2019 IEEE Intl Conf on Parallel & Distributed Processing with Applications, Big Data & Cloud Computing, Sustainable Computing & Communications, Social Computing & Networking (ISPA/BDCloud/SocialCom/SustainCom)

Reducing Style Overfitting for Character Recognition via Parallel Neural Networks with Style to Content Connection

Wei Tang[1,2,3], Yiwen Jiang[1,2,3], Neng Gao[3], Ji Xiang[3], Jiahui Shen[3], Xiang Li[1,2,3], Yijun Su[1,2,3]
[1]*State Key Laboratory of Information Security, Chinese Academy of Sciences, Beijing, China*
[2]*School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China*
[3]*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China*
{tangwei, jiangyiwen} @iie.ac.cn

***Abstract*—There is a significant style overfitting problem in neural-based character recognition: insufficient generalization ability to recognize characters with unseen styles. To address this problem, we propose a novel framework named Style-Melt Nets (SMN), which disentangles the style and content factors to extract pure content feature. In this framework, a pair of parallel style net and content net is designed to respectively infer the style labels and content labels of input character images, and the style feature produced by the style net is fed to the content net for eliminating the style influence on content feature. In addition, the marginal distribution of character pixels is considered as an important structure indicator for enhancing the content representations. Furthermore, to increase the style diversity of training data, an efficient data augmentation approach for changing the thickness of the strokes and generating outline characters is presented. Extensive experimental results demonstrate the benefit of our methods, and the proposed SMN is able to achieve the state-of-the-art performance on multiple real world character sets.**

***Index Terms*—character recognition, style overfitting, neural network**

I. Introduction

Characters have been widely used in our daily lives. There are probably tens of thousands of different characters with multifarious styles, and most of them can be well recognized by most people. But in the field of artificial intelligence, character recognition is considered as an extremely difficult task due to the very large number of categories, complicated structures, similarity between characters and the variability of styles. Because of its unique technical challenges and great social needs, there are intensive research in this field and a rapid increase of successful techniques, especially the end to end neural-based models such as Convolutional Neural Network (CNN), which is specifically designed to deal with the variability of 2D shapes of characters in documents [10]. Furthermore, this neural network has played an important role to improve recent image recognition studies [8], [14], [17].

However, the variability of styles still challenges character recognition due to the significant style overfitting problem: lack of generalization ability to recognize characters with unseen styles. For example, the trained models always perform poorly when test on the character sets with the styles that have never appeared in training data (see Sec. III-D for details). Therefore, in traditional applications, the training set needs to contain as many styles as possible to enhance the generalization ability of the recognizer, but this naive method is absolutely costly [8] and is contrary to the intention of building a human-level learning model to learn rich concepts from limited data [9].

To our best knowledge, few works are directly devoted to reducing style overfitting of character recognition. This is not surprising as the intractable challenge posed by style and content entanglement with such a large number of styles and content categories. The first study to disentangle two factors is using a bilinear model [19]. This bilinear model has been wildly used in zero-shot learning to associate visual representation and auxiliary class text description [1], [3], [21]. Recent study [26] successfully aplay this bilinear model to separate the content and style representations of characters. Furthermore, an intercross pair-wise optimization [16] have been proposed to extract relatively pure style representation by considering content to style feed. It is imperative take into account style to content feed for the purpose of extracting pure content representation to overcome style overfitting.

In this paper, we propose a novel framework named Style-Melt Nets (SMN), which disentangle the content and style factors of characters to extract pure content feature via style to content feed. More specifically, we take advantage of the superior capabilities of deep neural network for feature representation to build a style net and a content net, which learn the style and content representations respectively. To automatically disentangle the content feature from the style information, we first pre-train the style net supervised by the style label, and then feed the style feature to the content net besides the character images to extract relatively pure content feature.

Utilizing the domain knowledge of character structure is an effective method to reduce overfitting. It is based on an empirical observation: if two characters contain the same content but different styles, their structures are commonly invariant. Previous methods [11], [13], [16] usually cast the domain knowledge as the one-hot embedding. But such embedding methods require a lot of human resources for annotation.

978-1-7281-4328-6/19/$31.00 ©2019 IEEE
DOI 10.1109/ISPA-BDCloud-SustainCom-SocialCom48970.2019.00117

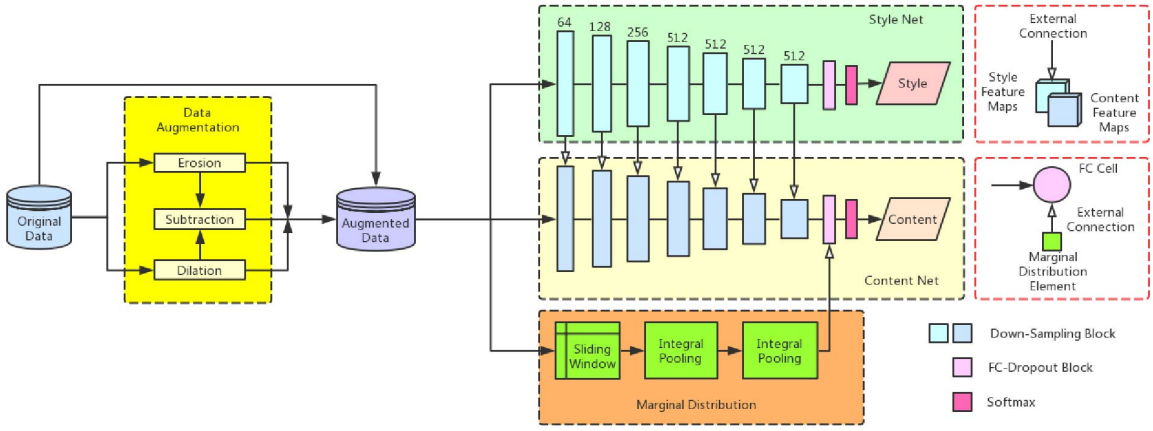

Fig. 1: The detailed architecture of the proposed framework of SMN. It consists of four components, including a style net $\mathcal{S}$, a content net $\mathcal{C}$, a data augmentation module $\mathcal{A}$ and a marginal distribution extractor $\mathcal{M}$. $\mathcal{S}$ and $\mathcal{C}$ extract the latent style feature and content feature respectively. The style feature extracted by $\mathcal{S}$ is fed to $\mathcal{C}$ for eliminating the style influence on content feature. The data augmentation $\mathcal{A}$ is used to increase the style diversity of training data. And the marginal distribution of character pixels $\mathcal{M}$ contain the domain knowledge of character structure.

We note that the marginal distribution of character pixels is an important structure indicator, and design a pair of narrow windows sliding along the axis of input image to extract these marginal distributions, which does not require any manual annotating. In addition, we propose integral pooling to make the marginal distribution more smooth for tolerating the variability of style.

Another useful method to reduce overfitting is data augmentation, which produces a large amount of generated sample to increase the diversity of training set. Traditional data augmentation methods for character recognition usually transform the input image into high dimensional space to generate deformed image by rotation and distortion [2], [8], [10]. And with the popularity of the deep generative models such as Variational Auto-Encoder [7] and Generative Adversarial Networks [4], generating more readable character images with specific styles has become feasible. But the neural based character generation or style transfer models [16], [26] require a large-scale training set for working desirable, which against our intention to learn rich concepts from limited data. Other methods for character generation [5], [9], [25] rely on online writing information, which are hardly applied to offline situation. We find that the thickness of strokes is one of the main manifestation of the diversity of styles, and the outline is one of the most hardly recognition character. Therefore, we present a simple but efficient data augmentation approach based on Erosion and Dilation to change the thickness of strokes and transfer the input character into outline style.

We conduct extensive experiments on 3 popular character sets: printed character, handwritten character, and wild character. Experimental results demonstrate the positive influence of our methods on reducing style overfitting and the proposed SMN is able to achieve state-of-the-art performance on both of the limited training sets and the relatively lager training sets. The main contributions of our study are summarized as follows:

- We propose Style-Melt Nets (SMN), which adopt parallel neural networks to disentangle the content and style features of character via style to content connection.
- We introduce the marginal distribution of character pixels as structure indicator, which contain informative domain knowledge with out any manual annotating.
- We present an efficient approach to control the thickness of strokes and transfer the input character into outline to increase the style diversity of training data.

II. Methodology

In this section, we first formulate the problem. And then we discuss the proposed framework SMN in detail by successively introducing 4 main modules: the Data Augmentation, the Marginal Distribution, the Style Net and the Content Net.

A. Problem Formulation

In order to formulate the problem, we first make the basic assumption that all images of characters are decided jointly by the content feature and style feature, which can be denoted as:

$$I^{(i,j)} \leftarrow (\mathbf{c}^{(i,j)}, \mathbf{s}^{(i,j)}) \tag{1}$$

where $I^{(i,j)}$ is the character with content i and style j, $\mathbf{c}^{(i,j)}$ represents the latent content feature of $I^{(i,j)}$, and $\mathbf{s}^{(i,j)}$ represents latent style feature of $I^{(i,j)}$.

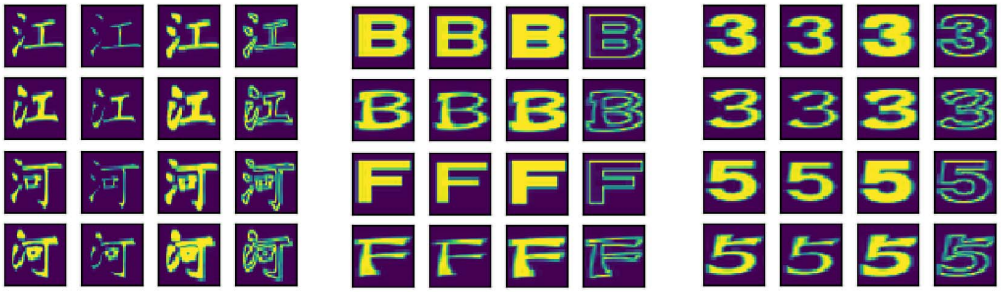

Fig. 2: Examples of the generated characters produced by the proposed data augmentation based on erosion and dilation. The left, center, left part of this figure respectively present several results of Chinese characters, English letters, and Arabic numerals. For each character set, the 1th, 2th, 3th and 4th columns respectively list the original, the lightface, the boldface and the outline fonts.

We use C and S to represent the content label and style label. Then the training data $\mathcal{D}_{tr}$ and the test data $\mathcal{D}_{te}$ could be present as:

$$\mathcal{D}_{tr} = \{I^{(i,j)}, C^{(i)}, S^{(j)}\}, \tag{2}$$

$$\mathcal{D}_{te} = \{I^{(i,k)}, C^{(i)}, S^{(k)}\} \tag{3}$$

where $i = 1, 2, ..., n$ and n is the total number of contents in training data and in test data. The difference between training data and test data is $j = 1, 2, ..., m$ and $k = m+1, m+2, ..., m+l$, in which m and l respectively represent the total number of styles in training data and test data.

The content net $\mathcal{C}$ and style net $\mathcal{S}$ aim to respectively obtain the latent features $\mathbf{c}^{(i,j)}$ and $\mathbf{s}^{(i,j)}$ of the input character $I^{(i,j)}$ in $\mathcal{D}_{tr}$. The purpose of the proposed framework is to accurately infer the content label when observing the characters with unseen style in $\mathcal{D}_{te}$.

B. Model Architecture

In this section, we present the details of the proposed SMN. The main architecture of this framework consists of a style net $\mathcal{S}$ and a content net $\mathcal{C}$. Furthermore, the data augmentation $\mathcal{A}$ and marginal distribution $\mathcal{M}$ are integrated into this framework as auxiliary components. The whole framework is illustrated in Figure 1. First, the data augmentation module generate the augmented training data given $\mathcal{D}_{tr}$. Next, the style net is pre-trained to extract the latent style feature given character images and their style labels. Then, the content net is trained given character images and content labels, along with their marginal distributions and the style feature maps produced by the pre-trained style net. Finally, the trained model is test on $\mathcal{D}_{te}$.

a) Data Augmentation.: In order to improve the generalization ability of the trained model, we use data augmentation to increase the style diversity of training data. Extensive experimental results show that some special factors have great influence on the style overfitting problem, for example the outline fonts and the thickness of character strokes. Therefore, we design a simple but effective approaches to change the thickness of strokes and generate outline characters. The input is a single character image, and the output is a series of transformed characters with the same content label and extra style labels. This data augmentation method is a specialized application of the morphology operations of erosion and dilation. It could be denoted as:

$$\mathcal{A} : \mathcal{D}_{tr}^{'} = g(\mathcal{D}_{tr}), \tag{4}$$

where $\mathcal{D}_{tr}^{'}$ denotes the augmented training data, $\mathcal{D}_{tr}$ is the original training data.

Figure 2 shows some examples of the generated characters. The left part of this figure show the result of Chinese characters, the center part are English litters, and the right part are Arabic numerals. For each part of this figure, the first column present the original characters; the 2th column are the lightface characters produced by erosion with kernel 2×2; the 3th column are the boldface characters produced by dilation 2×2; the 4th column show the outline characters produced by the subtracting of boldface and lightface. Proved by the experimental results, the proposed data augmentation method significantly increase the generalization ability of the trained model by increasing the style diversity of training data (see Sec. III-E for details).

b) Marginal Distribution.: The marginal distribution of character pixels proposed here is aim to capture the similar structure of the same character content with different styles. We designed as a pair of long narrow rectangle windows to obtain the marginal distribution. These windows sliding along the vertical or horizontal axis of the input images. The length of these windows are predetermined to be the height or width of the input image. They accumulates the pixel values in it, and produce a vector that represent the marginal distribution.

Notic that an excessively thin window has insufficient resilience to tolerate the deformation and scaling of a stroke, and an excessively fat window could be confused by too many strokes in it. To overcome this problem, more than one window with different size and stride could be designed to obtain diverse marginal distribution vector. More effectively, we use only one pair of window with its width to be 1 and stride to be 1, and then propose integral pooling to process the vectors produced by these minimal windows to achieve the same

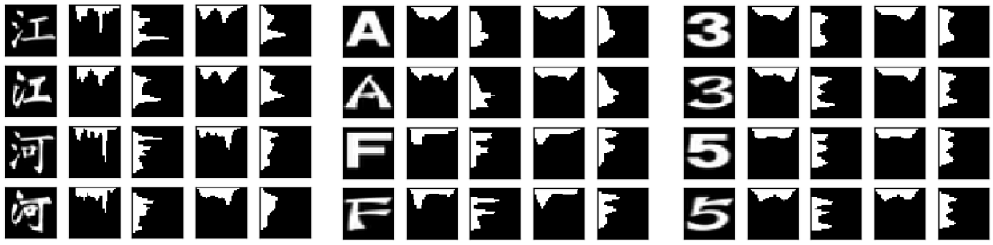

Fig. 3: Examples of the marginal distributions of character pixels. For each character group in this figure, the 1th column lists the original characters; the 2th and 3th columns successively list the the vertical and horizontal marginal distributions; the 4th and 5th columns respectively list the vertical and horizontal marginal distributions after double integral pooling operations.

effect as using a lot of sliding windows with different width and stride, which avoid tedious pixel-level operations. Integral pooling cuts a vector into a lot of chunks with special size and stride and then calculates the sum of all the elements in each chunk to produce a new vector. The Marginal Distribution module could be denoted as:

$$\mathcal{M} : \mathbf{m} = h(I), \tag{5}$$

where I denotes the input character image, $\mathbf{m}$ is the marginal distribution.

Figure 3 shows some examples of the extracted marginal distributions. For each group of characters, original characters are presented in the 1th column. The 2th and 2th columns respectively list the normalized vertical and horizontal marginal distributions without any integral pooling. The 4th and 5th columns respectively list the vertical and horizontal marginal distributions after twice integral pooling operations with their size and stride to be 3 and 1. It could be seen that the marginal distributions are obviously smoothed by integral pooling, and the style variability toleration is enhanced by this smooth. Experimental results demonstrate that the marginal distribution has a great positive influence on the proposed SMN (see Sec. III-F for details).

c) Style Net.: Style net is pre-trained to extract latent style features of input character images. This neural network consist of a series of Convolution-BatchNorm-LeakyReLU down-sampling blocks and a FC-Softmax classifier which yield the inferred style labels of input images. The style net could be formulated as:

$$\mathcal{S} : \mathbf{s}^{(i,j)} = p_\theta(I^{(i,j)}), \tag{6}$$

where θ denotes the parameters of network, $I^{(i,j)}$ is the input character image, and $\mathbf{s}^{(i,j)}$ is the latent styel feature of $I^{(i,j)}$.

Given a set of training examples $\mathcal{D}'_{tr}$, the training objective of $\mathcal{S}$ is defined as:

$$\theta = \arg\min_{\theta} \sum_{I \in \mathcal{D}'_{tr}} \mathcal{L}_s(I, S), \tag{7}$$

$$\mathcal{L}_s(I, S) = -S \log p_\theta(I), \tag{8}$$

where I denotes the characters image in training set, and $S^{(j)}$ is the style labels of input images. Driven by this loss function, the inferred style label produced by softmax layer will tend to be close to the ground truth. When test on the character sets with unseen styles, the style net will yield the inferred style label indicating which seen style is more similar with the unseen style.

d) Content Net.: The architecture of content net is similar to the style net. The main difference between style net and content net is that the style net is pre-trained by only feeding the character images and style labels, while the content net is fed by not only the character images and content labels but also the style features produced by the style net and the marginal distribution of input character images. The content net could be formulated as:

$$\mathcal{C} : \mathbf{c}^{(i,j)} = q_\phi(I^{(i,j)}, \mathbf{s}^{(i,j)}, \mathbf{m}^{(i,j)}), \tag{9}$$

where ϕ denotes the parameters of the network, $I^{(i,j)}$ denotes the input character image, $\mathbf{s}^{(i,j)}$ is the style feature produced by the style net, $\mathbf{m}^{(i,j)}$ is the marginal distribution of the input image, and $\mathbf{c}^{(i,j)}$ is the content feature extracted by content net.

Given the training data $\mathcal{D}'_{tr}$, the training objective of $\mathcal{C}$ is defined as:

$$\phi = \arg\min_{\phi} \sum_{I \in \mathcal{D}'_{tr}} \mathcal{L}_c(I, C, \mathbf{s}, \mathbf{m}), \tag{10}$$

$$\mathcal{L}_c(I, C, \mathbf{s}, \mathbf{m}) = -C \log q_\phi(I, \mathbf{s}, \mathbf{m}), \tag{11}$$

where I denotes character image, C is the content labels of input image, $\mathbf{s}$ is the style feature produced by style net, and $\mathbf{m}$ is the marginal distribution of input image.

The way to feed style feature and marginal distribution to content net is using External Connections, which inspired by Skip-Connection that has been commonly used in semantic segmentation task [12]. They are presented as the hollow arrows in Figure 1. External Connection enables the style feature maps produced by each convolution layers of style net to be transmitted to the corresponding layers in content net as the superposed input, and it also feeds the marginal distribution vectors to the full connection layer as an additional regularization. The external style feature maps and marginal distribution vectors does not affect the convergence of the content net, because the pre-trained style net and the marginal distribution extractor have no parameters to learn during the

Algorithm 1 Training Algorithm

Input: Characters dataset $\mathcal{D}_{tr} = \{I^{(i,j)}, C^{(i)}, S^{(j)}\}$
Output: The model parameters: θ, ϕ

$\mathcal{D}_{tr}^{'} \Leftarrow$ Get the augmented data of $\mathcal{D}_{tr}$ using $\mathcal{A}$
Randomly initialize θ, ϕ
$\theta \Leftarrow$ Pre-train $\mathcal{S}$ using $\nabla_\theta \mathcal{L}_s(I, S; \theta)$
repeat
    $(I, C) \Leftarrow$ Randomly select a mini-batch from $\mathcal{D}_{tr}^{'}$
    $\mathbf{s} \Leftarrow$ Get the style feature of I using $\mathcal{S}$
    $\mathbf{m} \Leftarrow$ Get the marginal distribution of I using $\mathcal{M}$
    $\phi \Leftarrow$ Train $\mathcal{C}$ using $\nabla_\phi \mathcal{L}_c(I, C, \mathbf{s}, \mathbf{m}; \phi)$
until Stable

training process of the content net. The details of the training procedure is shown in Algorithm 1. Leveraging the external style information and the structure indicator, the content net is able to produce relatively pure content feature.

III. Experiments

In this section, we first introduce the data set we used and then empirically demonstrate the existing of style overfitting. Next, we show the benefit of the proposed data augmentation and marginal distribution. Finally, we evaluate the performence of the proposed SMN compared with several baselines.

A. Data Set

After a long period of development and research, character recognition technology is currently divided into three different tasks according to the application scenarios, from easy to hard including: printed character recognition, handwritten character recognition, and wild character recognition. Therefore, we evaluate SMN on 3 real world character sets. The details of these 3 real world data sets are presented as follow:

a) ***Printed Characters****:* In the task of printed character recognition, following previous work [18], [27], we build a dataset that contains 800 font styles and each style consist of 3755 frequently used Chinese characters, 146 Japanese kanas, 52 English letters, and 9 Arabic numerals. All of these character sets are extracted from True Type Font (TTF) files collected on internet, which jointly launched by Apple Inc. and Microsoft Corp. and wildly used in printed document. We randomly select 80% and 10% of styles for training and test, and leave the rest part of styles for validation.

b) ***Handwritten Characters****:* In the task of handwritten character recognition, we evaluate all baseline models and our SMN model on the most popular handwritten benchmark dataset ICDAR 2013 [22]. Note that ICDAR'13 dose not provide explicit writing style annotations. Therefore, to train SMN with style supervision, we build a training set form 956 scanned documents written by 463 individuals (denoted as 463 different writing styles). We randomly select a part of the built training set for training and all trained models are tested on ICDAR'13. Contents in ICDAR'13 but not in the built training set dose not calculate the test error.

c) ***Wild Characters****:* In the task of reading character in the wild, we evaluate SMN and all baseline models on the large-scale public dataset CTW 2019 [23]. CTW 2019 is a very large dataset of Chinese character in street view images, which contain about 1 million Chinese characters annotated by experts in over 30 thousand street view images. For each character in the dataset, the annotation includes its underlying character, its bounding box, and 6 attributes. The attributes indicate whether it has complex background, whether it is raised, whether it is handwritten or printed, etc. We combine all the 6 attributes as 64 different styles to provide style supervision. Consider the complex background noise in the wild and the diversity of character orientations and rotations, the 64 supervision styles based on 6 annotated attribute hardly to cover all style concepts, but It does not prevent us from introducing incomplete style supervision to validate our model. In our experiments, we consider the recognition task as a classification problem of 1001 categories.Since some characters have very few samples in dataset and also have relatively rare usage in practice, we only consider recognition of the top 1000 frequent observed character categories. Besides the used 1000 character categories, a category of 'others' is added. We randomly select a part of styles for training, and leave the rest part of them for test.

B. Baseline Models

We compare our models with several state-of-the-art baseline models on character recognition. The description of these baselines are listed below:

a) ***CMP****:* [27] adopt multi-polling layer to capture multi-scale features of printed character.

b) ***CSK****:* [18] leverage a side skeleton channel to facilitate printed character recognition.

c) ***NCDD****:* [24] investigate domain-specific knowledge of normalization-cooperated direction-decomposed feature map to boost the performance of deep neural network.

d) ***FCN-S****:* [20] leverage an effective fully convolutional network to extract stroke skeletons for handwritten characters to assist handwritten character recognition task.

e) ***VGG****:* [15] is one of the most influential convolutional network because it reinforced the notion that deep architacture is better then shallow.

f) ***ResNet****:* [6] is the well-known deep neural network performs well in many computer vision tasks.

C. Implement Details

In the proposed model of SMN, the output channels of convolutional layers are 1, 2, 4, 8, 8, 8, 8 times of 64 respectively. The first convolution layer is with 5×5 kernel and stride 1 and the rest are with 3×3 kernel and stride 1, and all ReLUs are leaky with slope 0.1. The size of FC layer is 1024. We augment the input training data twice respectively using 2×2 kernel and 4×4 kernel. All the morphology operations are provided by OpenCV. And we use double integral pooling operations with the same size and stride to be 3 and 1 to process the marginal distribution vectors.

In training process, we use random values drawn from the Gaussian distribution ($\mu = 0$ and $\sigma^2 = 0.01$) to initialize all trainable weights, and all bias are initialized at 0. Adam is used as the optimization algorithm and the mini-batch size is 128, and the learning rate is set to be $1e^{-4}$. In printed, handwritten, and wild cases, the total number of epoches are 100, 200, and 1000, respectively. After each epoch, we shuffle the training data to make different mini-batches. The size of input image is 128×128. For noise avoiding, all the experiments are repeated 10 times with random training set selection to calculate the average top-1 accuracy for content recognition.

D. Existing of Style Overfitting

People learning new concepts can often generalize successfully from just a single example, yet machine learning algorithms typically require tens or hundreds of examples to perform with similar accuracy, and people can also use learned concepts in richer ways than neural-based algorithms [9]. Due to this situation, here we first demonstrate the existing of style overfitting by using a single candidate set for training.

Data	te-1	te-2	te-3	te-4	te-5
tr-1	**0.9635**	0.1313	0.0765	0.1305	0.0983
tr-2	0.1385	**0.9646**	0.0914	0.0823	0.0842
tr-3	0.0818	0.0959	**0.9659**	0.1236	0.0823
tr-4	0.1172	0.0879	0.1188	**0.9553**	0.1172
tr-5	0.0996	0.0972	0.0852	0.1103	**0.9553**

TABLE I: Results of VGG-11 on printed Chinese characters.

Data	te-1	te-2	te-3	te-4	te-5
tr-1	**0.9747**	0.1686	0.0941	0.1305	0.1073
tr-2	0.1606	**0.9826**	0.1047	0.1007	0.1081
tr-3	0.0885	0.1002	**0.9726**	0.1260	0.0927
tr-4	0.1390	0.0884	0.1292	**0.9614**	0.1191
tr-5	0.1113	0.0925	0.0895	0.1263	**0.9630**

TABLE II: Results of VGG-19 on printed Chinese characters.

Data	te-1	te-2	te-3	te-4	te-5
tr-1	**0.9835**	0.1694	0.1060	0.1593	0.1145
tr-2	0.1648	**0.9887**	0.1372	0.1052	0.1012
tr-3	0.1009	0.1262	**0.9851**	0.1531	0.1006
tr-4	0.1492	0.1075	0.1574	**0.9715**	0.1209
tr-5	0.1246	0.0903	0.0953	0.1321	**0.9793**

TABLE III: Results of CSK on printed Chinese characters.

Considering that Chinese character is more difficult to recognize than other characters due to its larger dictionary and complicated structures, here we randomly select several candidate sets of Chinese character with different font styles for cross-validation. The training sets and test sets are denoted as "tr-style label" and "te-style label", for example "tr-1" and "te-1". We evaluate the content recognition top-1 accuracy of popular CNN model such as VGG Nets [14] on these candidate sets. Table I and Table II respectively show the experimental results of VGG-11 (contain 8 convolution layers) and VGG-19 (contain 16 convolution layers). It could be seen that the train accuracy are very high (highlighted with bold font), but the test accuracy on other character sets with unseen font styles are extremely low. And even adding more amount of convolution layers, the style overfitting problem still exists.

In addition to the general classification models, we also test the classification model specialized for character recognition on the same way. Table III show the experimental results of the state-of-the-art Chinese character recognition method CSK [18], unfortunately, it is also challenged by style overfitting.

Furthermore, we test more extensively with the relatively large training sets on other character sets such as handwritten characters and wild characters. It could be see that the style overfitting is more serious in these cases.

E. Effect of Data Augmentation

Extensive experimental results strongly point out that some morphological factors of character have great impact on style overfitting. For example, a model trained on lightface fonts is hardly generalized to perform well on boldface fonts, and vice versa. More significantly, unseen outline fonts are more difficult to recognize than ordinary unseen fonts. Table IV presents the results of VGG-19 test on the representative character sets. In this table, the first column and row respectively lists the training sets and test sets. It could be seen that the trained models achieve relatively high performance when the test set possess the similar stroke thickness of the training set, and the performances are extremely poor when they test on the character set with unseen outline style.

Data	江	江	**江**	**江**	江
江	**0.9862**	0.1752	0.1260	0.1221	0.0753
江	0.1716	**0.9875**	0.1301	0.1259	0.0796
江	0.1251	0.1259	**0.9834**	0.1693	0.0812
江	0.1213	0.1248	0.1695	**0.9816**	0.0823
江	0.0743	0.0781	0.0815	0.0827	**0.9793**

TABLE IV: Results of Fast-HCCR on the representative fonts

Data	江	江	**江**	**江**	江
江	**0.9771**	0.3919	0.3761	0.3812	0.2731
江	0.3896	**0.9785**	0.3861	0.3694	0.2826
江	0.3852	0.3760	**0.9753**	0.3639	0.2903
江	0.3791	0.3710	0.3681	**0.9735**	0.2890
江	0.0804	0.0851	0.0862	0.0886	**0.9721**

TABLE V: Results of SMN on the representative fonts

Due to this situation, the proposed data augmentation approach aim to transfer the input character image to lightface and boldface, furthermore, generate the outline font. Here we augment the input training data twice respectively using 2×2 kernel and 4×4 kernel. Table V show the experimental results of VGG-19 which integrate data augmentation to increase the style diversity of training data. It could be seen that, compared to the previous results of VGG-19 without data augmentation, although the training accuracy is slightly reduced, the testing accuracy is significantly improved.

F. Influence of Marginal Distribution

To verify the effectiveness of the proposed marginal distribution, we compare the converge curves of VGG-19 training on printed Chinese characters. Figure 4 shows the training curves of VGG-19 integrating marginal distribution or not, where the y-axis represents the training loss and the x-axis represents the number of epochs during training. It could be seen that the style net integrated marginal distribution provide higher converge speed and lower loss. It demonstrates the marginal distribution improve the performance of the character recognition by providing domain knowledge of the structure similarity within same character.

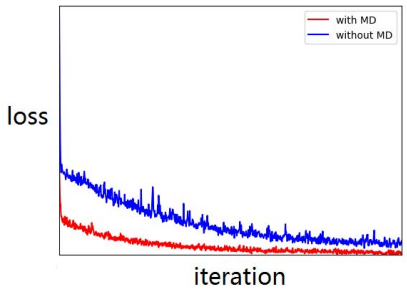

Fig. 4: Training curves of VGG-19 on printed Chinese characters.

G. Comparison with Baseline Methods

We compare the proposed SMN with several stat-of-the-art baseline models on multiple character sets. Table VI presents the exprimental results. Besides, we also report the results on relatively small training set in Figure 5. These experiments on small training data test the ability of baseline models and our SMN to learn rich concepts from limited data, which directly represent the generalization ability of the trained models. Specifically, we randomly choose $N\%$ candidate styles for training, where $N = 5, 10, 15, 20, 25, 30$. Note that in wild character recognition, the data augmentation and marginal attribution is abandoned due to the large nosie level in wild images, and the experiments on limited training data is also abandoned as the style annotation is insufficient in wild character dataset.

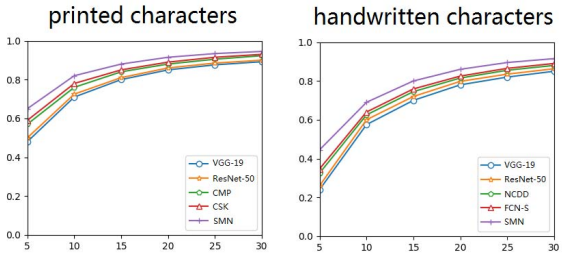

Fig. 5: Results on partial training data.

Methods	Printed	Handwritten	Wild
CMP [27]	0.9778	–	–
CSK [18]	0.9795	–	–
NCDD [24]	–	0.9621	–
FCN-S [20]	–	0.9636	–
VGG-19 [15]	0.9657	0.9352	0.6172
ResNet-50 [6]	0.9759	0.9497	0.7043
SMN (our)	**0.9836**	**0.9775**	**0.7956**

TABLE VI: Results on full training data.

When training on full training data, our SMN outperforms all of the baseline methods in all kind of character recognition tasks. In the case of printed character recognition, SMN outperform the best baseline of CSK with 0.41% improvement. In the case of handwritten character recognition, SMN outperform the best baseline of FAN-S with 1.39% improvement. Note that there are few public models designed for wild character recognition [23]. Therefore, in the case of wild character recognition, we use the popular convolutional neural networks such as VGG-19 and ResNet-50 as baseline. Our SMN outperform these popular baselines with more than 9% improvement. These results empirically prove the reasonable and effectiveness of the proposed SMN.

When training on partial training data, our SMN also outperform the all baseline models with the impressive improvement on printed characters and handwritten characters. In printed character recognition, we find all models suffer from performance degradation when the training styles is shrinking. When the training styles reducing from 30% to 5%, VGG-19 and ResNet-50 lose more than 40% accuracy, CMP and CSK lose more than 20% accuracy, and our SMN is more robust than baselines. The similar results is found in the task of handwritten character recognition. These results explicitly prove the existence of style overffiting, and also demonstrate that our SMN obtain relatively strong robustness in the case of insufficient training styles.

IV. Conclution

Character recognition is challenged by style overffiting problem due to that characters do not have an explicit separation between style and content. To address this issue, we propose a novel framework of Style-Melt Nets (SMN), which consisting of a pair of parallel content net and style net to separately learn the representations of the content features and the style features, and we take into account the style to content feed to melt the style influence within content features. Moreover, an effective data augmentation approach and an useful structure indicator are presented as the auxiliary components to improve the generalization ability of character recognition models.

References

[1] S. Changpinyo, W. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 5327–5336, 2016.

[2] D. C. Ciresan, U. Meier, and J. Schmidhuber. Multi-column deep neural networks for image classification. In *2012 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2012, Providence, RI, USA, June 16-21, 2012*, pages 3642–3649, 2012.

[3] A. Frome, G. S. Corrado, J. Shlens, S. Bengio, J. Dean, M. Ranzato, and T. Mikolov. Devise: A deep visual-semantic embedding model. In *Annual Conference on Neural Information Processing Systems 2013, NIPS 2013. December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 2121–2129, 2013.

[4] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. C. Courville, and Y. Bengio. Generative adversarial nets. In *Annual Conference on Neural Information Processing Systems 2014, NIPS 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 2672–2680, 2014.

[5] A. Graves. Generating sequences with recurrent neural networks. *CoRR*, abs/1308.0850, 2013.

[6] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition, ICCV 2016*, pages 770–778, 2016.

[7] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *CoRR*, abs/1312.6114, 2013.

[8] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Annual Conference on Neural Information Processing Systems 2012, NIPS 2012.*, pages 1106–1114, 2012.

[9] B. M. Lake, R. Salakhutdinov, and J. B. Tenenbaum. Human-level concept learning through probabilistic program induction. *Science*, 350(6266):1332–1338, 2015.

[10] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, Nov 1998.

[11] C. Li, J. Zhu, and B. Zhang. Max-margin deep generative models for (semi-)supervised learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(11):2762–2775, 2018.

[12] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 3431–3440, 2015.

[13] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

[14] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[15] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations, ICLR 2015, Conference Track Proceedings*, 2015.

[16] D. Sun, T. Ren, C. Li, H. Su, and J. Zhu. Learning to write stylized chinese characters by reading a handful of examples. In *International Joint Conference on Artificial Intelligence, IJCAI 2018, July 13-19, 2018, Stockholm, Sweden.*, pages 920–927, 2018.

[17] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015*, pages 1–9, 2015.

[18] W. Tang, Y. Su, X. Li, D. Zha, W. Jiang, N. Gao, and J. Xiang. Cnn-based chinese character recognition with skeleton feature. In *International Conference on Neural Information Processing, ICONIP 2018, Part V*, pages 461–472, 2018.

[19] J. B. Tenenbaum and W. T. Freeman. Separating style and content. In *Annual Conference on Neural Information Processing Systems 1996, NIPS 1996, Denver, CO, USA, December 2-5, 1996*, pages 662–668, 1996.

[20] T. Wang and C. Liu. Fully convolutional network based skeletonization for handwritten chinese characters. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, AAAI-18*, pages 2540–2547, 2018.

[21] Y. Xian, Z. Akata, G. Sharma, Q. N. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 69–77, 2016.

[22] F. Yin, Q. Wang, X. Zhang, and C. Liu. ICDAR 2013 chinese handwriting recognition competition. In *12th International Conference on Document Analysis and Recognition, ICDAR 2013, Washington, DC, USA, August 25-28, 2013*, pages 1464–1470, 2013.

[23] T. Yuan, Z. Zhu, K. Xu, C. Li, T. Mu, and S. Hu. A large chinese text dataset in the wild. *J. Comput. Sci. Technol.*, 34(3):509–521, 2019.

[24] X. Zhang, Y. Bengio, and C. Liu. Online and offline handwritten chinese character recognition: A comprehensive study and new benchmark. *Pattern Recognition*, 61:348–360, 2017.

[25] X. Zhang, F. Yin, Y. Zhang, C. Liu, and Y. Bengio. Drawing and recognizing chinese characters with recurrent neural network. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(4):849–862, 2018.

[26] Y. Zhang, Y. Zhang, and W. Cai. Separating style and content for generalized style transfer. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 8447–8455, 2018.

[27] Z. Zhong, L. Jin, and Z. Feng. Multi-font printed chinese character recognition using multi-pooling convolutional neural network. In *International Conference on Document Analysis and Recognition, ICDAR 2015*, pages 96–100, 2015.