تمرین سری اول کامیار مرادیان زه آب ۹۹۵۲۲۱۰۴

سوال ۱:

برای کوآنتیزه کردن شدت روشنایی به ۶۴ سطح نیاز است که به تعداد ۶۴ سطح را تعریف کنیم و هر مقدار میانی هر دو سطح را به سطح زیرین نگاشت کنیم. بنابراین محدوده شدت روشنایی پیکسل‌ها تغییر نخواهد کرد و همچنان همان ۰ تا ۲۵۵ باقی خواهد ماند. چیزی که در این میانه تغییر می‌کند، گام‌هایی است که از یک سطح به سطح بعدی نیاز است که طی شود. به عبارتی شدت روشنایی‌های کمتری به نمایش در خواهند آمد و برخی از جزئیات تصویر مانند وارد زیر از بین خواهند رفت:

- نواحی‌ای که تغییرات جزئی در شدت نور آن‌ها رخ می‌دهد، به صورت نوارهایی با رنگ یکسان در خواهند آمد.
- نواحی‌ای که تغییرات جزئی‌ای در شدت نور آن‌ها وجود دارد، ممکن است در دو سطح متفاوت از شدت‌های تعریف شده قرار گیرند و نواحی مرزگونه را در تصویر به وجود آورند.

بنابراین به طور کلی در نواحی با کنتراست پایین ممکن است که جزئیات از میان بروند و دیگر کنتراستی وجود نداشته باشد.

با این حال با توجه به اینکه چشم انسان، می‌تواند تا ۵۰ سایهٔ مختلف از رنگ خاکستری را که با هم تفاوت دارند را به طور واضح از بقیه تشخیص دهد، بنابراین حداقل تعداد بیت‌های ۶ و ۷ (یعنی ۶۴ و ۱۲۸ سطح) ممکن است که برای ذخیره و نمایش تصویر کافی باشد. اما با این حال کنتراست تصویر از بین می‌رود و یک سری از جزئیات قابل نمایش نخواهند بود. (منبع این پاراگراف: https://hamamatsu.magnet.fsu.edu/articles/digitalimagebasics.html)

سوال ۲:

هر چه سرعت Shutter بیشتر باشد، در مدت زمان کوتاه‌تری دریچه باز خواهد ماند و درنتیجه انرژی کمتری وارد دوربین شده و تصویر از برهم‌نهی لحظات کمتری تشکیل خواهد شد. از طرفی با توجه به این که پرندهٔ در حال پرواز، در حرکت است بنابراین در صورتی که سرعت Shutter کم باشد، لحظات بیشتری از ناحیه تصویربرداری شود و در نتیجه بر اثر برهم‌نهی چندین صحنهٔ پشت سر هم، تصویری تار و نه چندان شفاف به دست آید. از طرفی در صورتی که مدت زمان باز ماندن دریچه، خیلی کم باشد میزان نور کمی وارد دوربین خواهد شد و تصویری تیره به دست خواهد آمد، که می‌توان با استفاده پردازش‌های مراحل بعدی این تاریکی را کم کرده و کنتراست تصویر را بالا برد اما با اینحال نباید سرعت شاتر را خیلی زیاد انتخاب کرد که تصویر خیلی تیره ثبت شود. لذا؛ در چنین شرایط بهتر است که سرعت Shutter زیاد باشد تا پرنده به وضوح در تصویر قرار گیرد اما نه آنقدر زیاد که تصویر خیلی تیره‌ای تیره ثبت شود.

سوال ۳:

تعریف هر یک از پارامترهای گفته شده و شرح تاثیر آن‌ها بر کیفیت عکس به صورت زیر هستند:

- سرعت Shutter: مدت زمانی است که دریچهٔ دوربین باز می‌ماند تا نور بازتاب شده از اشیا به دوربین برسد و تصویر مورد نظر ثبت شود. همانطور که در بالا گفته شد، این پارامتر بر روی شدت روشنایی تصویر نهایی و همچنین تصاویر متحرک تاثیرگذار است. به طوری که اگر سرعت آن زیاد باشد، تصویر با شدت روشنایی پایین و حالت فریز شدن به

اشیایی که سریع حرکت می‌کنند را ثبت می‌کند. از طرفی مقدار پایین آن، باعث شدت روشنایی بیشتر و همچنین ایجاد تاری حرکت می‌شود که خود عاملی است که حرکت شی را در تصویر نمایش می‌دهد.

- میدان دید: مشخص کنندهٔ زاویه‌ای از ناحیه است که می‌توانیم با استفاده از دوربین خود ثبت کنیم بدون اینکه دوربین را حرکت دهیم و تنها با استفاده از تغییر فاصلهٔ کانونی لنز دوربین. هر چه میدان دید گسترده‌تر باشد می‌توانیم ناحیهٔ بیشتری را در تصویر ثبت کنیم، درحالی که زاویهٔ میدان دید کوچک‌تر، باعث می‌شود که بتوانیم برای اشیا خاصی فوکوس کنیم. بنابراین حالت اول برای تصویربرداری از مناظر و یا مجموعه‌ای از اشیا است، در حالی که مورد دوم برای اشیای خاص و یا دور از دوربین مناسب‌تر است.
- عمق دید: مشخص‌کنندهٔ عمقی از ناحیه است که می‌خواهیم فوکوس بیشتری بر روی آن داشته باشیم. به‌طوری که اگر بزرگ باشد، کل تصویر در فوکوس قرار خواهد گرفت و هر چه کمتر باشد، می‌توانیم با استفاده از بخش‌های کوچکتری را در فوکوس قرار بدهیم و پس‌زمینه را تار کنیم.

مشکلات بوجود آمده برای عکاس ممکن است به دلیل موارد زیر باشند:

- تاری تصویر: سریع بودن شاتر و در نتیجه مدت زمان بیشتری باز ماندن دریچه که باعث ثبت لحظات بیشتر و تاری تصویر می‌شود. از طرفی ممکن است که به دلیل عمق دید پایین نیز باشد که در نتیجهٔ آن فوکوس پایینی روی نواحی دورتر بوجود آمده است.
- نبودن یوزپلنگ در فوکوس: عمق دید پایین
- عدم پوشش فضای زیادی از منظره: زاویهٔ میدان دید کوچک بوده و در نتیجه فضای کوچکی از منظره ثبت شده است.

با توجه به توضیحات داده شده و موارد ذکر شده در مسئله، برای هر مورد درخواست شده داریم:

- حس تعقیب و گریز و سرعت: برای این مورد نیاز است که شاتر مقدار نسبتاً کمی داشته باشد تا اینکه بتوان حرکت یوزپلنگ را در تصویر نمایان کرد. به عنوان مثال در حد ۱/۳۰ ممکن است که مناسب باشد. (با توجه به موارد گفته شده در کلاس و اسلایدهای در رابطه با حرکت و سرعت شاتر)
- فوکوس روی یوزپلنگ و تار کردن پس‌زمینه: استفاده از عمق دید مناسب طوری که بر روی یوزپلنگ فوکوس قرار بگیرد.
- قرارگیری منظره‌ای وسیع در کادر: برای این منظور نیاز است که میدان دید را افزایش دهد تا اینکه منظرهٔ بزرگی در تصویر قرار گیرد.

بنابراین عکاس می‌تواند در ابتدا میدان دید بزرگی را انتخاب کند تا محل یوزپلنگ را شناسایی کند و سپس میدان دید را اندکی کاهش دهد (اما نه آنقدر که مقدار منظرهٔ مورد نظرش کاسته شود) و سپس با تنظیم فوکوس از یوزپلنگ تصویربرداری کند.

سوال ۴:

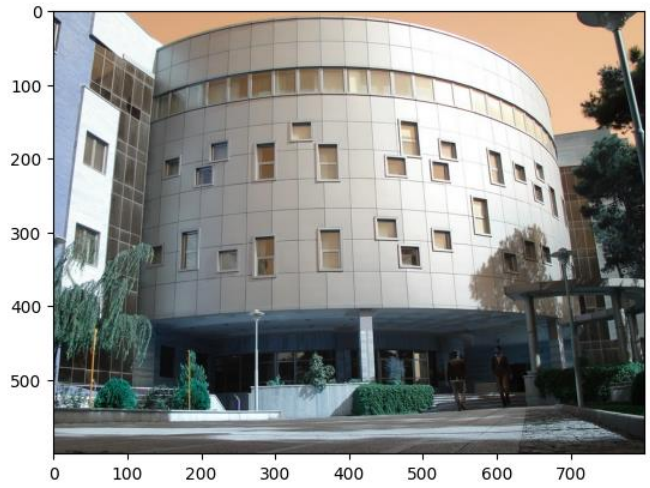
دو فایل نوتبوک بررسی شدند.

سوال ۵

بخش الف: دو تصویر بدست آمده به صورت زیر هستند:

- OpenCV:

- MatplotLib:



همانطور که مشاهده می‌شود بخش‌هایی از تصویر که آبی بوده‌اند به رنگ قرمز و بخش‌هایی که قرمز بوده‌اند به رنگ آبی در آمده‌اند. بنابراین می‌توان دریافت که جای کانال‌های قرمز و آبی در این دو ماژول با یکدیگر تفاوت دارد. با یک مقدار سرچ نیز محتوای زیر بدست آمد:

دلیل اصلی تفاوت در نتایج هنگام نمایش تصاویر با استفاده از تابع plt.imshow() از کتابخانهٔ Matplotlib و تابع cv2.imshow() از کتابخانهٔ OpenCV این است که OpenCV تصاویر را با فرمت BGR (آبی، سبز، قرمز) می خواند و نمایش می دهد، در حالی که Matplotlib فرض می کند که تصویر با فرمت RGB (قرمز، سبز، آبی) است.

بنابراین نیاز است که پیش از استفاده از plt.imshow() در ابتدا تصویر را از فرمت BGR به فرمت RGB تبدیل کنیم تا اینکه مقدار منتسب به هر پیکسل در هر کانال صحیح باشد و سپس از تابع نام‌برده استفاده کنیم.

برای این تبدیل می‌توانیم از قطعه کد زیر استفاده کنیم:

```
image_rgb = cv2.cvtColor(image, cv2.COLOR_BGR2RGB)
```

بعد از استفاده از کد ذکر شده، نتیجه به صورت زیر خواهد شد:

```
[30] ##########################Start Code#####################################
     I_RGB = cv2.cvtColor(I, cv2.COLOR_BGR2RGB)
     plt.imshow(I_RGB)
     plt.show()
     ############################End Code#####################################
```

بخش ب: برای عملیات گفته شده در هر بخش، می‌توان از توابع زیر استفاده کرد:

- cv2.flip: برای وارونه کردن تصویر است. این تابع دو آرگومان را دریافت می‌کند که یکی خود تصویر است و دیگری محوری است که حول آن تصویر وارونه می‌شود. به عنوان مثال با استفاده از axis=0 به صورت عمودی وارونه شده و در صورت استفاده از axis=1 به صورت افقی وارونه خواهد شد.
- cv2.hconcat: یک لیست را دریافت می‌کند و تصاویر را به ترتیب از چپ به راست در لیست بر روی تصویر خروجی قرار می‌دهیم.
- cv2.vconcat: یک لیست را دریافت می‌کند و تصاویر را به ترتیب قرار گرفته در لیست از چپ به راست، بر روی خروجی از بالا به پایین قرار می‌دهد.

یک نمونه کد به صورت زیر است:

```
##########################Start Code#####################################
I_vertical_revert = cv2.flip(I, 0)
I_1 = cv2.vconcat([I, I_vertical_revert])
cv2_imshow(I_1)
############################End Code#####################################
```

بنابراین تصاویر به دست آمده به ترتیب زیر هستند:

۱.

تمرین سری اول کامیار مرادیان زه آب ۹۹۵۲۲۱۰۴

۲.

۳.

۴.

۵. برای این مورد نیاز است که در ابتدا محل پنجره‌ها را پیدا کنیم. این کار را با استفاده از نمایش تصویر با استفاده از کتابخانهٔ `MatplotLib` انجام می‌دهیم. پس از این که حدود تصویر را بدست آوردیم با امتحان چند مقدار مختلف این محدوده را دقیق‌تر انتخاب می‌کنیم. سپس این محدوده از تصویر را با استفاده از `Slicing` انتخاب کرده و رنگ این بخش را تغییر می‌دهیم. این بخش را با توجه به اینکه فرمت کانال‌ها در `OpenCV` به صورت `BGR` است، انجام می‌دهیم. در پایان تصویر را با استفاده از تابع `cv2.imwrite` ذخیره می‌کنیم. تصویر نهایی بدست آمده به صورت زیر است:

۶. در ابتدا با توجه به اینکه هر تصویر رنگی شامل سه کانال است و تصاویر رنگی سه کاناله در `OpenCV` با فرمت `BGR` مورد استفاده قرار می‌گیرند، مقادیر مربوط به کانال‌های مختلف تصویر را از آن جدا می‌کنیم. بنابراین در هر بخش یک بردار بدست می‌آوریم که مربوط به یک کانال مشخص از تصویر است. سپس برای اینکه بتوانیم تصویر را در سه کانال و با استفاده از `OpenCV` نمایش دهیم، نیاز است که بردار بدست آمده را به یک آرایهٔ سه بعدی با همان ابعاد تصویر اولیه تبدیل کنیم، به طوری که برای هر کانال، مقادیر مربوط به کانال‌های دیگر صفر باشد. به عنوان مثال اگر بخواهیم که کانال آبی را نمایش دهیم، نیاز است که کانال‌های سبز و قرمز را صفر کنیم و مقادیر کانال آبی را برابر با کانال آبی تصویر اولیه قرار دهیم. به عنوان مثال یک نمونه کد در زیر آورده شده است که مربوط به نمایش کانال آبی تصویر است:

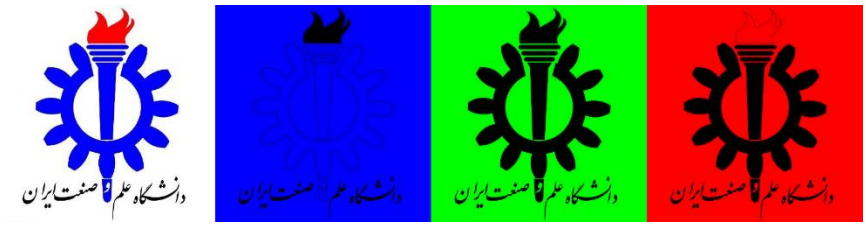
```
##########################Start Code#####################################
I_logo_B_one_dimensional = I_logo[:,:,0] # Format is BRG -> 1st dimension is blue
I_logo_B = np.zeros(I_logo.shape, dtype="uint8")
I_logo_B[:,:,0] += I_logo_B_one_dimensional
cv2_imshow(I_logo_B)
##########################End Code#######################################
```

در پایان نیز با استفاده از قطعه کد زیر تصاویر را در یک ردیف در کنار یکدیگر نمایش می‌دهیم:

```
# Concatenate the images horizontally
concatenated_image = cv2.hconcat([I_logo, I_logo_B, I_logo_G, I_logo_R])

cv2_imshow(concatenated_image)
```

نتیجهٔ نهایی به صورت زیر است:



تحلیل تصاویر بدست آمده: برای تشکیل رنگ سفید نیاز است که مقدار موجود در هر سه کانال بیشترین مقدار ممکن، یعنی ۲۵۵ باشد. همانطور که واضح است در پس زمینهٔ لوگوی دانشگاه، رنگ سفید وجود دارد و از طرفی مقدار مربوط به این پیکسل‌ها در تمام تصاویر بدست آمده از کانال‌ها نیز به رنگ موجود در همان کانال است. به عبارتی این پیکسل‌ها در کانال آبی، به رنگ آبی و در کانال قرمز، به رنگ قرمز و در کانال سبز، به رنگ سبز هستند. از طرفی برای اینکه یک بخش از تصویر به رنگ آبی در بیاید نیاز است که تنها کانال مربوط به رنگ آبی آن دارای مقدار غیر صفر و برابر با ۲۵۵ باشد (در اینجا و با توجه به تصویر لوگو می‌توان دریافت که تنها رنگ آبی در بخش دایره‌ای شکل لوگو نقش دارد.) همین موضوع برای بخش شعله‌های مشعل نیز برقرار است با این تفاوت که رنگ آن قرمز بوده، پس تنها کانال مربوط به رنگ قرمز آن مقدار غیر صفر دارد و سایر کانال‌ها در این پیکسل‌ها صفر هستند. به همین دلیل است که در بخش مشعل تنها کانال قرمز به رنگ قرمز است و سایر کانال‌ها سیاه‌رنگ هستند. همچنین با توجه به اینکه رنگ سبز در تصویر موجود نیست، در کانال مربوط به رنگ سبز تنها پس‌زمینه و حرف «و» به رنگ سبز هستند و یا به عبارتی مقدار کانال سبزشان ۲۵۵ است؛ که این بخش‌ها خود جزئی از رنگ سفید موجود در لوگو می‌باشند و همانطور که پیش از این ذکر شد نیاز است که مقدار این پیکسل‌ها در هر سه کانال مقدار غیر صفر و برابر با ۲۵۵ را داشته باشد. همچنین نوشته‌ها که به رنگ سیاه هستند نیاز است که مقدار موجود در هر سه کانال به صفر بوده و به همین دلیل است که این پیکسل‌ها در هر سه تصویر مربوط به سه کانال سبز، آبی و قرمز به رنگ سیاه هستند.