Dictionary-Guided Text Recognition for Smart Street Parking

Deyang Zhong
dyzhong@uw.edu
Jiayu Li
jiayu7@uw.edu
Wei Cheng
uwcheng@uw.edu
Juhua Hu
juhuah@uw.edu

School of Engineering and Technology
University of Washington
Tacoma, WA USA

Abstract

Smart detection and recognition of the driving environment are critical tasks in the automobile industry, while understanding road signs is a complicated task. When the traffic is heavy or the parking sign is unclear, drivers cannot finish street curbside parking efficiently, which blocks traffic and makes it worse. Numerous object detection and recognition techniques have been adopted to address this issue, but the study for automatic street parking sign understanding, particularly street parking text recognition, is relatively limited. This work bridges the gap between scene text recognition and a smart street curbside parking system. Concretely, we propose a smart street parking sign text recognition method that utilizes a large synthetic data and a small real parking sign text data. We focus on providing a multi-candidates technique built upon one general text recognition method and including parking sign specific text words in the candidates' dictionary. The former collects more text information and reduces potential errors, while the latter increases specificity and performance for the parking sign text recognition task. We compare the performance of leading text recognition engines with our proposed method in a real parking sign text data set. We show significant improvements, demonstrating the feasibility and superiority of our new proposal.

1 Introduction

The current technology for autonomous driving relies on detecting and analyzing road conditions, including but not limited to adjacent cars, traffic lanes, traffic lights, and pedestrians. However, the current road condition detection and recognition systems [3, 9] can only recognize parking signs and their text in a too simplified way to be applicable in a real smart street parking system. Specifically, they cannot precisely understand the parking sign, which can result in parking tickets. For example, they cannot tell the specific time range that parking is allowed that may come from missing the punctuation marks inside the parking sign or misunderstanding the abbreviation. In other words, the existing methods seldom design the dataset or pipeline for precise text recognition of street parking signs. In addition, there are significant differences and diversities in parking signs among regions [33]. Consequently, it is important to develop a system for a precise understanding of parking signs in terms of text, which will enhance autonomous driving technology especially for smart street parking.

Although some researchers have designed and created numerous algorithms and pipelines for smart driving and parking tasks [19, 27, 44], they focused on traffic light detection or parking in the parking lot. The most related existing work includes [17, 24], but they still have a shortcoming in terms of recognition for marks, numbers, and abbreviations. Besides, the accuracy of their models is not satisfying for a real smart street parking system that can avoid parking tickets. Recently, numerous large models have been introduced. Notably, the latest iteration, Generative Pre-trained Transformer 4 [34], purports to possess the capability of comprehending both images and language simultaneously, although this functionality has not yet been made available. Moreover, TrOCR [25] model exhibits commendable performance in Optical Character Recognition (OCR), which can also be adopted for text recognition in images and will be evaluated in this work. Conversely, Segment Anything Model [20] demonstrates remarkable proficiency in object segmentation, though lacking the capacity for text recognition.

Therefore, to do a precise understanding of parking sign text, we propose a new text recognition framework. Recently, CDistNet [54] has become the state-of-art scene text recognition method in various tasks, since it integrates the recognition clues among visual, semantic, and position spaces. However, it ignores the punctuation recognition that is critical for street parking signs, e.g., “Friday, Sunday” is very different from “Friday-Sunday”. Therefore, we first extend CDistNet by adding the punctuation recognition feature, named E-CDistNet. To further improve the text recognition performance, we have observed that street parking signs have a very specific language based on parking-specific dictionary. Thereafter, inspired by [32], we build a parking-specific dictionary that can be used to guide the text recognition procedure for street parking signs. Compared to existing state-of-the-art scene text recognition and OCR methods, our proposal demonstrates a significant improvement in recognizing street parking sign text. The main contributions of this work can be summarized as follows.

- We are the first who aim to do a precise understanding of street parking signs for smart street parking in terms of text recognition, which is critical for avoiding parking tickets.

- Without publicly available street parking sign text recognition data, we are the first who collected and annotated a 9k dataset for street parking sign text recognition.

- We extend the state-of-art scene text recognition method, CDistNet [54], to handle important punctuation marks in parking signs. Moreover, to do further precise understanding of parking text, we build a parking-specific dictionary that is used to guide the text recognition procedure.

- Experiments on a real parking sign text recognition task show significant improvement from our proposed method compared to existing state-of-the-art scene text recognition and OCR methods.

2 Related Work

2.1 Scene Text Recognition

Scene Text Recognition (STR) has developed for years. In the early days, researchers manually constructed text features and characteristics by personal observations or mathematical-statistical models. For example, Wang et al. [46] utilized histograms of oriented gradient (HOG) features in multiple steps to analyze characters and Yao et al. [50] proposed a multi-scale representation termed stroke lets that captures the essential substructures of characters at different granularities. A more comprehensive survey regarding traditional STR methods can be referred to [51]. Although traditional methods deal with specific scene texts, there are common problems especially concerning accuracy and automation.

Recently, deep neural networks (DNNs) have been adopted extensively in STR, which can be divided into two categories, that is, segmentation-based methods and segmentation-free methods. The segmentation-based approaches aim to create a character segmentation from the text image, identify each character separately, and gather them into one text line. This strategy is a combination of character detection and character recognition, thus each step affects the whole model performance. As an instance, Wang et al. [47] employed Random Ferns for character detection and Pictorial Structures formulation for word detection which takes the locations and scores of detected characters as input, while Mishra et al. [31] exploited the spacial structures and higher-order language priors to improve recognition capacity. Despite that these models and furthermore sophisticated lexicon-free methods performed well for specific circumstances, there are two obvious limitations of segmentation-based methods: 1) contextual information and character correlation information are ignored, which are useful for inference; 2) it asks for high requirements of the character detection model, which has become the prominent bottleneck.

Compared to the segmentation-based methods, the segmentation-free approaches regard cropped word images as a whole and infer the target text from the image without character segmentation. This enables the model to utilize semantic information. There are generally four steps for this approach: image preprocessing, feature extraction, sequence modeling, and prediction. Image preprocessing aims to improve the image quality by multiple techniques including background removal [4, 30], text image super-resolution (textSR) [35, 48], and rectification represented by Spatial Transformer Network (STN) and Thin-Plate-Spline(TPS) [1, 39] that could deal with irregular text and reduce the distortion. Then, feature extraction has been greatly improved using DNNs like Convolution Neural Networks (CNN) [7, 16], ResNet [1, 6], and Recurrent Convolution Neural Networks (RCNNs) [26, 45]. In sequence modeling, Bidirectional Long Short-Term Memory(BiLSTM) has been widely used for its potential to capture long-range dependencies despite large time and space consumption [10, 23, 28], while CNNs are also adopted recently for efficiency [11, 36, 52].

Finally, the prediction stage attracts considerable attention and extends to various applicable methods. Considering that CTC [12] functioned well in speech recognition, it was thus introduced to STR [14, 40, 52]. Although CTC demonstrated thrilling performance and stability in multiple models [10, 36, 41], the underlying sophistication and inadequate applicability to 2-dimensional images restricted its capacity. Recently, attention-based methods are gradually replacing the CTC and become the mainstream and evolve into various models [8, 21, 29, 49]. These models build an implicit language model and reduce complexity significantly. Nonetheless, it requires larger training data and vocabulary to construct language model and is incapable of solving the attention drift problem. For the purpose of utilizing the spacial information and overcoming the attention drift problem, many methods are proposed, e.g., localization supervision [7], TextScanner [43], RobustScanner [53], and CDistNet [54]. Among them, CDistNet [54] achieved unprecedented performance in numerous tasks. Nonetheless, as with other STR models, CDistNet dismisses punctuation marks directly that matters a lot in street parking sign text recognition task, e.g., "Friday, Sunday" vs. "Friday-Sunday".

2.2 Street Parking Sign Text Recognition

Although there are many applications of text recognition, from early straightforward documentary text recognition to multiple modern types of scene text recognition on billboards and storefronts, few studies specifically focused on recognizing the text on street parking signs. Irshad [15] introduced a novel framework for parking sign recognition but failed to process compound signs, which limits the applicability. Later, Jiang [17] proposed a parking sign model that included text recognition, sign detection, and sign understanding and Li [24] used a similar text recognition technique (i.e., CRNN [38]) and concentrated on generating rules from the recognized text. Nevertheless, both of them only utilized the semantic clues of text images and thus could not achieve the accuracy of CDistNet [54] which explored multiple clues. Recently, Faraji et al. [9] achieved outstanding performance in parking sign detection and classification tasks while unable to recognize sign text in detail, for example, the time duration or days permitted. In this work, we aim to do precise understanding of street parking sign text by integrating the special vocabulary of street parking signs into CDistNet but also capturing important punctuations that were ignored in CDistNet. It should be noted that our work is built upon the advanced parking sign detection methods in [5, 18] that lacks the parking sign text recognition component.

3 Methodology

As mentioned, in a real smart street parking system, it is critical to precisely understand a street parking sign. Otherwise, even recognizing one character wrong can result in parking tickets. Therefore, in this work, we aim to improve the recognition performance in terms of all characters appearing in the street parking sign. CDistNet [54] has demonstrated state-of-the-art scene text recognition performance in various tasks; however, it lacks the ability to handle characters other than numbers and alphabets. Therefore, we propose an improved framework specifically for street parking sign text recognition based on CDistNet.

3.1 An extension on CDistNet

CDistNet is a Transformer-based encoder-decoder framework with three branches in the encoding step: visual branch, positional branch, and semantic branch. Given a text image $I_i^{W \times H}$ with width W and height H as shown in Fig. 1, the visual branch makes use of TPS [39] to reduce the distortion. Then, a ResNet-50 and Transformer unit are employed to extract features. The process could be summarized as

$$F_i = T_2(R(T_1(I_i^{W \times H}))) \in R^{P \times E}$$

where T_1, R, and T_2 are TPS, ResNet-50, and Transformer, respectively. P is the length of reshaped visual features and E denotes channels of visual features.

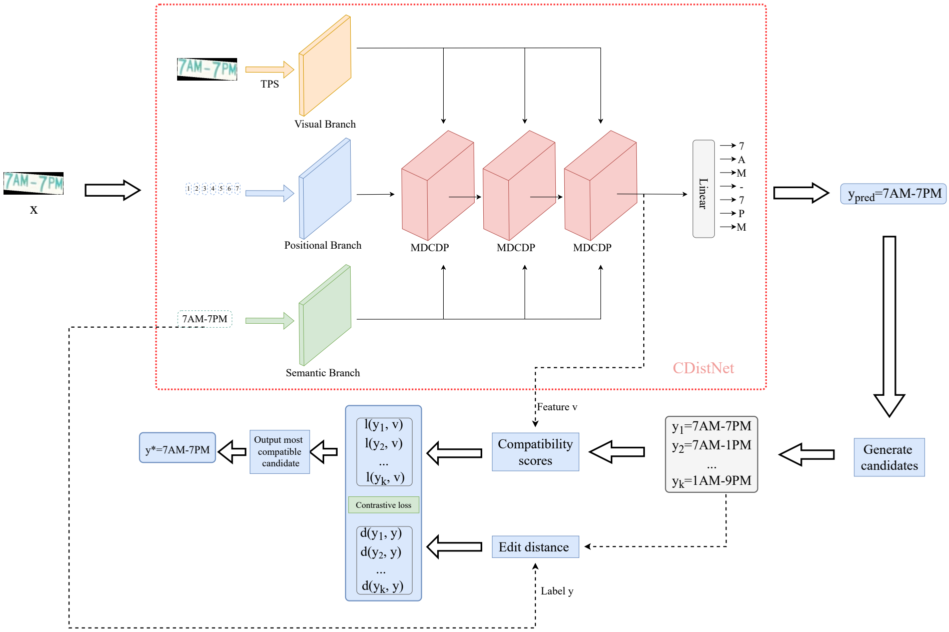


Figure 1: The proposed street parking sign text recognition pipeline.

The positional and semantic branches encode the labels of images, e.g., "7AM-7PM" as shown in Fig. 1. A superior semantic method as in [37] and an outstanding positional method used in [42] are applied to capture the positional and semantic information. The two outputs are respectively:

$$F_i^{pos},\ F_i^{sem} \in R^{T\times E}$$

where T denotes the number of characters. It should be noted that in the reference stage, T is unknown ahead of time. We can initialize it as 1 and generate the first token as F_{i0}^{pos}, $F_{i0}^{sem} \in R^{1\times E}$. Then, they are used to produce second token and the process repeats until the last token is processed.

In the decoding part, CDistNet [54] designed a module called Multi-Domain Character Distance Perception (MDCDP) that is used three times to improve the ability to extract input features. Specifically, the three feature branches extracted in the previous stage are fed into the first MDCDP. Subsequently, the output of the first MDCDP along with visual and positional branches is received as input in the second layer of MDCDP, then the same operation is executed by the third layer. At last, a linear classifier is employed to infer each decoded character as y_{pred} shown in Fig. 1. However, it should be noted that '-' is ignored by the original CDistNet method, which is very important for street parking sign understanding. Therefore, to address the limitation of the original CDistNet model that fails to handle punctuation marks, we introduce a punctuation recognition feature into the model. This enhancement allows for recognition of punctuation marks within the parking sign text, named Edited CDistNet (E-CDistNet). Specifically, the ground-truth label for the image in Fig. 1 is "7AM7PM" in CDistNet and it becomes "7AM-7PM" in E-CDistNet, in which the recognition of "-" is also considered as shown in Fig. 1.

3.2 Dictionary-guided E-CDistNet

However, the above simple extension by adding only the punctuation recognition may not be sufficient to do a precise understanding of the street parking sign text. Fortunately, we observe that although we have a lot of varieties in street parking signs in different cities across the country, we have very specific parking sign languages compared to general natural languages. Therefore, a finite parking-specific word dictionary can be built. Concretely, the dictionary contains special words especially used in street parking text. For instance, in the dictionary, we build the abbreviation of days and times including "Mon", "Fri", and "AM", the words containing punctuation marks including "8:00AM", and even the complex time duration like "7AM-3PM". Then, we use this dictionary to guide the training to further improve the recognition performance for street parking sign text recognition as follows.

3.2.1 Candidate Generation

Given the E-CDistNet prediction y_{pred}, we can generate a list of candidate words, e.g., $y_1, y_2, ..., y_k$. These words have the smallest distance regarding y_{pred} in our parking-specific dictionary. To determine the words with the smallest distance, we utilize the Levenshtein distance [22] that measures the difference between two sequences by the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other. k is set to 10 as suggested by our experiments in Sec. 4.3. It should be noted that we use '###' to fill candidates if less than k candidates are within three characters distance away from y_{pred}. Here, We choose three in accordance with the precedent set by the previous paper[32]. It aims to maximize our pool of potential options to enhance accuracy, and simultaneously to minimize the degree of mismatch.

3.2.2 Updated Training Loss

Thereafter, during the training, we consider two components, where the first is the traditional text recognition empirical loss comparing the prediction to the ground truth, while the second compares the prediction's candidate list to the ground truth. The second aims to give feedback to the training process on how bad the current prediction is from the perspective of the closest candidates in the dictionary.

Concretely, the first loss is the negative log likelihood of the ground truth $\mathbf{y}$ in terms of recognition, formally as

$$\ell(\mathbf{y}, \mathbf{v}) = -\sum_{j=1}^{len(\mathbf{y})} \log \mathbf{P}[y^j, j]$$

where the feature $\mathbf{v}$ is the output from the third MDCDP layer as shown in Fig. 1. We generate a probability matrix denoted as $\mathbf{P}$ through the utilization of a recurrent neural network with attention mechanisms [2], which is of size $s \times m$, where m represents the maximum length of all $\mathbf{y}$, and s corresponds to the size of the character dictionary, encompassing both special symbols and characters. y^j denotes the j-th character of $\mathbf{y}$. Therefore, given the extracted feature $\mathbf{v}$, if the probability of the ground truth character is high for each position, the loss is low.

The second loss is based on the KL-divergence between the compatibility scores and distances as

$$KL(D||L) = -\sum_{i=1}^{k} D_i \log(L_i)$$

where k denotes the number of candidates, D_i and L_i are the softmax value of $d(\mathbf{y_i},\mathbf{y})$ (i.e., Levenshtein edit distance) and $\ell(\mathbf{y_i},\mathbf{v})$ (i.e., the negative log likelihood). Specifically, L_i is $\exp\left(-\ell(\mathbf{y_i},\mathbf{v})\right)/\sum_{j=1}^{k}\exp\left(-\ell(\mathbf{y_j},\mathbf{v})\right)$, and D_i is $\exp\left(-d(\mathbf{y_i},\mathbf{y})\right)/\sum_{j=1}^{k}\exp\left(-d(\mathbf{y_j},\mathbf{y})\right)$. Thereafter, the comprehensive loss for each text recognition image can be represented as

$$\ell(\mathbf{y},\mathbf{v})+\lambda KL(D||L)$$

where the λ is a trade-off parameter and is set to 1 in this work. Fig. 1 summarizes the overall procedure of our proposed method to enhance the text recognition performance for street parking signs.

4 Experiment

To evaluate the proposed method, we conducted experiments on a desktop PC with an Intel(R) Core(TM) i9-9820X CPU, 128GB of RAM, and four GeForce RTX-2080 graphics cards with 11GB of memory. To demonstrate our proposal, we compare it with four state-of-the-art text recognition baselines as follows. These models are carefully selected based on their relevance and demonstrated performance in the field of text recognition.

- **CRNN** [38]: an established deep neural network that has demonstrated competitiveness in recognizing sequence-like images, making it a suitable candidate for our task.

- **CDistNet** [54]: a recently proposed state-of-the-art architecture specifically designed for text recognition tasks. It incorporates multiple branches for information extraction, which enhances its overall performance.

- **E-CDistNet**: an edited model based on CDistNet [54] as a variant baseline to address the limitation related to the capability to handle punctuation marks.

- **TrOCR** [25]: a recently proposed state-of-the-art OCR method that has gained recognition as an exceptional model for text recognition.

To the best of our knowledge, there is no publicly available street parking sign text recognition data. Therefore, we manually collected and annotated our own parking text dataset. We firstly collected about 4k street view images containing street parking signs from various cities in the US (including but not limited to New York, Connecticut, Boulder, Seattle, and San Francisco), of which 2k are street-level images taken by hand, while the rest of them are street-level video frames from dash cameras. After that, we adopted the parking sign detection and symbol detection methods in [5, 18] to crop word images out of each street-level image, resulting in parking text images as shown in Fig. 2. In total, our street parking text data consists of approximately 9k parking text images. The images in our dataset reflect the complexities and variations encountered in parking environments, such as different lighting conditions, text sizes, orientations, and backgrounds. Considering that 9k images can be insufficient to provide good text recognition performance, we also added the SynthText [13] data during the training. SynthText is a synthetically generated dataset that incorporates word instances into natural scene images as shown in Fig. 3, which comprises approximately 7 million synthetic word instances, with about 20% of them containing punctuation marks.

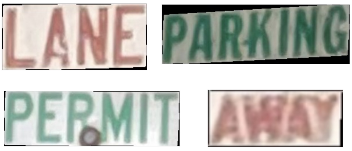


Figure 2: Street Parking Text

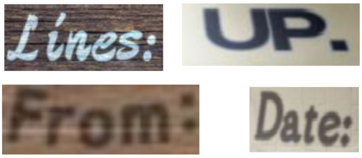


Figure 3: SynthText

4.1 Evaluation Setup

For each compared text recognition method, we use 80% of the parking text data and the whole SynthText for training, and the remaining 20% of the parking text data for test. However, the pre-trained TrOCR [25], which was trained on 684 million text lines, is directly used to test 20% of the parking text test data for comparison without fine-tuning, as authors of TrOCR used 8 V100 GPUs for fine-tuning, a resource beyond our capacity. A traditional evaluation metric for text recognition is the recognition accuracy in the word level. For example, if we can recognize all characters in “7AM-7PM” correctly, this is a correct recognition. Otherwise, at least one character including ‘-’ is not recognized, this is a wrong recognition.

However, in a real smart street parking system, some minor mistakes can be easily corrected during a post-processing stage. Therefore, in this work, we also separate the recognition errors into two categories, that is, repairable errors and irreparable errors. The repairable errors are words that could be corrected to standard parking language in the post-processing step. For instance, recognizing “7;00” as “7:00” is a repairable error because there is only one punctuation difference that can be fixed in the post-processing. However, misidentifying “ALL” as “AT” is irreparable due to the non-negligible gap. We believe it is crucial to differentiate between these two aspects in order to highlight the specific improvements made. This results in two new evaluation metrics, that is, repairable error rate and irreparable error rate. Then, the overall error rate as the traditional method is their summation. It should be noted that during the evaluation, we pass each mis-recognized result to our post-processing step. If the output from the post-processing stage is correct, it is treated as a repairable error. Moreover, to evaluate their combined effects on the overall smart street parking system, we implement a new error rate named Comprehensive Error Rate (CER). Considering that the repairable ER will not affect the final output but may consume time and computation, we decide to shrink its weight, instead of directly removing it. Concretely, based on the error distribution and its effect on subsequent steps, we maintain the irreparable error rate and reduce the repairable error rate to one-third of its previous level by assigning different weights to each type of error as

$$CER = \frac{1}{3}ER_{repariable} + ER_{irreparble}.$$

4.2 Performance Comparison

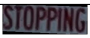
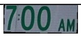
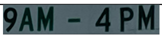
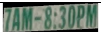
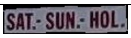
Table 1 summarizes the results, in which the best performance for each evaluation metric is in bold. First, it can be observed that the original CDistNet model, due to its inability to handle punctuation, yields the highest error rate among the evaluated models, while the CRNN model, which was used in [17, 24] and has the capability in punctuation, still exhibits poor

Table 1: Parking Sign Text Recognition Performance Comparison (ER: Error Rate).

Model	CRNN	CDistNet	E-CDistNet	TrOCR	Our Model
ER (%)	13.3	16.7	10.0	7.4	**4.3**
Repairable ER (%)	4.8	5.3	3.7	5.4	**3.2**
Irreparable ER (%)	8.5	11.4	6.3	2.0	**1.1**
CER (%)	10.1	13.2	7.5	3.8	**2.2**

performance. Second, through our modifications to the punctuation recognition function, the edited CDistNet model shows significant improvement and surpasses the CRNN model in terms of performance, which confirms the state-of-the-art performance of CDistNet in terms of text recognition using multiple clues. Additionally, the TrOCR model, despite not being specifically designed for this task, exhibits remarkable performance during the evaluation on our parking text test data. More importantly, our model, which builds a parking-specific dictionary as an additional text recognition guidance, demonstrates significantly better performance across all metrics. It improved a lot in Irreparable ER, while the improvement in Repairable ER is limited, mainly due to minor punctuation recognition problems such as ";" and ":", or missing ".". Fortunately, these repairable ones can be corrected in the post-processing step.

Table 2: Recognition Examples.

Images					
CRNN	STOPPING	700AM	9AM-4PM	7AM;8:3OPM	SAT-SUN:HOL
CDistNet	STOPPING	700AM	9AM4PM	7AN830PM	SATSUNHOL
E-CDistNet	STOPPING	7:00AM	9AM-4PM	7AN-8:30PM	SAT-SUN.-HOL
TrOCR	STOPPING	7:00AM	9AM-4PM	7AM-8:30PM	SAT-SUN.-HOL
Our Model	STOPPING	7:00AM	9AM-4PM	7AM-8:30PM	SAT.-SUN.-HOL.

In Table 2, we present five examples to showcase the recognized results of the five evaluated models. The CRNN model exhibits accurate recognition for simple, single-word images, demonstrating its fundamental capability in scene text recognition. However, when faced with images containing punctuation or complex multi-word instances, the model struggles to perform well. For instance in image "7AM-8:30PM", the CRNN model recognizes '-' as ';' and '0' as 'O', highlighting its limitations in handling more complex instances. The original CDistNet model encounters difficulties in dealing with punctuation. As a result, it only outputs the alphabet, while dismissing all punctuation marks present in the text, with a reduced level of mis-recognition on alphabets, e.g., only 'M' as 'N'. The Edited CDistNet (E-CDistNet) model, on the other hand, addresses this issue by modifying the punctuation recognition function. Although it improves punctuation recognition, it still exhibits limitations in accuracy, resulting in some missing punctuation in the last two sample images. In comparison to the previous models, the TrOCR model demonstrates exceptional performance across all sample images. It excels in recognizing one-word images (sample image 1), images containing punctuation (sample image 2), and images displaying time durations (sample images 3 and 4). However, when faced with a combination of date abbreviations and punctuation, the TrOCR model may still miss some punctuation marks.

Our proposed model, which builds upon E-CDistNet but adds the guidance of a parking-specific dictionary, exhibits the best performance across all the sample images presented in

Table 2. It addresses the challenges posed by different scenarios and demonstrates superior recognition accuracy. It is important to point out that these five images were specifically chosen to highlight the most challenging problems encountered by the models and facilitate a comprehensive comparison of their performance. By selecting samples that showcase the most severe issues faced by the models, we can gain a deeper understanding of their limitations and strengths. However, it is essential to consider that these specific scenarios may not represent the entirety of the models' performance in real-world applications.

4.3 Ablation Study

To show the effect of the number of candidates to consider, we change k in $\{3,5,8,10,15\}$ and show the results in Table 3. This result is consistent with our expectation that too small k cannot provide sufficient candidates to guide, while too large k combined with the constraint of smaller than 3 distance will not change the candidates. Therefore, we set $k = 10$.

Table 3: Effect of k.

k	3	5	8	10	15
ER(%)	7.8	4.9	4.5	4.3	4.3
Repairable ER(%)	4.9	3.6	3.4	3.2	3.2
Irreparable ER(%)	2.9	1.3	1.1	1.1	1.1
CER	4.5	2.5	**2.2**	**2.2**	**2.2**

5 Conclusion

To do precise street parking sign understanding in terms of text, we propose a new text recognition framework based on an existing state-of-the-art scene text recognition model (CDistNet). First, punctuation marks are critical to understanding parking sign text correctly without leading to parking tickets, which is ignored in CDistNet, so we extend CDistNet by adding the punctuation recognition feature. Moreover, based on the observation that street parking signs have very specific languages compared to the general languages, we are able to build a parking-specific dictionary that can be used to guide the text recognition procedure to further improve performance. Our proposed method shows significantly better performance in street parking sign text recognition compared to existing state-of-the-art text recognition methods and OCR techniques using various metrics considering a real smart street parking system. However, because of the limited number of parking text images that we were able to collect, our model is not flawless and cannot guarantee the same level of performance in environments that involve unusual or more complicated street parking, which will be our future work. For example, the remaining irreparable errors are mainly due to external factors (e.g., lightning and angle), we can improve by doing specific data augmentation or collecting more such conditioned data in the future.

References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition

model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4715–4723, 2019.

[2] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[3] Pietro Carnelli, Joy Yeh, Mahesh Sooriyabandara, and Aftab Khan. Parkus: A novel vehicle parking detection system. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 31, pages 4650–4656, 2017.

[4] Richard G Casey and Eric Lecolinet. A survey of methods and strategies in character segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18 (7):690–706, 1996.

[5] Hieu Chau, Yin Jin, Jiayu Li, Juhua Hu, and Wei Cheng. Real-time street parking sign detection and recognition. 2021. URL `https://learn-to-race.org/workshop-ai4ad-ijcai2022/assets/papers/paper_13.pdf`.

[6] Xiaoxue Chen, Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, and Canjie Luo. Adaptive embedding gate for attention-based scene text recognition. *Neurocomputing*, 381:261–271, 2020.

[7] Zhanzhan Cheng, Fan Bai, Yunlu Xu, Gang Zheng, Shiliang Pu, and Shuigeng Zhou. Focusing attention: Towards accurate text recognition in natural images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5076–5084, 2017.

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

[9] Parnia Haji Faraji, Hamid Reza Tohidypour, Yixiao Wang, Panos Nasiopoulos, Simon Ren, Arash Rizvi, Cloris Feng, Mahsa T Pourazad, and Victor CM Leung. Deep learning based street parking sign detection and classification for smart cities. In *Proceedings of the Conference on Information Technology for Social Good*, pages 254–258, 2021.

[10] Yunze Gao, Yingying Chen, Jinqiao Wang, Ming Tang, and Hanqing Lu. Dense chained attention network for scene text recognition. In *2018 25th IEEE International Conference on Image Processing (ICIP)*, pages 679–683. IEEE, 2018.

[11] Yunze Gao, Yingying Chen, Jinqiao Wang, Ming Tang, and Hanqing Lu. Reading scene text with fully convolutional sequence modeling. *Neurocomputing*, 339:161–170, 2019.

[12] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 369–376, 2006.

[13] 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.

[14] Pan He, Weilin Huang, Yu Qiao, Chen Loy, and Xiaoou Tang. Reading scene text in deep convolutional sequences. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 30, 2016.

[15] Humayun Irshad, Qazaleh Mirsharif, and Jennifer Prendki. Crowd sourcing based active learning approach for parking sign recognition. *arXiv preprint arXiv:1812.01081*, 2018.

[16] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Deep structured output learning for unconstrained text recognition. *arXiv preprint arXiv:1412.5903*, 2014.

[17] Zhongyu Jiang. Street parking sign detection, recognition and trust system. Master's thesis, University of Washington, 2019.

[18] Yin Jin. Real-time parking sign detection for smart street parking. Master's thesis, University of Washington, 2022.

[19] Abhirup Khanna and Rishi Anand. Iot based smart parking system. In *2016 International Conference on Internet of Things and Applications (IOTA)*, pages 266–270. IEEE, 2016.

[20] Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C Berg, Wan-Yen Lo, et al. Segment anything. *arXiv preprint arXiv:2304.02643*, 2023.

[21] Chen-Yu Lee and Simon Osindero. Recursive recurrent nets with attention modeling for ocr in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2231–2239, 2016.

[22] Vladimir I Levenshtein et al. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710. Soviet Union, 1966.

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

[24] Jiayu Li. An algorithm for street parking sign rule generation. Master's thesis, University of Washington, 2020.

[25] Minghao Li, Tengchao Lv, Jingye Chen, Lei Cui, Yijuan Lu, Dinei Florencio, Cha Zhang, Zhoujun Li, and Furu Wei. Trocr: Transformer-based optical character recognition with pre-trained models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13094–13102, 2023.

[26] Ming Liang and Xiaolin Hu. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3367–3375, 2015.

[27] Trista Lin, Hervé Rivano, and Frédéric Le Mouël. A survey of smart parking solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3229–3253, 2017.

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

[29] Canjie Luo, Lianwen Jin, and Zenghui Sun. Moran: A multi-object rectified attention network for scene text recognition. *Pattern Recognition*, 90:109–118, 2019.

[30] Canjie Luo, Qingxiang Lin, Yuliang Liu, Lianwen Jin, and Chunhua Shen. Separating content from style using adversarial learning for recognizing text in the wild. *International Journal of Computer Vision*, 129(4):960–976, 2021.

[31] Anand Mishra, Karteek Alahari, and CV Jawahar. Enhancing energy minimization framework for scene text recognition with top-down cues. *Computer Vision and Image Understanding*, 145:30–42, 2016.

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

[33] Rafael Martín Nieto, Alvaro Garcia-Martin, Alexander G Hauptmann, and Jose M Martinez. Automatic vacant parking places management system using multicamera vehicle detection. *IEEE Transactions on Intelligent Transportation Systems*, 20(3):1069–1080, 2018.

[34] OpenAI. Gpt-4 technical report, 2023.

[35] Clément Peyrard, Moez Baccouche, Franck Mamalet, and Christophe Garcia. Icdar2015 competition on text image super-resolution. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1201–1205. IEEE, 2015.

[36] Xianbiao Qi, Yihao Chen, Rong Xiao, Chun-Guang Li, Qin Zou, and Shuguang Cui. A novel joint character categorization and localization approach for character-level scene text recognition. In *2019 International Conference on Document Analysis and Recognition Workshops (ICDARW)*, volume 5, pages 83–90. IEEE, 2019.

[37] Fenfen Sheng, Zhineng Chen, and Bo Xu. Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition. In *2019 International conference on document analysis and recognition (ICDAR)*, pages 781–786. IEEE, 2019.

[38] Baoguang Shi, Xiang Bai, and Cong Yao. An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition. *IEEE transactions on pattern analysis and machine intelligence*, 39(11):2298–2304, 2016.

[39] Palaiahnakote Shivakumara, Weihua Huang, Trung Quy Phan, and Chew Lim Tan. Accurate video text detection through classification of low and high contrast images. *Pattern Recognition*, 43(6):2165–2185, 2010.

[40] Palaiahnakote Shivakumara, Souvik Bhowmick, Bolan Su, Chew Lim Tan, and Umapada Pal. A new gradient based character segmentation method for video text recognition. In *2011 International conference on document analysis and recognition*, pages 126–130. IEEE, 2011.

[41] Bolan Su and Shijian Lu. Accurate recognition of words in scenes without character segmentation using recurrent neural network. *Pattern Recognition*, 63:397–405, 2017.

[42] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

[43] Zhaoyi Wan, Minghang He, Haoran Chen, Xiang Bai, and Cong Yao. Textscanner: Reading characters in order for robust scene text recognition. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 12120–12127, 2020.

[44] Hongwei Wang and Wenbo He. A reservation-based smart parking system. In *2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, pages 690–695. IEEE, 2011.

[45] Jianfeng Wang and Xiaolin Hu. Gated recurrent convolution neural network for ocr. *Advances in Neural Information Processing Systems*, 30, 2017.

[46] Kai Wang and Serge Belongie. Word spotting in the wild. In *Computer Vision–ECCV 2010: 11th European Conference on Computer Vision, Heraklion, Crete, Greece, September 5-11, 2010, Proceedings, Part I 11*, pages 591–604. Springer, 2010.

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

[48] Wenjia Wang, Enze Xie, Peize Sun, Wenhai Wang, Lixun Tian, Chunhua Shen, and Ping Luo. Textsr: Content-aware text super-resolution guided by recognition. *arXiv preprint arXiv:1909.07113*, 2019.

[49] Mingkun Yang, Yushuo Guan, Minghui Liao, Xin He, Kaigui Bian, Song Bai, Cong Yao, and Xiang Bai. Symmetry-constrained rectification network for scene text recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9147–9156, 2019.

[50] Cong Yao, Xiang Bai, Baoguang Shi, and Wenyu Liu. Strokelets: A learned multi-scale representation for scene text recognition. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4042–4049, 2014. doi: 10.1109/CVPR.2014.515.

[51] Qixiang Ye and David Doermann. Text detection and recognition in imagery: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(7):1480–1500, 2015. doi: 10.1109/TPAMI.2014.2366765.

[52] Deli Yu, Xuan Li, Chengquan Zhang, Tao Liu, Junyu Han, Jingtuo Liu, and Errui Ding. Towards accurate scene text recognition with semantic reasoning networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 12113–12122, 2020.

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

[54] Tianlun Zheng, Zhineng Chen, Shancheng Fang, Hongtao Xie, and Yu-Gang Jiang. Cdistnet: Perceiving multi-domain character distance for robust text recognition. *International Journal of Computer Vision*, pages 1–19, 2023.