

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

موضوع ارایه: تحلیل باینری تصویر (فصل ۶)

استاد:

آقای دکتر

محلوجی

تهیه کننده:

لیلا زارع

- (۱) مقدمه ای بر تصاویر باینری
 - (۲) Label زدن شی و شمارش آن
 - (۳) آموزش تصاویر باینری در متلب
- خواندن تصویر
- تخمین پس زمینه
- مشاهده و نمایش تقریب پس زمینه به صورت یک سطح
- تفریق پس زمینه از تصویر اصلی
- افزایش کنتراست تصویر
- آستانه گیری
- شناسایی اشیا در تصویر
- بررسی یک شی
- مشاهده تمام اشیا
- محاسبه مساحت اشیا

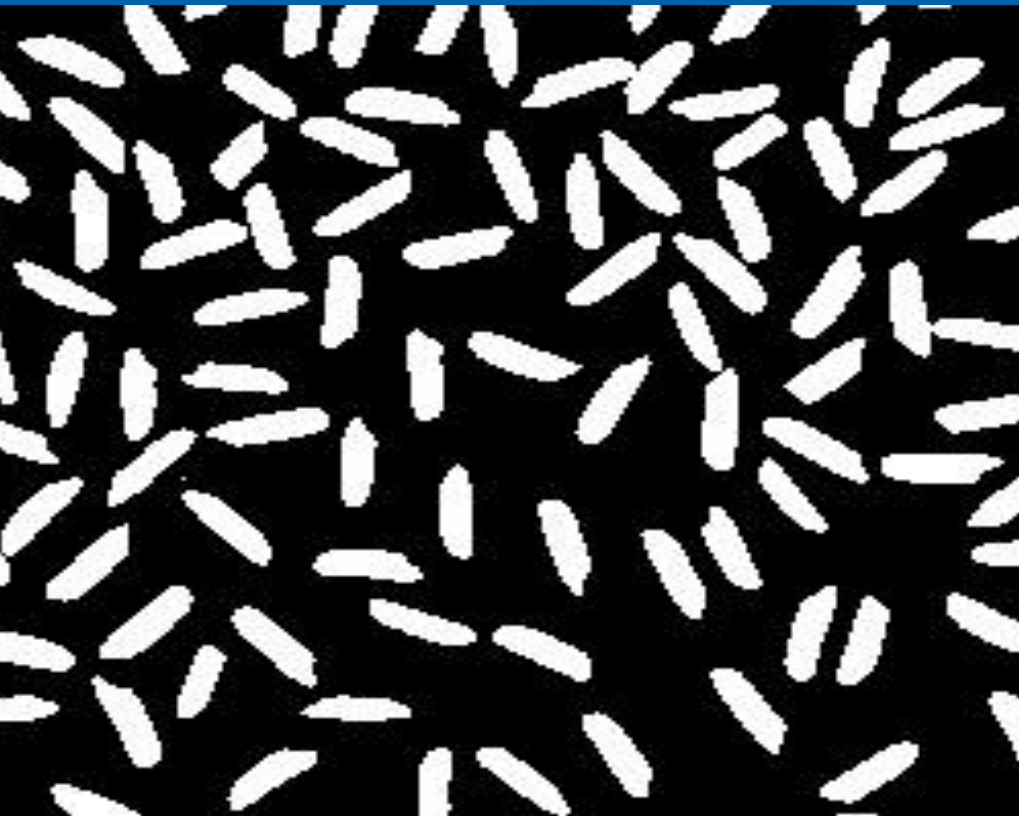
مقدمه :

در طول چند دهه گذشته، تجزیه و تحلیل تصاویر دو بعدی بسیار دارای اهمیت می باشند چرا که از آن بطورمنحصر به فردی جهت توصیف بسیاری از انواع شی، از کلید تا نهادها، از واشر تا مهره ، و از اثر انگشت تا کروموزوم را تحت پوشش قرار میدهد .

بررسی و تحلیل بافت یا باینری، زمینه مطالعاتی مهم و سودمندی در روند پردازش تصویر و بینایی ماشین می باشد. در واقع می توان گفت که بافت هر شی سبب تمایز اشیا از همدیگر می شود. پردازش تصاویر ماهواره ای و پزشکی، دریافت از راه دور، کنترل کیفیت اتوماتیک و پردازش اسناد، تنهاتعدادی از کاربردهای پردازش بافت تصویر است.

تصویر باینری (سیاه و سفید) :

به تصویری گفته می شود که تمام نقاط آن یا سفید باشند و یا سیاه که معمولا رنگ سیاه با عدد ۰ و رنگ سفید با عدد ۱ نشان داده می شود.



رنگ سیاه یا ۰ به قسمتی از تصویر اطلاق می شود که برای ما اهمیتی ندارد و زمینه تصویر به حساب می آید. همچنین رنگ سفید یا ۱ قسمتی از تصویر است که برای ما با اهمیت است.





```
y=imread('c:\3.jpg');  
imshow(y)  
figure  
y1=rgb2gray(y);  
imshow(y1)
```



این بخش در واقع با این فرض آغاز شده است که اشیاء در یک تقسیم بندی بنیادی از آستانه تا زمینه قرار گرفته اند و بر این اساس باید نسبت به بررسی آنها اقدام گردد.



این امر موجب سهولت در مشخص شدن مرزهای بین اشیاء در تصاویر باینری می‌گردد.

لذا پیش زمینه ۸ بعد اتصال شونده و پس زمینه ۴ بعد اتصال شونده را بمنظور اتصال مورد استفاده قرار میدهد.

۶.۳. شمارش و نشانه گذاری شیء:

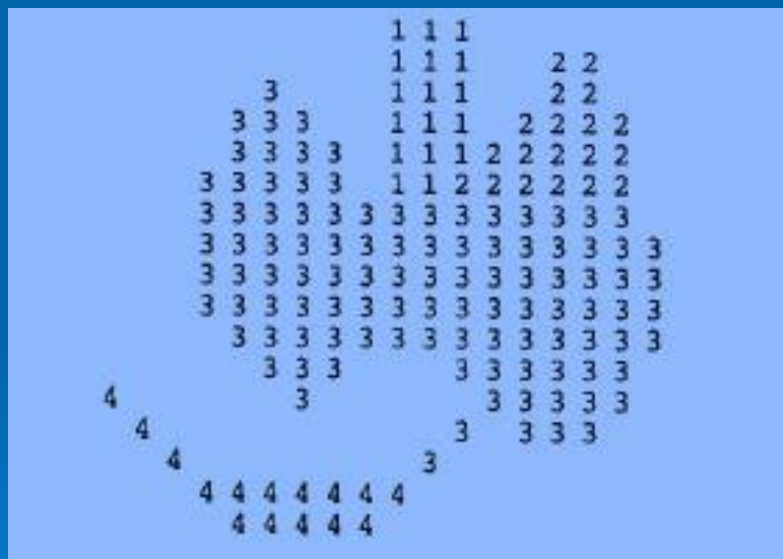
نشانه گذاری ممکن است توسط اسکن پی در پی تصویر انجام پذیرد تا زمانی که نقاط ۱ نما بر جسم اول ایجاد و مشخص گردند.

با استفاده از فضای موجود در تصویر اصلی یک فضای تصویر جداگانه برای نشانه گذاری ایجاد می گردد. سپس اسکن از سر گرفته میشود البته باید نقاطی که قبلاً نشانه گذاری شده است نادیده گرفته شود. این روش ادامه داشته تا کل تصویر اسکن شده و تمام اشیاء نشانه گذاری شوند.

در این روش شمارش شیء و نشانه گذاری معمولاً نیاز به حداقل $2N+1$ عبور در فضای تصویر دارد، و در عمل از لحاظ تعداد نزدیک تر به $NW/2$ که در آن W عرض متوسط از اشیاء است خواهد بود. از این رو، این الگوریتم ذاتاً ناکارآمد است.

یک احتمال در کاهش پیمودن تصویر وجود دارد (تا در محاسبات بتوان خلاصه وار عمل نموده و به نوعی دیگر صرفه جوئی نمود) بدین گونه که ابتدا نشانه گذاریها آماده شود و سپس عملیات قبلی اجرا شود.

البته یکی از مهمترین مشکلاتی که در این حالت رخ خواهد داد پیرامون اشکال U شکل است که نمیتوان براحتی این احتمالات را پیاده سازی نمود.



✓ در این راستا ابزاری دیگر وجود دارد و آن هم استفاده از تصویر برداری به شیوه تلاقی میباشد که میتواند مثر ثمر واقع شود.

کاربرد تحلیل باینری تصویر در متناب

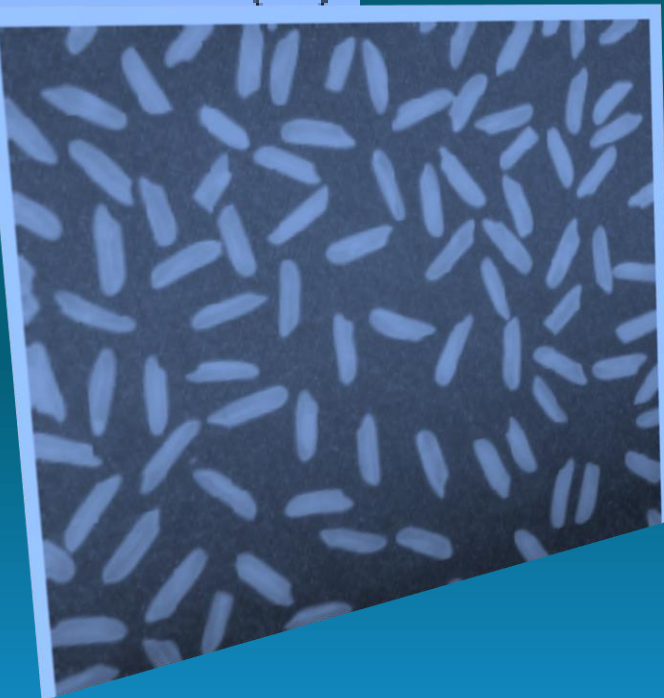
۱- خواندن تصویر

تصویر سطح خاکستری rice.png را خوانده و در یک آرایه به نام ذخیره می کند.

```
I = imread('rice.png');  
imshow(I)
```

حال آن را نمایش دهید:

نتیجه:



✓ ملاحظه میکنید که پس زمینه تصویر یکنواخت نیست بلکه در قسمتهای مرکزی روشنتر و در قسمتهای پایینی تیره تر است. عدم یکنواختی پس زمینه، اغلب باعث دشوارتر شدن کار جدا کردن اشیاء از تصویر می شود.

۲- تخمین پس زمینه:

از عملگر مورفولوژی بازکردن برای حذف دانه های برنج استفاده می کنیم تا بتوانیم به پس زمینه دست پیدا کنیم. اثر نهایی عملگر بازکردن این است که اشیایی را که نمی توانند ماسک را در بر بگیرند، حذف میکند.

برای بازکردن از دستور `imopen` به صورت زیر استفاده می کنیم:

```
background = imopen(I, strel('disk', 15));
```

- ✓ آرگومان اول، تصویر مورد نظرمان را باز می کند.
- ✓ آرگومان دوم ماسک مورد استفاده برای انجام عمل بازکردن است. (در اینجا برای تشکیل ماسک از دستور `strel` استفاده شده است.)



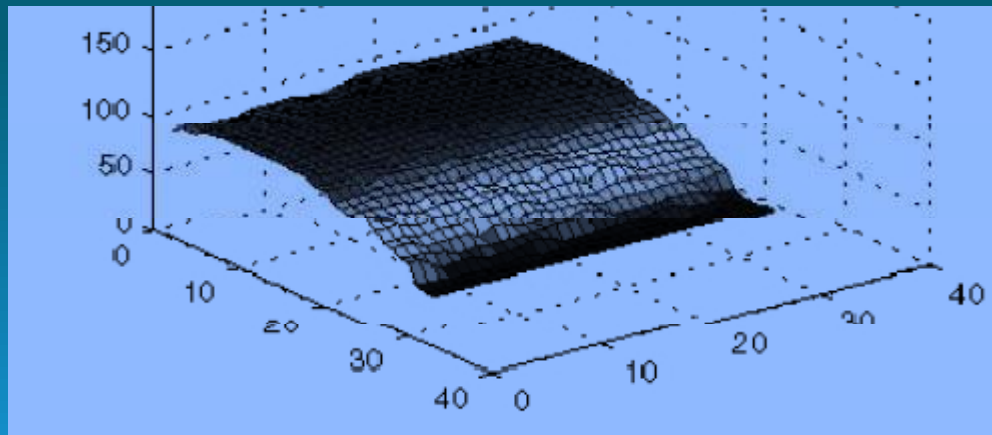
اگر تقریب فوق را به کمک دستور `imshow` نمایش دهیم نتیجه چنین خواهد بود:



۳-مشاهده (و نمایش) تقریب پس زمینه به صورت يك سطح :

برای نمایش پس زمینه به صورت يك سطح از دستور `surf` استفاده میکنیم؛ اما این دستور نیاز به آرگومان ورودی از نوع `double` دارد در حالیکه تصویر موجود در آرایه `background` از نوع `uint8` است. بنابراین، در حین استفاده از دستور `surf` کار تبدیل نوع را هم انجام می دهیم:

```
figure, surf(double(background(1:8:end,1:8:end))),zlim([0 255]);  
set(gca,'ydir','reverse');
```



۴-تفریق پس زمینه از تصویر اصلی :

برای ایجاد یک تصویر با پس زمینه یکنواخت، می توان تقریب بدست آمده از پس زمینه را از تصویر اصلی کم کرد و حاصل را نمایش داد:

```
I2 = I - background;  
figure, imshow(I2)
```

نتیجه :



Image with Uniform Background

۵-افزایش کانتراست تصویر:

تصویری که تا بحال بدست آوردیم پس زمینه تقریباً یکنواختی دارد اما تا حدی تاریک است و باید کانتراست آن را افزایش دهیم. در اینجا می‌خواهیم از دستور `imadjust` برای انجام این کار استفاده کنیم.

```
I3 = imadjust(I2);  
figure, imshow(I3);
```



نتیجه:

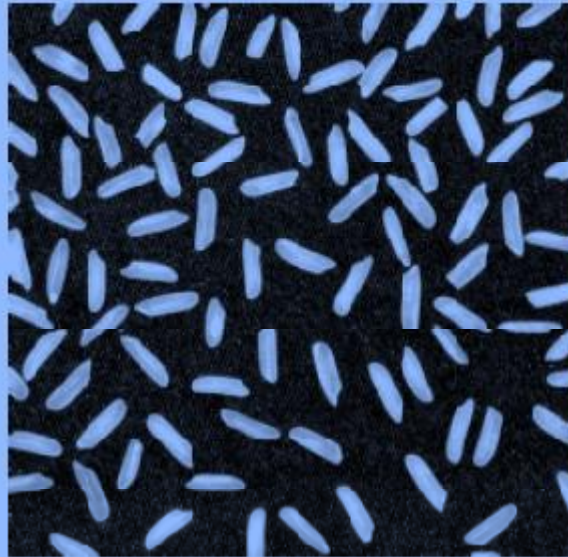


Image After Intensity Adjustment

۶- آستانه گيري تصوير:

براي تبديل يك تصوير سطح خاکستري به تصوير باينري از دستور `im2bw` ميتوان استفاده کرد. اين دستور نياز به مقدار يك آستانه دارد. براي انتخاب مناسب مقدار آستانه، ميتوانيم از دستور `graythresh` استفاده كنيم:

```
level = graythresh(I3);  
bw = im2bw(I3,level);
```

✓ براي حذف نويز، ميتوانيم از دستور `bwareaopen` استفاده كنيم. ايده ي اساسي اين است كه نويز عموماً شامل نقاط با مساحت كم است. بنابر اين ميتوان هر شيء با مساحتي كمتر از يك حد را نويز دانست.

بنابراین به کمک عملگر مورفولوژی بازکردن میتوان تمام اشیایی که مساحتشان کمتر از یک مقدار مشخص (در اینجا ۵۰) باشد را حذف کرد.

```
bw = bwareaopen(bw, 50);  
figure, imshow(bw)
```



Binary Version of the Image

- ✓ مقدار ۵۰ در حقیقت تابعی از رزولوشن ۱۲ (یا درجه تفکیک تصویر) است.
- ✓ هر چه رزولوشن بیشتر باشد، باید مقدار این آستانه را نیز بزرگتر در نظر گرفت.

۷-شناسایی اشیاء در تصویر:

برای استخراج اجزاء به هم پیوسته از یک تصویر باینری میتوان از هر یک از دستورات `bwconncomp` - `bwlabeln` - `bwlabel` استفاده کرد .
✓ جدیدترین آنها دستور `bwconncomp` است که حافظه ی کمتری مصرف کرده و گاهی سریعتر نیز می باشد.

```
cc = bwconncomp(bw, 4)
cc.NumObjects
```

نتیجه:

```
cc =  
  
Connectivity: 4  
ImageSize: [256 256]  
NumObjects: 95  
PixelIdxList: {1x95 cell}
```

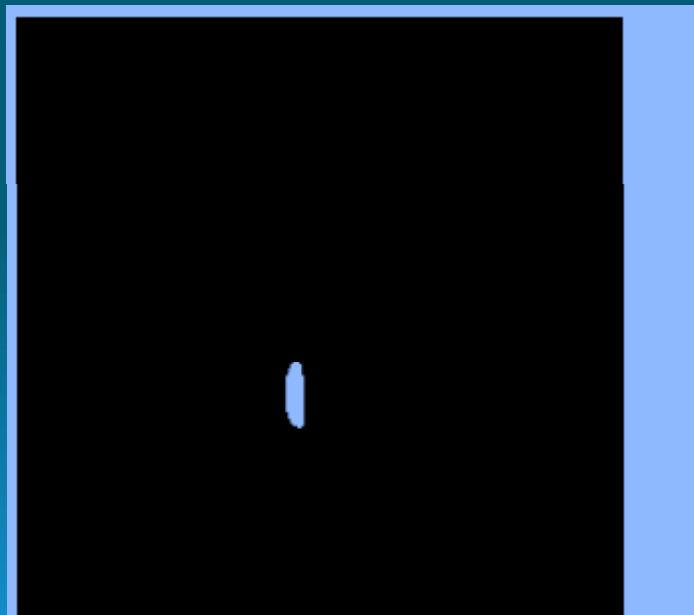
```
ans =  
  
95
```

۸- بررسی یک شی :

نشان دادن دانه برنج شماره ی ۵۰ :

```
grain = false(size(bw));  
grain(cc.PixelIdxList{50}) = true;  
figure, imshow(grain);
```

نتیجه:



The 50th Connected Component

۹-مشاهده ي تمام اشياء:

ابتدا از دستور `labelmatrix` براي تهيه ماتريس برچسب استفاده ميکنيم:

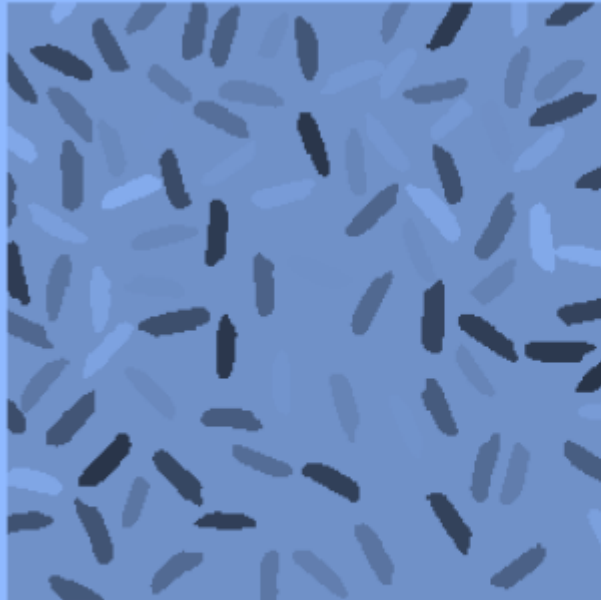
```
labeled = labelmatrix(cc);  
whos labeled
```

Name	Size	Bytes	Class	Attributes
labeled	256x256	65536	uint8	

حال براي اينکه هر دانه برنج را با يك رنگ نشان دهيم تا بتوانيم تصويري رنگي مشاهده کنيم، ماتريس به دست آمده را به کمک دستور `label2rgb` به يك تصوير رنگي تبديل کرده و آن را نمايش مي دهيم:

```
RGB_label = label2rgb(labeled, @spring, 'c', 'shuffle');  
figure, imshow(RGB_label)
```

نتيجه:



Label Matrix Displayed as Pseudocolor Image

Label Matrix Displayed as Pseudocolor Image



۱۰- محاسبه مساحت اشیاء مختلف:

هر دانه برنج يك جزء به هم پیوسته است. به کمک دستور `regionprops` اطلاعاتي را میتوان در مورد هر دانه برنج کسب کرد:

```
graindata = regionprops(cc, 'basic')
```

نتیجه:

```
graindata =  
  
95x1 struct array with fields:  
    Area  
    Centroid  
    BoundingBox
```

براي دسترسي به مساحت برنج ۵۰ أم :

```
graindata(50).Area
```

```
ans =
```

```
194
```

```
194
```

```
ans =
```

```
graindata(50).Area
```

با تشکر از نگاه گرمتان

