UNITS: UNSUPERVISED INTERMEDIATE TRAINING STAGE FOR SCENE TEXT DETECTION

Youhui Guo\textsuperscript{1,2}, Yu Zhou\textsuperscript{1,2,*}, Xugong Qin\textsuperscript{1,2}, Enze Xie\textsuperscript{1}, Weiping Wang\textsuperscript{1}

\textsuperscript{1}Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China  
\textsuperscript{2}School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China  
\{guoyouhui, zhouyu, qinxugong, wangweiping\}@iie.ac.cn, xez18@mails.tsinghua.edu.cn

ABSTRACT

Recent scene text detection methods are almost based on deep learning and data-driven. Synthetic data is commonly adopted for pre-training due to expensive annotation cost. However, there are obvious domain discrepancies between synthetic data and real-world data. It may lead to sub-optimal performance to directly adopt the model initialized by synthetic data in the fine-tuning stage. In this paper, we propose a new training paradigm for scene text detection, which introduces an \textit{UN}supervised \textit{I}ntermediate \textit{T}raining \textit{S}tage (UNITS) that builds a buffer path to real-world data and can alleviate the gap between the pre-training stage and fine-tuning stage. Three training strategies are further explored to perceive information from real-world data in an unsupervised way. With UNITS, scene text detectors are improved without introducing any parameters and computations during inference. Extensive experimental results show consistent performance improvements on three public datasets.

Index Terms— Scene text detection, domain discrepancies, unsupervised training

1. INTRODUCTION

Scene text detection is a fundamental and crucial task in computer vision because it is an important step in many practical applications, such as scene text recognition\cite{1, 2}, image/video understanding, and text visual question answering\cite{3}.

In recent years, deep learning based methods have been the mainstream in scene text detection, which require a large amount of data for training. However, labeling large sets of training images is extremely expensive and time-consuming. Most methods adopt synthetic data\cite{4}, which is easy to obtain and the annotations can be freely and automatically generated, to pre-train the models before fine-tuning on real-world data for better performance as shown in Figure 1 (a). Text instances are born with large variations in shapes, colors, fonts, sizes, orientations, and blend naturally with the background, resulting in a large domain discrepancy between synthetic data and real-world data. It may be sub-optimal to initialize the model directly using the pre-trained parameters on synthetic data when fine-tuning on real-world data.

Most of the existing methods are dedicated to designing a better model, while few consider the data inconsistency between the pre-training stage and fine-tuning stage. Since 2017, the “pre-train and fine-tune” paradigm dominates in natural language processing (NLP) tasks. Recently, a new paradigm dubbed “pre-train, prompt, and predict” is proposed in which downstream tasks are reformulated to look more like those solved during pre-training with the help of a textual prompt. Similarly, we propose a new training paradigm for scene text detection which includes an unsupervised intermediate training stage (UNITS) that can bridge the pre-training stage and fine-tuning stage and introduce information from real-world data into the pre-trained model. Inspired by the semi-supervised learning\cite{5, 6, 7}, we design several unsupervised training strategies that only use the synthetic data and the unlabeled real-world data in UNITS.
Specifically, for each unlabeled real image, we augment it to obtain an augmented version, and use the prediction of the original image to generate pseudo labels to supervise the prediction of the augmented image. In this way, the pre-trained model perceives the text information from the real-world data and provides a better initialization for the fine-tuning stage. Meanwhile, synthetic data is used to maintain the detection capability of the pre-trained model. Our proposed training pipeline with UNITS is illustrated in Figure 1 (b).

The proposed training paradigm is not constrained to one specific scene text detection model. We conduct extensive experiments on several classic scene text detection methods, e.g., EAST [8], PSENet [9], PAN [10] and DB [11] to verify the effectiveness and the generalization of the proposed method. Consistent performance improvements are obtained across different datasets and different scene text detectors.

The contributions of this work are summarized as follows:

- A new training paradigm is proposed for scene text detection which includes an unsupervised intermediate training stage that can bridge the pre-training stage and fine-tuning stage, and no extra computations or model parameters are involved in inference.
- In the unsupervised intermediate training stage, we explore a class of unsupervised training strategies that only use synthetic data and unlabeled real-world data.
- Experiments show that our method can appreciably improve the performance of several scene text detectors, e.g., DB [11] (83.8%→85.1%), EAST [8] (85.3%→86.5%), PSENet [9] (81.2%→82.1%), and PAN [10] (82.5%→82.9%) on ICDAR2015. Similar performance improvements can also be achieved on TotalText and MSRA-TD500 datasets.

2. RELATED WORK

With the revival of deep learning, the majority of recent scene text detectors are based on deep neural networks. These methods can be roughly classified into two categories: bottom-up methods and top-down methods.

Bottom-up methods detect the fundamental elements and group these elements into the final detection results. Some methods treat scene text detection as a semantic segmentation task and predict some auxiliary information to better differentiate pixels belonging to different text regions. PSENet [9] proposes a progressive scale expansion algorithm that can separate the dense text instances and detect text instances with arbitrary shapes. DB [11] performs the binarization process in a segmentation network with a differentiable binarization module to simplify post-processing. Some methods [12, 13, 14, 15] first detect individual text parts or characters, and group them into texts with some special post-processing steps. CRAFT [14] detects the text instances by exploring each character and affinity between characters. DRRG [15] constructs each text instance by a series of ordered rectangular components and utilizes graph convolution network to reason the relations of those components.

Top-down methods [16, 17, 8, 18] usually follow the general object detection methods and directly output the word/line-level detection results. TextBoxes++ [16] utilizes quadrilateral regression and RRPN [17] proposes rotation region proposal to detect multi-oriented text. Differently, EAST [8] is an anchor-free method that directly detects the quadrangles of words in a pixel-level manner. However, most of them show limited representation for irregular shapes, such as curved shapes. Different from methods that focus on specific model design, we focus on the incoherence of the two stages of pre-training and fine-tuning caused by domain discrepancies between synthetic data and real-world data. The proposed method generalizes well to various text detection methods.

Domain adaptation aims to reduce the domain gap between training and testing data. There are also some methods [19, 20, 21, 22] to solve the domain adaptation problem in scene text detection. GA-DAN [21] converts a source-domain image into multiple images of different spatial views as in target domain. Wu et al. [22] aims at the serious domain difference between synthetic data and real-world data, and proposes a synthetic-to-real domain adaptation method for scene text detection, which transfers knowledge from synthetic data to real-world data. In this work, we focus on how to use unlabeled real-world data to improve the pre-trained model to obtain better initialization and final performance during fine-tuning.

3. PROPOSED METHOD

3.1. Overview

In the training process of most previous scene text detection methods, the model is first trained using synthetic data (pre-training stage) and is trained on the target dataset (fine-tuning stage) with the initialization of the pre-trained model. In this section, we introduce a new training paradigm that includes an unsupervised intermediate training stage which bridges the pre-training stage and fine-tuning stage. The whole training paradigm is illustrated in Figure 2. Next, we first introduce the unsupervised intermediate training stage in detail. At the end of this section, we introduce how UNITS bridges the pre-training stage and fine-tuning stage.

3.2. Unsupervised Intermediate Training Stage

Due to domain discrepancies between synthetic data and real-world data, directly applying a pre-trained model trained with synthetic data as initialization may not be optimal for the fine-tuning stage. To alleviate this, we explore three training strategies for the unsupervised intermediate training stage inspired by semi-supervised learning. With these training
 strategies, the pre-trained model can perceive text information from real-world data and provide better initialization compatible with the fine-tuning stage. The specific structures are shown in Figure 3, then we will introduce them separately.

3.2.1. Double Branches Single Supervision

The Double Branches Single Supervision (DBSS) consists of two branches with the same structure but different parameters. One branch inputs the unlabeled real image $X_u$ to get the prediction result $P1$, and the other branch inputs the augmented version of $X_u$ to get the prediction result $P2$. We use $P1$ to construct a new pseudo-label $Y1$ to supervise $P2$. For unlabeled real-world data, only the branch $f(\theta_2)$ performs parameter update. In the fine-tuning stage, we use the branch $f(\theta_2)$ to initialize the model and train specific epochs for different scene text detection detectors.

3.2.2. Double Branches Double Supervision

The Double Branches Double Supervision (DBDS) is a modification of DBSS. We hope that the branch $f(\theta_2)$ can obtain information from the real-world data in DBSS, then we consider whether we can make these two branches interactively promote each other to obtain a better $f(\theta_1)$ and $f(\theta_2)$. On the basis of DBSS, we swap $X_u$ and $X_u'$, and update the parameters of the branch $f(\theta_1)$. Since the two branches are completely symmetrical for training, we use the branch $f(\theta_1)$ or branch $f(\theta_2)$ to initialize the model in the fine-tuning stage.

3.2.3. Single Branch Single Supervision

The implementation of Single Branch Single Supervision (SBSS) refers to PseudoSeg [6] and UDA [25]. The input unlabeled real image $X_u$ is augmented to obtain the image $X_u'$, and both enter the model $f(\theta)$ to get the prediction results, then we use the prediction result of $X_u$ to construct a pseudo-label to supervise the prediction result of $X_u'$. If we equate $f(\theta_1)$ to $f(\theta_2)$ in DBDS, we can get SBSS. Because there is

4. EXPERIMENTS

4.1. Datasets

The datasets used for the experiments in this paper are briefly introduced below:

- **SynthText** consists of 800k synthetic images.
- **TotalText** is a dataset that includes horizontal, oriented, and...
Table 1. Detection results with different unsupervised training strategies and comparisons with previous methods. “P”, “R”, and “F” indicate precision, recall, and f-measure respectively. “*” means our implementation.

| Method                      | ICDAR2015 | MSRA-TD500 | TotalText |
|-----------------------------|-----------|------------|-----------|
|                            | P   | R   | F  | P   | R   | F  | P   | R   | F  |
| SegLink [23]                | 76.8| 73.1| 75.0| 86.0| 70.0| 77.0| 79.0| 77.0| 77.0|
| RRPN [17]                   | 82.0| 73.0| 77.0| 82.0| 68.0| 74.0| 82.0| 67.0| 77.0|
| TextRex++ [16]              | 87.8| 78.5| 82.9| -   | -   | -   | -   | -   | -   |
| TextSnake [13]              | 84.9| 80.4| 82.6| 83.2| 73.9| 78.3| 82.7| 74.5| 78.4|
| CRAFT [14]                  | 89.9| 84.3| 86.9| 88.2| 78.2| 82.9| 87.6| 79.9| 83.6|
| PSENet + STKM [24]          | 87.8| 84.1| 85.9| -   | -   | -   | 86.32| 78.4| 82.0|
| DB* [11]                    | 89.7| 78.7| 83.8| 91.5| 81.1| 86.0| 88.4| 82.2| 85.2|
| DB* (1800 epochs)           | 89.8| 79.0| 84.0| 89.2| 80.6| 84.7| 88.2| 81.2| 84.6|
| DB + SBSS                   | 89.2| 79.4| 84.0| 91.9| 82.3| 86.9| 88.7| 83.1| 85.8|
| DB + DBDS                   | 89.9| 79.4| 84.3| 91.5| 84.7| 88.0| 89.7| 82.0| 85.7|
| DB + DBSS                   | 89.6| 81.0| 85.1| 91.0| 85.2| 88.1| 88.6| 83.2| 85.8|

Table 2. Detection results with different detectors.

| Detectors | Backbone | DBSS | ICDAR2015 | |
|-----------|----------|------|-----------|----|
| DB [11]   | ResNet-50 | ×    | 89.7 | 78.7 | 83.8 |
|           |          | √    | 89.6 | 81.0 | 85.1 |
| EAST [8]  | ResNet-50 | ×    | 85.9 | 84.6 | 85.3 |
|           |          | √    | 87.5 | 85.4 | 86.5 |
| PSENet [9]| ResNet-50 | ×    | 86.4 | 76.6 | 81.2 |
|           |          | √    | 84.8 | 79.5 | 82.1 |
| PAN [10]  | ResNet-18 | ×    | 86.1 | 79.2 | 82.5 |
|           |          | √    | 87.2 | 79.0 | 82.9 |

Table 3. Ablation study of the magnitude of data augmentation used in UNITS on ICDAR2015.

| Augmentation Type | EAST | DB |
|-------------------|------|----|
|                   | P   | R   | F  | P   | R   | F  |
| Baseline          | 85.9| 84.6| 85.3| 89.7| 78.7| 83.8|
| Color Jitter      | 85.8| 84.7| 85.3| 89.5| 79.1| 83.9|
| Randon Rotate     | 87.5| 85.4| 86.5| 89.6| 81.0| 85.1|

4.3. Ablation Study

4.3.1. Different Training Strategies

To verify the effectiveness of the three different training strategies in UNITS, we conduct experimental verification on DB. Following the open-source project, we conduct experiments on ICDAR2015, TotalText and MSRA-TD500. Specifically, we train 600 epochs in UNITS, then use the trained model as the initialization, and finally fine-tune 1200 epochs on the real-world dataset. The experimental results are shown in Table 1. It is worth mentioning that the baseline uses branch $f(\theta_1)$ in DBSS to initialize the model in order to eliminate the effect of using synthetic data in UNITS. It can be seen that the three training strategies can bring performance improvements on three datasets, which verifies the effectiveness of our new training pipeline. The DBSS achieves the best performance among the three training strategies, so we adopt it as the default training strategy for the subsequent experiments.

Also, we fine-tune 1800 epochs for the baseline DB and the performances have not increased or even decreased due to overfitting, which validates that the performance improvements do not come from additional training epochs.

4.3.2. Data Augmentation

First, we verify whether weak augmentation or strong augmentation should be used in UNITS. Based on DB and EAST, we conduct experiments on ICDAR2015, and the training strategy is DBSS. The experimental results are shown in Table 3. It can be seen that the performance is basically unchanged.

1https://github.com/SakuraRiven/EAST
2https://github.com/MhLiao/DB
3https://github.com/whai362/pan_pp_pytorch
4curved text. It consists of 1,255 training images and 300 testing images.

ICDAR2015 contains 1000 images for training and 500 images for testing. The images are captured by Google Glass and the text incidentally appears in the scene.

MSRA-TD500 is an oriented dataset that includes English and Chinese text instances with a large aspect ratio in natural scenes. It contains 300 training images and 200 testing images. Following the previous methods [11, 26, 27], we include extra 400 images from HUST-TR400 for training.

4.2. Implementation Details

We experiment on multiple scene text detectors, including EAST, DB, PSENet and PAN. For simplicity, we use the pre-trained models provided by the open-source code of these methods. If the initial learning rate of the pre-training stage is $lr$, the initial learning rate of UNITS is set to $0.1 \times lr$. And if the training epoch of the fine-tuning stage is $epoch$, the training epoch of UNITS is set to $0.5 \times epoch$. In the fine-tuning stage, all training settings are the same as the original methods. We implement our method based on PyTorch framework and all the models run on GeForce RTX-2080Ti.

For simplicity, we use the configuration used in UNITS on ICDAR2015.
Table 4. The effectiveness of multiple data augmentations. “Single Augmentation” means random rotation, and “Multiple Augmentations” means random rotation, random crop, and random scale.

| Method          | TotalText | ICDAR2015 | MSRA-TD500 |
|-----------------|-----------|-----------|------------|
|                 | P         | R         | F          | P         | R         | F          |
| Baseline        | 88.4      | 82.2      | 85.2       | 89.7      | 78.7      | 83.8       | 91.5      | 81.1      | 86.0      |
| Single Augmentation | 88.6      | 83.2      | 85.8       | 89.6      | 81.0      | 85.1       | 91.0      | 85.2      | 88.1      |
| Multiple Augmentations | **88.8**  | **83.0**  | **85.8**   | **89.3**  | **80.7**  | **84.8**   | **92.4**  | **83.0**  | **87.4**  |

Table 5. Ablation study of single dataset v.s. multiple datasets in UNITS.

| Method          | TotalText | ICDAR2015 | MSRA-TD500 |
|-----------------|-----------|-----------|------------|
|                 | P         | R         | F          | P         | R         | F          |
| Baseline        | 88.4      | 82.2      | 85.2       | 89.7      | 78.7      | 83.8       | 91.5      | 81.1      | 86.0      |
| Single Dataset  | 88.6      | 83.2      | 85.8       | 89.6      | **81.0**  | **85.1**   | 91.0      | **85.2**  | **88.1**  |
| Multiple Datasets | **90.9**  | **81.6**  | **86.0**   | **90.8**  | 79.5      | 84.8       | **91.6**  | **82.1**  | **86.6**  |

When weak augmentation (color jitter) is adopted. Because the input images of the two branches are basically the same and the output is almost the same, the provided supervision is roughly equivalent to supervision from pre-training with synthetic data alone. When strong augmentation (random rotation) is used, it can bring performance improvement of 1.2% and 1.4% on EAST and DB respectively.

Then we explore the role of multiple data augmentations. On the basis of random rotation, we add two additional data augmentations: random crop and random scale. We conduct experiments on DB, and the experimental results are shown in Table 4. Both can bring close performance improvements on MSRA-TD500. The performances of single augmentation are slightly better than multiple augmentations on ICDAR2015 and TotalText. We use random rotation as the default augmentation setting.

4.3.3. Domain Disparities

To verify whether our method can alleviate the domain discrepancies, we directly test the models of pre-training and UNITS on the test dataset. The model of UNITS is consistently superior to the model of pre-training, e.g., MSRA-TD500 (39.9% → 44.6%), ICDAR2015 (49.5% → 52.6%), and TotalText (49.8% → 51.5%) with DBSS.

The domain adaptation methods are to achieve the highest possible performance in the target domain. We use the labeled real-world data in DBSS to approximate the best domain adaptation result which can achieve 83.9% in F-measure on MSRA-TD500. After fine-tuning, the F-measure increases to 86.5% which is 0.5% higher than baseline, and the slight performance improvement may come from strong data augmentation. However, the F-measure of DBSS increases from 44.6% to 88.1% after fine-tuning. It shows that better performance before fine-tuning does not mean better performance after fine-tuning. The domain adaptation methods migrate the model to the target domain as much as possible but lose the knowledge learned by pre-training on large-scale synthetic data. UNITS is a trade-off between them and serves as a good bridge between the pre-training and fine-tuning stages.

4.3.4. Multiple Datasets for UNITS

As shown in Figure 2, we explore the possibility of using other unlabeled data. We use the existing public training datasets (ART, MLT2017, MSRA-TD500, HUST-TR400, and ICDAR2015) to construct a large dataset for UNITS training, denoted as UnlabeledDataset. By training UNITS on UnlabeledDataset, we can get an initialization model that is directly used to fine-tune all target datasets, which can reduce the time required to retrain for each dataset. Also, we use DBSS in UNITS and experiment on DB. The results are shown in Table 5. More data can bring similar performance improvement for ICDAR2015 and TotalText, but the improvement is tiny for MSRA-TD500. Since MSRA-TD500 consists of mostly text instances with a large aspect ratio which occupy a small proportion in UnlabeledDataset.

4.4. Comparison on Different Scene Text Detectors

To further verify the effectiveness of our proposed method, we conduct experiments on different scene text detectors. Specifically, we use DBSS as the training strategy in UNITS and the experimental dataset is ICDAR2015. The experimental results are shown in the Table 2. With UNITS, UNITS achieve substantial improvements of 1.3%, 1.2%, 0.9%, and 0.5% in F-measure on DB, EAST, PSENet, and PAN respectively, which demonstrates that our method generalizes well across different detectors.

4.5. Comparisons with State-of-the-Art Methods

We also compare DB with UNITS with state-of-the-art methods and the results are shown in Table 1. Our method can achieve better performance among the previous methods on MSRA-TD500 and TotalText. Especially, in particular, our method achieves 88.1% in F-measure and is significantly higher than other methods on MSRA-TD500. For ICDAR2015, our method is lower than CRAFT [14] which adopts additional character-level supervision. Some qualitative results on DB are displayed in Figure 4. Compared with the baseline, our method can detect more easily missed texts and suppress some false positives similar to the text instances.
In this paper, we propose a simple and effective training paradigm for scene text detection in which an unsupervised intermediate training stage (UNITs) can bridge the pre-training stage and fine-tuning stage and introduce information from real-world data into the pre-trained model. The experiments conducted on various benchmarks verify the effectiveness of UNITs across various scene text detectors. In the future, we hope to explore more training strategies like unsupervised representation learning methods [28, 29, 30, 31] or incremental learning methods [32] in UNITs and apply our training paradigm to more types of text detection methods.

6. REFERENCES

[1] Zhi Qiao, Yu Zhou, and et al., “SEED: semantics enhanced encoder-decoder framework for scene text recognition,” in CVPR, 2020, pp. 13525–13534.
[2] Zhi Qiao, Yu Zhou, and et al., “Pimnet: A parallel, iterative and mimicking network for scene text recognition,” in ACM MM, 2021, pp. 2046–2055.
[3] Gangyan Zeng, Yuan Zhang, Yu Zhou, and Xiaomeng Yang, “Beyond OCR + VQA: involving OCR into the flow for robust and accurate textvqa,” in ACM MM, 2021, pp. 376–385.
[4] Ankush Gupta, Andrea Vedaldi, and Andrew Zisserman, “Synthetic data for text localisation in natural images,” in CVPR, 2016, IEEE.
[5] Xiaokang Chen, Yuhui Yuan, Gang Zeng, and Jingdong Wang, “Semi-supervised semantic segmentation with cross pseudo supervision,” in CVPR, 2021, pp. 2613–2622.
[6] Yuliang Zou, Zizhao Zhang, Han Zhang, Chun-Liang Li, Xiao Bian, Jia-Bin Huang, and Tomas Pfister, “Pseudoseg: Designing pseudo labels for semantic segmentation,” in ICLR, 2021.
[7] Xugong Qin, Yu Zhou, Dongbao Yang, and Weiping Wang, “Curved text detection in natural scene images with semi- and weakly-supervised learning,” in ICDAR, 2019, pp. 559–564.
[8] Xinyu Zhou, Cong Yao, He Wen, Yuzhi Wang, Shuchang Zhou, Weiran He, and Jiajun Liang, “EAST: an efficient and accurate scene text detector,” in CVPR, 2017, pp. 2642–2651.
[9] Wenhai Wang, Enze Xie, Xiang Li, Wenbo Hou, Tong Lu, Gang Yu, and Shuai Shao, “Shape robust text detection with progressive scale expansion network,” in CVPR, 2019, pp. 9336–9345.
[10] Wenhai Wang, Enze Xie, and et al., “Efficient and accurate arbitrary-shaped text detection with pixel aggregation network,” in ICCV, 2019, pp. 8439–8448.
[11] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xiang Bai, “Real-time scene text detection with differentiable binarization,” in AAAI, 2020, pp. 11474–11481.
[12] Yudi Chen, Yu Zhou, and et al., “Constrained relation network for character detection in scene images,” in PRICAI, 2019, pp. 137–149.
[13] Shangbang Long, Jiaqiang Ruan, and et al., “Textsnake: A flexible representation for detecting text of arbitrary shapes,” in ECCV, 2018, pp. 19–35.
[14] Youngmin Baek, Bado Lee, and et al., “Character region awareness for text detection,” in CVPR, 2019, pp. 9365–9374.
[15] Shi-Xue Zhang, Xiaobin Zhu, and et al., “Deep relational reasoning graph network for arbitrary shape text detection,” in CVPR, 2020, pp. 9696–9705.
[16] Minghui Liao, Baoguang Shi, and Xiang Bai, “Textboxes++: A single-shot oriented scene text detector,” in IEEE Trans. Image Process., vol. 27, no. 8, pp. 3676–3690, 2018.
[17] Jianqi Ma, Weiyuan Shao, and et al., “Arbitrary-oriented scene text detection via rotation proposals,” in TMM, vol. 20, no. 11, pp. 3111–3122, 2018.
[18] Youhui Guo, Yu Zhou, Xugong Qin, and Weiping Wang, “Which and where to focus: A simple yet accurate framework for arbitrary-shaped nearby text detection in scene images,” in ICANN, 2021.
[19] Gangyan Zeng, Yuan Zhang, and et al., “A cost-efficient framework for scene text detection in the wild,” in PRICAI, 2021.
[20] Yudi Chen, Wei Wang, and et al., “Self-training for domain adaptive scene text detection,” in IJCAI, 2020, pp. 850–857.
[21] Fangneng Zhan, Chuhui Xue, and Shijian Lu, “GA-DAN: geometry-aware domain adaptation network for scene text detection and recognition,” in ICCV, 2019, pp. 9104–9114.
[22] Weijia Wu, Ning Lu, Enze Xie, Yuxing Wang, Wenwen Yu, Cheng Yang, and Hong Zhou, “Synthetic-to-real unsupervised domain adaptation for scene text detection in the wild,” in ACCV, 2020, pp. 289–303.
[23] Baoguang Shi, Xiang Bai, and et al., “Detecting oriented text in natural images by linking segments,” in CVPR, 2017, pp. 3482–3490.
[24] Qi Wan, Haoqin Ji, and Linlin Shen, “Self-attention based text knowledge mining for text detection,” in CVPR, 2021, pp. 5983–5992.
[25] Qizhe Xie, Zihang Dai, and et al., “Unsupervised data augmentation for consistency training,” in NeurIPS, 2020.
[26] Xugong Qin, Yu Zhou, Youhui Guo, Dayan Wu, and Weiping Wang, “Fc2rn: A fully convolutional corner refinement network for accurate multi-oriented scene text detection,” in ICASSP, 2021, pp. 4350–4354.
[27] Xugong Qin, Yu Zhou, and et al., “Mask is all you need: Rethinking mask R-CNN for dense and arbitrary-shaped scene text detection,” in ACM MM, 2021, pp. 414–423.
[28] Dezhaon Luo, Chang Liu, and et al., “Video cloze procedure for self-supervised spatio-temporal learning,” in AAAI, 2020, pp. 11701–11708.
[29] Yuan Yao, Chang Liu, Dezhaon Luo, Yu Zhou, and Qixiang Ye, “Video playback rate perception for self-supervised spatio-temporal representation learning,” in CVPR, 2020, pp. 6547–6556.
[30] Xiaoni Li, Yu Zhou, and et al., “Dense semantic contrast for self-supervised visual representation learning,” in ACM MM, 2021, pp. 1368–1376.
[31] Dezhaon Luo, Yu Zhou, and et al., “Exploring relations in untrimmed videos for self-supervised learning,” ACM TOMM, vol. 18, no. 1s, pp. 1–21, 2022.
[32] Dongbao Yang, Yu Zhou, and et al., “Rd-iod: Two-level residual-distillation-based triple-network for incremental object detection,” ACM TOMM, vol. 18, no. 1, pp. 1–23, 2022.