One-to-one Example-based Automatic Image Coloring Using Deep Convolutional Generative Adversarial Network

Junghoon Seo, Taewon Yoon, and Jinwoo Kim
Electrical Engineering and Computer Science Department, GIST College, Gwangju, Republic of Korea
Email: {sjh14, dbsus13, yd8012mw2}@gist.ac.kr

Kin Choong Yow
Division of Liberal Arts and Sciences, GIST College, Gwangju, Republic of Korea
Email: kcyow@gist.ac.kr

***Abstract*—Due to the indeterminate nature of the problem, image colorization techniques currently rely heavily on human intuition. Using deep convolutional networks, we can build a system that takes a source image to guide local-dependent feature color mapping and color a grayscale target image. Unlike most other convolutional neural network approaches that require a lot of training data, our proposed system uses only one image for training for each target image. Our system is based on deep convolutional generative adversarial networks, which contains concepts of both supervised and unsupervised learning. We proposed a model architecture, objective functions, and both preprocessing and postprocessing algorithms for the image coloring process. We evaluated our system on a variety of input images and showed that it produce excellent results.**

***Index Terms*—automatic image coloring, deep convolutional generative adversarial network, image processing, computer vision, neural network**

I. Introduction

Image colorization technique, which makes grayscale image colored, is one of the classical topics of computer vision. Traditionally, there are two approaches for image colorization. One is a scribble-based approach and the other is an example-induced approach. The former approach requires the user to provide local image color information, which is a color distribution of each feature in the image for coloring. The latter approach does not rely on user guidance, rather, it requires unique color information which is not identical to the target grayscale image to be colored. Strictly, example-induced colorization transfers the color composition from a full-color image to a grayscale image. The main objective of our work is to color the grayscale image based on a similar, colored image using example-induced colorization.

There are many existing research that use example-based image colorization. Recent approaches use neural network (NN) methods based on a large number of images' color information for image colorization. However, in practice, it is unrealistic to expect that one would have a large number of colored images that is similar to the target grayscale image. Hence, we focus our work on one-to-one image colorization, i.e. we use only one source image to color the target grayscale image. To achieve this, our method uses an NN approach that is focused on local image features rather than global image features.

Deep Convolutional Generative Adversarial Network (DCGAN), a modified model of the generative adversarial network (GAN), is known to be a suitable model for image generation in the unsupervised context. Our method adopts a variation of DCGAN to generate a bunch of local colored images from a grayscale target image, which includes both supervised and unsupervised learning. After the generation of local colored images, they are merged into one single colored image using the criteria of structural similarity (SSIM) index [1]. Subsequently, LAB space separation and Quick segmentation [2] techniques are applied to produce an enhanced result of the model.

II. Related Works

A. Colorization Methods

In approaches using user inserted color scribbles, Levin et al. [3] used optimization techniques based on the premise that neighboring pixels that have similar luminosity should have similar color. Yatziv and Sapiro [4] added the concept of color blending and Nie et al. [5] improved the computational time using quadtree decomposition. In example-induced approaches, Welsh et al. [6] transferred the entire color model to the target image by comparing luminance and texture between source and target images. Gupta et al. [7] used a fast cascade feature matching scheme, and Liu and Zhang [8] performed locally weighted regression on both the grayscale image and the source image for faster computation. Chariot et al. [9] used the probability distribution of all possible colors instead of choosing the

Manuscript received October 21, 2016; revised May 17, 2017.

doi: 10.12720/jait.8.2.80-85

most probable color. Recent papers used neural network for colorization and made fully automatic approaches. Cheng et al. [10] applied deep learning techniques and used a joint bilateral filtering for post-processing. Iizuka et al. [11] extracted global and local feature separately, and removed the dependency on segmentation.

B. GAN and Its Extensions

Goodfellow et al. [12] proposed an adversarial network structure of which both the generator model and the discriminative model are trained simultaneously. Their paper theoretically formulates global optimality and convergence of a generated distribution. One of the notable advancement of GAN architecture is by Denton et al. [13] which uses a cascade of convolutional neural networks (CNN) [14] with a Laplacian pyramid structure. Radford et al. [15] adds unsupervised learning methods to the image processing, and introduces the DCGAN. In their paper, the authors argue that DCGAN is a result of joining GAN and CNN.

III. Local Image Colorization by DCGAN

We propose a variation of DCGAN where the generator network maps from local images of a target to a generative distribution of colored images. Unlike most neural network methods, our approach does not contain any level of feature extractor such as a local descriptor or semantic segmentation in a network configuration. This is because we intend to achieve a native performance of DCGAN for an image colorization task.

A. Our Proposed DCGAN Architecture

The original DCGAN architecture [12] consists of two deep neural networks, a discriminator and a generator. We adopt a basic DCGAN but transform it to take input as two-dimensional random images, not a one-dimensional random vector. Fig. 1 presents the whole architecture of our DCGAN. Note both the discriminator and generator input and output. The shape of the discriminator input is 64 rows by 64 columns with 3 dimensions and the shape of the discriminator output is 100 rows with 1 dimension. The shape of the generator input is 64 rows by 64 columns with 1 dimension and the shape of the generator output is 64 rows by 64 columns with 3 dimensions.

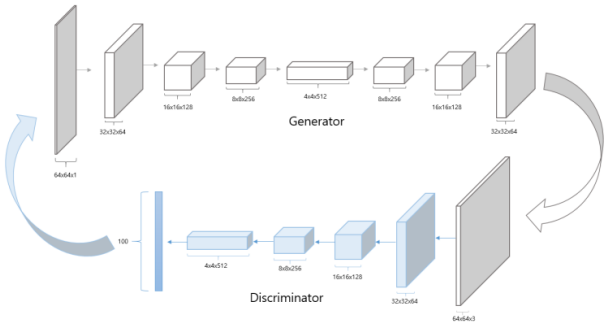

Figure 1. Flow of our DCGAN architecture. It consists of two NN architectures. Achromatic color part represents a generator and blue color part represents a discriminator. Each of the plane refers to each NN layer. Note that a big blue arrow and the white one do not indicate the same thing; The white one means image input as training data of the discriminator, but the blue one means a result of the discriminator that is dealt as one of objective function terms in the generator training.

The discriminator in our architecture is totally the same with the original model of DCGAN paper [12], which is a discriminant model of a CNN approach. In the entire architecture, the discriminator works to distinguish between imitation image and authentic image. It consists of four convolution layers that are connected to leaky ReLU activators [16] and batch normalizations [17] and one linear fully connected layer. The sigmoid function applied to the last part of the model interprets vector to probability. Table I refers to details of our discriminator architecture.

TABLE I. Architecture of the Discriminator

Type	*Kernel*	*Stride*	*Outputs*
Conv.	5 × 5	2 × 2	64
Conv.	5 × 5	2 × 2	128
Conv.	5 × 5	2 × 2	256
Conv.	5 × 5	2 × 2	512

The generator in our architecture differs slightly from the original model. The generator in our generative neural network model takes the task of generating colored images from monotone images. It consists of four convolution layers and four transpose convolution layers. After all the convolution or transpose convolution layers, ReLU activator [18] and batch normalization technique are applied. The hyper-tangent function applied to an end of generator architecture. Table II refers to details of our generator architecture.

TABLE II. Architecture of the Generator

Type	*Kernel*	*Stride*	*Outputs*
Conv.	5 × 5	2 × 2	64
Conv.	5 × 5	2 × 2	128
Conv.	5 × 5	2 × 2	256
Conv.	5 × 5	2 × 2	512
Deconv.	5 × 5	2 × 2	256
Deconv.	5 × 5	2 × 2	128
Deconv.	5 × 5	2 × 2	64
Deconv.	5 × 5	2 × 2	3

B. Objective Function of DCGAN and Training

A discriminator and a generator are the core concepts of GAN, which are different but are also correlated competitive networks. This is why it is called an "adversarial" network. Our model also follows this strategy roughly, but we adjust the objective function of the discriminator because we have information of the original color images and do not need to depend totally on unsupervised results.

The objective function of the discriminator is given in (1). D is a function which is defined as the discriminator and G is a function which is defined as the generator. x_i is the ith full-color image of the cut pieces of the source image and z_i is the ith grayscale image coming from x_i.

$$\frac{1}{m}\sum_{i=1}^{m}\left[logD\left(x_i\right)+\log\left(1-D\left(G\left(z_i\right)\right)\right)\right] \quad (1)$$

The objective function of a generator is given in (2). FRO means the Frobenius norm. Note that the first term of (2) is a kind of cross entropy and the second term of (2) is MSE.

$$\frac{1}{m}\sum_{i=1}^{m}\log\left(1-D\left(G\left(z_i\right)\right)\right)+\frac{1}{m}\sum_{i=1}^{m}\left\|G\left(z_i\right)-x_i\right\|^2{}_{FRO} \quad (2)$$

The results of objective function are computed from the result of the training process of the discriminator and generator using a stochastic gradient descent method. Each network's result works as training loss for another network. The total process is a type of minimax game and can be formulated by (3) using the same notation as (1) and (2).

$$\min_G\left[\max_D\left[E_{x\sim P_{data}}logD(x)+E_{z\sim P_z}\left[\log\left(1-D\left(G(z)\right)\right)\right]+E_{x\sim P_{data},z\sim P_z}\left\|x-G(z)\right\|^2{}_{FRO}\right]\right] \quad (3)$$

C. Preprocessing amd Postprocessing

We fix the size of the input in our architecture to 64 rows by 64 columns. Our model is set to achieve the goal that the model colorizes a grayscale image with 256 rows by 256 columns. The objective of preprocessing is to generate numerous locally spliced training data to feed the generator. The source image is cut into random regions of 64 x 64, and then random geometrical transformations, such as rotating, flipping, warping, resizing, are applied to these pieces of images. Brightness or contrast change can also be applied. The target image is also cut to the conforming shape of a regular square lattice in a random way. For example, a target image is cut into 64 pieces as defined by (4). Z_n is the nth image being cut and $Z^{original}{}_{a:b,c:d}$ is a piece of the target image that has row-wise pixel index from a to b and column-wise pixel index from c to d.

$$Z_{7i+j}=Z^{original}{}_{32i:32i+64,32j:32j+64} \quad (4)$$

After the preprocessing, our model treats the cut pieces of source image and target image as training data and testing data, respectively.

The objective of post-processing is to merge the colorized pieces of target images into a single complete colorized image and to enhance the colorized result. The problem of this amalgamation is that there are overlapped regions of 64 pieces. We introduce SSIM to solve this problem. The colorized pieces are transformed into grayscale image again, and SSIM is calculated by taking the structure difference between the localized target image and the generated piece. The merger process is a type of weighted average method, and the resulting image of the amalgamation is defined by the product of the relative SSIM fraction and the generated color distribution. Gaussian filters are used in the merged colorized image to eliminate the visual boundary line that forms from merging each piece of the image. Then, to enhance the colorization result, we introduce a Quickshift algorithm, which is a two-dimensional image segmentation method that is based on an approximation of a kernelized mean-shift. By applying Quickshift to the original grayscale target image, we obtain segmentation regions from the target image, and along with each label of segmentation, the color of each region is replaced by a median of discrete color distributions in the region. Finally, the original target image and colorization result image in RGB space are transformed into LAB space, and the AB space of the colorized result and the L space of the target image are merged into a single LAB space image.

IV. Experimental Results and Discussion

We implement our entire model using Tensorflow [19] 0.9.0. The details of our machine environment is as follows: Ubuntu 14.04, GeForce GTX1070 with 8GB VRAM, Intel(R) Core(TM) i5-6600 CPU @ 3.30GHz, 16GB main RAM, CUDA Toolkit 8.0 RC and cuDNN 5.0.

The learning rate of both the discriminator and the generator is set as 0.0006. We select Adam [20] as the stochastic gradient descent method and its momentum term is 0.5. The number of generated pieces of the target image is 12,800. The scale of the local density approximation, the level in the hierarchical segmentation, and the width of the Gaussian kernel used in smoothing the sample density of Quickshift are set as 0, 10, and 2, respectively. Also, in the training step, the number of epoch ranges from 12 to 16.

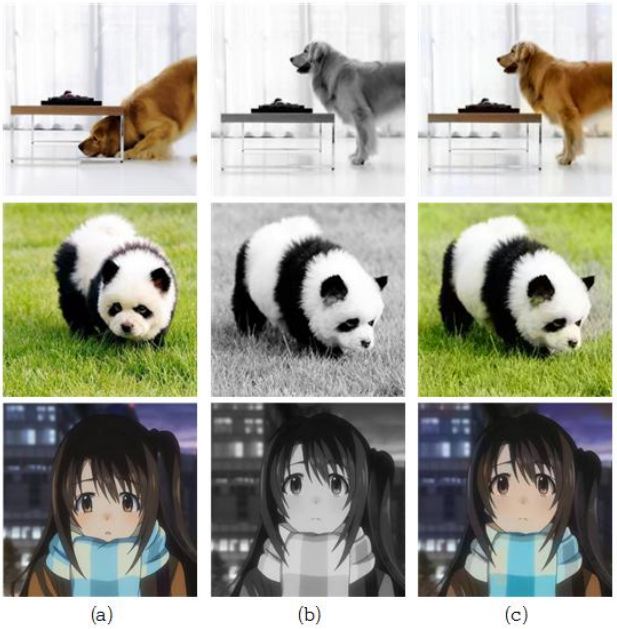

Figure 2. Demonstration of the model: (a) Source images which provide color distribution to the model. (b) Target images that the model has to colorize. (c) Results of target colorization. Those images are selected for showing successful results of our model. The most parts of each result are quite plausible.

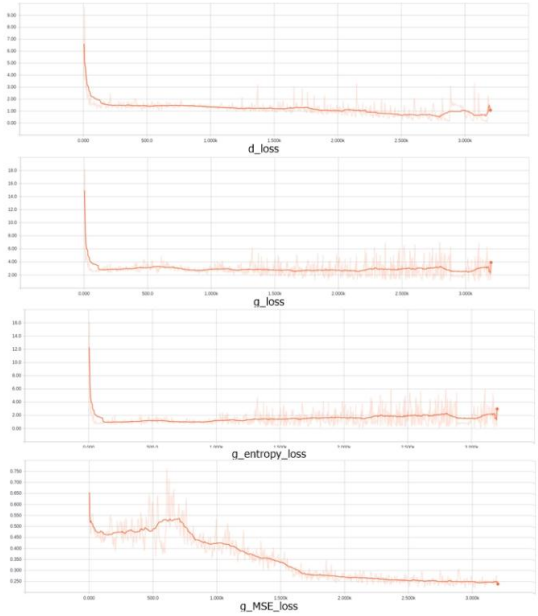

Figure 3. Changes of loss of the discriminator and the generator over the training step. This result comes from an instance of Fig. 4. Horizontal axis indicates epoch. From top to bottom, vertical axis indicates loss of a discriminator, a generator, cross entropy of the generator, and MSE of the generator. In common, a sharp decline appears in early epochs. However, like usual GAN, it is hard to reduce the loss of the generator and the discriminator.

Figure 4. Series changes of result over epochs. The top left image is the source image and the top right image is the target image. The remaining sixteen images indicate the changes of result over epochs. In the horizontal direction, training epoch increases from zero to sixteen. There are noticeable differences of performance appearing with each epoch.

Fig. 2 shows some results of our model. We can see that the model works very well in coloring the target grayscale image from only one single source image. Fig. 3 shows that change of the loss of the discriminator and the generator over the training step. Fig. 4 shows the changes of results over epoch and Fig. 5 shows the impact of the Quickshift postprocessing step.

Figure 5. Quickshift segmentation and its application into our model. The blurred image (left) is transformed into a clear, discernible image (right). A pink blob appears on one part of right upper of the blurred image and it is obvious that color is assigned wrongly. Looking the transformed image, this erroneous inference is reduced by Quickshift segmentation and mean color selection.

Figure 6. Example of significant failure of our metric. (left) Source image. (middle) Target grayscale image. (right) Target colored image. In human's semantic sense, a ground truth result is obvious; Color of buildings should be gray or blue, and color of needle-leaf tree should be brown and green. Note that much of local color distribution is generated wrongly.

Fig. 6 illustrates a limitation of our approach. We suppose the error of estimation of color distribution come from two causes. Our model focuses on a distribution of local grayscale image so it cannot see the global context of an image. On the other hand, if pieces of the source image do not appear similarly with the local grayscale target image when pieces are generated, the generator cannot generate a proper color distribution.

V. Conclusion and Future Work

We proposed a DCGAN-based model that aims at one-to-one colorization of grayscale image, using only a single source image that is similar to the target image. Our model has the advantage over other NN approaches that require numerous image instances. In this paper, we provide the designed model architecture, objective function, and both the preprocessing and post-processing algorithms. We tested our model on a variety of input images and showed that it produced excellent results.

Now we propose two possible approaches to overstep the limitation of our model. First, in GAN training, it always matters to find out proper data argumentation for a specific task. Here we are not concerned too much with a various type of data argumentation. If we try to consider it more, the performance of our model could be better. Second, we focus on the local image features, so we do

not consider any other feature descriptor or image classification task. This is intentional because we tend to measure the performance of a sole DCGAN model which carries out example-based image colorization. However, if the main purpose is set for improving the model performance, it may be better to consider a combination with other feature descriptor in preprocessing and postprocessing, or NN model architecture. It is plausible because those approaches also were applied to the other example-based colorization model including NN concept to improve their own model performance. For example, Cheng et al. [10] and Iizuka et al. [11] introduce those techniques to their own model and report improvement of the colorization performance.

Furthermore, although we emphasize our model as the model for an example-based task, our model also can be easily transposed to the model for a general scribble-based colorization task. Generally, generating a colored image from a scribbled image is algorithmically harder than the opposite one. This is because morphological segmentation is easier and more robust than semantic one in recent image processing trend. If we design a proper preprocessing which reduces a colored image to user-likely scribbled result of a grayscale image, we can readjust our model a little bit and train the adjusted model which solves scribble-based colorization task. NN approach is rarely applied to a scribble-based colorization task so it is worthy of future work.

ACKNOWLEDGMENT

This work was supported in part by the Institute for Information & communications Technology Promotion(IITP) grant funded by the Korea government(MSIP) (No.B0101-16-0525, Development of global multi-target tracking and event prediction techniques based on real-time large-scale video analysis).

REFERENCES

[1] Z. Wang, A. Bovid, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity", *IEEE Transactions on Image Processing*, vol. 13, Apr. 2004, pp. 600-612.

[2] A. Vedaldi and S. Soatto, "Quick shift and kernel methods for mode seeking," in *Proc. the 10th European Conference on Computer Vision: Part IV(ECCV 08)*, Springer Berlin Heidelberg, Oct. 2008, pp. 705-718.

[3] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using Optimization", *ACM Transactions on Graphics*, vol. 23, Aug. 2004, pp. 689-694.

[4] L. Yatziv and G. Sapiro, "Fast image and video colorization Using chrominance blending," *IEEE Transactions on Image Processing*, vol. 15, May. 2006, pp. 1120-1129.

[5] D. Nie, Q. Ma, L. Ma, and S. Xiao, "Optimization based grayscale image colorization," *Pattern Recognition Letters*, vol. 28, Sep. 2007, pp. 1445-1451.

[6] T. Welsh, M. Ashikhmin, and K. Mueller, "Transferring color to greyscale images," *ACM Transactions on Graphics*, vol. 21, July 2002, pp. 277-280.

[7] R. Gupta, A. Chia, D. Rajan, E. Ng, and H. Zhiyong, "Image colorization using similar images," in *Proc. the 20th ACM international conference on Multimedia, ACM*, Oct. 2012, pp. 369-378.

[8] S. Liu and X. Zhang, "Automatic grayscale image colorization using histogram regression," *Pattern Recognition Letters*, vol. 33, Oct. 2012, pp. 1673-1681.

[9] G. Chariot, M. Hofmann, and B. Schölkopf, "Automatic image colorization via multimodal predictions," in *Proc. the 10th European Conference on Computer Vision: Part III(ECCV 08)*, Springer Berlin Heidelberg, Oct. 2008, pp. 126-139.

[10] Z. Cheng, Q. Yang, and B. Sheng, "Deep colorization", in *Proc. the 2015 IEEE International Conference on Computer Vision, IEEE Computer Society*, Dec. 2015, pp. 415-423.

[11] S. Iizuka, E. Simo-Serra, and H. Ishikawa, "Let there be color!: joint end-to-end learning of global and local image priors for automatic image colorization with simultaneous classification", *ACM Transactions on Graphics*, vol. 35, July 2016, 110.

[12] I. Goodfellow, J. Pouget-Abadie_, M. Mirza, B. Xu, D. Warde-Farley, S. Ozairy, A. Courville, and Y. Bengio "Generative adversarial nets." *Advances in Neural Information Processing Systems 27 (NIPS 2014)*, pp. 2672-2680. 2014.

[13] E. L. Denton, S. Chintala, A. Szlam, and Rob Fergus. "Deep generative image models using a laplacian pyramid of adversarial networks," *Advances in Neural Information Processing Systems*, 28 (NIPS 2015). 2015.

[14] A. Krizhevsky, I. Sutskever, G. E. Hinton, "Imagenet classification with deep convolutional neural networks," Advances in Neural Information Processing Systems 25 (NIPS 2012), 2012

[15] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," in *Proc. 4th International Conference on Learning Representations (ICLR 2016)*, 2016.

[16] B. Xu, N. Wang, T. Chen, and M. Li, "Empirical evaluation of rectified activations in convolutional network," *Deep Learning Workshop*, ICML 2015

[17] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. the 32nd International Conference on Machine Learning*, Lille, France, 2015. pp. 448–456

[18] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th International Conference on Arti_cial Intelligence and Statistics (AISTATS)*, Fort Lauderdale, FL, USA. 2011.

[19] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous distributed systems," arXiv:1603.04467

[20] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. 3rd International Conference on Learning Representations (ICLR 2015)*, San Diego, 2015

Junghoon Seo is a B.S student in GIST College, Gwangju Institute of Science and Technology, Republic of Korea. He is now majoring in Electrical Engineering and Computer Science. His current research interests includes deep learning, image processing, recommender system, and natural language processing. Mr. Seo is a student member of KIISE.

Taewon Yoon is a B.S student in GIST College, Gwangju Institute of Science and Technology, Republic of Korea. He is now majoring in Electrical Engineering and Computer Science. He will take M.S course in KAIST School of Computing, Korea Advanced Institute of Science and Technology, Republic of Korea.

Jinwoo Kim is a B.S student in GIST College, Gwangju Institute of Science and Technology, Republic of Korea. He is now majoring in Electrical Engineering and Computer Science.

Mr. Kim got the grand prize in Korea Supercomputing Challenge 2014 from KISTI.

Kin Choong Yow is a Professor with the GIST College, Gwangju Institute of Science and Technology, Republic of Korea. He obtained his B.Eng. (Elect) with First Class Honours from the National University of Singapore in 1993, and his Ph.D. from the University of Cambridge, UK, in 1998.

Prior to joining GIST College, he was Professor at the Shenzhen Institute of Advanced Technology, P.R.China, and Associate Professor at the Nanyang Technological University of Singapore. He runs the Generic Intelligence and Smart Environment Lab (GISEL) in GIST, and his current research interests includes cognitive models, biologically inspired vision, artificial consciousness, commonsense reasoning, deep learning, intelligent CCTV systems, and autonomous robot operations in smart environments. Prof. Yow has published over 80 research papers, and has held a number of leadership roles in various international journals and conferences including JAIT, IntJIT, CCGrid2013, ICPADS2012 and MobileHCI2007.