Pattern Recognition Letters 106 (2018) 20–26

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Robust offline handwritten character recognition through exploring writer-independent features under the guidance of printed data

Yaping Zhang[a,b], Shan Liang[a], Shuai Nie[a,b], Wenju Liu[a,*], Shouye Peng[c]

[a] National Laboratory of Patten Recognition, Institute of Automation, Chinese Academy of Sciences, No.95, Zhongguancun East Road, Beijing 100190, PR China
[b] University of Chinese Academy of Sciences, China
[c] Xueersi Online School, China

ARTICLE INFO

Article history:
Received 12 August 2017
Available online 15 February 2018

MSC:
41A05
41A10
65D05
65D17

Keywords:
Handwritten character recognition
Writer-independent features
Adversarial feature learning
Convolutional neural network

ABSTRACT

Deep convolutional neural networks have made great progress in recent handwritten character recognition (HCR) by learning discriminative features from large amounts of labeled data. However, the large variance of handwriting styles across writers is still a big challenge to the robust HCR. To alleviate this issue, an intuitional idea is to extract writer-independent semantic features from handwritten characters, while standard printed characters are writer-independent stencils for handwritten characters. They could be used as prior knowledge to guide models to exploit writer-independent semantic features for HCR. In this paper, we propose a novel adversarial feature learning (AFL) model to incorporate the prior knowledge of printed data and writer-independent semantic features to improve the performance of HCR on limited training data. Different from available handcrafted features methods, the proposed AFL model exploits writer-independent semantic features automatically, and standard printed data as prior knowledge is learnt objectively. Systematic experiments on MNIST and CASIA–HWDB show that the proposed model is competitive with the state-of-the-art methods on the offline HCR task.

1. Introduction

[1] Handwritten character recognition (HCR) has been widely applied in mail sorting [29], historical documents recognition [24], handwritten notes transcription and bank check reading. In addition, it is also an important component of handwritten text recognition [25]. To this end, decades of efforts have been devoted to HCR, but robust HCR is still a challenging task due to the huge variance of handwriting styles. In particular, the recognition task is more difficult when the handwritten characters are offline, while we focus on the offline HCR in this paper.

HCR is a typical classification task, while deep convolutional neural network (DCNN) is one of the most exciting classification models and makes great progress in many fields. At present, DCNN has been widely applied to HCR and achieved significant performance improvements [5,7,8,31,32,35]. However, its success heavily relies on a large amount of labeled data which is high-cost. Moreover, DCNN-based HCR system suffers from the shift between the training and test distributions [33] when the variance of handwriting styles is large. In the previous works, there are mainly three methods proposed to alleviate this issue, which are summarized as follows: 1) using data augmentation techniques [5,7] to generate more data with different handwriting styles, such as affine transformation [19] and distorted generation [14]; 2) adopting writer-adaptation to match the feature distributions from the source domain to the target domain [30,32,33]; 3) designing writer-independent features manually to reduce within-class variation of character shape [32,35], such as normalization-cooperated gradient features [16]. These methods are well-designed on the basis of domain-specific knowledge to compensate for shape variation caused by various handwriting styles. However, generating more distorted data is insufficient to cover all the variations of handwritten characters, while writer adaptation methods need to match specific writers, and handcrafted writer-independent feature is so subjective that some pretty important information in characters may be lost. Besides, they don't explicitly model the final recognition objective, and they cannot take advantage of extra information.

As known to all, standard printed characters are writer-independent and present more semantic contents of characters. While handwritten characters contain various handwriting styles information of writers, as shown in Fig. 1, which tremendously

* Corresponding author:
E-mail address: lwj@nlpr.ia.ac.cn (W. Liu).
[1] Shuai Nie makes equal contribution to this work with Yaping Zhang.

https://doi.org/10.1016/j.patrec.2018.02.006
0167-8655/

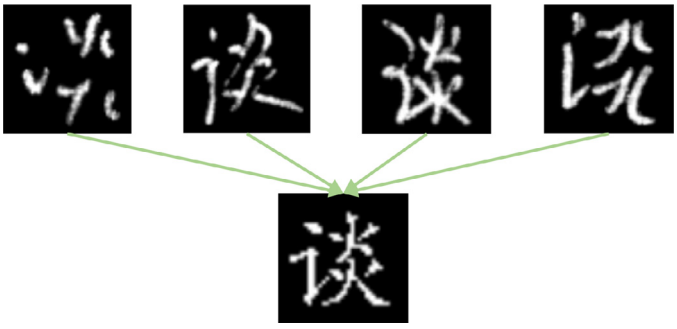

Fig. 1. All different handwritten characters could be representable in standard printed characters which are writer-independent.

interferes with the recognition of handwritten characters. Many experiments show that the printed characters can be more easily recognized without the interference of handwriting styles of writers [1]. In fact, standard printed characters are usually used as stencils to instruct us to recognize new characters, especially when the pupils learn to read from a textbook. Therefore, it is possible to improve the performance of HCR by exploiting writer-independent semantic features, while standard printed characters could be used as prior knowledge to guide models to exploit these features.

Generative adversarial network (GAN) is a well-known adversarial learning model [10]. It is composed of a generator and a discriminator. The discriminator can guide the generator to adjust a complex data distribution to another specific distribution by adversarial learning. When GAN is performed on a handwritten character set, such as MNIST dataset [13], it's interesting to observe that the generator could transfer noise vectors to realistic character images [6,26]. What's more, [28] used GAN to transfer images from street view house number (SVHN) dataset to the domain of MNIST dataset via GAN. Therefore, as shown in Fig. 2(a), it could be expected that the handwritten characters with various handwriting styles could be transferred into standard printed characters by adversarial learning.

Inspired by GAN, we propose a novel adversarial feature learning (AFL) model to exploit writer-independent semantic features for HCR. As shown in Fig. 2(b), AFL is a variant of GAN and composed of a feature extractor, a discriminator and a classifier, which concentrates the strengths of discriminative model and generative model for HCR. The feature extractor is used to extract encoding features of handwritten and printed characters. The discriminator judges whether the extracted features come from handwritten or standard printed characters. With the prior knowledge provided by standard printed characters, it can guide the feature extractor to exploit writer-independent semantic features from handwritten characters automatically. Finally, the extracted features are fed into the classifier to recognize the handwritten characters. The feature extractor, the discriminator and the classifier are jointly optimized by adversarial training. In the process of adversarial training, the prior knowledge from standard printed characters and writer-independent semantic features are incorporated, and hence we could get better performance of HCR.

We summarize our contributions as follows: 1) we introduce the writer-independent standard printed data as prior knowledge, which is learnt by AFL objectively, rather than use handcrafted writer-independent features which need a great amount of domain knowledge; 2) the proposed model could exploit writer-independent semantic features automatically, which in turn alleviates the large variance of handwriting styles for HCR; 3) the proposed AFL model could make better classification, which concentrates the strengths of discriminative model and generative model; 4) we achieve superior performance than the state-of-the-art model results on offline ICDAR-2013 handwritten Chinese character recognition competition dataset.

The remaining parts of this paper are organized as follows. Section 2 firstly reviews the related works. Then, we describe the proposed AFL method in Section 3. Experimental results and its detailed analysis are presented in Section 4. Finally, we draw concluding remarks in Section 5.

2. Related work

GAN [10] is to learn a generative model synthesizing images similar to real images through a two-player game between a generator and discriminator (in Fig. 2(a)). The key idea of GAN is an adversarial loss that forces the generator could find the mapping between noise vectors and real images. Despite many promising developments [2,11], more recent works focus on image synthesis, such as image generation [4,22] and representation learning [23]. There are a few trials to use GAN to make classification. [27] and [21] generalized GAN to learn a discriminative classifier, which is trained to not only classify images but also identify real and fake images. However, it's incompatible that a single discriminator network plays two competing roles of identifying fake samples and predicting labels, which results in hard achieving equilibrium between discriminator and generator, and in turn prevents the model from predicting labels accurately. Recently, [15] proposed triple-GAN to make two competing roles in discriminator separated to introduce three components for both classification and class-conditional generation. But the structure of triple-GAN is complex for classification. In the proposed AFL model, we concentrate on extracting writer-independent features for classification directly. The domain adversarial training proposed by [9] is closest to our AFL model, while it promotes invariant with respect to the shift between two domains for unsupervised domain adaptation. Our task is quite different from theirs. We introduce the standard printed data to guide the model to extract writer-independent semantic features for character recognition directly.

3. Adversarial feature learning

The proposed AFL model tries to improve the performance of HCR by learning writer-independent semantic features of a handwritten characters, where we provide standard printed characters as prior knowledge. This is different from the conventional feature learning of DCNN, where the primary goal is to extract discriminative feature matching training set. AFL is composed of three neural network components: a feature extractor (F) that characterizes the features of handwritten and standard printed characters, a classifier (C) that could make correct classification for the extracted features, and a discriminator (D) that acts as a teacher could guide F to learn prior knowledge. F, C, and D are jointly optimized by the adversarial training algorithm. The desired equilibrium is that F will concentrate on writer-independent features and can keep discriminative with the constraints of D and C.

3.1. Formulation

Let's denote a handwritten character image as x_h. The corresponding feature vector is denoted as $F(x_h, \theta_f)$ that is extracted by the feature extractor F. The extracted feature $F(x_h, \theta_f)$ is input into the classifier C to compute its classification probability:

$$P(y_c = y_h | x_h; \theta_f, \theta_c)) = \frac{e^{-C(y_c = y_h | F(x_h, \theta_f); \theta_c))}}{\sum_{j=0}^{L} e^{-C(y_c = j | F(x_h, \theta_f); \theta_c)}} \quad (1)$$

where $y_h \in Y_h$, $Y_h = \{1, 2, \ldots, L\}$ is a finite label set, L is the number of character classes, y_c denotes the predicted label, θ_f and θ_c

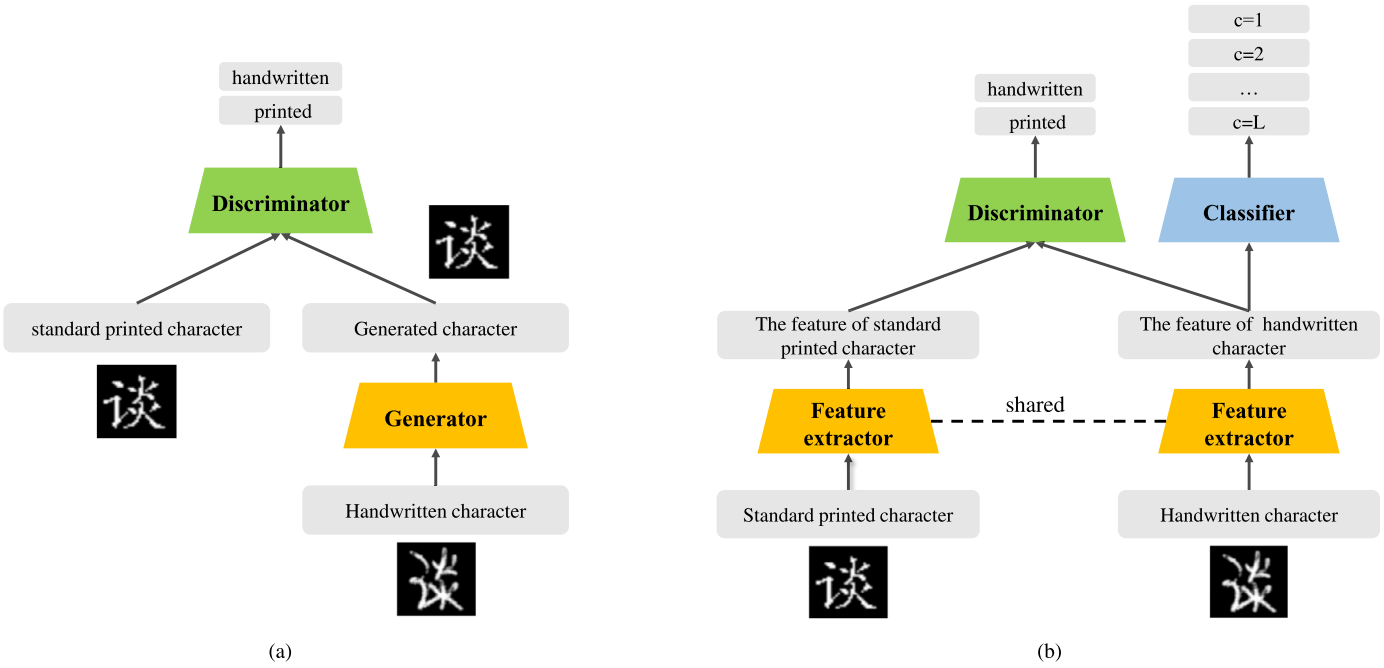

Fig. 2. *(a)* The structure of GAN: the discriminator is to identify whether a sample is real data or synthetic data generated by the generator. *(b)* The structure of AFL: the shared feature extractor is to extract writer-independent features from standard printed characters and handwritten characters, respectively; the discriminator is to distinguish that the extracted feature is from a standard printed character or not; and the classifier is to make correct classification of the extracted feature of handwritten characters, and equations $c = 1, \ldots, c = L$ denote the character category. (best view in color).

denote the parameters of F and C, respectively. The combination of F and C as a whole can be regarded as a character recognizer. C is optimized by minimizing the category loss $L_c(x_h; \theta_f, \theta_c)$ expressed as:

$$L_c(x_h; \theta_f, \theta_c) = -E_{x_h \in X_h}[log(P(y_c = y_h | x_h; \theta_f, \theta_c))] \tag{2}$$

where X_h denotes a handwritten character set containing characters with different handwriting styles.

The joint optimization of F and C is equal to the optimization of DCNN, while DCNN is a discriminative model which achieves high performance only when the training set and test set are identically distributed. However, the huge variance of handwriting styles across different writers causes the shift between the distributions of handwritten character training set and test set. To alleviate this issue, the proposed AFL model introduces standard printed characters to guide F to discover the writer-independent semantic features by adversarial learning. Standard printed characters can be regarded as prior knowledge. This is exploited by a discriminator.

The discriminator D is used to judge whether the extracted feature vector by F comes from handwritten or standard printed characters. We denote the feature label as y_d. The feature vector of a standard printed character corresponds to $y_d = 1$, while the feature vector of a handwritten character corresponds to $y_d = 0$. The probability of $y_d = 1$ and $y_d = 0$ are expressed as equation (3) and (4), respectively:

$$P(y_d = 1 | x; \theta_f, \theta_d) = \frac{1}{1 + e^{-D[F(x, \theta_f), \theta_d]}} \tag{3}$$

$$P(y_d = 0 | x; \theta_f, \theta_d) = 1 - P(y_d = 1 | x; \theta_f, \theta_d) \tag{4}$$

where x denotes the input from handwritten characters or standard printed characters, and θ_d denotes the parameters of D. D is optimized by minimizing the discriminator loss $L_d(x_h, x_p; \theta_f, \theta_d)$ as:

$$\begin{aligned} L_d(x_h, x_p; \theta_f, \theta_d) = & -E_{x_h \in X_h}[log(P(y_d = 0 | x_h; \theta_f, \theta_d))] \\ & -E_{x_p \in X_p}[log(P(y_d = 1 | x_p; \theta_f, \theta_d))] \end{aligned} \tag{5}$$

where X_p is a standard printed character set, and x_p is a sample from X_p.

F in the proposed AFL model is learnt to discover writer-independent semantic features that can fool D. The writer-independent semantic features could be seen as underlying features which standard characters and handwritten characters share in common, D can't distinguish them. Namely, the more writer-independent features F extracts, the larger discriminator loss $L_d(x_h, x_p; \theta_f, \theta_d)$ is. Therefore, F could be optimized by minimizing the adversarial feature loss $L_f(x_h, x_p; \theta_f, \theta_d, \theta_c)$ expressed as:

$$L_f(x_h, x_p; \theta_f, \theta_d, \theta_c) = L_c(x_h; \theta_f, \theta_c) - \alpha L_d(x_h, x_p; \theta_f, \theta_d) \tag{6}$$

where α is a hyper-parameter to control the tradeoff between the category loss L_c and the discriminator loss L_d. When $\alpha = 0$, the model is equivalent to a traditional deep neural network model. Furthermore, when the value of α is too large, the feature extractor may only focus on writer-independent but not discriminative features.

3.2. Optimization

AFL model is trained iteratively for robust handwritten character recognition. D is optimized to distinguish the features of standard printed characters X_p from the features of handwritten characters X_h, F is optimized to extract writer-independent and discriminative features of X_h that can fool D, and C is optimized by a fine-tuning strategy to make correct classification of the extracted features. With D being more powerful in distinguishing whether features are writer-independent or not and C being more precise in classifying features, F strives for extracting better features to compete with D and C. Finally, D, F and C improve each other in the progress of adversarial training.

In other words, D is optimized by minimizing the discriminator loss function L_d, while the writer-independent features extracted by F, which are similar with the features extracted from printed data, will result in maximizing the discriminators loss L_d. In other

words, *F*, *C* and *D* play the minimax game with loss function:

$$\min_{\theta_f,\theta_c}\max_{\theta_d} L_f(\theta_f,\theta_d,\theta_c) = L_c(x_h;\theta_f,\theta_c) - \alpha L_d(x_h,x_p;\theta_f,\theta_d) \tag{7}$$

Suppose the optimal parameters are $\hat{\theta}_f, \hat{\theta}_c, \hat{\theta}_d$, then we have

$$\hat{\theta}_f, \hat{\theta}_c = \text{argmin}\ L_f(\theta_f, \hat{\theta}_d, \theta_c) \tag{8}$$

$$\hat{\theta}_d = \text{argmax}\ L_f(\hat{\theta}_f, \theta_d, \hat{\theta}_c) \tag{9}$$

The opposite optimization goals for the parameters θ_f in *F* and the parameters θ_d in *D* make difficulties for developing *D*, *F* and *C* in one updating procedure. Therefore, we limit the update to the respective specific component in AFL method. *D*, *F*, and *C* are alternatively trained following the adversarial learning framework. As shown in Algorithm 1, we connect *F* and *C* to pretrain a primary character recognizer. Then we alternatively train the parameters of *D*, *F* and *C* to fine-tune the model. *D*, *F* and *C* are implemented with neural networks in the proposed AFL model, and the parameters are updated by gradient descent:

Algorithm 1 The adversarial feature learning algorithm.

Input: Dataset $(\boldsymbol{x}_h, \boldsymbol{y}_h) \in (\boldsymbol{X}_h, \boldsymbol{Y}_h)$, $\boldsymbol{x}_p \in \boldsymbol{X}_p$, randomly initialized a feature extractor *F*, a classifier *C*, a discriminator *D* parameterized by $\theta_f, \theta_c, \theta_d$

Output: the optimized *F*, *C* and *D* parameterized by $\hat{\theta}_f, \hat{\theta}_c, \hat{\theta}_d$

```
// Pretrain a whole character classifier
for number of pretraining epochs do
    for number of Mini-batches do
        (θ_f, θ_c) ← (θ_f, θ_c) − μ(∂L_c(x_h;θ_f,θ_c)/∂θ_f, ∂L_c(x_h;θ_f,θ_c)/∂θ_c)
    end for
end for
// the adversarial training
repeat
    for number of training epochs do
        for number of Mini-batches do
            // for discriminator
            θ_d ← θ_d − μ ∂L_d(x_h,x_p;θ_f,θ_d)/∂θ_d
            // for feature extractor
            θ_f ← θ_f − μ(∂L_c(x_h;θ_f,θ_c)/∂θ_f − α ∂L_d(x_h,x_p;θ_f,θ_d)/∂θ_f)
            // for classifier
            θ_c ← θ_c − μ ∂L_c(x_h;θ_f,θ_c)/∂θ_c
        end for
    end for
until convergence
θ̂_f = θ_f, θ̂_c = θ_c, θ̂_d = θ_d
return θ̂_f, θ̂_c, θ̂_d
```

- For discriminator:

$$\theta_d \leftarrow \theta_d - \mu\frac{\partial L_d(x_h,x_p;\theta_f,\theta_d)}{\partial\theta_d} \tag{10}$$

- For feature extractor:

$$\theta_f \leftarrow \theta_f - \mu\left(\frac{\partial L_c(x_h;\theta_f,\theta_c)}{\partial\theta_f} - \alpha\frac{\partial L_d(x_h,x_p;\theta_f,\theta_d)}{\partial\theta_f}\right) \tag{11}$$

- For classifier:

$$\theta_c \leftarrow \theta_c - \mu\frac{\partial L_c(x_h;\theta_f,\theta_c)}{\partial\theta_c} \tag{12}$$

where, μ is the learning rate we used for training model.

Table 1
The configuration of generating standard printed characters.

Constitution	Variations	Description
Font	10	LiSu, FangSong, STKaiti, NSimSun, KaiTi, SimSun, SimHei, Microsoft YaHei, YouYuan, STXinwei
Size	5	12, 15, 18, 21, 24
Boldness	3	Font weight: 200, 400, 700

4. Experiment

4.1. Datasets

We have conducted preliminary experiments on the widely adopted MNIST [13]. MNIST consists of 60,000 training samples and 10,000 test samples of handwritten digits of size 28×28, which have 10 different classes from 0 to 9. To further evaluate the effectiveness of the proposed AFL algorithm for HCR, we present our method on the challenging ICDAR-2013 offline handwritten Chinese character recognition competition dataset [31], which is a large scale classification task with great diversity in handwriting styles. It consists of 224,419 Chinese character samples produced by 60 different writers, and the number of character classes is 3755 (in level-1 set of GB2312-80). For training set, we use the offline handwritten Chinese characters datasets CASIA–HWDB1.0 and CASIA–HWDB1.1 collected by CASIA [17], which are totally 2,678,424 samples of 3755 character classes considered in the ICDAR-2013 competition dataset.

Standard printed characters which are introduced as prior knowledge are generated from Microsoft Windows fonts [3]. As shown in Table 1, the grayscale character images that correspond to MNIST and CASIA-HWDB are both generated with variations commonly seen in printed character images, including 10 fonts, 5 sizes, and 3 degrees of boldness. Given these variations, each standard printed character is associated with 150 different representations.

4.2. Setup

We briefly summarize our experimental settings in this scenario. *F* is composed of one input layer, one flatten output layer, and several convolutional blocks. Each convolutional block consists of a convolutional layer with 3×3 filers, followed by batch normalization [12] layer and leaky rectified linear unit (Leaky Relu) [18] activation function. *C* is a multilayer perceptron (MLP) only with one hidden fully connected layer. We use rectified linear unit (Relu) [20] as the activation of the hidden layer, and a softmax function as the activation of the output layer. And we also only use a fully connected hidden layer to construct *D* with one input layer and one output layer. The detail configurations of the proposed AFL model for MNIST and CASIA-HWDB are shown in Tables 2 and3, respectively. All the weights in AFL components are first initialized with normalized initialization (Glorot and Bengio, 2010). Then, all of our networks are optimized with Adam where the learning rate is initially set to 2e-4 and the first order of momentum is 0.5.

4.3. Comparison of different hyper-parameter α

This scenario aims to evaluate how the hyper-parameter α in equation (6) affects the performance of HCR. α controls the trade-off between the writer-independent and discriminative feature. On the one hand, when α is tiny, the category loss plays a main role in model and the discriminator loss rarely works, which may result in the feature extractor only focus on discriminative feature.

Table 2
MNIST.

Feature extractor F	
Input: 28×28 Gray image	
3×3 conv, 32, BN, LeakyReLU	
3×3 conv, 32, 2×2 subsample, BN, LeakyReLU	
3×3 conv, 64, BN, LeakyReLU	
3×3 conv, 64, 2×2 subsample, BN, LeakyReLU	
3×3 conv, 128, BN, LeakyReLU	
3×3 conv, 128, 2×2 subsample, BN, LeakyReLU	
Classifier C	Discriminator D
Input: feature vector	Input: feature vector
MLP 256 units, ReLU	MLP 128 units, ReLU
10-class softmax	MLP 1 unit, sigmoid

Table 3
CASIA–HWDB.

Feature extractor F	
Input: 64×64 Gray image	
3×3 conv, 96, BN, LeakyReLU	
3×3 conv, 96, 2×2 subsample, BN, LeakyReLU	
3×3 conv, 128, BN, LeakyReLU	
3×3 conv, 128, 2×2 subsample, BN, LeakyReLU	
3×3 conv, 160, BN, LeakyReLU	
3×3 conv, 160, 2×2 subsample, BN, LeakyReLU	
3×3 conv, 256, BN, LeakyReLU	
3×3 conv, 256, 2×2 subsample, BN, LeakyReLU	
3×3 conv, 256, BN, LeakyReLU	
Classifier C	Discriminator D
Input: feature vector	Input: feature vector
MLP 512 units, ReLU	MLP 512 units, ReLU
0.5 dropout	0.5 dropout
3755-class softmax	MLP 1 unit, sigmoid

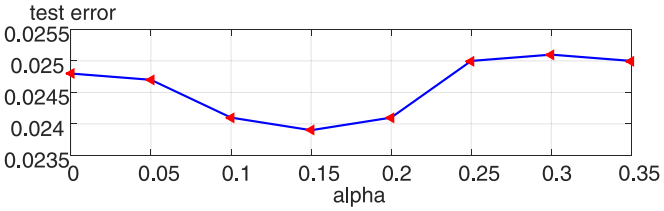

Fig. 3. Comparison of different hyper-parameter α.

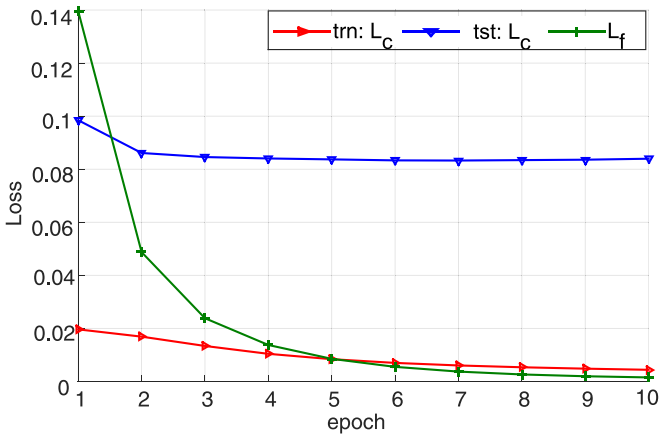

Fig. 4. The learning curve of AFL. 'trn' and 'tst' denote the results are on the training and test set, respectively. Note that the AFL model is trained by a fine-tune strategy on a pretrain character classifier which consists of F and C. (best view in color).

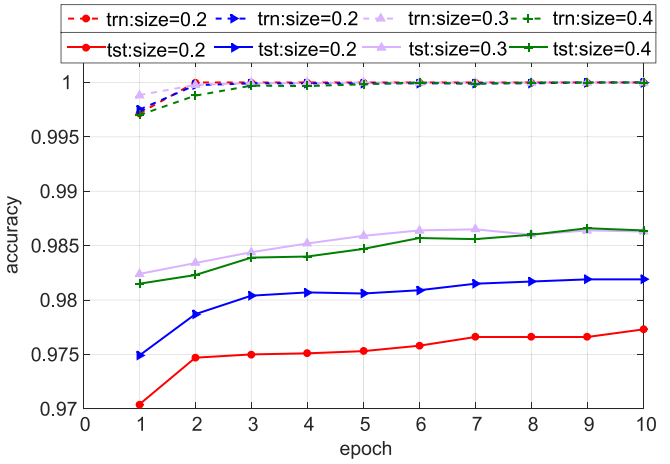

Fig. 5. The comparison between the accuracy of training set and test set. 'trn' and 'tst' denote the results are on the training and test set, respectively. 'size= 0.1', 'size= 0.2', 'size= 0.3' and 'size= 0.4' denote the ratio of samples randomly selected from MNIST training set. (best view in color).

On the other hand, when α is huge, the model concentrates on the writer-independent feature which may be lack of distinctiveness with weak constraint of category loss. Thus, an appropriate α is very important for the feature extractor to extract better feature to get better performance of HCR. For simplicity, we only conduct the experiments on MNIST. We use 6000 handwritten digit samples randomly selected from MNIST training set to train model, and evaluate the performance of HCR on MNIST test set. Specifically, we explore the different α from {0, 0.05, 0.1, 0.15, 0.20, 0.25, 0.30, 0.35} while keeping the other hyper-parameters fixed in the experiments. As shown in Fig. 3, the test error rate decreases with the increase of α. However, when α reaches a specific value, the error rate will increase. Especially, when $\alpha > 0.25$, the improvement of the performance of HCR will disappear. This suggests that α should be carefully selected for AFL to balance the writer-independent and discriminative feature. In this paper, we choose α from {0.1, 0.15, 0.2} for the following experiments.

4.4. Analysis on the adversarial feature learning

In this section, we preliminarily analyze how the AFL affects the performance of HCR. Fig. 4 shows the learning curve of AFL. The pretty dramatic decline of the adversarial feature loss L_f predicates that we get better features which are writer-independent and discriminative. Therefore, we can get better performance, even though the category loss L_c in training set keeps almost unchangeable. As shown in Fig. 5, when the accuracy of HCR in training set approaches to saturation, the accuracy in the test set can still get constant improvement.

Fig. 6 presents the comparison between AFL and non-AFL training method with different training data size. In the experiment, we use different size of data randomly sampling from MNIST training set while keeping the other hyper-parameters fixed. It's exciting that AFL could get significant improvement when available training data is small. Although, a sufficiently large training data could approximately describe the distribution of data, we observe that AFL can still get better performance of HCR. These phenomena imply that the variance between different handwriting styles could be alleviated by extracting writer-independent semantic features, and hence could improve the performance of HCR.

4.5. Handwritten Chinese character recognition

In this section, we study the effects of AFL method for handwritten Chinese character recognition(HCCR). First, we train a DCNN character classifier without adversarial learning as the base-

Table 4
Comparison of different methods for ICDAR-2013 offline HCCR competition.

Method	Accuracy	Memory	Handcrafted feature	Data augmentation
CNN-single [5]	96.58%	190.0 MB	No	Yes
Ensemble-CNN-voting [5]	96.79%	950.0 MB	No	Yes
HoG-CNN [35]	96.25%	–	Yes	No
Gradient-CNN [35]	96.28%	–	Yes	No
Gabor-CNN [35]	96.35%	27.77 MB	Yes	No
Ensemble-CNN-4 [35]	96.64%	110.9 MB	Yes	No
Ensemble-CNN-10 [35]	96.74%	270.0 MB	Yes	No
DCNN-Similarity ranking [7]	97.07%	36.20 MB	No	Yes
Ensemble-DCNN-Similarity ranking [7]	**97.64%**	144.8 MB	No	Yes
DirectMap-CNN [32]	**96.95%**	23.50 MB	Yes	No
DirectMap-CNN-Adaptation [32]	97.37%	**23.50** MB	Yes	No
DCNN case1	96.64%	18.20 MB	No	No
DCNN case2	96.60%	18.20 MB	No	No
AFL(ours)	**98.29%**	**18.20** MB	No	No

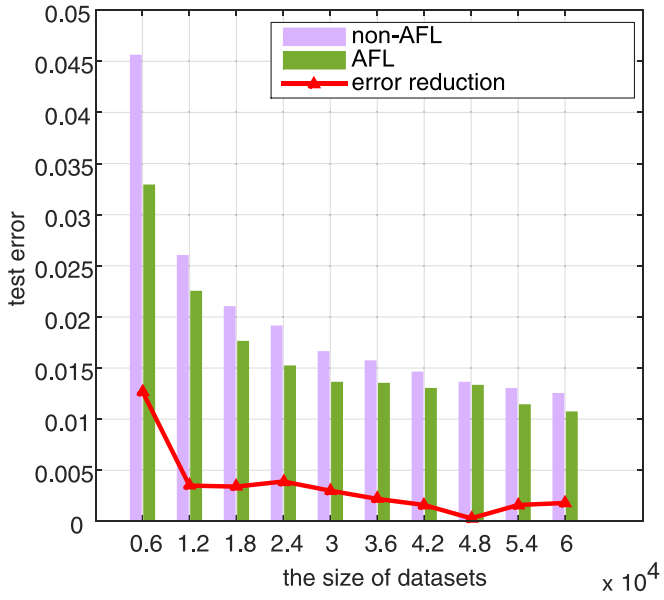

Fig. 6. Comparison of non-AFL and AFL classifier on MNIST. Here, error reduction means the difference between two classifiers' test error. (best view in color).

line. The architecture of DCNN model is the same as the architecture of character recognizer in AFL, which is composed of F and C. To study the effects of standard printed characters, we train the DCNN model in two different cases with the same training procedure. In DCNN case1, we only use CASIA–HWDB1.0-1.1 dataset without any data augmentation, and the character recognition accuracy rate is 96.64%. While we supply printed samples for the model in DCNN case2, we get the recognition accuracy of 96.60%, and we fail to get obvious improvement on the performance of HCCR. However, when we use the AFL method, we could achieve the best recognition accuracy of 98.29%, and outperforms the baseline result with a relative 49.11% error rate reduction. We could make an inference that the improved performance owes to robust writer-independent feature being learnt, rather than more printed samples.

Furthermore, we compare the proposed AFL model with different DCNN methods on HCCR. Table 4 presents the results of different methods on ICDAR-2013 offline competition database. It can be seen that the proposed AFL method achieves the state-of-the-art recognition accuracy of 98.29%, and outperforms the previous best result with a relative 27.54% error rate reduction [7]. Moreover, [7] used data augmentation techniques to generate more samples, while we only use the available printed data. Besides, it's worth noting that the proposed model doesn't use any handcrafted feature, but we achieve better performance than the methods that use well-designed manual features [32,35]. This mainly owes to AFL being able to automatically exploit writer-independent semantic features with the guidance of standard printed characters. We also observe that our model outperforms the writer-adaptation model that uses domain-specific knowledge [32]. It is more exciting that our model uses the least model parameters and achieves the best performance. Even compared with the ensemble models [5,7,35] which usually need to train several sub-models, the proposed model can still have a distinct advantage. This is very important for the practical HCR system which requires high real-time.

5. Conclusion

In this paper, we improve the performance of HCR by exploiting writer-independent semantic features with the prior knowledge of standard printed character, which is implemented by the proposed AFL model. Compared with the DCNN methods for HCR, the proposed AFL model could get significant performance improvement, especially when the available training data is inadequate. Specifically, we achieve the state-of-the-art result on offline handwritten Chinese character recognition.

We mention that our AFL model based on CNN could extend to online handwritten character recognition, where we need to transform the online handwriting trajectories into image-like representations. The RNN based approach, which is proposed by [34], directly deals with the sequential structure for online handwritten character recognition. We agree its on the list of things worth studying. When AFL is combined with RNN, the network structures should be redesigned carefully, and we will exploring it with AFL in future work.

Moreover, our AFL model is easily to extend with a few changes to alleviate other robust recognition problem, such as speech recognition with accents and robust face recognition with different poses.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (no. 61573357, no. 61503382, no. 61403370, no. 61273267, no. 91120303).

References

[1] N. Arica, F.T. Yarman-Vural, An overview of character recognition focused on off-line handwriting, IEEE Trans. Syst. Man. Cybern. Part C 31 (2) (2001) 216–233.
[2] M. Arjovsky, S. Chintala, L. Bottou, Wasserstein gan, arXiv:1701.07875 (2017).

[3] J. Bai, Z. Chen, B. Feng, B. Xu, Chinese image character recognition using dnn and machine simulated training samples, in: International Conference on Artificial Neural Networks, Springer, 2014, pp. 209–216.
[4] D. Berthelot, T. Schumm, L. Metz, Began: boundary equilibrium generative adversarial networks, arXiv:1703.10717 (2017).
[5] L. Chen, S. Wang, W. Fan, J. Sun, S. Naoi, Beyond human recognition: A cnn-based framework for handwritten character recognition, in: Pattern Recognition (ACPR), 2015 3rd IAPR Asian Conference on, IEEE, 2015, pp. 695–699.
[6] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, P. Abbeel, Infogan: Interpretable representation learning by information maximizing generative adversarial nets, in: Advances in Neural Information Processing Systems, 2016, pp. 2172–2180.
[7] C. Cheng, X.-Y. Zhang, X.-H. Shao, X.-D. Zhou, Handwritten chinese character recognition by joint classification and similarity ranking, in: Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on, IEEE, 2016, pp. 507–511.
[8] D. Cireşan, U. Meier, Multi-column deep neural networks for offline handwritten chinese character classification, in: Neural Networks (IJCNN), 2015 International Joint Conference on, IEEE, 2015, pp. 1–6.
[9] Y. Ganin, V. Lempitsky, Unsupervised domain adaptation by backpropagation, in: International Conference on Machine Learning, 2015, pp. 1180–1189.
[10] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, Y. Bengio, Generative adversarial nets, in: Advances in neural information processing systems, 2014, pp. 2672–2680.
[11] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, A. Courville, Improved training of wasserstein gans, arXiv:1704.00028 (2017).
[12] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.
[13] Y. LeCun, C. Cortes, C.J. Burges, Mnist handwritten digit database, AT&T Labs 2 (2010). [Online]. Available: http://yann.lecun.com/exdb/mnist.
[14] K. Leung, C.H. Leung, Recognition of handwritten chinese characters by combining regularization, fisher's discriminant and distorted sample generation, in: Document Analysis and Recognition, 2009. ICDAR'09. 10th International Conference on, IEEE, 2009, pp. 1026–1030.
[15] C. Li, K. Xu, J. Zhu, B. Zhang, Triple generative adversarial nets, arXiv:1703.02291 (2017).
[16] C.-L. Liu, Normalization-cooperated gradient feature extraction for handwritten character recognition, IEEE Trans. Pattern Anal. Mach. Intell. 29 (8) (2007) 1465–1469.
[17] C.-L. Liu, F. Yin, D.-H. Wang, Q.-F. Wang, Casia online and offline chinese handwriting databases, in: Document Analysis and Recognition (ICDAR), 2011 International Conference on, IEEE, 2011, pp. 37–41.
[18] A.L. Maas, A.Y. Hannun, A.Y. Ng, Rectifier nonlinearities improve neural network acoustic models, in: Proc. ICML, vol. 30, 2013.
[19] H. Miyao, M. Maruyama, Virtual example synthesis based on pca for off–line handwritten character recognition, in: Document Analysis Systems, vol. 7, Springer, 2006, pp. 96–105.
[20] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Proceedings of the 27th international conference on machine learning (ICML-10), 2010, pp. 807–814.
[21] A. Odena, Semi-supervised learning with generative adversarial networks, arXiv:1606.01583 (2016).
[22] A. Odena, C. Olah, J. Shlens, Conditional image synthesis with auxiliary classifier gans, arXiv:1610.09585 (2016).
[23] A. Radford, L. Metz, S. Chintala, Unsupervised representation learning with deep convolutional generative adversarial networks, arXiv:1511.06434 (2015).
[24] J. Richarz, S. Vajda, R. Grzeszick, G.A. Fink, Semi-supervised learning for character recognition in historical archive documents, Pattern Recognit. 47 (3) (2014) 1011–1020.
[25] T. Saba, A. Rehman, M. Elarbi-Boudihir, Methods and strategies on off-line cursive touched characters segmentation: a directional review, Artif. Intell. Rev. (2014) 1–20.
[26] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, X. Chen, Improved techniques for training gans, in: Advances in Neural Information Processing Systems, 2016, pp. 2234–2242.
[27] J.T. Springenberg, Unsupervised and semi-supervised learning with categorical generative adversarial networks, arXiv:1511.06390 (2015).
[28] Y. Taigman, A. Polyak, L. Wolf, Unsupervised cross-domain image generation, arXiv:1611.02200 (2016).
[29] X. Wei, S. Lu, Y. Wen, Y. Lu, Recognition of handwritten chinese address with writing variations, Pattern Recognit. Lett. 73 (2016) 68–75.
[30] H.-M. Yang, X.-Y. Zhang, F. Yin, Z. Luo, C.-L. Liu, Unsupervised adaptation of neural networks for Chinese handwriting recognition, in: Frontiers in Handwriting Recognition (ICFHR), 2016 15th International Conference on, IEEE, 2016, pp. 512–517.
[31] F. Yin, Q.-F. Wang, X.-Y. Zhang, C.-L. Liu, Icdar 2013 Chinese handwriting recognition competition, in: Document Analysis and Recognition (ICDAR), 2013 12th International Conference on, IEEE, 2013, pp. 1464–1470.
[32] X.-Y. Zhang, Y. Bengio, C.-L. Liu, Online and offline handwritten chinese character recognition: a comprehensive study and new benchmark, Pattern Recognit. 61 (2017) 348–360.
[33] X.-Y. Zhang, C.-L. Liu, Writer adaptation with style transfer mapping, IEEE Trans. Pattern Anal. Mach. Intell. 35 (7) (2013) 1773–1787.
[34] X.Y. Zhang, F. Yin, Y.M. Zhang, C.L. Liu, Y. Bengio, Drawing and recognizing chinese characters with recurrent neural network, IEEE Trans. Pattern Anal. Mach. Intell. PP (99) (2017). 1–1, doi: 10.1109/TPAMI.2017.2695539.
[35] Z. Zhong, L. Jin, Z. Xie, High performance offline handwritten Chinese character recognition using googlenet and directional feature maps, in: Document Analysis and Recognition (ICDAR), 2015 13th International Conference on, IEEE, 2015, pp. 846–850.