Meta Self-Learning for Multi-Source Domain Adaptation: A Benchmark

Shuhao Qiu, Chuang Zhu[*], Wenli Zhou
Beijing Laboratory of Advanced Information Networks
Beijing Key Laboratory of Network System Architecture and Convergence
Beijing University of Posts and Telecommunications
Beijing 100876, China
{qiushuhao, czhu, zwl}@bupt.edu.cn

Abstract

In recent years, deep learning-based methods have shown promising results in computer vision area. However, a common deep learning model requires a large amount of labeled data, which is labor-intensive to collect and label. What's more, the model can be ruined due to the domain shift between training data and testing data. Text recognition is a broadly studied field in computer vision and suffers from the same problems noted above due to the diversity of fonts and complicated backgrounds. In this paper, we focus on the text recognition problem and mainly make three contributions toward these problems. First, we collect a multi-source domain adaptation dataset for text recognition, including five different domains with over five million images, which is the first multi-domain text recognition dataset to our best knowledge. Secondly, we propose a new method called Meta Self-Learning, which combines the self-learning method with the meta-learning paradigm and achieves a better recognition result under the scene of multi-domain adaptation. Thirdly, extensive experiments are conducted on the dataset to provide a benchmark and also show the effectiveness of our method. The code of our work and dataset are available soon at `https://bupt-ai-cz.github.io/Meta-SelfLearning/`.

1. Introduction

In recent years, the booming development of deep learning leads to great progress in computer vision field. Text recognition has always been an important field in computer vision, for texts are everywhere in daily life and the understanding of them can be very meaningful. However, to realize accurate recognition in the real scene (which is known as scene text recognition) is still a challenging field because of the diversity of fonts and complicated environment (distortion, variation of font, occlude, etc.). Many deep-learning-based methods are proposed over the past few years to solve this problem [29, 27, 28, 36, 8].

As a data-driven method, the performance of the deep learning model highly relies on the amount of training data. The common way for addressing the above problem is to build publicly available large-scale datasets. However, collecting and labeling a large amount of real scene text data can be a time-consuming and labor-intensive work. What's more, the direct use of these datasets can not produce good results sometimes because of the distribution shift between the training data (source domain) and the testing data (target domain). Domain adaptation is a research field that focuses on aligning the source domain and target domain and thus obtaining better results. In recent years, many domain adaptation methods [33, 31, 20] have been proposed to solve the domain shift problem in image classification problems. Some researches in the text recognition area are also proposed that have good results in different domains [39, 11, 38].

While single-source domain adaptation is widely researched, multi-source domain adaptation is actually more suitable for the scene text recognition problem, for the training data of scene texts are always collected from many different sources. As a generalization form of single-domain, multi-source domain adaptation universally gets a better result than single-source domain adaption for the larger training corpus. Most works in this area focus on image classification problems [40, 24], and some datasets are proposed for this field [24, 17]. However, to our best knowledge, there are no publicly available datasets for text recognition in this area, and therefore, there is almost no related research work. In this paper, we collect a multi-source domain adaption dataset and provide a benchmark to fill this gap.

Some recent studies combined multi-source domain adaptation methods and meta-learning together. Hieu *et al.* [25] proposed a self-learning method combined with meta-

[*] the corresponding author: Chuang Zhu (czhu@bupt.edu.cn)

learning, which achieved SOTA on many image classification tasks. Li *et al.* [15] proposed a framework to fit any domain adaptation methods and got better results for the good initialization provided by the meta-learning method. However, during the meta-update, this method didn't make use of the information from the target, is very important for unsupervised domain adaptation problems. Inspired by this work, we proposed a method called meta self-learning in this paper. Our method adequately utilizes the information of the target domain by adding the target domain data to the meta-update process and get pseudo-labels with higher quality. The main contributions of our work are summarized as follows:

- We collect a multi-source domain adaptation dataset for text recognition with over 5 million images from 5 different domains. To the best of our knowledge, this is the first multi-domain adaptation dataset for text recognition.
- We propose a new self-learning framework for multi-source domain adaptation, which is effective and can be easily fit into any MDA and self-learning problem.
- Experiments are conducted on our dataset, which provide a benchmark and show the effectiveness of our method.

2. Related Works

2.1. Text Recognition

A common text recognition system can be divided into four stages: image preprocessing stage, feature extraction stage, sequence modeling stage, and prediction stage. The preprocessing stage mainly focuses on normalizing the image and rotating the text images into an appropriate position; STN [10] is a commonly used method in the computer vision area. Recent works [28, 19, 29] proposed methods like thin-plate spline (TPS) transformation to get a better result. In the feature extraction stage, CNN is used to extract the generate feature maps for images. In the field of text recognition, images are always be transformed into a feature map with a height of 1, therefore can be processed as a sequence in the following stages. In the sequence modeling stage, models like Bi-LSTM or GRU are used to learn the sequence information from the feature extracted in the last stage. Due to the variable length of the data, connectionist temporal classification (CTC) [6] or attention mechanism [2, 3] are commonly used in the prediction stage to make the final prediction.

2.2. Domain adaptation

The original domain adaptation only focuses on the single-source domain problem, and the main idea is to align the distribution between the source domain and target domain. Multi-source domain adaptation problem is also getting popular in recent years for it is more closer to the real scene and can get a better result than single-source domain adaptation generally. The main domain adaptation methods can be mainly classified into three types.

Discrepancy based domain adaptation: Tzeng *et al.* [33] proposed a domain confusion loss by calculating the maximum mean discrepancy (MMD) between the source domain data and the target domain data. Long *et al.* [20] proposed to calculate the MMD of more than one layer and used a multi-kernel MMD (MK-MMD) to achieve a better alignment. Sun *et al.* [31] proposed CORAL loss to align the second-order statistics of the source and target distributions. Peng *et al.* [24] proposed a multi-source domain adaptation method to calculate the moment distance not only between the source domain and target domain, but also among source domains.

Adversarial training based domain adaptation: Ganin *et al.* [5] used a domain discriminator and proposed a gradient reversal layer to separate the feature extractor and the domain discriminator, which forces the feature extractor to extract the domain-invariant feature. Zhao *et al.* [40] proposed an adversarial method to solve the multi-source domain adaptation problem using the gradient reversal layer and provided a thorough analysis.

Self-training-based domain adaptation: Self-training method has been widely used in image classification and segmentation problems. The method trains the model iteratively by generating pseudo-label of target data and adding them into the training data [32]. However, the direct use of this mechanism may only be helpful for the easy class and lead to a bias among classes in classification problems. Zou *et al.* [42] proposed a confidence regularized self-training method by adding regularizers to the network and achieve a better result.

2.3. Meta-Learning

Meta-learning, also known as learning-to-learn, is a broadly studied field in recent years. Different from traditional learning methods focusing on a specific task, meta-learning methods aim to learn "how to learn" from multiple tasks and achieve fast adaptation on a new task with a few samples.

MAML [4] is a very famous meta-learning algorithm, which aims to learn a good initialization of parameters and can guarantee a fast convergence to local minimal with a small amount of data on a new task. However, the computation overhead of MAML during training is very high due to the calculation of second-order derivatives. To reduce the computational cost of MAML, Reptile [23] provides a family of first-order meta-learning algorithms to approximate the original MAML. Some meta-learning methods are de-

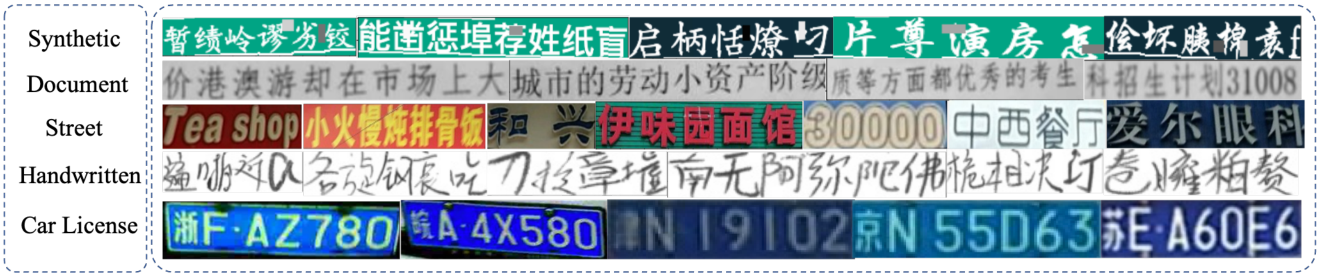


Figure 1. Dataset Overview: We address a multi-source domain adaptation dataset for text recognition, which contains more than 5 million images from five different domains, which are synthetic domain, document domain, street view domain, handwritten domain, and car license domain respectively. Some examples of data are shown in the figure.

signed based on metric learning, such as matching network and prototypical network [30, 34], which also make great influence in the field of few-shot learning.

Due to the inherent "bi-level update" property [7], some meta-learning-based domain adaptation and domain generalization methods were proposed in past few years. Li *et al*. [16] proposed a domain generalization method by dividing the source domains into meta-train domains and meta-test domains to simulate the real training process, which achieves better results on the real target domain. An online meta-learning method was proposed to enhance the effectiveness of any domain adaptation method [15]. The online meta-learning paradigm also enables the long-term effect of meta-learning, instead of only being effective at the beginning of the training stage.

2.4. Self-Learning

Self-learning methods predict labels for the unlabeled data using the model trained on source domains and take them as correct labels if the predict confidence is higher than a threshold [14]. The self-learning method can always bring considerable improvement because of the direct use of target domain data. However, there also exist some problems. The generated pseudo-labels can be noisy sometimes, and lead the model to a bad local minimal. Therefore, most works focus on how to generate pseudo-labels with high quality. Recent works [41, 42] provide methods for balancing different classes or adding regularizers to the model. Hieu *et al*. [25] provided a meta-learning paradigm combining with the teacher-student model, thus the parameters of the teacher model can be evaluated by the pseudo-label and get updated with better quality. In this paper, we provide a new way to combine the meta-learning paradigm and the self-learning methods on the basis of [15], by utilizing the information of pseudo-label during the meta-update.

3. Multi-Domain Text Recognition Dataset

In this section, we will introduce the details of our dataset. Our multi-domain text dataset consists of 5,209,215 images in total, and is divided into five domains, which are synthetic domain, handwritten domain, document domain, street view domain, and car plate domain. The character set size is set to 3,816, with 3,754 common Chinese characters and 62 alphanumeric characters, which can be represented as C. All of the five different domains are split into the training set and the test set with the ratio of $9:1$. The details of different domains are shown as follows.

Synthetic Domain. A good deep-learning-based text recognition model requires millions of images, which is very hard to collect and label. To fill up the gap of data size, synthetic texts are widely used during the training procedure. Therefore, a synthetic domain is necessary for the multi-domain dataset.

Our synthetic dataset contains 1,110,620 images in total. We generate all the training data with 5 different fonts and three different backgrounds. Random blocks and lines are added for the augmentation. When generating corpora, we found it impossible to cover all the characters in the charset only using real corpus. To achieve a better result on uncommon characters, we sacrifice the semantic information of this domain and generated all samples by random sampling from the C. The length of each corpus is between 4 and 10.

Document Domain. The data of document domain is collected from an open-source project[1], and the dataset contains about 3 million images. We filtered out the images that contains characters not in C and got 1,710,885 images in total. The corpora in this domain are from documents and news, and have the same length of 10.

Street View Domain. There are many publicly available street view datasets on the internet, however, most of them only contain street view text images from one region, which means only contains Chinese character or alphanumeric characters. In order to make a better recognition result on both Chinese and alphabetical characters, we merged the images from both Chinese scene text recogni-

[1] https://github.com/YCG09/chinese_ocr

tion datasets and English scene text recognition datasets, including SVT [35], SVT perspective [9], ICDAR2013 [13], ICDAR2015 [12], RCTW17 [22], ICDAR-2019 [21], and CUTE80 [26]. After the same filtering operation with the document domain, we got 199,346 images in this domain.

Handwritten Domain. The data of the handwritten domain is generated using the images in CASIA Online and Offline Chinese Handwriting Databases [18]. Out of the same consideration in the synthetic domain, we think that a better coverage of the charset is more important. Therefore, we use the same corpora with the synthetic domain. What's more, we also use some corpora from the street view domain to balance between the semantic information and the coverage of the charset. Images are generated by concatenating single-char images together according to the corpora. We get 1,897,021 images in total for handwritten images.

Car License Domain. The car license domain is composed of two parts. The first part is the largest Chinese car license dataset CCPD [37], and we only use the base part of this dataset, which contains 199,996 images. Although the CCPD contains a large amount of data, there exists a severe problem in this dataset. A Chinese license plate consists of 7 characters, the first one is the Chinese abbreviation of the province, and the remaining six are letters or numbers. Among the 7 characters, the abbreviation of the province is the most difficult to recognize for the Chinese characters are more complicated than alphabetic characters. However, most images in this dataset are collected from the same city, which means most of the images have the same province identity. This situation leads to a severe imbalance of the dataset, and the model being trained on this dataset can not get good performance in recognizing the province identity of other provinces. To solve this problem, we provide extra 7,932 images collected from surveillance cameras in 26 different provinces and alleviate this problem. We finally got 207,928 images in total, including 31 Chinese characters and 34 alphabetic characters.

4. Method

In this paper, we focus on the problem of multi-source domain adaptation and provide a new method combining the self-learning method with the online meta-learning method. The overview of our method is shown in Fig. 2. Given $D_S = \{S_1, S_2, \dots, S_N\}$ as the data with labels from multiple source domains, $\overline{D}_T$ as the target domain without labels, our goal is to get a model $f(\cdot)$ with parameters θ that achieves good results on the target domain using D_S and $\overline{D}_T$. By using the self-learning method, pseudo-label will be generated using $\overline{D}_T$, the data with pseudo-label can be represented as $\breve{D}_T$. While previous work using online meta-learning method didn't take pseudo-label into account during the meat-update, we add $\breve{D}_T$ into the meta-train set and the model will be updated with both D_S and $\breve{D}_T$ during the meta-update. This setting brings great gain for the model, for the information of the target domain can be very valuable. What's more, pseudo-labels with higher quality can be acquired under this paradigm. In the following section, we will first introduce the detail of our proposed method, and then introduce the text recognition model we used.

Algorithm 1: Meta Self-Learning for Multi-source Domain Adaptation

```
Data: $D_S = \{S_1, S_2, \dots, S_N\}, \overline{D}_T = T_1$
Input: Initial model $f(\theta)$
Input: Meta train learning rate $\alpha$, meta test learning
       rate $\beta$, outer learning rate $\gamma$
Input: pseudo-label threshold $\tau$
Result: Optimized parameter $\theta$
Warming up using $D_s$, get $\theta$;
while not converge do
    $\breve{D}_T = f(\theta; T_1) > \tau$;
    Random Split: $D_S + \breve{D}_T \rightarrow \overline{M} + \widetilde{M}$ ;
    Meta train: evaluate $\nabla_\theta = \frac{\partial l(\theta;\overline{M})}{\partial \theta}$;
    Update: $\theta' = \theta - \alpha \nabla_\theta$;
    Meta test: evaluate $\nabla_{\theta'} = \frac{\partial l(\theta';\widetilde{M})}{\partial \theta'}$;
    Update $\theta = \theta - \beta \nabla_{\theta'}$;
    Outer optimization: evaluate $\nabla_\theta = \frac{\partial l(\theta;D_S)}{\partial \theta}$;
    Update $\theta = \theta - \gamma \nabla_\theta$;
end
```

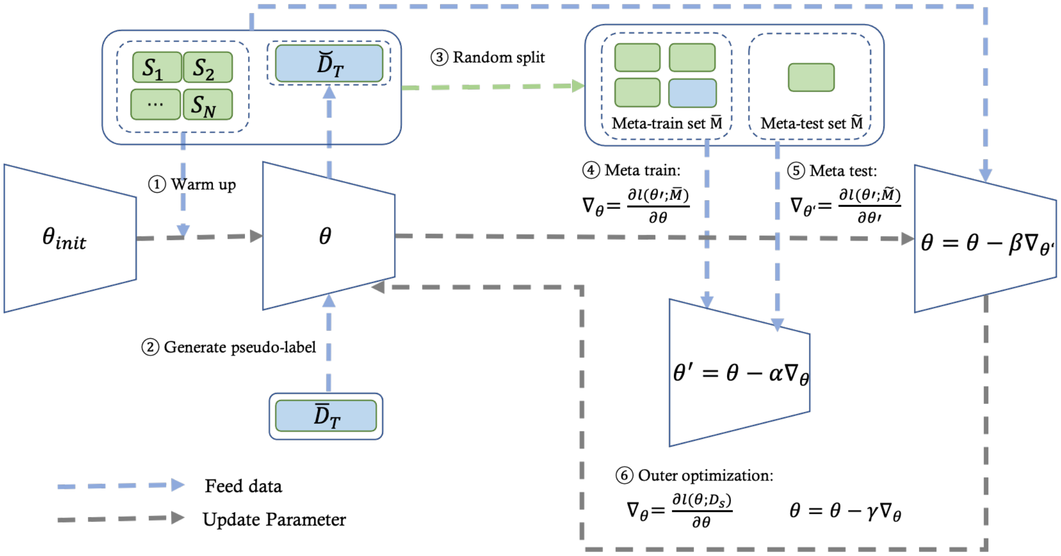
4.1. Meta Self-Learning

The whole procedure of our meta self-learning method is described in Algorithm 1.

Warm-Up and Generate Pseudo-Label. The model will first be trained on D_S as the warm-up phase. Warm-up is a necessary process for the self-learning method, and this process will greatly improve the quality of the generated pseudo-label and lead to a better result. Without warm-up process, the generated pseudo-labels will either have low confidence or wrong content, which will greatly jeopardize the predict accuracy on the target domain. After the warm-up, the target data with pseudo-label $\breve{D}_T$ will be generated.

Random Split. The usage of the pseudo-label is one of the most important issue. As the raw pseudo-label can be noisy, a meta-update is used in our method. During the meta-update, both D_S and $\breve{D}_T$ will be used, and are divided randomly into meta-train set $\overline{M}$ and meta-test set $\widetilde{M}$, which corresponds to the support set and query set in vanilla MAML.

Meta-Train. The network will first update on the meta-



S_1
S_2
$\cdots$
S_N
$\breve{D}_T$
③ Random split
Meta-train set $\overline{M}$
Meta-test set $\widetilde{M}$
① Warm up
θ_{init}
θ
④ Meta train:
$\nabla_\theta = \frac{\partial l(\theta';\overline{M})}{\partial \theta}$
⑤ Meta test:
$\nabla_{\theta'} = \frac{\partial l(\theta';\widetilde{M})}{\partial \theta'}$
$\theta = \theta - \beta \nabla_{\theta'}$
$\theta' = \theta - \alpha \nabla_\theta$
② Generate pseudo-label
$\overline{D}_T$
Feed data
Update Parameter
⑥ Outer optimization:
$\nabla_\theta = \frac{\partial l(\theta;D_s)}{\partial \theta}$
$\theta = \theta - \gamma \nabla_\theta$

Figure 2. Illustration of Meta Self-Learning method: The procedure of our method can be summarized as follow steps: 1. The data from source domains with labels D_S are used for warm-up; 2. The model is evaluated on the target domain data without labels $\overline{D}_T$ and generates pseudo-labels; 3. The target domain data with pseudo-labels D_S and $\breve{D}_T$ are split randomly as $\overline{M}$ and $\widetilde{M}$; 4. Meta train using $\overline{M}$; 5. Meta test using $\widetilde{M}$; 6. Outer optimization using a subset of D_S and $\breve{D}_T$.

train set $\overline{M}$ using the text recognition loss l in Eq. 12

$$l_a = \frac{1}{||\overline{M}||} \sum_{i=0}^{||\overline{M}||} l(\theta; \hat{y^i}, y^i), \tag{1}$$

where $||\overline{M}||$ is the size of meta-train set, $\hat{y^i}$ and y^i are the predicted label and the ground-truth label of text image, respectively. Then, the gradient of the model parameter θ is calculated as ∇_θ, where

$$\nabla_\theta = \frac{\partial l_a(\theta)}{\partial \theta}. \tag{2}$$

Then, the parameter will be updated as

$$\theta' = \theta - \alpha \nabla_\theta, \tag{3}$$

where α is the learning rate for the meta-train phase.

Meta-test. The meta-test phase is used to evaluate the model using meta-test set $\widetilde{M}$. In this phase, the loss function l_b is calculated with parameter updated in meta-train phase θ'.

$$l_b = \frac{1}{||\widetilde{M}||} \sum_{i=0}^{||\widetilde{M}||} l(\theta'; \hat{y^i}, y^i), \tag{4}$$

where $||\widetilde{M}||$ is the size of meta-test set. Following the vanilla MAML, we need to calculate the gradient of the original parameter θ using l_b , which is

$$\begin{aligned} \frac{\partial l_b(\theta')}{\partial \theta} &= \frac{\partial l_b(\theta')}{\partial \theta'} \cdot \frac{\partial \theta'}{\partial \theta} \\ &= \frac{\partial l_b(\theta')}{\partial \theta'} \cdot \frac{\theta - \alpha \frac{\partial l_s(\theta)}{\partial \theta}}{\partial \theta} \\ &= \frac{\partial l_b(\theta')}{\partial \theta'} \cdot (1 - \alpha \frac{\partial^2 l_s(\theta)}{\partial \theta^2}) \end{aligned} \tag{5}$$

It can be seen that a second-order derivative needs to be calculated. However, calculating the second-order derivative can be prohibitively expensive for a deep learning framework, especially for a large model with a long computation graph. Therefore, a first-order approximation of MAML is widely used, we can simply neglect the second-order entry in Eq. 5, and the gradient ∇_θ can be approximated as

$$\frac{\partial l_b(\theta')}{\partial \theta} \approx \frac{\partial l_b(\theta')}{\partial \theta'}. \tag{6}$$

After the approximation, the gradient we need to update the original parameter θ can be replaced by $\nabla_{\theta'}$, where

$$\nabla_{\theta'} = \frac{\partial l_b(\theta')}{\partial \theta'}. \tag{7}$$

Therefore, the initial parameter θ can be directly updated using $\nabla_{\theta'}$ as $\theta = \theta - \beta \nabla_{\theta'}$, where β is the learning rate for the meta-test phase.

Outer optimization As the meta-update uses the pseudo-label which can be noisy, an outer optimization is added additionally after it. In this phase, only the data from

D_S with real label is used to update the model with learning rate γ, as $\theta = \theta - \gamma\nabla_{\theta}$.

As MAML learns the initialization of network parameters only, it can not be applied to a normal network training with a consecutive update every iteration, for the influence of the initialization can be very trivial after times of iteration. In this paper, we use the online meta-learning method [15], and implements the procedure above in each iteration, therefore the network can benefit from the meta-learning paradigm throughout the training process.

4.2. Text Recognition Model

In this section, we will introduce the text recognition model we used following the flow in Section 2. For most of our images don't have severe deformation, the TPS module is not used in our model. During the feature extraction stage, the raw input image X is fed into the CNN (we use ResNet-50 in our experiment) and generates the output feature map $F(X) = \boldsymbol{x}$, where $\boldsymbol{x}$ has the shape of $D \times 1 \times T$ and can be represented as $\boldsymbol{x} = \{x_1, x_2, \ldots, x_t\}, x_i \subseteq R^d$. Note that D and T represent the channel number and the length of the feature map respectively, and the height of the feature map is set to 1. During the sequence modeling stage, we use a BiLSTM, and the hidden state of each time step can be represented as $\boldsymbol{h} = \{h_1, h_2, \ldots, h_t\}, h_i \subseteq R^h$. The hidden state $\boldsymbol{h}$ is then used for the final prediction. In our model, we use an attention mechanism in [2]. During the prediction of each time step, a context vector c_t is calculated by weighting the importance of different time steps

$$c_t = \sum_{i=0}^{T} \alpha_{t,i} h_i, \tag{8}$$

where the weight $\alpha_{t,i}$ is

$$\alpha_{t,i} = \frac{exp(c_{t,i})}{\sum_{j=0}^{T} exp(c_{t,j})}. \tag{9}$$

The $c_{t,i}$ in Eq. 9 is the importance of the i-th time step to the t-th time step, calculated with

$$c_{t,i} = tanh(W_S s_{t-1} + W_h h_i), \tag{10}$$

where W_s, W_h are the learnable parameters and s_{t-1} is the hidden state of the decoder. The hidden state s_t is calculated using the hidden state and the ground truth of the last time step g_{t-1} (which is teacher forcing method), together with the context vector of current time step c_t using a LSTM

$$s_t = LSTM(g_{t-1}, s_{t-1}, c_t). \tag{11}$$

Finally, a cross-entropy loss is used to calculate the classification loss on each time step

$$L = \prod_{i=1}^{T} \sum_{j=1}^{k} -y_{ik} log \hat{y}_{ik}, \tag{12}$$

where k is the size of the charset.

5. Experiments

5.1. Experimental Settings

In this section, we provide a benchmark on the dataset we proposed, and also show the experimental results and ablation studies to demonstrate the effectiveness of our method. The base text recognition model is modified on the best model in [1].

We implement our model using PyTorch on an NVIDIA Tesla T4. Adam is used as the outer optimizer, and SGD is used as the meta optimizer. α is set to $1e-3$, β and γ are changed during the training process. During training, we pick one domain as the target domain while the other four domains as source domains. We set the batch size to 24 per domain, which is 96 for 4 source domains and all the images are resized into 100×32. When using pseudo-label, we will use the training set of the target domain to generate the pseudo-label and the result is tested on the test set, which is unavailable during training.

The size of the character set in these experiments is set to 3818, which includes 3756 common Chinese characters and 62 alphanumeric characters.

5.2. Experimental Results

Baseline. The baseline model is trained with only source domains without any multi-source domain adaptation methods. The test accuracy of each domain is shown in Table 1. It can be seen that, directly using the source domain data performs badly on the target domain, which indicates that there are non-negligible domain gaps among different domains. The average accuracy among the 5 domains is 17.86%.

MLDG [16]. As described in the last section, our algorithm is a combination of meta-learning paradigm and pseudo-label method. In order to figure out the effectiveness of each part, we conduct experiments with two methods respectively. Li *et al.* [16] provide a training method using the meta-learning paradigm only. During the training, the source domains are divided into meta-train set and meta-test set. The model will first update one step using the meta-train set and then validate on the meta-test set. The final model converged on source domains will be deployed on the truly held-out target domain. According to the experiment results shown in table, the MLDG is not very effective for text recognition tasks for there is only a 1.16% improvement on average. We think the reason is that the difference between source domains and target domain is not only on appearance but also on the semantic level. For example, the document domain has a fixed length of 10 characters per sample, while other domains only contain few samples of the same length. What's more, the training data are sampled

Table 1. Experiment results on five different target domains. St represents street domain; Sy represents synthetic domain; D represents documentation domain; H represents Handwritten domain; C represents car license domain

	St,Sy,D,H→C	St,Sy,D,C→H	St,Sy,C,H→D	C,St,D,H→Sy	C,Sy,D,H→St	Average
Source Only	22.43%	3.50%	29.39%	24.75%	9.24%	17.86%
MLDG [16]	23.85%	3.39%	30.31%	25.11%	12.46%	19.02%
Pseudo-Label [14]	44.97%	3.77%	51.60%	54.11%	15.00%	33.89%
Meta Self-Learning (Ours)	**58.64%**	**5.41%**	**64.09%**	**65.33%**	**16.52%**	**42.00%**

Table 2. Experiment results of different settings on meta self-learning method.

	St,Sy,D,H→C	St,Sy,D,C→H	St,Sy,C,H→D	C,St,D,H→Sy	C,Sy,D,H→St
IAOS	**58.64%**	4.93%	42.94%	37.06%	**16.52%**
IPOA	44.94%	**5.41%**	53.35%	56.72%	15.34%
IPOP	41.05%	3.41%	**64.09%**	**65.33%**	15.02%

from different corpora, making it hard to learn the target domain's distribution with source domain data only.

Pseudo-Label. The experiment results using pseudo-label methods are shown in Table 1. As the warm-up is a necessary step for the pseudo-label method, we use the baseline model as the pre-trained model and start training using pseudo-label directly on it. For car, street, synthetic, and document domains, we set the threshold of pseudo-label confidence as 0.9. The threshold for handwritten domain is set to 0.98, for the pre-trained model performs bad on this domain. For the number of training data is very large in all domains, testing on the whole training set can be very time-consuming. Therefore, we only evaluate 50,000 images per domain, and the evaluation is done every 5,000 iterations. As shown in table, using pseudo-label can bring an up to 22.54% gain on accuracy for a single domain, and the average accuracy increases by 16.03%.

Meta Self-Learning. Using the same setting with the pseudo-label method, the same experiments are conducted with our meta self-learning method, and the results are shown in Table 1. It can be seen that our method brings up to 13.67% gain and 8.11% gain on average accuracy compared with the vanilla pseudo-label method, and achieves best result on every target domain. It's worth noting that, we actually used different settings for the different target domains. It is based on the finding that for each target domain, the selection of domains for meta-update and outer optimization may affect the result greatly, and we will discuss the detail in the next section. The result shown in the table is the result corresponds to the best setting for each target domain.

5.3. Discussion on Different Target Domains

During training, we found that the training procedure shown in Algorithm 1 not always performs best, therefore, we did experiments on three different settings as shown in Table 2. The main difference between these three settings is mainly reflected on the usage of pseudo-label images.

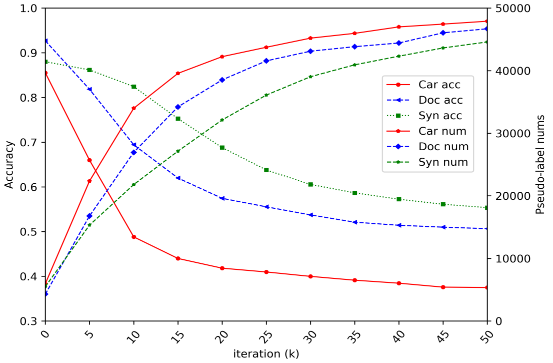


Figure 3. The accuracy of pseudo-label for car plate, document and synthetic domain during training.

IAOA represents the training procedure shown in algorithm 1, which use all 5 domains during the meta-update, and only use source domains during the outer optimization. Under this setting, we got the best results on the car license domain and the street domain. However, this setting didn't get good results on document domain and synthetic domain, and even worse than using the vanilla pseudo-label method. This may indicate that, in some domains, the use of source domain data may jeopardize the effectiveness of the pseudo-label image. Therefore, we tried to make pseudo-label play a more important role during training in some domains.

IPOA represents using the pseudo-label domain as meta-test set only during meta-update and use all five domains during outer optimization. In this setting, images with pseudo-label are added into the outer optimization and therefore influence the result more. It can be seen that, this setting achieves a better result than the pseudo-label method on synthetic, document and handwritten domains, which verifies our inference above. Therefore, we exploit the pseudo-label further in the next setting.

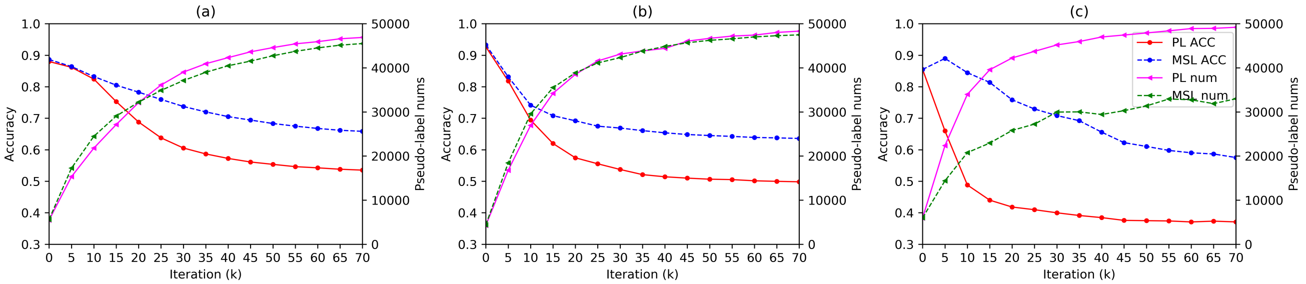


Figure 4. The accuracy of pseudo-label during training for vanilla pseudo-label method and our method from 3 domains. (a) is the synthetic domain, (b) is the document domain, (c) is the car license domain

IPOP has the same setting with IPOA during meta-update, while only use images with pseudo-label during the outer optimization. This setting achieves great improvements on synthetic and document domains, which is 64.09% on the document domain and 65.33% on the synthetic domain.

From the experiment results, we can see that the different usage of pseudo-label images can produce great gaps in the final accuracy, however, the impact is also different among different domains, but why this happens? Here we provide an intuitive explanation by the following experiments.

During training, we record the number of generated pseudo-label images and images that have the correct pseudo-label. The results from the car plate, document and synthetic domain using vanilla pseudo-label method are shown in Fig. 3. In all three domains, the number of the generated pseudo-labels keep increasing during the training process and finally reach nearly 50,000, which is the maximum number of images we set that allowed to be used as pseudo-label. Meanwhile, the accuracy of generated pseudo-labels keep decreasing and finally converge to a value. It can be seen that the pseudo-label accuracy of the car plate domain finally converges to about 0.4 while the in document and synthetic domain, this value is between 0.5 and 0.6, which means that, the pseudo-label quality of the car plate domain is relatively low. Therefore, it is reasonable to see that the accuracy of car license domain gets lower when rely more on the pseudo-label domain, while the accuracy of document and synthetic domain get better results.

The effectiveness of our method can also be shown in Fig. 4. We demonstrate the number of pseudo-label and pseudo-label accuracy in both vanilla pseudo-label method and our methods on car, synthetic, and document domains during training. The results of synthetic domain and document domain are similar. The pseudo-label number will converge to nearly 50,000 in both two methods, while our method stably gets a higher accuracy on the generated pseudo-label, which is on average 10% higher than the vanilla pseudo-label method. For the car plate domain, the number of pseudo-label in our method is controlled to about 30,000, and get a 20% promotion on the accuracy. These indicate that our method can produce pseudo-label with higher quality and get better training results.

5.4. Application Analysis

Our method provides a self-learning method for multi-source domain adaptation problems, but it can also be transferred into the single-source domain adaptation problem, where D_S contains only one domain. Actually, our method is a self-learning framework and is model-agnostic, therefore can be easily applied to any task using self-learning methods. However, as discussed in the last section, the paradigm for different tasks may need to be changed according to the quality of pseudo-labels. In future work, we will try to find a theoretical explanation and a unified framework for our method.

6. Conclusion

In this paper, we collect and generate a multi-source domain adaptation dataset for text recognition. To our best knowledge, this is the first and the largest publicly available dataset for this area, which is very meaningful for the great significance of both domain adaptation and text recognition problems. We also propose a new meta self-learning method for the multi-source domain adaptation problem, which is model-agnostic and can be easily applied to other tasks. Extensive experiments are done on our dataset to provide a benchmark and demonstrate the effectiveness of our method. However, our dataset is still very challenging because of the large scale of charset and notable domain shift among domains, and worth more exploration in the future.

Acknowledgements

This work was supported in part by 111 Project of China (B17007), and in part by the National Natural Science Foundation of China (61602011).

References

[1] Jeonghun Baek, Geewook Kim, Junyeop Lee, Sungrae Park, Dongyoon Han, Sangdoo Yun, Seong Joon Oh, and Hwalsuk Lee. What is wrong with scene text recognition model comparisons? dataset and model analysis. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 4715–4723, 2019.

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

[3] Jan Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. Attention-based models for speech recognition. *arXiv preprint arXiv:1506.07503*, 2015.

[4] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International Conference on Machine Learning*, pages 1126–1135. PMLR, 2017.

[5] Yaroslav Ganin and Victor Lempitsky. Unsupervised domain adaptation by backpropagation. In *International conference on machine learning*, pages 1180–1189. PMLR, 2015.

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

[7] Timothy Hospedales, Antreas Antoniou, Paul Micaelli, and Amos Storkey. Meta-learning in neural networks: A survey. *arXiv preprint arXiv:2004.05439*, 2020.

[8] Wenyang Hu, Xiaocong Cai, Jun Hou, Shuai Yi, and Zhiping Lin. Gtc: Guided training of ctc towards efficient and accurate scene text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 11005–11012, 2020.

[9] Max Jaderberg, Karen Simonyan, Andrea Vedaldi, and Andrew Zisserman. Synthetic data and artificial neural networks for natural scene text recognition. *arXiv preprint arXiv:1406.2227*, 2014.

[10] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

[11] Lei Kang, Marçal Rusinol, Alicia Fornés, Pau Riba, and Mauricio Villegas. Unsupervised writer adaptation for synthetic-to-real handwritten word recognition. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 3502–3511, 2020.

[12] Dimosthenis Karatzas, Lluis Gomez-Bigorda, Anguelos Nicolaou, Suman Ghosh, Andrew Bagdanov, Masakazu Iwamura, Jiri Matas, Lukas Neumann, Vijay Ramaseshan Chandrasekhar, Shijian Lu, et al. Icdar 2015 competition on robust reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160. IEEE, 2015.

[13] Dimosthenis Karatzas, Faisal Shafait, Seiichi Uchida, Masakazu Iwamura, Lluis Gomez i Bigorda, Sergi Robles Mestre, Joan Mas, David Fernandez Mota, Jon Almazan Almazan, and Lluis Pere De Las Heras. Icdar 2013 robust reading competition. In *2013 12th International Conference on Document Analysis and Recognition*, pages 1484–1493. IEEE, 2013.

[14] Dong-Hyun Lee et al. Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks. In *Workshop on challenges in representation learning, ICML*, volume 3, 2013.

[15] Da Li and Timothy Hospedales. Online meta-learning for multi-source and semi-supervised domain adaptation. In *European Conference on Computer Vision*, pages 382–403. Springer, 2020.

[16] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy Hospedales. Learning to generalize: Meta-learning for domain generalization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

[17] Da Li, Yongxin Yang, Yi-Zhe Song, and Timothy M Hospedales. Deeper, broader and artier domain generalization. In *Proceedings of the IEEE international conference on computer vision*, pages 5542–5550, 2017.

[18] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Casia online and offline chinese handwriting databases. In *2011 International Conference on Document Analysis and Recognition*, pages 37–41. IEEE, 2011.

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

[20] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael Jordan. Learning transferable features with deep adaptation networks. In *International conference on machine learning*, pages 97–105. PMLR, 2015.

[21] Nibal Nayef, Yash Patel, Michal Busta, Pinaki Nath Chowdhury, Dimosthenis Karatzas, Wafa Khlif, Jiri Matas, Umapada Pal, Jean-Christophe Burie, Cheng-lin Liu, et al. Icdar2019 robust reading challenge on multi-lingual scene text detection and recognition—rrc-mlt-2019. In *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pages 1582–1587. IEEE, 2019.

[22] Nibal Nayef, Fei Yin, Imen Bizid, Hyunsoo Choi, Yuan Feng, Dimosthenis Karatzas, Zhenbo Luo, Umapada Pal, Christophe Rigaud, Joseph Chazalon, et al. Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification-rrc-mlt. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, volume 1, pages 1454–1459. IEEE, 2017.

[23] Alex Nichol, Joshua Achiam, and John Schulman. On first-order meta-learning algorithms. *arXiv preprint arXiv:1803.02999*, 2018.

[24] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1406–1415, 2019.

[25] Hieu Pham, Zihang Dai, Qizhe Xie, Minh-Thang Luong, and Quoc V Le. Meta pseudo labels. *arXiv preprint arXiv:2003.10580*, 2020.

[26] Anhar Risnumawan, Palaiahankote Shivakumara, Chee Seng Chan, and Chew Lim Tan. A robust arbitrary text detection

system for natural scene images. *Expert Systems with Applications*, 41(18):8027–8048, 2014.

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

[28] Baoguang Shi, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Robust scene text recognition with automatic rectification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4168–4176, 2016.

[29] Baoguang Shi, Mingkun Yang, Xinggang Wang, Pengyuan Lyu, Cong Yao, and Xiang Bai. Aster: An attentional scene text recognizer with flexible rectification. *IEEE transactions on pattern analysis and machine intelligence*, 41(9):2035–2048, 2018.

[30] Jake Snell, Kevin Swersky, and Richard S Zemel. Prototypical networks for few-shot learning. *arXiv preprint arXiv:1703.05175*, 2017.

[31] Baochen Sun and Kate Saenko. Deep coral: Correlation alignment for deep domain adaptation. In *European conference on computer vision*, pages 443–450. Springer, 2016.

[32] Isaac Triguero, Salvador García, and Francisco Herrera. Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study. *Knowledge and Information systems*, 42(2):245–284, 2015.

[33] Eric Tzeng, Judy Hoffman, Ning Zhang, Kate Saenko, and Trevor Darrell. Deep domain confusion: Maximizing for domain invariance. *arXiv preprint arXiv:1412.3474*, 2014.

[34] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. *arXiv preprint arXiv:1606.04080*, 2016.

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

[36] Tianwei Wang, Yuanzhi Zhu, Lianwen Jin, Canjie Luo, Xiaoxue Chen, Yaqiang Wu, Qianying Wang, and Mingxiang Cai. Decoupled attention network for text recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 12216–12224, 2020.

[37] Zhenbo Xu, Wei Yang, Ajin Meng, Nanxue Lu, Huan Huang, Changchun Ying, and Liusheng Huang. Towards end-to-end license plate detection and recognition: A large dataset and baseline. In *Proceedings of the European conference on computer vision (ECCV)*, pages 255–271, 2018.

[38] Fangneng Zhan, Chuhui Xue, and Shijian Lu. Ga-dan: Geometry-aware domain adaptation network for scene text detection and recognition. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9105–9115, 2019.

[39] Yaping Zhang, Shuai Nie, Wenju Liu, Xing Xu, Dongxiang Zhang, and Heng Tao Shen. Sequence-to-sequence domain adaptation network for robust text image recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2740–2749, 2019.

[40] Han Zhao, Shanghang Zhang, Guanhang Wu, José MF Moura, Joao P Costeira, and Geoffrey J Gordon. Adversarial multiple source domain adaptation. *Advances in neural information processing systems*, 31:8559–8570, 2018.

[41] Yang Zou, Zhiding Yu, BVK Kumar, and Jinsong Wang. Unsupervised domain adaptation for semantic segmentation via class-balanced self-training. In *Proceedings of the European conference on computer vision (ECCV)*, pages 289–305, 2018.

[42] Yang Zou, Zhiding Yu, Xiaofeng Liu, BVK Kumar, and Jinsong Wang. Confidence regularized self-training. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5982–5991, 2019.