2023 IEEE International Conference on Image Processing (ICIP) | 978-1-7281-9835-4/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIP49359.2023.10222793

DIFFUSIONSTR: DIFFUSION MODEL FOR SCENE TEXT RECOGNITION

Masato Fujitake

FA Research, Fast Accounting Co., Ltd.
fujitake@fastaccounting.co.jp

ABSTRACT

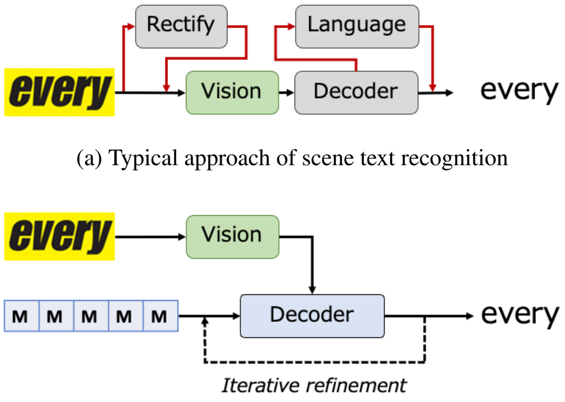
This paper presents Diffusion Model for Scene Text Recognition (DiffusionSTR), an end-to-end text recognition framework using diffusion models for recognizing text in the wild. While existing studies have viewed the scene text recognition task as an image-to-text transformation, we rethought it as a text-text one under images in a diffusion model. We show for the first time that the diffusion model can be applied to text recognition. Furthermore, experimental results on publicly available datasets show that the proposed method achieves competitive accuracy compared to state-of-the-art methods.

Index Terms— Scene text recognition, Document analysis, Diffusion model, Deep learning, Machine learning

1. INTRODUCTION

Text recognition in natural images is one of the active areas in computer vision and a fundamental and vital task in real-world applications such as document analysis and automated driving [1, 2]. However, scene text recognition is challenging because it requires recognizing text in various fonts, colors, and shapes. Many methods have been proposed to address this challenge. Early research proposed methods that utilize information from images using Convolutional Neural Networks (CNNs) and recognizes text sequences using Recurrent Neural Networks (RNNs) [3]. In order to deal with curved text images, pre-processing methods, such as Rectification Networks, were introduced before encoding visual features to improve accuracy [4]. Recently, methods with strong language models have been proposed for more robust recognition [5,6]. While these methods contribute to high accuracy, they tend to be complex because they use multiple modules.

We propose a new approach by reviewing the input-output relationship of the scene text recognition task. While existing approaches generate text sequences directly from images, as shown in Figure 1a, our method iteratively transforms a text sequence into a correctly recognized one under the condition of image information, as shown in Figure 1b. In other words, we regard the text recognition task as a text-to-text transformation task rather than an image-to-text task. By matching the input-output relationship of the same domain, rather than directly converting the different domains of image and text, prediction is possible even in simple structures, unlike existing methods.



(a) Typical approach of scene text recognition

(b) Our approach using diffusion model

Fig. 1: **The overview of the typical approach and our proposed one.** Typical approaches consist of a vision module to obtain information from an image and a decoder to convert it to a text sequence. Some methods employ rectified modules and language models to boost accuracy, which leads to complex architecture. In contrast, our approach consists of two main components. The decoder converts a sequence to a recognized result iteratively under the visual condition.

We introduce a diffusion model into the scene text recognition task to realize our approach. In recent years, generative models based on diffusion models have succeeded in image generation [7, 8]. The probabilistic diffusion models pipeline consists of a chain of Markov latent variables, with data flowing in two directions: the diffusion process and the denoising process. The denoising process generates data from Gaussian noise through inference. In contrast, the diffusion process is the training process that learns to transform data samples into Gaussian noise. We leverage a diffusion model in our proposed approach. The main structural feature of the diffusion model in the image generation task is that the input-output relationship corresponds to the same resolution—a fixed dimension—for images. Unlike image generation, however, the dimension of scene text recognition varies from image to image due to the length of a text sequence. This makes learning difficult because there is a different challenge of where a text ends within a fixed-length sequence in addition to the categorical classification of characters. To solve this problem, we propose a character-aware head, which predicts whether a character exists at the position in the sequence. By doing so, we achieved accuracy comparable to leading competitive methods despite its simple structure.

Our contribution is listed as follows.

- We propose DiffusionSTR, the first framework for scene text recognition with diffusion models.
- Experimental results show the proposed method's effectiveness, achieving competitive accuracy with state-of-the-art methods on public datasets.

2. RELATED WORKS

Scene Text Recognition. Scene Text Recognition can be roughly divided into two categories: language-free and language-based approaches. The language-free approaches predict the sequence of a character directly from input images without any language constraint. The main methods are CTC-based and segmentation-based methods. The CTC-based methods [3, 9] combine CNN to extract visual features and sequence models, such as RNN, to predict a sequence of characters with end-to-end training using CTC loss [10]. The segmentation-based methods segment characters at pixel level and recognize them by grouping [11]. However, these approaches do not use linguistic information, only image information, making them vulnerable to noise, such as occlusion and distortion.

The language-based approaches have been studied recently to alleviate the above problems [5]. Early research proposed methods utilizing N-grams [12], but recent methods have been proposed using powerful language models such as those represented by RNNs [13] and Transformer [5]. For instance, ABINet inputs the recognition results from vision into a language model and obtains text results with added language information. Then, it predicts the refined results fusing the two recognition results. This process is iterative for improvement [5]. Moreover, PARSeq uses Permutation Language Modeling to sophisticate the iterative improvement process [6]. These methods have contributed to higher accuracy by introducing powerful language models, but they are complex mechanisms.

Our proposed method differs from existing research in two ways. First, we do not use a language model for simplicity. Second, while existing research performs direct inference from images to text results, our proposed method differs because we prepare token sequences in advance and transform them to correct sequences through the vision condition.

Diffusion Models. Diffusion models are latent variable generative frameworks in [14] and are improved in [7]. Due to their remarkable power, they have recently attracted attention in image generation [7, 14, 15], mainly in continuous data. Adaptations have also been proposed for object detection [16] and audio [17]. Although initially not directly applicable to discrete data such as text, the extension to the discrete data has recently been proposed [8]. However, it has yet to be shown to what extent the diffusion model is effective in text recognition tasks. This is the first work of the diffusion models for the text recognition task.

3. METHOD

Figure 2 shows the pipeline of the proposed method. Our proposed method learns to transform text-to-text for scene text recognition through a diffusion model process based on the transformer architecture, including vision and text. The model comprises a vision encoder, a transformer, linear layers—FFN, and a character-aware head—that transform the results. The overall flow begins with generating visual features from images using the vision encoder. Next, a noise-filled token sequence x_T is used as input to generate a refined one x_{T-1} through the Transformer [19] under visual feature conditions. The new token sequence is refined T times, and finally, the output x_0 is converted to the recognized text through FFN, and the character's position is predicted through the character-aware head. We describe each detail below.

3.1. Diffusion Model

The original diffusion model [14] constructs forward and reverse processes. The forward process gradually deteriorates a data point x_0, sampled from a real-world data distribution $x_0 \sim q(x)$. It adds a small amount of noise at each time step where $t \in 1, \cdots T$ and makes the data point into a Gaussian noise $x_T \sim \mathcal{N}(0, \mathbf{I})$. On the other hand, the reverse denoising process tries to gradually reconstruct the original data x_0 through sampling from x_T. It is described as a learnable distribution $p(x_{t-1} \mid x_t)$.

In applying a diffusion model to the text recognition task, we basically follow the diffusion models [8, 14] but with some modifications. First, in the image generation task, the data x_t corresponds to a single image. However, our scene text recognition corresponds to a single text sequence comprising character tokens. Moreover, we use the multinomial diffusion model [8] for categorical data because, unlike the image generation task whose variable is continuance, scene text recognition's variable is discrete. In addition to characters, we also use special tokens to correspond to text recognition: the `[EOS]` token indicates the end of the recognized text. The `[PAD]` token indicates the padding of the fixed-length sequence after `[EOS]`. The `[MASK]` token is a special token that signifies noise state.

Next, although the original diffusion model is optimized to maximize the marginal likelihood of the data in [14], we use the devised objective L_{simple} [7] with a mean-squared error loss for stable training.

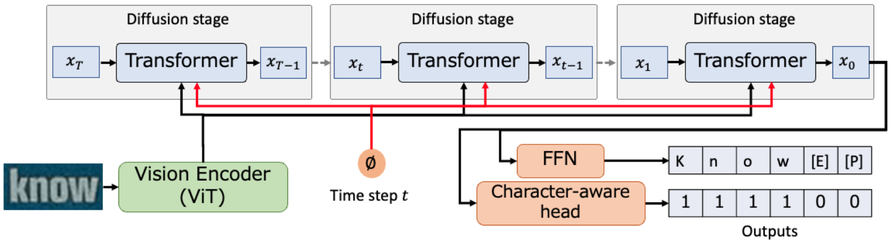


Fig. 2: **The pipeline of the proposed scene text recognition using a diffusion model.** It consists of the vision encoder [18], Transformer [19] with an additional time-based positional encoding, FFN and character-aware head. `[E]` and `[P]` are abbreviations for special tokens `[EOS]` and `[PAD]`, respectively. The proposed method performs text recognition by repeatedly refining the input sequence x_t based on image information. FFN performs character classification, while the character-aware head predicts where a character exists in a fixed-length sequence.

3.2. Vision Encoder

The vision encoder $VisEnc$ extracts information from a given image. For this purpose, we used ViT [18], an extension of Transformer [19] to the image field. That is, image $I \in \mathbb{R}^{H \times W \times C}$ is tokenized per $p_w \times p_h$ patch and encoded with 12 ViT-layer, where H, W, C are the height, width, and channel of the image. We used a learned positional encoding. Therefore, the visual features z are defined,

$$z = \mathrm{VisEnc}(I) \in \mathbb{R}^{\frac{HW}{p_w p_h} \times d}, \tag{1}$$

where d is the embedding dimension of ViT.

3.3. Transformer

We employ the Transformer decoder [19] Dec with an additional time-based positional encoding ϕ to convert token sequences in diffusion models. The architecture is similar to the decoder in the unconditional text translation with a diffusion model [8]. However, it differs in three respects. The first is the text transformation under the condition of the vision features z. We use the Transformer's cross-attention mechanism within the decoder to condition the text sequence. Second, while the Transformer typically infers one token at a time to handle text, our text recognition task simultaneously outputs the probabilities of all tokens. Third, the Transformer output results are used in two ways: one is converted into a string for text recognition using the Feedforward Network (FFN), following previous research [6]; the other is input into a character-aware head, which has an identical structure to the FFN, to predict whether a character region contains a real character, such as the alphabet. We use the cross-entropy loss for FFN.

The time positional encoding utilizes sinusoidal positional embedding and linear layers. The output is added to the typical sequence positional encoding in the Transformer and put into each decoder layer.

3.4. Character-aware Head

We propose a character-aware head for text recognition in the diffusion model. Since text recognition in the diffusion model is performed within a fixed-length sequence, there are two issues: the categorical classification of characters and the extent to which a character is a character sequence. Therefore, we propose classifying at a large level what is a character domain in a fixed-length sequence and facilitating categorical classification of characters. As shown in the output of Figure 2, the character-aware head performs a binary classification of whether the position corresponds to a character (1) or not (0). Binary Cross-entropy was used as the loss function.

3.5. Training Protocol

During training, we randomly sample a time step t. Then, we calculate the posteriors using the noise schedule $\alpha_t, \bar{\alpha}_t$, as defined in the original diffusion model, and a noisy sequence x_t as following [8]. We perform $x_{t-1} = Dec(x_t, z, t)$ to compute the loss.

3.6. Inference Protocol

The Inference process starts with the sequence x_T, filled with a mask token. The process is conditioned on z from the vision encoder. It then iterates T times to obtain x_0. We convert the final token sequence to text with FFN.

4. EXPERIMENTS

4.1. Dataset and Evaluation

For a fair comparison, we conducted experiments following the setup of [20]. We train the models on two synthetic datasets MJSynth (MJ) [12, 21] and SynthText (ST) [22]. We evaluated models on six standard benchmarks: ICDAR 2013 (IC13) [23], ICDAR 2015 (IC15) [24], IIIT 5KWords (IIIT) [25], Street View Text (SVT) [26], Street View Text-Perspective (SVTP) [27] and CUTE80 (CUTE) [28]. For evaluation, we use word-level accuracy on the six benchmark datasets. A prediction is considered correct if characters at all positions match. We report the mean score of the four experiments by following the previous research [6].

4.2. Implementation Details

All experiments are performed on four Nvidia A100 GPUs in mixed precision using PyTorch. We used the ADAMW optimizer [29] with a learning rate, which warms up to 10^{-4} linearly and drops to 0 following cosine decay. We set the hyperparameters of the optimizer ϵ and β as 10^{-8} and (0.9, 0.999), respectively. The number of training epochs is 20, and the warm-up epoch is 5. The batch size is 384. We set the weight decay to 0.01 For the Label preprocessing, we follow the previous work [13]. Concretely, we set a maximum label length of 25 and a charset size of 94, which includes mixed-case alphanumeric characters and punctuation marks without special tokens. We strictly follow the way of image preprocessing [6] including augmentation. For a fair comparison, we used a vision encoder [18] with the same settings as PARSeq [6]. We have six transformer layers with 16 attention heads and hidden dimension d of 384 for the transformer decoder. The balanced weights of the cross entropy loss function for character-aware head and character recognition are equal. We set the time step T to 1000.

4.3. Main Results

In this section, we compare the performances of the proposed method with state-of-the-art methods on public datasets. Table 1 shows the results of state-of-the-art methods and our experiments. For a fair comparison, we list the methods with only MJ and ST datasets for training. Our method has reached competitive accuracy against the latest strong methods and outperforms them on several datasets. ABINet [5] and PARSeq [6] use powerful language models, while TRBA [4] uses an image rectification module as preprocessing. Against such models, our model achieves comparable accuracy despite its simple structure without using either, demonstrating the effectiveness of the proposed method.

4.4. Detailed Analysis

To confirm the effectiveness of the proposed method in detail, we conducted a detailed analysis. In this section, we report the average accuracy of all test sets.

Table 1: **Word accuracy on the six benchmark datasets.**

Method	Test datasets and # of samples							
	IIIT5k	SVT	IC13		IC15		SVTP	CUTE
	3,000	647	857	1,015	1,811	2,077	645	288
CRNN [3]	81.8	80.1	89.4	88.4	65.3	60.4	65.9	61.5
ViTSTR [30]	88.4	87.7	93.2	92.4	78.5	72.6	81.8	81.3
TRBA [4]	92.1	88.9	–	93.1	–	74.7	79.5	78.2
ABINet [5]	96.2	93.5	**97.4**	–	86.0	–	**89.3**	89.2
PARSeq [6]	97.0	**93.6**	97.0	96.2	**86.5**	**82.9**	88.9	92.2
DiffusionSTR (Ours)	**97.3**	**93.6**	97.1	**96.4**	86.0	82.2	89.2	**92.5**

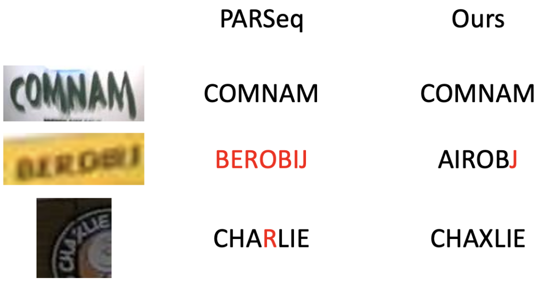


Fig. 3: **Visualized comparison against the state-of-the-art method [6].** The leftmost column shows the input image, and each column displays each method's output. Black text indicates a match with the ground truth; red indicates a different case. Our method recognizes more robustly.

Table 2: **Impact of character-aware head.**

Model	Character-aware head	Average accuracy
Complete model	✓	91.8
Ablation model	–	63.4

Qualitative analysis. Figure 3 shows the text recognition results of the proposed method and the latest one using the language model [6]. Our method outputs reasonable results, even for blurred or curved images. It produces more appropriate results for noisy images than the existing one.

Impact of character-aware head. In addition to the categorical classification of characters, DiffusionSTR proposes to predict the position of character presence. Table 2 shows the effectiveness of the proposed character-aware head. We see that the accuracy without predicting the presence is significantly degraded. The diffusion model needs to infer the location of character presence for text recognition.

Impact of time step. The diffusion step is a vital factor in the diffusion model. Table 3 shows how the total number of steps affects accuracy. We confirm that the accuracy increases as the number of diffusion steps increases; after 1000, only a little difference can be observed.

5. CONCLUSION

This work proposed a novel scene text recognition framework using diffusion models, which refines a noisy text sequence to the recognized one iteratively. Unlike existing methods that directly transform visual information into a text sequence, the proposed method leverages visual information to refine noisy text sequences conditionally. Experiment results showed that our method achieved competitive results compared to state-of-the-art methods with simple architecture.

Table 3: **Impact of the time step T.**

Total step (T)	100	500	1000	2000	4000
Average accuracy	17.6	86.1	91.8	91.9	91.3

6. REFERENCES

[1] Masato Fujitake and Hongpeng Ge, "Temporally-aware convolutional block attention module for video text detection," in *IEEE SMC*, 2021, pp. 220–225. 1

[2] Masato Fujitake, "A3s: Adversarial learning of semantic representations for scene-text spotting," in *ICASSP*, 2023, pp. 1–5. 1

[3] Baoguang Shi, Xiang Bai, and Cong Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *TPAMI*, vol. 39, no. 11, pp. 2298–2304, 2016. 1, 2, 4

[4] Jeonghun Baek, Yusuke Matsui, and Kiyoharu Aizawa, "What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels," in *CVPR*, 2021, pp. 3113–3122. 1, 4

[5] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in *CVPR*, 2021, pp. 7098–7107. 1, 2, 4

[6] Darwin Bautista and Rowel Atienza, "Scene text recognition with permuted autoregressive sequence models," in *ECCV*, 2022, pp. 178–196. 1, 2, 3, 4

[7] Jonathan Ho, Ajay Jain, and Pieter Abbeel, "Denoising diffusion probabilistic models," in *NeurIPS*, 2020, vol. 33, pp. 6840–6851. 1, 2

[8] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling, "Argmax flows and multinomial diffusion: Learning categorical distributions," in *NeurIPS*, 2021, vol. 34, pp. 12454–12465. 1, 2, 3

[9] Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin, "Gtc: Guided training of ctc towards efficient and accurate scene text recognition," in *AAAI*, 2020, vol. 34, pp. 11005–11012. 2

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

[11] Zhaoyi Wan, Minghang He, Haoran Chen, Xiang Bai, and Cong Yao, "Textscanner: Reading characters in order for robust scene text recognition," in *AAAI*, 2020, vol. 34, pp. 12120–12127. 2

[12] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Synthetic data and artificial neural networks for natural scene text recognition," in *NIPS Workshop*, 2014. 2, 3

[13] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai, "Aster: An attentional scene text recognizer with flexible rectification," *TPAMI*, vol. 41, no. 9, pp. 2035–2048, 2018. 2, 4

[14] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli, "Deep unsupervised learning using nonequilibrium thermodynamics," in *ICML*. PMLR, 2015, pp. 2256–2265. 2

[15] Aditya Ramesh, Prafulla Dhariwal, Alex Nichol, Casey Chu, and Mark Chen, "Hierarchical text-conditional image generation with clip latents," *arXiv preprint arXiv:2204.06125*, 2022. 2

[16] Shoufa Chen, Peize Sun, Yibing Song, and Ping Luo, "Diffusiondet: Diffusion model for object detection," *arXiv preprint arXiv:2211.09788*, 2022. 2

[17] Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro, "Diffwave: A versatile diffusion model for audio synthesis," in *ICLR*, 2021. 2

[18] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al., "An image is worth 16x16 words: Transformers for image recognition at scale," in *ICLR*, 2020. 3, 4

[19] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, L ukasz Kaiser, and Illia Polosukhin, "Attention is all you need," in *NIPS*, 2017, pp. 5998–6008. 2, 3

[20] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding, "Towards accurate scene text recognition with semantic reasoning networks," in *CVPR*, 2020, pp. 12113–12122. 3

[21] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman, "Reading text in the wild with convolutional neural networks," *IJCV*, vol. 116, pp. 1–20, 2016. 3

[22] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman, "Synthetic data for text localisation in natural images," in *CVPR*, 2016, pp. 2315–2324. 3

[23] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras, "Icdar 2013 robust reading competition," in *ICDAR*, 2013, pp. 1484–1493. 3

[24] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al., "Icdar 2015 competition on robust reading," in *ICDAR*, 2015, pp. 1156–1160. 3

[25] Anand Mishra, Karteek Alahari, and CV Jawahar, "Scene text recognition using higher order language priors," in *BMVC*, 2012. 3

[26] Kai Wang, Boris Babenko, and Serge Belongie, "End-to-end scene text recognition," in *ICCV*, 2011, pp. 1457–1464. 3

[27] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan, "Recognizing text with perspective distortion in natural scenes," in *ICCV*, 2013, pp. 569–576. 3

[28] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan, "A robust arbitrary text detection system for natural scene images," *Expert Systems with Applications*, vol. 41, no. 18, pp. 8027–8048, 2014. 3

[29] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," in *ICLR*, 2018, pp. 1–10. 4

[30] Rowel Atienza, "Vision transformer for fast and efficient scene text recognition," in *ICDAR*, 2021, pp. 319–334. 4