

تمرین سری سوم – بخش عملی

سوال پنجم – گزارش کار

مهرشاد فلاح اسطلخ‌زیر – 401521462

منطق کد:

بخش اول (a):

```
image_path = "Flower.png"          #TO DO
image = cv2.imread(image_path)

# Convert to RGB for displaying with matplotlib
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)

# Split the image into R, G, B channels
R, G, B = image_rgb[:, :, 0], image_rgb[:, :, 1], image_rgb[:, :, 2]          #TO DO

# Convert to HSV and split into H, S, V channels
image_hsv = cv2.cvtColor(image, cv2.COLOR_BGR2HSV)          #TO DO
H, S, V = image_hsv[:, :, 0], image_hsv[:, :, 1], image_hsv[:, :, 2]          #TO DO
```

در این بخش صرفاً از `cvtColor` استفاده کردیم و عکس را به حالت RGB و HSV تبدیل کردیم و بعد هم کانال‌ها را جدا کردیم.



نتیجه این کار هم در عکس بالا قابل مشاهده است.

بخش دوم (b):

```
# Create an RGB image where:
# - Red channel is img1
# - Green channel is img2
# - Blue channel is zeros
blue_channel = np.zeros(img1.shape) #TO DO

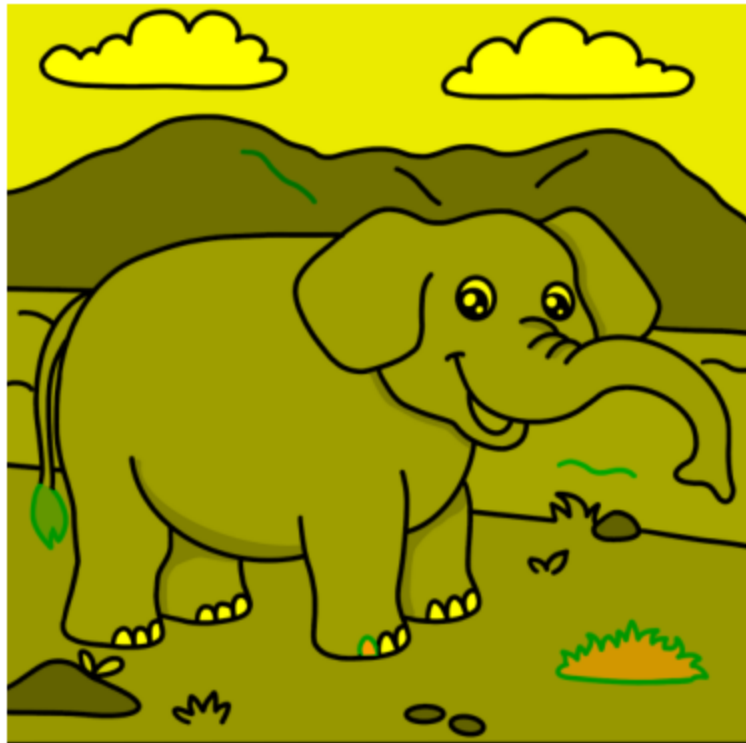
# Stack the channels
color_diff = np.stack((blue_channel, img2, img1), axis = -1) # (B, G, R) #TO DO
color_diff = color_diff.astype('uint8')

# Convert to RGB for displaying with matplotlib (OpenCV uses BGR)
color_diff_rgb = cv2.cvtColor(color_diff, cv2.COLOR_BGR2RGB) #TO DO

# Show the result
plt.imshow(color_diff_rgb)
plt.title('Difference Highlighted in Color')
plt.axis('off')
plt.show()
```

برای این بخش ابتدا کانال آبی را با صفر، کانال سبز را با عکس دوم و کانال قرمز را با عکس اول پر می‌کنم. برای این کار هم از `np.stack` استفاده می‌کنم. در نهایت چون دیفالت `opencv` کانال رنگی BGR است یک تبدیل به RGB می‌زنم و عکس خروجی را نمایش می‌دهم.

Difference Highlighted in Color



همانطور که در خروجی مشخص است. جاهایی که متفاوت هستند در دو عکس با رنگ سبز مشخص است.

```
# Create an empty RGB image
color_diff = np.zeros((img1.shape[0], img1.shape[1], 3)) #TO DO

# Where images are equal → keep grayscale
mask_equal = img1 == img2
grayscale = img1[mask_equal]
color_diff[mask_equal] = np.stack([grayscale] * 3, axis = -1) #TO DO

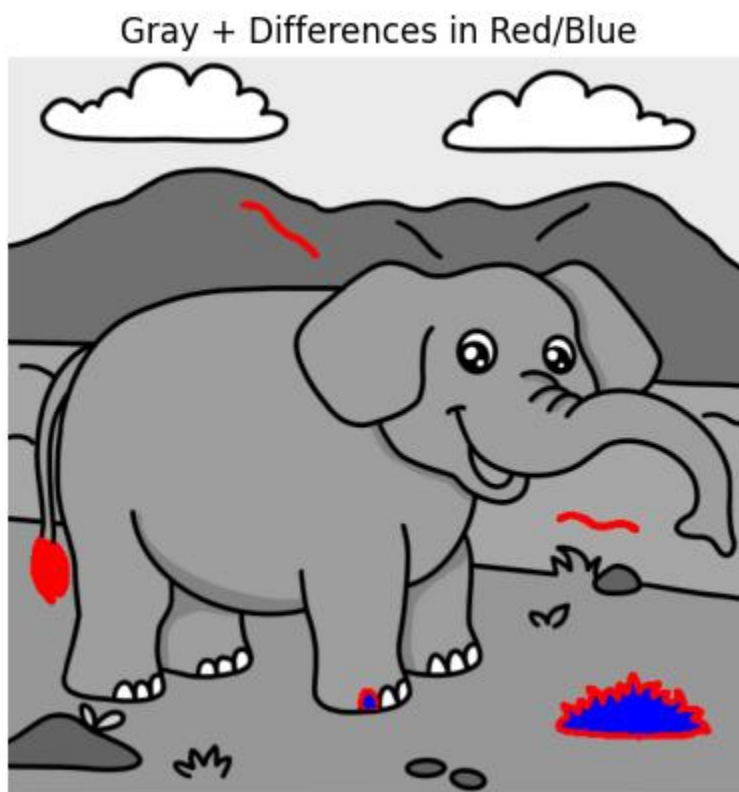
# Where img1 > img2 → red
mask_red = img1 > img2
color_diff[mask_red] = [255, 0, 0] #TO DO

# Where img2 > img1 → blue
mask_blue = img2 > img1
color_diff[mask_blue] = [0, 0, 255] #TO DO

color_diff = color_diff.astype('uint8')

# Show result
plt.imshow(cv2.cvtColor(color_diff, cv2.COLOR_BGR2RGB))
plt.title("Gray + Differences in Red/Blue")
plt.axis('off')
plt.show()
```

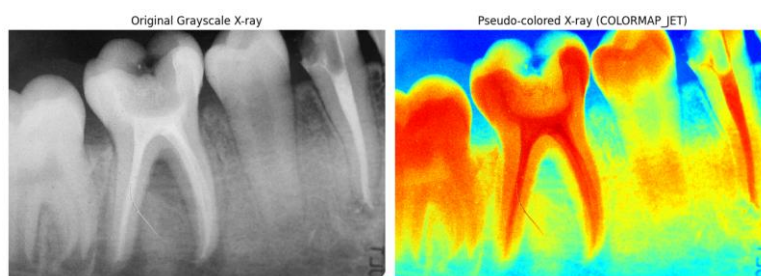
همانطور که در کد مشخص است برای روش دوم ابتدا یک آرایه سه کاناله با صفر می‌سازم. در مرحله بعد مکان‌هایی که دو عکس برابر هستند را با خود عکس پر می‌کنیم. جاهایی که عکس یک بزرگتر است کانال قرمز و جاهایی که عکس دو بزرگتر است را با کانال آبی پر می‌کنم.



نتیجه متد دوم هم در عکس مشخص است.

بخش سوم (C):

برای این بخش صرفاً از تابع `cv2.applyColorMap` استفاده می‌کنم و `COLORMAP_JET` طبق خواسته طراح. خروجی هم در عکس پایین مشخص است. [لینک](#) برای توضیحات بیشتر.

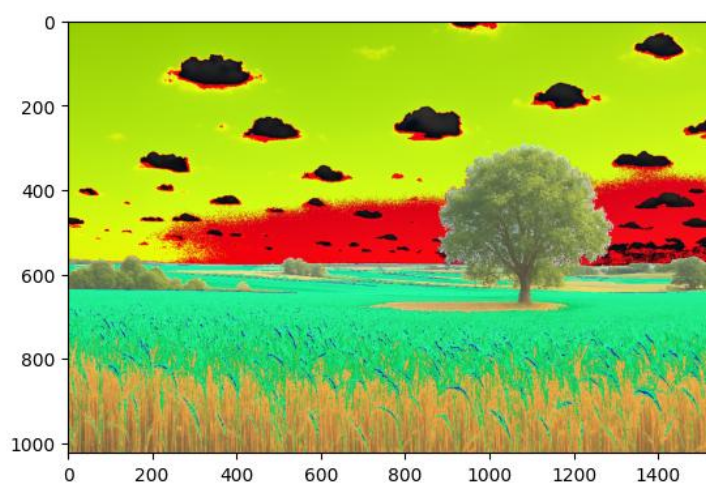


بخش چهارم(d):

****** برای این بخش از چت برای جنریت کردن یک عکس مناسب کمک گرفتیم.****** [لینک](#)

برای این بخش ابتدا به کل المان‌های آرایه 64 تا اضافه کردم و بعدش هر جا که از 255 بیشتر بود را با 255 برابر کردم. برای حالت بعدی هم ابتدا یک **convert** به **HLS** زدم و بعدش کانال **L** را به اندازه 64 تا افزایش دادم و مجدداً هر جا از 255 بیشتر بود برابر با 255 شد.

خروجی اول



خروجی دوم:

