Self-attention based Text Knowledge Mining for Text Detection

Qi Wan[1,2,3], Haoqin Ji[1,2,3], and Linlin Shen[1,2,3*]
[1]Computer Vision Institute, School of Computer Science and Software Engineering
[2]Shenzhen Institute of Artificial Intelligence and Robotics for Society
[3]Guangdong Key Laboratory of Intelligent Information Processing
Nanhai Avenue 3688, Shenzhen University, Shenzhen, China
{wanqi2019, jihaoqin2019}@email.szu.edu.cn, llshen@szu.edu.cn

Abstract

Pre-trained models play an important role in deep learning based text detectors. However, most methods ignore the gap between natural images and scene text images and directly apply ImageNet for pre-training. To address such a problem, some of them firstly pre-train the model using a large amount of synthetic data and then fine-tune it on target datasets, which is task-specific and has limited generalization capability. In this paper, we focus on providing general pre-trained models for text detectors. Considering the importance of exploring text contents for text detection, we propose STKM (Self-attention based Text Knowledge Mining), which consists of a CNN Encoder and a Self-attention Decoder, to learn general prior knowledge for text detection from SynthText. Given only image level text labels, Self-attention Decoder directly decodes features extracted from CNN Encoder to texts without requirement of detection, which guides the CNN backbone to explicitly learn discriminative semantic representations ignored by previous approaches. After that, the text knowledge learned by the backbone can be transferred to various text detectors to significantly improve their detection performance (e.g., 5.89% higher F-measure for EAST on ICDAR15 dataset) without bells and whistles. Pre-trained model is available at: https://github.com/CVI-SZU/STKM

1. Introduction

Scene text detection has drawn much attention in both academic communities and industries due to its ubiquitous real-world applications, such as online education, product search, instant translation, and video scene parsing. Benefited from the rapid development of deep Convolutional Neural Networks [11] in Object Detection [16, 24, 25] and Image Segmentation [19, 2, 26] over the past few years, re-

[*]Corresponding Author: Linlin Shen.

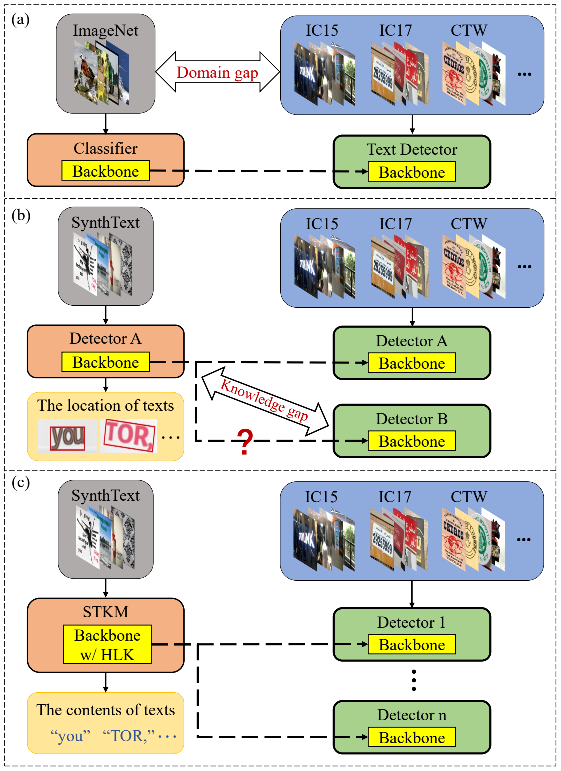


Figure 1. Comparisons of different pre-training strategies. Dashed arrows indicate fine-tuning. (a) Fine-tuned from ImageNet pre-training. (b) Fine-tuned from SynthText pre-training, where the red question mark indicates the knowledge gap between detector A and detector B. (c) Our approach, where the "HLK" denotes high level knowledge.

cent scene text detectors have achieved significant progress. Most of these methods apply ImageNet [27] pre-training to speed up convergence as well as improve final accuracy. However, there exists an obvious domain gap between natural images and scene text images. Some methods [1, 41, 18, 37] try to fine-tune models using initialization parameters pre-trained on a large amount of synthetic data for text detection, which outperforms ImageNet based pre-training, but still suffers from these shortcomings: (1) The generalization capability of such pre-trained backbones is limited. The weights trained with a specific detector may not be able to obtain good results in other detectors. Therefore, each text detection algorithm needs to be pre-trained again, which leads to a lot of redundant computational costs. (2) Text content is usually ignored. Different from generic object detection, text detection only classifies region proposals as text or non-text, no information about text content is extracted. Therefore, texture-like content in the image is easily detected as text.

As shown in Figure 1, in this paper, our motivation is to provide powerful pre-trained deep models for text detection, which contains more general prior knowledge by aggregating semantic representations of text contents. Inspired by transformer [33], we utilize self-attention mechanisms to devise a dedicated network called STKM (Self-attention based Text Knowledge Mining) to learn useful prior knowledge for text detection. As a result, the pre-trained backbone of STKM can be fine-tuned for various text detectors to significantly improve their detection performance. To be specific, we firstly extract features from standard CNN backbone, and then decode the flatten features using a cascaded self-attention architecture to directly recognize all the texts without requirement of detection. Our model can be trained end-to-end and only requires image-level text annotations, whose labeling cost is much cheaper than that of text location annotation. Furthermore, STKM is able to provide more general text knowledge, i.e., the backbone can be transferred to different text detection networks to achieve state-of-the-art performance. We fine-tune the pre-trained STKM backbone with diverse networks and datasets to verify its effectiveness. In particular, by replacing ImageNet pre-training with STKM, EAST [43] and PSENet [34] can achieve 5.89% and 5.64% higher F-measure on ICDAR2015 [10] dataset, respectively.

In summary, the main contributions of this paper are two-fold:

- We propose STKM, which can be trained end-to-end, to acquire general text knowledge for following text detection tasks. To the best of our knowledge, we are the first attempt to provide general pre-trained models for text detection.
- Without bells and whistles, extensive experiments show that the STKM can improve the performance of various detectors by a great margin on different benchmarks.

2. Related Work

Deep learning based scene text detectors available in literature can be mainly categorized into regression-based, segmentation-based and connected component-based (CC-Based) methods.

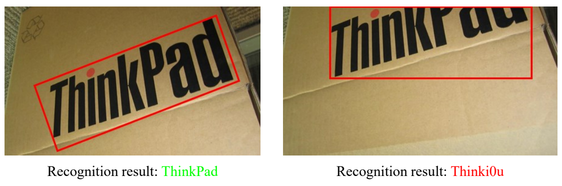


Figure 2. An example of incomplete text for recognition. Left: Complete text is both detected and recognized correctly. Right: Incomplete text is easily detected but difficult to be recognized. The detector EAST [43] and recognition network CRNN [29] are used.

Regression-Based Methods takes scene text as general object, general detection frameworks [16, 24, 25] are applied to locate text boxes by predicting the offsets from anchors or pixels. However, texts are often presented in irregular boxes with various aspect ratios. To handle this problem, TextBoxes [13] extends SSD [16] to effectively capture various text shapes by modifying the size of convolutional kernel and anchor box. RRPN [22] introduces rotation to both anchors and RoI-Pooling in Faster R-CNN, to detect arbitrary-oriented scene texts. EAST [43] uses FCN [19] to directly predict pixel-level quadrangles of word candidates without region proposals and preset anchors.

Segmentation-Based Methods regards detection task as a semantic segmentation problem by directly segmenting text regions with irregular shapes. PixelLink [4] firstly segments text instances with linking pixels and then generates bounding boxes from segmentation results. SSTD [9] proposes an attention mechanism by FCN to substantially suppresses background interference in the feature maps, which achieves accurate detection of words, particularly at small sizes. PSENet [34] generates multi scale of kernels for each text instance, and gradually expands the minimal scale kernel to the complete text instance with a progressive scale algorithm. TextField [36] learns a direction field to link neighbor pixels and use a simple morphological-based post-processing to achieve the final detection.

CC-Based Methods firstly extract individual text parts or characters, then use a link or group post-processing procedure to generate final texts. CTPN [31] leverages a modified Faster R-CNN [6] to extract horizontal text proposals with fixed-width for naturally connecting dense text components and generating horizontal text lines by a recurrent neural network. CRAFT [1] detects the text area by exploring the affinity between characters. DRRG [41] is the first attempt to perform deep relational reasoning between different small components via graph convolutional network for arbitrary shape text detection.

The above works have achieved remarkable progress in text detection area. However, most of these approaches

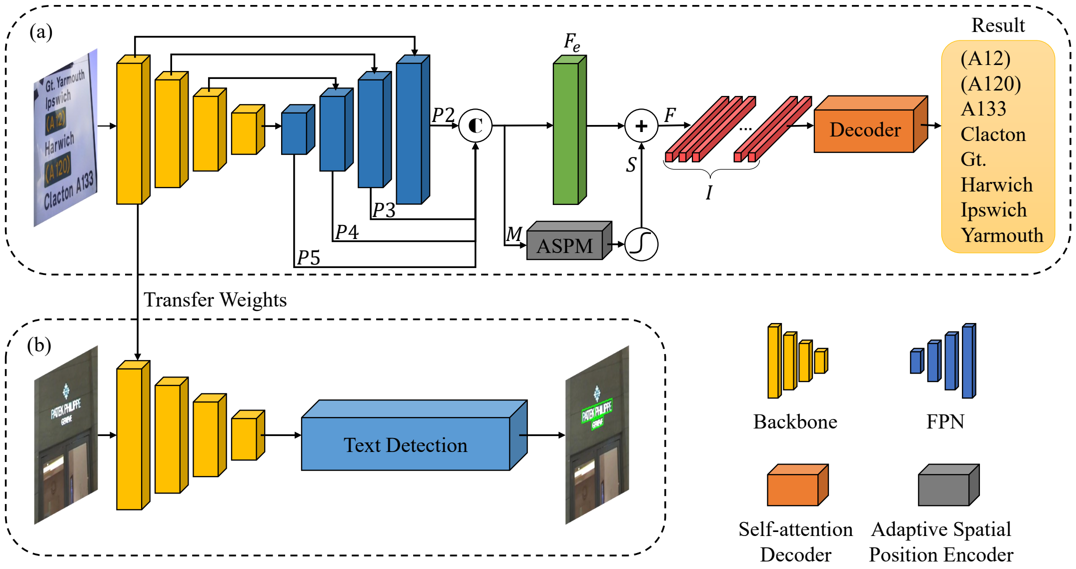


Figure 3. Illustration of our overall pipeline, (a) is the structure of STKM, (b) represents general text detection task. We firstly train STKM using an image level text recognition task. After that, the text knowledge learned by the backbone can be transferred to general text detectors to significantly improve their detection performances.

directly apply networks pre-trained using ImageNet, without paying particular attention to the importance of pre-training methodology. Although there are a few methods [41, 1, 13, 20, 5, 37, 40] proposed to pre-train their models using synthetic data, they are primarily designed to address labeling deficiencies in target datasets, such as the lack of character level annotations. In this paper, we focus on providing general and powerful network models for text detection. To achieve this goal, we design an efficient text feature mining network, to enable the standard backbone to learn strong prior knowledge helpful for text detection.

3. Method

In this section, we first introduce the overall pipeline of the proposed STKM network. Next, we present details of ordered text sequence generation and design of loss function. At last, we introduce the scheme of fine-tuning.

Models pre-trained using ImageNet have been proven to capture rich semantic information from images. However, as the text information of image is not explicitly studied, ImageNet pre-training can not capture text knowledge. In order to learn text specific representation in image and provide better prior knowledge for text detection, we propose STKM for text knowledge mining. For scene text images, text recognition requires more complete information than text detection. As shown in Figure 2, it is hard to recognize the incomplete text which, however, can be detected easily. Therefor our STKM aims at mining rich prior knowledge of text into pre-trained models by text recognition task. Due to the small size available and large number of text instances in scene text images, STKM is designed to directly recognize all word instances in the image, without requirement of detection

3.1. Network Architecture

Inspired by transformer [33] , we propose STKM to acquire general prior knowledge for text detection using attention mechanism. As shown in Figure 3, STKM consists of two main components, a CNN Encoder to transform the input image into the feature map with high-level semantic information, and a Self-attention Decoder to decode features extracted by the CNN Encoder into text sequence.

CNN Encoder. The basic framework of CNN Encoder consists of standard CNN backbone and FPN [15]. We firstly extract four feature maps $\{P_2, P_3, P_4, P_5\}$ from different layers of backbone with 256 channels. In order to combine different levels of semantic features together, we fuse four feature maps to obtain the enhanced feature map F_e through operation *C(.)*, which is shown as :

$$\begin{aligned} F_e &= C(P_2, P_3, P_4, P_5) \\ &= Down_{\times 2}(P_2) \parallel P_3 \parallel Up_{\times 2}(P_4) \parallel Up_{\times 4}(P_5) \end{aligned} \tag{1}$$

where "$\|$" refers to concatenation, $Down_{\times 2}$ denotes 2 times down-sampling and $Up_{\times 2}$, $Up_{\times 4}$ represent 2 and 4 times up-sampling, respectively. After that, feature map F_e is fed into a 1×1 convolution layer to reduce the number of channels from 1024 to 512. The shape of F_e is $H \times W \times C$, where H, W is $\frac{1}{4}$ height and width of the input image, and C presents channels.

Since the subsequent operations will flat the output of CNN Encoder into a sequence, the spatial relationship of F_e will be lost. We design an Adaptive Spatial Position Encoder Module(ASPM) to relieve this problem. ASPM is a fully convolutional network, with a sigmoid layer. We generate a coordinate matrix M as input of ASPM, which can be shown as:

$$\begin{cases} M(j,i,0) = j/H & j \in [0,H) \\ M(j,i,1) = i/W & i \in [0,W) \end{cases} \tag{2}$$

where H and W denote the height and wight of the feature map F_e, respectively. The generated coordinate matrix M has the same length and width as F_e. The ASPM can adaptively encode the coordinate matrix M into spatial position encoding S. Finally, we add S and F_e to obtain the output of CNN Encoder F, which has the same shape as F_e.

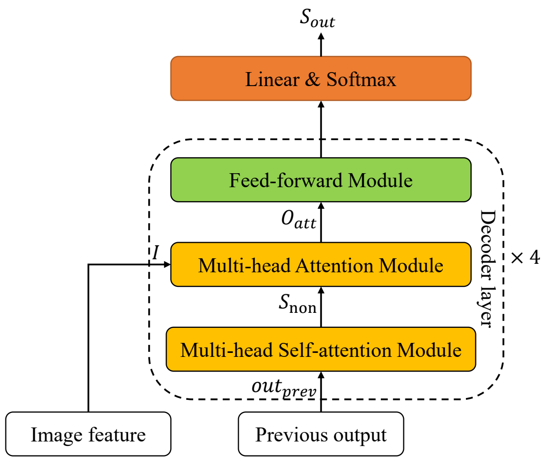


Figure 4. The overview of the Self-Attention module.

Self-attention Decoder. The main challenge of our design is how to convert the feature extracted from backbone into a single text sequence, which determine the quality of text knowledge mined by STKM. Inspired by transformer, we design a Self-attention Decoder to achieve this goal. As shown in Figure 4, we stack four identical decoder layers to construct the Self-attention Decoder, and each decoder layer consists of a masked multi-head self-attention module, a multi-head attention module and a feed-forward module. We present the detail of these modules in the supplementary material due to the page limit. First, the masked multi-head self-attention module obtains the non-local sequence S_{non} by modeling dependencies between different characters within previous outputs out_{prev}. After that, we flat the feature map F output by CNN Encoder into a feature sequence I with the shape of $K \times C$, where K equals to $H \times W$, and C is the dimensions of each vector in the feature sequence. The output of multi-head attention module is computed by a weighted operator applied to flatten

Algorithm 1 OTSG

```
Input: The list of $N_w$ words present in an input image: $W$;
       The maximum number of characters for the text sequence: $L$;
Output: An ordered text sequence inserted with required flags: $G$;
 1: function GENERATION($W$, $L$)
 2:     $G \leftarrow \varnothing$
 3:     Sort the words in list $W$ according to ASCII order, get $W_{sorted}$;
 4:     Add $SOS$ to $G$;
 5:     for $i = 1$ to $N_w$ do
 6:         Add the characters of the $i$-th word in $W_{sorted}$ to $G$;
 7:         Add $WOS$ to $G$;
 8:     end for;
 9:     Add $EOS$ to $G$;
10:     if the length of $G$ is less than $L$ then
11:         Add padding flag $POS$ to $G$;
12:     end if
13:     for each character $g_i$ in $G$ do
14:         generate the one-hot vector with dimension $N_c$:
            $[g_i(1) \dots g_i(n) \dots g_i(N_c)]$;
15:     end for
16:     return $G$.
17: end function
```

sequence I. The attention weights are denoted as α with the shape of $L \times K$, where L represents the length of output sequence. We compute the output of attention module as follows:

$$\begin{cases} \alpha = Attn(I, S_{non}) \\ O_{att} = \alpha \times I \end{cases} \tag{3}$$

where $Attn$ is an operation that calculates α using I and S_{non}. We present the details of $Attn$ in the supplementary material. The shape of O_{att} is $L \times C$. Each vector in O_{att} can be decoded into a character. After that O_{att} is fed into a feed-forward module, which consists of two linear layers and one ReLU layer in order to provide the network with non-linear transformations. At last, we take the output of the last decoder layer as the input of classification module, which consists of a single linear layer and softmax layer, to get the output sequence S_{out}.

The S_{out} is a sequence consisting of L characters $\{c_1, c_2, \cdots, c_i, \cdots, c_L\}$, where each character is represented by a probability vector with dimension N_c, $c_i = [c_i(1), c_i(2), \cdots, c_i(n), \cdots, c_i(N_c)]$. In this paper, we consider to recognize $N_c = 98$ characters, *i.e.* 10 digits, 52 case-sensitive letters, 32 punctuation characters and 4 special flags for word separation and sequence paddings (introduced in the next section). In the probability vector, the value of $c_i(n)$ represent the probability of c_i to be the n-th character in the list of $N_c(98)$ candidate characters.

3.2. Ordered Word Sequence Generation for Text Recognition

As a network designed for text recognition task, our STKM only requires image level text labels for training. When there are a number of text instances available in the input image, our network output a sequence of recognized words. See Figure 3 (a) for an example of such an ordered word sequence. While the list of words in the output is sorted in ASCII order, the list of labelled word instances in the image need to be sorted in ASCII order as well, such that the two sequences are consistent in order and can be properly matched. We propose an OWSG (Ordered Word Sequence Generation) algorithm to transform the available image level text annotations into an ordered word sequence. In the sequence, the list or words are sorted in the order of appearance of characters based on ASCII table.

To make the output of the network separable for each word, we add SOS (Start of Sequence), EOS (End of Sequence), WOS (Word of Sequence) and POS (Padding of Sequence) to the label. While tags like SOS and EOS are used to indicate the start and end of sequence, a line feed flag WOS is employed to separate different words in the sequence. As the length of sequence is required to be fixed for the output of network, we use flag POS to pad the word sequence to a preset length L.

The detail of text sequence generation is summarized in Algorithm 1. Given an unordered list of words W available in an image, our OWSG algorithm first sorts the words in ASCII order, adds SOS, EOS and WOS flags to beginning, end and middle of the word list, respectively, and pads the sequence to a fixed length L with POS. Finally, each character g_i in the sequence $G = \{g_1, g_2, \ldots, g_i, \ldots g_L\}$ is represented by a one-hot vector $[g_i(1) \ldots g_i(n) \ldots g_i(N_c)]$ with dimension N_c. While all others are set as zero, the n-th element corresponding to character g_i will be set as one.

3.3. Loss Function

Given a ground truth sequence $G = \{g_1, g_2, \ldots, g_i, \ldots, g_L\}$ generated by our OWSG and the output $S_{out} = \{c_1 \ldots c_i \ldots c_L\}$ of our network STKM, a KLDiv (Kullback-Leibler Divergence) loss function is designed in our paper to calculate the loss:

$$D(G \parallel S_{out}) = \sum_{i=1}^{L} \sum_{n=1}^{N_c} g_i^{smooth}(n) \log \frac{g_i^{smooth}(n)}{c_i(n)} \quad (4)$$

$$g_i^{smooth}(n) = g_i(n) \times (1 - \varepsilon) + \varepsilon / N_c \quad (5)$$

where L denotes the number of characters in the two sequences, N_c represent the dimension of one-hot/probability vector for each character and $g_i(n)$, $c_i(n)$ represent the n-th code/probability for the i-th character g_i and c_i, respectively.

We adopt label smoothing presented in [30] to soft the label G and avoid the cases where $g_i(n)/c_i(n) = 0$. As a result, $g_i(n)$ is transformed to $g_i^{smooth}(n)$ first before it's input to equation for calculation of the loss. In equation 5, ε is a pre-set hyper-parameter that adjusts the degree of smoothness and we set $\varepsilon = 0.1$ in this paper. While $D(G \parallel S_{out})$ become calculable for all different G, the smoothing operation can also reduce the gap between the true category probability and the mean probability of other categories, so as to avoid over-fitting.

3.4. Network Training and Backbone Fine-tuning

To learn the useful text knowledge, image level text recognition task is used to train our STKM. In this task, the network is required to recognize all of the texts (word level) available in an input image. Through this task, the backbone in STKM can learn important features about the contents of text, rather than looking at the bounding box of texts only. Once the text knowledge has been learned by the backbone, it can be transferred to general text detectors and fine-tuned by target datasets.

4. Experiments

In this session, we first briefly introduce the datasets and present the implementation details of our methods. Then we transfer the backbone of proposed STKM to EAST and PSENet, and compare them with other state-of-the-art methods on various challenging benchmarks. While EAST is one of the most commonly used algorithms for detecting linear texts of arbitrary orientation, PSENet is designed to detect curved text with the help of segmentation. Finally, we conduct ablation studies and different comparison experiments to demonstrate the effectiveness and generalization ability of our method.

4.1. Datasets

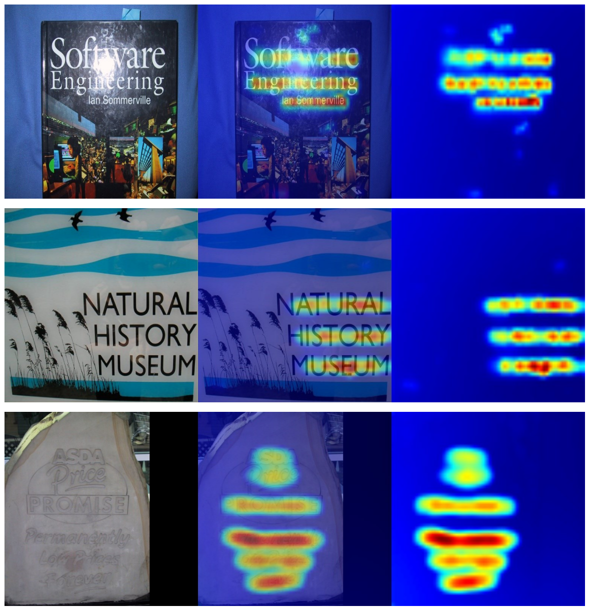
Syhthtext [7] contains more than 800K synthetic images, which are created by pasting words on the image according to certain rules. Most of these images are at word level annotations. This dataset is used to train our STKM.

ICDAR2015 [10] was introduced in the ICDAR 2015 Robust Reading Competition for incidental scene text detection, which consists of 1000 training images and 500 testing images, both with texts in English. The annotations are at the word level and the locations of texts are labeled with quadrilateral boxes.

ICDAR2017 [23] is a multi-lingual (9 languages) datasets, which contains 7,200 training images, 1,800 validation images, and 9,000 testing images. Similar to IC15, the text regions in IC17 are also annotated by the 4 vertices of quadrilaterals.

MSRA-TD500 [38] consists of 300 training and 200 testing images where texts includes Chinese and English. The annotations are at the word line level, and text regions are labeled with rotated rectangles.

TotalText [3] has 1255 training images and 300 testing images. Recently presented in ICDAR 2017, the text regions are labeled with polygon and the annotations are at word-level.


(a)

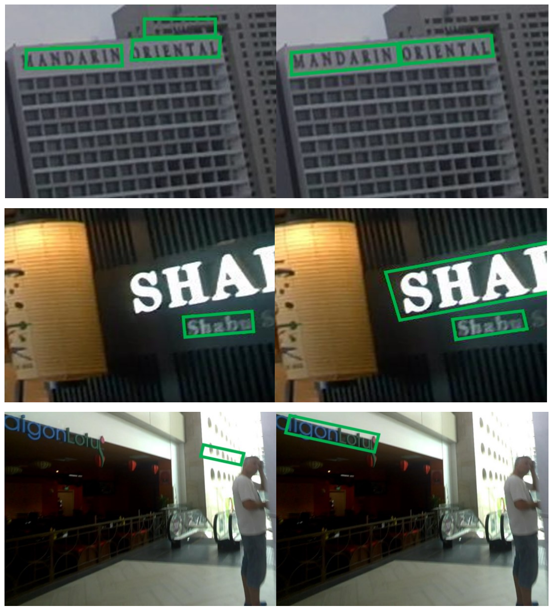

(b)

Figure 5. Visual results of the proposed method. (a) Heatmap visualization. (b) Results on ICDAR2015. Left column: detection results of original EAST. Right column: detection results of EAST with backbone pre-trained in STKM.

CTW-1500 [39] consists of 1000 training and 500 testing images, with total of 10751 text instances. Every image has curved text instances, which are annotated at text-line level by a region of 14 vertices.

4.2. Implementation Details

Training. We use ResNet [8] as the backbone of the proposed STKM and use SynthText dataset mentioned above as training set. For data augmentation, we rotate the original images in an angle range of -20° to +20°. Random cropping is not applied because we don't use location annotations and couldn't confirm whether the text region is clipped or not. The rotated image is resized to 512 $\times$ 512. We apply Adam optimizer to train our model with an initial learning rate 1×10^{-4}. The model is trained end-to-end for 10 epochs on 2 Tesla V100 GPUs with batch size of 32. Unlike other methods, STKM only needs to be trained once, and the weights of backbone can be transferred to different text detectors for fine-tuning on the target dataset.

Backbone Fine-tuning. We fine-tune EAST and PSENet on various public datasets. For EAST, we choose ICDAR2015, ICDAR2017 and MSRA-TD500 as target datasets due to its limitation on processing curved texts. We apply random crop and rotation for input images, which are then resized to different sizes, due to various text lengths and image sizes in different datasets. Specifically, the images of ICDAR2015, ICDAR2017, and MSRA-TD500 are resized to 512 $\times$ 512, 640 $\times$ 640, 640 $\times$ 640, and the batch size is 32, 22, and 22, respectively. Adam optimizer is used for training with initial learning rate 1×10^{-4}. We train EAST for 600 epochs on ICDAR2015 and MSRA-TD500 datasets and the learning rate is decreased by 0.1 every 200 epochs. For ICDAR2017, a total of 150 epochs are trained with a learning rate decrease of 0.1 per 50 epochs.

For PSENet, we select ICDAR2015, TotalText and CTW-1500 as target datasets. The data augmentation for training data follows the steps of the original paper. We train PSENet for 600 epochs using stochastic gradient descent (SGD) optimizer where the batch size is set to 16. Weight decay and momentum are set as 0.0001 and 0.9, respectively. The initial learning rate was set to 1×10^{-3} and decreased by 0.1 after epoch 200 and 400. All the experiments of EAST and PSENet are performed on a single Tesla V100 GPU.

4.3. Visual Results

We show some qualitative results of our method in Figure 5. Inspired by CAM [42]. we firstly visualize the features output by the CNN Encoder to better understand their properties. As shown in Figure 5 (a), the highlighted text area of 2D attention masks indicates that our method is able to learn the positions of characters without text location annotations. Meanwhile, such rich semantic information mined by standard backbones can be useful prior knowledge for subsequent text detection. Take EAST for example, Figure 5 (b)shows that EAST with the backbone pre-trained in STKM can detect text more accurately while the original EAST may easily produce false positives in texture-like areas, like windows. Lot of texts like "SHA" (the 2nd row) and "aigonlofu" (the 3rd row) are also missed by the original EAST. The Self-attention Decoder of our STKM guides the backbone to learn both position information and discriminative semantic representations useful for text detection, so as to reasonably improve the detection performance on target datasets after fine-tuning.

4.4. Ablation Study

To demonstrate the effectiveness of the proposed Self-attention decoder in STKM, we conduct ablation experiments by evaluating the performances of EAST and PSENet on ICDAR2015, ICDAR2017, MSRA-TD500, CTW-1500 and TotalText, and report their results in Table 1. When the self-attention decoder is not included, we replace it with standard ResNet classification head and convert STKM into a character-level multi-label classification network. As shown in Table 1, the inclusion of Self-attention Decoder significantly increases the F-measure of both EAST and PSENet. While the improvement of EAST on ICDAR2015, ICDAR2017 and MSRA-TD500 are 6.74%, 6.99% and 9.34%, respectively, that of PSENet on ICDAR2015, CTW-1500 and TotalText are 5.06%, 3.02% and 5.61%, respectively.

Table 1. Ablation study for Self-attention Decoder. "P", "R" and "F" represent the precision, recall and F-measure respectively. "SD?" means whether Self-attention Decoder is included or not.

Algorithms	Datasets	SD?	P	R	F	Gain
EAST	IC15		78.01	82.14	80.02	
		✓	**88.72**	**84.88**	**86.76**	**+6.74**
	IC17		72.88	54.50	62.36	
		✓	**73.23**	**66.85**	**69.35**	**+6.99**
	TD500		70.05	69.93	69.99	
		✓	**81.64**	**77.15**	**79.33**	**+9.34**
PSENet	IC15		84.31	77.61	80.82	
		✓	**87.78**	**84.06**	**85.88**	**+5.06**
	CTW		79.71	74.25	76.89	
		✓	**80.67**	**79.17**	**79.91**	**+3.02**
	TotalText		80.68	72.81	76.54	
		✓	**86.32**	**78.36**	**82.15**	**+5.61**

4.5. Comparisons with Different Pre-training Strategies

As ImageNet and SynText are widely used datasets to pre-train different text detectors, we perform evaluation of our STKM with these baselines in this section. For a fair comparison, we firstly re-implement the original EAST and PSENet, so that ImageNet pre-training can be applied. Then we replace ImageNet pre-training with other strategies to fine-tune these two text detectors on target datasets. Figure 6 shows the variances of F-measure for EAST on ICDAR2015 dataset, when it is pre-trained by different strategies for various epochs. As shown in the figure, the performance of EAST generally improve with the increase of epochs and become stable when the number approaches 10. There is a large margin between the performances of SKTM, SynthText pre-training and ImageNet pre-training.

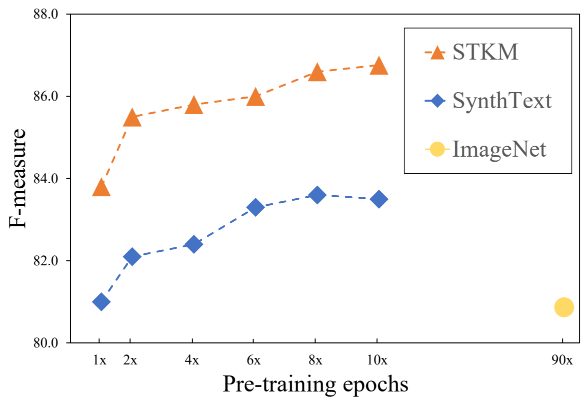


Figure 6. F-measure for different pre-training epochs.

Table 4 shows the performances of different pre-training strategies, together with that of our approach. As there is a large domain gap between natural scene images with text images, the performances of EAST and PSENet using the backbones pre-trained by ImageNet achieves the lowest F-measure. The F-measure gains of EAST using the backbone trained by our SKTM on ICDAR2015 and ICDAR2017 are 5.89% and 3.18%, respectively, which are significantly higher than that of EAST pre-trained by ImageNet and SynthText. Similar results are also suggested by PSENet.

To further justify the generalization capability of our SKTM, we also tested the performance of backbone pre-trained by EAST/PSENet using SynthText and list their results in Table 4. Take IC15 for example, while the performance of EAST using backbone pre-trained by PSENet only achieves slightly better performance (+0.67%) than ImageNet pre-training, that of EAST for IC17 dataset is even substantially lower (-4.35%) than ImageNet pre-training. The poor robustness shows that the backbone trained with a specific detector, e.g PSENet, may not be suitable for other detectors like EAST. When PSENet is concerned, the F-measure gains of our SKTM are also significantly higher than that of backbone pre-trained using EAST for both IC15 and CTW-1500 datasets.

4.6. Comparisons with State-of-the-Arts

In this section, we compare EAST and PSENet using the backbone pre-trained in STKM with other state-of-the-art methods on different challenging benchmarks and report their results in Table 2 and Table 3, respectively. Without using any tricks, the last row of Table 2 shows that the F-measure of EAST applying our STKM has improved significantly on all datasets (5.89% on ICDAR2015, 3.18% on ICDAR2017 and 10.86% on MSRA-TD500). Similarly, as

Table 2. Experimental results on ICDAR2015, ICDAR2017 and MSRA-TD500. The symbol "*" means our implementation. The number in [ ] denotes the relative improvement. "Addl" indicates additional data.

Algorithms	Addl	IC15			IC17			TD500		
		Precsion	Recall	F-measure	Precsion	Recall	F-measure	Precsion	Recall	F-measure
SegLink [28]	✓	76.8	73.1	75	-	-	-	86	70	77
RRD [14]	✓	85.6	79.0	82.2	-	-	-	69	79	74
PixelLink [4]	-	81.7	82.9	82.3	-	-	-	81.1	73	76.8
MSR [37]	-	86.6	78.4	82,3	-	-	-	-	-	-
TextBoxes++ [12]	✓	87.8	78.5	82.9	-	-	-	-	-	-
PAN [35]	-	82.9	77.8	80.3	-	-	-	-	-	-
FOTS [17]	✓	88.8	82	85.3	57.5	79.5	66.7	-	-	-
Lyu et al [21]	✓	94.1	70.7	80.7	83.6	55.6	66.8	-	-	-
EAST* [43]	-	82.83	79.08	80.87	70.27	62.52	66.17	69.43	67.52	68.47
EAST + STKM (ours)	-	**88.72**	**84.88**	**86.76**[**+5.89**]	73.23	**66.85**	**69.35**[**+3.18**]	**81.64**	**77.15**	**79.33**[**+10.86**]

Table 3. Experimental results on ICDAR2015, TotalText and CTW-1500. The symbol "*" means our implementation. The number in [ ] denotes the relative improvement. "Addl" indicates additional data.

Algorithms	Addl	IC15			TotalText			CTW		
		Precsion	Recall	F-measure	Precsion	Recall	F-measure	Precsion	Recall	F-measure
TextSnake [20]	✓	84.9	80.4	82.6	82.7	74.5	78.4	67.9	85.3	75.6
SegLink [28]	✓	73.1	76.8	75.0	30.3	23.8	26.7	42.3	40.0	40.8
Textfield [36]	✓	84.3	83.3	84.1	81.2	79.9	80.6	83	79.8	81.3
LOMO [40]	✓	91.9	83.5	**87.2**	88.6	75.7	81.6	89.2	69.6	78.4
TextDragon [5]	✓	84.8	81.8	83.1	79.5	81.0	80.2	84.5	74.2	79.0
Tian et al. [32]	✓	85.1	84.5	84.8	-	-	-	82,7	77.8	80.1
PSENet* [34]	-	82.5	78.09	80.24	81.8	75.1	78.31	81.63	76.04	78.74
PSENet + STKM (ours)	-	87.78	84.06	85.88[**+5.64**]	86.32	78.36	**82.15**[**+3.84**]	85.08	78.23	**81.51**[**+2.77**]

Table 4. Comparisons of different Pre-training strategies. "P", "R" and "F" represent the precision, recall and F-measure respectively.

Algorithms	Datasets	Pre-train	P	R	F	Gain
EAST	IC15	ImageNet	82.83	79.08	80.87	-
		SynthText	83.22	83.87	83.55	+2.68
		PSENet	83.76	79.53	81.54	+0.67
		STKM(ours)	**88.72**	**84.88**	**86.76**	**+5.89**
	IC17	ImageNet	70.27	62.52	66.17	-
		SynthText	71.25	66.29	68.68	+2.51
		PSENet	71.39	54.51	61.82	-4.35
		STKM(ours)	**73.23**	**66.85**	**69.35**	**+3.18**
PSENet	IC15	ImageNet	82.50	78.09	80.24	-
		SynthText	86.32	80.54	83.33	+3.09
		EAST	86.44	81.34	83.81	+3.57
		STKM(ours)	**87.78**	**84.06**	**85.88**	**+5.64**
	CTW	ImageNet	81.63	76.04	78.74	-
		SynthText	81.80	77.80	79.72	+0.98
		EAST	80.67	79.17	79.91	+1.17
		STKM(ours)	**85.08**	78.23	**81.51**	**+2.77**

show in Table 3, PSENet equipped with the backbone of STKM achieves substantial improvements of 5.64%, 3.76% and 2.77% in F-measure on ICDAR2015, TotalText and CTW-1500, respectively.

Among state of the art approaches like SegLink [28], TexBoxes++ [12] and PAN [35] etc., the F-measure of EAST using the backbone of STKM is consistently the highest on ICDAR2015, ICDAR2017 and MSRA-TD500 datasets. The performance gains of our approach over the runner up are around 1.4%, 3% and 3% on ICDAR2015, ICDAR2017 and MSRA-TD500 datasets, respectively. The F-measure of PSENet using the backbone of STKM is also the highest among state-of-the-art approaches for TotalText and CTW-1500 datasets. For ICDAR2015 dataset, the F-measure of our approach ranks the second, which is about 1.32% lower than that of LOMO [40]. However, compared with PSENet, the network of LOMO is much more complicated. LOMO firstly use SynthText and its location annotations to train their detector for 10 epochs, and then fine-turn it on the IC15 dataset. Note that our approach can also be integrated with LOMO to further improve its performance.

5. Conclusion

We propose in this paper the first text knowledge mining network, whose backbone can be transferred to different text detectors to improve their detection performance. Integrated with the proposed self-attention module, the proposed STKM can learn text knowledge from datasets with only image level text annotations. The learned text knowledge can thereafter be transferred to different text detectors to improve their performance. Extensive experiments on challenging benchmarks demonstrate the effectiveness and generalization ability of our STKM.

6. Acknowledgement

The work was supported by the National Natural Science Foundation of China under grants no. 91959108 and U1713214.

References

[1] Youngmin Baek, Bado Lee, Dongyoon Han, Sangdoo Yun, and Hwalsuk Lee. Character region awareness for text detection. In *CVPR*, pages 9365–9374, 2019. 1, 2, 3

[2] Liang-Chieh Chen, George Papandreou, Iasonas Kokkinos, Kevin Murphy, and Alan L Yuille. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *PAMI*, 40(4):834–848, 2017. 1

[3] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *ICDAR*, volume 1, pages 935–942. IEEE, 2017. 5

[4] Dan Deng, Haifeng Liu, Xuelong Li, and Deng Cai. Pixellink: Detecting scene text via instance segmentation. In *AAAI*, pages 6773–6780, 2018. 2, 8

[5] Wei Feng, Wenhao He, Fei Yin, Xu-Yao Zhang, and Cheng-Lin Liu. Textdragon: An end-to-end framework for arbitrary shaped text spotting. In *ICCV*, pages 9076–9085, 2019. 3, 8

[6] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, pages 580–587, 2014. 2

[7] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *CVPR*, pages 2315–2324, 2016. 5

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016. 6

[9] Pan He, Weilin Huang, Tong He, Qile Zhu, Yu Qiao, and Xiaolin Li. Single shot text detector with regional attention. In *ICCV*, pages 3047–3055, 2017. 2

[10] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *ICDAR*, pages 1156–1160. IEEE, 2015. 2, 5

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[12] Minghui Liao, Baoguang Shi, and Xiang Bai. Textboxes++: A single-shot oriented scene text detector. *TIP*, 27(8):3676–3690, 2018. 8

[13] Minghui Liao, Baoguang Shi, Xiang Bai, Xinggang Wang, and Wenyu Liu. Textboxes: a fast text detector with a single deep neural network. In *AAAI*, pages 4161–4167, 2017. 2, 3

[14] Minghui Liao, Zhen Zhu, Baoguang Shi, Gui-song Xia, and Xiang Bai. Rotation-sensitive regression for oriented scene text detection. In *CVPR*, pages 5909–5918, 2018. 8

[15] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *CVPR*, pages 2117–2125, 2017. 3

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In *ECCV*, pages 21–37. Springer, 2016. 1, 2

[17] Xuebo Liu, Ding Liang, Shi Yan, Dagui Chen, Yu Qiao, and Junjie Yan. Fots: Fast oriented text spotting with a unified network. In *CVPR*, pages 5676–5685, 2018. 8

[18] Yuliang Liu, Hao Chen, Chunhua Shen, Tong He, Lianwen Jin, and Liangwei Wang. Abcnet: Real-time scene text spotting with adaptive bezier-curve network. In *CVPR*, pages 9809–9818, 2020. 1

[19] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *CVPR*, pages 3431–3440, 2015. 1, 2

[20] Shangbang Long, Jiaqiang Ruan, Wenjie Zhang, Xin He, Wenhao Wu, and Cong Yao. Textsnake: A flexible representation for detecting text of arbitrary shapes. In *ECCV*, pages 20–36, 2018. 3, 8

[21] Pengyuan Lyu, Cong Yao, Wenhao Wu, Shuicheng Yan, and Xiang Bai. Multi-oriented scene text detection via corner localization and region segmentation. In *CVPR*, pages 7553–7563, 2018. 8

[22] Jianqi Ma, Weiyuan Shao, Hao Ye, Li Wang, Hong Wang, Yingbin Zheng, and Xiangyang Xue. Arbitrary-oriented scene text detection via rotation proposals. *TMM*, 20(11):3111–3122, 2018. 2

[23] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *ICDAR*, volume 1, pages 1454–1459. IEEE, 2017. 5

[24] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, pages 779–788, 2016. 1, 2

[25] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, pages 91–99, 2015. 1, 2

[26] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 1

[27] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1

[28] Baoguang Shi, Xiang Bai, and Serge Belongie. Detecting oriented text in natural images by linking segments. In *CVPR*, pages 2550–2558, 2017. 8

[29] 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. *PAMI*, 39(11):2298–2304, 2016. 2

[30] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *CVPR*, pages 2818–2826, 2016. 5

[31] Zhi Tian, Weilin Huang, Tong He, Pan He, and Yu Qiao. Detecting text in natural image with connectionist text proposal network. In *ECCV*, pages 56–72. Springer, 2016. 2

[32] Zhuotao Tian, Michelle Shu, Pengyuan Lyu, Ruiyu Li, Chao Zhou, Xiaoyong Shen, and Jiaya Jia. Learning shape-aware embedding for scene text detection. In *CVPR*, pages 4234–4243, 2019. 8

[33] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NIPS*, pages 5998–6008, 2017. 2, 3

[34] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao. Shape robust text detection with progressive scale expansion network. In *CVPR*, pages 9328–9337. IEEE, 2019. 2, 8

[35] Wenhai Wang, Enze Xie, Xiaoge Song, Yuhang Zang, Wenjia Wang, Tong Lu, Gang Yu, and Chunhua Shen. Efficient and accurate arbitrary-shaped text detection with pixel aggregation network. In *ICCV*, pages 8440–8449, 2019. 8

[36] Yongchao Xu, Yukang Wang, Wei Zhou, Yongpan Wang, Zhibo Yang, and Xiang Bai. Textfield: Learning a deep direction field for irregular scene text detection. *TIP*, 28(11):5566–5579, 2019. 2, 8

[37] Chuhui Xue, Shijian Lu, and Wei Zhang. Msr: multi-scale shape regression for scene text detection. In *IJCAI*, pages 989–995. AAAI Press, 2019. 1, 3, 8

[38] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *CVPR*, pages 1083–1090. IEEE, 2012. 5

[39] Liu Yuliang, Jin Lianwen, Zhang Shuaitao, and Zhang Sheng. Detecting curve text in the wild: New dataset and new solution. *arXiv preprint arXiv:1712.02170*, 2017. 6

[40] Chengquan Zhang, Borong Liang, Zuming Huang, Mengyi En, Junyu Han, Errui Ding, and Xinghao Ding. Look more than once: An accurate detector for text of arbitrary shapes. In *CVPR*, pages 10552–10561, 2019. 3, 8

[41] Shi-Xue Zhang, Xiaobin Zhu, Jie-Bo Hou, Chang Liu, Chun Yang, Hongfa Wang, and Xu-Cheng Yin. Deep relational reasoning graph network for arbitrary shape text detection. In *CVPR*, pages 9699–9708, 2020. 1, 2, 3

[42] Bolei Zhou, Aditya Khosla, Agata Lapedriza, Aude Oliva, and Antonio Torralba. Learning deep features for discriminative localization. In *CVPR*, pages 2921–2929, 2016. 6

[43] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang. East: an efficient and accurate scene text detector. In *CVPR*, pages 5551–5560, 2017. 2, 8