A Scene Text Recognition Model Based On Text Encoding Input

Anonymous ECCV 2024 Submission

Paper ID #10358

Abstract. Scene Text Recognition (STR) is a fundamental and popular area in imaging, particularly within Optical Character Recognition (OCR). With language models advancing, the STR task, which encompasses both image and language domains, has seen a growing adoption of multimodal models combining text and images. This paper introduces a novel integration method for text and images, named Text Encode Input (XInput). Experimentation has shown that the XInput methodl can be effectively integrated into most STR models, significantly enhancing performance while maintaining model size and inference time. The XInput-based model achieved state-of-the-art performance on public test datasets.

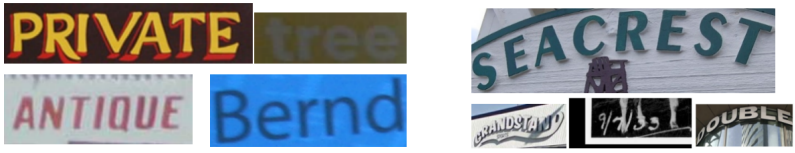
Keywords: OCR · STR · text-image multimodal model

1 Introduction

Optical Character Recognition (OCR) converts text from images into an editable format. It is widely used for digitizing documents, automating office tasks, and retrieving information. Meanwhile, Scene Text Recognition (STR) is a vital branch of OCR that focuses on recognizing and understanding text in images. Unlike traditional OCR systems, STR extends beyond simple character detection and recognition to include understanding of text layout, semantics, and context. STR's applications extend across multiple fields, including traffic sign recognition in autonomous driving, image annotation in retrieval systems, and text extraction in document digitization. Consequently, STR plays a pivotal role in practical applications, offering more efficient and intelligent text processing solutions across industries.

Despite its significant achievements, STR still faces numerous challenges. One major challenge is recognizing text in complex conditions, including uneven lighting, occlusion, distortion, and diverse languages and fonts. These challenges necessitate the design of robust and versatile STR systems, posing an urgent and complex issue that demands extensive research and innovation. Fig. 1 displays several samples from open STR datasets.

As tasks become more complex, the traditional CRNN (Convolutional Recurrent Neural Network) model [34] faces challenges in recognizing difficult samples. The attention mechanism [40] and the Vision Transformer (ViT) [9] have inspired researchers to investigate advanced and intricate models for STR tasks.



(a) simple samples **(b)** complex samples

Fig. 1: Sample images from open STR datasets

The attention mechanism enhances the model's focus on text-related areas, improving its processing of long texts and complex scenes. ViT model excels in tasks such as image classification by segmenting images into blocks and utilizing the Transformer architecture for processing. In STR tasks, the adoption of Vision Transformers (ViTs) opens up new opportunities for improving global image comprehension [2, 4, 8, 19, 27, 47, 50].

In multimodal research, as language models demonstrate increasing effectiveness, researchers are incorporating these models into STR tasks to improve performance [7, 11, 12, 23, 48, 51, 54]. For example, ABINet (Adaptive Bidirectional Integration Network) [11] and TrOCR (Transformer-based Optical Character Recognition with Pre-trained Models) [23] represent two key multimodal approaches. ABINet effectively merges text and image information, enhancing the accuracy and robustness of text recognition. TrOCR employs the Transformer architecture to integrate image patches and text information, achieving excellent performance across STR benchmarks, highlighting the potential of multimodal strategies in STR. These studies expand the design concepts for STR models and provide valuable insights for future deep learning research.

Addressing challenges in current STR tasks and reflecting on existing models, we propose a novel method for multimodal text-image fusion. This method can be applied in any encoder-decoder structure, named as text-input, or XInput. Unlike complex image-text multimodal model structure, our method directly concats label text and image encoding. This concatenated data is then passed to the decoder as the encoder's output, without relying on large language/text models.

We decided to abandon complex language models based on thorough observations of how humans recognize text in natural settings. Often, human text recognition does not heavily depend on linguistic semantics. In other words, humans recognize characters before they recognize words. Specifically, a person familiar with dictionary characters can often accurately identify characters by comparison when faced with an object to recognize, without needing to know the word as a whole. This observation becomes particularly significant with texts that deviate from the traditional semantic order. For example, with the text "TYPO", someone with knowledge of semantics might mistakenly correct it to "TYPE", while someone relying on a dictionary without semantic understanding could identify it correctly. Moreover, introducing a language model can lead to errors, such as incorrectly correcting "TYPO" to "TYPE", especially

considering that textual images provide abundant clues for precise recognition. Therefore, we conclude that a language model may not significantly improve recognition accuracy. If any improvement is observed, comparable benefits could be attained by utilizing image-based information.

Our new method is inspired by traditional learning methods, aiming to replicate human learning processes. Traditional learning often begins with guessing whether an image contains specific characters, which is then refined through the assessment of accuracy by the loss function. In contrast, human learning to read and write typically involves guidance from teachers, parents, or dictionaries, and gradually builds character knowledge through shape comparison. When learning a new character, a child visually learns its shape, while a teacher provides details on pronunciation, explanation, and meanings. Unlike waiting for a child's guess to be evaluated by a professional, our method directly provides textual information to the teacher or professional for immediate feedback. The new method directly encodes text information into the model, simulating human learning processes. We believe that this design aligns with the natural human literacy process, where knowledge of characters is built by comparing shapes during an external ground truth indoctrination. We believe that encoding textual information into the model simulates human guidance, closely mirroring human learning and potentially enhancing text recognition outcomes. Due to the incorporation of text information, the STR model with the new method will leverage image information in multimodal fusion, aiming for more efficient and robust text recognition. By fusing text and image information directly, we have experimentally validated the performance improvements of this design.

2 Related Word

Representative models such as the Convolutional Recurrent Neural Network (CRNN) [10], the attention-based Transform structure (ViTSTR [2], PARSeq [4]), and multimodal models incorporating language model (ABINet [11], DTrOCR [12]) serve as valuable resources for inspiring new algorithm designs. Fig. 2 illustrates the comparison between these STR models and the XInput method. Next, we will detail the characteristics of each model.

2.1 Convolutional Neural Networks

CRNN Traditional convolutional structures, such as the CRNN (Convolutional Recurrent Neural Network), have achieved significant results in STR. The CRNN model extracts image features using a convolutional neural network and then processes these features with a recurrent neural network for text recognition. While this structure excels in standard STR benchmarks, it faces challenges with long texts or complex scenes.

We consider the CRNN as a transformer encoder-decoder structure, with CNN serving as an encoder to extract image information and RNN acting as a decoder to process this information. However, due to the inherent limitations of

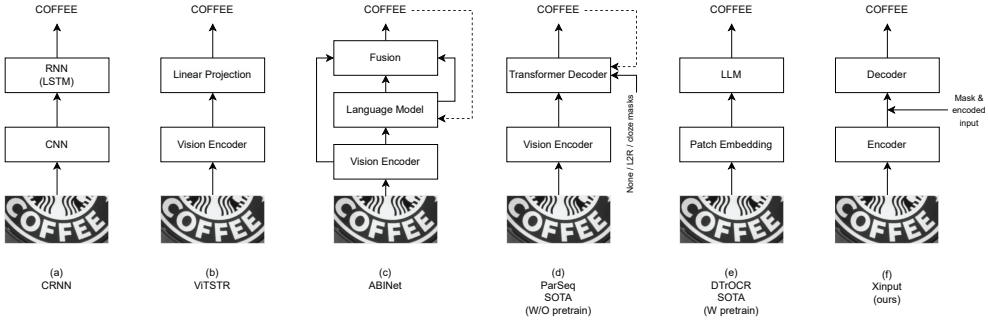


Fig. 2: Comparison of various STR model architectures. a) CRNN: The most widely applied method. b) ViTSTR: Represents STR models using transformer architectures. c) ABINet: Represents STR models incorporating language models. d) PARseq: State-of-the-art (SOTA) models without additional pre-training. e) DTrOCR: Uses large language models as a decoder, representing SOTA STR models with pre-training. f) XInput: The new method we proposed can be integrated into any encoder-decoder based model structure.

CNN and RNN, there remains significant room for improvement in recognition performance. The main limitation is sequence information decay. A significant issue with RNNs is their struggle with long sequences, which can potentially lead to gradient vanishing or explosion.

Optimizations of the CRNN structure, such as MA-CRNN [39], enhance semantic information extraction and address challenges by incorporating asymmetric convolutional layers and feature reuse networks. These include integrating attention mechanisms for improved contextual understanding and long-text prediction. Other optimizations include R2AM [22], GCRNN [10], and Rosetta [5], which leverage attention modeling, gated mechanisms, and complementary FPNs alongside advanced RNNs such as LSTMs and GRUs to enhance text recognition capabilities.

2.2 Transformer-based Models

ViTSTR Vision Transformer for Scene Text Recognition (ViTSTR) [2] is a scene text recognition model that utilizes transformer architecture. Inspired by the success of the Vision Transformer (ViT) in image classification, ViTSTR applies these principles to scene text recognition. ViT demonstrated that state-of-the-art (SOTA) ImageNet recognition results could be achieved solely with the transformer's encoders in series. ViT retains all transformer properties, including speed and computational efficiency. Unlike recognizing a single object class, ViTSTR must identify multiple characters in the correct sequence and length. Predictions are made in parallel in ViTSTR, which significantly increases the reasoning speed. ViTSTR represents the application of transformer models in the STR domain. It is a comprehensive transformer encoder-decoder model

leveraging the attention mechanism. It features a low-parameter, efficient computational design within the STR model framework. Its main advantage is speed with maintained accuracy. Additionally, the model's accuracy achieved SOTA at that time.

PARSeq PARSeq [4] is the state-of-the-art (SOTA) model that utilizes Transformer architecture without relying on pre-trained weights. PARSeq is characterized by learning a collection of internal autoregressive (AR) language models (LMS) with shared weights through substitution language modeling. The Permuted Sequence Modeling (PSM) technique was used to further optimize the relationship between characters in text prediction.

Traditional STR recognition approaches output the model sequentially from left to right, right to left, or by combining both directions [55]. However, the authors of PARSeq argue that the correlation of character information in English texts with their adjacent characters is not strong, but rather random. For instance, when identifying the letter "o" in the word "model," the context "m_" (identified from left to right) differs from "_del" (identified from right to left). "_del" is less likely to be semantically related to "o", whereas the fixed phrase "m_de" has a more direct relation to "o". Building on this concept, PARSeq introduces Permuted Sequence Modeling (PSM), which utilizes a random mask $\mathcal{M}$ in attention operations to create random dependencies between input contexts.

The structural diagrams of ViTSTR and PARSeq indicate that both models belong to the transformer encoder-decoder category. Both models use a VIT-encoder for encoding. ViTSTR's decoder includes a fully connected layer for the final output. Meanwhile, PARSeq uses an optimized transform-decoder structure. Unlike later text-image multimodal models, ViTSTR and PARSeq achieve remarkable performance without any pre-training, and they have fewer parameters than text-image multimodal models.

2.3 Image-Text multimodal models

Multimodal models have gained significant attention in Scene Text Recognition (STR) due to their versatility and effectiveness. Integrating text and image information, given their rich correlation, effectively enhances scene text recognition performance. Text-image multimodal models, an innovative approach, integrate both semantic text information and contextual visual features from images. In multimodal models, a fusion layer typically integrates text embeddings and image features, enabling the model to process semantic information from both sources. Attention mechanisms enable models to dynamically focus on text-relevant image regions, enhancing text detection and recognition accuracy.

ABINet ABINet, a prominent STR model, utilizes a multimodal structure for language and images. It blocks gradient flow between visual and linguistic models to explicitly model language. It introduces a Bi-directional Completion Network (BCN) language model based on bidirectional feature representation.

Additionally, it proposes iterative correction of the language model to effectively mitigate noisy input. The network structure involves inputting images into the vision model to extract features and predict outcomes, using these predictions to derive linguistic features and make further predictions in the language model, combining visual and linguistic features for enhanced predictions, and iteratively refining these combined predictions in the language model to achieve final outcomes.

with LLM Models Given the proven significant effects of Large Language Models (LLMs), their use in STR modeling is increasing. The application of large language models in STR notably enhances the modeling of contextual information. Through pre-training on extensive corpora, these models learn profound linguistic representations, enabling a deeper understanding of word-context relationships. This capability is crucial for processing texts rich in semantic information, including those with polysemous words, specialized terminology, or domain-specific contexts.

Additionally, the use of large language models partially addresses the semantic limitations of traditional STR models. Incorporating linguistic models enhances the STR model's understanding of text semantics and its ability to recognize complex scenes and lengthy texts.

TrOCR, Clip4STR, and DtrOCR are well-known STR models that combine large language models and achieve the state-of-the-art (SOTA) level. TrOCR initializes its decoder using the RoBERTa [24] and MiniLM [44] models. Clip4STR uses Clip [31] to initialize the model, while DtrOCR employs GPT-2 [32] as its decoder. TrOCR and Clip4STR display more prominent transformer encoder-decoder structures with LLM. Conversely, DtrOCR can be considered a simpler model, with its encoder performing linear chunking and the decoder based on a LLM.

Large language models used in STR significantly increase the number of model parameters and the difficulty of training. Furthermore, since the language model is integrated through pre-training, the STR model necessitates a significantly larger training sample size compared to traditional STR models. Therefore, the enhanced STR accuracy metrics achieved through the integration of language models do not directly demonstrate structural optimization but may stem from the larger training sample size.

Fig. 3 illustrates the details of the proposed XInput methods and STR models combined with LLM. Compared to other text-image multimodal models, the language input of XInput consists of ground truth labels with masks, unlike other approaches that use image output as input. The key difference is that in the XInput method, text information primarily assists in teaching the image encoding side to extract features, whereas other language models mainly correct decoder errors and enhance the encoder.

In Parseq's research, they assess the performance of ABINet's language model (LM) using the ground truth label as input (Tab. 10 in [4]). It can be inferred that when the pre-trained LLM is integrated with xinput and the model is frozen

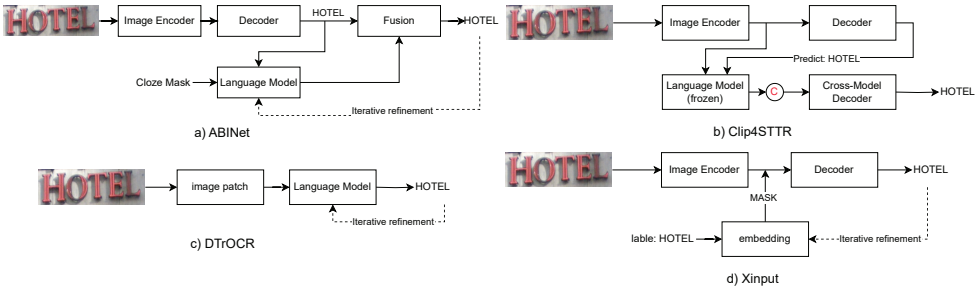


Fig. 3: Comparison of Text-Image Multimodal Models. a) ABINet combines a vision encoder-decoder, using its output as the input for the language model. Both vision and language components are trainable. b) Clip4STR utilizes a large language model (CLIP) and freezes it during training. The language model's input is also the output of the vision component. c) DTrOCR omits the visual component and solely relies on a large language model. d) XInput: The final output is derived from the vision component, where the text input is the ground truth label text, distinct from the vision output.

instead of fine-tuned, it fails to effectively utilize the ground truth label to aid in the training of the visual component. Moreover, the significant resources required for fine-tuning a large model only result in marginal improvements in its effectiveness. Consequently, we will shift our focus away from integrating the STR model structure with a large language model.

3 Text Encoding Input Method

We introduce the Text Encoding Input (XInput) training approach as a universal method applicable to all existing text recognition models. PARSeq, a state-of-the-art model that does not require pre-training, exemplifies how XInput can enhance model performance. This section first outlines XInput's model architecture. Next, we will discuss PARSeq-XInput, which merges PARSeq with XInput, and briefly review ViTSTR and ViTSTR-XInput, integrating ViTSTR with XInput.

3.1 Model Architecture

Based on previous analyses, it can be argued that most current STR models have a structure similar to a transformer encoder-decoder. During training, the masked ground truth is concatenated with the encoder's output, whereas during inference, a generated blank tensor replaces the ground truth. In a broad sense, the XInput method can be integrated into an encoder-decoder structure, as illustrated in Fig. 4,

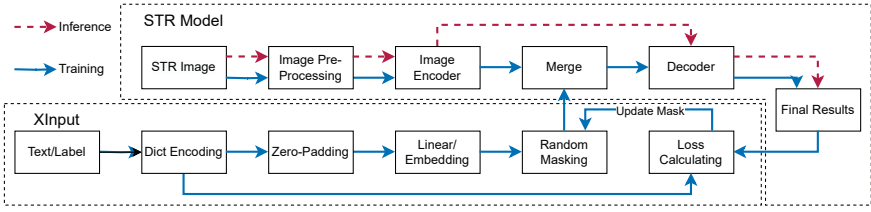


Fig. 4: Diagram of XInput architecture and training/inference. Details of the encoder/decoder layers are omitted due to space constraints. Image Pre-Processing encompasses normalization, augmentation, and additional steps. Dict Encoding transforms text into an encoding. Linear/Embedding handles the text encoding process. Zero-Padding applies a zero-complement operation to align the size of the image encoder output, incorporating group batching. The output size of XInput should be $B \times L_{max} \times E_s$, where B represents the batch size, L_{max} denotes the maximum text length, and E_s is the embedding size parameter of the image encoder. Merge combines text encoding with image encoding and integrates a new encoding into the decoder.

3.2 Vision Encoder

The XInput method utilizes the same vision encoder as the original STR models, typically the ViT encoder. The vision encoder's role is to extract intrinsic information from the $H \times W \times C$ image matrix and provide it to the decoder to achieve the desired output, where H, W, and C represent the dimensions of the input image. In an ideal STR model scenario, the encoder should extract complete textual information and disregard errors caused by background noise, interference, and distortion. The decoder should accurately convert complete picture information into text. During model initialization, without a pre-trained model, the encoder fails to provide valid information to the decoder, consequently rendering the decoder unable to generate valid information. Well-designed hyperparameters can ensure rapid model convergence and the timely delivery of accurate information. However, it is not guaranteed that all interference will be eliminated. Given that most data lacks interference, there is a high chance that some challenging interference samples will be retained, encoded by the encoder, and passed to the decoder. The decoder treats these samples as valid information.

3.3 Text encoding

Text information must be encoded before any calculations in the model network. XInput utilizes two encoding methods for text information. The first method, known as direct dictionary mapping, assigns fixed integer values to characters using a dictionary. This method offers the advantage of being intuitive and concise. However, a major drawback is the length of the dictionary, which can lead to significant discrepancies between the absolute values of text inputs and the vision encoder's outputs, creating challenges during training. The second method adds an embedding module after dictionary encoding. This approach facilitates model

convergence and is compatible with the language model's architecture due to the inclusion of an embedding layer. However, this method increases the number of parameters in the model. Text information enhances the initial, intermediate, and final stages of model training. During initialization, the vision encoder is randomly initialized, and its output contains invalid information, while the text input introduces complete required output of the decoder. At this stage, the decoder's task can be seen as replicating the introduced textual information.

Initial stage For simplicity, we omit the batch dimension by default. We represent textual information as $X = x_1, x_2, \ldots, x_n$. After encoding and expanding the text, we obtain the encoded text information $\mathbf{X}_{input} \in \mathbb{D}^{L_t \times E_s}$, where $\mathbb{D}$ represents the encoding space for encoding text information; L_t represents the text length; E_s represents the image embedding size. For the simplest case, $\mathbb{D}$ represents the space for encode the input text to the dictionary space. And it may also represent for an embedding space for the text input, as we discussed earlier. Eq. (1) shows the model training process in the initialization phase.

$$\mathbf{y} = T^{-1}(\mathbf{X}_{input}) = Decoder(Encoder(img)|\mathbf{X}_{input}) \quad (1)$$

where y represent forecast the output, $T^{-1}(\mathbf{X}_{input})$ on behalf of the inverse operation of text encoding input; *Decoder* represents decoder operation, $Encoder(img)$represent image coding operation, and $|$represents the stitching of image coding information and text coding information.

During the initialization phase, with the complete $\mathbf{X}_{input}$, the image-text encoder introduces what is considered noisy input. Meanwhile, the decoder focuses on learning the inverse process of the output, denoted as $T^{-1}()$. When only the dictionary-encoded label is applied, the operator is trained to function as the inverse embedding operator. When the embedding layer is added to XInput, the operator is trained to mimic an identity matrix. In both cases, the process should be straightforward, allowing for rapid convergence.

Intermediate stage Once the model is sufficiently trained and has significantly converged, the decoder should be capable of deriving more accurate outputs from the text part, while the vision encoder may temporarily provided few information. However, since no known label information is input during the inference process, the importance of text information should be gradually reduced in the training process. Consequently, vision encoding information should be gradually integrated into the decoder, transitioning from Eq. (1) to Eq. (2).

$$\mathbf{y} = T^{-1}(\mathbf{X}_{input}) = Decoder(Encoder(img)|(\mathbf{X}_{input} \cdot \mathbb{M} + pad_{id} \times (\mathbf{1} - \mathbb{M}))) \quad (2)$$

where $\mathbb{M}$ represents a random 0,1 mask matching the size of $\mathbf{X}_{input}$; pad_{id} present padding tokens.

A portion of the text information is masked and then supplemented with filler information before being combined with image information for input into the

decoder module, as indicated in Eq. (2) . Consequently, due to the masked text information, the initial stage of the decoder model fails to accurately produce the expected output y. The missing information can only be sourced from the image encoding part at this stage. Thus, the model gradually transitions from relying on textual to image information.

The crucial aspect of batch design involves selecting an appropriate random threshold ratio to retain the text information. In XInput, the ratio for the current batch is determined by considering the loss from the previous batch.

$$ratio = \alpha \cdot (Loss - \beta) \tag{3}$$

where *ratio* denotes the randomness rate of the random mask. A higher *ratio* means more ones in the mask and consequently, more text information is preserved; a lower *ratio* results in less text information being preserved. α and β are floating-point numbers that link the loss value *Loss* with the *ratio*. *Loss* signifies the loss value of the model's last batch. Automating the determination of the ratio based on the loss value enables a smooth transition from initialization through intermediate training to the final phase. A high loss suggests that the model has not converged, indicating the need for a higher ratio. A decreasing loss indicates that the model is progressively extracting valid information, allowing for a reduction in the ratio to facilitate more information extraction from the image. After reducing the loss, backtracking may occur due to insufficient text information. However, based on the information extracted from the image, the loss should be lower than before, indicating ongoing model convergence. Furthermore, the information obtained from the image should result in a reduced loss backtracking amplitude compared to before, ensuring continuous model convergence.

Final stage and inference When the loss falls below the threshold β, the text side stops providing information, enabling full information extraction from the image side and transitioning to the form described in Eq. (4). If difficult samples cause an increase in loss, the model will revert to using information from the text side to support ongoing training, returning to the form outlined in Eq. (2).

$$\mathbf{y} = T^{-1}(\mathbf{X}_{input}) = Decoder(Encoder(img)|pad_{id} \cdot \mathbf{1}) \tag{4}$$

It can be seen that when the model loss is stable below the value of β, the STR model supplemented with the XInput method essentially reverts to the original model. During the reasoning process, the text side does not provide additional information. In the worst case scenario, a STR model supplemented with XInput should have the same effect as the original STR model.

3.4 XInput with SOTA models

This article primarily compares the impact of XInput on two STR models: ViTSTR and PARSeq. As previously mentioned, the XInput method can be integrated into other STR models with an encoder-decoder structure, ensuring

that, at the very least, it is as effective as the original STR model. ViTSTR employs a fully connected layer immediately after the encoder for its final output. PARSeq utilizes the Transform-decoder structure for decoding. The final layer after the transformer decoder is also a fully connected layer. By comparing the model integrated with the XInput method to the original model, we aim to demonstrate that the XInput method is not only applicable to transform-based encoder-decoders but also enhances the fully connected layer. Furthermore, we can generalize that combining XInput can enhance most STR models.

3.5 XInput with other models

In addition, it is worth explaining that, as previously analyzed, CRNN can indeed be categorized into an encoder-decoder structure in a broad sense, where CNN serves as the image encoder and RNN as the decoder. Thus, the CRNN model can also integrate the XInput method. However, RNNs is designed for the sequential data, which means that theoretically, combining image and text features does not optimize RNN's backpropagation training. Our experiments also demonstrate that it is challenging for RNNs to extract information from both the text encoding input and the CNN component simultaneously in order to mutually optimize each other. Consequently, we do not present a model structure that combines CRNN with XInput.

The STR model, when combined with LLM, can also incorporate our proposed XInput method, as it aligns well with coding and decoding structures. However, as all other STR models integrating language models utilize pre-trained language models, the choice of these pre-trained models significantly influences the final STR model's accuracy metrics. We will not specifically assessed the impact of combining these models with XInput in subsequent evaluations.

4 Experiment

4.1 Dataset

Following previous work [3, 4], we utilize the same synthetic datasets as traditional STR models: MJSynth (MJ) [15] with 9M samples and SynthText (ST) [14] with 6.9M samples. Additionally, we employ COCO-Text (COCO) [41], RCTW17 [36], Uber-Text (Uber) [53], ArT [6], LSVT [38], MLT19 [28], and ReCTS [52]. A comprehensive discussion on these datasets is available in Baek et al [3]. Furthermore, we utilized two recent large-scale real datasets based on Open Image [20]: TextOCR [37] and annotations from the OpenVINO toolkit [21].

For testing, based on prior studies, we utilize IIIT 5k-word (IIIT5k) [26], CUTE80 (CUTE) [33], Street View Text (SVT) [43], SVT-Perspective (SVTP) [29], ICDAR 2013 (IC13) [18], and ICDAR 2015 (IC15) [17] for evaluation, as benchmark. Case-sensitive annotations for IIIT5k, CUTE, SVT, and SVTP are based on Long and Yao [25]. It is noted that IC13 and IC15 often appear in two versions in literature - with IC13 having 857 and 1,015 test samples, and IC15

having 1,811 and 2,077 samples. To avoid confusion, we collectively refer to these benchmarks as IIIT5k, CUTE, SVT, SVTP, IC13(1,015), and IC15(2,077). The six benchmark datasets collectively contain 7,672 test samples, termed the ALL test set, while the versions with IC13 (857) and IC15 (1,811) samples are known as the benchmark (subset).

4.2 Experimental details

XInput variants The XInput structure is integrated after the image encoding phase of the generalized encoder. The size of $\mathbf{X}_{input}$ is defined as $\in \mathbb{Q}^{B\times S\times E_s}$, where B denotes the batch size and S represents the fixed length of text encoding. The value of the variable depends on the labeled data and should be greater than or equal to the maximum length of the label L_{max}. For STR English word recognition, S is set to 25 in this study, unless noted otherwise.

The text content is dynamically masked based on the model training's loss values, as demonstrated in Eq. (3). Various factors, such as datasets, loss functions, model structures, and hyperparameter settings, impact the loss function during model training. Therefore, using fixed α or β values for different models is not advisable. This paper outlines the following method for setting relevant parameters. First, we train the model without XInput until it stabilizes, recording the stable loss value and the number of rounds required to reach this state. If the model stabilizes quickly, we set a smaller α to accelerate the reduction of textual information during training. Conversely, we set a larger α to retain more textual information for training support. Simultaneously, we set β less than the final loss value to ensure model stabilization at later stages, with a $ratio \leq 0$, blocking text information completely.

Other variants For parameters unrelated to XInput, we adopt the optimal settings for PARSeq, CRNN, and ViTSTR models as outlined in [4], with the exception of the maximum training epochs. In subsequent result presentations, any deviations in parameters will be noted. Adding text information increases the number of epochs needed for model convergence and stabilization. In our experiments, the final number of training epochs is determined by the model's loss curve. For comparison, other models are trained for an equivalent number of epochs.

Hardware and Software The experimental setup includes a dual V100 GPU configuration, Cuda V11.7, Python 3.9.18, and PyTorch 2.1.2. Additional details can be found on GitHub.

4.3 Comparison of State-of-the-art

In Tab. 1 and Tab. 2, we compare the XInput methods against previous state-of-the-art (SOTA) methods, which do not use external pre-trained models, across

Method	Venue	Train Dataset	Paper Cite Code Repoduce	IIIT5K 3,000	SVT 647	IC13 1,015	IC15 1,811	IC15 2,077	SVTP 645	CUTE 288
ASTER [35]	PAMI'19	MJ+ST	paper [54]	93.4	89.5	–	76.1	–	78.5	79.5
SRN [49]	CVPR'20	MJ+ST	paper [54]	94.8	91.5	–	82.7	–	85.1	87.8
TextScanner [42]	AAAI'20	MJ+ST	paper [54]	95.7	92.7	94.9	–	83.5	84.8	91.6
SE-ASTER [30]	CVPR'20	MJ+ST	paper [54]	93.8	89.6	92.8	80	–	81.4	83.6
TRBA [3]	CVPR'21	MJ+ST	paper [54]	92.1	88.9	–	86	–	89.3	89.2
VisionLAN [45]	ICCV'21	MJ+ST	paper [54]	95.8	91.7	–	83.7	–	86	88.5
ABINet [11]	CVPR'21	MJ+ST	paper [54]	96.2	93.5	–	86	–	89.3	89.2
ViTSTR-B [2]	ICDAR'21	MJ+ST	paper [54]	88.4	87.7	92.4	78.5	72.6	81.8	81.3
ViTSTR-S [2]	ICDAR'21	MJ+ST	Code	94.2	93.8	94.2	82.8	79.1	85.1	87.5
LevOCR [8]	ECCV'22	MJ+ST	paper [54]	96.6	92.9	–	86.4	–	88.1	91.7
MATRN [27]	ECCV'22	MJ+ST	paper [54]	96.6	95	95.8	86.6	82.8	90.6	**93.5**
PETR [46]	TIP'22	MJ+ST	paper [54]	95.8	92.4	97	83.3	–	86.2	89.9
DiG-ViT-B [47]	MM'22	MJ+ST	paper [54]	96.7	94.6	96.9	**87.1**	–	**91.0**	91.3
PARSeqA [4]	ECCV'22	MJ+ST	paper [54]	97	93.6	96.2	86.5	82.9	88.9	92.2
SIGAT [13]	CVPR'23	MJ+ST	paper [54]	96.6	**95.1**	**96.8**	86.6	83	90.5	93.1
ViTSTR+XInput	ours	MJ+ST	Code	94.8	91.3	95.5	84.8	80.7	86.8	89.9
PARSeq+XInput	ours	MJ+ST	Code	**97.5**	94.1	96.3	**87.1**	**83.4**	90.7	93.4

Table 1: For Synthetic datasets, Word accuracy on the six benchmark datasets (36-char). Synthetic datasets - MJ [30] and ST [28]; Benchmark datasets - SVT, IIIT5k, IC13, IC15, SVTP, and CUTE; In our experiments, bold indicates the highest word accuracy per column.

Method	Venue	Train Dataset	Paper Cite Code Repoduce	IIIT5K 3,000	SVT 647	IC13 1,015	IC15 1,811	IC15 2,077	SVTP 645	CUTE 288
PARSeq+CLIPTER [1]	ICCV'23	N/A	paper [54]	–	96.6	–	–	85.9	–	–
DiG-ViT-B [47]	MM'22	Real(2.8M)	paper [54]		96.5	97.6	88.9	–	92.9	96.5
ViTSTR-S [2]	ICDAR'21	Real(3.3M)	paper [54]	97.9	96	97.8	89	87.5	91.5	96.2
ViTSTR-S [2]	ICDAR'21	Real(3.3M)	Code	98.0	95.0	97.2	88.3	87.4	91.8	97.6
ABINet [11]	CVPR'21	Real(3.3M)	paper [54]	98.6	**98.2**	98	90.5	88.7	94.1	97.2
PARSeqA [4]	ECCV'22	Real(3.3M)	paper [54]	99.1	97.9	**98.4**	90.7	**89.6**	95.7	98.3
MAERec-B [16]	ICCV'23	Union14M-L [16]	paper [54]	98.5	97.8	98.1	-	89.5	94.4	**98.6**
ViTSTR+XInput		Real(3.3M)	Code	97.8	96.0	97.6	88.2	86.7	92.6	96.2
PARSeq+XInput		Real(3.3M)	Code	**99.2**	97.5	98.1	**90.8**	**89.6**	**96.0**	97.6

Table 2: For Synthetic datasets, Word accuracy on the six benchmark datasets (36-char). Real datasets - COCO, RCTW17, Uber, ArT, LSVT, MLT19, ReCTS, TextOCR, and OpenVINO; Benchmark datasets - SVT, IIIT5k, IC13, IC15, SVTP, and CUTE. In our experiments, bold indicates the highest word accuracy per column.

six common STR benchmarks. The XInput methods effectively enhance the original model's performance.

In Tab. 3, with larger and more challenging datasets, the model significantly improves after combining XInput. It is important to note that the average data presented in Tab. 3 originate from the results of the first three models in the validation set and the final model in optimal model training. Theoretically, utilizing more suitable parameters should lead to higher indicators than those currently reported.

5 Conclusion

We introduce the XInput method, designed to complement the STR model by enabling parallel computation within any encoder-decoder structure. We argue that integrating XInput improves the compatibility of STR networks with human learning styles during training. Experiments show that PARSeq models enhanced with XInput outperform the baseline PARSeq across 11 benchmark test sets. These optimal outcomes are achieved without the necessity of additional pre-trained large models. Additionally, it has been demonstrated that XInput can

Method	Train data	ArT 35,149	COCO 9,825	Uber 80,551	Total 125,525
CRNN	MJ+ST	57.3±0.1	49.3±0.6	33.1±0.3	41.1±0.3
ViTSTR-S	MJ+ST	66.1±0.1	56.4±0.5	37.6±0.3	47.0±0.2
TRBA	MJ+ST	68.2±0.1	61.4±0.4	38.0±0.3	48.3±0.2
ABINet	MJ+ST	65.4±0.4	57.1±0.8	34.9±0.3	45.2±0.3
PARSeqN	MJ+ST	69.1±0.2	60.2±0.8	39.9±0.5	49.7±0.3
PARSeqA	MJ+ST	70.7±0.1	64.0±0.9	42.0±0.5	51.8±0.4
PARSeq+XInput	MJ+ST	71.0±0.1	64.6±0.1	43.0±0.3	52.5±0.1
ViTSTR+XInput	MJ+ST	66.7±0.2	56.2±1.3	38.4±0.3	47.7±0.1
CRNN	Real(3.3M)	66.8±0.2	62.2±0.3	51.0±0.2	56.3±0.2
ViTSTR-S	Real(3.3M)	81.1±0.1	74.1±0.4	78.2±0.1	78.7±0.1
TRBA	Real(3.3M)	82.5±0.2	77.5±0.2	81.2±0.3	81.3±0.2
ABINet	Real(3.3M)	81.2±0.1	76.4±0.1	71.5±0.7	74.6±0.4
PARSeqN	Real(3.3M)	83.0±0.2	77.0±0.2	82.4±0.3	82.1±0.2
PARSeqA	Real(3.3M)	84.5±0.1	79.8±0.1	84.5±0.1	84.1±0.0
PARSeq+XInput	Real(3.3M)	84.0±0.2	78.6±0.4	85.0±0.5	84.2±0.4
ViTSTR+XInput	Real(3.3M)	81.1±0.1	73.1±0.4	78.8±0.2	79.0±0.1

Table 3: 36-char word accuracy on larger and more challenging datasets

be integrated with various parallel encoders to enhance model performance, as exemplified by ViTSTR. We believe that the XInput method offers a promising direction for future STR model research.

References

1. Aberdam, A., Bensaïd, D., Golts, A., Ganz, R., Nuriel, O., Tichauer, R., Mazor, S., Litman, R.: Clipter: Looking at the bigger picture in scene text recognition. arXiv preprint arXiv:2301.07464 (2023) 13
2. Atienza, R.: Vision transformer for fast and efficient scene text recognition. In: International Conference on Document Analysis and Recognition. pp. 319–334. Springer (2021) 2, 3, 4, 13
3. Baek, J., Matsui, Y., Aizawa, K.: What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3113–3122 (2021) 11, 13
4. Bautista, D., Atienza, R.: Scene text recognition with permuted autoregressive sequence models. In: European Conference on Computer Vision. pp. 178–196. Springer Nature Switzerland, Cham (10 2022). https://doi.org/10.1007/978-3-031-19815-1_11, https://doi.org/10.1007/978-3-031-19815-1_11 2, 3, 5, 6, 11, 12, 13
5. Borisyuk, F., Gordo, A., Sivakumar, V.: Rosetta: Large scale system for text detection and recognition in images. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 71–79 (2018) 4
6. Chng, C.K., Liu, Y., Sun, Y., Ng, C.C., Luo, C., Ni, Z., Fang, C., Zhang, S., Han, J., Ding, E., et al.: Icdar2019 robust reading challenge on arbitrary-shaped text-

rrc-art. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1571–1576. IEEE (2019) 11
7. Coquenet, D., Chatelain, C., Paquet, T.: Dan: a segmentation-free document attention network for handwritten document recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence (2023) 2
8. Da, C., Wang, P., Yao, C.: Levenshtein ocr. In: European Conference on Computer Vision. pp. 322–338. Springer (2022) 2, 13
9. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., et al.: An image is worth 16x16 words: Transformers for image recognition at scale. arXiv preprint arXiv:2010.11929 (2020) 1
10. Elbasani, E., Njimbouom, S.N., Oh, T.J., Kim, E.H., Lee, H., Kim, J.D.: Gcrnn: graph convolutional recurrent neural network for compound–protein interaction prediction. BMC bioinformatics **22**(5), 1–14 (2021) 3, 4
11. Fang, S., Xie, H., Wang, Y., Mao, Z., Zhang, Y.: Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7098–7107 (2021) 2, 3, 13
12. Fujitake, M.: Dtrocr: Decoder-only transformer for optical character recognition. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 8025–8035 (2024) 2, 3
13. Guan, T., Gu, C., Tu, J., Yang, X., Feng, Q., Zhao, Y., Shen, W.: Self-supervised implicit glyph attention for text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 15285–15294 (2023) 13
14. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2315–2324 (2016) 11
15. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. arXiv preprint arXiv:1406.2227 (2014) 11
16. Jiang, Q., Wang, J., Peng, D., Liu, C., Jin, L.: Revisiting scene text recognition: A data perspective. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 20543–20554 (2023) 13
17. Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V.R., Lu, S., et al.: Icdar 2015 competition on robust reading. In: 2015 13th international conference on document analysis and recognition (ICDAR). pp. 1156–1160. IEEE (2015) 11
18. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazan, J.A., De Las Heras, L.P.: Icdar 2013 robust reading competition. In: 2013 12th international conference on document analysis and recognition. pp. 1484–1493. IEEE (2013) 11
19. Khan, S., Naseer, M., Hayat, M., Zamir, S.W., Khan, F.S., Shah, M.: Transformers in vision: A survey. ACM computing surveys (CSUR) **54**(10s), 1–41 (2022) 2
20. Krasin, I., Duerig, T., Alldrin, N., Ferrari, V., Abu-El-Haija, S., Kuznetsova, A., Rom, H., Uijlings, J., Popov, S., Veit, A., et al.: Openimages: A public dataset for large-scale multi-label and multi-class image classification. Dataset available from https://github. com/openimages **2**(3), 18 (2017) 11
21. Krylov, I., Nosov, S., Sovrasov, V.: Open images v5 text annotation and yet another mask text spotter. In: Asian Conference on Machine Learning. pp. 379–389. PMLR (2021) 11

22. Lee, C.Y., Osindero, S.: Recursive recurrent nets with attention modeling for ocr in the wild. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2231–2239 (2016) 4
23. Li, M., Lv, T., Chen, J., Cui, L., Lu, Y., Florencio, D., Zhang, C., Li, Z., Wei, F.: Trocr: Transformer-based optical character recognition with pre-trained models. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 13094–13102 (2023) 2
24. Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V.: Roberta: A robustly optimized bert pretraining approach. arXiv preprint arXiv:1907.11692 (2019) 6
25. Long, S., Yao, C.: Unrealtext: Synthesizing realistic scene text images from the unreal world. arXiv preprint arXiv:2003.10608 (2020) 11
26. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: BMVC-British machine vision conference. BMVA (2012) 11
27. Na, B., Kim, Y., Park, S.: Multi-modal text recognition networks: Interactive enhancements between visual and semantic features. In: European Conference on Computer Vision. pp. 446–463. Springer (2022) 2, 13
28. Nayef, N., Patel, Y., Busta, M., Chowdhury, P.N., Karatzas, D., Khlif, W., Matas, J., Pal, U., Burie, J.C., Liu, C.l., et al.: Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In: 2019 International conference on document analysis and recognition (ICDAR). pp. 1582–1587. IEEE (2019) 11
29. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: Proceedings of the IEEE international conference on computer vision. pp. 569–576 (2013) 11
30. Qiao, Z., Zhou, Y., Yang, D., Zhou, Y., Wang, W.: Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 13528–13537 (2020) 13
31. Radford, A., Kim, J.W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., et al.: Learning transferable visual models from natural language supervision. In: International conference on machine learning. pp. 8748–8763. PMLR (2021) 6
32. Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al.: Language models are unsupervised multitask learners. OpenAI blog **1**(8), 9 (2019) 6
33. 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**(18), 8027–8048 (2014) 11
34. 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**(11), 2298–2304 (2016) 1
35. Shi, B., Yang, M., Wang, X., Lyu, P., Yao, C., Bai, X.: Aster: An attentional scene text recognizer with flexible rectification. IEEE transactions on pattern analysis and machine intelligence **41**(9), 2035–2048 (2018) 13
36. Shi, B., Yao, C., Liao, M., Yang, M., Xu, P., Cui, L., Belongie, S., Lu, S., Bai, X.: Icdar2017 competition on reading chinese text in the wild (rctw-17). In: 2017 14th iapr international conference on document analysis and recognition (ICDAR). vol. 1, pp. 1429–1434. IEEE (2017) 11
37. Singh, A., Pang, G., Toh, M., Huang, J., Galuba, W., Hassner, T.: Textocr: Towards large-scale end-to-end reasoning for arbitrary-shaped scene text. In: Proceedings of

the IEEE/CVF conference on computer vision and pattern recognition. pp. 8802–8812 (2021) 11
38. Sun, Y., Ni, Z., Chng, C.K., Liu, Y., Luo, C., Ng, C.C., Han, J., Ding, E., Liu, J., Karatzas, D., et al.: Icdar 2019 competition on large-scale street view text with partial labeling-rrc-lsvt. In: 2019 International Conference on Document Analysis and Recognition (ICDAR). pp. 1557–1562. IEEE (2019) 11
39. Tong, G., Li, Y., Gao, H., Chen, H., Wang, H., Yang, X.: Ma-crnn: a multi-scale attention crnn for chinese text line recognition in natural scenes. International Journal on Document Analysis and Recognition (IJDAR) **23**, 103–114 (2020) 4
40. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. Advances in neural information processing systems **30** (2017) 1
41. Veit, A., Matera, T., Neumann, L., Matas, J., Belongie, S.: Coco-text: Dataset and benchmark for text detection and recognition in natural images. arXiv preprint arXiv:1601.07140 (2016) 11
42. Wan, Z., He, M., Chen, H., Bai, X., Yao, C.: Textscanner: Reading characters in order for robust scene text recognition. In: Proceedings of the AAAI conference on artificial intelligence. vol. 34, pp. 12120–12127 (2020) 13
43. Wang, K., Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International conference on computer vision. pp. 1457–1464. IEEE (2011) 11
44. Wang, W., Wei, F., Dong, L., Bao, H., Yang, N., Zhou, M.: Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers. Advances in Neural Information Processing Systems **33**, 5776–5788 (2020) 6
45. Wang, Y., Xie, H., Fang, S., Wang, J., Zhu, S., Zhang, Y.: From two to one: A new scene text recognizer with visual language modeling network. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14194–14203 (2021) 13
46. Wang, Y., Xie, H., Fang, S., Xing, M., Wang, J., Zhu, S., Zhang, Y.: Petr: Rethinking the capability of transformer-based language model in scene text recognition. IEEE Transactions on Image Processing **31**, 5585–5598 (2022) 13
47. Yang, M., Liao, M., Lu, P., Wang, J., Zhu, S., Luo, H., Tian, Q., Bai, X.: Reading and writing: Discriminative and generative modeling for self-supervised text recognition. In: Proceedings of the 30th ACM International Conference on Multimedia. pp. 4214–4223 (2022) 2, 13
48. Ye, J., Hu, A., Xu, H., Ye, Q., Yan, M., Xu, G., Li, C., Tian, J., Qian, Q., Zhang, J., et al.: Ureader: Universal ocr-free visually-situated language understanding with multimodal large language model. arXiv preprint arXiv:2310.05126 (2023) 2
49. Yu, D., Li, X., Zhang, C., Liu, T., Han, J., Liu, J., Ding, E.: Towards accurate scene text recognition with semantic reasoning networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 12113–12122 (2020) 13
50. Yuan, Y., Fu, R., Huang, L., Lin, W., Zhang, C., Chen, X., Wang, J.: Hrformer: High-resolution vision transformer for dense predict. Advances in Neural Information Processing Systems **34**, 7281–7293 (2021) 2
51. Zeng, F., Gan, W., Wang, Y., Liu, N., Yu, P.S.: Large language models for robotics: A survey. arXiv preprint arXiv:2311.07226 (2023) 2
52. Zhang, R., Zhou, Y., Jiang, Q., Song, Q., Li, N., Zhou, K., Wang, L., Wang, D., Liao, M., Yang, M., et al.: Icdar 2019 robust reading challenge on reading chinese text on signboard. In: 2019 international conference on document analysis and recognition (ICDAR). pp. 1577–1581. IEEE (2019) 11

53. Zhang, Y., Gueguen, L., Zharkov, I., Zhang, P., Seifert, K., Kadlec, B.: Uber-text: A large-scale dataset for optical character recognition from street-level imagery. In: SUNw: Scene Understanding Workshop-CVPR. vol. 2017, p. 5 (2017) 11
54. Zhao, S., Wang, X., Zhu, L., Yang, Y.: Clip4str: A simple baseline for scene text recognition with pre-trained vision-language model. arXiv preprint arXiv:2305.14014 (2023) 2, 13
55. Zhao, W., Gao, L., Yan, Z., Peng, S., Du, L., Zhang, Z.: Handwritten mathematical expression recognition with bidirectionally trained transformer. arXiv preprint arXiv:2105.02412 (2021) 5