



درس پردازش تصویر دیجیتال
نیم سال دوم ۰۱-۰۲
استاد: دکتر پوره
تمرین سری اول

- کد هریک از سوالات را در فایل جداگانه بنویسید.
- کسب نمره مستلزم تحویل کدها، توضیحات و نتایج می باشد. در صورت عدم تحویل هر یک از این سه مورد، نمره آن سوال صفر خواهد بود.
- کدها ۴۰ درصد و نتایج ۳۰ درصد نمره هر سوال را شامل می شوند، حتما کدهای استفاده شده که منجر به نتایج فرستاده شده است را بفرستید و با اجرای کدها باید همان نتایجی که فرستاده اید قابل بازیابی باشد. در صورت اجرا نشدن کد یا بدست نیامدن نتایجی که فرستاده اید، نمره آن سوال صفر خواهد بود.
- توضیحات ۳۰ درصد نمره هر سوال را تشکیل می دهد. برای تمامی سوالات باید جزئیات روشی را که استفاده کرده اید را بیان کنید که مصحح متوجه کاری که شما برای رسیدن به نتیجه انجام داده اید، شود. توضیحات می تواند در یکی از دو فرمت های pdf یا ipynb باشد.
- کدهای شما تماماً باید توسط خودتان نوشته شده باشند. هرگونه استفاده از کد دیگران، اعم از دوستان و اینترنت، به هر شکل ممکن، اعم از کپی کردن یا همکاری کردن، تقلب محسوب می شود.
- در صورتیکه در انجام دادن تمرینات خود از هم فکری دیگران استفاده نموده اید باید نام آنها را در ابتدای گزارش خود ذکر نمایید.
- ددلاین این تمرین ۹ فروردین ۱۴۰۲ خواهد بود.
- ابهامات و پرسش های خود را می توانید از طریق آیدی تلگرام زیر بپرسید.

@MM_Mehran79

۱. Histogram Processing (۲۰ نمره)

Histogram Matching (آ)

تصویر Cecropia.jpg را طوری تغییر دهید تا هیستوگرام آن مشابه تصویر GeorgiaCypress.jpg شود. تمام مراحل پیاده سازی باید توسط خود شما انجام شود و نمیتوانید از کتابخانه های مختلف استفاده کنید. هیستوگرام تصویر بدست آمده را با نام result-01.jpg و خود تصویر بدست آمده را با نام result-02.jpg ذخیره نمایید. روش خود را به طور کامل در فایل pdf توضیحات یا در فایل ipynb به طور کامل توضیح دهید. فایل کد اصلی خود را با نام q1-1.py ذخیره نمایید.

Histogram Equalization (ب)

با استفاده از یکی از روش‌های Histogram Equalization (ordinary-HE or AHE) هیستوگرام تصویر LuebeckCityGate.jpg را اکوالایز کنید؛ هیستوگرام تصویر نهایی را نمایش دهید و با نام result-03.jpg ذخیره نمایید. همچنین تصویر نهایی را نیز نمایش دهید و با نام result-04.jpg ذخیره نمایید. روش خود را به طور کامل در فایل pdf توضیحات یا در فایل ipynb به طور کامل توضیح دهید. فایل کد اصلی خود را با نام q1-2.py ذخیره نمایید.

۲. Prokudin-Gorskii Images (۴۰ نمره)

در سال ۱۸۵۵ میلادی، دانشمندی اسکاتلندی به نام Maxwell Clerk James روشی برای تصویربرداری رنگی پیشنهاد نمود. در این روش از سه فیلتر استفاده می‌گردد که هریک به یکی از سه رنگ اصلی قرمز، سبز، و آبی حساس هستند. هر یک از این فیلترها شدت روشنایی یکی از سه رنگ اصلی را ثبت می‌کند. سپس، با ترکیب این سه رنگ می‌توان یک تصویر رنگی بازسازی کرد. ثبت تصاویر با این روش در زمان Maxwell به دلیل کیفیت پایین موادی که در اختیار بود در عمل به نتایج مطلوبی نرسید.

در اوایل قرن بیستم میلادی، دانشمندی روسی به نام Gorskii-Prokudin Mikhailovich Sergei از این روش استفاده کرده و با اجازه تزار در تمام روسیه سفر کرده و تصاویر زیادی از جاهای مختلف ثبت کرد. ولی در آن زمان هیچ تکنولوژی برای چاپ تصاویر رنگی وجود نداشت. در سال ۱۹۴۸ میلادی، کتابخانه ملی آمریکا این تصاویر را از پسران وی خریداری نموده و سعی در چاپ تصاویر رنگی از آنها نمود. ولی از آنجاییکه سه تصویر مربوط به سه رنگ اصلی کاملاً با هم منطبق نبودند، زمان زیادی برای تطابق آنها و ایجاد یک تصویر رنگی مناسب با ابزار و روش‌های آن موقع لازم بود. به همین دلیل تعداد کمی از آن تصاویر تبدیل به تصویر رنگی شدند. با پیدایش و گسترش تصاویر رقمی و استفاده از کامپیوتر، تصاویر سیاه و سفید Gorskii-Prokudin تبدیل به تصاویر رقمی شدند و تطابق آن‌ها با استفاده از کامپیوتر آسان‌تر و سریع‌تر شد. اکنون تمام این تصاویر در نسخه رنگی درست شده و در اختیار عموم قرار داده شده‌اند.

هدف شما در این تمرین نوشتن یک برنامه می‌باشد که با دادن هریک از این تصاویر رقمی شده به برنامه شما، تصویر رنگی متناظر آن به دست آید. سه مورد از این تصویرها که شامل تصویر **موزه بورودینو، لوکوموتیو بخار** و **تصویر دختران دهقان** هستند برای شما در نظر گرفته شده‌اند که در این تمرین برنامه خود را روی آنها اجرا نمایید. دقت کنید که این تصاویر ۱۶ بیتی هستند و شما باید روی همین تصاویر ۱۶ بیتی کار کنید. هریک از این تصاویر رقمی شده شامل سه تصویر سیاه و سفید است که به ترتیب از بالا به پایین مربوط به کانال‌های رنگ آبی، سبز، و قرمز هستند. این تصویر رقمی را در راستای عمودی به سه قسمت مساوی تقسیم کنید تا سه کانال قرمز و سبز و آبی به دست آیند. به دلیل نحوه تصویربرداری و همچنین وجود خطا، با روی هم گذاشتن این سه تصویر تطابق بین پیکسل‌های آنها به وجود نمی‌آید و تصویر واضحی ایجاد نمی‌شود. برای پیدا کردن تطابق بین سه کانال، فرض می‌کنیم که با یک جا به جایی در راستای X و Y می‌توان کانال‌ها را بر هم منطبق کرد، به این شکل که یکی از آن‌ها را ثابت گرفته و هر کدام از دو کانال دیگر را با مقداری جابجایی با کانال ثابت تطابق می‌دهید. توجه کنید که به دلیل اندازه بزرگ این تصاویر باید دقت کنید که پیاده سازی شما قادر به انجام این کار با سرعت بالا باشد.

برای مثال، تصویر قرمز را ثابت در نظر گرفته و تصویر سبز را روی آن جابجا کنید تا مکان تطابق را پیدا کنید. اگر پیکسل اول تصویر سبز که در گوشه بالا سمت چپ است را روی پیکسل اول تصویر قرمز دهید، پیکسل‌های دو تصویر با هم متناظر نخواهند بود. پیکسل اول تصویر سبز را باید در یک بازه در جهت افقی و عمودی حرکت دهید و در هر مکان میزان تطابق دو تصویر را حساب کنید و در نهایت منطبق‌ترین مکان را انتخاب کنید.

برای محاسبه میزان انطباق، می‌توانید پیکسل‌های دو تصویر که روی هم قرار می‌گیرند را از هم کم کنید و قدر مطلق

تفاضلات (فاصله L_1) و یا مربع تفاضلات (فاصله L_2) تمام پیکسل ها را با هم جمع کنید. هر جا که این مقدار کمینه باشد بهترین انطباق را به دست آورده‌اید. دلیل این امر این است که در تعداد زیادی پیکسل ها، پیکسل‌هایی که نقاط آنها تحت نور خورشید بوده‌اند، فارغ از رنگ آنها، در هر سه کانال مقدار زیادی دارند، ولی پیکسل‌هایی که در سایه بوده‌اند در هر سه کانال مقدار کمی دارند. البته در نظر داشته باشید که کناره‌های تصویر در سه کانال انطباق خوبی با هم ندارند و باعث می شوند حتی بهترین تطابق هم خطای زیادی داشته باشد.

به دلیل اندازه بزرگ تصاویر، اگر دو کانال را در تمام مکان ها در یک بازه بزرگ مانند $[100, 100]$ مقایسه کنید اجرای برنامه شما بسیار کند خواهد بود. برای حل کردن این مشکل می توانید ابتدا سه تصویر کانال ها را به اندازه کافی کوچک کنید و تطابق بین آن ها را به دست آورید و سپس به تدریج تصاویر با اندازه بزرگ تر را در نظر گرفته و در یک بازه در نزدیکی تطابق مرحله قبل جستجو نموده و جواب خود را بهتر کنید. در هر مرحله، تصویر نتیجه را به دست آورده و در گزارش خود این تصاویر را وارد نمایید. این مرحله را باید خودتان پیاده‌سازی نمایید و نمی‌توانید از توابع آماده استفاده نمایید. (از Gaussian Pyramid استفاده کنید.)

پس از پیدا کردن تطابق‌ها، یک تصویر رنگی از این سه تصویر به دست آورید. اطراف تصویر رنگی انطباق خوبی ندارد و رنگ‌های نادرستی ایجاد می شوند. این قسمت از کناره‌های تصویر از هر چهار طرف را به صورت اتوماتیک پیدا کرده و آن‌ها را حذف کنید.

تصاویر نهایی حاصل را با نام‌های `Girls-result-07.jpg` ، `Train-result-06.jpg` ، `Museum-result-05.jpg` ذخیره نمایید. این تصاویر را به صورت ۸ بیتی و با فشردن سازی `jpg` ذخیره نمایید تا حجم آنها زیاد نشود، ولی اندازه این تصاویر (تعداد سطرها و ستونها) همان اندازه تصاویر اصلی سه لایه باشد. میزان جابجایی لازم برای هر کدام از تطابق‌ها (برای مثال از سبز به قرمز و از آبی به قرمز) برای هر سه تصویر را در توضیحات خود ذکر کنید. کدهای شما باید به گونه‌ای باشند که از هر تصویر ورودی از این نوع یک تصویر رنگی مناسب با شرایط گفته شده بسازد.

روش خود را به طور کامل در فایل `pdf` توضیحات یا در فایل `ipynb` به طور کامل توضیح دهید. فایل کد اصلی خود را با نام `q2.py` ذخیره نمایید.

۳. Sharpening (۳۰ نمره)

در این سوال هدف این است که تصویر `OrcusMouth-blured.jpg` را شارپ‌تر کنید. این کار را به چهار روش زیر که در ادامه آمده است انجام دهید.

(آ) از رابطه $f - \alpha(f - f * g)$ استفاده کنید. مشابه آنچه در کلاس بررسی شد، از فیلتر گوسی به عنوان فیلتر هموارسازی استفاده کنید. اندازه و انحراف معیار فیلتر گوسی مناسب را در گزارش خود ذکر کنید. فیلتر گوسی خود را نمایش دهید و آن را با نام `result-08.jpg` ذخیره نمایید. تصویر هموار شده $(f * g)$ را با نام `result-09.jpg` ذخیره کنید. تصویر ماسک آنشارپ $(f - (f * g))$ را نیز با نام `result-10.jpg` ذخیره کنید. مقدار α را در گزارش خود ذکر کنید و تصویر نهایی را با نام `result-11.jpg` ذخیره کنید.

(ب) اینبار به جای استفاده از فیلتر گوسی، از `box filter` استفاده کنید. تصویر هموار شده $(f * g)$ را با نام `result-12.jpg` ذخیره کنید. تصویر ماسک آنشارپ $(f - (f * g))$ را نیز با نام `result-13.jpg` ذخیره کنید. مقدار α در نظر گرفته شده برای این روش را نیز در گزارش خود ذکر کنید و تصویر نهایی را با نام `result-14.jpg` ذخیره کنید.

(ج) این بار روند شارپ کردن تصویر را در حوزه فرکانس اعمال میکنیم. به این منظور از رابطه زیر استفاده میکنیم:

$$\mathcal{F}^{-1}\{(1 + kH_{HP}).F\}$$

H_{HP} فیلتر بالاگذر استفاده شده است که در این قسمت میخواهیم معادل فرکانسی فیلتر استفاده شده در بخش (آ) را به این عنوان استفاده کنیم تا نتیجه مشابه نتیجه قسمت (آ) شود. تبدیل فوریه تصویر را نمایش دهید و آن را با نام result-15.jpg ذخیره کنید. فیلتر بالاگذر را نمایش دهید و با نام result-16.jpg ذخیره کنید. مقدار k را در گزارش خود ذکر کنید. اندازه $(1 + kH_{HP}).F$ را نمایش دهید و با نام result-17.jpg ذخیره کنید. تصویر نهایی را با نام result-18.jpg ذخیره کنید.

(د) در این قسمت نیز از رابطه استفاده شده در بخش قبل استفاده میکنیم و اینبار از فیلتر بالاگذر ایده آل استفاده میکنیم. فیلتر بالاگذر را نمایش دهید و با نام result-19.jpg ذخیره کنید. مقدار k را در گزارش خود ذکر کنید. اندازه $(1 + kH_{HP}).F$ را نمایش دهید و با نام result-20.jpg ذخیره کنید. تصویر نهایی را با نام result-21.jpg ذخیره کنید.

- تصویر نهایی هر یک از چهار روش بالا را با هم مقایسه و تحلیل کنید.
- سرعت پردازش کد بخش (آ) را با بخش (ج) مقایسه و تحلیل کنید.

روش خود را به طور کامل در فایل pdf توضیحات یا در فایل ipynb به طور کامل توضیح دهید. فایل کد اصلی خود را با نام q3.py ذخیره نمایید.

۴. Color Processing (۱۰ نمره)

تصویر Poinsettia.jpg را در نظر بگیرید. رنگ برگهای قرمز رنگ را به سبز تغییر دهید. تصویر حاصل را با نام result-22.jpg ذخیره نمایید.

روش خود را به طور کامل در فایل pdf توضیحات یا در فایل ipynb به طور کامل توضیح دهید. فایل کد اصلی خود را با نام q4.py ذخیره نمایید.