (CTC)	52.0	95.9	92.6	95.3	84.2	85.0	89.2	91.5
DeiT-S (FA) (baseline)	21.6	91.4	86.4	89.6	74.2	75.8	79.1	84.7
DeiT-M (FA) (baseline)	38.9	92.0	87.3	91.4	77.4	78.9	82.2	86.4
DeiT-B (FA) (baseline)	**85.7**	93.1	88.7	92.9	77.3	79.7	85.7	87.4
CaiT-S (FA) (baseline)	46.5	94.3	87.2	92.5	79.5	79.4	87.1	88.2
DeiT-S (M) (ours)	21.6	91.4	85.5	91.3	75.3	76.7	82.2	85.3
DeiT-M (M) (ours)	38.9	92.5	87.8	92.2	76.6	79.5	81.9	86.6
DeiT-B (M) (ours)	**85.7**	93.0	86.9	92.2	78.6	79.1	84.0	87.3
CaiT-S (M) (ours)	46.5	93.5	86.9	91.9	77.6	77.8	85.4	87.2
DeiT-S (FA) + FT (baseline)	21.6	95.0	88.4	94.2	81.6	82.0	88.5	89.6
DeiT-M (FA) + FT (baseline)	38.9	95.5	91.2	95.4	83.4	83.4	92.0	91.0
DeiT-B (FA) + FT (baseline)	**85.7**	95.9	92.1	95.9	83.9	84.2	92.7	91.5
CaiT-S (FA) + FT (baseline)	46.5	96.0	92.3	95.8	84.5	84.7	**93.7**	**91.7**
DeiT-S (M) + FT (ours)	21.6	94.6	89.2	95.4	81.5	83.1	91.3	89.9
DeiT-M (M) + FT (ours)	38.9	95.0	92.3	95.2	83.5	84.0	90.9	90.9
DeiT-B(M) + FT (ours)	**85.7**	95.9	**92.6**	**96.1**	84.4	84.3	92.7	**91.7**
CaiT-S (M) + FT (ours)	46.5	**96.1**	90.6	95.4	**84.9**	**85.4**	92.7	**91.7**

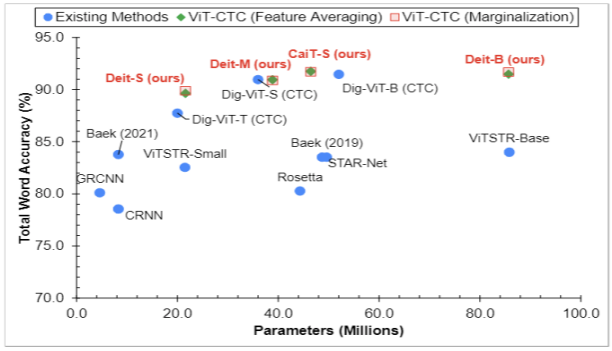

Figure 5: Model performance vs. parameters on all datasets (total).

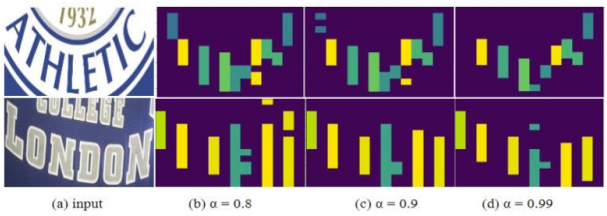

Figure 6: Association maps for different values of α. The color bars show regions on input images, corresponding to predicted characters.

8. Conclusion

In this paper, we considered a ViT-CTC STR design that uses a CTC decoder and a pretrained vision transformer (ViT) as a two-dimensional (2D) feature extractor. In addition to a simple vertical feature averaging technique, we presented a marginalization-based technique that collapses the height dimension in class probability space. Based on the proposed method, we derived association maps that explain why a model predicts what it predicts, and determine character locations without any visual-linguistic cross-attention mechanism. The inoculated ViT-CTC models outperform the recent CTC-based state-of-the-art (SOTA) methods on the aggregated public benchmark dataset. The proposed method can be easily adopted, implemented and generalized to other 2D pretrained feature extractors.

Acknowledgements

This work was supported by JSPS Kakenhi Grant Number 22H00540 and RUPP-OMU/HEIP.

Reference

[1] Atienza, R. Vision transformer for fast and efficient scene text recognition. Document Analysis and Recognition – ICDAR 2021 319–334 (2021).
[2] Liao, M. et al. Scene text recognition from two-dimensional perspective. Proceedings of the AAAI Conference on Artificial Intelligence 33, 8714–8721 (2019).
[3] Yang, M. et al. Reading and writing: Discriminative and generative modeling for self-supervised text recognition. Proceedings of the 30th ACM International Conference on Multimedia (2022).
[4] Diaz, D. H., Qin, S., Ingle, R. R., Fujii, Y. & Bissacco, A. Rethinking text line recognition models. arXiv preprint arXiv:104.07787(2021). URL https://arxiv.org/abs/2104.07787.
[5] Hannun, A. Sequence modeling with ctc. Distill (2017). Https://distill.pub/2017/ctc.
[6] Graves, A., Fern´andez, S., Gomez, F. & Schmidhuber, J. Connectionist temporal classification. Proceedings of the 23rd international conference on Machine learning-ICML '06 (2006).
[7] Jurafsky, D. & Martin, J. H. Speech and language processing: An introduction to natural language processing, computational linguistics, and speech recognition (Pearson, 2022).
[8] Shi, B., Bai, X. & Yao, C. 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, 2298–2304 (2017).
[9] Borisyuk, F., Gordo, A. & Sivakumar, V. Rosetta. Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (2018).
[10] Wang, J. & Hu, X. Guyon, I. et al.(eds) Gated recurrent convolution neural network for ocr. (eds Guyon, I. et al.) Advances in Neural Information Processing Systems, Vol. 30 (Curran Associates, Inc., 2017).
[11] Liu, W., Chen, C., Wong, K.-Y., Su, Z. & Han, J. Star-net: A spatial attention residue network for scene text recognition. Procedings of the British Machine Vision Conference 2016 (2016).
[12] Baek, J. et al. What is wrong with scene text recognition model comparisons? dataset and model analysis. 2019 IEEE/CVF International Conference on Computer Vision (ICCV) (2019).
[13] Li, H., Wang, P., Shen, C. & Zhang, G. Show, attend and read: A simple and strong baseline for irregular text recognition. Proceedings of the AAAI Conference on Artificial Intelligence 33, 8610–8617 (2019).
[14] Chen, X., Jin, L., Zhu, Y., Luo, C. & Wang, T. Text recognition in the wild. ACM Computing Surveys 54, 1–35 (2021).
[15] Xie, Z. et al. Aggregation cross-entropy for sequence recognition. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019).
[16] Wan, Z., Xie, F., Liu, Y., Bai, X. & Yao, C. 2d-ctc for scene text recognition. arXiv preprint arXiv:1907.09705 (2019).
[17] Touvron, H., Cord, M. & Jegou, H. Deit III: Revenge of the vit. arXiv preprint arXiv:2204.07118 (2022).
[18] Touvron, H., Cord, M., Sablayrolles, A., Synnaeve, G. & J'egou, H. Going deeper with image transformers. 2021 IEEE/CVF International Conference on Computer Vision (ICCV) 32–42 (2021).
[19] Jaderberg, M., Simonyan, K., Vedaldi, A. & Zisserman, A. Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 abs/1406.2227 (2014).
[20] Gupta, A., Vedaldi, A. & Zisserman, A. Synthetic data for text localisation in natural images. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2016).
[21] Yim, M., Kim, Y., Cho, H.-C. & Park, S. Synthtiger: Synthetic text image generator towards better text recognition models. Document Analysis and Recognition – ICDAR 2021 109–124 (2021).
[22] Veit, A., Matera, T., Neumann, L., Matas, J. & Belongie, S. J. Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 abs/1601.07140 (2016).
[23] Shi, B. et al. Icdar2017 competition on reading chinese text in the wild (rctw-17). 2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR) (2017).
[24] Zhang, Y. et al. Uber-text: A large-scale dataset for optical character recognition from street-level imagery (2017).
[25] Chng, C. K. et al. Icdar2019 robust reading challenge on arbitrary-shaped text - rrc-art. 2019 International Conference on Document Analysis and Recognition (ICDAR) (2019).
[26] Sun, Y. et al. Icdar 2019 competition on large-scale street view text with partial labeling - rrc-lsvt. 2019 International Conference on Document Analysis and Recognition (ICDAR) (2019).
[27] Zhang, R. et al. Icdar 2019 robust reading challenge on reading chinese text on signboard. 2019 International Conference on Document Analysis and Recognition (ICDAR) (2019).
[28] Singh, A. et al. Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021).
[29] Krylov, I., Nosov, S. & Sovrasov, V. Balasubramanian, V. N. & Tsang, I. W. (eds) Openimages V5 text annotation and yet another mask text spotter. (eds Balasubramanian, V. N. & Tsang, I. W.) Asian Conference on Machine Learning, ACML 2021, 17-19 November 2021, Virtual Event, Vol. 157 of Proceedings of Machine Learning Research, 379–389 (PMLR, 2021).
[30] Baek, J., Matsui, Y. & Aizawa, K. What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2021).
[31] Wang, K., Babenko, B. & Belongie, S. End-to-end scene text recognition. 2011 International Conference on Computer Vision (2011).
[32] Mishra, A., Alahari, K. & Jawahar, C. Scene text recognition using higher order language priors. Procedings of the British Machine Vision Conference 2012 (2012).
[33] Karatzas, D. et al. Icdar 2013 robust reading competition. 2013 12th International Conference on Document Analysis and Recognition (2013).
[34] Karatzas, D. et al. Icdar 2015 competition on robust reading. 2015 13th International Conference on Document Analysis and Recognition (ICDAR) (2015).
[35] Phan, T. Q., Shivakumara, P., Tian, S. & Tan, C. L. Recognizing text with perspective distortion in natural scenes. 2013 IEEE International Conference on Computer Vision (2013).
[36] Risnumawan, A., Shivakumara, P., Chan, C. S. & Tan, C. L. A robust arbitrary text detection system for natural scene images. Expert Systems with Applications 41, 8027–8048 (2014).
[37] Liu, N. F., Schwartz, R. & Smith, N. A. Inoculation by fine-tuning: A method for analyzing challenge datasets. Proceedings of the 2019 Conference of the North (2019).
[38] Hu, W., Cai, X., Hou, J., Yi, S. & Lin, Z. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. Proceedings of the AAAI Conference on Artificial Intelligence 34, 11005–11012 (2020).