

# توضیح تکلیف چهارم

آریا ادبی ۹۲۱۱۰۴۷۶

## نحوهی قرارگیری فایل‌ها

فایل‌های مربوط به هر پرسش در پوششی با شماره‌ی همان پرسش قرار گرفته‌اند. فرض شده است که تمامی فایل‌های لازم که شما در اختیار ما قرار دادید در پوششی inputs قرار دارد. متأسفانه به دلیل محدودیت اینترنت امکان قرار دادن آن‌ها توسط من نیست.

### پرسش ۱:



Figure 1: SVD - Compression - Resulted Image

نتیجه‌ی کار بعد کارهای انجام شده بالا آمده است.  
برای این کار  $m = 260$  قرار داده شده است. در ادامه اطلاعات خواسته شده آمده است. کلاً نحوه‌ی محاسبه‌ی هر کدام از اطلاعات به روشنی در کد قابل دیدن است.  
دقت کنید که آنتروپی برای هر سه لایه (و البته به علاوه میانگین آن‌ها) محاسبه شده است.  
دقت کنید که مجموع مقدار منفردها برای حالت فشرده‌سازی شده و حالت نافشره محاسبه شده است و مقدار حذفی به سادگی به دست می‌آید. اثبات می‌شود تقسیم این ۲ عدد به دست آمده میزان فشرده‌سازی را می‌گوید. ماتریس‌های سازنده نیز توسط  $imshow(\text{abs}(\log(M)))$  نمایش داده شده‌اند.  
کاری که توسط کد انجام می‌شود به راحتی قابل رویت است با استفاده از تابع Matlab ماتریس‌ها به دست می‌اید و بعد قسمتی از آن‌ها برداشته شده و دوباره در هم ضرب می‌شوند.

```

"Space saving of channel R"= 8.936083e-01
"Space saving of channel G"= 8.936083e-01
"Space saving of channel B"= 8.936083e-01
Mean "space saving"= 8.936083e-01

Entropy in original image:
Entropy R= 7.724430e+00
Entropy G= 7.681424e+00
Entropy B= 6.861159e+00
Mean Entropies= 7.422338e+00

Entropy in reduced image:
Entropy R= 7.732121e+00
Entropy G= 7.727176e+00
Entropy B= 7.044538e+00
Mean Entropies= 7.501278e+00

Sum of "Singular Values":
The sum in original image, channel R= 1.006888e+04
The sum in reduced image, channel R= 6.283779e+03

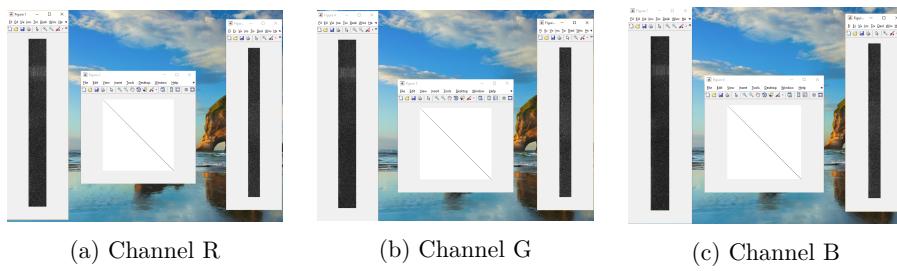
The sum in original image, channel G= 1.012856e+04
The sum in reduced image, channel G= 6.355981e+03

The sum in original image, channel B= 9.901532e+03
The sum in reduced image, channel B= 6.232602e+03

The sum in original image= 3.009898e+04
The sum in reduced image= 1.887236e+04

```

Figure 2: SVD - Compression - Informations



## پرسش ۲:

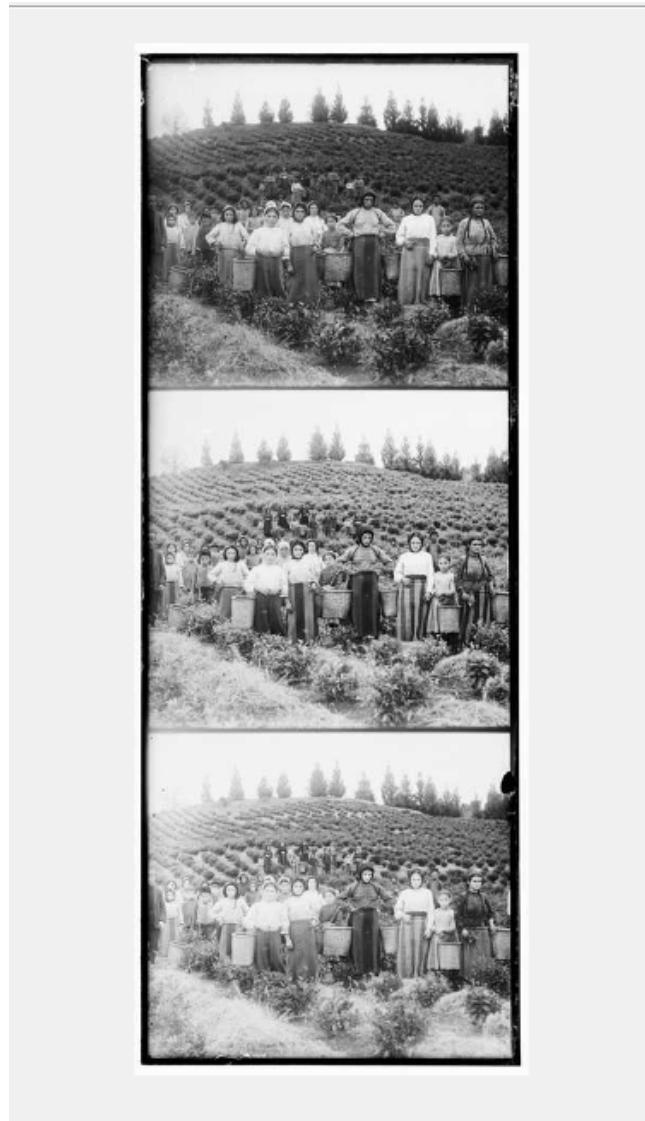


Figure 4: Compressed Image

در این پرسش همانند عمل کرد jpeg در حالت کلی عمل شده است. تنها تفاوت اساسی این است که در آن جا مقادیر ، بعد از quantization در نظر گرفته نمی شد ولی باید یک ۲ تایی کد می شد که جای هر مقدار غیر ، نیز مشخص شود. من ، ها را نیز مستقیم کد می کنم. با این که این عمل کمی ناکاراتر است ولی هنوز کاراست چون انتظار شمار زیادی . است که هافمن از این خاصیت خوب استفاده مناسبی می کند.

به طور خلاصه این کارها انجام می‌شود. به  $8^8$  ها افزای می‌شود. سپس هر کدام DCT اش گرفته می‌شود و بعد با ضرایب متفاوت quantize می‌کنیم. بعد از آن به صورت زیگزاگ جدول را ۱ بعدی می‌کنیم. پس از اتمام این کار برای همه  $8^8$  ها با هافمن کد می‌کنم.

نتیجه‌ی کار را می‌بینید گذاشته شده است.

طرز کار کد روشن است. کد به بخش‌ها تقسیم شده و وظیفه‌ی هر بخش نشان داده شده است. کد نوشته شده ساده است. نیاز خاصی با توجه به کامنت‌های گذاشته شده در خود کد ندارد. با استفاده از whos در مطلب به راحتی می‌توان فهمید که تصویر ایجاد شده  $35645400 \approx 0.499$  بایت جا می‌گیرد در حالی که تصویر اصلی  $71126096 \approx 0.499$  بایت جا می‌گرفت از این رو مقدار  $\frac{35645400}{71126096}$  معیار ذخیره سازی حافظه‌ی ماست. یعنی مقدار نصف.

## پرسش ۵:

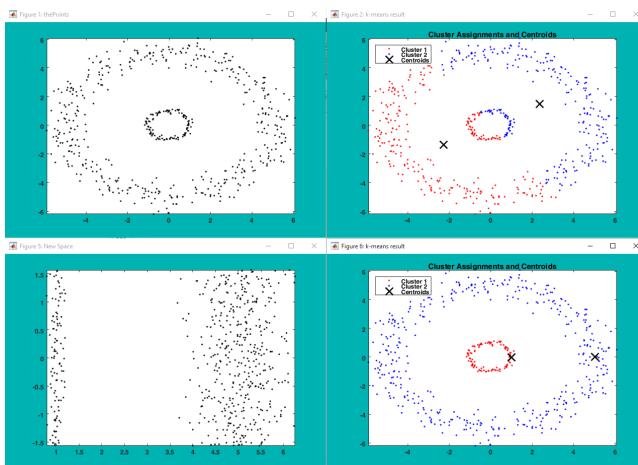


Figure 5: Feature Spaces and corresponding KMeans Clusters

در عکسی که بالا ملاحظه می‌کنید سمت چپ نقاط ما در فضای خاصیت‌هایی که گرفتیم نشان داده شده‌اند و سمت راست با استفاده از kmeans آن‌ها را دسته بندی کردیم. فضای اول همان فضای  $x$  و  $y$  است و فضای دوم  $R$  و  $\theta$  که همان مختصات قطبی است. همان طور که می‌بینید در فضای دکارتی جوابی که انتظار داریم را دریافت نمی‌کنیم. دلیل این موضوع روشن است چرا که فضاهایی که kmeans برای دسته‌هایی مشخص می‌کند کروی حالت (می‌تواند کشیده هم شود) است و هیچ نوع فضای دیگری به دسته‌هایی نمی‌تواند بدهد. از این رو دایره‌ی بیرونی چون یک دایره درون دارد در چنین فضاهایی نمی‌تواند توصیف شود. ولی در فضای قطبی همان طور که مشخص است این ۲ را با ۲ بیضی یکی برای قسمت چپ و دیگری برای قسمت راست می‌توان تقسیم کرد.

الگوریتم mean-shift اینگونه کار می‌کند که شماری پنجره‌ی جستجو می‌گیرد و این پنجره‌ها شروع می‌کنند در فضای خاصیت‌ها حرکت کردن تا همگرا شوند و حرکتی نکنند و یا میزانش هر بار بسیار کم باشد. این الگوریتم mode ها (peak - local-maxima) را پیدا می‌کند و همه‌ی نقاطی که یک پنجره از آن عبور کرده بود به همان mode ای نسبت داده می‌شوند که پنجره به آن همگرا شده است. دقت

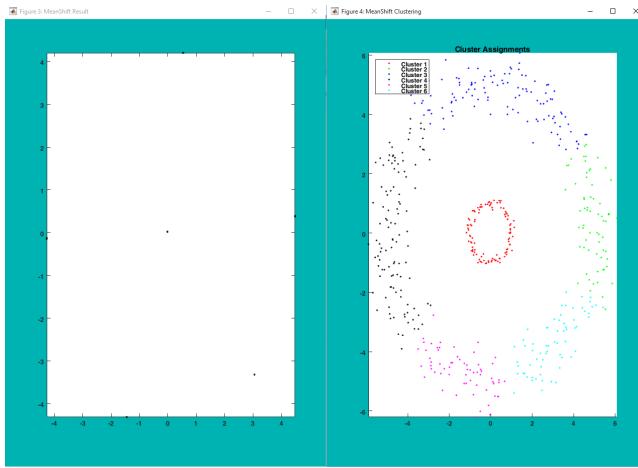


Figure 6: peaks(local maxima) and Mean Shift Clusters

کنید که این الگوریتم برخلاف kmeans خود در بارهی شمار دسته‌ها تصمیم می‌گیرد. البته این این تصمیم با دادن پارامترهای مختلف مثلاً `bandWith` مختلف احتمالاً متفاوت خواهد بود.  
با پارامترهای من همان طور که از تصویر نیز مشخص است نقاط ۶ دسته شده‌اند (در فضای دکارتی).  
منطق آن طبق حرف‌های زده شده و منطق `peak` که به سمتی متمايل است که نقاط بيشتری باشد روشن است.

دز اين جا نيز به خواسته‌ي خود نرسيديم ولی اگر فضا را قطبي کنيم درست می‌شود چون پرسش اين موضوع را نخواسته است من هم لازم ندانستم انجام دهم.  
تصویر سمت چپ mode ها را نشان مي‌دهد.

اين پرسش نيز به خاطر کامنت‌هایي که در کد هست کد بسیار ساده‌ای دارد اما ۲ نکته را خوب است توضیح دهم. من يك کپی از نقاط دارم پس با این کار حرکت پنجره‌ها را با حرکت نقطه‌ها متناظر می‌کنم  
دقت کنید تصمیم گیری بر اساس آن نقاطی است که کپی شده و ثابت‌اند از این رو این پیاده سازی اول شاید گیج کننده باشد. برای برچسب دهی به نقاط هم اینگونه عمل می‌کنیم: الگوریتم من به هر نقطه ۱ mode نسبت داده است کافیست هر نقطه‌ای که تا حالا برچسب نخورده را بگیرم و تمام چنین نقاط دیگری را با این نقطه مقایسه کنم اگر از میزانی به mode آن در نسبت دهی نزدیکتر بودند آنها دسته کنید و يك برچسب دهید. اين کار در `for` پس از الگوریتم اجرا می‌شود. چون من چک کردم و دیدم ۶ دسته است ۶ دسته را در `plot` آوردم. بقیه کد روشن است.

## پرسش ۶:

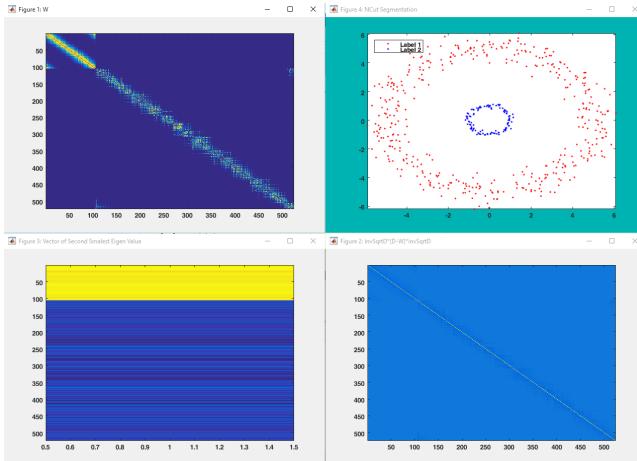


Figure 7: Requested Information

در این پرسش از NCut استفاده شده است. اولین کار در NCut تعریف یالهای گراف است اول از همه چون شمار نقطه‌ها زیاد نبود من گرافی کامل تشکیل دادم که وزن آن همانند آن چه در جزووهای درس بود از رابطه‌ی

$$e^{-\frac{dis^2}{2 \times \sigma^2}}$$

به دست می‌آید که در اینجا من dis را همان فاصله‌ی اقلیدسی گرفتم. ماتریس وزن‌ها عکس بالا سمت چپ است.

همانند راه حل ارائه شده عمل می‌کنم و ۲ ماتریس خواسته شده را به دست می‌آورم این ۲ ماتریس ۲ عکس پایین هستند. سپس با استفاده از بردار ویژه و یک عدد به عنوان جدا کننده تصویر را به ۲ قسمت تقسیم می‌کنم. تعیین این جدا کننده راههای مختلف دارد یکی از ساده ترین آن‌ها به علت شهود اثبات قرار دادن است که من نیز همین کار را کردم.

با این توضیحات و کامنت‌های کد، کد روشن است و نیازی به توضیح ندارد.

## پرسش ۸:

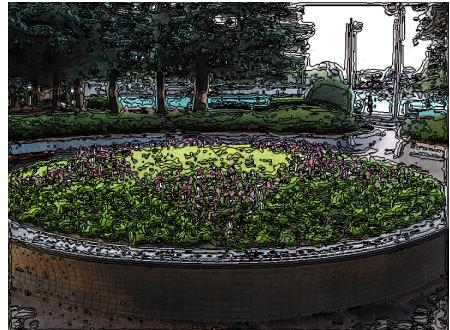
نکته: در این پرسش چون صورت پرسش هیچ روندی رو اجبار نکرده قرار داد می‌کنم که هر عکسی را که می‌خواهید ورودی دهید در پوشه‌ی پرسش با نام theImage.jpg بهگذارید.  
راه حل این پرسش ساده است. در همان سلایدهای شما یک منبع برای بانک فیلتر وجود داشت از همان جا یک بانک دانلود گرفتم و از آن استفاده کردم.

برای بهبود سرعت ابتدا عکس کوچک می‌شود و در انتهای به اندازه‌ی اصلی بر می‌گردد. فضای خاصیت ما به سادگی مقدار convolution هر فیلتر به مرکز آن pixel متناظر است. یعنی برای هر pixel به تعداد فیلترها مقدار داریم.

در انتهای روی این فضا kmeans می‌زنیم تا دسته بندی کنیم. حال هر pixel یک برچسب دارد. برای نشان دادن نتیجه راههای مختلفی هست. راهی که من انتخاب کردم این بود که فیلتر [۱، ۰، -۱] را به



(a) Image 1



(b) Image 2

صورت عمودی و افقی بزنید و هر کجا مقدار مخالف ۰ بود یعنی آنجا میان ۲ برچسب است آن جا را سیاه کنید. به این صورت دور دستهها خط کشیدیم.  
کد زده شده با توجه به کامنت‌ها و توضیحات بالا ساده است و نیازی به توضیح ندارد.