



# Automatic Natural Image Colorization

Tan-Bao Tran<sup>(✉)</sup> and Thai-Son Tran<sup>(✉)</sup>

University of Science, Vietnam National University, Ho Chi Minh City, Vietnam  
baotrankhtn@gmail.com, ttson@fit.hcmus.edu.vn

**Abstract.** We introduce a technique to automatically colorize natural grayscale images that combines both local and global features. Automatic colorization is a hard problem of computer vision and usually requires user interactions such as human-labelled color scribbles or reference images to achieve proper results. Based on Convolutional Neural Networks (CNN), our model is trained in an end-to-end fashion and can process images of any resolution. We improve the model of Iizuka *et al.* [1] by adding edge detection network and adjusting the input of the loss function. We also compare our model with the state of the art and show some improvements. Furthermore, we try colorizing ink wash paintings and achieve a special style.

**Keywords:** Colorization · Convolutional Neural Network · Self-supervised learning

## 1 Introduction

Colorizing grayscale images seems impossible because so much information (two out of three dimensions) has been lost. However, the semantics of the image provides meaningful information such as the sky is typically blue, the clouds are typically white. We can not recover the ground truth color, so we try to produce plausible results. Traditional approaches require significant user interaction to produce plausible colorization results while deep learning approaches have provided automatic methods with outstanding results recently. But deep learning models are still facing the problems of color bleeding and desaturation. The deep learning model of Iizuka *et al.* [1] provides good results and is currently one of the best colorization models; however, the colorized images are not



**Fig. 1.** Colorization results of some natural images

vibrant. Beside local features and global features (classification information) like the model of Iizuka *et al.* [1], we add Canny edge information. Local features are computed from small patches, provide information about low level to middle level of the images. Global features include classification information and Canny edge information. Classification information helps the model to know if the image is about forest or sea, indoors or outdoors, etc., while Canny edge information helps the model to reduce color bleeding over object boundaries. We will show that the model with edge information outperforms the model without edge information (Fig. 1).

Our contributions in this paper are in two areas. First, we adjust the model of Iizuka *et al.* [1] at some layers and introduce edge detection network that helps reduce color bleeding. Second, we change the input of the MSE loss function to make the colorization results more vibrant.

## 2 Related Work

Traditional approaches fall into two categories: scribble-based and transfer-based. Scribble-based methods, introduced by Levin *et al.* [2], require user to input color scribbles then solve a quadratic cost function derived from differences of intensities between a pixel and its neighboring pixels. It is assumed that adjacent pixels with similar luminance should have similar color. Huang *et al.* [3] improve this method by adjusting loss function and images will be processed by edge detection algorithm then each segmented region will be processed by colorization algorithm. Luan *et al.* [4] use the texture similarity to reduce user interactions.

Transfer-based methods require user to provide reference image(s). Welsh *et al.* [5] proposed a technique to colorize grayscale images by creating a set of sample pixels then each pixel in grayscale image will be scanned to find the best matching pixel in the set and transfer chrominance from sample pixel to the target pixel. Finding suitable reference image(s) is difficult for the user so Chia *et al.* [6] proposed a method that allows user to input the semantic label for each object in the scene then automatically download suitable images from the internet and apply colorization algorithm.

Recently, Larson *et al.* [7] proposed automatic colorization method based on deep convolutional neural network. They treat the problem as classification instead of regression and modify VGG-16 model to predict hue and chroma distributions for each pixel. Zhang *et al.* [8] proposed simple CNN architecture but they introduce annealed-mean to keep the vibrancy of the mode while maintaining the spatial coherence of the mean of the distribution. Iizuka *et al.* [1] introduce CNN architecture where global and local features are learned in parallel. They introduce fusion layer to fuse global and local features efficiently to achieve proper colorization results.

In conclusion, traditional methods require significant user interaction for proper colorization results. Automatic methods using CNN do not require any input but still face some problems such as desaturation and color bleeding.

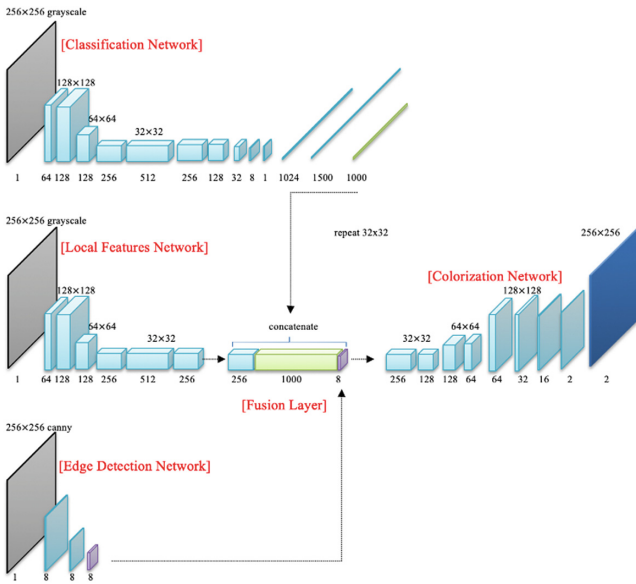
## 3 Approach

We add edge detection network to the CNN model of Iizuka *et al.* [1], adjust some layers and change the input of the loss function. We use CIE Lab and HSV colorspaces, the

input of the model is  $L$  and the output is  $a$  and  $b$  then  $a$  and  $b$  will be fused with  $L$  to form the output image. HSV is used to adjust the saturation of input images. The model is trained on about 200,000 natural images, validated on about 20,000 natural images of **Places** database and tested on natural images database of Massachusetts Institute of Technology. Activation function is ReLU, only the last convolutional layer uses tanh function. To help the model learn complex features and make the computation more effective, we use  $3 \times 3$  filter.

### 3.1 Architecture

There are four main parts in the model: a local features network, a classification network, an edge detection network and a colorization network. The local features, classification information and edge information are fused at “fusion layer”. Fusion layer is introduced in the paper of Iizuka *et al.* [1] and really efficient in colorization problem. We improve the model of Iizuka *et al.* [1] by adding edge detection network and adjusting the input of the loss function as well as some layers (Fig. 2).



**Fig. 2.** Overview of our model for automatic colorization of natural grayscale images

**Classification Network.** The input is  $L$  channel and the output is 1000-dimensional vector. Classification information plays an important role in colorization results; for instance, the green colors of leaves will be different between day and night. However, determining the exact class of the image is not necessary in colorization problem. Iizuka *et al.* [1] suppose there are only 256 classes but we suppose there are 1000 classes in the dataset to make the model more adaptive (Table 1).

**Table 1.** Architecture of classification network

Layer	Filter	Stride	Outputs
Convolution	$3 \times 3$	2	64
Convolution	$3 \times 3$	1	128
Convolution	$3 \times 3$	2	128
Convolution	$3 \times 3$	2	256
Convolution	$3 \times 3$	1	512
Convolution	$3 \times 3$	1	256
Convolution	$3 \times 3$	1	128
Convolution	$3 \times 3$	1	32
Convolution	$3 \times 3$	1	8
Convolution	$3 \times 3$	1	1
Fully-connected	–	–	1024
Fully-connected	–	–	1500
Fully-connected	–	–	1000

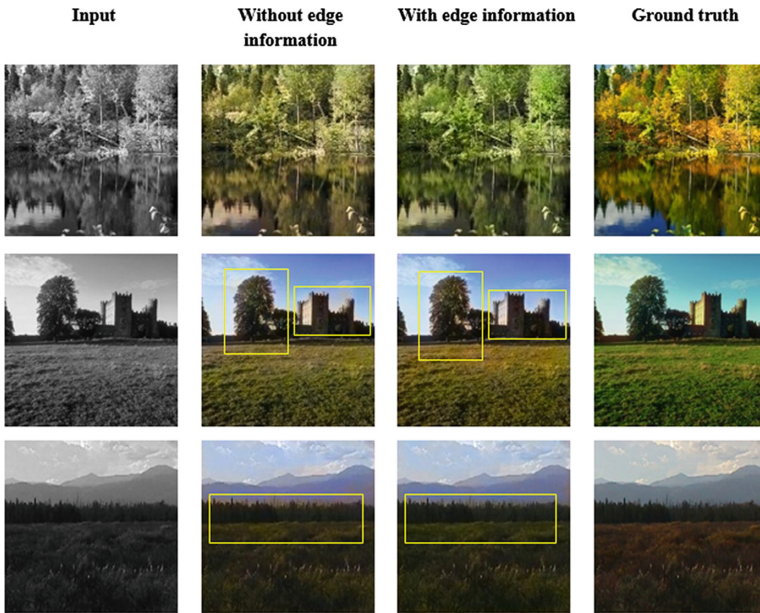
**Edge Detection Network.** The input is canny edge image extracted from grayscale image, the output is just a small  $32 \times 32 \times 8$  matrix because when edge detection network output is bigger it will affect how the model uses colors for objects and make the colorization results desaturated (Table 2).

**Table 2.** Architecture of edge detection network

Layer	Filter	Stride	Outputs
Convolution	$3 \times 3$	2	8
Convolution	$3 \times 3$	2	8
Convolution	$3 \times 3$	2	8

Canny edge is used because it is the most popular algorithm, strictly defined and provides reliable detection. Furthermore, we can adjust the width of the Gaussian, the low and high threshold for the hysteresis thresholding. Edge detection network can reduce color bleeding and make colorization results more vibrant and realistic (Fig. 3).

**Local Features Network.** Low level and middle level features will be extracted at this network. The input is  $L$  channel and the output is  $32 \times 32 \times 256$  matrix. Local features network is responsible for colorizing details in images. This network is similar to the combination of the low-level features network and middle-level features network of Iizuka *et al.* [1] but we eliminate the last two layers of their low-level features network because we train our model on smaller dataset (Table 3).



**Fig. 3.** Colorization results with and without edge information. Color bleeding improvements are in yellow bounding boxes (Color figure online)

**Table 3.** Architecture of local features network

Layer	Filter	Stride	Outputs
Convolution	$3 \times 3$	2	64
Convolution	$3 \times 3$	1	128
Convolution	$3 \times 3$	2	128
Convolution	$3 \times 3$	2	256
Convolution	$3 \times 3$	1	512
Convolution	$3 \times 3$	1	256

**Colorization Network.** The input is the output of fusion layer and the output is  $256 \times 256 \times 2$  matrix. The output represents  $a$  and  $b$  channel. To increase the resolution of the output by a factor of 2, upsampling layers consisting of using the nearest neighbor approach are used (Table 4).

**Fusion Layer.** The output of classification network is 1000-dimensional vector, this will be cloned  $32 \times 32$  times then arranged to form  $32 \times 32 \times 1000$  matrix. Then it will be fused with the output of edge detection network ( $32 \times 32 \times 8$  matrix) and local features network ( $32 \times 32 \times 256$  matrix) to form  $32 \times 32 \times 1264$  matrix.

**Table 4.** Architecture of colorization network

Layer	Filter	Stride	Outputs
Convolution	$3 \times 3$	1	256
Convolution	$3 \times 3$	1	128
Upsampling	–	–	128
Convolution	$3 \times 3$	1	64
Upsampling	–	–	64
Convolution	$3 \times 3$	1	32
Convolution	$3 \times 3$	1	16
Convolution	$3 \times 3$	1	2
Upsampling	–	–	2

### 3.2 Loss Function

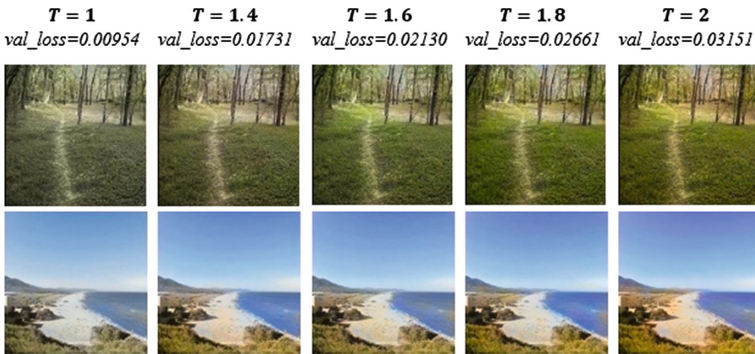
With  $N$  is the number of samples,  $Y = [y_1; y_2; \dots; y_N]$  is the expected matrix with each row represents (a, b) from a ground truth image,  $\hat{Y} = [\hat{y}_1; \hat{y}_2; \dots; \hat{y}_N]$  is the model's output matrix, MSE function used in the paper of Iizuka *et al.* is (1):

$$E = \frac{1}{N} \sum_{i=1}^N \|y_i - \hat{y}_i\|^2 \quad (1)$$

To make colorization results more vibrant, we will use  $Y'$  instead of  $Y$ . We first convert input images to HSV then increase  $S$  by multiplying with a constant  $T$ , shown in (2). Then we convert them back to Lab to get  $Y'$ , so our loss function becomes (3). We found that  $T = 1.8$  provides the best results (Fig. 4).

$$S' = T \times S \quad (2)$$

$$E = \frac{1}{N} \sum_{i=1}^N \|y'_i - \hat{y}_i\|^2 \quad (3)$$

**Fig. 4.** Colorization results affected by T

## 4 Experiments

See Fig. 5.



**Fig. 5.** Some colorization results of our model

### 4.1 Computation Time

Our model is trained on CPU using an Intel Core i5-8400 2.8 GHz with 6 cores, GPU using NVIDIA GeForce RTX 2060 6 GB. Each epoch takes about 6,000 s to complete. It takes about 2.5 s to colorize a  $256 \times 256$  grayscale image and about 3 s to colorize a  $1920 \times 1080$  grayscale image. We reach the best model after 14 epochs.

### 4.2 Compare with Models of Zhang *et al.* and Iizuka *et al.*

We compare our model with the models of Zhang *et al.* [8] and Iizuka *et al.* [1]. Our model performs well on natural grayscale images and can not only reduce color bleeding but also make the results more vibrant and realistic. We do a survey on 45 candidates, ask them to choose the “better” image between the one colorized by our model and the one colorized by the model of Iizuka *et al.* [1] in 50 couples of images. The survey’s

result shows that our model achieves better results on 68% of the couples of images (Fig. 6 and Table 5).



**Fig. 6.** Compare our model with the models of Zhang et al. [8] and Iizuka et al. [1]

We also do another survey on 30 candidates, ask them to recognize which one is the artificial image between the image colorized by our model and the ground truth image in 10 couples of images. The candidates are classified into 3 groups: **Art** (including people who have artistic jobs such as graphic designer, architect), **Technology** (including people who are familiar with digital images) and **Other**. The result shows that the candidates can recognize 5.6 artificial images in 10 artificial images .

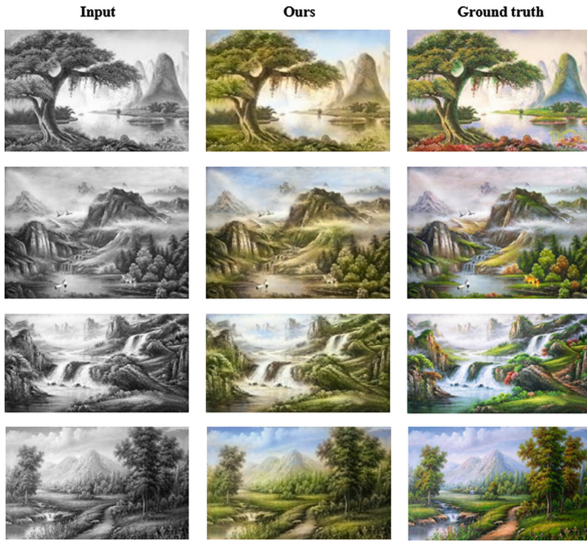
### 4.3 A Try on Ink Wash Paintings

We also try colorizing ink wash paintings and achieve a special style. Ink wash paintings do not require many colors and typically about natural scenes so our model performs well (Fig. 7).



**Table 5.** The result of the survey on how good our model is

Group	The number of candidates	The number of right answers (per 10)
Art	3	4.3
Technology	14	6.1
Other	13	5.4
<b>Total</b>	<b>30</b>	<b>5.6</b>



**Fig. 7.** A try on ink wash paintings

## 5 Conclusion

Our model has improved the model of Iizuka *et al.* [1]. Edge detection network helps the model reduce color bleeding while the adjustment in the input of the loss function helps the model achieve more vibrant colorization results. Our deep learning model can process images of any resolution and is trained in an end-to-end fashion. Our model is applied for natural grayscale images only but this can be extended by enlarging the dataset. Automatic colorization is a challenging and an interesting problem that needs more researches to achieve plausible results.

**Acknowledgements.** This research was supported, in part, by Ngoc Dung Beauty Center. We thank members of Ngoc Dung AI Lab for helpful discussions and Duy-Phu Nguyen for his helpful advice.

## References

1. 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), 1–11 (2016)
2. Levin, A., Lischinski, D., Weiss, Y.: Colorization using optimization. *ACM Trans. Graph.* **23**(3) (2004)
3. Huang, Y.-C., Tung, Y.-S., Chen, J.-C., Wang, S.-W., Wu, J.-L.: An adaptive edge detection based colorization algorithm. In: *ACM International Conference on Multimedia* (2005)
4. Luan, Q., Wen, F., Cohen-Or, D., Liang, L., Xu, Y.-Q., Shum, H.-Y.: Natural image colorization. In: *Eurographics Conference on Rendering Techniques* (2007)
5. Welsh, T., Ashikhmin, M., Mueller, K.: Transferring color to greyscale images. *ACM Trans. Graph.* **21**(3) (2002)
6. Chia, A.-S., et al.: Semantic colorization with internet images. *ACM Trans. Graph.* **30**(6), 1–8 (2011)
7. Larsson, G., Maire, M., Shakhnarovich, G.: Learning representations for automatic colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9908, pp. 577–593. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46493-0\\_35](https://doi.org/10.1007/978-3-319-46493-0_35)
8. Zhang, R., Isola, P., Efros, A.A.: Colorful image colorization. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) *ECCV 2016*. LNCS, vol. 9907, pp. 649–666. Springer, Cham (2016). [https://doi.org/10.1007/978-3-319-46487-9\\_40](https://doi.org/10.1007/978-3-319-46487-9_40)