Deep Correct: Deep Learning Color Correction for Color Blindness

Conference Paper · September 2017

DOI: 10.3233/978-1-61499-800-6-824

Gajo Petrovic
Meltin MMI

June 2017

Deep Correct: Deep Learning color correction for color blindness

Gajo Petrović [1], and Hamido Fujita
Iwate Prefectural University, Takizawa, Iwate, Japan

Abstract. Color vision deficiency affects 8% men and one in every 200 women. There are many different types of color blindness with the red-green as the most common. Most models for color correction are based on physiological models of how people with color vision deficiency perceive the world, with the goal of reducing errors derived from the color blindness simulation formula. In this paper we present Deep Correct, a novel Deep Learning based method for color correcting images in order to improve accessibility for people with color vision deficiency. The key elements of this work with regard to color blindness are two-fold: 1) we propose a data-driven Deep Learning approach for color correction and 2) we create an objective, quantitative metric for determining the distinguishability of images. Additionally, as a more general Deep Learning contribution, we propose a new method of training neural networks by utilizing error gradients from pretrained networks in order to train new, smaller networks.

Keywords. color blindness, color correction, deep learning

1. Introduction

Color blindness, or more correctly, color vision deficiency, is a decreased ability to see and correctly distinguish between colors. It affects 8% men and one in every 200 women. There are many different types of color blindness, with the red-green being the most common one. Color blindness is hereditary, and while it is mostly men who are color blind, women can still carry the related genes. There is no cure for color blindness yet, so it is often more common to provide assistance to people that have it.

While human (stereo) vision does not primarily rely on color to understand the world and recognize objects, it can still impact the ability to perform certain tasks, and some jobs (such as pilot, train driver, etc.) require people to have full color vision. The trouble with color vision is much more emphasized when using computers, as monitor displays present the entirety of information (including depth) as a color matrix. When creating software, it is therefore necessary to present elements in a way that they can still be clearly distinguished by color blind people.

Providing accessibility support for color blindness in software is therefore an important task. Ideally this is something that should be planned ahead, by not using color as identifiers. As an example, green and red color should not be used exclusively to present

[1]Corresponding Author: Gajo Petrović, Iwate Prefectural University, Takizawa, Iwate, Japan; E-mail: gajopetrovic@gmail.com

two opposite states, as people who have red-green vision deficiency will have trouble distinguishing between them. It is necessary to include some additional information (such as texture pattern, element shape, textual label, etc.) to identify elements. At the very least, the hue and saturation elements of color (in a HSV color mode) should not be used for presenting information.

Planning ahead is however difficult, and it is not uncommon that color blindness correction comes as an afterthought once the software product is already complete. It is then when developers are trying to accommodate accessibility, and it might be too late or too costly to redesign the user interface or rendering system. Because of that, it is common to provide support for color blindness accessibility as image color correction. This can be usually done by modifying the software output image to contain colors which make the desired elements clearly distinguishable.

The most common solution for color blindness is to apply a linear function that transforms the original color space into the new color space which is more suitable for people with a particular type of color blindness. There are multiple benefits of such an approach, as it is easy to implement and very fast to execute. As it only uses the original rendering of the software as input, it does not even have to be included in the original software, and may be provided as utility in the operating system or graphical processing unit driver. While this simplicity can be helpful, it is also a limitation. The primary purpose of color correction is to provide clear identification and differentiation of **objects** on the screen, and models that just utilize the original images lack this crucial software-specific information.

In this paper we have presented a Deep Learning color correction approach for color blindness. The proposed Deep Learning neural network is trained to output images which are most correctly distinguishable to color blind people. Instead of using human subjects, we simulate color blindness with software, and evaluate whether the simulated results are distinguishable enough. The two novel parts of this paper are: 1) we propose a quantifiable method for estimating image distinguishability for normal and color blind people, giving us both a clear metric to optimize and use for estimating correction method quality and 2) we present a data driven, deep learning approach for creating the color correction filter.

In section 2, we present the related work. In section 3, we describe the physiological condition in more detail and look into different types of color blindness. The problem setup and baseline approaches are presented section 4, while the proposed deep learning solution is in section 5. The results are shown in section 6, and we conclude the paper in section 7.

2. Related work

When creating software and considering accessibility, it is important to take color blindness into account as a considerable amount of people have color blindness or some sort of color vision deficiency. While there are still attempts to treat color blindness medically [15], there is so far no cure.

It is therefore necessary to create software that is usable by color blind people. This can be done during the software development process, and authors in [20] show how a color picker can be created to help user interface designers pick WCAG2.0[2] compliant colors. A simulation model for color blindness is presented in [10]. It attempts to demonstrate how people with specific types of color blindness perceive images, and it can be used during development and serve as a helpful visualization for verifying automated approaches. Manual approaches require effort during development, which is something that often does not happen, so we also need to consider automatic methods for accessibility.

There are a number of methods for dealing with color blindness. They can be separated into two groups depending on what they use for calculation: 1) pixel-based methods and 2) area-based methods. In paper [13] authors present state of the art methods for color blindness simulation, as well as a number of methods for both pixel and area based color correction. An adaptive method based on Fuzzy Logic to achieve color correction was presented in [8]. The authors have validated their results by using an experiment with four volunteers that have color vision deficiency. A skillet-based recoloring algorithm for dichromats is proposed in [18], where authors have replaced certain colors depending on their hue. They have shown improved results in comparison to previous methods, but this type of recoloring can sometimes introduce new and unexpected colors. A filter based on the Ishihara color test was designed in paper [17]. The resulting filter worked well for the test plates, and a modified version performed decently on real world images as well. A similar filter based method was created in [16]. Another method for recoloring was developed in [11], where the goal was to mitigate loss of color contrast due to color blindness.

There are a number of proposed methods [2] for hardware-based attempts of correcting color blindness, as well as additional issues that arise when using augmented reality. One such wearable device was developed in [12], and its usability was conducted in a clinical pilot study. In [27], authors have created a functional reflective polarizer that can be used in augmented reality systems. Chroma [23] is a wearable augmented reality solution for color blindness based on Google Glass. It automatically adapts to the type of the color blindness and includes a solution for color saliency. The authors have additionally conducted a user study to confirm the results. Real-time color and contrast correction for head-mounted displays is performed in [7]. The applied methods fix issues with color blindness due to color blending (mixing) of the augmented user interface and the real world. The authors also present a user evaluation of the proposed methods. Sometimes taking account of the hardware display can help with correcting for color blindness, as was done in [25], where authors have introduced a backlight control algorithm for LCD display.

Recently, a rise in Deep Learning has lead to many breakthroughs in machine learning areas, including computer vision. The annual competition ILSVRC [19] on the ImageNet [3] dataset has been dominated by deep neural network solutions in the past years, leading to well known architectures such as VGG16 [21], ResNet50 [6], InceptionV3 [22] and others.

Outside of image classification, Deep Learning has been used on a number of color-correction related tasks. An approach to photo adjustment was presented in [24], where authors implemented a image descriptor for local image semantics. The resulting photo enhancement system has been reported to perform better both quantitatively and qualitatively. In [9], authors demonstrate how Deep Neural Networks can be used for au-

[2]WCAG2.0: Web Content Accessibility Guidelines - `https://www.w3.org/TR/WCAG20/`

tomatic photo adjustment (with semantic-aware photo enhancement), replacing a time-consuming and difficult task that usually requires advanced skills. An unsupervised approach to color adjustment was done in [1], where authors have developed a system that learns style rankings by using a large photo database and a diverse set of styles for style transfer. The goal of this work was to create compelling, artefact-free images, and a user study was conducted to test it. Real-time image colorization was shown in [26], where authors have implemented a Deep Learning system that transforms greyscale images into colors. The system allows for user-guided image colorization, and it was trained with simulated user input.

3. Color blindness

Human color vision is based on three color photoreceptors, known as the L, M and S cones, due to their sensitivity to the long, medium and short wave bands of the spectrum [13] [10]. The combination of the three photoreceptors is interpreted as color sensation by the brain. For normal human vision (*normal trichromacy*), all three have to be properly sensitive. If any of the receptors are abnormal (usually due to the absence or presence of particular photopigments), the person will perceive color differently, and is said to have color vision deficiency (CVD). Depending on the type of abnormality, there are different types of CVDs.

In case of photopigment sensitivity shifts, the condition is called *anomalous trichromacy*, which causes colors to be perceived differently. Depending on which of the L, M and S cones are affected, the conditions are called *protanomaly*, *deuteranomaly* and *tritanomaly* respectively. In case of the *protanomaly*, the L cone is shifted towards the shorter wavelength M cone, while in the case of *deuteranomaly* and *tritanomaly*, the M and S cones are shifted towards the longer wavelengths Figure 1.

Dichromacy is the condition where the person is missing color vision in one of the three cones, and is thus only able to perceive two colors. Likewise, *monochromacy* is the case where two or all three of the photopigments are missing, and the person cannot distinguish any color.

The most common (99% of CVD) type of color blindness is the *red-green* blindness where users cannot distinguish between red and green colors. It is caused by the abnormality of the L or M cones, where three quarters are *analogous trichromats* and the rest are *dichromats*. In this paper we primarily focus on this type of CVD, although the approach can be extended to other types of *dichromacy* or *anomalous trichromacy*.

4. Color correction

In order to correct for color blindness, it is first necessary to create a model of how color blind people perceive the world. The simulation model used in this paper is based on [14], a simple linear transformation in the LMS color space. The transformation is shown in equation 1, and consists of first converting the input RGB image to the LMS color space 1a, applying a CVD simulation filter $C_i[y]$ and then reverting the image back to the RGB color space 1b. Specific CVD filter formulas for deuteranope, protanope and tritanope types are given in 2a, 2b and 2c respectively. Examples of applying these filters are shown in Figure 2.

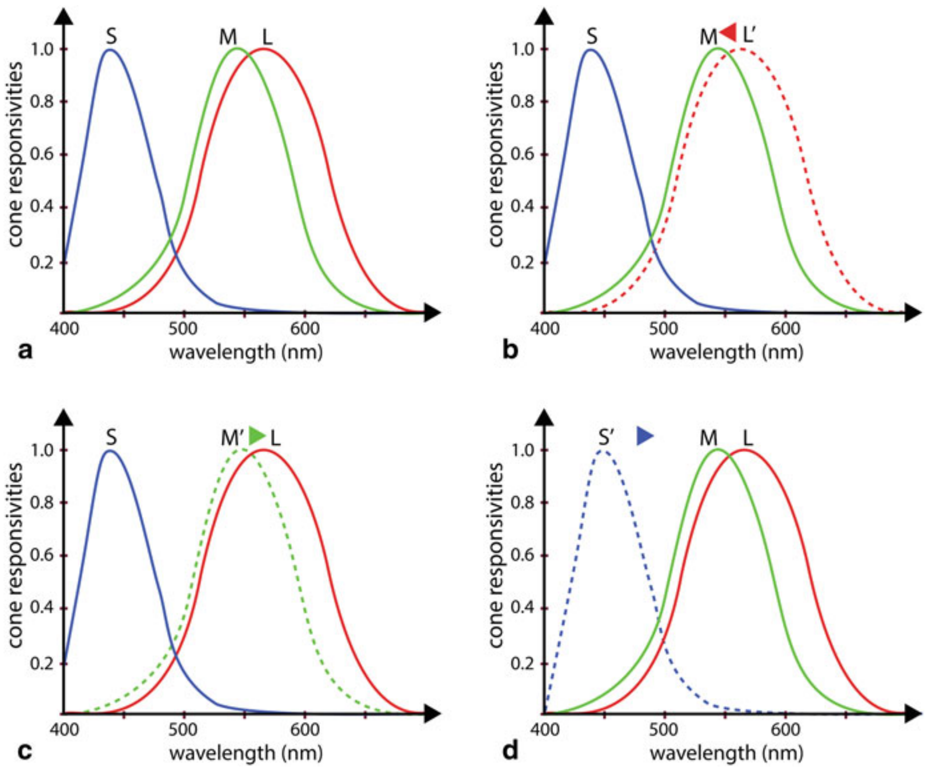

Figure 1. The spectral sensitivity functions. a) Normal trichromacy. b) Protanomaly (anomalous L cone type). c) Deuteranomaly (anomalous M cone type). d) Tritanomaly (anomalous S cone type). Figure source: [13]

$$RGB2LMS = \begin{pmatrix} 17.8824 & 43.5161 & 4.11935 \\ 3.45565 & 27.1554 & 3.86714 \\ 0.0299566 & 0.184309 & 1.46709 \end{pmatrix} \tag{1a}$$

$$SIM_i(I) = RGB2LMS^{-1} \cdot CVD_i \cdot RGB2LMS \cdot I \tag{1b}$$

$$CVD_{deuteranope} = \begin{pmatrix} 1 & 0 & 0 \\ 0.494207 & 0 & 1.24827 \\ 0 & 0 & 1 \end{pmatrix} \tag{2a}$$

$$CVD_{protanope} = \begin{pmatrix} 0 & 2.02344 & -2.52581 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \tag{2b}$$

$$CVD_{tritanope} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ -0.395913 & 0.801109 & 0 \end{pmatrix} \tag{2c}$$

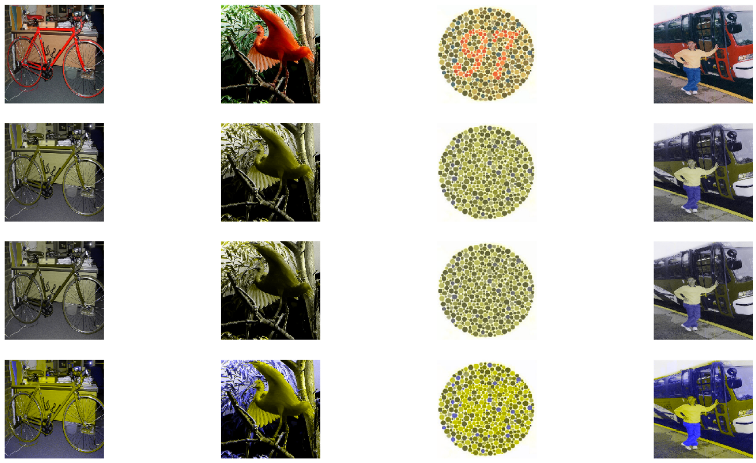

Figure 2. Different CVD simulations for example images. From top to bottom: normal vision, deuteranope, protanope and tritanope simulations.

The goal of color correction is to find a function R_i such as that when applied to an original image I, produces the best resulting image O_i, according to formula Equation 3.

$$O_i(I) = SIM_i(R_i(I)) \tag{3a}$$

Determining what constitutes the ideal output image is a non-trivial problem. According to the best of our knowledge, most papers, such as the ones listed in the related work section section 2, rely on conducting user studies. In the next section, we present a new metric for evaluating the quality of the output image.

The baseline method for color correction is given in Equation 4. It is a simple linear transformation, meant to emphasize contrast between colors that are hard to see. It works by subtracting the result obtained with CVD simulation from the original image (giving us colors which are not seen by people with CVD). This result is then shifted to a more visible specter by applying the matrix multiplication as shown.

$$R_{linear}(I) = I + \begin{pmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0.7 & 0 & 1 \end{pmatrix} (I - SIM_i(I)) \tag{4}$$

5. Deep Correct

In this paper we present the Deep Learning based color correction solution called Deep Correct. This system, as shown in figure Figure 3 consists of two modules: 1) *Corrector*

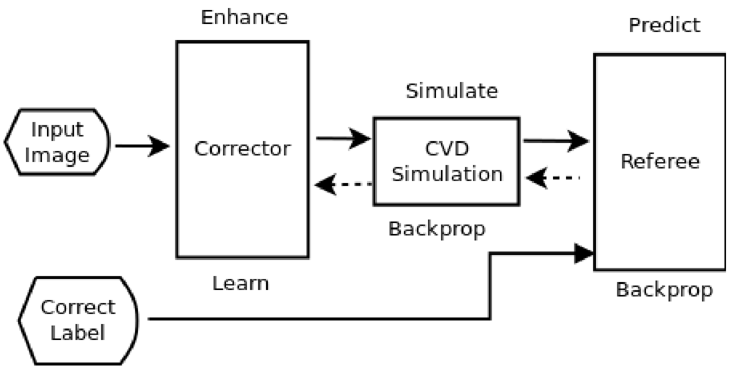

Figure 3. Deep Correct architecture. The top labels show actions taken in the forward pass, and the bottom labels show how errors are backpropagated.

- which performs the color correction and 2) *Referee* - which determines the quality of the result and is used to train the *Corrector*.

Corrector is a neural network that takes the original image as input and produces an output image of the same dimension. Its architecture consists of a linear corrector layer followed by 3 convolutional layers with 3x3 kernels and 16, 16 and 3 filters, respectively for the first, second and third layer. *Corrector* is trained by using backpropagation, with the gradients passed as result from another neural network - the *Referee*. On one hand, the goal of *Corrector* is to compensate for the loss of information in the color vision deficiency simulation filter, but more generally, it is designed to emphasize image features which are used by the *Referee* network. This can make it useful for enhancing images for normal color vision as well, by filtering noise or highlighting important elements.

Referee is another neural network, which is used to determine the quality of the color corrected images. It is trained separately (pretrained), on a different problem. During training of the *Corrector* we fix all of *Referee's* weights, and instead backpropagate the errors from the *Referee* to the *Corrector*. The *Referee* is set to solve a computer vision task, such as image classification, object recognition or similar, a task which relies on the high quality (discernibility) of input images. It is therefore optimizing the *Corrector* for input quality, that is, as it backpropagates the error gradients, it trains the *Corrector* to produce good output images - the goal of color correction.

The Corrector-Referee architecture is inspired by GAN (Generative Adversarial Networks) [5], although it is different in a number of key ways. With GANs, we normally train both the *Generator* and the *Discriminator* network simultaneously, in a competitive fashion. Here, the *Referee* is fixed during the *Corrector* training process, and is instead trained to solve a different problem. This allows us to utilize powerful, well performing networks, for entirely different problems. Additionally, in GANs, there is no clear stopping criteria, as the two networks are competing with each other, there is no clear metric. In our framework, the *Referee* provides us with a stopping criteria, making optimization a well defined goal.

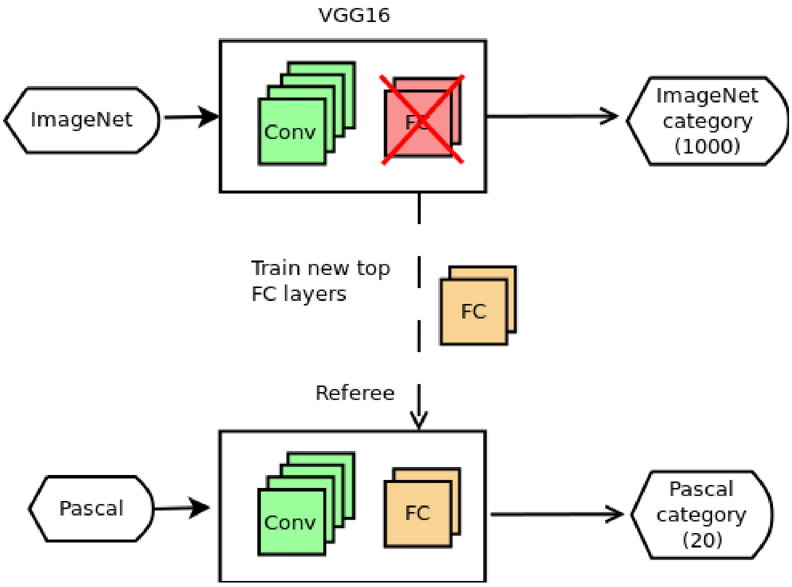

Figure 4. Referee model training and architecture

6. Results

Our *Referee* is trained on the *Pascal* dataset [4], with the goal of predicting the correct image label from 20 defined categories. We have created this network by using *VGG16* (one of the popular winning ImageNet models), removing the top fully connected layers (which are not useful for *Pascal*) and adding and training new fully connected layers to predict image categories for *Pascal*. VGG itself is trained on the much larger ImageNet dataset, and learns to produce one of the original 1000 categories, with relatively high accuracies (*70%* on *Top1* and *90%* on *Top5*). The fully connected layers are removed as we are not interested in the original 1000 categories, but we still utilize VGG's convolutional layers as they extract important features useful in training networks that perform computer vision tasks on real world objects. The Referee model creation process is shown in Figure 4.

We we have filtered down the *Pascal* dataset to a total of 11257 images, and separated into three sets, *D1*, *D2* and *D3* as shown in Table 1. *D1* and *D2* are used to train the *Referee* network, while *D2* and *D3* are used to train the *Corrector* network, as shown in Table 2. The original dataset has 17125 images, but we have selected a subset in order to reduce category imbalance that would misrepresent accuracy.

Dataset	D1	D2	D3	Total
Size	4471	4545	2241	11257

Table 1. Dataset size

We have conducted the experiments in the following order:

- First, we train the *Referee* network on the *D1* dataset, while initializing the VGG16 part of the model on the weights provided by the authors in [21]. We sep-

Network	D1	D2	D3
Referee	Train	Validation	
Corrector		Train	Validation

Table 2. Dataset usage in *Corrector* and *Referee* networks

	Corrector		Linear		Uncorrected	
	Top1	Top5	Top1	Top5	Top1	Top5
Deuteranope	49.95%	83.08%	48.81%	82.35%	49.13%	83.5%
Protanope	50.6%	83.76%	48.85%	82.44%	49.17%	82.86%
Tritanope	49.68%	8176%	48.53%	82.58%	47.2%	81.66%

Table 3. Comparative results of applying the *Corrector*, linear transformation corrector and uncorrected vision

arated the top-model training in two sub-phases. In the first phase, we fast train the top fully connected layers with VGG outputs precomputed for *100* epochs. In the second phase, we fine-tune the fully connected layers and the last VGG convolutional layer with data augmented inputs ($shear \in [0, 0.2]$, $zoom \in [0, 0.3]$, $rotation \in [-10^\circ, 10^\circ]$ and $horizontal flip$) for another *100* epochs. This resulted in *51.4% Top1* and *84.06% Top5* accuracy for the validation dataset *D2*. These results essentially describe the simulated normal color vision.

- We have then trained three *Corrector* networks on the *D2* dataset, by using a different CVD simulation filter for each of them. The training took *100* epochs, and we conducted data augmentation in the same way as in the previous step. This was tested on the *D3* dataset, where we compared the results achieved using *Corrector* with the linear transformation corrector from section 4 and uncorrected vision as baselines. The full results are shown in Table 3

The results show that the proposed *Corrector-Referee* architecture performs comparably well to the baseline method. In Figure 5 we show example images and how they look after applying color correction for Deuteranope CVD. We can see that the proposed Deep Learning solution produces images which are clearly distinguishable in color, without having that as an explicit objective. In fact, it managed to generalize color correction to even Ishihara plates which do not exist in the training sets. It does however introduce noise, and some of the colors may appear unnatural.

7. Conclusion

We have presented Deep Correct, a method for color correcting images in order to improve accessibility for people with CVD, and we have shown it outperforming the baseline linear transformation.

We also propose a new Deep Learning network architecture where we utilize a pretrained network's error gradients to train a new neural network. This allows us to create neural networks which optimize arbitrary, complex functions created by the pretrained network.

Our experimental results were obtained from tests on our *Referee* architecture, which simulates certain aspects of the human vision. As future work, we will also apply (artis-

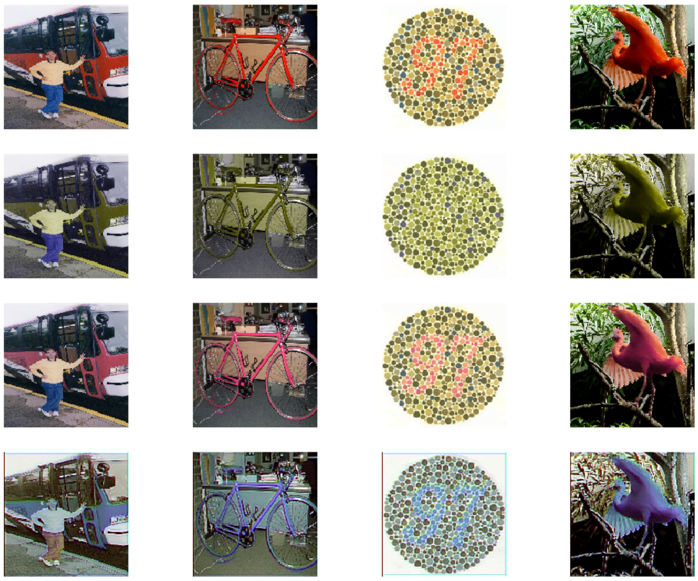

Figure 5. Correction of Deuteranope CVD. From top to bottom: normal vision, deuteranope simulation, linear correction and Deep Correct.

tic) style transfer or similar methods to obtain better, natural looking images. We plan to perform additional user studies where we can validate the correctness of using neural network simulations for human vision. Lastly, we will test our Corrector-Referee architecture on different Deep Learning problems as a more general approach to reusing large pretrained networks.

References

[1] Zezhou Cheng, Qingxiong Yang, and Bin Sheng. Deep colorization. *CoRR*, abs/1605.00075, 2016.

[2] Hélio M. de Oliveira, J. Ranhel, and R. B. A. Alves. Simulation of color blindness and a proposal for using google glass as color-correcting tool. *CoRR*, abs/1502.03723, 2015.

[3] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[4] M. Everingham, S. M. A. Eslami, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The pascal visual object classes challenge: A retrospective. *International Journal of Computer Vision*, 111(1):98–136, January 2015.

[5] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 2672–2680. Curran Associates, Inc., 2014.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

[7] J. David Hincapié-Ramos, L. Ivanchuk, S. K. Sridharan, and P. P. Irani. Smartcolor: Real-time color and contrast correction for optical see-through head-mounted displays. *IEEE Transactions on Visualization and Computer Graphics*, 21(12):1336–1348, Dec 2015.

[8] Jinmi Lee and Wellington Pinheiro dos Santos. An adaptive fuzzy-based system to simulate, quantify and compensate color blindness. *Integr. Comput.-Aided Eng.*, 18(1):29–40, January 2011.

[9] Joon-Young Lee, Kalyan Sunkavalli, Zhe L. Lin, Xiaohui Shen, and In So Kweon. Automatic content-aware color and tone stylization. *CoRR*, abs/1511.03748, 2015.

[10] G. M. Machado, M. M. Oliveira, and L. A. F. Fernandes. A physiologically-based model for simulation of color vision deficiency. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):1291–1298, Nov 2009.

[11] Gustavo Mello Machado. *A model for simulation of color vision deficiency and a color contrast enhancement technique for dichromats*. PhD thesis, Universidade Federal do Rio Grande do Sul, 2010.

[12] P. Melillo, D. Riccio, L. Di Perna, G. Sanniti Di Baja, M. De Nino, S. Rossi, F. Testa, F. Simonelli, and M. Frucci. Wearable improved vision system for color vision deficiency correction. *IEEE Journal of Translational Engineering in Health and Medicine*, 5:1–7, 2017.

[13] Neda Milić, Dragoljub Novaković, and Branko Milosavljević. *Enhancement of Image Content for Observers with Colour Vision Deficiencies*, pages 315–343. Springer International Publishing, Cham, 2015.

[14] John D Mollon, Françoise Viénot, and Hans Brettel. Digital video colourmaps for checking the legibility of displays by dichromats. *Color: Research and applications*, 24(4):243–252, 1999.

[15] M. Neitz and J. Neitz. Curing Color Blindness–Mice and Nonhuman Primates. *Cold Spring Harbor Perspectives in Medicine*, 4(11):a017418–a017418, 2014.

[16] P. K. Nigam and M. Bhattacharya. Colour vision deficiency correction in image processing. In *2013 IEEE International Conference on Bioinformatics and Biomedicine*, pages 79–79, Dec 2013.

[17] S. Poret, R. D. Dony, and S. Gregori. Image processing for colour blindness correction. In *2009 IEEE Toronto International Conference Science and Technology for Humanity (TIC-STH)*, pages 539–544, Sept 2009.

[18] M. G. Ribeiro and A. J. P. Gomes. A skillet-based recoloring algorithm for dichromats. In *2013 IEEE 15th International Conference on e-Health Networking, Applications and Services (Healthcom 2013)*, pages 702–706, Oct 2013.

[19] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015.

[20] Frode Eika Sandnes and Anqi Zhao. An interactive color picker that ensures wcag2.0 compliant color contrast levels. *Procedia Computer Science*, 67:87 – 94, 2015. Proceedings of the 6th International Conference on Software Development and Technologies for Enhancing Accessibility and Fighting Info-exclusion.

[21] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *CoRR*, abs/1409.1556, 2014.

[22] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *CoRR*, abs/1512.00567, 2015.

[23] Enrico Tanuwidjaja, Derek Huynh, Kirsten Koa, Calvin Nguyen, Churen Shao, Patrick Torbett, Colleen Emmenegger, and Nadir Weibel. Chroma: A wearable augmented-reality solution for color blindness. In *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '14, pages 799–810, New York, NY, USA, 2014. ACM.

[24] Zhicheng Yan, Hao Zhang, Baoyuan Wang, Sylvain Paris, and Yizhou Yu. Automatic photo adjustment using deep learning. *CoRR*, abs/1412.7725, 2014.

[25] J. I. You and K. C. Park. Image processing with color compensation using lcd display for color vision deficiency. *Journal of Display Technology*, 12(6):562–566, June 2016.

[26] Richard Zhang, Jun-Yan Zhu, Phillip Isola, Xinyang Geng, Angela S Lin, Tianhe Yu, and Alexei A Efros. Real-time user-guided image colorization with learned deep priors. *ACM Transactions on Graphics (TOG)*, 9(4), 2017.

[27] Ruidong Zhu, Guanjun Tan, Jiamin Yuan, and Shin-Tson Wu. Functional reflective polarizer for augmented reality and color vision deficiency. *Opt. Express*, 24(5):5431–5441, Mar 2016.