Automatic Cartoon Colorization Based on Convolutional Neural Network

Domonkos Varga
MTA SZTAKI, Institute for Computer Science and Control
Kende u. 13-17.
Budapest, Hungary
varga.domonkos@sztaki.mta.hu

Csaba Attila Szabó
Budapest University of Technology and Economics, Department of Networked Systems and Services
Magyar tudósok krt. 2.
Budapest, Hungary
szabo@hit.bme.hu

Tamás Szirányi
MTA SZTAKI, Institute for Computer Science and Control
Kende u. 13-17.
Budapest, Hungary
sziranyi.tamas@sztaki.mta.hu

ABSTRACT

This paper deals with automatic cartoon colorization. This is a hard issue, since it is an ill-posed problem that usually requires user intervention to achieve high quality. Motivated by the recent successes in natural image colorization based on deep learning techniques, we investigate the colorization problem at the cartoon domain using Convolutional Neural Network. To our best knowledge, no existing papers or research studies address this problem using deep learning techniques. Here we investigate a deep Convolutional Neural Network based automatic color filling method for cartoons.

CCS CONCEPTS

• **Computing methodologies** → *Image processing*;

KEYWORDS

Colorization, Cartoon Colorization, Convolutional Neural Network

ACM Reference format:
Domonkos Varga, Csaba Attila Szabó, and Tamás Szirányi. 2017. Automatic Cartoon Colorization Based on Convolutional Neural Network. In *Proceedings of CBMI, Florence, Italy, June 19-21, 2017,* 6 pages.
https://doi.org/10.1145/3095713.3095742

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.

Automatic cartoon colorization is a more difficult task than automatic natural image colorization because the drawer's or the designer's individual style implies an additional factor in the ill-posed problem. Consequently, we put the emphasis on to create plausible colorization that is convincing and aesthetic for a human observer. In this paper, we introduce a cartoon colorization method using deep learning techniques to produce plausible colorization of black-and-white cartoons.

Main contributions. This paper deals with automatic colorization of cartoons using Convolutional Neural Network. To our best knowledge, no existing papers or research studies addresses the problem of cartoon colorization using deep learning techniques.

Paper organization. This paper is organized as follows. In Section 2, the related and previous works are reviewed. 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. The existing algorithms can be divided into three classes.

Scribble-based approaches interpolate colors in the grayscale image based on color scribbles produced by a user. Levin et al. [15] 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. [9] improved further this algorithm in order to reduce color blending at image edges. Yatziv et al. [25] developed the algorithm of Levin et al. [15] 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. [18] applied simple statistical analysis to impose one image's color characteristics on another. Welsh et al. [24] 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. [11] 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 [15] algorithm. Charpiat et al. [2] predicted the expected variation of color at each pixel, thus defining a non-uniform spatial

CBMI, June 19-21, 2017, Florence, Italy

ACM ISBN 978-1-4503-5333-5/17/06…$15.00
https://doi.org/10.1145/3095713.3095742

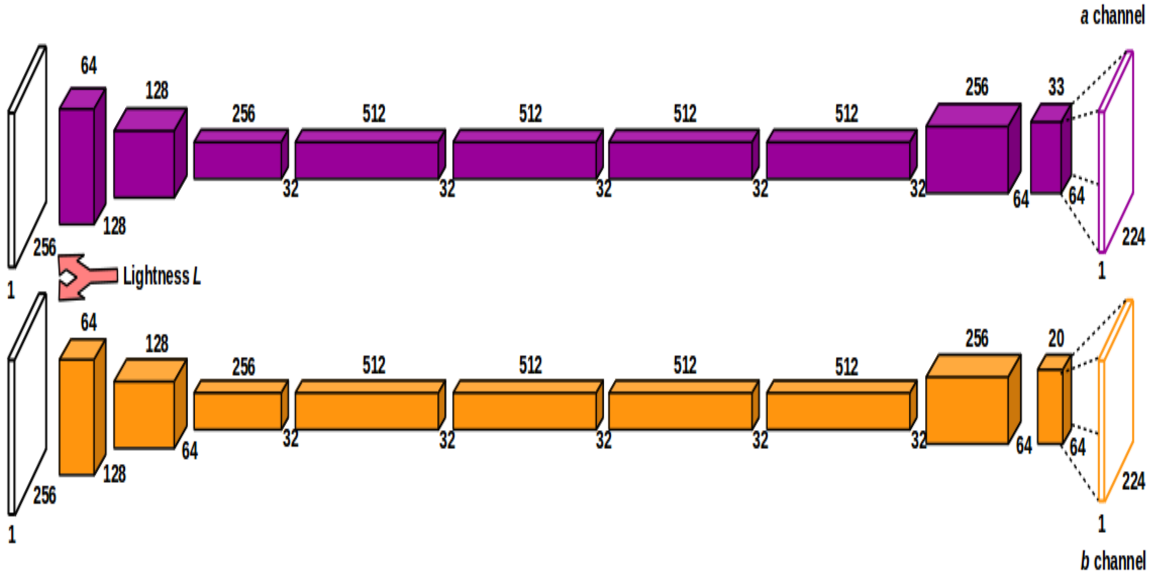

Figure 1: The architecture of the implemented system.

coherency criterion. Then graph cuts were applied to maximize the probability of the whole colored image at the global level.

Learning-based approaches model the variables of the image colorization process by applying different machine learning techniques and algorithms. Bugeau and Ta [1] 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. [3] 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 [22], and high-level semantic features.

In recent years, Convolutional Neural Network based approaches appeared to solve the colorization problem. Iizuka et al. [10] elaborated a colorization method that jointly extracts global and local features from an image and then merge them together. In [23], the authors proposed a fully automatic algorithm based on VGG-16 [19] 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 [13] for quality evaluation. Zhang et al. [26] 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 [16], 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 colorized independently. Guided image filtering [8] is applied as postprocessing. Larsson et al. [14] processed a grayscale image through VGG-16 [19] architecture and obtained hypercolumns [7] as feature vectors. The system learns to predict *hue* and *chroma* distributions for each pixel from its hypercolumn.

Cartoon colorization algorithms fall into scribble-based or example-based approaches. To our knowledge, no existing paper deals with cartoon colorization using deep learning.

Sykora et al. [20] modeled the dynamic part of a scene by a set of outlined homogeneous regions which covers the static background. The authors developed an unsupervised segmentation algorithm for black-and-white cartoon animations able to produce segmentation. Qu et al. [17] proposed a method similar to Levin's method [15] but Gabor wavelet filters were applied to measure pattern-continuity. The algorithm is initialized by a curve at the user-provided color scribbles and evolves until it achieves boundaries of regions of interest. The progression of the moving facade depends on local and global features as well.

3 OUR APPROACH

We reimplemented the algorithm of [26] with modifications using Keras [4] deep learning library. The algorithm of [26] has some appealing properties which makes it ideal for cartoon colorization. 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. The problem is very similar in the case of cartoons. 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. Figure 2 shows the empirical distribution of

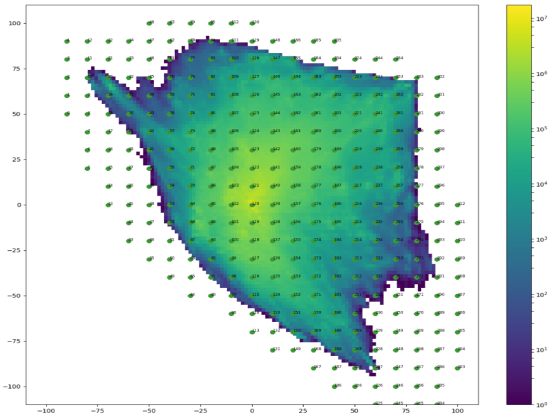

Figure 2: Empirical probability distribution of *ab* values in our cartoon database, shown in *log* scale. The horizontal axis represents the *b* values and the vertical axis represents the *a* values. The green dots denote the quantized *ab* value pairs.

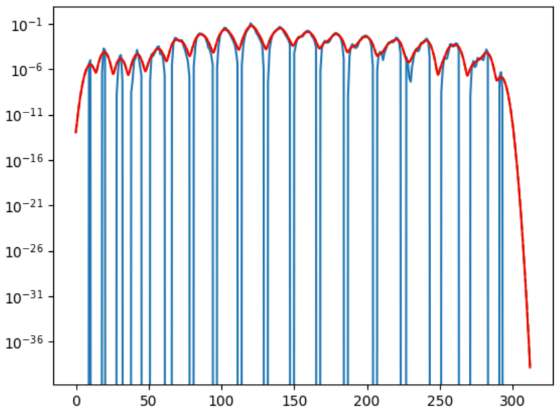

Figure 3: Empirical (blue curve) and smoothed empirical distribution (red curve) of *ab* pairs in the quantized space of our cartoon database.

pixels in *ab* space, gathered from our cartoon database which consists of 100,000 images. It can be clearly seen that this distribution significantly differs from the distribution of natural images.

Unlike [26], we train two CNNs for *a* and *b* channels in order to boost performance (see Figure 1). Given an input $\mathbf{G} \in \mathbb{R}^{H \times W \times 1}$ grayscale image (H is the input image's height, W is the width), one CNN learns a mapping $\hat{\mathbf{A}} = \mathcal{F}_1(\mathbf{G})$ while the other CNN learns $\hat{\mathbf{B}} = \mathcal{F}_1(\mathbf{G})$. As pointed out in many papers, Euclidean loss function is not an optimal solution because this 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 $\hat{\mathbf{A}}$ against the ground truth $\mathbf{A}$:

$$L(\hat{\mathbf{A}}, \mathbf{A}) = -\sum_{h=1, w=1}^{H,W} v(\mathbf{Z}_{h,w}) \sum_{q=1}^{Q_1=33} \mathbf{A}_{h,w,q} \cdot log(\hat{\mathbf{A}}_{h,w,q}) \quad (1)$$

where $\mathbf{A} \in [0,1]^{H \times W \times Q_1}$ and $\hat{\mathbf{A}} \in [0,1]^{H \times W \times Q_1}$ is quantized to $Q_1 = 33$ values (see Figure 2), $v(\mathbf{Z}_{h,w})$ stands for the weighting term that is used to re-balance the loss function with respect to the color distribution, and $\mathbf{Z} \in [0,1]^{H \times W \times Q}$ is used for to search the nearest quantized *ab* bin, where $Q = 313$ is the number of quantized *ab* value pairs. The loss function of the other CNN is exactly same but we predict the *b* values instead of *a* values:

$$L(\hat{\mathbf{B}}, \mathbf{B}) = -\sum_{h=1, w=1}^{H,W} v(\mathbf{Z}_{h,w}) \sum_{q=1}^{Q_2=20} \mathbf{B}_{h,w,q} \cdot log(\hat{\mathbf{B}}_{h,w,q}) \quad (2)$$

where $\mathbf{B} \in [0,1]^{H \times W \times Q_2}$ and $\hat{\mathbf{B}} \in [0,1]^{H \times W \times Q_2}$ is quantized to $Q_2 = 20$ values (see Figure 2) and the meaning of the other terms is the same as in Eq. 1. Based on the algorithm of [26], each pixel is weighted by $\mathbf{w} \in \mathcal{R}^Q$, with respect to its closest *ab* bin:

$$v(\mathbf{Z}_{h,w}) = \mathbf{w}_{q^*}, \text{ where } q^* = \arg\max_q \mathbf{Z}_{h,w,q} \quad (3)$$

$$\mathbf{w} \propto ((1-\lambda)\tilde{\mathbf{p}} + \frac{\lambda}{Q})^{-1}, \quad (4)$$

where $\tilde{\mathbf{p}}$ is the smoothed empirical distribution which is obtained from the empirical distribution of colors in the quantized *ab* space with a Gaussian kernel $\mathbf{G}_\sigma$. We use in our experiments the following values: $\lambda = \frac{1}{2}$ and $\sigma = 5$. Figure 3 shows the empirical distribution and the smoothed empirical distribution.

As we mentioned our cartoon database contains 100,000 images. Table 1 shows the main sources of our cartoon database. We take out frames from the listed cartoons by the table. In order to avoid sampling similar scenes we sampled the video stream randomly but with the criteria that the temporal distance between any two frames must be at least 20 frames and the number of sampled frames cannot exceed the 5% of the total frames.

Subsequently, we trained the two CNNs with the help of our database using ADAM optimizer [12] and early stopping [5] with the following parameters: $\alpha = 0.0001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, $d = 0.0$, and $\varepsilon = 1e - 8$ where α is the learning rate, ε is the fuzz factor, and d is the learning rate decay over each update. 70% of our cartoon images have been in the training set and 30% of the images have been in the validation set, respectively. On the other hand we have tested our method on cartoon images whose source video is not in Table 1.

4 EXPERIMENTAL RESULTS

Our experiments have been performed on various cartoon images whose title or images were not in our training or validation database. The proposed cartoon colorization approach has been implemented in Keras [4] and is able to colorize properly various kind of cartoon images fully automatically. Figure 4 presents several colorization results obtained by the proposed method with respect to the grayscale inputs and ground-truth colorful cartoons. It can be seen that we could produce plausible colors. Moreover, neighboring regions with different grayscale values always get different *ab* values. Furthermore, the amount of artifacts by edges is minimal. The color filling within a region is nearly flawless, there are only few false edges within adjacent regions. These occur mainly in homogeneous wide backgrounds.

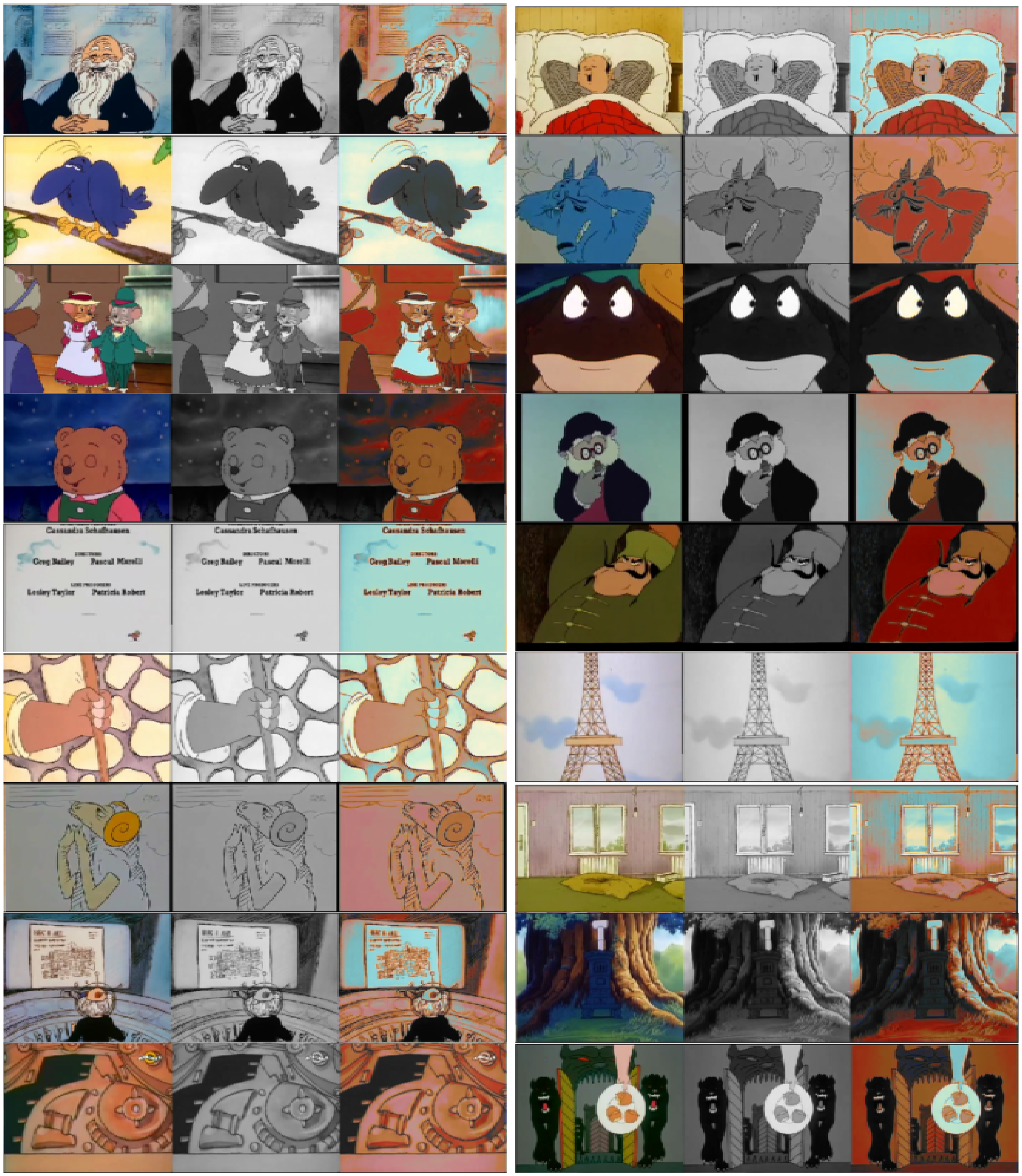

Figure 4: Colorized results. In every sequence the first image is the ground-truth, the second is the grayscale input, and the third is the colorized results.

Table 1: Our database contains 100,000 cartoon images. This table describes our sources with respect to the title, country of origin, release date, and the number of sampled frames.

Title	Country of Origin	Original Release	Number of frames
Les Mondes Engloutis	France	1985	9076
Hófehér	Hungary	1983	5119
Az erdő kapitánya	Hungary	1988	5487
Il était une fois... l'homme	France	1978	7668
The Princess and the Goblin	Hungary-UK-Japan	1992	8860
A nagy ho-ho-horgász	Hungary	1986	8872
Nu, pogodi!	Soviet Union	1980	9788
Fabulák	Hungary	1989	6661
Macskafogó	Hungary	1986	9201
Les Maîtres du temps	Hungary-France	1982	7427
Tiny Heroes	Germany-USA-Hungary	1997	5567
Čudnovate zgode šegrta Hlapića	Croatia-UK-Germany	1997	9511
Lucky Luke	France-USA	1971	6772

Table 2: Quantitative evaluation using the source code of [6]. In the best case Normalized Cross-correlation and Structural Content are 1. By the other indices the lower value is better.

Index number	Value
Mean Square Error	0.43
Peak Signal to Noise Ratio	60.5 *dB*
Normalized Cross-correlation	0.96
Average Difference	0.2
Structural Content	1.01
Maximum Difference	23.9
Normalized Absolute Error	0.05

Unfortunately, methodical quality evaluation by showing colorized cartoon images to human observers to rate the quality, is slow, expensive, and subjective. That is why, we have looked for quantitative indices. Unfortunately, there is no exact index number which could tell unequivocally the quality of a colorization. Using the source code of [6] we have done some qualitative evaluation. The colorized cartoon images and the ground truth images were compared by measuring indices implemented in [6]. Table 2 shows the averaged values of 100 colorized cartoon image - ground-truth image pairs. In the best case Normalized Cross-correlation and Structural Content are 1. By the other indices the lower value is the better. In the field of image compression if Peak Signal to Noise Ratio is 30 *dB* then the quality of the compression is considered to be good.

To our best knowledge, no existing papers or research studies address cartoon colorization using deep learning techniques. Since the result of existing scribble-based or example-based techniques heavily depends on the user's artistic skills or the quality of the reference image, we have not any comparisons to them.

5 CONCLUSION

This paper have investigated the colorization problem at the cartoon domain using deep learning techniques. We have shown that proper color filling is possible in different scales of cartoon images. There are many directions for further research. Unfortunately, the color uncertainty in cartoons is much higher than in natural images. To overcome this problem, we want to examine the learning-based colorization problem using additional information such as color-scribbles or a colorful reference image. Another direction of research would be the fine-tuning of VGG-16 based algorithms on the cartoon domain. The method can also be extended to re-colorize highly textured natural images as cartoon-like samples, where the cartoon part and the textures can be separated [21], and this artificially de-textured image can be filled by re-coloring to get a cartoon-like colored image.

ACKNOWLEDGMENTS

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] Aurélie Bugeau and Vinh-Thong Ta. 2012. Patch-based image colorization. In *Pattern Recognition (ICPR), 2012 21st International Conference on.* IEEE, 3058–3061.

[2] Guillaume Charpiat, Matthias Hofmann, and Bernhard Schölkopf. 2008. Automatic image colorization via multimodal predictions. *Computer Vision–ECCV 2008* (2008), 126–139.

[3] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. 2015. Deep colorization. In *Proceedings of the IEEE International Conference on Computer Vision.* 415–423.

[4] François Chollet. 2015. Keras. (2015).

[5] Federico Girosi, Michael Jones, and Tomaso Poggio. 1995. Regularization theory and neural networks architectures. *Neural computation* 7, 2 (1995), 219–269.

[6] Karunesh Kumar Gupta and RP Pareek. 2014. A Survey of Image Quality Assessment Techniques for Medical Imaging. *New Delhi Nov 1st and 2nd* (2014), 114.

[7] Bharath Hariharan, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. 2015. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 447–456.

[8] Kaiming He, Jian Sun, and Xiaoou Tang. 2013. Guided image filtering. *IEEE transactions on pattern analysis and machine intelligence* 35, 6 (2013), 1397–1409.

[9] Yi-Chin Huang, Yi-Shin Tung, Jun-Cheng Chen, Sung-Wen Wang, and Ja-Ling Wu. 2005. An adaptive edge detection based colorization algorithm and its applications. In *Proceedings of the 13th annual ACM international conference on Multimedia.* ACM, 351–354.

[10] Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. 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 (TOG)* 35, 4 (2016), 110.

[11] Revital Ironi, Daniel Cohen-Or, and Dani Lischinski. 2005. Colorization by Example.. In *Rendering Techniques*. Citeseer, 201–210.

[12] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[13] Amir Kolaman and Orly Yadid-Pecht. 2012. Quaternion structural similarity: a new quality index for color images. *IEEE Transactions on Image Processing* 21, 4 (2012), 1526–1536.

[14] Gustav Larsson, Michael Maire, and Gregory Shakhnarovich. 2016. Learning representations for automatic colorization. In *European Conference on Computer Vision*. Springer, 577–593.

[15] Anat Levin, Dani Lischinski, and Yair Weiss. 2004. Colorization using optimization. In *ACM Transactions on Graphics (ToG)*, Vol. 23. ACM, 689–694.

[16] Xiangguo Liang, Zhuo Su, Yiqi Xiao, Jiaming Guo, and Xiaonnan Luo. 2016. Deep patch-wise colorization model for grayscale images. In *SIGGRAPH ASIA 2016 Technical Briefs*. ACM, 13.

[17] Yingge Qu, Tien-Tsin Wong, and Pheng-Ann Heng. 2006. Manga colorization. In *ACM Transactions on Graphics (TOG)*, Vol. 25. ACM, 1214–1220.

[18] Erik Reinhard, Michael Adhikhmin, Bruce Gooch, and Peter Shirley. 2001. Color transfer between images. *IEEE Computer graphics and applications* 21, 5 (2001), 34–41.

[19] Karen Simonyan and Andrew Zisserman. 2014. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014).

[20] Daniel Sỳkora, Jan Buriánek, and Jiří Žára. 2004. Unsupervised colorization of black-and-white cartoons. In *Proceedings of the 3rd international symposium on Non-photorealistic animation and rendering*. ACM, 121–127.

[21] Dániel Szolgay and Tamás Szirányi. 2012. Adaptive image decomposition into cartoon and texture parts optimized by the orthogonality criterion. *IEEE Transactions on Image Processing* 21, 8 (2012), 3405–3415.

[22] Engin Tola, Vincent Lepetit, and Pascal Fua. 2010. Daisy: An efficient dense descriptor applied to wide-baseline stereo. *IEEE transactions on pattern analysis and machine intelligence* 32, 5 (2010), 815–830.

[23] Domonkos Varga and Tamás Szirányi. 2016. Fully automatic image colorization based on Convolutional Neural Network. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*. IEEE, 3691–3696.

[24] Tomihisa Welsh, Michael Ashikhmin, and Klaus Mueller. 2002. Transferring color to greyscale images. In *ACM Transactions on Graphics (TOG)*, Vol. 21. ACM, 277–280.

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

[26] Richard Zhang, Phillip Isola, and Alexei A Efros. 2016. Colorful image colorization. In *European Conference on Computer Vision*. Springer, 649–666.