Syrian Arab Republic

Syrian Private University

Faculty of Computer and Informatics Engineering

Color Vision Deficiency Assistance System Design

Graduation project

Submitted by:

Najla albakri

Abdulsattar Alkeddeh

Rami turki

Supervisor

Dr. Eng. Talal Hammoud

May 2023

Acknowledgments

We take this opportunity to express gratitude to all of the Department faculty members for their help and support. We also thank our parents for the unceasing encouragement, support and attention. we are also grateful to my brothers and sisters who supported us through this venture. We place on record, our sincere thanks to Dr. Talal Hammoud for the continuous encouragement, supervision, patience and advices.

Abstract

Color blindness is the failure to distinguish color differences under normal lighting conditions. It is a real issue which affects a significant portion of the population. It affects 1 in 12 men and 1 in 200 women or around 4.5 % of the entire world population. If the world population is estimated at 7 billion, around 315 million people suffer from some form of color blindness. This is roughly 80 times the population of Ireland. Due to the significant number of people affected, a viable solution for real time color perception enhancement is needed. The proposed solution consists of an Arduino board connected to camera to identify colors to diseases people and help them to recognize the true colors. We can also detect color and track its movement in the background.

General Introduction

Being color blind does not mean that a person cannot see any colors at all. There are many different kinds of color blindness, Usually, people will have a hard time telling certain colors apart, such as red and green. There are different causes of color blindness. For the vast majority of people with deficient color vision the condition is genetic and has been inherited from their mother, although some people become color blind as a result of other diseases. Most color blind people are able to see things as clearly as other people but they are unable to fully 'see' red, green or blue light. The effects of color vision deficiency can be mild, moderate or severe so, for example, approximately 40% of color blind pupils currently leaving secondary school are unaware that they are color blind, whilst 60% of sufferers experience many problems in everyday life. This project tends to design a system for helping color blind people to identify colors clearly. We start Chapter 1 with theoretical study that spot the light on the human visual system and color processing then we move in Chapter 2 to define color deficiency, causes, and diagnosis. We move then in Chapter 3 to describe the hardware implementation for the system using ESP32-CAM.

Contents

Abstract i

General introduction ii

List of Figures v

List of Tables vi

List of Abbreviations vi

Chapter 1: Color Image Processing

1.1. Introduction 2

1.2. Human Vision System 2

1.3. Photoreceptors 6

1.4. Color fundamentals 11

1.5. Color models 16

1.5.1. RGB color model 17

1.5.2. CMY and CMYK color model 18

1.5.3. HSI color model 20

Chapter 2: Color Vision Deficiency

2.1. Introduction 28

2.2. Color Vision Deficiencies 30

2.2.1. CVD Causes 33

2.2.2. CVD Diagnosis 35

Chapter 3: Color Detection and Tracking System Design

3.1. Introduction 39

3.2. ESP32-CAM 39

3.3. Design Coding 43

Conclusion 51

References 52

List of Figures

Figure 1-1: Anatomy of human eye. 3

Figure 1-2: Connection between cells in the human retina. 7

Figure 1-3: Rod and cone photoreceptors. 8

Figure 1-4: Rod and cone photoreceptors. 14

Figure 1-5: The CIE chromaticity diagram. 9

Figure 1-6: Illustrative color gamut of color monitors. 15

Figure 1-7: Schematic of the RGB color cube. 18

Figure 1-8: Relationships between the RGB and HSI color models. 22

Figure 1-9: Hue and saturation in the HSI color model. 24

Figure 1-10: The HSI color model. 26

Figure 2-1: how colorblind people see the traffic lights. 29

Figure 2-2: Google glass color identification system. 30

Figure 2-3: Color vision spectrum for color blind people. 32

Figure 2-4: The spectral sensitivity functions 32

Figure 2-5: Genetic Tree showing possible outcomes. 35

Figure 2-6: Ishihara test for colorblindness. 36

Figure 2-7: Ishihara test for colorblindness. 37

Figure 3-1: ESP32-CAM. 39

Figure 3-2: ESP32-CAM pinout. 40

Figure 3-3: Color track bar. 44

List of Tables

Table 1-1: Types of cones. 9

Table 3-1: MicroSD card Connection 42

Table 3-2: Camera Connection 43

Table 3-3: Frequency scaling. 48

List of Abbreviations

HVS	Human Vision System
RGB	Red, Green, Blue
CIE	Commission Internationale de l'Eclairage
CMY	Cyan, Magenta, Yellow
CMYK	Cyan, Magenta, Yellow, Black
HSI	Hue, Saturation, Intensity
CVD	Color Vision Deficiency

Chapter 1

Color Image Processing

1.1. Introduction

Using color in image processing is motivated by two principal factors. First, color is a powerful descriptor that often simplifies object identification and extraction from a scene. Second, humans can discern thousands of color shades, compared to only about two dozen shades of gray. The latter factor is particularly important in manual image analysis. Color image processing is divided into two major areas, pseudo- and full-color processing. In the first category, the issue is one of assigning color(s) to a particular grayscale intensity or range of intensities. In the second, images typically are acquired using a full-color sensor, such as a digital camera, or color scanner. Until just a few years ago, most digital color image processing was done at the pseudo- or reduced-color level. However, because color sensors and processing hardware have become available at reasonable prices, full-color image processing techniques are now used in a broad range of applications.

1.2. Human Vision System

The Human Vision System (HVS) is complex; however, a basic understanding is necessary to ensure that the final application solves the root cause of the problem. The human visual color response is driven by specialized cells called cones, which exist in three types, viz. R, G, and B. HVS perceptions are initiated and strongly influenced by the anatomical structure of the eye. Figure

1-1 shows a schematic representation of the optical structure of the human eye with some key features labeled. The human eye acts like a camera. The cornea and lens act together like a camera lens to focus an image of the visual world on the retina at the back of the eye, which acts like the film or other image sensor of a camera. These and other structures have a significant impact on our perception of color.

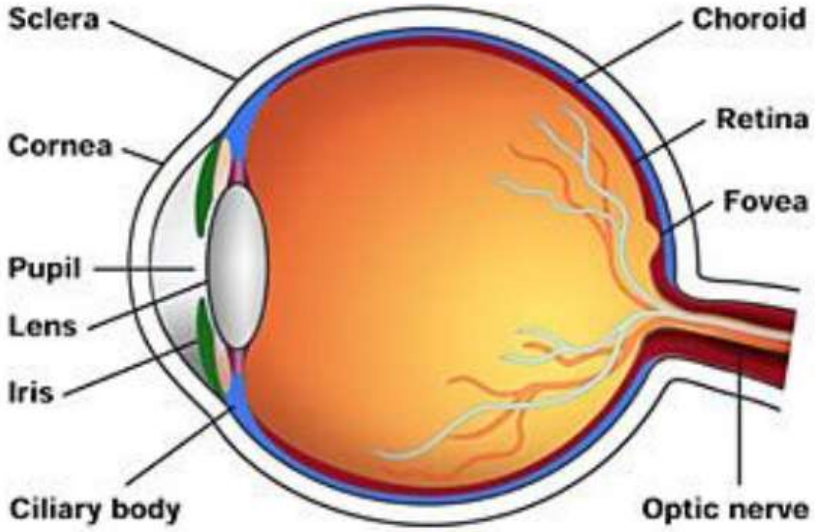


Figure 1-1: Anatomy of human eye.

- **The cornea** is the transparent outer surface of the front of the eye through which light passes. It serves as the most significant image-forming element of the eye since its curved surface at the interface with air represents the largest change in index of refraction in the eye's optical system. The cornea is avascular, receiving its nutrients from marginal blood vessels and the fluids surrounding it.

- **The lens** serves the function of accommodation. It is a layered, flexible structure that varies in index of refraction. It is a naturally occurring gradient-index optical element with the index of refraction higher in the center of the lens than at the edges. This feature serves to reduce some of the aberrations that might normally be present in a simple optical system. The shape of the lens is controlled by the ciliary muscles. When we gaze at a nearby object, the lens becomes 'fatter' and thus has increased optical power to allow us to focus on the near object. When we gaze at a distant object, the lens becomes 'flatter' resulting in the decreased optical power required to bring far away objects into sharp focus.
- **The Humors** is the volume between the cornea and lens is filled with the aqueous humor, which is essentially water. The region between the lens and retina is filled with vitreous humor, which is also a fluid, but with a higher viscosity, similar to that of gelatin. Both humors exist in a state of slightly elevated pressure (relative to air pressure) to assure that the flexible eyeball retains its shape and dimensions in order to avoid the deleterious effects of wavering retinal images.
- **The iris** is the sphincter muscle that controls pupil size. The iris is pigmented, giving each of us our specific eye color. Eye color is determined by the concentration and distribution of melanin within the iris.

- **The Retina** is the optical image formed by the eye is projected onto the retina. The retina is a thin layer of cells, approximately the thickness of tissue paper, located at the back of the eye and incorporating the visual system's photosensitive cells and initial signal processing and transmission 'circuitry.' These cells are neurons, part of the central nervous system, and can appropriately be considered a part of the brain. The photoreceptors, rods and cones, serve to transduce the information present in the optical image into chemical and electrical signals that can be transmitted to the later stages of the visual sys-tem. These signals are then processed by a network of cells and transmitted to the brain through the optic nerve.
- **The Fovea** is the most important structural area on the retina is the fovea. The fovea is the area on the retina where we have the best spatial and color vision. When we look at, or fixate, an object in our visual field, we move our head and eyes such that the image of the object falls on the fovea.
- **The macula** serves to protect this critical area of the retina from intense exposures to short-wavelength energy. It might also serve to reduce the effects of chromatic aberration that cause the short-wavelength image to be rather severely out of focus most of the time.
- **The optic nerve** is made up of the axons (outputs) of the ganglion cells, the last level of neural processing in the retina.

It is interesting to note that the optic nerve is made up of approximately one million fibers, carrying information generated by approximately 130 million photoreceptors. Thus there is a clear compression of the visual signal prior to transmission to higher levels of the visual system.

1.3. Photoreceptors

The retina includes several layers of neural cells, beginning with the photoreceptors, the rods and cones. A vertical signal processing chain through the retina can be constructed by examining the connections of photoreceptors to bipolar cells, which are in turn connected to ganglion cells, which form the optic nerve. Even this simple pathway results in the signals from multiple photoreceptors being compared and combined. This is because multiple photoreceptors provide input to many of the bipolar cells and multiple bipolar cells provide input to many of the ganglion cells. More importantly, this simple concept of retinal signal processing ignores two other significant types of cells. These are the horizontal cells, that connect photoreceptors and bipolar cells laterally to one another, and the amacrine cells, that connect bipolar cells and ganglion cells laterally to one another. Figure 1-2 provides only a slight indication of the extent of these various interconnections. it is important to realize that the signals transmitted from the retina to the higher levels of the brain via the ganglion cells are not simple point-wise representations of the receptor signals, but rather consist of sophisticated

combinations of the receptor signals. To envision the complexity of the retinal processing, keep in mind that each synapse between neural cells can effectively perform a mathematical operation (add, subtract, multiply, divide) in addition to the amplification, gain control, and nonlinearities that can occur within the neural cells.

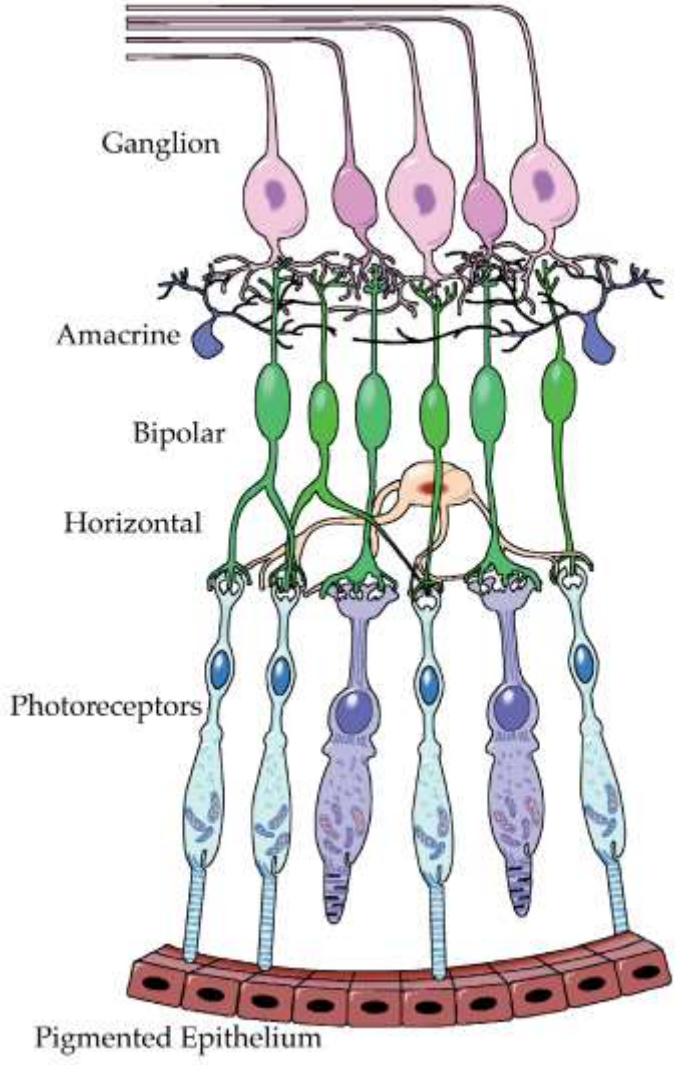


Figure 1-2: Connection between cells in the human retina.

Thus the network of cells within the retina can serve as a sophisticated image computer. This is how the information from

130 million photoreceptors can be reduced to signals in approximately one million ganglion cells without loss of visually meaningful data. It is interesting to note that light passes through all of the neural machinery of the retina prior to reaching the photoreceptors. This has little impact on visual performance since these cells are transparent and in fixed position, thus not perceived. It also allows the significant amounts of nutrients required and waste produced by the photoreceptors to be processed through the back of the eye. Figure 1-3 provides a representation of the two classes of retinal photoreceptors, rods and cones. Rods and cones derive their respective names from their prototypical shape. Rods tend to be long and slender while peripheral cones are conical. This distinction is misleading since foveal cones, which are tightly packed due to their high density in the fovea, are long and slender, resembling peripheral rods.

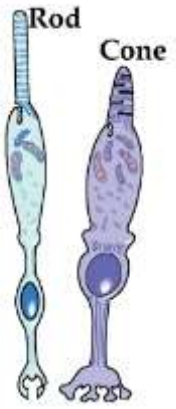


Figure 1-3: Rod and cone photoreceptors.

The more important distinction between rods and cones is in visual function. Rods serve vision at low luminance levels while

cones serve vision at higher luminance levels. Thus the transition from rod to cone vision is one mechanism that allows our visual system to function over a large range of luminance levels. At high luminance the rods are effectively saturated and only the cones function. In the intermediate luminance levels, both rods and cones function and contribute to vision. Vision when only rods are active is referred to as scotopic vision. Vision served only by cones is referred to as photopic vision and the term mesopic vision is used to refer to vision in which both rods and cones are active. There is only one type of rod receptor with a peak spectral responsivity at approximately 510 nm. There are three types of cone receptors with peak spectral responsivities spaced through the visual spectrum. The three types of cones are most properly referred to as L, M, and S cones. These names refer to the long-wavelength, middle-wavelength, and short-wavelength sensitive cones (Table 1-1), respectively. Sometimes the cones are denoted with other symbols such as RGB or ρ γ β suggestive of red, green, and blue sensitivities.

Table 1-1: Types of cones.

Cone	**pigment**	**spectral response**	**Peak**
S	β (blue)	400 - 500 nm	424 nm
M	γ (greenish blue)	450 - 630 nm	534 nm
L	ρ (yellowish green)	500 - 700 nm	564 nm

cones are the sensors in the eye responsible for color vision. Detailed experimental evidence has established that the 6 to 7

million cones in the human eye can be divided into three principal sensing categories, corresponding roughly to red, green, and blue. Approximately 65% of all cones are sensitive to red light, 33% are sensitive to green light, and only about 2% are sensitive to blue. How-ever, the blue cones are the most sensitive. Figure 1-4 shows average experimental curves detailing the absorption of light by the red, green, and blue cones in the eye. Because of these absorption characteristics, the human eye sees colors as variable combinations of the so-called primary colors: red (R), green (G), and blue (B).

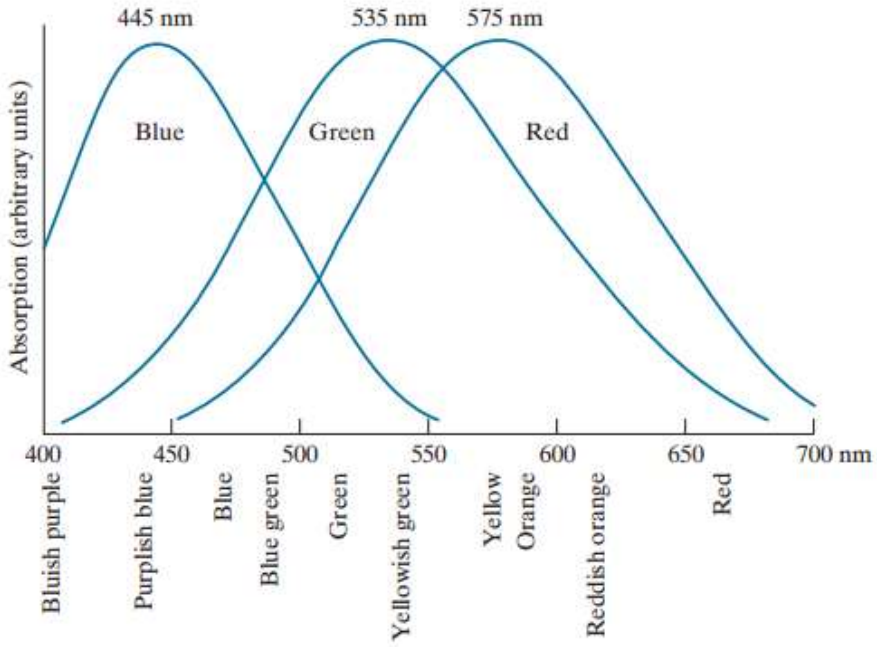


Figure 1-4: Rod and cone photoreceptors.

For the purpose of standardization, the CIE (Commission Internationale de l'Eclairage - the International Commission on Illumination) designated in 1931 the following specific wavelength values to the three primary colors: blue 435.8 nm,

green 546.1 nm and red 700 nm. This standard was set before results such as those in Figure 1-4 became available in 1965. Thus, the CIE standards correspond only approximately with experimental data. It is important to keep in mind that defining three specific primary color wavelengths for the purpose of standardization does not mean that these three fixed RGB components acting alone can generate all spectrum colors. Use of the word primary has been widely misinterpreted to mean that the three standard primaries, when mixed in various intensity proportions, can produce all visible colors. As you will see shortly, this interpretation is not correct unless the wavelength also is allowed to vary, in which case we would no longer have three fixed primary colors

1.4. Color fundamentals

Although the process employed by the human brain in perceiving and interpreting color is a physiopsychological phenomenon that is not fully understood, the physical nature of color can be expressed on a formal basis supported by experimental and theoretical results. Basically, the colors that humans and some other animals perceive in an object are determined by the nature of the light reflected from the object. The primary colors can be added together to produce the secondary colors of light— magenta (red plus blue), cyan (green plus blue), and yellow (red plus green). Mixing the three primaries, or a secondary with its

opposite primary color, in the right intensities produces white light. The characteristics generally used to distinguish one color from another are brightness, hue, and saturation.

- **Brightness**: embodies the achromatic notion of intensity.
- **Hue**: is an attribute associated with the dominant wavelength in a mixture of light waves. Hue represents dominant color as perceived by an observer. Thus, when we call an object red, orange, or yellow, we are referring to its hue.
- **Saturation**: refers to the relative purity or the amount of white light mixed with a hue.

The pure spectrum colors are fully saturated. Colors such as pink (red and white) and lavender (violet and white) are less saturated, with the degree of saturation being inversely proportional to the amount of white light added. Hue and saturation taken together are called chromaticity and, therefore, a color may be characterized by its brightness and chromaticity. The amounts of red, green, and blue needed to form any particular color are called the tristimulus values, and are denoted, X, Y, and Z, respectively. A color is then specified by its trichromatic coefficients, defined as:

$$x = \frac{X}{X + Y + Z} \tag{1-1}$$

$$y = \frac{Y}{X + Y + Z} \tag{1-2}$$

$$z = \frac{Z}{X + Y + Z} \tag{1-3}$$

We see from these equations that

$$x + y + z = 1 \tag{1-4}$$

For any wavelength of light in the visible spectrum, the tristimulus values needed to produce the color corresponding to that wavelength can be obtained directly from curves or tables that have been compiled from extensive experimental results. Another approach for specifying colors is to use the CIE chromaticity diagram (Figure 1-5), which shows color composition as a function of x (red) and y (green). For any value of x and y, the corresponding value of z (blue) is obtained from Eq. (1-4). The point marked green in Figure 1-5, for example, has approximately 62% green and 25% red content. It follows from Eq. (1-4) that the composition of blue is approximately 13%. The positions of the various spectrum colors— from violet at 380 nm to red at 780 nm — are indicated around the boundary of the tongue-shaped chromaticity diagram. These are the pure colors. Any point not actually on the boundary, but within the diagram, represents some mixture of the pure spectrum colors. The point of equal energy shown in Figure 1-5 corresponds to equal fractions of the three primary colors; it represents the CIE standard for white light. Any point located on the boundary of the chromaticity chart is fully saturated.

As a point leaves the boundary and approaches the point of equal energy, more white light is added to the color, and it becomes less saturated.

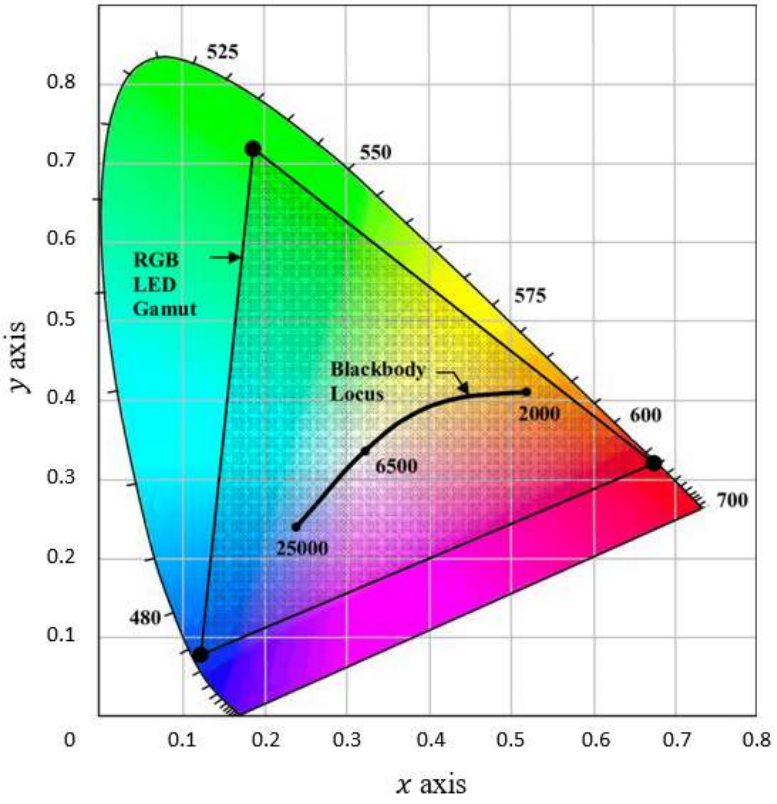


Figure 1-5: The CIE chromaticity diagram.

The chromaticity diagram is useful for color mixing because a straight-line segment joining any two points in the diagram defines all the different color variations that can be obtained by combining these two colors additively. Consider, for example, a straight line drawn from the red to the green points shown in Figure 1-5. If there is more red than green light, the exact point representing the new color will be on the line segment, but it will be closer to the red point than to the green point. Similarly, a line drawn from the point of equal energy to any point on the

boundary of the chart will define all the shades of that particular spectrum color. Extending this procedure to three colors is straightforward. To determine the range of colors that can be obtained from any three given colors in the chromaticity diagram, we simply draw connecting lines to each of the three color points. The result is a triangle, and any color inside the triangle, or on its boundary, can be produced by various combinations of the three vertex colors. A triangle with vertices at any three fixed colors cannot enclose the entire color region in Figure 1-5. This observation supports graphically the remark made earlier that not all colors can be obtained with three single, fixed primaries, because three colors form a triangle. The triangle in Figure 1-6 shows a representative range of colors (called the color gamut) produced by RGB monitors.

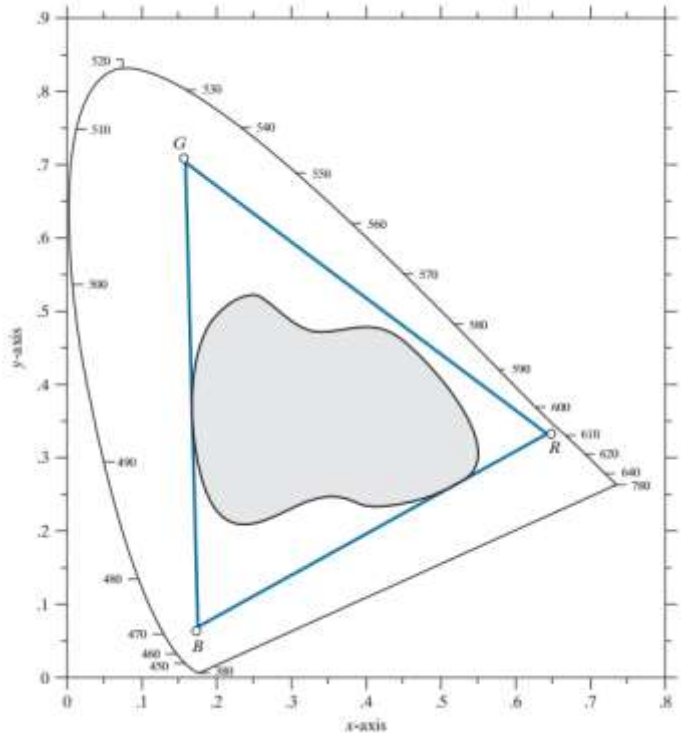


Figure 1-6: Illustrative color gamut of color monitors.

The shaded region inside the triangle illustrates the color gamut of today's high-quality color printing devices. The boundary of the color printing gamut is irregular because color printing is a combination of additive and subtractive color mixing, a process that is much more difficult to control than that of displaying colors on a monitor, which is based on the addition of three highly controllable light primaries.

1.5. Color models

The purpose of a color model (also called a color space or color system) is to facilitate the specification of colors in some standard way. In essence, a color model is a specification of:

- A coordinate system
- A subspace within that system

such that each color in the model is represented by a single point contained in that subspace. Most color models in use today are oriented either toward hardware (such as for color monitors and printers) or toward applications, where color manipulation is a goal. In terms of digital image processing, the hardware-oriented models most commonly used in practice are the RGB model for color monitors and a broad class of color video cameras; the CMY (Cyan, Magenta, Yellow) and CMYK (Cyan, Magenta, Yellow, Black) models for color printing; and the HSI (Hue, Saturation, Intensity) model, which corresponds closely with the way

humans describe and interpret color. The HSI model also has the advantage that it decouples the color and grayscale information in an image. There are numerous color models in use today. This is a reflection of the fact that color science is a broad field that encompasses many areas of application. It is tempting to dwell on some of these models here, simply because they are interesting and useful. However, keeping to the task at hand, we focus attention on a few models that are representative of those used in image processing.

1.5.1. RGB color model

In the RGB model, each color appears in its primary spectral components of red, green, and blue. This model is based on a Cartesian coordinate system. The color subspace of interest is the cube shown in Figure 1-7, in which RGB primary values are at three corners; the secondary colors cyan, magenta, and yellow are at three other corners; black is at the origin; and white is at the corner farthest from the origin. In this model, the grayscale (points of equal RGB values) extends from black to white along the line joining these two points. The different colors in this model are points on or inside the cube, and are defined by vectors extending from the origin. For convenience, the assumption is that all color values have been normalized so the cube is the unit cube. That is, all values of R, G, and B in this representation are assumed to be in the range [0, 1]. Note that the RGB primaries

can be interpreted as unit vectors emanating from the origin of the cube. Images represented in the RGB color model consist of three component images, one for each primary color. When fed into an RGB monitor, these three images combine on the screen to produce a composite color image. The number of bits used to represent each pixel in RGB space is called the pixel depth. Consider an RGB image in which each of the red, green, and blue images is an 8-bit image. Under these conditions, each RGB color pixel has a depth of 24 bits. The term full color image is used often to denote a 24-bit RGB color image.

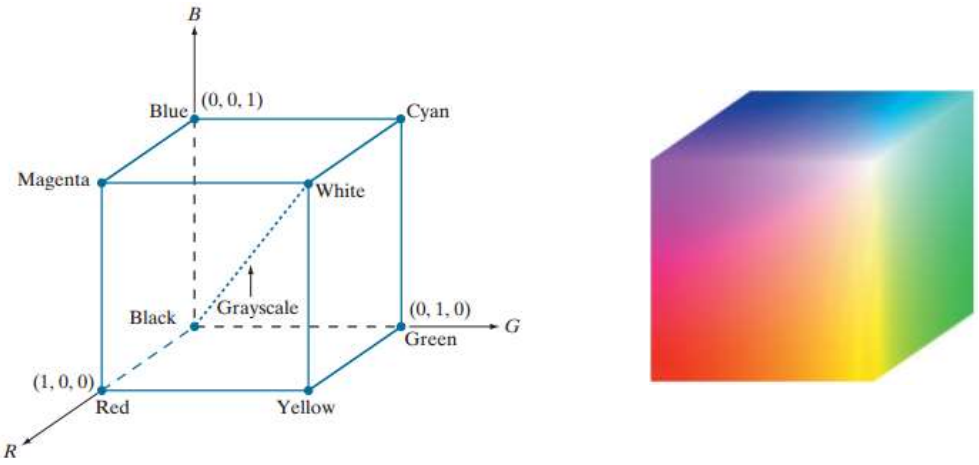


Figure 1-7: Schematic of the RGB color cube.

1.5.2. CMY and CMYK color model

Cyan, Magenta, and Yellow are the secondary colors of light or, alternatively, they are the primary colors of pigments. For example, when a surface coated with cyan pigment is illuminated with white light, no red light is reflected from the surface. That is, cyan subtracts red light from reflected white light, which itself

is composed of equal amounts of red, green, and blue light. Most devices that deposit colored pigments on paper, such as color printers and copiers, require CMY data input or perform an RGB to CMY conversion internally. This conversion is performed using the simple operation:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ G \end{bmatrix} \tag{1-5}$$

where the assumption is that all RGB color values have been normalized to the range [0,1]. Eq. (1-5) demonstrates that light reflected from a surface coated with pure cyan does not contain red. Similarly, pure magenta does not reflect green, and pure yellow does not reflect blue. Eq. (1-5) also reveals that RGB values can be obtained easily from a set of CMY values by subtracting the individual CMY values from 1.

In order to produce true black, a fourth color, black, denoted by K, is added, giving rise to the CMYK color model. The black is added in just the proportions needed to produce true black. Thus, when publishers talk about "four-color printing," they are referring to the three CMY colors, plus a portion of black. The conversion from CMY to CMYK begins by letting:

$$K = \min(C, M, Y) \tag{1-6}$$

If $K = 1$, then we have pure black, with no color contributions, from which it follows that:

$$C = M = Y = 0 \quad (1\text{-}7)$$

Otherwise,

$$C = (C - K)/(1 - K) \quad (1\text{-}8)$$

$$M = (M - K)/(1 - K) \quad (1\text{-}9)$$

$$Y = (Y - K)/(1 - K) \quad (1\text{-}10)$$

where all values are assumed to be in the range [0,1]. The conversions from CMYK back to CMY are:

$$C = C * (1 - K) + K \quad (1\text{-}11)$$

$$M = M * (1 - K) + K \quad (1\text{-}12)$$

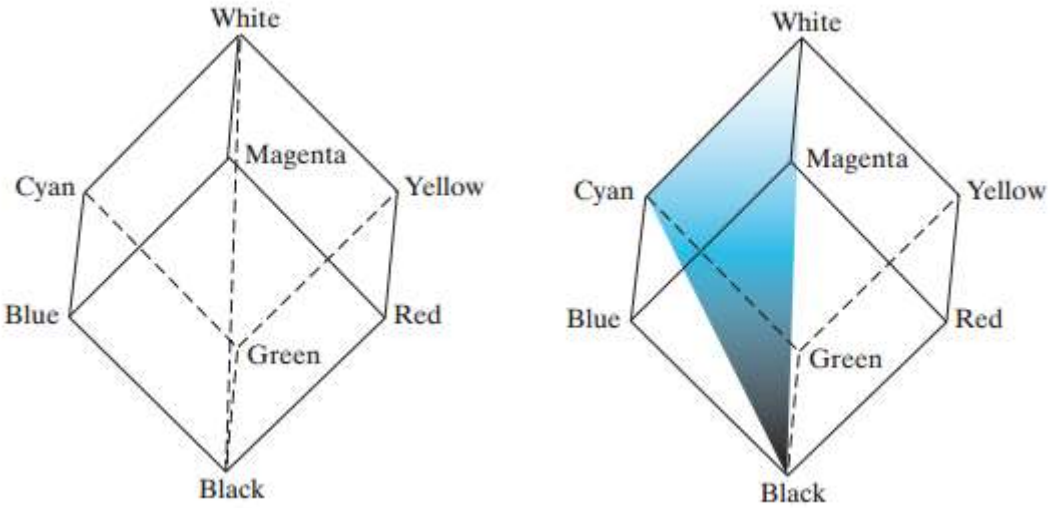
$$Y = Y * (1 - K) + K \quad (1\text{-}13)$$

1.5.3. HSI color model

Creating colors in the RGB, CMY, and CMYK models, and changing from one model to the other, is straightforward. These color systems are ideally suited for hardware implementations. In addition, the RGB system matches nicely with the fact that the human eye is strongly perceptive to red, green, and blue primaries. Unfortunately, the RGB, CMY, and other similar color models are not well suited for describing colors in terms that are practical for human interpretation. For example, one does not refer to the color of an automobile by giving the percentage of each of the primaries composing its color. Furthermore, we do not think of color images as being composed of three primary images that combine to form a single image. When humans view a color object, we describe it by its hue, saturation, and brightness. hue is

a color attribute that describes a pure color (pure yellow, orange, or red), whereas saturation gives a measure of the degree to which a pure color is diluted by white light. Brightness is a subjective descriptor that is practically impossible to measure. It embodies the achromatic notion of intensity and is one of the key factors in describing color sensation. We do know that intensity (gray level) is a most useful descriptor of achromatic images. This quantity definitely is measurable and easily interpretable. The model we are about to present, called the HSI (Hue, Saturation, Intensity) color model, decouples the intensity component from the color-carrying information (hue and saturation) in a color image. As a result, the HSI model is a useful tool for developing image processing algorithms based on color descriptions that are natural and intuitive to humans, who, after all, are the developers and users of these algorithms. We can summarize by saying that RGB is ideal for image color generation (as in image capture by a color camera or image display on a monitor screen), but its use for color description is much more limited. The material that follows provides an effective way to do this. An RGB color image is composed three grayscale intensity images, so it should come as no surprise that we can to extract intensity from an RGB image. This becomes clear if we take the color cube from Figure 1-7 and stand it on the black, (0,0,0) vertex, with the white, (1,1,1) vertex directly above it. The intensity (gray) scale is along the line joining these two vertices. In Figures 1-8a and 1-8b, the line

(intensity axis) joining the black and white vertices is vertical. Thus, if we wanted to determine the intensity component of any color point in Figure 1-8, we would simply define a plane that contains the color point and, at the same time, is perpendicular to the intensity axis.



(a) **(b)**

Figure 1-8: Conceptual relationships between the RGB and HSI color models.

The intersection of the plane with the intensity axis would give us a point with intensity value in the range [0,1]. A little thought would reveal that the saturation (purity) of a color increases as a function of distance from the intensity axis. In fact, the saturation of points on the intensity axis is zero, as evidenced by the fact that all points along this axis are gray. Hue can be determined from an RGB value also. Consider Figure 1-8b, which shows a plane defined by three points (black, white, and cyan). The fact that the black and white points are contained in the plane tells us that the

intensity axis also is contained in the plane. Furthermore, we see that all points contained in the plane segment defined by the intensity axis and the boundaries of the cube have the same hue (cyan in this case). We could arrive at the same conclusion that all colors generated by three colors lie in the triangle defined by those colors. If two of those points are black and white, and the third is a color point, all points on the triangle would have the same hue, because the black and white components cannot change the hue (of course, the intensity and saturation of points in this triangle would be different). By rotating the shaded plane about the vertical intensity axis, we would obtain different hues. From these concepts, we arrive at the conclusion that the hue, saturation, and intensity values required to form the HSI space can be obtained from the RGB color cube. That is, we can convert any RGB point to a corresponding point in the HSI color space by working out the formulas that describe the reasoning outlined in the preceding discussion. The key point regarding the cube arrangement in Figure 1-8, and its corresponding HSI color space, is that the HSI space is represented by a vertical intensity axis, and the locus of color points that lie on planes perpendicular to that axis. As the planes move up and down the intensity axis, the boundaries defined by the intersection of each plane with the faces of the cube have either a triangular or a hexagonal shape. This can be visualized much more readily by looking at the cube straight down its grayscale axis, as shown in Figure 1-9a. We see

that the primary colors are separated by 120°. The secondary colors are 60° from the primaries, which means that the angle between secondaries is 120° also. Figure 1-9b shows the same hexagonal shape and an arbitrary color point (shown as a dot).

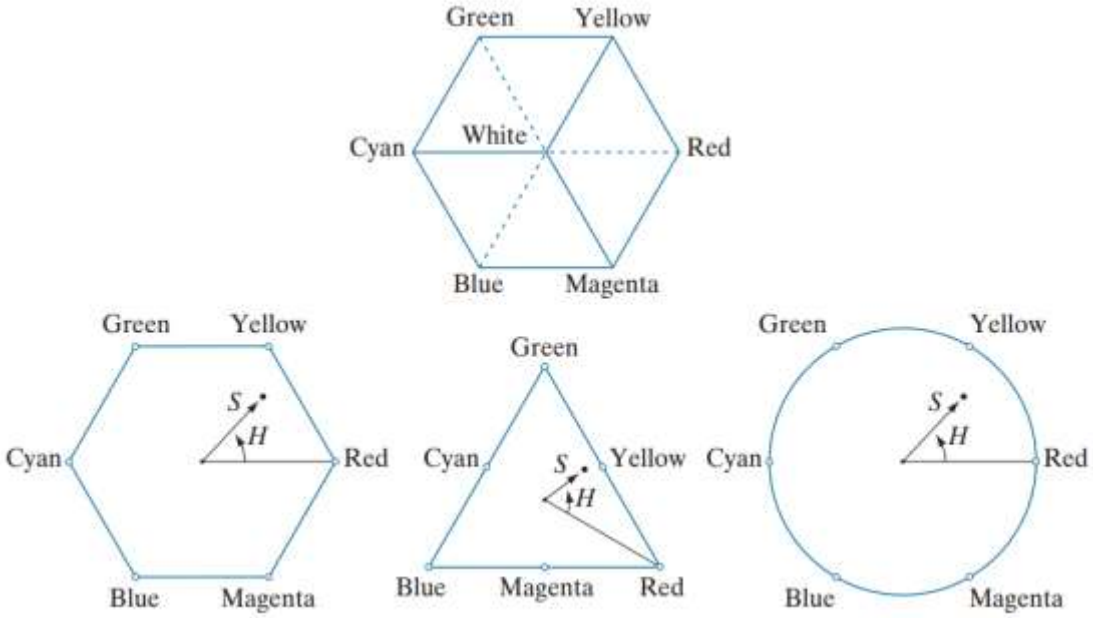


Figure 1-9: Hue and saturation in the HSI color model.

- **The hue of the point** is determined by an angle from some reference point. Usually (but not always) an angle of 0° from the red axis designates 0 hue, and the hue increases counter clockwise from there.
- **The saturation** (distance from the vertical axis) is the length of the vector from the origin to the point. Note that the origin is defined by the intersection of the color plane with the vertical intensity axis.

- **The vertical intensity axis** is the important components of the HSI color space, the length of the vector to a color point, and the angle this vector makes with the red axis.

Therefore, it is not unusual to see the HSI planes defined in terms of the hexagon just discussed, a triangle, or even a circle, as in Figure 1-9 show. The shape chosen does not matter because any one of these shapes can be warped into one of the other two by a geometric transformation. Figure 1-10 shows the HSI model based on color triangles, and on circles. Given an image in RGB color format, the H component of each RGB pixel is obtained using the equation:

$$H = \begin{cases} \theta & if\ B \leq G \\ 360 - \theta & if\ B > G \end{cases} \tag{1-14}$$

With

$$\theta = \left\{ \frac{\frac{1}{2}[(R - G) + (R - B)]}{[(R - G)^2 + (R - B)(G - B)]^{1/2}} \right\} \tag{1-15}$$

The saturation component is given by

$$S = 1 - \frac{3}{(R + G + B)}[\min(R, G, B)] \tag{1-16}$$

Finally, the intensity component is obtained from the equation

$$I = \frac{1}{3}(R + G + B) \tag{1-17}$$

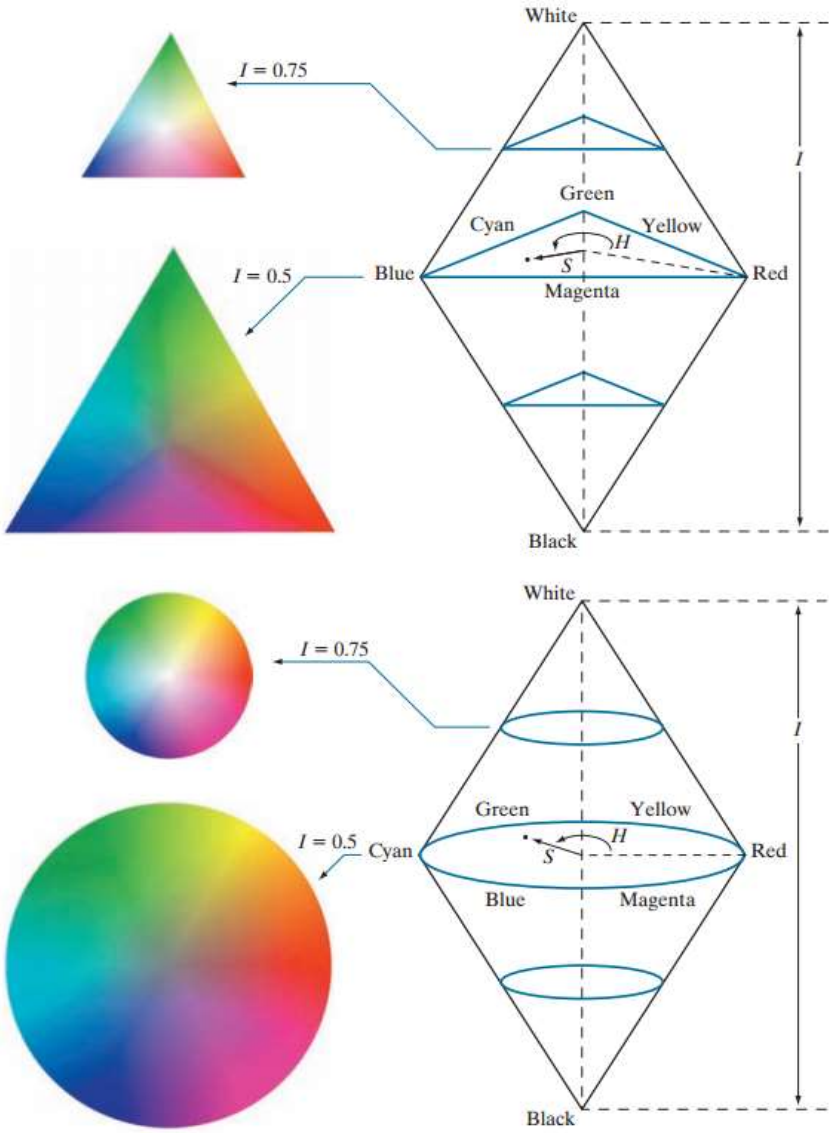


Figure 1-10: The HSI color model.

These equations assume that the RGB values have been normalized to the range [0,1], and that angle θ is measured with respect to the red axis of the HSI space, as in Figure 1-9. Hue can be normalized to the range [0,1] by dividing by 360° all values resulting from Eq.(1-14). The other two HSI components already are in this range if the given RGB values are in the interval [0,1].

Chapter 2

Color Vision Deficiency

2.1. Introduction

Color vision deficiency (CVD) is characterized by the inability or decreased ability in perception of some specific colors. CVD is further divisible into several types, of which red-green color vision defects are the most common whereas blue-yellow color vision defect is rarer. These kinds of CVDs interrupt color perception but do not impinge on the sharpness of vision. CVD may be X-linked recessive or an autosomal dominant or very rarely an autosomal recessive trait associated with X chromosome or chromosome 7 or 2 or 8 or 10. CVD in a subject is detected and its severity is analyzed by means of several tests, e.g. Ishihara Pseudo Chromatic Test, Lantern Test and Anomaloscope test. The prevalence of CVD varies worldwide and some parts of the world show considerable prevalence. It is very important to identify CVD affected individuals, mild or severe, at an early age by proper counseling in order to guide them in selection of the suitable career path. Few countries have initiated awareness and counseling program for CVD affected individuals, however it is imperative for all countries to take necessary steps in this regard. Color blindness affects a significant portion of the population. It affects 1 in 12 men and 1 in 200 women or around 4.5% of the entire world population. If the world population is estimated at 7 billion, around 315 million people suffer from some form of color blindness. People with color blindness have difficulties in many aspects of life. Problems range from occupations where color

identification is an integral part (assembling electronics) to driving (traffic lights as in Figure 2-1). Some countries have even refused to grant driving licenses to color blind people (Romania). Color is a vital tool for processing information. Therefore, color blind people also have difficulty while cooking (unable to see when the food is "done"), understanding graphs, playing sports etc.



Figure 2-1: how colorblind people see the traffic lights.

Due to the significant number of people affected and the lack of feasible existing solutions the research to develop an accurate color identification system needs to be undertaken. This system could help identify the parts when building a circuit; help differentiate colors to make reading a map easier etc. The system could be adapted to use with augmented reality systems such as the Google glass (Figure 2-2). The applications of this system are many therefore research in this field is vital.



Figure 2-2: Google glass color identification system.

2.2. Color Vision Deficiencies

There are various types of inherited and acquired CVD according to its character and severity. CVD common types as in [1,2] are:

- Protanopia (Red deficiency).
- Deuteranopia (Green deficiency).
- Tritanopia (Blue deficiency).
- Protanomaly (Red weakness).
- Deuteranomaly (Green weakness).
- Tritanomaly (Blue weakness).
- Achromatopia (Absolute color blindness).

Some CVDs are caused by the lack of a particular type of cone photo pigment. Since there are three types of cone photo pigments, there are three general classes of these color vision deficiencies, namely Protanopia, Deuteranopia, and Tritanopia. An observer with protanopia, known as a protanope, is missing the L-cone photo pigment and therefore is unable to discriminate

reddish and greenish hues since the red–green opponent mechanism cannot be constructed. A deuteranope is missing the M-cone photo pigment and therefore also cannot distinguish reddish and greenish hues due to the lack of a viable red–green opponent mechanism (Figure 2-3). Protanopes and deuteranopes can be distinguished by their relative luminous sensitivity since it is constructed from the summation of different cone types. The protanopic luminous sensitivity function is shifted toward shorter wave-lengths. A tritanope is missing the S-cone photo pigment and therefore cannot discriminate yellowish and bluish hues due to the lack of a yellow–blue opponent mechanism. There are also anomalous trichromats who have trichromatic vision, but the ability to discriminate particular hues is reduced either due to shifts in the spectral sensitivities of the photo pigments or contamination of photo-pigments (e.g., some L-cone photo pigment in the M-cones). Among the anomalous trichromats are those with any of the following: Protanomaly (weak in L-cone photo pigment or L-cone absorption shifted toward shorter wavelengths), Deuteranomaly is weakness in M-cone photo pigment or M-cone absorption shifted toward longer wavelengths. Tritanomaly weakness in S-cone photo pigment or S-cone absorption shifted toward longer wavelengths). There are also some cases of cone monochromatic (effectively only one cone type) and rod monochromatic (no cone responses). While it is impossible for a person with normal color vision to experience

what the visual world looks like to a person with a color vision deficiency, it is possible to illustrate the hues that become indistinguishable.

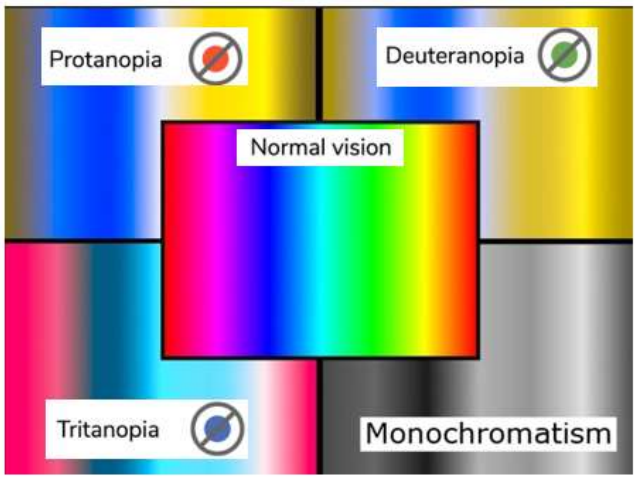


Figure 2-3: Color vision spectrum for color blind people.

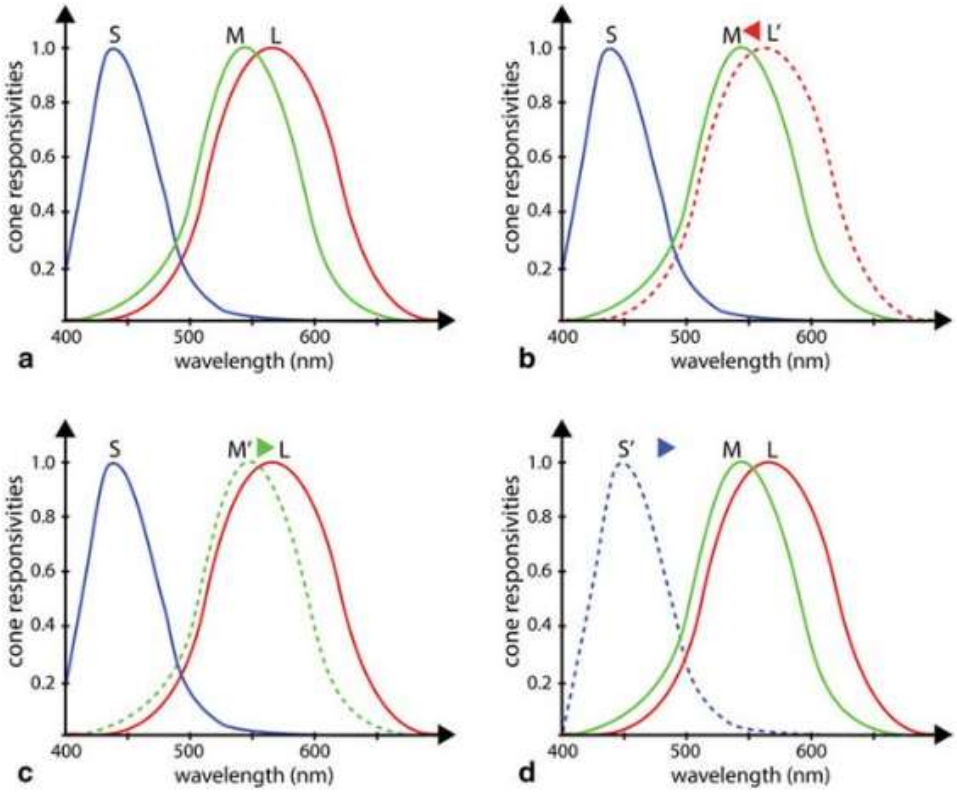


Figure 2-4: The spectral sensitivity functions. a) Normal trichromacy. b) Protanomaly (anomalous L cone type). c) Deuteranomaly (anomalous M cone type). d) Tritanomaly (anomalous S cone type).

2.2.1. CVD Causes

The most common cause of color blindness is the fault in the development of one or more sets of cones due to genetics. However, there are other non-hereditary causes:

- **Kallman's syndrome**: This is an inherited condition where the primary symptom is a failure to start puberty or to complete it. It is prevalent in both genders and has many additional symptoms such as an altered sense of smell. Color blindness is sometimes also a symptom of this condition.
- **Parkinson's disease**: This is a degenerative disorder of the central nervous system caused by the death of dopamine generative cells. Early symptoms include shaking, rigidity, slowness of movement, behavioral problems etc. Because this is a neurological disorder, light sensitive nerve cells in the retina may be damaged leading to color blindness.
- **Shaken Baby Syndrome**: Any physical damage at a young age to the brain or the retina can cause permanent damage to an infant. One effect of such damage can be permanent color blindness.
- **UV damage**: Over exposure to ultra-violet rays causes degeneration of the retina. This is one of the most common causes of color blindness however only presents itself later in the patient's life.

- **Cataracts**: A cataract is a clouding of the lens of the eye. The most common cause of cataracts is ageing. By age 80, more than half of all Americans either have had a cataract or have had cataract surgery. The cloudiness caused makes colors less bright making it harder to distinguish them.
- **Medications**: Ample prescribed medications have a side effect which causes color blindness. For example, Tiagabine which is an antiepileptic drug which has been shown to reduce the user's color vision.
- **Hereditary Causes**: The most usual cause of color blindness is the inheritance of a genetically mutated gene on the X chromosome. Research has also shown that mutations that can lead to color blindness originate from in excess of 19 different chromosomes. Females have two X chromosomes where males only have one. Therefore, in females, if one X chromosome is defective, the normal one will prevent the woman from becoming color blind.

This explains why 1 in 12 men but only 1 in 200 women suffer from any type of color blindness. Women who have a flawed gene and are carriers, their sons have a 50% chance of being color blind, and their daughters have a 50% chance of becoming a carrier. Women who suffer from color blindness as a result of mutated X chromosomes will typically have a carrier mother, and a color blind father, getting two mutated X chromosomes. There are also inherited diseases such as Cone Dystrophy,

Achromatopsia, Blue cone monochratism, Retinitis pigmentosa, Retinoblastoma etc. that can lead to color blindness.

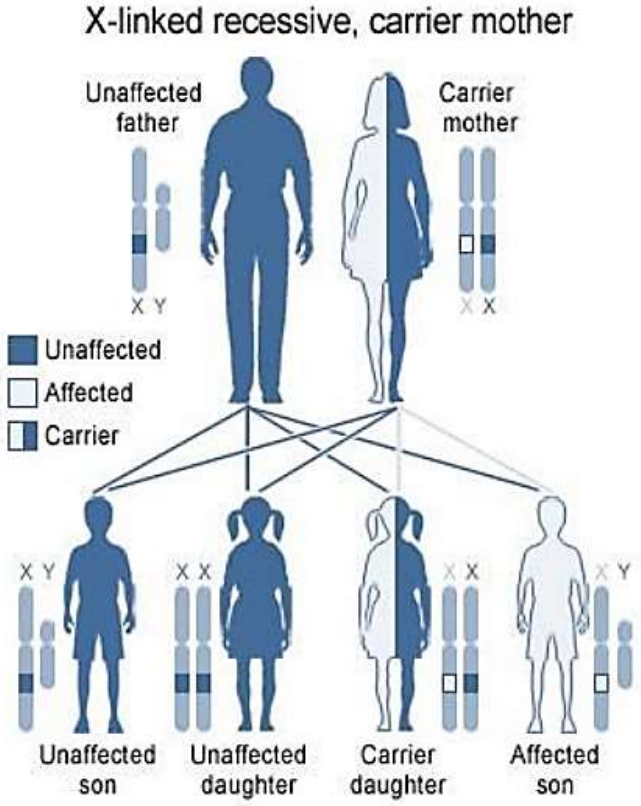
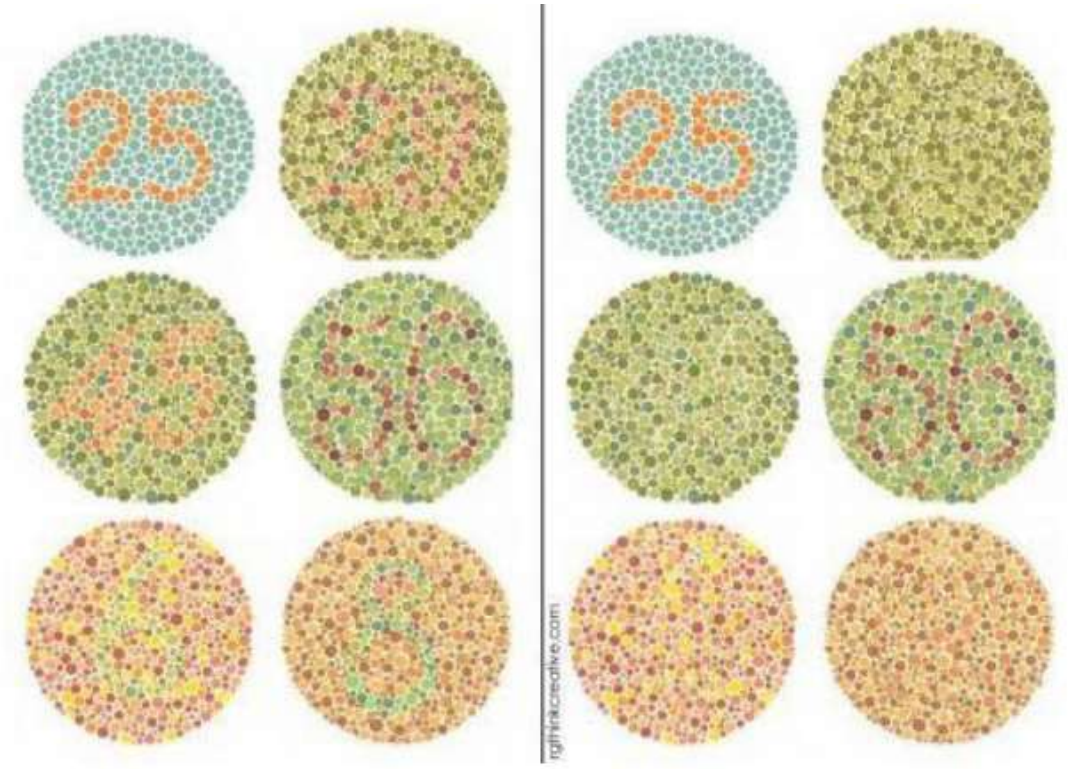


Figure 2-5: Genetic Tree showing possible outcomes.

2.2.2. CVD Diagnosis

Color blindness can be hard to identify, predominantly in children with inherited color vision deficiency as they may be uninformed that they have any problems with their color vision. A child with a condition such as Deuteranopia may seemingly be able to accurately identify colors which they can't see (e.g. red) because they have been taught the color of objects from an early age and will know for example that the sky is blue and cucumbers are green even if they cannot actually see those colors. The most common test for color blindness is the Ishihara test. This test is

useful for red/green color blindness but does not work for blue color blindness. The test consists of 38 plates of circles which contains numbers imbedded with an irregular pattern. The number is of a different color therefore those with normal vision should be able to identify them. However, color blind people cannot accurately distinguish colors therefore cannot identify the number. Special tests have been developed for young infants as they cannot correctly identify numbers. The plates can identify the type and the severity of color blindness present.

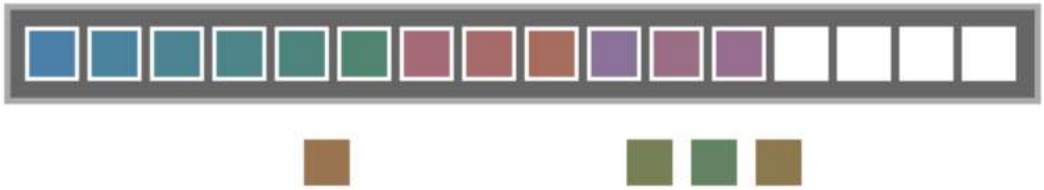


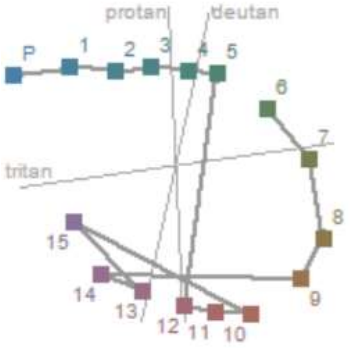
Normal View Color blind view

Figure 2-6: Ishihara test for colorblindness.

There's another an Alternative test called color arrangement test. This test differs from the Ishihara test style as it utilizes vector mathematics to plot and formulate results, offering a formulated

method of defining the user's condition. To take the test the user is asked to arrange the colors in an ordered list. After the user has completed this task a graph is generated outlining the user's performance. A detailed table outlining the metrics of the user's performance along with the severity of the user's condition is also displayed.





The thick line describes your order of the test plates. People with normal color vision order them in a circle (P, 1, 2, ..., 15). Crossings indicate some form of color blindness. Paralleslism of crossings to a confusion line (protan, deutan, tritan) is a clue for the type of your color blindness.

Figure 2-7: Ishihara test for colorblindness.

Chapter 3

Color Detection and Tracking System Design

3.1. Introduction

In this section a color detection and tracking system is designed. The design based on ESP32-CAM Arduino board (Figure 3-1). The ESP32 can act as a server for a browser client. A web page is designed for clients. OpenCV.js runs in the web page.

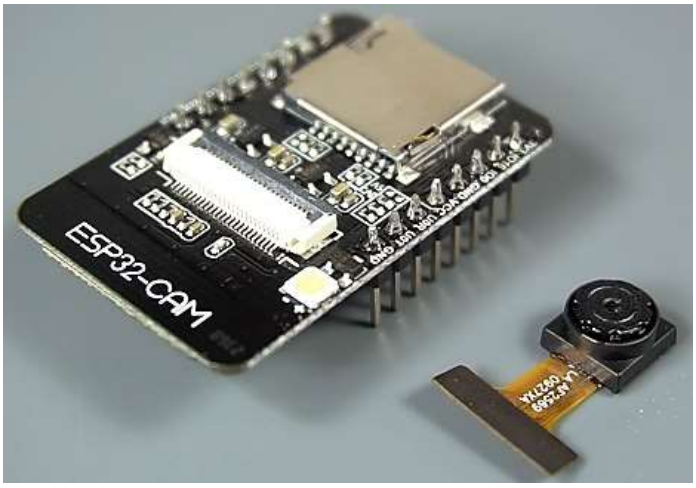


Figure 3-1: ESP32-CAM.

3.2. ESP32-CAM

The ESP32-CAM is a very small camera module with the ESP32-S chip that costs approximately 10$. Besides the OV2640 camera, and several General Purposes Inputs Outputs (GPIOs) to connect peripherals, it also features a microSD card slot that can be useful to store images taken with the camera or to store files to serve to clients. ESP32-CAM has a lot of features, such as:

- The smallest 802.11b/g/n Wi-Fi BT SoC module
- Low power 32-bit CPU, can also serve the application processor

- Up to 160MHz clock speed, summary computing power up to 600 DMIPS.
- Built-in 520 KB SRAM, external 4MPSRAM.
- Supports UART/SPI/I2C/PWM/ADC/DAC.
- Support OV2640 and OV7670 cameras, built-in flash lamp.
- Support image WiFI upload.
- Support TF card.
- Supports multiple sleep modes.
- Supports STA/AP/STA+AP operation mode.
- Support SmartConfig/AirKiss technology.
- Support for serial port local and remote firmware upgrades.

Figure 3-2 shows the pinout of ESP32-CAM.

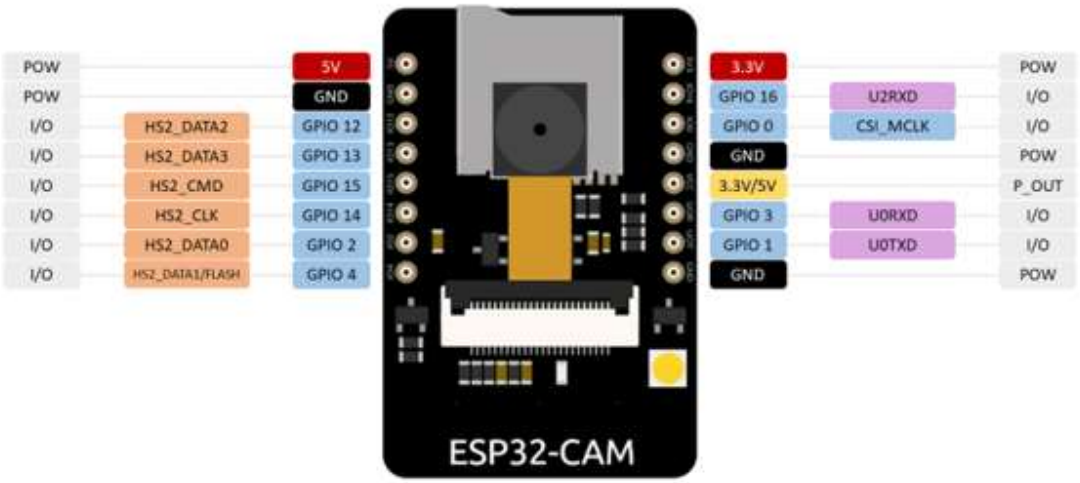


Figure 3-2: ESP32-CAM pinout.

The ESP32-CAM comes with three GND pins (colored in black color) and two power pins (colored with red color): 3.3V and 5V. ESP32-CAM can be powered through the 3.3V or 5V pins.

However, many people reported errors when powering the ESP32-CAM with 3.3V.

Power output pin

There's also the pin labeled on the silkscreen as VCC (colored with a yellow rectangle). This pin should not use to power the ESP32-CAM. That is an output power pin. It can either output 5V or 3.3V. In our case, the ESP32-CAM outputs 3.3V whether it is powered with 5V or 3.3V. Next to the VCC pin, there are two pads. One labeled as 3.3V and other as 5V.

Serial Pins

GPIO 1 and GPIO 3 are the serial pins (TX and RX, respectively). Because the ESP32-CAM doesn't have a built-in programmer, you need to use these pins to communicate with the board and upload code. GPIO 1 and GPIO 3 can be used to connect other peripherals like outputs or sensors after uploading the code.

GPIO 0

GPIO 0 determines whether the ESP32 is in flashing mode or not. This GPIO is internally connected to a pull-up 10k Ohm resistor. When GPIO 0 is connected to GND, the ESP32 goes into flashing mode and you can upload code to the board. To make the ESP32 run "normally", you just need to disconnect GPIO 0 from GND.

MicroSD Card Connections

The following pins are used to interface with the microSD card when it is on operation.

Table 3-1: MicroSD card Connection

MicroSD card	ESP32
CLK	GPIO 14
CMD	GPIO 15
DATA0	GPIO 2
DATA1 / flashlight	GPIO 4
DATA2	GPIO 12
DATA3	GPIO 13

All these GPIOs are RTC and support ADC: GPIOs 2, 4, 12, 13, 14, and 15.

Flashlight (GPIO 4)

The ESP32-CAM has a very bright built-in LED that can work as a flash when taking photos. That LED is internally connected to GPIO 4. That GPIO is also connected to the microSD card slot, so you may have troubles when trying to use both at the same time – the flashlight will light up when using the microSD card. Note: one of our readers shared that if you initialize the microSD card as follows, you won't have this problem because the microSD card won't use that data line. That LED works with inverted logic, so you send a LOW signal to turn it on and a HIGH signal to turn it off.

Camera Connections

The connections between the camera and the ESP32-CAM AI-Thinker are shown in the following table.

Table 3-2: Camera Connection

OV2640 CAMERA	ESP32	Variable name in code
D0	GPIO 5	Y2_GPIO_NUM
D1	GPIO 18	Y3_GPIO_NUM
D2	GPIO 19	Y4_GPIO_NUM
D3	GPIO 21	Y5_GPIO_NUM
D4	GPIO 36	Y6_GPIO_NUM
D5	GPIO 39	Y7_GPIO_NUM
D6	GPIO 34	Y8_GPIO_NUM
D7	GPIO 35	Y9_GPIO_NUM
XCLK	GPIO 0	XCLK_GPIO_NUM
PCLK	GPIO 22	PCLK_GPIO_NUM
VSYNC	GPIO 25	VSYNC_GPIO_NUM
HREF	GPIO 23	HREF_GPIO_NUM
SDA	GPIO 26	SIOD_GPIO_NUM
SCL	GPIO 27	SIOC_GPIO_NUM
POWER PIN	GPIO 32	PWDN_GPIO_NUM

3.3. Design Coding

The program consists of two parts:

- The server program which runs on the ESP32 Camera.
- The client program which runs on the Web browser.

The web server allows color tracking of a moving object. On the web server interface, we can properly select the color we want to track. Then, the browser sends the real time x and y coordinates of the center of mass of the moving object to the ESP32 board.



Figure 3-3: Color track bar.

We code the next instructions in browser to allow the Camera image and OpenCV.js, which have different origins to work together in the program. Without these instructions, the browser throws errors.

```
client.println("HTTP/1.1 200 OK");
client.println("Access-Control-Allow-Origin: *");
client.println("Access-Control-Allow-Headers:
Origin, X-Requested-With, Content-Type, Accept");
client.println("Access-Control-Allow-Methods:
GET,POST,PUT,DELETE,OPTIONS");
client.println("Content-Type: image/jpeg");
client.println("Content-Disposition: form-data;
name=\"imageFile\"; filename=\"picture.jpg\"");
client.println("Content-Length: " + String(fb-
>len));
client.println("Connection: close");
client.println();
```

We communicate the “center-of-mass” of a color target detected by the client to the ESP32 server to detect colors.

```
void ExecuteCommand() {
  if (cmd!="colorDetect")
  }
  if (cmd=="resetwifi") {
  WiFi.begin(P1.c_str(), P2.c_str());
  Serial.print("Connecting to ");
  Serial.println(P1);
  long int StartTime=millis();
```

```
while (WiFi.status() != WL_CONNECTED)
      {
            delay(500);
            if ((StartTime+5000) < millis()) break;
      }
      Serial.println("");
      Serial.println("STAIP:
"+WiFi.localIP().toString());
      Feedback="STAIP: "+WiFi.localIP().toString();
   }
   else if (cmd=="restart") {
      ESP.restart();
   }
   else if (cmd=="cm"){
      int XcmVal = P1.toInt();
      int YcmVal = P2.toInt();
      Serial.println("cmd= "+cmd+" ,VALXCM=
"+XcmVal);
      Serial.println("cmd= "+cmd+" ,VALYCM=
"+YcmVal);
   }
   else if (cmd=="quality") {
      sensor_t * s = esp_camera_sensor_get();
      int val = P1.toInt();
      s->set_quality(s, val);
   }
   else if (cmd=="contrast") {
      sensor_t * s = esp_camera_sensor_get();
      int val = P1.toInt();
      s->set_contrast(s, val);
   }
   else if (cmd=="brightness") {
      sensor_t * s = esp_camera_sensor_get();
      int val = P1.toInt();
      s->set_brightness(s, val);
   }
   else {
      Feedback="Command is not defined.";
   }
   if (Feedback=="") {
      Feedback=Command;
   }
}
```

These are used to set limits to the color range of colors allowed in the "processed" image in the CV application.

```
<div class="section">
<h2>RGB Color Trackbars</h2>
<table>
   <tr>
      <td>R min:   <span
id="RMINdemo"></span></td>
      <td><input type="range" id="rmin" min="0"
max="255" value="0" class = "slider"></td>
      <td>R max:   <span
id="RMAXdemo"></span></td>
      <td><input type="range" id="rmax" min="0"
max="255" value="50" class = "slider"></td>
   </tr>
   <tr>
      <td>G min:   <span
id="GMINdemo"></span></td>
      <td><input type="range" id="gmin" min="0"
max="255" value="0" class = "slider"></td>
      <td>G max:   <span
id="GMAXdemo"></span></td>
      <td><input type="range" id="gmax" min="0"
max="255" value="50" class = "slider"></td>
   </tr>
   <tr>
      <td>B min:   <span id
="BMINdemo"></span></td>
      <td><input type="range" id="bmin" min="0"
max="255" value="0" class = "slider"></td>
      <td>B max:<span id="BMAXdemo"></span></td>
      <td> <input type="range" id="bmax" min="0"
max="255" value="50" class = "slider"></td>
   </tr>
</table>
</div>
var RMAXslider = document.getElementById("rmax");
var RMAXoutput =
document.getElementById("RMAXdemo");
RMAXoutput.innerHTML = RMAXslider.value;
RMAXslider.oninput = function() {
   RMAXoutput.innerHTML = this.value;
   RMAX = parseInt(RMAXoutput.innerHTML,10);
   console.log("RMAX=" + RMAX);
}
```

```
console.log("RMAX=" + RMAX);
var RMINslider = document.getElementById("rmin");
var RMINoutput =
document.getElementById("RMINdemo");
RMINoutput.innerHTML = RMINslider.value;
RMINslider.oninput = function(){
RMINoutput.innerHTML = this.value;
  RMIN = parseInt(RMINoutput.innerHTML,10);
  console.log("RMIN=" + RMIN);}
console.log("RMIN=" + RMIN);
var GMAXslider = document.getElementById("gmax");
var GMAXoutput =
document.getElementById("GMAXdemo");
GMAXoutput.innerHTML = GMAXslider.value;
GMAXslider.oninput = function(){
  GMAXoutput.innerHTML = this.value;
  GMAX = parseInt(GMAXoutput.innerHTML,10);}
console.log("GMAX=" + GMAX);
var GMINslider = document.getElementById("gmin");
var GMINoutput =
document.getElementById("GMINdemo");
GMINoutput.innerHTML = GMINslider.value;
GMINslider.oninput = function(){
  GMINoutput.innerHTML = this.value;
  GMIN = parseInt(GMINoutput.innerHTML,10);}
console.log("GMIN=" + GMIN);
var BMAXslider = document.getElementById("bmax");
var BMAXoutput =
document.getElementById("BMAXdemo");
BMAXoutput.innerHTML = BMAXslider.value;
BMAXslider.oninput = function(){
  BMAXoutput.innerHTML = this.value;
  BMAX = parseInt(BMAXoutput.innerHTML,10);}
console.log("BMAX=" + BMAX);
var BMINslider = document.getElementById("bmin");
var BMINoutput =
document.getElementById("BMINdemo");
BMINoutput.innerHTML = BMINslider.value;
BMINslider.oninput = function(){
  BMINoutput.innerHTML = this.value;
  BMIN = parseInt(BMINoutput.innerHTML,10);
}
console.log("BMIN=" + BMIN);
```

The maximum and minimum values of red, green, and blue (RGB) are applied to the OpenCV function, inRange()

```
let high = new
cv.Mat(src.rows,src.cols,src.type(),[RMAX,GMAX,BMAX,
255]);
let low = new
cv.Mat(src.rows,src.cols,src.type(),[RMIN,GMIN,BMIN,
0]);
cv.inRange(src,low,high,mask1);
//inRange(source image, lower limit, higher limit,
destination image)
cv.threshold(mask1,mask,THRESH_MIN,255,cv.THRESH_BIN
ARY);
//threshold(source image,destination
image,threshold,255,threshold method);
```

The image is 4channel; RGBA where A is the level of transparency. In this tutorial, A will set at 100% opacity, namely 255. The code is based on the fact that; besides the A plane, the image has 3 color planes, RGB, each pixel in each plane having a value between 0 and 255. The high/low limits are applied to the corresponding color planes for each pixel. The threshold() function examines the composite source image pixel value and sets the corresponding destination value at either 0 or 255 depending on whether the source value is less or greater than the threshold value. The top image in the right-hand column shows this binary image. For the sake of completeness, an invert capability has been added to the binary image. When the INVERT button in the web page is clicked, the binary image is inverted (black becomes white, white becomes black) and subsequent processing is performed on the new image. The button is bistable, so that a second push returns the binary image to its original state.

Once the binary image is deemed acceptable, the **TRACKING** button, which is bistable, can be clicked.

```
if(b_tracker == true){
try{
 if(b_invert==false){
cv.findContours(mask,contours,hierarchy,cv.RETR_C
OMP,cv.CHAIN_APPROX_SIMPLE);
//findContours(source image, array of contours
found, hierarchy of contours
// if contours are inside other contours, method
of contour data retrieval,
//algorithm method)
}
else{

cv.findContours(mask2,contours,hierarchy,cv.RETR_
CCOMP,cv.CHAIN_APPROX_SIMPLE);
}
console.log("CONTOUR_SIZE = " + contours.size());

//draw contours
if(b_contour==true){
   for(let i = 0; i < contours.size(); i++){

cv.drawContours(src,contours,i,[0,0,0,255],2,cv.L
INE_8,hierarchy,100)
   }
}
```

If the tracking button is pressed when the binary image is fully black, the instructions depending on findContours output will throw exceptions; the try-catch allows the program to continue safely, posting an output in the console log and the text box.

Contours.sizecontours is the output of findContours and is an array of contours of the white object(s) found in the binary image. Contours.size() finds the number of elements in the array. The

hierarchy (contours inside other contours) output is not of concern here as there will be no white objects (outlined in black) inside other white objects. marks finding the largest area contour in the array of contours using the MaxAreaArg function and transmitting the centroid, x_cm, y_cm to the ESP32 via a fetch instruction.

```
for(let k = 0; k < contours.size(); k++){
   cnt = contours.get(k);
   Moments = cv.moments(cnt,false);
   M00Array[k] = Moments.m00;}
let max_area_arg = MaxAreaArg(M00Array);
console.log("MAXAREAARG = "+max_area_arg);
let ArgMaxArea = MaxAreaArg(M00Array);
if(ArgMaxArea >= 0){
cnt = contours.get(MaxAreaArg(M00Array));   //use
the contour with biggest M00
Moments = cv.moments(cnt,false);
M00 = Moments.m00;
M10 = Moments.m10;
M01 = Moments.m01;
x_cm = M10/M00;      // 75 for circle_9.jpg
y_cm = M01/M00;      // 41 for circle_9.jpg
XCMoutput.innerHTML = Math.round(x_cm);
YCMoutput.innerHTML = Math.round(y_cm);
console.log("M00 = "+M00);
console.log("XCM = "+Math.round(x_cm));
console.log("YCM = "+Math.round(y_cm));
fetch(document.location.origin+'/?cm='+Math.round(
x_cm)+';'+Math.round(y_cm)+';stop');
console.log("M00ARRAY = " + M00Array);
```

Conclusion

We designed a system for helping people with color vision deficiency for recognizing colors correctly. The design consisted of an Arduino board ESP32-CAM

References

[1] Color Blindness, Available at:

http://www.colorblindawareness.org/color-blindness/

[2] Causes & incidence of Color Blindness, Available

at: http://www.webexhibits.org/causesofcolor/2C.html

[3] Color Blindness, Available at:

http://www.allaboutvision.com/conditions/colordeficiency.htm

[4] Deeb SS, Motulsky AG [2011]. Red Green Color vision defect. GeneRevies ® [Internet]; Available from http://www.ncbi.nlm.nih.gov/books/NBK1301/

[5] Karim JK and Mohammed AS. Prevalence of congenital Red Green color vision defects among various ethnic groups of students in Erbil city. Jordan Journal of Biological Sciences. Volume 6[3]: 235-238

[6] Deller, J.R; Proakis, J.G; Hansen, J.HL, "Discrete-Time Processing of Speech Signals". Macmillan Publishing Company, New York 1993.

[7] Hess, W, "Pitch Determination of Speech Signals". Springer-Verlag, Berlin 1983.

[8] Jayant, N.S.; Noll, P. "Digital Coding of Waveforms.", Prentice-Hall, 1984.