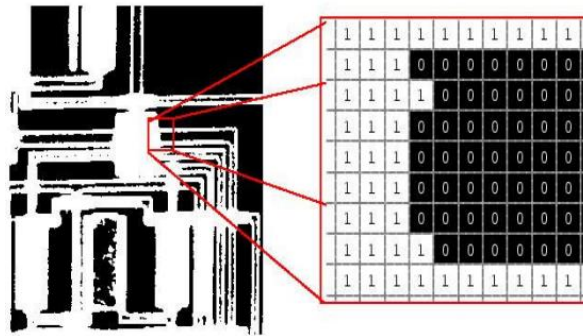


محاضرات العملي لمادة معالجة الصور قسم علوم الحاسبات كلية التربية

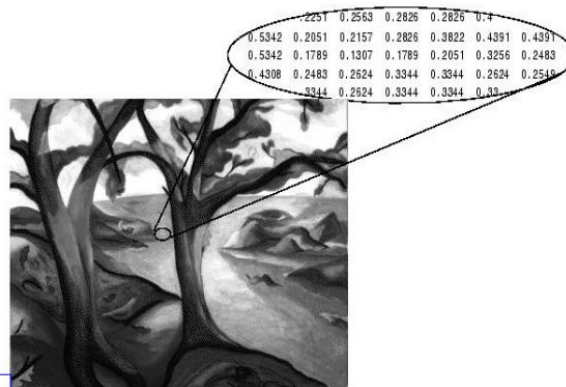
م.م رافد عيدان حليوت

. أنواع الصورة الرقمية في الماتلاب:

- **Binary Image** هي الصورة التي تمثل بمصفوفة بعدها $M \times N$ وهي تحوي اللونين الأسود والأبيض فقط. اذ يأخذ كل بكسل قيمة تمثل لون فالأسود يأخذ قيمة 0 بينما الأبيض يأخذ قيمة 1. (لا يوجد لون رمادي في هذا النوع من الصور).



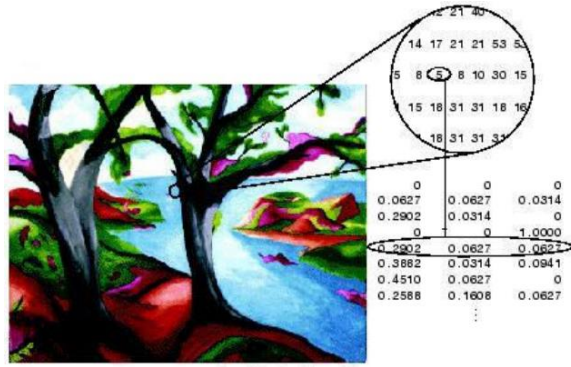
Grayscale Image - هي صورة تمثل بمصفوفة ثنائية البعد $M \times N$ وعناصرها من النوع double. تقع بين المجال $[0, 1]$ اذ 0 يمثل الأسود و 1 الأبيض وما بينهما درجات اللون الرمادي. كل عنصر في المصفوفة يمثل درجة لون لبكسل واحد ابيض او اسود او ما بينهما.



TrueColor RGB Image - تمثل بمصفوفة ثلاثية البعد $M \times N \times 3$ عناصرها من نوع double او unit8، فكل بكسل في الصورة ينتج عن دمج ثلاثة مركبات لإعطاء اللون المناسب فالمركبة الحمراء مكونة من مصفوفة ثنائية (تستخرج من المصفوفة ثلاثية البعد) $M \times N$ يكون فيها قيمة اللون بين المجال $[0, 1]$ الأسود و الأحمر و 1 وما بينهما هو درجات اللون الأحمر، وكذلك بالنسبة للأزرق والأخضر. ودمج ثلاثة الألوان RGB من ثلاثة مصفوفات ثنائية البعد نحصل على الألوان الحقيقية.



index Image - الصور المفهرسة تحتوي الصور المفهرسة مصفوفة من البيانات x وخارطة اللون للمصفوفة map، بيانات المصفوفة ممكن ان تكون من نوع 8-bit او 16-bit. اما خارطة اللون للصورة map هي مصفوفة ابعادها $m-3$ محتوية على قيم ذات مدى يتراوح من (0 الى 1). كل صف من الخارطة يحدد مركبات الالوان الاحمر والاخضر والازرق كلون واحد. تستخدم الصور المفهرسة الرسم المباشر للقيم في الصورة حيث ان كل لون لنقطة صورية يحدد بواسطة القيم الماثلة لـ x في خارطة الالوان. القيمة 1 تشير الى الصف الاول في خارطة الالوان والقيم 2 تشير الى الصف الثاني وهكذا خارطة اللوات تخزن مع الصور المفهرسة وتحمل تلقائيا مع الصورة عندما يتم قراءة الصورة المفهرسة.



```
Mycolormap(5,:)= [0.2902 0.0627 0.0627];
```

```
&&
```

```
Myimage(m,n)=5;
```

وهذا الدليل 5 يشير إلى اللون القريب من الأزرق الموجود في مصفوفة خارطة اللون .

١,٣ يوجد نوعان لمجال اللون:

- **Double**: يكون مجاله [٠,١] اذ يكون الأسود ٠ واللون ١ ودرجات اللون بينهما اذ تكون قيم اللون اصغر من الواحد (فاصلة) مثلا ٠,٢٣٤.

- **Unit8**: تكون قيمته صحيحة Integer ويتراوح مجاله [٠,٢٥٥] اذ ٠ يمثل الأسود و٢٥٥ اللون وما بينهما درجات اللون.

يستخدم نوع unit8 بدلاً من double للتقليل من مساحة الذاكرة ولتسريع من عملية معالجة الصور.

```
>> c=imread('caribou.tif');
>> cd=double(c);
>> imshow(c),figure,imshow(cd)
```

٢-٣ فتح وقراءة صورة:

يتم فتح وقراءة صورة من أي نوع باستخدام تعليمة **Imread**: ويتم عرضها باستخدام التعليمة **imshow** ونميز حالتين:

١- فتح صورة من الحاسوب: لفتح صورة وعرضها من الجهاز الحاسوب نستخدم الصيغة الآتية:

```
X=imread ('path\filename,format');
Imshow(x);
```

اذ تتم قراءة الصورة عبر المسار filename ذات الإمتداد format ومن ثم تخزينها في المصفوفة x.

٢- فتح صورة من برنامج الماتلاب نفسه نستخدم الصيغة الآتية:

```
X=imread ('filename,format');
Imshow(x)
```

مثال ١: صورة من نوع RGB

```
X = imread ('onion.png');
imshow(X);
```

Imwrite: اذ تم معالجة صورة معينة وأجرينا عليها التغييرات المناسبة ثم أردنا حفظ أو طباعة هذه الصورة على جهاز الحاسوب بإسم جديد وإمتداد جديد نستخدم هذه التعليمة بالصيغة الآتية:

```
Imwrite (image, filename);
```

مثال ٢:

```
x=imread('C:\Users\DigitalNet\Desktop\ ferrari','jpeg')
imshow(x)
imwrite(x,'newferrari.bmp');
```

هنا تم طباعة نفس الصورة الموجودة على سطح المكتب إلى مسار الماتلاب الآني (current folder) بإسم جديد newferrari وإمتداد جديد .bmp.

Iminfo: تمكن هذه التعليمة بالحصول على معلومات كاملة عن الصورة وبكتابة الصيغة الاتية:

```
info=imfinfo(filename,format)
```

نحصل على معلومات أهمها:

مسار ملف الصورة – حجم الملف – العرض – الإرتفاع – الإمتداد – نظام الألوان.

مثال :

```
info=imfinfo('D:\sky','jpeg')
```

والنتيجة الظاهرة :

```
info =
```

```
Filename: 'D:\sky.jpg'
```

```
FileModDate: '08-١٦:٠٩:٥٢ ٢٠٠٧-ابريل'
```

```
FileSize: 575314
```

```
Format: 'jpg'
```

```
FormatVersion: ''
```

```
Width: 1280
```

```
Height: 960
```

```
BitDepth: 24
```

```
ColorType: 'truecolor'
```

```
FormatSignature: ''
```

```
NumberOfSamples: 3
```

```
CodingMethod: 'Huffman'
```

```
CodingProcess: 'Sequential'
```

```
Comment: {}
```

```
Orientation: 1
```

```
XResolution: 72
```

```
YResolution: 72
```

```
ResolutionUnit: 'Inch'
```

```
Software: 'ACD Systems Digital Imaging '
```

```
DateTime: '2007:03:25 01:04:50 '
```

```
YCbCrPositioning: 'Centered'
```

```
DigitalCamera: [1x1 struct]
```

الملخص:

1. الصور (Images)

يتعامل (Matlab) مع اربعة انواع من الصور:

- الصورة الملونة (24 bit/pixel) ويشار اليها ب (rgb) وتظهر في (work space) على شكل مصفوفة مكونة من ثلاث مستويات (width x height x 3) تمثل هذه المستويات الحزم الثلاثة (red, green, blue)، هذه الصورة ذو وضوحية عالية وتشغل مساحة تخزينية عالية.
- الصورة الملونة (256 colors) ويشار اليها ب (ind) تظهر في (work space) على شكل مصفوفتين، مصفوفة الشدة (intensity) بابعاد (width x height) ومصفوفة الخارطة (map) وتكون بابعاد (256 x 3) تكون ذو وضوحية وخرن اقل من الصورة السابقة.
- الصورة الرمادية (gray) ويشار اليها ب (gray) تظهر في (work space) على شكل مصفوفة واحدة (intensity) وتكون بابعاد (width x height).
- الصورة الثنائية (binary)

2. قراءة صورة (Image read)

- لقراءة صوره من نوع (rgb) نستخدم الايعاز: `rgb= imread ('c:/water.jpg','jpg');`
- لقراءة صوره من نوع (ind) نستخدم الايعاز: `[I map]= imread ('c:/water.bmp','bmp');`
- لقراءة صوره من نوع (gray) نستخدم الايعاز: `I = imread ('c:/water.bmp','bmp');`

3. لعرض صورة (Image show)

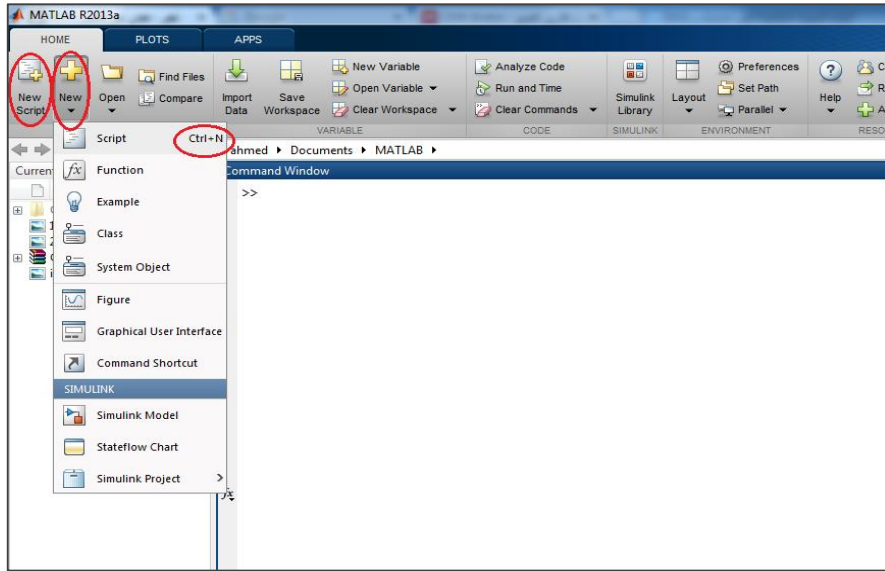
- لعرض صوره من نوع (rgb) نستخدم الايعاز: `imshow(rgb);`
- لعرض صوره من نوع (ind) نستخدم الايعاز: `imshow(I, map);`
- لعرض صوره من نوع (gray) نستخدم الايعاز: `imshow(I);`

4. لخرن صورة (Image write)

- لخرن صوره من نوع (rgb) نستخدم الايعاز: `imwrite(rgb,filename,fmt);`
- لخرن صوره من نوع (ind) نستخدم الايعاز: `imwrite(I,map,filename,fmt);`
- لخرن صوره من نوع (gray) نستخدم الايعاز: `imwrite(I,filename,fmt);`

ملاحظة مهمة:

يعمل المفتاح Enter في الماتلاب للنزول من سطر الى اخر والتنفيذ بنفس الوقت، ولكتابة جميع اسطر البرنامج ثم اجراء التنفيذ (بعد الانتهاء من كتابة البرنامج)، يستخدم الملف *.m او m-file. وهو ملف يستخدم ايضا لحفظ البرامج عند التطبيق في الماتلاب.



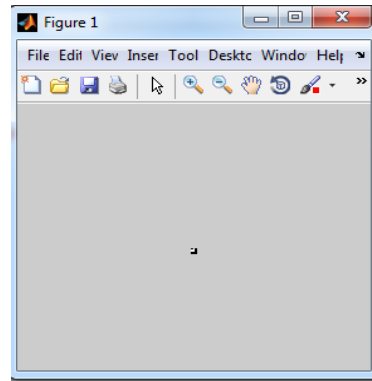
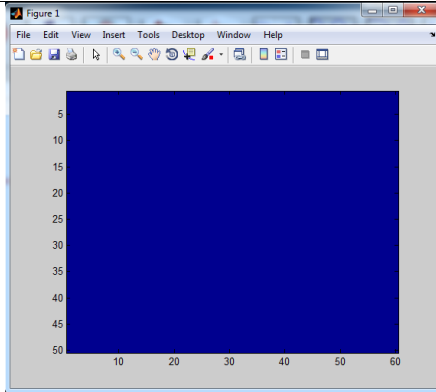
ويتم انشاء ملف m-file من خلال الاماكن المؤشر عليها في الشكل اعلاه.

مثال: إنشاء صورة ثنائية:
تخزين هذه القيم في المتغير X.

```
x=[0 1 0; 1 1 0; 0 0 0];
imshow(x)
imwrite(x,'im1.jpg');
```

او باستخدام الدالة zeros.

```
x=zeros (50,60);
imshow(x)
imwrite(x,'im۲.jpg');
```



تمرين: على ماذا نحصل اذا اجرينا الاتي:
١. كتابة الشفرة الاتية في الخطوة الاولى من البرنامج الاخير:

```
x=zeros (50)؛
```

٢. استخدام الدالة ones بدل zeros:

```
x= ones (50, 60)؛
```

٣. نفذ البرنامج:

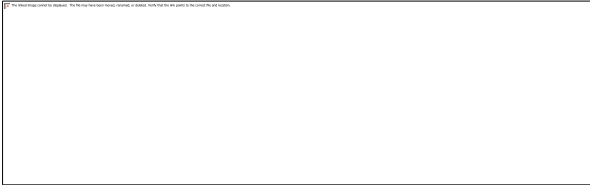
```
x= zeros (500, 600)؛
imshow(x)
x(100:200, 100:200)=1
imshow*x)
```

مثال: إنشاء صورة Grayscale:

في البداية نكون قيم عشوائية بين ٠ و ١ على شكل مصفوفة ببعدين، باستخدام الدالة Rand، ثم تخزين هذه القيم في المتغير x.

```
x = rand (5);
```

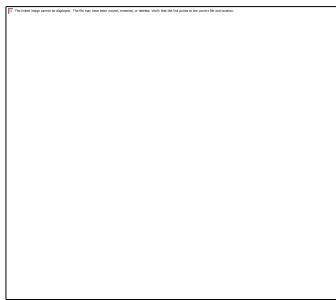
القيم العشوائية التي حصلنا عليها هي كما في الشكل الاتي:



مثال:

```
image (x*64);  
colormap(gray) ;
```

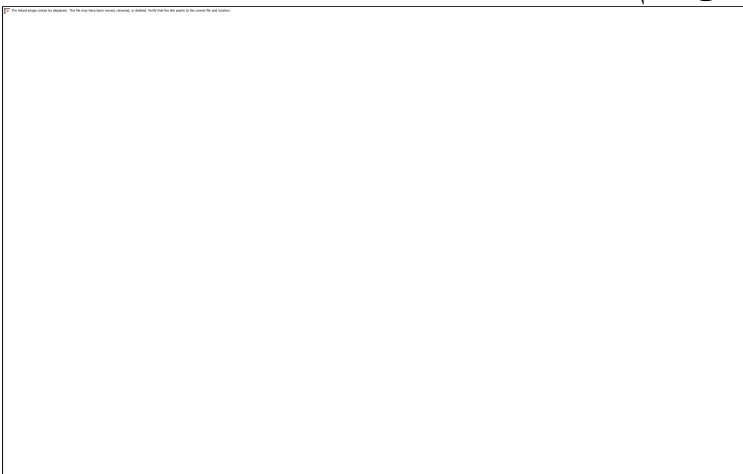
والنتيجة هي:



مثال: إنشاء صورة من نوع RGB:

```
x (:,:,1) = rand (7) ;  
x (:,:,2) = rand (7) ;  
x (:,:,3) = rand (7) ;
```

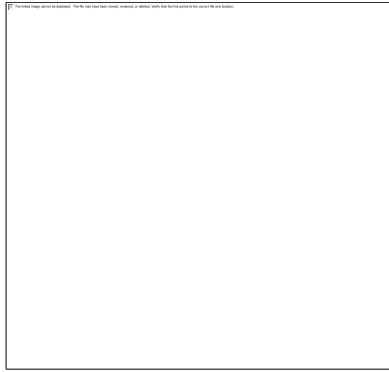
هذه الشفرة تسمح بتشكيل مصفوفة بثلاث أبعاد $M \times N \times 3$ كما ذكرنا سابقاً، وفي هذه الحالة فإن أبعاد المصفوفة هي $7 \times 7 \times 3$ وفي حالتنا في هذا المثال حصلنا على القيم الاتية:



وعند كتابة الأمر:

```
image(x);
```

تكون النتيجة:



مثال: قراءة صورة من نوع Indexed:

```
[X,map] = imread('trees.tif');
```

```
imshow(X,map)
```



مثال: إنشاء صورة من نوع Indexed:

كما ذكرنا سابقا فإن هذا النوع من الصور هو عبارة عن مصفوفتين يتم تعريف كل من المصفوفة y والمصفوفة map_y . المصفوفة y عبارة عن بعدين وتمثل أبعاد الصورة، وسنفرض أن أبعاد الصورة هي 4×4 ولذلك فإن مصفوفة map_y ستكون أبعادها 3×16 (ما تفسير هذا الرقم؟). للحصول على أرقام عشوائية من 1 إلى 16 على شكل مصفوفة 4×4 قم بكتابة الكود:

```
y = reshape(randperm(16), 4, 4);
```

فتحصل على:

$y =$

3	2	11	1
4	15	16	6
9	10	12	5
8	13	7	14

ومصفوفة map_y :

```
map_y = rand(16,3);
```

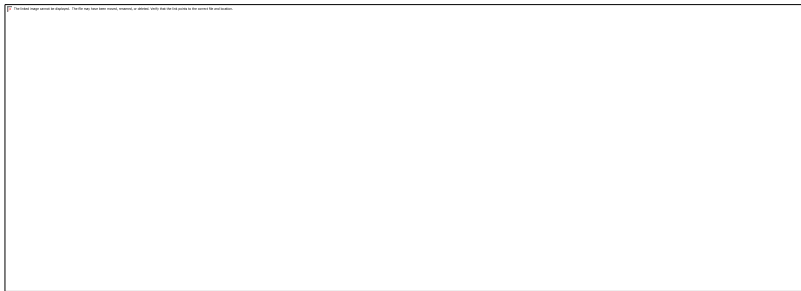
فتحصل على:


```
map_y =
    0.1500    0.0616    0.2458
    0.6386    0.2168    0.1116
    0.6108    0.7550    0.9532
    0.3400    0.2273    0.7989
    0.0259    0.8489    0.2591
    0.7309    0.1790    0.5006
    0.3661    0.2906    0.0523
    0.4330    0.9923    0.4138
    0.1236    0.4001    0.2778
    0.9153    0.9388    0.2735
    0.3221    0.9883    0.9242
    0.0182    0.2901    0.2608
    0.9255    0.9005    0.7504
    0.6506    0.8332    0.6298
    0.4335    0.6657    0.3355
    0.7470    0.6555    0.1815
```

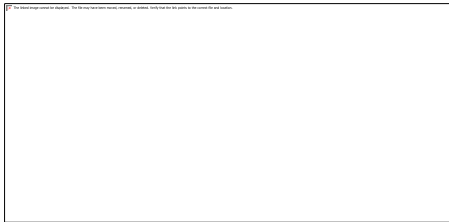
لتشكيل الصورة قم بإظهار المصفوفة y أولاً، ثم ادمجها مع map_y كآتي:

```
image(y)
colormap(map_y)
```

فتحصل على:

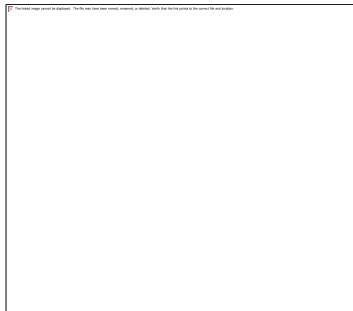


مثال) إنشاء صورة من نوع Binary: نجرب المصفوفة التالية



```
image(64*x)
colormap(gray)
```

نحصل على:



تمرين: نفذ الشفرة الآتية:

```
myrgb(:,1)=rand(7,7);
myrgb(:,2)=rand(7,7);
myrgb(:,3)=rand(7,7);
image(myrgb);
axis image
```

٤- تحويل بين الصور:

Im2bw: يمكن تحويل الصورة أيا كان نوعها إلى صورة ثنائية (أبيض وأسود) باستخدام هذه الصيغة:

```
binary image=im2bw(image,level);
```

اذ `image` تمثل الصورة الأصلية أما `level` فهي شدة العتبة تتراوح بين الصفر والواحد اذ تتحول جميع البكسلات التي قيمتها فوق العتبة إلى الأبيض أو تحمل القيمة 1 أما البكسلات التي تحمل قيم تحت العتبة فتتحول إلى اللون الأسود أو تحمل القيمة صفر.

مثال:

```
x=imread('C:\Users\DigitalNet\Desktop\mosque','jpeg');
imshow(x);
y=im2bw(x,0.25);
imshow(y);
```

١- التحويل من grayscale إلى RGB :

```
myrgb = cat(3,mygray,mygray,mygray);
```

مثال :

```
mygray=imread('pout.tif');
imshow(mygray)
myrgb = cat(3,mygray,mygray,mygray);
figure,imshow(myrgb)
```

والنتيجة هي :



٢- التحويل من RGB إلى grayscale باستخدام القيمة المتوسطة :

```
mygray = mean(myrgb,3);
```

مثال :

```
myrgb=imread('onion.png');
imshow(myrgb)
mygray = round(mean(myrgb,3))/255;
figure,imshow(mygray)
```

والنتيجة هي :



٣- التحويل من RGB إلى grayscale باستخدام الوزن NTSC :

```
mygray = rgb2gray(myrgb);
```

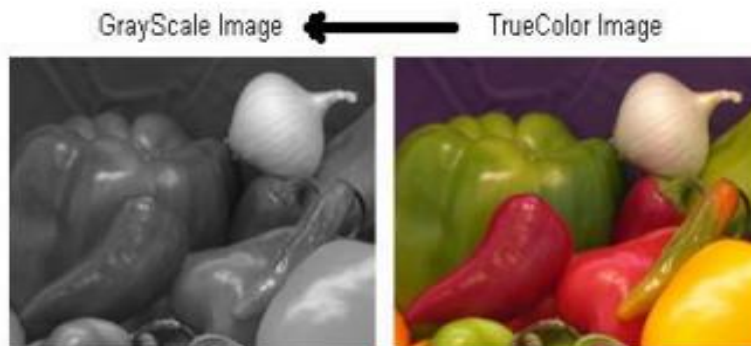
ملاحظة :

- التحويل من RGB إلى Grayscale باستخدام الوزن NTSC يتم على الشكل التالي :

```
mygray =  
0.2989* myrgb (:,:,1) + 0.5870* myrgb (:,:,2) + 0.1140* myrgb (:,:,3);
```

مثال :

```
myrgb=imread('onion.png');  
imshow(myrgb)  
mygray = rgb2gray(myrgb);  
figure,imshow(mygray)
```

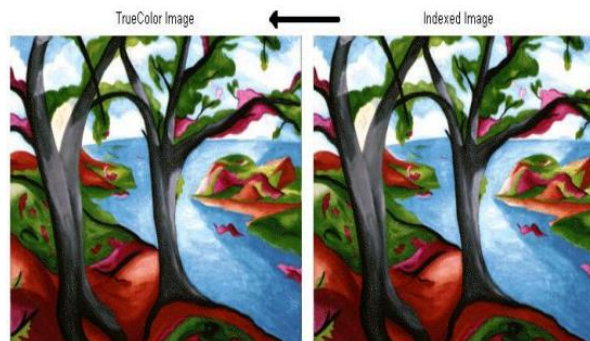


٤- التحويل من Indexed إلى RGB :

```
myrgb = ind2rgb(myindexed,mycolormap);
```

مثال :

```
[myindexed, mycolormap] =imread('trees.tif');  
imshow(myindexed, mycolormap)  
myrgb = ind2rgb(myindexed,mycolormap);  
figure,imshow(myrgb)
```



نلاحظ أنه لا فرق بين الصورتين لأن كلا الصورتين تملكان نفس نظام الألوان RGB ولكن تختلفان عن بعضهما بطريقة التمثيل في الماتلاب .

٥- التحويل من RGB إلى Indexed باستخدام K لون :

```
[myindexed,mycolormap] = rgb2ind(myrgb,K);
```

مثال :

```
myrgb=imread('peppers.png');  
imshow(myrgb)  
[myindexed,mycolormap] = rgb2ind(myrgb,256);  
figure,imshow(myindexed,mycolormap)
```



طبعا في هذا المثال Indexed Image تملك 256 لون تتدرج من الأبيض للأسود في خارطة اللون Colormap الخاصة بالصورة.

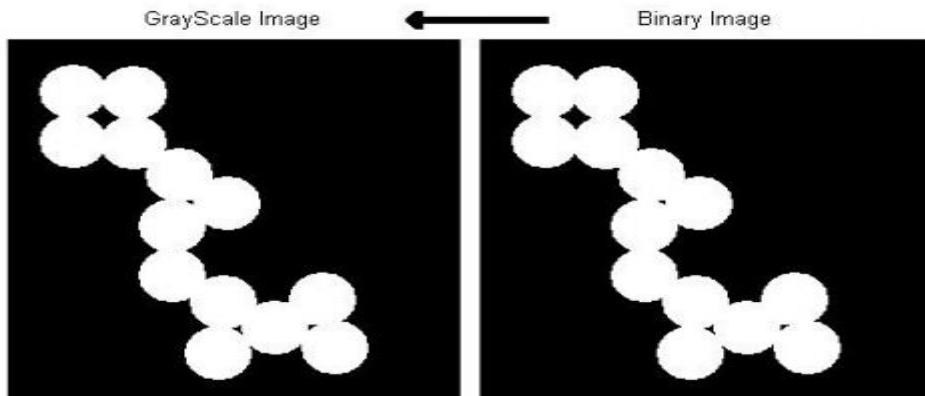
٦- التحويل من Binary إلى Grayscale :

```
mygray = double(mybinary);
```

مثال :

```
mybinary=imread('circles.png');  
imshow(mybinary)  
mygray = double(mybinary);  
figure, imshow(mygray)
```

والنتيجة هي :



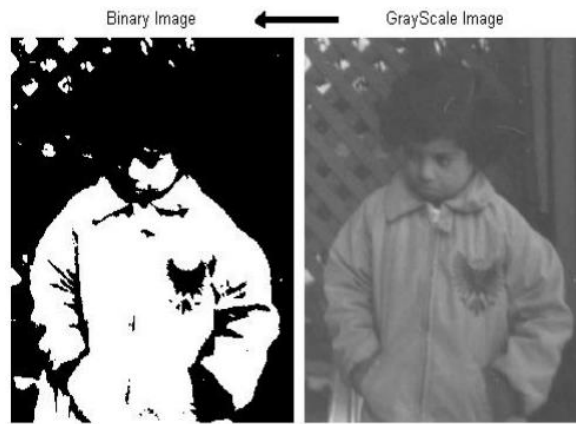
٧- التحويل من Grayscale إلى Binary :

```
mybinary = (mygray > a);
```

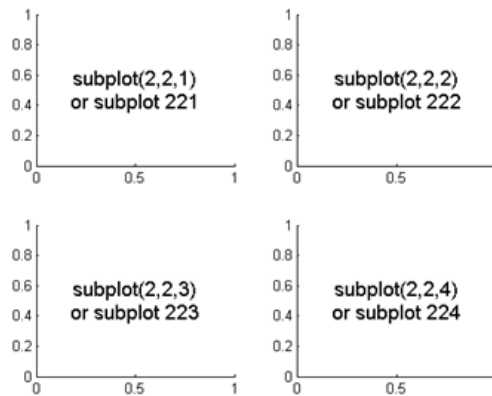
حيث جميع الألوان الرمادية التي تقع فوق a تتحول إلى لون أبيض والتي تقع تحت a تتحول لون أسود .

مثال :

```
mygray=imread('pout.tif');  
imshow(mygray)  
level=graythresh(mygray)*256;  
mybinary = (mygray > level);  
figure , imshow(mybinary)
```



مثال اهمية الدالة subplot: تعمل هذه الدالة على وضع مجموعة رسوم او صور في اطار واحد لمعاينتها جميعا في نفس الوقت.

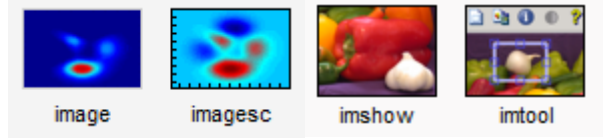


مثال

```
a=imread(...);  
b=imread(...);  
subplot(2,2,1); imshow(a); title('Outgo')  
subplot(2,2,2); imshow(b); title('Income')
```

ملاحظة: تستخدم الدالة title لوضع عنوان فوق الصورة عند معاينتها.

تمرين: اكتب الفرق بين اهمية دوال (معاينة او اظهار الصورة)



الملخص:

<code>(I,map) = rgb2ind(rgb, 256);</code>	لتحويل من صورة (rgb) الى صورة (ind) نستخدم الابعاز:
<code>I = rgb2gray(rgb);</code>	لتحويل من صورة (rgb) الى صورة (gray) نستخدم الابعاز:
<code>rgb = ind2rgb(I,map)</code>	لتحويل من صورة (ind) الى صورة (rgb) نستخدم الابعاز:
<code>I = ind2gray(x,map)</code>	لتحويل من صورة (ind) الى صورة (gray) نستخدم الابعاز:
<code>(I, map) = gray2ind(I,256)</code>	لتحويل من صورة (gray) الى صورة (ind) نستخدم الابعاز:

هـ. عمليات على الصورة:

- تكبير وتصغير حجم الصورة **Imresize**: تستخدم لتغيير حجم او ابعاد الصورة. يمكن تغيير حجم الصورة ثنائية البعد باستخدام الصيغة الأولى:

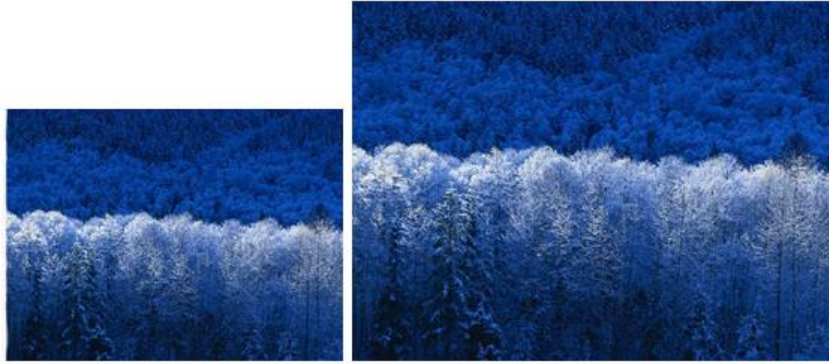
```
J=imresize(I,Scale)
```

اذ I هي الصورة قبل إعادة التقييس و scale نسبة التقييس فإذا كانت أكبر من الواحد فسيزداد حجم الصورة أما إذا كانت أصغر من الواحد فسينقص حجم الصورة.

مثال: لتكبير صورة، تم تعريف معامل التكبير بالقيمة: 1.25

```
clc;
clear all;
I = imread('Winter.jpg');
J = imresize(I,1.25);
imshow(I)
figure, imshow(J)
```

النتائج:



الشكل الثاني :

```
J=imresize(I,[nrows ncol])
```

في هذا الشكل يمكننا إعادة تحجيم الصورة إلى الحجم الذي نرغب به

مثال :

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

```
J = imresize(I,[240 320]);
```

```
Original_size = size(I)
```

```
After_size = size(J)
```

```
figure, imshow(I), figure, imshow(J)
```

والنتائج كما يلي :

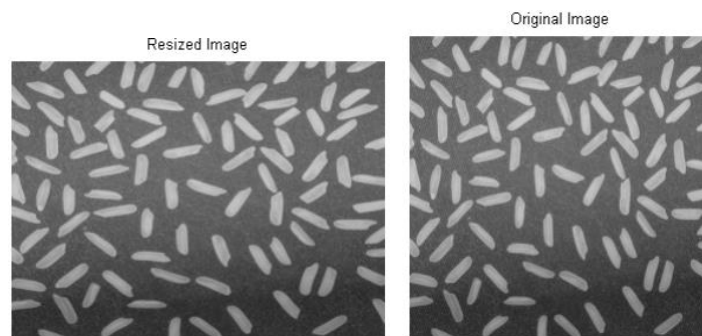
```
Original_size =
```

```
256 256
```

```
After_size =
```

```
240 320
```

والصور



الشكل الثالث :

وهو خاص بالصور من نوع GrayScale :

```
[X, map] = imread('trees.tif');  
imshow(X, map)  
[Y, newmap] = imresize(X, map, 1.5);  
figure,imshow(Y, newmap)  
Original_size = size(X)  
After_size = size(Y)
```

والنتائج كما يلي :

```
Original_size =  
258 350
```

```
After_size =  
387 525
```

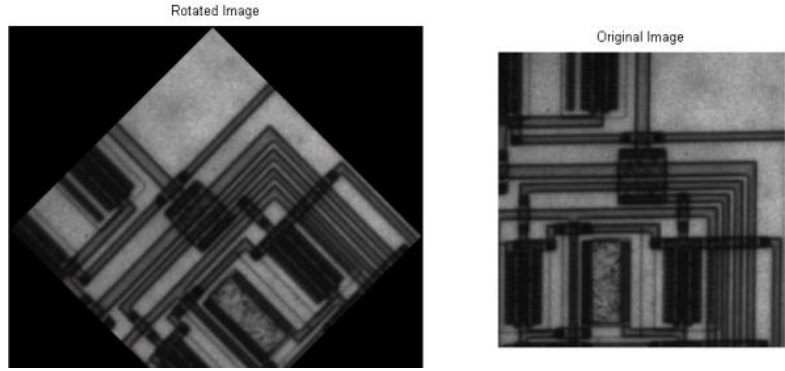
تدوير (دوران) الصورة:

Imrotate: تدوير الصورة بزاوية معينة بالدرجات.

```
J=imrotate(I,Angle);
```

اذ I هي الصورة الأصلية المراد تدويرها و angle زاوية الدوران مقدرة بالدرجات.
مثال:

```
I = imread('circuit.tif');  
J = imrotate(I,45,'bilinear');  
imshow(I)  
figure,  
imshow(J)
```

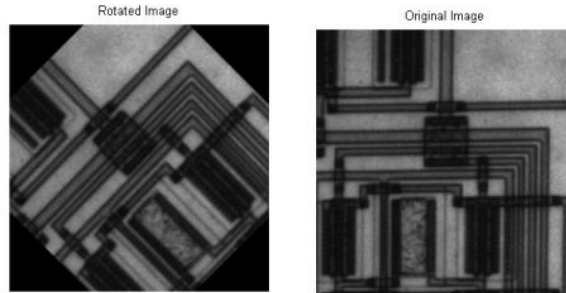
أما أردنا أن يكون للصورة الناتجة نفس حجم الصورة الأصلية لكن نلاحظ إقتطاع جزء من الصورة .

مثال ٢ :

```
I = imread('circuit.tif');
J = imrotate(I, 45,'bilinear','crop');
Original_size = size(I)
After_size = size(J)
imshow(I)
figure, imshow(J)
```

والنتائج كما يلي :

```
Original_size =
    280    272
After_size =
    280    272
```



- إقتطاع جزء من الصورة I: imcrop يمكن إقتطاع جزء من الصورة وإنشاء صورة جديدة من الصورة المقطعة.

```
J=imcrop(I);
```

اذ z هي الصورة المقطعة يدويا اذ يتوجب عليك تحديد الجزء من الصورة الذي نريد إقتطاعه يدويا.
- قلب الصورة Fliplr: تمكن من إجراء قلب للصورة من اليمين لليساار (لا ننسى أن الصورة عبارة عن مصفوفة) أي وكأن الصورة وضعت أمام مرآة.

```
J=fliplr(I);
```

اذ I الصورة الأصلية Flipud: تمكنا هذه التعليمة من إجراء قلب للصورة من الأعلى إلى الأسفل (لا ننسى أن الصورة عبارة عن مصفوفة).

```
J=flipud(I);
```

اذ I الصورة الأصلية.

العمليات على الصور الثنائية

١- اختيار عدة Objects من صورة ثنائية bwselect :

يمكن اختيار عدة Objects من صورة ثنائية وتخزينها في صورة أخرى تحوي فقط على Objects التي تم اختيارها باستخدام التعليمة bwselect وفق شكلين :

الشكل الأول :

```
bw2=bwselet(bw1);
```

يتم اختيار Objects من الصورة bw يدوياً بالضغط مرة واحدة على Object وبعد الانتهاء من الاختيار نضغط Enter .

مثال :

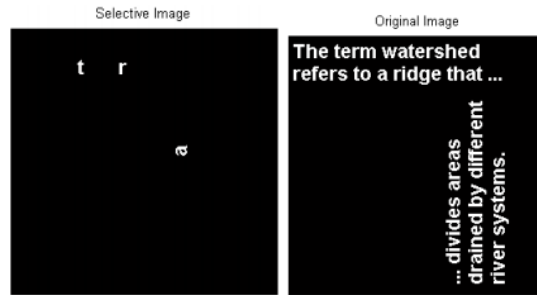
```
bw1 = imread('text.png');
```

```
bw2=bwselect(bw1);
```

```
imshow(bw1)
```

```
figure , imshow(bw2)
```

والنتائج كما يلي :



الشكل الثاني :

```
bw2=bwselet(bw1,c,r,n);
```

يتم اختيار Objects من خلال تحديد إحداثيات نقطة واحدة من كل Object وبالتالي نحدد عدة نقاط حيث c تحوي الإحداثيات الأفقية x و r تحوي الإحداثيات الشاقولية y أما n تأخذ إحدى القيمتين 8 , 4 وهي تعبر أن أصغر Object مؤلف من تجمع 4 أو 8 بكسلات متجمعة .

مثال :

```
bw1 = imread('text.png');
```

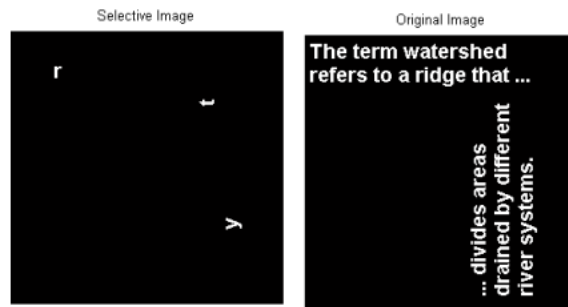
```
c = [43 185 212];
```

```
r = [38 68 181];
```

```
bw2=bwselect(bw1,c,r,4);
```

```
imshow(bw1)
```

```
figure, imshow(bw2)
```



٢- مساحة جميع Objects في الصورة الثنائية bwarea:

يمكن الحصول على مساحة جميع Objects في الصورة الثنائية Binary image باستخدام التعليمة bwarea وفق الشكل التالي:

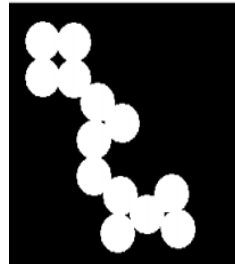
```
Total_Area=bwarea(bw)
```

مثال:

```
bw= imread('circles.png');
```

```
imshow(bw);
```

```
Total_Area=bwarea(bw)
```



٣- حذف Objects من الصورة الثنائية ذات مساحة أقل من N بكسل:

يمكن حذف Objects من الصورة الثنائية ذات مساحة ثنائية أقل من N بكسل والصورة الجديدة تحوي جميع Objects ذات المساحة أكبر من N بكسل باستخدام التعليمة bwareaopen وفق الشكل:

```
bw2=bwareaopen(bw1,N)
```

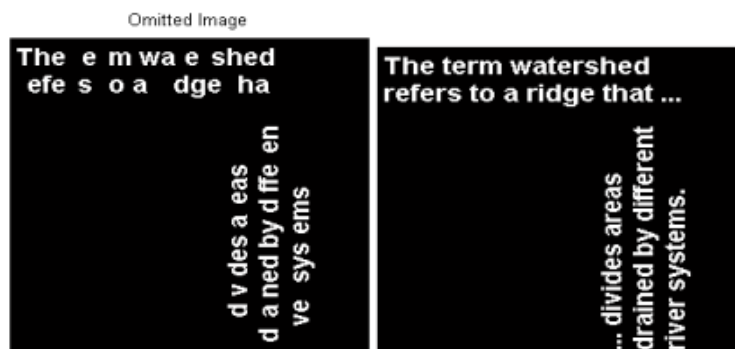
حيث N المساحة التي تحتها يحذف ال- Object من الصورة.

```
bw1 = imread('text.png');
```

```
imshow(bw1)
```

```
bw2 =bwareaopen(bw1,50);
```

```
figure , imshow(bw2)
```



٤ - تحديد المسافة بين كل بكسل والبكسل الأقرب غير المساوي للصفر :

يمكن الحصول على مصفوفة المسافة للمصفوفة الثنائية Binary Image باستخدام التعليمة bwdist وفق الشكل :

[D,L]=bwdist (bw ,method)

حيث D مصفوفة المسافة هي مصفوفة كل عنصر فيها هو المسافة بين العنصر المقابل و اقرب عنصر يساوي الواحد له في المصفوفة الثنائية .

أما L مصفوفة تحوي الدليل الخطي للعنصر الأقرب المساوي للواحد من الصورة الثنائية .

تعطى المسافة بحسب طريقة الحساب :

chessboard : Distance = max(|x1-x2| , |y1-y2|) ;

cityblock : Distance = |x1-x2| + |y1-y2| ;

euclidean : Distance = sqrt((x1-x2).^2+(y1-y2).^2) ;

quasi-euclidean :

Distance = |x1-x2| + (sqrt(2) -1) |y1-y2| ; if |x1-x2| > |y1-y2|

Distance = (sqrt(2) -1) |x1-x2| + |y1-y2| ; otherwise

افتراضياً تحسب المسافة وفق طريقة Euclidean .

مثال :

```
bw = zeros(5,5); bw(2,2) = 1; bw(4,4) = 1;
bw
```

```
[D,L] = bwdist(bw);
```

```
D
```

```
L
```

والنتائج كما يلي :

```
bw =
    0    0    0    0    0
    0    1    0    0    0
    0    0    0    0    0
    0    0    0    1    0
    0    0    0    0    0
```

D =

```
1.4142  1.0000  1.4142  2.2361  3.1623
1.0000   0  1.0000  2.0000  2.2361
1.4142  1.0000  1.4142  1.0000  1.4142
2.2361  2.0000  1.0000   0  1.0000
3.1623  2.2361  1.4142  1.0000  1.4142
```

L =

```
7  7  7  7  7
7  7  7  7 19
7  7  7 19 19
7  7 19 19 19
7 19 19 19 19
```

٥- إيجاد حواف Objects في صورة ثنائية bwperim :

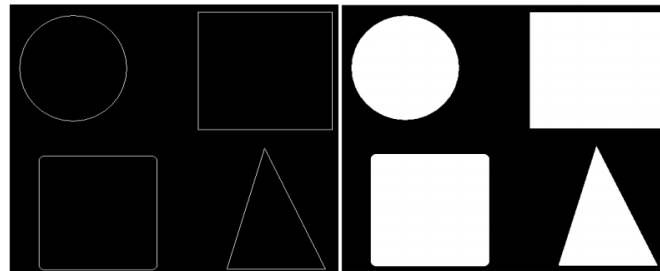
يمكن إيجاد حواف Objects في الصورة الثنائية باستخدام التعليمة bwperim وفق الشكل :

```
bw2=bwperim(bw1,conn)
```

حيث bw1 الصورة الأصلية أما conn هو 4 أو 8 وهو أقل مساحة للـ Object .
و bw2 صورة ثنائية تظهر حواف كل Object وكل Hole في الصورة الأصلية .

مثال :

```
I1=imread('D:\Shapes.jpg');
level=graythresh(I1)
bw1=im2bw(I1);
imshow(bw1)
bw2=bwperim(bw1,4);
figure , imshow(bw2)
```



٦- رقم Euler للصورة الثنائية bweuler :

للحصول على رقم Euler للصورة كاملة نستخدم التعليمة bweuler :
رقم Euler للصورة = عدد Objects في الصورة - عدد الثقوب في الصورة
ومنه

عدد الثقوب في الصورة = عدد Objects في الصورة - رقم Euler للصورة

```
I1=imread('D:\Shapes1.jpg');  
level=graythresh(I1)  
bw1=im2bw(I1,level);  
[labeled,numObjects]=bwlabel(bw1,4);  
info=regionprops(labeled,'all');  
Euler_Number=bweuler(bw1,4);  
NumHoles=numObjects-Euler_Number
```

والنتائج كما يلي :

```
level =  
    0.4843  
NumHoles =  
    8
```

٧- متممة الصورة image complement الرمادية او ما يسمى negative:

```
rgb=imread('image1.format');  
I=rgb2gray(rgb);  
Ic=imcomplement(I);  
subplot(1,2,1); imshow(I), title ('image1');  
subplot(1,2,2); imshow(Ic), title ('image complement');
```