This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3262692

Unsupervised Domain Adaptation via Class Aggregation for Text Recognition

Xiao-Qian Liu, Xue-Ying Ding, Xin Luo, Xin-Shun Xu*

***Abstract*—Cross-domain text recognition is a very challenging task due to the domain drift problem. One solution is aligning feature distributions between domains through Unsupervised Domain Adaptation (UDA). All existing methods perform feature alignment based on the whole image or semantic character features. However, visual character features without contextual semantics also contain much valuable information, *e.g.*, stroke features of individual characters, which also benefits domain transfer. To this end, we propose a dual intra-Class Aggregation based unsupervised Domain Adaptation method (CADA) for text recognition, which aligns both visual and semantic character feature distributions. To our knowledge, CADA is the first to consider visual character features without contextual semantics in cross-domain text recognition tasks. Accordingly, a Single-head Self-Attention (SSA) mechanism is introduced for extracting visual character features. Thereafter, a dual intra-class aggregation strategy is designed, which performs class aggregations in both visual and semantic spaces. We test the proposed method on widely-used datasets by combining it with representative text recognition models with various decoding methods. Extensive experimental results demonstrate the superiority and generality. Moreover, there is no additional inference time introduced compared to the baselines.**

***Index Terms*—Unsupervised Learning; Domain Adaptation; Text Recognition; Class Aggregation.**

I. Introduction

CURRENTLY, deep learning based text recognition methods have made great progress [1], [2], [3]. Although text recognition in single-domain can achieve good performance [4], [5], [6], it is still much challenging in cross-domain due to domain gaps, such as variations in strokes, fluency, appearance, and background. Consequently, a model may perform well in a source domain but poorly in a target domain, known as domain drift problem [7]. For example, as shown in Fig. 1, there are domain discrepancies among synthetic text, real scene text, and handwritten text, which is a typical domain drift phenomenon. Generally, these domains have different low-level feature distributions, such as color and pixel features, and similar high-level feature distributions, such as category and target type.

One way to solve the domain drift is to fine-tune a pre-trained model with labeled target data. For example, Ayan *et al.* [8] proposed a unified model for scene text and handwriting text recognition based on knowledge distillation, which is a supervised domain adaptation method. However, in many scenarios, less labeled data is available in some target domains; moreover, image annotation is time-consuming and labor-intensive. Therefore, more recently, unsupervised domain adaptation has been proposed, which transfers knowledge learned in source domains to unlabeled target domains to alleviate domain drift. For instance, SMILE [9], a UDA-based method, optimizes source data by supervised cross-entropy and target data by unsupervised entropy minimization, respectively. However, it ignores the interaction of source and target domains, and its performance decreases when the two domains differ significantly.

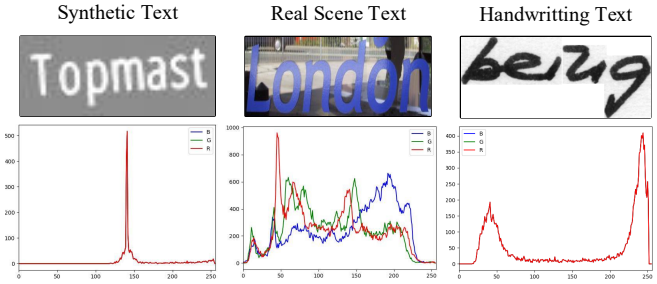


Fig. 1. Multiple domain samples. They have different low-level feature, such as RGB feature below each image, and the same high-level feature, such as both for the text recognition task.

Although several UDA-based methods have been proposed for text recognition, they ignore that text recognition is a sequence recognition task. The premise of correct sequence recognition is the accurate recognition of each character. To address this, ASSDA [10] is proposed, considering global and local-level features via adversarial learning. However, it only extracts contextual semantic local-level character features. It neglects visual character features, such as stroke and character structure, which contain valuable appearance information and are beneficial to accurately recognizing individual characters.

To address these issues mentioned above, we propose a novel intra-**C**lass **A**ggregation based unsupervised **D**omain **A**daptation method named CADA, which performs dual intra-class aggregation of character features in both visual space and semantic space. Specifically, it first adopts cross-entropy to optimize word-level labeled data and entropy minimization to optimize unlabeled target data. Then, a single-head self-attention module is introduced to extract visual character features without contextual semantics in visual space. Correspondingly, sequence-to-sequence cross attention is employed

This work was supported in part by: (1) National Natural Science Foundation of China (Grant No. 62172256, 62202278, and 62202272); (2) Natural Science Foundation of Shandong Province (Grant No. ZR2019ZD06 and ZR2020QF036), and in part by the Quan Cheng Laboratory, Jinan, China. (*Corresponding atuhor: Xin-Shun Xu.*)

X.-Q. Liu, X.-Y. Ding, X. Luo, and X.-S. Xu are with the School of Software, Shandong University, Jinan 250101, China (e-mail: jlrxqxq370322@126.com; 202015188@mail.sdu.edu.cn; luoxin.lxin@gmail.com; xuxinshun@sdu.edu.cn).

in semantic space to extract semantic character features. Finally, based on the visual character features and semantic character features, a dual intra-class aggregation is conducted, where similar character features from source and target domains are pulled close within classes. In this way, it can extract domain-invariant fine-grained features, thus alleviating the domain drift problem. It is tested on several representative text recognition models with various decoding ways. Extensive domain adaptation experiments demonstrate that CADA achieves state-of-the-art average results on benchmark datasets and can achieve competitive results compared to a supervised finetune model.

To summarize, our contributions are as follows:

- We propose an unsupervised Domain Adaptation method based on intra-Class Aggregation for text recognition, alleviating the domain drift problem.
- The visual character features without contextual semantics are first exploited in the UDA-based text recognition task. Based on this, a single-head self-attention module is introduced to extract visual character features.
- A dual intra-class aggregation strategy is designed based on a center loss, which performs visual and semantic space class aggregation, respectively.
- Extensive experiments are conducted on widely-used benchmark datasets. The results demonstrate the superiority and generality of CADA over some state-of-the-art methods.

II. Related Works

This section reviews the literature on deep learning based text recognition, unsupervised domain adaptation, and domain adaptation for text recognition.

A. Deep Learning based Text Recognition

In the past decade, encode-decode based deep learning models have made much progress on text recognition [11], [12]. Some adopt the connectionist temporal classification (CTC) [13] layer, making end-to-end sequence discriminative learning possible. For example, Shi *et al.* [14] proposed the CRNN framework based on CTC loss for scene text recognition. Subsequently, the CTC-based methods are gradually replaced by attention-based methods [7]. For example, Baek *et al.* [12] proposed a four-stage model, namely TRBA, and experimentally demonstrated that the attention-based decoding method was superior to the CTC-based decoding method, but was less efficient than the latter due to the autoregressive decoding. Based on the joint modeling of visual and semantic features, Bhunia *et al.* [15] proposed a multi-stage and multi-scale 2D-attention model. However, it is also an attention-based decoding model, making it less computationally efficient. Considering the speed of CTC-based decoding, Hu *et al.* [16] proposed a GTC model. In the training process, the attention-based decoding branch is utilized to guide the training of the CTC-based decoding branch, but only the CTC branch is used in the inference stage. More recently, to address the parallel limitation of attention-based methods, many transformer-based recognition models are proposed [17], [18]. For instance, MASTER [19] is a CNN-transformer based model which extracts global information and features of different spaces. In addition, a novel memory caching mechanism eliminates unnecessary calculations and saves intermediate calculation results, thus improving inference speed. Fang *et al.* [17] innovatively proposed a language-based model, which optimizes a visual model and a language model separately by blocking the transmission of the gradient flow. The methods mentioned above have achieved competitive results; however, they mainly focus on single-domain text recognition or only leverage global features while ignoring local features.

B. Unsupervised Domain Adaptation

Unsupervised domain adaptation has gained increasing attention in recent several years [20], [21], which aims to obtain a good performance model on unlabeled target data by mitigating the domain drift. Existing UDA-based methods can be divided into three categories. The first one is statistics-based, which maps data from source and target domains into a shared space where domain alignment is achieved by minimizing measurements, *e.g.*, Maximum Mean Discrepancy (MMD) [22], correlation alignment distance (CORAL) [23], [24]. The second category is adversarial learning based methods [25], [26]. For example, Zhang *et al.* [27] proposed a collaborative and adversarial network CAN, where an adversarial training scheme is used to learn both discriminative low-level representations and high-level representations. Coupled GANs [26] directly applies GANs to domain adaptation to explicitly reduce the domain drift by learning a joint distribution of multi-domain images. The third category is self-training based methods, which use pseudo-labeled target samples to retrain the model through specific training strategies. For instance, Huang *et al.* [28] constructed a categorical domain-mixed dictionary from the labels of the source domain and the pseudo labels of the target domain and proposed a novel Category Contrast technique (CaCo) that introduces semantic priors on top of instance discrimination for visual UDA tasks. Zou *et al.* [29] introduced a class-balanced self-training UDA for semantic segmentation. Different from the above methods based on adversarial learning or self-training, our CADA aligns fine-grained category features from source and target domains in embedding space to learn domain-invariant features.

C. Domain Adaptation for Text Recognition

Due to the inter-domain variability, text recognition models trained on a source domain, *e.g.*, large-scale synthetic text, can hardly be applied directly to a target domain, *e.g.*, real scene text, or handwriting text. Meanwhile, in some scenarios, a target domain has insufficient labeled data. To address this problem, unsupervised domain adaptation methods for text recognition have attracted much attention recently. These methods can be roughly divided into two categories, *i.e.*, writing style adaptation and scene adaptation. Thereinto, the former treats an author as a domain that adapts writing styles among multiple authors [30], [31]. For example, MetaHTR [31] is a writer-adapted model via a single gradient step update during inference, which exploits additional new-writer text

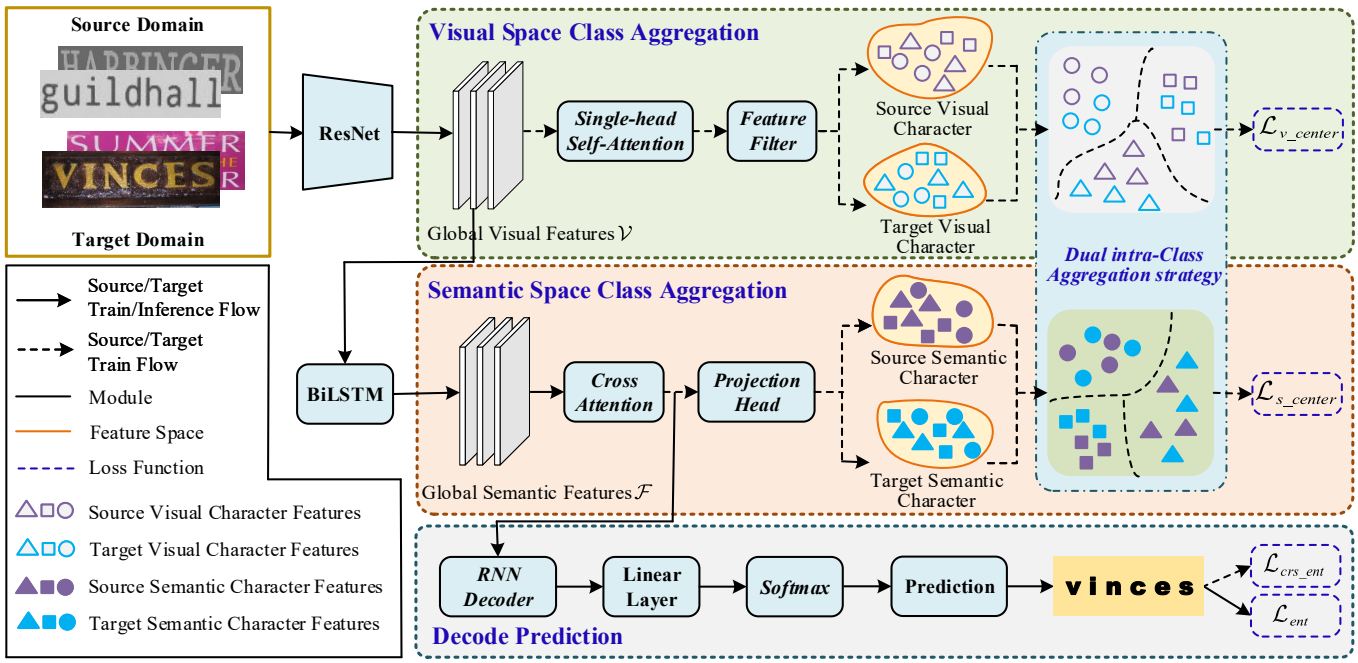


Fig. 2. The pipline of CADA. It consists of a ResNet to extract global visual features $\mathcal{V}$, a BiLSTM to capture global semantic features $\mathcal{F}$, a Decode Prediction module that autoregressively predicts sequences based on an RNN decoder, a visual space class aggregation module, and a semantic space class aggregation module. Thereinto, the intra-class aggregation in visual and semantic space constitutes the dual intra-class aggregation strategy. In the Decode Prediction module, the source data and target data are optimized using supervised cross-entropy $\mathcal{L}_{crs\_ent}$ and unsupervised entropy minimization $\mathcal{L}_{ent}$, respectively.

via a novel meta-learning framework. In contrast, the latter treats a scene as a domain that performs multiple scenes adaptation, typical methods including GA-DAN [32], ASSDA [33], and SMILE [9]. GA-DAN introduces a geometry-aware domain adaptation network to convert synthetic text images to real scene text images and then uses the converted text images to train the model for target domain recognition. However, it is a two-stage model instead of an end-to-end unified framework. ASSDA is an improved version of SSDAN [34], which uses adversarial learning to extract global and local character features, respectively, and realizes the domain adaptation of the synthetic text to real scene text. SMILE performs sequence-to-sequence unsupervised domain adaptation via entropy minimization for text recognition. However, it optimizes the source and target domains separately. More specifically, the source domain is optimized by supervised cross-entropy, and the target domain is optimized by unsupervised entropy minimization. In other words, it ignores the interaction between the source and target domains; therefore, its performance on the target domain degrades when the distributions of source and target domains differ significantly. Unlike these UDA-based methods, our method starts from the task itself, where the correct sequence recognition depends on the accurate recognition of each character. For this purpose, CADA extracts visual and semantic character features and conducts the alignment interaction between source and target domains in visual space and semantic space, respectively.

III. Our Method

A. Problem Definition

Our work aims to achieve domain adaptation for sequence-to-sequence text recognition in cross-domains. In other words, the labeled source data and unlabeled target data are used to learn domain-invariant feature representations so that the model can perform well in target domains. Formally, given N^s word-level annotated source samples $\mathcal{D}^S = \{(x_i, y_i)\}_{i=1}^{N^s}$ and N^t unlabeled target samples $\mathcal{D}^T = \{(x_j)\}_{j=1}^{N^t}$, where x_i or x_j is an image containing a text sequence, and y_i is a word-level label $y_i = \{y_{i,1}, y_{i,2}, \cdots, y_{i,L}\}$, where L is the actual length of the text sequence. The task aims to learn a model on $\mathcal{D}^S$ and $\mathcal{D}^T$ that performs well on $\mathcal{D}^T$.

B. Overview

The full framework of CADA is shown in Fig. 2, which comprises five modules, *i.e.*, ResNet, BiLSTM, decode prediction, visual space class aggregation, and semantic space class aggregation.

More specifically, given the input image x, ResNet first extracts global visual features $\mathcal{V}(x) = [v_1, v_2, \cdots, v_T] \in \mathbb{R}^{T\times D}$, where T is the maximum decoding length, and D is the feature dimension. Then, a two-layer BiLSTM is adopted to capture the global semantic features for sequence modeling, denoted as $\mathcal{F}(x) = [f_1, f_2, \cdots, f_T] \in \mathbb{R}^{T\times D}$. Last, the semantic features $\mathcal{F}$ is decoded to a text sequence prediction by an RNN decoder. During training, the source and target samples are optimized individually using supervised cross-entropy and unsupervised entropy minimization.

Our proposed visual space class aggregation and semantic space class aggregation constitute a dual intra-class aggregation strategy. The visual space class aggregation module is employed for clustering visually similar characters with pseudo-labels from the source and target domains. The semantic space class aggregation module is used to aggregate pseudo-labeled semantic characters from the source and target domains.

In the following subsections, we first introduce the main parts of the decode prediction module, *e.g.*, the RNN decoder and the entropy minimization, and then the proposed visual space class aggregation and semantic space class aggregation, respectively.

C. Decode Prediction

As shown in Fig. 2, the RNN first autoregressively decodes the attention-based semantic features in the decode prediction module. Then, the unlabeled target data are optimized by unsupervised entropy minimization.

1) ***RNN Decoder****:* The RNN decodes the attention-based sequence features. At decoding time-step t, the representation of the most relevant part to the character y_t of the semantic features $\mathcal{F}$ is defined as a context vector g_t,

$$g_t = \sum_{i=1}^{T} \alpha_{t,i} f_i \,, \tag{1}$$

where $f_i \in \mathbb{R}^D$ is the i-th semantic feature, and $\alpha_{t,i} \in (0, 1)$ is attention weight, which is calculated as follows,

$$\alpha_{t,i} = \frac{\exp(e_{t,i})}{\sum_{j=1}^{T} \exp(e_{t,j})} \,, \tag{2}$$

where the attention score $e_{t,i}$ indicates the attention of the decoding character y_t to the i-th sequence feature. Specifically, $e_{t,i}$ is defined as,

$$e_{t,i} = \omega^T \tanh\left(\mathcal{W}_s s_{t-1} + \mathcal{W}_f \mathcal{F} + b\right), \tag{3}$$

where ω, $\mathcal{W}_s$, $\mathcal{W}_f$, and b are trainable parameters, and s_{t-1} is the hidden state of the RNN at time t-1.

Then, the current hidden state s_t is updated as follows,

$$s_t = RNN\left(s_{t-1}, y_{t-1}, g_t\right), \tag{4}$$

where y_{t-1} is the one-hot encoding at time t-1. In our method, y_{t-1} is one-hot encoding of the label for source domain and the prediction for target domain.

Thereafter, the probability of the current predicted character y_t is computed by,

$$p(y_t|x) = softmax\left(\mathcal{W}_o s_t + b_o\right), \tag{5}$$

where $\mathcal{W}_o$ and b_o are trainable parameters of a linear layer.

2) ***Entropy Minimization****:* Entropy minimization, known as entropy regularization, has been widely used in self-supervised and unsupervised learning. To enlarge the margins of features across different categories of characters, we adopt unsupervised entropy minimization to optimize the unlabeled target data. For a target sample $x_j \in \mathcal{D}^T$, each predicted character y_t has an entropy value. Then, the entropy value of a predicted text sequence is the sum of the entropy value of each character. Thus, the entropy minimization of a target domain is defined as follows,

$$\mathcal{L}_{ent} = \frac{1}{N^t} \sum_{j=1}^{N^t} \sum_{t=1}^{T} -p(y_{j,t}|x_j) \log p(y_{j,t}|x_j) \,, \tag{6}$$

where $p(y_{j,t}|x_j)$ is the predicted probability of each character after the softmax function, N^t is the number of target samples, and T is the pre-defined maximum decoding length.

It is worth noting that since the target domain has no supervised information, entropy minimization could suffer from overconfidence, even if the predictions are incorrect. To mitigate this problem, the target data is optimized based on a pre-trained supervised *Baseline* model, which is trained only on source data by cross-entropy,

$$\mathcal{L}_{crs\_ent} = \frac{1}{N^s} \sum_{i=1}^{N^s} \sum_{t=1}^{T} -\log p(y_{i,t}|x_i) \,, \tag{7}$$

where $p(y_{i,t}|x_i)$ is the predicted probability of each character after the softmax function, and N^s is the number of source samples.

D. Visual Space Class Aggregation

As mentioned previously, besides considering the contextual semantics, the accurate recognition of each character is also crucial for a sequence recognition task such as text recognition. Moreover, the visual features of each character may contain valuable information, such as stroke features and appearance features, which are essential for the recognition of individual characters. Therefore, extracting and processing the visual character features becomes critical to the text recognition task. For this purpose, we propose the visual space class aggregation module, mainly composed of a newly introduced single-head self-attention submodule, feature filter, and intra-class aggregation with a visual space center loss. Thereinto, the single-head self-attention module extracts fine-grained visual character features. After that, the low-confidence characters are filtered out in the feature filter step. Accordingly, intra-class aggregation in visual space with a center loss is performed based on these visual character features. In this way, the domain drift problem is mitigated by fine-grained alignment of visual character features.

1) ***Single-head Self-Attention (SSA)****:* To extract fine-grained visual character features from the global visual features $\mathcal{V}$, we introduce a single-head self-attention module according to the multi-head self-attention (MSA) [35], which maps the global visual features $\mathcal{V}$ to the enhanced visual features $\mathcal{A}_\mathcal{V}$. Technically, as shown in Fig. 3, the global visual features $\mathcal{V} \in \mathbb{R}^{T\times D}$ are input to three independent linear transformations to obtain $Q \in \mathbb{R}^{T\times D}$, $K \in \mathbb{R}^{T\times D}$, and $V \in \mathbb{R}^{T\times D}$, respectively. Then, the enhanced visual features $\mathcal{A}_\mathcal{V} \in \mathbb{R}^{T\times D}$ are defined as follows,

$$\mathcal{A}_\mathcal{V} = softmax\left(\frac{QK^T}{\sqrt{D}}\right)V \,. \tag{8}$$

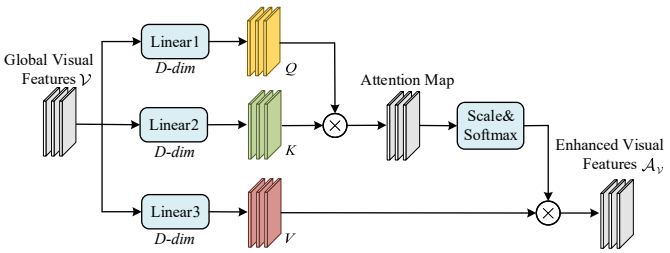


Fig. 3. The structure of SSA module. It maps the global visual features $\mathcal{V}$ to enhanced visual features $\mathcal{A}_\mathcal{V}$. $\otimes$ denotes pixel-wise multiplication.

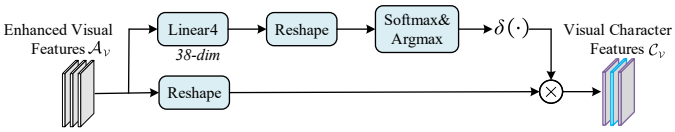


Fig. 4. The structure of feature filter module. It filters out characters with low confidence from the enhanced visual features $\mathcal{A}_\mathcal{V}$. $\delta(\cdot)$ is a threshold function. $\otimes$ denotes pixel-wise multiplication.

*2) **Feature Filter**:* Since no character-level annotation exists for the source and target domains, the extracted fine-grained visual character features are based on pseudo-label. This may lead to relatively inaccurate visual character features. To mitigate this problem, we introduce a feature filter module for filtering out visual character features that are not easily distinguishable, shown in Fig. 4. Intuitively, the character is easier to distinguish if the probability is higher. Technically, the enhanced visual character features $\mathcal{A}_\mathcal{V}$ are processed by a linear layer for classification output, and the maximum probability $p(y_t|x)$ can be obtained by *argmax*. Then, a threshold function $\delta(\cdot)$ is defined,

$$\delta(p) = \begin{cases} 1, & p(y_t|x) \geq \tau \\ 0, & otherwise \end{cases}, \tag{9}$$

where τ is a threshold parameters, $p(y_t|x)$ is the maximum probability after the softmax function.

Meanwhile, the enhanced visual character features $\mathcal{A}_\mathcal{V}$ are reshaped to obtain features $C'_\mathcal{V} = \{(c_i^\mathcal{V}, z_i^\mathcal{V})\}_{i=1}^{M'}$, where $z_i^\mathcal{V}$ is the pseudo-label of visual character feature $c_i^\mathcal{V}$, and M' is the number of visual characters before feature filter. Finally, based on the threshold function $\delta(\cdot)$, we get the more distinguished visual character features $C_\mathcal{V}$,

$$C_\mathcal{V} = C'_\mathcal{V} \otimes \delta(p), \tag{10}$$

where $\otimes$ denotes pixel-wise multiplication. The key idea is if the probability is greater than the threshold τ, the character feature $c_i^\mathcal{V}$ is retained; otherwise, it is discarded. That is, we obtain the set of visual characters $C_\mathcal{V} = \{(c_i^\mathcal{V}, z_i^\mathcal{V})\}_{i=1}^{M}$, where $z_i^\mathcal{V}$ is the pseudo-label of visual character feature $c_i^\mathcal{V}$, and M is the number of valid visual characters from the source and target domains after feature filter.

*3) **Visual Space Center Loss**:* Generally, whether in the source or target domain, the same characters always have similar visual features. We hope the character-specific visual knowledge learned from the source domain can be transferred to the target domain. For this purpose, we propose an intra-class aggregation step in visual space. More specifically, we cluster similar visual characters originating from the source and target domains with a center loss to make similar visual characters close in the embedding space. In this way, the character-specific knowledge learned from the source domain is transferred to the target domain by aligning visual characters to learn the domain-invariant fine-grained visual character features. Thereinto, the visual space intra-class aggregation based on visual character features is implemented by a center loss defined as follows,

$$\mathcal{L}_{v\_center} = \frac{1}{2M} \sum_{i=1}^{M} \|c_i^\mathcal{V} - \kappa_{z_i^\mathcal{V}}\|_2^2, \tag{11}$$

where $\kappa_{z_i^\mathcal{V}}$ is the class center of category $z_i^\mathcal{V}$ in visual embedding space, which is randomly initialized. The attention-based text recognition task has 38 class centers: 10 numbers, 26 case-insensitive letters, a start symbol ['GO'], and a stop symbol ['S'].

E. Semantic Space Class Aggregation

It is necessary to infer the contextual semantics of each character for the recognition of individual characters in sequence recognition tasks. For this purpose, we propose the semantic space class aggregation module as shown in Fig. 2, composed of cross attention, projection head, and intra-class aggregation with a semantic space center loss. Specifically, the cross attention locates the semantic character features. After that, the projection head maps the character features to a semantic embedding space. Similarly, the feature filter module is applied to obtain more distinguishable semantic character features. Finally, class aggregation with a center loss in semantic space is conducted. In this way, the domain migration could be mitigated by fine-grained alignment of semantic character features.

*1) **Cross Attention**:* Sequence-to-sequence cross attention described in III-C1 is utilized to extract semantic character features [33], [36], [37]. After the cross attention on global semantic features $\mathcal{F}(x) = [f_1, f_2, \cdots, f_T]$, the semantic character feature $\tilde{c}^\mathcal{F}$ can be directly denoted as,

$$\tilde{c}^\mathcal{F} \overset{def}{=} g_t = \sum_{i=1}^{T} \alpha_{t,i} f_i. \tag{12}$$

In this way, the character feature $\tilde{c}^\mathcal{F}$ is the weighted sum of all the semantic features $\mathcal{F}$.

As lexical dependence exists among characters, it can be encoded to the character feature representation. Thus, another available semantic character feature representation can be denoted as,

$$\tilde{c}^\mathcal{F} \overset{def}{=} s_t = RNN(s_{t-1}, y_{t-1}, g_t). \tag{13}$$

In this way, the character feature integrates the hidden state s_{t-1} of RNN, the output y_{t-1} at time t-1 (the ground truth for source domain and the prediction for target domain) and the weighted sum of all the semantic features $\mathcal{F}$.

*2) **Projection Head**:* To improve the robustness of the extracted character features, a projection head submodule is designed to map the character features to a semantic embedding space. More specifically, two different feature projection heads are created in our method.

The first one is *Linear Mapping*, where the transformed character features $c^{\mathcal{F}}$ is obtained through a linear layer, defined as follows,

$$c^{\mathcal{F}} = \mathcal{W}_c \tilde{c}^{\mathcal{F}} + b_c \,, \tag{14}$$

where $\mathcal{W}_c$ and b_c are trainable parameters of a linear layer.

The other one is *Identity Mapping*, which directly adopts the character features decoded by cross attention as the components of semantic space class aggregation, where $c^{\mathcal{F}} = \tilde{c}^{\mathcal{F}}$.

In addition, similar to the visual character feature filter, the feature filter scheme is also adopted to filter out semantic characters with low confidence. Finally, we get distinguishable semantic character features $C_{\mathcal{F}} = \{(c_i^{\mathcal{F}}, z_i^{\mathcal{F}})\}_{i=1}^{N}$, where $z_i^{\mathcal{F}}$ is the pseudo-label of semantic character feature $c_i^{\mathcal{F}}$, and N is the number of valid semantic characters from the source and target domains after feature filter.

*3) **Semantic Space Center Loss**:* The intra-class aggregation in visual space is based on visual character features; however, for sequences, each character also contains rich contextual semantics, which is critical for correctly recognizing sequences. Therefore, to align semantic character features, we similarly cluster similar semantic characters originating from the source and target domains. Semantic space intra-class aggregation is also implemented by a center loss. In this way, domain-invariant fine-grained character features can be learned to mitigate the domain drift problem. In specific, the semantic space intra-class aggregation based on semantic character features is defined as follows,

$$\mathcal{L}_{s\_center} = \frac{1}{2N} \sum_{i=1}^{N} \| c_i^{\mathcal{F}} - \rho_{z_i^{\mathcal{F}}} \|_2^2 \,, \tag{15}$$

where $\rho_{z_i^{\mathcal{F}}}$ is the class center of category $z_i^{\mathcal{F}}$ in semantic space, which is randomly initialized. Similar to center loss in visual space, there are 38 class centers in semantic space.

F. Overall Objective Function

Jointly considering the losses defined above, *i.e.*, supervised cross-entropy loss for the source domain, unsupervised entropy minimization loss for the target domain, center loss for visual space, and center loss for semantic space, we define the overall objective of our method as follows,

$$\mathcal{L} = \mathcal{L}_{crs\_ent} + \lambda_1 \mathcal{L}_{ent} + \lambda_2 \mathcal{L}_{v\_center} + \lambda_3 \mathcal{L}_{s\_center} \,, \tag{16}$$

where λ_1, λ_2, and λ_3 are trade-off parameters.

IV. Experiments

In this section, we first introduce the datasets and the experimental settings, including implementation details and evaluation metrics. Thereafter, we show the experimental results compared with the state-of-the-art methods, followed by some ablation experiments. Finally, we give some further analyses and visualizations.

A. Datasets

We conduct extensive experiments to validate the proposed CADA on widely-used benchmark datasets and our large-scale handwritten dataset.

1) Synthetic Text Datasets (Syn): Synth90k (MJ) [38] is a synthetic text dataset. It contains 8.9 million images generated from a set of 90k common English words, which are annotated with word sequences. SynthText (ST) [39] is another widely-used synthetic text dataset containing 5.5 million images with English words. In our experiments, MJ and ST are jointly used only for the training set of the source domain.

2) Real Scene Text Datasets: IIIT5k-Words (IIIT5K) [40] is crawled from Google image searches with query words such as 'billboards' and 'movie posters'. It contains 2000 cropped training images and 3000 cropped test images. Street View Text (SVT) [41] is collected from Google Street View. It contains 2000 training images and 3000 test images. ICDAR-2003 (IC03) [42] is a camera-captured text dataset. It contains 1156 cropped training images and 860 cropped test images. ICDAR-2013 (IC13) [43] contains 848 training images and 857 test images. SVT-Perspective (SVTP) [44] contains 645 test images, which are also collected from Google Street View but have many perspective texts. CUTE80 [45] is collected for curved text containing 288 cropped test images. ICDAR-2015 (IC15) [46] is collected by people who wear Google Glass. Consequently, it contains some perspective and blurry texts. It contains 4468 cropped training images and 1811 cropped test images.

These real scene text datasets are generally divided into regular scene texts, including IIIT5K, SVT, IC03, and IC13, and irregular scene texts, including SVTP, CUTE80, and IC15.

3) Handwritten Text Datasets: IAM [47] is a handwritten English text dataset written by 657 writers. It contains 1539 text pages, 13353 text lines, and 115320 words. According to the standard partition[1] [48], it includes 53841 training words, 8566 validation words, and 17616 test words, including numbers, upper and lower case letters, and special characters. In our experiments, special characters are filtered, and upper and lower case letters are case-insensitive. CVL [49] is a public dataset for writer retrieval, identification, and word spotting, written by 310 writers. It contains 12289 training words and 84949 test words. We also conduct experiments on our large-scale handwritten English text dataset (SDU-OM), written by 39920 writers. It contains 2,027k training words, 290k validation words, and 579k test words.

B. Experimental Settings

1) Implementation Details: For a fair comparison, our model adopts the same protocols following [12]. For example, a *Baseline* model is first trained with only labeled source data, serving as a pre-trained model. Thereafter, it can be fine-tuned or domain adaptive to reduce the domain gaps. The trade-off parameters λ_1, λ_2, and λ_3 in Eq. 16 are emperically set to $\{0.1, 0.0001, 0.00001\}$ on synthetic text to real scene text task, and $\{0.1, 0.0001, 0.0001\}$ on other tasks, respectively. We use

[1] https://fki.tic.heia-fr.ch/databases/iam-handwriting-database

TABLE I
THE WORD PREDICTION ACCURACY RESULTS OF CADA AND SOME STATE-OF-THE-ART METHODS ON DOMAIN ADAPTATION FROM SYNTHETIC TEXT TO REAL SCENE TEXT, INCLUDING BOTH REGULAR AND IRREGULAR DATASETS. PRI: PRIVATE DATASET, R: REAL SCENE TEXT.

	Methods	Reference	UDA	Labeled Train	Syn→RST				Syn→IST			Average
					IIIT5K	SVT	IC03	IC13	SVTP	CUTE80	IC15	
SOAT Methods	STAR-Net [50]	BMCV2016	N	MJ	87	86.9	94.4	92.8		71.7	76.1	
	RARE [51]	CVPR2016	N	MJ	86.2	85.8	93.9	92.6	~	70.4	74.5	~
	CRNN [14]	TPAMI2017	N	MJ	82.9	81.6	93.1	91.1	~	~	~	~
	GRCNN [52]	NIPS2017	N	MJ+PRI	84.2	83.7	93.5	90.9	~	~	~	~
	Char-Net [53]	AAAI2018	N	MJ	83.6	84.4	91.5	90.8	~	~	60	~
	TRBA [12]	ICCV2019	N	MJ+ST	87.9	87.5	94.9	93.6	79.2	74	77.6	~
	SSDAN [34]	CVPR2019	Y	MJ+ST	87.6	88.1	94.6	93.8	~	73.9	78.7	~
	ASSDA [33]	TIP2021	Y	MJ+ST	88.3	88.6	95.5	93.7	~	76.3	78.7	~
	SMILE [9]	arXiv2022	Y	MJ+ST	89.3	87.6	**96**	94.9	~	75.6	78.9	~
Ours	Baseline	~	N	MJ+ST	87.40	87.02	95.12	92.88	80.00	74.22	78.08	85.63
	CADA	~	Y	MJ+ST	**89.30**	**89.03**	95.35	**95.10**	**81.55**	**78.40**	**80.12**	**87.48**
	Finetune	~	N	MJ+ST+R	91.73	91.04	95.35	95.10	81.71	75.26	81.67	88.79

Adadelta as the optimizer, with a learning rate initialized to 0.1. The maximum number of training steps is set to 300k. If not specified, all experiments are conducted on an NVIDIA 2080Ti GPU with batch size 128.

2) Evaluation Metric: In the experiments, word accuracy is adopted to evaluate scene text recognition. In addition, we use *Average* to measure performance comprehensively, which is the word accuracy on all test datasets. Word Error Rate (WER) and Character Error Rate (CER) are commonly applied to measure the performance of handwritten recognition models. WER represents the proportion of words improperly recognized. CER is the Levenstein distance between the predicted character sequence and the ground truth sequence.

C. Domain Adaptation of Synthetic Datasets

In this section, we explore the domain adaptation of synthetic text to real scene text. The source domain is labeled synthetic text, MJ and ST. Following the protocol in [33], the target domain is unlabeled real scene text, *i.e.*, the union of training sets IIIT5K, SVT, IC13, and IC15. Further, the real scene text is divided into regular and irregular text. Therefore, two adaptation experiments are conducted: synthetic text to regular scene text (Syn→RST) and synthetic text to irregular scene text (Syn→IST).

Additionally, to verify the domain adaptation performance of CADA, we focus on unconstrained text recognition without any lexicon. The *Baseline* model is trained only with labeled source data and acts as a pre-trained model in all experiments. The *Finetune* model is trained on labeled source data and a small number of labeled target data.

1) Synthetic Text to Regular Real Text: Although the synthetic text is based on the style of scene text for synthesized data, there are still domain discrepancies between similar domains due to differences in illumination, background, *etc*. To demonstrate the adaptability of CADA between similar domains, we conduct domain adaptation experiments from synthetic text to regular scene text. The results are summarized in Table I, from which we can observe:

- Compared with the *Baseline*, CADA gains improvement on most regular datasets. The enhancement shows that CADA can use the knowledge learned in the source domain and transfer it to the target domain even if the target data has no supervised information.
- Compared with the state-of-the-art methods ASSDA [33] and SMILE [9], CADA obtains the best results, except for the IC03 dataset. One of the possible reasons is that the IC03 dataset is too simple for baseline alone to extract sufficient feature representation. Therefore, the performance through domain adaptation is restricted.
- CADA achieves comparable results compared with *Finetune* model. One reason is that it can transfer the knowledge learned in source data to target data; the other reason is that it can automatically learn domain-invariance features to reduce domain gaps.

2) Synthetic Text to Irregular Real Text: The experimental results of domain adaptation from synthetic text to irregular scene text (Syn→IST) are also shown in Table I. From the results on irregular scene text, we can find:

- Compared with the *Baseline*, CADA performs better on the three irregular scene text datasets. In particular, CADA achieves a 4.18% (74.22% *vs.* 78.40%) improvement on CUTE80.
- Compared with the state-of-the-art methods ASSDA and SMILE, CADA achieves much better results than them. The main reasons include that CADA considers the interaction between the source and target domains and optimizes the target domain using visual and semantic character features.
- CADA achieves comparable results to the *Finetune* model. It obtains better results on CUTE80 than the finetune model, further demonstrating the effectiveness of adaptation ability.

To summarize, CADA performs well on the tasks of domain adaptation from synthetic text to real scene text. It confirms the effectiveness of the proposed method.

D. Domain Adaptation of Cross-Domain Tasks

We further evaluate the performance of CADA on two cross-domain adaptation tasks, *i.e.*, handwritten text to real scene text and synthetic text to handwritten text.

1) Handwritten Text to Real Scene Text: In the task from handwritten text to real scene text (STR), the source domain

TABLE II
THE WORD PREDICTION ACCURACY RESULTS OF CADA, BASELINE AND FINETUNE ON THE DOMAIN ADAPTATION FROM HANDWRITTEN TEXT TO REAL SCENE TEXT, INCLUDING REGULAR SCENE TEXT AND IRREGULAR SCENE TEXT. *TEST REPRESENTS THE TEST RESULT ON THE HANDWRITTEN TEXT.

Domain	Model	*Test	Regular Scene Text				Irregular Scene Text		
			IIIT5K	SVT	IC03	IC13	SVTP	CUTE80	IC15
IAM→STR	Baseline	80.53	10.63	0.62	10.58	9.80	0.47	1.74	0.83
	CADA	~	13.20	1.55	19.42	16.92	1.40	2.44	2.54
	Finetune	~	72.83	62.13	77.44	76.08	48.68	34.5	57.43
CVL→STR	Baseline	86.51	1.00	0	0.47	0.12	0	0.35	0
	CADA	~	10.37	1.70	14.42	13.19	1.09	2.09	2.26
	Finetune	~	67.83	58.89	70.00	68.73	46.36	32.40	55.00
SDU-OM→STR	Baseline	93.27	8.53	0.31	6.16	5.37	0	3.14	0.77
	CADA	~	12.43	1.08	18.37	16.57	1.40	2.79	2.65
	Finetune	~	73.87	65.07	80.47	79.00	50.54	39.02	60.19

TABLE III
THE WORD ERROR RATE (WER) AND CHARACTER ERROR RATE (CER) RESULTS ON THE TASK FROM SYNTHETIC TEXT TO HANDWRITTEN TEXT.

Model	Syn→IAM		Syn→CVL		Syn→SDU-OM	
	WER	CER	WER	CER	WER	CER
Base [33]	54.30	28.41	~	~	~	~
SSDAN [34]	53.65	27.26	~	~	~	~
ASSDA [33]	**43.78**	19.96	~	~	~	~
Baseline	57.07	30.90	72.28	40.08	30.78	18.92
CADA	45.70	**19.67**	67.34	32.88	20.89	13.47
Finetune	15.96	6.02	18.41	7.23	15.03	11.65

is handwritten text, and the target domain is real scene text. We adopt three handwritten datasets, *i.e.*, IAM, CVL, and SDU-OM; therefore, there are three domain adaptation tasks from handwritten text to real scene text, *i.e.*, IAM→STR, CVL→STR and SDU-OM→STR. CADA is compared with the *Baseline* model and the *Finetune* model. Thereinto, the *Baseline* model is trained with the training set of handwritten text only and tested directly on the real scene text. The *Finetune* model is fine-tuned with a collection of real scene text training sets. In general, the source data of CADA is the handwritten text training set, and the target data is a collection of the real scene text training set. The results are summarized in Table II, from which we can observe:

- The *Baseline* model performs well on the test data of handwritten text, while its performance drops sharply on real scene text, indicating the discrepancy between the two domains and that the feature representation learned in handwritten text fails to generalize well to scene text.
- Compared to the *Baseline* model, however, CADA achieves performance improvements on the domain adaptation experiments for all three datasets, demonstrating the effectiveness of domain adaptation of CADA.
- The performance gap between CADA and the supervised *Finetune* model is apparent. Significantly, compared with the results in Table I, the performance of CADA, *Baseline* and *Finetune* deteriorates significantly, which demonstrates the hardness of domain adaptation in different domains. Just as stated in [8], exploring a general model for multiple domains is essential.

2) Synthetic Text to Handwritten Text: The experimental results of domain adaptation from synthetic text to handwritten text are shown in Table III. From the results, we can find:

- Compared with *Baseline* model, CADA improves on all tasks, further demonstrating that CADA can explore the potential inter-domain similarity between different domains.
- On the task of Syn→IAM, we find that CADA achieves about 8% performance improvement over SSDAN in terms of both WER and CER. Moreover, it achieves the best CER results among these three methods, while it is inferior to ASSDA in terms of WER.
- Similar to the results in Table II, CADA still has a severe gap compared to the supervised *Finetune* model, further demonstrating the hardness of cross-domain adaptation.

E. Generality of CADA

The proposed core modules can be deployed to most encoder-decoder-based text recognition methods. For example, the visual space class aggregation module can be applied to global visual features extracted by an encoder. The semantic space category aggregation module can be applied to contextual semantic features extracted by semantic modeling. In addition, the entropy minimization of the target domain can be applied to the prediction output after softmax. To validate this, we further perform experiments based on several representative text recognition models with different decoding ways, including RNN-based, transformer-based, and language-based models.

It is inappropriate to compare directly with these results in the original papers since many key settings differ from our task. Specifically, first, different training set types, *e.g.*, Scatter [55] utilized an additional synthetic dataset SA containing 1.2 million images; second, different training set versions, *e.g.*, Aster [54], and SRN [56] used an ST containing 7.3 million images, while the ST used in our method contains 5.5 million images; third, different test set versions, such as IC03 (860 *vs.* 867), IC13 (857 *vs.* 1015), and IC15 (1811 *vs.* 2077) all contain two versions (numbers in parentheses denote the number of samples in the test set). Therefore, for a fair comparison, we reproduced these methods based on the official code under the same training set and test set version, as shown in the *-Baseline of Table IV. In addition, we also reproduced the ABINet [17] with the same setting as in our method. The

TABLE IV
THE WORD PREDICTION ACCURACY RESULTS OF TEXT RECOGNITION METHODS BASED ON DIFFERENT DECODING WAYS. TRANS: TRANSFORMER; LAN: LANGUAGE; ST(7.3): ANOTHER VERSION OF ST CONTAINING 7.3 MILLION IMAGES; SA: A SYNTHETIC DATASET CONTAINING 1.2 MILLION IMAGES.

Methods	Decoder	Labeled Train	Syn-RST				Syn-IST			Average
			IIIT5K	SVT	IC03	IC13	SVTP	CUTE80	IC15	
Aster [54]	RNN	MJ+ST(7.3)	93.40	89.50	94.50	~	78.50	79.50	76.10	~
Aster-Baseline		MJ+ST	92.50	88.56	93.02	93.58	80.00	82.29	77.03	87.50
Aster-CADA		MJ+ST	92.70	87.94	92.44	93.12	81.40	81.94	79.07	88.00
Scatter [55]	RNN	MJ+ST+SA	93.70	92.70	~	~	86.90	87.50	~	~
Scatter-Baseline		MJ+ST	89.57	89.49	96.05	95.22	81.24	77.78	80.40	87.72
Scatter-CADA		MJ+ST	90.10	90.88	95.93	95.45	81.61	82.95	77.08	88.42
SRN [56]	Trans	MJ+ST(7.3)	94.80	91.50	~	95.50	85.10	87.80	82.70	~
SRN-Baseline		MJ+ST	94.60	91.34	93.95	95.22	82.79	83.33	80.07	89.60
SRN-CADA		MJ+ST	94.43	91.81	94.42	94.63	83.88	84.38	80.18	89.90
ABINet [17]	Trans+Lan	MJ+ST	96.20	93.50	~	97.40	89.30	89.20	86.00	~
ABINet-Baseline		MJ+ST	96.10	93.97	95.93	96.50	89.30	91.67	85.31	92.84
ABINet-CADA		MJ+ST	95.50	93.66	97.44	97.55	88.99	92.36	86.31	93.10
ABINet-CADA-single		MJ+ST	96.33	94.44	~	98.02	~	~	86.64	~

results of ABINet-Baseline are comparable to or even better than those published in the original paper. The results of these baselines provide a more fair basis to show the superiority of the core modules of our method. These baselines act as pre-trained models for *-CADA. From the results in Table IV, we can find that:

- Compared with *-Baseline, *-CADA obtains different degrees of improvement according to the *Average*. Specifically, it improves by 0.5%, 0.7%, 0.3%, and 0.26% on Aster, Scatter, SRN, and ABINet, respectively. This indicates that based on the global visual and semantic features extracted by baselines with varying capacities, the two-class intra-aggregation strategy can further extract fine-grained character features.
- Our method achieves a clear improvement on the task of synthetic text to irregular scene text. Specifically, our method is improved on 2-3 datasets over three irregular text datasets. The reason is that fine-grained visual features are more beneficial for recognizing individual characters in irregular text, such as curved text. In particular, SRN improves on all three datasets because its bidirectional inference provides more accurate global semantic features for intra-class aggregation in semantic space.

From Table IV, we can also observe that the performance of *-CADA decreases on several datasets. One reason may be that our optimization aims at the global optimum, *i.e.*, the *Average* metric that more comprehensively reflects the performance of the model. When the model achieves the global optimum, there may be cases that it is not optimal on some individual datasets. In addition, the performance degradation on some individual datasets mainly occurs on regular scene text. Another reason may be that with the goal of global optimization, the *Baseline* model is sufficient to extract visual and semantic features of regular text, making the model perform well on more challenging irregular scene text but limiting the performance on regular text. Following ASSDA, we conducted experiments where the target domain was a single scene dataset to validate the above conjecture, taking ABINet as an example. Due to the limitation of dataset availability, we performed experiments on the III5K, SVT, IC13, and IC15 datasets. From the ABINet-CADA-single in Table IV, we can see that our method can further improve the performance when the target domain is a single scene dataset. Benefiting from dual intra-class aggregation in visual and semantic space, the model can automatically transfer knowledge beneficial to the single dataset to improve recognition performance.

Overall, global performance can be improved by simply deploying off-the-shelf text recognition models with different decoding ways. These experimental results demonstrate the generality of our method.

F. Ablation Study

1) The Effect of Each Module: To sufficiently investigate the effect of entropy minimization $\mathcal{T}ent$, visual space class aggregation $\mathcal{V}center$, and semantic space class aggregation $\mathcal{S}center$, we conducted domain adaptation on two types of tasks: synthetic text to real scene text and handwritten text. Thereinto, the results of synthetic text to real scene text are summarized in Table V; those of synthetic text to handwritten text (SDU-OM) are shown in Table VI. We index all submodels and CADA from top to bottom in both tables as 1-8. From these results, we have the following observations:

- Model-2, model-3, and model-4 outperform model-1, demonstrating that each module is effective when applied individually to the adaptation. Meanwhile, the improvement of model-4 is the most significant, indicating that the entropy minimization can fully explore the feature representation in unsupervised learning.
- Compared with model-4, model-6 and model-7 also improve in most cases, including intra-class aggregation in visual and semantic space, respectively. It indicates that the class aggregation based on center loss can align feature distribution from source and target domains to transfer knowledge from source data to target data and improve model performance in the target domain.
- It is worth noting that CADA (model-8) achieves the best results on most datasets and the best average performance, as shown in Table V. In addition, it also achieves the best

TABLE V
THE WORD PREDICTION ACCURACY RESULTS OF DIFFERENT COMPONENTS ON THE DOMAIN ADAPTATION FROM SYNTHETIC TEXT TO REAL SCENE TEXT DATASETS, INCLUDING BOTH REGULAR SCENE TEXT AND IRREGULAR SCENE TEXT.

Model	$\mathcal{S}center$	$\mathcal{V}center$	$\mathcal{T}ent$	Syn→RST				Syn→RST			Average
				IIIT5K	SVT	IC03	IC13	SVTP	CUTE80	IC15	
1. Baseline	✗	✗	✗	87.40	87.02	95.12	92.88	80.00	74.22	78.08	85.630
2. +$\mathcal{S}center$	✔	✗	✗	88.80	88.56	95.12	93.40	80.62	78.40	78.80	86.789
3. +$\mathcal{V}center$	✗	✔	✗	88.03	89.18	95.58	94.63	81.24	78.05	78.58	86.617
4. +$\mathcal{T}ent$	✗	✗	✔	88.53	89.80	95.23	94.89	80.16	77.35	80.18	87.085
5. +$\mathcal{S}center$+$\mathcal{V}center$	✔	✔	✗	88.87	87.48	95.35	94.40	79.38	76.31	78.96	86.617
6. +$\mathcal{T}ent$+$\mathcal{S}center$	✔	✗	✔	89.10	88.56	95.47	94.87	79.85	78.05	80.07	87.196
7. +$\mathcal{T}ent$+$\mathcal{V}center$	✗	✔	✔	89.33	89.80	95.47	94.63	80.31	78.40	80.34	87.468
8. CADA	✔	✔	✔	89.30	89.03	95.35	95.10	81.55	78.40	80.12	87.480

TABLE VI
THE WORD ERROR RATE (WER) AND CHARACTER ERROR RATE (CER) RESULTS OF DIFFERENT COMPONENTS ON THE DOMAIN ADAPTATION FROM SYNTHETIC TEXT TO HANDWRITTEN TEXT.

Model	$\mathcal{S}center$	$\mathcal{V}center$	$\mathcal{T}ent$	Syn→SDU-OM	
				WER	CER
1. Baseline	✗	✗	✗	30.78	18.92
2. +$\mathcal{S}center$	✔	✗	✗	26.11	15.73
3. +$\mathcal{V}center$	✗	✔	✗	31.01	18.90
4. +$\mathcal{T}ent$	✗	✗	✔	21.02	14.14
5. +$\mathcal{S}center$+$\mathcal{V}center$	✔	✔	✗	26.04	15.90
6. +$\mathcal{T}ent$+$\mathcal{S}center$	✔	✗	✔	27.57	16.52
7. +$\mathcal{T}ent$+$\mathcal{V}center$	✗	✔	✔	20.98	13.80
8. CADA	✔	✔	✔	20.89	13.47

results in terms of both WER and CER on the task from synthetic text to handwritten text, as shown in Table VI, further demonstrating the effectiveness of different modules.

2) The Effect of Semantic Character Feature: As mentioned previously, we can adopt two schemes to extract semantic character features, *i.e.*, g_t and s_t. To evaluate whether semantic character feature can contribute to performance improvement or which type of character feature is more effective, we conduct adaptation experiments from synthetic text to real scene text on two submodels, *i.e.*, Baseline+$\mathcal{S}center$ (model-2 in Table V) and Baseline+$\mathcal{T}ent$+$\mathcal{S}center$ (model-6 in Table V). The results are shown in Table VII. We observe that:

- Character features extracted based on g_t are more valuable for intra-class aggregation in semantic space than those extracted based on s_t.
- The performance of models with g_t or s_t to extract character features is better than that of the *Baseline* model, demonstrating the effectiveness of leveraging the character features. It also shows the effectiveness of the semantic space class aggregation in CADA on domain knowledge transfer.

3) The Effect of the Projection Head: As mentioned previously, we can adopt two schemes to process the character features in the projection head module, *i.e.*, *Identity mapping* and *Linear mapping*. Thereinto, the first does not transform the features; in contrast, the latter adopts a fully-connected layer with dimension 256×256 to further transform the features. To evaluate which one is better, we also conduct experiments. The results are also incorporated into Table VII. We find that the *Identity mapping* has a slight advantage over the *Linear mapping*. It means that the *Identity mapping* is robust enough to extract character features. However, a linear transformation may overly modify the original embedding and weakens the effect of semantic space intra-class aggregation.

4) The Effect of Visual Space Class Center: Visual character features also contain appearance information and are important for text recognition. Maintaining the number of semantic space character class centers as 38, we further explore the effect of the number of visual space character centers. By default, the number of character centers in visual space is 38, *i.e.*, similar characters in the source and target domains share a common class center, defined as Shared. In addition, we tested the number of character centers in visual space is 76, *i.e.*, similar characters in the source and target domains have their class center, defined as Respective. As presented in Table VIII, the best result is obtained by generating a shared class center for each category. One of the main reasons is that *Respective* implements intra-domain clustering, while *Shared* enables cross-domain clustering, which is essentially an alignment of fine-grained features of the source and target domains of UDA.

G. Algorithm Analysis

1) Analysis of Parameter Sensitivity: We evaluate the sensitivity of the hyper-parameters, *i.e.*, λ_1, λ_2, and λ_3. The evaluation is conducted by changing one parameter while keeping the other hyper-parameters fixed. Specifically, we first explore the effect of λ_1 with different values, *i.e.*, $\{1, 0.1, 0.01\}$. Fig. 5(a) shows the model achieves the best results when $\lambda_1 = 0.1$. Thereafter, we explore the relative weights of class aggregation in visual space and semantic space, *i.e.*, λ_2 and λ_3. As shown in Table IX, the model achieves the best average result when $\lambda_2 = 0.0001$ and $\lambda_3 = 0.00001$. In addition, the model achieves better performance when λ_2 takes a larger value than λ_3. It demonstrates that the character features in visual space are more capable of facilitating intra-class aggregation, which also means that visual character features contain much valuable information to promote positive transfer. In addition, we also conduct parameter sensitivity experiments for the threshold τ. From Fig. 5(b), it can be seen that the performance of intra-class aggregation in visual space and semantic space can be better when $\tau = 0.3$ on balance.

2) Analysis of Model Complexity: We analyze the spatial and temporal complexity of several representative models in

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3262692

TABLE VII

THE WORD PREDICTION ACCURACY RESULTS OF DIFFERENT CHARACTER FEATURES EXTRACTING AND PROJECTION HEADS ON THE DOMAIN ADAPTATION FROM SYNTHETIC TEXT TO REAL SCENE TEXT DATASETS, INCLUDING BOTH REGULAR SCENE TEXT AND IRREGULAR SCENE TEXT.

Model	Projection	Char	Syn→RST			Syn→IST			Average
			IIIT5K	SVT	IC03	IC13	SVTP	CUTE80	IC15	
Baseline	~	~	87.40	87.02	95.12	92.88	80.00	74.22	78.08	85.630
Baseline+$\mathcal{S}center$	Identity	g_t	88.80	88.56	95.12	93.40	80.62	78.40	78.80	86.789
		s_t	88.07	87.79	95.58	94.28	79.69	78.40	77.86	86.209
	Linear	g_t	88.63	88.56	95.58	94.28	79.85	76.66	79.18	86.728
		s_t	88.20	88.25	94.77	94.40	78.92	77.35	77.69	86.086
Baseline+$\mathcal{T}ent$+$\mathcal{S}center$	Identity	g_t	89.10	88.56	95.47	94.87	79.85	78.05	80.07	87.196
		s_t	89.10	88.10	95.35	94.63	80.78	77.35	79.46	87.036
	Linear	g_t	89.00	88.72	95.35	94.75	80.62	78.75	79.90	87.196
		s_t	89.37	88.87	95.58	94.63	80.47	78.75	79.90	87.024

TABLE VIII

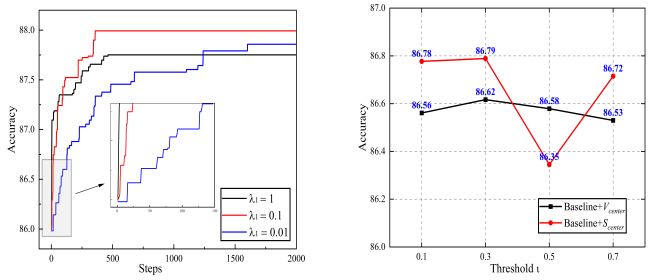
THE WORD PREDICTION ACCURACY RESULTS OF VISUAL CHARACTER FEATURES ON THE NUMBER OF CLASS CENTERS ON THE DOMAIN ADAPTATION FROM SYNTHETIC TEXT TO REAL SCENE TEXT DATASETS, INCLUDING BOTH REGULAR SCENE TEXT AND IRREGULAR SCENE TEXT.

Model	Centers	Syn→RST			Syn→IST			Average
		IIIT5k	SVT	IC03	IC13	SVTP	CUTE80	IC15	
Baseline+$\mathcal{V}center$	Shared	88.03	89.18	95.58	94.63	81.24	78.05	78.58	86.617
	Respective	88.27	88.41	95.35	93.58	80.78	77.70	78.24	86.382
CADA	Shared	89.30	89.03	95.35	95.10	81.55	78.40	80.12	87.480
	Respective	89.50	88.50	95.12	95.10	81.81	75.26	80.40	87.406

TABLE IX

THE WORD PREDICTION ACCURACY RESULTS OF DIFFERENT COMBINATION OF λ_2 AND λ_3 ON THE DOMAIN ADAPTATION FROM SYNTHETIC TEXT TO REAL SCENE TEXT DATASETS, INCLUDING BOTH REGULAR SCENE TEXT AND IRREGULAR SCENE TEXT. * REPRESENTS THE PARAMETERS USED IN OUR EXPERIMENTS.

Model	λ_2	λ_3	Syn→RST			Syn→IST			Average
			IIIT5K	SVT	IC03	IC13	SVTP	CUTE80	IC15	
CADA (λ_1=0.1)	0.0001	0.0001	89.00	88.25	95.47	94.40	80.78	78.05	80.51	87.258
	0.001	0.0001	89.37	88.72	95.23	94.75	81.40	77.70	79.96	87.357
	0.0001*	0.00001*	89.30	89.03	95.35	95.10	81.55	78.40	80.12	87.480



(a) λ_1 (b) τ

Fig. 5. The results of parameter sensitivity. (a): the model training accuracy with different λ_1 on the domain adaptation of synthetic text to real scene text. (b): the results of two variants with different threshod values.

TABLE X

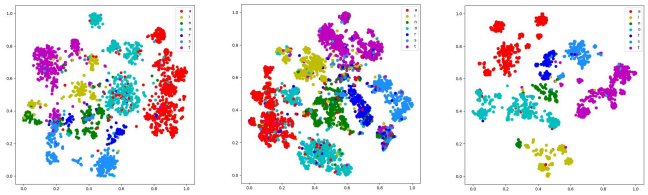
MODEL COMPLEXITY ANALYSIS RESULTS. ALL RESULTS ARE THE AVERAGE OF 5 RUNS WITH BATCHSIZE=48. TRANS: TRANSFORMER; LAN: LANGUAGE.

Methods	Decoder	Params ($\times 10^6$)	Memory (MiB)	Train (s)	Infer (ms)
Baseline	RNN	49.959	5349	33	0.84
CADA	RNN	49.959	4377	45	0.85
Scatter-CADA	RNN	120.529	8641	151	3.78
SRN-CADA	Trans	54.672	14429	73	1.71
ABINet-CADA	Trans+Lan	36.739	12053	75	1.75

terms of trainable parameters, memory footprint, training time, and inference time. For a fair comparison, we test all models with the same batch size on the same hardware. For the temporal complexity, we run five times to take the average. Specifically, the *training time* is the 100 iterations of each model, and the *inference time* is the average time over all the test datasets. As seen from Table X, CADA does not introduce additional inference time compared to *Baseline*. This is because the dual intra-class aggregation and entropy minimization are only introduced in the training phase and removed in the inference phase. With comparable training parameters, the memory consumption of the transformer-based model SRN-CADA is about 3.3x more than that of CADA, possibly due to the more layers of encoding and the increased dimensions of the features. In addition, the inference time of Scatter-CADA is 4.4x that of CADA due to its multiple decoding blocks.

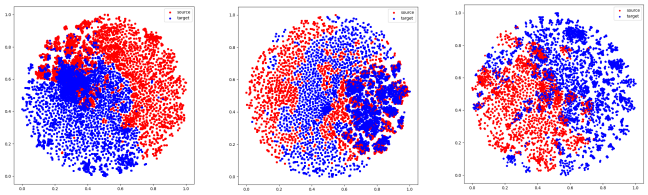
3) Visualization of the Feature Distribution: To further demonstrate the effectiveness of intra-class aggregation, we visualize the semantic character features using the t-SNE tool. Specifically, in synthetic text to real scene text, we randomly select several characters from the target domain to visualize the semantic character features of the *Baseline*, CADA, and *Finetune* models, respectively. As shown in Fig. 6, compared

This article has been accepted for publication in IEEE Transactions on Circuits and Systems for Video Technology. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TCSVT.2023.3262692



(a) Baseline (b) CADA (c) Finetune

Fig. 6. Visualization of semantic character features from target domain by t-SNE tool on synthetic text to real scene text task.



(a) Baseline (b) CADA (c) Finetune

Fig. 7. Visualization of semantic character features from source and target domains by t-SNE tool on synthetic text to handwritten text task.

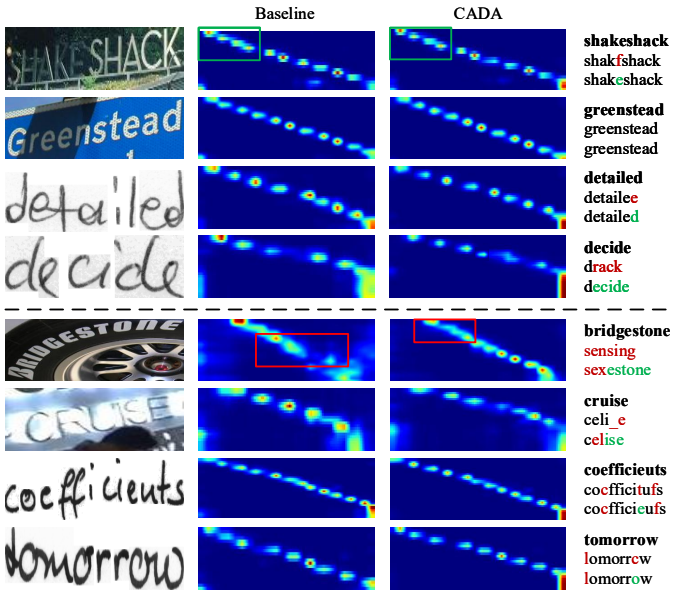


Fig. 8. Attention visualization and prediction reuslts. The top four rows are right cases, and the bottom four rows are failed cases. For the prediction column, the first row with bold font is the label; the second row is the prediction of baseline; the third row is the prediction of CADA. All reds denote failures, including boxes and fonts, and greens denote correct.

to the *Baseline*, CADA can more clearly distinguish the feature distribution of each character, which indicates that the semantic space class aggregation can well cluster the same characters from source and target domains.

Further, Fig. 7 illustrates the feature distribution of source and target data of *Baseline*, CADA, and *Finetune* models in the task of synthetic text to handwritten text. We can observe that, for *Baseline*, the distribution of target samples is more clearly bounded from the distribution of source samples. However, after domain adaptation by CADA, the two distributions are brought closer, making the target distribution indistinguishable from the source one.

4) Visualization of Attention Result: To illustrate the effectiveness of the attention scheme in CADA, we also visualize the attention results of each decoding time step in the semantic space. As shown in the first four rows of Fig. 8, CADA can locate fine-grained character features more precisely than the *Baseline* model, which means that it can further perform character-level class aggregation.

5) Analysis of Failure Case: The failure cases of CADA are shown in the bottom four rows of Fig. 8, which can be divided into two categories. 1) CADA suffers from the attention drift [7] problem, which limits the effect of intra-class aggregation by interfering with character features. This is mainly reflected in the misidentification of sequence lengths. 2) The CADA may not work when both visual and semantic information is missing since the gain of CADA is mainly due to the dual-class intra-aggregation, which is based on the visual and semantic character features. With low-quality images, it is difficult for CADA to extract visual and semantic features, such as 'r' and 'u' in 'cruise'. In addition, CADA may fail if the handwritten text is too scribbled to resemble other characters visually. For instance, the 't' in 'tomorrow' seems like 'l', or the 't' in 'coefficieuts' looks like 'f ', which leads to wrong recognition of both *Baseline* and CADA. The visual similarity caused by the strokes can interfere with the semantic understanding and thus affect the recognition results. In general, our proposed method can perform well when visual and semantic information is not missing and attention is accurately perceived.

V. Conclusion

This paper proposes a novel unsupervised domain adaptation method based on class aggregation. We extract the character features in visual and semantic spaces, respectively. Thereinto, a single-head self-attention module is introduced to extract visual character features. Besides, we employ cross attention to extract semantic character features. In addition, a center loss is used for dual intra-class aggregation to pull close similar characters. Extensive experiments on multiple domains have been conducted. The results demonstrate the superiority of our proposed method over several state-of-the-art methods.

It is worth noting that our proposed method has several limitations. Our approach may suffer from attention drift problems and low-quality images in some cases, which may limit the performance of CADA. We will further explore these in our future work.

References

[1] Y. Wang, H. Xie, S. Fang, J. Wang, S. Zhu, and Y. Zhang, "From two to one: A new scene text recognizer with visual language modeling network," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14 174–14 183.

[2] B. Li, X. Tang, X. Qi, Y. Chen, C.-G. Li, and R. Xiao, "Emu: Effective multi-hot encoding net for lightweight scene text recognition with a large character set," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5374–5385, 2022.

[3] L. Nandanwar, P. Shivakumara, R. Ramachandra, T. Lu, U. Pal, A. Antonacopoulos, and Y. Lu, "A new deep wavefront based model for text localization in 3d video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 6, pp. 3375–3389, 2022.

[4] J. Baek, Y. Matsui, and K. Aizawa, "What if we only use real datasets for scene text recognition? toward scene text recognition with fewer labels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3113–3122.

[5] A. Aberdam, R. Litman, S. Tsiper, O. Anschel, R. Slossberg, S. Mazor, R. Manmatha, and P. Perona, "Sequence-to-sequence contrastive learning for text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 302–15 312.

[6] K. S. Raghunandan, P. Shivakumara, S. Roy, G. H. Kumar, U. Pal, and T. Lu, "Multi-script-oriented text detection and recognition in video/scene/born digital images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 29, no. 4, pp. 1145–1162, 2019.

[7] Z. Cheng, F. Bai, Y. Xu, G. Zheng, S. Pu, and S. Zhou, "Focusing attention: Towards accurate text recognition in natural images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 5086–5094.

[8] A. K. Bhunia, A. Sain, P. N. Chowdhury, and Y. Song, "Text is text, no matter what: Unifying text recognition using knowledge distillation," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 963–972.

[9] Y.-C. Chang, Y.-C. Chen, Y.-C. Chang, and Y.-R. Yeh, "Smile: Sequence-to-sequence domain adaption with minimizing latent entropy for text image recognition," *arXiv preprint arXiv:2202.11949*, 2022.

[10] Y. Zhang, S. Nie, S. Liang, and W. Liu, "Robust text image recognition via adversarial sequence-to-sequence domain adaptation," *IEEE Trans. Image Process.*, vol. 30, pp. 3922–3933, 2021.

[11] H. Qin, C. Yang, X. Zhu, and X. Yin, "Dynamic receptive field adaptation for attention-based text recognition," in *Proc. Int. Conf. Document Anal. Recog.*, vol. 12822, 2021, pp. 225–239.

[12] J. Baek, G. Kim, J. Lee, S. Park, D. Han, S. Yun, S. J. Oh, and H. Lee, "What is wrong with scene text recognition model comparisons? dataset and model analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 4714–4722.

[13] A. Graves, S. Fernández, F. J. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. Int. Conf. Mach. Learn.*, vol. 148, 2006, pp. 369–376.

[14] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 11, pp. 2298–2304, 2017.

[15] A. K. Bhunia, A. Sain, A. Kumar, S. Ghose, P. N. Chowdhury, and Y. Song, "Joint visual semantic reasoning: Multi-stage decoder for text recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2021, pp. 14 920–14 929.

[16] W. Hu, X. Cai, J. Hou, S. Yi, and Z. Lin, "GTC: guided training of CTC towards efficient and accurate scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, pp. 11 005–11 012.

[17] S. Fang, H. Xie, Y. Wang, Z. Mao, and Y. Zhang, "Read like humans: Autonomous, bidirectional and iterative language modeling for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 7098–7107.

[18] R. Yan, L. Peng, S. Xiao, and G. Yao, "Primitive representation learning for scene text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 284–293.

[19] N. Lu, W. Yu, X. Qi, Y. Chen, P. Gong, R. Xiao, and X. Bai, "MASTER: multi-aspect non-local network for scene text recognition," *Pattern Recognit.*, vol. 117, p. 107980, 2021.

[20] Y. Wang, L. Qi, Y. Shi, and Y. Gao, "Feature-based style randomization for domain generalization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 8, pp. 5495–5509, 2022.

[21] Y. Su, Y. Li, W. Nie, D. Song, and A. Liu, "Joint heterogeneous feature learning and distribution alignment for 2d image-based 3d object retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 10, pp. 3765–3776, 2020.

[22] A. Gretton, K. M. Borgwardt, M. J. Rasch, B. Schölkopf, and A. J. Smola, "A kernel approach to comparing distributions," in *Proc. AAAI Conf. Artif. Intell.*, 2007, pp. 1637–1641.

[23] B. Sun and K. Saenko, "Deep CORAL: correlation alignment for deep domain adaptation," in *Proc. Eur. Conf. Comput. Vis.*, vol. 9915, 2016, pp. 443–450.

[24] J. Zhuo, S. Wang, W. Zhang, and Q. Huang, "Deep unsupervised convolutional domain adaptation," in *Proc. ACM Multimedia Conf.*, 2017, pp. 261–269.

[25] L. Wu, R. Hong, Y. Wang, and M. Wang, "Cross-entropy adversarial view adaptation for person re-identification," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 7, pp. 2081–2092, 2020.

[26] M. Liu and O. Tuzel, "Coupled generative adversarial networks," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 469–477.

[27] W. Zhang, W. Ouyang, W. Li, and D. Xu, "Collaborative and adversarial network for unsupervised domain adaptation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3801–3809.

[28] J. Huang, D. Guan, A. Xiao, S. Lu, and L. Shao, "Category contrast for unsupervised domain adaptation in visual tasks," *arXiv preprint arXiv:2106.02885*, 2021.

[29] Y. Zou, Z. Yu, B. V. K. V. Kumar, and J. Wang, "Unsupervised domain adaptation for semantic segmentation via class-balanced self-training," in *Proc. Eur. Conf. Comput. Vis.*, vol. 11207, 2018, pp. 297–313.

[30] S. Azadi, M. Fisher, V. G. Kim, Z. Wang, E. Shechtman, and T. Darrell, "Multi-content GAN for few-shot font style transfer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7564–7573.

[31] A. K. Bhunia, S. Ghose, A. Kumar, P. N. Chowdhury, A. Sain, and Y. Song, "Metahtr: Towards writer-adaptive handwritten text recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 15 830–15 839.

[32] F. Zhan, C. Xue, and S. Lu, "GA-DAN: geometry-aware domain adaptation network for scene text detection and recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 9104–9114.

[33] Y. Zhang, S. Nie, S. Liang, and W. Liu, "Robust text image recognition via adversarial sequence-to-sequence domain adaptation," *IEEE Trans. Image Process.*, vol. 30, pp. 3922–3933, 2021.

[34] Y. Zhang, S. Nie, W. Liu, X. Xu, D. Zhang, and H. T. Shen, "Sequence-to-sequence domain adaptation network for robust text image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 2740–2749.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Neural Inf. Process. Syst*, 2017, pp. 5998–6008.

[36] J. Lin, Z. Cheng, F. Bai, Y. Niu, S. Pu, and S. Zhou, "Text recognition in real scenarios with a few labeled samples," in *Proc. Int. Conf. Pattern Recognit.*, 2020, pp. 370–377.

[37] X. Zhang, B. Zhu, X. Yao, Q. Sun, R. Li, and B. Yu, "Context-based contrastive learning for scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2022, pp. 3353–3361.

[38] 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.

[39] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 2315–2324.

[40] A. Mishra, K. Alahari, and C. V. Jawahar, "Scene text recognition using higher order language priors," in *Proc. Brit. Mach. Vis. Conf.*, 2012, pp. 1–11.

[41] K. Wang, B. Babenko, and S. J. Belongie, "End-to-end scene text recognition," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2011, pp. 1457–1464.

[42] 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. Jolion, L. Todoran, M. Worring, and X. Lin, "ICDAR 2003 robust reading competitions: entries, results, and future directions," *Int. J. Document Anal. Recognit.*, vol. 7, no. 2-3, pp. 105–122, 2005.

[43] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. G. i Bigorda, S. R. Mestre, J. Mas, D. F. Mota, J. Almazán, and L. de las Heras, "ICDAR 2013 robust reading competition," in *Proc. Int. Conf. Document Anal. Recog.*, 2013, pp. 1484–1493.

[44] T. Q. Phan, P. Shivakumara, S. Tian, and C. L. Tan, "Recognizing text with perspective distortion in natural scenes," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2013, pp. 569–576.

[45] A. Risnumawan, P. Shivakumara, C. S. Chan, and C. L. Tan, "A robust arbitrary text detection system for natural scene images," *Expert Syst. Appl.*, vol. 41, no. 18, pp. 8027–8048, 2014.

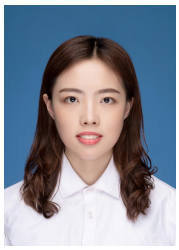
[46] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. K. Ghosh, A. D. Bagdanov, M. Iwamura, J. Matas, L. Neumann, V. R. Chandrasekhar, S. Lu, F. Shafait, S. Uchida, and E. Valveny, "ICDAR 2015 competition on robust reading," in *Proc. Int. Conf. Document Anal. Recog.*, 2015, pp. 1156–1160.

[47] U. Marti and H. Bunke, "The iam-database: an english sentence database for offline handwriting recognition," *Int. J. Document Anal. Recognit.*, vol. 5, no. 1, pp. 39–46, 2002.

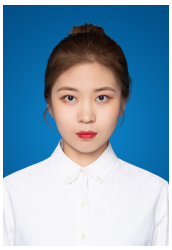
[48] J. Sueiras, "Continuous offline handwriting recognition using deep learning models," *arXiv preprint arXiv:2112.13328*, 2021.

[49] F. Kleber, S. Fiel, M. Diem, and R. Sablatnig, "Cvl-database: An off-line database for writer retrieval, writer identification and word spotting," in *Proc. Int. Conf. Document Anal. Recog.*, 2013, pp. 560–564.

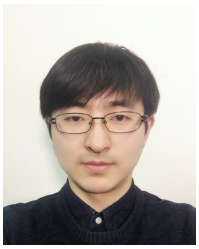
[50] W. Liu, C. Chen, K. K. Wong, Z. Su, and J. Han, "Star-net: A spatial attention residue network for scene text recognition," in *Proc. Brit. Mach. Vis. Conf.*, 2016.
[51] B. Shi, X. Wang, P. Lyu, C. Yao, and X. Bai, "Robust scene text recognition with automatic rectification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 4168–4176.
[52] J. Wang and X. Hu, "Gated recurrent convolution neural network for OCR," in *Proc. Neural Inf. Process. Syst.*, 2017, pp. 335–344.
[53] W. Liu, C. Chen, and K. K. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2018, pp. 7154–7161.
[54] B. Shi, M. Yang, X. Wang, P. Lyu, C. Yao, and X. Bai, "ASTER: an attentional scene text recognizer with flexible rectification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 9, pp. 2035–2048, 2019.
[55] R. Litman, O. Anschel, S. Tsiper, R. Litman, S. Mazor, and R. Manmatha, "SCATTER: selective context attentional scene text recognizer," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 11 959–11 969.
[56] D. Yu, X. Li, C. Zhang, T. Liu, J. Han, J. Liu, and E. Ding, "Towards accurate scene text recognition with semantic reasoning networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12 110–12 119.



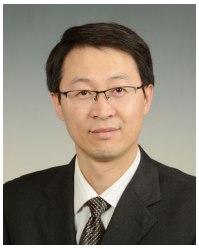
Xiao-Qian Liu received the M.S. degree in Control engineering in 2020 from Shandong University, China. She is currently pursuing the Ph.D. degree in artificial intelligence at the School of Software, Shandong University, Jinan, China. Her research interests include deep learning, computer vision, domain adaptation, and OCR.



Xue-Ying Ding received the bachelor's degree in software engineering from Shandong University, China, in 2020. Currently, she is a graduate student at the School of Software, Shandong University, Jinan, China. Her current research interests include machine learning, computer vision, and scene text recognition.



Xin Luo received the Ph.D. degree in computer science from Shandong University, Jinan, China, in 2019. He is currently an assistant professor with the School of Software, Shandong University, Jinan, China. His research interests mainly include machine learning, multimedia retrieval and computer vision. He has published over 20 papers on TIP, TKDE, ACM MM, SIGIR, WWW, IJCAI, et al. He serves as a Reviewer for ACM International Conference on Multimedia, International Joint Conference on Artificial Intelligence, AAAI Conference on Artificial Intelligence, the IEEE Transactions on Cybernetics, Pattern Recognition, and other prestigious conferences and journals.



Xin-Shun Xu is currently a professor with the School of Software, Shandong University. He received his M.S. and Ph.D. degrees in computer science from Shandong University, China, in 2002, and Toyama University, Japan, in 2005, respectively. He joined the School of Computer Science and Technology at Shandong University as an associate professor in 2005, and joined the LAMDA group of Nanjing University, China, as a postdoctoral fellow in 2009. From 2010 to 2017, he was a professor at the School of Computer Science and Technology, Shandong University. He is the founder and the leader of MIMA (Machine Intelligence and Media Analysis) group of Shandong University. His research interests include machine learning, information retrieval, data mining and image/video analysis and retrieval. He has published in TIP, TKDE, TMM, TCSVT, AAAI, CIKM, IJCAI, MM, SIGIR, WWW and other venues. He also serves as a SPC/PC member or a reviewer for various international conferences and journals, *e.g.* AAAI, CIKM, CVPR, ICCV, IJCAI, MM, TCSVT, TIP, TKDE, TMM and TPAMI.