

به نام خدا

امیرمحمد کمیجانی ۹۹۵۲۲۰۳۲

\* سوالات تشریحی یعنی سوالات ۳ و ۵ و ۷ و ۸ در فایل دیگری در همین فایل زیپ ارسال شده اند.

\* توضیحات کد ها یعنی سوالات ۱ و ۲ و ۴ + جدول سوال ۶ در این فایل است.

(۱)

ابتدا تبدیل RGB به CMYK را شرح میدهیم

ابتدا باید در نظر بگیریم که مقیاس در RGB, CMYK متفاوت است.

ابتدا عملیات نورمالیزیشن را برای هر یک از کانال های r, g, b انجام میدهیم یعنی تقسیم بر ۲۵۵ میکنیم.

سپس طبق فرمول زیر که در جزوه نیز موجود است عمل میکنیم:

$$K = 1 - \max(R, G, B)$$

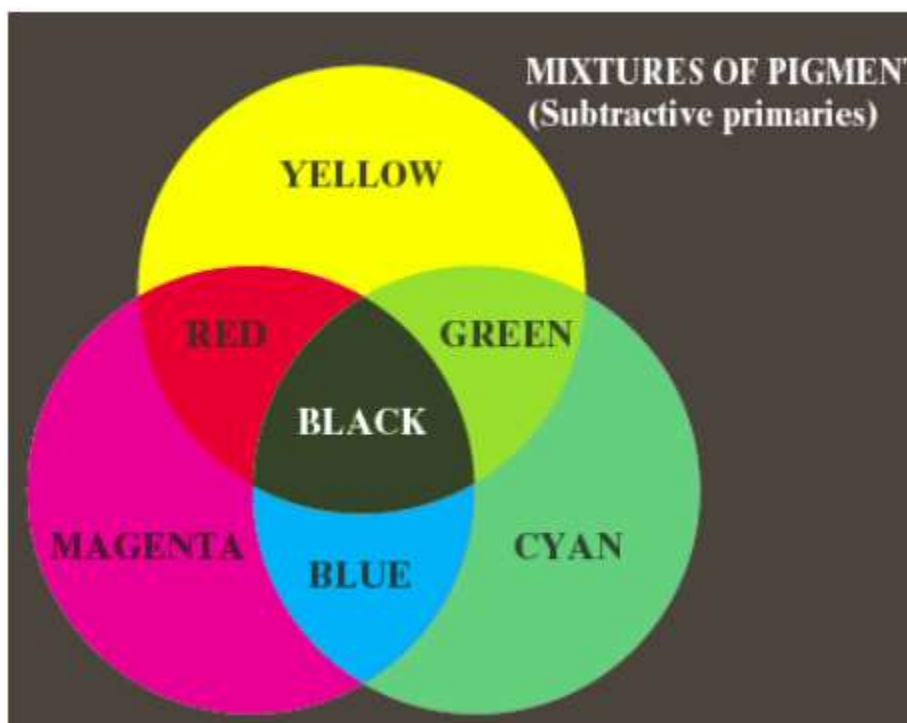
$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 - K \\ 1 - K \\ 1 - K \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

در نهایت بعد از اعمال این فرمول ها در کد خود باید مقادیر نورمالیزه شده را وارد مقیاس مناسب CMYK کنیم.

### تبدیل CMYK به RGB :

برای عکس این حرکت ابتدا به این صورت عمل میکنیم که چه مقدار نور توسط رنگ سیاه یا همان K جذب شده است و مقداری که جذب شده است را کم میکنیم تا به آن میزان از مقادیر  $c, m, y$  مقدار جذب شده را می‌کاهیم و در هم ضرب میکنیم تا مقدار  $r, g, b$  بدست آید.

این قسمت از اسلاید در مفهوم این قسمت به ما کمک میکند:



لازم به ذکر است که مقیاس بندی یا همان **scaling** را در این تابع نیز انجام میدهیم.

### تبدیل RGB به HSI :

برای پیاده سازی این تابع با استفاده از فرمول ها و اسلاید درس عمل میکنیم.

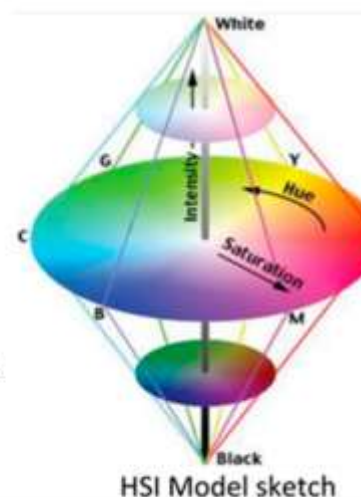
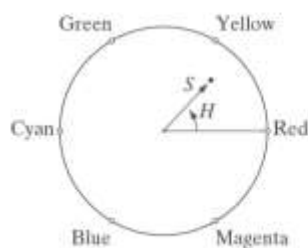
## تبدیل RGB به HSI

$$\theta = \cos^{-1} \left( \frac{(R - G) + (R - B)}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \right)$$

$$H = \begin{cases} \theta, & \text{if } B \leq G \\ 360 - \theta & \text{if } B > G \end{cases}$$

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

$$I = \frac{R + G + B}{3}$$



HSI Model sketch

و در نهایت در cell بعدی از هر تابع استفاده کرده ایم و خروجی را نمایش داده ایم.

(۲)

ابتدا دو تصویر را دریافت کرده و به RGB بردیم.

برای پیاده سازی تابع و پیدا کردن اختلاف دو تصویر باید به این صورت عمل کنیم که ابتدا تصویر را به فضای gray scale برده و سپس کار خود را شروع کنیم.

ابتدا با توجه به اینکه shape دو عکس یکسان نیست عکس با shape کمتر را در نظر میگیریم.

با توجه به اینکه shape دو تصویر را به صورت دستی پیدا کردیم در کد هم به همان صورت میگذاریم. البته در خطی که کامنت شده کد ما به صورتی است که میتواند به صورت داینامیک نیز عکس با اندازه کمتر را انتخاب کند. دلیل انتخاب عکس با اندازه کمتر نیز مشخص است در تصویر بزرگتر نقاطی را داریم که در تصویر کوچکتر نداریم در نتیجه برای اندازه باید اشتراک یا مینیموم بگیریم.

حال هر دو تصویر وارد grayscale شدند حال ایده اصلی به این صورت است که یک تصویر را مثلا کانال r و تصویر دیگر را کانال b,g در نظر بگیریم.

در کد ما تصویر اول را کانال r,g در نظر گرفتیم و تصویر دو را b در نظر گرفتیم.

نتیجه این است که در تصویر خروجی جاهایی که در تصویر ۲ متمایز از تصویر ۱ هستند به رنگ آبی و جاهایی که در تصویر ۱ متمایز از تصویر ۲ هستند به رنگ زرد یعنی ترکیب قرمز و سبز نمایش داده میشوند.

(۴)

ابتدا هر دو تصویر را خواندیم و نمایش دادیم.

سپس به صورت دستی و حدس و آزمایش توانستیم روبان مشکی رنگ را در گوشه سمت چپ تصویر ایجاد کنیم. برای ایجاد آن از کتابخانه cv2.line استفاده کردیم و نقاط ابتدا و انتها و سپس رنگ مشکی و ضخامت ۱۰۰ در نظر گرفتیم.

علت اینکه برای تولید رنگ مشکی را در سه کانال ۱ قرار دادیم این بود که اگر آنرا 0 میگذاشتیم بعد از projective مقدار سیاه آن از بین میرفت (آزمون و خطا)

در مرحله بعدی ۴ نقطه مربوط به destination یا به اصطلاح فضایی بر روی دیوار که عکس باید روی آن قرار میگرفت را انتخاب کردیم. (۴ نقطه گوشه)

سپس چهار نقطه گوشه عکس پدربزرگ را انتخاب کردیم.  
نقاط گوشه مشخص تر هستند و برای قرار دادن عکس منطقی تر.

```
mat = cv2.getPerspectiveTransform(src_points, dst_points[, solveMethod])  
// src_points: Coordinates of quadrangle vertices in the source image  
// dst_points: Coordinates of the corresponding quadrangle vertices in the destination image  
// mat: Perspective transform from four pairs of the corresponding points
```

ساختار این تابع مشخص است نقاط سورس و مقصد را میدهم (۴ نقطه) و سپس ماتریس تبدیل آنرا به ما میدهد.

در نهایت ما میخواهیم که این عکس بر روی قسمتی از عکس دیوار که مشخص کرده ایم project شود.

`cv::warpPerspective()`

Perspective transform a whole image

طبق اسلاید ها و کاربرد توابع از این تابع استفاده میکنیم.

یک کپی از عکس دیوار میگیریم تا تصویر اصلی دچار تغییرات نشود. سپس تمام مناطق و بازه ای که projected image در آنها مقدار 0 ندارد انتخاب میکنیم و در تصویر نهایی نیز همان بازه ها را مشخص میکنیم و تصویر پدربزرگ را روی آن قسمت قرار میدهم و در نهایت تصویر نهایی را خروجی میدهم.

(۶)

نوع تبدیل	انتقال	rigid	شباهت	affine	تصویری
فاصله جفت نقاط ثابت میماند	صحیح	صحیح	غلط	غلط	غلط
زاویه بین جفت خط ثابت میماند	صحیح	صحیح	صحیح	غلط	غلط
خط ها، خط باقی مانند	صحیح	صحیح	صحیح	صحیح	صحیح
زاویه بین هر خط و محور ایکس ثابت میماند	صحیح	غلط	غلط	غلط	غلط
چهار ضلعی ها، چهار ضلعی باقی می مانند	صحیح	صحیح	صحیح	صحیح	صحیح
خطوط موازی، موازی باقی می مانند	صحیح	صحیح	صحیح	صحیح	غلط

دایره ها، دایره باقی می مانند	صحیح	صحیح	صحیح	غلط
نسبت بین مساحت دو شکل ثابت باقی می مانند	صحیح	صحیح	صحیح	غلط