Unsupervised Structure-Texture Separation Network for Oracle Character Recognition

Mei Wang, Weihong Deng, Cheng-Lin Liu

Abstract— Oracle bone script is the earliest-known Chinese writing system of the Shang dynasty and is precious to archeology and philology. However, real-world scanned oracle data are rare and few experts are available for annotation which make the automatic recognition of scanned oracle characters become a challenging task. Therefore, we aim to explore unsupervised domain adaptation to transfer knowledge from handprinted oracle data, which are easy to acquire, to scanned domain. We propose a structure-texture separation network (STSN), which is an end-to-end learning framework for joint disentanglement, transformation, adaptation and recognition. First, STSN disentangles features into structure (glyph) and texture (noise) components by generative models, and then aligns handprinted and scanned data in structure feature space such that the negative influence caused by serious noises can be avoided when adapting. Second, transformation is achieved via swapping the learned textures across domains and a classifier for final classification is trained to predict the labels of the transformed scanned characters. This not only guarantees the absolute separation, but also enhances the discriminative ability of the learned features. Extensive experiments on Oracle-241 dataset show that STSN outperforms other adaptation methods and successfully improves recognition performance on scanned data even when they are contaminated by long burial and careless excavation.

Index Terms— oracle character recognition, unsupervised domain adaptation, feature disentanglement, generative adversarial network.

I. Introduction

Historical hieroglyphs are the major sources for knowing ancient civilization and have played an important role in archeology and philology. As the oldest hieroglyphs in China, oracle characters [1], [2] were inscribed on cattle bones or turtle shells about 3000 years ago and provided as key documentary records of human activities during that period. Therefore, the study of oracle characters is of vital importance for Chinese etymology and calligraphy as well as learning the culture and history of ancient China and even the world.

Research on oracle characters began in the late of 1890s shortly after oracle bones were unearthed. Thus far, nearly 4,500 different oracle characters have been discovered, and only about 2,200 characters have been successfully deciphered [3], [4]. To help with the excavation of new oracle bones and the identification of unseen characters, deep convolutional neural networks (CNN) [5], [6], [7], [8] are recently introduced to oracle character recognition [9], [4], [10]. However, training deep models always requires a large number of labeled samples, which becomes the major difficulty. On the one hand, oracle bones are extremely rare, and only limited number of scanned oracle data are available for training. On the other hand, scanned characters are broken and contain serious noises due to long burial and careless excavation, as shown in Fig. 1 and 3. Annotating these data requires a high level of expertise and is extremely hard and time-consuming even for experts in archeology or paleography. Therefore, lacking of enough scanned data with labels is the main obstacle for the development of oracle character recognition systems.

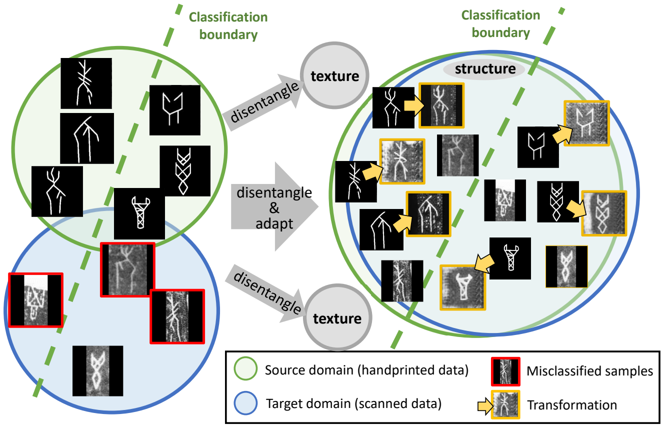


Fig. 1. The model trained on handprinted data cannot perform well on real scanned data due to distribution discrepancy (left). Benefiting from disentanglement, transformation and alignment, handprinted and scanned data are aligned in structure-shared space and the network is optimized by transformed characters, which improve performance on scanned data (right).

In this paper, we propose to transfer knowledge acquired for handprinted character recognition to solve the recognition task of real scanned data. Handprinted oracle characters written by experts are available in large quantities, and can overcome the annotation burden. However, we experimentally prove that these high-quality data suffer from the distribution difference from real scanned data, which results in a dramatic performance drop when directly applying the model trained on hand-

Mei Wang and Weihong Deng are with the Pattern Recognition and Intelligent System Laboratory, School of Artificial Intelligence, Beijing University of Posts and Telecommunications, Beijing, 100876, China. E-mail: {wangmei1,whdeng}@bupt.edu.cn. (Corresponding Author: Weihong Deng)

Cheng-Lin Liu is with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing, 100190, China, and also with the School of Artificial Intelligence, University of Chinese Academy of Sciences, Beijing, 100049, China. E-mail: liucl@nlpr.ia.ac.cn.

printed data to recognize the newly excavated oracle characters, as shown in Fig. 1. Therefore, we propose to adopt unsupervised domain adaptation (UDA) [11] to mitigate the discrepancy, where labeled handprinted samples are set as source data and unlabeled scanned samples as target data. It aims to align the distributions between these two domains by finding a domain-invariant feature space [12], [13], [14], such that the learned model can generalize to scanned domain. While this, in principle, is a good idea, it still faces problems when adapting. 1) Large domain shift between handprinted and scanned oracle data caused by serious noises makes it hard to align the two domains well, and this direct alignment leaves the learned domain-invariant representations vulnerable to the contamination with aspects of noises. 2) Different writing styles lead to high degrees of intra-class variance. Characters belonging to the same category largely vary in stroke and even topology which brings great difficulty for learning discriminative target features when lack of labels.

To address this issue, we propose a structure-texture separation network (STSN) as shown in Fig. 1. First, our STSN learns a triplet of encoders to disentangle the features of individual domains into structure (e.g., glyph) and texture (e.g., noise and resolution) components, and then distribution alignment is safely taken in structure-related feature space. Unlike previous approaches [12], [13] which directly align the entire domains in the whole feature space, separating and adapting the feature space in this manner can mitigate the negative influence caused by serious noises and abrasions. Second, inspired by the generative adversarial nets (GANs) [15], a generator and duplex discriminators are designed to transform images by swapping the learned texture information between any pairs of images, which guarantees the absolute separation of features via a min-max game rather than feature orthogonality [16]. Third, benefitting from image transformation, data augmentation and label transfer are achieved at the same time for target domain. The transformed target-like data follow the distribution of scanned domain and maintain the category information transferred from handprinted domain, which addresses the problem of lacking target labels. A classifier is stacked on the structure representations for the final classification, which also predicts the labels of these target-like data, to pull the samples belonging to the same classes together and thus make target representations more generalized and discriminative.

To facilitate the research towards UDA for oracle character recognition, we present an Oracle-241 dataset[1]. It contains about 80K images of corresponding handprinted and scanned characters belonging to 241 categories. These images were selected from the oracle datasets collected by AnYang Normal University [17], and were rebuilt under unsupervised setting. Each image is centered by one oracle character. Oracle-241 contains extremely serious and unique noises caused by thousands of years of burial and aging, and contains various writing styles in most categories which all make it more difficult to recognize and adapt. Compared with the commonly-used UDA datasets, the categories of Oracle-241 are more than 20 times as many as those of MNIST-USPS-SVHN digit datasets [18], [19], [20].

Our contributions can be summarized into three aspects.

1) To the best of our knowledge, it is the first time that unsupervised domain adaptation is applied in oracle character recognition problem. This research aims to make recognition systems more practical and accurate when utilized to recognize real-world oracle characters, which contributes not only to technology but also to culture heritage preservation and the understanding of oracle characters and ancient civilization.

2) Our proposed STSN simultaneously learns to disentangle, transform, adapt and recognize. Separating texture from structure can avoid the negative influence caused by serious noises when adapting. Instead of transforming conditioned on a binary domain label, our STSN swaps the learned texture information between any pairs of images so that texture and structure of our transformed images are both from real images leading to realistic and diverse results. Simultaneously, the labels of the transformed images are further used in classifier training stage to improve the discrimination ability.

3) Extensive experimental results on a large Oracle-241 dataset show that our method successfully transfers recognition knowledge from handprinted oracle characters to scanned data. It outperforms other UDA methods and offers a state-of-the-art (SOTA) in performance on Oracle-241. Experiments on MNIST-USPS-SVHN digit datasets also show the effectiveness of the proposed method and prove that our STSN has the potential to generalize to a variety of scenarios.

II. Related work

A. Oracle character recognition

Identifying and deciphering oracle characters is one of the most important topics in oracle bone study. As technologies evolve, a handful of works have began to concentrate on oracle character recognition from the perspective of computer vision. Earliest researches [21], [22], [23], [24] regarded oracle character as an undirected graph, and utilized its topological properties as features to perform classification. Liu et al. [25] extracted block histogram-based features and applied support vector machine (SVM) to recognize characters. Guo et al. [9] collected a Oracle-20K dataset which composes of 20K handprinted oracle characters belonging to 261 categories and constructed a novel hierarchical representation. Recently, CNNs have achieved great progress in some computer vision tasks and are introduced into oracle character recognition. Huang et al. [4] released a dataset of scanned oracle characters called OBC306 and presented the CNN-based

[1] Data and code are available on https://github.com/wm-bupt/STSN

evaluation for this dataset to serve as a benchmark. OracleNet [26] considered the radical-level composition of oracle characters, and detected radicals using a Capsule network [27] to recognize characters. Zhang et al. [10] trained a DenseNet [28] with triplet loss and performed classification by nearest neighbor classifier. Cross-modal oracle character recognition was first proposed by Zhang et al. [29]. They utilized adversarial learning and cross-modal triplet loss to perform classification on handprinted and scanned data in a supervised manner. However, our method focuses on UDA where the labels of scanned data are unavailable which is a more realistic but difficult scenario.

B. Unsupervised domain adaptation

Unsupervised domain adaptation [11] has been proposed to address domain shift between labeled source domain and unlabeled target domain. Popular UDA methods [13], [30], [31], [32] align distributions by moment matching. For example, maximum mean discrepancy (MMD) [13], [33], [34], [35] was applied to deep networks to reduce the distribution mismatch. Also, adversarial learning [36], [12], [37], [38] has been widely used for alignment. Domain-adversarial neural network (DANN) [12] made domain classifier fail to predict the domain labels of features by a gradient reversal layer (GRL) such that the feature distributions over the two domains are similar. Conditional adversarial domain adaptation (CDAN) [39] conditioned an adversarial adaptation model on discriminative information conveyed in the classifier predictions. By borrowing the idea of GANs [15], generative adversarial methods [40], [41], [42] transformed source images to target-like ones to improve target performance. For example, cycle-consistent adversarial domain adaptation (CyCADA) [40] utilized CycleGAN [43] to perform pixel-level and feature-level adaptation; while DupGAN [41] proposed a generator with duplex discriminators to restrict the latent representation to be domain invariant, with its category information preserved. Other methods [16], [44] explored to learn the disentangled feature representations to perform better adaptation. For example, domain separation network (DSN) [16] applied feature orthogonality and image reconstruction to disentangle features into domain-specific and domain-invariant components, and performed alignment in domain-invariant space. In this paper, it is the first time that unsupervised domain adaptation is applied in oracle character recognition problem. Different from DSN, our STSN applies image transformation to constrain the disentanglement process. Besides, our transformed images are generated by using the disentangled texture information instead of the binary domain labels, and the transformed images are further utilized to optimize network to improve the performance on scanned domain in an unsupervised manner.

C. Text image editing

Taken a content image and a style image as inputs, text image editing (TIE) [45], [46], [47] aims to replace the text instance in the style image while retaining the styles of both the background and text. SRNet [45] first generated the foreground text by style transfer, then obtained the background image by text erasure, and finally synthesized the edited text image by fusing foreground and background. SwapText [46] further improved SRNet by manipulating geometric points of characters to transform text locations. Shimoda et al. [47] formulated raster text editing as a de-rendering problem. They proposed a vectorization model to parse text information, and edited texts by a rendering engine using the parsed parameters. In principle, TIE can be utilized to mitigate the domain gap between handprinted and scanned images. Replacing the characters in scanned images with handprinted characters can generate labeled training images in scanned domain. However, there are several challenges when training such editing model for oracle characters. 1) Existing TIE methods require pairwise samples for training, i.e., a pair of images with same style but different texts. Although overlaying synthetic text to existing background images can generate pairwise synthetic images for TIE, it is quite difficult to apply this technology to oracle-bone images since background texture information is rare. 2) The ground-truths of style-transfer text and background image are also needed in TIE. However, the backgrounds and foregrounds of real-world scanned oracle images are difficult to segment due to serious noises. 3) Since oracle characters are less investigated and have not been digitalized, it is impossible to parse their rendering parameters, e.g., text, size and font. Considering of these challenges, we instead utilize image transformation in our STSN.

III. Structure-texture separation network

In this section, we describe our STSN method which simultaneously learns to disentangle, transform, adapt and recognize. By disentangling features, it not only aligns handprinted and scanned data in structure-shared feature space, but also swaps textures between any pairs of images to perform transformation. A classifier for final classification learns to predict the labels of transformed images which further enhances the discriminative ability of the learned features.

A. Overview

In our study, we assume that there are M labeled source domain (handprinted) samples $X^s = \{x_i^s\}_{i=1}^M$ with the corresponding labels $Y^s = \{y_i^s\}_{i=1}^M$, and N unlabeled target domain (scanned) samples $X^t = \{x_i^t\}_{i=1}^N$ without any available annotated labels. There is a discrepancy between the distributions of source and target domains. Our goal is to train a handprinted oracle character recognition system that can generalize well to scanned data through training on labeled source domain and unlabeled target domain. Specifically, we desire to obtain domain-invariant features that work equally well in both domains.

An overview of the proposed method is shown in Fig. 2. The networks consist of a structure-shared encoder E_g, two texture-specific encoders $\{E_n^s, E_n^t\}$, a generator G, a feature-level discriminator D_F, two image-level discriminators $\{D_I^s, D_I^t\}$, a pretrained VGGNet, and a classifier C. Given x^s and x^t,

- Encoder $\boldsymbol{E_g}$, $\boldsymbol{E_n^s}$ and $\boldsymbol{E_n^t}$ disentangle and extract structure-shared and texture-specific features of the input images so that the contamination by texture can be avoided in adaptation process, as shown in Fig. 2(a).
- Generator $\boldsymbol{G}$ utilizes the disentangled features to reconstruct the original input images and transform images across domains shown in Fig. 2(a). Better reconstruction and transformation would force encoders to disentangle well.
- Discriminator $\boldsymbol{D_I^s}$ and $\boldsymbol{D_I^t}$ are pitted against G to ensure the transformed images look real from image level. D_I^s and D_I^t distinguish between real and transformed images, while G aims for fooling D_I^s and D_I^t. They are optimized by image-level adversarial loss $\mathcal{L}_{advI}$ in Fig. 2(b).
- VGGNet aims to ensure the reconstructed and transformed images look real from feature level. It extracts their perceptual features and constrains the feature similarity of structure and texture between the generated and real images supervised by reconstruction loss $\mathcal{L}_{rec}$ and perceptual loss $\mathcal{L}_{per}$, as shown in Fig. 2(c).
- Discriminator $\boldsymbol{D_F}$ is pitted against E_g to align source and target features in structure-related space. D_F predicts the domain labels of features, while E_g aims to fool D_F. They are guided by feature-level adversarial loss $\mathcal{L}_{advF}$ shown in Fig. 2(d).
- Classifier $\boldsymbol{C}$ stacked on structure features tries to predict the labels of source and transformed target-like images supervised by classification loss $\mathcal{L}_{cls}^s$ and $\mathcal{L}_{cls}^{st}$, and thus reduces intra-class variance and improves the discriminative ability for target images, as shown in Fig. 2(e).

B. Disentanglement and transformation

Disentanglement: E_g parameterized by θ_{E_g} aims for mapping an image x from either source or target domain to a structure-shared representation f_g that encodes high-level glyph information and is expected to be domain invariant. E_n^s and E_n^t parameterized by $\theta_{E_n^s}$ and $\theta_{E_n^t}$ map source image x^s and target image x^t to texture-specific representations f_n^s and f_n^t respectively, which encode low-level noise information and are private to each domain.

$$f_g = E_g\left(x\right), x \in X^s \cup X^t, \tag{1}$$

$$f_n^s = E_n^s\left(x^s\right), \tag{2}$$

$$f_n^t = E_n^t\left(x^t\right). \tag{3}$$

Transformation: To induce the model to produce such seprated representations and guarantee that $\{f_g^*, f_n^*\}$ are complementary to each other, image reconstruction and transformation are performed later. Specifically, the reconstructed image is generated from G through recombining f_g^* and f_n^* of the same image:

$$x^{ss} = G\left(z^{ss}\right) = G\left(f_g^s, f_n^s, s\right), \tag{4}$$

$$x^{tt} = G\left(z^{tt}\right) = G\left(f_g^t, f_n^t, t\right), \tag{5}$$

where the domain code $\{s, t\}$ is used to specify which domain the latent representation is transformed to. For easy implementation, structure-shared feature, texture-specific feature and domain code are concatenated and utilized as the input of G, e.g., $z^{ss} = [f_g^s; f_n^s; s]$. We assume that the generated images x^{ss} and x^{tt} should be the same as the original input images, i.e., $x^{ss} \sim x^s$ and $x^{tt} \sim x^t$, which is constrained by reconstruction loss introduced in Section III-C. Moreover, to transform images across domains, we swap the texture-specific components between any pairs of source and target images, and then decode them into two unseen images by G:

$$x^{st} = G\left(z^{st}\right) = G\left(f_g^s, f_n^t, t\right), \tag{6}$$

$$x^{ts} = G\left(z^{ts}\right) = G\left(f_g^t, f_n^s, s\right), \tag{7}$$

where x^{st} is the transformed target-like image, which is generated by combining the structure of x^s and the texture of x^t, and x^{ts} is the transformed source-like image generated by the similar way. Intuitively, if E_g and E_n successfully encode the structure and texture representations respectively and these representations are complementary to each other as we expect, x^{st} should look alike the real scanned image x^t in terms of texture appearance while retaining the same glyph information as x^s. It is constrained by the image-level discriminator D_I^t and perceptual loss introduced in Section III-C. Similar constrains are applied on x^{ts}.

C. Objective functions

The overall loss $\mathcal{L}$ of our framework consists of classification loss $\mathcal{L}_{cls}$, feature-level adversarial loss $\mathcal{L}_{advF}$, image-level adversarial loss $\mathcal{L}_{advI}$, perceptual loss $\mathcal{L}_{per}$, and reconstruction loss $\mathcal{L}_{rec}$. Our overall loss can be computed as follows,

$$\mathcal{L} = \mathcal{L}_{cls} + \alpha_1 \mathcal{L}_{advF} + \alpha_2 \mathcal{L}_{advI} + \alpha_3 \mathcal{L}_{per} + \alpha_4 \mathcal{L}_{rec}, \tag{8}$$

where $\{\alpha_i\}_{i=1}^4$ are the parameters for the trade-off between different terms. The detailed optimization procedure of our proposed STSN is depicted in the Algorithm 1. The followings are the details of each loss.

Classification Loss (Fig. 2(e)). With the labeled source data, we are able to learn a basic recognition model in a supervised way,

$$\mathcal{L}_{cls}^s = \underset{C, E_g}{min} \mathbb{E}_{(x^s, y^s) \sim (X^s, Y^s)} L_{ce}\left(C\left(E_g(x^s)\right), y^s\right), \tag{9}$$

where $L_{ce}(\cdot)$ is the cross-entropy loss function. Considering of the lack of target labels, the transformed target-like image x^{st} is able to be utilized as an alternative to supervise learning proceeds for target domain. x^{st} holds the same glyph information with the labeled source data

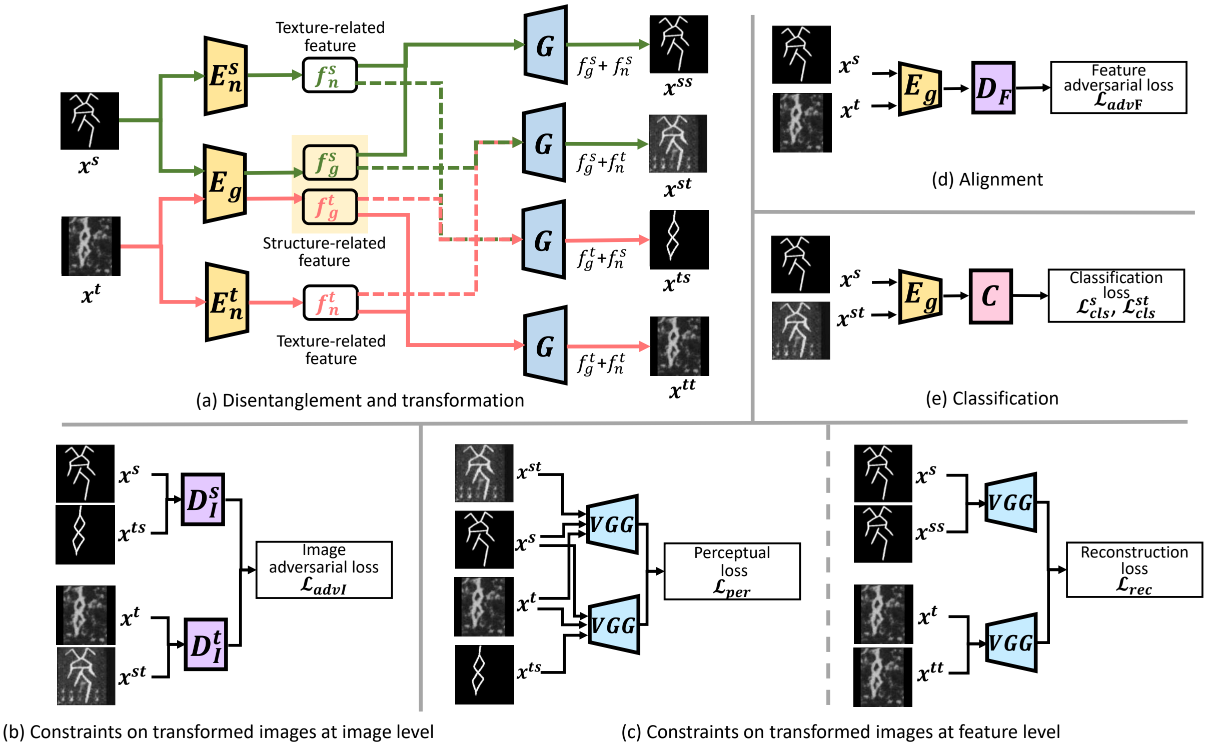

Texture-related feature
E_n^s
f_n^s
x^s
E_g
f_g^s
f_g^t
Structure-related feature
x^t
E_n^t
f_n^t
Texture-related feature
G
$f_g^s+f_n^s$
x^{ss}
$f_g^s+f_n^t$
x^{st}
$f_g^t+f_n^s$
x^{ts}
$f_g^t+f_n^t$
x^{tt}
(a) Disentanglement and transformation
x^s
x^t
E_g
D_F
Feature adversarial loss $\mathcal{L}_{advF}$
(d) Alignment
x^s
x^{st}
E_g
C
Classification loss $\mathcal{L}_{cls}^s, \mathcal{L}_{cls}^{st}$
(e) Classification
x^s
x^{ts}
D_I^s
x^t
x^{st}
D_I^t
Image adversarial loss $\mathcal{L}_{advI}$
(b) Constraints on transformed images at image level
x^{st}
x^s
x^t
x^{ts}
VGG
Perceptual loss $\mathcal{L}_{per}$
x^s
x^{ss}
x^t
x^{tt}
VGG
Reconstruction loss $\mathcal{L}_{rec}$
(c) Constraints on transformed images at feature level

Fig. 2. Illustration of STSN. (a) E_g, E_n^s and E_n^t encode images into structure and texture feature space. Based on the disentangled features, G generates the reconstructed and transformed images. (b) D_I^s and D_I^t are utilized to make the transformed images look real from image level. (c) VGGNet ensures the generated images contain proper structures and textures at feature level. (d) D_F is pitted against E_g to make structure-related features domain-invariant. (e) C stacked on E_g is trained on the source and transformed target-like images which further improves the generalization and discrimination of network.

x^s, so its category is expected to be the same as x^s's. Thus, we optimize the network by the target-like images x^{st} supervised with the labels of x^s,

$$\mathcal{L}_{cls}^{st} = \underset{C,E_g}{min}\mathbb{E}_{(z^{st},y^s)\sim(Z^{st},Y^s)} L_{ce}\left(C\left(E_g\left(G\left(z^{st}\right)\right)\right), y^s\right). \tag{10}$$

Moreover, through transferring glyph information from x^s to x^{st}, the transformed images x^{st} consist of various writing styles as well. With help of $\mathcal{L}_{cls}^{st}$, STSN can push these samples of the same class together and reduce intra-class variances for target domain. Therefore, the classification loss can be computed by $\mathcal{L}_{cls} = \mathcal{L}_{cls}^s + \mathcal{L}_{cls}^{st}$.

Feature-level adversarial loss (Fig. 2(d)). We optimize D_F and E_g in a min-max manner where D_F aims to classify whether a structure-related feature is drawn from the source or the target domain (D_F should output 0 for source features and 1 for the target ones), and E_g aims to fool D_F through generating domain-invariant features which reverse the outputs of D_F. The feature-level adversarial loss can be formulated as follows,

$$\begin{aligned}\mathcal{L}_{advF} = \underset{E_g}{min}\ \underset{D_F}{max}\ & \mathbb{E}_{x^t\sim X^t}\left[\log\left(D_F\left(E_g\left(x^t\right)\right)\right)\right] \\ & + \mathbb{E}_{x^s\sim X^s}\left[\log\left(1 - D_F\left(E_g\left(x^s\right)\right)\right)\right].\end{aligned} \tag{11}$$

By aligning source and target features into a domain-invariant space, we enable our model to learn on source data while still generalizing to target data. Moreover, our feature-level adversarial loss takes place in structure-related feature space so that the negative influence caused by serious noise is avoided when adapting leading to a better generalization.

Image-level adversarial loss (Fig. 2(b)). We employ image-level adversarial loss $\mathcal{L}_{advI}$ to encourage G to generate realistic transformed images. $\mathcal{L}_{advI}$ is applied in two mapping directions, and can be expressed as,

$$\begin{aligned}\mathcal{L}_{advI}^t = \underset{E_g,E_n^t,G}{min}\ \underset{D_I^t}{max}\ & \mathbb{E}_{x^t\sim X^t}\left[\log\left(D_I^t\left(x^t\right)\right)\right] \\ & + \mathbb{E}_{z^{st}\sim Z^{st}}\left[\log\left(1 - D_I^t\left(G\left(z^{st}\right)\right)\right)\right],\end{aligned} \tag{12}$$

$$\begin{aligned}\mathcal{L}_{advI}^s = \underset{E_g,E_n^s,G}{min}\ \underset{D_I^s}{max}\ & \mathbb{E}_{x^s\sim X^s}\left[\log\left(D_I^s\left(x^s\right)\right)\right] \\ & + \mathbb{E}_{z^{ts}\sim Z^{ts}}\left[\log\left(1 - D_I^s\left(G\left(z^{ts}\right)\right)\right)\right].\end{aligned} \tag{13}$$

For the mapping function $G: s+t \rightarrow st$, D_I^t tries to distinguish real target data x^t from the transformed target-like data x^{st}, while G tries to fool D_I^t. Since $z^{st} = [E_g(x^s); E_n^t(x^t); t]$ is utilized as the input of G, the quality of transformed image is also highly dependent on E_g and E_n^t. Therefore, we also optimize E_g and E_n^t to confuse D_I^t so that they can well encode the structure and texture features for realistic transformation. This optimization contributes not only to G, but also to feature disentanglement and so our adaptation process. For the mapping function $G: t+s \rightarrow ts$ and the discriminator

D_I^s, we impose the similar adversarial loss $\mathcal{L}_{advI}^s$ for the transformed source-like image x^{ts}. As proved in [48], the optimization of the standard GAN in Eq. (12) and (13) suffers from instability and may lead to the vanishing gradients problem. To address this issue, we utilize LSGAN [48] instead. Therefore, our image-level adversarial loss can be computed by $\mathcal{L}_{advI} = \mathcal{L}_{advI}^s + \mathcal{L}_{advI}^t$.

Perceptual Loss (Fig. 2(c)). $\mathcal{L}_{advI}$ only makes the transformed samples follow the distribution of real data without consideration of the glyph preservation when transforming. Therefore, we introduce the following perceptual loss to impose the constraints on both structure similarity and texture similarity. For the transformed image x^{st}, it constrains that x^{st} is similar to x^t in terms of texture because of the shared texture component f_n^t; it simultaneously requires that x^{st} is similar to x^s in terms of structure since they share the same f_g^s. Similar constraints are imposed on the transformed image x^{ts}. Some researches [49], [50] indicate that higher feature layers are good at keeping the structure information, while lower layers help to hold texture related things. Therefore, $\mathcal{L}_{per}$ is formulated as follows,

$$\begin{aligned}\mathcal{L}_{per} = \underset{E_g, E_n^t, E_n^s, G}{min} \sum_{l \in \ell_{str}} & \lambda_l (\left\| \phi_l \left(x^s\right) - \phi_l \left(G\left(z^{st}\right)\right) \right\|_1 \\ & + \left\| \phi_l \left(x^t\right) - \phi_l \left(G\left(z^{ts}\right)\right) \right\|_1) \\ + \sum_{l \in \ell_{tex}} & \lambda_l (\left\| \mathcal{U}\left(\phi_l \left(x^t\right)\right) - \mathcal{U}\left(\phi_l \left(G\left(z^{st}\right)\right)\right) \right\|_1 \\ & + \left\| \mathcal{U}\left(\phi_l \left(x^s\right)\right) - \mathcal{U}\left(\phi_l \left(G\left(z^{ts}\right)\right)\right) \right\|_1)\end{aligned} \tag{14}$$

where the first two terms represent structure loss which constrains the structure similarity; while the last two terms represent texture loss used to constrain the texture similarity. $\phi_l(\cdot)$ extracts the activation map from l-th layer of the pretrained VGGNet-16 model [6]. The set of layers ℓ_{str} are the higher layers $\{\ell^{relu4\_1}, \ell^{relu5\_1}\}$ of VGGNet, while ℓ_{tex} are the lower layers $\{\ell^{relu1\_1}, \ell^{relu2\_1}, \ell^{relu3\_1}\}$. λ_l is trade-off parameter for l-th layer. Structure loss constrains L1 distance between the features extracted by higher layers; while texture loss constrains the channel-wisely feature similarity computed on lower layers inspired by AdaIN [51]. In texture loss, $\mathcal{U}(\cdot)$ computes the mean activation across spatial dimensions independently for each channel,

$$\mathcal{U}(f) = \frac{1}{HW} \sum_{h=1}^{H} \sum_{w=1}^{W} f_{chw}, \tag{15}$$

where f_{chw} denotes chw-th element of the feature map, where h and w span spatial dimensions, c is the feature channel.

Reconstruction loss (Fig. 2(c)). As described previously in Section III-B, the reconstructed images x^{ss} and x^{tt} should be the same as the original input images x^s and x^t. Therefore, we employ reconstruction loss to force them to have similar feature representations as computed by VGGNet,

$$\begin{aligned}\mathcal{L}_{rec} = \underset{E_g, E_n^t, E_n^s, G}{min} \sum_{l \in \ell_{rec}} & \lambda_l \left\| \phi_l \left(x^t\right) - \phi_l \left(G\left(z^{tt}\right)\right) \right\|_1 \\ & + \lambda_l \left\| \phi_l \left(x^s\right) - \phi_l \left(G\left(z^{ss}\right)\right) \right\|_1,\end{aligned} \tag{16}$$

where the set of layers $\ell_{rec} = \ell_{tex} \cup \ell_{str}$ consists of both lower and higher layers.

Algorithm 1 Optimization procedure of STSN.

```
Input:
    Labeled handprinted data $\{x_i^s, y_i^s\}_{i=1}^M$, and unlabeled scanned data $\{x_i^t\}_{i=1}^N$.
Output:
    The parameters of whole network $\Theta = \{\hat{\theta}_{E_g}, \hat{\theta}_{E_n^s}, \hat{\theta}_{E_n^t}, \hat{\theta}_{D_F}, \hat{\theta}_{D_I^s}, \hat{\theta}_{D_I^t}, \hat{\theta}_G, \hat{\theta}_C\}$.
Randomly initialize $\hat{\theta}_{E_n^s}$, $\hat{\theta}_{E_n^t}$, $\hat{\theta}_{D_F}$, $\hat{\theta}_{D_I^s}$, $\hat{\theta}_{D_I^t}$, $\hat{\theta}_G$ and $\hat{\theta}_C$, and initialize $\hat{\theta}_{E_g}$ with pretrained ImageNet model.
repeat
    Sample a minibatch of samples from both $X^s$ and $X^t$
    / / Update discriminators by Eq. (11), (12) and (13)
    $\hat{\theta}_{D_F} = \theta_{D_F} - \eta_d \frac{\partial \mathcal{L}_{advF}}{\partial \theta_{D_F}}$
    $\hat{\theta}_{D_I^t} = \theta_{D_I^t} - \eta_d \frac{\partial \mathcal{L}_{advI}^t}{\partial \theta_{D_I^t}}$, $\hat{\theta}_{D_I^s} = \theta_{D_I^s} - \eta_d \frac{\partial \mathcal{L}_{advI}^s}{\partial \theta_{D_I^s}}$
    / / Update generator by Eq. (12), (13), (14) and (16)
    $\hat{\theta}_G = \theta_G - \eta\alpha_2 \frac{\partial \mathcal{L}_{advI}}{\partial \theta_G} - \eta\alpha_3 \frac{\partial \mathcal{L}_{per}}{\partial \theta_G} - \eta\alpha_4 \frac{\partial \mathcal{L}_{rec}}{\partial \theta_G}$
    / / Update encoders by Eq. (11), (12), (13), (14) and (16)
    $\hat{\theta}_{E_n^t} = \theta_{E_n^t} - \eta\alpha_2 \frac{\partial \mathcal{L}_{advI}^t}{\partial \theta_{E_n^t}} - \eta\alpha_3 \frac{\partial \mathcal{L}_{per}}{\partial \theta_{E_n^t}} - \eta\alpha_4 \frac{\partial \mathcal{L}_{rec}}{\partial \theta_{E_n^t}}$
    $\hat{\theta}_{E_n^s} = \theta_{E_n^s} - \eta\alpha_2 \frac{\partial \mathcal{L}_{advI}^s}{\partial \theta_{E_n^s}} - \eta\alpha_3 \frac{\partial \mathcal{L}_{per}}{\partial \theta_{E_n^s}} - \eta\alpha_4 \frac{\partial \mathcal{L}_{rec}}{\partial \theta_{E_n^s}}$
    $\hat{\theta}_{E_g} = \theta_{E_g} - \eta_e\alpha_1 \frac{\partial \mathcal{L}_{advF}}{\partial \theta_{E_g}} - \eta_e\alpha_2 \frac{\partial \mathcal{L}_{advI}}{\partial \theta_{E_g}} - \eta_e\alpha_3 \frac{\partial \mathcal{L}_{per}}{\partial \theta_{E_g}} - \eta_e\alpha_4 \frac{\partial \mathcal{L}_{rec}}{\partial \theta_{E_g}} - \eta_e \frac{\partial \mathcal{L}_{cls}}{\partial \theta_{E_g}}$
    / / Update classifier by Eq. (9) and (10)
    $\hat{\theta}_C = \theta_C - \eta_e \frac{\partial \mathcal{L}_{cls}^s}{\partial \theta_C} - \eta_e \frac{\partial \mathcal{L}_{cls}^{st}}{\partial \theta_C}$
until Convergence
```

D. Discussion

Difference from DSN [16], GSN [52] and the work of Kim et al. [53]. DSN, GSN and Kim et al. separated feature space by image reconstruction and feature orthogonality. Orthogonality of structure and texture is sometimes too weak to guarantee the absolute separation. Moreover, intra-class variances caused by writing style are not able to be well addressed by only alignment. Image transformation is introduced into our STSN. Transformation contributes not only to feature disentanglement but also to data augmentation for target domain. Variances can be reduced by supervising learning process with the transformed target-like images.

Difference from IDEL [54], DRLST [55] and CAAE [56]. IDEL, DRLST and CAAE focus on text style transfer

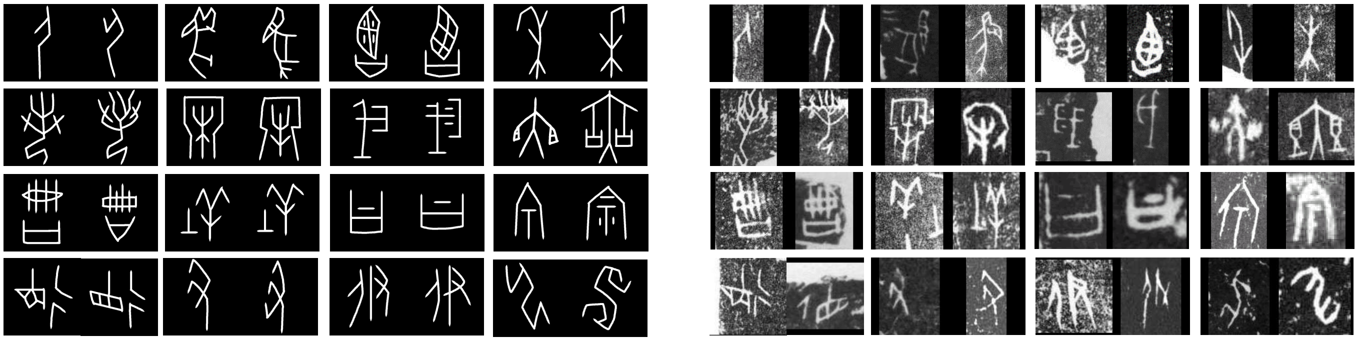


Fig. 3. The handprinted and scanned examples in Oracle-241 dataset. The left eight columns show handprinted samples belonging to 16 classes and the right eight columns show the corresponding scanned samples of the same 16 classes.

in the field of natural language processing. They learned the disentangled features, and aimed to generate style-transfer text. However, STSN focuses on a different recognition task, thus disentanglement and generation are just subtasks. Moreover, our method is different from them in terms of algorithm design. IDEL utilized mutual information to guarantee the disentanglement process, while STSN uses adversarial learning instead. DRLST only generated style-transfer text in inference stage, while STSN generates and utilizes the transformed images for network optimization in training stage. CAAE generated style-transfer samples conditioned on style labels, while STSN swaps the learned texture features to transform images. Instead of sharing the same encoder for learning the separated features, a triplet of encoders is utilized by STSN to avoid the negative mutual-effect.

Difference from CyCADA [40] and DupGAN [41]. All of CyCADA, DupGAN and our STSN follow the idea of GAN to achieve domain transformation for domain adaptation. CyCADA and DupGAN directly transformed images conditioned on a binary domain label. However, benefiting from feature disentanglement, the images in our STSN are generated by swapping texture-specific components between any pairs of source and target images so that texture and structure of our transformed images are both from real images.

Difference from ASSDA [57], AHWR [58] and the work of Luo et al. [59]. ASSDA and AHWR focus on UDA of text recognition. Although they took the characteristics of text into consideration and proposed temporal pooling and character-level alignment to address the adaptation of sequence-like word images, they directly aligned domains in the whole feature space. In our STSN, structure information is disentangled from texture and alignment is adopted in structure-related feature space. Moreover, our method is capable of image transformation across domains, while ASSDA and AHWR are not. Luo et al. [59] aim to improve text recognition by removing backgrounds while retaining the text content. They shared attention masks from the recognizer to the discriminator to enable a character-level adversarial training when performing image transformation. However, our method is capable of feature disentanglement and alignment, while the work of Luo et al. is not.

IV. Experiments

In this section, we perform UDA experiments on our Oracle-241 for oracle character recognition and on MNIST-USPS-SVHN [18], [19], [20] for digit classification, which prove the effectiveness of our method. The details are as follows.

A. Datasets

Oracle-241 is a large oracle character dataset for transferring knowledge from handprinted characters to scanned data, as shown in Fig. 3. Since oracle characters have not been digitalized, they are stored or displayed by images in computers. We selected the oracle images from the datasets collected by AnYang Normal University [17] and built them under UDA setting. Our Oracle-241 contains 78,565 handprinted and scanned characters of 241 categories, and each image is centered by one single character. Scanned oracle images are generated by reproducing the oracle-bone surface by placing a piece of paper over the subject and then rubbing the paper with rolled ink. The aging process has made the inscriptions less legible so that these scanned characters are broken and exist serious noises. Handprinted oracle characters are written by experts, and their corresponding images are clean and high-definition. We split Oracle-241 into two subsets for training and testing as shown in Table I. The training set consists of labeled 10,861 handprinted data and unlabeled 50,168 scanned data; while the testing set contains 3,730 handprinted data and 13,806 scanned data.

TABLE I

Statistic of our Oracle-241 dataset.

Subsets	Train		Test	
	# Classes	# Images	# Classes	# Images
Handprint	241	10,861	241	3,730
Scan	241	50,168	241	13,806

TABLE II

Network Architectures of our Encoder and Generator used for oracle recognition.

Encoder (E_n)	Generator (G)
Input: $x \in \mathbb{R}^{224\times224\times3}$	Input: f_g, f_n, domain code

Conv(k7n64s2), BN, ReLU
Conv(k7n64s2), BN, ReLU
Conv(k3n128s2), BN, ReLU
{Conv(k3n256s2), BN, ReLU}× 3
	Deconv(k4n256s2), IN, ReLU, ConvBlock(k3n128s1)
	Deconv(k4n128s2), IN, ReLU, ConvBlock(k3n64s1)
	Deconv(k4n64s2), IN, ReLU, ConvBlock(k3n32s1)
	{Deconv(k4n32s2), IN, ReLU, ConvBlock(k3n32s1)}× 2

AvgPool, Conv(k1n8s1)	Conv(k1n3s1), Tanh
Output: f_n	Output: x^{ss}/ x^{st}/ x^{ts}/ x^{tt}

TABLE III

Network Architectures of our Discriminators used for oracle recognition.

Discriminator (D_I)	Discriminator (D_F)
Input: $x \in \mathbb{R}^{224\times224\times3}$	Input: f_g^s, f_g^t

Conv(k6n64s2), IN, LReLU(0.2)
Conv(k4n128s2), IN, LReLU(0.2)
Conv(k4n256s2), IN, LReLU(0.2)
Conv(k4n512s2), IN, LReLU(0.2)
	Linear(1024), ReLU
	Dropout(0.5)
	Linear(1024), ReLU
	Dropout(0.5)

Linear(1)	Linear(1), Sigmoid
Output: Real/Fake	Output: Source/Target

MNIST-USPS-SVHN [18], [19], [20] are the commonly used digit datasets for domain adaptation. They contain digital images of 10 classes. MNIST consists of a training set of 60,000 grey images and a testing set of 10,000 grey images. USPS contains 7,291 and 2,007 grey images for training and testing. SVHN contains 73,257 and 26,032 digits for training and testing. Following previous works [40], we perform domain adaptation on three transfer tasks: MNIST→USPS (M→U), USPS→MNIST (U→M) and SVHN→MNIST (S→M).

B. Implementation detail

Network architecture. For oracle character recognition, ResNet-18 [8] pretrained on Imagenet [60] is utilized as structure encoder E_g. Texture encoders $\{E_n^s, E_n^t\}$ contain 6 convolution blocks (Conv-BN-ReLU) followed by an average pooling layer and a 1×1 convolution layer, and share the first convolution block with E_g. The texture information is encoded into a 8-dimension feature. For G and $\{D_I^s, D_I^t\}$ networks, we use architectures similar to those used in DCGAN [61]. We apply 5 stacked layer groups as G, and each layer group consists of a 4×4 transposed convolution layer [62], one IN layer [63], one ReLU layer and one convolution block (Conv-LN-ReLU). Then 1×1 convolution layer is applied to yield 224×224×3 images. $\{D_I^s, D_I^t\}$ contain 4 convolution blocks (Conv-IN-LReLU), followed by a fully connected layer. For D_F, we use the same discriminative network structure as DANN [12]. Details are shown in Table II and III.

For digit classification, we use a variant of the LeNet architecture as E_g following recent works [40], [39]. The same architecture is applied to $\{E_n^s, E_n^t\}$. Other networks are similar to those used in oracle character recognition.

Experimental setup. We train our model for 150,000 iterations over the data with batch size 16. For preprocessing, we randomly crop and flip the training samples. We adopt mini-batch SGD with momentum of 0.9 to train E_g and C, and set weight decay and initial learning rate to $5e{-}4$ and $2.5e{-}4$, respectively. We use Adam solver with base learning rate of $1e{-}4$ and momentum of 0.99 to train the discriminators D_F and $\{D_I^s, D_I^t\}$, and use Adam solver with base learning rate of $1e{-}3$ and momentum of 0.99 to train $\{E_n^s, E_n^t\}$ and G. The learning rate is annealed by $\eta = \eta_0 \left(\frac{1-T}{T_{max}}\right)^{0.9}$, where η_0 is the initial learning rate, T and T_{max} are the current and total iteration. The parameters $\{\alpha_i\}_{i=1}^4$ are set to be 1, 0.01, 0.05 and 0.5, respectively. $\{\lambda_l\}_{l\in\ell_{rec}}$ are set to be 1/32, 1/16, 1/8, 1/4 and 1, respectively, which emphasize more on higher layer. $\{\lambda_l\}_{l\in\ell_{str}}$ share the same value with $\{\lambda_l\}_{l\in\ell_{rec}}$, and $\{\lambda_l\}_{l\in\ell_{tex}}$ are set to be 1 for all layers.

Evaluation protocol. We follow the standard protocols for unsupervised domain adaptation as [12], [13]. All labeled source characters and all unlabeled target characters are utilized for training, and the average classification accuracy and standard deviation are reported based on three random experiments.

C. Ablation study

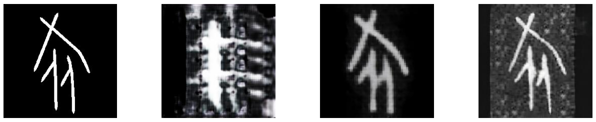
To take a further step into different components within our method, we report the results of 5 variations of STSN in Table IV. Specifically, we remove some of $\mathcal{L}_{advF}$, $\mathcal{L}_{advI}$, $\mathcal{L}_{rec}$, $\mathcal{L}_{per}$ and $\mathcal{L}_{cls}^{st}$ from the full model. From the results, we can find that each component makes an important contribution to the target performance and the quality of transformed images.

Effectiveness of alignment (Model-A v.s. STSN). We first investigate the effectiveness of $\mathcal{L}_{advF}$ which aligns source and target features in structure-related space. We perform ablation study with Model-A, which is built by

TABLE IV

Ablation study on Oracle-241 dataset.

Methods	Components						Scan
	$\mathcal{L}^{s}_{cls}$	$\mathcal{L}_{advF}$	$\mathcal{L}_{advI}$	$\mathcal{L}_{rec}$	$\mathcal{L}_{per}$	$\mathcal{L}^{st}_{cls}$	
Source-only	✓	✗	✗	✗	✗	✗	2.1
Model-A	✓	✗	✓	✓	✓	✓	31.8
Model-B	✓	✓	✗	✗	✗	✗	39.5
Model-C	✓	✓	✓	✗	✗	✗	40.3
Model-D	✓	✓	✓	✓	✗	✗	41.2
Model-E	✓	✓	✓	✓	✓	✗	42.7
STSN (ours)	✓	✓	✓	✓	✓	✓	47.1



(a) Source (b) Model-C (c) Model-D (d) Model-E

Fig. 4. Ablation: the effect of $\mathcal{L}_{advI}$, $\mathcal{L}_{rec}$ and $\mathcal{L}_{per}$ on transformation. (a) Source handprinted character. (b) Without $\mathcal{L}_{rec}+\mathcal{L}_{per}$, the transformed image is not satisfied. (c) Without $\mathcal{L}_{per}$, the transformed image lacks the proper texture. (d) With all the constraints, the transformed image looks real from both structure and texture aspects.

only eliminating $\mathcal{L}_{advF}$. Compared with STSN, the performance of Model-A drops 15.3%. It shows that alignment between the two domains indeed helps to improve the model generalization. Moreover, $\mathcal{L}_{advF}$ makes structure features domain-invariant and texture-independent leading to better disentanglement.

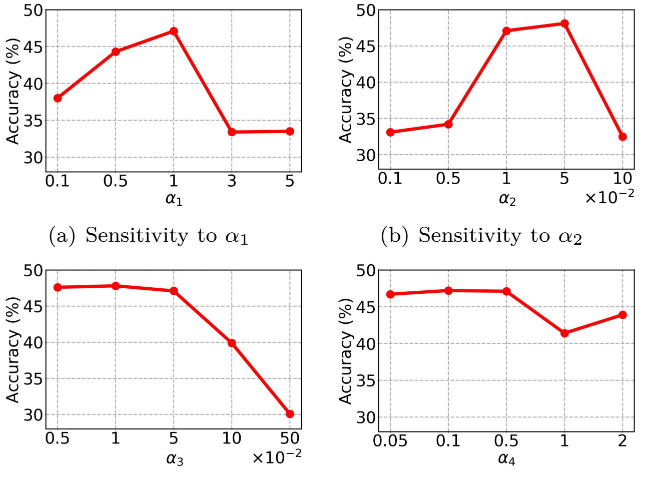
Effectiveness of disentanglement (Model-B v.s. STSN). Model-B removes all feature disentanglement modules from STSN, and thus it only globally aligns source and target domains in the whole feature space. From the comparison results between Model-B and our STSN, we can see that dramatic decrease happens in the target accuracy of Model-B. This result reveals the effect of disentanglement is important for adaptation process, since alignment can be adopted in structure-related feature space so that adaptation process is protected from being contaminating by texture leading to the performance improvement.

Effectiveness of transformation (Model-E v.s. STSN). STSN further optimizes the network using the transformed target-like images supervised by $\mathcal{L}^{st}_{cls}$. To explore its effect, we also perform ablation study with Model-E, i.e., STSN w/o $\mathcal{L}^{st}_{cls}$. Compared with STSN, Mode-E suffers a degradation of 4.4% w.r.t. target accuracy, manifesting the importance of the proposed transformation module. The transformed target-like images achieve data augmentation and label transfer, and supervision on them helps to reduce intra-class variance in target domain.

Quality of transformed image (Model-C v.s. Model-D v.s. Model-E). We further analyze the effect of $\mathcal{L}_{advI}$, $\mathcal{L}_{rec}$ and $\mathcal{L}_{per}$ on transformation. Model-C represents the transformation module w/o $\mathcal{L}_{rec}+\mathcal{L}_{per}$, Model-D denotes the module w/o $\mathcal{L}_{per}$, and Mode-E is the full transformation module. The transformed images generated by these models are visualized in Fig. 4. Fig. 4(b) demonstrates Model-C with $\mathcal{L}_{advI}$ alone fails to produce the desired behavior for our task. When adding $\mathcal{L}_{rec}$, Model-D can generate images with the proper glyph information but still fails to match the target style due to the absence of a proper constraint of texture similarity. In Fig. 4(d), the full transformation module can generate a satisfactory transformed image, indicating that $\mathcal{L}_{advI}$, $\mathcal{L}_{rec}$ and $\mathcal{L}_{per}$ all make important contributions to transformation. Moreover, we assume that the quality of transformed images will affect feature disentanglement and so the adaptation performance. Therefore, we also report the corresponding adaptation results of Model-C, Model-D and Model-E in Table IV. It can be observed Model-E achieves the best performance among them which also proves our assumption.

Parameter Sensitivity. $\{\alpha_i\}_{i=1}^{4}$ are the trade-off parameters for different losses in Eq. (8). We investigate the target performance of STSN when these parameters change, and show the results in Fig. 5. It can be observed that the accuracy first increases and then decreases as α_1 and α_2 vary, demonstrating a desirable bell-shaped curve. In principle, smaller value of α would avoid STSN learning the disentangled and aligned features well; while larger value of α will overemphasize on alignment and lead to negative transfer. We also observe that a small α_3 (e.g., 0.05) tends to have a better performance, but the accuracy would suffer an obvious decline when $\alpha_3 > 0.05$, since a larger α_3 will weaken the effect of $\mathcal{L}^{s}_{cls}$ term and degrade the classification performance. For α_4, the accuracy is stable under a wide range of parameter values, indicating STSN is not quite sensitive to α_4.

Moreover, we also check the sensitivities of λ_l in $\mathcal{L}_{rec}$ and $\mathcal{L}_{per}$. In our method, the values of $\{\lambda_l\}_{l\in\ell_{rec}}$ are increased along with the layer number, and we call it ASC model. We build two variants, i.e., SAME and DEC, by changing the values of λ_l, and compare ASC with them shown in Table V. $\{\lambda_l\}_{l\in\ell_{rec}}$ keep the same for all layers in SAME model, but are decreased along with the layer number in DEC model. From the results, we can see that ASC model outperforms other comparison models. Since the preservation of content and semantic information is



(a) Sensitivity to α_1

(b) Sensitivity to α_2

(c) Sensitivity to α_3

(d) Sensitivity to α_4

Fig. 5. The sensitivity of target accuracy to $\{\alpha_i\}_{i=1}^4$ when adapting handprinted oracle data to scanned data.

TABLE V

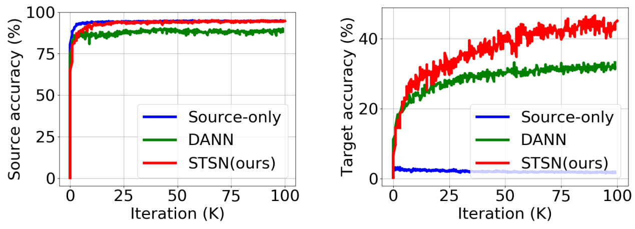
The empirical analysis results for the sensitivity of hyperparameter λ_l

Methods	Domain	ASC	SAME	DES
$\{\lambda_l\}_{l\in\ell_{rec}}$	Source	95.0	94.2	89.6
	Target	47.1	43.7	43.1
$\{\lambda_l\}_{l\in\ell_{str}}$	Source	95.0	94.1	94.1
	Target	47.1	32.0	36.3
$\{\lambda_l\}_{l\in\ell_{tex}}$	Source	94.2	95.0	94.6
	Target	34.3	47.1	35.0

more difficult but meaningful for oracle recognition, we prefer to pay more attention to higher layers in our feature reconstruction loss. Similar experiments are performed on $\{\lambda_l\}_{l\in\ell_{str}}$ and $\{\lambda_l\}_{l\in\ell_{tex}}$. We observe that ASC model also performs best in structure loss, while SAME model is superior to others in texture loss. Therefore, it is necessary to set proper weight for each layer.

D. Visualization

Convergence. We empirically study the convergence property of the proposed model. Fig. 6 illustrates the corresponding test accuracies with respect to the number of iterations. From the figure, we can see that the accuracy is monotonically increasing with the increase of iterations, and STSN gets converged after several iterations. STSN achieves much better results than DANN [12], but shows lower convergence rate on target domain. We conjecture the reason is as follows. Our STSN boosts target performance by disentanglement, alignment and transformation. However, both disentanglement and transformation largely depend on the generator G. Absolute disentanglement and realistic transformed images cannot even be achieved until G converges. Therefore, the convergence rate of STSN may be slowed down due to the optimization of the generator G.



(a) Source domain

(b) Target domain

Fig. 6. The testing accuracies of Source-only, DANN and our proposed method evaluated on source and target domain when adapting handprinted oracle characters to scanned data.

Image visualization. Fig. 7 visualizes the transformed target-like images of our method on Oracle-241. As seen, our method can effectively generate the target-like images with different types of noises and abrasions. The images in the first row are transformed into some incomplete characters covered by some white regions. The second row shows the transformed images suffered from serious abrasions similar to salt and pepper noise. In the third row, the generated images are contaminated by the added strokes or the omitted strokes. Moreover, scanned characters are generated by placing a piece of paper over oracle-bone and then rubbing the paper with rolled ink. The manually applied pressure while rubbing causes the variations of contrast and makes strokes differ in thickness. This type of transformed images is shown in the last row. Fig. 8 shows some transformed source-like images which also proves the effectiveness of our transformation method. Although these denoising images are not utilized to optimize network, they play important roles in disentanglement. We also show the transformed images on MNIST-USPS-SVHN datasets in Fig. 9. These transformed images all look alike the real target images regarding texture while successfully retaining the same glyph information as source images.

Feature visualization. To demonstrate the effectiveness of our method on reducing domain shift, we extract source and target features of Oracle-241 dataset using "Source-only" and our STSN model, and visualize the learned features using t-SNE [64]. "Source-only" model is trained on handprinted data without any adaptation. The results are shown in Fig. 10. As we can see, the distributions are separated between domains before adaptation. After adaptation, features are successfully fused and thus there is no domain discrepancy between different domains. The similar observation can be obtained on digit datasets shown in Fig. 11. Furthermore, to demonstrate the effectiveness of our method on reducing intra-class variance, we also visualize target features on Oracle-241 using t-SNE. We randomly select some scanned oracle characters of 20 classes, then extract and visualize their features by "Source-only" and our STSN model. The results are shown in Fig. 12. We can observe that features belonging to different classes are mixed together before adaptation while our STSN features are more discriminative. The features in the same class are mapped close and features

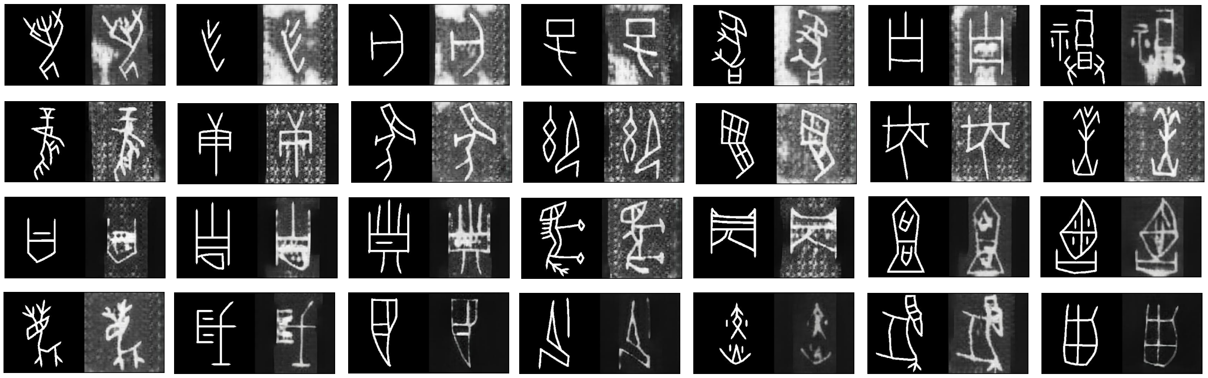


Fig. 7. The transformed target-like images on Oracle-241 dataset. In every two columns, the left and right images are the images from real handprinted domain and their corresponding transformed images in target domain.

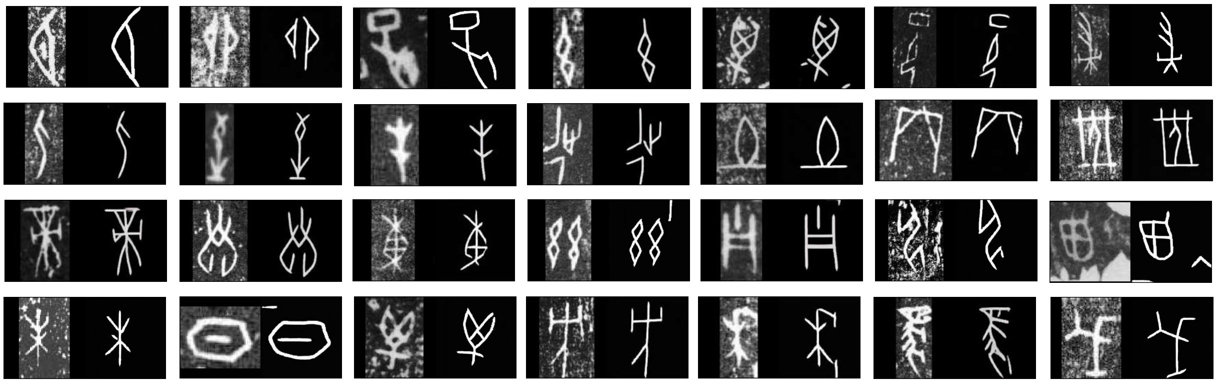
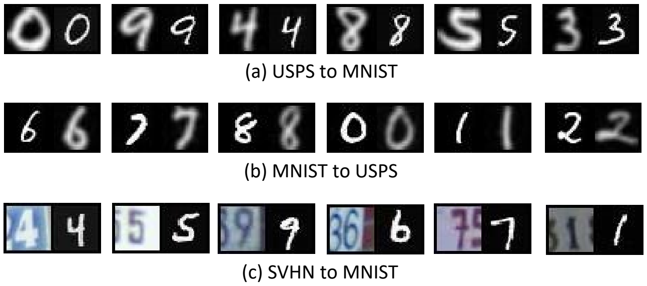


Fig. 8. The transformed source-like images on Oracle-241 dataset. In every two columns, the left and right images are the images from real scanned domain and their corresponding transformed images in source domain.

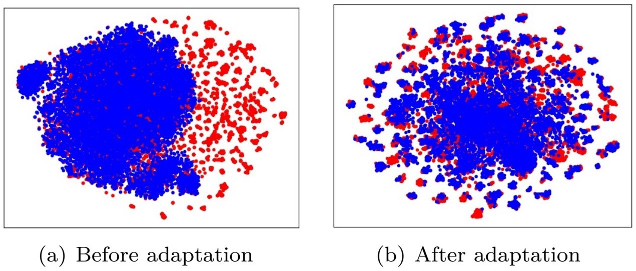


(a) USPS to MNIST

(b) MNIST to USPS

(c) SVHN to MNIST

Fig. 9. The transformed target-like images on (a) U→M, (b) M→U and (c) S→M tasks. In every two columns, the left and right images are the images from source domain and their corresponding transformed images in target domain.



(a) Before adaptation

(b) After adaptation

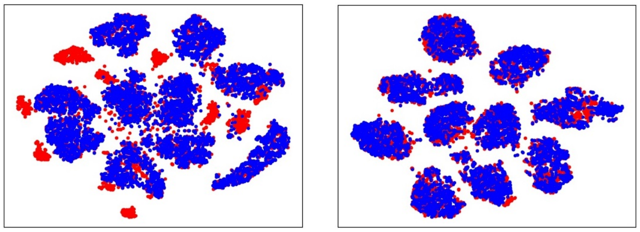
Fig. 10. Feature visualization on testing data of Oracle-241 dataset. Red points are handprinted samples and the blue ones represent scanned samples. (Best viewed in color)

of different classes are separated to some extent.

E. Comparison with state-of-the-arts

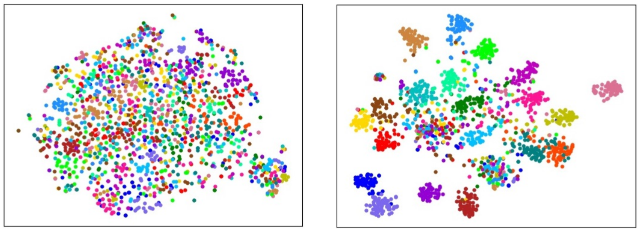
Results on Oracle-241. Here we transfer recognition knowledge from handprinted oracle characters to scanned data. To prove its effectiveness, our STSN is compared with some methods designed for oracle character recognition [4], [10]. Since they only trained models on handprinted data without any adaptation, we call them "Source-only" models. Several SOTA adaptation methods used for image classification, e.g., CDAN [39], BSP [65] and GVB [66], are also compared. All of them are utilized ResNet-18 as backbone for fair comparison. The quantitative results are described as follows.

From the results shown in Table VI, we have the following observations. Firstly, when trained on handprinted data and tested on the same domain, "Source-only" models can achieve high accuracies. However, the performances dramatically decrease when they are directly applied on scanned data, which brings large challenges on recognizing real-world oracle characters. Although Zhang et al. [10] introduced triplet loss and nearest neighbor classifier to oracle character recognition, they ignored the distribution discrepancy between handprinted and scanned data. STSN is the first one to discuss this issue, and applies UDA in



(a) Before adaptation (b) After adaptation

Fig. 11. Feature visualization on U→M task. Red points are USPS samples and the blue ones represent MNIST samples. (Best viewed in color)



(a) Before adaptation (b) After adaptation

Fig. 12. Feature visualization on target scanned data of Oracle-241 dataset. The color of a circle represents its class. (Best viewed in color)

oracle character recognition.

Secondly, we notice that existing UDA methods can successfully improve target performance by learning a domain-invariant feature space. This phenomenon shows the importance of mitigating the domain shift. However, this improvement is still limited for target domain and the source accuracy is decreased unfortunately after adapting. For example, ASSDA [57] which adopted both image-level and character-level alignment only achieves 32.6% on target domain. Without consideration of texture-specific information contained in each domain, it is quite challenging to align the whole source and target domains well. Moreover, hard examples with excessive domain characteristics would have negative effect in alignment and so the performances of both two domains. Although GVB [66] published in CVPR'20 utilized a bridge, i.e., one fully connected layer, to model domain-specific parts, the bridge with simple architecture is hard to capture these properties well.

TABLE VI
Source and target accuracy (mean±std%) on Oracle-241 dataset (the handprinted characters to scanned data setting). The best numbers are indicated in bold.

Methods		Handprint	Scan
Source-only	ResNet [4], [8]	94.9±0.1	2.1±0.6
	Zhang et al. [10]	94.5±0.4	8.4±1.0
Adaptation	CORAL [30]	89.5±0.6	18.4±1.3
	DDC [67]	90.8±1.5	25.6±1.9
	DAN [13]	90.2±1.5	28.9±1.6
	ASSDA [57]	85.8±0.1	32.6±0.2
	DANN [12]	87.1±1.7	32.7±1.5
	GVB [66]	92.8±0.4	36.8±1.1
	CDAN [39]	85.3±3.3	37.9±2.0
	BSP [65]	87.7±0.7	43.7±0.4
	STSN (ours)	95.0±0.2	47.1±0.8

TABLE VII
Target accuracy (mean± std%) on three transfer tasks of digit datasets. The best numbers are indicated in bold.

Methods	U→M	M→U	S→M	Avg
Source-only [8]	69.6±3.8	82.2±0.8	67.1±0.6	73.0
DSN w/ MMD [16]	-	-	72.2	-
DANN [12]	-	77.1±1.8	73.6	-
ADDA [36]	90.1±0.8	89.4±0.2	76.0±1.8	85.2
DAA [68]	92.8±1.1	90.3±0.2	78.3±0.5	87.1
LEL [69]	-	-	81.0±0.3	-
DSN w/ DANN [16]	-	-	82.7	-
DTN [70]	-	-	84.4	-
AsmTri [71]	-	-	86.0	-
CoGAN [72]	-	-	91.2±0.8	-
MSTN [73]	-	92.9±1.1	91.7±1.5	-
PixelDA [74]		95.9		-
UNIT [75]	93.6	95.9	90.5	93.4
CyCADA [40]	96.5±0.1	95.6±0.2	90.4±0.4	94.2
STSN (ours)	96.7±0.1	94.4±0.3	92.2±0.1	94.4

Lastly, benefitting from joint adaptation, disentanglement and transformation, our STSN outperforms other competitors significantly, which proves its effectiveness. To be specific, it achieves 47.1% when tested on scanned data, which is higher than CDAN [39] by 9.2% and higher than the best baseline BSP [65] by 3.4%. Moreover, it also maintains better performance on source domain after adapting because alignment in structure-related feature space protects adaptation process from the contamination by texture.

Results on MNIST-USPS-SVHN. We additionally perform UDA experiments on other text datasets, i.e., MNIST-USPS-SVHN. We follow the evaluation protocol in [40], [39] and report the target accuracy on each transfer task. Results of digit datasets are shown in Table VII. DSN [16] which disentangled features by image reconstruction and feature orthogonality only achieves 82.7% on S→M task. CyCADA [40] which took advantage of CycleGAN [43] to generate the transformed target-like images largely boosts the target performance and achieves 90.4% on S→M task. Our proposed model outperforms all comparison methods on almost all the tasks. The utilization of transformation to constrain the disentanglement process enables the absolute separation and the safe alignment. Instead of using the binary domain labels, transforming images using the learned texture information makes our transformed images more realistic and diverse such that intra-class variances can be better reduced. After adaptation, our method achieves an accuracy of 96.7% on

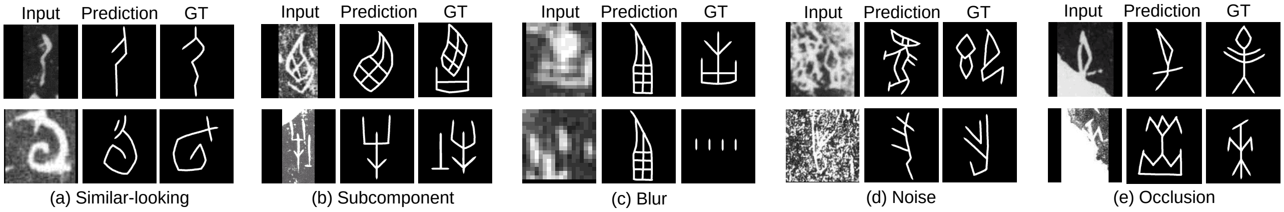


Fig. 13. Ten example samples which are misclassified by our STSN model. For each character, the left, middle and right images are the misclassified scanned sample, prediction and ground-truth, respectively. The predictions and ground-truths are showed visually by the corresponding handprinted images belonging to the same classes.

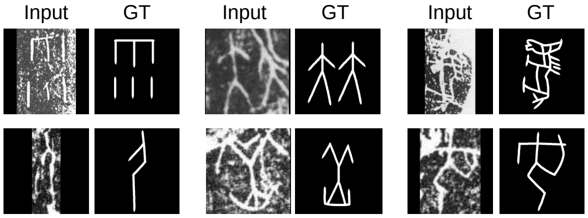


Fig. 14. Six example images which are misclassified by "Source-only" model but classified correctly by our method. The ground-truths are showed by the corresponding handprinted images belonging to the same classes.

U→M task, and achieves 92.2% on S→M task. The results further demonstrate the advantage of our STSN method.

F. Error analysis

Some example images which are misclassified by "Source-only" model [8] but classified correctly by our STSN are presented in Fig. 14. From the results, we can find that "Source-only" model fails to recognize the scanned images with serious noises and abrasions while our STSN can succeed. However, more than 50% of scanned images are still misclassified by our method. We show some of them in Fig. 13, and conjecture the reasons may be as follows. First, our model is sometimes confused by some similar-looking characters. For example, the characters of the predicted category and the ground-truth (GT) category only differ in some small details shown in Fig. 13 (a). Second, some characters are subcomponents of the others which also makes our model more error prone as shown in Fig. 13 (b). Third, the images are seriously degraded and even completely lost their discriminative glyph information caused by blur (Fig. 13 (c)), noise (Fig. 13 (d)) and occlusion (Fig. 13 (e)). Recognizing these characters is challenging even for human.

V. Conclusion and future work

To address the problem of lack of massive labeled real-world scanned oracle data, we are the first to apply unsupervised domain adaptation in oracle character recognition. The main contribution of this paper is that we present a publicly available oracle character database for domain adaptation, and propose a structure-texture separation network (STSN) to transfer recognition knowledge from handprinted oracle characters to real scanned data. STSN is an end-to-end learning framework for joint disentanglement, transformation, adaptation and recognition. Unlike previous approaches which globally align the two domains, our STSN disentangles features and applies alignment in structure-related space. Moreover, domain transformation by swapping textures is adopted to achieve data augmentation, reduce intra-class variance and guarantee the absolute separation of features. Experiments on our proposed Oracle-241 dataset have convincingly demonstrated that STSN successfully improves the performance on scanned data and outperforms several SOTA methods.

However, the accuracy on scanned data is less than 50%, and there is still room for improvement. Hence, one future trend is to investigate more effective and efficient methods, e.g., pseudo-labels, to further improve the generalization ability of recognition model. Moreover, considering that it is possible to access a small amount of labeled scanned data, we aim to extend our method from the unsupervised setting to semi-supervised setting in the future.

Recognizing oracle data accurately is an open issue for both human and automatic system. Human expert work is still needed, but our system can lighten their workload. Given an oracle character, our STSN can provide the predicted probability to experts and help them with decision-making. When the system confidently identifies which category the sample belongs to, experts only need to check the result. When the system fails to give a confident prediction, this sample may be difficult to recognize (or has not been deciphered) and thus needs to be further identified by experts. Considering that about 2,200 characters have been successfully deciphered, we plan to cooperate with some experts and collect more oracle characters to cover all of the 2,200 categories such that the trained system can be more practical.

VI. Acknowledgments

This work was partially supported by the National Natural Science Foundation of China under Grants No. 61871052 and 62192784.

References

[1] R. K. Flad, S. Allan, R. Campbell, X. Chen, L. von Falkenhausen, H. Fang, M. Fiskesjö, Z. Jing, D. N. Keightley, E. Kyriakidis et al., "Divination and power: a multiregional view of the development of oracle bone divination in early china," Current Anthropology, vol. 49, pp. 403–437, 2008.

[2] D. N. Keightley, "Graphs, words, and meanings: Three reference works for shang oracle-bone studies, with an excursus on the religious role of the day or sun," Journal of the American Oriental Society, vol. 117, pp. 507–524, 1997.

[3] B. Li, Q. Dai, F. Gao, W. Zhu, Q. Li, and Y. Liu, "HWOBC-A handwriting oracle bone character recognition database," Journal of Physics: Conference Series, vol. 1651, pp. 1–7, 2020.

[4] S. Huang, H. Wang, Y. Liu, X. Shi, and L. Jin, "OBC306: A large-scale oracle bone character recognition dataset," in Proceedings of International Conference on Document Analysis and Recognition, 2014, pp. 1–14.

[5] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in neural information processing systems, vol. 25, pp. 1097–1105, 2012.

[6] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in Proceedings of International Conference on Learning Representations, 2014, pp. 1–14.

[7] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1–9.

[8] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 770–778.

[9] J. Guo, C. Wang, E. Roman-Rangel, H. Chao, and Y. Rui, "Building hierarchical representations for oracle character and sketch recognition," IEEE Transactions on Image Processing, vol. 25, pp. 104–118, 2015.

[10] Y.-K. Zhang, H. Zhang, Y.-G. Liu, Q. Yang, and C.-L. Liu, "Oracle character recognition by nearest neighbor classification with deep metric learning," in Proceedings of International Conference on Document Analysis and Recognition, 2019, pp. 309–314.

[11] M. Wang and W. Deng, "Deep visual domain adaptation: A survey," Neurocomputing, vol. 312, pp. 135–153, 2018.

[12] Y. Ganin and V. Lempitsky, "Unsupervised domain adaptation by backpropagation," in Proceedings of International conference on machine learning, 2015, pp. 1180–1189.

[13] M. Long, Y. Cao, J. Wang, and M. Jordan, "Learning transferable features with deep adaptation networks," in Proceedings of International conference on machine learning, 2015, pp. 97–105.

[14] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Unsupervised domain adaptation with residual transfer networks," Advances in Neural Information Processing Systems, vol. 29, pp. 136–144, 2016.

[15] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," Advances in neural information processing systems, vol. 27, pp. 2672–2680, 2014.

[16] K. Bousmalis, G. Trigeorgis, N. Silberman, D. Krishnan, and D. Erhan, "Domain separation networks," Advances in neural information processing systems, vol. 29, pp. 343–351, 2016.

[17] "Anyang normal university. OBI database, 2019." http://jgw.aynu.edu.cn/ajaxpage/home2.0.

[18] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," Proceedings of the IEEE, vol. 86, pp. 2278–2324, 1998.

[19] Y. Netzer, T. Wang, A. Coates, A. Bissacco, B. Wu, and A. Y. Ng, "Reading digits in natural images with unsupervised feature learning," in Proceedings of International Conference on Neural Information Processing Systems Workshops, 2011, pp. 1–7.

[20] J. S. Denker, W. Gardner, H. P. Graf, D. Henderson, R. E. Howard, W. Hubbard, L. D. Jackel, H. S. Baird, and I. Guyon, "Neural network recognizer for hand-written zip code digits," in Proceedings of International Conference on Neural Information Processing Systems, 1989, pp. 323–331.

[21] X.-L. Zhou, X.-C. Hua, and F. Li, "A method of jia gu wen recognition based on a two-level classification," in Proceedings of International Conference on Document Analysis and Recognition, 1995, pp. 833–836.

[22] F. Li and P.-y. Woo, "The coding principle and method for automatic recognition of jia gu wen characters," International Journal of Human-Computer Studies, vol. 53, pp. 289–299, 2000.

[23] Q. Li, Y. Yang, and A. Wang, "Recognition of inscriptions on bones or tortoise shells based on graph isomorphism," Jisuanji Gongcheng yu Yingyong(Computer Engineering and Applications), vol. 47, pp. 112–114, 2011.

[24] S. Gu, "Identification of oracle-bone script fonts based on topological registration," Computer & Digital Engineering, vol. 44, pp. 2001–2006, 2016.

[25] Y. Liu and G. Liu, "Oracle bone inscription recognition based on svm," Journal of Anyang Normal University, vol. 2, pp. 54–56, 2017.

[26] X. Lu, H. Cai, and L. Lin, "Recognition of oracle radical based on the capsule network," CAAI transactions on intelligent systems, vol. 15, pp. 243–254, 2020.

[27] S. Sabour, N. Frosst, and G. E. Hinton, "Dynamic routing between capsules," Advances in Neural Information Processing Systems, vol. 30, pp. 3859–3869, 2017.

[28] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 4700–4708.

[29] Y.-K. Zhang, H. Zhang, Y.-G. Liu, and C.-L. Liu, "Oracle character recognition based on cross-modal deep metric learning," Acta Automatica Sinica, vol. 47, pp. 791–800, 2021.

[30] B. Sun and K. Saenko, "Deep coral: Correlation alignment for deep domain adaptation," in Proceedings of European conference on computer vision, 2016, pp. 443–450.

[31] W. Zellinger, T. Grubinger, E. Lughofer, T. Natschläger, and S. Saminger-Platz, "Central moment discrepancy (cmd) for domain-invariant representation learning," arXiv preprint arXiv:1702.08811, pp. 1–13, 2017.

[32] X. Peng, Q. Bai, X. Xia, Z. Huang, K. Saenko, and B. Wang, "Moment matching for multi-source domain adaptation," in Proceedings of the IEEE International Conference on Computer Vision, 2019, pp. 1406–1415.

[33] M. Long, H. Zhu, J. Wang, and M. I. Jordan, "Deep transfer learning with joint adaptation networks," in Proceedings of International conference on machine learning, 2017, pp. 2208–2217.

[34] H. Yan, Y. Ding, P. Li, Q. Wang, Y. Xu, and W. Zuo, "Mind the class weight bias: Weighted maximum mean discrepancy for unsupervised domain adaptation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2272–2281.

[35] Y. Chen, S. Song, S. Li, and C. Wu, "A graph embedding framework for maximum mean discrepancy-based domain adaptation algorithms," IEEE Transactions on Image Processing, vol. 29, pp. 199–213, 2019.

[36] E. Tzeng, J. Hoffman, K. Saenko, and T. Darrell, "Adversarial discriminative domain adaptation," in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 7167–7176.

[37] Z. Pei, Z. Cao, M. Long, and J. Wang, "Multi-adversarial domain adaptation," in Proceedings of the AAAI Conference on Artificial Intelligence, 2018, pp. 3934–3941.

[38] A. Chadha and Y. Andreopoulos, "Improved techniques for adversarial discriminative domain adaptation," IEEE Transactions on Image Processing, vol. 29, pp. 2622–2637, 2019.

[39] M. Long, Z. Cao, J. Wang, and M. I. Jordan, "Conditional adversarial domain adaptation," in Proceedings of International Conference on Neural Information Processing Systems, 2018, pp. 1647–1657.

[40] J. Hoffman, E. Tzeng, T. Park, J.-Y. Zhu, P. Isola, K. Saenko, A. Efros, and T. Darrell, "Cycada: Cycle-consistent adversarial domain adaptation," in Proceedings of International conference on machine learning, 2018, pp. 1989–1998.

[41] L. Hu, M. Kan, S. Shan, and X. Chen, "Duplex generative adversarial network for unsupervised domain adaptation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1498–1507.

[42] S. Sankaranarayanan, Y. Balaji, C. D. Castillo, and R. Chellappa, "Generate to adapt: Aligning domains using generative adversarial networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 8503–8512.

[43] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros, "Unpaired image-to-image translation using cycle-consistent adversarial networks," in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2223–2232.

[44] W. Zhang, W. Ouyang, W. Li, and D. Xu, “Collaborative and adversarial network for unsupervised domain adaptation,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2018, pp. 3801–3809.

[45] L. Wu, C. Zhang, J. Liu, J. Han, J. Liu, E. Ding, and X. Bai, “Editing text in the wild,” in Proceedings of the ACM international conference on multimedia, 2019, pp. 1500–1508.

[46] Q. Yang, J. Huang, and W. Lin, “Swaptext: Image based texts transfer in scenes,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 14 700–14 709.

[47] W. Shimoda, D. Haraguchi, S. Uchida, and K. Yamaguchi, “De-rendering stylized texts,” in Proceedings of the IEEE International Conference on Computer Vision, 2021, pp. 1076–1085.

[48] X. Mao, Q. Li, H. Xie, R. Y. Lau, Z. Wang, and S. Paul Smolley, “Least squares generative adversarial networks,” in Proceedings of the IEEE international conference on computer vision, 2017, pp. 2794–2802.

[49] J. Johnson, A. Alahi, and L. Fei-Fei, “Perceptual losses for real-time style transfer and super-resolution,” in Proceedings of European conference on computer vision, 2016, pp. 694–711.

[50] L. Gatys, A. Ecker, and M. Bethge, “A neural algorithm of artistic style,” Journal of Vision, vol. 16, pp. 326–326, 2016.

[51] X. Huang and S. Belongie, “Arbitrary style transfer in real-time with adaptive instance normalization,” in Proceedings of the IEEE International Conference on Computer Vision, 2017, pp. 1501–1510.

[52] G. Shi, C. Feng, L. Huang, B. Zhang, H. Ji, L. Liao, and H.-Y. Huang, “Genre separation network with adversarial training for cross-genre relation extraction,” in Proceedings of the Conference on Empirical Methods in Natural Language Processing, 2018, pp. 1018–1023.

[53] Y.-B. Kim, K. Stratos, and D. Kim, “Adversarial adaptation of synthetic or stale data,” in Proceedings of the Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), 2017, pp. 1297–1307.

[54] P. Cheng, M. R. Min, D. Shen, C. Malon, Y. Zhang, Y. Li, and L. Carin, “Improving disentangled text representation learning with information-theoretic guidance,” in Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2020, pp. 7530–7541.

[55] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, “Disentangled representation learning for non-parallel text style transfer,” in Proceedings of the Annual Meeting of the Association for Computational Linguistics, 2019, pp. 424–434.

[56] T. Shen, T. Lei, R. Barzilay, and T. Jaakkola, “Style transfer from non-parallel text by cross-alignment,” in Proceedings of International Conference on Neural Information Processing Systems, 2017, pp. 6833–6844.

[57] Y. Zhang, S. Nie, S. Liang, and W. Liu, “Robust text image recognition via adversarial sequence-to-sequence domain adaptation,” IEEE Transactions on Image Processing, vol. 30, pp. 3922–3933, 2021.

[58] L. Kang, M. Rusinol, A. Fornés, P. Riba, and M. Villegas, “Unsupervised writer adaptation for synthetic-to-real handwritten word recognition,” in Proceedings of the IEEE Winter Conference on Applications of Computer Vision, 2020, pp. 3502–3511.

[59] C. Luo, Q. Lin, Y. Liu, L. Jin, and C. Shen, “Separating content from style using adversarial learning for recognizing text in the wild,” International Journal of Computer Vision, vol. 129, no. 4, pp. 960–976, 2021.

[60] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein et al., “Imagenet large scale visual recognition challenge,” International journal of computer vision, vol. 115, pp. 211–252, 2015.

[61] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” in Proceedings of International Conference on Learning Representations, 2015, pp. 1–16.

[62] V. Dumoulin and F. Visin, “A guide to convolution arithmetic for deep learning,” arXiv preprint arXiv:1603.07285, pp. 1–31, 2016.

[63] D. Ulyanov, A. Vedaldi, and V. Lempitsky, “Instance normalization: The missing ingredient for fast stylization,” arXiv preprint arXiv:1607.08022, pp. 1–6, 2016.

[64] L. v. d. Maaten and G. Hinton, “Visualizing data using t-sne,” Journal of machine learning research, vol. 9, pp. 2579–2605, 2008.

[65] X. Chen, S. Wang, M. Long, and J. Wang, “Transferability vs. discriminability: Batch spectral penalization for adversarial domain adaptation,” in Proceedings of International conference on machine learning, 2019, pp. 1081–1090.

[66] S. Cui, S. Wang, J. Zhuo, C. Su, Q. Huang, and Q. Tian, “Gradually vanishing bridge for adversarial domain adaptation,” in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2020, pp. 12 455–12 464.

[67] E. Tzeng, J. Hoffman, N. Zhang, K. Saenko, and T. Darrell, “Deep domain confusion: Maximizing for domain invariance,” arXiv preprint arXiv:1412.3474, pp. 1–9, 2014.

[68] X. Jia, Y. Jin, X. Su, and Y. Hu, “Domain-invariant representation learning using an unsupervised domain adversarial adaptation deep neural network,” Neurocomputing, vol. 355, pp. 209–220, 2019.

[69] Z. Luo, Y. Zou, J. Hoffman, and L. Fei-Fei, “Label efficient learning of transferable representations across domains and tasks,” in Proceedings of International Conference on Neural Information Processing Systems, 2017, pp. 164–176.

[70] Y. Taigman, A. Polyak, and L. Wolf, “Unsupervised cross-domain image generation,” in Proceedings of International Conference on Learning Representations, 2017, pp. 1–14.

[71] K. Saito, Y. Ushiku, and T. Harada, “Asymmetric tri-training for unsupervised domain adaptation,” in Proceedings of International Conference on Machine Learning, 2017, pp. 2988–2997.

[72] M.-Y. Liu and O. Tuzel, “Coupled generative adversarial networks,” Advances in neural information processing systems, vol. 29, pp. 469–477, 2016.

[73] S. Xie, Z. Zheng, L. Chen, and C. Chen, “Learning semantic representations for unsupervised domain adaptation,” in Proceedings of International conference on machine learning, 2018, pp. 5423–5432.

[74] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan, “Unsupervised pixel-level domain adaptation with generative adversarial networks,” in Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 3722–3731.

[75] M.-Y. Liu, T. Breuel, and J. Kautz, “Unsupervised image-to-image translation networks,” Advances in neural information processing systems, vol. 30, pp. 700–708, 2017.