

Various Color Models Used in Digital Image Processing

This article illustrates different color models and their properties with use-cases



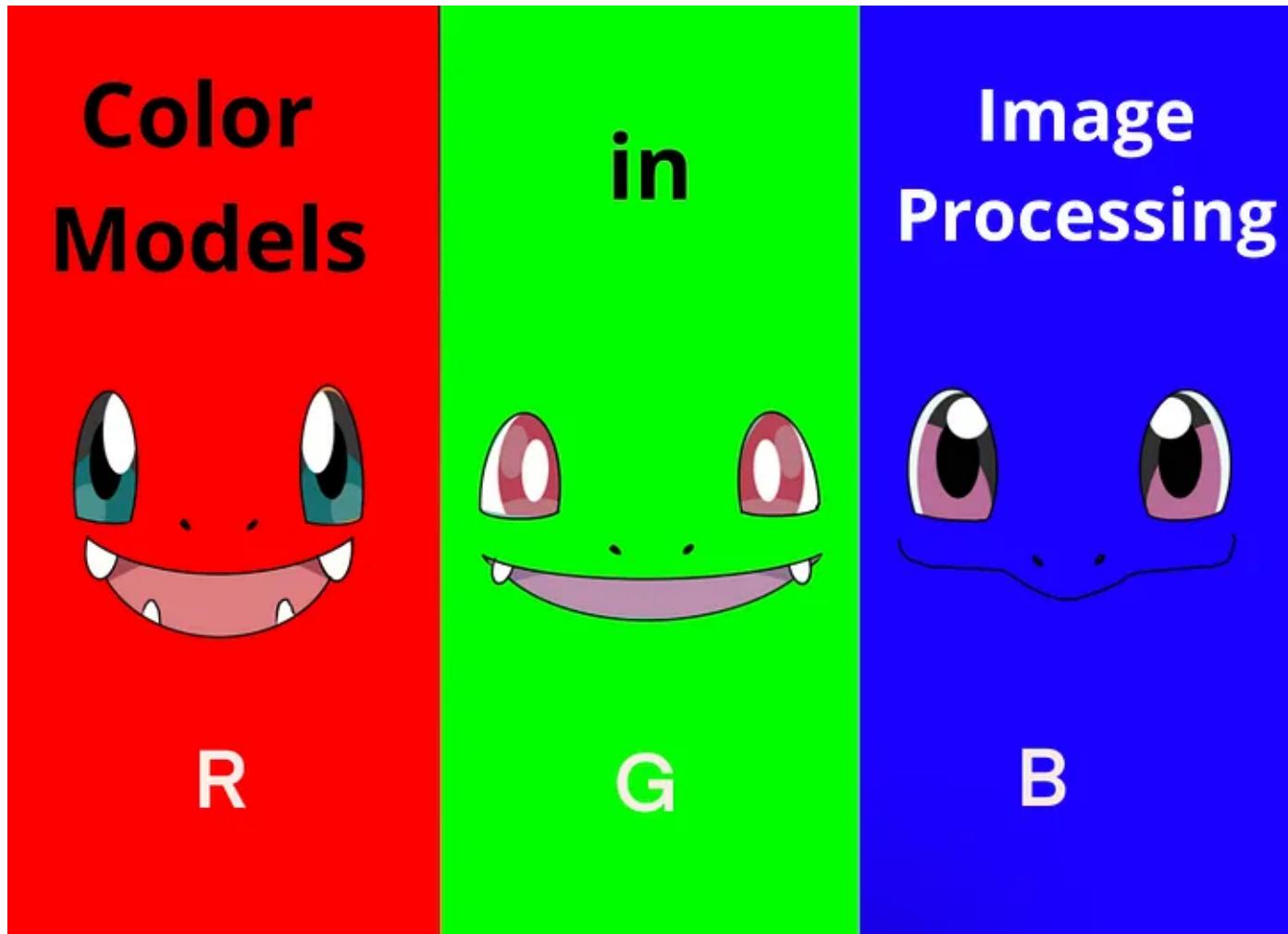
Prateek Chhikara · [Follow](#)

Published in Level Up Coding

7 min read · Apr 12, 2022

Listen

Share



Cover Art depicting RGB channels. (Source: Image by the author)

The somewhat psychological way human cognition formulates colored imagery based on wavelengths cannot be used in machines. Instead, machines use a color model, a mathematical approximation of the naturally non-quantifiable nature of human visual perception [1]. Many color models exist, and presumably, they all have advantages and disadvantages that make them more or less suitable for a given application. This article discusses the most commonly used color models that are as follows.

1. **RGB (RED – GREEN – BLUE)**
2. **CMY and CMYK (CYAN – MAGENTA – YELLOW)**
3. **HSI (HUE – SATURATION – INTENSITY)**

Before deep diving into the color models let us first understand the difference between Additive and Subtractive color models [6].

Additive Color Model

1. These type of models use light which is emitted directly from a source to display colors.
2. These models mixes different amount of RED, GREEN, and BLUE (primary colors) light to produce rest of the colors.
3. Adding these three primary colors results in WHITE image.
4. Example: RGB model is used for digital displays such as laptops, TVs, tablets, etc.

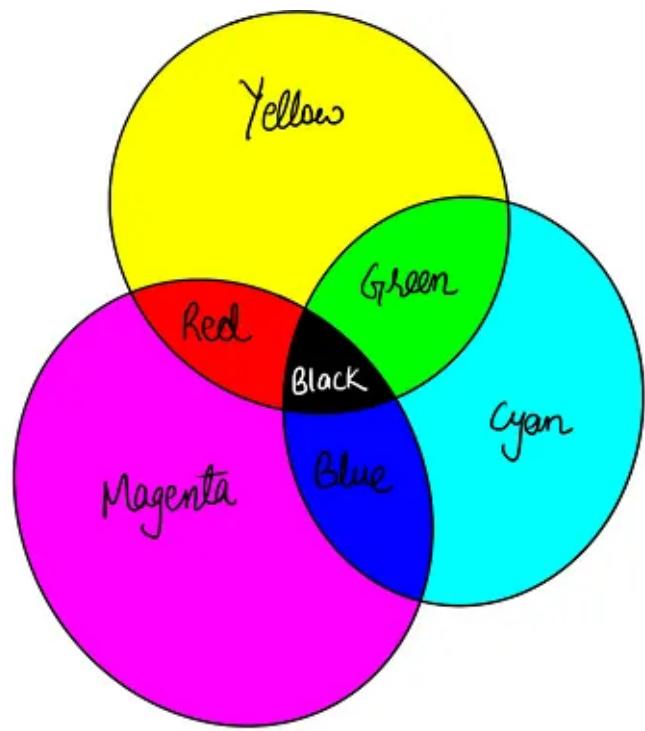
Subtractive Color Model

1. These type of models use printing inks to display colors.
2. Subtractive color starts with an object that reflects light and uses colorants to subtract portions of the white light illuminating an object to produce other colors.
3. If an object reflects all the white light back to the viewer, it appears white, and if it absorbs all the light then it appears black.
4. Example: Graphic designers used the CMYK model for printing purpose.

Figure 1 shows the vein diagrams of both Additive and Subtractive color models.



**Mixture of Light
(Additive Primaries)**



**Mixture of Pigments
(Subtractive Primaries)**

Figure 1. Vein Diagram for Additive and Subtractive Color Model types. (Source: Image by the author)

1. RGB

The model's name comes from the initials of the three additive primary colors, red, green, and blue. The RGB color model is an additive color model in which red, green, and blue are added together in various ways to reproduce a wide range of colors [4].

Usually, in RGB a pixel is represented using 8 bits for each red, green, and blue. This creates a total of around 16.7 million colors (2^{24}). Equal values of these three primary colors represents shade of gray color ranging from black to white.

Let's plot these three primary colors on a 3-dimensional plane in the form of a cube, as shown in Figure 2; the RGB values will be at the corners present on the three axes. The origin will be black, and the diagonal opposite to the origin will be black. The rest three corners of the cube will be cyan, magenta, and yellow. Inside the cube, we get a variety of colors represented by the RGB vector (origin at black).

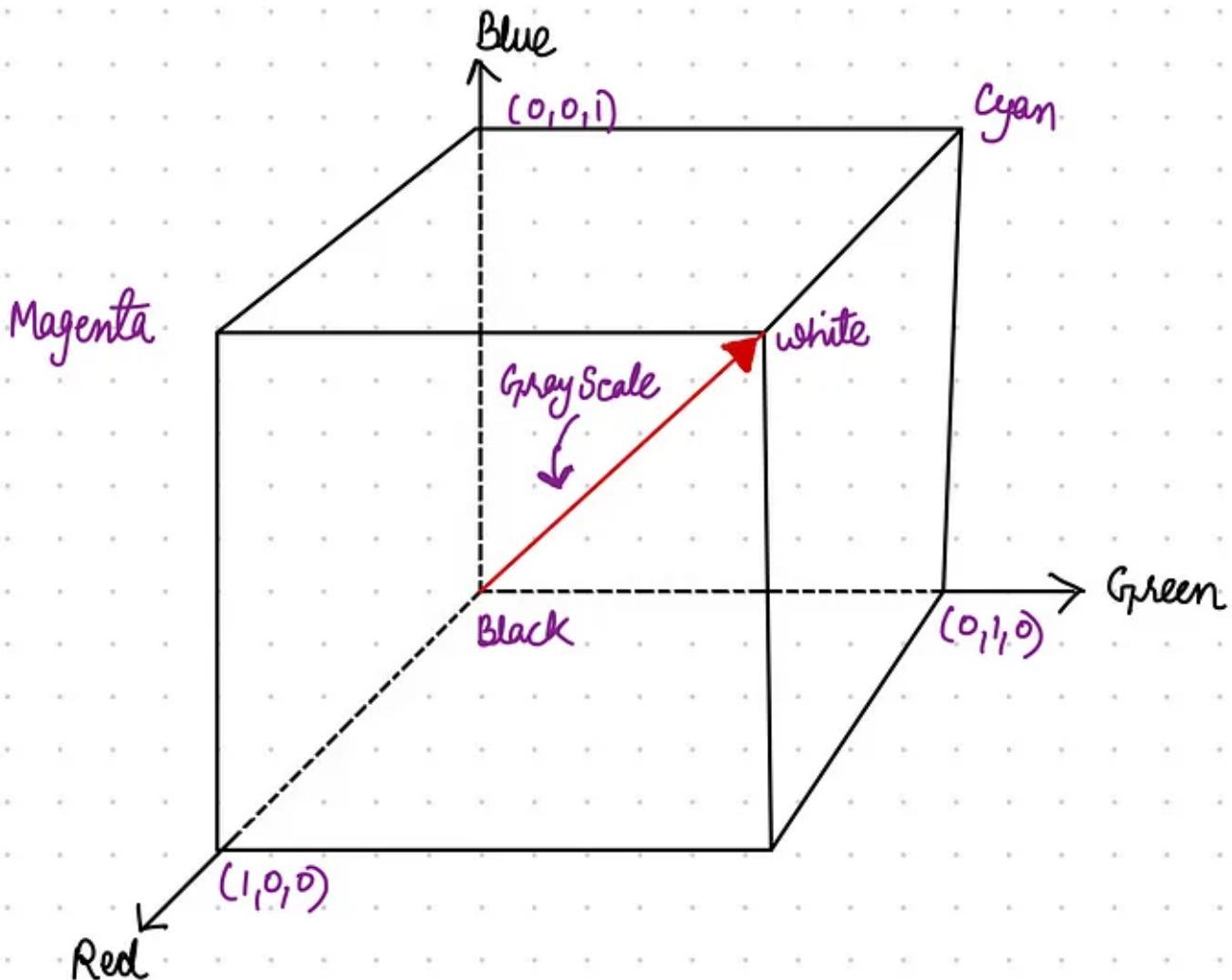


Figure 2. RGB color plot in a 3-dimensional plane. (Source: Image by the author)

With the help of the primary colors, we can generate secondary colors (Yellow, Cyan, and Magenta) as follows.

Colour combination:

Green(255) + Red(255) = Yellow

Green(255) + Blue(255) = Cyab

Red(255) + Blue(255) = Magenta

Red(255) + Greeb(255) + Blue(255) = White

2. CMY and CMYK

The CMY color model is a subtractive color model in which cyan, magenta, and yellow (secondary colors) pigments or dyes are mixed in different ways to produce a broad range of colors [5]. The secondary colors are also called the primary color pigments. The CMY color model itself does not describe what is meant by cyan, magenta, and

yellow colorimetrically, so the mixing results are not specified as absolute but relative to the primary colors. When the exact chromaticities of the cyan, magenta, and yellow primaries are defined, the color model then becomes an absolute color space.

The Process of Color Subtraction

The methodology of color subtraction is a valuable way of predicting the ultimate color appearance of an object if the color of the incident light and the pigments are known [2]. The relationship between the RGB and CMY color models is given by:

$$\text{RGB} = 1 - \text{CMY} \text{ or } \text{CMY} = 1 - \text{RGB}$$

In other words, red light consumes cyan and vice versa, magenta absorbs green and vice versa, & yellow absorbs blue and vice versa. To understand the process of color subtraction, let's consider a surface of yellow pigment. When a white light ($R+G+B$) is incident on this yellow surface, the blue lights will get absorbed and we will see only the combination of red and green light as shown in Figure 3.

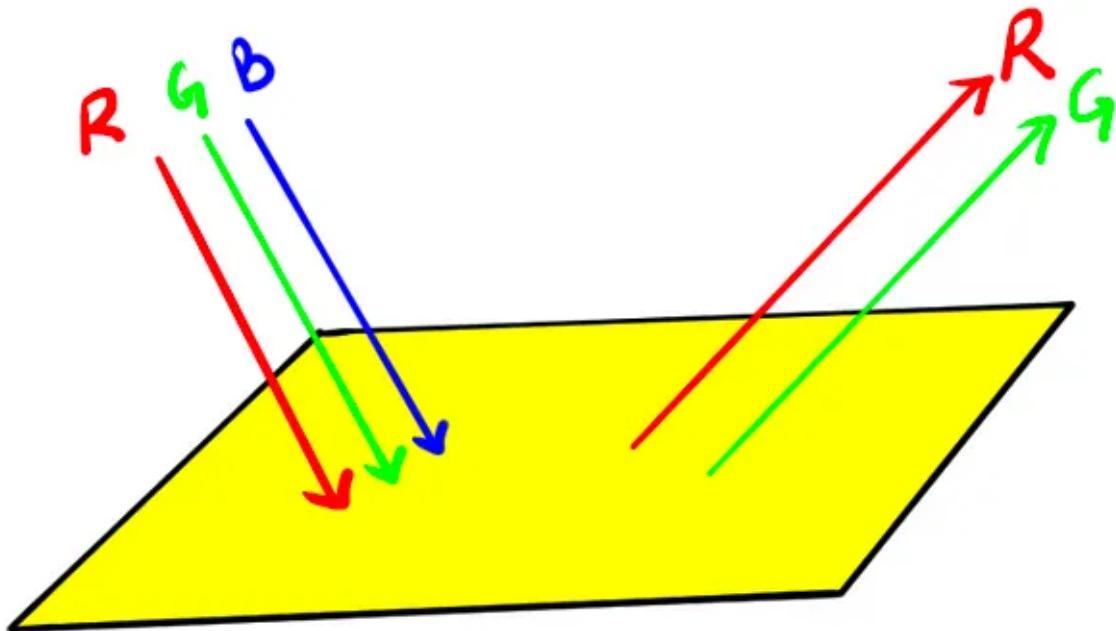


Figure 3. Color Subtraction Example 1. (Source: Image by the author)

Similarly, if we throw a magenta light, a combination of red and blue, on a yellow pigment, the result will be a red light because the yellow pigment absorbs the blue light, as shown in Figure 4.

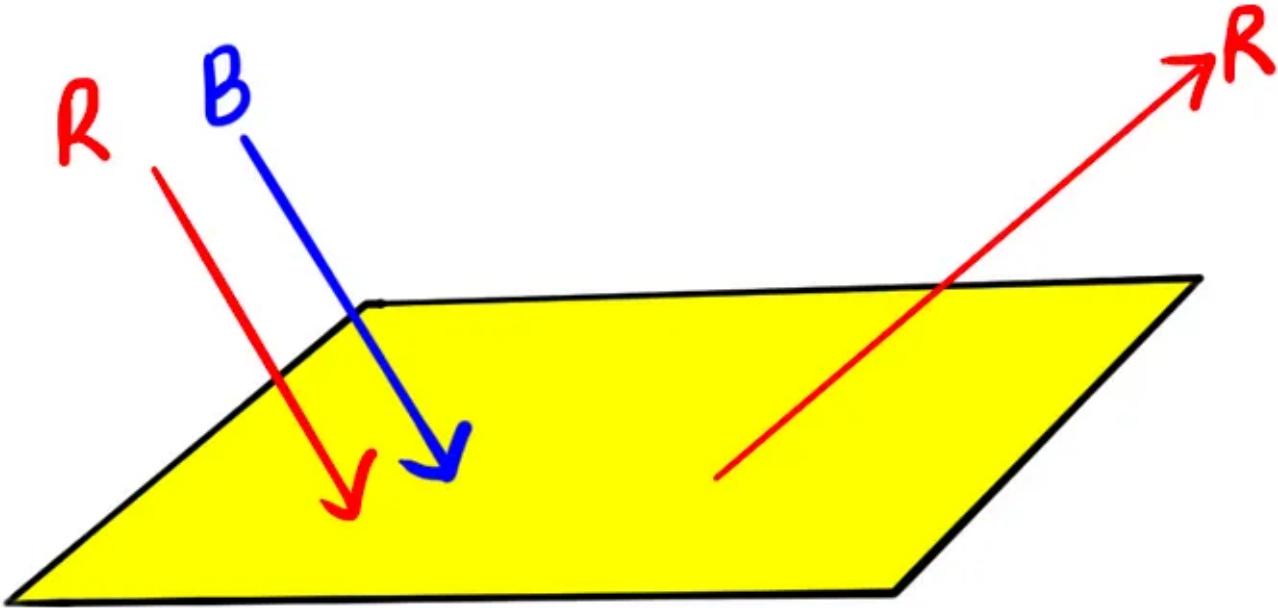


Figure 4. Color Subtraction Example 2. (Source: Image by the author)

CMYK

CYMK color model is used in hardcopy devices. According to the theory, 100% cyan, 100% magenta, and 100% yellow would result in pure black [3]. With today's printing colors, it is not practicable to achieve this, so in the area of printing, the additional component key 'K' (black) is required. The black color obtained by mixing the C, M, and Y color pigments is muddy or sluggish.

3. HSI

The RGB and CMY color models are ideally suited for hardware implementations; however, these are not reasonably suited for representing colors in terms that are practical for human interpretation [7]. RGB is not a particularly intuitive way to describe colors. HSI stands for Hue, Saturation, and Intensity. When humans view a color object, its hue, saturation, and brightness are described.

1. **Hue:** It is a color attribute that describes a pure color.
2. **Saturation:** It measures the extent to which a pure color is diluted by white light.
3. **Brightness:** It depends upon color intensity, which is a key factor in describing the color sensation. The intensity is easily measurable, and the results are also easily interpretable.

Note: *RGB is great for color generation, but HSI is great for color description.*

Calculation of Hue, Sauturation, and Intensity

If we stand the cube (shown in Figure 2) on the black vertex and position the white vertex directly above it, we will get something similar to what is shown in Figure 5. The diagonal (from black to white) or the height of the standing cube represents intensity. At the lowest level, the intensity is zero, whereas at the top the intensity is maximum ($=1$).

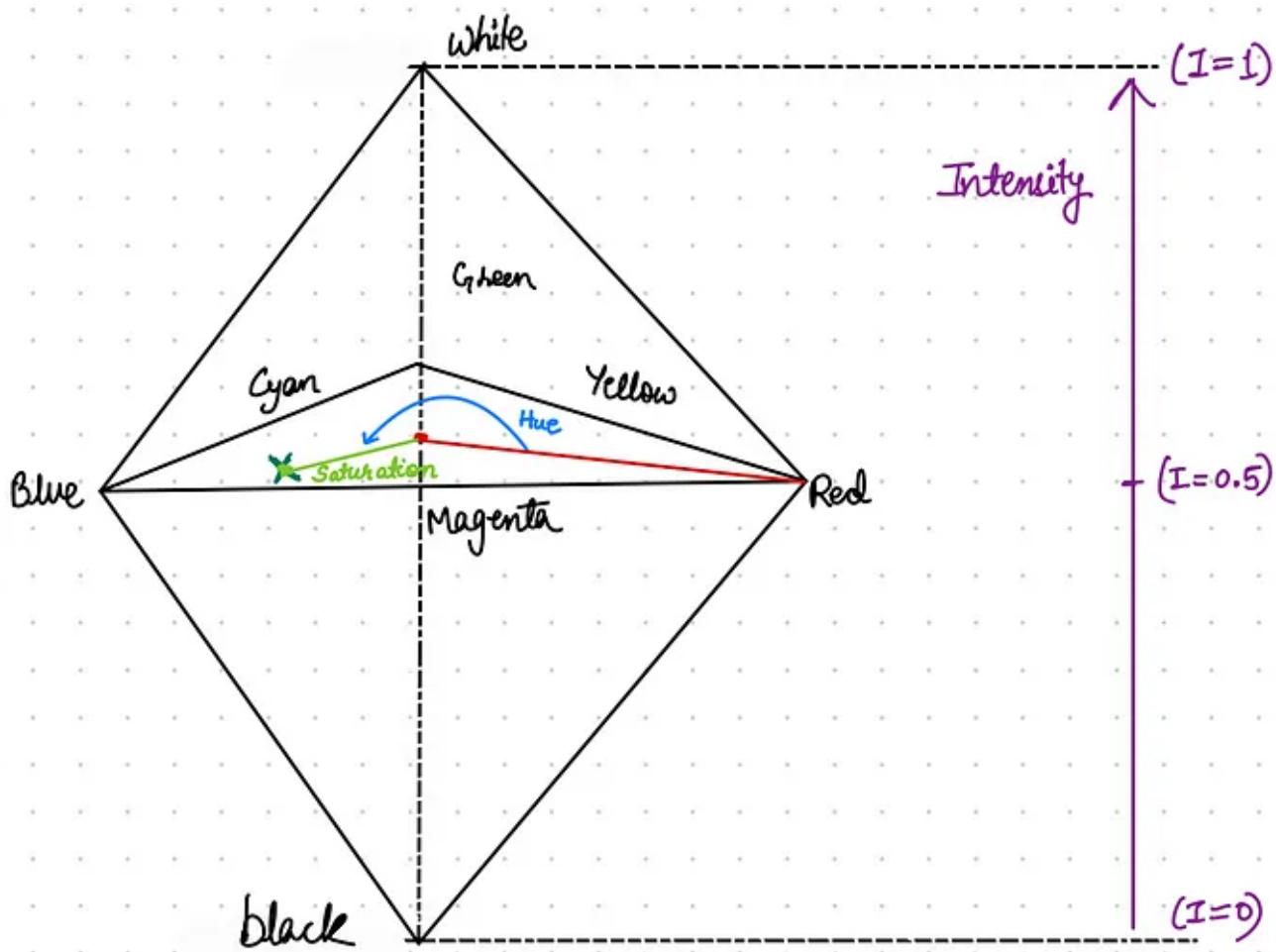


Figure 5. Color cube . (Source: Image by the author)

Therefore the HSI model comprises a vertical intensity axis and the locus of color points that lie on planes perpendicular to that axis.

Now, suppose we look straight down at the RGB cube as arranged previously; we would see a hexagonal shape with each primary color separated by 120° and secondary colors at 60° from the primaries, as shown in Figure 6. In this, the hue is the angle from a reference point, usually red, and saturation is the distance from the origin to the point.

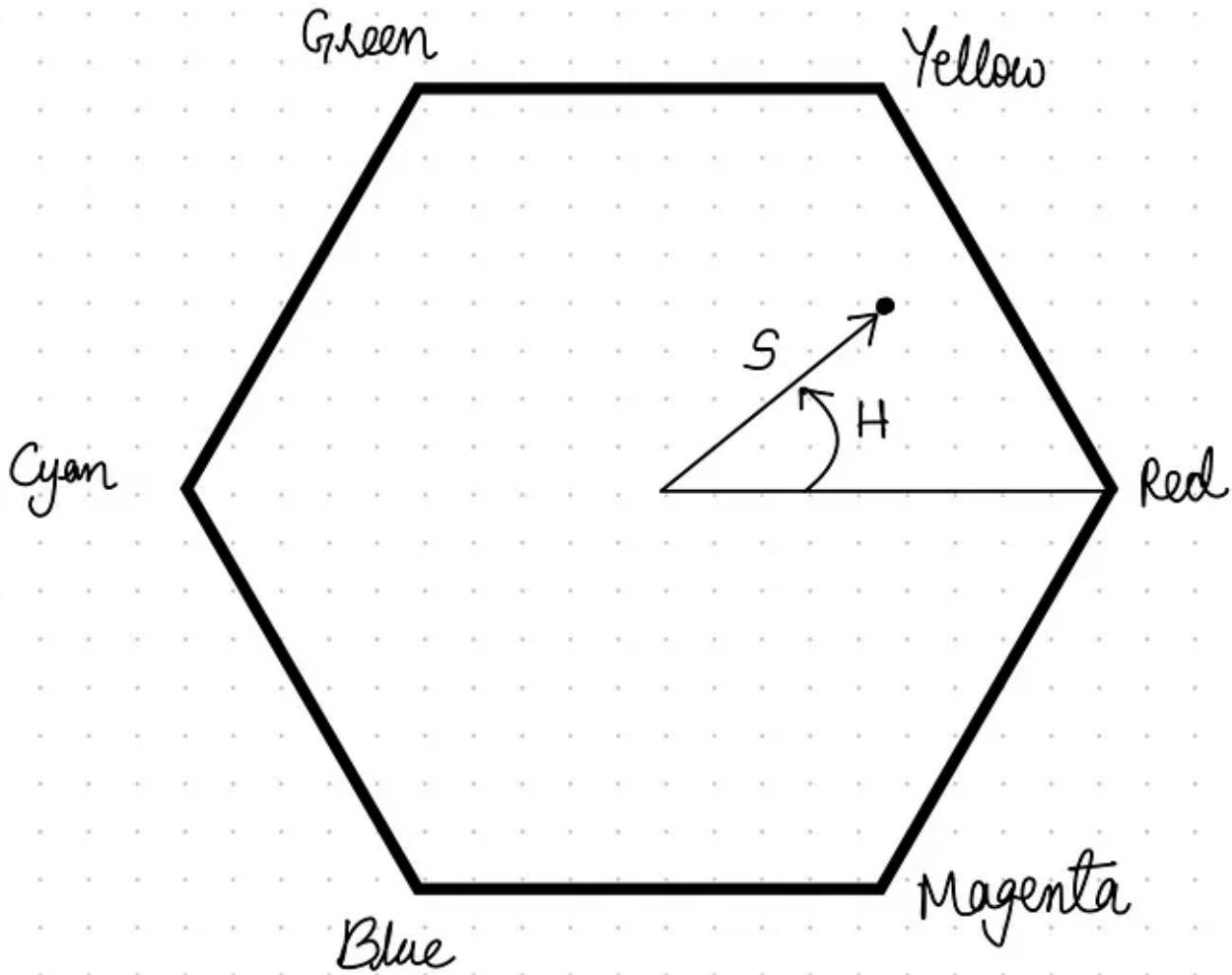


Figure 6. Top view of the color cube. (Source: Image by the author)

Mathematical Equations to convert RGB to HSI

Hue: Its value ranges from 0° to 360° or can be normalized to 0–1. It can be described as an angle on the above circle, and each degree represents a distinct color.

Saturation: It denotes the amount of color or, more precisely, its percentage or represents the vibrancy of the color. Its value ranges from 0 to 1; 0 means no color, while 1 illustrates the full color. The lower the saturation value, the grayer is present in color, causing it to appear.

Intensity: It has a range of 0–255 or normalized 0–1.

Given a color in RGB format, the HSI values can be calculated as shown in Figure 7 [8].

$$\text{Hue} = \begin{cases} \theta & ; \text{ if } B \leq G \\ 360^\circ - \theta & ; \text{ if } B > G \end{cases}$$

$$\text{where, } \theta = \cos^{-1} \left\{ \frac{0.5 [(R-G) + (R-B)]}{[(R-G)^2 + (R-B)(G-B)]^{1/2}} \right\}$$

Open in app ↗

[Sign up](#)

[Sign In](#)



Search Medium



(R+G+B)

$$\text{Intensity} = \frac{R+G+B}{3}$$

Figure 7. Mathematical equations to convert RGB pixel to HSI pixel. (Source: Image by the author)

Conclusion

This article first discussed three types of color models: RGB, CMY/CMYK, and HSI. Further, we understand each of these models through visualization. We also discussed the need for the HSI model even when the RGB model was existing. Finally, we see the mathematical illustration to convert the RGB pixel to an HSI pixel.

References:

1. <https://www.allaboutcircuits.com/technical-articles/understanding-color-models-used-in-digital-image-processing>

2. Color Subtraction

The ultimate color appearance of an object is determined by beginning with a single color or mixture of colors and...

3. CMY & CMYK Color printing. What is the difference?

The characters stand for the color cyan (C), magenta (M), yellow (Y) and key (K, black). Cyan is a kind of blue...

www.b2bsolutions.gr

4. RGB color model

The RGB color model is an additive color model in which red, green and blue light are added together in various ways to...

www.hisour.com

5. CMY color model - Wikipedia

The CMY color model is a subtractive color model in which cyan, magenta and yellow pigments or dyes are added together...

en.wikipedia.org

6. Additive & Subtractive Color Models

Color models explain how colors work, interact and how we replicate color. Additive and subtractive color models are an...

pavilion.dinfos.edu

7. HSI Color Model in Digital Image Processing | Asquero

HSI stands for Hue, Saturation, and Intensity. When humans view a color object it is described by its hue, saturation...

www.asquero.com

8. Converting RGB Image to HSI

A blog for beginners. MATLAB image processing codes with examples, explanations and flow charts. MATLAB GUI codes are...

www.imageprocessing.com

Image Processing

Computer Vision

Engineering

Science

Colors



Follow



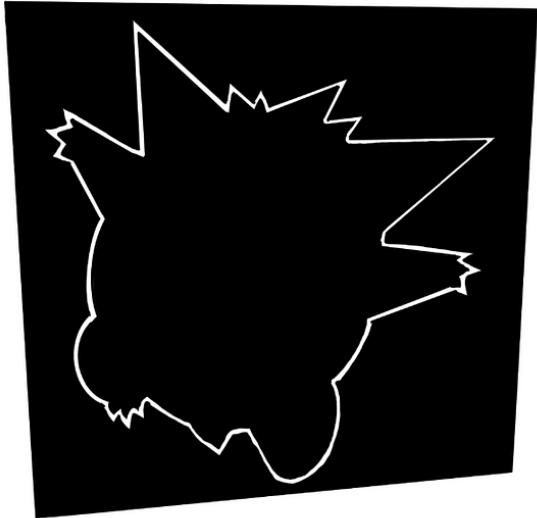
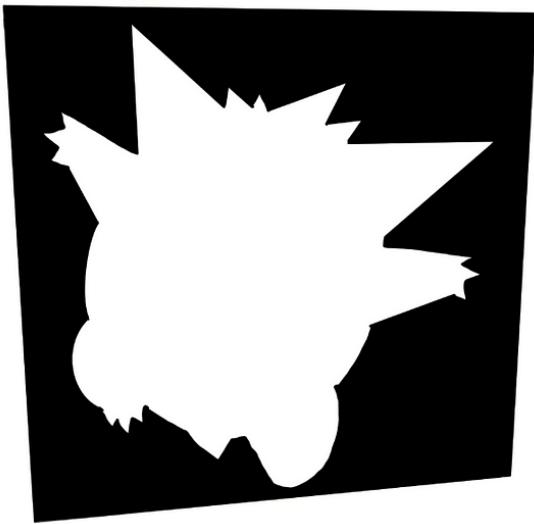
Written by Prateek Chhikara

268 Followers · Writer for Level Up Coding

Computer Science Graduate at University of Southern California | Data Scientist with 2+ years of industrial experience. Website: <https://www.prateekchhikara.com>

More from Prateek Chhikara and Level Up Coding

Image Processing



Prateek Chhikara in Towards Data Science

Understanding Morphological Image Processing and Its Operations

This article illustrates Morphological Image Processing in more straightforward terms; readers can understand how Morphology works in...

5 min read · Mar 30, 2022

299





Arslan Mirza in Level Up Coding

10 Coding Habits That Make You a Bad Programmer

The Coder's Playbook

◆ · 7 min read · May 30

👏 559

💬 5

↗ +

```
commit ffcf2c01b7ef612893529cef188cc1961ed64521 (HEAD -> master, origin/master, origin/bors/staging, origin/HEAD)
Merge: fc991bf81 5159211da
Author: iohk-bors[bot] <43231472+iohk-bors[bot]@users.noreply.github.com>
Date:   Tue Nov 8 17:44:34 2022 +0000

Merge #4563

4563: New p2p topology file format r=coot a=coot

Fixes #4559.

Co-authored-by: Marcin Szamotulski <coot@coot.me>
Co-authored-by: olgahryniuk <67585499+olgahryniuk@users.noreply.github.com>

commit fc991bf814891a9349f22cf278632d39b04d4628
Merge: 5633d1c05 5cd94d372
Author: iohk-bors[bot] <43231472+iohk-bors[bot]@users.noreply.github.com>
Date:   Tue Nov 8 13:07:58 2022 +0000

Merge #4613

4613: Update building-the-node-using-nix.md r=CarlosLopezDeLara a=CarlosLopezDeLara

Build the cardano-node executable. No default configuration.

Co-authored-by: CarlosLopezDeLara <carlos.lopezdelara@iohk.io>

commit 5159211da7a644686a973e4fb316b64ebblaa34c
Author: olgahryniuk <67585499+olgahryniuk@users.noreply.github.com>
Date:   Tue Nov 8 13:25:10 2022 +0200
```



Jacob Bennett in Level Up Coding

Use Git like a senior engineer

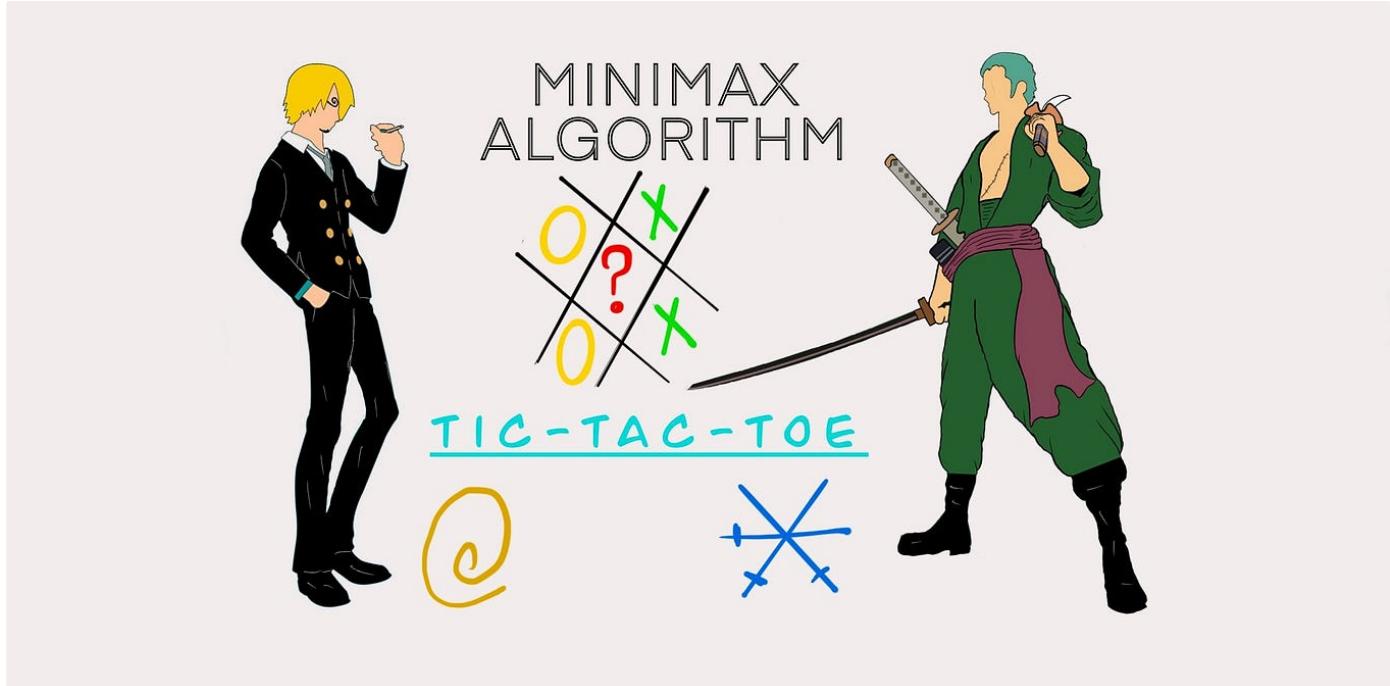
Git is a powerful tool that feels great to use when you know how to use it.

◆ · 4 min read · Nov 15, 2022

👏 6.6K

💬 68

↗ +



 Prateek Chhikara in Level Up Coding

MinMax Algorithm Explanation using Tic-Tac-Toe Game

This article explains the MiniMax algorithm in game theory using the tic-tac-toe game.

5 min read · Jan 17, 2022

 495

 2



See all from Prateek Chhikara

See all from Level Up Coding

Recommended from Medium

```
34  
35     self.debug = debug  
36     self.logger = logging.getLogger(__name__)  
37     if path:  
38         self.file = open(os.path.join(job_dir, "request.log"))  
39         self.file.seek(0)  
40         self.fingerprints.update(fp.read(), os.linesep)  
41  
42     @classmethod  
43     def from_settings(cls, settings):  
44         debug = settings.getbool("SUPERFLUX_DEBUG")  
45         return cls(job_dir(settings), debug)  
46  
47     def request_seen(self, request):  
48         fp = self.request_fingerprint(request)  
49         if fp in self.fingerprints:  
50             return True  
51         self.fingerprints.add(fp)  
52         if self.file:  
53             self.file.write(fp + os.linesep)
```



Maximilian Strauss in Better Programming

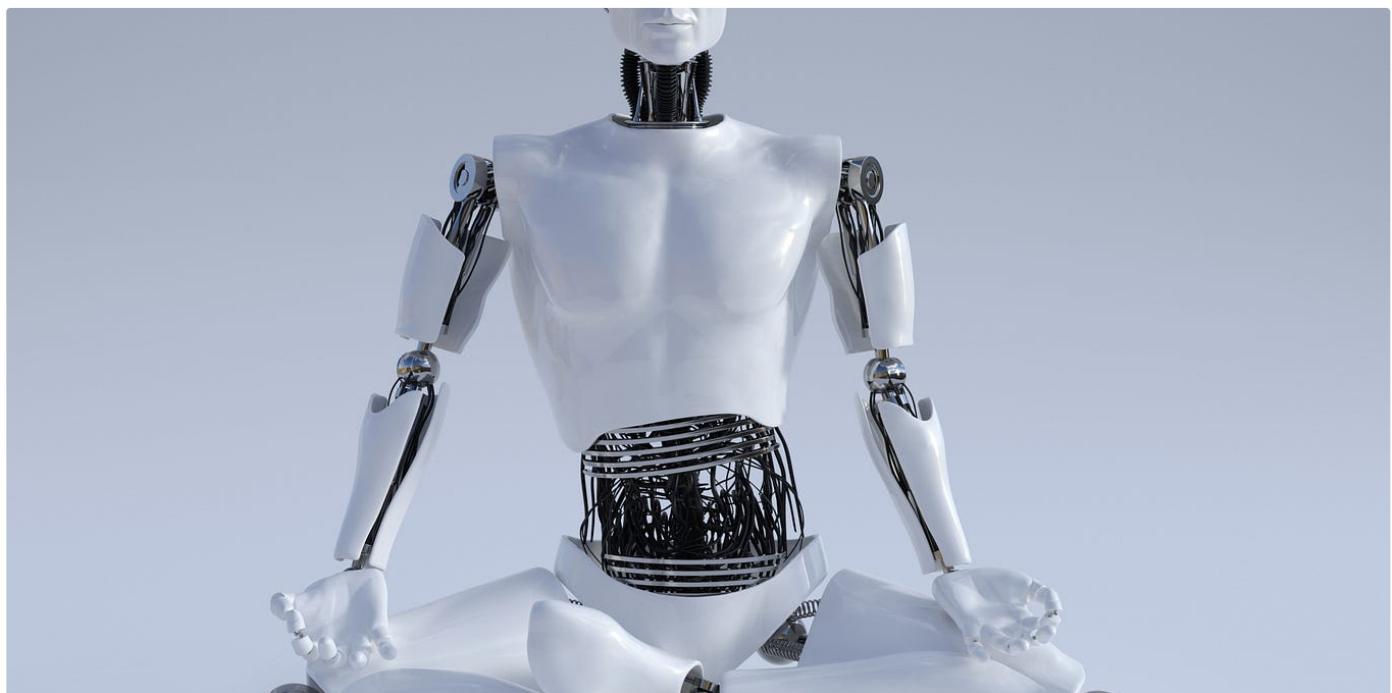
A Practical Guide To Extract Text From Images (OCR) in Python

How to use optical character recognition with three libraries

◆ · 7 min read · Jan 10

96

3





The PyCoach in Artificial Corner

You're Using ChatGPT Wrong! Here's How to Be Ahead of 99% of ChatGPT Users

Master ChatGPT by learning prompt engineering.

★ · 7 min read · Mar 17

25K 453



Lists



6 Science-Backed Health Stories on Covid, Sleep, and More

6 stories · 16 saves



ChatGPT

20 stories · 9 saves

Leadership

31 stories · 67 saves

Stories to Help You Grow as a Software Developer

19 stories · 149 saves

 Zahid Parvez

Creating panoramas using python (image stitching)

To create a panorama in python we can multiple images taken with overlapping fields of view and stich them into a wider panorama. In fact...

 · 2 min read · Jan 19

 8 

 +

 Mastafa Foufa in Towards Data Science

Machine Learning in a Non-Euclidean Space

Chapter I. Why you should learn about non-Euclidean ML

 · 10 min read · Jun 16

 356 

 +



Love Sharma in Dev Genius

System Design Blueprint: The Ultimate Guide

Developing a robust, scalable, and efficient system can be daunting. However, understanding the key concepts and components can make the...

★ · 9 min read · Apr 20

👏 4.3K

💬 32





Leonie Monigatti in Towards Data Science

Getting Started with LangChain: A Beginner's Guide to Building LLM-Powered Applications

A LangChain tutorial to build anything with large language models in Python

★ · 12 min read · Apr 25

👏 3K

💬 19



See more recommendations