Deep Patch-wise Colorization Model for Grayscale Images

Xiangguo Liang [†] Zhuo Su [†, *] Yiqi Xiao [†] Jiaming Guo [†] Xiaonnan Luo [‡, §]
[†] School of Data and Computer Science, Sun Yat-sen University
[‡] School of Computer Science and Information Security, Guilin University of Electronic Technology
[§] National Engineering Research Center of Digital Life, Sun Yat-sen University

Figure 1: *Automatic colorization for grayscale images. Top: target grayscale images. Bottom: colorized results of our deep patch-wise clolorization model.*

Abstract

To handle the colorization problem, we propose a deep patch-wise colorization model for grayscale images. Distinguished with some constructive color mapping models with complicated mathematical priors, we alternately apply two loss metric functions in the deep model to suppress the training errors under the convolutional neural network. To address the potential boundary artifacts, a refinement scheme is presented inspired by guided filtering. In the experiment section, we summarize our network parameters setting in practice, including the patch size, amount of layers and the convolution kernels. Our experiments demonstrate this model can output more satisfactory visual colorizations compared with the state-of-the-art methods. Moreover, we prove our method has extensive application domains and can be applied to stylistic colorization.

Keywords: Colorization, Convolutional Neural Network, Image Enhancement

Concepts: •**Computing methodologies** → **Image manipulation;** *Computational photography;*

1 Introduction

Constrained by the photography in the 20th century, amounts of monochrome photos record some memorable moments but regretably lack rich colors, which arouses the demands for color restoration and re-colorization. These demands may be summarized to the image colorization problem that is a process of adding color to grayscale images. Existing colorization methods usually require users' color scribbles or a suitable reference image which needs some elaborate collections. These methods require many sophisticated skills, but may not generate the satisfactory results. In recent years, some researchers began to employ the deep learning models to tackle this task. Their methods may be roughly divided into two categories: local [Cheng et al. 2015] and global [Iizuka et al. 2016] colorization.

In addition, it should not be neglected that the colorizations are often accompanied with many visual artifacts. To pursue artifact-free high quality generation without professional skills, we propose a new automatic data-driven colorization model. In the training process, two loss functions are used alternately to reduce training errors. Since the color diffusive artifacts may appear along the region boundary, we utilize a guided filtering to refine our final result. The experimental results demonstrate the effectiveness of the proposed model and the high visual quality of our colorizations.

Our contributions are as follows. 1) Propose a new deep patch-wise colorization model for grayscale images and design an automatic end-to-end colorization network, which can generate visual satisfactory colors. 2) Alternately use two loss functions to train the model and achieve a better performance. 3) Utilize a refinement scheme with guided filtering to overcome the boundary artifacts, and successfully extend our model to stylistic colorization.

2 Related Work

Scribble-based colorization. [Levin et al. 2004] proposed an effective method that required users to provide color scribbles on the target grayscale image. The color information of scribbles was then propagated to the whole target image by least-square optimization. To relieve the burden on users, [Xu et al. 2013] proposed a sparse control method which automatically determines the influence of edit samples across the whole image jointly considering spatial distance, sample location, and appearance. These methods heavily depend on users' scribbles to get the color information.

Example-based colorization. [Welsh et al. 2002] utilized the pixel intensity and neighborhood statistics to find the similar pixels in reference image and then transferred the color of the matched pixel to the target pixel. Since finding a suitable reference image is difficult, [Chia et al. 2011] proposed an image filter framework to distill suitable reference images from collected Internet images. However, it still needs to provide semantic text labels to search on

*suzhuo3@mail.sysu.edu.cn (corresponding author)

SIGGRAPH Asia 2016 Technical Briefs, December 05-08, 2016, Macao
ISBN: 978-1-4503-4541-5/16/12 $15.00
DOI: http://dx.doi.org/10.1145/3005358.3005375

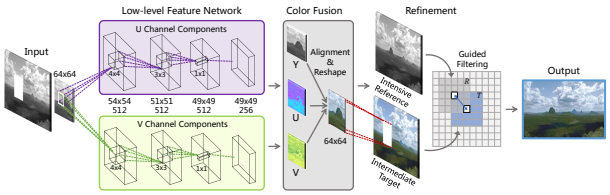


Figure 2: *An overview of our model. Two networks are used to output U and V. Then the recombined image is refined by a guided filter to get the final output.*

the web. [Su et al. 2014] employed a self-learning filter to handle color distortion, grain effect and loss of details.

Deep learning based colorization. In recent years, deep learning has been widely used to solve the colorization problem. [Cheng et al. 2015] extracted the low-level, mid-level and high-level features from grayscale images as the input of fully connected network. [Iizuka et al. 2016] designed a colorization model based on CNN, which took the entire image as a training unit. The methods based on deep learning have some disadvantages in common, such as time-consuming and lack of pertinence, which may cause unsatisfactory colorizations and large computation costs.

3 Patch-wise Colorization Model

We propose our model on the basis of vectorized convolutional neural network (VCNN). It mainly consists of three parts: low-level feature network, color fusion and refinement. An overview of our model and its subcomponents are illustrated in Fig. 2.

3.1 Deep network

Nowadays, CNN has been widely used in vision tasks. Its convolution operation can be typically expressed as

$$y = \sigma\left(W * x + b\right), \tag{1}$$

where $y \in R^m$ and $x \in R^n$. W is a $k \times k$ convolution kernel, b is a bias vector and $*$ is the convolution operator. σ is a nonlinear transfer function, which is ReLU in our model. The most important part of colorization is to extract features from target grayscale image and colorize based on these features, enlightening us to use CNN. Since the inputs of CNN are usually multichannel images, we modify CNN to process the grayscale inputs.

Inspired by [Ren and Xu 2015], we find that VCNN can be a better choice to pursue a faster training speed than traditional CNN. Vectorization refers to the process that transforms the original data structure into a vector representation so that the scalar operators can be converted into a vector implementation, as illustrated in Fig. 3. We implement a vectorized colorization network, whose convolution layer realizes a function in this form:

$$[y_i]_i = \sigma\left([\varphi(x)]_i * [W_i]_i + [b_i]_i\right), \tag{2}$$

where x is the input of convolution layer and y_i is the ith output. W_i is the ith convolution kernel, φ refers to the vectorization operator, and operator $[\,]_i$ is to assemble vectors with index i to form a matrix. This operation is a simplification of extracting matrix from original data or feature map. On the basis of vectorized paradigm, we may reduce the time consumption of the colorization network. We implement our model in YUV color space, where U and V are independent chrominance signals. Therefore, two networks of the same architecture are used to output U and V respectively. Unlike other methods which get U and V in one network [Cheng et al. 2015], our method simplifies the network and improves the accuracy of the results.

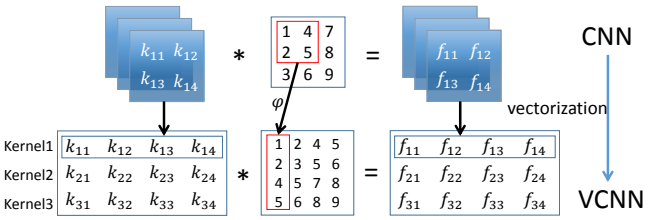


Figure 3: *Convolution operation of VCNN, which transfers the matrix operations into vector operations.*

3.2 Colorization network structure

Deep colorization models could be divided into the following two categories. The local colorization maps color to pixels using the low-, mid- and high-level features [Cheng et al. 2015] while the global colorization incorporates the global features into their models [Iizuka et al. 2016]. In local model, the colorization results of pixels are independent to each other, leading to an unsatisfactory performance. However, it could be a difficult task to evaluate the semantic relationship between neighboring pixels when it comes to global colorization, so the results may not be reasonable. Therefore, the appropriate patch size is a key to the success of our model. We respectively emphasize on the patch sizes of 16x16, 32x32 and 64x64. It turns out that 64x64 outperforms the others. To take the influence of neighboring pixels into consideration, the output of the network should be smaller than the input. Therefore, we set the output as the center 56x56 area of the input 64x64 patch. Under this setting, our model gets a more compact implementation when compared with the global colorization.

Another important network parameter is the number of layers, which is essential to avoid under-fitting and over-fitting. In our model, two networks are used separately to output U and V for reducing the layers of network. Inspired by the effectiveness of VGG network [Simonyan and Zisserman 2014], we build our network based on its architecture. [Ren and Xu 2015] proposed a denoise network that is similar to what we need. However, since the colorization problem is more complicated, we increase the layers in the network, which refers to four convolution layers and a mapping layer. Our experiments demonstrate it performs better in fitting the mapping than other networks of different layers. (Sec. 4.2)

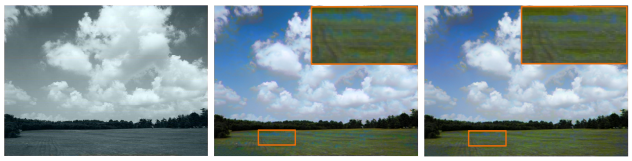
The convolution kernel size and its amount also contribute significantly to the final performance. Based on VGG network architecture and the understanding of convolutional network [Zeiler and Fergus 2014], we conduct experiments on various combinations of kernels. We finally choose our convolutional kernels as a combination of 11x11, 4x4, 3x3 and 1x1. The corresponding amounts are respectively 512, 512, 512 and 256. In the convolution layers, there are sufficient features extracted from the grayscale image to generate satisfactory results.

3.3 Training

We get the predicted $\widehat{U}$ matrix by inputting Y to the network as the initial parameters. The mapping function F can be denoted as

$$\widehat{U} = F\left(Y\right). \tag{3}$$

Training error is the result of subtracting the predicte $\widehat{U}$ matrix from the ground truth U. With the back-propagation algorithm, the training error propagates to all parameters. Then the stochastic gradient descent algorithm is used to update the parameters. To pursue a better training performance, we alternately use two loss functions ($L1$ and $L2$) in the process. At the beginning, we use the following



(a) $L2$ only (b) $L1$ only (c) $L1$+$L2$

Figure 4: *Comparison between different loss functions. It is obvious that merely using $L2$ causes a failure. Combining $L1$ with $L2$ can reduce more artifacts than using $L1$ only.*

loss function:

$$L1(\widehat{U}, U) = \min \sum \left\| \widehat{U} - U \right\|_2. \quad (4)$$

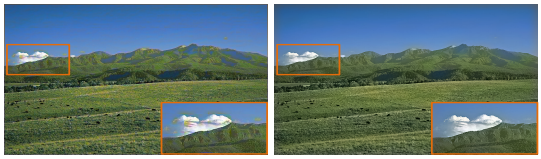
It computes the minimal sum of L_2-norm of the difference between $\widehat{U}$ and U. Then the training error will propagate through the network by BP algorithm. Since this loss function aims to optimize the global error, after 100 iterations, we change our loss function to

$$L2(\widehat{U}, U) = \min \sum U \log(\widehat{U}). \quad (5)$$

The loss function focuses on minimizing the local error. It can be a complementary to the previous one for improvement. Therefore, we use $L1$ to train the network in the first 100 iterations. When the training errors would not decrease any more, we change the loss function to $L2$ in the next 100 iterations. The experiment result is illustrated in Fig. 4.

3.4 Refinement scheme

We obtain U and V by neural networks. They are then combined with Y to reform the color image. However, it can be seen that there are some artifacts at the image edges. To remove the visible artifacts, we use guided filter in the refinement scheme and make full use of the target grayscale image by choosing it as the guidance. The input of the guided filter [He et al. 2013] is the output colorization of neural network, a color image. This scheme can help to reduce artifacts and preserve edges. We set the filter size as 40 and the edge-preserving parameter $1e^{-4}$. From the experimental results, most of the artifacts are suppressed after the refinement.



(a) Before image refinement (b) After image refinement

Figure 5: *Image refinement using guided filtering. After being refined, most of the artifacts have been removed.*

4 Experimental Results and Discussion

In this section, we introduce the experimental setting of our colorization network, and compare our results with the state-of-the-art methods to validate our performance. We also successfully apply the proposed method to stylistic colorization.

4.1 Dataset

The previous methods usually choose big datasets as the training input. However, the results reveal that they typically work well in colorizing images of common types. In ordre to reduce the uncertainty of colorizations, we train our model on small databases to be capable of coloring images whose type is consistent with training data. We select 10^3 high quality images of outdoor scenes and extract image patches from them. As a result, the colored images in this paper would be similar to the outdoor scenes in colors. The patches are segmented into 21 files, 20 of which are used for training and one for validating. In each file, there are 10^4 Y-U pairs.

Table 1: *Three network architectures with different convolution kernels used in our experiments.*

Network1		Network2		Network3	
size	number	size	number	size	number
16x16	512	11x11	512	14x14	512
1x1	512	6x6	512	3x3	512
1x1	256	1x1	256	1x1	256

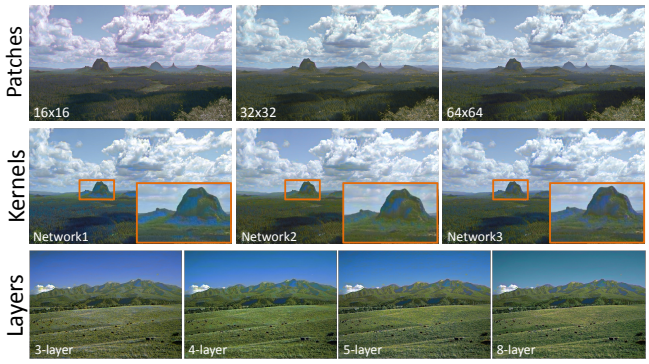


Figure 6: *Experiment on network design and parameter setting, including the patch size, amount of layers and the convolution kernels.*

4.2 Network design and parameter setting

Numerous experiments are conducted to determine the network parameter setting. We explore the optimal patch size by training our network on image patches of different sizes, and the results are shown in the first row of Fig. 6. When the patch size is 16x16, the colorization is unnatural with many artifacts. The reason is the patch size is so small that there are insufficient pixel combinations for training, increasing the probability of underfitting. As the patch size increases, it can be seen that the color is more natural and the artifacts are reduced. However, if the patch size is as large as 128x128, the complexity of this task is similar to global colorization and the result may be unreasonable. For the best performance, we finally set the patch size as 64x64.

The amount of layers is another important parameter. Based on VGG network structure, we adjust the number of layers to compare the corresponding network performances. As illustrated in the third row of Fig. 6, we respectively experiment on the networks of 3-layer, 4-layer, 5-layer and 8-layer (including the last layer which is an identical mapping). Comparing these four colorizations, we can see the results of both 3-layer and 8-layer networks are not very close to the natural scenery. 4-layer and 5-layer networks could generate more natural colorizations, but the grass colors of 5-layer are more hierarchical, indicating the network is sensitive to slight changes in images. Therefore, our network consists of 5 layers, which are 4 convolution layers and a mapping layer.

We seek to understand the role of convolution kernels by choosing various kernels but keeping the patch size and number of layers unchanged. Table 1 presents three network structures and the second row of Fig. 6 are the experimental results. Network1 produces more

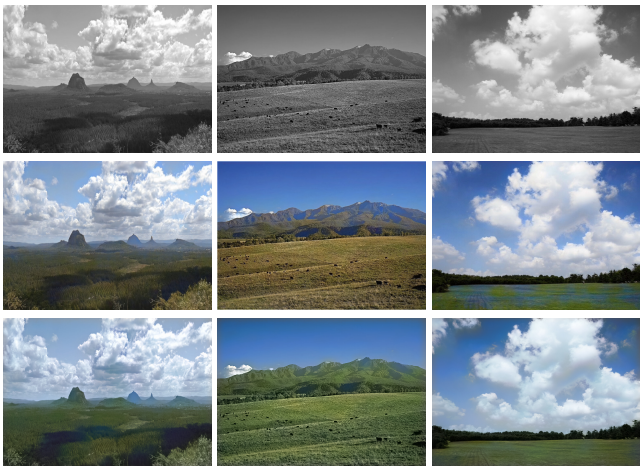


Figure 7: *Compared with [Iizuka et al. 2016]. Row 2 shows the results of [Iizuka et al. 2016] while Row 3 presents ours.*

artifacts than the other two, which demonstrates the features extracted by 16x16 kernels are not effective enough. The performance of Network2 and Network3 are roughly the same. They can both output more natural colorizations. In VGG network, small convolution kernels show better quality in extracting image features. Combining our experiments and the prior, we set our convolution kernels as a combination of 11x11, 4x4, 3x3 and 1x1 kernels. From previous work, we learn that when the amount of kernels reaches 512, it is able to extract sufficient features. The amount of kernels in the four convolution layers are 512, 512, 512, 256 respectively.

4.3 Result analysis and stylization rendering

The state-of-the-art colorization method [Iizuka et al. 2016] declared that it can be successfully applied to various types of images. This method fuses local and global features together to output the final colorization while we only utilize local features. In their implementation, the entire image is taken as a unit for training and colorizing. By contrast, our model adopts a patch-wise implementation. Patch-level evaluation may establish more exact color mapping to achieve visual satisfactory result and runtime performance. We compare our results with theirs, and the comparisons are illustrated in Fig. 7. It can be seen that the colors of our forests and grass are more vivid and natural. We extend our model to stylistic colorization (Fig. 8). If the training dataset consists of images in the same style, the colorized results will be closer to this style, which gives the model the ability to colorize images in various styles.

5 Conclusion

In this paper, we propose a deep patch-wise colorization model with alternative loss metrics under VCNN framework. During the colorizing process, image patches are extracted with an 8-pixel overlap to make the output a complete image. If the target image is black-and-white, we could use histogram equalization to get its grayscale version as input, which makes the performance better than inputing a black-and-white image.

Acknowledgements

This work is supported by the Natural Science Foundation of China (No. 61502541, 61320106008, 61370186), the Natural Science Foundation of Guangdong Province (No. 2016A030310202), and the Fundamental Research Funds for the Central Universities (Sun Yat-sen University).

Figure 8: *Stylistic colorization. Left: bronze style. Middle: retro style. Right: bright style. We use three styles of images to train the network respectively. Results demonstrate that our model can be successfully applied to stylistic colorization.*

References

CHENG, Z., YANG, Q., AND SHENG, B. 2015. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision*, 415–423.

CHIA, A. Y.-S., ZHUO, S., GUPTA, R. K., TAI, Y.-W., CHO, S.-Y., TAN, P., AND LIN, S. 2011. Semantic colorization with internet images. *ACM Transactions on Graphics 30*, 6, 156.

HE, K., SUN, J., AND TANG, X. 2013. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence 35*, 6, 1397–1409.

IIZUKA, S., SIMO-SERRA, E., AND ISHIKAWA, H. 2016. 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 35*, 4.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. *ACM Transactions on Graphics 23*, 3, 689–694.

LUAN, Q., WEN, F., COHEN-OR, D., LIANG, L., XU, Y.-Q., AND SHUM, H.-Y. 2007. Natural image colorization. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, Eurographics, 309–320.

REN, J. S., AND XU, L. 2015. On vectorization of deep convolutional neural networks for vision tasks. In *29th AAAI Conference on Artificial Intelligence*, 25–30.

SIMONYAN, K., AND ZISSERMAN, A. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

SU, Z., ZENG, K., LIU, L., LI, B., AND LUO, X. 2014. Corruptive artifacts suppression for example-based color transfer. *IEEE Transactions on Multimedia 16*, 4, 988–999.

WELSH, T., ASHIKHMIN, M., AND MUELLER, K. 2002. Transferring color to greyscale images. vol. 21, ACM, 277–280.

XU, L., YAN, Q., AND JIA, J. 2013. A sparse control model for image and video editing. *ACM Transactions on Graphics 32*, 6, 197.

XU, L., REN, J. S., LIU, C., AND JIA, J. 2014. Deep convolutional neural network for image deconvolution. In *Advances in Neural Information Processing Systems*, 1790–1798.

ZEILER, M. D., AND FERGUS, R. 2014. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision*, Springer, 818–833.