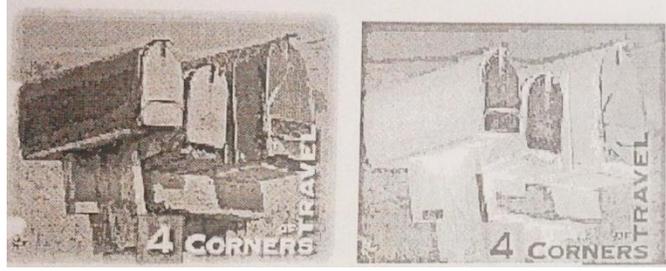


**محاضرات العملي لمادة معالجة
الصور
قسم علوم الحاسبات
كلية التربية**

م. امل عباس كاظم



مثال :

```
bw = imread('text.png');
imshow(bw)
bw2 = imcomplement(bw);
figure , imshow(bw2)
```

والنتائج كما يلي :



العمليات الحسابية على الصور الرقمية

١- جمع (دمج) صورتين رقميتين **imadd** :

يمكن جمع صورتين أي دمجهما في صورة واحدة من خلال التعليمة **imadd** على الشكل :

```
K=imadd(I,J) ;
```

مثال :

```
I = imread('rice.png');
imshow(I)
J = imread('cameraman.tif');
figure , imshow(J)
K=imadd(I,J);
figure, imshow(K)
```

Image 1

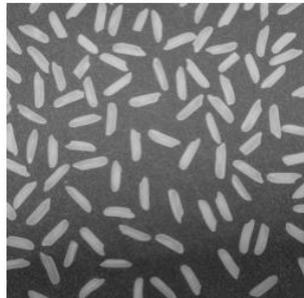


Image 2





نحن نعلم أن الألوان قيمها تتراوح بين 0 (اللون الأسود) و 255 (اللون الأبيض) لكن هناك مشكلة عند جمع قيمتين ويكون الناتج يزيد عن 255 فإنه يقص إلى 255 لأن نوع الصورة uint8 مما يؤدي إلى ضياع بعض الألوان حيث تذهب إلى اللون الأبيض لذلك يفضل توسيع مجال الألوان إلى [0 512] أي التحويل إلى نوع uint16 .

مثال :

```
I = imread('rice.png');
J = imread('cameraman.tif');
K = imadd(I,J,'uint16');
imshow(K,[])
```

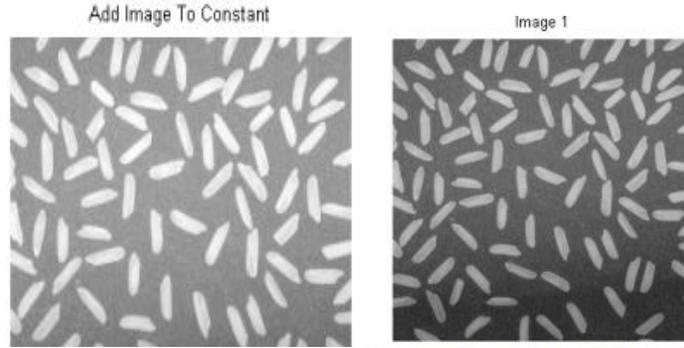


أو يمكن إضافة عدد ثابت إلى الصورة (تفتيح الصورة) .

مثال :

```
I = imread('rice.png');
imshow(I)
J = imadd(I,50);
figure , imshow(J)
```

والنتائج كما يلي :



٢- طرح صورة رقمية من أخرى `imsubtract` :

يمكن طرح صورة رقمية من أخرى من خلال التعليمة `imsubtract` على الشكل التالي :

`K=imsubtract(I,J) ;`

غالبا من غير المفيد طرح صورة رقمية من أخرى لكن من الممكن طرح صورة رقمية من خلفية نفس الصورة لجعل الخلفية سوداء تماما كما ذكرنا سابقا

مثال :

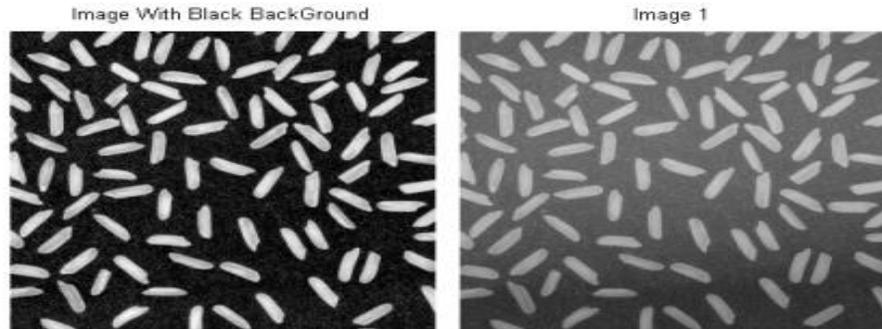
`I = imread('rice.png');`

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

`Ip = imsubtract(I,background);`

`imshow(Ip,[])`

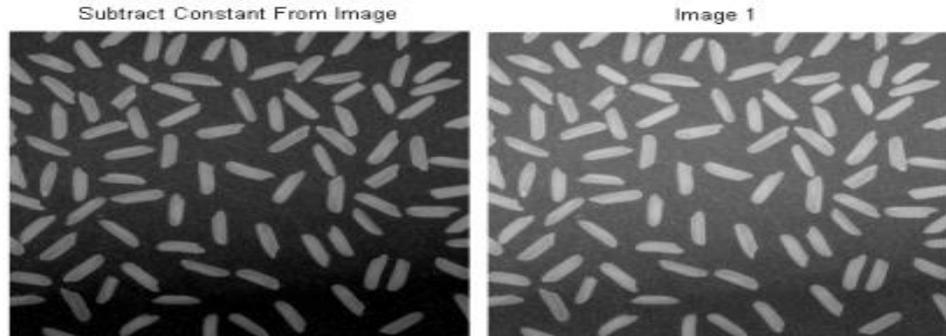
والنتائج كما يلي :



أو يمكن طرح عدد ثابت من الصورة (تعتيم الصورة) .
مثال :

```
I = imread('rice.png');  
Iq = imsubtract(I,50);  
figure, imshow(I), figure, imshow(Iq)
```

والنتائج كما يلي :



٣- ضرب صورتين رقميتين ببعضهما **immultiply** :

يمكن ضرب صورة رقمية بأخرى من خلال التعليمة **immultiply** على الشكل التالي :

```
K= immultiply (I,J) ;
```

حيث يمكن ضرب صورة رقمية بعدد ثابت .

مثال :

```
I = imread('moon.tif');  
imshow(I)  
J = immultiply(I,1.2);  
figure , imshow(J)
```

والنتائج كما يلي :



٤ - قسمة صورتين رقميتين من بعضهما **imdivide** :

يمكن قسمة صورة رقمية من أخرى من خلال التعليمة **imdivide** على الشكل التالي :

```
K= imdivide (I,J) ;
```

حيث يمكن قسمة صورة رقمية على عدد ثابت .

مثال :

```
I = imread('moon.tif');
```

```
imshow(I)
```

```
J = imdivide(I,1.5);
```

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

و النتائج كما يلي :



٦ - الجمع الخطي للصور الرقمية **imlincomb** :

يمكن القيام بالجمع الخطي لعدة صور رقمية بعد تحديد عامل الضرب الخاص بكل صورة من خلال التعليمة **imlincomb** على الشكل التالي :

```
Z = imlincomb(K1,A1,K2,A2,....,Kn,An)
```

حيث $K1, K2, \dots, Kn$ عوامل الضرب

و $A1, A2, \dots, An$ الصور المراد جمعها

والصورة الناتجة هي :

```
Z=K1*A1+K2*A2+.....+Kn*An
```

والصورة ذات عامل الضرب الأكبر هي الصورة الأكثر أهمية .

مثال :

```
I = imread('rice.png');  
J = imread('cameraman.tif');  
K=imlincomb(0.2,I,0.8,J);  
imshow(K)
```

أي الصورة الجديدة تحوي 20% من I و 80% من J .
والنتائج كما يلي :



- تصميم مرشحات:

Fspecial: تصميم مرشح ثنائي البعد من أحد الأنواع الشهيرة مثل مرشح غاوص أو لابلاسيان أو اللوغاريتمي أو المتوسط.... ويمكن كتابته بالصيغة الآتية:

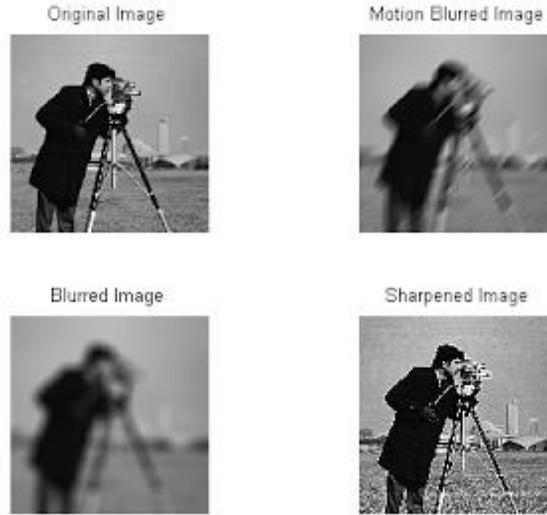
```
h = fspecial(type, parameters);
```

اذ تمثل نوع المرشح و parameter مواصفات هذا المرشح.
مثال:

```
I = imread('cameraman.tif');  
subplot(2,2,1);  
imshow(I); title('Original Image');
```

```
H = fspecial('motion',20,45);  
MotionBlur = imfilter(I,H,'replicate');  
subplot(2,2,2);  
imshow(MotionBlur);title('Motion Blurred Image');  
H = fspecial('disk',10);  
blurred = imfilter(I,H,'replicate');  
subplot(2,2,3);  
imshow(blurred); title('Blurred Image');
```

```
H = fspecial('unsharp');  
sharpened = imfilter(I,H,'replicate');  
subplot(2,2,4);  
imshow(sharpened); title('Sharpened Image');
```



- إضافة ضجيج:

Imnoise: إضافة ضجيج للصورة بكتابة الصيغة التالية

`J = imnoise(I,type)`

اذ I هي الصورة الأصلية type يمثل نوع الضجيج فقد يكون Gaussian أو salt and papper أو poisson
مثال:

```
I = imread('eight.tif');
J = imnoise(I,'salt & pepper',0.02);
figure,
imshow(I)
figure,
imshow(J)
```

- **Videoinput**: يمكن منها تشغيل الفيديو وإستعراضه من كاميرا الحاسوب أو أي كاميرا أو أي كاميرا موصولة معه عن طريق الUSB وهذه الصيغة:

```
obj = videoinput(adaptorname,deviceID,format)
```

اذ **adaptorname** إسم المتغير للكاميرا و **deviceID** هو رقم يدل على ID الكاميرا و **format** تمثل صيغة العرض قد تكون YUY2-640*480 أو MJPG-640*480 وأن الرقم 640*480 يمثل حجم الصورة ويمكن تغييره.
مثال:

```
cam=videoinput('winvideo',1,'YUY2_640x480');
set(cam,'returnedcolorspace','rgb');
vidRes = get(cam, 'VideoResolution');
nBands = get(cam, 'NumberOfBands');
hImage = image(zeros(vidRes(2), vidRes(1), nBands));
preview(cam,hImage);
```

ملاحظة: preview: لإستعراض الفيديو ومشاهدته في نافذة ضمن واجهة الماتلاب.
Getsnapshot: مهمته إلتقاط صورة من الكاميرا وتكتب بالصيغة الآتية:

```
Image=getsnapshot(obj);
```

يأتي هذا التابع مرافقا لتابع videoinput اذ obj إسم متحول الذي خزن فيه الفيديو
مثال:

```
cam=videoinput('winvideo',1,'YUY2_640x480');
set(cam,'returnedcolorspace','rgb');
vidRes = get(cam, 'VideoResolution');
nBands = get(cam, 'NumberOfBands');
hImage = image(zeros(vidRes(2), vidRes(1), nBands));
preview(cam,hImage);
pause(10);
image1=getsnapshot(cam);
figure;
imshow(image1);
```

٨. منحنى الهستوغرام:

يظهر هذا المنحنى توزيع الاضاءة في الصورة على البكسلات، اذ يعبر محور السينات على شدة الاضاءة ومحور الصادات بكسلات الصورة باستخدام:

Imhist(I)

تظهر هذه التعليلة الهستوغرام (تقسيم الإضاءة داخل الصورة) للصورة (I)، وإن عدد الـ bins في الهستوغرام محدد بنوع الصورة فإذا كان نوع الصورة grayscale فإن Imhist يستخدم قيم افتراضية من ٢٥٦ bins، أما إذا كانت الصورة ثنائية فإن Imhist تستخدم ٢ bins.

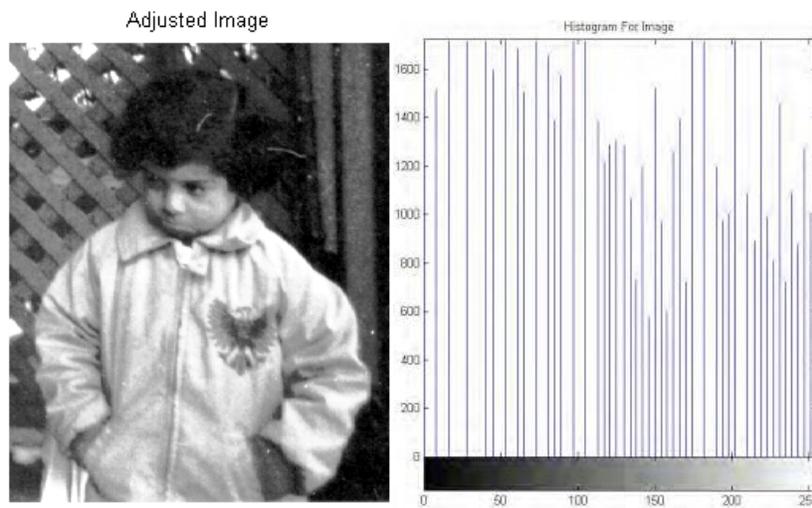
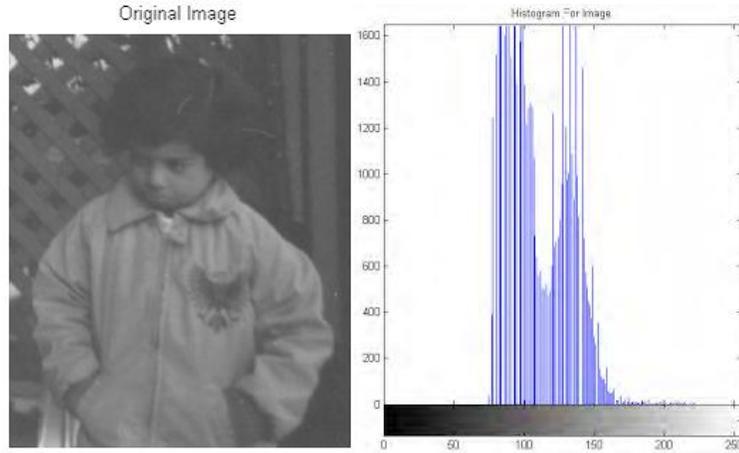
Imhist(I,n):

تعرض الهستوغرام اذ n تحدد عدد الـ bins المستخدم في الهستوغرام، وتحدد ايضا طول حاجز اللون (إذا كانت الصورة ثنائية n تأخذ قيمتين فقط).

تحسين تباين الصورة:

يمكن تحسين تباين الصورة باستخدام تعليمة histeq كما يأتي: اذ تقوم هذه التعليمة بفرد مستويات الاضاءة على كافة بكسلات الصورة، ولا تبقى موزعة على بكسلات معينة.

```
I = imread('pout.tif');
imshow(I)
figure, imhist(I)
I2 = histeq(I);
figure, imshow(I2)
figure, imhist(I2)
```

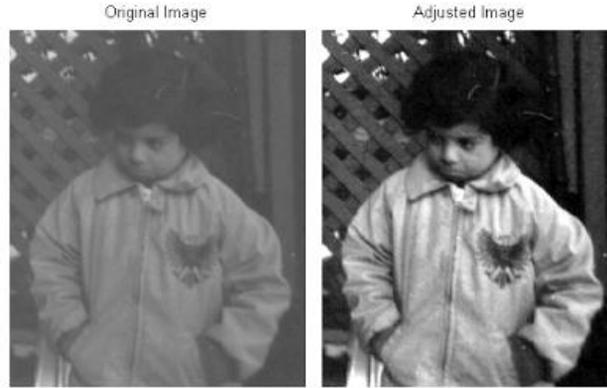


ويمكن تحسين تباين الصورة من خلال:
imadjust: يمكننا تعديل تباين الصورة باستخدام هذه التعليمة ولها عدة أشكال أحدها:

```
j=imadjust(i)
```

وبذلك تكون الصورة J هي نفس الصورة I مع زيادة تباين الصورة من خلال زيادة درجة الإشباع ١% للبيانات.
 مثال:

```
I=imread('pout.tif');
imshow(I);
J =imadjust(I);
figure;
imshow(J);
```



الشكل الثاني :

```
J =imadjust(I,[low_in;high_in],[low_out;high_out]);
```

حيث يتم تعديل تباين الصورة من خلال تعديل جميع البكسلات في الصورة I والتي قيمها تتراوح بين low_in و High_in والتي ستتغير قيمها وتنحصر ما بين low_out و high_out بشكل متقابل أما البكسلات التي تقع قيمها خارج low_in فتصبح قيمها 0 أي تحمل اللون الأسود والبكسلات التي تقع قيمها خارج high_in فتصبح قيمها 1 أي تحمل اللون الأبيض إذا كانت الصورة من نوع grayscale أو الأحمر أو الأخضر أو الأزرق إذا كانت الصورة من نوع RGB .

مثال :

```
I=imread('pout.tif');
imshow(I)
J =imadjust(I,[0.2;0.6],[0.4;0.8]);
figure,imshow(J)
```

والنتائج كما يلي :



الشكل الثاني :

```
J =imadjust(I,[low_in;high_in],[low_out;high_out]);
```

حيث يتم تعديل تباين الصورة من خلال تعديل جميع البكسلات في الصورة I والتي قيمها تتراوح بين low_in و High_in والتي ستتغير قيمها وتتنحصر ما بين low_out و high_out بشكل متقابل أما البكسلات التي تقع قيمها خارج low_in فتصبح قيمها 0 أي تحمل اللون الأسود والبكسلات التي تقع قيمها خارج high_in فتصبح قيمها 1 أي تحمل اللون الأبيض إذا كانت الصورة من نوع grayscale أو الأحمر أو الأخضر أو الأزرق إذا كانت الصورة من نوع RGB .

مثال :

```
I=imread('pout.tif');  
imshow(I)  
J =imadjust(I,[0.2;0.6],[0.4;0.8]);  
figure,imshow(J)
```

والنتائج كما يلي :



ويمكن اختيار القيم [low_in;high_in] او القيم [low_out;high_out] كـ []
للدلالة على أنها تحمل القيم الافتراضية [0 1] .

مثال :

```
I=imread('pout.tif');
```

```
imshow(I)
```

```
J =imadjust(I,[0.2;0.7],[ ]);
```

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

كما يمكن الحصول على **Negative** من الصورة نجعل $Low_out > High_out$.

مثال :

```
I=imread('pout.tif');
```

```
imshow(I)
```

```
J1=imadjust(I,[ ],[1;0]);
```

```
figure,imshow(J1)
```

والنتائج كما يلي :

Original Image



Adjusted Image

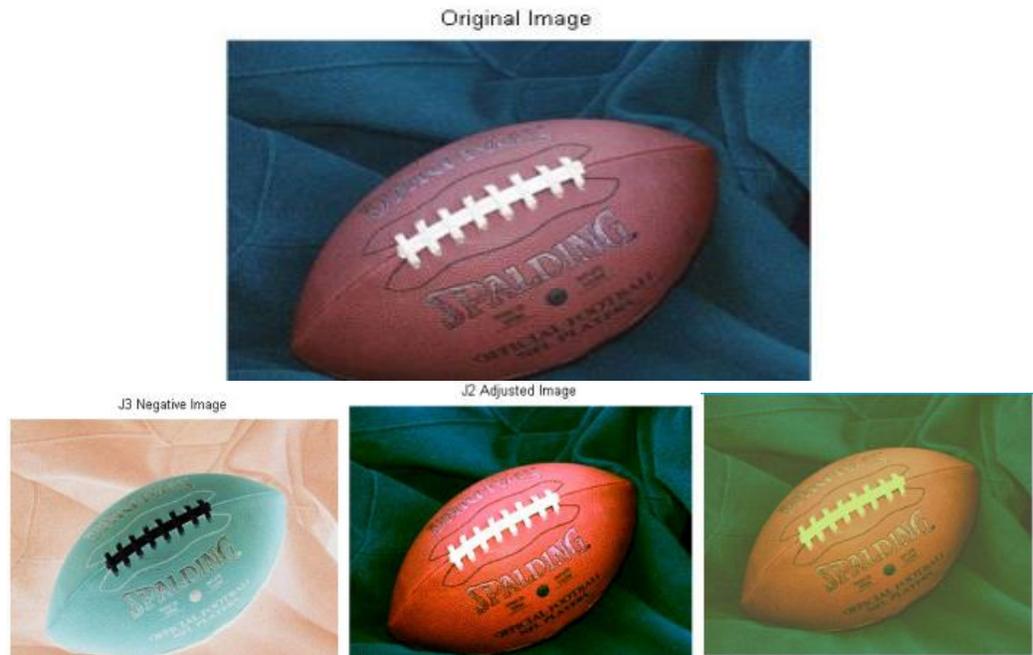


الأمر لا يختلف بالنسبة للصورة الملونة :

مثال :

```
I=imread('football.jpg');  
imshow(I)  
J1=imadjust(I,[0.2 0.1 0.3;0.6 0.8 0.7],[0.3 0.3 0.2;0.8 0.9 0.5]);  
figure,imshow(J1)  
J2=imadjust(I,[0.2 0.1 0.3;0.6 0.8 0.7],[]);  
figure,imshow(J2)  
J3=imadjust(I,[],[1 1 1;0 0 0]);  
figure,imshow(J3)
```

والنتائج كما يلي :



حذف ارضية صورة:

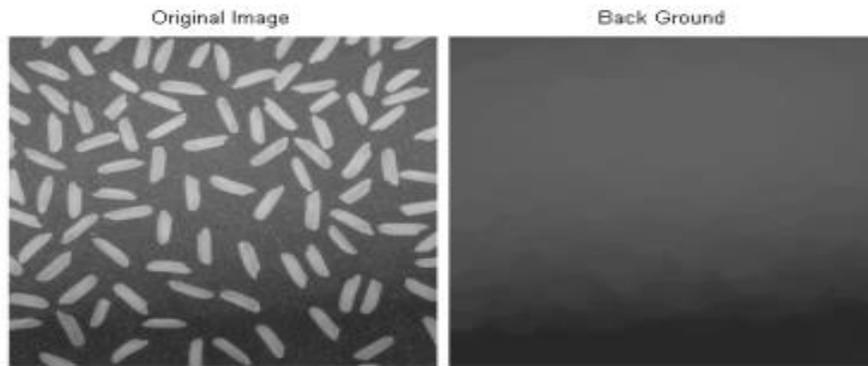
يمكن اقتطاع خلفية الصورة باستخدام التعليمة `imopen` كما في المثال التالي :

مثال :

```
I=imread('rice.png');  
imshow(I)  
background = imopen(I,strel('disk',15));  
figure, imshow(background)
```

التعليمة `strel` تنشئ مصفوفة أصفار وواحدات تكون الواحدات فيها على شكل قرص نصف قطره 15 والتعليمة `imopen` تقوم بفتح الصورة بشكل صرفي متناسب مع عنصر التركيب `strel` .

والنتائج كما يلي :



جعل ارضية صورة سوداء :

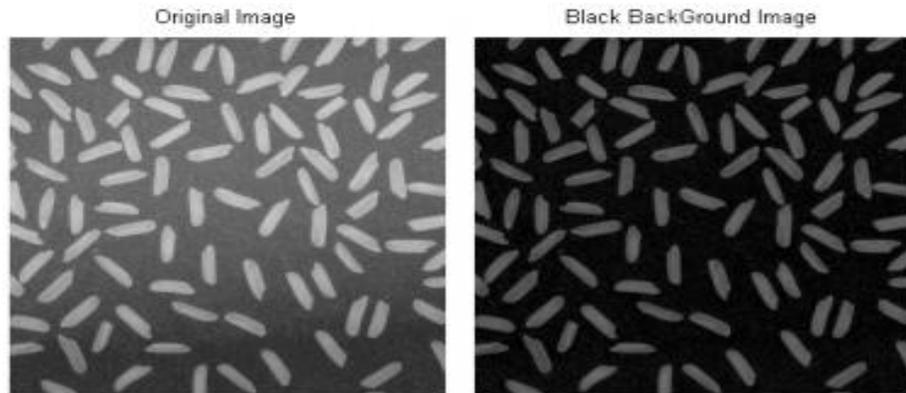
بعد اقتطاع خلفية الصورة كما في المثال في الموضوع السابق يمكن طرح الخلفية من الصورة الأصلية فنحصل على صورة فيها خلفية سوداء تماما .

مثال :

```
I=imread('rice.png');  
imshow(I)  
background = imopen(I,strel('disk',15));  
I2=imsubtract(I,background);  
figure, imshow(I2)
```

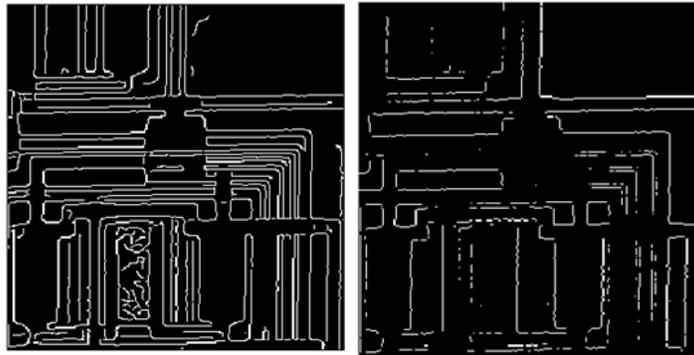
حيث تقوم التعليمة `imsubtract` بطرح الصورتين من بعضهما وستشرح هذه التعليمة لاحقاً.

والنتائج كما يلي :



ايجاد او تحديد حواف الصور:

```
%Find the edges of an image using the Prewitt and Canny methods.  
I = imread('circuit.tif');  
BW1 = edge(I,'prewitt');  
BW2 = edge(I,'canny');  
imshow(BW1);
```



Edge sharpening تحسين الحواف

```
% Edge enhancement with unsharp masking  
>>p=imread('pelicans.tif');  
>>u=fspecial('unsharp',0.5);  
>>pu=filter2(u,p);  
>>imshow(p),figure,imshow(pu/255)
```



(a) The original



(b) After unsharp masking

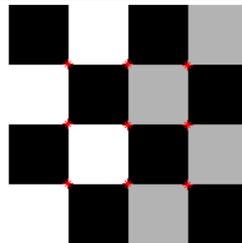
تمارين:

```
>>bc=imread('board.tif');
>>bg=im2uint8(rgb2gray(bc));
>>b=bg(100:355,50:305);
>>imshow(b)
```

```
>>m=fspecial('motion',7,0);
>>bm=imfilter(b,m);
>>imshow(bm)
```

مثال

```
%Find and plot corner points in a checkerboard image.
I = checkerboard(50,2,2);
C = corner(I);
imshow(I);
hold on
plot(C(:,1), C(:,2), 'r*');
```



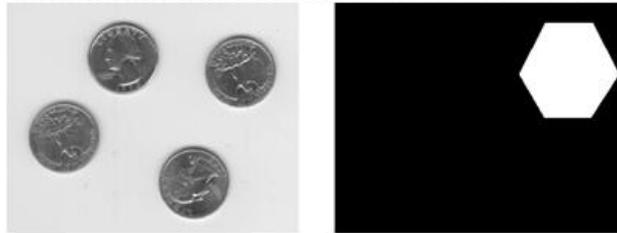
مثال

```
I = imread('eight.tif');
c = [222 272 300 270 221 194];
r = [21 21 75 121 121 75];
J = roifill(I,c,r);
imshow(I)
figure, imshow(J)
```



مثال

```
I = imread('eight.tif');
c = [222 272 300 270 221 194];
r = [21 21 75 121 121 75];
BW = roipoly(I,c,r);
figure, imshow(I)
figure, imshow(BW)
```



مثال

```
load clown
BW = roicolor(X,10,20);
imshow(X,map)
figure, imshow(BW)
```



مثال

```
figure, imshow('pout.tif');
h = imellipse;
position = wait(h);
```

مثال

```
figure, imshow('pout.tif');
h = imfreehand;
position = wait(h);
```

مثال

```
figure, imshow('pout.tif');
h = imline;
```

```
position = wait(h);
```

مثال

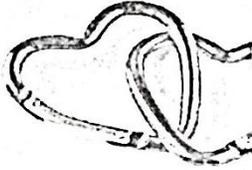
```
figure, imshow('pout.tif');
```

```
h = imrect;
```

```
position = wait(h);
```

تطبيقات عمليّة

الصورة الثانية (Picture2.jpg)



الصورة الأولى (Picture1.jpg)



• اكتب برنامج لقلب الصورة الأولى من الأعلى إلى الأسفل :

```
rgb=imread('c:/picture1.jpg','jpg');  
I=rgb2gray(rgb);  
I=flipud(I);  
imshow(I);
```



• اكتب برنامج لوضع الصورة الأولى اسفل الصورة الثانية :

```
rgb1=imread('c:/picture1.jpg','jpg');  
rgb2=imread('c:/picture2.jpg','jpg');  
I1=rgb2gray(rgb1);  
I2=rgb2gray(rgb2);  
I=[I1; I2];  
imshow(I);
```



• اكتب برنامج لوضع الصورة الأولى بجانب الصورة الثانية:

```
rgb1=imread('c:/picture1.jpg','jpg');  
rgb2=imread('c:/picture2.jpg','jpg');  
I1=rgb2gray(rgb1);  
I2=rgb2gray(rgb2);  
I=[I1 I2];  
imshow(I);
```



• اكتب برنامج لوضع النصف اليسار من الصورة الاولى بجانب النصف اليسار من الصورة الثانية:



```
rgb1=imread('c:/picture1.jpg','jpg');
rgb2=imread('c:/picture2.jpg','jpg');
I1=rgb2gray(rgb1);
I2=rgb2gray(rgb2);
[r c]=size(I1);
I=[I1(:,1:c/2) I2(:,1:c/2)];
imshow(I);
```

• اكتب برنامج لتبديل الاعمدة الفردية من الصورة الاولى مع الاعمدة الفردية من الصورة الثانية:



```
rgb1=imread('c:/picture1.jpg','jpg');
rgb2=imread('c:/picture2.jpg','jpg');
I1=rgb2gray(rgb1);
I2=rgb2gray(rgb2);
I1(:,1:2:end)=I2(:,1:2:end);
imshow(I1);
```

• اكتب برنامج لوضع الربع اليسار الاعلى من الصورة الثانية في الربع اليسار الاعلى من الصورة الاولى:



```
rgb1=imread('c:/picture1.jpg','jpg');
rgb2=imread('c:/picture2.jpg','jpg');
I1=rgb2gray(rgb1);
I2=rgb2gray(rgb2);
[r c]=size(I1);
I1(1:r/2,1:c/2)=I2(1:r/2,1:c/2);
imshow(I1);
```

عمليات احصائية للصور:

Max, std, min, sum, mean, ...

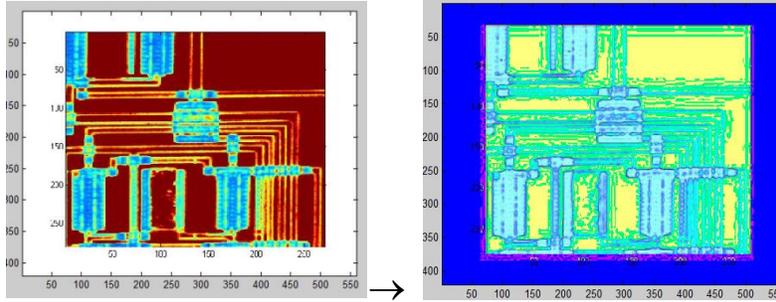
Example

```
I = imread('circuit.tif');
d=mean(I)
dd= mean(mean(I))
```

التحويل بين انظمة الالوان:

- **rgb2hsv:** Convert RGB colormap to HSV colormap

```
a=imread('1.jpg');
aa=rgb2hsv(a);
figure; image(a);
figure; image(aa);
```

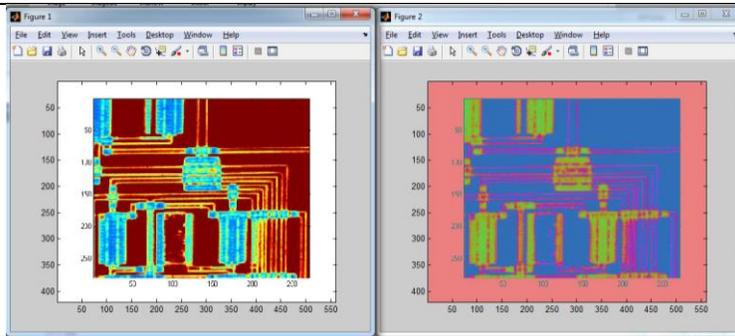


- **hsv2rgb**: Convert HSV colormap to RGB colormap.

```
a1=imread('aa.jpg');
aa= hsv2rgb (a);
figure; image(a1);
figure; image(aa);
```

- **rgb2ycbcr**: Convert RGB color values to YCbCr color space.

```
a=imread('aa.jpg');
aa= rgb2ycbcr (a);
figure; image(a);
figure; image(aa);
```



- **rgb2ntsc**: Convert RGB color values to YIQ color space.

```
RGB = imread('peppers.png');
YIQ = rgb2ntsc(RGB);
imshow(YIQ);figure
image(RGB)
```