

# بسم الله الرحمن الرحيم

دانشگاه علم و صنعت ایران

پاییز ۱۳۹۹

تحويل: دوشنبه ۲۱ مهر

تمرین سری سوم

مبانی بینایی کامپیوتر

## توضیحات

- در ابتدای هر سوال یک نشانگر وجود دارد که مشخص می‌کند آن سوال تشریحی است یا برنامه‌نویسی؛ اگر نشانگر ✎ یعنی سوال تشریحی است و اگر نشانگر ☐ باشد یعنی سوال یک سوال برنامه‌نویسی است.
- تمامی تصاویری که برای حل سوالات نیاز هستند در پوشه images قرار دارند.
- برای حل سوالات برنامه‌نویسی فایل HW3.ipynb را تا انتها دنبال کنید.
- برای تمامی سوال‌های پیاده‌سازی سعی شود توضیحی تشریحی در مورد نحوه پیاده‌سازی کد و بررسی و مقایسه نتایج در پاسخ‌های تشریحی آورده شود.
- خواهشمندیم اگر مطلبی را از اینترنت استفاده کردید حتماً منبع آن را مشخص کنید.
- تصاویر خروجی در کنار فایل HW3.ipynb ذخیره می‌شوند. لطفاً آن‌ها را در گزارش خود بیاورید و همراه با فایل pdf مربوط به پاسخ سوالات تشریحی فشرده کرده و ارسال کنید.

## سوالات

۱. ✎ در شکل زیر ماتریس بالا، فیلتر و ماتریس پایین، تصویر هستند. برای zero-padding چند ردیف صفر به هر سمت از تصویر اضافه می‌گردد؟ با در نظر گرفتن zero-padding، تصویر را با فیلتر کانوالو کنید. در حالت BOARDER\_REFLECT تصویر را با فیلتر کانوالو کنید و نتیجه را با حالت قبل مقایسه کنید (۲۰).

۱	۳	۱
۱	۰	۱
۱	۲	۱

فیلتر

۱	۲	۱	۶
۷	۱	۱	۱
۳	۱	۲	۰
۱	۴	۰	۲

تصویر

۲. به صورت مفهومی توضیح دهید دو کرنل زیر چه پردازشی بر روی تصویر انجام می‌دهند (۱۰)؟

a.

-۱	۰	۱
----	---	---

b.

۱
۲
۱

۳. فرض کنید تصویر زیر یک تصویر سطح خاکستری است (محدوده رنگ ممکن بین صفر تا ۲۵۵ است).

به صورت مرحله به مرحله الگوریتم CLAHE را بر روی این تصویر با اندازه پنجره  $3 \times 3$  و حالت‌های

$\text{ClipLimit} = 1$  و  $\text{ClipLimit} = 2$  اعمال کنید (۲۰).

۴۶	۵۱	۵۷	۵۹
۴۶	۵۲	۵۸	۶۰
۴۶	۵۲	۵۸	۶۰

برای این سوال ممکن است بخواهید برای ارزیابی پاسخ خود، پیاده سازی‌ای انجام دهید ولی نتیجه‌ای که در پیاده سازی به دست می‌آورید متفاوت با نتیجه‌ای است که حساب خواهید کرد. بررسی کنید که چرا

پاسخ شما و پاسخ پیاده سازی‌ای که کردید متفاوت است (۲۰)؟

۴. گاهی برای کارهای پردازش تصویر لازم است هیستوگرام یک تصویر را شبیه به هیستوگرام یک تصویر

دیگر یا هیستوگرام از پیش تعیین شده بکنیم برای این کار از تطبیق هیستوگرام استفاده می‌کنیم که در

آن از تابع متعادل سازی هیستوگرام تصویر ورودی استفاده می‌کنیم و معکوس تابع متعادل سازی

هیستوگرام تصویر مرجع را بر روی آن اعمال می‌کنیم. برای تصویر Q4.jpg و تصویر مرجع Q4\_ref.jpg

الگوریتم تطبیق هیستوگرام را بدون توابع پیش ساخته پیاده سازی کنید. برای این کار تابع

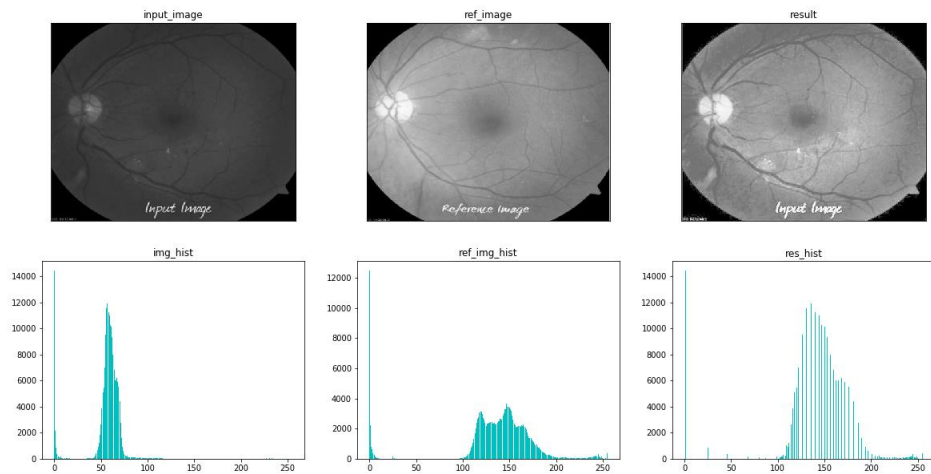
histogram\_matching را کامل کنید. در این تابع ورودی تصویر اولیه و تصویر مرجع هستند و خروجی

تصویر حاصل از تطبیق هیستوگرام تصویر اولیه بر تصویر مرجع است. تصاویر ورودی را به صورت تک

کاناله بخوانید. در تصویر پایین یک نمونه از چیزی که مورد انتظار است برای ورودی‌های دیگری غیر از

ورودی‌های این سوال، آمده است. شاید در خود تصویر تغییرات اعمال شده محسوس نباشد ولی در

هیستوگرام‌ها مشخص است که چه کاری صورت گرفته است (۳۰).



۵. تابع `Gaussian_filter` را کامل کنید. ورودی این تابع اندازه کرنل (کرنل را مربعی با اندازه فرد در نظر بگیرید) و انحراف معیار تابع گاوسی است. در این تابع بدون استفاده از توابع آماده در `OpenCV` کرنل مد نظر را ساخته با استفاده از تابع `filter2D` از کتابخانه `OpenCV` آن را با تصویر کانالو کنید. بار دیگر تصویر را با استفاده از تابع `GaussianBlur` در کتابخانه `OpenCV` فیلتر کرده و نتایج را از نظر کیفیت و سرعت با هم مقایسه کنید. خروجی مورد انتظار برای این سوال این است که دو خروجی تقریباً مشابه هم شوند. برای این سوال تصویر `img5.jpg` را به صورت تک کاناله خوانده و تغییرات بالا را روی آن اعمال کنید. برای نحوه استفاده از توابع `filter2D` و `GaussianBlur` از کتابخانه `OpenCV` به این [لینک](#) مراجعه کنید (۲۰).

$$\left( G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \right)$$

• موفق باشید.