Multi-modal Text Recognition Networks: Interactive Enhancements between Visual and Semantic Features

Byeonghu Na[1], Yoonsik Kim[2], and Sungrae Park[3]*

[1] Dept. of Industrial & Systems Engineering, KAIST, `wp03052@kaist.ac.kr`
[2] Clova AI Research, NAVER Corp., `yoonsik.kim90@navercorp.com`
[3] Upstage AI Research, Upstage, `sungrae.park@upstage.ai`

Abstract. Linguistic knowledge has brought great benefits to scene text recognition by providing semantics to refine character sequences. However, since linguistic knowledge has been applied individually on the output sequence, previous methods have not fully utilized the semantics to understand visual clues for text recognition. This paper introduces a novel method, called **M**ulti-mod**A**l **T**ext **R**ecognition **N**etwork (**MATRN**), that enables interactions between visual and semantic features for better recognition performances. Specifically, MATRN identifies visual and semantic feature pairs and encodes spatial information into semantic features. Based on the spatial encoding, visual and semantic features are enhanced by referring to related features in the other modality. Furthermore, MATRN stimulates combining semantic features into visual features by hiding visual clues related to the character in the training phase. Our experiments demonstrate that MATRN achieves state-of-the-art performances on seven benchmarks with large margins, while naive combinations of two modalities show less-effective improvements. Further ablative studies prove the effectiveness of our proposed components. Our implementation is available at `https://github.com/wp03052/MATRN`.

1   Introduction

Scene text recognition (STR), a major component of optical character recognition (OCR) technology, identifies a character sequence in a given text image patch (e.g. words in a traffic sign). Applications of deep neural networks have brought great improvements in the performance of STR models [2,26,27,29,33,34]. They typically consist of a visual feature extractor, abstracting the image patch, and a character sequence generator, responsible for character decoding. Despite wide explorations to find better visual feature extractors and character sequence generators, existing methods still suffer from challenging environments: occlusion, blurs, distortions, and other artifacts [2,3].

To address these remaining challenges, STR methods have tried to utilize linguistic knowledge on the output character sequence. The mainstream of the approaches had been to model recursive operations learning linguistic knowledge for next character prediction. RNN [2,27] and Transformer [16,25,32] have

been applied to learn the auto-regressive language model (LM). However, the auto-regressive process requires expensive repeated computations and also learns limited linguistic knowledge from the uni-directional transmission.

To compensate for the issues, Yu *et al.* [33] propose SRN that refines an output sequence without auto-regressive operations. After identifying a seed character sequence, SRN re-estimates the character for each position at once by utilizing a Transformer encoder with a subsequent mask. Based on SRN, Fang *et al.* [7] improve the iterative refinement stages by explicitly dividing a vision model (VM) and an LM by blocking gradient flows and employing a bi-directional LM pre-trained on unlabeled text datasets. These methods incorporating semantic knowledge of LMs provide breakthroughs in recognizing challenging examples with ambiguous visual clues. However, the character refinements without visual features might lead to wrong answers by missing existent visual clues.

For better combinations of semantics and visual clues, Bhunia *et al.* [3] propose a multi-stage decoder referring to visual features multiple times to enhance semantic features. At each stage, a character sequence, designed as differentiable with Gumbel-softmax, is re-fined by re-assessing visual clues. Concurrently, Wang *et al.* [30] propose VisionLAN utilizing a language-aware visual mask that occludes selected character regions for enhancing the visual clues at the training phase. They prove that combining visual clues and semantic knowledge leads to better STR performances. Inspired by them, we raise a novel question: what is the best way to model the interactions between visual and semantic features identified by VM and LM, respectively?

To answer the question, this paper introduces a simple-but-effective extension of a STR model, named **M**ulti-mod**A**l **T**ext **R**ecognition **N**etwork (**MATRN**), that enhances visual and semantic features by referring to features in both modalities. MATRN consists of three proposed modules applied upon visual and semantic features: (1) multi-modal feature enhancement, incorporating bi-modalities to enhance each feature, (2) spatial encoding for semantics, linking two different modalities, (3) visual clue masking strategy, stimulating the cross-references between visual and semantic features. Figure 1 shows four types of visual and semantic feature fusions. MATRN is positioned in the bi-directional feature fusion (Figure 1d) by applying multi-modal feature enhancement. To the best of our knowledge, this natural but simple extension has never been explored.

The resulting model, MATRN, is architecturally simple but effective. In addition, the visual and semantic feature fusions are not expensive because the whole process is conducted in parallel. When we evaluate simple combinations of visual and semantic features without our proposed components, the performance improvements are observed as less-effective. However, interestingly, the proposed components contribute to STR performances effectively and lead MATRNs to achieve superior performances with notable gains from the current state-of-the-art. Consequently, our paper proves that semantics is helpful to capture better visual clues as well as that combining visual and semantic features reaches better STR performances.

Our contributions are threefold.
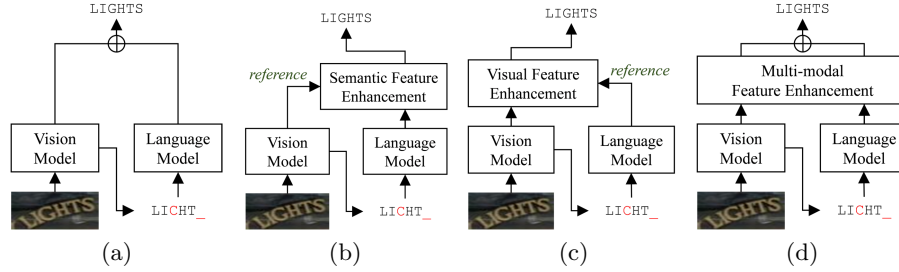
Fig. 1. Four types of visual and semantic feature fusions for STR: (a) simple combination of outputs from VM and LM, (b) visual-to-semantic feature fusion, (c) semantic-to-visual feature fusion, and (d) bi-directional feature fusion. SRN [33] is placed in (a) by applying LM to refine the output of VM. ABINet [7], PIMNet [22], and JVSR [3] can be aligned in (b) because their decoders refer to visual features iteratively during refining the final output sequence. VisionLAN [30] combines semantic information into visual features in a similar way to (c). Our method, MATRN, is positioned in (d) by enhancing both features through the bi-directional reference.

- We explore the combinations of visual and semantic features, identified by VM and LM, and prove their benefits. To the best of our knowledge, multi-modal feature enhancements with bi-directional fusions are novel components, that are natural extensions but have never been explored.
- We propose a STR method, named MATRN, that contains three major components, spatial encoding for semantics, multi-modal feature enhancements, and visual clue masking strategy, for better combinations of two modalities. Thanks to the effective contributions of the proposed components, MATRN achieves state-of-the-art performances on seven STR benchmarks.
- We provide empirical analyses that illustrate how our components improve STR performances as well as how MATRN contributes to the existing challenges.

2   Related Work

To utilize the benefits of a bi-directional Transformer, the non-autoregressive decoder has been introduced in the STR community. The general decoding process of them [1,7,22,30,33] lies in the effective construction of a sequence processed parallelly in the decoder. Specifically, positional embeddings describing the order of the output sequence are used to align visual (or semantic) features. Although the output sequence is generated in parallel, the bi-directional Transformer has shown comparable performances with the auto-regressive approaches.

ViTSTR [1] mainly focused on their VM without explicitly learning the LM. Inspired by the success of ViT [6], ViTSTR [1] has adopted ViT training scheme to STR. Specifically, its composition is very simple composed of the Transformer encoder and is trained with un-overlapped patches.

In order to incorporate linguistic knowledge, PIMNet [22], SRN [33] and ABINet [7] have been proposed. To learn linguistic knowledge from the auto-regressive model, PIMNet [22] proposed step-wise predictions and similarity distance between non-autoregressive and auto-regressive models. SRN and ABINet introduced a language modality that refines the output sequence of VM. Then, the final predictions are achieved by fusing the output sequences of LMs and VMs. In SRN [33], the LM is trained along with VM where the LM learns semantic information from other words. Based on SRN, ABINet [7] improves the iterative refinement stages by explicitly dividing the VM and LM. With a pre-training LM on unlabeled text datasets, it provides breakthroughs in recognizing challenging examples with ambiguous visual clues.

To interactively combine LM and VM, multi-modal recognizers are also introduced [3,30]. JVSR [3] proposes a multi-stage decoder referring to visual features multiple times to enhance semantic features. Specifically, it is based on an RNN-attention decoder with multi-stages where each stage generates an output sequence and visual features are employed for updating each hidden state. Since the decoder takes a hidden state as an input, the visual feature can iteratively enhance the semantic features. Concurrently, VisionLAN [30] proposes a language-aware visual mask that refers to semantic features for enhancing the visual features. Given a masking position of the word, the masking module occludes corresponding visual feature maps of the character region at the training phase. The previous multi-modal recognizers focus on one modality for final prediction and they utilize the other modality to improve their chosen one. In contrast, we explore the multiple combinations of multi-modal processes and propose MATRN which conducts both bi-directional enhancements.
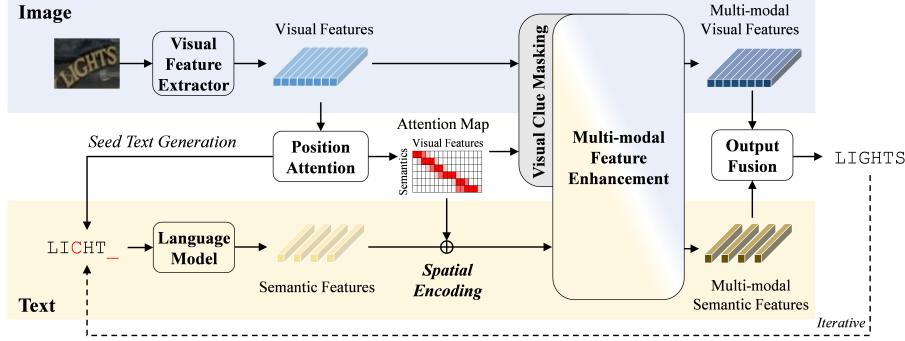
3   MATRN

Here, we describe our recognition model, *MATRN*, which incorporates both visual and semantic features. We will provide an overview of our method, and then describe each component in detail.

3.1   Overview of MATRNs

Figure 2 shows the overview of our model. It includes a visual feature extractor and a seed text generator to embed an image and provide an initial sequence of characters, as traditional STR models do. LM is applied to the seed text to extract semantic features.

Our contributions are focused on incorporating the visual and semantic features for better STR performances. Our method first encodes spatial positions into semantic features by utilizing an attention map identified during the seed text generation. The multi-modal feature enhancement module enriches individual visual and semantic features by incorporating multi-modalities. The enhanced features are named as *multi-modal visual features*, enhanced visual features with semantic knowledge, and *multi-modal semantic features*, enhanced

Fig. 2. An overview of MATRN. A visual feature extractor and an LM extract visual and semantic features, respectively. By utilizing the attention map, representing relations between visual features and character positions, MATRNs encode spatial information into the semantic features and hide visual features related to a randomly selected character. Through the multi-modal feature enhancement module, visual and semantic features interact with each other and the enhanced features in the two modalities are fused to finalize the output sequence.

semantic features with visual clues. Finally, both features are combined to provide the output sequence.

In the training phase of MATRN, the visual clue masking module hides visual features, related to a single character to stimulate the combination of semantics. Furthermore, the output sequence can be iteratively applied into the seed text for the LM, as follows [7].

3.2 Visual and Semantic Feature Extraction

To identify visual and semantic features, we construct three components: visual feature extractor, seed text generator, and language model. The following describes each module.

For the visual feature extractor, ResNet and Transformer units [7,33] are applied. The ResNet, $F^{\text{V.R}}$, with 45 layers embeds an input image, $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$, into convolution visual features, $\mathbf{V}^{\text{conv}} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times D}$. H, W are height and width of the image and D is the feature dimension. Before applying the Transformer, sinusoidal spatial position embeddings, $\mathbf{P}^{\text{V}} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times D}$, are added. Then, the Transformer, $F^{\text{V.T}}$, with three layers is applied:

$$\mathbf{V} = F^{\text{V.T}}(F^{\text{V.R}}(\mathbf{X}) + \mathbf{P}^{\text{V}}), \tag{1}$$

where $\mathbf{V} \in \mathbb{R}^{\frac{H}{4} \times \frac{W}{4} \times D}$ indicates visual features that are the outputs of the visual feature extractor.

For the seed text generation, an attention mechanism is utilized to transcribe visual features into character sequences. Specifically, an attention map, $\mathbf{A}^{\text{V-S}} \in \mathbb{R}^{T \times \frac{HW}{16}}$, is calculated by setting queries as text positional embeddings,

$\mathbf{P}^S \in \mathbb{R}^{T \times D}$, and keys as $\mathcal{G}(\mathbf{V}) \in \mathbb{R}^{\frac{HW}{16} \times D}$ in the attention mechanism, where T is the maximum length of sequence and $\mathcal{G}(\cdot)$ is a mini U-Net. Through the attention map, visual features are abstracted upon sequential features, $\mathbf{E}^{\mathbf{V}} = \mathbf{A}^{\text{V-S}} \widetilde{\mathbf{V}}$, where $\widetilde{\mathbf{V}} \in \mathbb{R}^{\frac{HW}{16} \times D}$ indicates the flattened visual features. By applying a linear layer and softmax function, a seed character sequence, $\mathbf{Y}_{(0)} \in \mathbb{R}^{T \times C}$, is generated, where C indicates the number of character classes. The whole process can be formalized as follows;

$$\mathbf{A}^{\text{V-S}} = \text{softmax} \left(\mathbf{P}^S \mathcal{G}(\mathbf{V})^\top / \sqrt{D} \right), \tag{2}$$

$$\mathbf{Y}_{(0)} = \text{softmax} \left(\mathbf{A}^{\text{V-S}} \widetilde{\mathbf{V}} \mathbf{W} \right), \tag{3}$$

where $\mathbf{W} \in \mathbb{R}^{D \times C}$ indicates a linear transition matrix.

The LM, introduced by [7], consists of four Transformer decoder blocks. It uses $\mathbf{P}^S$ as inputs and $\mathbf{Y}_{(0)}$ as the key/value of the attention layer. By processing the multiple decoder blocks, the LM identifies semantic features, $\mathbf{S} \in \mathbb{R}^{T \times D}$;

$$\mathbf{S} = F^{\text{LM}}(\mathbf{Y}_{(0)}), \tag{4}$$

where F^{LM} indicates the LM. We initialize the LM with the weights, pre-trained on WikiText-103, provided by [7].

3.3   Spatial Encoding to Semantic Features

One important point of combining the visual and semantic features is how to align each piece of information of different modalities. To guide the relationship between visual and semantic features, MATRN encodes spatial positions of visual features into semantic features. We call this process *spatial encoding to semantics (SES)*.

The key idea of SES is utilizing the attention map $\mathbf{A}^{\text{V-S}}$, used for the seed text generation, and the spatial position embedding $\mathbf{P}^V$, introduced in the visual feature extractor. Since $\mathbf{A}^{\text{V-S}}$ provides which visual features are used to estimate a character at each position, the spatial positions for semantic features, $\mathbf{P}^{\text{Align}} \in \mathbb{R}^{T \times D}$, are calculated as follows;

$$\mathbf{P}^{\text{Align}} = \mathbf{A}^{\text{V-S}} \widetilde{\mathbf{P}}^V, \tag{5}$$

where $\widetilde{\mathbf{P}}^V \in \mathbb{R}^{\frac{HW}{16} \times D}$ is the flattened sinusoidal spatial position embedding, $\mathbf{P}^V$. Then, we encode the spatial information into semantic features:

$$\mathbf{S}^{\text{Align}} = \mathbf{S} + \mathbf{P}^{\text{Align}}. \tag{6}$$

From this encoding process, the spatially aligned semantic features, $\mathbf{S}^{\text{Align}}$, contain spatial clues of visual features are highly related. It should be noted that SES does not need any additional parameters as well as it is simple and effective for the cross-references between visual and semantic features.

3.4   Multi-modal Features Enhancement

Now, we hold visual features, $\tilde{\mathbf{V}}$, that learn visual clues for character estimations, and semantic features, $\mathbf{S}^{\text{Align}}$, that contain linguistic knowledge for a character sequence. Previous methods [33,7] simply use a gated mechanism to seed character feature $\mathbf{E}^{\text{V}}$ and semantic feature $\mathbf{S}$. However, this simple fusion mechanism might not completely utilize these two features. Therefore, we propose a way in which the visual and semantic features refer to each other effectively and consequently enhance the features.

Multi-modal transformer [28], which consists of transformer layers processing multiple types of features at once, has been introduced in several domains such as visual question answering [10], vision-language navigation [4], autonomous driving [21], video retrieval [8], and many others. Inspired by them, we employ the multi-modal transformer for visual and semantic features enhancement for STR. The multi-modal transformers have multiple Transformer encoder blocks that consist of an attention layer and a feed-forward layer. At the attention layer, both visual and semantic features are processed through self-attentions. Since the queries determine their major modality, visual features are enhanced as multi-modal visual features, $\mathbf{V}^{\text{M}} \in \mathbb{R}^{\frac{HW}{16} \times D}$, and semantic features are updated into multi-modal semantic features, $\mathbf{S}^{\text{M}} \in \mathbb{R}^{T \times D}$.

3.5   Final Output Fusion

Both multi-modal features are utilized to finalize the output character sequence. While multi-modal semantic features are already aligned as a sequence, multi-modal visual features are required to be re-organized to estimate characters. To align the visual features into a sequence, we apply a character generator, which has the same architecture of the seed text generator, to aggregate $\mathbf{V}^{\text{M}}$ into sequential features, $\mathbf{E}^{\mathbf{V}^{\text{M}}}$ (See §3.2). Afterward, two sequential features, $\mathbf{E}^{\mathbf{V}^{\text{M}}}$ and $\mathbf{S}^{\text{M}}$, are combined through a gate mechanism to identify features, $\mathbf{F} \in \mathbb{R}^{T \times D}$, used for final character estimations:

$$\mathbf{G} = \sigma \left(\left[\mathbf{E}^{\mathbf{V}^{\text{M}}}; \mathbf{S}^{\text{M}} \right] \mathbf{W}^{\text{gated}} \right), \tag{7}$$

$$\mathbf{F} = \mathbf{G} \odot \mathbf{E}^{\mathbf{V}^{\text{M}}} + (1 - \mathbf{G}) \odot \mathbf{S}^{\text{M}}, \tag{8}$$

where $\mathbf{W}^{\text{gated}} \in \mathbb{R}^{2D \times D}$ is a weight, $[;]$ indicates concatenation, and $\odot$ is element-wise product. Finally, a linear layer and softmax function are applied on $\mathbf{F}$ to estimate a character sequence, $\mathbf{Y}_{(1)} \in \mathbb{R}^{T \times C}$.

3.6   Visual Clue Masking Strategy

To facilitate a better blend of the visual and semantic features, we propose a visual clue masking strategy, which is motivated by VisionLAN [30]. This strategy selects a single character randomly and hides corresponding visual features based on the attention map, $\mathbf{A}^{\text{V-S}}$, identified in the seed text generation. By explicitly deleting influential features for the character estimation, the multi-modal
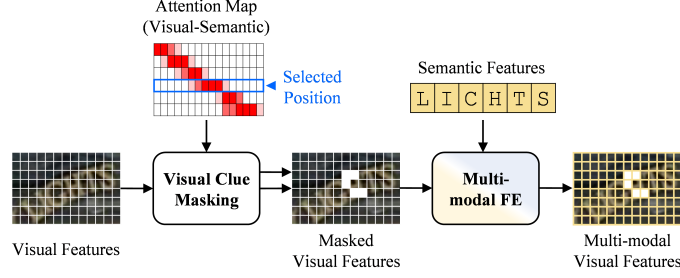
Fig. 3. Conceptual descriptions of visual clue masking strategy. Based on the attention map, representing relations between visual features and characters, influential features for a randomly selected character position are masked. In the multi-modal FE stage, semantic features are stimulated to be merged more strongly to compensate for the missing visual clues.

FE module becomes stimulated to encode semantic knowledge into the visual features in order to compensate for the missing visual clues. Figure 3 provides the conceptual description of the visual clue masking strategy.

The masking process chooses a position randomly in a character sequence and finds the top-K visual features relevant to the chosen position. For example, if the fourth position is selected, the process identifies the front K visual features in the descending order of the attention scores at the fourth position. The identified visual features are replaced into $\mathbf{v}_{[\text{MASK}]} \in \mathbb{R}^D$. The visual clue masking is only applied in the training phase. To reduce the discrepancy between training and evaluating phases, we keep the identified features unchanged with probability 0.1, as like [5].

3.7   Training Objective

MATRN is trained as end-to-end learning with multi-task cross-entropy objectives from multi-level visual and semantic features. We denote $\mathcal{L}_*$ is a cross-entropy loss for estimated character sequences from a feature $*$. For the estimations, a linear layer and a softmax function are utilized. In addition, MATRN applies iterative semantic feature correction to resolve the noisy input for the LM, as follows [7,15]. At the iteration, the input of LM is replaced into the output of the output fusion layer (See Figure 2). The loss of MATRN is formed as follows;

$$\mathcal{L} = \mathcal{L}_{\mathbf{E}^{\mathbf{V}}} + \frac{1}{M} \sum_{i=1}^{M} \left(\mathcal{L}_{\mathbf{S}_{(i)}} + \mathcal{L}_{\mathbf{S}_{(i)}^{\text{M}}} + \mathcal{L}_{\mathbf{E}_{(i)}^{\mathbf{V}^{\text{M}}}} + \mathcal{L}_{\mathbf{F}_{(i)}} \right), \tag{9}$$

where M is the number of iterations. Here, $\mathbf{S}_{(i)}, \mathbf{S}_{(i)}^{\text{M}}, \mathbf{E}_{(i)}^{\mathbf{V}^{\text{M}}}$, and $\mathbf{F}_{(i)}$ indicate the semantic, multi-modal semantic, multi-modal visual, and final fused features at the i-th iteration, respectively.

4   Experiments

4.1   Datasets

For fair comparisons, we use the same training dataset and evaluation protocol with [7,33]. For the training set, we use two widely-used synthetic datasets, MJSynth [11] and SynthText [9]. MJSynth has 9M synthetic text images and SynthText consists of 7M images including examples with special characters. Most previous works have used these two synthetic datasets together: MJSynth + SynthText [2].

For evaluation, eight widely used real-world STR benchmark datasets are used as test datasets. The datasets are categorized into two groups: "regular" and "irregular" datasets, according to the geometric layout of texts. "regular" datasets mainly contain horizontally aligned text images. IIIT5K (IIIT) [19] consists of 3,000 images collected from the web. Street View Text (SVT) [12] has 647 images collected from Google Street View. ICDAR2013 (IC13) [13] represents images cropped from mall pictures and has two variants; 857 images ($IC13_S$) and 1015 images ($IC13_L$). We utilize all two variants for providing fair comparisons. We skipped the evaluation on ICDAR2003 [17] because it contains duplicated images with IC13 [2].

"irregular" datasets consist of more examples of text in arbitrary shapes. IC-DAR2015 (IC15) consists of images taken from scenes and also has two versions; 1,811 images ($IC15_S$) and 2,077 images ($IC15_L$). Street View Text Perspective (SVTP) [20] contains 645 images of which texts are captured in perspective views. CUTE80 (CUTE) [24] consists of 288 images of which texts are heavily curved.

In our analyses, we measure the word prediction accuracy on each dataset. For "Total.", we evaluate the accuracy of unified evaluation datasets except for $IC13_L$ and $IC15_L$. It should be noted that we follow the philosophy of Baek *et al.* [2] which compares STR models upon the common evaluation datasets.

4.2   Implementation Details

The height and width of the input image are 32 and 128 by re-sizing text images and we apply image augmentation methods [7,18,33] such as rotation, color jittering, and noise injection. The number of character classes is 37; 10 for digits, 26 for alphabets, and a single padding token.

We borrow the network structures of the visual feature extractor, seed text generator, and language model from ABINet [7]. We set the feature dimension, D, as 512 and the maximum length of the sequence, T, as 25. For the multi-modal transformer, we use 2 Transformer blocks with 8 heads and a hidden size of 512. The iteration number M is set to 3 unless otherwise specified. We fixed the number of visual features mask, K, as 10.

We adopt the code from ABINet[4], and keep the experiment configuration. We use a pre-trained visual feature extractor and a pre-trained language model,

---

[4] https://github.com/FangShancheng/ABINet

Table 1. Recognition accuracies (%) on eight benchmark datasets, including the variant versions. The underlined values represent the best performances among the previous STR methods and the bold values indicate the best performances among all models including ours. For our implementation, we conduct repeated experiments with three different random seeds and report the averaged accuracy with standard deviation.

Model	Year	Regular test dataset				Irregular test dataset			
		IIIT	SVT	IC13$_\mathrm{S}$	IC13$_\mathrm{L}$	IC15$_\mathrm{S}$	IC15$_\mathrm{L}$	SVTP	CUTE
CombBest [2]	2019	87.9	87.5	93.6	92.3	77.6	71.8	79.2	74.0
ESIR [35]	2019	93.3	90.2	-	91.3	-	76.9	79.6	83.3
SE-ASTER [23]	2020	93.8	89.6	-	92.8	80.0		81.4	83.6
DAN [29]	2020	94.3	89.2	-	93.9	-	74.5	80.0	84.4
RobustScanner [34]	2020	95.3	88.1	-	94.8	-	77.1	79.5	90.3
AutoSTR [37]	2020	94.7	90.9	-	94.2	81.8	-	81.7	-
Yang *et al.* [32]	2020	94.7	88.9	-	93.2	79.5	77.1	80.9	85.4
SATRN [16]	2020	92.8	91.3	-	94.1	-	79.0	86.5	87.8
SRN [33]	2020	94.8	91.5	95.5	-	82.7	-	85.1	87.8
GA-SPIN [36]	2021	95.2	90.9	-	94.8	82.8	79.5	83.2	87.5
PREN2D [31]	2021	95.6	<u>94.0</u>	96.4	-	83.0	-	87.6	<u>91.7</u>
JVSR [3]	2021	95.2	92.2	-	<u>95.5</u>	-	**<u>84.0</u>**	85.7	89.7
VisionLAN [30]	2021	95.8	91.7	95.7	-	83.7	-	86.0	88.5
ABINet [7]	2021	<u>96.2</u>	93.5	<u>97.4</u>	-	<u>86.0</u>	-	<u>89.3</u>	89.2
ABINet (reproduced)		96.2	93.7	97.2	95.4	85.9	82.1	89.3	89.0
		±0.2	±0.4	±0.2	±0.2	±0.2	±0.1	±0.4	±0.3
MATRN (ours)		**96.6**	**95.0**	**97.9**	**95.8**	**86.6**	82.8	**90.6**	**93.5**
		±0.1	±0.2	±0.1	±0.1	±0.1	±0.0	±0.2	±0.6

which are provided by [7]. We use 4 NVIDIA GeForce RTX 3090 GPUs to train our models with batch size of 384. We used Adam [14] optimizer of initial learning rate 10^{-4}, and the learning rate is decayed to 10^{-5} after six epochs.

4.3   Comparison to State-of-the-Arts

Table 1 shows the existing STR methods and their performances on the eight STR benchmark datasets, including the variant versions of IC13 and IC15. In this comparison, we only consider the existing methods that are trained on MJSynth and SynthText.

When comparing the existing STR methods, PREN2D, JVSR, and ABINet showed state-of-the-art performances (See underlined values in the table). When compared to them, MATRN achieves the state-of-the-art performances on all evaluation datasets except IC15$_\mathrm{L}$. Specifically, our model achieved superior performance improvements on SVTP and CUTE, 1.3 percent point (pp) and 1.8pp respectively, because these datasets contain low-quality images, curved images, or proper nouns. Therefore, we found that our multi-modal fusion modules resolve the difficulties of scene text images, which cannot solve alone. JVSR [3] still holds the best position on IC15$_\mathrm{L}$, but MATRN shows huge performance gains

Table 2. Comparison of ABINet and MATRN under the comparable resources. **Param.** indicates the number of model parameters (M) and **Time** represents the inference time (ms/image) with batch size of 1 under AMD 32 cores, RTX 3090 GPU, and SSD 2TB. The underline indicates the similar or more resource when comparing from those of MATRN and the bold represents the best performer.

Model	Param.	Time	IIIT	SVT	IC13$_S$	IC15$_S$	SVTP	CUTE	Total.
ABINet	36.7M	21.6ms	96.2	93.7	97.2	85.9	89.3	89.0	92.6
ABINet w/ VM-*Big*	<u>46.2M</u>	22.6ms	96.4	93.8	97.9	86.3	89.5	88.5	92.9
ABINet w/ LM-*Big*	<u>46.2M</u>	26.6ms	96.0	94.3	97.5	86.3	89.9	88.9	92.8
ABINet w/ VM-*Bigger*	<u>90.3M</u>	<u>29.7ms</u>	96.3	94.9	**98.0**	86.5	89.9	89.9	93.1
ABINet w/ LM-*Bigger*	<u>52.5M</u>	<u>30.0ms</u>	96.1	94.3	97.7	86.0	90.1	89.2	92.8
MATRN (ours)	<u>44.2M</u>	<u>29.6ms</u>	**96.6** ±0.1	**95.0** ±0.2	97.9 ±0.1	**86.6** ±0.1	**90.6** ±0.2	**93.5** ±0.6	**93.5** ±0.1

on the other datasets: 1.4pp on IIIT, 2.8pp on SVT, 0.3pp on IC13$_L$, 4.9pp on SVTP, and 3.8pp on CUTE.

For apples-to-apples comparisons, we reproduced ABINet, which is one of the state-of-the-art methods and also our baseline before adding multi-modal fusion modules. In the sanity check, we observed that all reproduced performances are aligned upon confidence intervals from the reported scores. When comparing MATRN from the reproduced ABINet, the performance improvements are statistically significant over all datasets and the gaps are 0.4pp, 1.3pp, 0.7pp, 0.7pp, 1.3pp, and 4.5pp on IIIT, SVT, IC13$_S$, IC15$_S$, SVTP, and CUTE, respectively.

Many previous works, such as SE-ASTER, SRN, ABINet, JVSR, and VisionLAN, also analyzed how semantic information can be utilized for text recognition. When compared to them, MATRN shows the best performances on all but one datasets. This result implies that our incorporation method for visual and semantic features is effective compared to the existing methods that utilized semantic information.

4.4   Performance Comparison under the Comparable Resources

Since MATRN employs additional layers and modules upon ABINet, the performance gains might be considered as the effect of the additional memories and computational costs. To prove the pure benefits of the proposed methods, we evaluate large ABINets that utilize additional memories and computational costs as much as that MATRN requires. Specifically, the scale-up is conducted in two parts; adding transformer layers into VM (or LM) until the models have a similar number of parameters (*Big* models) and a similar inference speed (*Bigger* models). Table 2 shows the evaluation results. The *Big* models have similar parameters with MATRN but their speeds are faster since there is no cross-references between visual and semantic features. By scaling up the models, the *Bigger* models have similar inference speeds with MATRN but hold more parameters. When comparing the performances, the *Big* models provide relatively

Table 3. Performance improvements through gradual applications of our proposed modules. At the last line, all modules are applied and the model becomes MATRN.

	Multi-modal FE	SES Encoding	Visual Clue Masking	IIIT	SVT	IC13$_S$	IC15$_S$	SVTP	CUTE	Total.
ABINet→				96.2	93.7	97.2	85.9	89.3	89.0	92.6
	✓			96.5	94.3	98.0	85.9	90.1	91.0	93.0
	✓	✓		96.4	94.7	**98.1**	**86.9**	90.4	89.9	93.3
MATRN→	✓	✓	✓	**96.6**	**95.0**	97.9	86.6	**90.6**	**93.5**	**93.5**

small performance improvements: 0.3pp of VM-*Big* and 0.2pp of LM-*Big* in Total. The *Bigger* models show better performance improvements than the *Big* models; 0.5pp of VM-*Bigger* and 0.2pp of LM-*Bigger* in Total. However, the performance gains from the scaling-up are restricted when comparing the performance improvements of MATRN; 0.9pp in Total. In addition, the performances of MATRN are statistically significant when comparing all large ABINets. The experiments prove that the benefits of MATRN have not solely lie in the increasing computation resources.

4.5   Ablation Studies on Proposed Modules

Here, we analyze how the proposed modules contribute to the final performances. Table 3 shows the ablation studies that start from ABINet and add the proposed modules one by one. As can be seen, the total performances increase when adding proposed modules gradually. The application of the multi-modal transformer provides 0.4pp performance improvements based on ABINet. By applying SES on the multi-modal transformer, the total performance increases by 0.3pp. When adding the visual clue masking, the total performance finally becomes 93.5% with the 0.2pp improvement. Consequently, the simple application of a multi-modal transformer brings 0.4pp and our novel modules provide 0.5pp of performance improvements. We should note that applying a multi-modal transformer requires additional computations and parameters but the other proposed modules use quite small computations without any additional parameters. The ablation study indicates that our proposed modules for better multi-modal fusion lead to better STR performances effectively.

4.6   Discussion

Uni-modal vs. Multi-modal Feature Enhancement. The existing methods [3,30] focus on uni-modal FE by utilizing the other modality. To analyze the benefit of multi-modal FE, we compare the uni-modal FE that updates only a single modality utilizing the multi-modal transformer. In this experiment, we use SES for better fusion through the multi-modal transformer but do not apply the visual clue masking strategy for a fair comparison. Table 4 provides the

Table 4. Comparisons between uni-modal and multi-model FEs.

| Visual FE	Semantic FE	IIIT	SVT	IC13$_S$	IC15$_S$	SVTP	CUTE	Total.
		96.5	93.2	97.0	85.9	89.0	89.2	92.7
	✓	**96.5**	93.8	97.2	85.8	89.0	90.3	92.8
✓		96.1	93.5	97.5	86.1	89.8	**91.3**	92.8
✓	✓	96.4	**94.7**	**98.1**	**86.9**	**90.4**	89.9	**93.3**

Table 5. STR Performances with each level of features from VM, LM, and their fusions. Each value indicates total STR accuracy (%). **V** and **S** represent the output features from VM and LM, respectively. $\mathbf{V}^M$ and $\mathbf{S}^M$ indicate the enhanced features through the cross-reference. **F** represents the combined features for the final output.

Model	**V**	**S**	$\mathbf{V}^M$	$\mathbf{S}^M$	**F**
ABINet	90.9	49.5	-	-	92.6
MATRN	91.2	52.6	93.4	93.4	93.5

comparison results. In the table, the first model is identical to ABINet with SES and the uni-modal FE models (the second and third models) update the target features through the multi-modal transformer. As can be seen, the uni-modal FEs provide marginal performance improvements; 0.1pp in Total. When enhancing both modalities (the last model), the STR model enjoys two benefits of the semantic and visual FE and shows large performance improvements in Total. Given these points, we found that combining visual and semantic features improves the recognition performance, but one-way information flows are not enough to fuse two modalities. Besides, the multi-modal FE enables the two features to communicate in both directions and provides better performance.

STR Performance at Each Level of Features MATRN utilizes multi-task cross-entropy objectives, described in §3.7. Here, we evaluate the STR performances from the multiple features; **V**, **S**, $\mathbf{V}^M$, $\mathbf{S}^M$, and **F**. Table 5 shows the results of ABINet and MATRN. Interestingly, the results of **S** show insufficient performances in both models by refining character sequences without consideration of visual clues. However, the semantic features are combined and lead to better performances; **F** (ABINet), $\mathbf{V}^M$, and $\mathbf{S}^M$ (MATRN). In addition, the multi-modal features in MATRN show better performances than the final performances of ABINet and their combination shows the best.

Analysis on Cross-references Figure 4 shows attention map examples identified by the multi-modal FE of MATRN. At each attention map, the top-left and the bottom-right show the uni-modal attentions referring to their uni-modal features and the others provide the cross attentions between two different modal-
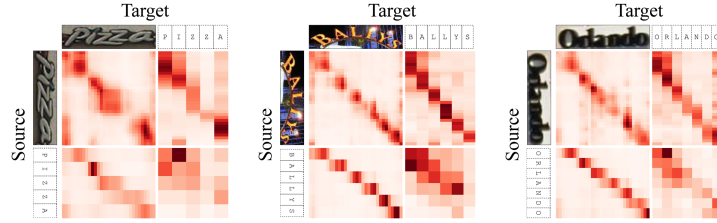
Fig. 4. Examples of self-attention maps in multi-modal FE. Attention maps on the top-right and the bottom-left indicate the cross attentions between two modalities.

Fig. 5. Examples that ABINet fails (*first line*) but MATRN succeeds (*second line*).

ities. As can be seen in the examples, visual and semantic features refer to their own modality as well as interact with each other.

Analysis on Previous Failure Cases Figure 5 shows the test examples that ABINet fails but MATRN successes. As can be seen, MATRN provides robust results on "cropped characters", "heavily curved text", "ambiguous visual clues" and "low resolutions". The results show that MATRN tackles the existing challenges.

5   Conclusion

This paper explores the combinations of visual and semantic features identified by VM and LM for better STR performances. Specifically, we propose MATRN that enhances visual and semantic features through cross-references between two modalities. MATRN consists of SES, that matches semantic features on 2D space that visual features are aligned in, multi-modal FE, that updates visual and semantic features together through the multi-modal transformer, and visual clue masking strategy, that stimulates the semantic references of visual features. In our experiments, naive applications of the multi-modal transformer lead to marginal improvements from the baseline. To this end, the components of MATRN effectively contribute to the multi-modal combinations and MATRN finally achieves state-of-the-art performances on seven STR benchmarks with large margins.

References

1. Atienza, R.: Vision transformer for fast and efficient scene text recognition. In: Lladós, J., Lopresti, D., Uchida, S. (eds.) Document Analysis and Recognition – ICDAR 2021. pp. 319–334. Springer International Publishing, Cham (2021)
2. Baek, J., Kim, G., Lee, J., Park, S., Han, D., Yun, S., Oh, S.J., Lee, H.: What is wrong with scene text recognition model comparisons? dataset and model analysis. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV) (October 2019)
3. Bhunia, A.K., Sain, A., Kumar, A., Ghose, S., Chowdhury, P.N., Song, Y.Z.: Joint visual semantic reasoning: Multi-stage decoder for text recognition. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 14940–14949 (October 2021)
4. Chen, S., Guhur, P.L., Schmid, C., Laptev, I.: History aware multimodal transformer for vision-and-language navigation. In: NeurIPS (2021)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers). pp. 4171–4186. Association for Computational Linguistics, Minneapolis, Minnesota (Jun 2019). https://doi.org/10.18653/v1/N19-1423, `https://www.aclweb.org/anthology/N19-1423`
6. Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N.: An image is worth 16x16 words: Transformers for image recognition at scale. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=YicbFdNTTy`
7. 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 (CVPR). pp. 7098–7107 (June 2021)
8. Gabeur, V., Sun, C., Alahari, K., Schmid, C.: Multi-modal transformer for video retrieval. In: European Conference on Computer Vision. pp. 214–229. Springer (2020)
9. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localisation in natural images. In: IEEE Conference on Computer Vision and Pattern Recognition (2016)
10. Hu, R., Singh, A., Darrell, T., Rohrbach, M.: Iterative answer prediction with pointer-augmented multimodal transformers for textvqa. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 9992–10002 (2020)
11. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Synthetic data and artificial neural networks for natural scene text recognition. In: Workshop on Deep Learning, NIPS (2014)
12. Kai Wang, Babenko, B., Belongie, S.: End-to-end scene text recognition. In: 2011 International Conference on Computer Vision. pp. 1457–1464 (2011). https://doi.org/10.1109/ICCV.2011.6126402
13. Karatzas, D., Shafait, F., Uchida, S., Iwamura, M., i. Bigorda, L.G., Mestre, S.R., Mas, J., Mota, D.F., Almazàn, 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 (2013). https://doi.org/10.1109/ICDAR.2013.221

14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (2015), `http://arxiv.org/abs/1412.6980`

15. Lee, J., Mansimov, E., Cho, K.: Deterministic non-autoregressive neural sequence modeling by iterative refinement. In: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. pp. 1173–1182. Association for Computational Linguistics, Brussels, Belgium (Oct-Nov 2018). https://doi.org/10.18653/v1/D18-1149, `https://www.aclweb.org/anthology/D18-1149`

16. Lee, J., Park, S., Baek, J., Oh, S.J., Kim, S., Lee, H.: On recognizing texts of arbitrary shapes with 2d self-attention. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops (June 2020)

17. Lucas, S.M., Panaretos, A., Sosa, L., Tang, A., Wong, S., Young, R.: Icdar 2003 robust reading competitions. In: Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings. pp. 682–687 (2003). https://doi.org/10.1109/ICDAR.2003.1227749

18. Lyu, P., Liao, M., Yao, C., Wu, W., Bai, X.: Mask textspotter: An end-to-end trainable neural network for spotting text with arbitrary shapes. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 67–83 (2018)

19. Mishra, A., Alahari, K., Jawahar, C.: Scene text recognition using higher order language priors. In: Proceedings of the British Machine Vision Conference. pp. 127.1–127.11. BMVA Press (2012). https://doi.org/http://dx.doi.org/10.5244/C.26.127

20. Phan, T.Q., Shivakumara, P., Tian, S., Tan, C.L.: Recognizing text with perspective distortion in natural scenes. In: 2013 IEEE International Conference on Computer Vision. pp. 569–576 (2013). https://doi.org/10.1109/ICCV.2013.76

21. Prakash, A., Chitta, K., Geiger, A.: Multi-modal fusion transformer for end-to-end autonomous driving. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7077–7087 (2021)

22. Qiao, Z., Zhou, Y., Wei, J., Wang, W., Zhang, Y., Jiang, N., Wang, H., Wang, W.: Pimnet: a parallel, iterative and mimicking network for scene text recognition. In: Proceedings of the 29th ACM International Conference on Multimedia. pp. 2046–2055 (2021)

23. 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 (CVPR) (June 2020)

24. 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). https://doi.org/https://doi.org/10.1016/j.eswa.2014.07.008, `https://www.sciencedirect.com/science/article/pii/S0957417414004060`

25. Sheng, F., Chen, Z., Xu, B.: NRTR: A no-recurrence sequence-to-sequence model for scene text recognition. In: 2019 International Conference on Document Analysis and Recognition, ICDAR 2019, Sydney, Australia, September 20-25, 2019. pp. 781–786. IEEE (2019). https://doi.org/10.1109/ICDAR.2019.00130, `https://doi.org/10.1109/ICDAR.2019.00130`

26. 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 & Machine Intelligence **39**(11), 2298–2304 (nov 2017). https://doi.org/10.1109/TPAMI.2016.2646371

27. 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 (2019). https://doi.org/10.1109/TPAMI.2018.2848939

28. Tsai, Y.H.H., Bai, S., Liang, P.P., Kolter, J.Z., Morency, L.P., Salakhutdinov, R.: Multimodal transformer for unaligned multimodal language sequences. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). Association for Computational Linguistics, Florence, Italy (7 2019)

29. Wang, T., Zhu, Y., Jin, L., Luo, C., Chen, X., Wu, Y., Wang, Q., Cai, M.: Decoupled attention network for text recognition. Proceedings of the AAAI Conference on Artificial Intelligence **34**(07), 12216–12224 (Apr 2020). https://doi.org/10.1609/aaai.v34i07.6903, `https://ojs.aaai.org/index.php/AAAI/article/view/6903`

30. 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)

31. Yan, R., Peng, L., Xiao, S., Yao, G.: Primitive representation learning for scene text recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). pp. 284–293 (June 2021)

32. Yang, L., Wang, P., Li, H., Li, Z., Zhang, Y.: A holistic representation guided attention network for scene text recognition. Neurocomputing **414**, 67–75 (2020). https://doi.org/https://doi.org/10.1016/j.neucom.2020.07.010, `https://www.sciencedirect.com/science/article/pii/S0925231220311176`

33. 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 (CVPR) (June 2020)

34. Yue, X., Kuang, Z., Lin, C., Sun, H., Zhang, W.: Robustscanner: Dynamically enhancing positional clues for robust text recognition. In: Vedaldi, A., Bischof, H., Brox, T., Frahm, J.M. (eds.) Computer Vision – ECCV 2020. pp. 135–151. Springer International Publishing, Cham (2020)

35. Zhan, F., Lu, S.: Esir: End-to-end scene text recognition via iterative image rectification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)

36. Zhang, C., Xu, Y., Cheng, Z., Pu, S., Niu, Y., Wu, F., Zou, F.: Spin: Structure-preserving inner offset network for scene text recognition. Proceedings of the AAAI Conference on Artificial Intelligence **35**(4), 3305–3314 (May 2021), `https://ojs.aaai.org/index.php/AAAI/article/view/16442`

37. Zhang, H., Yao, Q., Yang, M., Xu, Y., Bai, X.: Autostr: Efficient backbone search for scene text recognition. In: Proceedings of the European Conference on Computer Vision (ECCV) (2020)