Rethinking Text Segmentation: A Novel Dataset and A Text-Specific Refinement Approach

Xingqian Xu[1], Zhifei Zhang[2], Zhaowen Wang[2], Brian Price[2], Zhonghao Wang[1], Humphrey Shi[3,1]

[1]UIUC, [2]Adobe Research, [3]University of Oregon

Abstract

Text segmentation is a prerequisite in many real-world text-related tasks, e.g.*, text style transfer, and scene text removal. However, facing the lack of high-quality datasets and dedicated investigations, this critical prerequisite has been left as an assumption in many works, and has been largely overlooked by current research. To bridge this gap, we proposed* ***TextSeg****, a large-scale fine-annotated text dataset with six types of annotations: word- and character-wise bounding polygons, masks, and transcriptions. We also introduce Text Refinement Network (****TexRNet****), a novel text segmentation approach that adapts to the unique properties of text,* e.g. *non-convex boundary, diverse texture, etc., which often impose burdens on traditional segmentation models. In our TexRNet, we propose text-specific network designs to address such challenges, including key features pooling and attention-based similarity checking. We also introduce trimap and discriminator losses that show significant improvement in text segmentation. Extensive experiments are carried out on both our TextSeg dataset and other existing datasets. We demonstrate that TexRNet consistently improves text segmentation performance by nearly 2% compared to other state-of-the-art segmentation methods. Our dataset and code can be found at https://github.com/SHI-Labs/Rethinking-Text-Segmentation.*

1. Introduction

Text segmentation is the foundation of many text-related computer vision tasks. It has been studied for decades as one of the major research directions in computer vision, and it continuously plays an important role in many applications [2, 55, 56, 47, 9]. Meanwhile, the rapid advances of deep neural nets in recent years promoted all sorts of new text-related research topics, as well as new vision challenges on text. Smart applications, such as font style transfer, scene text removal, and interactive text image editing, require effective text segmentation approaches to parse

Figure 1: Example images and annotations from the proposed TextSeg dataset. From left to right are images, word and character bounding polygons, pixel-level word (dark gray) and word-effect (light gray) masks, and pixel-level character masks.

text accurately from complex scenes. Without any doubt, text segmentation is critical for industrial usages because it could upgrade the traditional text processing tools to be more intelligent and automatic, relaxing tedious efforts on manually specifying text regions.

However, modern text segmentation has been left behind in both datasets and methods. The latest public text segmentation challenge was in 2013-2015, hosted by ICDAR [26]. Since then, three datasets: Total-Text [10], COCO_TS [5], and MLT_S [6], were introduced. However, Total-Text is limited in scale, and the labeling quality in COCO_TS and MLT_S needs further improvement (Figure 5). Moreover, all the three datasets contain only common scene text, discouraging text in other visual conditions, *e.g.*, artistic design and text effects. As a result, these datasets do not meet modern research standards, such as large-scale and fine-annotated. Thus, we propose a new text segmentation dataset: **TextSeg**, that collects images from a wider range of sources, including both scene and design text, and with a richer set of accurate annotations. This dataset would lead to further advancements in text segmentation research.

Additionally, text segmentation algorithms and methods

Hand-written style Artistic Font Stereo Effect

Scene Text Design Text

Figure 2: Images examples from the proposed TextSeg dataset. The left four columns show scene text that dominantly presents in existing text segmentation datasets, and the rest columns are design text w/ or w/o text effects, which distinguishes TextSeg from all the other related datasets.

in recent years fall behind other research topics, partially due to the lack of a proper dataset. Unlike the rapid advances in other segmentation research, only a few studies [46, 6, 14] have brought new text segmentation ideas. Meanwhile, these studies did not provide an intuitive comparison with modern SOTA segmentation approaches and were unable to demonstrate their advantages over other techniques. As aforementioned, effective text segmentation models are valuable in applications. With our strong motivation to bridge this gap, we propose Text Refinement Network (**TexRNet**), and we thoroughly exam its performance on five text segmentation datasets including the proposed TextSeg dataset. The details of our design principles and our network structure are given in Session 3, and experiments and ablations studies are shown in Session 5.

In summary, the main contributions of this paper are in three-folds:

- We introduce a new large-scale fine-annotated text segmentation dataset, TextSeg, consisting of 4,024 text images, including scene text and design text with various artistic effects. TextSeg has six types of annotations for each image, *i.e.*, word- and character-wise quadrilateral bounding polygons, pixel-level masks, and transcriptions. TextSeg surpasses prior datasets on these aspects: 1) more diverse text fonts/styles from diverse sources/collections, 2) more comprehensive annotations, and 3) more accurate segmentation masks.

- We provide a new text segmentation approach, Text Refinement Network (TexRNet), aiming to solve the unique challenges from text segmentation. We design effective network modules (*i.e.*, key features pooling and attention-based similarity checking) and losses (*i.e.*, trimap loss and glyph discriminator) to tackle those challenges, *e.g.*, diverse texture and arbitrary scales/shapes.

- Exhaustive experiments are conducted to demonstrate the effectiveness of the proposed TexRNet, which outperforms SOTA on our TextSeg and on another four representative datasets. Besides, we give prospects for downstream applications that could significantly benefit from text segmentation.

2. Related Work

2.1. Segmentation in Modern Research

Semantic and instance segmentation are popular tasks for modern research. In semantic segmentation, pixels are categorized into a fixed set of labels. Datasets such as PASCAL VOC [15], Cityscapes [12], COCO [34], and ADE20K [61] are frequently used in this task. Traditional graph models,*e.g.*, MRF [31] and CRF [29], predict segments by exploring inter-pixel relationship. After CNNs became popular [28], numerous deep models were proposed using dilated convolutions [60, 7, 8, 50], encoder-decoder structures [44, 60, 8, 33], and attention modules [51, 48, 16]. Instance segmentation methods predict distinct pixel labels for each object instance. These methods can be roughly categorized into top-down approaches [20, 32, 21, 35, 53, 27] and bottom-up approaches [4, 17, 37, 54, 40]. Top-down approaches are two-stage methods that first locate object bounding boxes and then segment object masks within those boxes. Bottom-up approaches locate keypoints [57, 40] and find edges and affinities [17, 37, 54, 4] to assist the segmentation process.

2.2. Text Segmentation

Early methods frequently used thresholding [41, 45] for segmentation particularly on document text images. Yet such methods cannot produce satisfactory results on scene text images with complex colors and textures. Other approaches used low-level features [36, 52, 3] and Markov Random Field (MRF) [38] to bipartite scene text images. In [36], text features created from edge density/orientation were fed into an multiscale edge-based extraction algorithm for segmentation. In [52], a two-stage method was introduced in which foreground color distribution from stage one was used to refine the result for stage two. In [3], seed points of both text and background were extracted from low-level features and were later used in segmentation. Inspired by MRF, [38] formulated pixels as random variables in a graph model, and then graph-cut this model with two pre-selected seeds. In recent years, several deep learning methods [46, 14, 6] were proposed for text segmentation. The method proposed by [46] is a three-stage CNN-based model, in which candidate text regions were detected, refined, and filtered in those stages correspondingly. Another method SMANet was jointly proposed with the dataset MLT_S in [6]. They adopted the encoder-decoder structure from PSPNet [60], and created a new multiscale attention module for accurate text segmentation.

2.3. Text Dataset

Spotlight datasets motivate researchers to invent effective methods to tackle computer vision problems. For example, the MNIST dataset of handwritten digits [30] illustrated the effectiveness of a set of classical algorithms, *e.g.*, KNN [1], PCA [42], SVM [13], etc. In recent years, the huge success of deep learning inspires researchers to create more challenging datasets to push forward the vision research front. Many text datasets are created for OCR purpose like CUTE80 [43], MSRA-TD500 [58], ICDARs [25, 26, 39], COCO-Text [49], and Total-Text [10], which are scene text datasets with word-level bounding boxes. Other datasets such as Synth90K [24] and SynthText [19] are synthetic text dataset for recognition and detection. Among these dataset, ICDAR13 [26] and Total-Text [10] provide pixel-level labels for text segmentation. Recently, Bonechi *et al*. introduced segmentation labels to COCO-Text and ICDAR17 MLT, forming up two new text segmentation datasets COCO_TS [5] and MLT_S [6]. In general, ICDAR13 and Total-Text are relatively smaller sets, and COCO_TS and MLT_S are of larger scale but their labeling quality is not precise.

3. Text Refinement Network

We propose a new approach, namely Text Refinement Network (TexRNet), specifically targets text segmentation. Since text segmentation is intrinsically similar to modern semantic segmentation, the related state-of-the-art methods can be leveraged to provide the base for our proposed TexRNet. Figure 3 overviews the pipeline of TexRNet, which consists of two components: 1) a backbone and 2) the key features pooling and attention module that refines the backbone for the text-domain. The design of the latter module is inspired by the uniqueness of text segmentation, and the principles will be discussed in Section 3.1. The network structure and corresponding loss functions will be detailed in Sections 3.2 and 3.3, respectively.

3.1. Design Principle

Multiple unique challenges distinguish text segmentation from modern semantic segmentation, thus motivating specific designs for text segmentation. In semantic segmentation, common objects, *e.g.*, trees, sky, cars, etc., tend to share texture across different scenes. However, in text segmentation, the text texture may be extremely diverse across different words, although it could be homogeneous inside each word. To accommodate larger texture diversity, TexRNet dynamically activates low-confidence areas according to their global similarity to high-confidence regions, *i.e.*, the yellow block in Figure 3, which aims to adaptively find similar textures in the same scene while relaxing the model from "remembering" those diverse textures.

Another challenge of text segmentation is the arbitrarily scaled text. The commonly adopted convolutional layers in semantic segmentation would limit the receptive field, reducing adaptiveness to diverse scale and aspect ratio. To achieve higher adaptiveness to scale, we adopt the popular non-local concept [51, 48]. We use dot product and softmax to enforce attention on similar texture across the entire image.

3.2. Network Structure

As aforementioned, the backbone can employ an arbitrary semantic segmentation network. Here, we choose two representative works, *i.e.*, ResNet101-DeeplabV3+ [8] and HRNetV2-W48 [50], because they are the milestone and state-of-the-art in semantic segmentation, respectively. The rest of this section will focus on the new designs of TexRNet, *i.e.*, the yellow block in Figure 3, which is the key to boosting text segmentation performance.

Assume an input image $x \in \mathbb{R}^{H\times W\times 3}$, where H and W denote image height and width, respectively. The feature map extracted from the backbone is x_f. The remainder of the proposed TexRNet could be described in the following three sequential components.

Initial Prediction: Similar to most traditional segmentation models, the feature map x_f is mapped to the semantic map x_{sem} through a convolutional layer (the kernel size is 1×1) with bias. After the softmax layer, x_{sem} becomes the initial segmentation prediction x'_{sem}, which can be supervised by ground truth labels as the following.

$$\mathcal{L}_{sem} = \text{CrossEntropy}(x'_{sem}, x_{gt}), \tag{1}$$

where $x'_{sem} = \text{Softmax}(x_{sem})$, and x_{gt} indicates the ground truth label.

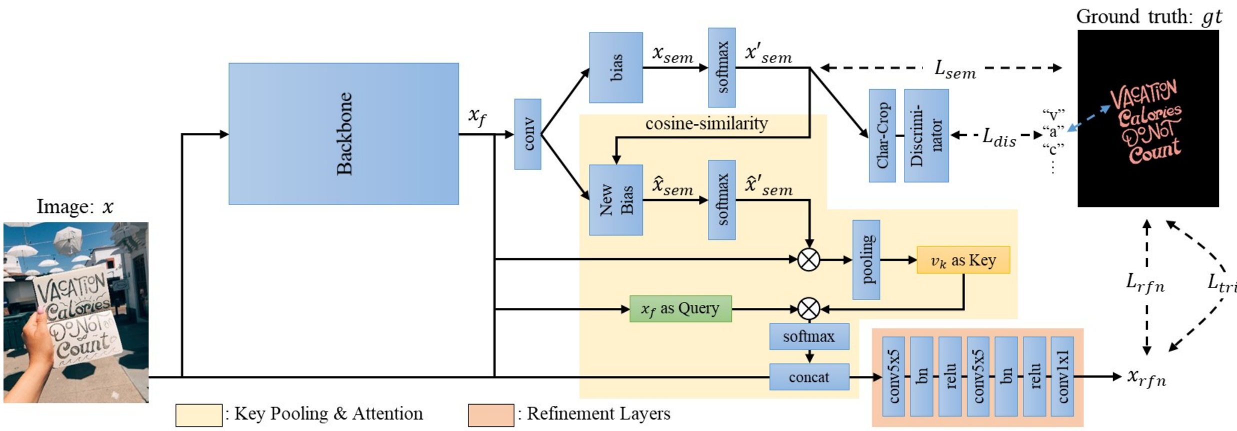


Figure 3: Overview of the proposed TexRNet. Most parts of the network are well-explained in Session 3.2. Besides, "Char-Crop" is the modules that help cropping characters for classifier. It requires ground truth character bounding boxes as input, and therefore will only be available if those boxes are provided. During inference time, neither "Char-Crop" nor "Classifier" need to be loaded, and x_{rfn} will be the model's final output.

Key Features Pooling: Because text does not have a standard texture that can be learned during training, the network must determine that text texture during inference. Specifically, the network should revise low-confidence regions if they share similar texture with high-confidence regions of the same class. To achieve this goal, we need to pool the key feature vector from high-confidence regions for each class $i \in C$ to summarize the global visual property of that class. In our case, $|C| = 2$, corresponding to text and background. More specifically, we conduct a modified cosine-similarity on the initial prediction x'_{sem} and use its output as new biases to transform x'_{sem} into $\hat{x}'_{sem}$ which is the weight map for key pooling. The cosine-similarity is written in Eq. 2, assuming $x'_{sem} \in \mathbb{R}^{c \times n}$, where $c = |C|$ denotes the number of classes, and n is the number of pixels in a single channel.

$$
\begin{aligned}
&\text{CosSim}(x'_{sem}) = X \in \mathbb{R}^{c \times c},\\
&X_{ij} = \begin{cases} \dfrac{x_i x_j^T}{||x_i|| \cdot ||x_j||}, & i \neq j \\ 0, & i = j \end{cases}, \\
&x_i = x'^{(i,)}_{sem} \in \mathbb{R}^{1 \times n},\ i = 1, \cdots, c,
\end{aligned}
\tag{2}
$$

where $\text{CosSim}(\cdot)$ denotes the modified cosine-similarity function, and $x'^{(i,)}_{sem}$ denotes the ith channel of x'_{sem}, *i.e.*, the predicted score map on class i. From our empirical study, the cosine-similarity value X_{ij} indicates the ambiguity between prediction on classes i and j. For example, when X_{ij} is close to 1, pixels are activated similarly in both $x'^{(i,)}_{sem}$ and $x'^{(j,)}_{sem}$, and thus cannot be trusted. Therefore, we use zero bias on class i and use biases in proportional to X_{ij} on class $j \neq i$ equivalent to decrease the confidence scores on class i. Those regions remains high-activated in class i are then confidence enough for the key pooling. The final key pooling is a normalized weighted sum between the weight map $\hat{x}'_{sem}$ and feature map x_f:

$$
v_i = \frac{x_f \cdot \left(\hat{x}'^{(i,)}_{sem}\right)^T}{\left\|\hat{x}'^{(i,)}_{sem}\right\|},\ v = \left[v_i, \cdots\right],\ i = 1, \cdots, c, \tag{3}
$$

where $x_f \in \mathbb{R}^{m \times n}$ denotes the feature map with m channels and n pixels in each channel, $v_i \in \mathbb{R}^{m \times 1}$ denotes the pooled vector for class i, and $v \in \mathbb{R}^{m \times c}$ is the concatenated matrix from v_i.

Attention-based Similarity Checking: We then adopt an attention layer, which uses v as key and x_f as query, and computes the query-key similarity x_{att} through dot-product followed by softmax:

$$
x_{att} = \text{Softmax}(v^T \cdot x_f),\ x_{att} \in \mathbb{R}^{c \times n}. \tag{4}
$$

The x_{att} will activate those text regions that may be ignored due to low-confidence in the initial prediction x'_{sem}. Then, we fuse x_{att} with the input image x and backbone feature x_f into our refined result x_{rfn} through several extra convolutional layers (orange block in Figure 3). Note that our attention layer differs from the traditional query-key-value attention [48] in several ways. Traditional attention requires identical matrix dimensions on query and key, while our approach uses a key v that is significant smaller than the query x_f. Also, traditional attention fuses value and attention through dot product, while ours fuses x_{att} with other features through a deep model. The final output x_{rfn} is supervised by the ground truth as shown in the following.

$$
\mathcal{L}_{rfn} = \text{CrossEntropy}(x_{rfn}, x_{gt}). \tag{5}
$$

3.3. Trimap Loss and Glyph Discriminator

Since human vision is sensitive to text boundaries, segmentation accuracy along the text boundary is of central importance. In addition, text typically has a relatively high contrast between the foreground and background to make it more readable. Therefore, a loss function that focuses on the boundary would further improve the precision of text segmentation. Inspired by [23], we proposed the trimap loss as expressed as follows,

$$\mathcal{L}_{tri} = \text{WCE}\left(x_{rfn}, x_{gt}, w_{tri}\right),$$
$$\text{WCE}(x, y, w) = -\frac{\sum_{j=1}^{n} w_j \sum_{i=1}^{c} x_{i,j} \log(y_{i,j})}{\sum_{j=1}^{n} w_j} \tag{6}$$

where w_{tri} is the binary map with value 1 on text boundaries and 0 elsewhere, and WCE(x, y, w) is cross-entropy between x and y weighted by the spatial map w.

Another unique attribute of text is its readable nature, *i.e.*, the segments of glyphs should be perceptually recognizable. Given that the partial segmentation of a glyph diminishes its readability, we train a glyph discriminator to improve text segments' readability. It is worth noting that the glyph discriminator also improves the evaluation score, as shown in the evaluation. More specifically, we pre-train a classifier for character recognition given the ground-truth character bounding boxes in the training set (the proposed dataset TextSeg provides these annotations). In our case, there are 37 classes, *i.e.*, 26 letters, 10 digits, and misc. During the training of TexRNet, the pre-trained classifier is frozen and applied to the initial prediction x'_{sem}, serving as the glyph discriminator. As illustrated in Figure 3, x'_{sem} is cropped into patches according to the character locations and then fed into the discriminator to obtain the discriminator loss $\mathcal{L}_{dis}$, which indicates whether and how these patches are recognizable.

Unlike $\mathcal{L}_{tri}$ that operates on x_{rfn}, the glyph discriminator is applied on the initial prediction x'_{sem} for mainly two reasons: 1) $\mathcal{L}_{tri}$ focuses on boundary accuracy while $\mathcal{L}_{dis}$ focuses on the body structure of the text, which "distracts" each other if they are applied on the same prediction map. Our empirical studies also show that the improvements from $\mathcal{L}_{tri}$ and $\mathcal{L}_{dis}$ would be diminished if they work together on the same output, which aligns with our analysis. 2) $\mathcal{L}_{tri}$ can directly impact the performance, so it oversees the model's final output x_{rfn}, while $\mathcal{L}_{dis}$ reinforces the deep perception on text thus it can be placed on earlier layers. Above all, the final loss of TexRNet will be

$$\mathcal{L} = \mathcal{L}_{sem} + \alpha\mathcal{L}_{rfn} + \beta\mathcal{L}_{tri} + \gamma\mathcal{L}_{dis}, \tag{7}$$

where α, β, and γ are weights from 0 to 1. In the following experiments, $\alpha = 0.5$, $\beta = 0.5$, and $\gamma = 0.1$. We select these loss weights in the way that the weight sums on two branches are roughly balanced (*i.e.* $0.5 + 0.5 \approx 1 + 0.1$).

4. The New Dataset TextSeg

As text in the real world is extremely diverse, to bridge text segmentation to the real world and accommodate the rapid advances of the text vision research, we propose a new dataset TextSeg, a multi-purpose text dataset focused on but not limited to segmentation.

4.1. Image Collection

The 4,024 images in TextSeg are collected from posters, greeting cards, covers, logos, road signs, billboards, digital designs, handwriting, etc. The diverse image sources could be roughly divided into two text types: 1) scene text, *e.g.*, road signs and billboards, and 2) design text, *e.g.*, artistic text on poster designs. Figure 2 shows examples of the two types. Existing text-related datasets tend to focus on scene text, while TextSeg balances the two text types to achieve a more real-world and diverse dataset. In addition, rather than focusing on text lines, the proposed TextSeg includes a large amount of stylish text. Sharing the language setting from those representative text segmentation datasets, the proposed TextSeg mainly focuses on English (*i.e.*, case-sensitive alphabet, numbers, and punctuation).

4.2. Annotations

TextSeg provides more comprehensive annotations as compared to existing datasets. More specifically, TextSeg has annotated the smallest quadrilateral, pixel-level mask, and transcription for every single word and character. Besides, text effects, *e.g.*, shadow, 3D, halo, etc., are annotated in TextSeg, which distinguishes text from traditional objects and significantly affects text segmentation. To the best of our knowledge, the proposed TextSeg is the only dataset with such comprehensive annotation for text segmentation.

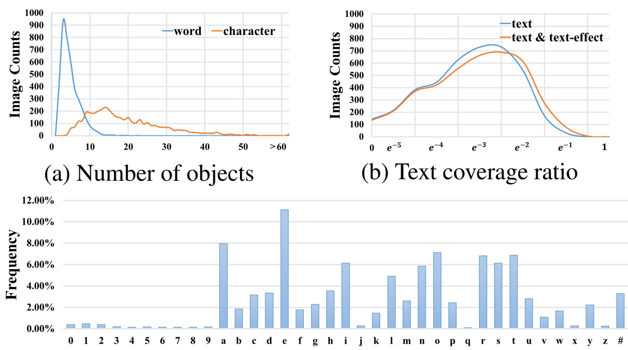
Smallest Quadrilaterals are annotated to tightly bound words, characters, and punctuation. These quadrilaterals are recorded in the image coordinate (*i.e.*, top-left origin, x axis is horizontally right, and y axis is vertically down), and the four vertices are ordered clockwise starting from the top-left corner in the natural reading direction. A smallest quadrilateral tightly bounds a word or character, as shown in Figure 1. In certain cases like blurry text or long strokes, the quadrilaterals would cover the text's core area by ignoring the ambiguous boundary or decorative strokes.

Pixel-level Masks consist of word masks, character masks, and word-effect masks. The word mask is a subset of the word-effect mask since the word mask labels the word surface without the effects like shadow and decoration, while the effect mask covers both word and effects. Similar to word masks, the character masks label character surfaces without those effects. Borrowing the concept from modern segmentation, word masks enable semantic segmentation, and character masks allow instance segmentation. For character masks, the most challenging cases are

Dataset	# Images	Approx. Image Size	Text Type	# Bounding Polygons		Word-level Masks	# Character Masks
Scene Text Segmentation				# Word	# Char		
ICDAR13 FST [26]	462	1000 × 700	Scene	1,944	6,620	Word	4,786
COCO_TS† [5]	14,690	600 × 480	Scene	139,034	–	Word	–
MLT_S† [6]	6,896	1800 × 1400	Scene	30,691	–	Word	–
Total-Text [10]	1,555	800 × 700	Scene	9,330	–	Word	–
TextSeg (Ours)	4,024	1000 × 800	Scene + Design	15,691	73,790	Word, Word-Effect	72,254

† The 14,690 images in COCO_TS is a subset of the totally 53,686 images in COCO-Text [49]. Similarly, the 6,898 images in MLT_S is a subset of the 10,000 images in ICDAR17 MLT [39]. Thus, their word bounding polygons can be directly extracted from their parent datasets.

Table 1: Statistical comparison between TextSeg and other datasets for text segmentation. The "–" marker indicates absence of the corresponding annotation in a dataset.



(a) Number of objects (b) Text coverage ratio

(c) Letter frequency

Figure 4: Statistics of TextSeg. (a) Number of images with different numbers of words and characters. (b) Text coverage ratio against the image. (c) Character frequency of the whole dataset.

handwriting and artistic styles, where there are no clear boundaries between characters. Thus, the criterion is to keep all masks perceptually recognizable.

4.3. Statistical Analysis

Statistical comparison between TextSeg and four representative text segmentation datasets is listed in Table 1, *i.e.*, ICDAR13 FST [26], MLT_S [6], COCO_TS [5], and Total-Text [10]. In general, TextSeg has more diverse text types and all types of annotations. Another dataset that provides character-level annotations is ICDAR13 FST, but its size is far smaller than other datasets. COCO_TS and MLT_S are relatively large, but they lack character-level annotations and mainly focus on scene text. The Total-Text was proposed with similar scope to those existing datasets.

The 4,024 images in TextSeg are split into training, validation, and testing sets with 2,646, 340, and 1,038 images, respectively. In TextSeg and all its splits, the ratio between the number of scene text and design text is roughly 1:1. Figure 4a counts the number of images with different numbers of words and characters, where 12-16 characters and 2-4 words per image is the majority. Figure 4b shows the distribution of the text coverage ratio, where the blue line is set up for word masks and the orange line is for word-effect masks. The rightward shifting from blue to orange indicates the coverage increment due to the word-effect. Finally, Figure 4c displays the character frequency in TextSeg, which roughly aligns with that of English corpus.

4.4. Qualitative Comparison

Figure 5 shows qualitative comparison between TextSeg, ICDAR13 FST, COCO_TS, MLT_S, and Total-Text. ICDAR13 FST has many box-shape masks (considered as ignored characters), which is not a common case in the proposed TextSeg. Other datasets, *i.e.*, COCO_TS, MLT_S, and Total-Text, have only word masks. Note that COCO_TS and MLT_S introduce a large number of ignored areas, especially along text boundaries, which would hinder models from precisely predicting text boundaries. Those boundary-ignored annotations are caused by automatic labeling using weekly supervised models. Similar to TextSeg, Total-Text is labeled manually, but it is of a much smaller size than ours and lacks annotations of characters and text effects.

5. Experimental Evaluation

To demonstrate the effectiveness of the proposed TexRNet, it will be compared to the state-of-the-art methods DeeplabV3+ [8] and HRNet-W48 [50] on five datasets, *i.e.*, ICDAR13 FST [26], COCO_TS [5], MLT_S [6], Total-Text [10], and the proposed TextSeg.

5.1. Experiment Setup

Each model in comparison will be re-trained on each of the aforementioned text segmentation datasets. The models are initialized by ImageNet pretrains and then trained on 4 GPUs in parallel using SGD with weight decay of $5e^{-4}$ for 20,500 iterations. The first 500 iterations are linear warmups [18], and the rest iterations use poly decayed learning rates starting from 0.01 [8]. Note that 5,500 iterations are performed on ICDAR13 FST due to its small size as shown in Table 1. For TextSeg, our model train and evaluate using word masks as foreground instead of the word-effect masks. For the data augmentation, we randomly scale the short side

(a) TextSeg (Ours) (b) ICDAR13 FST (c) COCO_TS (d) MLT_S (e) Total-Text

Figure 5: Comparison of annotations from multiple text segmentation datasets. The proposed TextSeg and ICDAR13 FST [26] provide character-level annotations (color-coded characters). COCO_TS [5], MLT_S [6], and Total-Text [10] only provide word-level annotations, where masks in red and white denote text regions and ignored regions, respectively.

Method	TextSeg (Ours)		ICDAR13 FST		COCO_TS		MLT_S		Total-Text	
	fgIoU	F-score	fgIoU	F-score	fgIoU	F-score	fgIoU	F-score	fgIoU	F-score
PSPNet† [60, 5]	–	–	–	0.797	–	–	–	–	–	0.740
SMANet† [6]	–	–	–	0.785	–	–	–	–	–	0.770
DeeplabV3+ [8]	84.07	0.914	69.27	0.802	72.07	0.641	84.63	0.837	74.44	0.824
HRNetV2-W48 [50]	85.03	0.914	70.98	0.822	68.93	0.629	83.26	0.836	75.29	0.825
HRNetV2-W48 + OCR [59]	85.98	0.918	72.45	0.830	69.54	0.627	83.49	0.838	76.23	0.832
Ours: TexRNet + DeeplabV3+	86.06	0.921	72.16	0.835	**73.98**	**0.722**	**86.31**	0.860	76.53	0.844
Ours: TexRNet + HRNetV2-W48	**86.84**	**0.924**	**73.38**	**0.850**	72.39	0.720	86.09	**0.865**	**78.47**	**0.848**

† In [5, 6], the author augmented the original training dataset with SynthText [19] in both ICDAR13 FST and Total-Text experiments.

Table 2: Performance comparison between TexRNet and other models on TextSeg and other representative text segmentation datasets. The bold numbers indicate the best results.

of the input images from 513 to 1025 and randomly crop a 513×513 patch as input in training.

The glyph discriminator in TexRNet adopts a ResNet50 classifier [22], which is trained on character patches from TextSeg training and validation sets. It achieves the classification accuracy of 93.38% on the TextSeg testing set. Since only the proposed TextSeg and ICDAR13 FST provide character bounding boxes, the glyph discriminator is only applied on these two datasets and disabled on COCO_TS, MLT_S, and Total-Text.

We evaluate our models using multi-scale no-flip ensemble 8 scales from 0.75x to 2.5x of a standard 513 short side image. To align with modern segmentation tasks, we use foreground Intersection-over-Union (fgIoU) as our major metric. Also, the typical F-score measurement on foreground pixels is provided in the same fashion as [11, 26]. The foreground here indicates the text region in both prediction and ground truth.

5.2. Model Performance

This section compares TexRNet to other text and semantic segmentation methods. To demonstrate the effectiveness of TexRNet, the comparison is conducted on five datasets including our TextSeg. As previously claimed, we adopt DeeplabV3+ [8] and HRNetV2-W48 [50] as our backbone and baseline. We also compares with the SOTA semantic segmentation model: HRNetV2-W48 + Object-Contextual Representations (OCR) [59]. The PSPNet and SMANet results are from [5, 6] in which their models were trained on ICDAR13 FST and Total-Text augmented with SynthText [19]. Tables 2 shows the overall results. As the table shows, our proposed TexRNet outperforms other methods on all datasets.

5.3. Ablation Studies

This section performs ablation studies on the key pooling and attention (the yellow block in Figure 3), trimap loss, and glyph discriminator in the proposed TexRNet. In this experiment, DeeplabV3+ is adopted as the backbone, and the models are trained and evaluated on TextSeg. Starting from the base version of TexRNet, the key pooling and attention (Att.), trimap loss ($\mathcal{L}_{tri}$), and glyph discriminator ($\mathcal{L}_{dis}$) are added incrementally as shown in Table 3, where the fgIoU and F-score are reported, presenting a consistently increasing trend. The final TexRNet achieves the best performance, around 2% increase in fgIoU as compared to DeeplabV3+.

An interesting observation is that TexRNet (final) have exactly the same number of parameters as TexRNet (base), but the part between them contributes the most improvement. To further investigate whether the performance increase comes from parameter increase, we compared TexRNet with HRNetV2-W48+OCR and other models in Figure 6. We discover that TexRNet achieves higher accuracy with less parameters as compared to HRNetV2-W48+OCR,

Method	Att.	$\mathcal{L}_{tri}$	$\mathcal{L}_{dis}$	fgIoU	F-score
DeeplabV3+				84.07	0.914
TexRNet (base)				84.86	0.917
TexRNet	✓			85.36	0.919
TexRNet	✓	✓		85.55	0.921
TexRNet (final)	✓	✓	✓	**86.06**	**0.921**

Table 3: Ablation studies of TexRNet on TextSeg. All models are training on TextSeg train and validation sets, and all TexRNet use DeeplabV3+ as backbone. The column "Attn." represents whether attention layers are included. Similarly, columns "$\mathcal{L}_{tri}$" and "$\mathcal{L}_{dis}$" indicate whether the trimap loss and glyph discriminator are used.

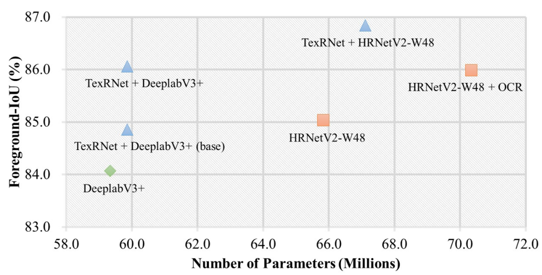


Figure 6: Comparison of different methods in the number of parameter vs. text segmentation performance in fgIoU.

demonstrating the effectiveness of our design in TexRNet.

5.4. Downstream applications

This section gives prospects of TexRNet and TextSeg dataset, especially in driving downstream applications.

Text Removal is a practical problem in photo and video editing, and it is also an application with high industrial demand. For example, media service providers frequently need to erase brands from their videos to avoid legal issues. Since this task is a hole filling problem, Deep Image Prior [47] is employed, and different types of text masks are provided to compare the performance of text removal. Typically, word or character bounding boxes are standard text masks because they are easy to get from existing text detection methods. By contrast, the proposed TexRNet provides much more accurate text masks. Figure 7 compares the results using these three types of text masks, *i.e.*, text segmentation mask, character bounding polygon, and word bounding polygon. Obviously, finer mask yields to better performance.

Text Style Transfer is another popular task for both research and industry. Mostly, text style transfer relies on accurate text masks. In this experiment, we use Shape-Matching GAN [56] as our downstream method, which requires text masks as an input. In their paper, all demo images are generated using ground truth text masks, which may be impractical in real-world applications. Therefore, we extend TexRNet with Shape-Matching GAN to achieve scene text style transfer on an arbitrary text image. A few examples are visualized in Figure 8, and more examples can be found in the supplementary.

Figure 7: Examples of text removal with different types of text masks. From left to right, the top row shows the input image, predicted text mask from our TexRNet, character bounding polygons, and word bounding polygons from ground truth. The second row are text removing results using corresponding text masks on the same column.

Figure 8: Examples of text style transfer with styles of fire and maple on the first column. The rest are results with their original images attached to the bottom-left corner.

6. Conclusions

We introduce a novel text segmentation dataset TextSeg, which consists of 4,024 scene text and design text images with comprehensive annotations including word- and character-wise bounding polygons, masks, and transcriptions. We also propose a new and effective text segmentation method, TexRNet. We demonstrate that our model outperforms state-of-the-art semantic segmentation models on TextSeg and another four datasets. To support our idea that text segmentation has great potential in the industry, we introduce two downstream applications, *i.e.*, text removal and text style transfer, to show promising results using text segmentation masks from TexRNet. In conclusion, text segmentation is a critical task. We hope that our new dataset and method would become the corner-stone for future text segmentation research.

References

[1] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.

[2] Samaneh Azadi, Matthew Fisher, Vladimir G Kim, Zhaowen Wang, Eli Shechtman, and Trevor Darrell. Multi-content gan for few-shot font style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7564–7573, 2018.

[3] Bo Bai, Fei Yin, and Cheng Lin Liu. A seed-based segmentation method for scene text extraction. In *2014 11th IAPR International Workshop on Document Analysis Systems*, pages 262–266. IEEE, 2014.

[4] Min Bai and Raquel Urtasun. Deep watershed transform for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[5] Simone Bonechi, Paolo Andreini, Monica Bianchini, and Franco Scarselli. Coco_ts dataset: Pixel–level annotations based on weak supervision for scene text segmentation. In *International Conference on Artificial Neural Networks*, pages 238–250. Springer, 2019.

[6] Simone Bonechi, Monica Bianchini, Franco Scarselli, and Paolo Andreini. Weak supervision for generating pixel–level annotations in scene text segmentation. *Pattern Recognition Letters*, 138:1–7, 2020.

[7] Liang-Chieh Chen, George Papandreou, Florian Schroff, and Hartwig Adam. Rethinking atrous convolution for semantic image segmentation. *arXiv preprint arXiv:1706.05587*, 2017.

[8] Liang-Chieh Chen, Yukun Zhu, George Papandreou, Florian Schroff, and Hartwig Adam. Encoder-decoder with atrous separable convolution for semantic image segmentation. In *European Conference on Computer Vision*, 2018.

[9] Yu Cheng, Zhe Gan, Yitong Li, Jingjing Liu, and Jianfeng Gao. Sequential attention gan for interactive image editing. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 4383–4391, 2020.

[10] Chee Kheng Ch'ng and Chee Seng Chan. Total-text: A comprehensive dataset for scene text detection and recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 935–942. IEEE, 2017.

[11] Antonio Clavelli, Dimosthenis Karatzas, and Josep Llados. A framework for the assessment of text extraction algorithms on complex colour images. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 19–26, 2010.

[12] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.

[13] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[14] Julia Diaz-Escobar and Vitaly Kober. Natural scene text detection and segmentation using phase-based regions and character retrieval. *Mathematical Problems in Engineering*, 2020, 2020.

[15] Mark Everingham, SM Ali Eslami, Luc Van Gool, Christopher KI Williams, John Winn, and Andrew Zisserman. The pascal visual object classes challenge: A retrospective. *International journal of computer vision*, 111(1):98–136, 2015.

[16] Jun Fu, Jing Liu, Haijie Tian, Yong Li, Yongjun Bao, Zhiwei Fang, and Hanqing Lu. Dual attention network for scene segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3146–3154, 2019.

[17] Naiyu Gao, Yanhu Shan, Yupei Wang, Xin Zhao, Yinan Yu, Ming Yang, and Kaiqi Huang. Ssap: Single-shot instance segmentation with affinity pyramid. In *Proceedings of the IEEE International Conference on Computer Vision*, 2019.

[18] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch sgd: Training imagenet in 1 hour. *arXiv preprint arXiv:1706.02677*, 2017.

[19] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman. Synthetic data for text localisation in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2315–2324, 2016.

[20] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Simultaneous detection and segmentation. In *European Conference on Computer Vision*, 2014.

[21] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE International Conference on Computer Vision*, 2017.

[22] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016.

[23] Yu-Hui Huang, Xu Jia, Stamatios Georgoulis, Tinne Tuytelaars, and Luc Van Gool. Error correction for dense semantic image labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 998–1006, 2018.

[24] 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.

[25] 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 *2015 13th International Conference on Document Analysis and Recognition*, pages 1156–1160. IEEE, 2015.

[26] 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. IEEE, 2013.

[27] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic feature pyramid networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[28] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems*, 2012.

[29] John Lafferty, Andrew McCallum, and Fernando CN Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. 2001.

[30] Yann LeCun, Corinna Cortes, and CJ Burges. Mnist handwritten digit database. 2010.

[31] Stan Z Li. *Markov random field modeling in image analysis*. Springer Science & Business Media, 2009.

[32] Yi Li, Haozhi Qi, Jifeng Dai, Xiangyang Ji, and Yichen Wei. Fully convolutional instance-aware semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[33] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2117–2125, 2017.

[34] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision*, pages 740–755. Springer, 2014.

[35] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018.

[36] Xiaoqing Liu and Jagath Samarabandu. Multiscale edge-based text extraction from complex images. In *2006 IEEE International Conference on Multimedia and Expo*, pages 1721–1724. IEEE, 2006.

[37] Yiding Liu, Siyu Yang, Bin Li, Wengang Zhou, Jizheng Xu, Houqiang Li, and Yan Lu. Affinity derivation and graph merge for instance segmentation. In *European Conference on Computer Vision*, 2018.

[38] Anand Mishra, Karteek Alahari, and CV Jawahar. An mrf model for binarization of natural scene text. In *2011 International Conference on Document Analysis and Recognition*, pages 11–16. IEEE, 2011.

[39] 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 *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1454–1459. IEEE, 2017.

[40] Davy Neven, Bert De Brabandere, Marc Proesmans, and Luc Van Gool. Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[41] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[42] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[43] 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.

[44] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[45] Bolan Su, Shijian Lu, and Chew Lim Tan. Binarization of historical document images using the local maximum and minimum. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 159–166, 2010.

[46] Youbao Tang and Xiangqian Wu. Scene text detection and segmentation based on cascaded convolution neural networks. *IEEE Transactions on Image Processing*, 26(3):1509–1520, 2017.

[47] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Deep image prior. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9446–9454, 2018.

[48] 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.

[49] Andreas Veit, Tomas Matera, Lukas Neumann, Jiri Matas, and Serge Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. *arXiv preprint arXiv:1601.07140*, 2016.

[50] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[51] Xiaolong Wang, Ross Girshick, Abhinav Gupta, and Kaiming He. Non-local neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7794–7803, 2018.

[52] Xiufei Wang, Lei Huang, and Changping Liu. A novel method for embedded text segmentation based on stroke and color. In *2011 International Conference on Document Analysis and Recognition*, pages 151–155. IEEE, 2011.

[53] Yuwen Xiong, Renjie Liao, Hengshuang Zhao, Rui Hu, Min Bai, Ersin Yumer, and Raquel Urtasun. Upsnet: A unified panoptic segmentation network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

[54] Xingqian Xu, Mang Tik Chiu, Thomas S Huang, and Honghui Shi. Deep affinity net: Instance segmentation via affinity. *arXiv preprint arXiv:2003.06849*, 2020.

[55] Shuai Yang, Jiaying Liu, Zhouhui Lian, and Zongming Guo. Awesome typography: Statistics-based text effects transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7464–7473, 2017.

[56] Shuai Yang, Zhangyang Wang, Zhaowen Wang, Ning Xu, Jiaying Liu, and Zongming Guo. Controllable artistic text style transfer via shape-matching gan. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4442–4451, 2019.

[57] Tien-Ju Yang, Maxwell D Collins, Yukun Zhu, Jyh-Jing Hwang, Ting Liu, Xiao Zhang, Vivienne Sze, George Papandreou, and Liang-Chieh Chen. Deeperlab: Single-shot image parser. *arXiv preprint arXiv:1902.05093*, 2019.

[58] Cong Yao, Xiang Bai, Wenyu Liu, Yi Ma, and Zhuowen Tu. Detecting texts of arbitrary orientations in natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1083–1090. IEEE, 2012.

[59] Yuhui Yuan, Xilin Chen, and Jingdong Wang. Object-contextual representations for semantic segmentation. *arXiv preprint arXiv:1909.11065*, 2019.

[60] Hengshuang Zhao, Jianping Shi, Xiaojuan Qi, Xiaogang Wang, and Jiaya Jia. Pyramid scene parsing network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2881–2890, 2017.

[61] Bolei Zhou, Hang Zhao, Xavier Puig, Sanja Fidler, Adela Barriuso, and Antonio Torralba. Scene parsing through ade20k dataset. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.