TRANSFORMING PHOTOS TO COMICS USING CONVOLUTIONAL NEURAL NETWORKS

Yang Chen[†] *Yu-Kun Lai*[⋆] *Yong-Jin Liu*[†]

[†] Tsinghua University, China
[⋆] Cardiff University, UK

ABSTRACT

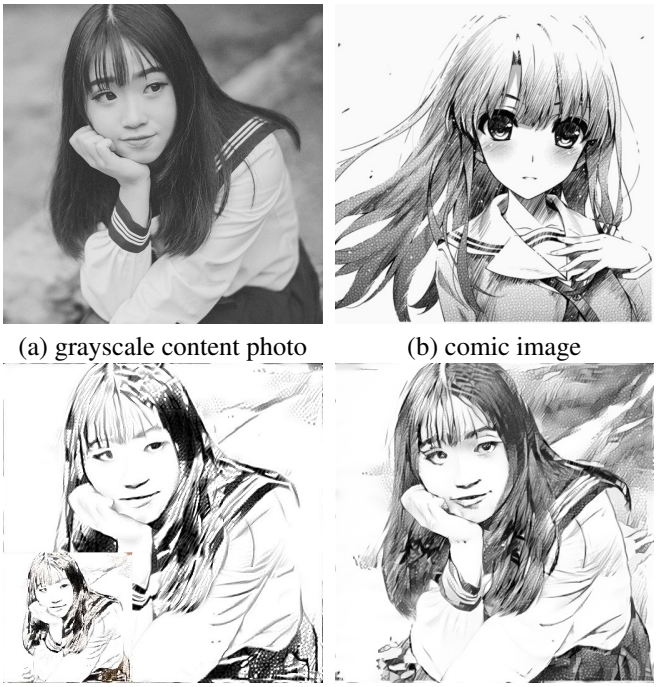
In this paper, inspired by Gatys's recent work, we propose a novel approach that transforms photos to comics using deep convolutional neural networks (CNNs). While Gatys's method that uses a pre-trained VGG network generally works well for transferring artistic styles such as painting from a style image to a content image, for more minimalist styles such as comics, the method often fails to produce satisfactory results. To address this, we further introduce a dedicated comic style CNN, which is trained for classifying comic images and photos. This new network is effective in capturing various comic styles and thus helps to produce better comic stylization results. Even with a grayscale style image, Gatys's method can still produce colored output, which is not desirable for comics. We develop a modified optimization framework such that a grayscale image is guaranteed to be synthesized. To avoid converging to poor local minima, we further initialize the output image using grayscale version of the content image. Various examples show that our method synthesizes better comic images than the state-of-the-art method.

***Index Terms*—** style transfer, deep learning, convolutional neural networks, comics

1. INTRODUCTION

Nowadays cartoons and comics are getting more and more popular worldwide. Many famous comics are created based on real world scenery. However, comic drawing involves substantial artistic skills and is very time-consuming. An effective computer program to transform photos of real world scenes to comic styles will be a very useful tool for artists to build their work on. In addition, such techniques can also be integrated into photo editing software such as Photoshop and Instagram, for turning everyday snapshots into comic styles.

Recent methods [1, 2] based on deep convolutional neural networks (CNNs) [3, 4] have shown decent performance in automatically transferring a variety of artistic styles from a style image to a content image. They produce good results for painting-like styles that have rich details, but often fail to produce satisfactory results for comics which typically contain minimalistic lines and shading. See Fig. 1 for an example. Gatys's result [1] contains artifacts of color patches, not existing in the content or style images. We turn their output images to grayscale to hide such artifacts (which is used for the remaining experiments in this paper). Their result also fails to produce essential lines to clearly delineate object boundaries, and shading similar to the given example.



(a) grayscale content photo (b) comic image

(c) Gatys's result (d) Our result

Fig. 1. An example of comic style transfer. (a) content photo, (b) comic image providing style, (c) Gatys's result turned into grayscale with the original colored output in the left corner, (d) our result. Our method avoids the artifacts of Gatys's and nicely reproduces the given comic style.

In this paper, inspired by Gatys's method, we propose a novel solution based on CNNs to transform photos to comics. The overview of our pipeline is illustrated in Fig. 2. Our method takes a content photo and a comic image as input, and produces a comic stylized output image. Since comic images are always grayscale, we first turn the content photo to a grayscale image. We formulate comic style transfer as an optimization problem combining content and style con-

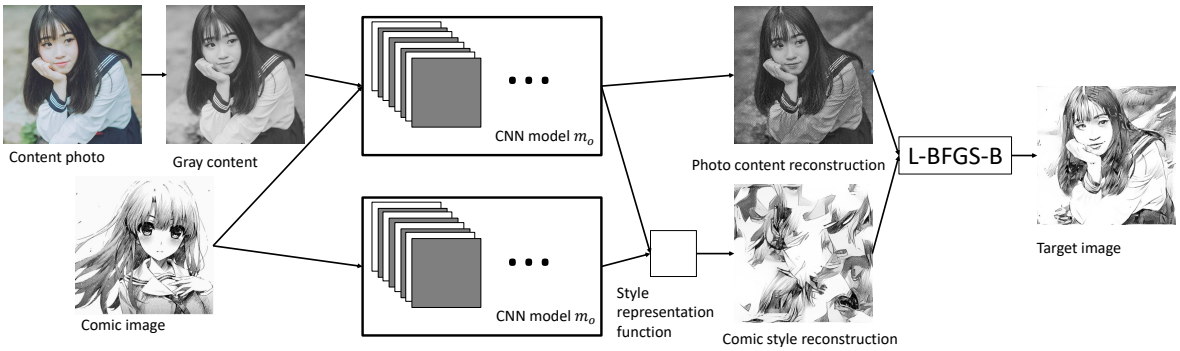


Fig. 2. Overview of the algorithm pipeline of our method.

straints. For the content constraint, we use a standard CNN model for feature extraction, similar to Gatys's method. For the style constraint, we observe that standard CNN models used in Gatys's method are typically trained using photos, and thus may not represent comic styles well, so we introduce a dedicated deep neural network trained for comic/photo classification, which is able to better extract comic style features. We formulate the optimization of the synthesized image in the grayscale domain, to suit the needs of comic stylization and avoid color artifacts. Moreover, we initialize the optimization with the content image (rather than the default white noise image as in [1]) along with a higher weight to the style constraint, to further improve the results.

2. RELATED WORK

Many non-photorealistic rendering methods [5–11] have been proposed, which aim at producing stylized images that mimic specific artistic styles including comic styles using algorithms. Different methods are usually needed for different styles, and they may work well only for certain input images.

Recently, Gatys et al. [1] propose a new way to create artistic images using deep CNNs. This method takes a content image and a style image as input and uses the original VGG network [12] trained for object recognition to transfer the texture information of the style image to the content image. It works very well when style images are more abstract or contain rich textures (e.g. painting), but fail to produce ideal results for comics. Li and Wand [2] combine a CNN with a Markov Random Field (MRF) for style transfer. Their method is also based on the VGG network [12]. In this paper, we propose a novel approach that introduces a dedicated comic style network for more effective comic style transfer.

3. OUR COMIC STYLE TRANSFER METHOD

3.1. Training the comic style network

To better represent comic styles, we introduce a dedicated comic style network. It is a new deep neural network trained for classification of comics and photos. We train our model based on the 16-layer VGG-network [12], a CNN that has outstanding performance in classification tasks. The same network architecture trained by [13] for object classification is used for content feature extraction.

To train our comic style network, we take comic images drawn by 10 different artists and photos of real world objects and scenes. Altogether 1482 comic images and 4234 photos, as well as their horizontally mirrored images are used as the training data, 100 comic images and 300 photos are used as validation data and another dataset with 100 comic images and 300 photos as test data. Because all the comic images are square grayscale images, we fix the resolution of all the images to 224×224. We then change the input layer of VGG-16 to a grayscale image, and set the number of output labels to 2, namely comics and photos.

The classification accuracy of our network is 99.25% in the validation data and 99.5% in the test data, which shows that our network has the ability to extract useful comic style features and differentiates comics from photos. As we will show later in Sec. 4, our comic style network is capable of extracting comic features effectively.

3.2. Transforming photos to comics

We now describe our method to synthesize comic style images with the given content. Similar to [1], we use convolution layers and pooling layers to extract feature maps of content and style in different network levels. The output image is reconstructed using gradient descent by minimizing joint losses between its feature maps and those of input images.

3.2.1. Content features and loss function

To calculate the features representing the content of the photo, we use the model m_o of the pre-trained VGG-16 network [13], which is available in the Caffe framework [14].

For each layer l of the CNN m_o, N_l feature maps are obtained using N_l different filters. For both the content photo p and the target image t, we can obtain their filter responses P^l

and T^l through m_o. Following [1], the content feature loss is defined as:

$$\mathcal{L}_{ontent}(p,t,l) = \frac{1}{2}\sum_{i,j}(T_{ij}^l - P_{ij}^l)^2 \tag{1}$$

where T_{ij}^l and P_{ij}^l are the i^{th} feature map at position j in layer l of the model m_o. The derivative of the content loss can be worked out as follows:

$$\frac{\partial \mathcal{L}_{content}(p,t,l)}{\partial T_{ij}^l} = \begin{cases} (T^l - P^l)_{ij} & T_{ij}^l > 0 \\ 0 & T_{ij}^l < 0 \end{cases} \tag{2}$$

which is used to reconstruct the target image using back propagation. In the content representation, we use the feature maps in 'conv4_2' to compute the content loss.

3.2.2. Comic style features and loss function

To better represent comic style features, we propose to use two VGG-16 models, where one is the same model m_o for content feature extraction which captures generic styles, and the other is our comic style network m_s described in Sec. 3.1, which represents comic specific styles. To represent styles in a spatially independent way, Gram matrices of size $N_l \times N_l$ are used [15]: $G_{ij}^l = \sum_k T_{ik}^l T_{jk}^l$. Let $S^{l,m}$ and $C^{l,m}$ be the Gram matrices of the target image t and comic image c, for the model $m \in \{m_o, m_s\}$. The contribution of each layer in model m to the style loss is defined as:

$$E^{l,m} = \frac{1}{4N_l^2 M_l^2}\sum_{i,j}(S_{ij}^{l,m} - C_{ij}^{l,m})^2 \tag{3}$$

where N_l and M_l are the number of the feature maps and the size of each feature map, respectively. The derivative of $E^{l,m}$ is:

$$\frac{\partial E^{l,m}}{\partial T_{ij}^l} = \begin{cases} \frac{1}{N_l^2 M_l^2}\left((T^l)^T(S^l - C^l)\right)_{ij} & T_{ij}^l > 0 \\ 0 & T_{ij}^l < 0 \end{cases} \tag{4}$$

We define the total style loss using features of both models:

$$\mathcal{L}_{style}(c,t) = \alpha\sum_{l=1}^{L}\frac{1}{L}E^{l,m_s} + (1-\alpha)\sum_{l=1}^{L}\frac{1}{L}E^{l,m_o} \tag{5}$$

where $\alpha \in [0,1]$ is the weight to balance the two style models. Its influence will be discussed in Sec. 4. l iterates over the style representation layers which we set to 'conv1_1', 'conv2_1', 'conv3_1', 'conv4_1' and 'conv5_1' in this paper, and $L = 5$ is the number of layers used.

3.2.3. Target image reconstruction

We define the joint loss function by combining the content and style losses defined in the previous subsection:

$$\mathcal{L}(p,c,t) = \mathcal{L}_{content}(p,t) + \beta\mathcal{L}_{style}(c,t) \tag{6}$$

(a) input images (b) Gatys's results (c) our results

Fig. 3. Comparison of comic style transfer results. (a) input content and style images given by the user, (b) results by Gatys's method, (c) our results.

where $\beta \in \mathbb{R}$ is the weighting factor for style conformity; we will illustrate its influence in Sec. 4.

We can then reconstruct our target image by minimizing Eq. 6 using L-BFGS-B [16, 17]. To ensure the target image is grayscale, we set the gradient ∇T for updating the target image as the average of the gradients in the three color channels to ensure consistent update in different channels:

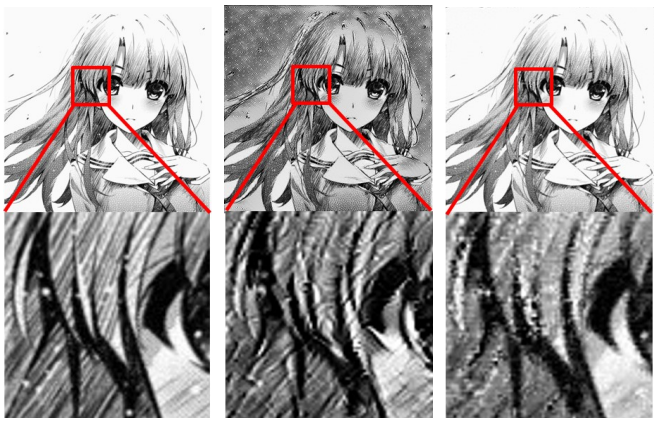
$$\nabla T = \frac{1}{3}(\nabla T_r + \nabla T_g + \nabla T_b). \tag{7}$$

We initialize the target image using the grayscale version of the content photo to provide more content constraint, and set a higher β to the joint loss function (Eq. 6) to better transfer the comic style while preserving the content scenery.

4. EXPERIMENTAL RESULTS

We have given an example of our method in Fig. 1. Fig. 3 shows more results and compare them with Gatys's method [1]. To avoid bias, input images are not present in the dataset used to train our comic style network. For fair comparison, we optimize the Gatys's results by using the grayscale content image for initialization, setting $\beta = 10^4$, and turning their output images to grayscale to remove color artifacts. The results are otherwise much worse. We use fixed parameters $\alpha = 0.5$, $\beta = 10^4$ for our method. We can see that our method produces lines and shading that better mimic the given comic style, and better maintains object information of the content photo, leading to visually improved comic style images. Our method (as well as [1]) does not take semantic

Fig. 4. Image stylization results using different combinations of parameters. Images in the same row share the same α value and images in the same column share the same β value, as labeled. The input images of these results are the same as Fig. 1.



(a) original comic image (b) reconstruction using standard VGG (c) our reconstructed image

Fig. 5. The reconstructed images from a white noise image using standard VGG (b) and our comic style network (c).

information into account, so it may transfer semantically different regions from the style images to target images. This could be improved by using semantic maps [18, 19].

Presentation ability of our model. To demonstrate the presentation ability of our style network for comic images, we reconstruct the comic image from a white noise image using the features computed by our comic style network as $\mathcal{L}_{content}$ with the $\mathcal{L}_{style}$ term ignored. As shown in Fig. 5, our model can extract useful information to effectively recover the comic image, whereas using the standard VGG network for $\mathcal{L}_{content}$, the reconstructed image fails to preserve essential lines, object boundaries and shading.

Influence of the parameter setting. Our comic style transfer method has two parameters: the weight between two style models α and the weight of style loss β. Fig. 4 illustrates how different parameters influence the results. We can see that our comic style network m_s provides more detailed information for comic shading while m_o provides more outline information (see rows of Fig. 4). Regarding β, larger β leads to more style constraint and less content constraint in the target image (see columns of Fig. 4). Choosing $\alpha = 0.5$ and $\beta = 10^4$ achieves a good balance between content and style preservation.

5. CONCLUSION

In this paper, we propose a novel approach based on deep neural networks to transform photos to comic styles. In particular, we address the limitation of [1] in transferring comic styles, by introducing a dedicated comic style network to the loss function for optimizing target images. We further constrain the optimization of target images to be in the grayscale image domain, avoiding artifacts of color patches. The experimental results show that our method preserves line structures, especially object boundaries better with improved lines and shading closer to the given example. As future work, we would like to investigate building a feed forward neural network [20] to approximate the solution, to improve the efficiency for real-time applications.

Acknowledgements

This work was supported by Royal Society-Newton Advanced Fellowship (NA150431), the Natural Science Foundation of China (61661130156) and Beijing Higher Institution Engineering Research Center of Visual Media Intelligent Processing and Security.

References

[1] L.A. Gatys, A.S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2414–2423.

[2] C. Li and M. Wand, "Combining Markov random fields and convolutional neural networks for image synthesis," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 2479–2486.

[3] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[4] S. Lawrence, C. L. Giles, A. C. Tsoi, and A. D. Back, "Face recognition: A convolutional neural-network approach," *IEEE Transactions on Neural Networks*, vol. 8, no. 1, pp. 98–113, 1997.

[5] A. Hertzmann, "Painterly rendering with curved brush strokes of multiple sizes," in *ACM SIGGRAPH*, 1998, pp. 453–460.

[6] D. Mould, "A stained glass image filter," in *Eurographics Workshop on Rendering*, 2003, pp. 20–25.

[7] C.-K. Yang and H.-L. Yang, "Realization of Seurats pointillism via non-photorealistic rendering," *The Visual Computer*, vol. 24, no. 5, pp. 303–322, 2008.

[8] J. E. Kyprianidis and J. Döllner, "Image abstraction by structure adaptive filtering," in *EG UK Theory and Practice of Computer Graphics*, 2008, pp. 51–58.

[9] M. Zhao and S.-C. Zhu, "Sisley the abstract painter," in *International Symposium on Non-Photorealistic Animation and Rendering*, 2010, pp. 99–107.

[10] S.-H. Zhang, X.-Y. Li, S.-M. Hu, and R. R. Martin, "Online video stream abstraction and stylization," *IEEE Transactions on Multimedia*, vol. 13, no. 6, pp. 1286–1294, 2011.

[11] P. L. Rosin and Y.-K. Lai, "Artistic minimal rendering with lines and blocks," *Graphical Models*, vol. 75, no. 4, pp. 208–229, 2013.

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

[13] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg, and L. Fei-Fei, "ImageNet large scale visual recognition challenge," *International Journal of Computer Vision*, vol. 115, no. 3, pp. 211–252, 2015.

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," in *ACM International Conference on Multimedia*, 2014, pp. 675–678.

[15] L. A. Gatys, A. S. Ecker, and M. Bethge, "Texture synthesis and the controlled generation of natural stimuli using convolutional neural networks," *arXiv preprint arXiv:1505.07376*, 2015.

[16] R. H. Byrd, P. Lu, J. Nocedal, and C. Zhu, "A limited memory algorithm for bound constrained optimization," *SIAM Journal on Scientific Computing*, vol. 16, no. 5, pp. 1190–1208, 1995.

[17] C. Zhu, R. H. Byrd, P. Lu, and J. Nocedal, "Algorithm 778: L-BFGS-B: Fortran subroutines for large-scale bound-constrained optimization," *ACM Transactions on Mathematical Software (TOMS)*, vol. 23, no. 4, pp. 550–560, 1997.

[18] A. J. Champandard, "Semantic style transfer and turning two-bit doodles into fine artworks," *arXiv preprint arXiv:1603.01768*, 2016.

[19] L. A. Gatys, A. S. Ecker, M. Bethge, A. Hertzmann, and E. Shechtman, "Controlling perceptual factors in neural style transfer," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[20] T. Q. Chen and M. Schmidt, "Fast patch-based style transfer of arbitrary style," in *Advances in neural information processing systems*, 2016.