DeepStyleCam: A Real-Time Style Transfer App on iOS

Ryosuke Tanno, Shin Matsuo, Wataru Shimoda, and Keiji Yanai(✉)

Department of Informatics, The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo 182-8585, Japan
{tanno-r,yanai}@mm.inf.uec.ac.jp

Abstract. In this demo, we present a very fast CNN-based style transfer system running on normal iPhones. The proposed app can transfer multiple pre-trained styles to the video stream captured from the built-in camera of an iPhone around 140ms (7fps). We extended the network proposed as a real-time neural style transfer network by Johnson et al. [1] so that the network can learn multiple styles at the same time. In addition, we modified the CNN network so that the amount of computation is reduced one tenth compared to the original network. The very fast mobile implementation of the app are based on our paper [2] which describes several new ideas to implement CNN on mobile devices efficiently. Figure 1 shows an example usage of DeepStyleCam which is running on an iPhone SE.

1 Introduction

In 2015, Gatys et al. proposed an algorithm on neural artistic style transfer [3,4] which synthesizes an image which has the style of a given style image and the contents of a given content image using Convolutional Neural Network (CNN) as shows in Fig. 2. This method enables us to modify the style of an image keeping the content of the image easily. It replaces the information which are degraded while the signal of the content image goes forward through the CNN layers with style information extracted from the style image, and reconstructs a new image which has the same content as a given content images and the same style as a given style image. In this method, they introduced "style matrix" which was presented by Gram matrix of the feature maps, that is, correlation matrix between feature maps in CNN.

However, since the method proposed by Gatys et al. required forward and backward computation many times (in general several hundreds times), the processing time tends to become longer (several seconds) even using GPU.

Then, several methods using only feed-forward computation of CNN to realize style transfer have been proposed so far. One of them is the method proposed by Johnson et al. [1]. They proposed perceptual loss functions to train the ConvDeconvNetwork as a feed-forward style transfer network. The ConvDeconvNetwork consists of down-sampling layers, convolutional layers and up-sampling layers,

L. Amsaleg et al. (Eds.): MMM 2017, Part II, LNCS 10133, pp. 446–449, 2017.
DOI: 10.1007/978-3-319-51814-5_39

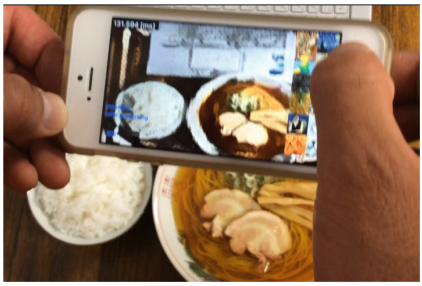

Fig. 1. "DeepStyleCam" running on an iPhone SE.

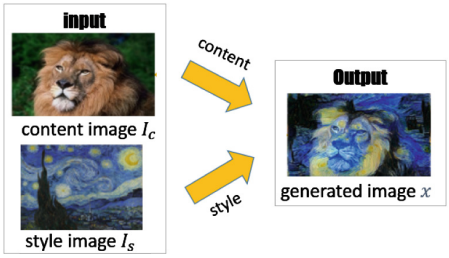

Fig. 2. "Neural style transfer" which creates a novel image by mixing the content and the style of two given images.

which accepts an image and outputs an modified image. This networks is commonly used for super-resolution [5] and coloring of gray-scale images [6]. In their method, they train a ConvDeconvNetwork so that the style matrix of its output image becomes closer to the style matrix of the given fixed style image and the CNN features of the input image leaves unchanged by using the proposed perceptual losses.

However, the ConvDeconvNetwork trained by their method can treat only one fixed style. If transferring ten kinds of styles, we have to train ten different ConvDeconvNetwork independently. This is not good for mobile implementation in terms of required memory size. Then, we modified Johnson et al.'s method so that one ConvDeconvNetwork can train multiple styles at the same time.

2 Proposed System

We modified the ConvDeconvNetwork used in [1] by adding a fusion layer and a style input stream as shown in Fig. 3. This is inspired by Iizuka et al's CNN-based coloring work [6]. They proposed adding a contextual stream to the ConvDeconvNetwork. With this improvement, they achieved coloring depending on the content of a target image.

We propose a style transfer network with style input as shown in Fig. 3. When training, we provide sample images to the content stream and style images to the style stream. The training method is the same as [1]. Please refer the detail on training to [1].

When transferring images, we input a target image to the content stream and one of the style images used in the training phrase to the style stream. Then, we can obtained the results corresponding on the selected style images by using only one trained network as shown in the top row of Fig. 4.

In addition, we shrunk the ConvDeconvNetwork compared to [1] to save computation costs. We added one down-sampling layer and up-sampling layer, replaced 9×9 kernels with smaller 5×5 kernels in the first and last convolutional

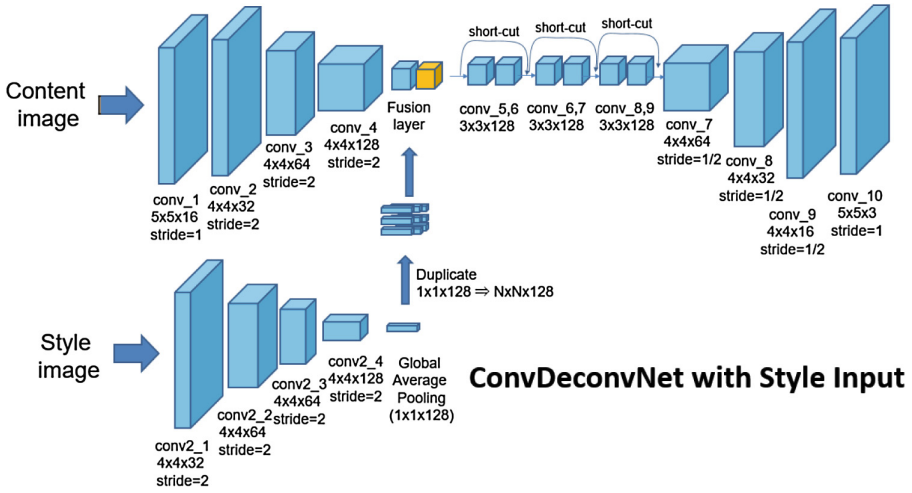

Fig. 3. Style transfer network with style input.

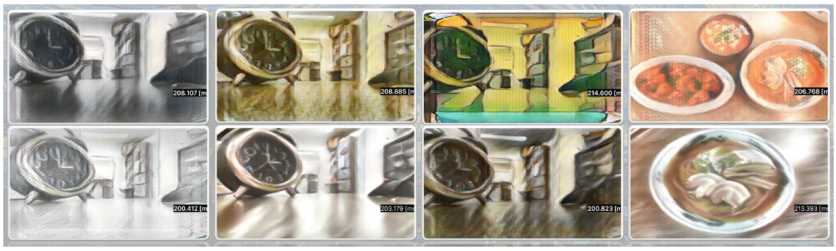

Fig. 4. Examples of the results. (Top) normal mode, (bottom) color preserving mode. (Color figure online)

layers, and reduced five Residual Elements into three. We confirmed that these network shrinking did not harm the quality of outputs significantly.

Regarding mobile implementation, we followed our work on Efficient Mobile CNN Implementation [2]. In the method, CNN networks are directly converted to a C source code which utilized multi-threading and iOS Accelerate Framework for CNN computation.

In addition, we implemented color preserving mode [7] which transfers a selected style only into gray-scale elements of an input image. The results in the color preserving mode as shown in the bottom row of Fig. 4. It can keep color of the content image while changing only the intensity of pixels, which is especially suitable for food images.

3 Demo Video and App on the iOS App Store

We prepared the videos recorded that the DeepStryleCam app was running in the practical settings. We released the app on the iOS app store. You can try "DeepStyleCam" on your iPhone or iPad. Note that we strongly recommend to use iPhone 6s/7/SE or iPad Pro for this app, because the app requires much computational power.

- Demo video of the DeepStyleCam.
  http://foodcam.mobi/deepstylecam/
- iOS app on the App Store, "RealTimeMultiStyleTransfer"
  https://itunes.apple.com/jp/app/realtimemultistyletransfer/id1161707531

References

1. Johnson, J., Alahi, A., Fei, L.F.: Perceptual losses for real-time style transfer and super-resolution. In: Proceedings of European Conference on Computer Vision (2016)
2. Yanai, K., Tanno, R., Okamoto, K.: Efficient mobile implementation of a CNN-based object recognition system. In: Proceedings of ACM Multimedia (2016)
3. Gatys, L.A., Ecker, A.S., Bethge, M.: A neural algorithm of artistic style. arXiv:1508.06576 (2015)
4. Gatys, L.A., Ecker, A.S., Bethge, M.: Image style transfer using convolutional neural networks. In: Proceedings of IEEE Computer Vision and Pattern Recognition (2016)
5. Dong, C., Loy, C.C., He, K., Tang, X.: Learning a deep convolutional network for image super-resolution. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8692, pp. 184–199. Springer, Heidelberg (2014). doi:10.1007/978-3-319-10593-2_13
6. Iizuka, S., Simo-Serra, E., Ishikawa, H.: Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification. ACM Trans. Graph. **35**(4), 110 (2016). (Proceedings of SIGGRAPH 2016)
7. Gatys, L.A., Bethge, M., Hertzmann, A., Shechtman, E.: Preserving color in neural artistic style transfer. arXiv:1606.05897 (2016)