

## توضیح تکلیف دوم

آریا ادیبی ۹۲۱۱۰۴۷۶

### نحوه‌ی قرارگیری فایل‌ها

فایل‌های مربوط به هر پرسش در پوشه‌ی با شماره‌ی همان پرسش قرار گرفته‌اند. فرض شده است که در هر کدام از این پوشه‌ها تصویرهای مورد نیاز اولیه قرار داده شده‌اند. همچنین در پرسش ۷ نیز باید تصویر حاصل از پرسش ۴ قرار بگیرد.

#### پرسش ۱:

یک فیلتر ۲ بعدی جدا پذیر است اگر و تنها اگر رتبه‌ی<sup>۱</sup> آن ۱ باشد. یعنی، هر سطر دلخواه را به سطر دلخواه دیگر می‌توان تنها با ضرب آن در یک عدد به دست آورد (یا به صورت هم ارز در مورد ستون‌ها). بنابراین این یک بررسی ساده برای جدا پذیری به دست آوردن شمار سطرها/ستون‌های مستقل خطی در ماتریس است. اگر این شمار ۱ بود آنگاه فیلتر ما جدایی پذیر است. بنابراین این یک روش ساده برای بررسی این موضوع می‌تواند اینگونه باشد:

۱. سطر اول را کنید.

۲. از ۲ به بعد را انتخاب کرده و روی هر یک بررسی کنید که:  
آیا می‌توان سطر اول را در عددی ضرب کرد که به سطر انتخاب شده تبدیل شود (تنها کافیس ضرب با یک عدد بررسی شود آن هم ضربی است که عضو اول سطر ۱ را به عضو اول سطر کنونی تبدیل کند).

#### پرسش ۲:

در این پرسش فیلتر استفاده شده فیلتر Sobel است که در زیر نیز نشان داده شده است. برای درست تشخیص دادن لبه‌ها همان طور که در کلاس نیز توضیح داده شد، ابتدا تصویر را با فیلتر `imgaussfilter` هموار می‌کنم. تمام مرحله‌ها دقیقاً همان گونه است که در صورت پرسش ذکر شده است. برای به دست آوردن زاویه نیز به سادگی با داشتن مقدار و جهت در هر بعد می‌توان از تابع `arctan` استفاده کرد و این مقدار را به دست آورد.

---

<sup>۱</sup>rank

1	2	1
0	0	0
-1	-2	-1

1	0	-1
2	0	-2
1	0	-1

(a) Sobel-Horizontal

(b) Sobel-Vertical

### پرسش ۳:

فیلتر لاپلاس این پرسش، فیلتریست که `fspecial('laplacian')` می‌دهد که این فیلتر در زیر نشان داده شده است.

0.1667	0.6667	0.1667
0.6667	-3.3333	0.6667
0.1667	0.6667	0.1667

این فیلتر روی عکس زده می‌شود و مقدارهای به دست آمده به تصویر اصلی جمع می‌شود که مرزها بیشتر خود را نشان دهد یا به عبارتی تصویر sharp شود.

### پرسش ۴:

در این پرسش تقریباً تمام کارها توسط تابع `imhist`<sup>۲</sup> و `histeq`<sup>۳</sup> انجام می‌شود که توضیح‌های این ۲ در پاورقی آورده شده است.

به سادگی Histogram مقصد را با `imhist` گرفته و با `histeq`، Histogram مبدأ را به آن تقریب می‌زنیم.

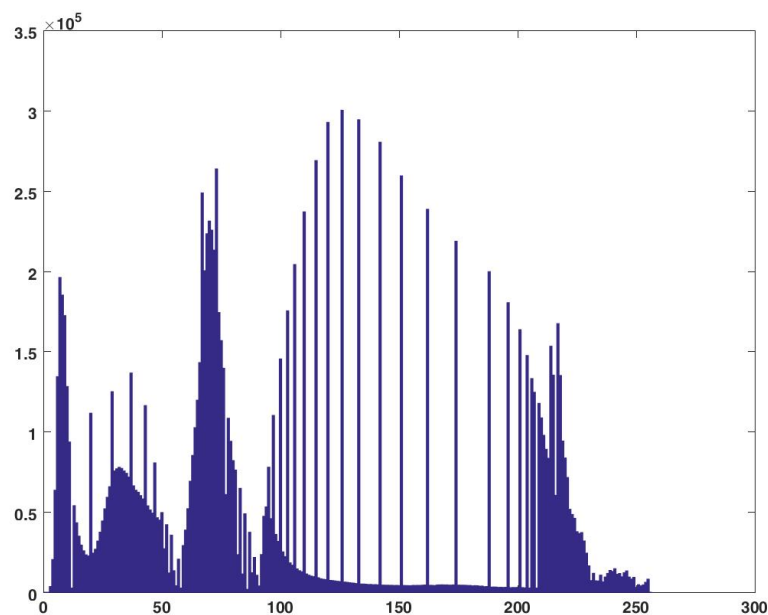
تفاوت در Histogram قبل و بعد از تبدیل هم در نمودار زیر نشان داده شده است. برای از بین بردن noise ها نیز از فیلتر میانه‌ی خود مطلب یعنی `medfilt2` استفاده شده است.

۲

`imhist(I)` calculates the histogram for the intensity image I and displays a plot of the histogram. The number of bins in the histogram is determined by the image type.

۳

`J = histeq(I,hgram)` transforms the intensity image I so that the histogram of the output intensity image J with `length(hgram)` bins approximately matches hgram. The vector hgram should contain integer counts for equally spaced bins with intensity values in the appropriate range: [0, 1] for images of class double, [0, 255] for images of class uint8, and [0, 65535] for images of class uint16. `histeq` automatically scales hgram so that `sum(hgram) = prod(size(I))`. The histogram of J will better match hgram when `length(hgram)` is much smaller than the number of discrete levels in I.



## پرسش ۵:

در این پرسش تصویر به فضای رنگی hsv برده می‌شود. در این پرسش به صورت کاملاً تجربی عمل شده و با استفاده از mask رنگ مورد نظر در این فضا انتخاب شده و به رنگ دلخواه برده می‌شود. همان طور که گفته شده عددها به صورت تجربی به دست آمده‌اند.

## پرسش ۶:

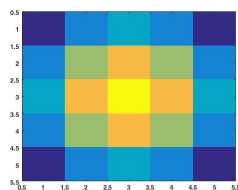
در این پرسش به سادگی تابع `integralImage` تصویر انتگرال را به دست می‌آورد.

## پرسش ۷:

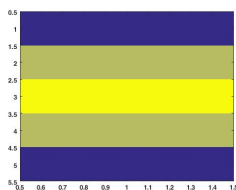
فیلتر گوس استفاده شده توسط تابع `fspecial('gaussian')` گرفته شده و توسط تابع `isfilterseparable` از هم جدا شده است. این فیلترها را توسط تابع `imagesc` نشان داده شده‌اند که نتیجه در این جا هم گذاشته شده است.

• اگر مایل به دیدن مقدار عددی آن‌ها هستید به سادگی می‌توان در کد به جای `imagesc` از `disp` استفاده کرد که مقدارها را در `terminal` نشان می‌دهد.

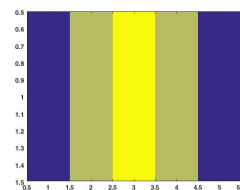
برای اعمال این فیلترها هم به سادگی با ۲ حلقه‌ی تو در تو و با استفاده از تابع `imfilter` برای تصویر `CrossCorrelation` را پیاده سازی کردم. از این جا به بعد می‌توان گفت که همانند دستورهای پرسش عمل شده است، تنها نتیجه پرسش این قرار است.



(a) Gaussian Filter  
2D



(b) Gaussian Filter  
Row



(c) Gaussian Filter  
Column

#### ● با پیاده‌سازی خودم:

همانطور که انتظار می‌رفت زمان فیلترهای جدا بسیار کمتر است. اگر اندازه‌ی فیلتر ما  $m \times n$  باشد و اندازه‌ی تصویر اصلی  $a \times b$  انتظار داریم که زمان بدون جدا کردن از  $O(abmn)$  باشد در صورتی که با جدا کردن  $O(ab(m+n) + mn)$  است که برای اعداد به اندازه‌ی کافی بزرگ به مراتب کوچک‌تر است،  $mn$  نیز برای به دست آوردن ۲ ماتریس جدا شده است که البته به نظر از این سریع‌تر هم الگوریتم دارد.

#### ● با imfilter:

می‌توان دید که این‌جا فیلتر جدا شده زمان بیشتری می‌گیرد. این موضوع چند علت دارد ولی علتی که من اینجا به آن اشاره می‌کنم این است که خود تابع جداپذیری را بررسی می‌کنید.

در مورد تصویر نهایی نیز چه جدا کنیم و چه نکنیم تصویر یکی است.

## پرسش ۸:

در این پرسش پیاده سازی فیلتر bilateral از اینترنت گرفته شده است با نام bfilter2 که مولف آن، به صورت کامل توضیحات را آورده است.

تصویر انتخاب شده تصویر دختر بچه‌ای است که سایه روشن روی صورت و موی نقش بزرگی دارد. همچنین تارهای جدا شده‌ی مو قابل توجه است. به سادگی می‌توان دید که فیلتر گوس معمولی تصویر را مات می‌کند ولی با استفاده از bilateral به همان میزان تصویر هموار می‌شود ولی این سایه روشن‌ها و تارهای مو در نظر گرفته می‌شود و تصویر به مراتب بهتری به ما می‌دهد همزمان با این که هدف ما یعنی هموار سازی را نیز انجام می‌دهد.