

# Smartphone Based Image Color Correction for Color Blindness

<https://doi.org/10.3991/ijim.v12i3.8160>

Lamiaa A. Elrefaei

King Abdulaziz University, Jeddah, Saudi Arabia

Benha University, Cairo, Egypt

laelrefaei@kau.edu.sa, lamia.alrefaai@feng.bu.edu.eg

**Abstract**—Color blind is a type of Color Vision Deficiency, which is the inability that a person could not realize the differences between some colors. There are three types of color blindness: Monochromacy, Dichromacy, and Anomalous Trichromacy. Color blind cannot be cured. Today, technology gets up with solutions to help people with color blindness to see the image and distinguish between the different colors using some algorithms. This paper presents a smartphone based experimental comparison of color correction algorithms for all Dichromacy color-blind viewers: Protanopia, Duteranopia, and Tritanopia. This comparison includes LMS Daltonization algorithm, Color-blind Filter Service (CBFS) algorithm, LAB color corrector algorithm, and the shifting color algorithm. The LMS algorithm is implemented for all the three types of Dichromacy. While CBFS, LAB adjustment, and Shifting color algorithms are applied to correct colors for Protanopia, Duteranopia, and Tritanopia respectively. The results show that the processing time for LMS algorithm is slow compared to other algorithms. For Protanopia people, the LMS algorithm is better than CBFS algorithm as the LMS algorithm only changes color of confused areas with no change in the brightness. For Duteranopia people, the LAB color correction is better than the LMS algorithm. For Tritanopia people, both the shifting color algorithm and the LMS algorithm may produce a new confusion in the proceed images. An application interface is implemented to enable the user to choose the algorithm that gives the most appropriate results.

**Keywords**—Color-blindness, Protanopia, Duteranopia, Tritanopia, Daltonization, LMS Detonaization, LAB color, OpenCV, Android.

## 1 Introduction

Color blindness affects approximately 1 in 12 men and 1 in 200 women in the world. Most people with color blindness have an ability to see things as clearly as other people but they unable to see red, green or blue light clearly. There are extremely rare cases where people are unable to see any color at all [1], [2].

There are different color blindness causes. For the most people with color vision deficiency, the condition is genetic and has been inherited from their mother. Also,

there are some diseases causing color blindness such as diabetes and multiple sclerosis or acquiring the condition over time due to the aging process, medication and so on [1], [2].

For human vision, there are two types of photoreceptors: rods and cones. Rods are sensitive to light while cones are sensitive to colors. Cones have three types; L-cones which are sensitive to long wavelength (red), M-cones which are sensitive to middle wavelength (green), and S-cones which are sensitive to short wavelength (blue). According to these cones, there are three types of color blindness [3], [4]; 1) Monochromacy, in which no cones or only one cone type exist, 2) Dichromacy, in which one cone type is missing, which can be of three types: Protanopia, in which L-cones are missing, Deuteranopia, in which M-cones are missing, and Tritanopia, in which S-cones are missing, and 3) Anomalous Trichromacy, in which there is a reduction in the sensitivity to a particular color, which can be of three types: Protanomaly which corresponds to a reduced sensitivity to red light, Deuteranomaly which corresponds to a reduced sensitivity to green light, and Tritanomaly which corresponds to a reduced sensitivity to blue light. Protanopia and Deuteranopia are the two types of red-green color blindness. Tritanopia is known to be blue-yellow color blindness. Table 1 summarizes the different color-blind types, their causes and effects.

**Table 1.** Color Blindness Types, Causes and Effects

Color blind type	Cause	Effect
<b>Monochromacy</b>	No cones or only one cone type exist.	Inability to see any color (see the world in grey shades)
<b>Dichromacy</b>	One cone type is missing, three types: 1) Protanopia: L-cones are missing 2) Deuteranopia: M-cones are missing 3) Tritanopia: S-cones are missing	Inability to see the color corresponding to the missed cone type. Inability to see Red color (Red Blind) Inability to see Green color (Green Blind) Inability to see Blue color (Blue Blind)
<b>Anomalous Trichromacy</b>	All cone types are exist, but they are not aligned, three types: 1) Protanomaly: L-cones are not aligned 2) Deuteranomaly: M-cones are not aligned 3) Tritanomaly: S-cones are not aligned	Reduction in the sensitivity to a particular color  Less sensitivity to Red color (Red weak)  Less sensitivity to Green color (Green weak)  Less sensitivity to Blue color (Blue weak)

Color blind cannot be cured. Therefore, the only method to help color blind people to recognize or differentiate color is with a vision aid kit. Today, technology gets up with solutions to help people with color blindness to see the image and distinguish between the different colors using some algorithms.

This paper focuses on Dichromacy color blind type. Figure 1 shows how each Dichromacy color blind person see an image against the normal person. This paper pre-

sents a smartphone based experimental comparison of color correction algorithms for Dichromacy viewers. This comparison includes LMS Daltonization algorithm [5], [6], [7], Color-blind Filter Service (CBFS) algorithm [8], LAB color corrector algorithm [7], and the shifting color algorithm [9]. The LMS algorithm is implemented for all the three types of Dichromacy: Protanopia, Duteranopia, and Tritanopia. The CBFS algorithm is implemented only for Protanopia viewers. The LAB color corrector algorithm is implemented only for Duteranopia viewers. The shifting color algorithm is implemented only for Tritanopia viewers.

The rest of this paper is organized as follows: section 2 introduces the literature review. The tools and methodologies are introduced in section 3. In section 4, the details of the color correction algorithms implementations and experiments are presented. Section 5 shows the results. Finally, Section 6 concludes the paper.

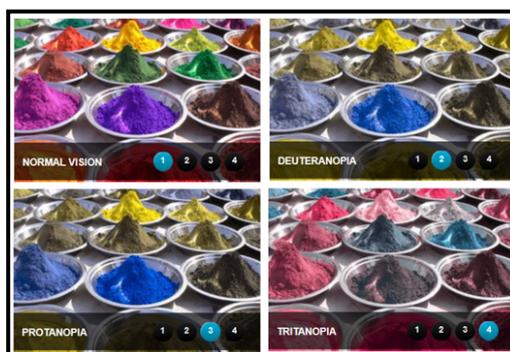


Fig. 1. Colors as seen by Dichromacy color blind [4].

## 2 Literature Review

In internet world and websites, there are many computerized aids developed for color blindness:

- Color-blind Testers
- Color-blind Simulators
- Color correction/recognition.

### 2.1 Color-blind Testers

Color-blind testers are used for testing if the person has a color blind, and if yes, determine the type of color blind.

There are many online web-based and smartphone-based applications tests that are available to test the color vision. Examples of these online web-based tests are<sup>1</sup>: the *Ishihara Plate test*, which was introduced by Dr. Shinobu Ishihara from Japan, the

---

<sup>1</sup> <http://www.color-blindness.com/color-blindness-tests/>

*Farnsworth D-15 arrangement test* which was introduced by Farnsworth in 1947, the *PseudIsochromatic Plate (PIP) Test*, and finally *Farnsworth-Munsell 100 Hue Test* which was developed by Dean Farnsworth in the 1943. An example of mobile applications tests in Android is *Color Blind Check*<sup>2</sup>. It is important to mention that the test results of these tests may be varied with different lighting conditions and with different trials on different devices, as each computer/mobile phone screen has different color settings.

## 2.2 Color-blind Simulators

Using color blindness simulators reveals how images may appear to users with a variety of color blindness conditions. People use color-blind simulators to check their art work, pictures, documents and web pages for color blind visibility [10] [11]. Some simulators are offered as an offline software program, an online web page, or a mobile application. Table 2 shows some examples of these simulators and their supported color blindness types and platforms.

**Table 2.** Color Blind Simulator Examples

Color-blind Simulator	The supported color blindness types	Platforms
<b>Color Oracle</b> <sup>3</sup>	all types of Dichromacy: <ul style="list-style-type: none"> <li>• Deuteranopia</li> <li>• Protanopia</li> <li>• Tritanopia</li> </ul>	<ul style="list-style-type: none"> <li>• a downloaded program that works offline for Window, Mac and Linux.</li> </ul>
<b>Vischeck</b> <sup>4</sup>	all types of Dichromacy: <ul style="list-style-type: none"> <li>• Deuteranopia</li> <li>• Protanopia</li> <li>• Tritanopia</li> </ul>	<ul style="list-style-type: none"> <li>• a website that can run on a chosen uploaded image after selecting the type of color vision to simulate</li> <li>• a web-page.</li> <li>• an offline plugin with the help of Adobe Photoshop or ImageJ for Window, Mac and Linux.</li> </ul>
<b>CVD simulator(Coblis)</b> <sup>5</sup>	<ul style="list-style-type: none"> <li>• Monochromacy</li> <li>all types of Dichromacy:                             <ul style="list-style-type: none"> <li>• Deuteranopia</li> <li>• Protanopia</li> <li>• Tritanopia</li> </ul> </li> <li>all types of Anomalous Trichromacy:                             <ul style="list-style-type: none"> <li>• Protanomaly</li> <li>• Deuteranomaly</li> <li>• Tritanomaly</li> </ul> </li> </ul>	<ul style="list-style-type: none"> <li>• a website that can run on a chosen uploaded image after selecting the type of color vision to simulate. Support image zooming feature.</li> </ul>
<b>Color Blindness Simulate Correct</b> <sup>6</sup>	all common colorblindness types are supported.	<ul style="list-style-type: none"> <li>• an Android application that simulates and corrects color blindness in real-time using the built-in camera of the mobile phone.</li> </ul>

<sup>2</sup> <https://play.google.com/store/apps/details?id=ch.colblindor.colorblindcheck&hl=en>.

<sup>3</sup> <http://colororacle.org/>

<sup>4</sup> <http://www.vischeck.com/>

<sup>5</sup> <http://www.color-blindness.com/coblis-color-blindness-simulator/>

<sup>6</sup> [http://www.seewald.at/en/2012/01/color\\_blindness\\_correction\\_and\\_simulator](http://www.seewald.at/en/2012/01/color_blindness_correction_and_simulator)

### 2.3 Color Correction for Dichromacy

There are many researchers proposed color correction techniques for Dichromacy. These techniques vary in terms of the supported Dichromacy types and the algorithms used. Techniques for modifying images for Deuteranopia viewers are presented in [6] and [7]. They used three algorithms: LMS Daltonization, color contrast enhancement, and LAB color adjustment. The LMS Daltonization algorithm, also used in [5], [12].

K. Erdoğan and N. Yılmaz [9] suggested a method that shifting colors based on HSV color space. The main concept is shifting the Hue channel for all spectrum by a specific value in the range [0.1-0.9].

The Color-blind Filter Service (CBFS) algorithm, suggested by [7], is a user-assisted recoloring technique. It modifies the original image by using six parameters provided from the user. In [13], a color compensation vision system is proposed for red and green color blindness people. This system is depending on converting RGB image to HSL.

C. L. Lai *et al.* [14] developed a portal vision auxiliary agency system for people who have color blindness or serious low vision. Their system passes through three steps, the vision model measurement, perceptual compensation and pattern recognition.

The mentioned color correction algorithms can be used for each type of Dichromacy color blindness. Hence, algorithms applied in the papers are suggested for a particular type of color blindness, but this is not precluded to apply them and work properly for another type.

## 3 Tools and Methodology

In this paper, a smartphone based experimental comparison of color correction algorithms for Dichromacy viewers is presented. Android platform is chosen because it is free and open source [15]. Android Studio, which is one of the most used Integrated Development Environment (IDE), is used to develop the application. Android Software Development Kit (SDK) is used to help developing mobile applications for Android. The Android SDK tools include development tools, emulator, and required libraries to build Android applications<sup>7</sup>. OpenCV is an open source computer vision software library<sup>8</sup>, it is used for image processing in the developed application.

In this paper, four algorithms are implemented to correct colors for Dichromacy. As shown in Table 3, the LMS Daltonization algorithm is applied to correct colors for all Dichromacy types. While CBFS, LAB adjustment, and Shifting color algorithms are applied to correct colors only for Protanopia, Duteranopia, and Tritanopia respectively. In Section 4, the smartphone-based implementations of these four algorithms are explained.

---

<sup>7</sup> <https://developer.android.com/>

<sup>8</sup> <http://opencv.org/>

**Table 3.** The implemented algorithms for each Dichromacy type in this paper.

Color blindness type	The implemented algorithms
Protanopia	LMS Daltonization & CBFS
Duteranopia	LMS Daltonization & LAB adjustment
Tritanopia	LMS Daltonization & Shifting colors

Application interfaces, shown in Figure 2, are created and ported on LG g2 smartphone that run on Android jelly bean OS v.4.2.2. Figure 2 (a) shows the application icon. The main interface that appears when the user opens the application is shown in Figure 2 (b). In the main interface the user can choose his/her color-blindness type. Then one of the interfaces, shown in Figures 2 (c)-(e), enables the user to choose one of the two solutions created for each color-blindness type. The user can try both solutions, then choose the most suitable and comfortable one to his/her daily use. After choosing one solution, the mobile camera will open; user can capture any view then press OK. The application will process this view image and display the processed image on the mobile screen, as shown in Figure 2 (f).

For comparison purposes, laptops are used to run Android Studio and to run the simulation software. An offline methodology is followed as illustrated in Figure 3. The RGB original image is first entered to a color-blind simulator to see how the original image is seen by Dichromate. Then the RGB original image is processed by the implemented color correction algorithms to get the color corrected image, which is fed to the color-blind simulator to see how the corrected image is seen by Dichromate. Vischeck and Coblis simulators are used to see the difference between the original and the processed images for Dichromate people.

## 4 Algorithms implementation and Experiments

### 4.1 LMS Daltonization Algorithm

The LMS algorithm in the developed application is implemented for Protanopia, Duteranopia, and Tritanopia. It is the most famous algorithm used for color-blindness correction. Its idea is to use the information lost in the simulation of color blindness and use LMS color space to compensate colors missing in each group/type of cones, long (L), medium (M), and short (S) in order to be predictable to the viewer [5], [6], [7]. Algorithm 1 lists the steps of LMS Daltonization algorithm. In step 5 of algorithm 1, there are error modification matrices for Protanopia and Deuteranopia, but there is no special matrix found for Tritanopia. Although the work in [5] mentioned the Tritanopia matrix that convert from 3D LMS space to 2D space, it is not mentioned the Tritanopia matrix of error modification. The same error modification matrix of Protanopia is used also for Tritanopia. In our implementation, a new proposed error modification matrix for tritanopia is used, as in equation (11).

Figure 4 shows the differences for Tritanopia on some sample images processed using the Protanopia and the proposed error modification matrices. As shown in Figure 4, both error modification matrices distinguish between blue and green and be-

tween yellow and purple. But the Protanopia error modification matrix shifts all spectrum colors to blue and green. As the Tritanopia people have a problem with blue color, shifting colors to blue is not a good solution. The proposed error modification matrix shifts all spectrum colors to green and red.

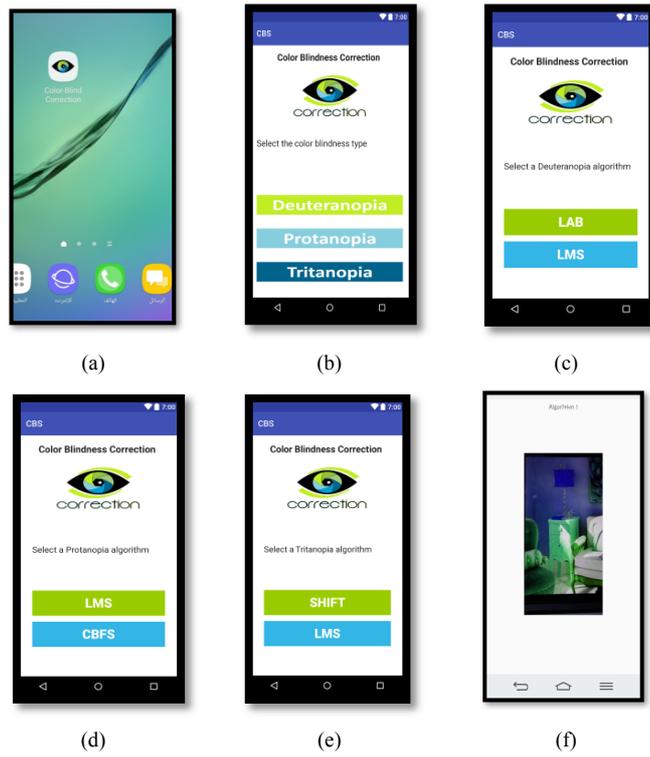


Fig. 2. The developed application interfaces

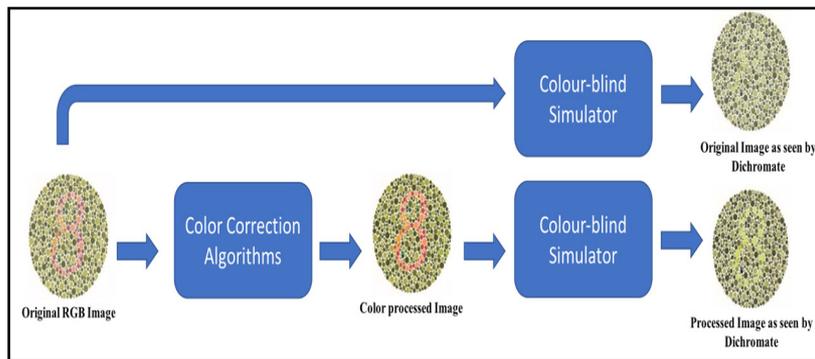


Fig. 3. The offline comparison methodology

---

**Algorithm 1: LMS Daltonization**

---

//Input: RGB input image

//Output: RGB color corrected image

---

- 1: Convert RGB image to LMS color space using equation (1)

$$\begin{bmatrix} L \\ M \\ S \end{bmatrix} = \begin{bmatrix} 17.8824 & 43.5161 & 4.11935 \\ 3.45565 & 27.1554 & 3.86714 \\ 0.0299566 & 0.184309 & 1.46709 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

- 2: Simulate color-blindness using equation (2) for Protanopia, (3) for Duteranopia and (4) for Tritanopia

$$\begin{bmatrix} L_P \\ M_P \\ S_P \end{bmatrix} = \begin{bmatrix} 0 & 2.02344 & -2.52581 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (2)$$

$$\begin{bmatrix} L_D \\ M_D \\ S_D \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0.49421 & 0 & 1.24827 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (3)$$

$$\begin{bmatrix} L_T \\ M_T \\ S_T \end{bmatrix} = \begin{bmatrix} 1.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ -0.395913 & 0.801109 & 0.0 \end{bmatrix} \begin{bmatrix} L \\ M \\ S \end{bmatrix} \quad (4)$$

- 3: Convert  $L_i M_i S_i$  back to  $R_i G_i B_i$  using equation (5),  $i = \{P, D, T\}$

$$\begin{bmatrix} R_i \\ G_i \\ B_i \end{bmatrix} = \begin{bmatrix} 0.0809444479 & -0.130504409 & 0.116721066 \\ 0.113614708 & -0.0102485335 & 0.0540193266 \\ -0.000365296938 & -0.00412161469 & 0.693511405 \end{bmatrix} \begin{bmatrix} L_i \\ M_i \\ S_i \end{bmatrix} \quad (5)$$

- 4: Find Difference between original and simulated images by (6), (7) and (8)

$$D_{R(i)} = R - R_i \quad (6)$$

$$D_{G(i)} = G - G_i \quad (7)$$

$$D_{B(i)} = B - B_i \quad (8)$$

- 5: Shift colors towards visible spectrum by multiplying by error matrices using (9) for Protanopia, (10) for Duteranopia and (11) for Tritanopia

$$\begin{bmatrix} R_{map(P)} \\ G_{map(P)} \\ B_{map(P)} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0.7 & 1 & 0 \\ 0.7 & 0 & 1 \end{bmatrix} \begin{bmatrix} D_{R(P)} \\ D_{G(P)} \\ D_{B(P)} \end{bmatrix} \quad (9)$$

$$\begin{bmatrix} R_{map(D)} \\ G_{map(D)} \\ B_{map(D)} \end{bmatrix} = \begin{bmatrix} 1 & 0.7 & 0 \\ 0 & 0 & 0 \\ 0 & 0.7 & 1 \end{bmatrix} \begin{bmatrix} D_{R(D)} \\ D_{G(D)} \\ D_{B(D)} \end{bmatrix} \quad (10)$$

$$\begin{bmatrix} R_{map(T)} \\ G_{map(T)} \\ B_{map(T)} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0.7 \\ 0 & 1 & 0.7 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} D_{R(T)} \\ D_{G(T)} \\ D_{B(T)} \end{bmatrix} \quad (11)$$

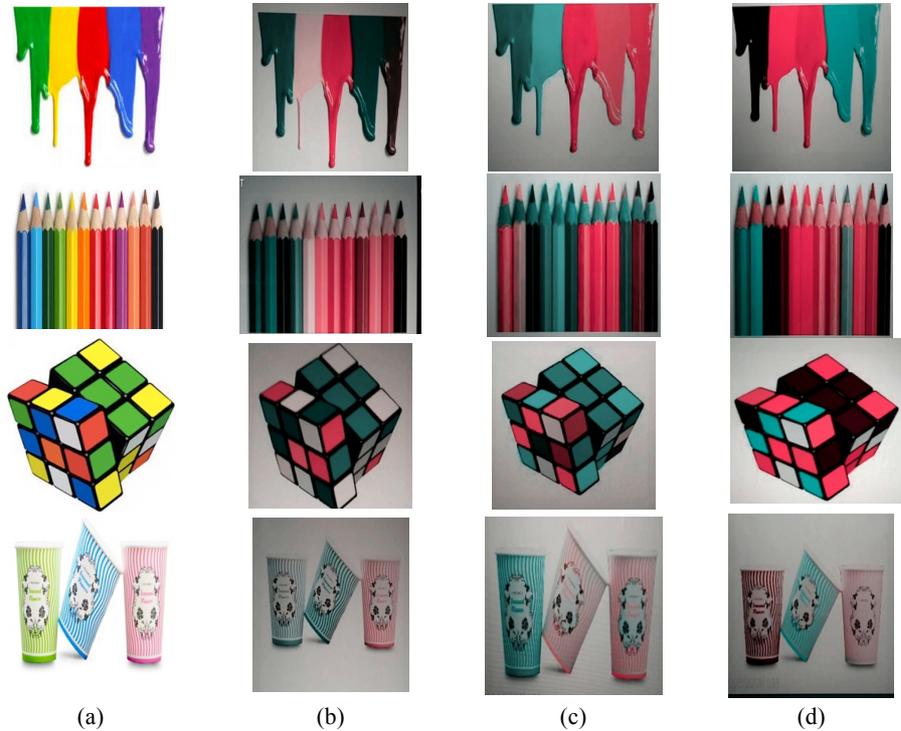
- 6: Add shifted colors to original image using (12), (13) and (14)

$$R_{F(i)} = R + R_{map(i)} \quad (12)$$

$$G_{F(i)} = G + G_{map(i)} \quad (13)$$

$$B_{F(i)} = B + B_{map(i)} \quad (14)$$


---



**Fig. 4.** LMS Daltonization algorithm Tritonopia experiments

- (a) Original images (b) Original images as seen by Tritonopia
- (c) LMS processed images as seen by Tritonopia using the Protanopia error modification matrix in (10)
- (d) LMS processed Images as seen by Tritonopia using the proposed error modification matrix in (11).

#### 4.2 Color-Blind Filter Service (CBFS) Algorithm

The CBFS algorithm in the developed application is implemented for Protanopia. It uses HSL (Hue, Saturation, Lightness) color space to represent colors [8]. Hue represents the color itself, Saturation is the amount of color, while Lightness is how much light or dark the color is. Algorithm 2 specifies the steps of this algorithm. In step 2 of algorithm 2, a closeness parameter is defined in order to decide how much the HSL pixel color is close to the dominant color of the RGB image pixels and it is equal to the absolute difference between them. For Protanopia, many values for this parameter from 10 to 90 are tested. Figure 5 shows the testing results on some images for closeness parameter equal to 30, 70, and 90. Based on the results shown in Figure 5, the closeness parameter value is set to 70 in the proposed application.

---

**Algorithm 2: Color-Blind Filter Service**

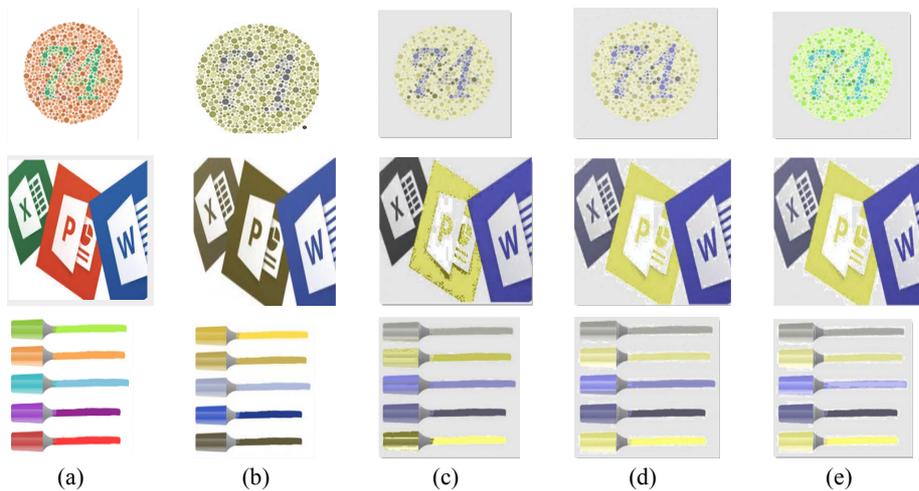
---

//Input: RGB input image

//Output: RGB color corrected image

---

- 1: Convert RGB image to HSL.
  - 2: for each image pixel, **If** the pixel color is close to the dominant color of the RGB image pixels (red/green) **then**
    - Hue  $\leftarrow$  Hue - 30% (15)
    - Saturation  $\leftarrow$  Saturation - 10% (16)
    - Lightness  $\leftarrow$  Lightness + 25% (17)
  - else**
    - Saturation  $\leftarrow$  Saturation + 10% (18)
    - Lightness  $\leftarrow$  Lightness - 10% (19)
  - 3: Convert HSL image back to RGB.
- 



**Fig. 5.** The CBFS algorithm Protanopia experiments

(a) Original images (b) Original images as seen by Protanopia

(c) CBFS processed Images as seen by Protanopia (Closeness parameter =30)

(d) CBFS processed Images as seen by Protanopia (Closeness parameter =70)

(e) CBFS processed Images as seen by Protanopia (Closeness parameter =90)

### 4.3 LAB Color Corrector

The LAB Color Corrector algorithm in the developed application is implemented for Duteranopia. It corrects colors for color blinder by modifying reds and greens of an image to increase color contrast for a color blinder [7]. The LAB Color Corrector implementation steps are listed in algorithm 3. The first operation is to adjust each pixel of A component, where a positive A means it is closer to red while negative A means it is closer to green as the LAB color range [-100, 100]. However, in Android platform, LAB ranges from 0 to 255. Hence, we considered the half of 255 (127) as 0

value in the range  $[-100, 100]$ , values greater than 127 is the positive part, and values less than or equal 127 is the negative part. This algorithm lacks clear theoretical basis. It is based upon experimental procedures relying mostly on trial and error in the presence of a color-blind viewer. Hence, the best adjustments between L, B, and A components for Duteranopia is determined based on experimenting different trials. According to algorithm 3, these trials were chosen to increase the color contrast between red-green to be noticeable regards to Duteranopia case.

#### 4.4 Shifting Color Algorithm

The Shifting color algorithm in the developed application is implemented for Tritanopia. In the Shifting color algorithm RGB image is read and colors are shifted by different ratios in HSV color space (Hue, Saturation, and Value) [9]. In algorithm 4, shifting color steps are explained. The  $h$  value in the algorithm defines the shifting ratio and it would be changed according to the color vision deficiency type [9]. To specify the  $h$  value for Tritanopia algorithm,  $h$  values are changed from 0.1 to 0.9 and tested for each one on some images as shown in Figure 6. The Hue channel in OpenCV has a range value from 0 to 179, so, the  $h$  value is multiplied by 180. Hue values of the whole image are shifted with a fixed rate, the colors will be changed but the information will be preserved. As shown in Figure 6, the green color is shifted to the inverse range of the blue color. We can see clearly the difference in the range of blue and green when  $h$  between 0.3 and 0.4. The same result of blue and green colors occurs with yellow and purple colors. The best value of  $h$  is 0.3, it gives a better result during experiment. The other values (0.4, 0.7, and 0.8) were excluded as they affect the other colors.

---

#### Algorithm 3: LAB Color Corrector

---

//Input: RGB input image

//Output: RGB color corrected image

---

- 1: Convert RGB image to LAB color space.
  - 2: Adjust the A values relative to its maximum, making positive values a bit more positive and negative values a bit more negative.
  - 3: Adjust the B values relative to how green or red it is.
  - 4: Change L pixels which is the brightness of the pixel relative to pixels A values.
  - 5: The image is converted back to the RGB color space and concatenated to ensure that pixel values lie between zero and one.
-

---

**Algorithm 4: Shifting Color**

---

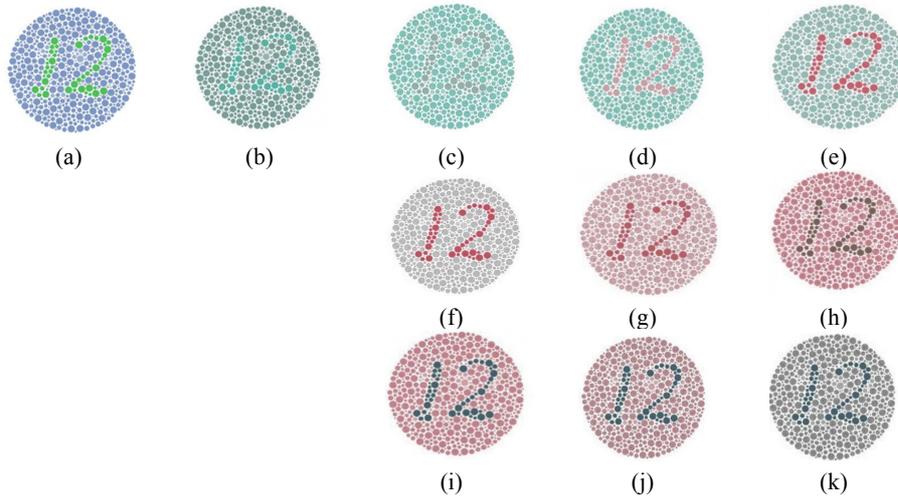
//Input: RGB input image

//Output: RGB color corrected image

---

- 1: Convert RGB image to HSV color space
  - 2: Shifting colors for each pixel  $(x, y)$  using equation (20)
 
$$H_{xy} = H_{xy} + h \quad (20)$$
  - 3: Check the values of hue channel to be in the range (0 to 0.9). **If**  $(H_{xy} > 1)$  **then** change the hue value using Equation 21. **If**  $(H_{xy} < 0)$  **then** change the hue value using Equation 22.
 
$$H_{xy} = 1 - H_{xy} \quad (21)$$

$$H_{xy} = 0 \quad (22)$$
  - 4: Convert HSV image back to RGB.
- 



**Fig. 6.** The Shifting algorithm Tritanopia experiments

(a) Original images (b) Original images as seen by Tritanopia

(c) - (k) The Shifting algorithm processed images as seen by Tritanopia for  $h$  equal to 0.1 – 0.9

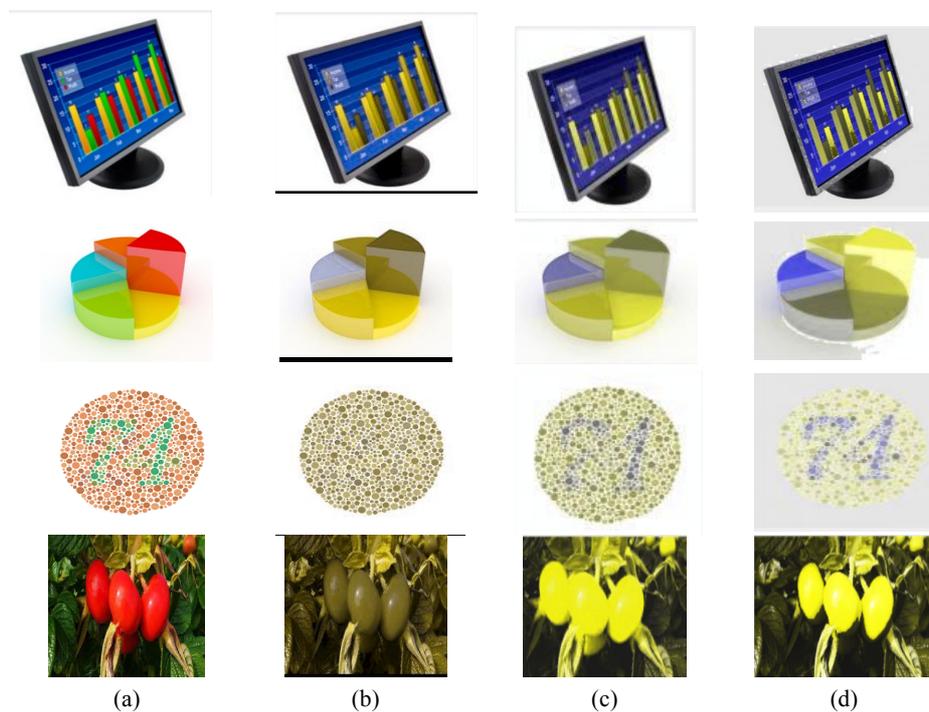
## 5 Results

The implemented algorithms explained in section 4 are ported on LG G2 smartphone with 2 GB Ram, 13 MP camera, quad-core 2.26 GHz CPU, and run on Android jelly bean OS v.4.2.2.

A set of test images are used to compare the results from the implemented algorithms for each Dicromate type, as listed in Table 2.

Figure 7 shows the resulted processed images from the implemented algorithms as seen by Protanopia: LMS Deltonization, Figure 7 (c), and CBFS, Figure 7 (d). People

with Protanopia have a difficulty to distinguish between black and many shades of red, dark brown and dark green, dark orange and dark red, blue and red, purple and dark pink, and mid green with some oranges [4]. The tested images processing average time is 5.43 seconds for LMS and 2.65 seconds for CBFS. The LMS algorithm only changes color of confused areas with no change in the brightness, while almost all colors of the image are changed by the CBFS algorithm. Unlike the CBFS processed images, in the LMS processed images, different colors are converted to different colors after processing.

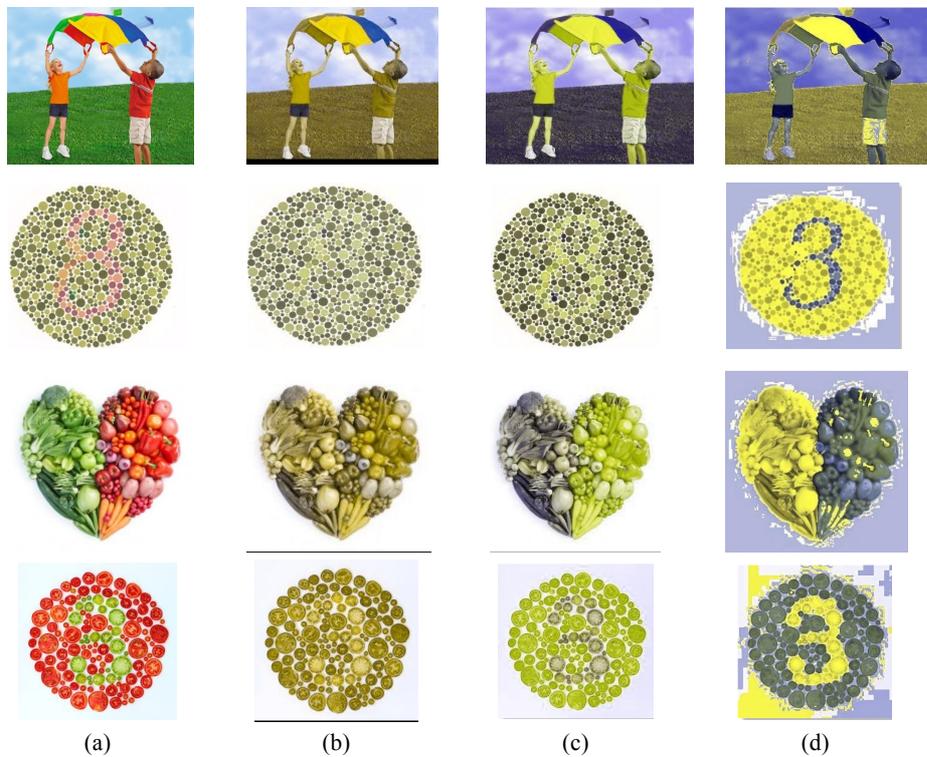


**Fig. 7.** The Protanopia Results

- (a) Original images
- (b) Original images as seen by Protanopia
- (c) LMS processed images as seen by Protanopia
- (d) CBFS processed images as seen by Protanopia

Figure 8 shows the resulted processed images from the implemented algorithms as seen by Duteranopia: LMS, Figure 8 (c), and LAB, Figure 8 (d). People who suffer from Duteranopia confuse mid-reds with mid-greens, bright greens with yellows, pale pinks with light grey, mid-reds with mid-brown, and light blues with lilac [4]. The tested images processing average time is 9.1 seconds for LMS and 1.3 seconds for LAB. Both algorithms achieved their purpose. In LAB color correction, there is a noticeable difference in the color range that is not visible for color-blind. In the LMS Daltonization, the Colors after processing are not quite noticeable for color-blind.

Figure 9 shows the resulted processed images from the implemented algorithms as seen by Tritanopia: LMS algorithm, using the proposed error modification matrix in equation (11), Figure 9 (c), and the shifting color algorithm, Figure 9 (d). Tritanopia is a very rare form of color blindness, occurring in less than 0.003% of males and females. The most colors that Tritanopia people confuse are light blues with greys, dark purples with black, mid-greens with blues and oranges with reds [4]. The tested images processing average time is 12.2 seconds for LMS and 3.4 seconds for the Shifting algorithm. In LMS, some converted colors become similar and cause a new confusion. The shifting color algorithm makes a difference between the colors that must be different for Tritanopia people, but some other colors may be affected.



**Fig. 8.** The Duteranopia Results

- (a) Original images
- (b) Original images as seen by Duteranopia
- (c) LMS processed images as seen by Duteranopia
- (d) LAB processed images as seen by Duteranopia

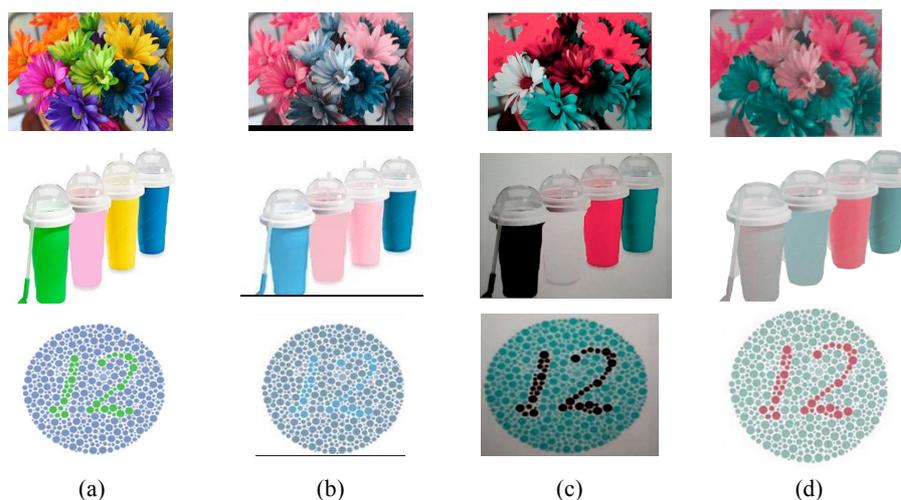
## 6 Conclusions

The goal of this paper is to present a smartphone based experimental comparison of color correction algorithms for Dichromacy viewers. This comparison included the LMS Daltonization algorithm, Color-blind Filter Service (CBFS) algorithm, LAB

color corrector algorithm, and the shifting color algorithm. The LMS algorithm is implemented for all the three types of Dichromacy: Protanopia, Duteranopia, and Tritanopia. In our implementation of the LMS algorithm, a new proposed error modification matrix for Tritanopia is used, as in equation (11). The CBFS algorithm is implemented only for Protanopia viewers. The LAB color corrector algorithm is implemented only for Duteranopia viewers. The shifting color algorithm is implemented only for Tritanopia viewers. These algorithms convert colors to other colors that color-blind persons can distinguish.

The results show that the processing time for LMS algorithm is slow compared to other algorithms. For Protanopia people, the LMS algorithm is better than CBFS algorithm as the LMS algorithm only changes color of confused areas with no change in the brightness. For Duteranopia people, the LAB color correction is better than the LMS algorithm. For Tritanopia people, both the shifting color algorithm and the LMS algorithm may produce a new confusion in the proceed images.

Although these algorithms have some weaknesses, they can solve their specific color-blindness type problem properly. In the developed application, the user can try solutions from the different implemented algorithms, then choose the most suitable and comfortable one to his/her daily use.



**Fig. 9.** The Tritanopia Results

(a) Original images (b) Original images as seen by Tritanopia (c) LMS processed images using equation (11) as seen by Tritanopia (d) The Shifting color algorithm processed images as seen by Tritanopia

## 7 References

- [1] Deane B. Judd, "The Color Perceptions of Deuteranopic and Protanopic Observers," J. Opt. Soc. Am. 39, 1949, pp. 252-256. <https://doi.org/10.1364/JOSA.39.000252>
- [2] "Color Blindness," 2016. [Online]. Available: <http://www.colourblindawareness.org/colour-blindness/>. [Accessed 19 February 2018].
- [3] B. L. Cole, "Assessment of inherited color vision defects in clinical practice," Clin Exp Optom., vol. 90(3), 2007, pp. 157-75. <https://doi.org/10.1111/j.1444-0938.2007.00135.x>

- [4] "Types of Color Blindness," 2016. [Online]. Available: <http://www.colourblindawareness.org/colour-blindness/types-of-colour-blindness/>. [Accessed 19 February 2018].
- [5] J. Rumiński, J. Wtorek, M. Kaczmarek, A. Bujnowski, T. Kocejko and A. Poliński, "Image Simulation and Annotation for Color Blinded," in 2010 2nd International Conference on Information Technology, (2010 ICIT), Gdansk, 2010.
- [6] Jyoti D. Badlani, C.N. Deshmukh, "A Novel Technique for Modification of Images for Deuteranopic Viewers," International Journal of Advanced Research in Computer and Communication Engineering, Vol. 5, Issue 4, April 2016, pp. 467-473.
- [7] N. Halder, D. Roy, T. Chowdhury, A. Chattaraj and P. Roy, "Image Color Transformation for Deuteranopia Patients using Daltonization," IOSR Journal of VLSI and Signal Processing (IOSR-JVSP), vol. 5, pp. 15-20, 2015.
- [8] G. Iaccarino, D. Malandrino, M. D. Percio and V. Scarano, "Efficient Edge-Services for Colorblind Users," in Proceedings of the 15th international conference on World Wide Web, New York, NY, USA, 2006. <https://doi.org/10.1145/1135777.1135944>
- [9] K. Erdoğan and N. Yılmaz, "Shifting Colors to Overcome Not Realizing Objects Problem due to Color Vision Deficiency," in Proc. of the Second Intl. Conf. on Advances in Computing, Electronics and Electrical Technology - CEET 2014, Kuala Lumpur, 2014.
- [10] Bernhard Jenny and Nathaniel Vaughn Kelso, "Designing maps for the colour-vision impaired," Bulletin of the Society of Cartographers SoC, 41, p. 9-12, 2007.
- [11] Bernhard Jenny and Nathaniel Vaughn Kelso, "Color Design for the Color Vision Impaired," Cartographic Perspectives, 58, p. 61-67, 2007. <https://doi.org/10.14714/CP58.270>
- [12] C.-N. Anagnostopoulos, G. Tsekouras, I. Anagnostopoulos and C. Kalloniatis, "Intelligent modification for the daltonization process of digitized paintings," in Proceedings of the 5th International Conference on Computer Vision Systems (ICVS 2007), Mytilene, Lesvos, 2007.
- [13] T. Ohkubo and K. Kobayashi, "A Color Compensation Vision System for Color-blind People," in SICE Annual Conference, Tokyo, 2008.
- [14] C. L. Lai, S. W. Chang and J. Sheen, "An Integrated Portable Vision Assistant Agency for the Visual Impaired People," in IEEE International Conference on Control and Automation, Christchurch, 2009. <https://doi.org/10.1109/ICCA.2009.5410541>
- [15] M. Kelvin, "what are the benefits of android application development," 2012.

## 8 Acknowledgment

The author would like to thank Duaa Hamid for her help in designing the application icon. The author also is grateful to the anonymous reviewers for their constructive suggestions to improve the quality of the paper.

## 9 Author

**Lamiaa A. Elrefaei** received her B.Sc. degree with honors in Electrical Engineering (Electronics and Telecommunications) in 1997, The M.Sc. in 2003 and Ph.D. in 2008 in Electrical Engineering (Electronics) from Faculty of Engineering at Shoubra, Benha University, Egypt. She has held a number of faculty positions at Benha University, as Teaching Assistant from 1998 to 2003, as an Assistant Lecturer from 2003 to 2008, and as Lecturer from 2008 to date. She is currently an Associate Professor at Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia. She is a senior member of IEEE.

Article submitted 22 December 2017. Resubmitted 19 February 2018. Final acceptance 14 March 2018. Final version published as submitted by the author.