Visual-Semantic Transformer for Scene Text Recognition

Liang Diao[1†]
diaoliang91@gmail.com
Xin Tang[1†]
tangxint@gmail.com
Jun Wang[2]
deeplearning.pku@qq.com
Rui Fang[1]
rui_fang_science@163.com
Guotong Xie[2]
xieguotong_cmt@163.com
Weifu Chen[3*]
waifook.chan@gmail.com

[1] Visual Computing Group, Ping An Property & Casualty Insurance Company, Shenzhen, China

[2] Ping An Technology (Shenzhen) Co. Ltd.

[3] School of Information and Telecommunication Engineering, Guangzhou Maritime University, Guangzhou, China

Abstract

Semantic information plays an important role in scene text recognition (STR) as well as visual information. Although state-of-the-art models have achieved great improvement in STR, they usually rely on extra external language models to refine the semantic features through context information, and the separate utilization of semantic and visual information leads to biased results, which limits the performance of those models. In this paper, we propose a novel model called Visual-Semantic Transformer (VST) for text recognition. VST consists of several key modules, including a ConvNet, a visual module, two visual-semantic modules, a visual-semantic feature interaction module and a semantic module. VST is a conceptually much simpler model. Different from existing STR models, VST can efficiently extract semantic features without using external language models and it also allows visual features and semantic features to interact with each other parallel so that global information from two domains can be fully exploited and more powerful representations can be learned. The working mechanism of VST is highly similar to our cognitive system, where the visual information is first captured by our sensory organ, and is simultaneously transformed to semantic information by our brain. Extensive experiments on seven public benchmarks including regular/ irregular text recognition datasets verify the effectiveness of VST, it outperformed other 14 popular models on four out of seven benchmark datasets and yielded competitive performance on the other three datasets.

† : Equal contribution; *: Corresponding author.

1 Introduction

Scene text recognition (STR) is the task of recognizing text from images taken in complex and has many real-world applications, such as self-driving cars [48] and street image understanding [43]. Though STR is an active research field [2, 17, 22, 35, 37], it is an inherently difficult task. Clutter background, large perspective distortion, lighting condition, degraded image quality (due to motion/out-of-focus blur), all those factors impose serious challenges to successfully solving the task, which lead to severe miss-prediction. It should be noted that typically a complete approach to recognizing text from scene images usually involves text detection and text recognition. However, following a large portion of the previous work and many open-source datasets [15, 16], we assume that text detection is done and we only need to focus on text recognition in this work. That is, we assume that the input images are cropped with regular or irregular characters lying in. We still use STR to refer to the text recognition systems under this assumption.

The approaches to solving STR problem can be roughly divided into two categories: linguistic-based [8, 32] and linguistic-free [28, 34]. Linguistic-based methods refer to those which incorporate vocabulary (dictionary), lexicon (parts of words) or the context information among semantic sequence, while linguistic-free methods use only visual information without relying on explicit language modeling.

The visual based methods are difficult to perceive contextual information, which can lead to false recognition when encountering occlusion, blurring and other situations. To overcome this limitation, most of the state-of-the-art models consider how to incorporate linguistic features with visual features for better performance. [27] added a dictionary in training and inference stage to help the model select the most compatible outcomes for STR. [32] used a language model to build the semantic correlation between scene and text in order to re-rank the recognition results. [42] proposed to learn the linguistic rules in the visual space by randomly masking out some characters from the image and predicting them back. Similar to speech recognition, scene text recognition can be treated as a sequence-to-sequence (seq2seq) mapping problem [49]. [17] combined convolution and LSTM as an encoder, then used another LSTM as the decoder to predict text attentively, quite similar to the *show, attend and tell* work on image captioning [44]. ASTER [35] took a two-stage approach to first rectify curved text images and then performs recognition using a seq2seq model with attention. [20] proposed a stacked block architecture with intermediate supervision to train a deep BiLSTM encoder, while attention was used in decoding stage to exploit contextualized visual features. [1] proposed seq2seq contrastive learning of visual representations which can be applied to text recognition. DAN [41] decoupled the alignment module from the decoding stage into the early conventional encoder network.

Transformers have also been successfully applied to STR. STAR-Net [21] used a spatial transformer [12] to tackle challenges brought by image distortion. ABINet [8] enforced a bidirectional language-model (LM) to only learn linguistic rules by gradient-stopping in training. The decoding is in an iterative way allowing the predictions to be refined progressively. HRGAT [46] connected CNN feature maps to a transformer-based autoregressive decoder, where the cross-attention is guided by holistic representation obtained by average-pooling of 2D feature maps. SRN [47] incorporated a visual-to-semantic embedding block and cross-entropy loss to align with ground-truth text, which is close to our model, but their model uses argmax embedding while our model directly uses probability vectors that enable smooth gradient flows in training. Instead of using argmax, [4] used Gumbel-softmax [14] for extracting semantic information, which was then fed into the succeeding transformer-based visual-semantic reasoning module. The decoding involves complex multiple-stage attentional LSTM that couples with feature pyramid networks.

In this paper, we propose a novel model called *Visual-Semantic Transformer* (VST) which interacts the visual and semantic feature in an explicit way. Although it is also a transformer-based multiple-stage semantic processing model, the proposed model can efficiently solve the visual-semantic alignment problem and learn more powerful representations, and hence obviously improve text prediction accuracy. Major contributions of the proposed model can be summarized as:

- We introduce an interaction module that allows visual features and semantic features to globally interact with each other and promotes the learning ability of the model.

- We design a new visual-semantic fusion strategy for merging and fusing semantic features with visual features in a cascade of two visual-semantic alignment modules concatenated by an interaction module.

- Experiments conducted on seven popular benchmark datasets demonstrate that VST can achieve higher or competitive prediction accuracy in scene text recognition without the aid of explicit language models.

2 Visual-Semantic Transformer

2.1 Motivation

When a text image is passing through our visual system, the sensory organ (the eye) and parts of the central nervous systems (the retina, the optic nerve, the optic tract and the visual cortex) detect and interpret information from the optical spectrum to build a representation (aka visual features) of the image. However, the visual information alone is inadequate for accurate text recognition, in particular, for text images that are blurred or occluded. Modern STR models require linguistic features to further improve the prediction accuracy. Although those algorithms have achieved great improvement in text recognition, there are some major limitations of the existing frameworks, including (1) visual information and linguistic information have been processed independently, or (2) an extra language model or a pre-trained model is required to extract the linguistic features [30] .

To alleviate the limitations, the proposed model employs two core components: visual-semantic alignment modules and visual-semantic interaction modules. The visual-semantic module is able to learn the semantic features from visual features without explicitly using linguistic features or external language models, which makes the model much simpler and more efficient. On the other hand, the proposal of the visual-semantic interaction module is inspired by wav2vec [3] in audio community. The wav2vec model first extract primary audio features from waveform using 1D convolution and the features are processed by a succeeding transformer module, resulting in secondary contextualized audio features. The secondary audio features interact with the primary features by predicting them back at the next few time steps. Similarly, we extract semantic and visual features at different stages, making them interact with each other so that more meaningful representation can be learned from the interaction, which is more in line with our cognitive system [6] [18].

Since both the visual-semantic alignment modules and visual-semantic interaction modules are based on attention mechanism and transformers, we name the proposed model

Visual-Semantic Transformer (VST), implying that it is a transformer which explicitly models visual and semantic information. Fig. 1 demonstrates the architecture of the model. VST consists of several key modules, namely ConvNet ($\mathcal{C}$), Visual module ($\mathcal{V}$), Interaction module ($\mathcal{I}$), Semantic module ($\mathcal{S}$) and two Visual-Semantic Alignment ($\mathcal{A}_1$ and $\mathcal{A}_2$) modules. Module $\mathcal{C}$ can be any convnet, Module $\mathcal{V}$, $\mathcal{I}$, $\mathcal{S}$ are basic transformer blocks, while Visual-Semantic Alignment (VS-Align) is an attention-based alignment block. We will introduce each module in details in the following subsections.

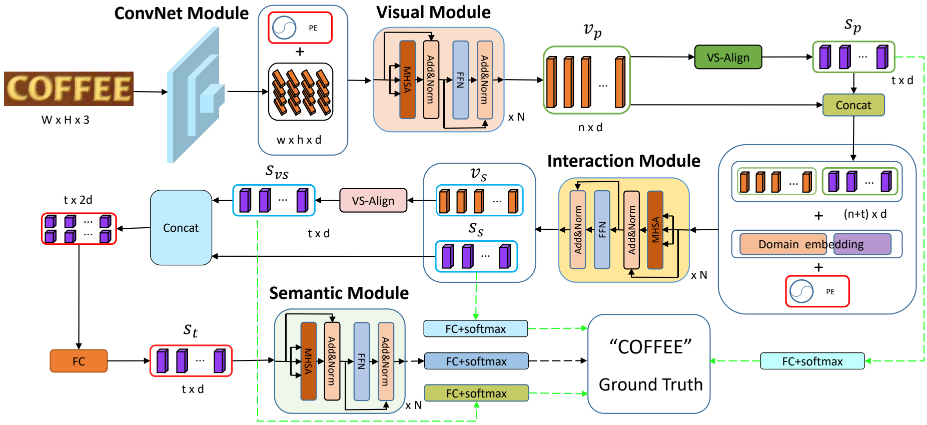

Figure 1: The architecture of Visual-Semantic Transformer (VST). The VST consists of several key modules, namely ConvNet ($\mathcal{C}$), Visual module ($\mathcal{V}$), Interaction module ($\mathcal{I}$), Semantic module ($\mathcal{S}$) and two Visual-Semantic Alignment ($\mathcal{A}_1$ in green, $\mathcal{A}_2$ in pink) modules. $v_p, s_p, v_s, s_s, s_{vs}, s_t$ represents primary visual, primary semantic, secondary visual, secondary semantic, visually semantic and tertiary semantic features respectively. Best view in color.

2.2 The ConvNet ($\mathcal{C}$) Module

Theoretically, ConvNet ($\mathcal{C}$) can be any kind of convolutional networks. In this work, we use a resnet-like architecture in extracting local features. Input images are first resized to have the same height, with aspect ratio kept unchanged. During training and testing, replication-padding (padding using values from the image border) is used for batch processing. Assume that input text images are of size $W \times H \times 3$ and the feature maps generated by the convnet module are of size $w \times h \times d$. The features will be fed into the visual module.

2.3 The Visual ($\mathcal{V}$) Module

Convnets are good at learning local features but hard to learn global correlation between features that locate far apart. Inspired from the success of transformers in vision tasks, we add a transformer-based visual module $\mathcal{V}$ after the ConvNet Module $\mathcal{C}$ in order to learn the global correlation and enhance the feature maps. The visual module consists of multi-head self-attentions (MHSA), layer-norm, and feed-forward network, as described in [36], but with minor modification that we put layer-norm before MHSA [7, 40]. Note that Fig. 1 does NOT display the modification. The feature maps obtained by Module ($\mathcal{C}$) are reshaped as a sequence of dimension d and length $n = w \times h$ and fed to the Visual Module $\mathcal{V}$. The Module

$\mathcal{V}$ generates a sequence of the same dimension and length as the size of the input sequence. We call the output *primary visual features* which encode the appearance information of the input image and will be further converted to *primary semantic features* by the VS-Align module that will be discussed in next subsection.

2.4 The Visual-Semantic Alignment ($\mathcal{A}$) Module

Although parallel attentions have been used to map visual features into semantics[25, 42], compared with previous work, the Visual-Semantic Alignment (VS-Align) module adopted in this work is much simpler but still effective. The architecture of the VS-Align module is depicted in Fig. 2. The visual features (feature maps) are first projected using a linear layer (the transform matrix is denoted as Q) and normalized using softmax operator, obtaining t attention maps (i.e., $\text{softmax}(QV^T)$), each of which has the same spatial dimension as the origin feature maps. Then the visual features are weightily summed by each heat map to obtain t semantic features. The alignment module can be mathematically formulated as,

$$S = \text{softmax}(QV^T)V \tag{1}$$

where $S \in R^{t\times d}$ is the semantic sequence, $V \in R^{w\times h\times d}$ is the visual feature maps and $Q \in R^{t\times d}$ is the trainable projection matrix.

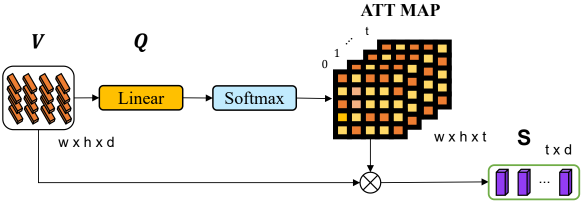

Figure 2: The architecture of VS-Align module. Visual features (viewed as feature maps) are projected using a linear layer and further normalized using sofmtax operator, obtaining t attention maps, each of which has the same spatial dimension as the origin feature map. The t attention maps and the original d visual feature maps together will reduce (by multiplication) to t semantic features in R^d.

As shown in Fig. 1, there are two VS-Align modules sharing the same structure but with different weights, we denote the first VS-Align module and the second VS-Align module as $\mathcal{A}_1$ and $\mathcal{A}_2$ (the green and pink VS-Align in Fig. 1). The reason that we use two VS-Align modules in this work is because it has been verified that it is difficult to achieve absolute alignment for multiple modalities especial for visual language models. So we suppose that the second VS-Align module can further reduce the gap between the primary and the secondary visual features. Indeed, we found that the attention maps are more precise to extract semantic features with the second VS-Align module, as shown in Fig. 3.

2.5 The Interaction ($\mathcal{I}$) Module

The interaction module plays a key role in fusing semantic features and visual features. The module takes two streams , namely *the primary semantic features* and *the primary visual* *features* as input, producing *the secondary visual features* and *the secondary semantic features*. The Interaction Module $\mathcal{I}$ is a transformer-based model, and following the notations in [36], the interaction module can be formulated as,

$$S = \text{softmax}(\frac{[Q_s;Q_v][K_s;K_v]^T}{\sqrt{d_k}})[V_s;V_v], \tag{2}$$

where the subscripts s and v denote the semantic and the visual features respectively, ';' is column representation. Since the semantic features can be seen as pseudo-linguistic features, which are distinguished from visual domain, we learn two embeddings and add them to the corresponding streams to distinguish the feature domain before feeding into the interaction module (See Fig. 1). In practice, since positional encodings can help the module to break the permutation invariant property of transformers, fixed positional encoding is added into the visual stream, and learnable positional embedding is added into the semantic stream.

Inside the module, two kinds of information can be interacted: (a) the visual information and the semantic information; (b) the features in the same stream. The interaction will help to learn meaningful semantic features since they now can access all spatial location of the visual features. Visual features are also benefited from the interaction because they can now not only internally interact with all other spatial locations regardless of distance, but also learn from semantic features, which is useful for dealing with appearance degradation.

One of the outputs of the interaction module is the secondary visual sequence in $R^{n\times d}$, which will be converted to *the visually semantic features* in $R^{t\times d}$ using another VS-Align module, and be compared against ground-truth text by cross-entropy loss after classification mapping layer and softmax. The other output is the secondary semantic features in $R^{t\times d}$, the loss is calculated in the same way. Note that the two classification mapping layers and loss branches are unneeded on the inference stage.

2.6 The Semantic ($\mathcal{S}$) Module

The Semantic Module ($\mathcal{S}$) is used to further fuse the two semantic streams, the secondary semantic features and the visually semantic features. However, Module $\mathcal{S}$ is optional. When $\mathcal{S}$ is inserted, the model is named VST-F (full); otherwise, it is named VST-B (basic). The architecture of the Semantic Module ($\mathcal{S}$) is the same as that of the Visual Module $\mathcal{V}$. We neglect to describe the details here. To balance the speed and the performance, we fuse the semantic features from two streams in a simpler but efficient manner with a concatenation operation followed by a linear layer, which transforms the dimension along channel direction from $t\times 2d$ into $t\times d$ before the transformer encoder. We denote the fusion feature as *tertiary semantic features* which is the input for Module $\mathcal{S}$. The output features of Module $\mathcal{S}$ are feed into a linear and softmax layer to decode the final text prediction.

2.7 Training Objective

Let $X = [x^{(1)}, x^{(2)}, \cdots, x^{(m)}]$ and $Y = [y^{(1)}, y^{(2)}, \cdots, y^{(m)}]$ denote the training set and the ground truth text label set, S_p, S_s, S_{vs}, S_t represent the primary semantic, the secondary semantic, the visually semantic and the tertiary semantic features respectively. Then the objective function of VST-F is composed of four losses. Each loss is defined as the cross-entropy loss calculated between the ground truth and the prediction results given by intermediate semantic features,

S_p, S_s, S_{vs} or S_t.

$$\begin{aligned} \min_{\theta_v,\theta_a,\theta_i,\theta_s} \quad & \alpha_1 * L(fc_{S_p}(S_p),Y) + \alpha_2 * L(fc_{S_s}(S_s),Y) + \alpha_3 * L(fc_{S_{vs}}(S_{vs}),Y) + \alpha_4 * L(fc_{S_t}(S_t),Y) \\ \text{subject to} \quad & V_p = f_{\theta_v}(X), S_p = f_{\theta_{a1}}(V_p), (S_s,V_s) = f_{\theta_i}(S_p,V_p), S_{vs} = f_{\theta_{a2}}(V_s), S_t = f_{\theta_s}(V_s) \end{aligned} \tag{3}$$

where $\theta_v, \theta_{a1}, \theta_i, \theta_{a2}, \theta_s$ denote the parameters of module $\mathcal{C}$+$\mathcal{V}$, $\mathcal{A}_1$, $\mathcal{I}$, $\mathcal{A}_2$ and $\mathcal{S}$ respectively, $fc()$ represents a fully connected layer concatenating with a softmax operator as a classifier, V_p and V_s represent the primary visual and the secondary visual features. We use $L(fc_{S_p}(S_p),Y)$ as an example to demonstrate how to compute the losses:

$$L(fc_{S_p}(S_p),Y) = -\sum_{i=1}^{m}\sum_{j=1}^{t} y_j^{(i)} \log P(\hat{y}_j^{(i)}), \tag{4}$$

where $\hat{y}^{(i)} = fc_{S_p}(S_p^{(i)})$ is the predicted result given by the primary semantic features of the ith training sample. In Fig. 1, three green dashed lines illustrate the first three losses in Eq. (3), and the black dash line illustrates the the fourth loss in Eq. (3), which is a main loss. Since VST-B doesn't have Module $\mathcal{S}$, its objective function only contains the first three losses, and $\alpha_4 = 0$.

3 Experiments

3.1 Datasets

VST-F and VST-B were independently trained on two commonly used synthetic datasets: SynthText (ST) [9], MJSynth (MJ) [11, 13]. All models were trained from scratch without finetuning on any real datasets. We evaluated the proposed models on four regular text datasets: IIIT 5K-words (IIIT) [26] , Street View Text (SVT) [38], ICDAR 2003 (IC03) [24], ICDAR 2013 (IC13) [15] , and three irregular text datasets: ICDAR 2015 (IC15) [16], Street View Text Perspective (SVTP) [29] and CUTE [31] . IIIT contains 3000 cropped images collected from Google image search, from which we randomly selected 647 Google Street View images as the testing set. Following [39], we selected 867 cropped images from IC03 for testing. IC13 contains 1095 testing images and following previous work [47] we discarded images containing non-alphanumeric characters or having fewer than three characters, and finally the testing set contained 1015 images. For irregular datasets, IC15 contains 2077 cropped images by Google Glasses, and we used 1811 images after discarding extremely distorted images. SVTP and CUTE contain 639 and 288 images respectively.

3.2 Experiment Configurations

Module $\mathcal{C}$ was implemented as a four-stage resnet as [23], with each stage having 1,2,5 and 3 blocks. Module $\mathcal{V}$, $\mathcal{I}$ and $\mathcal{S}$ each consisted of 3 transformer layers. The number of parameters was about 63.99M for VST-F, and 63.65M for VST-B. Image patches were resized with aspect-ratio unchanged so that $H = 48$, and the width was trimmed or padded to $W = 160$ pixels. Feature maps were of size $6 \times 40 \times 512$, and we set $w = 40, h = 6, n = 240$. The number of classes was set to 38, including 0-9,a-z, and two control characters, [unk] and [eos]. We assumed that the maximum character length was $t = 25$. We used online data augmentation techniques for training, including distortion, stretch, perspective

Method	Regular test datasets				Irregular test datasets		
	IIIT	SVT	IC03	IC13	IC15	SVTP	CUTE
AON [6]	87.0	82.8	91.5	_	68.2	73.0	76.8
ASTER [35]	93.4	89.5	94.5	91.8	76.1	78.5	79.5
NRTR [33]	86.5	88.3	95.4	94.7	_	_	_
SAR [17]	91.5	84.5		91.0	69.2	76.4	83.3
DAN [41]	94.3	89.2	95.0	93.9	74.5	80.0	84.4
HRGAT [46]	94.7	88.9	_	93.2	79.5	80.9	85.4
SRN [47]	94.8	91.5	_	95.5	82.7	85.1	87.8
SCATTER [19]	93.7	92.7	96.3	93.9	82.2	86.9	87.5
GTC [10]	95.5	92.9	95.2	94.3	82.5	86.2	92.3
RobustScanner [49]	95.3	88.1	_	94.8	77.1	79.5	90.3
JVSR [4]	95.2	92.2	_	95.5	84.0	85.7	89.7
PREN [45]	95.6	94.0	95.8	96.4	83.0	87.6	91.7
ABINet-SV [8]	95.4	93.2	_	96.8	84.0	87.0	88.9
ABINet-LV [8]	96.2	93.5	_	**97.4**	**86.0**	**89.3**	89.2
VST-B (OURS)	**96.7**	93.8	**97.5**	96.6	85.2	88.4	**95.5**
VST-F (OURS)	**96.7**	**94.0**	97.3	96.7	85.4	89.0	**95.5**

Table 1: When compared with previous work, our approach achieves very competitive results. VST-B denotes the $\mathcal{C}$+$\mathcal{V}$+$\mathcal{A}_1$+I+$\mathcal{A}_2$ basic model and VST-F the $\mathcal{C}$+$\mathcal{V}$+$\mathcal{A}_1$+I+$\mathcal{A}_2$+$\mathcal{S}$ full model. '-' denotes data not available or config not the same.

transform, blurring, colour jitter etc. We set $\alpha_1 = 0.5$, $\alpha_2 = 0.3$, $\alpha_3 = 0.3$, $\alpha_4 = 0.3$ as the tuning coefficients for each loss in Eq. 3. On training stage, the batch size was set to be 256 and sampled from ST, MJ datasets with weight 0.5,0.5 to balance the datasets [2]. We used Adam optimizer with the initial learning rate 1e-4 and decreased to 1-e5 when the loss plateaued. The models were trained from scratch without any extra data. All experiments were implemented on 4 NVIDIA Tesla V100 GPUs and it took roughly 3 days to finish the training.

3.3 Comparison with State-of-the-art

We compared VST-B and VST-F with popular models on seven public benchmarks. The results were listed in Tab.1. Comparing with state-of-the-art approaches, both VST-B and VST-F achieved competitive recognition accuracy. Though ABINET-LV [8] performed slightly better than ours on three out of the seven datasets, it used extra curbersome pretrained language models. Extra corpus helps ABINet correctly classify some texts that can be only inferenced from specific contexts, especially unseen texts that never appear in the training sets. On the rest four datasets, our models outperformed SOTA by wide margins (0.5% IIIT, 1.2% IC03, 2.8% CUTE). Moreover, VST-B and VST-F were about three times faster than ABINET in inference, where the inference time for ABINet was 42ms per image and the it was 17ms per image for VST-B and VST-F.

3.4 Ablation Study

To verify whether each module is critical for the final performance, ablation study was conducted by subsequently adding one module starting from the most basic $\mathcal{C}$+$\mathcal{V}$ configuration.

Module	Regular test datasets				Irregular test datasets		
	IIIT	SVT	IC03	IC13	IC15	SVTP	CUTE
$\mathcal{CV}$	95.10	91.94	95.72	95.52	81.83	86.23	91.33
$\mathcal{CVA}_1$	95.63	91.94	96.41	95.55	82.31	86.95	91.76
$\mathcal{CVA}_1\mathcal{IA}_2/S_s$	96.57	93.82	97.35	96.55	85.20	88.37	94.79
$\mathcal{CVA}_1\mathcal{IA}_2/S_{vs}$	96.67	93.82	**97.46**	96.55	85.15	88.37	95.49
$\mathcal{CVA}_1\mathcal{IA}_2\mathcal{S}$	**96.73**	**94.00**	97.34	**96.65**	**85.42**	**89.00**	**95.49**

Table 2: Recognition accuracy increases when extra module is added. $/S_s$ and $/S_{vs}$ denote decoding from secondary semantics and visually semantics respectively, $\mathcal{CVA}_1\mathcal{IA}_2\mathcal{S}$ means modules $\mathcal{C}$+$\mathcal{V}$+$\mathcal{A}_1$+I+$\mathcal{A}_2$+$\mathcal{S}$.

The results are shown in Tab. 2, from which we observe that a consistent performance gained when a new module was added. Also note that $\mathcal{C}$+$\mathcal{V}$+$\mathcal{A}_1$ performed better than $\mathcal{C}$+$\mathcal{V}$, which verifies the effectiveness of VS-Align module. For the $\mathcal{C}$+$\mathcal{V}$+$\mathcal{A}_1$+I+$\mathcal{A}_2$ configuration, there are two ways of decoding: decoding from the secondary semantics (s_2) or from the visually semantics (s_{vs}). When the full setting $\mathcal{C}$+$\mathcal{V}$+$\mathcal{A}_1$+$\mathcal{I}$+$\mathcal{A}_2$+S was used, it achieved the best performance on 6 datasets. From Tab. 1 and Tab. 2 we can notice that under the same experimental configurations and the same initialization, the performance of VST-F was steadily superior to the performance of VST-B, which indicates that the semantic module is useful for learning more powerful representations.

3.5 Visualization

To visualize how the semantic features of an image make impact on the visual features in Module $\mathcal{I}$, at each time step, we computed the heatmap by averaging all the 8 attention heads and superimposed it on the image. Remember that in Module $\mathcal{I}$, each head focuses on different aspects of an image, but on average the attention should focus on the spatial location which corresponds to the character that is being dealt with at that time step. Fig. 3 shows three examples. As we can see, the foci of the heatmaps were basically consistent with the characters being dealt with. For some characters, the foci were slightly shifted, probably due to the translation-invariant property of the convnet we used in the work. Overall, the visualization illustrates that the interaction module worked as expected. .

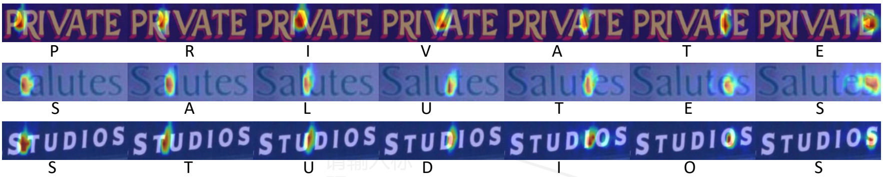

Figure 3: Visualization of attention maps for decoding each character.

We also visualized the attention heatmaps of the primary and the secondary VS-Align modules. As expected, Fig. 4 verifies that the attention maps of the second VS-Align module appeared to be more precise than the first one.

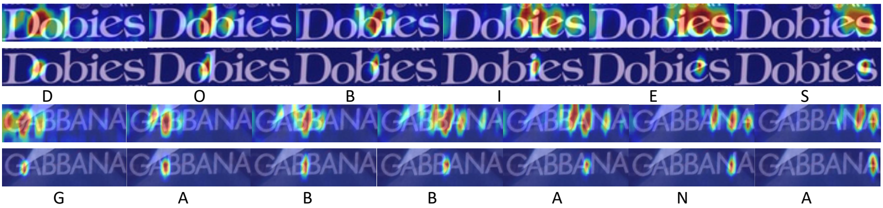

Figure 4: Visualization of attention heatmaps of the primary and secondary VS-Align modules. For each of the two examples, the top (bottom) row shows the heatmaps of the primary (secondary) VS-Align module.

4 Conclusion

In this paper, we propose a Visual-Semantic Transformer (VST) for scene text recognition. VST is a transformer-based model that can efficiently learn the semantic feature without using an external language model. The superiority of VST is mainly due to its concise architecture and the innovative module designs, in particular, the visual-semantic alignment module and the interaction module. Extensive experiments on regular and irregular scene text recognition datasets have verified the effectiveness of the model.

References

[1] Aviad Aberdam, Ron Litman, Shahar Tsiper, Oron Anschel, Ron Slossberg, Shai Mazor, R Manmatha, and Pietro Perona. Sequence-to-sequence contrastive learning for text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15302–15312, 2021.

[2] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 4715–4723, 2019.

[3] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *arXiv preprint arXiv:2006.11477*, 2020.

[4] A. Bhunia, Aneeshan Sain, Amandeep Kumar, S. Ghose, Pinaki Nath Chowdhury, and Yi-Zhe Song. Joint visual semantic reasoning: Multi-stage decoder for text recognition. *ArXiv*, abs/2107.12090, 2021.

[5] Ayan Kumar Bhunia, Aneeshan Sain, Amandeep Kumar, Shuvozit Ghose, Pinaki Nath Chowdhury, and Yi-Zhe Song. Joint visual semantic reasoning: Multi-stage decoder for text recognition. In *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 14920–14929, 2021. doi: 10.1109/ICCV48922.2021.01467.

[6] Zhanzhan Cheng, Yangliu Xu, Fan Bai, Yi Niu, Shiliang Pu, and Shuigeng Zhou. Aon: Towards arbitrarily-oriented text recognition. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5571–5579, 2018.

[7] 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. *arXiv preprint arXiv:2010.11929*, 2020.

[8] Shancheng Fang, Hongtao Xie, Yuxin Wang, Zhendong Mao, and Yongdong Zhang. Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition. 2021.

[9] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[10] 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.

[11] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[12] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in neural information processing systems*, pages 2017–2025, 2015.

[13] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Reading text in the wild with convolutional neural networks. *International Journal of Computer Vision*, 116(1):1–20, 2016.

[14] Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

[15] 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 *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493, 2013.

[16] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, Faisal Shafait, Seiichi Uchida, and Ernest Valveny. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160, 2015.

[17] Hui Li, Peng Wang, Chunhua Shen, and Guyu Zhang. Show, attend and read: A simple and strong baseline for irregular text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8610–8617, 2019.

[18] Ling Li, Matthew J. Connors, Mathias Kolle, Grant T. England, Daniel I. Speiser, Xianghui Xiao, Joanna Aizenberg, and Christine Ortiz. Multifunctionality of chiton biomineralized armor with an integrated visual system. *Science*, 350(6263):952–956, 2015. doi: 10.1126/science.aad1246. URL `https://www.science.org/doi/abs/10.1126/science.aad1246`.

[19] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R. Manmatha. Scatter: Selective context attentional scene text recognizer. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11962–11972, 2020.

[20] Ron Litman, Oron Anschel, Shahar Tsiper, Roee Litman, Shai Mazor, and R Manmatha. Scatter: selective context attentional scene text recognizer. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11962–11972, 2020.

[21] Wei Liu, Chaofeng Chen, Kwan-Yee K Wong, Zhizhong Su, and Junyu Han. Star-net: A spatial attention residue network for scene text recognition. In *BMVC*, volume 2, page 7, 2016.

[22] Shangbang Long, Xin He, and Cong Yao. Scene text detection and recognition: The deep learning era. *International Journal of Computer Vision*, 129(1):161–184, 2021.

[23] Ning Lu, Wenwen Yu, Xianbiao Qi, Yihao Chen, Ping Gong, Rong Xiao, and Xiang Bai. Master: Multi-aspect non-local network for scene text recognition. *Pattern Recognition*, 117:107980, 2021.

[24] S.M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, and R. Young. Icdar 2003 robust reading competitions. In *Seventh International Conference on Document Analysis and Recognition, 2003. Proceedings.*, volume 3, pages 682–687, 2003.

[25] Pengyuan Lyu, Zhicheng Yang, Xinhang Leng, Xiaojun Wu, Ruiyu Li, and Xiaoyong Shen. 2d attentional irregular scene text recognizer. *arXiv preprint arXiv:1906.05708*, 2019.

[26] Anand Mishra, Karteek Alahari, and Cv Jawahar. Scene text recognition using higher order language priors. In *Proceedings of the British Machine Vision Conference*, pages 127.1–127.11. BMVA Press, 2012. ISBN 1-901725-46-4. doi: http://dx.doi.org/10.5244/C.26.127.

[27] Nguyen Nguyen, Thu Nguyen, Vinh Tran, Minh-Triet Tran, Thanh Duc Ngo, Thien Huu Nguyen, and Minh Hoai. Dictionary-guided scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7383–7392, 2021.

[28] Yash Patel, Lluis Gomez, Marçal Rusinol, and Dimosthenis Karatzas. Dynamic lexicon generation for natural scene images. In *European Conference on Computer Vision*, pages 395–410. Springer, 2016.

[29] Trung Quy Phan, Palaiahnakote Shivakumara, Shangxuan Tian, and Chew Lim Tan. Recognizing text with perspective distortion in natural scenes. In *2013 IEEE International Conference on Computer Vision*, pages 569–576, 2013. doi: 10.1109/ICCV.2013.76.

[30] Zhi Qiao, Yu Zhou, Dongbao Yang, Yucan Zhou, and Weiping Wang. Seed: Semantics enhanced encoder-decoder framework for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13528–13537, 2020.

[31] 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*, 41(18):8027–8048, 2014.

[32] Ahmed Sabir, Francesc Moreno-Noguer, and Lluís Padró. Visual re-ranking with natural language understanding for text spotting. In *Asian Conference on Computer Vision*, pages 68–82. Springer, 2018.

[33] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 781–786, 2019.

[34] 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. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[35] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018.

[36] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[37] Zhaoyi Wan, Jielei Zhang, Liang Zhang, Jiebo Luo, and Cong Yao. On vocabulary reliance in scene text recognition. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11422–11431, 2020.

[38] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464, 2011.

[39] Kai Wang, Boris Babenko, and Serge Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464. IEEE, 2011.

[40] Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. Learning deep transformer models for machine translation. *arXiv preprint arXiv:1906.01787*, 2019.

[41] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. In *AAAI*, pages 12216–12224, 2020.

[42] Yuxin Wang, Hongtao Xie, Shancheng Fang, Jing Wang, Shenggao Zhu, and Yongdong Zhang. From two to one: A new scene text recognizer with visual language modeling network. *arXiv preprint arXiv:2108.09661*, 2021.

[43] Liang Wu, Chengquan Zhang, Jiaming Liu, Junyu Han, Jingtuo Liu, Errui Ding, and Xiang Bai. Editing text in the wild. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 1500–1508, 2019.

[44] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhudinov, Rich Zemel, and Yoshua Bengio. Show, attend and tell: Neural image caption generation with visual attention. In *International conference on machine learning*, pages 2048–2057. PMLR, 2015.

[45] Ruijie Yan, Liangrui Peng, Shanyu Xiao, and Gang Yao. Primitive representation learning for scene text recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 284–293, 2021.

[46] Lu Yang, Peng Wang, Hui Li, Zhen Li, and Yanning Zhang. A holistic representation guided attention network for scene text recognition. *Neurocomputing*, 2020.

[47] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 12113–12122, 2020.

[48] Hongyuan Yu, Chengquan Zhang, Xuan Li, Junyu Han, Errui Ding, and Liang Wang. An end-to-end video text detector with online tracking. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 601–606, 2019.

[49] Xiaoyu Yue, Zhanghui Kuang, Chenhao Lin, Hongbin Sun, and Wayne Zhang. Robustscanner: Dynamically enhancing positional clues for robust text recognition. In *European Conference on Computer Vision*, pages 135–151. Springer, 2020.