به نام خدا

Table of Contents

[**سوال 1 : فضا رنگ‌ها** 3](#_Toc196424144)

[**بخش الف** 3](#_Toc196424145)

[**فضا رنگ RGB** 3](#_Toc196424146)

[**فضا رنگ CMYK** 4](#_Toc196424147)

[**فضا رنگ HSV** 5](#_Toc196424148)

[**بخش ب** 6](#_Toc196424149)

[**گزارش کار** 7](#_Toc196424150)

# **سوال 1 : فضا رنگ‌ها**

## **بخش الف**

یک فضا رنگ، به نوعی یک مدل ریاضی برای توصیف کردن رنگ‌ها با اعداد است. به طور خاص برای مدیریت کردن و مشخص کردن رنگ‌ها استفاده میشود. هر مدل برای استفاده‌ای خاص به کار میرود که در ادامه هر کدام را بررسی میکنیم:

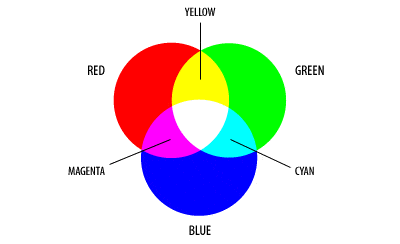
### **فضا رنگ RGB**

این مدل که مخفف سه رنگ آبی، سبز و قرمز هست، بر اساس نحوه ترکیب نور است که با رنگ سیاه (یعنی نبود نور) شروع میشود و با افزودن مقادیر مختلفی از نور قرمز، سبز و آبی (که رنگ های اصلی نور هستند)، طیف وسیعی از رنگ‌ها را تولید میکند.

نحوه کارکرد مدل به این شکل است که اگر این نورهای مختلف با هم ترکیب شوند رنگ های مختلفی تولید میکنند اگر هر 3 نور با تمام شدت ترکیب شوند رنگ سفید را پدید میاورند. متغیر بودن شدت هر نور (روشنایی هر کدام) باعث تولید میلیون های رنگ جدید میشود.

هر مولفه از این فضا رنگ با یک عدد صحیح بین 0 تا 255 نمایش داده میشوند. 0 به معنای عدم حضور این نور و 255 به معنای حضور پر رنگ این نور معنا دارد. هر کدام از این موارد را نیز میتوان با 8 بیت در هر کانال نمایش داد که اجازه میدهد 256 مقدار ممکن برای هر مولفه داشته باشیم. در نتیجه این اتفاق ما 256\*256\*256\*=16,777,216 رنگ ممکن خواهیم داشت. همچنین برای گزارش دقیق از هر رنگ میتوانیم از تعداد بیت‌های بیشتری برای توصیف کنیم.

این مدل در مانیتورها و دستگاه‌های هوشمند معمولا استفاده میشود که یک مدل اصلی برای دستگاه‌هایی است که نور منتشر میکنند. البته مشکلی که در این سیستم‌ها هست وابسته بودن آنها به یک دستگاه خاص است به طوری که مقدار یکسان RGB در صفحات مختلف، اندکی متفاوت هستند. همچنین این مدل میتواند طیف وسیعی از رنگ‌ها را بازنمایی کند مخصوصا رنگ‌های روشن و درخشان.

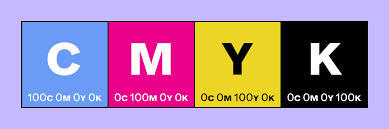


### **فضا رنگ CMYK**

این فضا رنگ که مخفف cyan, magenta, yellow, key (Black) است، اساس این مدل بر نحوه جذب نور توسط جوهر یا رنگدانه‌ها استوار است. کار آن با یک سطح سفید (مانند کاغذ) آغاز می‌شود و با استفاده از جوهر یا رنگدانه‌ها، طول موج‌های خاصی از نور کم جذب شده و طول موج‌های باقی‌مانده به چشم بیننده بازتاب داده می‌شود.

هر کدام از جوهرهای اشاره شده وظیفه جذب یک نور به خصوص را دارند، مثلا رنگدانه فیروزه‌ای (cyan) وظیفه جذب نور قرمز همراه با بازتاب آبی و سبز است. یا رنگ ارغوانی (magneta) وظیفه جذب رنگ سبز، و آبی و قرمز را بازتاب میدهد. وقتی رنگدانه‌ها روی هم قرار میگیرند، نور بیشتری را جذب میکنند. از نظر تئوری، ترکیب خالص رنگدانه‌های گفته شده باید تمام نور را جذب کرده و رنگ سیاه تولید کند ولی اینکار معمولا امکان پذیر نیست.

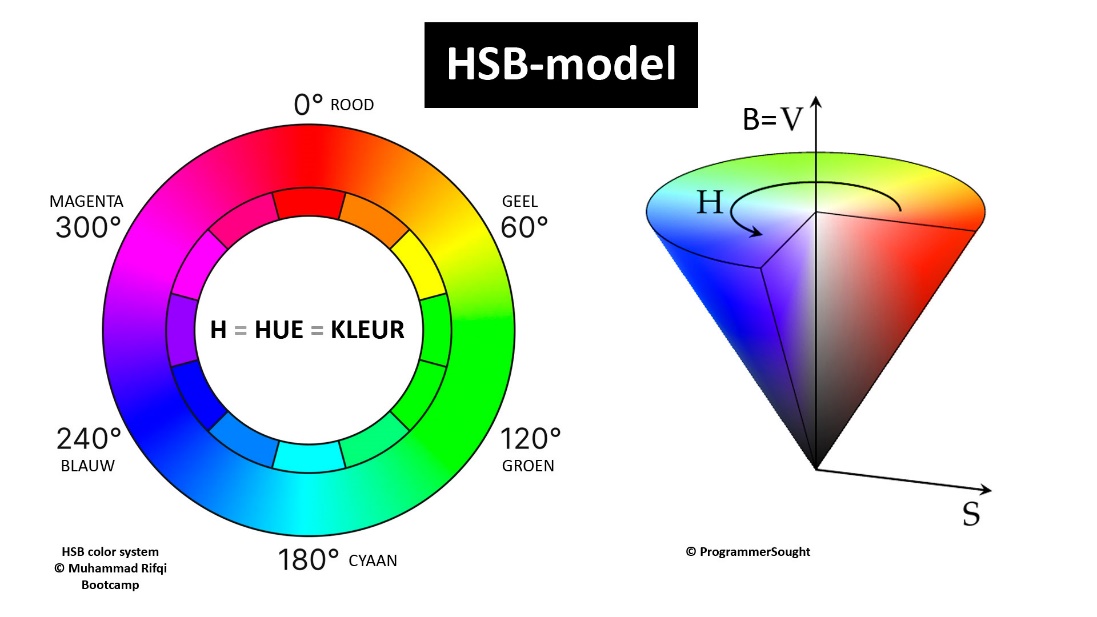
هر کدام از رنگدانه‌های گفته شده مقداری 0 تا 100 درصد میتوانند بگیرند و رنگ های جدیدی تولید کنند. از این فضا رنگ معمولا در پرینت گرفتن استفاده میشود. هر چند که در مقایسه با RGB رنگ‌های کمتری را پوشش میدهد، همچنین وقتی RGB را به این فضا رنگ تبدیل میکنیم سبب از دست رفتن درخشندگی بعضی رنگ‌ها میشود و نیاز به تنظیمات مجدد دارد.



### **فضا رنگ HSV**

این فضا رنگ که مخفف رنگ (hue)، خالص بودن رنگ یا اشباع (saturation) و ارزش (value) است که البته در بعضی موارد به HSB هم معروف است که به جای مقدار، روشنایی brightness استفاده میشود. این مدل برخلاف RGB یا CMYK است، این مدل تبدیلی از فضای رنگی RGB است که با هدف سازگاری بیشتر با شیوه درک رنگ و ویژگی‌های آن توسط انسان، طراحی شده است. این مدل رنگ‌های را در یک ساختار استوانه‌ای نمایش میدهد.

نحوه کارکرد فضا رنگ به اینگونه است که hue بیانگر خود رنگ خالص است یعنی چیزی که ما بهش رنگ میگیم به نوعی یک زاویه‌ای روی چرخه رنگ تعریف میشود که معمولا در بازه 0 تا 360 درجه قرار دارد. زاویه 0 یا 360 درجه معمولا قرمز و 120 درجه سبز و 240 درجه آبی است. زوایای میانی نشان دهنده رنگ های میانی هستند مثلا 60 درجه زرد است. مولفه saturation بیانگر شدت یا خلوص رنگ است، این مولفه مشخص میکند که چه مقدار ناخالصی یا خاکستری با خود رنگ خالص ترکیب شده‌اند که معمولا بین 0 تا 100 درصد است. 0 درصد یعنی رنگ کاملا از اشباغ خارج شده است که دیگر رنگ خالص نیست و یک طیفی به سیاه تا سفید متغیر است، به نوعی انگار صفحه‌ای خاکستری شده است. 100 درصد یعنی رنگ در خالص ترین و زنده ترین حالت ممکن آن رنگ خالص (hue) قرار دارد. مولفه ارزش نشان دهنده روشنی یا تاریکی کلی رنگ است که مقدار آن معمولا بین 0 تا 100 درصد است. ارزش 0 درصد همیشه سیاه است صرف نظر از مقدار رنگ یا اشباع. ارزش 100 درصد نشان دهنده روشن‌ترین حالت ممکن رنگی است که توسط رنگ خالص و اشباع تعریف شده است. کاهش دادن مقدار ارزش، رنگ را تیره‌تر میکند. این فضا رنگ بیشتر مورد استفاده موارد گرافیکی و بسری در کامپیوترها است و برای برنامه نویسی و مباحث پردازش تصویر و بینایی کامپیوتر به کار میرود زیرا میتواند اطلاعات رنگ (hue) را از خالص بودن و میزان روشنایی رنگ جدا میکند و برای تشخیص شی مورد مناسب است. این فضا رنگ برای انسان بسیار شهودی و قابل درک تر است تا بتواند رنگ‌ها را بر اساس درک خودش دستکاری کند، اگر چه از این فضا رنگ نمیتوان برای خروجی فیزیکی مستقیم مانند RGB که در مانیتور ها استفاده میشود یا CMYK که در پرینت گرفتن استفاده میشود. معمولا این فضا رنگ نیاز دارد مقادیر خودش را به RGB تبدیل کند اگر بخواهد که به نمایش در مانیتور در بیاید.



## **بخش ب**

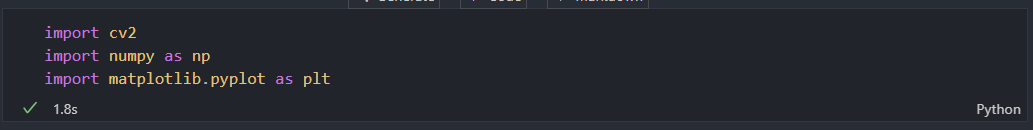
ی

### **گزارش کار**

در این پروژه ما ابتدا تصویر ورودی را در که در فضا رنگ RGB قرار دارد به فضا رنگ HSV تبدیل میکنیم. زیرا این فضا رنگ برای تشخیص پوست انسان معمولا مناسب تر است زیرا به کمک 3 مولفه این فضا رنگ میتوانیم پوست انسان را تشخیص دهیم.

Hue بیانگر تن رنگ خواهد بود، تن‌های رنگ پوست انسان، در میان قومیت‌های مختلف، معمولا در محدوده‌ی از رنگ‌ها قرار دارد. saturation بیانگر خلوص یک رنگ است همانطور که قبلا توضیح دادیم. پوست انسان دارای طیف محدودی از اشباع را دارا است که نه به طور کامل خاکستری است (مقدار پایین اشباع) و نه به شدت پر رنگ و شدید (مقدار بالای اشباع). Value نیز میزان روشنایی را مشخص میکند که تحت تاثیر شرایط نوری قرار دارد.

با استفاده از HSV، میشود یک سری آستانه بر اساس hue و saturation در نظر گرفت که کمتر به تغییرات نور پردازی نسبت به RGB حساس است. که همانطور که قبلا توضیح دادیم برای پوست انسان یک محدوده خاصی در نظر گرفتیم و آستانه ها را تعیین کردیم و ماسک را اعمال کردیم.

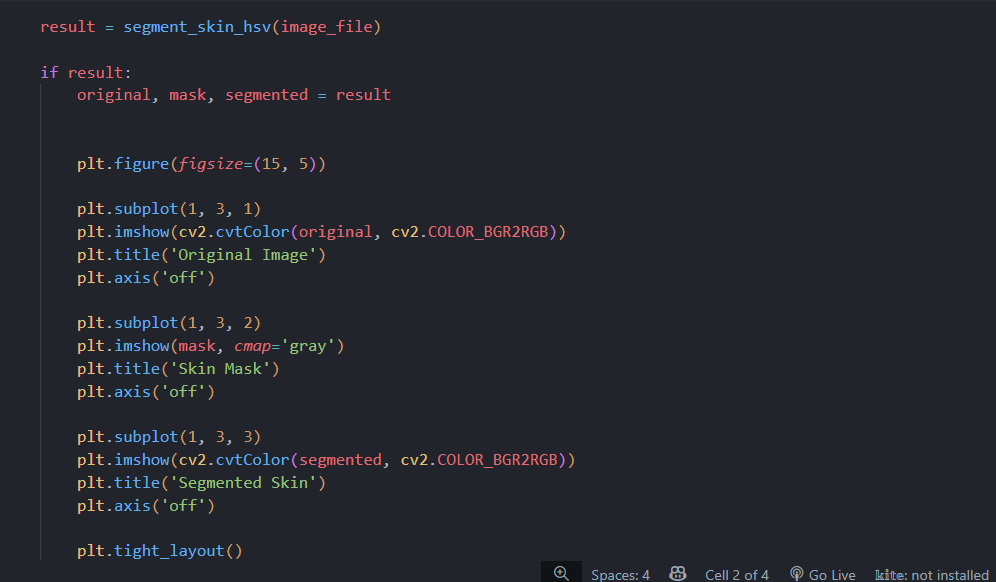


ابتدا کتابخانه cv2 برای کار با تصویر پیاده‌سازی کردیم. از کتابخانه matplotlib برای گزارش و خروجی تصویری استفاده کردیم.



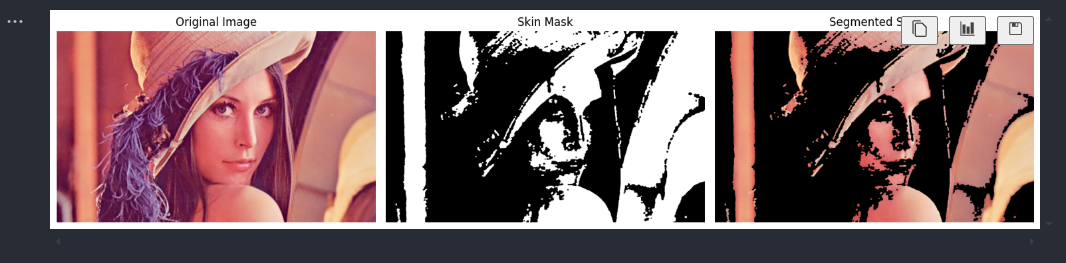
بر اساس توضیحات گفته شده، در تابع segment\_skin\_hsv توضیحات را اعمال کردیم. ابتدا تصویر ورودی را گرفتیم و از RGB به HSV تبدیل کردیم. سپس با کمک lower\_skin و upper\_skin یک محدوده‌ای برای پوست انسان در نظر گرفتیم (hue: 0-179, s:0-255,v:0-255) البته بر اساس مشاهده سایر خروجی‌های گرفته شده میتوانیم مقداری این بازه‌های را جابجا کنیم. و بعد با استفاده از skin\_mask یک ماسک باینری تعریف کردیم. در ادامه با کمک kernel سعی کردیم مقداری عکس را بهتر کنیم و نویز های آن را حذف کنیم در خط اول skin\_mask نویزهای کوچک را حذف کردیم و در خط بعدی سعی کردیم حفره‌های کوچک را پر کنیم.

در مرحله بعدی با کمک segmented\_skin\_image ماسک به عکس اولیه اعمال کردیم. اینکار به صورت bitwise صورت گرفته که نیازمند این است که تصاویر از تعداد کانال یکسان پیروی کنند پس ماسک را به تصویر اولیه RGB اعمال کردیم. در نهایت خروجی را برگرداندیم.



با کمک کتابخانه matplotlib و عکس ورودی که عکس لنا است به تابع به عنوان ورودی دادیم و خروجی را گزارش دادیم.

#### **خروجی**



بخش پ