Chinese Text in the Wild

Tai-Ling Yuan
Tsinghua University
Beijing, China
yuantailing@gmail.com

Zhe Zhu
Tsinghua University
Beijing, China
ajex1988@gmail.com

Kun Xu
Tsinghua University
Beijing, China
xukun@tsinghua.edu.cn

Cheng-Jun Li
Tencent
Beijing, China
chengjunli@tencent.com

Shi-Min Hu
Tsinghua University
Beijing, China
shimin@tsinghua.edu.cn

Abstract

We introduce Chinese Text in the Wild, a very large dataset of Chinese text in street view images. While optical character recognition (OCR) in document images is well studied and many commercial tools are available, detection and recognition of text in natural images is still a challenging problem, especially for 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 annotated by experts in over 30 thousand street view images. This is a challenging dataset with good diversity. It contains planar text, raised text, text in cities, text in rural areas, text under poor illumination, distant text, partially occluded text, etc. For each character in the dataset, the annotation includes its underlying character, its bounding box, and 6 attributes. The attributes indicate whether it has complex background, whether it is raised, whether it is handwritten or printed, etc. The large size and diversity of this dataset make it suitable for training robust neural networks for various tasks, particularly detection and recognition. We give baseline results using several state-of-the-art networks, including AlexNet, OverFeat, Google Inception and ResNet for character recognition, and YOLOv2 for character detection in images. Overall Google Inception has the best performance on recognition with 80.5% top-1 accuracy, while YOLOv2 achieves an mAP of 71.0% on detection. Dataset, source code and trained models will all be publicly available on the website[1].

[1]https://ctwdataset.github.io/

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

1. Introduction

Automatic text detection and recognition is an important task in computer vision, with many uses ranging from autonomous driving to book digitization. This problem has been extensively studied, and has been divided into two problems at different levels of difficulty: text detection and recognition in document images, and text detection and recognition in natural images. The former is less challenging and many commercial tools are already available. However, text detection and recognition in natural images are 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 [2]. As shown in Figure 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 learn-

ing in many fields, including image classification, speech recognition, and so on. Very deep networks with tens 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 a variety of detection, classification, recognition tasks. These models need massive amount of data for training. Availability of data is, indeed, a key factor in the success of deep neural networks. The public datasets, such as Image-Net dataset [4], the Microsoft COCO dataset [13] and the ADE20K dataset [33], have become a key driver for progress in computer vision.

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, 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 cities, text in rural areas, text under poor illumination, distant text, partially occluded text, etc. For each image, we annotate all Chinese texts in it. For each Chinese character, we annotate its underlying character, its bounding box, and 6 attributes to indicate whether it is occluded, having complex background, distorted, 3D raised, wordart, and handwritten, respectively.

We have used the dataset as a basis to train deep models using several state-of-the-art approaches for character recognition, and character detection in images. These models are also presented as baseline algorithms. The dataset, source code and baseline algorithms will all be publicly available. We expect the dataset to greatly stimulate 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 given in Section 4, and conclusions are presented 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 numbers. We briefly review both benchmark datasets and approaches from 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 [24, 26, 16, 14], and those that contain synthetic text [3, 8]. Images in these datasets are mainly of two kinds: Internet images, and Google Street View images. For example, the SVHN [18] and SVT [27] datasets utilize Google Street View images, augmented by annotations for numbers 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 [9] for Korean text, FSNS [24] for French text, and MSRA-TD500 [29] for Chinese text. However, MSRA-TD500 dataset [29] only contains 500 natural images, which is far from sufficient for training deep models such as convolutional neural networks. In contrast, our dataset contains over 30 thousand images and about 1 million Chinese characters.

2.2. Text detection and recognition

Text detection and recognition approaches can be classified as those approaches that use hand-crafted features, and those approaches that use automatically learned features (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.

The most widely used approach to text detection based on hand-crafted features is the *stroke width transform* (SWT) [5]. The SWT transforms an image into a new stroke-width image with 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) [15, 1, 10, 19]. Such MSERs always contain non-text regions, so a robust filter is needed for candidate text region selection. Recently, deep learning based approaches have been adopted for text detection, including both fully convolutional networks (FCN) [32] and cascaded convolutional text networks (CCTN) [7].

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 [22] and part-based models [23] have been applied and found to work well. Word recognition provides additional contextual information, so Bayesian inferencing [31], conditional random fields [17] and graph models [12] can now be used.

A recent trend is to focus on 'end-to-end' recognition [28]; more can be found in a detailed survey by Ye and Doermann [30].

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, 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 as a *category*, and refer to an observed instance of a Chinese character in an image as a *character instance*, or as 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 2048×2048. We manually check all street view images, and remove those images which do not contain any Chinese characters. Besides, since the street view images were captured at fixed intervals (i.e., 10 to 20 meters), successive images may have large duplicated areas. Hence, we manually check each pair of successive images, if duplicated areas cover more than 70% of the total image size, we also remove one image. Finally, 32,285 images are selected.

3.2. Annotation

We now describe the annotation process in detail. For each image, all Chinese character instances are annotated. Characters in English and other languages are not annotated. Our annotation pipeline is illustrated in Figure 2. A bounding box is first drawn around a sentence of Chinese text. Next, for each character instance, a more tight bounding box is drawn around it, and its corresponding character instance and its attributes are also specified.

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, *yes* or *no* is specified for each attribute. The occlusion attribute indicates whether the character is occluded, 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 it is frontal. The raised attribute indicates whether the character is 3D raised or it is planar. The wordart attribute indicates whether the character uses a 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 Figure 3. We provide these attributes since the texts have large appearance variations due to color, font, occlusion, and background differences, etc. With the help of these attributes, it will be easier to analyze the algorithm performance on different styles of texts. Researcher may also design algorithms for specific styles of Chinese texts, i.e., 3D raised texts.

In order to ensure high quality, we invite 40 annotation experts for the annotation process. They are employed by a professional image annotation company and are 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 on annotation instructions. The whole annotation process took about 2 months. In total, 1,018,402 Chinese character instances are annotated. Figure 4 shows two images in our dataset and the corresponding annotation.

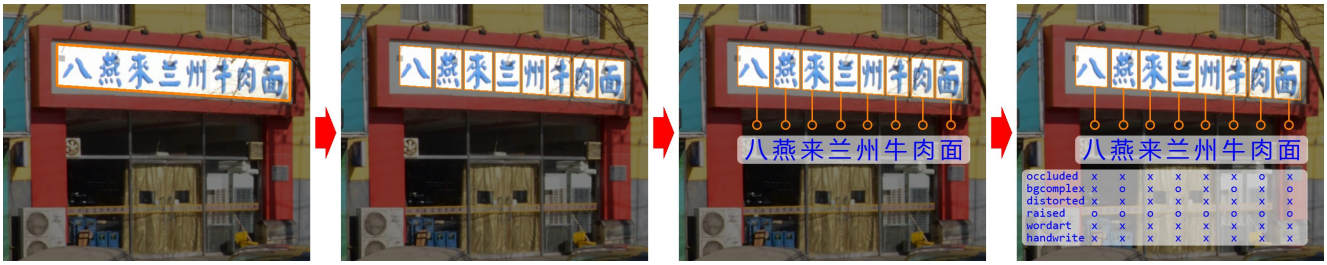
3.3. Dataset splitting

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 (Section 4.1) and a detection testing set for the detection task (Section 4.2). We set the ratio of the sizes of the three sets to $8:1:1$. We randomly distribute all the images into the three sets according to the ratio. To avoid correlation between training and testing images, we constrain that the images captured on the same street must be in the same set. After splitting, 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

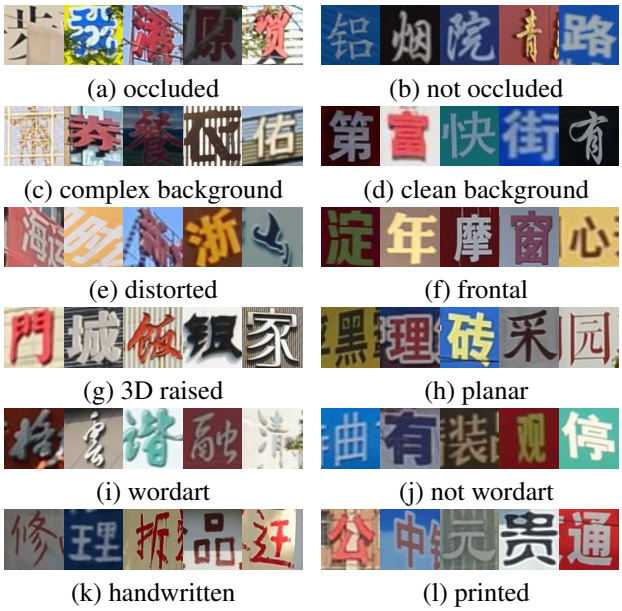
Our CTW dataset contains 32,285 images with 1,018,402 Chinese character instances. It contains 3,850 character categories (i.e., unique Chinese characters).

In Figure 5, for the top 50 most frequent observed character categories, we show the number of character instances in each category in the training set and in the testing set, respectively. In Figure 6 (a), we show the number of images contains specific number of character instances in the training set and in the testing set, respectively. In Figure 6 (b), we show the number of images containing specific number of character categories in the training set and in the testing set, respectively. In Figure 7, 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 Figure 8, we provide the percentage of character instances with different attributes in all/large/medium/small character instances, respectively. Small, medium, and large refer to character size < 16, $\in [16, 32)$ and ≥ 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



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

Figure 2. Annotation pipeline: drawing a bounding box for the sentence (a), drawing a bounding box for each character instance (b), labeling its corresponding character category (c), and labeling its attributes (d).



(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

Figure 3. Examples with different attributes.

Figure 4. Left: two images in our dataset. Right: corresponding ground truth annotation.

character instances, a higher proportion (19.2%) of them are occluded.

Diversity The above statistics show that our dataset has good diversity on character categories, character sizes, and attributes (i.e., occlusion, background complexity, 3D raised, etc.). As shown in Figure 8, 13.2% character instances are occluded, 28.0% have complex background, 26.0% are distorted, and 26.9% are raised text. As shown in Figure 9, our dataset contains planar text (a), raised text (b), text in cities (c), text in rural areas (d), horizontal text (e), vertical text (f), distant text (g), nearby text (h), text under poor illumination (i), and partially occluded text (j). Due to such diversity and complexity, our CTW dataset is a challenging dataset.

4. Baseline Algorithms and Performance

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

We considered two tasks: character recognition from cropped regions, and character detection from images.

4.1. Recognition

Given a cropped rectangular region showing a Chinese character instance, the goal of the character recognition task is to predict its character category.

We have tested several state-of-the-art convolutional neural network structures for the recognition task using TensorFlow, including: AlexNet [11], OverFeat [21], Google Inception [25], 50 layer ResNet [6](ResNet50) and 152 layer ResNet (ResNet152). We use the training set and the recognition testing set as described in Section 3.3 for training and testing, respectively. Since a majority of the character categories are rarely-used Chinese characters, which

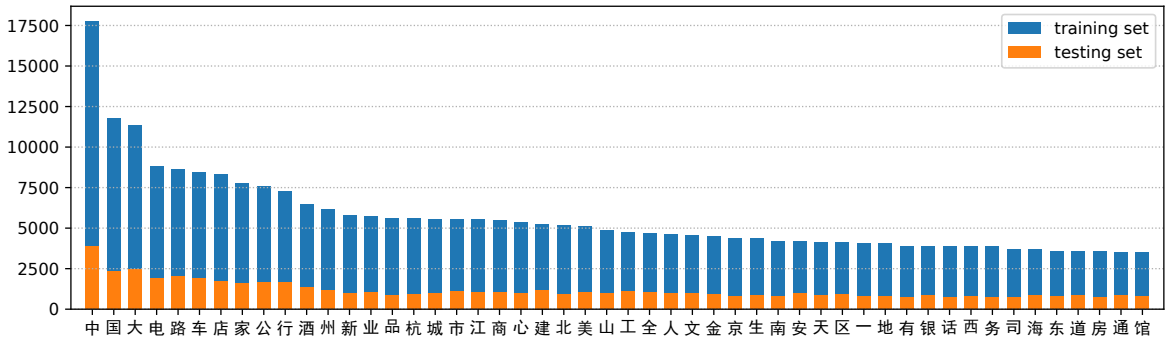
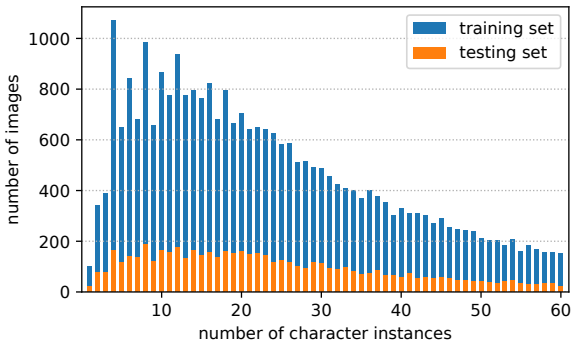
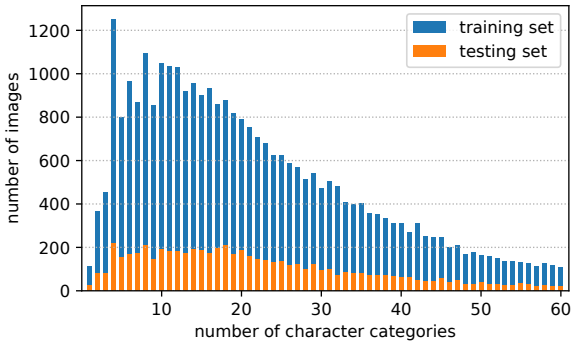


Figure 5. Number of character instances for the 50 most frequent observed character categories in our dataset.



(a)



(b)

Figure 6. Histograms. (a) Number of images containing specific number of character instances; (b) Number of images containing specific number of character categories.

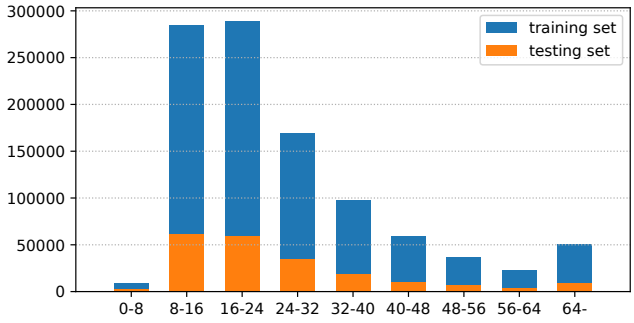


Figure 7. The number of character instances with different sizes. The size is measured by the long side of its bounding box in pixels.

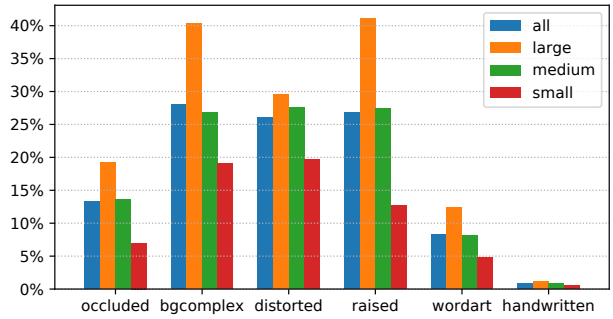


Figure 8. The percentage of character instances with different attributes in all/large/medium/small character instances, respectively. Small, medium, and large refer to character size < 16, $\in [16, 32)$ and ≥ 32, respectively.

have very few samples in the training data and also have very rare usage in practice, we only consider recognition of the top 1000 frequent observed character categories. We consider recognition as a classification problem of 1001 categories. Besides the used 1000 character categories, an 'others' category is added. We trained each network using tens of thousands of iterations, and the parameters of each model are finely tuned. On the testing set, the top-1 accuracy achieved by these networks was: 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 accuracy of the top 10 frequent observed character categories. In Figure 10, 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 best accuracy rate.

In Figure 11, we provide the top-1 accuracy using

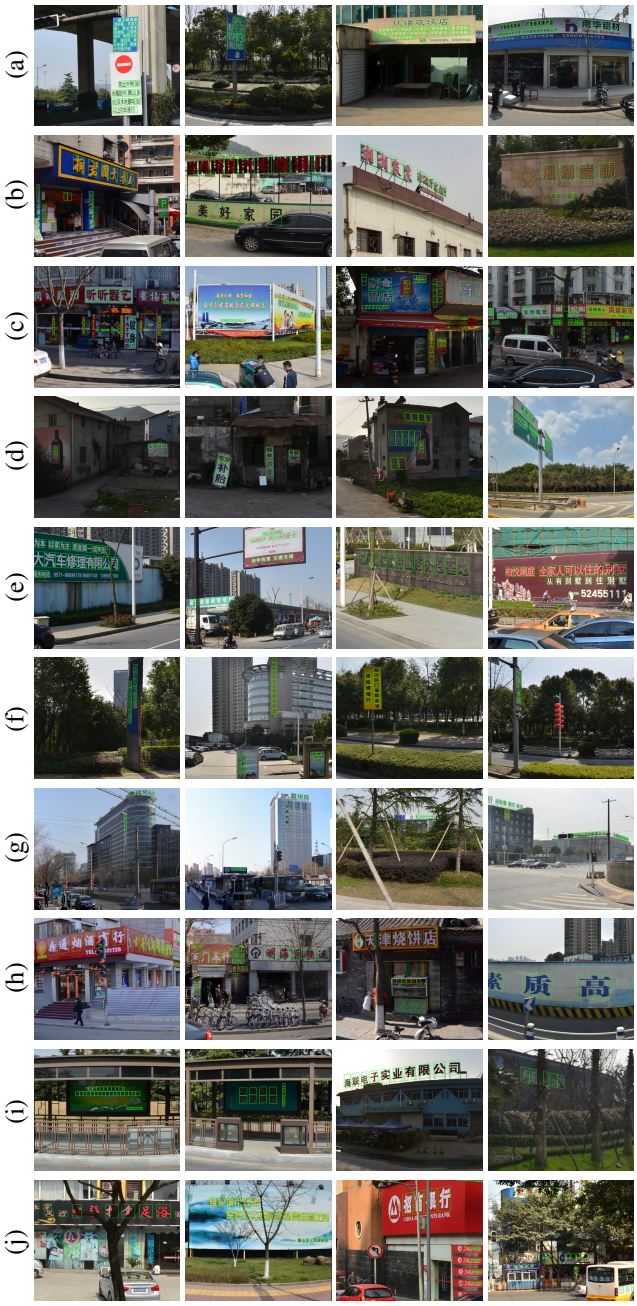


Figure 9. Dataset diversity. (a) planar text, (b) raised text, (c) text in cities, (d) text in rural areas, (e) horizontal text, (f) vertical text, (g) distant text, (h) nearby text, (i) text under poor illumination, (j) partially occluded text.

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 recognize than those with complex background, handwritten characters, and small characters, respectively. An interesting observation is that the recognition accuracy

Figure 10. Some examples of the recognition task. 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 with green. The percentage number shows the confidence of the results.

of large wordart characters (70.0%) is lower than the accuracy of medium wardart characters (72.3%). The reason is that large characters are more likely to be occluded or have complex background (as shown in Figure 8), making them harder to be recognized.

More details can be found on the website.

4.2. Detection

Given an image, the goal of the character detection task is to detect the bounding boxes of all character instances and also recognize each character instance, i.e., predict its character category.

We have tested the YOLOv2 algorithm [20] for the detection task. Given an image, the output of YOLOv2 is a list of recognized character instances, each is associated with a character category, a bounding box, and a confidence score in $[0, 1]$. We use the training set and the detection testing set as described in Section 3.3 for training and testing, respectively. Following the recognition task (Section 4.1), we also limit the number of categories to 1001, i.e., the top 1000 frequent observed character categories and an 'others' category. Since the resolution of images in our dataset is large (i.e., 2048×2048), we have slightly modified YOLOv2 to

Table 1. Top-1 accuracy of the 10 most frequent character categories.

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

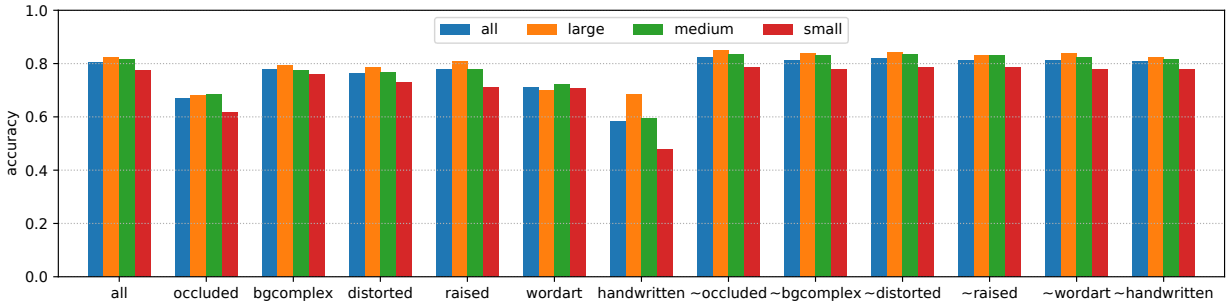


Figure 11. the top-1 accuracy using Google Inception for character instances with different attributes and different sizes. Small, medium, and large refer to character size < 16, $\in [16, 32)$ and ≥ 32, respectively. $\sim$ denotes without a specific attribute , e.g., $\sim$occluded means not occluded character instances.

adapt it to our dataset. For training, first, we set input resolution of YOLOv2 to 672×672. Secondly, each image (2048×2048 resolution) is uniformly segmented into 196 subimages, each of which has resolution of 168×168 and is overlapped with each other by 23-24 pixels. The subimages are scaled to resolution of 672×672, and then are fed into YOLOv2 as input. For testing, since character instances vary a lot in sizes, in order to detect character instances of different sizes, we perform a multi-scale scheme. First, we set input resolution of YOLOv2 to 1216×1216. Secondly, we segment each input image into 16 subimages (608×608 resolution) with overlapping of 128 pixels, and also segment the same input image into 64 smaller subimages (304×304 resolution) with overlapping of 54-55 pixels. After that, all the 80 subimages from both scales are resized to resolution of 1216×1216 and then are fed into YOLOv2 as input. Finally, non-maximum suppression is applied to remove duplicated detections.

In the testing set, YOLOv2 achieves an mAP of 71.0%. In Table 2, we also show the AP scores for the top 10 frequent observed character categories. The AP scores range from 80.3% to 90.3%. In Figure 12, we give the overall precision-recall curve, and the precision-recall curve for characters with different sizes.

In Figure 13, 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 recognized character instances with the highest confidences as 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 (71.6%), printed characters (69.8%) are higher than the recall rates of occluded characters (56.7%), handwritten characters (53.8%), respectively. However, the recall rate of planar characters (69.4%) is lower than that of raised characters (70.1%). The reason might be that raised characters have stronger structures than planar characters and hence they are easier to be detected. We also illustrate some detection results of YOLOv2 in Figure 14. More details can be found on the website.

5. Conclusions

We have 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 will be the largest publicly available dataset for Chinese text in natural images. We annotate all Chinese characters in all images. For each Chinese character, the annotation includes its underlying character, the bounding box, and six attributes. We also provide baseline algorithms for two tasks: character recognition from cropped regions, and character detection from images. We believe that our dataset will greatly stimulate future works in Chinese text detection and recognition.

References

[1] H. Chen, S. S. Tsai, G. Schroth, D. M. Chen, R. Grzeszczuk, and B. Girod. Robust text detection in natural images with edge-enhanced maximally stable extremal regions. In *2011*

Table 2. AP of the 10 most frequent Chinese characters.

	中	国	大	电	路	车	店	家	公	行
YOLOv2	86.2%	81.6%	87.0%	82.3%	89.7%	81.6%	90.3%	81.6%	80.3%	84.2%

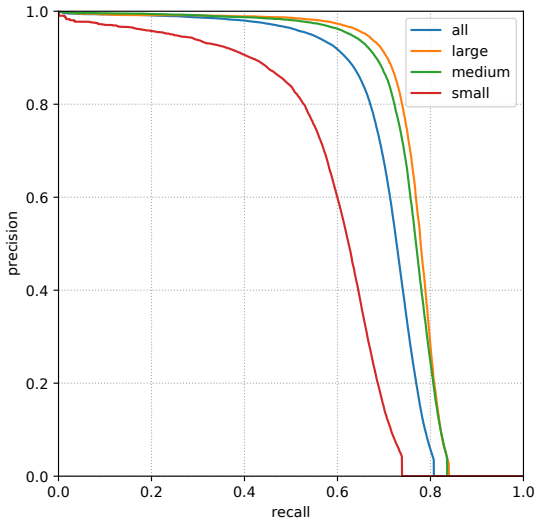


Figure 12. Precision-recall curves of the detection task using YOLOv2 [20]. We show precision-recall curve for all character instances (blue), and curves for character instances with large (yellow), medium (green), and small sizes (red), respectively.

18th IEEE International Conference on Image Processing, pages 2609–2612, Sept 2011.

[2] Y. Cui, F. Zhou, Y. Lin, and S. J. Belongie. Fine-grained categorization and dataset bootstrapping using deep metric learning with humans in the loop. *CoRR*, abs/1512.05227, 2015.

[3] T. E. de Campos, B. R. Babu, and M. Varma. Character recognition in natural images. In *VISAPP (2)*, pages 273–280, 2009.

[4] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[5] B. Epshtein, E. Ofek, and Y. Wexler. Detecting text in natural scenes with stroke width transform. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2963–2970, June 2010.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] T. He, W. Huang, Y. Qiao, and J. Yao. Accurate text localization in natural image with cascaded convolutional text network. *CoRR*, abs/1603.09423, 2016.

[8] M. Jaderberg, K. Simonyan, A. Vedaldi, and A. Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[9] J. Jung, S. Lee, M. S. Cho, and J. H. Kim. Touch tt: Scene text extractor using touchscreen interface. *ETRI Journal*, 33(1):78–88, 2011.

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

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

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

[13] T. Lin, M. Maire, S. J. Belongie, L. D. Bourdev, R. B. Girshick, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft COCO: common objects in context. *CoRR*, abs/1405.0312, 2014.

[14] S. M. Lucas, A. Panaretos, L. Sosa, A. Tang, S. Wong, R. Young, K. Ashida, H. Nagai, M. Okamoto, H. Yamamoto, H. Miyao, J. Zhu, W. Ou, C. Wolf, J.-M. Jolion, L. Todoran, M. Worring, and X. Lin. Icdar 2003 robust reading competitions: entries, results, and future directions. *International Journal of Document Analysis and Recognition (IJDAR)*, 7(2):105–122, 2005.

[15] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761 – 767, 2004. British Machine Vision Computing 2002.

[16] A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012.

[17] A. Mishra, K. Alahari, and C. V. Jawahar. Top-down and bottom-up cues for scene text recognition. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2687–2694, June 2012.

[18] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng. Reading digits in natural images with unsupervised feature learning. In *NIPS Workshop on Deep Learning and Unsupervised Feature Learning 2011*, 2011.

[19] L. Neumann and J. Matas. *A Method for Text Localization and Recognition in Real-World Images*, pages 770–783. Springer Berlin Heidelberg, Berlin, Heidelberg, 2011.

[20] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *CoRR*, abs/1612.08242, 2016.

[21] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. *CoRR*, abs/1312.6229, 2013.

[22] K. Sheshadri and S. Divvala. Exemplar driven character recognition in the wild. In *Proceedings of the British Machine Vision Conference*, pages 13.1–13.10, 2012.

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

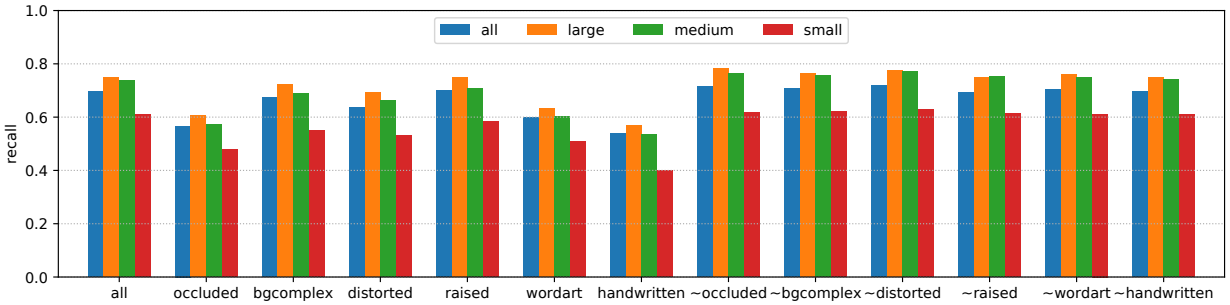


Figure 13. The recall rates of YOLOv2 [20] for character instances with different attributes and different sizes. Small, medium, and large refer to character size < 16, $\in [16, 32)$ and ≥ 32, respectively. $\sim$ denotes without a specific attribute , e.g., $\sim$occluded means not occluded character instances.

Figure 14. Detection results by YOLOv2 [20]. 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.

[24] R. Smith, C. Gu, D.-S. Lee, H. Hu, R. Unnikrishnan, J. Ibarz, S. Arnoud, and S. Lin. *End-to-End Interpretation of the French Street Name Signs Dataset*, pages 411–426. Springer International Publishing, Cham, 2016.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *CoRR*, abs/1409.4842, 2014.

[26] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie. Coco-text: Dataset and benchmark for text detection and recognition in natural images. In *arXiv preprint arXiv:1601.07140*, 2016.

[27] K. Wang, B. Babenko, and S. Belongie. End-to-end scene text recognition. In *2011 International Conference on Computer Vision*, pages 1457–1464, Nov 2011.

[28] T. Wang, D. J. Wu, A. Coates, and A. Y. Ng. End-to-end text recognition with convolutional neural networks. In *Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012)*, pages 3304–3308, Nov 2012.

[29] C. Yao, X. Bai, W. Liu, Y. Ma, and Z. Tu. Detecting texts of arbitrary orientations in natural images. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 1083–1090, Washington, DC, USA, 2012. IEEE Computer Society.

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

[31] D. Zhang and S.-F. Chang. A bayesian framework for fusing multiple word knowledge models in videotext recognition. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, volume 2, pages II–528–II–533 vol.2, June 2003.

[32] Z. Zhang, C. Zhang, W. Shen, C. Yao, W. Liu, and X. Bai. Multi-oriented text detection with fully convolutional networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4159–4167, 2016.

[33] B. Zhou, H. Zhao, X. Puig, S. Fidler, A. Barriuso, and A. Torralba. Semantic understanding of scenes through the ade20k dataset. *arXiv preprint arXiv:1608.05442*, 2016.

[34] Z. Zhu, D. Liang, S. Zhang, X. Huang, B. Li, and S. Hu. Traffic-sign detection and classification in the wild. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.