Yuan TL, Zhu Z, Xu K *et al.* A large Chinese text dataset in the wild. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 34(3): 509–521 May 2019. DOI 10.1007/s11390-019-1923-y

A Large Chinese Text Dataset in the Wild

Tai-Ling Yuan[1], Zhe Zhu[2], Kun Xu[1], *Member*, *CCF*, *IEEE*, Cheng-Jun Li[3], Tai-Jiang Mu[1,*], *Member*, *CCF* and Shi-Min Hu[1], *Fellow*, *CCF*, *Senior Member*, *IEEE*

[1] *Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China*

[2] *Department of Radiology, Duke University, North Carolina 27708, U.S.A.*

[3] *Tencent Technology (Beijing) Co. Ltd., Beijing 100080, China*

E-mail: {yuantailing, ajex1988}@gmail.com; xukun@tsinghua.edu.cn; chengjunli@tencent.com
mmmutj@gmail.com; shimin@tsinghua.edu.cn

Received December 24, 2018; revised March 20, 2019.

Abstract In this paper, we introduce a very large Chinese text dataset in the wild. While optical character recognition (OCR) in document images is well studied and many commercial tools are available, the detection and recognition of text in natural images is still a challenging problem, especially for some more complicated character sets such as Chinese text. Lack of training data has always been a problem, especially for deep learning methods which require massive training data. In this paper, we provide details of a newly created dataset of Chinese text with about 1 million Chinese characters from 3 850 unique ones annotated by experts in over 30 000 street view images. This is a challenging dataset with good diversity containing planar text, raised text, text under poor illumination, distant text, partially occluded text, etc. For each character, the annotation includes its underlying character, bounding box, and six attributes. The attributes indicate the character's background complexity, appearance, style, etc. Besides the dataset, we give baseline results using state-of-the-art methods for three tasks: character recognition (top-1 accuracy of 80.5%), character detection (AP of 70.9%), and text line detection (AED of 22.1). The dataset, source code, and trained models are publicly available.

Keywords Chinese text dataset, Chinese text detection, Chinese text recognition

1 Introduction

Automatic text detection and recognition is an important task in computer vision, with applications ranging from autonomous driving to book digitization. This problem has been extensively studied, and has been divided into two scenarios at different levels of difficulty, i.e., text detection and recognition in document images, and in natural images. The former is less challenging and many commercial tools are already available. However, the latter is still challenging. For example, a character may have very different appearances in different images due to style, font, resolution, or illumination differences; characters may also be partially occluded, distorted, or have complex background, which makes detection and recognition even harder. Sometimes we even have to deal with high intra-class versus low inter-class differences[1]. As shown in Fig.1, the three characters differ a little, but the instances of the same character could have large appearance differences.

The past few years have witnessed a boom of deep learning in many fields, including image classification, speech recognition, machine translation, and so on. Very deep networks with tens of or even more than a hundred layers (such as VGG-19, Google Inception or ResNet) have nice modeling capacity, and have shown promising performance in both detection and classification tasks. These models usually require massive amount of data for training. Availability of data is,

Regular Paper

Special Section of CVM 2019

This work was supported by the National Natural Science Foundation of China under Grant Nos. 61822204 and 61521002, a research grant from the Beijing Higher Institution Engineering Research Center, and the Tsinghua-Tencent Joint Laboratory for Internet Innovation Technology.

*Corresponding Author

indeed, a key factor in the success of deep neural networks. The public datasets, such as the Image-Net[2], the Microsoft COCO[3] and the ADE20K[4], have become a key driver for progress in computer vision.

Fig.1. High intra-class variance versus low inter-class variance. Each row shows instances of a Chinese character (rightmost). The first character differs from the second character by a single stroke, and the second character differs from the third character by another small stroke. While the three characters are very similar in shape, the instances of the same character have very different appearances, due to color, font, occlusion, background differences, etc.

In this paper, we present a large dataset of Chinese text in natural images, named Chinese Text in the Wild (CTW). The dataset contains 32 285 images with 1 018 402 Chinese characters for 3 850 distinct ones, going much beyond previous datasets. The images are from Tencent Street View. They are captured from tens of different cities in China, without preference for any particular purpose. The dataset is a challenging dataset, due to its diversity and complexity. It contains planar text, raised text, text in urban areas, text in rural areas, text under poor illumination, distant text, partially occluded text, etc. Within each image, all Chinese texts are annotated. For each Chinese character, we annotate their underlying character, bounding box, and six attributes to indicate whether it is occluded, having complex background, distorted, 3D raised, wordart, and handwritten, respectively.

Based on this character dataset, we train deep models using several state-of-the-art approaches for character recognition and detection in natural images. These models are also presented as baseline algorithms. The dataset, source code, and trained models are publicly available①. We expect the dataset to greatly stimulate the future development of detection and recognition algorithms of Chinese texts in natural images.

The rest of this paper is organized as follows. We discuss related work in Section 2, and give details of our dataset in Section 3. The baseline algorithms trained using our dataset and experimental results are presented in Section 4, and conclusions are made in Section 5.

2 Related Work

Text detection and recognition has received much attention during the past decades in the computer vision community, albeit mostly for English text and digits. We have briefly reviewed both benchmark datasets and approaches in recent years. Here, we treat text recognition in documents as a separate well-studied problem, and only focus on text detection and recognition in natural images.

2.1 Datasets of Text in Natural Images

Datasets of text in natural images could be classified into two categories: those that only contain real-world text[5−8], and those that contain synthetic text[9,10]. Images in these datasets are mainly of two kinds: Internet images, and Google Street View images. For example, the SVHN[11] and SVT[12] datasets utilize Google Street View images, augmented by annotations for digits and text, respectively. Most previous datasets target English text in the Roman alphabet, and digits, although several recent datasets consider text in other languages and character sets. Notable amongst them are the KAIST scene text dataset[13] for Korean text, FSNS[7] for French text, and MSRA-TD500[14] and RCTW-17[15] for Chinese text. However, MSRA-TD500 dataset[14] only contains 500 natural images, which is far from sufficient for training deep models such as convolutional neural networks. RCTW-17[15] dataset contains 12 263 images, but their texts are annotated by lines, which is unfriendly to some end-to-end algorithms. In contrast, our dataset contains over 30 000 images and about 1 million Chinese characters from 3 850 unique ones, and the text lines have been further segmented into separated characters.

2.2 Text Detection and Recognition

Text detection and recognition approaches can be classified with regard to the feature adopted, i.e., hand-crafted or automatically learned (as deep learning does). We draw a distinction between text detection, detecting a region of an image that (potentially) contains text, and text recognition, determining which characters and text are present, typically using the cropped areas returned by text detection.

Text Detection. The most widely used approach to text detection based on hand-crafted features is the stroke width transform (SWT)[16]. SWT transforms an

①https://ctwdataset.github.io/, March 2019.

image into a new stroke-width image with an equal size, in which the value of each pixel is the stroke width associated with the original pixel. This approach works quite well for relatively clean images containing English characters and digits, but often fails on more cluttered images. Another widely used approach is to seek text as maximally stable extremal regions (MSERs)[17−20]. Such MSERs always contain non-text regions; thus a robust filter is needed for candidate text region selection. Recently, deep learning based approaches have been adopted for text detection, including fully convolutional networks (FCN)[21,22], cascaded convolutional text networks (CCTN)[23], and connectionist text proposal network (CTPN)[24].

Text Recognition. Given cropped text, recognition methods for general objects can be adapted to text recognition. Characters and words are at two different levels in English, and different approaches have been proposed for character recognition and word recognition separately. For character recognition, both SVM-based approaches[25] and part-based models[26] have been applied and found to work well. Word recognition provides additional contextual information; therefore Bayesian inferencing[27], conditional random fields (CRFs)[28] and graph models[29] can be used.

A recent trend is to focus on "end-to-end" recognition[30−32]; more can be found in a detailed survey by Ye and Doermann[33].

3 Chinese Text in the Wild Dataset

In this section, we present Chinese Text in the Wild (CTW), a very large dataset of Chinese text in street view images. We will discuss how the images are selected, annotated, and split into training and testing sets, and we also provide statistics of the dataset. For denotation clearness, we refer to each unique Chinese character as a character category or a category, and refer to an observed instance of a Chinese character in an image as a character instance, or an instance.

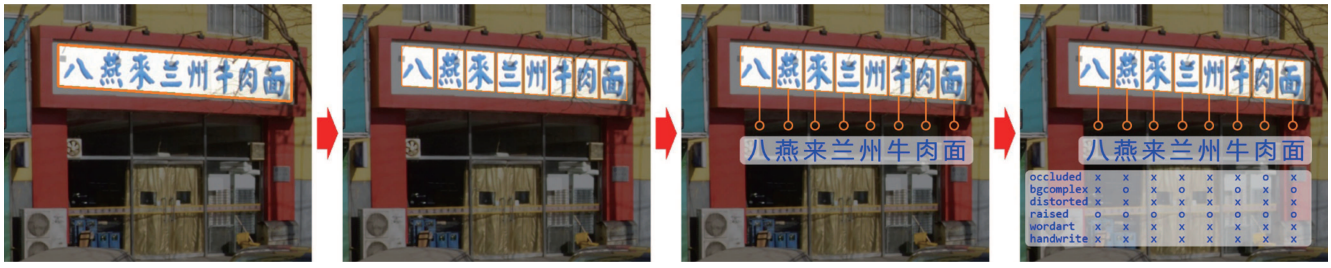
3.1 Image Selection

We have collected 122 903 street view images from Tencent Street View. Among them, 98 903 images are from the Tsinghua-Tencent 100K dataset[34], and 24 000 directly from Tencent Street View. These images are captured from tens of different cities in China, and each image has a resolution of 2 048 × 2 048. We manually check all street view images, and remove those images which do not contain any Chinese characters. Besides, since the street view images are captured at fixed distance intervals (i.e., 10–20 meters), successive images may have large duplicated areas. Hence, we manually check each pair of successive images, and if duplicated areas cover more than 70% of the total image size, we also remove one image from the pair. Finally, 32 285 images are selected.

3.2 Annotation

We now describe the annotation process in detail. For each image, only all Chinese character instances are annotated. Our annotation pipeline is illustrated in Fig.2. A bounding box is first drawn around a sentence of Chinese text. Next, for each character instance, a tighter bounding box is drawn around it, and its corresponding category and its attributes are also specified. Additionally, illegible text regions are annotated as "ignore" regions. Examples are illustrated in Fig.3.

There are six attributes to annotate, which are occlusion attribute, complex background attribute, distortion attribute, raised attribute, wordart attribute, and handwritten attribute. For each character instance, yes or no is specified for each attribute. The occlusion attribute indicates whether the character is occluded,



(a) (b) (c) (d)

Fig.2. Annotation pipeline: (a) drawing a bounding box for a line of texts, (b) drawing a bounding box for each character instance, (c) labeling its corresponding character category, and (d) labeling its attributes.

partially occluded by other objects or not. The complex background attribute indicates whether the character has complex background, shadows on it or not. The distortion attribute indicates whether the character is distorted, rotated or frontal. The raised attribute indicates whether the character is 3D raised or planar. The wordart attribute indicates whether the character uses an artistic style or uses a traditional font. The handwritten attribute indicates whether the character is handwritten or printed. Character examples of each attribute are illustrated in Fig.4. We provide these attributes since the texts have large appearance variations due to color, font, occlusion, background differences, etc. With the help of these attributes, it will be easier to analyze the algorithm performance on different styles of texts. Researchers may also design algorithms for specific styles of Chinese texts, e.g., 3D raised texts.

Fig.3. Two cropped images in our dataset with corresponding annotation. Character instances are shown in green while "ignore" regions are shown in yellow.

Fig.4. Examples with different attributes. (a) Occluded. (b) Not occluded. (c) Complex background. (d) Clean background. (e) Distorted. (f) Frontal. (g) 3D raised. (h) Planar. (i) Wordart. (j) Not wordart. (k) Handwritten. (l) Printed.

To guarantee annotation of high quality, we recruit 40 annotation experts, all employed by a professional image annotation company and well trained for image annotations tasks. We also invite two inspectors to verify the quality of annotations. Before annotating, we first invite them to take a training session following annotation instructions. The whole annotation process takes about two months. In total, 1 018 402 Chinese character instances are annotated. Fig.3 shows two cropped images in our dataset with corresponding annotation.

3.3 Dataset Splitting

We split our dataset to a training set and a testing set. The testing set is further split into a recognition testing set for the recognition task (Subsection 4.1) and a detection testing set for the detection task (Subsection 4.2). The ratio of the sizes of the three sets is set to 8 : 1 : 1, and we randomly distribute all the images into the three sets according to this ratio. To avoid the correlation between training and testing images, we constrain that the images captured on the same street must be in the same set. Finally, the training set contains 25 887 images with 812 872 Chinese characters, the recognition testing set contains 3 269 images with 103 519 Chinese characters, and the detection testing set contains 3 129 images with 102 011 Chinese characters.

3.4 Statistics

Our CTW dataset contains 32 285 images with 1 018 402 Chinese character instances, 98.5% of which are commonly used characters in GB2312②. It contains 3 850 character categories (i.e., unique Chinese characters), 2 913 of which cover 77.6% of the commonly used characters in GB2312.

Fig.5 shows the frequencies for the top 50 most frequently observed character categories in the training set and in the testing set, respectively. Fig.6(a) shows the number of images containing a specific number of character instances in the training set and in the testing set, respectively. Fig.6(b) shows the number of images containing a specific number of character categories in the training set and in the testing set, respectively. In Fig.7(a), we provide the number of character instances with different sizes in the training set and in the testing set, respectively, where the size is measured by the long side of its bounding box in pixels. In Fig.7(b), we provide the percentage of character instances with different attributes in all, large, medium, and small character instances, respectively. Small, medium, and large refer to the character size less than ($<$) 16, ranged in ($\in$) [16, 32) and no less than ($\geqslant$) 32, respectively. We could find that large character instances are more likely to have complex attributes. For example, in all character instances, 13.2% of them are occluded, while in all large character instances, a higher proportion (19.2%) of them are occluded.

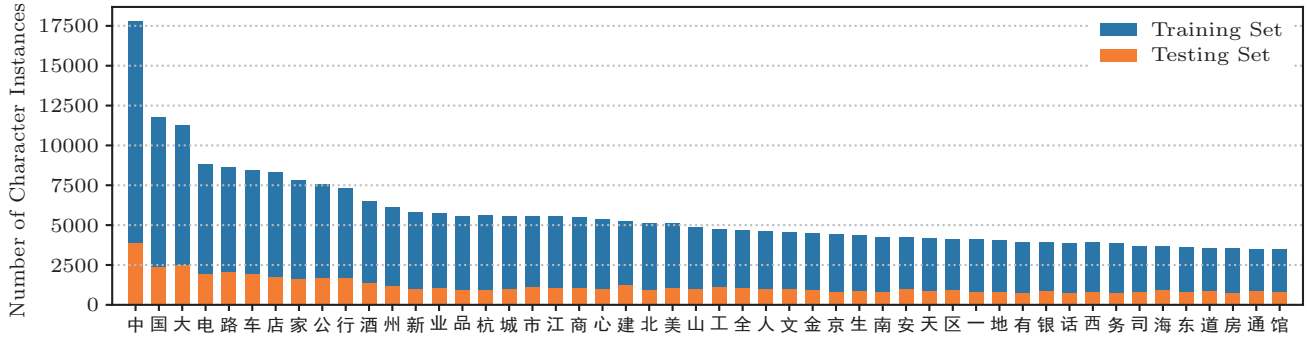


Fig.5. Number of character instances for the 50 most frequently observed character categories in our dataset.

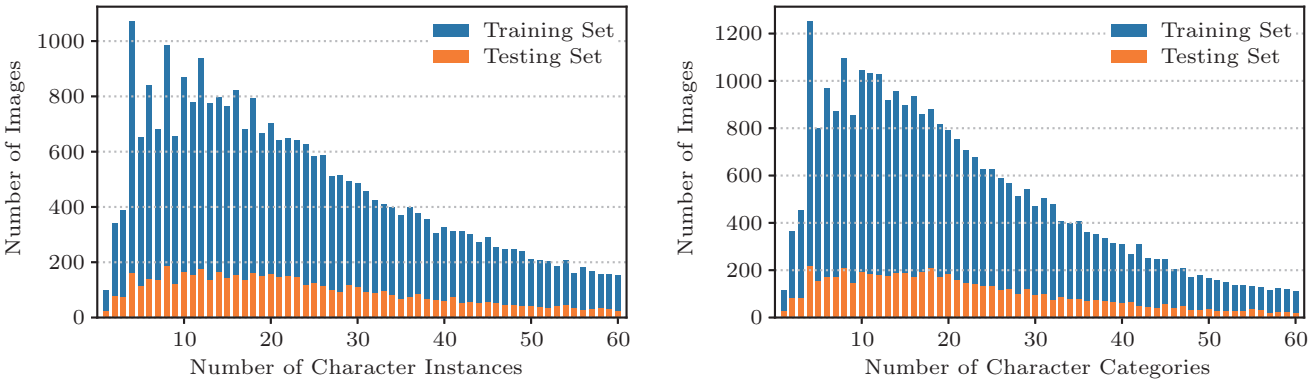


Fig.6. Statistics for the number of character instances and categories in the dataset. (a) Number of images containing a specific number of character instances. (b) Number of images containing a specific number of character categories.

Diversity. As shown in Fig.7(b), 13.2% character instances are occluded, 28.0% have complex background, 26.0% are distorted, and 26.9% are raised text. Also note that only 36.0% of the character instances are normal (i.e., all the attributes are "no"). As shown in Fig.8, our dataset contains planar text, raised text, text in urban areas, text in rural areas, vertical text, distant text, nearby text, text under poor illumination, and partially occluded text. The above statistics show that our dataset has good diversity on character cate-

②http://www.moe.gov.cn/s78/A19/yxs_left/moe_810/s230/201206/t20120601_136847.html, March 2019.

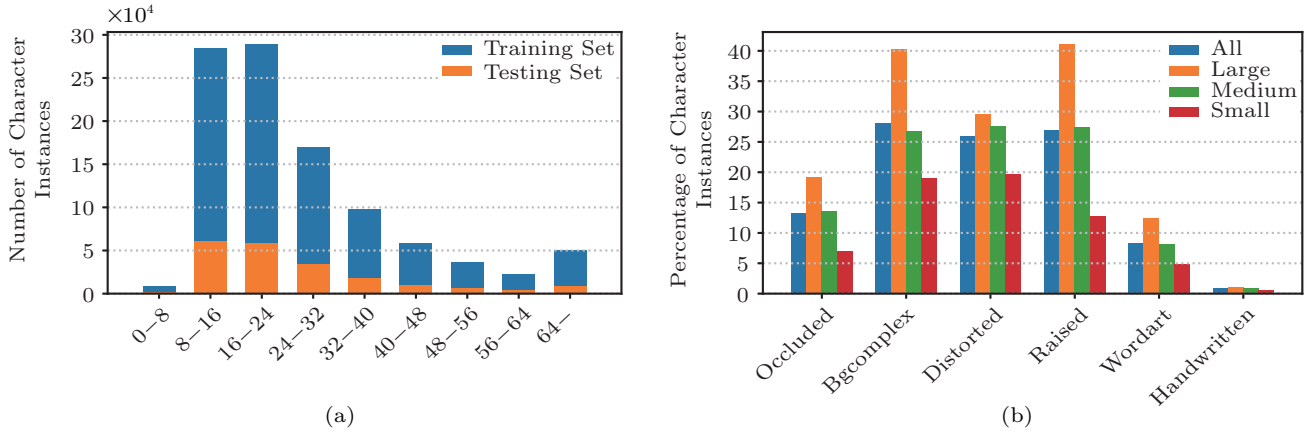


Fig.7. Statistics for the size of character instances in the dataset. (a) Number of character instances with different sizes. The size is measured by the long side of its bounding box in pixels. (b) Percentage of character instances with different attributes in all, large, medium, and small character instances, respectively. Small, medium, and large refer to character size < 16, $\in [16, 32)$ and $\geqslant 32$, respectively.

gories, character sizes, and attributes (i.e., occlusion, background complexity, 3D raised, etc.). Due to such diversity and complexity, our CTW dataset is a challenging Chinese character dataset for real images.

4 Baseline Algorithms and Performance

We now describe the baseline algorithms and their performance using the proposed CTW dataset. Our experiments are performed on a desktop with a 3.5 GHz Intel Core i7-5930K CPU, NVIDIA GTX TITAN GPU and 32 GB RAM.

We consider three tasks: character recognition from cropped regions, character detection from images, and text line detection from images.

4.1 Recognition

Given a cropped region showing a Chinese character instance, the goal of the character recognition task is to predict its character category. We adopt top-1 accuracy as the primary metric. Only Chinese character instances are considered in the evaluation, i.e., no digits or English characters in the ground truth set.

We test several state-of-the-art convolutional neural network structures for the recognition task using TensorFlow, including AlexNet[35], OverFeat[36], Google Inception[37], 50 layer ResNet[38] (ResNet50) and 152 layer ResNet (ResNet152). We use the training set and the recognition testing set as described in Subsection 3.3 for training and testing, respectively. Since a majority of the character categories are rarely-used Chinese characters, which have very few samples in the training data and also have very rare usage in practice, we only consider the recognition of the top 1 000 frequently observed character categories while we regard the other 2 850 character categories as one category. This makes our recognition task as a classification problem of 1 001 categories: the used 1 000 character categories plus an "others" category. We train each network using tens of thousands of iterations, and the parameters of each model are finely tuned. On the testing set, the top-1 accuracies achieved by these networks are AlexNet (73.0%), OverFeat (76.0%), Google Inception (80.5%), ResNet50 (78.2%) and ResNet152 (79.0%), respectively. In Table 1, we also give the top-1 accuracies of the top 10 frequent observed character categories and the highest accuracy for each category is indicated in bold. In Table 2, we show 20 character instances randomly chosen from the testing set. In each row, from left to right, we show the cropped region of a character instance, the ground truth character category, and the recognition results of different methods. Among the above methods, Google Inception achieves the highest accuracy rate.

In Fig.9, we provide the top-1 accuracy using Google Inception for character instances with different attributes and different sizes, respectively. The results are consistent with our intuition, e.g., characters with clean backgrounds, printed characters, and large characters are easier to be recognized than those with complex background, handwritten characters, and small characters, respectively. An interesting observation is that the recognition accuracy of large wordart characters (70.0%) is lower than the accuracy of medium wardart characters (72.3%). The reason is that large characters

Fig. 8. Dataset diversity. (a) Planar text. (b) Raised text. (c) Text in urban areas. (d) Text in rural areas. (e) Vertical text. (f) Distant text. (g) Nearby text. (h) Text under poor illumination. (i) Partially occluded text.

Table 1. Top-1 Accuracies (%) of the 10 Most Frequent Character Categories

Method	中	国	大	电	路	车	店	家	公	行	All
AlexNet	79.8	66.2	78.4	83.6	87.3	82.7	79.9	78.9	80.4	84.1	73.0
OverFeat	82.7	69.5	84.0	87.2	89.0	86.3	83.4	83.6	82.0	87.1	76.0
Google Inception	**88.9**	**74.6**	**88.1**	**90.9**	**91.2**	**89.2**	**90.3**	**88.4**	**87.8**	**90.6**	**80.5**
ResNet50	86.4	72.6	84.0	89.1	90.3	87.1	86.5	84.7	84.1	87.5	78.2
ResNet152	87.4	73.0	85.5	89.3	91.0	87.6	87.1	86.8	84.3	88.4	79.0

Table 2. Some Examples of the Recognition Task

Instance	Category	AlexNet	OverFeat	Google Inception	ResNet50	ResNet152
	华	华 33.7	货 52.8	货 42.2	华 46.7	货 30.1
	务	类 43.6	务 36.8	务 70.3	务 73.7	务 71.9
	永	永 100.0	永 100.0	永 100.0	永 99.7	永 99.8
	店	质 6.2	质 36.3	店 81.0	店 81.6	店 86.2
	同	月 28.4	列 33.4	同 95.8	同 72.5	同 36.3
	華	疗 2.9	麻 6.9	華 98.1	華 94.5	華 74.0
	收	收 34.6	收 64.0	收 99.9	收 93.7	收 95.8
	创	创 97.8	创 99.7	创 99.4	创 99.3	创 99.2
	动	动 99.1	动 99.9	动 99.3	动 99.8	动 96.5
	好	的 11.5	好 45.0	好 95.4	妇 40.8	好 62.3

Note: In each row, from left to right, we give the cropped region of a character instance, the ground truth character category, and the recognition results of different methods. Corrected recognitions are painted in green. The number shows the confidence (%) of the results.

are more likely to be occluded or have a complex background (as shown in Fig.7(b)), making them harder to be recognized. More details can be found in the supplemental material③.

4.2 Character Detection

Given an image, the goal of the character detection task is to detect the bounding boxes of all character instances and predict their character categories at the same time. Since Chinese greatly differs from English: Chinese text line consists of characters with no additional space between two words, and many individual characters themselves are words, we evaluate different methods on character level rather than on word level. We adopt average precision (AP), which follows PASCAL VOC[39], as the primary metric. A detection is considered as true positive when the detected region and the ground truth region have an IoU over 0.5 and also have an identical character category. A detection which does not match a ground truth but matches an "ignore" region is excluded during the evaluation. Only Chinese characters are considered in the evaluation.

We run two methods for the detection task, i.e., YOLOv2[40] and SSD512[41]. Given an image, the output of YOLOv2 is a list of detected character instances,

③The supplemental material is available at https://ctwdataset.github.io/, March 2019.

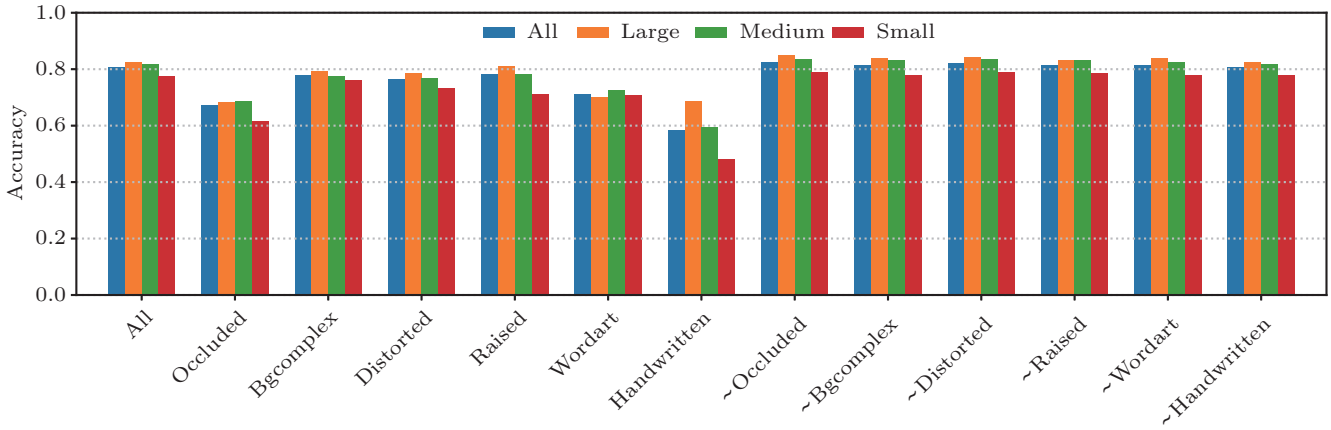


Fig.9. Top-1 accuracy using Google Inception for character instances with different attributes and different sizes. Small, medium, and large refer to character size < 16, $\in [16, 32)$ and $\geqslant 32$, respectively. $\sim$ denotes "without a specific attribute", e.g., $\sim$ occluded means "not occluded character instances".

each of which is associated with a character category, a bounding box, and a confidence score in $[0, 1]$. Same as in the recognition task (Subsection 4.1), we set the number of categories to 1 001, i.e., the top 1 000 frequent observed character categories and an "others" category. Since the images in our dataset are of high resolution, we slightly modify YOLOv2 to adapt it to our dataset. In the training phase, we set the input resolution of YOLOv2 to 672×672. Each training image ($2\,048 \times 2\,048$) is uniformly cropped into 196 subimages, each of which has the resolution of 168×168 with the overlap of 23 or 24 pixels. Then these cropped images are resized to 672×672, and fed into YOLOv2 as input. In the testing phase, since character instances vary a lot in sizes, we perform a multi-scale scheme. We segment each input image ($2\,048 \times 2\,048$) into three scales: 64 subimages (304×304) with overlapping of 54–55 pixels, 16 larger subimages (608×608) with the overlap of 128 pixels, and four larger subimages ($1\,216 \times 1\,216$) with the overlap of 384 pixels. After that, all the 84 subimages from all scales are resized to the resolution of $1\,216 \times 1\,216$ and then fed into YOLOv2 as input. Finally, non-maximum suppression is applied to remove duplicated detections. Data preprocessing and network modification for SSD512 are performed in a similar way.

In the testing set, YOLOv2 achieves an overall AP of 70.9%. More specifically the APs for character instances of large, medium and small sizes are 77.6%, 75.2% and 58.4%, respectively. SSD512 achieves an overall AP of 64.7%, and the APs for character instances of large, medium and small sizes are 71.7%, 68.5% and 54.5%, respectively. In Table 3, we show the AP scores for the top 10 frequent observed character categories. Fig.10 gives overall precision-recall curves, and the precision-recall curves for characters with different sizes.

In Fig.11, we provide the recall rates of YOLOv2 for character instances with different attributes and different sizes, respectively. To compute the recall rates, for each image in the testing set, denoting the number of annotated character instances as n, we select n detected character instances with the highest confidences as the output of YOLOv2. The results are also consistent with our intuition, i.e., simple characters are easier to be detected and recognized. For example, the recall rates of not occluded characters (71.9%) and printed characters (70.0%) are higher than those of occluded characters (57.0%) and handwritten characters (54.2%), respectively. However, the recall rate of planar characters (69.6%) is lower than that of raised characters (70.6%). The reason might be that the raised characters have stronger structures than the

Table 3. AP (%) of the 10 Most Frequent Character Categories

Method	中	国	大	电	路	车	店	家	公	行	All
YOLOv2	**86.6**	**81.5**	**87.4**	**82.9**	89.7	**81.8**	**90.8**	**84.6**	**80.6**	**85.0**	**70.9**
SSD512	81.1	77.1	83.7	80.9	**90.3**	78.4	86.7	82.8	74.6	81.9	64.7

Note: The bold entries indicate the maximum AP for each character category.

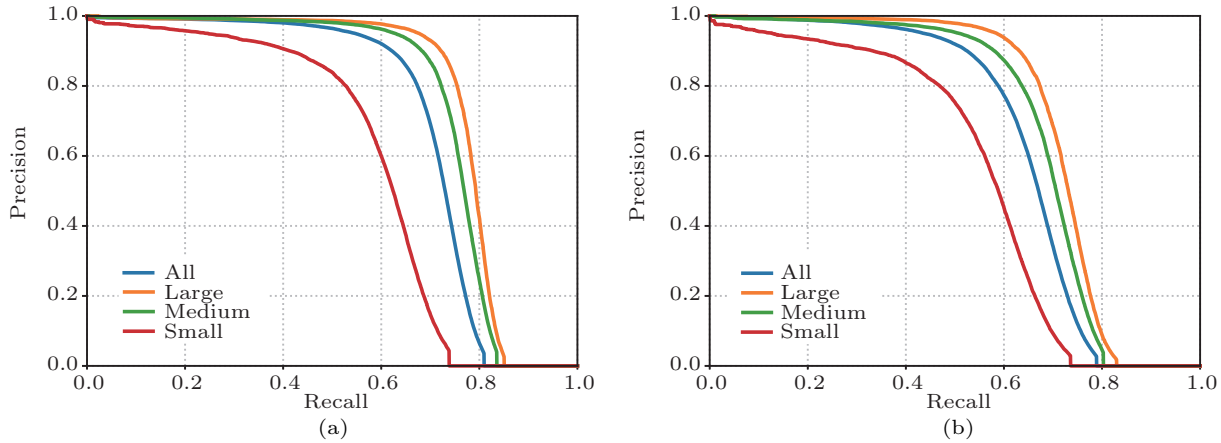


Fig. 10. Precision-recall curves of the detection tasks using (a) YOLOv2 and (b) SSD512, respectively. We show precision-recall curves for all character instances (blue), and curves for character instances with large (yellow), medium (green), and small (red) sizes, respectively.

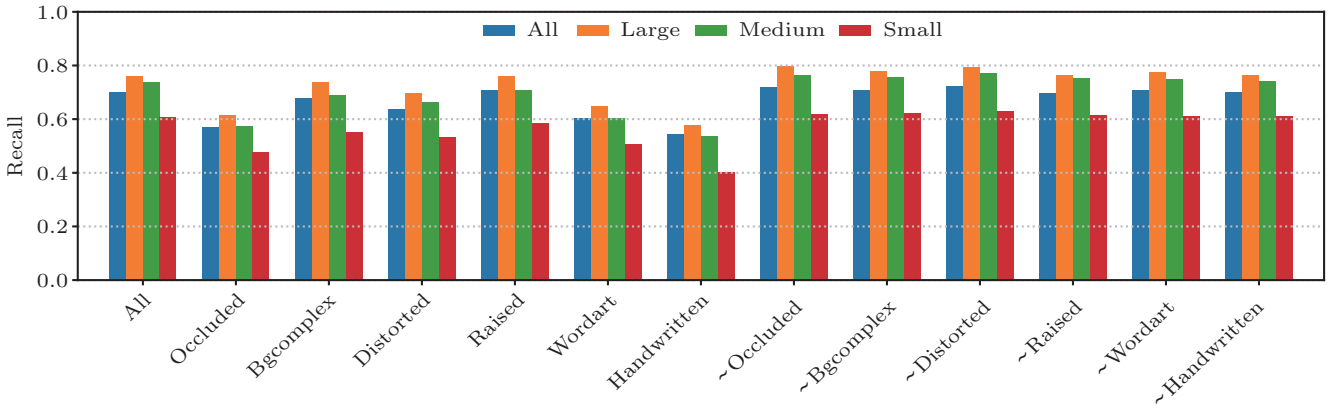


Fig.11. Recall rates of YOLOv2[40] for character instances with different attributes and different sizes. Small, medium, and large refer to character size < 16, $\in [16, 32)$ and $\geqslant 32$, respectively. $\sim$ denotes without a specific attribute, e.g., $\sim$occluded means "not occluded character instances".

planar characters and hence are easier to be detected. We also illustrate some detection results of YOLOv2 in Fig.12. More details can be found in the supplemental material④.

4.3 Text Line Detection

Given an image, the goal of the text line detection task is to detect all text lines. For each text line, we require to detect the bounding region and predict the text (i.e., all character instances). Ground truth of the bounding region of a text line is provided as the convex hull of all character instances in the line (seeing the first and the third images in Fig.13). The averaged edit distance (AED)[15] serves as the major metric for text line detection. To evaluate detection results, we first match all detections to all ground truths. We determine a matched pair between a detection and a ground truth if their regions have an IoU over 0.5, and no matches share a detection or a ground truth. A detection matching an "ignore" region is also excluded. After that, we compute the edit distance for each matched pair. For unmatched detections and ground truths, the edit distance is assigned as the length of themselves. Finally, the averaged edit distance is computed by the sum of all edit distances in all images divided by the number of images.

Our baseline method takes a bottom-up scheme. First, we use YOLOv2 to detect all character instances, as described in Subsection 4.2. Second, we remove character instances with low confidence, and apply non-

④https://ctwdataset.github.io/, March 2019.

Fig.12. Detection results by YOLOv2[40]. For each image, we give the detected characters and their bounding boxes. Correct detections are shown in green while wrong detections are shown in yellow.

Fig.13. Examples of text line detection. (a) and (c) illustrate the ground truths, where text lines are shown in green while ignored regions are shown in yellow. (b) and (d) present the corresponding detection results, where detected text lines are colored in red.

maximum suppression across all character categories. Finally, we concatenate character instances into text lines, if they are similar in size, aligned well and uniformly distributed along a horizontal or vertical line. The bounding region of each detected text line is set as the convex hull of its character instances. Two results are shown in Fig.13. The above described baseline method achieved an AED of 22.1. Note that the average number of character instances in each image on detection test set is 32.6.

5 Conclusions

We introduced Chinese Text in the Wild, a very large dataset of Chinese text in street view images. It contains 32 285 images with 1 018 402 Chinese character instances, and is the largest publicly available dataset for Chinese text in natural images. We annotated all Chinese characters in all images. For each Chinese character, the annotation includes its underlying character, bounding box, and six attributes. We also provided baseline algorithms for three tasks: character recognition from cropped regions, character detection from images, and text line detection from images. We believe that our dataset will greatly stimulate future work in Chinese text detection and recognition.

There are various directions for future work. Firstly, it will be worthwhile to use photorealistic rendering techniques to generate synthetic text images, to further enhance the size and diversity of the dataset, especially for those unusual characters. Secondly, while the current provided baseline algorithms are all character-based, context information will be certainly useful for designing future algorithms to further improve the performance of Chinese character detection and recognition.

References

[1] Cui Y, Zhou F, Lin Y, Belongie S. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. In *Proc. the 29th IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.1153-1162.

[2] Deng J, Dong W, Socher R, Li L J, Li K, L F F. ImageNet: A large-scale hierarchical image database. In *Proc. the 22nd IEEE Conference on Computer Vision and Pattern Recognition*, June 2009, pp.248-255.

[3] Lin T Y, Maire M, Belongie S, Hays J, Perona P, Ramanan D, Dollár P, Zitnick C L. Microsoft COCO: Common objects in context. In *Proc. the 13th European Conference on Computer Vision*, April 2014, pp.740-755.

[4] Zhou B, Zhao H, Puig X, Xiao T, Fidler S, Barriuso A, Torralba A. Semantic understanding of scenes through the ADE20K dataset. *International Journal of Computer Vision*, 2019, 127(3): 302-321.

[5] Lucas S M, Panaretos A, Sosa L *et al.* ICDAR 2003 robust reading competitions: Entries, results, and future directions. *International Journal on Document Analysis and Recognition*, 2005, 7(2/3): 105-122.

[6] Mishra A, Alahari K, Jawahar C V. Scene text recognition using higher order language priors. In *Proc. the 2012 British Machine Vision Conference*, September 2012, Article No. 127.

[7] Smith R, Gu C, Lee D, Hu H, Unnikrishnan R, Ibarz J, Arnoud S, Lin S. End-to-end interpretation of the French Street Name Signs dataset. In *Proc. the 14th European Conference on Computer Vision*, October 2016, pp.411-426.

[8] Veit A, Matera T, Neumann L, Matas J, Belongie S. COCO-text: Dataset and benchmark for text detection and recognition in natural images. arXiv:1601.07140, 2016. https://arxiv.org/abs/1601.07140, March 2019.

[9] de Campos T E, Babu B R, Varma M. Character recognition in natural images. In *Proc. the 4th International Conference on Computer Vision Theory and Applications*, February 2009, pp.273-280.

[10] Jaderberg M, Simonyan K, Vedaldi A, Zisserman A. Synthetic data and artificial neural networks for natural scene text recognition. arXiv:1406.2227, 2014. https://arxiv.org/abs/1406.2227, March 2019.

[11] Netzer Y, Wang T, Coates A, Bissacco A, Wu B, Ng A Y. Reading digits in natural images with unsupervised feature learning. https://ai.google/research/pubs/pub37648, March 2019.

[12] Wang K, Babenko B, Belongie S J. End-to-end scene text recognition. In *Proc. the 2011 International Conference on Computer Vision*, November 2011, pp.1457-1464.

[13] Jung J, Lee S, Cho M S, Kim J H. Touch TT: Scene text extractor using touchscreen interface. *Journal of Electronics and Telecommunications Research Institute*, 2011, 33(1): 78-88.

[14] Yao C, Bai X, Liu W, Ma Y, Tu Z. Detecting texts of arbitrary orientations in natural images. In *Proc. the 25th IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp.1083-1090.

[15] Shi B, Yao C, Liao M, Yang M, Xu P, Cui L, Belongie S, Lu S, Bai X. ICDAR2017 competition on reading Chinese text in the wild (RCTW-17). In *Proc. the 14th IAPR International Conference on Document Analysis and Recognition*, November 2017, pp.1429-1434.

[16] Epshtein B, Ofek E, Wexler Y. Detecting text in natural scenes with stroke width transform. In *Proc. the 23rd IEEE Conference on Computer Vision and Pattern Recognition*, June 2010, pp.2963-2970.

[17] Matas J, Chum O, Urban M, Pajdla T. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 2004, 22(10): 761-767.

[18] Chen H, Tsai S S, Schroth G, Chen D M, Grzeszczuk R, Girod B. Robust text detection in natural images with edge-enhanced Maximally Stable Extremal Regions. In *Proc. the 18th IEEE International Conference on Image Processing*, September 2011, pp.2609-2612.

[19] Koo H I, Kim D H. Scene text detection via connected component clustering and nontext filtering. *IEEE Transactions Image Processing*, 2013, 22(6): 2296-2305.

[20] Neumann L, Matas J. A method for text localization and recognition in real-world images. In *Proc. the 10th Asian Conference on Computer Vision*, November 2011, pp.770-783.

[21] Zhang Z, Zhang C, Shen W, Yao C, Liu W, Bai X. Multi-oriented text detection with fully convolutional networks. In *Proc. the 29th IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.4159-4167.

[22] Zhou X, Yao C, Wen H, Wang Y, Zhou S, He W, Liang J. EAST: An efficient and accurate scene text detector. In *Proc. the 30th IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp.2642-2651.

[23] He T, Huang W, Qiao Y, Yao J. Accurate text localization in natural image with cascaded convolutional text network. arXiv:1603.09423, 2016. https://arxiv.org/abs/1603.09423, March 2019.

[24] Tian Z, Huang W, He T, He P, Qiao Y. Detecting text in natural image with connectionist text proposal network. In *Proc. the 14th European Conference on Computer Vision*, October 2016, pp.56-72.

[25] Sheshadri K, Divvala S K. Exemplar driven character recognition in the wild. In *Proc. the 2012 British Machine Vision Conference*, September 2012, Article No. 13.

[26] Shi C, Wang C, Xiao B, Zhang Y, Gao S, Zhang Z. Scene text recognition using part-based tree-structured character detection. In *Proc. the 26th IEEE Conference on Computer Vision and Pattern Recognition*, June 2013, pp.2961-2968.

[27] Zhang D, Chang S F. A Bayesian framework for fusing multiple word knowledge models in videotext recognition. In *Proc. the 2003 IEEE Conference on Computer Vision and Pattern Recognition*, June 2003, pp.528-533.

[28] Mishra A, Alahari K, Jawahar C V. Top-down and bottom-up cues for scene text recognition. In *Proc. the 25th IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp.2687-2694.

[29] Lee S, Kim J. Complementary combination of holistic and component analysis for recognition of low-resolution video character images. *Pattern Recognition Letters*, 2008, 29(4): 383-391.

[30] Wang T, Wu D J, Coates A, Ng A Y. End-to-end text recognition with convolutional neural networks. In *Proc. the 21st International Conference on Pattern Recognition*, November 2012, pp.3304-3308.

[31] Shi B, Bai X, Yao C. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017, 39(11): 2298-2304.

[32] Liao M, Shi B, Bai X, Wang X, Liu W. TextBoxes: A fast text detector with a single deep neural network. In *Proc. the 31st AAAI Conference on Artificial Intelligence*, February 2017, pp.4161-4167.

[33] Ye Q, Doermann D. Text detection and recognition in imagery: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015, 37(7): 1480-1500.

[34] Zhu Z, Liang D, Zhang S, Huang X, Li B, Hu S. Traffic-sign detection and classification in the wild. In *Proc. the 29th IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.2110-2118.

[35] Krizhevsky A, Sutskever I, Hinton G E. ImageNet classification with deep convolutional neural networks. In *Proc. the 26th Annual Conference on Neural Information Processing Systems*, December 2012, pp.1106-1114.

[36] Sermanet P, Eigen D, Zhang X, Mathieu M, Fergus R, LeCun Y. OverFeat: Integrated recognition, localization and detection using convolutional networks. arXiv:1312.6229, 2013. https://arxiv.org/abs/1312.6229, March 2019.

[37] Szegedy C, Liu W, Jia Y, Sermanet P, Reed S E, Anguelov D, Erhan D, Vanhoucke V, Rabinovich A. Going deeper with convolutions. In *Proc. the 28th IEEE Conference on Computer Vision and Pattern Recognition*, June 2015, pp.1-9.

[38] He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In *Proc. the 29th IEEE Conference on Computer Vision and Pattern Recognition*, June 2016, pp.770-778.

[39] Everingham M, Eslami S A, Van Gool L, Williams C K, Winn J, Zisserman A. The PASCAL Visual Object Classes challenge: A retrospective. *International Journal of Computer Vision*, 2015, 111(1): 98-136.

[40] Redmon J, Farhadi A. YOLO9000: Better, faster, stronger. In *Proc. the 30th IEEE Conference on Computer Vision and Pattern Recognition*, July 2017, pp.6517-6525.

[41] Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu C Y, Berg A C. SSD: Single shot multibox detector. In *Proc. the 14th European Conference on Computer Vision*, October 2016, pp.21-37.

Tai-Ling Yuan is a Ph.D. candidate in the Department of Computer Science and Technology, Tsinghua University, Beijing. He received his B.S. degree in computer science and technology from the same university in 2016. His research interests include computer graphics and computer vision.

Zhe Zhu is a postdoctoral associate at the Department of Radiology, Duke University, North Carolina. He received his Ph.D. degree in computer science and technology from Tsinghua University, Beijing, in 2017. His research interests include computer graphics, computer vision and medical imaging.

Kun Xu is an associate professor in the Department of Computer Science and Technology, Tsinghua University, Beijing. Before that, he received his Ph.D. degree in computer science and technology from the same university in 2009. His research interests include realistic rendering and image/video editing.

Cheng-Jun Li is the team lead of Tencent Autonomous Driving Business Center, Beijing, and is responsible for the HD map, perception and localization systems. He received his Ph.D. degree in computer science from Peking University, Beijing, in 2007. His research interests include 3D modeling, virtual reality, AI and robotics.

Tai-Jiang Mu is currently a postdoctoral researcher in the Department of Computer Science and Technology, Tsinghua University, Beijing, where he received his Ph.D. degree in computer science and technology in 2016. His research area is computer graphics, mainly focusing on image and video processing, scene understanding and interaction for robot.

Shi-Min Hu received his Ph.D. degree in computer science from Zhejiang University, Hangzhou, in 1996. He is currently a professor in the Department of Computer Science and Technology, Tsinghua University, Beijing. His research interests include digital geometry processing, video processing, rendering, computer animation, and computer-aided geometric design. He has published more than 100 papers in journals and refereed conferences. He is the Editor-in-Chief of Computational Visual Media (Springer), and on the editorial boards of several journals, including Computer Aided Design (Elsevier), Computer & Graphics (Elsevier), and Journal of Computer Science and Technology (Springer).