Adaptively Transfer Category-Classifier for Handwritten Chinese Character Recognition

Yongchun Zhu[1,2], Fuzhen Zhuang[1,2(✉)], Jingyuan Yang[3], Xi Yang[4], and Qing He[1,2]

[1] Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS), Institute of Computing Technology, CAS, Beijing 100190, China
{zhuyongchun18s,zhuangfuzhen,heqing}@ict.ac.cn
[2] University of Chinese Academy of Sciences, Beijing 100049, China
[3] George Mason University, Fairfax, USA
jyang53@gmu.edu
[4] Sunny Education Inc., Beijing 100102, China
xi.yang@17zuoye.com

Abstract. Handwritten character recognition (HCR) plays an important role in real-world applications, such as bank check recognition, automatic sorting of postal mail, the digitization of old documents, intelligence education and so on. Last decades have witnessed the vast amount of interest and research on handwritten character recognition, especially in the competition of HCR tasks on the specific data sets. However, the HCR task in real-world applications is much more complex than the one in HCR competition, since everyone has their own handwriting style, e.g., the HCR task on middle school students is much harder than the one on adults. Therefore, state-of-the-art methods proposed by the competitors may fail. Moreover, there is not enough labeled data to train a good model, since manually labelling data is usually tedious and expensive. So one question arises, is it possible to transfer the knowledge from related domain data to train a good recognition model for the target domain, e.g., from the handwritten character data of adults to the one of students? To this end, we propose a new neural network structure for handwritten Chinese character recognition (HCCR), in which we try to make full use of a large amount of labeled source domain data and a small number of target domain data to learn the model parameters. Furthermore, we make a transfer on the category-classifier level, and adaptively assign different weights to category-classifiers according to the usefulness of source domain data. Finally, experiments constructed from three data sets demonstrate the effectiveness of our model compared with several state-of-the-art baselines.

1 Introduction

The handwritten character recognition problem has attracted much interest and research for a long time, and plays an important role in various kinds of

Q. Yang et al. (Eds.): PAKDD 2019, LNAI 11439, pp. 110–122, 2019.
https://doi.org/10.1007/978-3-030-16148-4_9

applications [10,20,22], such as bank cheque recognition, automatic sorting of postal mail, the digitization of old documents, intelligence education and so on. The previous handwritten character recognition works can be grouped into different types, including the recognition tasks concerning about digits [10], English characters [11], Chinese characters [20,22], French characters [7] etc. In this paper, we focus on the handwritten Chinese character recognition (HCCR) problem, and consider more challenging recognition scenarios which are much closer to approaching the real-world applications.

The HCCR problem has been extensively studied for more than 40 years [12], and can be further divided into two types: online and off-line recognition. The online recognizer identifies characters during the writing process using the digitised trace of the pen, while off-line recognition deals with images scanned of previously handwritten characters. Usually, the online recognition task is easier than the off-line one since there is much digitised trace information available for training the models. However, off-line recognition has broader applications, e.g., automatic sorting of postal mail and the editing of old documents. In the recent decade, there are many research works and competitions devoted to the off-line recognition of Chinese characters, especially based on the deep neural network framework [1,18]. Convolution neural network (CNN), which was originally developed by LeCun et al. [10], provided a new end-to-end approach to handwritten Chinese character recognition with very promising results in recent years [18,22], e.g., the extended deeper architectures of AlexNet, VGG, GoogLeNet, ResNet with dropout and nonlinear activation function ReLU. Ciresan et al. [2] proposed the multi-column deep neural network (MCDNN), which may be the first successful model based on deep neural network (DNN) used in the application of large-scale HCCR tasks. The winner of online and off-line handwritten Chinese character recognition competition in ICDAR2013 was based on MCDNN [20]. Zhong et al. [22] proposed HCCR-GoogLeNet and employed three types of directional feature maps, namely the Gabor, gradient and HoG feature maps, to enhance the performance of GoogLeNet, leading to the high accuracy of 96.74% on the ICDAR2013 off-line data set. Recently, Yang et al. [19] proposed a new training method DropSample to enhance deep convolutional neural networks for large-scale HCCR problems.

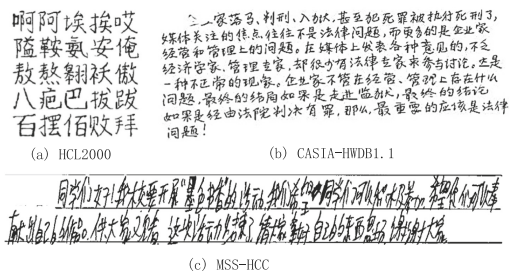

Fig. 1. The examples of three data sets, HCL2000, CASIA-HWDB1.1 and MSS-HCC.

Though there are many advanced algorithms proposed for HCCR, HCCR is still a challenging task. First, comparing English characters with only 26 basic categories, in Chinese the national standard GB2312-80 coding defined 6763 categories of commonly used Chinese characters. GB18010-2000 further expanded the Chinese character set to 27,533 categories. Even for the recently used Chinese character sets HCL2000 [21] and CASIA-HWDB1.1 [13], which both contain 3,755 categories of characters. Second, handwritten Chinese characters are very casual and diverse compared with regular printed characters. Third, there are many similar and confusing Chinese characters, which human can recognize easily but very hard for computer. For example, "己" v.s. "已", "曰" v.s. "日", "淡" v.s. "谈" and "何" v.s. "向". Moreover, there are also other technical challenges in addition to the complexity of Chinese characters. On one hand, recent HCCR competitions are targeted at improving the performance on a specific standard data set, e.g., CASIA-HWDB1.1 is adopted in ICDAR2013, but the recognition scenarios are much more complex and difficult in real-world applications. Figure 2 shows some examples from three data sets, in which HCL2000 and CASIA-HWDB1.1 are both well known off-line handwritten Chinese character data sets, while MSS-HCC is the one we collect from the middle school students writing answers and compositions in a specific exam[1]. From this figure, we can find that the recognition of MSS-HCC is much harder than HCL2000 and CASIA-HWDB1.1. On the other hand, there is usually not enough labeled data to train a satisfying model, since manually labelling is tedious, time-consuming and expensive. Also, it might fail when simply applying the model trained from one data set to another data set, e.g., the proposed model [22] trained on HCL2000 only obtains about 63% accuracy on CASIA-HWDB1.1.

Transfer learning aims to adapt the knowledge from related source domain data to the model learning in the target domain, which provides the possibility of success for HCCR tasks. Along this line, we propose a transfer handwritten Chinese character recognition model based on the successful deep network structure AlexNet [9]. Specifically, for both source and target domain data, we share the network parameters with five convolution layers and three pooling layers, and then learn the parameters of three fully connected layers separately. In addition, to adaptively transfer the category knowledge from the source domain to the target domain, we impose a regularization item with different weights to learn the similarity of category-classifiers trained from the source to the target domain. Finally, we conduct extensive experiments on three data sets to validate the effectiveness of our model.

The remainder of this paper is organized as follows. Section 2 briefly introduces the related work of handwritten character recognition and transfer learning, followed by the problem formalization and the details of the proposed model in Sect. 3. Section 4 presents the experimental results to demonstrate the effectiveness of our model, and finally Sect. 5 concludes this paper.

[1] This work does not consider how to segment the characters, but only focuses on the recognition of segmented isolate characters.

2 Related Work

In this section, we will briefly introduce the most related work about handwritten Chinese character recognition and transfer learning.

2.1 Handwritten Chinese Character Recognition

Ciresan et al. [4] first used CNNs to realize the classification of 1,000 types of handwritten digit characters, then based on which IDSIA Lab won the first place on off-line HCCR data set with the accuracy of 92.12% and the fourth place on online HCCR data set with 93.01% in ICDAR2011 competition [14]. The champions of off-line and online HCCR competition in ICDAR2013 [20] are based on MCDNN. Wu et al. [18] proposed to improve the performance of off-line HCCR task up to 96.06% by adopting the ensemble of four alternately trained relaxation convolutional neural networks (ATR-CNN). After that in 2015, Zhong et al. [22] proposed the HCCR-GoogLeNet with Gabor, gradient and HoG feature maps to obtain an accuracy of 96.74% accuracy on ICDAR2013 off-line data set, which is the first time computers outperformed human recognition accuracy 96.13%. Furthermore, Chen et al. [1] uses a deeper CNN network to achieve 96.79% accuracy, and Yang et al. [19] proposed a new DropSample to train ensemble CNNs model to get 97.06% accuracy on ICDAR2013 data set. However, these above methods are all designed for a specific standard data set, which might fail in more complex HCCR scenarios. Also, they don't consider to make use of the large amount of auxiliary data, i.e., HCL2000. Therefore, in this work we try to apply the knowledge from related source domain data for further improving the recognition performance in target domain.

2.2 Transfer Learning

Transfer learning targets at learning the knowledge from large amount of related source/auxiliary data to help improve the prediction performance of target domain data [16]. In recent decade, transfer learning has provoked vast amount of attention and research for variable kinds of applications, e.g., text classification [5], image classification [23], visual categorization [17] etc. To the best of our knowledge, there is little work of transfer learning for handwritten Chinese character recognition problems. The work [3] actually learnt Chinese characters by first pre-training a DNN on other data sets. Thus, we will employ transfer learning manner based on deep network to deal with HCCR. Furthermore, most previous work make transfer on the instance level [8], the model level [6], and the feature learning level [15], but we focus on transferring the knowledge on the category-classifier level.

3 Adaptively Transfer Category-Classifier for Chinese Handwriting Recognition

3.1 Problem Formulation

For clarity, the frequently used notations are listed in Table 1. Supposing we have data in both the source and the target domain $\mathcal{D}_s = \{x_i^{(s)}, y_i^{(s)}\}|_{i=1}^{n_s}$ and $\mathcal{D}_t = \mathcal{D}_t^L \cup \mathcal{D}_t^U = \{x_i^{(t)}, y_i^{(t)}\}|_{i=1}^{n_t^L} \cup \{x_i^{(t)}\}|_{i=1}^{n_t^U}$, respectively, where $x_i^{(s)}, x_i^{(t)} \in \mathbb{R}^{m \times m}$ are the data instances with image size $m \times m$, $y_i^{(s)}, y_i^{(t)} \in \{1, \cdots, c\}$ are their corresponding labels, c is the number of categories, n_s and $n_t = n_t^L + n_t^U$ are respectively the numbers of instances in source and target domains. Usually there are large number of labeled instances in source domain and only a small portion of labeled instances in target domain, i.e., $n_t^L \ll n_s$ and $n_t^L \ll n_t^U$, thus this is an challenging recognition task. Since the data distributions of source domain and target domain are different, our goal adopts transfer learning to make full use of source domain data $\mathcal{D}_s$ and a small portion of labeled target domain data $\mathcal{D}_t^L$ to train a recognition model and obtain satisfying performance on the unlabeled target domain data $\mathcal{D}_t^U$.

Table 1. The notation and denotation

$\mathcal{D}_s$, $\mathcal{D}_t$	The source and target domains
n_s	The number of instances in source domain
n_t	The number of instances in target domain
l	The index of layer
$n_t^L(n_t^U)$	The number of labeled (unlabeled) instances in target domain
m	The width and the height of original map
K_{ij}^l	The j-th kernel filter in the l-th layer connected to the i-th map in the $(l-1)$-th layer
k^l	The number of nodes in l-th full connected layer
κ^l	The kernel size in l-th convolutional layer
c	The number of categories
$\boldsymbol{x}_i^{(s)}$, $\boldsymbol{x}_i^{(t)}$	The i-th instance of source and target domains
$y_i^{(s)}$, $y_i^{(t)}$	The label of instances $\boldsymbol{x}_i^{(s)}$ and $\boldsymbol{x}_i^{(t)}$
$\boldsymbol{a}^l$	The output of the l-th full connected layer
$\boldsymbol{W}^l$, $\boldsymbol{b}^l$	Weight matrix and bias for the l-th full connected
$\boldsymbol{\theta}_j$	$\boldsymbol{\theta}_j (j \in \{1, \cdots, c\})$ is the j-th column of W^8
$\top$	The transposition of a matrix
$\xi^{(s)}, \xi^{(t)}$	The input of the softmax layer

3.2 Adaptively Transferring Category-Classifier Model

Motivated by the success of deep network structure AlexNet for image classification [9] and handwritten character recognition [22], we propose a new Adaptively Transferring Category-classifier model for HCCR (ATC-HCCR for short) based on AlexNet. The architecture of the network of ATC-HCCR is shown in Fig. 2. Particularly, this network has a total of eight layers including the first five successive convolutional layers $conv1, \cdots, conv5$ ($conv1, conv2, conv5$ are followed by pooling layers.) and three fully connected layers $fc6, fc7$, and $fc8$. In Fig. 2, two important transfer learning components are utilized to improve the performance. The first knowledge transfer is achieved when both the source and the target domain share the same parameters of the five convolutional layers and three pooling layers for the shared convolutional kernels and pooling operations. Then, the network is diverged into two branches to learn the parameters of three fully connected layers separately, where one is for the source domain and the other one is the target domain. Furthermore, we impose a regularization item with different weights as the second transfer knowledge component to adaptively transfer the category knowledge from the source domain to the target domain by learning the similarity of category-classifiers trained from the source domain to the target domain.

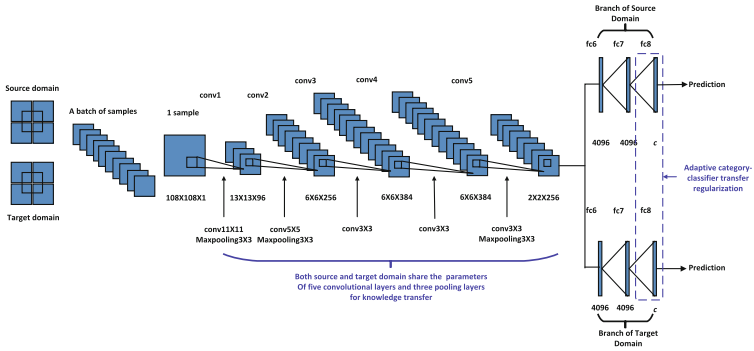

Fig. 2. The network structure of ATC-HCCR.

Given $x_i^l \in \mathbb{R}^{m^l \times m^l}$ represents the i-th map in the l-th layer and the map size is $m^l \times m^l$, j-th kernel filter in the l-th layer connected to the i-th map in the $(l-1)$-th layer denoted as $K_{ij}^l \in \mathbb{R}^{\kappa^l \times \kappa^l}$ and index maps set $M_j = \{i | i\text{-th map in the } (l-1)\text{-th layer connected to } j\text{-th map in the layer}\}$. So the convolutional operation is defined by the following equation,

$$x_j^l = f\Big(\sum_{i \in M_j} x_i^{l-1} * K_{ij}^l + b_j^l\Big), \tag{1}$$

where $*$ denotes convolutional operation, $f(z) = \max(0, z)$ is ReLU non-linearity activation function and b_j^l is bias.

Next, pooling layer then combines the output of the neuron cluster at one layer to single neuron in the next layer. Pooling operations are carried out to reduce the number of data points and to avoid overfitting. And pooling equation can be described as

$$x_j^l = down(x_j^l), \tag{2}$$

where $down(\cdot)$ is max pooling to computer the max value of each $p \times p$ region in x_j^lmap. After the last pooling layers, pixels of pooling layers are stretched to single column vector. These vectorized and concatenated data points are fed into fully connected layers for the classification.

The output of the l-th fully connected layer of the branch of source domain is listed as follows, and the one of target domain is similar.

$$a_s^l = f(W_s^l a_s^{l-1} + b_s^l) \tag{3}$$

The $fc6$ and $fc7$ in Fig. 2 both have an output $a_s^l \in \mathbb{R}^{k^l \times 1}$($l$-th full connected layer) of k^l nodes, a weight matrix $W_s^l \in \mathbb{R}^{k^l \times k^{(l-1)}}$, and a bias vector $b_s^l \in \mathbb{R}^{k^l \times 1}$(the l-th full connected layer). And the output of the fc7 is denoted as $\xi^{(s)} \in \mathbb{R}^{k^7 \times 1}$.

$$y^{(s)} = g(W_s^8 \xi^{(s)} + b_s^8), \tag{4}$$

$g(\cdot)$ is a softmax function, and $y^{(s)} \in \mathbb{R}^{c \times 1}, W_{(s)}^8 \in \mathbb{R}^{c \times k^7}$. Let $W_s^8 = [\theta_1^{(s)}, \theta_2^{(s)}, \cdots, \theta_c^{(s)}]^\top$, $W_t^8 = [\theta_1^{(t)}, \theta_2^{(t)}, \cdots, \theta_c^{(t)}]^\top$, $\gamma = [\gamma_1, \gamma_1, \cdots, \gamma_c]$, the objective function to be minimized in our proposed learning framework is formalized as follows:

$$\mathcal{L} = J_r(\mathcal{D}_s \cup \mathcal{D}_t^L, y) + \Omega(W_s^8, W_t^8), \tag{5}$$

where the first term is the Cross Entropy for both the source and target domain, which can be further detailed defined as

$$J_r(\mathcal{D}_s \cup \mathcal{D}_t^L, y) = -\frac{1}{n_s + n_t^L} \sum_{i=1}^{n_s+n_t^L} \sum_{j=1}^{c} 1\{y_i = j\} \log \frac{e^{\theta_j^\top \xi_i}}{\sum_{u=1}^{c} e^{\theta_u^\top \xi_i}}, \tag{6}$$

where ξ_i is the i-th instance, and $\theta_j^\top \in \mathbb{R}^{k^7 \times 1}(j \in \{1, ..., c\})$ is the j-th row of W_s^8 or W_t^8. The second term of the objective is the regularization term, which is defined as

$$\Omega(W_s^8, W_t^8) = \lambda \sum_{i=1}^{c} \gamma_i \cdot |\theta_i^{(s)} - \theta_i^{(t)}|. \tag{7}$$

In the regularization term, there are two parameters, namely, the trade-off parameter λ and the weights γ. λ controls the significance of the regularization term, while γ is for transferring category-classifiers between the source and target domain. The term $|\theta_i^{(s)} - \theta_i^{(t)}|$ represents the distance of i-th category-classifier between the two domains.

We use tensorflow to implement our network and AdamOptimizer as the optimizer, and the detailed algorithm is shown in Algorithm 1. Note that, there is

Algorithm 1. Transfer Learning with Adaptively Transfer Category-classifier

Input: Given one source domain $\mathcal{D}_s = \{x_i^{(s)}, y_i^{(s)}\}|_{i=1}^{n_s}$, and one target domain $\mathcal{D}_t = \mathcal{D}_t^L \cup \mathcal{D}_t^U = \{x_i^{(t)}, y_i^{(t)}\}|_{i=1}^{n_t^L} \cup \{x_i^{(t)}\}|_{i=1}^{n_t^U}$, trade-off parameters λ and weights γ, the number of nodes in full connected layer and label layer, k and c.
Output: Results of x_i belongs to the vector of probability for each category.

1. Use both $\mathcal{D}_s$ and $\mathcal{D}_t^L$ to train AlexNet.
2. Use the parameters in Step1's model to initialize ATC-HCCR shown in Figure 2.
3. Choose a batch of instances from $\mathcal{D}_s$ or $\mathcal{D}_t^L$ as input.
4. Use AdamOptimizer with loss function Eq. (5) to update all variables.
5. Continue Step3 and Step4 until the algorithm converges.
6. Input $\mathcal{D}_t^U$ and get the vector of probability for each category that x_i belongs to.

only a small amount of labeled data in the target area, oversampling is required. Besides, training a randomly initialized model can waste a lot of time, so pre-training method is used. Specifically, we first use the source domain and the small amount of labeled target domain data to train a AlexNet, and then we use the parameters of this model to initialize our model. In our experiment, oversampling and pre-training are both used in ATC-HCCR.

After all the parameters are learned, we can use the classifiers to predict the target domain. That is, for any instance $x^{(t)}$ in target domain, the output of the $y^{(t)}$ can indicate $x^{(t)}$ belonging to the vector of probability for each category. We choose the maximum probability and the corresponding label as the prediction.

4 Experimental Evaluation

In this section, we conduct extensive experiments on three real-world handwritten Chinese character data sets to validate the effectiveness of the proposed framework.

4.1 Data Preparation

Two of the three data sets are standard ones, i.e., HCL2000 and CASIA-HWDB1.0[2], and the rest one MSS-HCC is collected by ourselves. The statistics of three data sets are listed in Table 2.

Table 2. The statistics of three data sets.

	HCL2000	CASIA-HWDB1.1	MSS-HCC
#category	3,755	3,755	27
#instance	3,755,000	1,126,500	5,920

[2] We thank the authors for providing these two data sets.

HCL2000 [21] contains 3,755 categories of frequently used simplified Chinese characters written by 1,000 different persons. All images for each character in this data set is 64×64. As shown in Fig. 1(a), the characters in the data set are neat and orderly.

CASIA-HWDB1.1 [13] is produced by 300 persons, which includes 171 categories of alphanumeric characters and symbols, and 3,755 categories of Chinese characters. The Chinese characters are used as the experimental data. As shown in Fig. 1(b), the characters are written less neat and orderly.

MSS-HCC is labeled by ourselves. We collect this data set from the middle school students writing answers and compositions in a specific exam, and after segmentation the images with size 108×108 for each Chinese character are obtained. Then we labeled about 20,000 images, and those categories of Chinese characters whose number of instances larger than 100 are selected as the experimental data. As shown in Fig. 1(c), this data set is written much in messy.

For these three data sets, we conduct some preliminary tests applying the HCCR method in [22]. We use HCL2000 as training data and CASIA-HWDB1.1 for test, then the accuracy is 63%. In contrast, using CASIA-HWDB1.1 as training data and HCL2000 for test, the accuracy is achieved at 94%. This shows that HCL2000 is more neat than CASIA-HWDB1.1. In addition, respectively using HCL2000 and CASIA-HWDB1.1 as training sets, MSS-HCC for test has accuracies of 53% and 71%. In contrast, using MSS-HCC as the training data, the accuracies on HCL2000 and CASIA-HWDB1.1 are 86% and 87%. These results reveal that MSS-HCC is more messy than HCL2000 and CASIA-HWDB1.1. In the experiments, we focus on the knowledge transfer from source domain to improve the recognition performance of much more difficult tasks. Therefore, three transfer HCCR problems are finally constructed, i.e., HCL2000 → CASIA-HWDB1.1, HCL2000 → MSS-HCC and CASIA-HWDB1.1 → MSS-HCC.

4.2 Baselines and Implementation Details

Baselines: We mainly compare our model with following two state-of-the-art baselines,

- AlexNet-HCCR [22], which uses AlexNet for HCCR task, contains five convolutional layers, three pooling layers, and three full connected layers. There is not transfer mechanism for this method.
- preDNN [3], is actually an accelerated deep neural network (DNN) model by first pretraining a DNN on a small subset of all classes and then continuing to train on all classes. As claimed in their original paper, preDNN is a transfer learning approach for handling HCCR problems.

For AlexNet-HCCR, we record three values of accuracy for each transfer learning problem. Specifically, training the models on labeled source domain data $\mathcal{D}_s$, labeled target domain data $\mathcal{D}_t^L$, labeled source and target domain data $\mathcal{D}_s \cup \mathcal{D}_t^L$, respectively, and then testing unlabeled target domain data $\mathcal{D}_t^U$, denoted as AlexNet-HCCR(s), AlexNet-HCCR(t) and AlexNet-HCCR(s+t), respectively.

For preDNN, we first pretrain the model on $\mathcal{D}_s$, then continue to train on $\mathcal{D}_t^L$, and finally make prediction on $\mathcal{D}_t^U$.

Implementation Details: There are two parameters, i.e., trade-off parameter λ and weights $\gamma_i (1 \leq i \leq c)$ for transferring category-classifiers between source and target domains. We set $\lambda = 5$ for all experiments, and for $\gamma_i (1 \leq i \leq c)$ we simply set them according to the accuracies on $\mathcal{D}_t^L$ given by the AlexNet-HCCR model trained from $\mathcal{D}_s$, i.e., the higher value of accuracy for the i-th category, the larger value is set to $\gamma_i (1 \leq i \leq c)$, and vice versa. Certainly, it would be better to study the optimum setting for $\gamma_i (1 \leq i \leq c)$, which will be our future work. The number of iterations for optimization is 50,000, and the average values of accuracy are recorded for 3 trials. Finally, a small portion of target domain data are randomly sampled as labeled ones. Specifically, we set the sampling ratio from [1.67%, 10%] with interval 1.67% for CASIA-HWDB1.1 as target domain, and from [5%, 30%] with interval 5% for MSS-HCC as target domain. The prediction accuracy is adopted as the evaluation metric.

4.3 Experimental Results

We evaluate all the approaches under different sampling ratios of labeled target domain data, and all the results are shown in Table 3. From these results, we have the following insightful observations,

- Except AlexNet-HCCR(s) only using labeled source domain data, the performance of all the other algorithms improves with the increasing values of sampling ratio of target domain data as labeled data. Generally, the performance increases significantly with the increasing of sampling ratio, and then slowly, which coincides with our expectation. Because if there are enough labeled data for training a good model, incorporating more labeled data will not take much effect.
- Our model ATC-HCCR achieves the best results over all baselines, under different sampling ratios of target domain data, which demonstrates the effectiveness of the proposed transfer learning framework for HCCR tasks. Also, we observe that ATC-HCCR beats baselines with a large margin of improvement on the problem of HCL2000 → CASIA-HWDB1.1, and much smaller margin on the problems of HCL2000 → MSS-HCC and CASIA-HWDB1.1 → MSS-HCC. This is due to the fact that the recognition of MSS-HCC data set is more challenging. On the other hand, for a challenging problem, even a small value of 0.5% improvement is remarkable.
- Both transfer learning models ATC-HCCR and preDNN outperform AlexNet-HCCR, which indicates the importance and necessity of applying transfer learning for tackling HCCR problems. ATC-HCCR is better than preDNN, since preDNN, as a simple transfer learning algorithm, only tries to adopt all network parameters from the source domain for initialization but not considers the transfer of category-classifiers.

Table 3. The performance (%) comparison on three data sets among AlexNet-HCCR, preDNN and ATC-HCCR.

	HCL2000 → CASIA-HWDB1.1						Mean
	1.67%	3.33%	5%	6.67%	8.33%	10%	
AlexNet-HCCR(s)	63.30	63.31	63.40	63.48	63.53	63.41	63.41
AlexNet-HCCR(t)	30.83	61.64	79.52	78.04	81.01	81.85	68.82
AlexNet-HCCR(s+t)	73.07	76.78	79.52	81.13	82.24	82.08	79.14
preDNN	73.01	76.89	79.37	81.47	82.47	83.56	79.46
ATC-HCCR	76.79	79.78	82.37	84.13	85.08	85.06	**82.20**
	HCL2000 → MSS-HCC						Mean
	5%	10%	15%	20%	25%	30%	
AlexNet-HCCR(s)	61.49	63.18	62.61	62.75	63.92	64.30	63.04
AlexNet-HCCR(t)	66.44	82.83	89.77	90.96	92.57	93.00	85.93
AlexNet-HCCR(s+t)	86.31	88.95	91.02	91.55	92.22	94.76	90.80
preDNN	86.93	90.69	92.61	93.45	93.90	94.61	92.03
ATC-HCCR	87.76	91.12	93.24	93.71	94.57	94.88	**92.55**
	CASIA-HWDB1.1 → MSS-HCC						Mean
	5%	10%	15%	20%	25%	30%	
AlexNet-HCCR(s)	76.01	78.38	78.27	78.12	78.38	77.87	77.84
AlexNet-HCCR(t)	66.44	82.83	89.77	90.96	92.57	93.00	85.93
AlexNet-HCCR(s+t)	89.48	91.38	92.27	93.67	94.21	94.98	92.67
preDNN	89.19	92.64	92.74	93.58	94.61	94.98	92.96
ATC-HCCR	90.98	93.14	93.80	94.55	94.68	95.29	**93.74**

Table 4. The Influence of trade-off parameter λ on the performance (%) of ATC-HCCR.

λ	HCL2000 → CASIA-HWDB1.1					
	1.67%	3.33%	5%	6.67%	8.33%	10%
0	75.77	79.58	82.03	83.83	84.65	84.90
0.05	76.30	80.14	82.56	83.83	85.05	85.10
0.5	76.51	79.76	82.41	84.08	84.84	85.29
5	76.79	79.78	82.37	84.13	85.08	85.06

4.4 The Influence of Trade-Off Parameter λ

We investigate the influence of trade-off parameter λ on the performance of ATC-HCCR over the problem HCL2000 → CASIA-HWDB1.1, and λ is sampled from $\{0, 0.05, 0.5, 5\}$. The results are shown in Table 4. $\lambda = 0$ indicates that ATC-HCCR only considers the parameters sharing of five convolutional layers

and three pooling layers during the optimization, and even so ATC-HCCR can outperform preDNN. When $\lambda > 0$, the transfer category-classifier regularization is integrated in our model, and the performance of ATC-HCCR can be further improved, which shows the effectiveness of transfer category-classifier regularization. In our experiments, we simply set the weights $\gamma_i (1 \leq i \leq c)$ according to the accuracies of AlexNet-HCCR making predictions on $\mathcal{D}_t^L$. If the size of $\mathcal{D}_t^L$ is small, the estimation of γ_i may not be reliable, therefore λ is not set to a large value, i.e., $\lambda = 5$ in the experiments.

5 Conclusion

In this paper, we study the challenging handwritten Chinese character recognition (HCCR) problem in real-world applications. As there is little work about transfer learning for HCCR, based on Alexnet, we propose a new network framework by adaptively transferring category-classifier for HCCR problems. In our framework, there are actually two components for knowledge transfer. First, the parameters of five convolutional layers and three pooling operations are shared across the source and the target domain during the optimization; second, observing that the category-classifiers from two domains have different similarities, therefore different weights are imposed to regularize the category-classifier transfer. Furthermore, we also collect a small set of much more challenging HCCR data, and finally conduct experiments on three data sets to demonstrate the effectiveness of our model. In future work, we will collect more data and consider how to find the optimum values of weights.

Acknowledgments. The research work is supported by the National Key Research and Development Program of China under Grant No. 2018YFB1004300, the National Natural Science Foundation of China under Grant Nos. U1836206, U1811461, 61773361, the Project of Youth Innovation Promotion Association CAS under Grant No. 2017146.

References

1. Chen, L., Wang, S., Fan, W., Sun, J., Naoi, S.: Beyond human recognition: a CNN-based framework for handwritten character recognition. In: ACPR, pp. 695–699 (2015)
2. Ciresan, D., Meier, U., Schmidhuber, J.: Multi-column deep neural networks for image classification. In: CVPR, pp. 3642–3649 (2012)
3. Cireşan, D.C., Meier, U., Schmidhuber, J.: Transfer learning for Latin and Chinese characters with deep neural networks. In: IJCNN, pp. 1–6 (2012)
4. Ciresan, D.C., Meier, U., Gambardella, L.M., Schmidhuber, J.: Convolutional neural network committees for handwritten character classification. In: ICDAR, pp. 1135–1139 (2011)
5. Dai, W., Xue, G.R., Yang, Q., Yu, Y.: Co-clustering based classification for out-of-domain documents. In: ACM SIGKDD, pp. 210–219 (2007)
6. Gao, J., Fan, W., Jiang, J., Han, J.: Knowledge transfer via multiple model local structure mapping. In: SIGKDD, pp. 283–291 (2008)

7. Grosicki, E., El-Abed, H.: ICDAR 2011-French handwriting recognition competition. In: ICDAR, pp. 1459–1463 (2011)
8. Jiang, J., Zhai, C.: Instance weighting for domain adaptation in NLP. In: ACL, pp. 264–271 (2007)
9. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: NIPS, pp. 1097–1105 (2012)
10. LeCun, Y., et al.: Handwritten digit recognition with a back-propagation network. In: NIPS, pp. 396–404 (1990)
11. Lin, Z., Wan, L.: Style-preserving English handwriting synthesis. Pattern Recognit. **40**, 2097–2109 (2007)
12. Liu, C.L., Jaeger, S., Nakagawa, M.: 'Online recognition of Chinese characters: the state-of-the-art. IEEE TPAMI **26**, 198–213 (2004)
13. Liu, C.L., Yin, F., Wang, D.H., Wang, Q.F.: Online and offline handwritten Chinese character recognition: benchmarking on new databases. Pattern Recognit. **46**, 155–162 (2013)
14. Liu, C.L., Yin, F., Wang, Q.F., Wang, D.H.: ICDAR 2011 Chinese handwriting recognition competition. In: ICDAR (2011)
15. Pan, S.J., Tsang, I.W., Kwok, J.T., Yang, Q.: Domain adaptation via transfer component analysis. IEEE TNN **22**, 199–210 (2011)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. IEEE TKDE **22**, 1345–1359 (2010)
17. Shao, L., Zhu, F., Li, X.: Transfer learning for visual categorization: a survey. IEEE TNNLS **26**, 1019–1034 (2015)
18. Wu, C., Fan, W., He, Y., Sun, J., Naoi, S.: Handwritten character recognition by alternately trained relaxation convolutional neural network. In: Proceedings of 14th ICFHR, pp. 291–296 (2014)
19. Yang, W., Jin, L., Tao, D., Xie, Z., Feng, Z.: DropSample: a new training method to enhance deep convolutional neural networks for large-scale unconstrained handwritten Chinese character recognition. Pattern Recognit. **58**, 190–203 (2016)
20. Yin, F., Wang, Q.F., Zhang, X.Y., Liu, C.L.: ICDAR 2013 Chinese handwriting recognition competition. In: ICDAR, pp. 1464–1470 (2013)
21. Zhang, H., Guo, J., Chen, G., Li, C.: HCL2000-a large-scale handwritten Chinese character database for handwritten character recognition. In: ICDAR, pp. 286–290 (2009)
22. Zhong, Z., Jin, L., Xie, Z.: High performance offline handwritten Chinese character recognition using googlenet and directional feature maps. In: Proceedings of 13th ICDAR, pp. 846–850 (2015)
23. Zhu, Y., et al.: Heterogeneous transfer learning for image classification. In: AAAI (2011)