Handwriting Style Mixture Adaptation

Hong-Ming Yang[1,2], Xu-Yao Zhang[1,2], Fei Yin[1], Cheng-Lin Liu[1,2]
[1]National Laboratory of Pattern Recognition, Institute of Automation of Chinese Academy of Sciences
95 Zhongguancun East Road, Beijing 100190, P.R. China
[2]University of Chinese Academy of Sciences, Beijing, P.R. China
Email: {hongming.yang, xyz, fyin, liucl}@nlpr.ia.ac.cn

***Abstract*—In handwriting recognition, the test data usually come from multiple writers which are not shown in the training data. Therefore, adapting the base classifier towards the new style of each writer can significantly improve the generalization performance. Traditional writer adaptation methods usually assume that there is only one writer (one style) in the test data, and we call this situation as *style-clear adaptation*. However, a more common situation is that multiple handwriting styles exist in the test data, which is widely appeared in multi-font documents and handwriting data produced by the cooperation of multiple writers. We call the adaptation in this situation as *style-mixture adaptation*. To deal with this problem, in this paper, we propose a novel method called K-style mixture adaptation (K-SMA) with the assumption that there are totally K styles in the test data. Specifically, we first partition the test data into K groups (style clustering) according to their style consistency, which is measured by a newly designed style feature that can eliminate class (category) information and keep handwriting style information. After that, in each group, a style transfer mapping (STM) is used for writer adaptation. Since the initial style clustering may be not reliable, we repeat this process iteratively to improve the adaptation performance. The K-SMA model is fully unsupervised which do not require either the class label or the style index. Moreover, the K-SMA model can be effectively combined with the benchmark convolutional neural network (CNN) models. Experiments on the online Chinese handwriting database CASIA-OLHWDB demonstrate that K-SMA is an efficient and effective solution for style-mixture adaptation.**

I. Introduction

For most machine learning and pattern recognition algorithms, there is a basic assumption that the training and test data are independently and identically (iid) drawn from the same distribution. However, for many real applications, this assumption is usually unsatisfied. In handwriting recognition, because the large variability of handwriting styles, we often observe changing distributions across different writers, which makes handwriting recognition a challenging problem. To deal with this problem, we should transfer or adapt the classifier learned on the training data to the new distribution of the particular writer. This process is called writer adaptation [1] which is a special case of transfer learning [2].

In writer adaptation, for adapting the base classifier to a specific writer, we should collect some data from this particular writer. In other words, we have an assumption that the data used for adaptation are all from the same writer. However, when the test data come from multiple different writers, the situation becomes much more complicated. If we can partition the test data according to their writer index, then the adaptation will become much easier because each writer are assumed to have their own handwriting style consistency, and we call the adaptation in this situation as *style-clear adaptation*. However, as shown in Fig. 1, a more common situation is that we do not have the writer (or style) information, the test data are randomly mixed. In this case, writer adaptation is obviously more difficult and we call the adaptation in this situation as *style-mixture adaptation*. In style-mixture adaptation, we do not have the style information of the samples, hence it will reduce the cost of labelling the data into their style category, for example, in a multi-font document, identifying which font each character belongs to is a difficult task. Instead, in style-mixture adaptation, we consider the adaptation and style identification problem simultaneously in an unified framework. Compared with style-clear adaptation, style-mixture adaptation is more general and practical. However, to the best of our knowledge, most previous works are concentrated on the style-clear adaptation, and there are rare researches on the style-mixture adaptation problem.

In this paper, we propose a novel framework called K-SMA to deal with the style-mixture adaptation problem. The CNN is used as our base classifier. The first step in K-SMA is to partition the test data into K groups according to their style consistency. A style feature is designed to capture the handwriting style information by decreasing the influence from the class information and enhancing the style information of the particular writer. Based on this feature, the K-means algorithm is used for style clustering. In each cluster, the samples are style-consistent, then the style transfer mapping (STM) [1] can be used for writer adaptation. The processes of style clustering and writer adaptation are repeated iteratively to improve the performance. Experimental results show that our proposed method is both efficient and effective for style-mixture adaptation.

The rest of this paper is organized as following. Section II describes the related works; Section III introduces the style-mixture adaptation problem. After that, the proposed K-SMA method is described in Section IV, and the experimental results are presented in Section V. Finally, we give the conclusions in Section VI.

II. Related Works

In writer adaptation, the style transfer mapping (STM) [1] method uses a linear transformation to project the writer-specific data onto a style free space for adaptation. The linear

transformation has a closed form solution, and hence the learning of STM can be computed efficiently. Li et al. further apply the STM learning method to historical Chinese character recognition [3]. Feng et al. contribute an improvement for STM by using a nonlinear transformation of Gaussian Progress Regression as a replacement of the linear transformation in STM [4]. This modified method is proved to be useful for Dunhuang historical Chinese character recognition. Except for character recognition, Rodríguez et al. [5] first present a novel adaptation method based on a separation of the word-dependent parameters from the writer-dependent parameters in semi-continuous hidden Markov model for word spotting task.

Recently, the deep learning [6] becomes more and more popular, and many proposed writer adaptation methods are closely related to the deep neural network (DNN). Tang et al. [7] use some labeled target domain data to fine-tuning a CNN model trained on the source domain data for adaptation. Zhang et al. [8] train a high performance CNN and combine the CNN with STM for writer adaptation, this method achieves a new benchmark on handwritten Chinese character recognition. Du et al. [9] proposes a new criterion to learn a transformation for supervised writer adaptation, and the features used for learning are extracted by a trained CNN.

Writer adaptation is a special case of domain adaptation. In domain adaptation, Long et al. [10] propose combining multiple kernel maximum mean discrepancies (MK-MDD) loss with the negative log likelihood loss to train the DNN for learning transferable features for unsupervised domain adaptation. Ruder et al. [11] give an extension of knowledge distillation, called knowledge adaptation, to deal with the domain shift problem. Ghifary et al. [12] combine the unsupervised reconstruction (auto-encoder) loss of the test data with the supervised classification loss of the training data to train the DNN for unsupervised domain adaptation.

III. Style-Mixture Adaptation

We denote the source (training) and target (test) dataset as

$$Data_S = \{(x_i, y_i)_{i=1}^{N_S}\},\ Data_T = \{(x_i)_{i=1}^{N_T}\}. \quad (1)$$

In handwriting recognition, both datasets are composed of handwritten characters from completely different writers, which have their own exclusive styles and distributions. To deal with the variability of different styles in the test data, the base classifier C learned on the $Data_S$ should be adapted to each style in $Data_T$ respectively. This process is known as writer adaptation. According to whether we are given the style information of $Data_T$, we can classify the writer adaptation into *style-clear adaptation* and *style-mixture adaptation*.

A. Style-clear Adaptation

In this situation, the style information of $Data_T$ is clear to us. Therefore, we know which samples in $Data_T$ belong to the same style (writer), and the $Data_T$ in this case usually has the form as following

$$Data_T = \{(x_i)_{i=1}^{N_T}\} = \{data_k\}_{k=1}^{K} \quad (2)$$

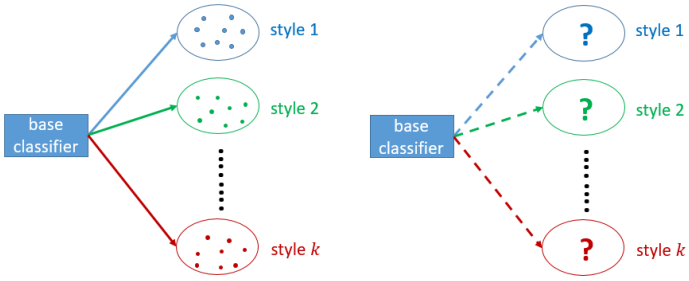

Fig. 1. Style-clear Adaptation (left) and Style-mixture Adaptation (right)

$data_k$ is the data of a particular writer k, and it can be denoted as

$$data_k = \{(x_{kj})_{j=1}^{N_k}\}. \quad (3)$$

With the help of the $data_k$, we can directly adapt the base classifier C to writer k. Because the data for each writer is clear to us, writer adaptation in this situation is relatively easy and most previous methods only consider adaptation under this setting. However, in some applications, the cost for labelling the style information or grouping the data according to their styles for $Data_T$ is expensive, especially when the samples in $Data_T$ are massive. Thus the style-clear assumption is sometimes idealized and lack of practicability. We give a graphic description for style-clear adaptation in Fig. 1.

B. Style-mixture Adaptation

In this situation, we are only given a style-mixture set $Data_T$ but without its style information. Hence we can not pick out the data from $Data_T$ for a specific writer k. Without the help of the writer specific data, adapting the base classifier C to writer k exactly as before is impossible. Thus the previous methods for style-clear adaptation are inapplicable in this case. Compared with style-clear adaptation, the style-mixture adaptation is more difficult, but it is more general and has more applications. We mainly concentrate on the style-mixture adaptation in this paper. A graphic description of style-mixture adaptation is presented in Fig. 1.

IV. K-SMA Model

In this paper, we propose a new method called K-SMA to deal with the style-mixture adaptation problem. Before the introduction of K-SMA, we first give a trivial method to deal with the style-mixture adaptation problem.

A. Direct Adaptation (DA): A Trivial Method

In style-mixture adaptation, due to the lacking of the style information of $Data_T$, adapting the base classifier C to each style k exactly is impossible. A direct method is that we can ignore the different styles in $Data_T$ and tread it as a single-style set. Then we can directly adapt the base classifier C to the whole set of $Data_T$. When the number of the styles in $Data_T$ is few or the styles in $Data_T$ are similar to each other, this method can really work. But in more complicated scenarios, this method is not a good choice.

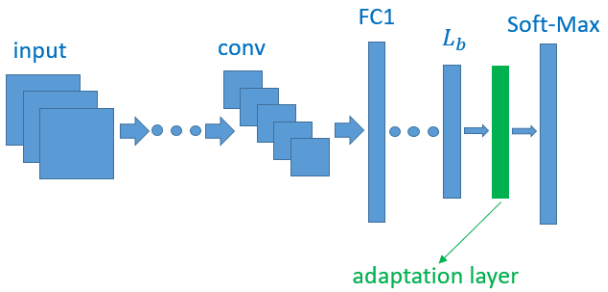

Fig. 2. The Structure of CNN for Adaptation

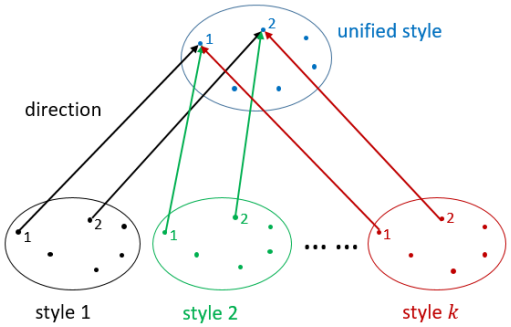

Fig. 3. The Projection Direction for Different Styles

B. K-Style Mixture Adaptation (K-SMA)

The basic assumption in K-SMA is that there are totally K styles in $Data_T$, and we want to partition the style-mixture set $Data_T$ into different subsets (style clustering) according to their style consistency. Then the base classifier C can be adapted to each subset respectively for writer adaptation. The style clustering is very important in K-SMA, if the clustering performance is poor, the writer adaptation performance can not be guaranteed. For better clustering, the features used for clustering must be carefully designed. In summary, the K-SMA includes three main parts: the design of the feature for style clustering, the clustering method, and the adaptation method to each cluster.

1) The Design of the Style Feature: The goal of style clustering is to group the style-mixture set $Data_T$ into multiple clusters, and the samples in each cluster have the same style. To realize this target, the feature used for clustering must represent the style information of the sample, we call this kind of feature as *style feature*. Style clustering is a task very similar to writer identification, therefore, the style feature can be extracted from a writer identification model (e.g. CNN model). In this way, two models (a character classification model and a writer identification model) are required. However, in this paper, we consider style clustering and writer adaptation simultaneously, and only one base classification model is used, the style feature is calculated directly from the classification model.

Fristly, we train a CNN model on $Data_S$ as the base classification model, then we insert an adaptation layer to the trained CNN for adaptation. This structure can be seen in Fig. 2. Let L_b denotes the layer before the adaptation layer, and let ϕ denotes the mapping from the input of the CNN to the output of layer L_b.

First, by feeding forward $Data_S$ to the trained CNN, we can get its new representation at layer L_b, i.e.

$$Data_S^\star = \{(\phi(x_i), y_i)_{i=1}^{N_S}\} \tag{4}$$

Based on $Data_S^\star$, we can compute the sample mean on L_b

$$\mu_y = \frac{1}{N_y} \sum_{y_k=y} \phi(x_k) \tag{5}$$

for each class y in $Data_S$ and form the set

$$U = \{(\mu_y)_{y=1}^{N_C}\} \tag{6}$$

which is called *unified style space*. After that, by feeding forward the unlabeled $Data_T$ to the trained CNN, we can get its new feature at layer L_b and the predicted class label by the trained CNN, we denote this as

$$Data_T^\star = \{(\phi(x_i), \hat{y}_i)_{i=1}^{N_T}\} \tag{7}$$

Given $(\phi(x_i), \hat{y}_i) \in Data_T^\star$, its corresponding *style feature* is defined as

$$d_i = \frac{\mu_{\hat{y}_i} - \phi(x_i)}{\|\mu_{\hat{y}_i} - \phi(x_i)\|_2} \tag{8}$$

which is the projection direction from $\phi(x_i)$ to the corresponding $\mu_{\hat{y}_i}$ (predicted class mean) in the unified style space.

Obviously, both $\phi(x_i)$ and $\mu_{\hat{y}_i}$ include the class information of $\hat{y}_i$, then by the subtraction $\mu_{\hat{y}_i} - \phi(x_i)$, the class information in d_i can be eliminated. For $x_1, x_2 \in Data_T$ with predicted class label $\hat{y}_1, \hat{y}_2$, if they belong to the same style and $\hat{y}_1 = \hat{y}_2$, then $\phi(x_1) \approx \phi(x_2)$, $\mu_{\hat{y}_1} = \mu_{\hat{y}_2}$, thus $d_1 \approx d_2$; if $\hat{y}_1 \neq \hat{y}_2$, because x_1, x_2 belong to the same style, then the relative location of $\phi(x_1), \phi(x_2)$ w.r.t. $\mu_{\hat{y}_1}, \mu_{\hat{y}_2}$ is similar, thus $\phi(x_1) - \phi(x_2) \approx \mu_{\hat{y}_1} - \mu_{\hat{y}_2}$ and further $d_1 \approx d_2$. If x_1, x_2 belong to different styles and $\hat{y}_1 = \hat{y}_2$, then $\mu_{\hat{y}_1} = \mu_{\hat{y}_2}$, but $\phi(x_1) \neq \phi(x_2)$ because the variability of different styles, thus $d_1 \neq d_2$; further, if $y_1 \neq y_2$, of course we can get $d_1 \neq d_2$. This process can be seen in Fig. 3. From the analysis before, we can see that when samples come from the same style, the corresponding style features d_i are similar no matter the labels of the samples are the same or not, however, when the samples come from different styles, the corresponding style features d_i are usually different. Thus the designed style feature can really present the style information of the samples. Meanwhile, it can also eliminate the class information of the samples. Therefore, the designed style feature d_i is appropriate and effective for style clustering.

The motivation of the above analysis is based on the assumption that each style cluster is an affine transformation of the unified style space (which is reasonable for handwriting recognition), and therefore, as shown in Fig. 3, in each cluster, the directions of each sample pointing to its class-mean in the unified style space are consistent.

2) Style Clustering: With (8), we can get the style feature of the data in $Data_T$, resulting in a new set as:

$$\widehat{Data}_T = \{(d_i)_{i=1}^{N_T}\} \tag{9}$$

which is calculated based on the class means on L_b and the predicted class labels from the pretrained CNN model. After that, we can group the set $\widehat{Data}_T$ into different styles by the K-means algorithm [13]. The initialization of the cluster centers is implemented by randomly choosing some data points from $\widehat{Data}_T$. The number of clusters K is a hyper-parameter which is equal to the number of styles. By using the K-means clustering, we can divide the style-mixture set $\widehat{Data}_T$ into multiple subsets according to the style consistency. Then we can adapt the base classifier C to each subset for better generalization performance.

3) Adaptation to Each Style Cluster: The base CNN classifier is trained on the large dataset $Data_S$. For adaptation to a specific cluster k, we first insert an adaptation layer to the trained CNN, then by using the specific cluster's data

$$data_{ck} = \{x_i\}_{i=1}^{N_{ck}} \tag{10}$$

we can train the adaptation layer for adapting to the cluster k. The training of the adaptation layer is based on the STM model [8], [14]. The objective function is

$$\min_{A\in\mathbf{R}^{d\times d}, b\in\mathbf{R}^d} \sum_{i=1}^{N_{ck}} f_i \|As_i+b-t_i\|_2^2+\beta\|A-I\|_F^2+\gamma\|b\|_2^2 \tag{11}$$

and N_{ck} is the number of samples in cluster k, $s_i = \phi(x_i)$ and $t_i = \mu_{\hat{y}_i}$ are the source and target points respectively, $f_i \in [0,1]$ is the transformation confidence for pair (s_i, t_i) which is given by the CNN model, i.e.,

$$f_i = max\{SoftMax_{CNN}(x_i)\}. \tag{12}$$

The $\|.\|_F$ is the matrix Frobenius norm and $\|.\|_2$ is the vector L_2 norm. This optimization problem is convex, thus it has a closed-form solution [1]. After solving this optimization problem, we can directly set its solution to the parameters of the adaptation layer. Note that there are totally K style clusters, and therefore, the basic STM adaptation will be implemented K times.

4) Whole Algorithm: Unsupervised Style-Mixture Adaptation: The problem considered in this paper is unsupervised style-mixture adaptation. In other words, we do not have the class labels and the style indexes. Both the style feature d_i in (8) and the target point t_i in (11) are based on the predicted class labels of the CNN. The initial prediction (without adaptation) is usually unreliable, however, with the style clustering and cluster-specific adaptation, the accuracy of label prediction will be improved, and then with the improved labels, we can re-calculate d_i and t_i and re-implement the style clustering and cluster-specific adaptation. This process is repeated iteratively to boost the whole performance, and the output of this method is the finally predicted class labels for $Data_T$. Lastly, we summarize the whole K-SMA model in Algorithm 1.

Algorithm 1 K-SMA Algorithm

Require:
labeled training data $Data_S$
unlabeled style-mixture test data $Data_T$
the trained base CNN classifier C
number of styles K

1: feed forward the $Data_S$ to C, compute sample-mean for each class on $Data_S$ by (5)
2: adapt C to the whole $Data_T$ with STM, get the prediction $\{\hat{y}_i\}_{i=1}^{N_T}$ of $Data_T$
3: feed forward the $Data_T$ to C, use the prediction $\{\hat{y}_i\}_{i=1}^{N_T}$, compute the style feature of $Data_T$ by (8)
4: cluster $\widehat{Data}_T$ into K subsets using K-means
5: adapt C to each subset respectively using STM and get new predictions $\{\hat{y}_i\}_{i=1}^{N_T}$
6: repeat steps 3-5 multiple times
7: **return** final predicted class labels $\{\hat{y}_i\}_{i=1}^{N_T}$ for $Data_T$

V. Experiments

A. Database

We use the CASIA-OLHWDB1.0-1.2 [15] as the training dataset, and we use this dataset to train the base CNN classifier. The test data are the ICDAR-2013 online competition dataset [16], which includes the data come from 60 writers. The writers in the training dataset and test dataset are disjoint from each other. The number of classes in the test dataset is 3755, hence we ignore the samples which are out of the 3755-class in the training dataset.

B. The Base Convolutional Neural Network

We train a CNN on the training dataset as the base classifier. To obtain a high accuracy, we adopt the same structure and training method as [8]. It includes 8 convolutional layers and 3 fully-connected layers. The activation function of the CNN is leaky-ReLU [17], besides, the dropout [18] method is adopted to prevent the CNN from over-fitting. Meanwhile, we use the same data pre-processing and training method as [8]. The accuracy of the trained CNN on the ICDAR-2013 online competition dataset is 97.55%.

C. The Setting of the Experiments

To build the style-mixture data, we randomly choose several different writers and mix their data together. We test the DA (Section IV-A) and K-SMA (Section IV-B) methods on the style-mixture data respectively. Meanwhile, we also give the style-clear adaptation results on the same data. In the style-clear adaptation, we know exactly which sample belongs to which style (writer), and then we can conduct writer adaptation multiple times with accurate style partition, and this is an ideal (best) situation.

In building the style-mixture data, different number of styles (writers), i.e., from two to four, are considered in our experiments. In each case, we conduct the experiments for ten times in which the style-mixture data are built from different combination of writers. We adopt different number of iterations in K-SMA for different experiments, and the average number of iteration is two. We randomly run the K-SMA algorithm for five times in each experiment and use the average as the final result. In the experiments of K-SMA, We also give

TABLE I
THE RESULTS OF MIXING TWO STYLES (%)

writers	before adaptation	style-clear adaptation	direct adaptation	K-SMA(2)
16,19	97.92	98.24	98.02	98.12
9,31	97.20	97.55	97.44	97.50
20,21	95.58	96.35	96.02	96.27
34,57	90.59	92.13	91.76	92.13
15,47	95.81	96.56	96.20	96.41
28,40	97.82	98.22	97.96	98.10
21,59	99.24	99.40	99.32	99.34
34,35	90.40	91.82	91.28	91.69
11,50	96.38	96.85	96.72	96.71
20,43	95.46	96.19	95.87	96.18
average	95.64	96.33	96.06	96.25
error reduction rate (%)		15.83	9.63	13.99

TABLE II
THE RESULTS OF MIXING THREE STYLES (%)

writers	before adaptation	style-clear adaptation	direct adaptation	K-SMA(2)	K-SMA(3)
1,3,12	97.24	97.67	97.35	97.41	97.45
2,31,34	93.11	94.18	93.73	93.82	93.96
9,14,50	95.50	96.07	95.69	95.75	95.76
10,55,60	98.13	98.47	98.34	98.36	98.39
44,46,47	96.30	96.78	96.45	96.49	96.59
18,20,57	96.80	97.37	97.07	97.11	97.16
20,49,59	96.52	97.09	96.79	96.86	96.93
21,27,34	93.52	94.50	94.02	94.29	94.15
51,54,55	98.56	98.92	98.73	98.81	98.79
34,43,58	93.45	94.42	93.97	94.02	94.12
average	95.91	96.55	96.21	96.29	96.33
error reduction rate (%)		15.65	7.33	9.29	10.27

the results under different setting of K (the number of style clusters).

D. Experimental Results

The results of our experiments are shown in table I, II and III. The number in the parenthesis after K-SMA in the table represents the number K used in K-SMA model. The error reduction rate appeared in three tables is an measurement used to measure the effectiveness of the adaptation method which is defined as

$$error\ reduction\ rate = \frac{error(before) - error(adapted)}{error(before)}. \tag{13}$$

E. Analysis of the Results

From the results of the three tables, we can see that both the DA and K-SMA methods are useful to improve the performance of the base classifier. After adaptation, the error rates are really decreased compared with the results before adaptation. The K-SMA outperforms the DA in all the three settings, and its results are closer to the style-clear adaptation, which is the most ideal scenario. This demonstrates the efficiency of the proposed K-SMA method for style-mixture adaptation.

When the results of DA being very closed to the results of style-clear adaptation, the K-SMA can only boost the performance by a small margin compared with DA (sometimes even worse). This can be seen in the writer combinations of $\{21,59\}$, $\{11,50\}$ in table I, $\{10,55,60\}$, $\{51,54,55\}$ in table II, $\{5,13,38,59\}$, $\{22,27,44,48\}$ in table III. This phenomenon is normal, the style-clear adaptation is the most ideal case and it can be viewed as the upper bound of K-SMA. Therefore, it is hard for K-SMA to further boost the performance when DA itself is already closed to style-clear adaptation. In these situations, the styles in $Data_T$ are usually similar to each other, then treat them as a unified (single) style and use DA for adaptation is a better choice.

However, when the gap between the results of DA and style-clear adaptation is large, K-SMA will consistently boost the performance, this can be seen in the style combination $\{34,57\}$, $\{20,43\}$ in table I, $\{2,31,34\}$, $\{21,27,34\}$ in table II, $\{2,12,20,56\}$, $\{1,7,21,49\}$ in table III. This indicates that the proposed K-SMA is more suitable for complicated style mixture scenarios.

F. Effect of the Style Number

From tables I to III, we can see that when the number of mixing styles increases, the promotion caused by DA will decrease gradually. This indicates that when the style number in $Data_T$ is large, treating it as a single style set and directly adapt to it is not suitable. Similarly, when there are more styles in $Data_T$, the improvement of K-SMA compared with DA will also decrease. This is because when the number of styles increases, it will become much more difficult to obtain an accurate style clustering results.

G. Effect of the Cluster Number in K-SMA

When the style number is small, the K-means algorithm can well divide the data into different styles, thus setting the cluster number equal to the real number of writers will be better. This can be seen in table II, K-SMA(3) outperforms K-SMA(2). However, When the style number is large, the unsupervised K-means algorithm is not capable to divide the data correctly, meanwhile, some styles in the data may be similar to each other, thus setting the cluster number smaller than the real number of writers may be better. This can be seen in table III, K-SMA(3) works better than K-SMA(4).

VI. CONCLUSION

This paper deals with a new problem called style-mixture adaptation, which is more general than style-clear adaptation. A new model called K-SMA is proposed to solve this problem. In K-SMA, a style feature representing the styles and eliminating the influence form class categories is used for style clustering, after that in each style cluster, the traditional writer adaptation method is adopted to improve the prediction. The

TABLE III
THE RESULTS OF MIXING FOUR STYLES (%)

writers	before adaptation	style-clear adaptation	direct adaptation	K-SMA(2)	K-SMA(3)	K-SMA(4)
5,13,38,59	98.69	98.86	98.77	98.81	98.78	98.81
24,34,46,57	94.59	95.42	95.01	95.06	95.08	95.13
1,16,30,58	97.81	98.12	97.80	97.85	97.90	97.88
26,30,34,43	94.62	95.42	94.89	95.00	95.00	94.96
2,12,20,56	96.98	97.50	97.11	97.19	97.25	97.24
22,27,44,48	97.95	98.13	98.07	98.06	98.07	98.07
1,21,45,50	97.51	97.86	97.60	97.63	97.68	97.67
16,19,25,35	98.32	98.50	98.32	98.37	98.40	98.39
1,7,21,49	97.90	98.33	97.97	98.06	98.08	98.06
24,25,53,55	98.33	98.57	98.39	98.43	98.44	98.43
average	97.27	97.67	97.39	97.45	97.47	97.46
error reduction rate (%)		14.65	4.40	6.59	7.33	6.96

process of style clustering and cluster-specific adaptation is repeated iteratively in an unsupervised manner. Experiments demonstrate that our model is effective, which can further improve the performance of the base classifier in the style-mixture setting. This framework is general and can be applied to other style-mixture adaptation problems, not limited to handwriting recognition.

The style feature used in K-SMA is simply calculated based on an initial classification model, which may be not powerful enough in distinguishing different styles. Therefore, a future work is to design better and more robust style features. Moreover, in the specific application of handwriting recognition, we usually have some prior knowledge, i.e., the characters in the same text line usually have the same style. This indicates that the style clustering should be conducted in a hierarchical manner, i.e., on the text lines or paragraphs, not directly on the characters. These improvements will be considered in our future work.

ACKNOWLEDGMENT

This work has been supported by the National Natural Science Foundation of China (NSFC) grants 61573355 and 61633021.

REFERENCES

[1] Xu-Yao Zhang and Cheng-Lin Liu. Writer adaptation with style transfer mapping. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(7):1773–1787, 2013.

[2] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.

[3] Bohan Li, Liangrui Peng, and Jingning Ji. Historical chinese character recognition method based on style transfer mapping. In *2014 11th IAPR International Workshop on Document Analysis Systems (DAS)*, pages 96–100. IEEE, 2014.

[4] Jixiong Feng, Liangrui Peng, and Franck Lebourgeois. Gaussian process style transfer mapping for historical chinese character recognition. In *SPIE/IS&T Electronic Imaging*, pages 94020D–94020D. International Society for Optics and Photonics, 2015.

[5] Jose A Rodriguez, Florent Perronnin, Gemma Sánchez, and Josep Lladós. Unsupervised writer style adaptation for handwritten word spotting. In *2008 19th International Conference on Pattern Recognition (ICPR)*, pages 1–4. IEEE, 2008.

[6] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.

[7] Yejun Tang, Liangrui Peng, Qian Xu, Yanwei Wang, and Akio Furuhata. Cnn based transfer learning for historical chinese character recognition. In *2016 12th IAPR Workshop on Document Analysis Systems (DAS)*, pages 25–29. IEEE, 2016.

[8] Xu-Yao Zhang, Yoshua Bengio, and Cheng-Lin Liu. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition*, 61:348–360, 2017.

[9] Jun Du, Jian-Fang Zhai, Jin-Shui Hu, Bo Zhu, Si Wei, and Li-Rong Dai. Writer adaptive feature extraction based on convolutional neural networks for online handwritten chinese character recognition. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 841–845. IEEE, 2015.

[10] Mingsheng Long, Yue Cao, Jianmin Wang, and Michael I Jordan. Learning transferable features with deep adaptation networks. In *2015 32th International Conference on Machine Learning (ICML)*, pages 97–105, 2015.

[11] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. Knowledge adaptation: Teaching to adapt. *arXiv preprint arXiv:1702.02052*, 2017.

[12] Muhammad Ghifary, W Bastiaan Kleijn, Mengjie Zhang, David Balduzzi, and Wen Li. Deep reconstruction-classification networks for unsupervised domain adaptation. In *2016 14th European Conference on Computer Vision (ECCV)*, pages 597–613. Springer, 2016.

[13] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[14] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, Zhenbo Luo, and Cheng-Lin Liu. Unsupervised adaptation of neural networks for chinese handwriting recognition. In *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pages 512–517. IEEE, 2016.

[15] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Casia online and offline chinese handwriting databases. In *2011 11th International Conference on Document Analysis and Recognition (ICDAR)*, pages 37–41. IEEE, 2011.

[16] Fei Yin, Qiu-Feng Wang, Xu-Yao Zhang, and Cheng-Lin Liu. Icdar 2013 chinese handwriting recognition competition. In *2013 12th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1464–1470. IEEE, 2013.

[17] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *2013 30th International Conference on Machine Learning (ICML)*, volume 30, 2013.

[18] Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958, 2014.