

به نام خدا

گزارش تمرین سری سوم - درس مبانی بینایی کامپیوتر

سید محمد علی خواری - شماره دانشجویی: 99521496

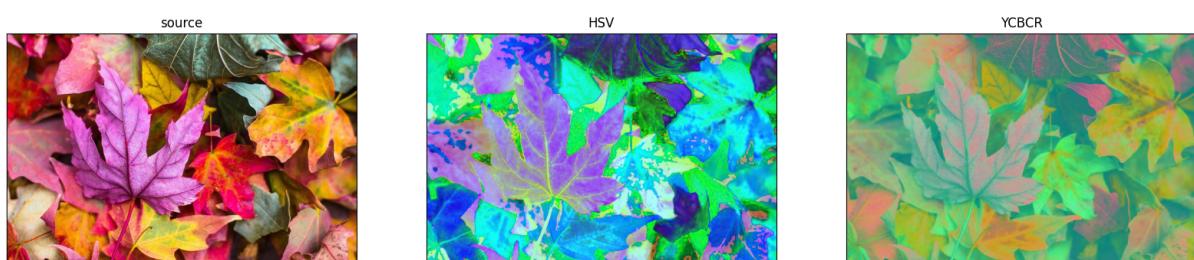
سوال اول)

الف) برای حل این سوال دوتابع به نام های `cmyk_to_rgb` و `rgb_to_cmyk` را تعریف میکنیم. درتابع اول یک ورودی از نوع رنگ RGB گرفته و با استفاده از فرمول های موجود در اسلاید به دست آورده و با ضرب کردن مقادیر به دست آمده در 100 نتیجه را بر میگردانیم. برای اینکه بتوان از فرمول نوشته شده استفاده کرد ابتدا مقادیر آن را نرمالایز میکنیم.

برای تبدیل رنگ از فضای cmyk به rgb هم مراحل مشابه با مراحل بالا طی میکنیم و نتایج را در 255 ضرب میکنیم تا به فرمت rgb قابل نمایش باشند.

```
The original RGB arr is: [ 50. 70. 130.]  
The CMYK of given RGB is: [31.37254902 23.52941176 0. 49.01960784]  
The RGB of given CMYK is: [ 50. 70. 130.]
```

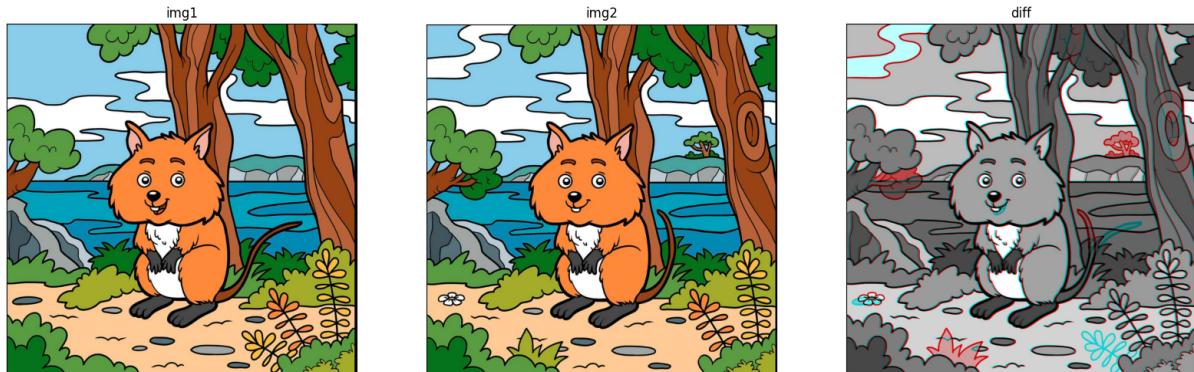
ب) برای این کار از تابع `cvtColor` و دادن عکس ورودی و تبدیل مورد نظر این کار را انجام میدهیم.



ج) برای نمایش یک عکس در قالب کانال های مجراء، ابتدا تصویر را به HSV تبدیل کرده و به صورت جداگانه هرکدام از کانال های آن را رسم میکنیم.



د) برای به دست آوردن تفاوت های موجود در دو تصویر ابتدا هردو را به فرمت gray در آورده و سپس یک متغیر برای تصویر نهایی که یک تصویر سه کanal هست ساخته و به کanal قرمز آن



تصویر اول و به دو کanal سبز و آبی، تصویر دوم را میدهیم. با این کار نقاطی از تصویر اول که مقدار روشنایی بیشتری نسبت به تصویر دوم دارند به صورت رنگ قرمز در می آید.

ه) هرکدام از فضاهای رنگی برای موارد خاصی استفاده میشود. به عنوان مثال برای نمایش تصویر در مانیتور از فضای رنگی RGB استفاده می کنیم. در حالیکه برای چاپ تصاویر در چاپگر ها از فضای رنگی cmyk استفاده میکنیم.

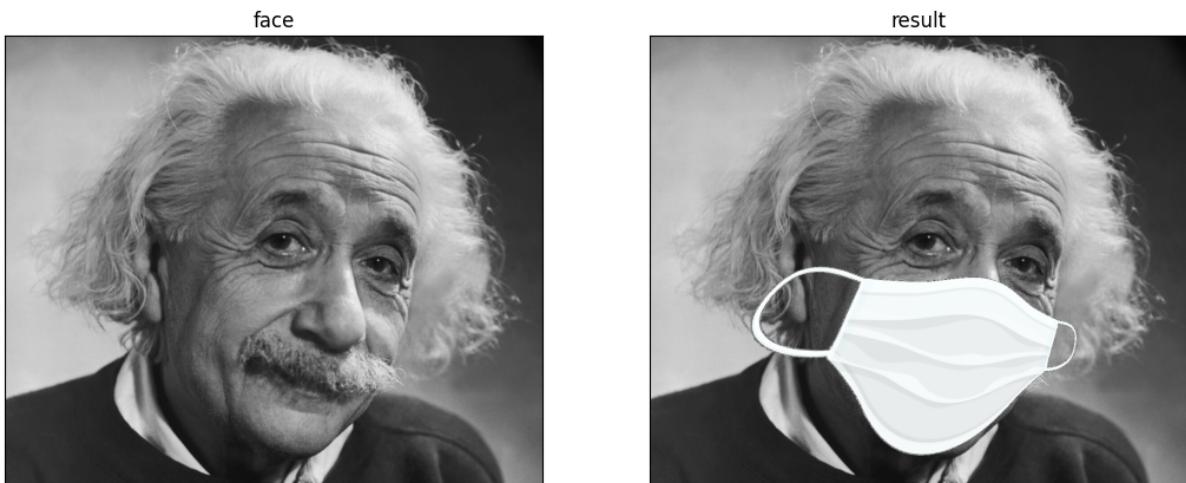
سوال دوم)

ابتدا با استفاده از یک حلقه for تصاویر موجود در پوشه داده شده را خوانده و نمایش میدهیم. در مرحله بعد از کلاس stitcher یک شی ساخته و با دادن عکس های خوانده شده در بالا آن ها را به هم متصل میکنیم.



سوال سوم)

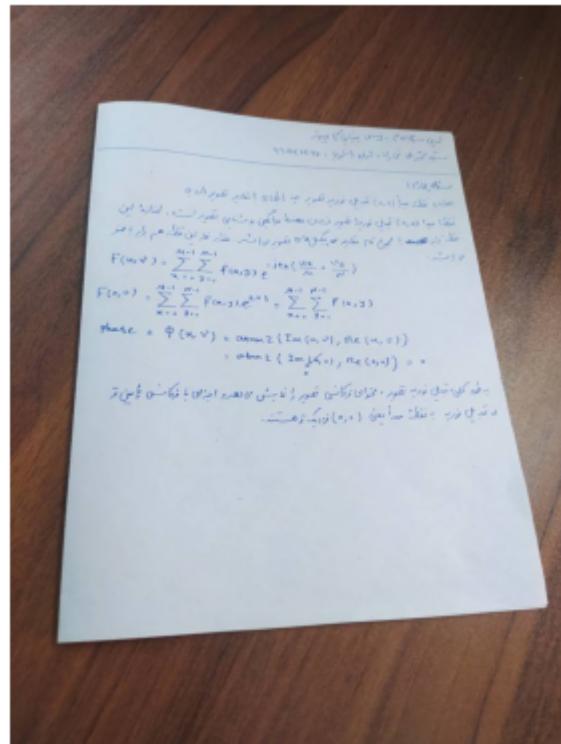
الف) در ابتدا فایل train شده برای تشخیص چهره را دانلود کرده و در مسیر colab قرار داده و با استفاده از توابع موجود در کتابخانه dlib ناحیه صورت تصویر داده شده را پیدا میکنیم. در مرحله بعد برای آنکه بتوانیم ماسک را بر روی صورت قرار دهیم نیاز است تا نقاط مورد نظر بر روی صورت را مشخص کرده و منتظر با آن نقاط در ماسک را تعیین کنیم. سپس با استفاده از تابع perspective در کتابخانه opencv، تبدیل مناسب را پیدا میکنیم. پس از تعیین تبدیل مناسب و اعمال آن بر روی تصویر ماسک باید ماسک را بر روی صورت قرار دهیم. برای این کار تصویر ماسک را به حالت gray تبدیل کرده و آن را با تابع threshold تصویر را به صورت باینری تبدیل میکنیم. و در آخر با استفاده از تابع add از کتابخانه opencv ماسک را بر روی صورت قرار میدهیم.



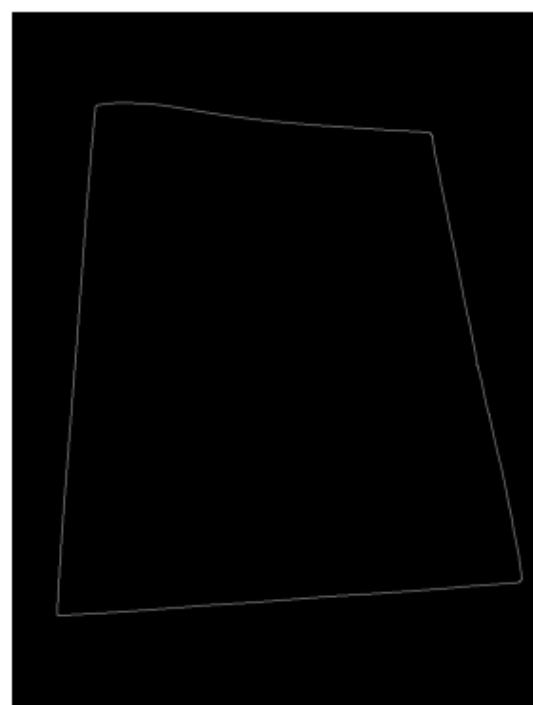
ب) به فرآیند تشخیص و قرار دادن نقاط بر روی صورت انسان مثل چشم، بینی، چانه و غیره گفته میشود. به این نقاط کلیدی landmark یا facial landmark یا هم گفته میشود. معمولا برای تشخیص نقاط از تکنیک های بینایی کامپیوتر مثل ماشین لرنینگ و یا پردازش تصویر استفاده کرد.

سوال چهارم)

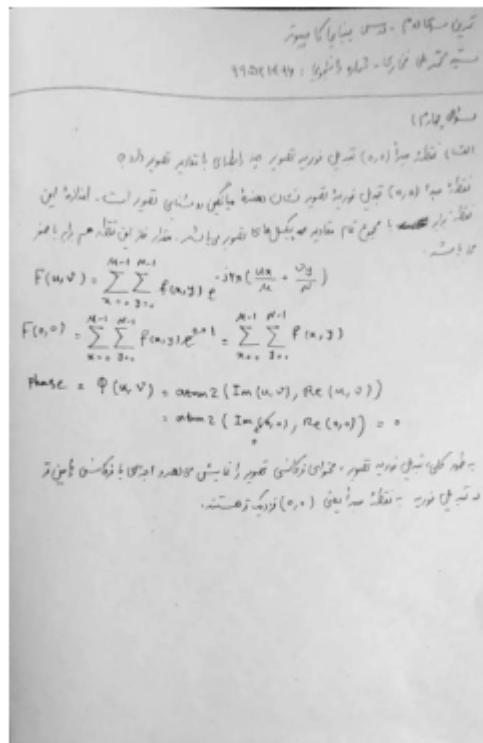
الف) ابتدا تصویر مورد نظر را در پوشه ای در مسیر تمرین قرار داده و با استفاده از کتابخانه opencv خوانده و با استفاده از تابع imshow نمایش میدهیم.



پس از آن تصویر را به فرمت grayscale تبدیل کرده و با استفاده از تابع gaussian کتابخانه opencv نویز های موجود در تصویر را از بین میبریم و لبه یاب کنی را با مقادیر T1, T2 مناسب اجرا میکنیم. انتخاب مقادیر T1, T2 بسته به کیفیت و شدت روشنایی تصویر دارد.



در مرحله بعد با استفاده ازتابع findContours ناحیه مستطيلي کاغذ را تشخیص داده و مختصات چهار گوشه آن را به دست می آوریم و با استفاده ازتابع perspective تبدیل مناسب برای اینکه کاغذ را به حالت قابل خوانایی برسانیم به دست می آوریم. نتیجه نهایی به صورت زیر است:



سوال پنجم)

الف) برای تشخیص گوشه های موجود در یک تصویر با استفاده از الگوریتم harris مراحل زیر را طی میکنیم:

- ابتدا تصویر خوانده شده را به فرمت gray درآورده و با استفاده از یک فیلتر مناسب مثل فیلتر گاوی، نویز تصویر را کاهش میدهیم.
- در مرحله بعد با توجه به پارامتر های مورد نیاز در فرمول الگوریتم، نیاز است که مشتق های افقی و عمودی تصویر را به دست بیاوریم، برای این کار میتوانیم از عملگر sobel استفاده کنیم. با این کار از تغییرات تصویر در هر دو راستای افقی و عمودی مطلع می شویم.

- پس از محاسبه های مشتق های افقی و عمودی، مربعات آن ها را محاسبه کرده و پنجره w را بر روی مربعات مشتق افقی و عمودی اعمال میکنیم.
- در مرحله بعد، مقادیر R برای تمام پیکسل های تصویر به دست می آوریم. برای این کار از فرمول نوشته شده در اسلاید استفاده کرده و با استفاده از دترمینان و تریس ماتریس M که شامل مربعات مشتق افقی و عمودی تصویر است به دست می آوریم.

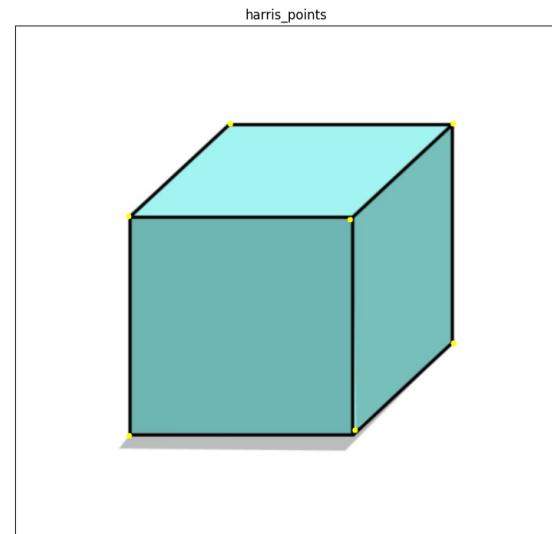
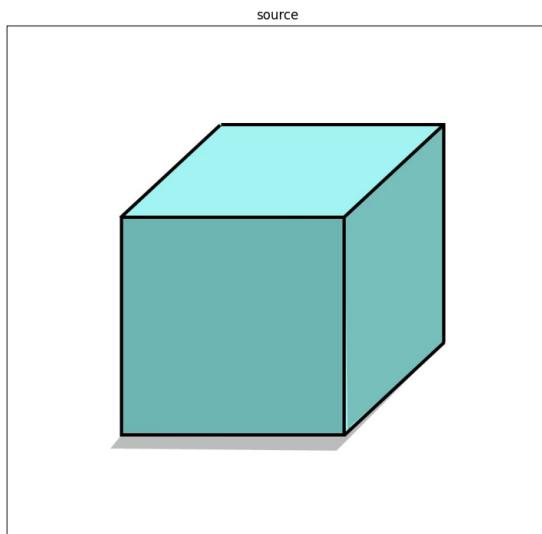
$$R = \det(M) - k(\text{trace}(M))^2$$

- پس از مرحله نیاز است که مقادیر کوچکتر از thersold را برابر با صفر قرار داده و با استفاده از non-maximum suppression مقادیر بیشینه محلی را تقویت کنیم و مقادیر غیر بیشینه محلی را از بین ببریم.

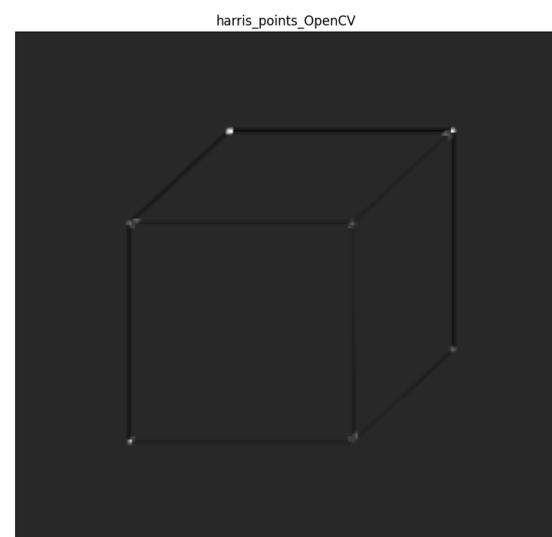
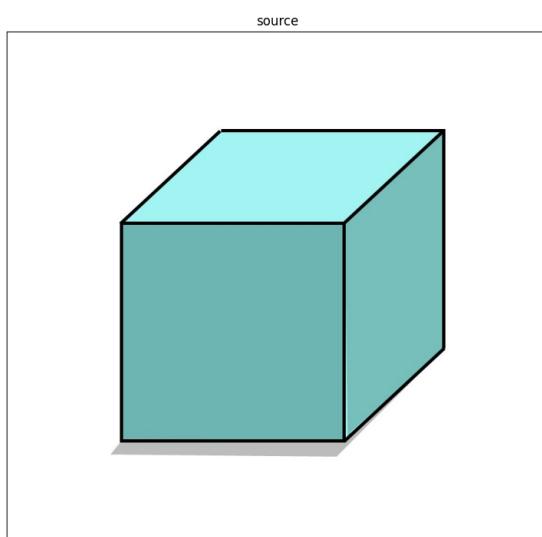
(ب)

تمامی مراحل نوشته شده در قسمت قبل در نوت بوک Q5 نوشته شده و در زیر نتایج آن را خواهیم دید:





و نتیجه استفاده از کتابخانه opencv به صورت زیر میباشد:



(سوال ششم)

SIFT: در این روش که مخفف عبارت scale-invariant feature transform میباشد، یکی از روش های مناسب و پر استفاده جهت تشخیص و استخراج ویژگی از تصویر میباشد. در این روش با استفاده از فیلتر های گاوی نقاط کلیدی استخراج میشود که به همین دلیل و استفاده از پردازش های طولانی و سنگین استفاده از این الگوریتم امکان پذیر نیست.

SURF: این روش به منظور بهبود الگوریتم قبلی ابداع شد و سرعت اجرا الگوریتم بهتر خواهد بود. این روش مخفف عبارت speed-up robust feature میباشد. علاوه بر سرعت بالای این الگوریتم نسبت به الگوریتم SIFT ، مقاومت بیشتری در برابر نویز دارد اما به دلیل استفاده از تبدیلات ویژه در این الگوریتم بسیاری از ویژگی های الگوریتم SIFT را ندارد.

ORB: مخفف oriented fast and robust brief ها برای استخراج نقاط کلیدی و ویژگی های موجود در تصویر است. این روش برای تصاویر با اندازه کوچک بدون توجه به جهت بسیار قوی است. همچنین این روش نسبت به دو روش قبلی, SURF سریع تر بوده و مقاومت بیشتری در برابر نویز دارد اما دقت آن کمتر است.