2016 23rd International Conference on Pattern Recognition (ICPR)
Cancún Center, Cancún, México, December 4-8, 2016

Fully automatic image colorization based on Convolutional Neural Network

Domonkos Varga*†, Tamás Szirányi*‡
*MTA SZTAKI, Institute for Computer Science and Control
{varga.domonkos, sziranyi.tamas}@sztaki.mta.hu
†Budapest University of Technology, Department of Networked Systems and Services
‡Budapest University of Technology, Department of Material Handling and Logistics Systems

***Abstract*—This paper deals with automatic image colorization. This is a very difficult task, since it is an ill-posed problem that usually requires user intervention to achieve high quality. A fully automatic approach is proposed that is able to produce realistic colorization of an input grayscale image. Motivated by the recent success of deep learning techniques in image processing, we propose a feed-forward, two-stage architecture based on Convolutional Neural Network that predicts the U and V color channels. Unlike most of the previous works, this paper presents a fully automatic colorization which is able to produce high-quality and realistic colorization even of complex scenes. Comprehensive experiments and qualitative and quantitative evaluations were conducted on the images of SUN database and on other images. We have found that Quaternion Structural Similarity (QSSIM) gives in some degree a good base for quantitative evaluation, that is why we chose QSSIM as an index-number for the quality of colorization.**

I. Introduction

Automatic image colorization deals with the problem of adding colors to monochrome images without any user intervention. Colorization has some practical applications such as colorizing old movies or photographs, color recovering, artist assistance and visual effects. On the other hand, automatic image colorization is a good model for many problems. There are 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.

To date, deep learning techniques have shown impressive results on both high-level and low-level vision problems. Researchers achieved impressive results in image classification [1], pedestrian detection [2], face detection [3], pedestrian tracking [4], handwritten character classification [5], image super-resolution [6], photo adjustment [7], etc. The goal of this paper is to present our novel automatic image colorization algorithm based on Convolutional Neural Network (CNN). Like [8], we formulate image colorization as a regression problem and CNNs are used to solve this problem.

Main contributions. We propose a novel fully automatic image colorization algorithm using the VGG-16 model [22] and Convolutional Neural Network (CNN). A feed-forward, two-stage architecture based on CNN is presented that predicts the U and V color channels.

Paper organization. This paper is organized as follows. In Section II, the related and previous works are reviewed. We describe the proposed CNN-based automatic colorization algorithm in Section III. Section IV shows experimental results and analysis. We draw the conclusions in Section V.

II. Related works

Existing works on image colorization can be broadly divided into three classes: scribble-based approaches, example-based approaches, and learning-based approaches. The first approach attempts to interpolate colors based on color scribbles provided by an artist. The example-based algorithms try to transfer the color information from a reference image to a target grayscale image. Unlike example-based approaches, the learning-based methods learn the variables of image colorization modeling in order to overcome the limitations of manual assignments.

Scribble-based colorization Levin et al. [9] introduced an interactive colorization technique that can be applied to both still images and image sequences. The user provides the information how each region should be colored by placing color scribbles in the image. The color information of the scribbles are then propagated to the remaining pixels of the target image. This algorithm improved by Huang et al. [10] in order to reduce color blending at image edges. Yatziv et al. [11] introduced a method which combines the color information of multiple scribbles to determine the color of a pixel. A distance metric was proposed in order to measure the distance between the pixel and the scribbles. The combination weights of the particular scribbles were calculated based on the measured distance.

Example-based colorization Reinhard et al. [12] proposed a color transfer algorithm based on statistical analysis to impose one image's color characteristics on another. Similarly, Welsh et al. [13] applied pixel instensity and neighborhood statistics to find similar pixels in a reference image and then transfer the color information to a matched pixel of the target grayscale image. Irony et al. [14] determined first for each pixel of the target image which example segment it should learn its color from, then the authors treated them as user scribble input for colorization. Charpiat et al. [15] 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.

978-1-5090-4847-2/16/$31.00 ©2016 IEEE

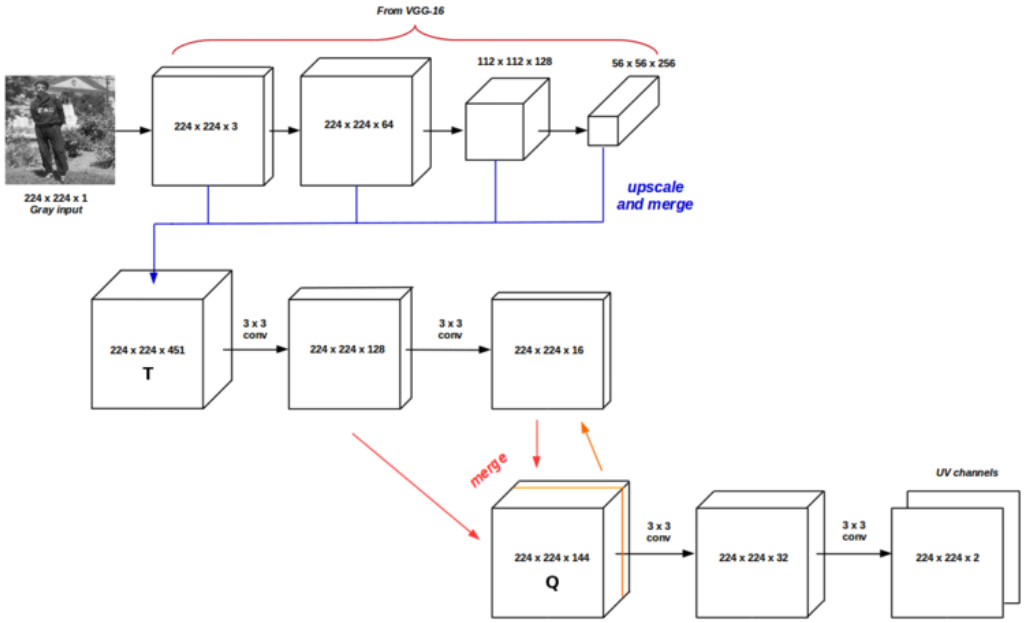


Fig. 1: The architecture of our proposed automatic image colorization algorithm. We have used in our work the pretrianed VGG-16 CNN model [22], since it incorporates a huge amount a semantic information.

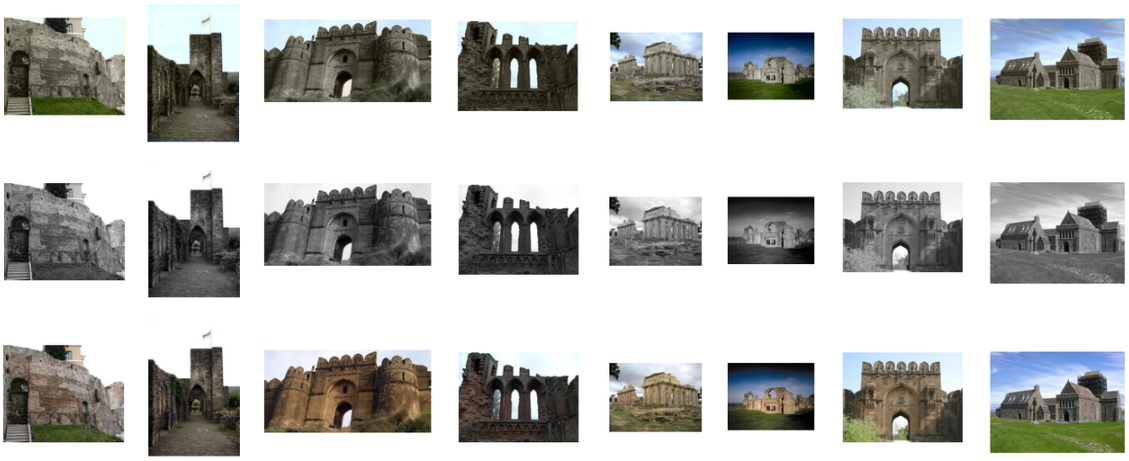


Fig. 2: Colorized results. In the first row the output of the proposed algorithm can be seen. The second row presents the grayscale input images. In the third row the ground-truth colorful images can be seen.

Learning-based colorization Bugeau and Ta [16] 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. Cheng et al. [8] proposed a fully-automatic colorization method based on three-layer deep neural network and hand-crafted features. Three levels of features were extracted from each pixel of the training images: raw grayscale values, DAISY features [17], and high-level semantic features. Then these feature were concatenated and used to train a deep neural network. Ryan Dahl [18] proposed a CNN-based approach and utilized a pretrained CNN for image classification [22] as a feature extractor. Then, a trained residual encoder was applied that provides color channels. The predicted colors are rational for the most part, although the system is heavily prone to desaturate or to caramelize images. We think that the Euclidean loss function is not susceptible in this case and we propose instead of Euclidean loss function a cross entropy like loss function.

The main disadvantage of scribble-based approaches is that the performance heavily depends on human intervention. Example-based colorization works well only if we were able to find appropriate color image that is given as one of the inputs to the algorithm. Our approach is most related to [8] in the sense that a training dataset is composed and a neural

(a) Deshpande et al. [24]

(b) Our proposed system.

(c) Ground-truth images.

Fig. 3: The comparison with the state-of-the-art algorithm of Deshpande et al. [24] for the castle and the abbey category of SUN database [23].

(a) Chen et al. [8]

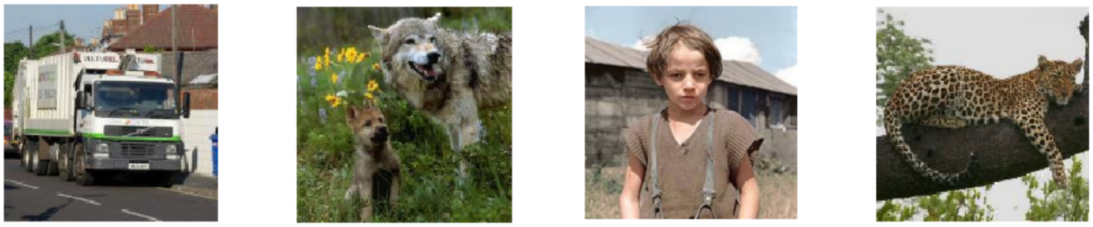
(b) Our proposed system.

(c) Ground-truth images.

Fig. 4: The comparison with the state-of-the-art algorithm of Chen et al. [8].

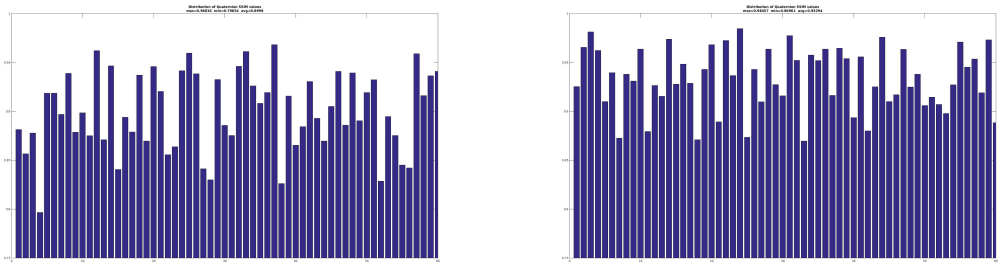
(a) Algorithm of Dahl [18].

(b) Our proposed system.



(c) Ground-truth images.

Fig. 5: The comparison with the state-of-the-art algorithm of Dahl [18].



(a) Deshpande et al. [24]

(b) Our proposed system.

Fig. 6: Quantitative comparison with the state-of-the-art algorithm of Deshpande et al. [24]. 60 images were selected randomly from the abbey, castle, and cathedral outdoor categories of SUN [23] database. Statistics of our method: Max.: 0.98457, Min.: 0.86961, Avg.: 0.93294. Statistics of Deshpande et al. [24]: Max.: 0.96816, Min.: 0.79654, Avg.: 0.8998.

network is trained in order to predict U and V values. Unlike [8], we extract the semantic information in a fully automatic way.

III. Our approach

An overview of the proposed colorization algorithm can be seen in Figure 1. As pointed out in [8], semantic information has an important and unavoidable role in colorization. In order to effectively colorize any images, the system should have information about the semantic composition of the scene and its localization. For example, the color of leaves on a tree may be some kind of green in spring, but they could be brown in autumn. That is why we have used in our work the pretrained VGG-16 CNN model [22] with slight modifications. This model incorporates a huge amount a semantic information, since it was trained on ILSVRC 2012 classification dataset which consists of more than 1 million images.

The input of VGG-16 is a fixed-sized 224×224 RGB image. Since the input in our case is a single-channel grayscale image, the input grayscale image is concatenated one after another three times. We have extracted several layers from VGG-16 as seen in the top of Figure 1. The VGG-16 was divided into

Fig. 7: Limitations: If the VGG-16 part of the proposed system is unable to resolve correctly or nearly correctly the semantic information of the input image, our algorithm tends to blur the image with some kind of sepia tone. Another drawback is that the proposed algorithm tends to transfer the color of big semantic parts to the surrounding subtler details, but with different semantics. Left: ground-truth images. Right: Colorized results.

two parts thataway the first part contains all the layers which can be found before the third pooling layer of VGG-16. We used only the first part and the second part was ignored.

The layers of the first part were upscaled to the input size of VGG-16 and were concatenated. This process has resulted in a $224 \times 224 \times 451$-size matrix which we denote by $\mathbf{T}$. In the bottom of Figure 1, the procedure can be seen that converts matrix $\mathbf{T}$ into U and V channels.

In this paper, we used Multi-Stage features in order to obtain richer representaion by adding complementary information from a preceding layer. Multi-Stage features have improved performance in other works [19], [20], [21] and in this work as well. The matrix $\mathbf{T}$ is followed by two convolutional layers and the pooling layers were missed out, because we want to obtain 224×224 resolution. The outputs of these two convolutional layers are concatenated and this process results in a $224 \times 224 \times 144$-size matrix which is denoted by $\mathbf{Q}$. Similarly, the matrix $\mathbf{Q}$ is followed by two convolutional layers and the ouputs of the second convolutional layer are the predicted U and V channels. In the backpropagation algorithm, we use from matrix $\mathbf{Q}$ that part that corresponds to the second convolutional layer in the first stage (the $224 \times 224 \times 16$-size matrix in Figure 1).

Given the training set $\mathcal{L} = \{(\mathbf{UV}_i, \mathbf{Y}_i)\}_{i=1}^{N}$, where $\mathbf{UV}$ denotes the target U and V values, and $\mathbf{Y}$ is the corresponding lightness channel. We used the old YUV standard, since it had been developed to add color channels to the Luminance value (Y), why this color space may fit the most to the present approach of finding the additional colors. Furthermore, YUV minimizes the correlation between the three coordinate axes. The analysis of the effects and advantages or drawbacks of different color spaces is beyond the scope of this paper. For the present, we are pleased with the above assumption.

As we mentioned, the input of the system will be $\mathbf{Y}$ which is used to create matrix $\mathbf{T}$ by computing feedforward VGG-16 then concatenating and upscaling the layers. Matrix $\mathbf{T}$ serves as input of the two-stage CNN described in the previous paragraphs. We train and modify the convolutional weights only this two-stage CNN, where the target is $\mathbf{UV}_i$ and the input is $\mathbf{T}$.

In the training process, Stochastic Gradient Descent algorithm was used with a learning rate of 8×10^{-2}, a weight decay of 8×10^{-7}, and a step decay of 0.5 for every 300 epochs. We trained our model for 1000 epochs. We learn a mapping $\mathcal{F}$ that predicts the U and V values:

$$\mathbf{UV}_p = \mathcal{F}(\mathbf{T}), \tag{1}$$

where $\mathbf{T}$ is the matrix obtained with the help of VGG-16, and $\mathbf{UV}_p$ consists of the predicted U and V values. We tested different loss functions because we found that the Euclidean distance produced desaturated results. That is why we then used a cross entropy like loss function:

$$L(\mathbf{UV}_p, \mathbf{UV}) = -\frac{1}{W \cdot H} \sum \mathbf{UV} \cdot log(\mathbf{UV}_p), \tag{2}$$

where $\mathbf{UV}$ contains the real U and V values, W and H stands for the width and height of the input image - in our case both of them equals to 224. We have trained our network on ILSVRC 2012 classification dataset. 500.000 images were selected randomly from this database then we converted the training color images into greyscale images. For each image the U and V channels were calculated, respectively.

IV. Experimental results and Evaluation metric

Figure 2 presents several colorization results obtained by the proposed method with respect to the inputs and ground-truth colorful images. Figure 3 shows a comparison with the method of Deshpande et al [24]. It can be seen that we could produce more realistic and less desaturated color images. The ground-truth images were taken from the abbey and castle categories of the SUN database [23]. Similarly, Figure 4 presents a comparison with the state-of-the-art algorithm of Cheng et al [8]. Note that we did not use any post-processing procedure like [8].

Methodical quality evaluation, by showing colorized images to human observers, is slow, expensive, and very subjective. Another reason to ignore human observers was that neither the Euclidean loss functions nor cross-entropy like loss functions are suitable to express the humans subjective opinion about the plausibility or rationality of the predicted colors. It is another direction of research to evaluate the loss functions in this regard or to find out how to back-propagate the errors in the CNN with respect to the opinion of human observers.

Unfortunately, there is no exact index-number which clearly indicates the quality of a colorization. Using obvious examples, where the result is definitely fixed by natural laws: e.g. images of natural scenes, containing blue sky, vegetation, grass, people (skin color) and some animals. In that case numerical evaluation may have a good value.

The simplest and most widely used quality metrics are the mean squared error (MSE) and the peak signal-to-noise-ratio (PSNR). It may occur easily that colorized images have roughly the same MSE or PSNR values with respect to the ground-truth image, but very different quality. Empirically, we have found that Quaternion Structural Similarity (QSSIM) [25] gives in some degree a good base for quantitative evaluation. QSSIM is a theoretically well based measure, which has been accepted by the color research community as a potential qualification value. We remark that in most cases the higher QSSIM values indicates better quality and more reasonable colors in colorization but the range of QSSIM proved non-linear and it sometimes reflects inconsistently the differences in quality. We selected randomly 60 images from the abbey, castle, and cathedral outdoor categories of the SUN [23] database. These images were converted to greyscale and then colorized using our algorithm and the algorithm of Deshpande et al. [24]. Figure 6 shows the QSSIM values of the images in a column chart. It can be seen that our QSSIM values (Max.: 0.98457, Min.: 0.86961, Avg.: 0.93294) outperforms the values of Deshpande et al. [24] (Max.: 0.96816, Min.: 0.79654, Avg.: 0.8998).

Finally, we compared our results with the algorithm of Dahl [18] and Figure 5 shows the comparison. It can be seen that we could produce slightly lifesomer colors.

If the VGG-16 part of the proposed system is unable to resolve correctly or nearly correctly the semantic information of the input image, our algorithm tends to blur the image with some kind of sepia tone. Another drawback is that the proposed algorithm tends to transfer the color of big semantic parts to the surrounding subtler details, but with different semantics (see Figure 7).

V. Conclusion

This paper introduced a novel, fully automatic colorization algorithm based on VGG-16 and a two-stage CNN. VGG-16 provided multiple discriminative, sematic information which was used to train a two-stage CNN architecture without pooling layers. The two-stage architecutre proved us a richer representation by adding information from a preceding layer. The U and V color channels were predicted because the YUV space had been developed to add color channels to the Luminance value (Y), why this color space may fit the most to the present approach of finding the additional colors. In addition to this, YUV minimizes the correlation between the three coordinate axes. Comprehensive qualitative and quantitative experiments were conducted on SUN database. The limitations of the algorithm were pointed out as well.

Acknowledgment

The research was supported by the Hungarian Scientific Research Fund. 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] Y. Tian, P. Luo, X. Wang, and X. Tang. Deep learning strong parts for pedestrian detection. *Proceedings of the IEEE International Conference on Computer Vision*, 1904–1912, 2015.

[3] 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.

[4] D. Varga, T. Szirányi, A. Kiss, L. Spórás, and L. Havasi. A multi-view pedestrian tracking method in an uncalibrated camera network. *Proceedings of the IEEE International Conference on Computer Vision Workshops*, 37–44, 2015.

[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. Wang, D. Liu, J. Yang, W. Han, and T. Huang. Deep networks for image super-resolution with sparse prior. *Proceedings of the IEEE International Conference on Computer Vision*, 370–378, 2015.

[7] Z. Yan, H. Zhang, B. Wang, S. Paris, and Y. Yu. Automatic photo adjustment using deep learning. *CoRR*, abs/1412.7725, 2014.

[8] Z. Cheng, Q. Yang, and B. Sheng. Deep colorization. *Proceedings of the IEEE International Conference on Computer Vision*, 415–423, 2015.

[9] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics*, **23**(3): 689–694, 2004.

[10] 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.

[11] L. Yatziv and G. Sapiro. Fast image and video colorization using chrominance blending. *IEEE Transactions on Image Processing*, **15**(5): 1120–1129, 2006.

[12] E. Reinhard, M. Ashikhmin, B. Gooch, and P. Shirley. Color transfer between images. *IEEE Computer Graphics and Applications*, **21**(5): 34–41, 2001.

[13] T. Welsh, M. Ashikhmin, and K. Mueller. Transfering color to greyscale images. *ACM Transactions on Graphics*, **21**(3): 277–280, 2002.

[14] R. Irony, D. Cohen-Or, and D. Lischinski. Colorization by example. *Eurographics Symp. on Rendering*, 2005.

[15] G. Charpiat, M. Hofmann, and B. Schölkopf. Automatic image colorization via multi-modal predictions. *Computer Vision-ECCV 2008*, 126–139, 2008.

[16] A. Bugeau and V.T. Ta. Patch-based image colorization. *Proceedings of the IEEE International Conference on Pattern Recognition*, 3058–3061, 2012.

[17] 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.

[18] Ryan Dahl. http://tinyclouds.org/colorize/.

[19] P. Sermanet, S. Chintala, and Y. LeCun. Convolutional neural networks applied to house numbers digit classification. *Proceedings of the International Conference on Pattern Recognition*, 3288–3291, 2012.

[20] P. Sermanet, K. Kavukcuoglu, and Y. LeCun. Traffic signs and pedestrians vision with multi-scale convolutional networks. *Snowbird Machine Learning Workshop*, **2**(3): 1–8, 2011.

[21] P. Sermanet and Y. LeCun. Traffic sign recognition with multi-scale convolutional networks. *The 2011 International Joint Conference on Neural Networks*, 2809–2813, 2011.

[22] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *CoRR*, abs/1409.1556, 2014.

[23] J. Xiao, J. Hays, K. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 3485–3492, 2010.

[24] 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.

[25] 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.