



انتقال داده‌ها

موضوع پروژه : اضافه و حذف کردن نویز به تصویر

استاد درس: ابولفضل دیانت

نام دانشجو: محمدحسین حسینی - ملیکا محمدی فخار

نیم‌سال اول

سال تحصیلی ۱۴۰۲-۱۴۰۳

۱.۰. گام اول

۱

۱.۰ گام اول

با توجه به این که حجم برنامه matlab به شدت بالاست (حدود ۱۸ گیگابایت) از کامپایلر آنلاین octave-online استفاده کردیم.

۲.۰ گام دوم

در این گام عکسی را به عنوان ورودی برنامه‌مان انتخاب می‌کنیم.



شکل ۱: عکس ورودی

۱.۲.۰ بررسی کد

حال کد متلب این گام را بررسی می‌کنیم:

```
1 filePath = 'first.png';  
2 img = imread(filePath);  
3 figure;  
4 imshow(img)
```

حال به بررسی خطوط کد می‌پردازیم:

- اول مسیر فایل داده‌شده را مشخص می‌کنیم
- سپس با دستور `imread` آن فایل را می‌خوانیم
- و با دستور `imshow` فایل مرحله قبل را در خروجی نشان می‌دهیم.

۳.۰ گام سوم

در این گام با توجه به راهنمایی گفته شده در داک پروژه از تابع پیش تعریف‌شده `rgb2gray` استفاده می‌کنیم.

۱.۳.۰ بررسی کد

حال کد متلب این گام را بررسی می‌کنیم :

```
1 grayImg = rgb2gray(img);
```

۲.۳.۰ انواع تصاویر

- `binary images` : این نوع تصاویر در یک ماتریس $m * n$ ذخیره می‌شوند که رنگ سیاه در آن معادل صفر و رنگ سفید معادل یک می‌باشد
- `grayscale images` : در این نوع تصاویر که در یک ماتریس $m * n$ ذخیره می‌شوند هر عضو آن شدت رنگ آن پیکسل را نشان می‌دهد. که بزرگترین عدد برای رنگ سفید و کوچکترین عدد برای رنگ سیاه می‌باشد که با توجه به تایپ داده‌ها می‌توانند رنج مختلفی برای خود بگیرند
- `truecolor images (rgb images)` : این نوع تصاویر در یک ماتریس سه بعدی یعنی $m * n * 3$ ذخیره می‌شوند که به جای ذخیره کردن عدد در مرحله قبل یک ماتریس ۳ عضوی متشکل از شدت رنگ های `rgb` را در خود نگه می‌دارد.

۴.۰ گام چهارم

تصویر تبدیل شده در مرحله قبل را در نشان داده و آن را ذخیره می‌کنیم.

۱.۴.۰ بررسی کد

```

1 figure ;
2 imwrite(grayImg , 'GrayImage.bmp') ;
3 imshow(grayImg) ;

```



شکل ۲: تصویر خاکستری شده

۲.۴.۰ تفاوت فرمت های متفاوت عکس

- **jpg** : این نوع تصاویر از نوع فشرده سازی با اتلاف^۱ می باشند. در واقع حجم فایل را تا حد زیادی کاهش می دهد این نوع داده برای نگهداری و فرستادن مناسب است
- **png** : این نوع تصاویر از نوع فشرده سازی بدون اتلاف^۲ می باشند. در واقع این نوع فشرده سازی

^۱ lossy compression

^۲ lossless compression

مخالف فشرده‌سازی قبلی می‌باشد و می‌توان به بازسازی دوباره داده اصلی از فایل فشرده رسید.

- **bmp** : توسط شرکت Microsoft توسعه یافته شده است. حجم فایل بیشتری دارد و از نوع فشرده‌سازی بدون اتلاف می‌باشد.
- **tiff** : این نوع نسبت به نوع های قبل بیشترین حجم فایل را دارد و از نوع فشرده‌سازی بدون اتلاف می‌باشد و معمولاً در صنعت هنر و عکس های حرفه ای استفاده می‌شود

اگر بخواهیم از بین این فرمت ها یک فرمت را به عنوان فشرده‌سازی بدون اتلاف انتخاب کنیم کدام فرمت می‌باشد؟ **tiff**.

۵.۰ گام پنجم

با توجه به جستجوهای صورت گرفته تصویرها از نوع سیگنال توان نیستند اما از نوع سیگنال انرژی می‌باشند. در پی جستجوهای مختلف متوجه شدیم هر عضو ماتریس نشان دهنده انرژی آن می‌باشد که با جمع آن‌ها می‌توان به انرژی کل تصویر رسید. دو راه را برای این کار انجام دادیم یکی که طبق نکته ذکر شده بالا و دیگری طبق رابطه پارسوال اول تبدیل فوریه آن را محاسبه کرده و سپس انرژی این سیگنال را در حوزه فرکانس حساب می‌کنیم.

۱.۵.۰ بررسی کد

```

1
2 totalEnergy = sum(grayImg(:));
3 display(totalEnergy);
4
5 F = fft2(grayImg);
6 magImage = abs(F).^2;
7 energy = sum(magImage(:));
8 display(energy);

```

```
totalEnergy : 28313081, energy : 1027740529051912
```

۶.۰ گام ششم

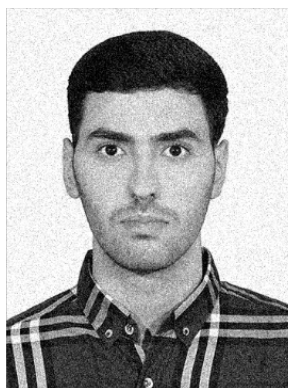
در این گام نیز با توجه به تابع از پیش تعریف شده `imnoise` که مقادیر پیش فرض مقادیر 0.01 برای واریانس و صفر برای میانگین در نظر گرفته شده است می توان به تصویر نویز اضافه کرد. و به عنوان ورودی دوم تابع نویز `gaussian` را به آن می دهیم و در ادامه نویز اضافه شده به تصویر را نمایش می دهیم.

۱.۶.۰ بررسی کد

```

1 figure ;
2 noisedPicture = imnoise(grayImg, 'gaussian');
3 imshow(noisedPicture);

```



شکل ۳: تصویر با نویز

۲.۶.۰ SNR(signal to noise ratio)

در این قسمت نوبت به محاسبه نسبت سیگنال به نویز اضافه شده است.

SNR : میزان قدرت یک سیگنال نسبت به نویز پس زمینه آن می باشد. که با واحد db اندازه گیری می شود.

$$SNR = 10 \cdot \log_{10} \frac{P_{signal}}{P_{noise}}$$

این نسبت می‌تواند هر عددی باشد که اعداد بزرگتر از صفر نشان دهنده این است که سطح سیگنال اصلی بیشتر از سطح noise است و اعداد کوچکتر برعکس. در واقع هر چه این نسبت بزرگتر باشد کیفیت سیگنال بهتر است.

۳.۶.۰ بررسی کد

```

1 x = im2double(grayImg);
2 y = im2double(abs(noisedPicture - grayImg));
3 signalNoiseRatio = snr(x,y);
4 display(signalNoiseRatio);

```

```
signalNoiseRatio = 14.59
```

۷.۰ گام هفتم

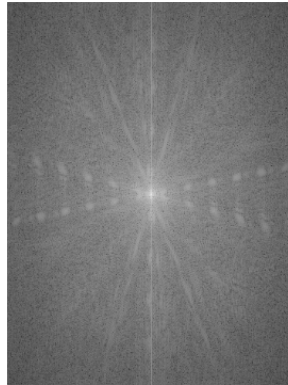
در این گام سعی به گرفتن تبدیل فوری از تصویر خاکستری شده و آن را طبق راهنمایی گفته شده رسم می‌کنیم.

۱.۷.۰ بررسی کد

```

1 figure;
2 ft = fftshift(log(abs(fft2(grayImg))));
3 imshow(ft, []);

```



شکل ۴: تصویر تبدیل فوریه گرفته شده

۲.۷.۰ تحلیل تبدیل فوریه گرفته شده

- هر پیکسل در عکس تبدیل فوریه نشان دهنده یک موج سینوسی دو بعدی با فرکانس وابسته به فاصله از مرکز می باشد.
- نتیجه تصویر نشان می دهد که عکس حاوی تمامی فرکانس ها می باشد اما بزرگی آن هر چه از مرکز تصویر دورتر می شویم کمتر می شود و در نتیجه فرکانس بیشتر می شود.
- فرکانس های کمتر حاوی اطلاعات بیشتری نسبت به فرکانس های بالاتر هستند

۳.۷.۰ پاسخ به سوال ها

۱. مرکز تصویر چرا از همه نقاط دیگر نورانی تر است؟ به دلیل اینکه بخش زیادی از عکس فرکانس کمتری دارند و مرکز تصویر حاوی اطلاعات بیشتری است. به تعبیری دیگر بخش زیادی از تصویر اصلی ما دارای تغییرات کم فرکانس هستند یعنی رنگ آنها به یکباره از سفید به سیاه تغییر نمی کند
۲. چرا هر چه از مرکز دورتر می شویم نقاط کم نورتر می شوند؟ چون این نقاط فرکانس بیشتر را در بر می گیرند در واقع این نقاط نشان دهنده تغییر ناگهانی در عکس اصلی می باشند.
۳. بالا و پایین ترین فرکانس در تصویر کدام نقاط است؟ مرکز دارای کمترین فرکانس و هر چه از مرکز دورتر می شویم فرکانس بیشتر می شود.

۸.۰ گام هشتم

از روش `median` استفاده می‌کنیم.

۱.۸.۰ بررسی کد

```

1 figure ;
2 Kmedian = medfilt2(noisedPicture);
3 imshow(Kmedian);

```



شکل ۵: تصویر با نویز رفع شده

۲.۸.۰ عملکرد رفع نویز

با توجه به راهنمایی گفته شده و تابع ازپیش تعریف شده `PSNR` به بررسی این تابع می‌پردازیم. هرچه مقدار خروجی این تابع بیشتر باشد ما بهتر عمل کرده‌ایم

range PSNR

- برای تصویر ها و ویدئو های فشرده شده بین ۳۰ تا ۵۰ دسی بل می‌باشد که ۸ بیتی است.

- برای ۱۲ بیتی خروجی از ۶۰ به بالا خوب است
- مقدارهای قابل قبول برای انتقال‌های بی‌سیم بین ۲۰ تا ۲۵ می‌باشد.

```
1 [peaksnr , outputSNR] = psnr(noisedPicture , grayImg);  
2 fprintf('\n The Peak-SNR value is %0.4f' , peaksnr );
```

```
The Peak-SNR value is 18.3259
```