



دانشگاه صنعتی شریف  
دانشکده‌ی مهندسی کامپیوتر

## سیستم‌های چندرسانه‌ای (۱-۴۰۳۴۲)

نیم‌سال دوم ۹۸-۹۹

استاد: مهدی امیری

پردازش تصویر

تاریخ تحویل: ۱۹ اردیبهشت

تمرین سری سوم

### به موارد زیر توجه کنید:

- پاسخ تمرین را به همراه تمامی فایل‌ها به صورت یک فایل فشرده، که نام آن در قالب MMS\_HW3\_LastName\_StudentID باشد، به آدرس ایمیل درس به نشانی mms2020spring@gmail.com بفرستید. لطفا عنوان ایمیل خود را همانند قالب فوق قرار دهید.
- مهلت ارسال پاسخ تمرین تا ساعت ۲۳:۵۹ روز اعلام‌شده است. بهتر است نوشتن تمرین را به ساعات پایانی موکول نکنید.
- همکاری و هم‌فکری شما در حل تمرین مانعی ندارد، اما پاسخ ارسالی هر کس حتما باید توسط خود شخص نوشته شده باشد.
- مبنای درس، اعتماد بر پاسخ ارسالی از سوی شماست؛ بنابراین ارسال پاسخ به این معناست که پاسخ آن تمرین، توسط شما نوشته شده‌است. در صورت تقلب یا اثبات عدم نوشتار پاسخ حتی یک سوال از تمرین، برخورد شدیدی صورت خواهد گرفت.
- پاسخ سوالات مطرح شده در صورت تمرین‌ها در قالب یک گزارش با فرمت PDF، نمودارها و شکل‌های خروجی m فایل‌ها و خود m فایل‌ها می‌باشد.

در این تمرین شما با برخی ابزارها و روش‌های اصلی پردازش تصویر آشنا خواهید شد. آنچه که بایستی تحویل داده شود، شامل پاسخ سوالات مطرح شده در صورت تمرین‌ها (در قالب يك گزارش با فرمت ، PDF نمودارها و شکل‌های خروجی و m فایل‌ها) می‌باشد.

## سوال ۱. تبدیل تصاویر

در این تمرین هدف آشنا شدن و درک ماتریس تصاویر است. یک تصویر متشکل از یک ماتریس است که در هر خانه آن، عددی قرار گرفته است. با جابه‌جایی صحیح خانه‌های ماتریس، می‌توان تصویر را تبدیل کرد. در این تمرین ابتدا تصویر cameraman.tif را با دستور `imread` بخوانید. سپس با فرض اینکه مرکز تصویر به عنوان مبدا مختصات قرار دارد، تصویر تقارن یافته آن را نسبت به محور  $x$  و  $y$  و خط  $y = x$  و  $y = -x$  بیابید و نمایش دهید. برای این کار می‌توانید یک تبدیل بیابید و از دستور `imwarp` متلب استفاده نمایید. برای این سوال یک تابع با نام `Q1_convert` ارائه دهید؛ که تصویر فوق به عنوان ورودی اول تابع باشد و با استفاده از ورودی دوم حالت تقارن را مشخص کنید. خروجی تابع، باید تصویر تقارن یافته باشد. در مورد کد نوشته شده، توضیح دهید.

## سوال ۲. تغییر سطوح روشنایی تصویر

الف. با استفاده از تابع `histeq` الگوریتم `Equalization Histogram` را که در درس با آن آشنا شده‌اید، بر روی تصویر `nrg.tif` پیاده کنید و نتیجه را مشاهده کنید. در خروجی علاوه بر تصویر اولیه و تصویر بهبودیافته، با استفاده از تابع `imhist` هیستوگرام دو تصویر را نیز نمایش دهید. تمامی این ۴ تصویر را در یک تصویر با دو سطر و دو ستون نمایش بدهید. برای این کار از تابع `subplot` کمک بگیرید. کد نوشته شده را در یک فایل با نام `Q2_1` ارائه دهید.

ب. در تابع `histeq` می‌توانیم تعداد سطوح خاکستری تصویر نهایی را به عنوان ورودی به این تابع بدهیم. با این کار سطوح روشنایی تصویر تغییر می‌کند. مقدار این پارامتر را ۲، ۴، ۱۶ و ۲۵۶ قرار دهید. تصاویر خروجی و `Histogram` آن‌ها را در یک تصویر با چهار سطر و دو ستون نمایش دهید. نتایج را با توجه به هیستوگرام تصاویر و کیفیت آنها مقایسه کنید و در مورد تفاوت آنها توضیح دهید. کد نوشته شده را در یک فایل با نام `Q2_2` ارائه دهید.

پ. تابع `imadjust` را بر روی همان تصویر قسمت الف اعمال کنید. تصویر اولیه، تصویر حاصل از اعمال این تابع و تصویر حاصل از اعمال تابع `histeq` را در یک تصویر نمایش دهید و در مورد تفاوت `imadjust` و `histeq` توضیح دهید. `Histogram` تصویر حاصل از اعمال تابع `imadjust` و تابع `histeq` را نیز در یک تصویر نمایش دهید و توضیح دهید با هم چه تفاوتی دارند. کد نوشته شده را در یک فایل با نام `Q2_3` ارائه دهید.

ت. با استفاده از تابع `imadjust` می‌توانیم عمل `Correction Gamma` را بر روی تصاویر انجام دهیم. تصویر `fomopan_grayscale_512.jpg` را با مقدارهای ۲.۰، ۶.۰، ۶.۱ و ۲.۲ برای گاما پردازش کنید و تصاویر حاصل را در یک تصویر نمایش دهید. در مورد نقش `Correction Gamma` در بهبود تصاویر و تاثیر مقادیر `Gamma` توضیح دهید. کد نوشته شده را در یک فایل با نام `Q2_4` ارائه دهید.

ث. تاکنون توابع معرفی شده را بر روی یک تصویر `Grayscale` اعمال کردیم. حال می‌خواهیم با یک تصویر رنگی کار کنیم. برای این منظور، باید از فضای رنگی استفاده کنید که در آن `Intensity` یک مولفه جدا باشد. برای این سوال این کار را با دو فضای رنگ `YCbCr` و `HSV` انجام دهید. از تصویر `mzrun_color_512.jpg` استفاده کنید و برای هر کدام از فضاهای رنگ خواسته شده، ابتدا آن را به آن فضای رنگ برده، سپس مولفه `Intensity` را شناسایی کنید. سپس

توابع histeq و imadjust را بر آن اعمال کنید. در نهایت تصویر را مجدداً به فضای رنگ RGB برگردانید و نمایش دهید. تصویری شامل تصاویر بهبود یافته از اعمال توابع histeq و imadjust و تصویر اصلی برای فضای رنگ Lab و YCbCr نمایش دهید. به عبارت دیگر برای هر فضای رنگ، یک تصویر شامل دو تصویر بهبود یافته و تصویر اصلی ارائه شود. کد نوشته شده را در یک فایل با نام Q2\_5 ارائه دهید و در مورد آن توضیح دهید.

### سوال ۳. بهبود تصاویر

در این قسمت می‌خواهیم با تعدادی دیگر از توابع Toolbox Processing Image که برای بهبود کیفیت تصاویر پیاده‌سازی شده‌اند، آشنا شویم. تصویر lena\_gray\_256.tif را در نظر بگیرید.

الف. با استفاده از تابع imnoise نویز noise paper & salt با Density Noise برابر ۰.۵۰ به تصویر اضافه کنید. سپس تصویر اصلی را با استفاده از فیلترهای averaging و median هر دو با اندازه پنجره ۳×۳ بازبینی کنید. تصاویر حاصل از ایجاد نویز و رفع نویز را در یک تصویر و کد نوشته شده را در یک فایل با نام Q3\_1 ارائه دهید. مشاهدات خود را گزارش دهید. کدام فیلتر بهتر عمل می‌کند؟

ب. یکی دیگر از انواع نویزی که در متلب پیاده‌سازی شده است، نویز گوسی است. با استفاده از تابع imnoise نویز گوسی به تصویر اضافه کنید. مقدار پارامتر میانگین در این تابع را صفر و واریانس را برابر ۰.۱۰ قرار دهید. با این دستور، به هر پیکسل از تصویر با احتمال بیشتر مقدار صفر و با احتمال کمتری مقادیر مثبت و منفی اضافه می‌شود. هر چه مقدار واریانس کمتر باشد، احتمال اضافه شدن صفر بیشتر می‌شود. تصویر حاصل از اضافه کردن نویز را ارائه دهید. سپس با استفاده از فیلتر گوسی، تصویر را بازسازی کنید. برای ساختن فیلتر گوسی از تابع fspecial با ابعاد پنجره ۵×۵ و مقدار واریانس برابر یک استفاده کنید. تصاویر اصلی، تصویر حاصل از ایجاد نویز و تصویر بازسازی شده را در یک تصویر و کد نوشته شده را در یک فایل با نام Q3\_2 ارائه دهید و مشاهدات خود را گزارش کنید.

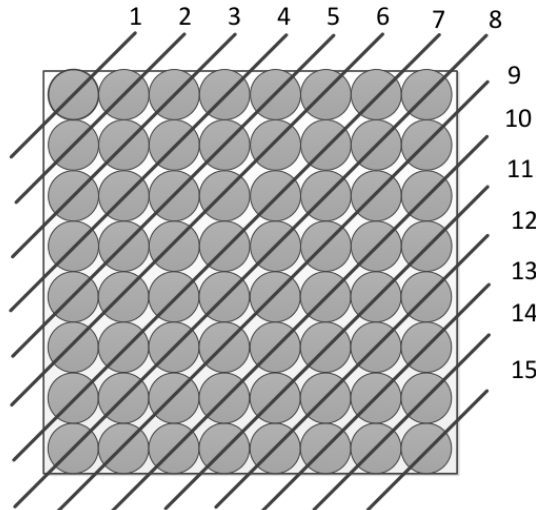
پ. یکی از راه‌های بررسی اختلاف دو تصویر با یکدیگر مقایسه error mean-squared آن دو است. اگر دو تصویر کاملاً یکسان باشند، مقدار mse برابر صفر خواهد بود. برای تصاویر رفع نویز شده توسط فیلترهای average، median و gaussian مقدار mse نسبت به تصویر اصلی را محاسبه کنید. کد نوشته شده را با نام Q3\_3 ارائه داده و مقادیر به دست آمده را در جدولی مشابه با جدول زیر ارائه دهید.

نوع فیلتر	median	average	gaussian
mse			

ت. در این قسمت می‌خواهیم با آشکارسازی لبه‌ها آشنا شویم. با کمک دستورات متلب می‌توانیم لبه‌های یک تصویر را در قالب یک تصویر باینری نمایش دهیم. لبه‌های تصویر ارائه شده در این سوال را با الگوریتم canny و sobel مشخص کنید. می‌توانید از تابع edge در متلب استفاده کنید. تصاویر حاصل را در یک تصویر نمایش دهید. کد نوشته شده را با نام Q3\_4 ارائه داده و مشاهدات خود را گزارش کنید.

## سوال ۴. فشرده‌سازی تصاویر

در این تمرین هدف پیاده‌سازی قسمتی از فشرده‌سازی JPEG است. در این سوال تابعی بنویسید که از تصویر DCT بگیرد. سپس آن را به بلاک‌های  $8 \times 8$  تبدیل کنید. مطابق با تصویر زیر، از خط شماره ۲ به بعد، همه ضرایب DCT را صفر کنید. سپس دوباره از کل تصویر معکوس DCT گرفته و تصویر اصلی را بازسازی کنید. عملیات بالا را مجدداً تکرار کنید. با این تفاوت که مطابق با شکل زیر یکبار از خط شماره ۶ به بعد و بار دیگر از خط شماره ۱۱ به بعد ضرایب تبدیل DCT را صفر کنید و مجدداً تصویر اصلی را بازسازی کنید. برای این سوال از تصویر lena\_gray\_256.tif استفاده کنید.



با این کار شما عملیاتی شبیه به فشرده‌سازی JPEG با کیفیت‌های مختلف را انجام داده‌اید. سه تصویر بازسازی شده به همراه تصاویر اصلی را در یک تصویر ارائه دهید. مقدار mse را برای هر سه تصویر بازسازی شده و تصویر اصلی در جدولی مشابه با جدول سوال قبل ارائه دهید و توضیح دهید صفر کردن تعداد بیشتری از ضرایب DCT در بلاک  $8 \times 8$  چه تاثیری بر روی کیفیت می‌گذارد. وجود تعداد بیشتری صفر در تبدیل DCT یک تصویر چگونه می‌تواند به کاهش حجم تصویر کمک می‌کند؟ کد نوشته شده را در یک فایل با نام Q4\_DCT ارائه دهید و در مورد آن توضیح دهید.

**راهنمایی:** برای تبدیل DCT یک تصویر راه‌های متفاوتی وجود دارد. برای نمونه می‌توانید بصورت زیر عمل کنید

```
D = dctmtx(8);
dct = @(block_struct) D * block_struct.data * D';
DCT_image = blockproc(A, [8 8], dct, 'PadPartialBlocks', true, 'PadMethod', 'replicate')
;
```

و برای عکس تبدیل DCT،

```
invdct = @(block_struct) D' * block_struct.data * D;
image = blockproc(DCT_image, [8 8], invdct, 'PadPartialBlocks', ...
    true, 'PadMethod', 'replicate');
image = uint8(image);
```

توجه کنید که خروجی تابع blockproc کل تصویر را بر می‌گرداند. شما باید تصویر را به بلاک‌های  $8 \times 8$  تقسیم کرده و در هر بلاک ضرایب معینی را پس از بدست آوردن DCT صفر کنید. برای این کار، یک راه ساختن یک mask و ادغام آن با تصویر با بلاک‌های  $8 \times 8$  است. این کار مجدداً با تابع blockproc قابل انجام است. در مورد این تابع و نحوه استفاده از آن جست‌وجو نمایید. همچنین در صورت نیاز توابع مربوط scanning zig-zag ضمیمه تمرین شده است. برای پاسخ به این سوال

می‌توانید از روش‌های دیگر هم استفاده کنید. در هر حال ارائه تصاویر حاصل از انجام عملیات و توضیح روش مورد استفاده و کد ارائه شده الزامی است.

### **موفق باشید**

موضوع تمرین بعد: موضوع تمرین بعد: پردازش ویدئو