SVTR: Scene Text Recognition with a Single Visual Model

Yongkun Du[1] , **Zhineng Chen**[2*] , **Caiyan Jia**[1] , **Xiaoting Yin**[3] , **Tianlun Zheng**[2] , **Chenxia Li**[3] , **Yuning Du**[3] , **Yu-Gang Jiang**[2]

[1]School of Computer and Information Technology and Beijing Key Lab of Traffic Data Analysis and Mining, Beijing Jiaotong University, China

[2]Shanghai Collaborative Innovation Center of Intelligent Visual Computing, School of Computer Science, Fudan University, China

[3]Baidu Inc., China

{yongkundu,cyjia}@bjtu.edu.cn, {zhinchen,ygj}@fudan.edu.cn, tlzheng21@m.fudan.edu.cn, {yinxiaoting,lichenxia,duyuning}@baidu.com

Abstract

Dominant scene text recognition models commonly contain two building blocks, a visual model for feature extraction and a sequence model for text transcription. This hybrid architecture, although accurate, is complex and less efficient. In this study, we propose a Single Visual model for Scene Text recognition within the patch-wise image tokenization framework, which dispenses with the sequential modeling entirely. The method, termed SVTR, firstly decomposes an image text into small patches named character components. Afterward, hierarchical stages are recurrently carried out by component-level mixing, merging and/or combining. Global and local mixing blocks are devised to perceive the inter-character and intra-character patterns, leading to a multi-grained character component perception. Thus, characters are recognized by a simple linear prediction. Experimental results on both English and Chinese scene text recognition tasks demonstrate the effectiveness of SVTR. SVTR-L (Large) achieves highly competitive accuracy in English and outperforms existing methods by a large margin in Chinese, while running faster. In addition, SVTR-T (Tiny) is an effective and much smaller model, which shows appealing speed at inference. The code is publicly available at https://github.com/PaddlePaddle/PaddleOCR.

1 Introduction

Scene text recognition aims to transcript a text in natural image to digital character sequence, which conveys high-level semantics vital for scene understanding. The task is challenging due to variations in text deformations, fonts, occlusions, cluttered background, etc. In the past years, many efforts have been made to improve the recognition accuracy. Modern text recognizers, besides accuracy, also take factors like inference speed into account because of practical requirements.

*Corresponding Author

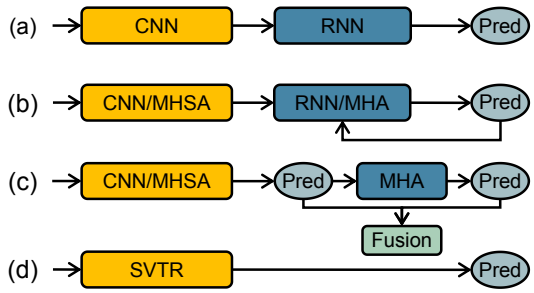


Figure 1: (a) CNN-RNN based models. (b) Encoder-Decoder models. MHSA and MHA denote multi-head self-attention and multi-head attention, respectively. (c) Vision-Language models. (d) Our SVTR, which recognizes scene text with a single visual model and enjoys efficient, accurate and cross-lingual versatile.

Methodologically, scene text recognition can be viewed as a cross-modal mapping from image to character sequence. Typically, the recognizer consists of two building blocks, a visual model for feature extraction and a sequence model for text transcription. For example, CNN-RNN based models [Zhai *et al.*, 2016; Shi *et al.*, 2017] first employed CNN for feature extraction. The feature was then reshaped as a sequence and modeled by BiLSTM and CTC loss to get the prediction (Figure 1(a)). They are featured by efficiency and remain the choice for some commercial recognizers. However, the reshaping is sensitive to text disturbances such as deformation, occlusion, etc, limiting their effectiveness.

Later, encoder-decoder based auto-regressive methods became popular [Sheng *et al.*, 2019; Li *et al.*, 2019; Zheng *et al.*, 2021], the methods transform the recognition as an iterative decoding procedure (Figure 1(b)). As a result, improved accuracy was obtained as the context information was considered. However, the inference speed is slow due to the character-by-character transcription. The pipeline was further extended to vision-language based framework [Yu *et al.*, 2020; Fang *et al.*, 2021], where language knowledge was incorporated (Figure 1(c)) and parallel prediction was conducted. However, the pipeline often requires a large capacity model or complex recognition paradigm to ensure the recog-

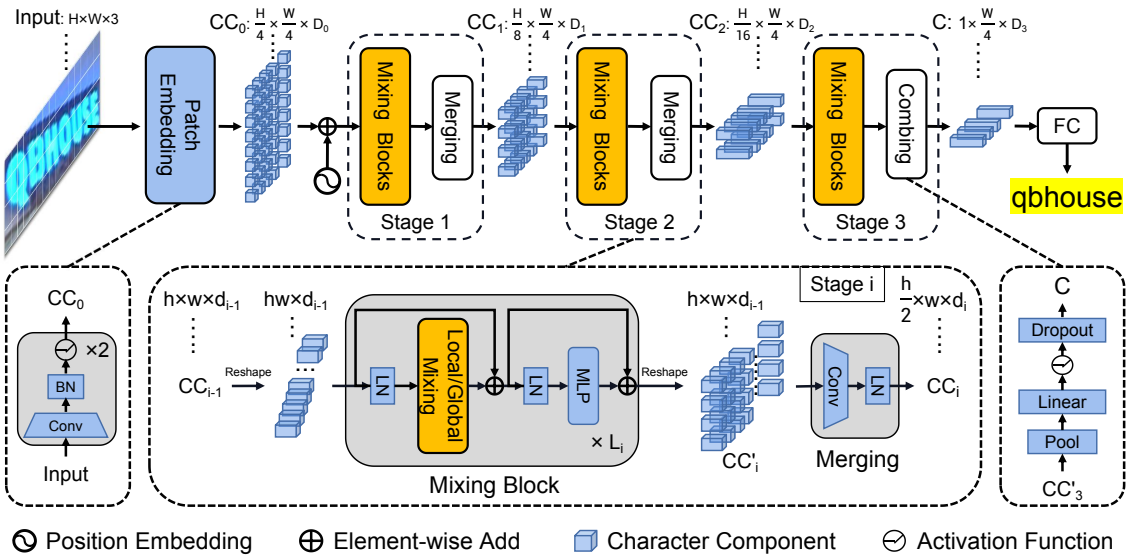


Figure 2: Overall architecture of the proposed SVTR. It is a three-stage height progressively decreased network. In each stage, a series of mixing blocks are carried out and followed by a merging or combining operation. At last, the recognition is conducted by a linear prediction.

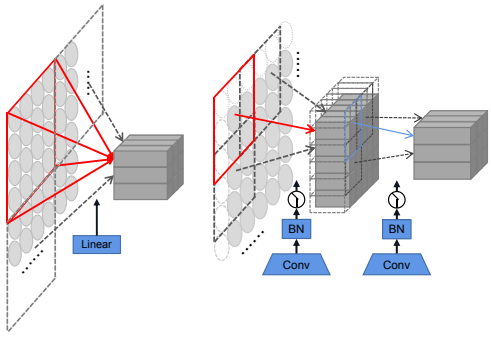
nition accuracy, restricting its efficiency.

Recently, there are efforts emphasized developing simplified architectures to accelerate the speed. For example, using complex training paradigm but simple model for inference. CRNN-RNN based solution was revisited in [Hu *et al.*, 2020]. It utilized the attention mechanism and graph neural network to aggregate sequential features corresponding to the same character. At inference, the attention modeling branch was discarded to balance accuracy and speed. PREN2D [Yan *et al.*, 2021] further simplified the recognition by aggregating and decoding the 1D sub-character features simultaneously. [Wang *et al.*, 2021] proposed VisionLAN. It introduced a character-wise occluded learning to endue the visual model with language capability. While at inference, the visual model was applied merely for speedup. In view of the simplicity of a single visual model based architecture, some recognizers were proposed by employing off-the-shelf CNN [Borisyuk *et al.*, 2018] or ViT [Atienza, 2021] as the feature extractor. Despite being efficiency, their accuracy is less competitive compared to state-of-the-art methods.

We argue that the single visual model based scheme is effective only if discriminative character features could be extracted. Specifically, the model could successfully catch both intra-character local patterns and inter-character long-term dependence. The former encodes stroke-like features that describe fine-grained features of a character, being a critical source for distinguishing characters. While the latter records language-analogous knowledge that describes the characters from a complementary aspect. However, the two properties are not well modeled by previous feature extractors. For example, CNN backbones are good at modeling local correlation rather than global dependence. Meanwhile, current transformer-based general-purpose backbones would not give privilege to local character patterns.

Motivated by the issues mentioned above, this work aims to enhance the recognition capability by reinforcing the visual model. To this end, we propose a visual-based model SVTR for accurate, fast and cross-lingual versatile scene text recognition. Inspired by the recent success of vision transformer [Dosovitskiy *et al.*, 2021; Liu *et al.*, 2021], SVTR first decomposes an image text into small 2D patches termed character components, as each of which may contain a part of a character only. Thus, patch-wise image tokenization followed by self-attention is applied to capture recognition clues among character components. Specifically, a text-customized architecture is developed for this purpose. It is a three-stage height progressively decreased backbone with mixing, merging and/or combining operations. Local and global mixing blocks are devised and recurrently employed at each stage, together with the merging or combining operation, acquiring the local component-level affinities that represent the stroke-like feature of a character, and long-term dependence among different characters. Therefore, the backbone extracts component features of different distance and at multiple scales, forming a multi-grained character feature perception. As a result, the recognition is reached by a simple linear prediction. In the whole process only one visual model is employed. We construct four architecture variants with varying capacities. Extensive experiments are carried out on both English and Chinese scene text recognition tasks. It is shown that SVTR-L (large) achieves highly competitive results in English and outperforms state-of-the-art models by a large margin in Chinese, while running faster. Meanwhile, SVTR-T (Tiny) is also an effective and much smaller model yet with appealing inference speed. The main contributions of this work can be summarized as follows.

- We demonstrate, for the first time, that a single visual model can achieve competitive or even higher accuracy as advanced vision-language models in scene text recognition. It is promising to practical applications due to its



(a) (b)

Figure 3: (a) The linear projection in ViT [Dosovitskiy *et al.*, 2021]. (b) Our progressive overlapping patch embedding.

efficiency and cross-lingual versatility.

- We propose SVTR, a text-customized model for recognition. It introduces local and global mixing blocks for extracting stroke-like features and inter-character dependence, respectively, together with the multi-scale backbone, forming a multi-grained feature description.
- Empirical studies on public benchmarks demonstrate the superiority of SVTR. SVTR-L achieves the state-of-the-art performance in recognizing both English and Chinese scene texts. While SVTR-T is effective yet efficient, with parameters of 6.03M and consuming 4.5ms per image text on average in one NVIDIA 1080Ti GPU.

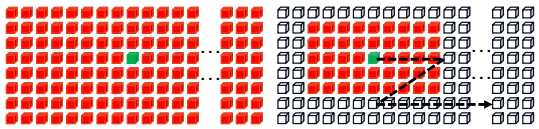
2 Method

2.1 Overall Architecture

Overview of the proposed SVTR is illustrated in Figure 2. It is a three-stage height progressively decreased network dedicated to text recognition. For an image text of size $H \times W \times 3$, it is first transformed to $\frac{H}{4} \times \frac{W}{4}$ patches of dimension D_0 via a progressive overlapping patch embedding. The patches are termed character components, each associating with a fraction of text character in the image. Then, three stages, each composed of a series of mixing blocks followed by a merging or combining operation, are carried out at different scales for feature extraction. local and global mixing blocks are devised for stroke-like local pattern extraction and inter-component dependence capturing. With the backbone, component feature and dependence of different distance and at multiple scales are characterized, generating a representation referred to as C of size $1 \times \frac{W}{4} \times D_3$, which perceives multi-grained character features. Finally, a parallel linear prediction with de-duplication is conducted to get the character sequence.

2.2 Progressive Overlapping Patch Embedding

With an image text, the first task is to obtain feature patches which represent character components from $X \in R^{H \times W \times 3}$ to $CC_0 \in R^{\frac{H}{4} \times \frac{W}{4} \times D_0}$. There exists two common one-step projections for this purpose, i.e., a 4×4 disjoint linear projection (see Figure 3(a)) and a 7×7 convolution with stride 4. Alternatively, we implement the patch embedding by using two consecutive 3×3 convolutions with stride 2 and batch normalization, as shown in Figure 3(b). The scheme, despite increasing the computational cost a little, adds the feature dimension progressively which is in favor of feature fusion. Ablation study in Section 3.3 shows its effectiveness.



(a) (b)

Figure 4: Illustration of (a) global mixing and (b) local mixing.

2.3 Mixing Block

Since two characters may differ slightly, text recognition heavily relies on features at character component level. However, existing studies mostly employ a feature sequence to represent the image text. Each feature corresponds to a thin-slice image region, which is often noisy especially for irregular text. It is not optimal for describing the character. Recent advancement of vision transformer introduces 2D feature representation, but how to leverage this representation in the context of text recognition is still worthy investigation.

More specifically, with the embedded components, we argue that text recognition requires two kinds of features. The first is local component patterns such as the stroke-like feature. It encodes the morphology feature and correlation between different parts of a character. The second is inter-character dependence such as the correlation between different characters or between text and non-text components. Therefore, we devise two mixing blocks to perceive the correlation by using self-attention with different reception fields.

Global Mixing. As seen in Figure 4(a), global mixing evaluates the dependence among all character components. Since text and non-text are two major elements in an image, such a general-purpose mixing could establish the long-term dependence among component from different characters. Besides, it also capable of weakening the influence of non-text components, while enhancing the importance of text components. Mathematically, for character components CC_{i-1} from the previous stage, it is first reshaped as a feature sequence. When feeding into the mixing block, a layer norm is applied and followed by a multi-head self-attention for dependence modeling. In the following, a layer norm and a MLP are sequentially applied for feature fusion. Together with the shortcut connections, forming the global mixing block.

Local Mixing. As seen in Figure 4(b), local mixing evaluates the correlation among components within a predefined window. Its objective is to encode the character morphology feature and establish the associations between components within a character, which simulates the stroke-like feature vital for character identification. Different from global mixing, local mixing considers a neighborhood for each component. Similar to convolution, the mixing is carried out in a sliding window manner. The window size is empirically set to

7×11. Compared with the global mixing, it implements the self-attention mechanism to capture local patterns.

As aforementioned, the two mixing blocks aims to extract different features that are complementary. In SVTR, the blocks are recurrently applied many times in each stage for comprehensive feature extraction. Permutation of the two kinds of blocks will be ablated later.

2.4 Merging

It is computational expensive to maintain a constant spatial resolution across stages, which also leads to redundant representation. As a consequence, we devise a merging operation following the mixing blocks at each stage (except the last one). With the features outputted from the last mixing block, we first reshape it to an embedding of size $h \times w \times d_{i-1}$, denoting current height, width and channels, respectively. Then, we employ a 3×3 convolution with stride 2 in the height dimension and 1 in the width dimension, followed by a layer norm, generating an embedding of size $\frac{h}{2} \times w \times d_i$.

The merging operation halve the height while keep a constant width. It not only reduce the computational cost, but also build a text-customized hierarchical structure. Typically, most image text appears horizontally or near horizontally. Compressing the height dimension can establish a multi-scale representation for each character, while not affecting the patch layout in the width dimension. Therefore, it would not increase the chance of encoding adjacent characters into the same component across stages. We also increase the channel dimension d_i to compensate the information loss.

2.5 Combining and Prediction

In the last stage, the merging operation is replaced by a combining operation. It pools the height dimension to 1 at first, followed by a fully-connected layer, non-linear activation and dropout. By doing this, character components are further compressed to a feature sequence, where each element is represented by a feature of length D_3. Compared to the merging operation, the combining operation can avoid applying convolution to an embedding whose size is very small in one dimension, e.g., with 2 in height.

With the combined feature, we implement the recognition by a simple parallel linear prediction. Concretely, a linear classifier with N nodes is employed. It generates a transcript sequence of size $\frac{W}{4}$, where ideally, components of the same character are transcribed to duplicate characters, components of non-text are transcribed to a blank symbol. The sequence is automatically condensed to the final result. In the implementation, N is set to 37 for English and 6625 for Chinese.

2.6 Architecture Variants

There are several hyper-parameters in SVTR, including the depth of channel and the number of heads at each stage, the number of mixing blocks and their permutation. By varying them, SVTR architectures with different capacities could be obtained and we construct four typical ones, i.e., SVTR-T (Tiny), SVTR-S (Small), SVTR-B (Base) and SVTR-L (Large). Their detail configurations are shown in Table 1.

3 Experiments

3.1 Datasets

For English recognition task, our models are trained on two commonly used synthetic scene text datasets, i.e., **MJSynth** (MJ) [Jaderberg *et al.*, 2014; Jaderberg *et al.*, 2015] and **SynthText** (ST) [Gupta *et al.*, 2016]. Then the models are tested on six public benchmarks as follows: **ICDAR 2013** (IC13) [KaratzasAU *et al.*, 2013] contains 1095 testing images. we discard images that contain non-alphanumeric characters or less than three characters. As a result, IC13 contains 857 images. **Street View Text** (SVT) [Wang *et al.*, 2011] has 647 testing images cropped form Google Street View. Many images are severely corrupted by noise, blur, and low resolution. **IIIT5K-Words** (IIIT) [Mishra *et al.*, 2012] is collected from the website and contains 3000 testing images. **ICDAR 2015** (IC15) [Karatzas *et al.*, 2015] is taken with Google Glasses without careful position and focusing. IC15 has two versions with 1,811 images and 2,077 images. We use the former one. **Street View Text-Perspective** (SVTP) [Phan *et al.*, 2013] is also cropped form Google Street View. There are 639 test images in this set and many of them are perspectively distorted. **CUTE80** (CUTE) is proposed in[Anhar *et al.*, 2014] for curved text recognition. 288 testing images are cropped from full images by using annotated words.

For Chinese recognition task, we use the **Chinese Scene Dataset** [Chen *et al.*, 2021]. It is a public dataset containing 509,164, 63,645 and 63,646 training, validation, and test images. The validation set is utilized to determine the best model, which is then assessed using the test set.

3.2 Implementation Details

SVTR uses the rectification module [Shi *et al.*, 2019], where the image text is resized to 32×64 for distortion correction. We use the AdamW optimizer with weight decay of 0.05 for training. For English models, the initial learning rate are set to $\frac{5}{10^4} \times \frac{batchsize}{2048}$. The cosine learning rate scheduler with 2 epochs linear warm-up is used in all 21 epochs. Data augmentation like rotation, perspective distortion, motion blur and Gaussian noise, are randomly performed during the training. The alphabet includes all case-insensitive alphanumerics. The maximum predict length is set to 25. The length exceeds the vast majority of English words. For Chinese models, the initial learning rate are set to $\frac{3}{10^4} \times \frac{batchsize}{512}$. The cosine learning rate scheduler with 5 epochs linear warm-up is used in all 100 epochs. Data augmentation was not used for training. The maximum predict length is set to 40. Word accuracy is used as the evaluation metric. All models are trained by using 4 Tesla V100 GPUs on PaddlePaddle.

3.3 Ablation Study

To better understand SVTR, we perform controlled experiments on both IC13 (regular text) and IC15 (irregular text) under different settings. For efficiency, all the experiments are carried out by using SVTR-T without rectification module and data augmentation.

The Effectiveness of Patch Embedding

As seen in Table 2 (the left half), different embedding strategies behave slightly different in recognition accuracy. Our

Models	$[D_0, D_1, D_2]$	$[L_1, L_2, L_3]$	Heads	D_3	Permutation	Params (M)	FLOPs (G)
SVTR-T	[64,128,256]	[3,6,3]	[2,4,8]	192	$[L]_6[G]_6$	4.15	0.29
SVTR-S	[96,192,256]	[3,6,6]	[3,6,8]	192	$[L]_8[G]_7$	8.45	0.63
SVTR-B	[128,256,384]	[3,6,9]	[4,8,12]	256	$[L]_8[G]_{10}$	22.66	3.55
SVTR-L	[192,256,512]	[3,9,9]	[6,8,16]	384	$[L]_{10}[G]_{11}$	38.81	6.07

Table 1: Architecture specifications of SVTR variants (w/o counting the rectification module and linear classifier).

Embedding	IC13	IC15	Merging	IC13	IC15	FLOPS
Linear	92.5	72.0	None	92.4	71.8	1.10
Overlap	93.0	73.9	Prog	**93.5**	**74.8**	0.29
Ours	**93.5**	**74.8**				

Table 2: Ablation study on patch embedding (the left half) and merging (the right half).

CP	IC13	IC15	CP	IC13	IC15
None	91.6	68.2	$[G]_{12}$	92.7	73.7
$[G]_6[L]_6$	92.2	71.9	$[L]_{12}$	91.3	70.2
$[L]_6[G]_6$	**93.5**	**74.8**	$[GL]_6$	91.9	72.4
			$[LG]_6$	**93.5**	73.5

Table 3: Ablation study on mixing block permutation.

progressive embedding scheme outperforms the two default ones by 0.75% and 2.8% on average on the two datasets, indicating that it is effective especially for irregular text.

The Effectiveness of Merging

There are also two choices, i.e., applying the merging operation to build a progressively decreased network ($prog$), and keeping a constant spatial resolution across stages ($none$). As indicated in the right half of Table 2. The merging not only reduces the computational cost, but also increases the accuracy on both datasets. It verifies that a multi-scale sampling on the height dimension is effective for text recognition.

The Permutation of Mixing Blocks

There are various ways to group the global and local mixing blocks in each stage. Several of them are assessed in Table 3, where $none$ means no mixing block is taken into account. $[G]_6[L]_6$ means for each stage, six global mixing blocks are carried out at first, and then six local mixing blocks. Others are defined similarly. As can be seen, almost every scheme gives certain accuracy improvement. We believe that the improvements are attributed to perceiving comprehensive character component features. The relatively large gains on irregular text further explains the mixing block is helpful to feature learning in complex scenarios. It is observed that $[L]_6[G]_6$ reports the best accuracy. It gives accuracy gains of 1.9% and 6.6% on IC13 and IC15 when compared with $none$. By placing the local mixing block in front of the global one, it is beneficial to guide the global mixing block to focus on long-term dependence capturing. On the contrary, switching their permutation is likely to confuse the role of the global mixing block, which may repetitively attend to local characteristics.

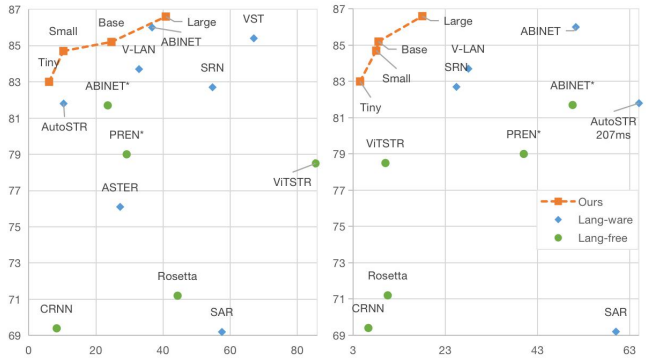


Figure 5: Accuracy-parameter (M) and Accuracy-speed (ms) plots of different models on IC15.

3.4 Comparison with State-of-the-Art

We compare SVTR with previous works on six English benchmarks covering regular and irregular text and one Chinese scene dataset in Table 4. The methods are grouped by whether utilizing the language information, i.e., lan-free and lan-aware. We first look into the result in English datasets. Even SVTR-T, the smallest one of the SVTR family, achieves highly competitive accuracy among the lan-free ones. While other SVTRs are new state-of-the-art on most datasets. When further compared with lan-aware ones, SVTR-L gains the best accuracy on IIIT, IC15 and CUTE among the six English benchmarks. Its overall accuracy is on par with recent studies [Fang *et al.*, 2021; Tang *et al.*, 2022], which used extra language models. As a contrast, SVTR enjoys its simplicity and runs faster. The results basically imply that a single visual model also could perform the recognition well, as discriminative character features are successfully extracted.

We then analyze the result on Chinese Scene Dataset. [Chen *et al.*, 2021] gives the accuracy of six existing methods as shown in the table. Encouragingly, SVTR performs considerably well. Compared with SAR, the best one among the listed ones, accuracy gains ranging from 5.4% to 9.6% are observed, which are noticeable improvements. The result is explained as SVTR comprehensively perceives multi-grained character component features. They are well suited to characterize Chinese words that have rich stroke patterns.

In Figure.5, we also depict the accuracy, parameter and inference speed of different models on IC15. Owning to their simpleness, SVTRs consistently rank top-tier in both accuracy-parameter and accuracy-speed plots, further demonstrating its superiority compared to existing methods.

	method	English regular			English unregular			Chinese	Params	Speed
		IC13	SVT	IIIT5k	IC15	SVTP	CUTE	Scene	(M)	(ms)
Lan-free	CRNN[Shi *et al.*, 2017]	91.1	81.6	82.9	69.4	70.0	65.5	53.4	8.3	6.3
	Rosetta[Borisyuk *et al.*, 2018]	90.9	84.7	84.3	71.2	73.8	69.2	-	44.3	10.5
	SRN*[Yu *et al.*, 2020]	93.2	88.1	92.3	77.5	79.4	84.7	-	-	-
	PREN*[Yan *et al.*, 2021]	94.7	92.0	92.1	79.2	83.9	81.3	-	29.1	40.0
	ViTSTR[Atienza, 2021]	93.2	87.7	88.4	78.5	81.8	81.3	-	85.5	11.2
	ABINet*[Fang *et al.*, 2021]	94.9	90.4	94.6	81.7	84.2	86.5	-	23.5	50.6
	VST*[Tang *et al.*, 2022]	95.6	91.9	95.6	82.3	87.0	91.8	-	-	-
Lan-aware	ASTER[Shi *et al.*, 2019]	-	89.5	93.4	76.1	78.5	79.5	54.5	27.2	-
	MORAN[Luo *et al.*, 2019]	-	88.3	91.2	-	76.1	77.4	51.8	28.5	-
	NRTR[Sheng *et al.*, 2019]	94.7	88.3	86.5	-	-	-	-	31.7	160
	SAR[Li *et al.*, 2019]	91.0	84.5	91.5	69.2	76.4	83.5	62.5	57.5	120
	AutoSTR[Zhang *et al.*, 2020]	-	90.9	94.7	81.8	81.7	84.0	-	10.4	207
	SRN[Yu *et al.*, 2020]	95.5	91.5	94.8	82.7	85.1	87.8	60.1	54.7	25.4
	PREN2D[Yan *et al.*, 2021]	96.4	94.0	95.6	83.0	87.6	91.7	-	-	-
	VisionLAN[Wang *et al.*, 2021]	95.7	91.7	95.8	83.7	86.0	88.5	-	32.8	28.0
	ABINet[Fang *et al.*, 2021]	**97.4**	93.5	96.2	86	89.3	89.2	-	36.7	51.3
	VST[Tang *et al.*, 2022]	96.4	**93.8**	**96.3**	85.4	88.7	**95.1**	**-**	64.0	-
Ours	SVTR-T (Tiny)	96.3	91.6	94.4	84.1	85.4	88.2	67.9	6.03	4.5
	SVTR-S (Small)	95.7	93.0	95.0	84.7	87.9	92.0	69.0	10.3	8.0
	SVTR-B (Base)	97.1	91.5	96.0	85.2	**89.9**	91.7	71.4	24.6	8.5
	SVTR-L (Large)	97.2	91.7	**96.3**	**86.6**	88.4	**95.1**	**72.1**	40.8	18.0

Table 4: Results on six English and one Chinese benchmarks tested against existing methods, where $CRNN$ and $Rosetta$ are from the reproduction of CombBest [Baek *et al.*, 2019]. Lan means language and * means the language-free version of the corresponding method. The speed is the inference time on one NVIDIA 1080Ti GPU averaged over 3000 English image text.

3.5 Visualization Analysis

We have visualized the attention maps of SVTR-T when decoding different character components. Each map can be explained as serving a different role in the whole recognition. Nine maps are selected for illustration, as shown in Figure.6. The first line shows three maps of gazing into a fraction of character "B", with emphasis on its left side, bottom and middle parts, respectively. It indicates that different character regions are contributed to the recognition. The second line exhibits three maps of gazing into different characters, i.e., "B", "L", and "S". SVTR-T is also able to learn its feature by viewing a character as a whole. While the third line exhibits three maps simultaneously activate multiple characters. It implies that dependencies among different characters are successfully captured. The three lines together reveal that sub-character, character-level and cross-character clues are all captured by the recognizer, in accordance with the claim that SVTR perceives multi-grained character component features. It again explains the effectiveness of SVTR.

With the results and discussion, we conclude with that SVTR-L enjoys the merits of accurate, fast and versatile, being a highly competitive choice in accuracy-oriented applications. While SVTR-T is an effective and much smaller model yet quite fast. It is appealing in resource-limited scenarios.

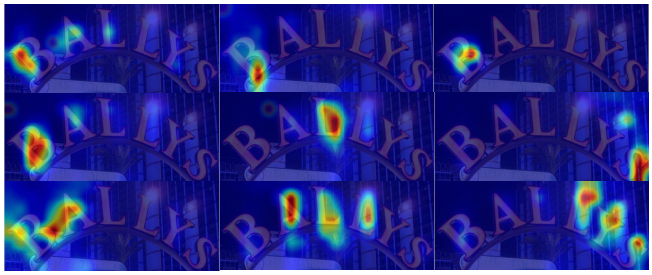


Figure 6: Visualization of SVTR-T attention maps.

4 Conclusion

We have presented SVTR, a customized visual model for scene text recognition. It extracts multi-grained character features that describe both stroke-like local patterns and inter-component dependence of varying distance at multiple height scales. Therefore, the recognition task can be conducted by using a single visual model, enjoying the merits of accuracy, efficiency and cross-lingual versatility. SVTR with different capacities are also devised to meet diverse application needs. Experiments on both English and Chinese benchmarks basically verify the proposed SVTR. Highly competitive or even better accuracy is observed compared to state-of-the-art methods, while running faster. We hope that SVTR will foster further research in scene text recognition.

Acknowledgments

The work is supported by the National Nature Foundation of China (No. 62172103, 61876016) and CCF-Baidu Open Fund (No.2021PP15002000).

References

[Anhar *et al.*, 2014] R. Anhar, S. Palaiahnakote, C. S. Chan, and C. L. Tan. A robust arbitrary text detection system for natural scene images. In *Expert Systems with Applications*, page 8027–8048, 2014.

[Atienza, 2021] R. Atienza. Vision transformer for fast and efficient scene text recognition. *arXiv:2105.08582*, 2021.

[Baek *et al.*, 2019] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *ICCV*, pages 4714–4722, 2019.

[Borisyuk *et al.*, 2018] F. Borisyuk, A. Gordo, and V. Sivakumar. Rosetta: Large scale system for text detection and recognition in images. In *ACM SIGKDD*, page 71–79, 2018.

[Chen *et al.*, 2021] J. Chen, H. Yu, J. Ma, M. Guan, X. Xu, X. Wang, S. Qu, B. Li, and X. Xue. Benchmarking chinese text recognition: Datasets, baselines, and an empirical study. *arXiv:2112.15093*, 2021.

[Dosovitskiy *et al.*, 2021] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, and N. Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[Fang *et al.*, 2021] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. In *CVPR*, 2021.

[Gupta *et al.*, 2016] A. Gupta, A. Vedaldi, and A. Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016.

[Hu *et al.*, 2020] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In *AAAI*, pages 11005–11012, 2020.

[Jaderberg *et al.*, 2014] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. In *NeurIPS Deep Learning Workshop*, 2014.

[Jaderberg *et al.*, 2015] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Reading text in the wild with convolutional neural networks. In *IJCV*, pages 1–20, 2015.

[Karatzas *et al.*, 2015] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny. Icdar 2015 competition on robust reading. In *ICDAR*, pages 1156–1160, 2015.

[KaratzasAU *et al.*, 2013] D. KaratzasAU, F. ShafaitAU, S. UchidaAU, M. IwamuraAU, L. G. i. BigordaAU, S. R. MestreAU, J. MasAU, D. F. MotaAU, J. A. AlmazànAU, and L. P. de las Heras. Icdar 2013 robust reading competition. In *ICDAR*, pages 1484–1493, 2013.

[Li *et al.*, 2019] H. Li, P. Wang, C. Shen, and G. Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *AAAI*, pages 8610–8617, 2019.

[Liu *et al.*, 2021] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *ICCV*, 2021.

[Luo *et al.*, 2019] C. Luo, L. Jin, and Z. Sun. A multi-object rectified attention network for scene text recognition. In *Pattern Recognit.*, pages 109–118, 2019.

[Mishra *et al.*, 2012] A. Mishra, A. Karteek, and C. V. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012.

[Phan *et al.*, 2013] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan. Recognizing text with perspective distortion in natural scenes. In *CVPR*, pages 569–576, 2013.

[Sheng *et al.*, 2019] F. Sheng, Z. Chen, and B. Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *ICDAR*, pages 781–786, 2019.

[Shi *et al.*, 2017] B. Shi, X. Bai, and C. Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. In *TPAMI*, pages 2298–2304, 2017.

[Shi *et al.*, 2019] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai. Aster: An attentional scene text recognizer with flexible rectification. In *TPAMI*, pages 2035–2048, 2019.

[Tang *et al.*, 2022] X. Tang, Y. Lai, Y. Liu, Y. Fu, and R. Fang. Visual-semantic transformer for scene text recognition. In *AAAI*, 2022.

[Wang *et al.*, 2011] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *ICCV*, pages 1457–1464, 2011.

[Wang *et al.*, 2021] Y. Wang, H. Xie, S. Fang, J. Wang, S. Zhu, and Y. Zhang. From two to one: A new scene text recognizer with visual language modeling network. In *ICCV*, pages 14194–14203, 2021.

[Yan *et al.*, 2021] R. Yan, L. Peng, S. Xiao, and G. Yao. Primitive representation learning for scene text recognition. In *CVPR*, pages 284–293, 2021.

[Yu *et al.*, 2020] D. Yu, X. Li, C. Zhang, T. Liu, J. Han, J. Liu, and E. Ding. Towards accurate scene text recognition with semantic reasoning networks. In *CVPR*, pages 12113–12122, 2020.

[Zhai *et al.*, 2016] C. Zhai, Z. Chen, J. Li, and B. Xu. Chinese image text recognition with blstm-ctc: A segmentation-free method. In *CCPR*, pages 525–536, 2016.

[Zhang *et al.*, 2020] H. Zhang, Q. Yao, M. Yang, Y. Xu, and X. Bai. Autostr: Efficient backbone search for scene text recognition. In *ECCV*, pages 751–767. Springer, 2020.

[Zheng *et al.*, 2021] T. Zheng, Z. Chen, S. Fang, H. Xie, and Y. Jiang. Cdistnet: Perceiving multi-domain character distance for robust text recognition. *arXiv:2111.11011*, 2021.