وزارة التعليم العالي والبحث العلمي
كلية المستقبل الجامعة
قسم الفيزياء الطبية

معالجة وتحليل الصور الطبية بالماتلاب

المرحلة الرابعة – ملزمة العملي

المحتويات

مقدمة عن لغة الماتلاب

الجزء الأول : مقدمة عن الصور وانواعها بلغة الماتلاب

الجزء الثاني : كتابة الصورة ومعالجة النقطة..

الجزء الثالث : معالجة الصور والجوار من النقاط

الجزء الرابع : التعرف على حواف الصورة وايجاد الكائنات لها.

الجزء الخامس : تصنيف الصور الطبية باستخدام تقنيات متطورة.

مقدمة عن لغة الماتلاب

تعتبر ماتلاب لغةً من لغات البرمجة عالية المستوى، كما أنها أيضًا عبارةً عن بيئةٍ تفاعليةٍ يُعتمد عليها في تطوير الخوارزميات والقيام بتحليل البيانات، وهي أيضًا جزءٌ لا يتجزأ في إنشاء

التطبيقات والنماذج، وتوفر للمستخدم مجموعةً من الأدوات والوظائف الرياضية التي تساعد في إيجاد حلولٍ سريعةٍ للغاية بالاعتماد على جداول البيانات أو حتى لغات البرمجة التقليدية؛ ومن أبرزها جافا(JAVA, C++, C) ، وتزداد رقعة استخدامها بين أوساط مبرمجي أنظمة التحكم والبيولوجيا الحاسوبية وغيرها من المجالات.

تعتبر ماتلاب **MATLAB**أ ايضًا بمثابةِ مصفوفةٍ أو خوارزميةٍ جيء بها خصيصًا لغايات خلق بيئة حوسبةٍ رقميةٍ ذات نماذجٍ متعددةٍ، ويعود الفضل في تطويرها إلى **Mathworks**، وتتيح هذه اللغة عالية المستوى الفرصة في إجراء تطويرٍ وتغييرٍ على المصفوفات وطرق تخطيط البيانات وتطبيقها كخوارزميات، وتترك بصمةً واضحةً في إنشاء واجهات المستخدم وإقامة حلقة وصلٍ مع البرامج المكتوبة بلغاتٍ أخرى كلغة بايثون وفورتران جافا.

خصائص لغة الماتلاب

تطغى على لغةِ ماتلاب مجموعةً من الخصائص التي تجعلها مميزةً عن غيرها من لغات البرمجة، ومن أبرز هذه الخصائص:

- سهولة الاستخدام، حيث تتيح الفرصة لمستخدميها في الوصول إلى الحلول بالطرق الرياضية المألوفة.
- توفير أدوات ووسائل تشكل حلولًا للمشاكل التي تواجه التطبيقات وتطويرها.
- وسيلةٌ تعليميةٌ فعالةٌ وقياسية لعدةِ مجالات منها مبادئ الهندسة والرياضيات والعلوم وغيرها.
- أنموذج حقيقي لتحقيق التطوير والتقدم في البرمجيات.
- أفضل خيارٍ للاستخدام في كتابة البرامج التي تحتاج إلى نطاقٍ معتدلٍ من الأوامر والتحرير في حل المشكلات.
- اختصار أداء اللغة إجمالًا على التحكم بالأرقام وتغييرها .

نظام ماتلاب

ينقسم نظام ماتلاب إلى عدة عناصرٍ أساسية، وهي:

لغة ماتلاب(MATLAB Language)

مصفوفةٌ عالية المستوى تتلائم بطبيعتها مع عملية التحكم بتدفق البيانات وهيكلة البيانات، وتطغى عليها ميزات البرمجة الشيئية.

بيئة عمل ماتلاب (The MATLAB working environment)

يمكن الإشارة إليها بأنها ذلك النطاق الذي يتم توظيف الأدوات الخاصة بلغة الماتلاب والتي يعتمد عليها المبرمج في كتابة برنامجه لإحداث التغيير أو إنشاء المهمة الموكولة إليه؛ وتحتضن هذه البيئة عمليات استيراد البيانات وتصديرها والتحكم بالبيانات والملفات المعتمدة على اللغة.

المخططات

عبارةٌ عن نظامٍ رسوميّ متخصصٍ قائمٍ على مجموعةٍ من الأوامر الخاصة لغايات إيجاد تصورٍ للبيانات ذات أبعادٍ ثنائيةٍ وثلاثيةٍ، كما يتضمن أيضًا عملية معالجةٍ دقيقةٍ للصور والرسوم المتحركة ورسومات العرض التقديمي، ولا يستحيل وجود أوامرٍ ذات مستوى منخفض فيها إطلاقًا، حيث تكمل الأوامر المكتوبة باللغة عالية المستوى للوصول إلى النتيجة المرجوة.

مكتبة الوظائف الرياضية (The MATLAB mathematical function library)

ترتكز لغة ماتلاب على كمٍ ضخمٍ من الخوارزميات والوظائف الحسابية بدءًا من الأساسية ووصولًا إلى الأكثر تعقيدًا على الإطلاق، فيتواجد بين أبعادها كل من الجمع والجذور التربيعية والتكعيبية والظل وتمام الجيب وغيرها، بالإضافة إلى القيام بإيجاد حلولٍ سريعةٍ دون تأخرٍ.

واجهة برمجة التطبيقات (Application Program Interface)

تتيح هذه الواجهة للمبرمج بكتابة الأوامر بالاعتماد على اللغات التي تتوافق بطبيعتها مع لغة ماتلاب، ومنها لغة فورتران FORTRAN ولغةC ، وتخلق جوًا من الألفة بينهما لغايات المضي قدمًا بالتشغيل والتعديل وقراءة الملفات بكل سهولةٍ، وتشير المعلومات إلى أن هناك علاقةً ديناميكية تربط بين اللغات المذكورة الثلاث لتحفزها على أداء وظيفتها على أكمل وجهٍ.

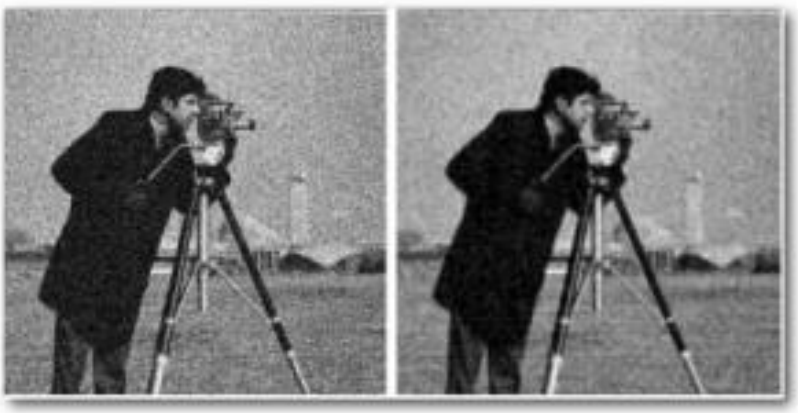
الجزء الأول : مقدمة عن الصور وانواعها بلغة الماتلاب

يستخدم الماتلاب في كثير من التطبيقات من أهمها-:

- تحليل البيانات.Data analysis
- رسم هندسي ورسم مجسمات ذات أبعاد ثلاثية 3D Dimensions
- اجراء العمليات الحسابيه بسرعه ودقه وأيضا المصفوفات.Math and Computation
- اللوغاريتمات .Algorithm.
- يستخدم في كثير من العلوم والصناعه في محاكاه وتصميم الأنظمه المختلفه Modeling and Simulation System.

معالجة الصور بالماتلاب :-

معالجه الصور الرقمية من التطبيقات التي تندرج تحت بند استخدام الماتلاب في المحاكاه وسنشرح في البداية أنواع الصور الرقمية وكيفية استخدام الماتلاب لفتح وقراءة صوره واستخدام منحني الهيستوغرام في تحسين اجراء بعض العمليات علي الصور ، انظر الى مثال الصورة ادناه



أنواع الصور الرقمية

1.الصور الثنائية Binary Images

وهي الصور التي تحتوي علي لونين فقط وهما اللون الأبيض واللون الأسود وتمثل بمصفوفه أبعادها MxN.تنقسم الصورة الي عدد من البيكسل ويأخذ قيمة اما 0 او 1 فاللون الأسود يأخد القيمه 0 واللون الأبيض يأخذ القيمه 1.

2. الصور الرمادية Grayscale Image

كما قلنا أن الصوره تمثل علي هيئه مصفوفة ثنائيه عناصرها MxN هذا النوع من الصور الرقمية تقع عناصرها بين المجال [0,255] كما قلنا ان اللون الأسود يمثل ب 0 واللون الأبيض يمثل ب255 وما بينهما درجات للون الرمادي.

3.الصور الملونة Indexed Image

تمثل بمصفوفتين تسمي احدهما مصفوفة الدليل وبعدها يكون MxN والمصفوفه الأخري تسمي colormapوابعادها Kx3 وتحتوي علي جميع الألوان المحتمل وجودها في الصورة،الثلاث أعمدة في مصفوفة Colormap تحوي علي مركبات الألوان الأحمر والأخضر والأزرق اما مصفوفة الدليل تحوي علي بيكسيلات الصورة التي تشير الي الألوان في في مصفوفه.colormap

4.الصور الحقيقية Truecolor Image

هذا النوع يتم تمثيله بمصفوفه ثلاثيه الابعاد MxNx3 وتكون عناصرها من النوعين ،هذا النوع يتم فيه دمج ثلاثة مركبات في البيكسل الواحد لاعطاء اللون المراد.

مثال: اللون الحمر

يستخرج من مصفوفه ثلاثيه الأبعاد فانها تحتوي علي قيمه اللون الاحمر التي تتراوح قيمتها في المجال [0,1] فاللون الأسود قيمته 0واللون الأحمر قيمة 255 وما بينهما هو درجات اللون الأحمر،وكذلك بالنسبة لباقي الألوان .يكون مجاله 0,255 فالأسود يمثله الرقم 0 واللون يمثله الرقم 255 وما بينهما درجات اللون.يفضل هذا النوع وذلك لتقليل مساحة الذاكرة وتسريع عملية معالجة الصور.

التحويلات بين أنواع الصور الرقمية:-

1- التحويل من Grayscle اليRGB
2- التحويل من RGB الي Grayscale باستخدام القيمة المتوسطة
3- التحويل من RGB الي Grayscale باستخدام الوزنNTSC
4- التحويل من Indexed اليRGB

5- التحويل من RGB الي Indexed باستخدام K لون

6- التحويل من Binary اليGrayscale

7- التحويل من Grayscale اليBinary

مثال علي ذلك: التحويل من RGB الي grayscale باستخدام القيمه المتوسطة باستخدام الأمر

mygray=mean(myrgb,3)

كيف يعمل هذا المثال كالاتي :-

255و255و255 255,255,255 اذا تمت عملية تحويلها من اللون الى الرمادي راح نستخدم المعدل

myrgb=imread('onion.png') هذا الايعاز يستخدم لقراء الصورة الموضح اسمها داخل الاقواس وخزنها في المتغير

imshow(myrgb); هذا الايعاز يستخدم لعرض الصورة الملونة

mygray=round(mean(myrgb,3))/255; هذا الايعاز يستخدم لقراءة معدل كل بكسل في الصورة وتقسيمه على 255 لكي يحول الى اللون الرمادي

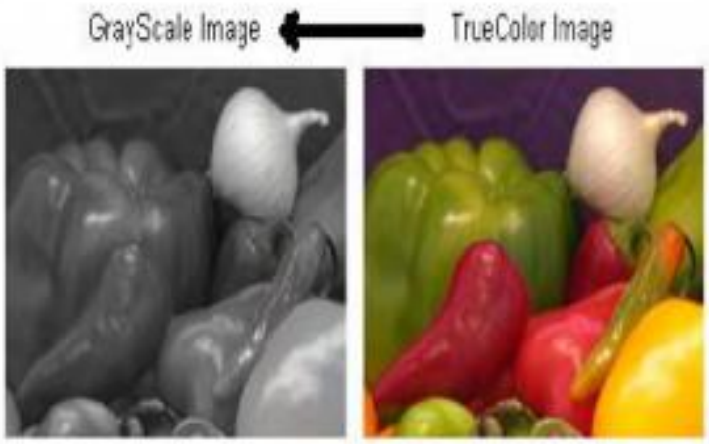
figure,imshow(mygray); عرض الصورة النهائية الرمادية

بعض الايعازات البسيطة في الماتلاب خاصة بالصور

1. **فتح الصورة من الحاسوب باستخدام الامر...**

```
x=imread ('filename.format');

Imshow(x);
```

هذا الأمر يعني قراءة الصوره من المسار filename وامتدادها format وتخزينها في المصفوفة x . وسوف تعرض صورة ال Gray وصورة RGB.



مثال: تحويل الصورة من اللون المون الحقيقي RGB الى اللون الرمادي

فتح صورة من برنامج الماتلاب نفسه وعرضها

- اذا كانت من النوع Image Indexed فيمكن قرائتها وعرضها عن طريق

```
[x,map]=imread('filename.format');
```

Imshow(x,map);

Xهي مصفوفة الدليل و map مصفوفة خارطة اللونKx3

مثال:-

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

imshow(X,map);
```

مثال 2: قراءة صورة تدعى Indexed Image

- **اما اذا كانت الصورة من بقية الأنواع ملونة من 0 الى 255 بكسل Image او ColorMap**

```
X=imread
('filename.format');

imshow(X);
```

مثال :فتح صوره من النوعRGB

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

imshow(X);
```

مثال3 : قراءة صورة ملونة من نوع الصور الحقيقة **True Color Image**

الجزء الثاني :كتابة صورة imwrite in matlab

لحفظ او طباعه الصوره علي الجهاز باسم وامتداد جديد نستخدم

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

مثال:

```
x=imread('ferrari.jpeg');
imshow(x);
imwrite(x,'newferrari.bmp');
```

هنا تم طباعة الصورة باسم newferrari وامتداد.bmp

للحصول علي معلومات عن الصورة مثل حجم الملف والعرض والإرتفاح والإمتداد ومسار الملف والألوان وهكذا عن طريق

```
info=imfinfo(filename.format);
```

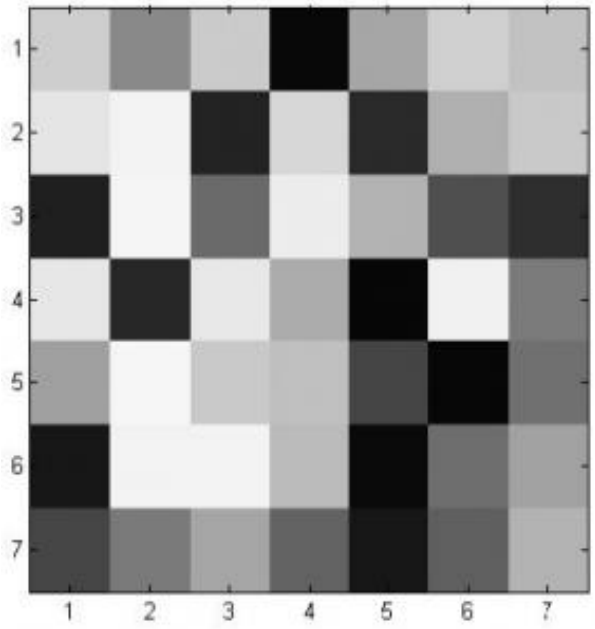
أمثلة :-

أنواع الصور

Imread (filename.format);ايعاز قراءة الصورة
Imshow (x);ايعاز عرض الصورة
Imwrite(x,y);طباعة الصورة اكس في الواي
Imfinfo(filename.format);عرض معلومات الصورة

انشاء الصور وتحويلاتها

```
mygray=rand(7,7);   هذا الايعازيستخدم لتوليد ارقام عشوائية للصورة
image(mygray*255); 255 عملية توليد صورة بضرب كل قيمة في
axis image
colormap(gray(256));
```



مثال4: طباعة صورة من نوع Grayscale Image

2. انشاء صورة من نوعTruecolor RGB

```
myrgb(:,:,1)=rand(7,7);➔ Red
myrgb(:,:,2)=rand(7,7);➔ Green
myrgb(:,:,3)=rand(7,7);➔ Blue
image(myrgb);
image(min(max(myrgb,0),1));
```

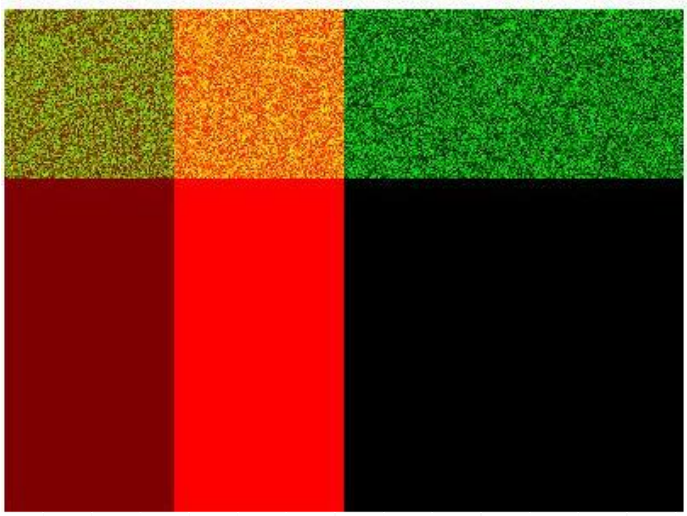
لحذف اللون الأبيض والأسود من الصورة أعلاه

Red

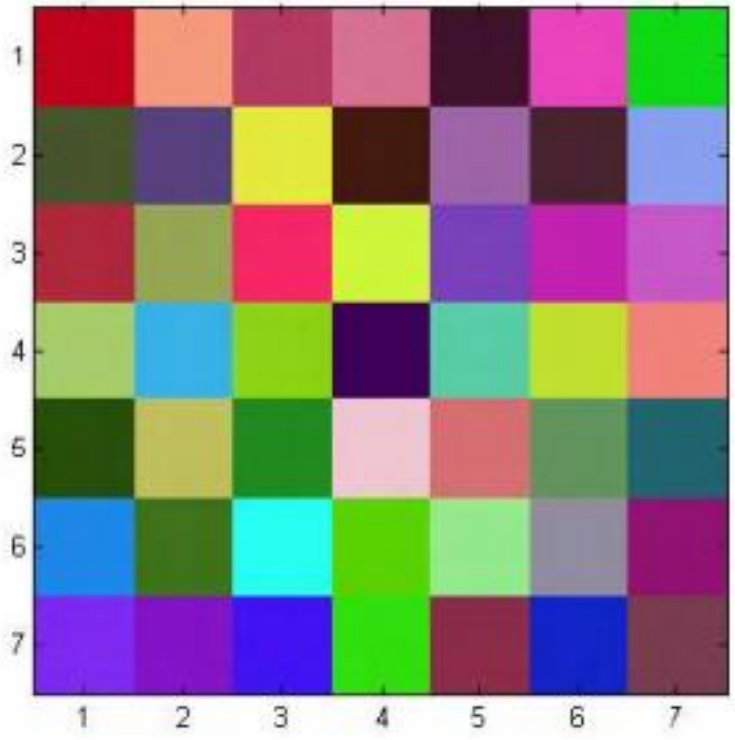
100	20	10
220	50	200
120	205	31

100	20	10
220	50	200
120	205	31

100	20	10
220	50	200
120	205	31



مثال5: طباعة صورة ملونة من نوع TrueColor Image



مثال6: طباعة صورة ملونة من نوع TrueColor Image

تحويلات الصورة

1. تحويل صورة من gray الى RGB

>> image(min(max(myrgb,0),1)); optionally

```
>> mygray=imread('pout.tif'); //(0.. 255) Gray Level
>> imshow(mygray); //Gray Level as Gray Image
>> mygray=cat(3,mygray,mygray,mygray); //show this image as RGB
>> imshow(mygray);
```

2. التحويل من صورة الرمادي الى الثنائي

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

100	78	0	54
123	250	31	200
89	78	45	56

Threshold = Sum(100+78+…….+56)/12.

Threshold= Sum(inside the mask)/ 9= k

If abs(xi- k)> D then ➔ 1 , otherwise =0

Threshold is being manually

3. الحصول علي معلومات عن الصورة-:

3.1. الصورة من الحاسب

يمكن الحصول علي معلومات عن الصورة وذلك عن طريقImfinfo

info= imfinfo **(filename,formate);**

ويمكن الحصول علي معلومات عن الصورة مثل حجم الملف والعرض والإرتفاح والإمتداد ومسار الملف والألوان وهكذا.

3.2. الصورة من برنامج الماتلاب

وذلك يتم عن طريق

info=imfinfo('cameraman','tif');

تطبيق العمليات الجبرية والمنطقية على الصور

Functions

Short-circuit &&, ||	Logical operations with short-circuiting
&	Find logical AND such as z=x & y;
~	Find logical NOT such as z= x ~ y;
|	Find logical OR
xor	Find logical exclusive-OR
all	Determine if all array elements are nonzero or `true`
any	Determine if any array elements are nonzero
false	Logical 0 (false)
find	Find indices and values of nonzero elements

islogical	Determine if input is logical array
logical	Convert numeric values to logicals
true	Logical 1 (true)

Algebra operations

```
x=imread('gray(1).png');
y=imresize(x,[100,100]);
subplot(3,1,1), imshow(y), title('Original Image');
x1=imread('gray(1).png');
y1=imresize(x1,[100,100]);
subplot(3,1,2), imshow(y1), title('second Image');
xx=imadd(y,y1);
subplot(3,1,3); imshow(xx); title('added Image');
```

also
>> Z = imsubtract(x,y);
>> Z = imdivide(x,y);
>> Z = immultiply(x,y);

الجزء الثالث : معالجة الصور ونقاط الجوار

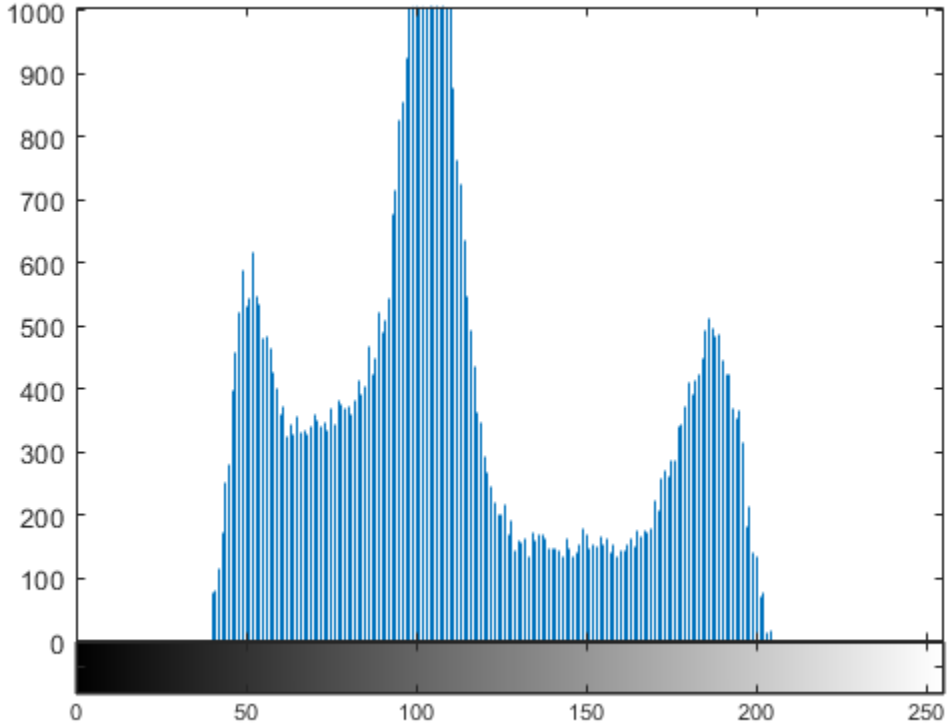
منحنى الهيستوغرام:

يوضح المنحني توزيعات الإضاءة في الصورة علي البيكسلات يتكون من محورين المحور السينات يعبر عن شدة الإضاءة والمحور الآخر الوايات يعبر عن بيكسلات الصورة وذلك باستخدام.

60	12	60	12	110
110	110	120	120	120
60	60	60	60	60
60	135	135	135	135
135	135	135	135	135

X	12	60	110	120	135
Y	2	8	3	3	9
j					

```
I = imread('rice.png');// rice.png  صورة رمادية
imshow(I)
figure;
imhist(I);// 0.. 255
```

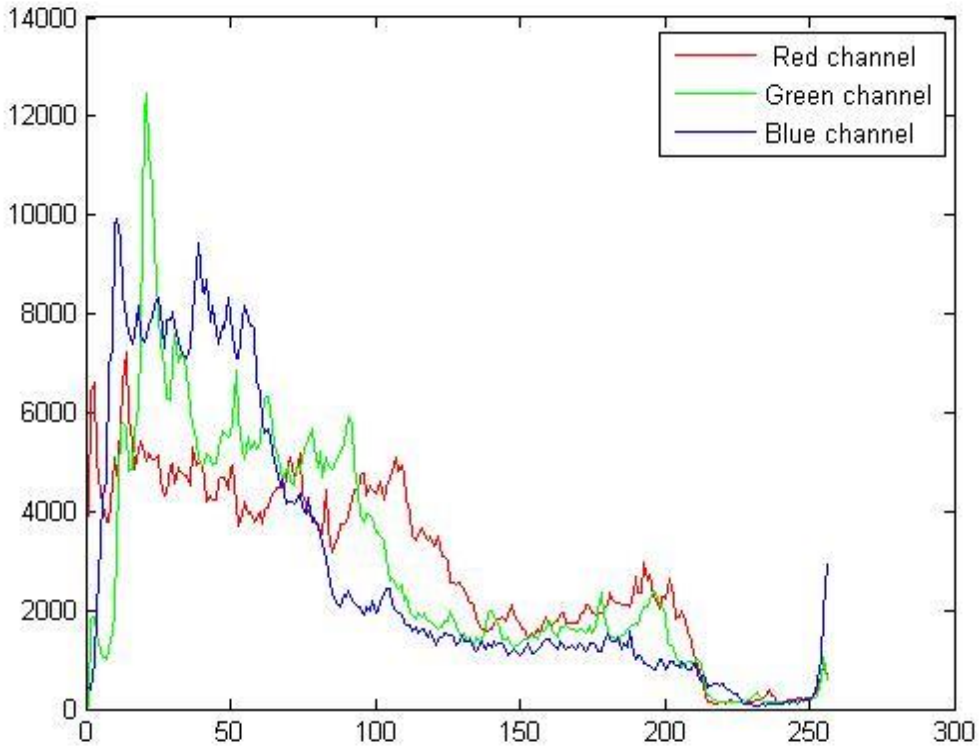


مثال7: حساب عدد تكرار الشدة اللونية لكل قيمة للصورة الرمادية GrayScale Image

حساب التكرار للصورة **RGB** في الماتلاب كالاتي

```
I=imread('1.jpg');
R=imhist(I(:,:,1));
G=imhist(I(:,:,2));
B=imhist(I(:,:,3)); figure, plot(R,'r')
hold on, plot(G,'g')
plot(B,'b'), legend(' Red channel','Green
channel','Blue channel');
hold off,
```

الصورة الناتجة للصورة الحقيقية الملونة كالاتي :-



مثال8: حساب عدد تكرار الشدة اللونية لكل قيمة للصورة الملونة RGB

تحسين تباين الصورة

يمكن تحسين تباين الوان الصورة عن طريق histeq حيث تقوم بفرد الإضاءة علي جميع بيكسلات الصورة.

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

Original Image

مثال 9: عرض الصورة الناتجة من histeq

Histogram Equalization for RGB Images

Histogram Equalization can be considered as redistribution of the intensity of the image. Color histogram equalization can be achieved by converting a color image into HSV/HSI image and enhancing the Intensity while preserving Hue and Saturation components.

However, performing histogram equalization on components of R,G and B independently will not enhance the image. At the end of this class, check the histogram of before and after histogram equalization of an image which is obtained by performing histogram equalization on the components(R,G and B) independently.

Steps to be performed:

1. Convert RGB image into HSI Image. RGB ➔ HSI == HE ➔ RGB output
2. Obtain the 'Intensity Matrix' from the HSI Image matrix

3. Perform Histogram Equalization on the intensity Matrix
4. Update the Intensity Matrix from the HSI Image matrix with the histogram equalized Intensity matrix

MATLAB CODE:

```
%COLOR HISTOGRAM EQUALIZATION

%READ THE INPUT IMAGE
I = imread('football.jpg');
%CONVERT THE RGB IMAGE INTO HSV IMAGE FORMAT
HSV = rgb2hsv(I);
%PERFORM HISTOGRAM EQUALIZATION ON INTENSITY COMPONENT
Heq = histeq(HSV(:,:,3));
HSV_mod = HSV;
HSV_mod(:,:,3) = Heq;
RGB = hsv2rgb(HSV_mod);

figure,subplot(1,2,1),imshow(I);title('Before Histogram Equalization');

       subplot(1,2,2),imshow(RGB);title('After Histogram Equalization');
```

EXPLANATION:

RGB image matrix is converted into HSI(Hue ,Saturation and Intensity) format and histogram equalization is applied only on the Intensity matrix . The Hue and Saturation matrix remains the same. The updated HSI image matrix is converted back to RGB image matrix.

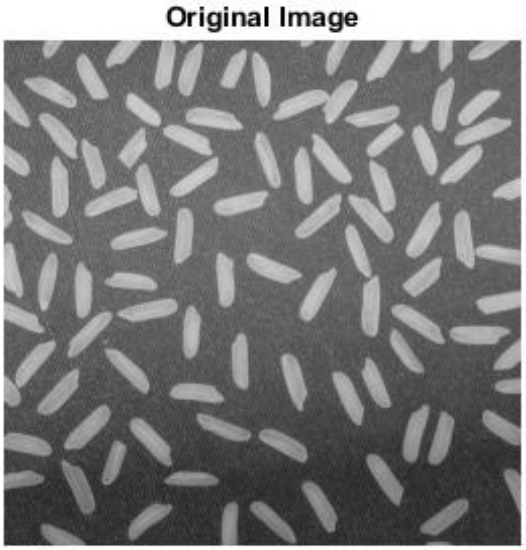
```
%DISPLAY THE HISTOGRAM OF THE ORIGINAL AND THE EQUALIZED IMAGE
HIST_IN = zeros([256 3]);
HIST_OUT = zeros([256 3]);
%HISTOGRAM OF THE RED,GREEN AND BLUE COMPONENTS
HIST_IN(:,1) = imhist(I(:,:,1),256); %RED
HIST_IN(:,2) = imhist(I(:,:,2),256); %GREEN
HIST_IN(:,3) = imhist(I(:,:,3),256); %BLUE

HIST_OUT(:,1) = imhist(RGB(:,:,1),256); %RED
HIST_OUT(:,2) = imhist(RGB(:,:,2),256); %GREEN
HIST_OUT(:,3) = imhist(RGB(:,:,3),256); %BLUE

mymap=[1 0 0; 0.2 1 0; 0 0.2 1];

figure,subplot(1,2,1),bar(HIST_IN);colormap(mymap);legend('RED
CHANNEL','GREEN CHANNEL','BLUE CHANNEL');title('Before Applying Histogram
Equalization');
       subplot(1,2,2),bar(HIST_OUT);colormap(mymap);legend('RED
CHANNEL','GREEN CHANNEL','BLUE CHANNEL');title('After Applying Histogram
Equalization');
```

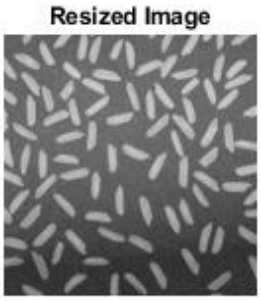
تكبير و تصغير حجم الصورة:

يمكن التحكم في حجم الصورة ثنائية الأبعاد وذلك من خلال الأمر.Imresize

مثال

```
I = imread('pout.jpg');//  قراءة الصورة الاصلية
J = imresize(I, 0.5);//  كل قيمة في الصورة تضرب
 في معامل النصف لتقليل حجمها
Figure/  ولد شكل معين
imshow(I)//  ارسم الصورة
title('Original Image') اسمها الصورة الاصلية
figure / ولد شكل
imshow(J)//  ارسم الصورة بعد التصغير او التكبير
title('Resized Image')// الصورة الناتجة
```





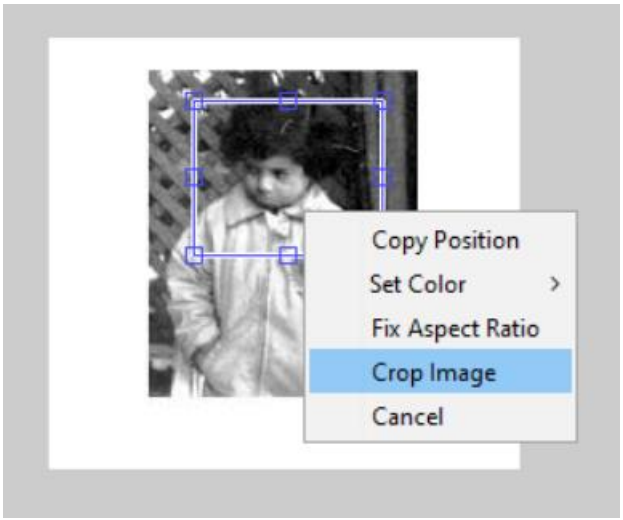
مثال 10: عرض الصورة الناتجة بعد التصغير

اقتطاع جزء من الصورة cropping

لإقتطاع جزء من الصورة وإنشاء صورة جديدة من الجزء المقطوع من خلال Imcrop.

```
I = imread('pout.jpg'); قراءة الصورة
[J,rect] = imcrop(I); قطع جزء من الصورة
```

الايعاز أعلاه يسمح باقتطاع جزء من الصورة من خلال الفأرة او المؤشر .



مثال 11: قطع جزء من الصورة عن طريق الايعاز أعلاه

لاقتطاع جزء من الصور عن طريق تحديد مستطيل او مربع من الصورة نستخدم الاتي

```
I = imread('pout.jpg');
I2 = imcrop(I,[75 68 130 112]);//  قطع جزء من
الصورة حسب القيم المحددة
subplot(1,2,1)/ عرض الصورة الاصلية
imshow(I)
title('Original Image')
subplot(1,2,2)/  عرض الصورة المقطوعة
imshow(I2)
title('Cropped Image')
```

قلب الصورة:

يمكننا أن نقوم بقلب الصورة من اليمين لليسار (لا ننسى أن الصورة عبارة عن مصفوفة)

أي وكأن الصورة وضعت أمام مرآة من خلال.Flipdim

يمكن قلب الصورة باتجاه المحاور الافقية والعمودية وكلاهما باستخدام الايعاز flipdim

```
I = imread('pout.jpg');
I2 = flipdim(I ,2);              %# horizontal flip
 قلب الصورة افقيا
I3 = flipdim(I ,1);              %# vertical flip قلب
 الصورة عموديا
I4 = flipdim(I3,2);     %# horizontal+vertical
flip قلب الصورة عموديا وافقيا
subplot(2,2,1), imshow(I) عرض الصورة الاصلية
subplot(2,2,2), imshow(I2) عرض الصورة مقلوبة افقيا
subplot(2,2,3), imshow(I3) عرض الصورة مقلوبة عموديا
subplot(2,2,4), imshow(I4) عرض الصورة مقلوبة افقيا
 وعموديا
```

مثال 12: قلب الصورة باتجاهات متعددة

تصميم مرشحات:

تصميم مرشح ثنائي البعد من أحد الأنواع الشهيرة مثل مرشح غاوص أو لابلاسيان أو اللوغاريتمي او المتوسط من خلالFspecial وتستخدم هذه المرشحات لتعديل الصورة المضببة والصور الغير واضحة . الجدول رقم (1) يمثل اهم المرشحات في لغة ماتلاب المستخدمة مع أنواع مختلفة من الصور الطبية وغيرها.

جدول رقم 1: أنواع المرشحات في لغة ماتلاب

Value	Description
'average'	Averaging filter
'disk'	Gaussian lowpass filter
'gaussian'	Gaussian lowpass filter
'laplacian'	Filter approximating the two-dimensional Laplacian operator
'log'	Laplacian of Gaussian filter
'motion'	Prewitt horizontal edge-emphasizing filter
'prewitt'	Prewitt horizontal edge-emphasizing filter
'sobel'	Sobel horizontal edge-emphasizing filter
'unsharp'	Unsharp contrast enhancement filter

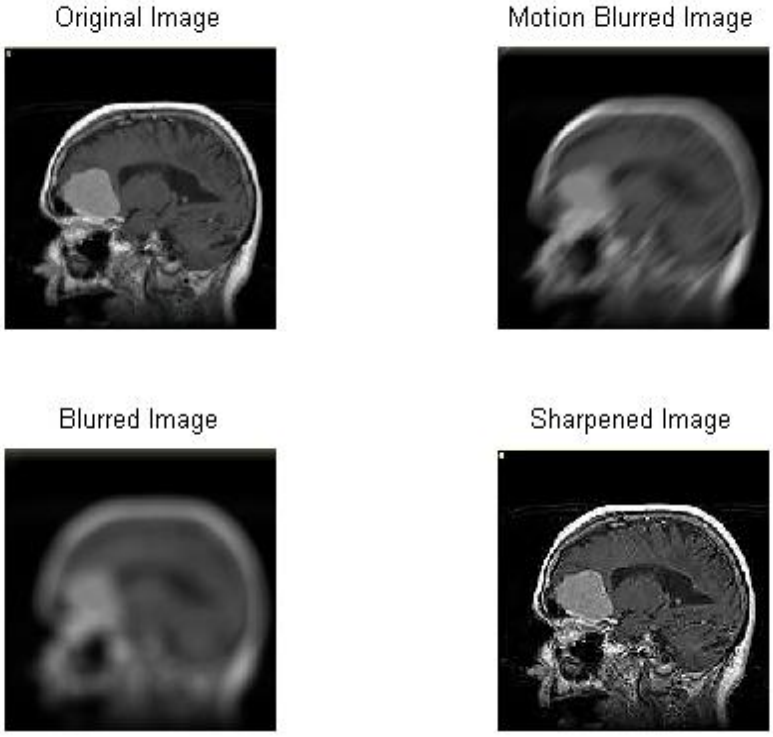
الصيغة المشهورة للمرشح في لغة الماتلاب كالاتي :-

h=fspecial(type)حيث ان type تمثل نوع الفلتر من الجدول أعلاه .

مثال 13: لدينا الصورة الطبية الاتية تم عمل عليها الفلتر motion خاص بالتأكيد على الحافة وفلتر غاوص ومرشح تحسين التباين

```
I = imread('3.jpg'); قراءة الصورة
subplot(2,2,1);  وضع اطار لها للرسم
imshow(I); title('Original Image'); عرض الصورة
 باسم الصورة الاصلية

H = fspecial('motion',20,45); تطبيق المرشح الأول
```

```
MotionBlur = imfilter(I,H,'replicate'); تطبيق
 الفلتر عليها
subplot(2,2,2);  وضع اطار رسم ثاني
imshow(MotionBlur);title('Motion Blurred
Image'); عرض الصورة

H = fspecial('disk',10); تطبيق المرشح الثاني
blurred = imfilter(I,H,'replicate');
subplot(2,2,3);
imshow(blurred); title('Blurred Image');

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



مثال 13: تطبيق مرشحات مختلفة على الصور الطبية

إضافة ضجيج Image Noise:

وتوجد أنواع Imnoise إضافة ضجيج للصورة من خلال مختلفة من الضجيج الذي يمكن ان يؤثر في الصورة ، جدول رقم 2 يمثل اهم أنواع الضجيج في الصورة في لغة الماتلاب (للاطلاع فقط)

Syntax

```
J = imnoise(I,'gaussian')
J = imnoise(I,'gaussian',m)
J = imnoise(I,'gaussian',m,var_gauss)

J = imnoise(I,'poisson')
J = imnoise(I,'salt & pepper')
J = imnoise(I,'salt & pepper',d)
J = imnoise(I,'speckle')
J = imnoise(I,'speckle',var_speckle)
```

Description

`J = imnoise(I,'gaussian')` adds zero-mean, Gaussian white noise with variance of 0.01 to grayscale image `I`.

`J = imnoise(I,'gaussian',m)` adds Gaussian white noise with mean `m` and variance of 0.01.

`J = imnoise(I,'gaussian',m,var_gauss)` adds Gaussian white noise with mean `m` and variance `var_gauss`.

`J = imnoise(I,'localvar',var_local)` adds zero-mean, Gaussian white noise of local variance `var_local`.

`J = imnoise(I,'localvar',intensity_map,var_local)` adds zero-mean, Gaussian white noise. The local variance of the noise, `var_local`, is a function of the image intensity values in `I`. The mapping of image intensity value to noise variance is specified by the vector `intensity_map`.

`J = imnoise(I,'poisson')` generates Poisson noise from the data instead of adding artificial noise to the data. See Algorithms for more information.

`J = imnoise(I,'salt & pepper')` adds salt and pepper noise, with default noise density 0.05. This affects approximately 5% of pixels.

`J = imnoise(I,'salt & pepper',d)` adds salt and pepper noise, where `d` is the noise density. This affects approximately `d*numel(I)` pixels.

`J = imnoise(I,'speckle')` adds multiplicative noise using the equation `J = I+n*I`, where `n` is uniformly distributed random noise with mean 0 and variance 0.05.

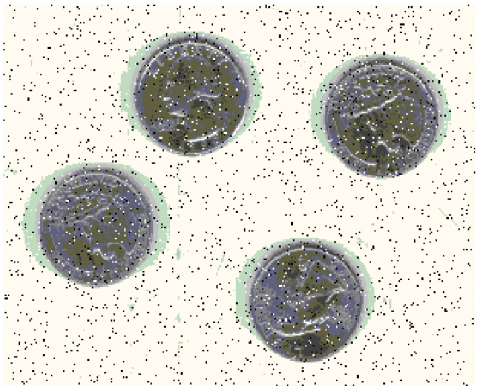
المثال التالي يوضح إضافة ضجيج للصورة من نوع الملح والفلفل

Salt and pepper

مثال 13: إضافة ضجيج للصورة من نوع الملح والفلفل

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

الصورة الاصلية



الصورة بعد إضافة الضوضاء

الصورة بعد إضافة الضجيج

ارجاع الصورة الاصلية بعد إزالة الضجيج

استخدام المعدل

Filter the noisy image, J, with an averaging filter and display the results. The example uses a 3-by-3 neighborhood.

```
Kaverage = filter2(fspecial('average',3),J)/255;
figure
imshow(Kaverage)
```

استخدام المنوال

Now use a median filter to filter the noisy image, `J`. The example also uses a 3-by-3 neighborhood. Display the two filtered images side-by-side for comparison. Notice that `medfilt2` does a better job of removing noise, with less blurring of edges of the coins.

```
Kmedian = medfilt2(J);
imshowpair(Kaverage,Kmedian,'montage')
```

إضافة الضجيج على الصورة الملونة

```
RGB = imread('saturn.png');
```

```
imshow(J(600:1000,1:600));
title('Portion of the Image with Added Gaussian Noise');
K = wiener2(J,[5 5]);
```

```
figure
imshow(K(600:1000,1:600));
title('Portion of the Image with Noise Removed by Wiener Filter');
```

محاضرة العملي لمادة Fast Fourier Transform FFT

Fast fourier and inverse fast fourier FFT and IFFT

Image is found in spatial domain : 2D array

Each fourier and transform is used to convert the image to ferquency domain

```
f=imread('1.jpg');

b=rgb2gray(f);

[r,c]=size(b);

a=imresize(b, [r r ]);

Whos a

g=fspecial('gaussian',r,10);

max(g(:));

g1=mat2gray(g);
```

```
max(g1(:));
v=fft2(a);
vf=fftshift(v);
ag1i=vf.* g1;fftshow(ag1);
ag1i=ifft2(ag1); ifftshow(ag1i);
```

قسم الفيزياء الطبية

نستخدم في هذه الدرس كيفية إيجاد حدية الصورة وتوضيح معالمها باستخدام ايعازين هما

imsharpen

Sharpen image using unsharp masking

Syntax

```
B = imsharpen(A) الايعاز الأول
B = imsharpen(A,Name,Value) الايعاز الثاني
```

Description

B = imsharpen(A) sharpens the grayscale or truecolor (RGB) input image A by using the unsharp masking method.

يستخدم الايعاز الأول لايجاد حدية للصورة سواء كانت ملونة او رمادية وتوضيحها اكثر ابراز الحواف الخاصة بها ...

B = imsharpen(A,Name,Value) uses name-value pairs to control aspects of the unsharp masking.

يستخدم هذا الايعاز الثاني لايجاد حدية اكثر مع وجود ايعازات **للسيطرة على الحدية** وهي كل من الاسم والقيمة يستخدم الاسم لايجاد حدية حول نقاط الصورة الملونة بدرجة معينة والقيمة قيمة الحدية التي سوف تستخدم للتوضيح

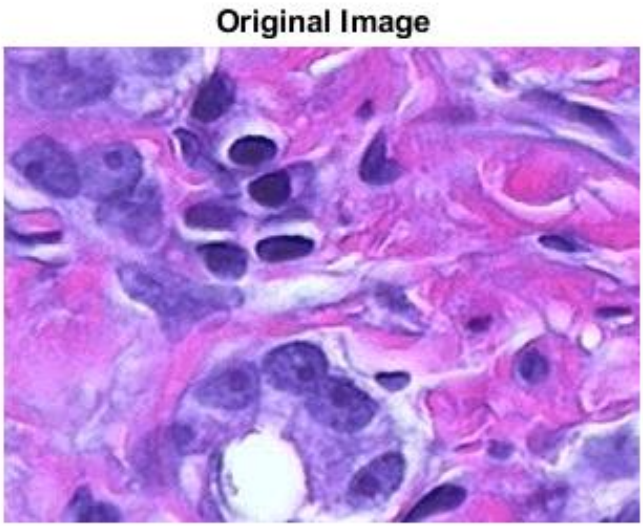
Examples

Sharpen Image

Read an image into the workspace and display it.

الايعازات ادناه لعرض وقراءة الصورة

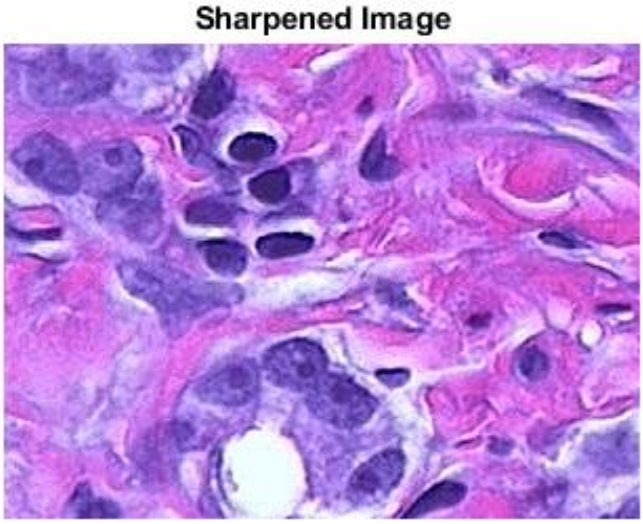
```
a = imread('hestain.png');
imshow(a)
title('Original Image');
```



Sharpen the image using the `imsharpen` function and display it.

الايعازات ادناه لعرض حدية اكثر للصورة

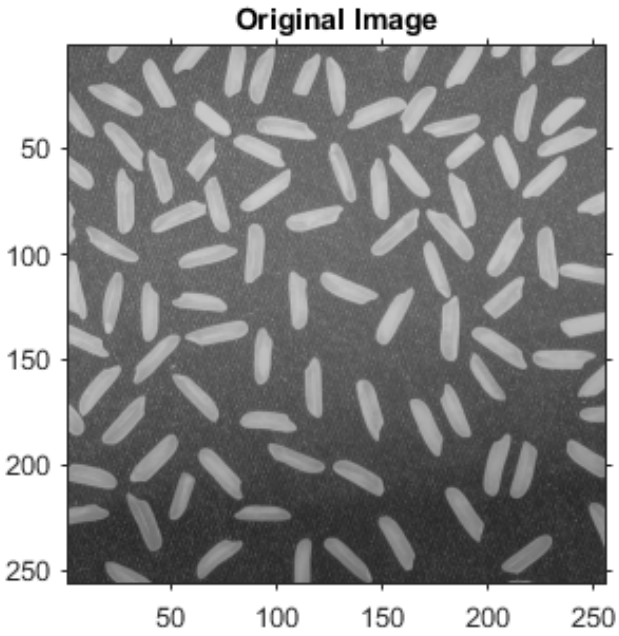
```
b = imsharpen(a); تطبيق حدية للصورة عن طريق هذا الايعاز
figure, imshow(b)
title('Sharpened Image');
```



قسم الفيزياء الطبية Lab#

Control the Amount of Sharpening at the Edges

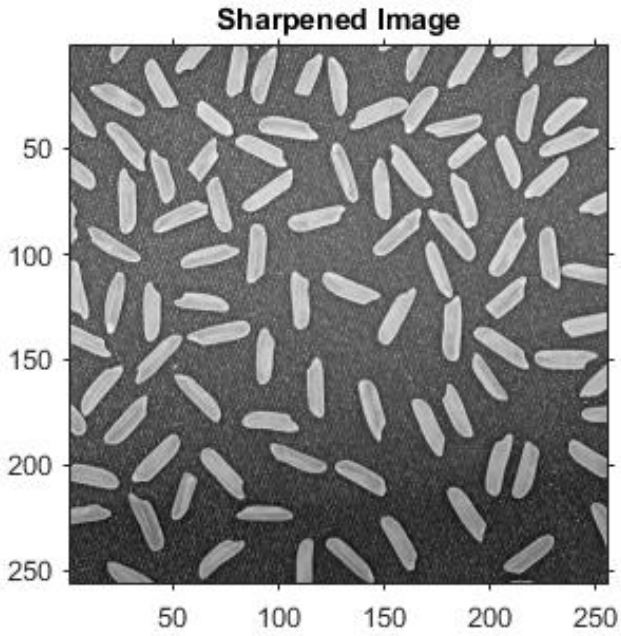
Read an image into the workspace and display it.



Sharpen image, specifying the `radius` and `amount` parameters.

```
b = imsharpen(a,'Radius',2,'Amount',1);
figure, imshow(b)
title('Sharpened Image');
```

يظهر هذا الايعاز حدية الصورة حول الحواف الناتجة باستخدام



Input Arguments

collapse all

A — Image to be sharpened
grayscale image | RGB image

Image to be sharpened, specified as a grayscale or RGB image.

Name-Value Pair Arguments

Specify optional comma-separated pairs of `Name,Value` arguments. `Name` is the argument name and `Value` is the corresponding value. `Name` must appear inside quotes. You can specify several name and value pair arguments in any order as `Name1,Value1,...,NameN,ValueN`.

نستخدم قيم لوصوف الاسم والقيمة اعلان مثلا استخدام طريقة النصف القطر وبقيمة

Example: `'Radius',1.5`

`'Radius'` — Standard deviation of the Gaussian lowpass filter
1 (default) | positive number

Standard deviation of the Gaussian lowpass filter, specified as a positive number. This value controls the size of the region around the edge pixels that is affected by sharpening. A large value sharpens wider regions around the edges, whereas a small value sharpens narrower regions around edges.

الانحراف المعياري لمرشح Gaussian lowpass ، المحدد كرقم موجب. تتحكم هذه القيمة في حجم المنطقة حول بكسلات الحواف المتأثرة بالحدية. تعمل القيمة الكبيرة على زيادة وضوح المناطق الأوسع حول الحواف ، بينما تعمل القيمة الصغيرة على زيادة حدة المناطق الأضيق حول الحواف.

Example: `'Radius',1.5`

`'Amount'` — Strength of the sharpening effect
`0.8` (default) | numeric scalar

Strength of the sharpening effect, specified as a numeric scalar. A higher value leads to larger increase in the contrast of the sharpened pixels. Typical values for this parameter are within the range `[0 2]`, although values greater than 2 are allowed. Very large values for this parameter may create undesirable effects in the output image.

Example: `'Amount',1.2`

قوة تأثير التوضيح ، المحددة كسلسلة رقمية. تؤدي القيمة الأعلى إلى زيادة أكبر في تباين وحدات البكسل الحادة. تقع القيم النموذجية لهذه المعلمة ضمن النطاق [0 2] ، على الرغم من السماح بالقيم الأكبر من 2. قد تؤدي القيم الكبيرة جدًا لهذه المعلمة إلى إنشاء تأثيرات غير مرغوب فيها في صورة الإخراج.

`'Threshold'` — Minimum contrast required for a pixel to be considered an edge pixel
`0` (default) | scalar in the range [0 1]

Minimum contrast required for a pixel to be considered an edge pixel, specified as a scalar in the range [0 1]. Higher values (closer to 1) allow sharpening only in high-contrast regions, such as strong edges, while leaving low-contrast regions unaffected. Lower values (closer to 0) additionally allow sharpening in relatively smoother regions of the image. This parameter is useful in avoiding sharpening noise in the output image.

Example: `'Threshold',0.7`

"الحد الأدنى" - الحد الأدنى من التباين المطلوب حتى يتم اعتبار البكسل بكسل حافة

0 (افتراضي) | سلمي في النطاق [0 1] الحد الأدنى من التباين المطلوب حتى يتم اعتبار البكسل بكسل حافة ، محددًا كسلسلة في النطاق [0 1]. تسمح القيم الأعلى (الأقرب إلى 1) بالوضوح فقط في المناطق عالية التباين ، مثل الحواف القوية ، مع ترك المناطق منخفضة التباين غير متأثرة. تسمح القيم المنخفضة (الأقرب إلى 0) أيضًا بالتوضيح في مناطق أكثر نعومة نسبيًا من الصورة. هذه المعلمة مفيدة في تجنب زيادة حدة التشويش في صورة الإخراج.

extractHOGFeatures

Extract histogram of oriented gradients (HOG) features

استخراج الرسم البياني لميزات التدرجات الموجهة (HOG)

Syntax

```
features = extractHOGFeatures(I)
```

```
[features,validPoints] = extractHOGFeatures(I,points)
[___, visualization] = extractHOGFeatures(I,___)
[___] = extractHOGFeatures(___,Name,Value)
```

Description

`features = extractHOGFeatures(I)` returns extracted HOG features from a truecolor or grayscale input image, `I`. The features are returned in a 1-by-*N* vector, where *N* is the HOG feature length. The returned features encode local shape information from regions within an image. You can use this information for many tasks including classification, detection, and tracking.

features = extractHOGFeatures (I)تُرجع ميزات HOG المستخرجة من صورة إدخال truecolor أو تدرج الرمادي ، .Iيتم إرجاع الميزات في متجه 1 N-by-، حيث N هو طول ميزة HOG. تقوم المعالم التي تم إرجاعها بترميز معلومات الشكل المحلي من مناطق داخل الصورة. يمكنك استخدام هذه المعلومات للعديد من المهام بما في ذلك التصنيف والكشف والتتبع.

`[features,validPoints] = extractHOGFeatures(I,points)` returns HOG features extracted around specified point locations. The function also returns `validPoints`, which contains the input point locations whose surrounding region is fully contained within `I`. Scale information associated with the points is ignored.

[features ،validPoints] = extractHOGFeatures (I ،Points)

ترجع ميزات HOG المستخرجة حول مواقع النقاط المحددة. تقوم الوظيفة أيضًا بإرجاع نقاط صالحة ، والتي تحتوي على مواقع نقاط الإدخال التي تكون المنطقة المحيطة بها متضمنة بالكامل في .I يتم تجاهل معلومات المقياس المرتبطة بالنقاط.

`[___, visualization] = extractHOGFeatures(I,___)` optionally returns a HOG feature visualization, using any of the preceding syntaxes. You can display this visualization using `plot(visualization)`.

`[___] = extractHOGFeatures(___,Name,Value)` uses additional options specified by one or more Name,Value pair arguments, using any of the preceding syntaxes.

[___ ،visualization] = extractHOGFeatures (I ،___) يعرض اختياريًا تصورًا لميزة

يقوم اختياريًا بإرجاع تصور ميزة HOG ، باستخدام أي من التركيبات السابقة. يمكنك عرض هذا التصور باستخدام الحبكة (التصور). [___ = [___ extractHOGFeatures، الاسم ، القيمة) يستخدم خيارات إضافية محددة بواسطة واحد أو أكثر من وسيطات زوج القيمة ، باستخدام أي من التركيبات السابقة.

Examples

Read the image of interest. قراءة الصورة

```
img = imread('cameraman.tif'); ايعاز قراءة الصورة
```

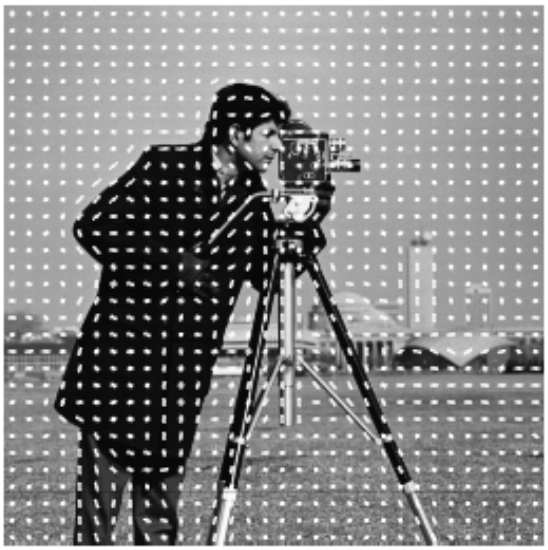
Extract HOG features. استخراج الرسم البياني لميزات التدرجات الموجهة

```
[featureVector,hogVisualization] = extractHOGFeatures(img);
```

Plot HOG features over the original image. ارسم الناتج

```
figure;
imshow(img);
```

```
hold on;
plot(hogVisualization); ايعاز عرض ورسم الصورة
```



Extract HOG Features using CellSize

قراءة الصورة

Read the image of interest.

```
I1 = imread('gantrycrane.png');
```

Extract HOG features.

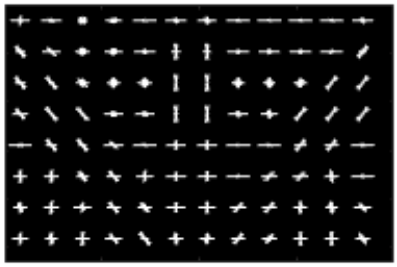
```
[hog1,visualization] = extractHOGFeatures(I1,'CellSize',[32 32]);
```

استخراج الرسم البياني لميزات التدرجات الموجهة

Display the original image and the HOG features. اعرض الصورة

```
subplot(1,2,1);
imshow(I1);
subplot(1,2,2);
```

```
plot(visualization);
```



Extract HOG Features Around Corner Points استخراج موجهات التدرج حول نقاط الحافات

```
I2 = imread('gantrycrane.png'); قراءة الصورة
```

Detect and select the strongest corners in the image.

```
corners   = detectFASTFeatures(im2gray(I2)); استخراج وتعين الحواف الحادة للصورة
strongest = selectStrongest(corners,3); استخراج الحافات
```

Extract HOG features.

```
[hog2,validPoints,ptVis] =
extractHOGFeatures(I2,strongest); استخراج الرسم البياني لميزات التدرجات
```

Display the original image with an overlay of HOG features around the strongest corners.

عرض الصورة مع الحافات القوية

```
figure;
imshow(I2);
hold on;
plot(ptVis,'Color','green');
```

Input Arguments معاملات الادخال في الايعازات أعلاه

`I` — Input image الصورة المدخلة

***M*-by-*N*-by-3 truecolor image | *M*-by-*N* 2-D grayscale image**

Input image, specified in either *M*-by-*N*-by-3 truecolor or *M*-by-*N* 2-D grayscale. The input image must be a real, nonsparse value. If you have tightly cropped images, you may lose shape information that the HOG function can encode. You can avoid losing this information by including an extra margin of pixels around the patch that contains background pixels.

M-by-N-3-truecolor image | M-by-N 2-D صورة بتدرج الرمادي
صورة الإدخال ، المحددة إما في M-by-N-by-3 truecolor أو M-by-N 2-D grayscale. يجب أن تكون صورة الإدخال قيمة حقيقية غير متفرقة. إذا قمت بقص الصور بإحكام ، فقد تفقد معلومات الشكل التي يمكن لوظيفة HOG ترميزها. يمكنك تجنب فقد هذه

المعلومات بتضمين هامش إضافي من البكسل حول التصحيح الذي يحتوي على بكسلات الخلفية.

points — Center location point
BRISKPoints object | cornerPoints object | SURFPoints objec t | MSERRegions object | *M*-by-2 matrix of [*x*, *y*] coordinates

Center location point of a square neighborhood, specified as either
a BRISKPoints, SURFPoints, MSERRegions, ORBPoints or corne rPoints object, or an *M*-by-2 matrix of *M* number of [*x*, *y*] coordinates. The function extracts descriptors from the neighborhoods that are fully contained within the image boundary. You can set the size of the neighborhood with the BlockSize parameter. Only neighborhoods fully contained within the image are used to determine the valid output points. The function ignores scale information associated with these points.

نقطة موقع مركزية لنقطة مجاورة، محدد إما كنقاط BRISK ، أو نقاط SURFP ، أو MSERRegions ، أو ORBPoints أو كائن نقاط الزاوية ، أو مصفوفة M في 2 لعدد M من إحداثيات [x، y]. تستخرج الوظيفة الواصفات من الأحياء المضمنة بالكامل داخل حدود الصورة. يمكنك تعيين حجم الحي باستخدام معلمة BlockSize. يتم استخدام الأحياء المتضمنة بالكامل في الصورة فقط لتحديد نقاط الإخراج الصالحة. تتجاهل الوظيفة معلومات المقياس المرتبطة بهذه النقاط.

Name-Value Pair Arguments

Specify optional comma-separated pairs of Name,Value arguments. Name is the argument name

and `Value` is the corresponding value. `Name` must appear inside quotes. You can specify several name and value pair arguments in any order as `Name1,Value1,...,NameN,ValueN`.

Example: `'BlockSize',[2 2]` sets the `BlockSize` to be a 2-by-2 square block.

`'CellSize'` — Size of HOG cell
`[8 8]` (default) | 2-element vector

Size of HOG cell, specified in pixels as a 2-element vector. To capture large-scale spatial information, increase the cell size. When you increase the cell size, you may lose small-scale detail.

`'BlockSize'` — Number of cells in block
`[2 2]` (default) | 2-element vector

Number of cells in a block, specified as a 2-element vector. A large block size value reduces the ability to suppress local illumination changes. Because of the number of pixels in a large block, these changes may get lost with averaging. Reducing the block size helps to capture the significance of local pixels. Smaller block size can help suppress illumination changes of HOG features.

وسيطات زوج الاسم والقيمة

حدد أزواجًا اختيارية مفصولة بفواصل من وسيطات الاسم والقيمة. الاسم هو اسم الوسيطة والقيمة هي القيمة المقابلة. يجب أن يظهر الاسم داخل علامات الاقتباس. يمكنك تحديد عدة وسيطات لزوج الاسم والقيمة بأي ترتيب مثل Name1 و Value1 و ... و NameN و ValueN.

مثال: " "BlockSize، [2 2] يعين BlockSize ليكون كتلة مربعة 2 × 2.

"حجم الخلية" - حجم خلية HOG

[8 8] (افتراضي) | 2-عنصر ناقل

حجم خلية HOG ، المحدد بالبكسل كمتجه ثنائي العنصر. لالتقاط معلومات مكانية واسعة النطاق ، قم بزيادة حجم الخلية. عند زيادة حجم الخلية ، قد تفقد التفاصيل الصغيرة الحجم.

" - "BlockSizeعدد الخلايا في الكتلة

[2 2] (افتراضي) | 2-عنصر ناقل

عدد الخلايا في الكتلة ، المحدد كمتجه ثنائي العنصر. تقلل قيمة حجم الكتلة الكبيرة من القدرة على قمع تغييرات الإضاءة المحلية. نظرًا لعدد البكسل في كتلة كبيرة ، فقد تضيع هذه التغييرات مع المتوسط. يساعد تقليل حجم الكتلة على التقاط أهمية وحدات البكسل المحلية. يمكن أن يساعد حجم الكتلة الأصغر في قمع تغييرات الإضاءة لميزات .HOG

قسم الفيزياء الطبية- الفصل الثاني تحليل الصور

نستخدم في هذه الدرس كيفية إيجاد حدية الصورة وتوضيح معالمها باستخدام ايعازين هما

الدرس الاول 8-3-2022 : حدية الصورة

imsharpen

Sharpen image using unsharp masking

Syntax

```
B = imsharpen(A) الايعاز الأول
B = imsharpen(A,Name,Value) الايعاز الثاني
```

Description

B = imsharpen(A) sharpens the grayscale or truecolor (RGB) input image A by using the unsharp masking method.

يستخدم الايعاز الأول لايجاد حدية للصورة سواء كانت ملونة او رمادية وتوضيحها اكثر

B = imsharpen(A,Name,Value) uses name-value pairs to control aspects of the unsharp masking.

يستخدم هذا الايعاز الثاني لايجاد حدية اكثر مع وجود ايعازات للسيطرة على الحدية وهي كل من الاسم والقيمة الاسم يستخدم لايجاد حدية حول نقاط الصورة الملونة بدرجة معينة القيمة قيمة الحدية التي سوف تستخدم للتوضيح

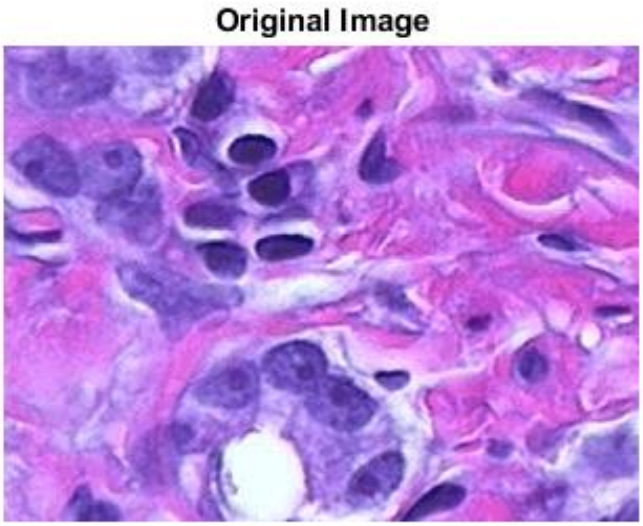
Examples

Sharpen Image

Read an image into the workspace and display it.

الايعازات ادناه لعرض وقراءة الصورة

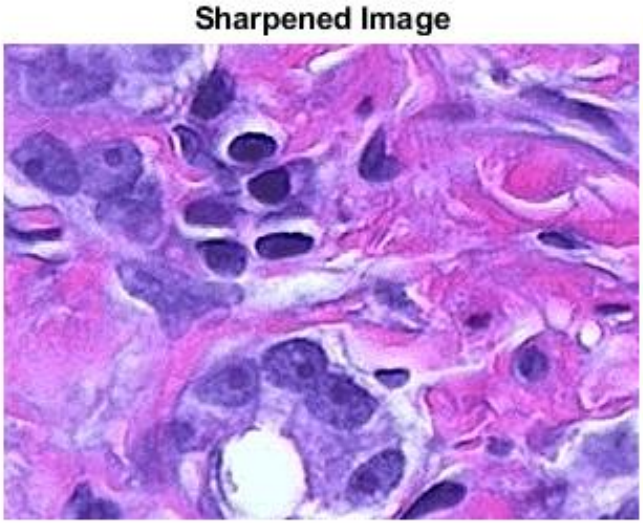
```
a = imread('hestain.png');
imshow(a)
title('Original Image');
```



Sharpen the image using the `imsharpen` function and display it.

الايعازات ادناه لعرض حدية اكثر للصورة

```
b = imsharpen(a); تطبيق حدية للصورة عن طريق هذا الايعاز
figure, imshow(b)
title('Sharpened Image');
```



الدرس الثانى 15-3-2022 تقطيع الصور Multithresh

Multilevel image thresholds using Otsu's method

Syntax

```
thresh = multithresh(A)
thresh = multithresh(A,N)
```

Description

`thresh = multithresh(A)` returns the single threshold value `thresh` computed for image `A` using Otsu's method. You can use `thresh` as an input argument to `imquantize` to convert an image into a two-level image.

العتبة = multithresh (A) تُرجع حد قيمة العتبة الفردية المحسوبة للصورة A باستخدام طريقة Otsu. يمكنك استخدام العتبة كوسيطة إدخال للتخمين لتحويل صورة إلى صورة ذات مستويين.

`thresh = multithresh(A,N)` returns `thresh` a 1-by-N vector containing `N` threshold values using Otsu's method. You can use `thresh` as an input argument to `imquantize` to convert image `A` into an image with `N+1` discrete levels.

عتبة = multithresh (A، N) ترجع العتبة متجه 1 × N يحتوي على قيم عتبة N باستخدام طريقة Otsu. يمكنك استخدام العتبة كوسيطة إدخال للتقليد لتحويل الصورة A إلى صورة ذات مستويات منفصلة N + 1.

Read image and display it.

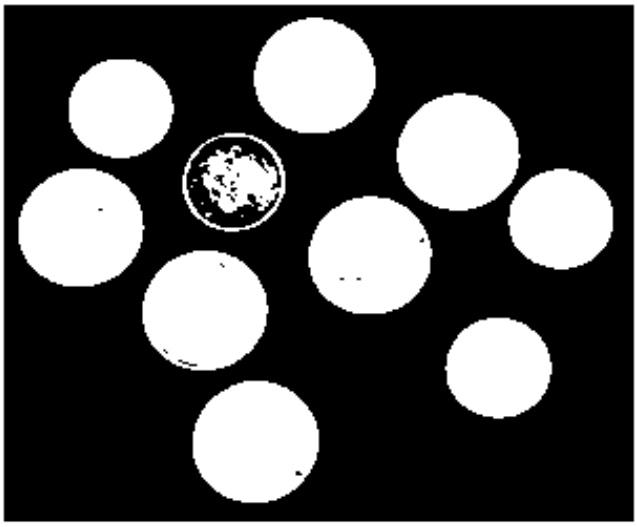
```
I = imread('coins.png'); قراءة الصورة
imshow(I) اظهار الصورة
```

Calculate a single threshold value for the image.

```
level = multithresh(I);    حساب حد العتبة للصورة
```

Segment the image into two regions using `imquantize` , specifying the threshold level returned by `multithresh` . الايعاز أعلاه يرجع حد العتبة للصورة وبعدها يتم ارسال لتكميم الصورة الى جزئين حسب حد العتبة

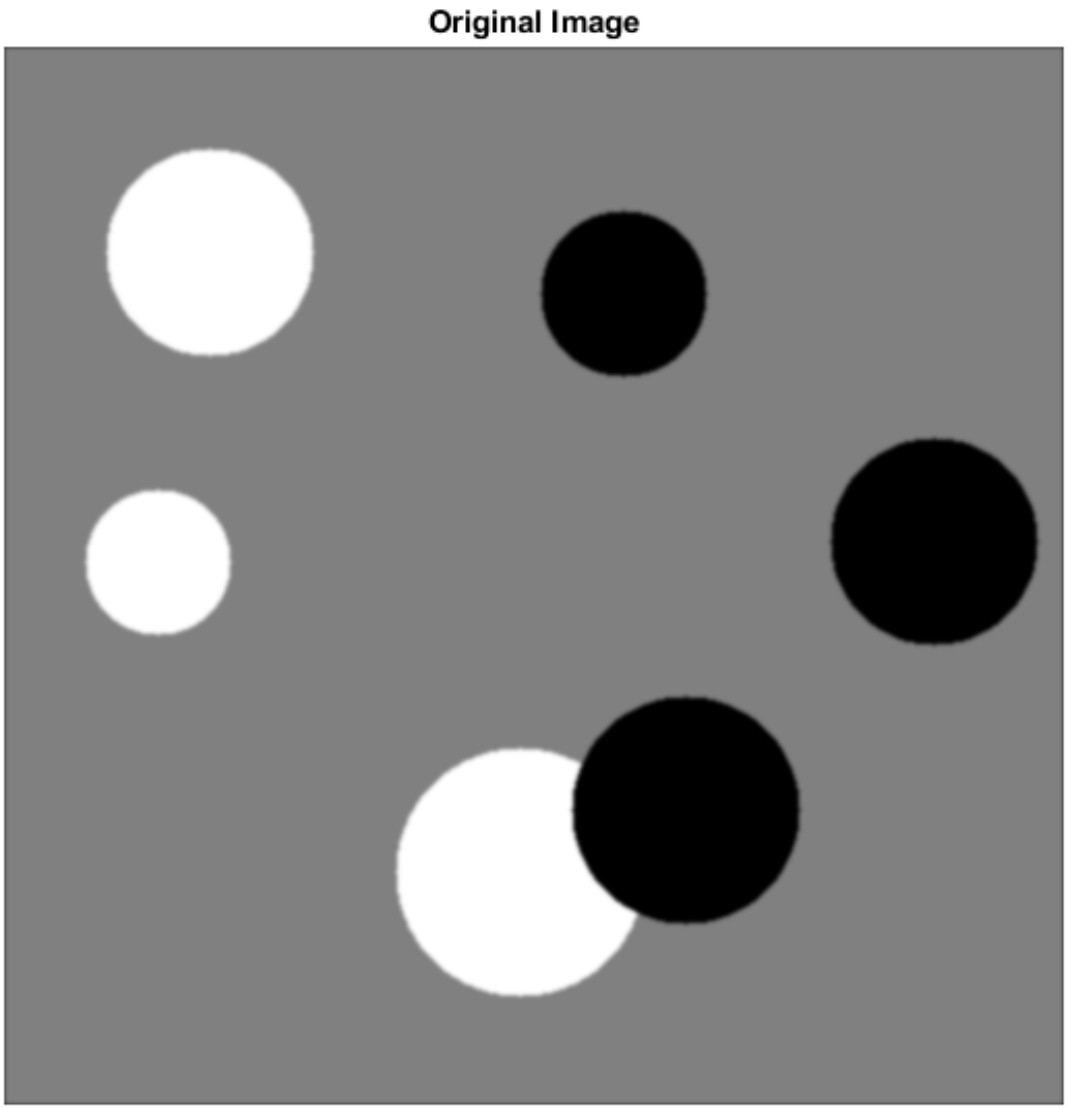
```
seg_I = imquantize(I,level); تكميم الصورة واظهارها
figure
imshow(seg_I,[])
```



Segment Image into Three Levels Using Two Thresholds

Read image and display it.

```
I = imread('circlesBrightDark.png'); اقرا الصورة
imshow(I) اظهار الصورة
axis off
title('Original Image')
```



Calculate two threshold levels.

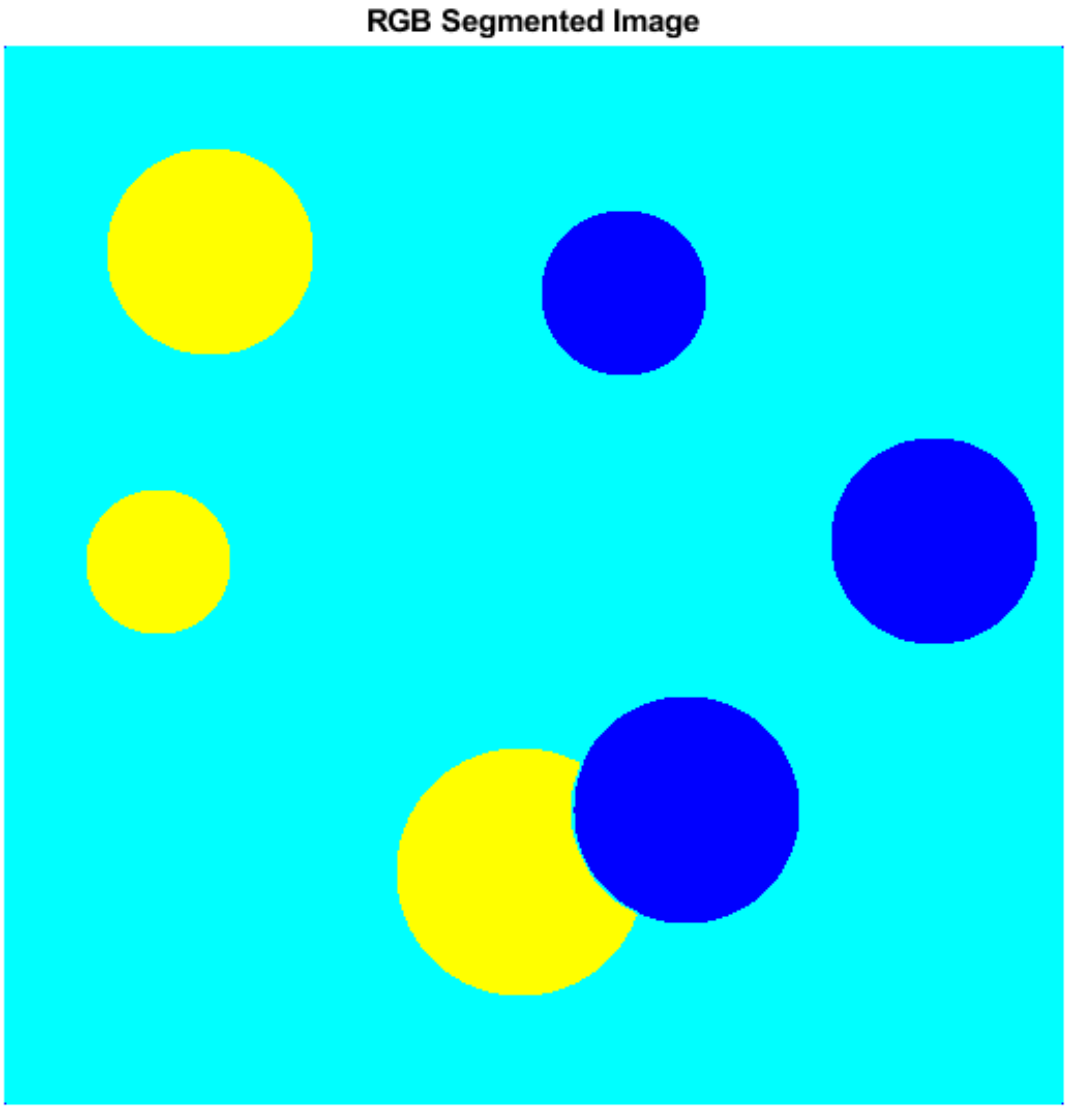
```
thresh = multithresh(I,2);
```
تكوين حد عتبتين للصورة أعلاه

Segment the image into three levels using `imquantize` . تكميم الصورة الى ثلاثة أجزاء حسب قيمة حد العتبة

```
seg_I = imquantize(I,thresh);
```
Convert segmented image into color image using `label2rgb` and display it. تحويل الصورة الى ملونة

```
RGB = label2rgb(seg_I);
```

```
figure;
imshow(RGB)
axis off
title('RGB Segmented Image')
```



الدرس الثالث : 2022-03-22

مشكلة اطراف الصورة Edges of the Images

مشكلة تظهر دائما مشكلة عند تطبيق الفلتر حيث نتساءل هل سيتم تطبيق الفلتر على النقاط التي في العمودين الأول والأخير من الصورة وكذلك الصفين الأول والأخير (هذا بالنسبة لفلتر بعده ٣×٣ ،(فجوار نقاط هذه الصفوف والأعمدة يقع جزء منه خارج مصفوفة الصورة، وهنالك في الحقيقة ثلاث تقنيات لحل هذه المشكلة Ignore the edges :

150	0	100
111	113	120
122	122	121

1. **تجاهل الاطراف Ignore of the Edges** ويرمز لهذا الأسلوب كما سيأتي بـ ' **'valid** حيث يتم تجاهل الصفوف والأعمدة على الأطراف وينتج عن هذا الأسلوب مصفوفة حجمها اصغر من المصفوفة الأصلية .
2. **الحشو بالاصفار : Padding with Zeros** وهذا التقنية تعتمد على إضافة أصفار خارج إطار المصفوفة (نقط الصورة الاصلية) مما يزيد من حجمها ويجعل أطرافها كأنها نقاط داخلية بصناعة جوار صفري لكل عمود وصفر لا جوار له، وبعد انجاز الفلتر لدينا خيارين a :
   a. الخيار ' **'same** وفيه يتم إلغاء الحشو بعد انجاز عملية الفلترة بحيث يكون الناتج مصفوفة بنفس حجم المصفوفة الأصلية b .
   b. .الخيار ' **'full** وفيه يتم الإبقاء على الصفوف والأعمدة الزائدة بقيمها الجديدة بعد الفلتر مما يسبب الحصول على مصفوفة مفلترة اكبر من الاصلية.

الفلترة في الماتلاب (Filtering in Matlab)

يتم استخدام الدالة filter2 ويمرر إليها ثلاثة وسائط parameters

Resulted_image=filter2(filter,image,shape)

المدخل الأول filter هو مصفوفة الفلتر والثاني image هو الصورة الاصلية والثالث shape هو إحدى ثلاث كلمات محجوزة لتحديد نوع المصفوفة الناتجة حسب طريقة الحشو إما same

وهي السلسلة النصية الافتراضية default أو valid أو full ،حسب الموضح أعلاه .مثال :

تطبيق الفلتر average ثلاثي البعد على المصفوفة x كما يلي :

parametersويمرر إليها ثلاثة وسائط filter2يتم استخدام الدالة

Resulted_image=filter2(filter,image,shape)

المدخل الأول filter هو مصفوفة الفلتر والثاني image هو الصورة الاصلية والثالث shape هو إحدى ثلاث كلمات محجوزة لتحديد نوع المصفوفة الناتجة حسب طريقة الحشو إما same

وهي السلسلة النصية الافتراضية default أو valid أو full ،حسب الموضح أعلاه. **مثال :**

تطبيق الفلتر average ثلاثي البعد على المصفوفة x كما يلي:

>> x=uint8(10*magic(5));// **magic** generate image with 5 x 5 dimensions

هذه المصفوفة عبارة عن صورة افتراضية تم توليدها عبر الدالة.magic

>> a=ones(3,3)/9

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

والمصفوفة a هي الفلتر المطلوب تطبيقه.

```
>> filter2(a,x,'same');
```

ولنعيد تطبيق نفس السطر مع تغيير السلسلة النصية shape مره valid ومره full ثم لاحظ الفرق .

الدالة fspecial لتوليد الفلاتر القياسية

هناك مجموعة كبيرة من الفلاتر القياسية نستطيع توليدها عبر الدالة fspecial حسب اسمائها المخزونة مسبقا.

فالسطر التالي:

```
>> f= fspecial('average',[5,7]);
```

يولد فلتر المتوسط الذي طبقناه سابقا ولكن بأبعاد مختلفة، وإذا لم يتم تحديد أبعاد فإن البعد القياسي: ٣×٣ هو

```
>> aa= fspecial('average');
```

المصفوفة aa هي نفسها المصفوفة a التي تم حجزها سابقا. وهناك فلاتر اخرى مثلا فلتر لابلاسيان

```
>> f=fspecial('laplacian')
```

والفلتر جاوسيان على لابلاسيان ويختصر:log

```
>> f1=fspecial('log')
```

Frequencies; low and high pass filters العبور ومنخفضة عالية الفلاتر توجد في كل صورة مجموعة من المكونات، ولو نظرنا إلى المصفوفة نفسها لوجدنا ان هذه المكونات تنعكس على قيم المصفوفة وإجمالا نستطيع ان نقول ان لدينا نوعين من المكونات هي المكونات عالية التردد (التكرار)، والمكونات منخفضة التكرار أو التردد. المكونات عالية التكرار: components frequency High هي مكونات تتميز بأنها تتغير بصورة كبيرة في قيم المستوى الرمادي في مسافات صغيرة، ومن أمثلتها الحواف edges والتشوشات noise. المكونات منخفضة التكرار: components frequency Low هي مكونات ثابتة او شبه ثابتة داخل الصورة وتغيراتها قليلة في قيم المستوى الرمادي ومن امثلتها الخلفيات backgrounds ونسيج الصورةtextures عموما. وهذا يقودنا إلى تعريف نوعين من الفلاتر المطبقة على الصورة:

- high pass filter: it 'passes over' the high frequency components, and reduces or eliminates low frequency components,

الدرس الرابع : 29-3-2022 تطبيق الفلاتر عالية ومنخفضة العبور

الفلاتر عالية العبور HPF تتجاوز المكونات عالية التكرار بدون تغيير ولكنها تقلل أو تلغي المكونات منخفضة التكرار.

- low pass filter: it 'passes over' the low frequency components, and reduces or eliminates high frequency components,

الفلاتر منخفضة العبور LPF تتجاوز المكونات المنخفضة التكرار بدون تغيير، ولكنها تقلل او تلغي المكونات عالية التكرار.

سنقول اختصار فلاتر عالية وفلاتر منخفضة.

مثال

```
>> c=imread('cameraman.tif');
>> f1=fspecial('average');// 3 x 3 filter
>> cf1=filter2(f1,c);
```

لاحظ شكل الصورة الناتجة مع الفلتر ذي البعد ٣×٣ ثم اعد تطبيقه مع نفس الفلتر ولكن بحجم اكبر ٩×٩ ثم اكبر ٢٥×٢٥ ،كما في الأشكال التالية:

(a) Original image

(b) Average filtering

(c) Using a 9×9 filter

(d) Using a 25×25 filter

والفلترين laplacian و log هما فلتران عاليا العبور HPF، كما في المثالين التاليين:

مثال 1 : الفلتر laplacian

```
>> f=fspecial('laplacian');
>> cf=filter2(f,c);
 >> imshow(cf/100);
```

مثال 2: الفلتر LoG

```
>> f1=fspecial('log');
>> cf1=filter2(f1,c);
 >> figure,imshow(cf1/100);
```

والشكل التالي يوضح نتيجة تطبيقهما على الصورة c صورة رجل الكاميرا كما سيأتي:

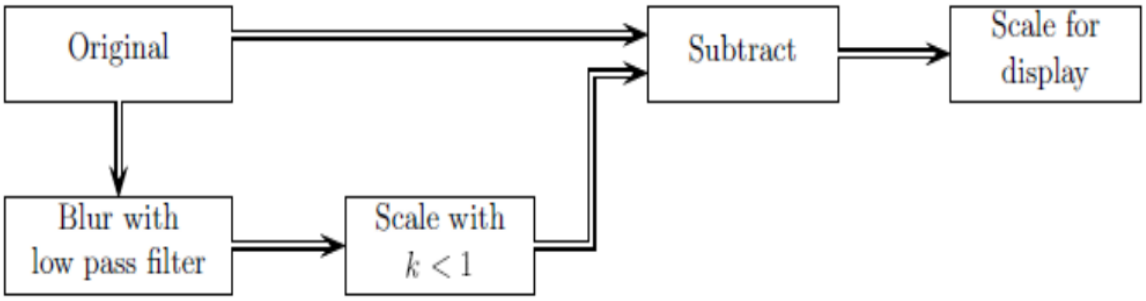
(a) Laplacian filter

(b) Laplacian of Gaussian ("log") filtering

لاحظ كيف يقوم الفلتر عالي العبور بتغيير المناطق المنخفضة التكرار وتقليل قيمها حتى صارت اقرب إلى الصفر، وهذا هو سبب ظهور اللون الأسود في الصورتين الأخيريتن، أما الفلتر منخفض العبور فهو يقوم بتقليل قيم المكونات المرتفعة مما يعني اقتراب الصورة من المستوى الرمادي المتوسط أكثر وأكثر.

الدرس الخامس : 5-4-2022

هي إحدى أهم تطبيقات الفلاتر المنخفضة حيث يتم في هذه التقنية عمل فلتر منخفض ثم طرحه من الصورة الأصلية فينتج لدينا صورة اصلية اكثر تحديدا لحواف مكوناتها، وتسمى هذه التقنية أيضا بتحسين الحواف أو القناع غير الحادunsharp masking والمخطط في الشكل التالي يوضح هذه العملية:



تطبيق هذه الفكرة في الكود التالي، لاحظ اننا استخدمنا فلترا منخفضا هو الفلترaverage

```
>> x=imread('cameraman.tif');
>>f=fspecial('average');
>> xf=filter2(f,x);
>> xu=double(x)-xf/1.5;
>> imshow(xu/70);
```

ولقد حصلت تعديلات عديدة على هذه الفكرة، وتم تطوير فلتر يختصر خطوتي التحجيم scaleوالطرح subtracatبخطوة واحدة عبر الفلتر unsharp الذي يقبل وسيطا

parameterواحد تحدد نسبة التحديد sharpingالمطلوب .المثال التالي يطبق هذا الفلتر على صورة الصبي

```
>> p=imread('pout.tif');

>> u=fspecial('unsharp',0.5);

>> pu=filter2(u,p);

>> imshow(p),figure,imshow(pu/255);
```

لاحظ قسمة المصفوفة pu على 255 قبل عرضها بسبب زيادة قيم الصورة عن القيمة القصوى للون الرمادي .وبطبيعة الحال فكل صورة تحتاج إلى قيمة مناسبة لتعديل القيم، في حين يتم استخدام الدالة im2gray بعض الأحيان لتقوم بنفس الدور.

الدرس السادس : 12-4-2022

اكتشاف الحواف Edge Detection: في الصورة ، الحافة عبارة عن منحنى يتبع مسار التغيير السريع في شدة تلك الصورة. غالبًا ما ترتبط الحواف بحدود الكائن في بيئة المشهد. يستخدم اكتشاف الحواف لتحديد الحواف في الصورة لتسهيل معالجة الصورة. يعمل الكشف

عن الحواف من خلال الكشف عن الانقطاعات في السطوع. يستخدم اكتشاف الحواف بشكل أساسي لتجزئة الصور Image Segmentation واستخراج البيانات في مجالات مثل معالجة الصور ورؤية الكمبيوتر ورؤية الآلة.

للعثور على الحواف ، يمكنك استخدام حافة وظيفة الحافة المضمنة...

edge(image, Edge detector) في الـــ Matlab. تبحث هذه الوظيفة المضمنة عن أماكن في الصورة تتغير فيها الكثافة بسرعة ، باستخدام أحد هذين المعيارين:

- الأماكن التي يكون فيها المشتق الأول للشدة أكبر في الحجم من قيمة حدية معينة Threshold.
- الأماكن التي يكون فيها المشتق الثاني من الشدة تقاطعًا صفريًا.

توفر كاشفات الحواف Edge Detector العديد من المقدرات المشتقة ، كل منها يطبق أحد التعريفات المذكورة أعلاه. بالنسبة لبعض هذه المقدرات ، يمكنك تحديد ما إذا كانت العملية يجب أن تكون حساسة للحواف الرأسية أو الحواف الأفقية أو كليهما. تعرض مقدرات الحواف صورة ثنائية تحتوي على 1 قيمة حيث توجد الحواف وصفر قيمة في مكان آخر. أقوى تقنية لاكتشاف الحواف توفرها الحافة هي طريقة Canny. تختلف طريقة Canny عن طرق اكتشاف الحواف الأخرى من حيث أنها تستخدم نوعين مختلفين من مستويات العتبات لاكتشاف الحواف القوية والضعيفة. تتضمن طريقة اكتشاف الحواف الحادة الحواف الضعيفة في الإخراج فقط إذا كانت متصلة بحواف قوية. لذلك ، تقل احتمالية تأثر هذه الطريقة بالضوضاء ، وتزيد احتمالية اكتشاف الحواف الضعيفة الحقيقية.

هناك العديد من الوظائف المضمنة لاكتشاف الحواف المتوفرة في Matlab مثل:

- ✓ كاشف حافة سوبل Sobel Edge Detector
- ✓ جهاز كشف الحواف بريويت Prewitt edge detector
- ✓ روبرت حافة كاشف Robert edge detector
- ✓ كاشف حافة لابلاشين وكاوشن Log Edge Detector
- ✓ كاشف الحافة الصفرية Zero Cross Edge Detector
- ✓ كاشف حافة كاني Canny Edge Detector

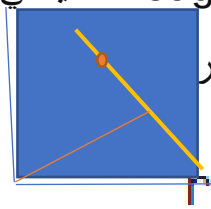
```
% importing the image
I = rgb2gray(imread("flowers.jpg"));
subplot(2, 4, 1),
imshow(I);
title("Gray Scale Image");

% Sobel Edge Detection
J = edge(I, 'Sobel');
subplot(2, 4, 2),
imshow(J);
title("Sobel");

% Prewitt Edge detection
K = edge(I, 'Prewitt');
subplot(2, 4, 3),
imshow(K);
title("Prewitt");
```

```
% Robert Edge Detection
L = edge(I, 'Roberts');
subplot(2, 4, 4),
imshow(L);
title("Robert");

% Log Edge Detection
M = edge(I, 'log');
subplot(2, 4, 5),
imshow(M);
title("Log");

% Zerocross Edge Detection
M = edge(I, 'zerocross');
subplot(2, 4, 6),
imshow(M);
title("Zerocross");

% Canny Edge Detection
N = edge(I, 'Canny');
subplot(2, 4, 7),
imshow(N);
title("Canny");
-
```

الدرس السابع والثامن 18-2022-4/26

تحويل هوغ

يدعم Image Processing Toolbox ™ الوظائف التي تتيح لك استخدام تحويل Hough لاكتشاف الخطوط في الصورة. وتنفذ وظيفة **hough** تحويل Hough القياسي (SHT). تم تصميم تحويل هوغ لاكتشاف الخطوط ، باستخدام التمثيل البارامتر



```
rho = x*cos(theta) + y*sin(theta)
```

المتغير rho هو المسافة من الأصل إلى الخط على طول متجه عمودي على الخط. ثيتا هي الزاوية بين المحور x وهذا المتجه. تنشئ الدالة hough مصفوفة مساحة معلمة تتوافق صفوفها وأعمدتها مع قيم rho و theta ، على التوالي.

بعد حساب تحويل Hough ، يمكنك استخدام وظيفة houghpeaks للعثور على قيم الذروة في مساحة المعلمة. تمثل هذه القمم الخطوط المحتملة في صورة الإدخال. بعد تحديد القمم في تحويل Hough ، يمكنك استخدام وظيفة houghlines للعثور على نقاط نهاية مقاطع الخط المقابلة للقمم في تحويل Hough. تملأ هذه الوظيفة الفجوات الصغيرة في مقاطع الخط تلقائيًا.

كشف الخطوط في الصور باستخدام Hough

يوضح هذا المثال كيفية اكتشاف الخطوط في صورة ما باستخدام تحويل هوغ.

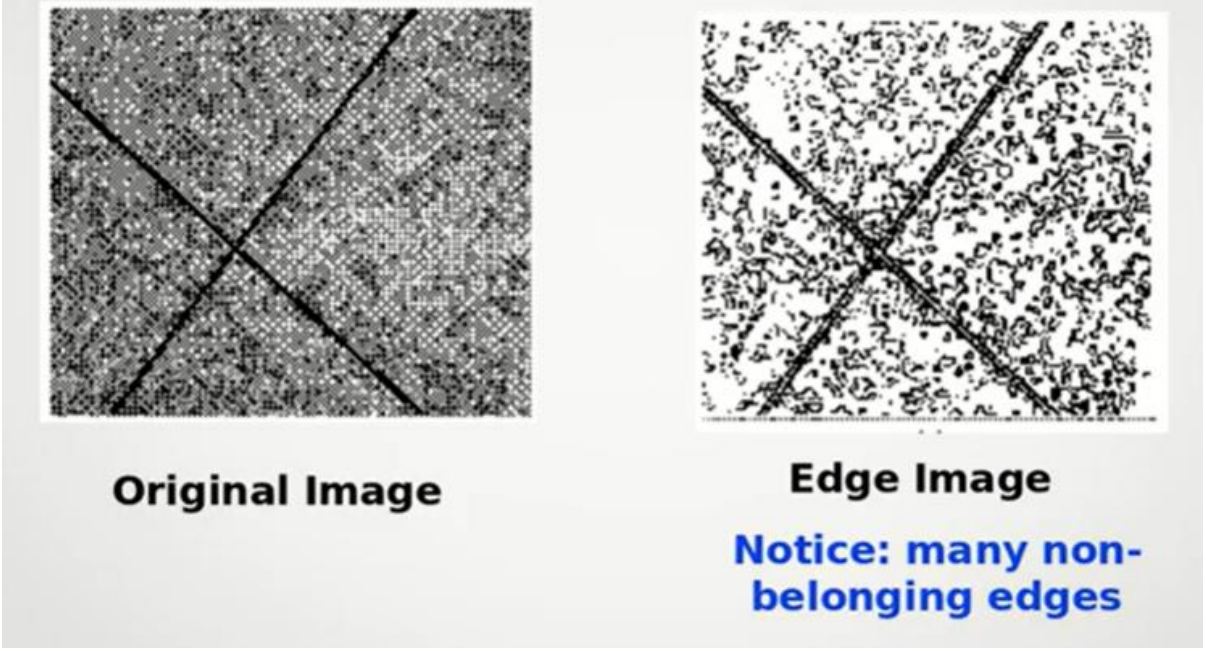
الخورامية

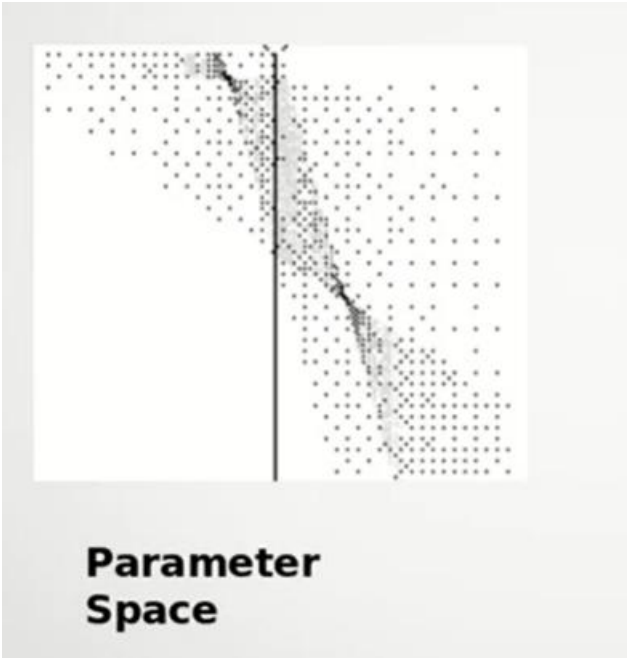
Algorithm: Hough Transform for line detection

1. Parameter space is divided into accumulator cells A, all, initially, set to zero.
2. For every point $p(x,y)$ in image, change m in the range and calculate c. $c = -xm + y$
3. $A(m, c) = A(m, c) + 1$

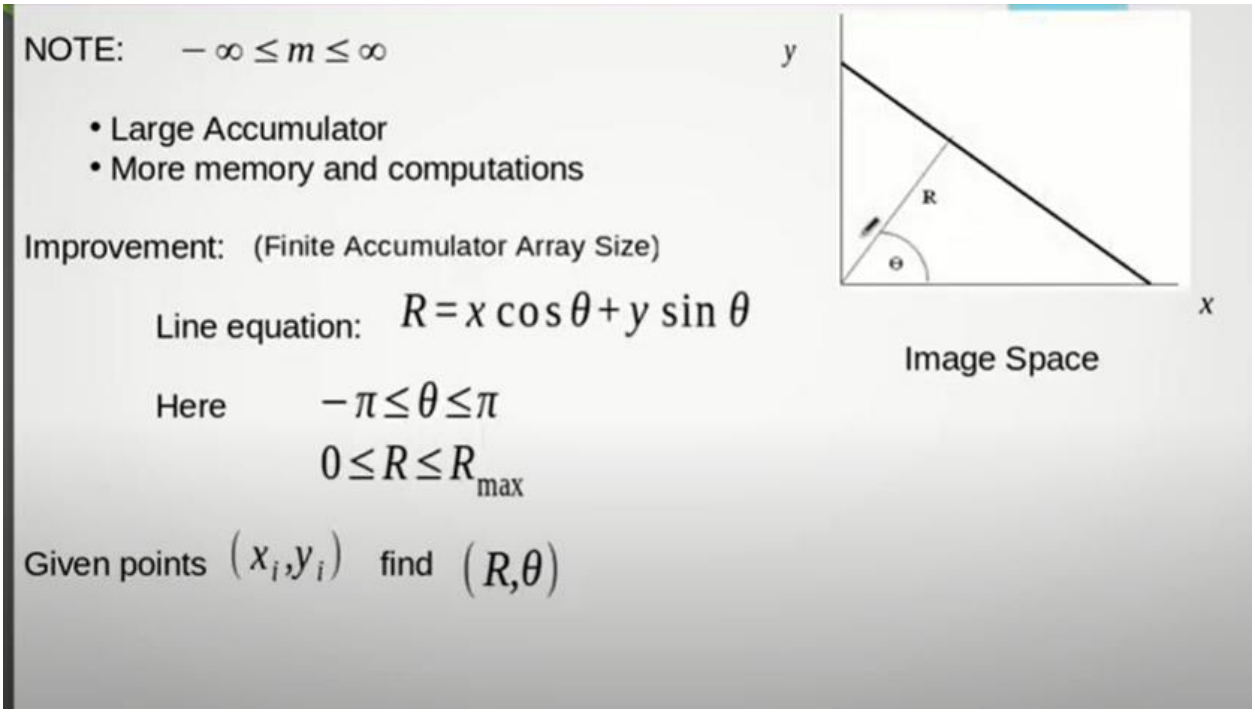
At the end the value of $A(m_i, c_j)$ corresponds to the number of points that lie on the line:

$$y = - m_i x + c_j$$


Original Image
Edge Image
Notice: many non-belonging edges

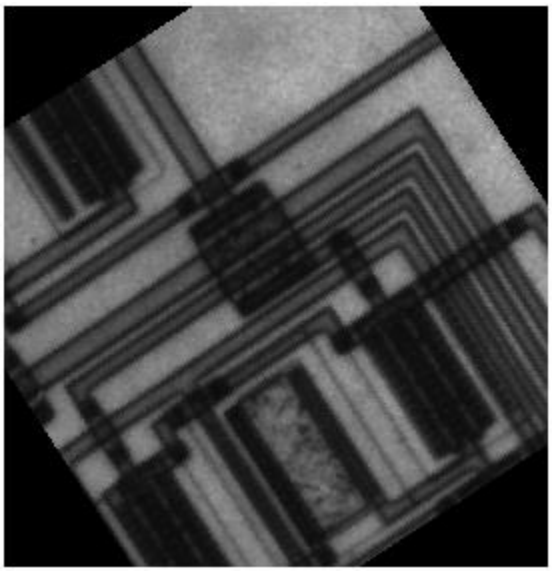

Parameter Space

طريقة اخرى



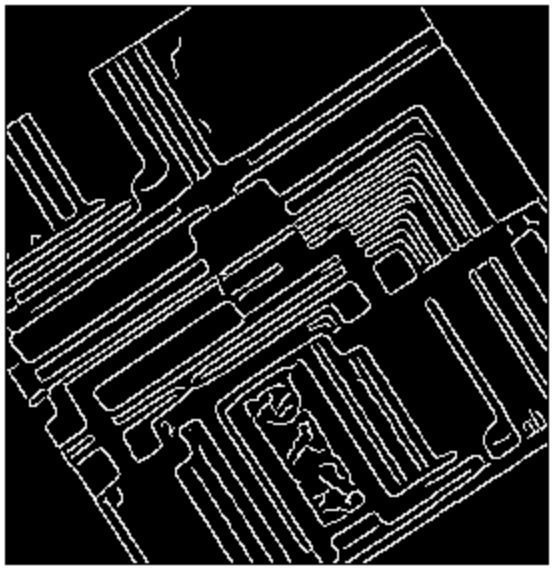
اقرأ صورة في مساحة العمل ، ولجعل هذا المثال أكثر توضيحًا ، قم بتدوير الصورة. اعرض الصورة.

```
I = imread('circuit.tif');
rotI = imrotate(I,33,'crop');
imshow(rotI)
```



ايجاد الزويا في الصورة بأستخدام دالة edge

```
BW = edge(rotI,'canny');
imshow(BW);
```

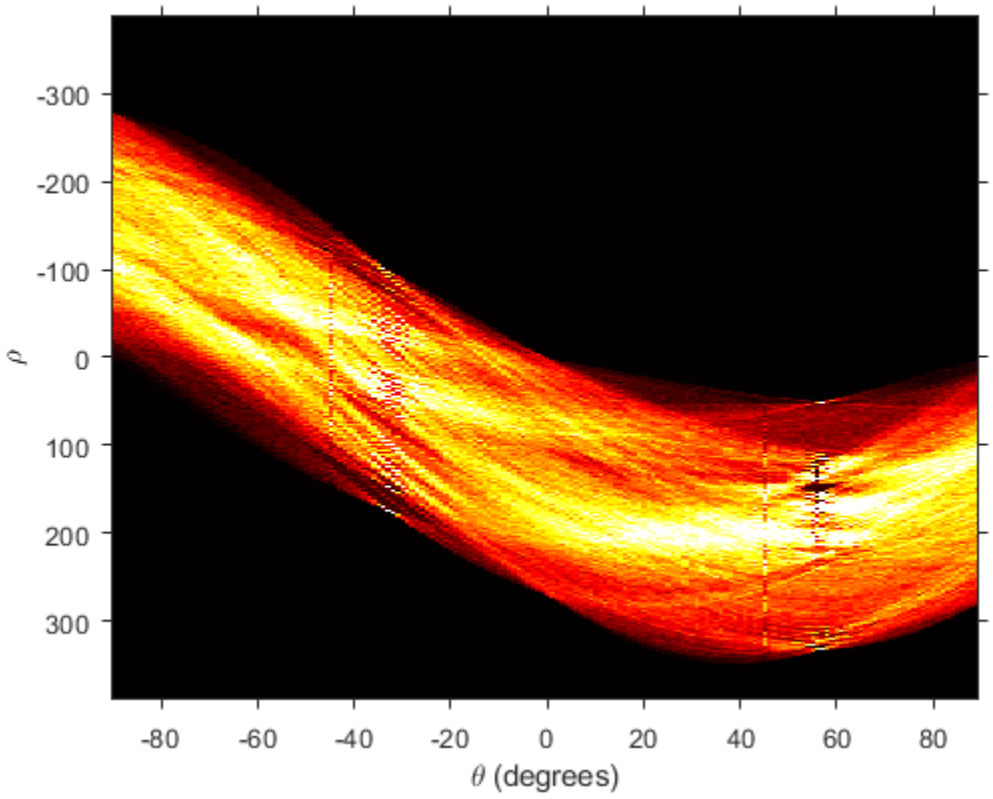


حساب تحويل هوغ من الصورة الثنائية الراجعة من دالة edge

```
[H,theta,rho] = hough(BW);
```

عرض تحويل هوغ الناتج من الدالة hough

```
figure
imshow(imadjust(rescale(H)),[],...
       'XData',theta,...
       'YData',rho,...
       'InitialMagnification','fit');
xlabel('\theta (degrees)')
ylabel('\rho')
axis on
axis normal
hold on
colormap(gca,hot)
```



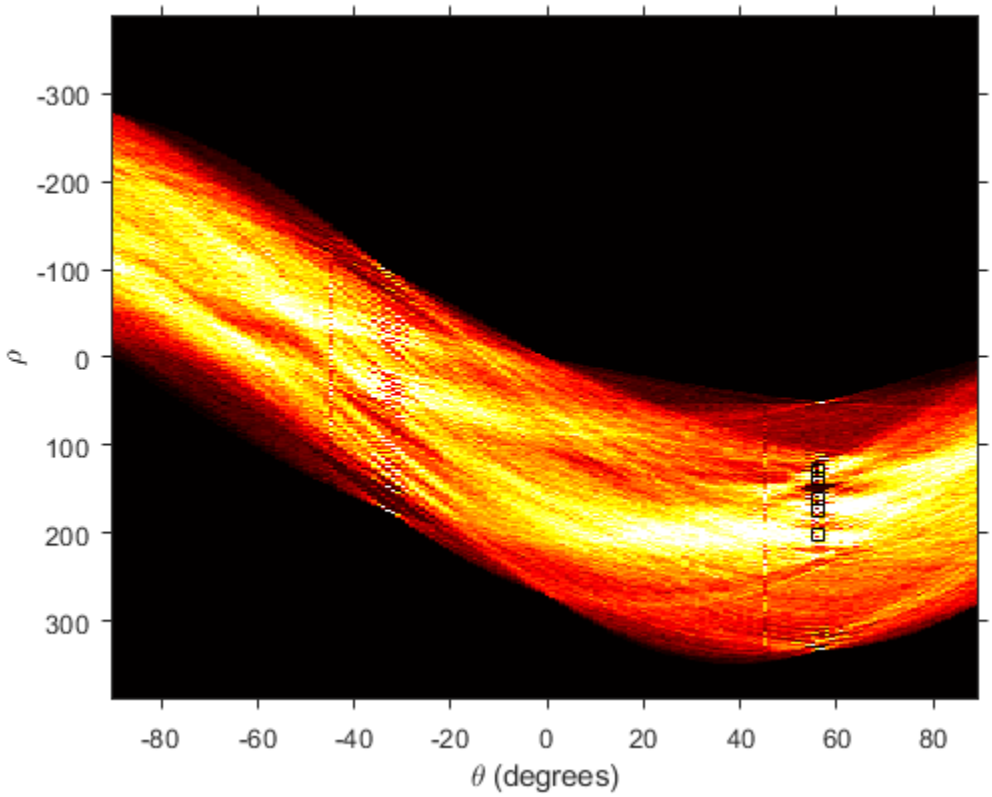
أوجد القمم في مصفوفة تحويل Hough، H ، باستخدام وظيفة houghpeaks.

```
P = houghpeaks(H,5,'threshold',ceil(0.3*max(H(:))));
```

قم بتركيب مخطط على صورة التحويل الذي يحدد القمم.

```
x = theta(P(:,2));
y = rho(P(:,1));
```

```
plot(x,y,'s','color','black');
```



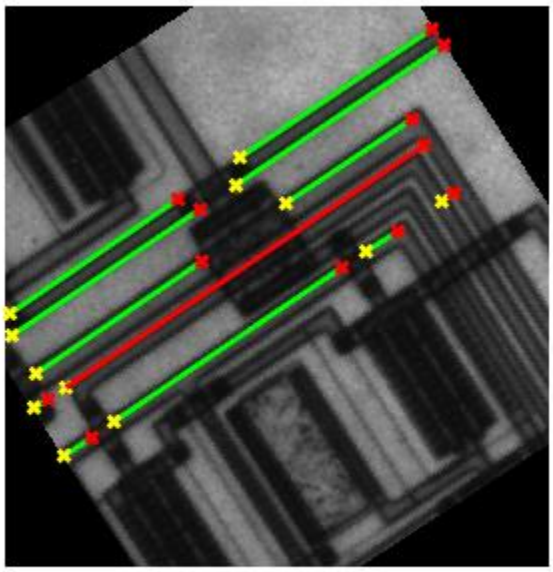
ابحث عن خطوط في الصورة باستخدام وظيفة houghlines.

```
lines = houghlines(BW,theta,rho,P,'FillGap',5,'MinLength',7);
```

قم بإنشاء مخطط يعرض الصورة الأصلية بالخطوط المتراكبة عليها.

```
figure, imshow(rotI), hold on
max_len = 0;
for k = 1:length(lines)
   xy = [lines(k).point1; lines(k).point2];
   plot(xy(:,1),xy(:,2),'LineWidth',2,'Color','green');

   % Plot beginnings and ends of lines
   plot(xy(1,1),xy(1,2),'x','LineWidth',2,'Color','yellow');
   plot(xy(2,1),xy(2,2),'x','LineWidth',2,'Color','red');
```

```
    % Determine the endpoints of the longest line segment
    len = norm(lines(k).point1 - lines(k).point2);
    if ( len > max_len)
       max_len = len;
       xy_long = xy;
    end
end
% highlight the longest line segment
plot(xy_long(:,1),xy_long(:,2),'LineWidth',2,'Color','red');
```



الدرس التاسع والعاشر 17-2022/5/24

Watershed transform (تحويل مجمعات المياه) for 2D images

يجد تحويل مستجمعات المياه "أحواض تجميع المياه" أو "خطوط حافة مستجمعات المياه" في صورة ما عن طريق معاملتها كسطح حيث تمثل ***وحدات البكسل الفاتحة ارتفاعات عالية*** وتمثل ***وحدات البكسل الداكنة ارتفاعات منخفضة***. يمكن استخدام تحويل مستجمعات المياه لتقسيم المناطق المتجاورة ذات الأهمية إلى كائنات مميزة.

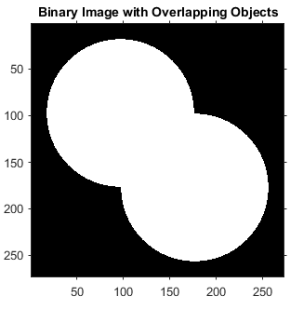
`L = watershed(A)` returns a label matrix `L` that identifies the watershed regions of the input matrix `A`.

`L = watershed(A,conn)` specifies the connectivity to be used in the watershed computation.

مثال

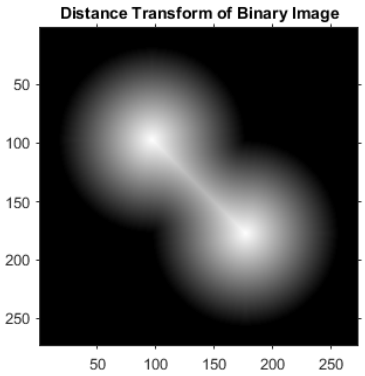
1. نقوم بإنشاء صورة ثنائية تحتوي على كائنين دائريين متداخلين. ونعرض الصورة.

```
center1 = -40;
center2 = -center1;
dist = sqrt(2*(2*center1)^2);
radius = dist/2 * 1.4;
lims = [floor(center1-1.2*radius)
ceil(center2+1.2*radius)];
[x,y] = meshgrid(lims(1):lims(2));
bw1 = sqrt((x-center1).^2 + (y-center1).^2) <= radius;
bw2 = sqrt((x-center2).^2 + (y-center2).^2) <= radius;
bw = bw1 | bw2;
imshow(bw)
title('Binary Image with Overlapping Objects')
```

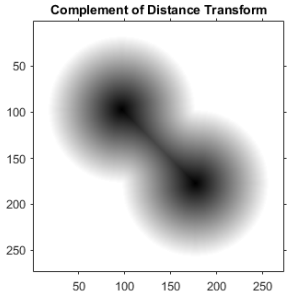


3. احسب تحويل المسافة لمعكوس او مكمل الصورة الثنائية. قيمة كل بكسل في صورة الإخراج هي المسافة بين ذلك البكسل وأقرب بكسل غير صفري من bw

```
D = bwdist(~bw);
imshow(D,[])
title('Distance Transform of Binary Image')
```



4. نأخذ تكملة الصورة التي تم تحويلها من مسافة بحيث تمثل وحدات البكسل الفاتحة ارتفاعات عالية وتمثل وحدات البكسل الداكنة ارتفاعات منخفضة لتحويل مستجمعات المياه.

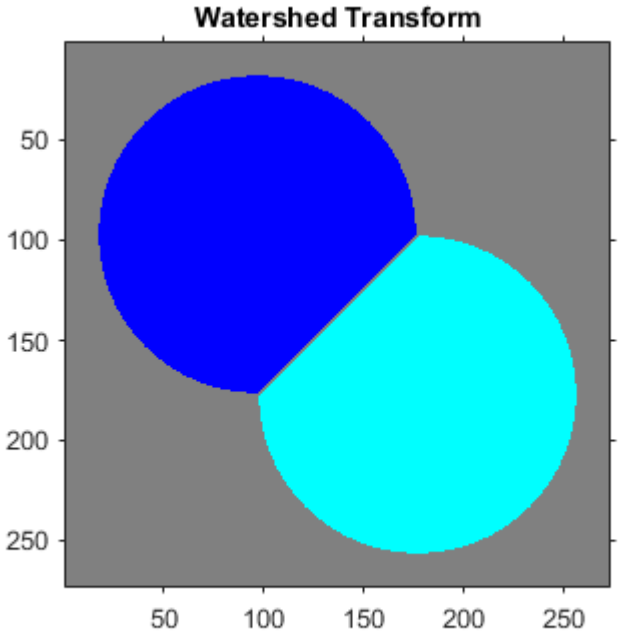


4. احسب تحويل مستجمعات المياه. عيّن وحدات البكسل التي تقع خارج الكائن المحدد تكون قيمتها هي 0.

```
L = watershed(D);
L(~bw) = 0;
```

5. عرض مصفوفة التسمية الناتجة كصورة RGB.

```
rgb = label2rgb(L,'jet',[.5 .5 .5]);
imshow(rgb)
title('Watershed Transform')
```



الدرس الحادي عشر 24-2022/5/31

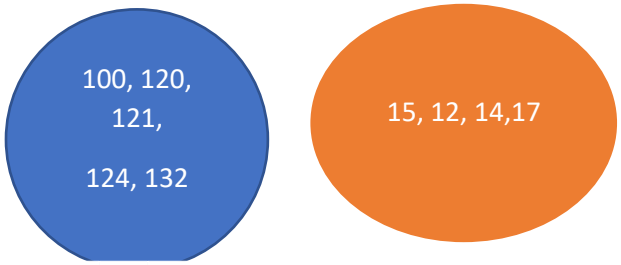
K means Clustering عنقدة البيانات

Cluster data using *k*-means clustering, then plot the cluster regions. عنقد البيانات ورسمها باستخدام الk-means

100	120	15
12	14	17
121	124	132

K means = K= عدد العناقيد التي تحتاج لها لفصل البيانات

K=2



Dist(120, Xi)= Dist(120, 100,15,…..132= SQRT(100-120)^2=SGRT(400)=SQRT(400)=1.2 the distance between 100, 120

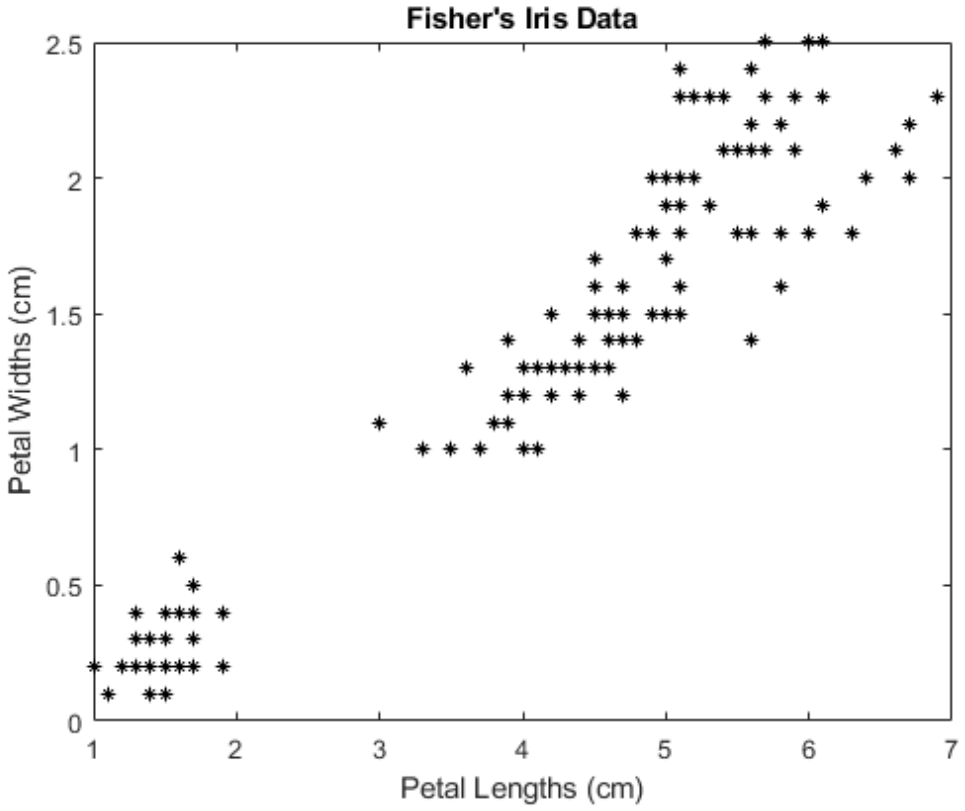
Min Distance 120=

New Mean = 120+100/2= 110 نقطة افتراضية

New Mean = 17+15/2= 16 نقطة افتراضية ثانية

Load Fisher's iris data set. Use the petal lengths and widths as predictors.

```
load fisheriris تحميل البيانات
X = meas(:,3:4);

figure;
plot(X(:,1),X(:,2),'k*','MarkerSize',5);
title 'Fisher''s Iris Data';
xlabel 'Petal Lengths (cm)';
ylabel 'Petal Widths (cm)';
```



The larger cluster seems to be split into a lower variance region and a higher variance region. This might indicate that the larger cluster is two, overlapping clusters. العنقود الاول يكون عنقودان اخران اكثر

Cluster the data. Specify *k* = 3 clusters.

```
rng(1); % For reproducibility
[idx,C] = kmeans(X,3);
```

`idx` is a vector of predicted cluster indices corresponding to the observations in `X`. `C` is a 3-by-2 matrix containing the final centroid locations.

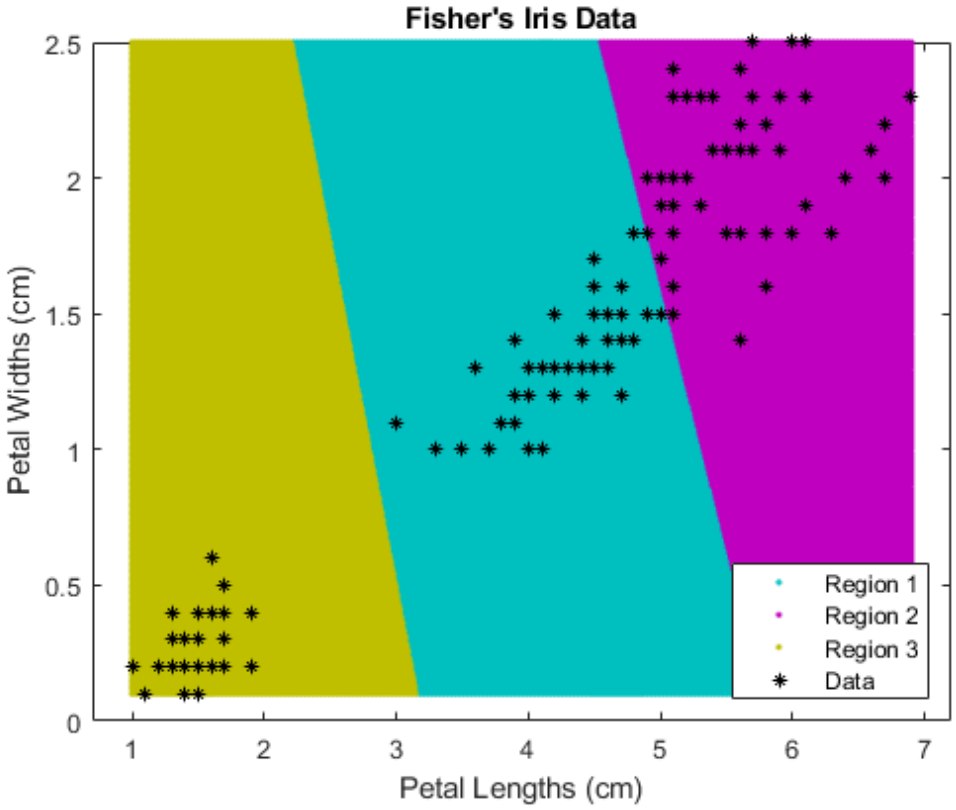
Use `kmeans` to compute the distance from each centroid to points on a grid. To do this, pass the centroids (`C`) and points on a grid to `kmeans`, and implement one iteration of the algorithm.

```
x1 = min(X(:,1)):0.01:max(X(:,1));
x2 = min(X(:,2)):0.01:max(X(:,2));
[x1G,x2G] = meshgrid(x1,x2);
XGrid = [x1G(:),x2G(:)]; % Defines a fine grid on the plot

idx2Region = kmeans(XGrid,3,'MaxIter',1,'Start',C);
Warning: Failed to converge in 1 iterations.
    % Assigns each node in the grid to the closest centroid
```

`kmeans` displays a warning stating that the algorithm did not converge, which you should expect since the software only implemented one iteration.

Plot the cluster regions.

```
figure;
gscatter(XGrid(:,1),XGrid(:,2),idx2Region,...
    [0,0.75,0.75;0.75,0,0.75;0.75,0.75,0],'..');
hold on;
plot(X(:,1),X(:,2),'k*','MarkerSize',5);
title 'Fisher''s Iris Data';
xlabel 'Petal Lengths (cm)';
ylabel 'Petal Widths (cm)';
legend('Region 1','Region 2','Region 3','Data','Location','SouthEast');
hold off;
```


Fisher's Iris Data
Petal Widths (cm)
Petal Lengths (cm)
Region 1
Region 2
Region 3
Data
0
0.5
1
1.5
2
2.5
1
2
3
4
5
6
7