Twin Deep Convolutional Neural Network for Example-based Image Colorization

Domonkos Varga[1,2] and Tamás Szirányi[1,3]

[1] MTA SZTAKI, Institute for Computer Science and Control
{varga.domonkos, sziranyi.tamas}@sztaki.mta.hu
[2] Budapest University of Technology and Economics, Department of Networked Systems and Services
[3] Budapest University of Technology and Economics, Department of Material Handling and Logistics Systems

Abstract. This paper deals with the colorization of grayscale images. Recent papers have shown remarkable results on image colorization utilizing various deep architectures. Unlike previous methods, we perform colorization using a deep architecture and a reference image. Our architecture utilizes two parallel Convolutional Neural Networks which have the same structure. One CNN, which uses the reference image, helps the other CNN in color prediction for the input image. On the other hand, the second CNN, which uses the input image, helps to identify the areas which holds essential information about the color scheme of the scene. Comprehensive experiments and qualitative and quantitative evaluations were conducted on the images of SUN database and on other images. Quantitative evaluations are based on Peak Signal-to-Noise Ratio (PSNR) and on Quaternion Structural Similarity (QSSIM).

Keywords: image colorization, deep learning, convolutional neural network

1 Introduction

Automatic image colorization examines the problem how to add realistic colors to grayscale images without any user intervention. It has some useful applications such as colorizing old photographs or movies, artist assistance, visual effects and color recovering. On the other hand, colorization is a heavily ill-posed problem. In order to effectively colorize any images, the algorithm or the user should have enough information about the scene's semantic composition.

As pointed out in [16], image colorization is also a good model for a huge number of applications where we want to take an arbitrary image and predict values or different distributions at each pixel of the input image, exploiting information only from this input image. This is a very common task in the image processing and pattern recognition community.

To date, deep learning techniques have shown impressive results on both high-level and low-level vision problems including image classification [1], removing phantom objects from point clouds [2], pedestrian detection [3], face detection [4], handwritten character classification [5], photo adjustment [6], etc. In recent years, deep learning based approaches appeared to address the colorization problem.

Main contributions. Image colorization algorithms can be divided into three classes: scribble-based, example-based, and learning-based. In this paper, we show a possible solution that utilizes the advantages of example-based and learning-based approaches. Unlike previous methods, we perform colorization using a deep architecture and a reference image.

Paper organization. This paper is organized as follows. In Section 2, the related and previous works are reviewed primarily focused on learning-based approaches. We describe our algorithm in Section 3. Section 4 shows experimental results and analysis. The conclusions are drawn in Section 5.

2 Related works

Image colorization has been intensively studied since 1970's. Broadly speaking, the existing algorithms can be divided into three groups: scribble-based, example-based, and learning-based approaches. In this section, we mainly concentrate on reviewing learning-based approaches.

Scribble-based approaches interpolate colors in the grayscale image based on color scribbles produced by a user or an artist. Levin et al. [7] presented an interactive colorization method which can be applied to still images and video sequences as well. The user places color scribbles on the image and these scribbles are propagated through the remaining pixels of the image. Huang et al. [8] improved further this algorithm in order to reduce color blending at image edges. Yatziv et al. [9] developed the algorithm of Levin et al. [7] in another direction. The user can provide overlapping color scribbles. Furthermore, a distance metric was proposed to measure the distance between a pixel and the color scribbles. Combinational weights belong to each scribbles which were determined based on the measured distance.

Example-based approaches require two images. These algorithms transfer color information from a colorful reference image to a grayscale target image. Reinhard et al. [10] applied simple statistical analysis to impose one image's color characteristics on another. Welsh et al. [11] utilized on pixel intensity values and different neighborhood statistics to match the pixels of the reference image with the pixels of grayscale target image. On the other hand, Irony et al. [12] determine first for each pixel which example segment it should learn its color from. This carried out by applying a supervised classification algorithm that considers the low-level feature space of each pixel neighborhood. Then each color assignment is treated as color micro-scribbles which were the inputs to Levin et al.'s [7] algorithm. Charpiat et al. [13] predicted the expected variation of color at each pixel, thus defining a non-uniform spatial coherency criterion. Then graph cuts were applied to maximize the probability of the whole colored image at the global level. Gupta et al. [14] extracted features from the target and reference images at the resolution of superpixels. Based on different kind of features, the superpixels of the reference image were matched with the superpixels of the target image and the color information was transfered to the center of the superpixels of the target image with the help of micro color-scribbles. Then these micro-scribbles were propagated through the target image.

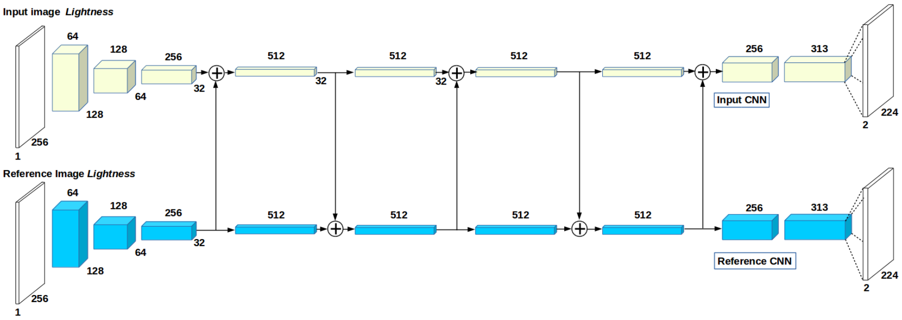

Fig. 1: The architecture of the proposed method. The input and the reference CNN have the same structure. First, only the reference CNN is trained then the input CNN and the reference CNN are trained simultaneously. Information is transmitted from input CNN to reference CNN and vica versa using element-wise addition operator to certain convolutional blocks.

Learning-based approaches model the variables of the image colorization process by applying different machine learning techniques and algorithms. Bugeau and Ta [15] introduced a patch-based image colorization algorithm that takes square patches around each pixel. Patch descriptors of luminance features were extracted in order to train a model and a color prediction model with a general distance selection strategy was proposed. Deshpande et al. [16] colorize an image by optimizing a linear system that considers local predictions of color, spatial consistency, and consistency with an overall histogram. Cheng et al. [17] introduced a fully-automatic method based on a deep neural network which was trained by hand-crafted features. Three levels of features were extracted from each pixel of the training images: raw grayscale values, DAISY features [18], and high-level semantic features.

In recent years, Convolutional Neural Network based approaches appeared to tackle the colorization problem. Iizuka et al. [19] elaborated a colorization method that jointly extracts global and local features from an image and then merge them together. In [20], the authors proposed a fully automatic algorithm based on VGG-16 [21] and a two-stage Convolutional Neural Network to provide richer representation by adding semantic information from a preceding layer. Furthermore, the authors proposed Quaternion Structural Similarity [22] for quality evaluation. Zhang et al. [23] trained a Convolutional Neural Network to map from a grayscale input to a distribution of quantized color values. This algorithm was evaluated with the help of human participants asking them to distinguish between colorized and ground-truth images. In [24], the authors introduced a patch-based colorization model using two different loss functions in a vectorized Convolutional Neural Network framework. During colorization patches are extracted from the image and are colorized independently. Guided image filtering [25] is applied as postprocessing. Larsson et al. [26] processed a grayscale image through VGG-16 [21] architecture and obtained hypercolumns [27] as feature vectors. The system learns to predict *hue* and *chroma* distributions for each pixel from its hypercolumn. Deshpande et al. [28] proposed a conditional model for predicting multiple colorizations. The low dimensional embedding of color fields was learned by a Variational Autoencoder. Similarly, Cao et al. [29] worked with a conditional model but a Conditional Generative Adversarial Network was utilized to model the distribution of real-world colors. Limmer and Lensch [30] proposed a method for transferring the RGB color spectrum to near-infrared images using deep multi-scale convolutional neural networks. The transfer between RGB and near-infrared images is trained.

3 Our approach

The objectiveness of our framework is to combine example-based and learning-based approaches in order to produce more realistic and plausible colors. To capitalize on the advantages of example-based and learning-based methods as well, we propose a novel architecture which is shown in Figure 1. Our architecture consists of two parallel CNNs which are called Input CNN and Reference CNN. These have the same structure. In the following, this structure is firstly described and then the co-operation of the two networks is discussed.

We reimplemented the algorithm of [23] using Keras [31] deep learning library. This algorithm has some appealing properties. First of all, the authors elaborated a *class rebalancing* method because the distribution of ab values in natural images is biased towards low ab values. Second, colorization is treated as multinomial classification instead of regression. This means that the ab output space is quantized into bins with grid size 10 and keep the Q = 313 values which are in gamut. For all details, we refer to [23].

We used SUN database [32] to compile our training database. We denote a reference image by R and an input image by I. Formally, our database can be defined as $\mathcal{L}_i = \{(I_i, R_i) | i = 1, ..., N\}$ where N is the number of image pairs and reference image R_i is semantically similar to input image I_i. That is why we opted to utilize SUN database [32] since this dataset contains images grouped by their semantic information. Figure 2 shows the empirical distribution of pixels in ab space gathered from our database. Figure 3 illustrates the empirical and smoothed empirical distribution of ab pairs in the quantized space. These curves were determined and were applied in the training process based on the algorithm of [23].

First, we train only the Reference CNN using only the R_i's from our database. We utilize *ADAM* optimizer [33] and early stopping [34] with the following parameters: $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $d = 0.0$, and $\varepsilon = 1e-8$ where

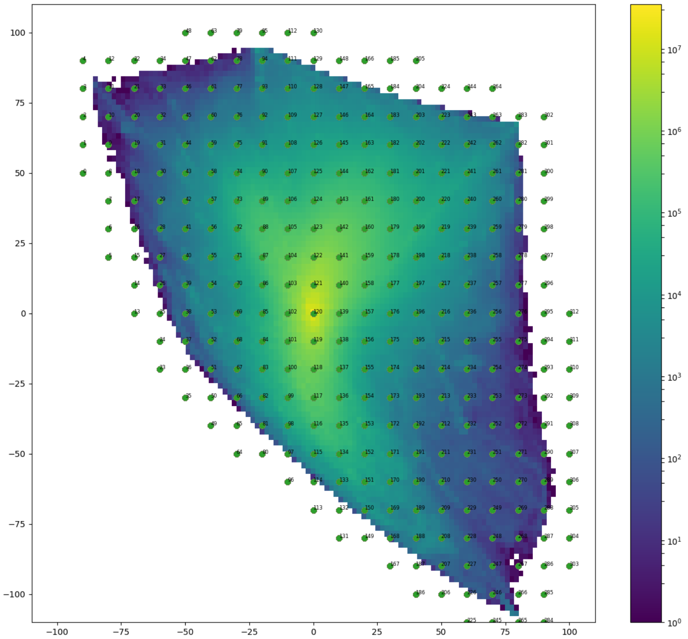

Fig. 2: Empirical probability distribution of *ab* values in our database, shown in *log* scale. The horizontal axis represents the *b* values and the vertical axis represents the *a* values. The green dots denote the quantized *ab* value pairs.

α is the learning rate, ε is the fuzz factor, and d is the learning rate decay over each update. Then the input CNN and the reference are trained simultaneously using the whole $\mathcal{L}_i = \{(I_i, R_i)|i = 1, ..., N\}$ database. As we mentioned the input and the reference CNN have the same structure. Information is transmitted from input CNN to reference CNN and vica versa using element-wise addition operator to certain convolutional blocks (see Figure 1). The image pairs $(I_i, R_i)_{i=1}^{N}$ are given to the input of the two CNNs. The values of the third convolutional block in the Reference CNN are added element-wise to those in the Input CNN. Next, the values of the fourth convolutional block in the input CNN are added to those in the Reference CNN. This process repeats to the second last convolutional block. In this process, we also applied ADAM optimizer and early stopping with the above mentioned parameters. In this way, the color information of the reference image is applied to facilitate the color prediction for the input image. On the other hand, information from the input image helps to identify the areas which

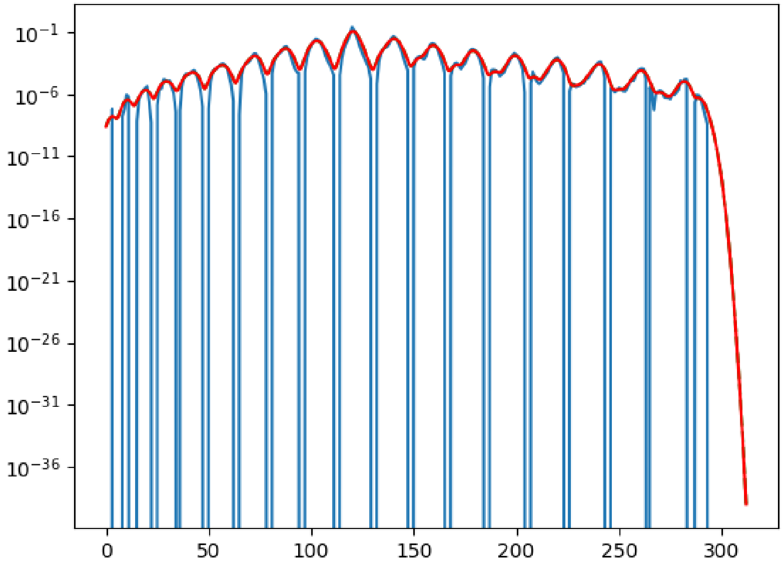

Fig. 3: Empirical (blue curve) and smoothed empirical distribution (red curve) of *ab* pairs in the quantized space of our cartoon database.

holds essential information about the color scheme of the scene. The proposed framework was trained on 60.000 image pairs of the SUN database.

As pointed out in many papers [20], [23], [24], [26], Euclidean loss function is not an optimal solution because it will result in the so-called averaging problem. Namely, the system will produce grayish sepia tone effects. That is why we use a cross-entropy like loss function to compare predicted $\hat{\mathbf{Z}} \in [0,1]^{H \times W \times Q}$ against the ground truth $\mathbf{Z} \in [0,1]^{H \times W \times Q}$:

$$L(\hat{\mathbf{Z}}, \mathbf{Z}) = - \sum_{h=1,w=1}^{H,W} v(\mathbf{Z}_{h,w}) \sum_{q=1}^{Q=313} \mathbf{Z}_{h,w,q} \cdot log(\hat{\mathbf{Z}}_{h,w,q}), \tag{1}$$

where $Q = 313$ is the number of quantized ab values (see Figure 2), $v(\cdot)$ is a weighting term used to rebalance the loss based on color-class rarity, and H and W denote the height and the width of the training images. The weighting term $v(\cdot)$ is obtained using the smoothed empirical distribution of ab pairs in the quantized space (see Figure 3). For all details of the weighting term, we refer to [23].

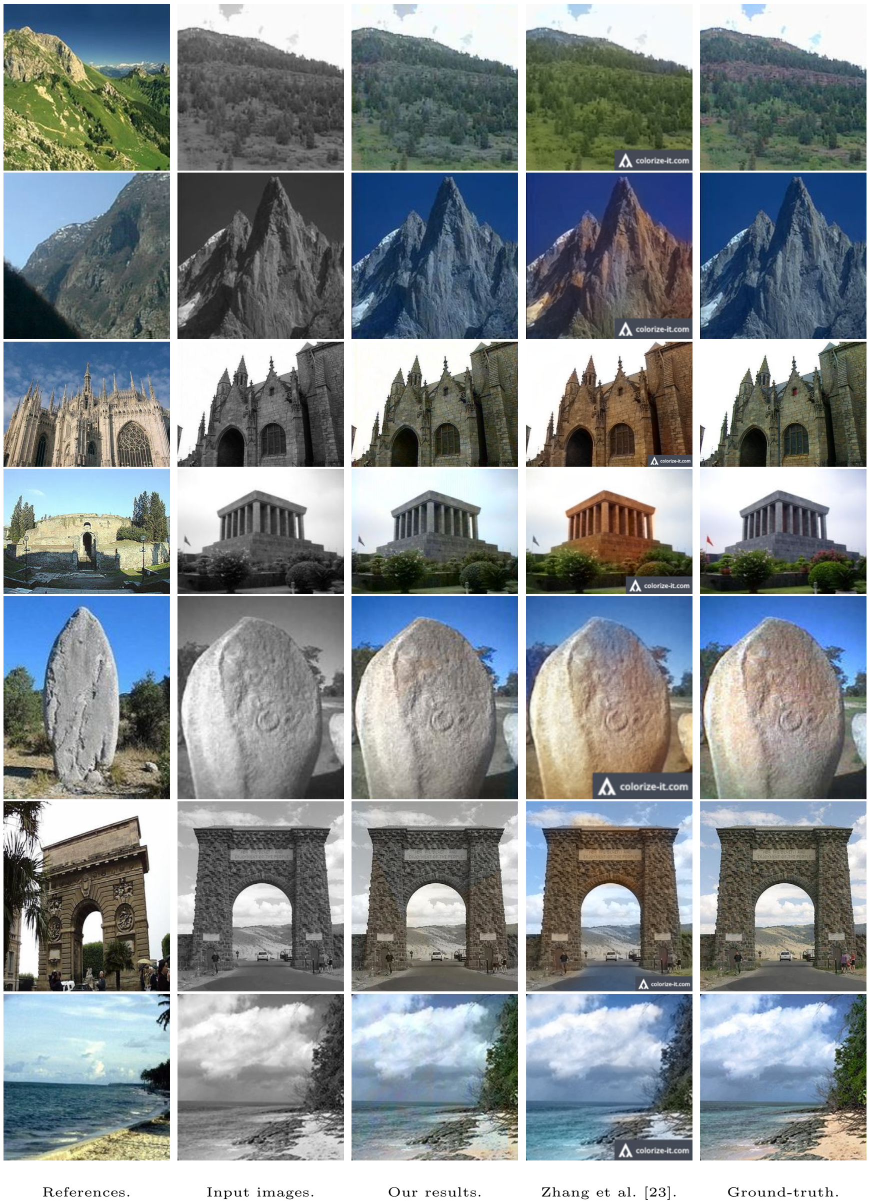

Fig. 5: Colorized results. The first image is the reference image, the second is the grayscale input, the third is our colorized result, and fourth is the result of [23], and the fifth is the ground-truth image. Digital watermarks in the lower right corners were embedded by the application of [23] (available: http://demos.algorithmia.com/colorize-photos).

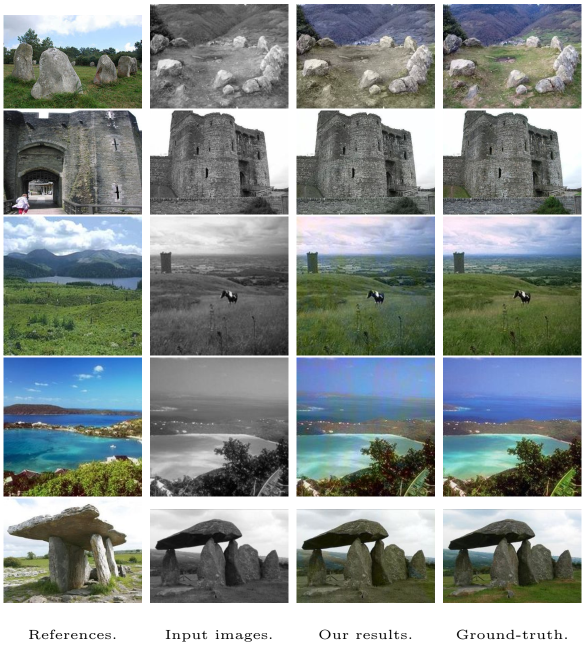

Fig. 7: Colorized results.

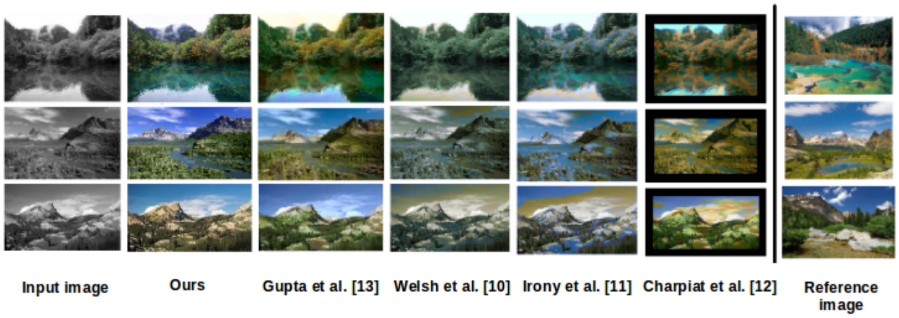

Fig. 8: Comparison with state-of-the-art example-based colorization algorithms.

4 Experimental results

Figure 5 presents several colorization results obtained by our proposed method with respect to the inputs, the ground-truth colorful images, and the reference images. Figure 5 also illustrates the results of [23] which were obtained using their web application (available: http://demos.algorithmia.com/colorize-photos). Note that the digital watermarks in the lower right corners were embedded by this application. From this qualitative comparison, we can see that our method is able to reduce visible artifacts, especially for detailed scenes, ob-

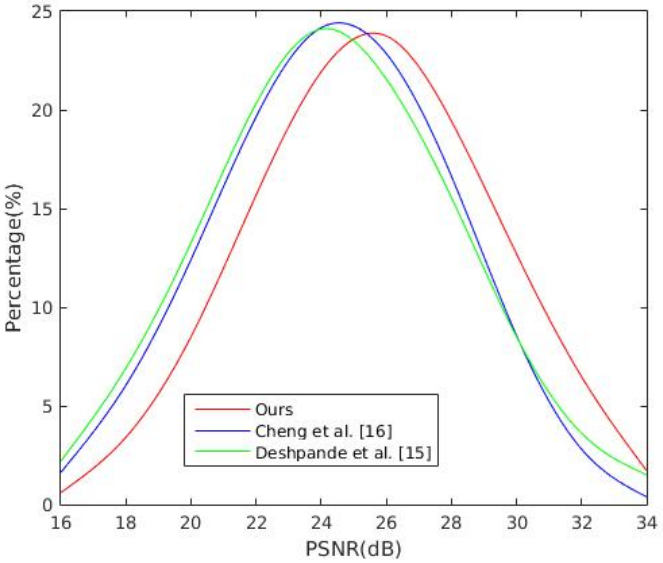

Fig. 9: Peak Signal-to-Noise Ratio (PSNR) distribution. It can be seen that the proposed method can improve the colorization accuracy.

jects with large color variances (*e.g.* building). The color filling is nearly flawless. We could reduce the amount of false edges near the object boundaries. Figure 7 shows further results of our method.

Figure 8 shows a comparison with the major state-of-the-art example-based colorization algorithms such as [11], [12], [13], and [14]. It can be seen that we could produce more realistic and plausible colors than most state-of-the-art example-based colorization algorithms.

Figure 9 presents the Peak Signal-to-Noise Ratio (PSNR) distribution of our method, Cheng et al. [17], and Deshpande et al. [16]. We have measured the PSNR distribution on 1500 test images from the SUN database [32]. Note that we reimplemented for this experiment the method of [17] using Keras deep learning library [31]. In our experiment, we have applied a 33-dimensional semantic feature vector for [17] and have trained the proposed deep neural network architecture using ADAM optimizer [33] and the images of SUN database. Besides, we have used the source code (available: http://vision.cs.illinois.edu/projects/lscolor) provided by Deshpande et al. [16]. Figure 9 illustrates that the proposed method is able to improve colorization accuracy since it outperforms these two state-of-the-art algorithms.

Unfortunately, there is no widely used quality metrics which clearly indicates the quality of a colorization. Methodical quality evaluation by showing colorized images to human observers is slow, expensive, and subjective. Empirically, we have found that Quaternion Structural Similarity (QSSIM) [22] gives a good base

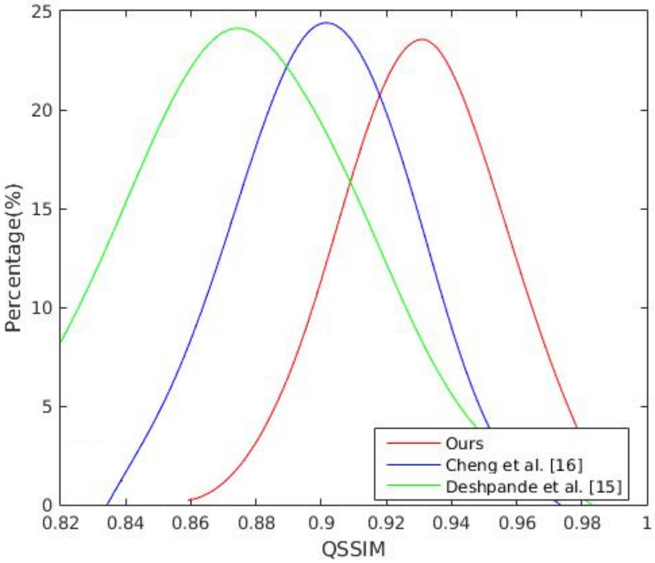

Fig. 10: Quaternion Structural Similarity (QSSIM) distribution. It can be seen that the proposed method can improve the colorization quality. A higher QSSIM value indicates better image quality.

for quantitative evaluation. It is a theoretically well based measure which has been accepted by the colorimetry research community as a potential qualification value. We have measured the QSSIM distribution on 1500 test images from the SUN database. Figure 10 presents the QSSIM distribution of our method, Cheng et al. [17], and Deshpande et al. [16]. It can be seen that the proposed method outperforms the two other state-of-the-art algorithms. A higher QSSIM values indicates better image quality. This experiment was based on the source code (available: http://www.ee.bgu.ac.il/˜kolaman/QSSIM) provided by Kolaman et al. [22].

5 Conclusion

In this paper, we have introduced a novel framework which capitalizes on the advantages of example-based and learning-based colorization approaches. Specifically, we have shown a possible solution that combines the information between two CNNs in order to help the input CNN in color prediction for the input image. To this end, we have trained first a reference CNN which facilitates the identification of the specific color scheme of the input scene. We have shown that the semantic enhancement capability of a deep CNN can be switched into a colorization scheme to result in an effective image analysis and interpretation framework. The QSSIM method has been proved a superior measuring method for color modeling. There are many directions for further research. First, it is worth to generalize the proposed method for arbitrary size input images. Another direction of research would be automatizing the search for a suitable reference image to an input image.

Acknowledgment

The research was supported by the Hungarian Scientific Research Fund (No. OTKA 120499). We are very thankful to Levente Kovács for helping us with professional advices in high-performance computing.

References

1. A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 1097–1105, 2012.
2. B. Nagy and C. Benedek. 3D CNN Based Phantom Object Removing from Mobile Laser Scanning Data. *International Joint Conference on Neural Networks*, 4429–4435, 2017.
3. E. Bochinski, V. Eiselein, and T. Sikora. Training a convolutional neural network for multi-class object detection using solely virtual world data. *IEEE International Conference on Advanced Video and Signal Based Surveillance*, 278–285, 2016.
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*, **8**(1): 98–113, 1997.
5. D. Ciresan and U. Meier. Multi-column deep neural networks for offline handwritten Chinese character classifiction. *Proceedings of the International Joint Conference on Neural Networks*, 1–6, 2015.
6. Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep learning. *CoRR*, abs/1412.7725, 2014.
7. A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics*, **23**(3): 689–694, 2004.
8. Y.C. Huang, Y.S. Tung, J.C. Chen, S.W. Wang, and J.L. Wu. An adaptive edge detection based colorization algorithm and its applications. *Proceedings of the 13th annual ACM international conference on Multimedia*, 351–354, 2005.
9. L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing*, **15**(5): 1120–1129, 2006.
10. E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, **21**(5): 34–41, 2001.
11. T. Welsh, M. Ashikhmin, and K. Mueller. Transfering color to greyscale images. *ACM Transactions on Graphics*, **21**(3): 277–280, 2002.
12. R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. *Eurographics Symp. on Rendering*, 2005.
13. G. Charpiat, M. Hofmann, and B. Schölkopf. Automatic image colorization via multi-modal predictions. *European Conference on Computer Vision*, 126–139, 2008.
14. R.K. Gupta, A.Y.S. Chia, D. Rajan, E.S. Ng, and H. Zhiyong. Image colorization using similar images. *Proceedings of the 20th ACM international conference on Multimedia*, 369–378, 2012.

15. A. Bugeau and V.T. Ta. Patch-based image colorization. *Proceedings of the IEEE International Conference on Pattern Recognition*, 3058–3061, 2012.
16. A. Deshpande, J. Rock, and D. Forsyth. Learning large-scale automatic image colorization. *Proceedings of the IEEE International Conference on Computer Vision*, 567–575, 2015.
17. Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. *Proceedings of the IEEE International Conference on Computer Vision*, 415–423, 2015.
18. E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **32**(5): 815–830, 2010.
19. 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 (TOG)*, **35**(4): 110, 2016.
20. D. Varga and T. Szirányi. Fully automatic image colorization based on Convolutional Neural Network. *International Conference on Pattern Recognition*, 2016.
21. K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.
22. A. Kolaman and O. Yadid-Pecht. Quaternion structural similarity: a new quality index for color images. *IEEE Transactions on Image Processing*, **21**(4): 1526–1536, 2012.
23. R. Zhang, P. Isola, and A.A. Efros. Colorful image colorization. *European Conference on Computer Vision*, 649–666, 2016.
24. X. Liang, Z. Su, Y. Xiao, J. Guo, and X. Luo. Deep patch-wise colorization model for grayscale images. *SIGGRAPH ASIA 2016 Technical Briefs*, 13, 2016.
25. K. He, J. Sun, and X. Tang. Guided image filtering. *European Conference on Computer Vision*, 1–14, 2010.
26. G. Larsson, M. Michael, and G. Shakhnarovich. Learning representations for automatic colorization. *European Conference on Computer Vision*, 577–593, 2016.
27. B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik. Hypercolumns for object segmentation and fine-grained localization. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 447–456, 2015.
28. A. Deshpande, J. Lu, M.C. Yeh, and D. Forsyth. Learning Diverse Image Colorization. *arXiv preprint arXiv:1612.01958*, 2016.
29. Y. Cao, Z. Zhou, W. Zhang, and Y. Yu. Unsupervised Diverse Colorization via Generative Adversarial Networks. *arXiv preprint arXiv:1702.06674*, 2017.
30. M. Limmer and H. Lensch. Infrared Colorization Using Deep Convolutional Neural Networks. *arXiv preprint arXiv:1604.02245*, 2016.
31. F. Chollet. Keras. *https://github.com/fchollet/keras*, 2015.
32. J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN Database: Large-scale Scene Recognition from Abbey to Zoo. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 3485–3492, 2010.
33. D. Kingma and J. B. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.
34. F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural computation*, **7**(2): 219–269, 1995.