Separating Style and Content for Generalized Style Transfer

Yexun Zhang
Shanghai Jiao Tong University
zhyxun@sjtu.edu.cn

Ya Zhang✉
Shanghai Jiao Tong University
ya_zhang@sjtu.edu.cn

Wenbin Cai
Microsoft
wenbca@microsoft.com

Abstract

Neural style transfer has drawn broad attention in recent years. However, most existing methods aim to explicitly model the transformation between different styles, and the learned model is thus not generalizable to new styles. We here attempt to separate the representations for styles and contents, and propose a generalized style transfer network consisting of style encoder, content encoder, mixer and decoder. The style encoder and content encoder are used to extract the style and content factors from the style reference images and content reference images, respectively. The mixer employs a bilinear model to integrate the above two factors and finally feeds it into a decoder to generate images with target style and content. To separate the style features and content features, we leverage the conditional dependence of styles and contents given an image. During training, the encoder network learns to extract styles and contents from two sets of reference images in limited size, one with shared style and the other with shared content. This learning framework allows simultaneous style transfer among multiple styles and can be deemed as a special 'multi-task' learning scenario. The encoders are expected to capture the underlying features for different styles and contents which is generalizable to new styles and contents. For validation, we applied the proposed algorithm to the Chinese Typeface transfer problem. Extensive experiment results on character generation have demonstrated the effectiveness and robustness of our method.

1. Introduction

In recent years, style transfer, as an interesting application of deep neural networks (DNNs), has increasingly attracted attention among the research community. Existing studies either apply an iterative optimization mechanism [8] or directly learn a feed-forward generator network to force the output image to be with target style and target contents [12, 23]. A set of losses are accordingly proposed for the transfer network, such as the pix-wise loss [10], the perceptual loss [12, 27], and the histogram loss [25]. Recently, several variations of generative adversarial networks (GANs) [14, 28] are introduced by adding a discriminator to the style transfer network which incorporates adversarial loss with transfer loss to generate better images. However, these studies aim to explicitly learn the transformation from a certain source style to a given target style, and the learned model is thus not generalizable to new styles, i.e. retraining is needed for transformations of new styles which is time-consuming.

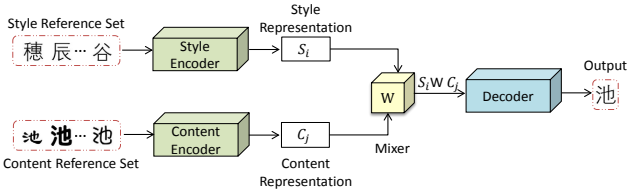

Figure 1. The framework of the proposed *EMD* model.

In this paper, we propose a novel generalized style transfer network which can extend well to new styles or contents. Different from existing supervised style transfer methods, where an individual transfer network is built for each pair of style transfer, the proposed network represents each style or content with a small set of reference images and attempts to learn separate representations for styles and contents. Then, to generate an image of a given style-content combination is simply to mix the corresponding two representations. This learning framework allows simultaneous style transfer among multiple styles and can be deemed as a special 'multi-task' learning scenario. Through separated style and content representations, the network is able to generate images of all style-content combination given the corresponding reference sets, and is therefore expected to generalize well to new styles and contents. To our best knowledge, the study most resembles to ours is the bilinear model proposed by Tenenbaum and Freeman [22], which obtained independent style and content representations through matrix decomposition. However, it usually requires an exhaustive enumeration of examples for accurate decomposition of new styles and contents, which may not be readily available for some styles/contents.

As shown in Figure 1, the proposed style transfer network, denoted as *EMD* thereafter, consists of a style en-

Table 1. Comparison of *EMD* with existing methods.

Methods	Data format	Generalizable to new styles?	Requirements for new style transfer	What the model learned?
Pix2pix [10]	paired	The learned model can only transfer images to styles which appeared in the training set. For new styles, the model has to be retrained.	Retrain on a lot of training images for a source style and a target style.	The translation from a certain source style to a specific target style.
CoGAN [14]	unpaired			
CycleGAN [28]	unpaired			
Rewrite [1]	paired			
Zi-to-zi [2]	paired			
AEGN [16]	paired			
Perceptual [12]	unpaired		Retrain on many input content images and one style image.	Transformation among specific styles.
StyleBank [5]	unpaired			
Patch-based [6]	unpaired	The learned model can be generalized to new styles.	One or a small set of style/content reference images.	The swap of style/content feature maps.
AdaIn [9]	unpaired			The transferring of feature statistics.
EMD	triplet			The feature representation of style/content.

coder, a content encoder, a mixer, and a decoder. Given a set of reference images, the style/content encoder leverages the conditional dependence of styles and contents to learn style/content representations. The mixer then combines the corresponding style and content representations using a bilinear model. The decoder finally generates the target images based on the combined representations. Each training example for the proposed network is provided as a triplet $<\mathcal{R}_{S_i}, \mathcal{R}_{C_j}, I_{ij}>$, where I_{ij} is the target image of style S_i and content C_j. $\mathcal{R}_{S_i}$ and $\mathcal{R}_{C_j}$ are respectively the style and content reference sets, each consisting of r random images of the corresponding style S_i and content C_j. The entire network is trained end-to-end with a weighted $L1$ loss measuring the difference between the generated images and the target images. As it is difficult to validate the decomposition of style and content for images, we here use the character typeface transfer as a special case of style transfer to validate the proposed method. Extensive experiment results have demonstrated the effectiveness and robustness of our method for style transfer. The main contributions of our study are summarized as follows.

- We propose a generalized style transfer network which is able to generate images of any unseen style/content given a small set of reference images.
- The network decomposes an image into separate style and content representations, taking advantages of the conditional dependence of contents and styles.
- This learning framework allows simultaneous style transfer among multiple styles and can be deemed as a special 'multi-task' learning scenario.

2. Related Work

Neural Style Transfer. DeepDream [17] may be the first attempt to generate artistic work using Convolution Neural Networks (CNNs). Then Gatys et. al successfully applied CNNs to neural style transfer [8]. They generate the target image by optimizing a noise image iteratively using a pretrained network, which is time-consuming. Therefore, many studies have been done for finding a way to directly learn a feed-forward generator network. Johnson et. al proposed a perceptual loss function to help neural style transfer [12]. Ulyanov et. al proposed a texture network for both texture synthesis and style transfer [23]. Further, Chen et. al proposed the stylebank to represent each style by a convolution filter, which can simultaneously learn numerous styles [5]. For arbitrary neural style transfer, [6] proposed a patch-based method to replace each content feature patch with the nearest style feature. Further, [9] proposed a faster method based on adaptive instance normalization which performed style transfer in the feature space by transferring feature statistics.

Image-to-Image Translation. Image-to-image translation is to learn the mapping from the input image to output image, such as from edges to real objects. Pix2pix [10] used a conditional GAN based network which needs paired data for training. However, paired data are hard to collect in many applications. Therefore, some methods with no need for paired data are proposed. Liu and Tuzel proposed the coupled GAN (CoGAN) [14] for learning a joint distribution of two domains by a weight sharing way. Later, Liu [13] extended the CoGAN to unsupervised image-to-image translation problem. Some other studies [3, 20, 21] encourage the input and output to share certain content even though they may differ in style by enforcing the output to be close to the input in a predefined metric space, such as class label space and so on. Recently, Zhu et. al proposed the cycle-consistent adversarial network (CycleGAN) [28] which performs well for many vision and graphics tasks.

Character Style Transfer. Most existing studies take character style transfer as an image translation task. The "Rewrite" project uses a simple traditional flavour top-down CNNs structure and can transfer a typographic font to another stylized typographic font [1]. As the improvement version, the "zi-to-zi" project can transfer multiple styles by assigning each style an one-hot category label and training the network by a supervised way [2]. The recent work "From A to Z" also adopts a supervised method and assigns each character an one-hot label [24]. Lyu et. al proposed an auto-encoder guided GAN network (AEGN) which can synthesize calligraphy images with specified style from standard Chinese font images [16].

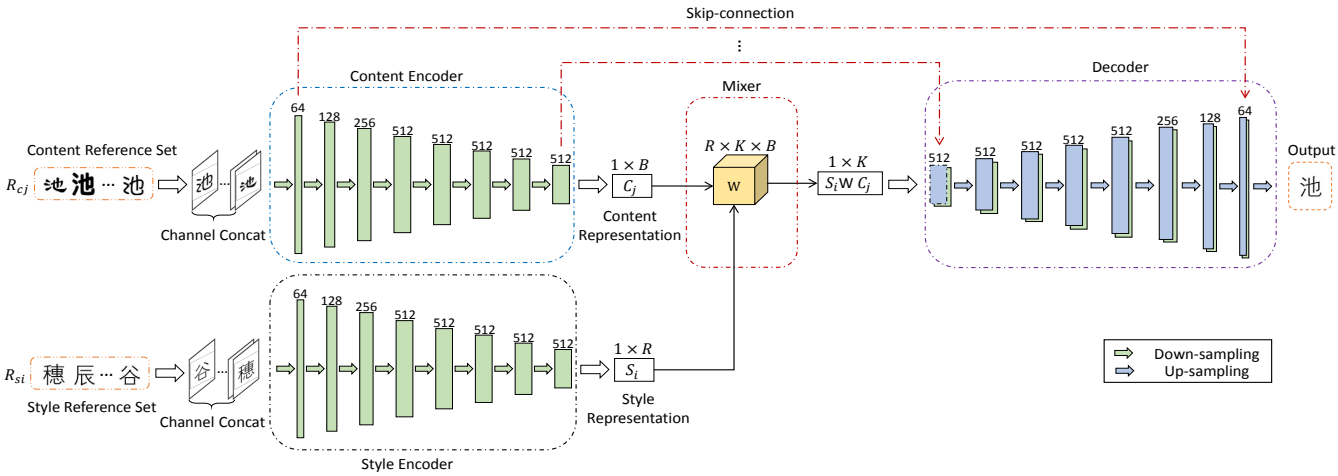

Figure 2. The detailed architecture of the proposed generalized *EMD* model for style transfer.

However, most of the methods reviewed above can only transfer styles in the training set and the network must be retrained for new styles. In contrast, the proposed *EMD* network can generate images with novel styles/contents given only a small set of reference images. We present a comparison of the methods in Table 1.

3. Generalized Style Transfer Model

In this section, we present the details of the proposed generalized style transfer model *EMD*. The whole model is an encoder-decoder network which consists of four subnets: *Style Encoder*, *Content Encoder*, *Mixer* and *Decoder*, as shown in Figure 2. First, the *Style/Content Encoder* extracts style/content representations given style/content reference images. Next, the *Mixer* integrates the style feature and content feature and the combined feature is then fed into the *Decoder*. Finally, the *Decoder* generates the image with the target style and content.

3.1. Encoder Network

To achieve the generation of images with arbitrary style and content, it is crucial to separate the style and content explicitly. The *Style Encoder* and *Content Encoder* are designed for this purpose. They both have the same architecture, consisting of a series of Convolution-BatchNorm-LeakyReLU down-sampling blocks which yield 1×1 latent feature representations of the input style/content reference images. The first convolution layer is with 5×5 kernel and stride 1 and the rest are with 3×3 kernel and stride 2. All ReLUs are leaky, with slope 0.2.

The input to the *Style Encoder* and *Content Encoder* are style reference set $\mathcal{R}_{S_i}$ and content reference set $\mathcal{R}_{C_j}$, respectively. $\mathcal{R}_{S_i}$ consists of r reference images with the same style S_i but different contents $C_{j_1}, C_{j_2}, \ldots, C_{j_r}$

$$\mathcal{R}_{S_i} = \{I_{ij_1}, I_{ij_2}, \ldots, I_{ij_r}\}, \tag{1}$$

where I_{ij} represents the image with style S_i and content C_j. Similarly, $\mathcal{R}_{C_j}$ is for content C_j $(j = 1, 2, \ldots, m)$ and consists of r reference images with the same content C_j but different styles $S_{i_1}, S_{i_2}, \ldots, S_{i_r}$

$$\mathcal{R}_{C_j} = \{I_{i_1 j}, I_{i_2 j}, \ldots, I_{i_r j}\}. \tag{2}$$

The r reference images are concatenated in the channel dimension to feed in to the encoders. This allows the encoders to capture the common characteristics among images of the same style/content.

3.2. Mixer Network

With the style representations and content representations obtained by the *Style Encoder* and *Content Encoder*, we combine the two factors by the *Mixer* which is a bilinear model. The bilinear models are two-factor models with the mathematical property of separability: their outputs are linear in either factor when the other is held constant, which has been demonstrated that the influences of two factors can be efficiently separated and combined in a flexible representation that can be naturally generalized to unfamiliar factor classes [22], such as new styles. Furthermore, the bilinear model has also been successfully used in zero-shot learning as a compatibility function to associate visual representation and auxiliary class text description [4, 7, 26]. The learned compatibility function can be seen as the shared knowledge and transferred to new classes. Here, we take the bilinear model to integrate styles and contents together and the combination function can be formulated as

$$F_{ij} = S_i \mathbf{W} C_j, \tag{3}$$

where $\mathbf{W}$ is a tensor with size $R \times K \times B$, S_i is the R-dimensional style feature and C_j is the B-dimensional content feature. F_{ij} can be seen as the K-dimensional feature vector of image I_{ij} which will be taken as the input of the *Decoder* to generate the image with style S_i and content C_j.

3.3. Decoder Network

The image generator is a typical decoder network which is symmetrical to the encoder and maps the combined feature representation to the output image with target style and content. The *Decoder* roughly follows the architectural guidelines set forth by Radford et. al [18] and consists of a series of Deconvolution-BatchNorm-ReLU up-sampling blocks except the last layer which only contains the deconvolution layer. Other than the last layer which uses 5×5 kernels and stride 1, all deconvolution layers use 3×3 kernels and stride 2. The outputs are transformed into [0,1] by the sigmoid function.

In addition, since the stride convolution in *Style Encoder* and *Content Encoder* is detrimental to spatial information extraction, we adopt the skip-connection which has been commonly used in semantic segmentation tasks [11, 15, 19] to refine the segmentation using spatial information from different resolutions. Here, based on the fact that though the content inputs and outputs differ in appearances, they share the same structure, we concatenate the input feature map of each up-sampling block with the corresponding output of the symmetrical down-sampling block in *Content Encoder* to allow the *Decoder* to learn back the relevant structure information lost during the down-sampling process.

3.4. Loss Function

Given a set of training examples $\mathcal{D}_t$, the training objective is defined as

$$\theta = \arg\min_{\theta} \sum_{I_{ij} \in \mathcal{D}_t} L(\hat{I}_{ij}, I_{ij} | \mathcal{R}_{S_i}, \mathcal{R}_{C_j}), \tag{4}$$

where θ represents model parameters, $\hat{I}_{ij}$ is the generated image and $L(\hat{I}_{ij}, I_{ij} | \mathcal{R}_{S_i}, \mathcal{R}_{C_j})$ is the generation loss which can be written as

$$L(\hat{I}_{ij}, I_{ij} | \mathcal{R}_{S_i}, \mathcal{R}_{C_j}) = W_{st}^{ij} \times W_b^{ij} \times ||\hat{I}_{ij} - I_{ij}||. \tag{5}$$

We use pixel-wise L1 loss as our generation loss for character typeface transfer problem rather than L2 loss since L1 loss tends to yield sharper and cleaner images [10, 16].

W_{st}^{ij} and W_b^{ij} are two weights for target image I_{ij} which are added to alleviate the imbalance in the target set induced by the random sampling. In each learning iteration, the size and thickness of target images in the target set may vary greatly and the model will be optimized mainly for target images containing characters which have more pixels and cause more losses, such as those big and thick characters. Moreover, models trained using L1 loss may pay more attention to blacker characters and perform poorly on images with lighter characters. To alleviate these imbalance, we add these two weights on the generation loss: W_{st}^{ij} about the size and thickness of characters, and W_b^{ij} about the darkness of characters.

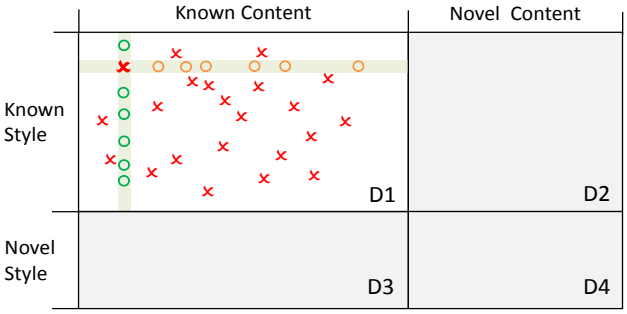

Figure 3. The illustration of data set partition, target images selection and reference set construction (best viewed in color).

As for W_{st}^{ij}, we first calculate the number of black pixels, i.e. pixels covered by characters. Then W_{st}^{ij} is defined as the reciprocal of the number of black pixels in each target image

$$W_{st}^{ij} = 1/N_b^{ij}, \tag{6}$$

where N_b^{ij} is the number of black pixels of target image I_{ij}.

As for W_b^{ij}, we calculate the mean value of black pixels for each target image and set a softmax weight

$$W_b^{ij} = \frac{exp(\text{mean}_{ij})}{\sum_{I_{ij} \in \mathcal{D}_t} exp(\text{mean}_{ij})}, \tag{7}$$

where mean_{ij} is the mean value of the black pixels of the target image I_{ij}.

4. Experiments

In this section, we evaluate the proposed network for Chinese Typeface transfer problem. We first introduce the data set we used followed by the implementation details. Finally, we present our experimental results.

4.1. Data Set

To evaluate the proposed *EMD* model with Chinese Typeface transfer tasks, we construct a data set which contains 832 fonts (styles) and each font has 1732 commonly used Chinese characters (contents). All images are 80×80 pixels. We randomly select 75% of the styles and contents as known styles and contents (i.e. 624 train styles and 1299 train contents) and leave the rest 25% as novel styles and contents (i.e. 208 novel styles and 433 novel contents). The entire data set is therefor partitioned into four subsets as shown in Figure 3: D_1, images with known styles and contents namely train styles and contents, D_2, images with known styles but novel contents, D_3, images with known contents but novel styles, and D_4, images with both novel styles and novel contents. The four data sets represent different levels of style transfer challenges.

4.2. Implementation Details

In our experiment, the output channels of convolution layers in the *Style Encoder* and *Content Encoder* are 1, 2,

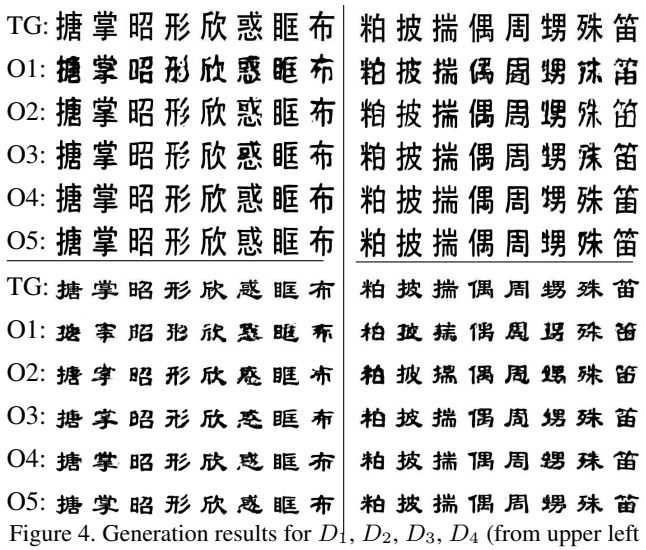

Figure 4. Generation results for D_1, D_2, D_3, D_4 (from upper left to lower right) with different training set size. TG: Target image, O1: Output for N_t=20k, O2: Output for N_t=50k, O3: Output for N_t=100k, O4: Output for N_t=300k, O5: Output for N_t=500k. In all cases, r=10.

4, 8, 8, 8, 8, 8 times of C respectively, where C=64. And for the *Mixer*, we set R=B=K in our implementation. The output channels of the first seven deconvolution layers in *Decoder* are 8, 8, 8, 8, 4, 2, 1 times of C respectively. We set the initial learning rate as 0.0002 and train the model end-to-end with the Adam optimization method until the output is stable.

In each experiment, we first randomly sample N_t target images with known content and known styles as training examples. We then construct the two reference sets for each target image by randomly sampling r images of the corresponding style/content. Figure 3 provides an illustration of target images selection and reference set construction. Each row represents one style and each column represents a content. The target images are represented by randomly scattered red "x" marks. The reference images for the target image are selected from corresponding style/content, shown as the orange circles for the style reference images and green circles for content reference images. When testing, taking D4 as an example, each target image in D4 can be generated with r style/content reference images. The style reference images can be randomly sampled from images with target style in D3 and the content reference images are randomly sampled from images with target content in D2.

4.3. Experimental Results

In this section, we present the experimental results. First, we analyze the influence of some factors influencing the model performance. Then, we validate the separation of style and content. Finally, we compare the proposed method with some baseline networks to prove the effectiveness of our method.

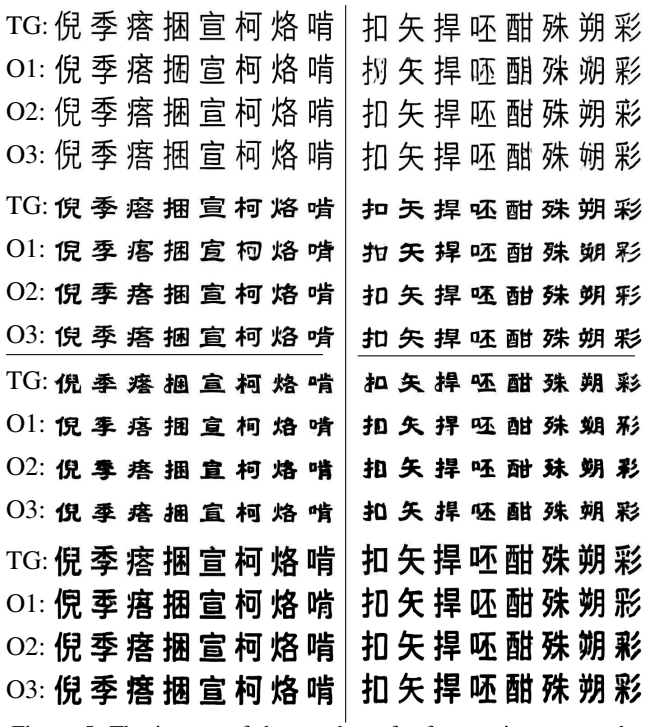

Figure 5. The impact of the number of reference images on the generation of images in D_1, D_2, D_3, D_4, respectively (from upper left to lower right). TG: Target image, O1: Output for r=5, O2: Output for r=10, O3: Output for r=15. In all cases, N_t=300k.

4.3.1 Influence of the Training Set Size

To evaluate the influence of the training set size on style transfer, we conduct experiments for N_t=20k, 50k, 100k, 300k and 500k. The generation results for D_1, D_2, D_3 and D_4 are shown in Figure 4. As we can see, the larger the training set, the better the performance, which is consistent with our intuition. The generated images with N_t=300k and 500k are clearly better than images generated with N_t=20k, 50k and 100k. Besides, the performance of N_t=300k and N_t=500k is close which implies that with more training images, the network performance tends to be saturated and N_t=300k is enough for good results. Therefore, we take N_t=300k for the following experiments.

4.3.2 Influence of the Reference Set Size

In addition, we conduct experiments with different number of reference images. Figure 5 displays the image generation results of N_t=300k with r=5, r=10 and r=15 respectively. From the figure, we can observe that with more reference images, characters are generated better in details. Besides, characters generated with r=5 are overall okay, meaning that our model can generalize to novel styles using only a few reference images. The generation results of r=10 and r=15 are close, therefore we take r=10 in our other experiments. Intuitively, more reference images will support

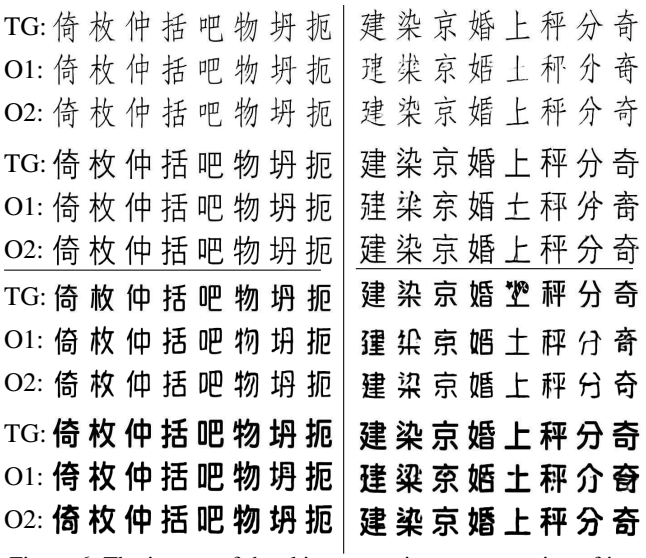

Figure 6. The impact of the skip-connection on generation of images in D_1, D_2, D_3, D_4, respectively (from upper left to lower right). TG is the target image, O1 and O2 are outputs of models without and with skip-connection. In all cases N_t=300k, r=10.

more information about strokes and styles of characters and the common points in the reference sets will be more obvious. Therefore, given $r > 1$, our model can achieve co-learning of images with the same style/content. Moreover, with $r > 1$ we can learn more images at once which will improve the efficiency but if we split the <r, r, 1> triplets to be r^2 <1, 1, 1> triplets, the time will increase nearly r^2 times under the same condition.

4.3.3 Effect of the Skip-connection

To evaluate the effectiveness of the skip-connection during image generation, we compare the results with and without skip-connection in Figure 6. As shown in the figure, images in D_1 are generated best, next is D_3 and last is D_2 and D_4, which conforms to the difficulty level and indicates that novel contents are more challenging to extract than novel styles. For known contents, models with and without skip-connection perform closely but for novel contents, images generated with skip-connection are much better in details. Besides, the model without skip-connection may generate images of novel characters to be similar characters which it has seen before. This is because the structure of novel characters is more challenging to extract and the structure information losing during down-sampling will lead the model to generate blurry even wrong characters. However, with content skip-connection, the location and structure information lost will be recaptured by the *Decoder* network.

4.3.4 Validation of Style and Content Separation

Separating style and content is the key feature of the proposed *EMD* model. To validate the clear separation of style

Figure 7. Validation of pure style extraction. CR: the content reference set, TG: the target image, O1, O2 and O3 are generated by CR and three different style reference sets SR1, SR2 and SR3.

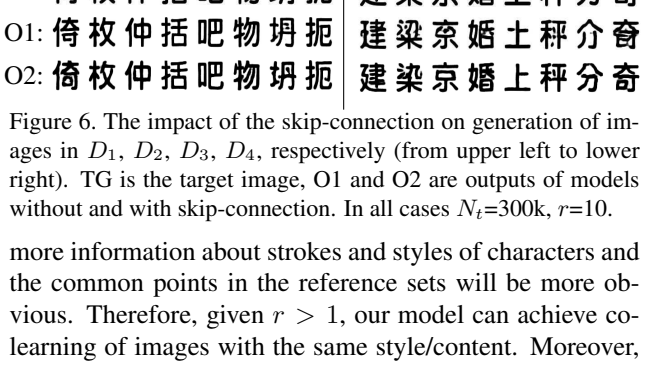

Figure 8. Validation of pure content extraction. SR: the style reference set, TG: the target image, O1, O2 and O3 are generated using SR but three different content reference sets CR1, CR2 and CR3.

and content, we combine one content representation with style representations from a few disjoint style reference sets for one style and check whether the generated images are the same. For better validation, the content reference sets and style reference sets are all for novel styles and contents and we generate images with novel style and novel content. Similarly, we combine one style representation with content representations from a few disjoint content reference sets. The results are displayed in Figure 7 and Figure 8, respectively. As shown in Figure 7, the generated O1, O2 and O3 are similar though the style reference sets used are different, demonstrating that the *Style Encoder* extracts accurate style representations since the only one thing the three style reference sets share is the style. Similar results can be found in Figure 8, showing that the *Content Encoder* extracts accurate content representations.

4.3.5 Comparison with Baseline Methods

In this subsection, we compare our method with the following baselines for character style transfer.

- Pix2pix [10]: Pix2pix is a conditional GAN based image translation network, which also adopts the skip-connection to connect encoder and decoder. Pix2pix is optimized by L1 distance loss and adversarial loss.

Figure 9. Comparison of image generation for known styles and novel contents. Equal number of image pairs with source and target styles are used to train the baselines.

- Auto-encoder guided GAN (AEGN) [16]: AEGN consists of two encoder-decoder networks, one for image transfer and another acting as an auto-encoder to guide the transfer to learn detailed stroke information.
- Zi-to-zi [2]: Zi-to-zi is proposed for Chinese typeface transfer which is based on the encoder-decoder architecture followed by a discriminator. In discriminator, there are two fully connected layers to predict the real/fake and the style category respectively.
- CycleGAN (C-GAN) [28]: CycleGAN consists of two mapping networks which translate images from style A to B and from style B to A, respectively and construct a cycle process.

For comparison, we use the font Song as the source font which is simple and commonly used and transfer it to target fonts. Our model is trained with N_t=300k and r=10 and as an average, we use less than 500 images for each style. We compare our method with baselines on generating images with known styles and novel styles, respectively. For novel style, the baselines is re-trained from scratch.

Known styles as target style. Taking known styles as the target style, baselines are trained using the same number of paired images as the images our model used for the target style. The results are displayed in Figure 9 where CycleGAN is denoted as C-GAN for simplicity. We can observe that for known styles and novel contents, our method performs much better than pix2pix, AEGN and CycleGAN and close to or a little better than zi-to-zi. This is because pix2pix and AEGN usually need more samples to learn a style as Lyu did in [16]. CycleGAN performs poorly and it only generates part of characters or some strokes, which may be because it learns the domain mappings and without the domain knowledge, it may perform poorly. Zitozi performs well since it learns multiple styles at the same time and the contrast among different styles helps the model learn styles better.

For quantification analysis, we calculate the L1 loss, Root Mean Square Error (RMSE) and the Pixel Disagreement Ratio (PDAR) [28] between generated images and target images. PDAR is the number of pixels with different values in the two images divided by the total image size after image binaryzation. We conduct experiments for 10 randomly sampled styles and the average results are displayed at the last three columns in Figure 9 and the best performance is bold. We can observe that our method performs best and achieves the lowest L1 loss, RMSE and PDAR.

Novel styles as target style. Taking novel styles as the target style, we test our model to generate images of novel styles and contents given r=10 style/content reference images without retraining. As for baselines, retraining is needed. Here, we conduct two experiments for baselines. One is that we first pretrain a model for each baseline method using the training set our method used and then fine-tune the pretrained model with the same 10 reference images as our method used. The results show that all baseline methods preform poorly and it is unfeasible to learn a style by fine-tuning on only 10 reference images. Thus, we omit the experiment results here.

The other setting is training the baseline model from scratch. Since it is unrealistic to train baseline models with only 10 samples, we train them using 300, 500, 1299 images of the target style respectively. Here we use 1299 is because the number of train contents is 1299 in our data set. The results are presented in Figure 10. As shown in the figure, the proposed *EMD* model can generalize to novel styles from only 10 style reference images but other methods need to be retrained with more samples. The pix2pix, AEGN and CycleGAN perform worst even learned on all 1299 training images, which demonstrates that these three methods are not effective for character style transfer especially when the training data are not enough. With only 10 style reference images, our model performs better than zi-to-zi-300 namely zi-to-zi model learned with 300 examples for each style, close to zi-to-zi-500 and a little worse than zi-to-zi-1299. This may be because zi-to-zi learns multiple styles at the same time and learning with style contrast helps model learning better.

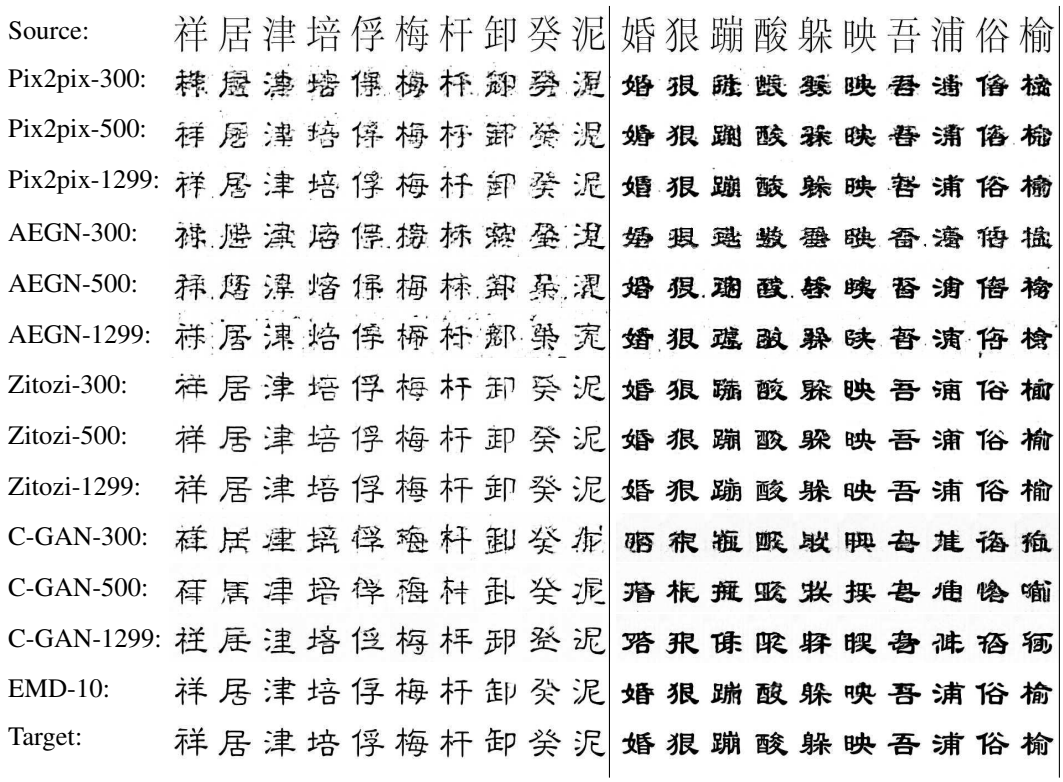

Figure 10. Comparison of image generation for novel styles and contents given r=10. The baseline methods are trained with 300, 500, 1299 image pairs respectively.

The quantitative comparison results including L1 loss, RMSE and PDAR are shown at the last three columns of Figure 10 and we can observe that though given only 10 style reference images, our method performs better than all pix2pix, AEGN and CycleGAN models and zi-to-zi-300, and close to zi-to-zi-500 and zi-to-zi-1299, which demonstrates the effectiveness of our method.

In conclusion, these baseline methods require many images of source styles and target styles to learn, which may hard to collect for some styles. Besides, the learned baseline model can only transfer styles appearing in train set and for new styles, they have to be retrained which is time-consuming. But our method can generalize to novel styles given only a few reference images. In addition, baseline models can only use images of target styles. However, since the proposed *EMD* model learns feature representations instead of transformation among specific styles, it can leverage images of any styles and make the most of existing data.

5. Conclusion and Future Work

In this paper, we propose a generalized style transfer network named *EMD* which could generate images with new styles and contents given only a few style and content reference images. The main idea is that from these reference images, the *Style Encoder* and *Content Encoder* could extract style and content representations, respectively. Then the extracted style and content representations will be mixed by a *Mixer* to generate images with target styles and contents. To separate style and content, we leverage the conditional dependence of styles and contents given an image. This learning framework allows simultaneous style transfer among multiple styles and can be deemed as a special 'multi-task' learning scenario. Then the learned encoders and mixer will be taken as the shared knowledge and transferred to new styles and contents. We evaluate the proposed method on Chinese Typeface transfer task and extensive experiments demonstrate its effectiveness.

In our study, the learning process consists of a series of image generation tasks and we try to learn a model which can generalize to novel but related tasks by learning a high-level strategy, namely learning the feature representations. This resembles to "learning-to-learn" program. In the future, we will explore more about "learning-to-learn" and integrate it with our framework.

Acknowledgment

The work is partially supported by the High Technology Research and Development Program of China 2015AA015801, NSFC 61521062, STCSM 15DZ2270400.

References

[1] Rewrite. https://github.com/kaonashi-tyc/Rewrite.

[2] Zi-to-zi. https://github.com/kaonashi-tyc/zi2zi.

[3] K. Bousmalis, N. Silberman, D. Dohan, D. Erhan, and D. Krishnan. Unsupervised pixel-level domain adaptation with generative adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[4] S. Changpinyo, W. Chao, B. Gong, and F. Sha. Synthesized classifiers for zero-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5327–5336, 2016.

[5] D. Chen, L. Yuan, J. Liao, N. Yu, and G. Hua. Stylebank: An explicit representation for neural image style transfer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[6] T. Q. Chen and M. Schmidt. Fast patch-based style transfer of arbitrary style. *arXiv preprint arXiv:1612.04337*, 2016.

[7] A. Frome, G. Corrado, J. Shlens, S. Bengio, J. Dean, T. Mikolov, et al. Devise: A deep visual-semantic embedding model. In *Advances in neural information processing systems*, pages 2121–2129, 2013.

[8] A. Gatys, A. Ecker, and M. Bethge. Image style transfer using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2414–2423, 2016.

[9] X. Huang and S. Belongie. Arbitrary style transfer in real-time with adaptive instance normalization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[10] P. Isola, J. Zhu, T. Zhou, and A. Efros. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[11] S. Jégou, M. Drozdzal, D. Vazquez, A. Romero, and Y. Bengio. The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1175–1183. IEEE, 2017.

[12] J. Johnson, A. Alahi, and F. Li. Perceptual losses for real-time style transfer and super-resolution. In *Proceedings of the European Conference on Computer Vision*, pages 694–711. Springer, 2016.

[13] M. Y. Liu, T. Breuel, and J. Kautz. Unsupervised image-to-image translation networks. In *Advances in Neural Information Processing Systems*, pages 700–708, 2017.

[14] M. Y. Liu and O. Tuzel. Coupled generative adversarial networks. In *Advances in Neural Information Processing Systems 29*, pages 469–477. 2016.

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

[16] P. Lyu, X. Bai, C. Yao, Z. Zhu, T. Huang, and W. Liu. Auto-encoder guided gan for chinese calligraphy synthesis. In *arXiv preprint arXiv:1706.08789*, 2017.

[17] A. Mordvintsev, C. Olah, and M. Tyka. Inceptionism: Going deeper into neural networks. *Google Research Blog. Retrieved June*, 20(14), 2015.

[18] A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. In *Proceedings of the International Conference on Learning Representations*, 2016.

[19] O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. In *Proceedings of the International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 234–241. Springer, 2015.

[20] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[21] Y. Taigman, A. Polyak, and L. Wolf. Unsupervised cross-domain image generation. In *arXiv preprint arXiv:1611.02200*, 2016.

[22] J. Tenenbaum and W. Freeman. Separating style and content. In *Proceedings of the Advances in neural information processing systems*, pages 662–668, 1997.

[23] D. Ulyanov, V. Lebedev, A. Vedaldi, and V. Lempitsky. Texture networks: Feed-forward synthesis of textures and stylized images. In *Proceedings of the International Conference on Machine Learning*, pages 1349–1357, 2016.

[24] P. Upchurch, N. Snavely, and K. Bala. From a to z: supervised transfer of style and content using deep neural network generators. In *arXiv preprint arXiv:1603.02003*, 2016.

[25] P. Wilmot, E. Risser, and C. Barnes. Stable and controllable neural texture synthesis and style transfer using histogram losses. *arXiv preprint arXiv:1701.08893*, 2017.

[26] Y. Xian, Z. Akata, G. Sharma, Q. Nguyen, M. Hein, and B. Schiele. Latent embeddings for zero-shot classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 69–77, 2016.

[27] H. Zhang and K. Dana. Multi-style generative network for real-time transfer. In *arXiv preprint arXiv:1703.06953*, 2017.

[28] J. Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.