گزارش پروژه اول

درس انتقال داده ها

نگین حقیقی – 99521226

استاد درس: دکتر دیانت

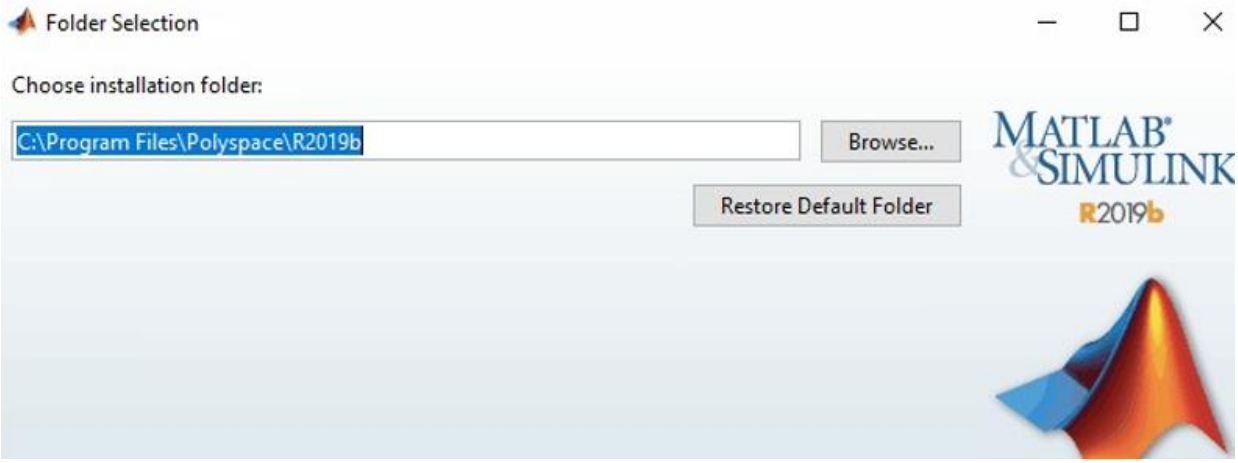
نیم سال دوم 1402-1401

مقدمه:

در این پروژه قصد داریم یک فایل تصویری را وارد نرم افزار MATLAB کرده و در ابتدا به آن نویز اضافه کنیم. و سپس نویز را حذف کنیم. برای انجام این کار، هشت گام نام برده شده در صورت پروژه را طی میکنیم و در هر مرحله توضیحاتی درباره کار انجام شده در آن میدهیم.

گام اول)

در این گام به نصب نرم افزار متلبMATLAB میپردازیم. از سایت soft98.ir آن را دانلود کرده و مراحل نصب را طی میکنیم. آدرس آن را همان مقدار دیفالت در پوشه program files گذاشتم.



گام دوم)

در این مرحله، ابتدا یک عکس دلخواه از اینترنت دانلود کرده (من عکس دسته گل انتخاب کردم تا رنگی بودن آن مشخص تر باشد). سپس عکس را وارد نرم افزار متلب میکنیم.

```
set(gcf, 'Position',  [100, 40, 900, 600])
% reading image
pic = imread('flower.jpg');
subplot(2,3,1), imshow(pic), title('<First pic>');
```

گام سوم)

در این گام قصد داریم فایل وارد شده را از RGB به یک فایل Gray-Scale 8 bit تبدیل کنیم. به همین منظور از دستور rgb2gray مطابق زیر استفاده کرده ام.

```
% Convert to grayscale
grayPic = rgb2gray(pic);
```

نکته 1) تصاویر در حالت کلی به 3 دسته باینری، رنگی، gray scale تقسیم میشوند.

تصاویر باینری، تنها شامل دو رنگ سیاه و سفید هستند. به طوری که هر پیکسل آنها اگر بیشتر از یک مقداری باشد سفید در نظر گرفته میشود. یعنی عدد 1 و اگر کمتر از آن باشد، سیاه یعنی صفر در نظر گرفته میشود.

تصاویر Gray-Scale شامل سطح های مختلفی از رنگ خاکستری هستند. این تصاویر دارای 652 سطح خاکستری هست.

تصاویر رنگی نیز شامل سه باند رنگ هستند که شامل rgb میشود (red, green, blue) که هرکدام رنگ متفاوتی دارند. مقایس این تصاویر در کل 24 bit/pixel است.

گام چهارم)

حال با استفاده از دستورات imshow و imwrite تصویر حاصل را مشاهده میکنیم و آن را در کامپیوتر خود ذخیره میکنیم.

نکته2) برای ذخیره تصویر به صورت کامل و بدون هیچ فشرده سازی با اتلاف در کامپیوتر، از فرمت png استفاده میکنیم..

```
% Convert to grayscale
grayPic = rgb2gray(pic);
subplot(2,3,2), imshow(grayPic), title('<Grayscale Pic>');

imwrite(grayPic, 'gray_pic.png');
```

تصویر حاصل:

<First pic>

<Grayscale Pic>

گام پنجم)

در این گام تصویر rgb ذخیره شده را دوباره با دستور imread میخوانیم و به تصویر خاکستری تبدیل میکنیم. سیگنال تصویر حاصل، از نوع سیگنال انرژی هست (و توان آن صفر است). مقدار انرژی آن اینگونه محاسبه میشود که تمام سطح های خاکستری عکس را باهم جمع میزنیم.

```
% Energy of gray pic
energy = sum(grayPic(:));
fprintf('The energy of gray phote = %d\n', energy);
```

گام ششم)

حال به تصویر خاکستری ای که در مرحله قبل ایجاد کردیم، یک نویز گاوسی با میانگین صفر و پراش (variance) دلخواه (اینجا 0.3) اضافه میکنیم. مطابق قطعه کد زیر، این کار را با دستور imnoise انجام داده ایم.

```
% Noise
gaussian_pic = imnoise(grayPic, 'gaussian', 0, 0.03);
imwrite(gaussian_pic, 'gaussian_pic.png');
subplot(2,3,3), imshow(gaussian_pic), title('Grayscale gaussian noise');
```

برای محاسبه snr از یک تابع آماده خود متلب به نام snr استفاده کرده ایم.

```
p_snr = snr(double(grayPic(:)), double(grayPic(:)) - double(grayPic(:)));
fprintf('this is <before> snr: %d\n', p_snr);

p_snr = snr(double(grayPic(:)), double(grayPic(:)) - double(gaussian_pic(:)));
fprintf('this is <after> snr: %d\n', p_snr);
```

نکته3) در کل snr ، حاصل تقسیم توان سیگنال به نویز است. که یکای آن dB است. در حالتی که نویزی نداشته باشیم، حاصل این متغیر، بینهایت است. زیرا مقدار نویز صفر بوده و طبق گفته بالا، صفر در مخرج کسر قراره گرفته و در نتیجه حاصل را بینهایت میکند.

گام هفتم)

در این گام، تصویر نویزی ساخته شده در مراحل قبل را به حوزه فرکانس برده ایم.

```
% Frequency domain
freqdomain = fftshift(log(abs(fft2(gaussian_pic))));
subplot(2,3,4), imshow(freqdomain, []), title('<Gaussian in Freq>');
```

نکته4) در FFT هر تابع به صورت جمع توابع سینوسی و کسینوسی نوشته میشود. هدف کلی fft جابه جایی عکس بین حوزه های frequencyوspatial هست. از آن برای بردن یک عکس به حوزه فرکانس استفاده میکنیم. بیشترین فرکانس در کناره ها و کمترین فرکانس در وسط اتفاق می افتد.

گام هشتم)

حال قصد داریم نویز ایجاد شده را از تصویر حذف کنیم. دو روش برای اینکار وجود دارد، که یکی روش average filter و دیگری روشmedian filter است که هر دو را روی تصویر انجام داده ایم. هر دوی این روش ها در تابع آماده متلب هستند و مطابق زیر انجام داده ام:

```
% Frequency domain
freqdomain = fftshift(log(abs(fft2(gaussian_pic))));
subplot(2,3,4), imshow(freqdomain, []), title('<Gaussian in Freq>');

X=ones(5,5)/25;
method1_pic=imfilter(gaussian_pic, X);
subplot(2,3,5), imshow(method1_pic), title('<Average filter method>');
imwrite(method1_pic, 'method1_pic.png');

method2_pic=medfilt2(gaussian_pic);
subplot(2,3,6), imshow(method2_pic), title('<Median filter method>');
imwrite(method2_pic, 'img2.png');

% Method1 psnr:
[psnr1, snr1] = psnr(gaussian_pic, method1_pic);
fprintf('\n Average filter psnr: ', psnr1);
% Method2 psnr:
[psnr2, snr2] = psnr(gaussian_pic, method2_pic);
fprintf('\n median filter psnr:', psnr2);
```

```
The energy of gray phote = 24695725
this is <before> snr: Inf
this is <after> snr: 8.009587e+00
```

```
Average filter psnr: 15.566
median filter psnr: 16.063
```

توابع آماده هر دو روش در شکل بالا استفاده شده است. برای روش اول از تابع imfilter و برای روش دوم از تابع medfilt2 استفاده میشود. و در آخر نیز طبق نکته5، مقدار psnr محاسبه شده است. که از همان تابع آماده psnr در متلب استفاده میشود. مطابق شکل آخر، طبق اعداد نشان داده شده، دقت روش دوم بالاتر است.

نگین حقیقی-بهار 1402