

Procesamiento de imágenes con Octave

Lectura de una imagen

```
>>A=imread('pin.jpg');  
  
>>A=imread('C:\Users\DFR\pin.jpg');
```

Formatos: jpg, TIFF, JPEG, GIF, BMP, PNG, XWD.

```
>> A=imread('C:\Users\DFR\pin.jpg');
>> whos
Variables in the current scope:
```

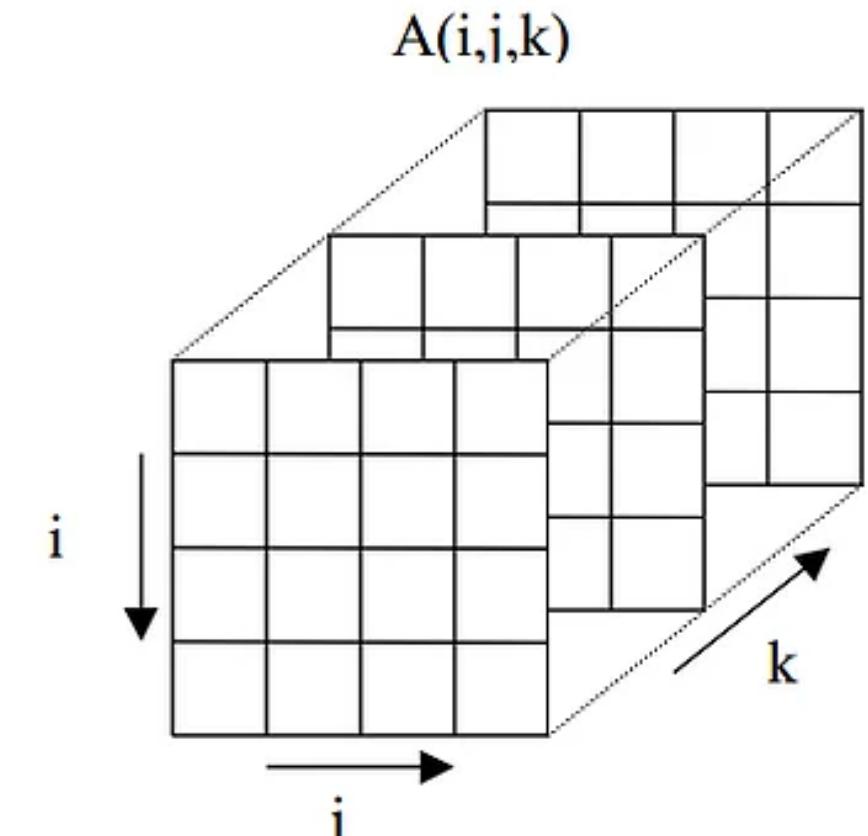
Attr	Name	Size	Bytes	Class
=====	=====	=====	=====	=====
	A	768x1024x3	2359296	uint8

Total is 2359296 elements using 2359296 bytes

A es una hipermatriz : Formada por tres matrices. RGB

Tipo de dato: uint 8: Entero: 0 - 255 → Escala de grises

$R = A(:,:,1)$
 $G = A(:,:,2)$
 $B = A(:,:,3)$



:Attr Name	Size	Bytes	Class
A	$768 \times 1024 \times 3$	2359296	uint8

Imagen original



>> imshow(A)

Imagen Red



Imagen Green



Imagen Blue



```
>>R=A(:,:1);  
>> imshow(R)
```

```
>>G=A(:,:2);  
>> imshow(G)
```

```
>>B=A(:,:3);  
>> imshow(B)
```

Tipo de datos

- **uint8**: Enteros de 8 bits en el rango de [0,255] (1 byte por elemento)
- **uint16**: Enteros de 16 bits en el rango de [0, 65535] (2 bytes por elemento)
- **uint32**: Enteros de 32 bits en el rango de [0, 4294967295] (4 bytes por elemento)
- **Double**: Valores reales entre 0- 1. (8 byte por elemento)
- **Single**: Valores reales entre 0-1 . (4 byte por elemento)

Imagen original



```
>> imshow(A)
```

Imagen en escala de grises



```
>> g=rgb2gray(A)  
>>imshow(g)
```

Transformación de la hipermatriz a una matriz

```
>> g=rgb2gray(A)  
>>imshow(g)
```

Variables in the current scope:

Attr Name	Size	Bytes	Class
====	=====	=====	=====
A	768x1024x3	2359296	uint8
g	768x1024	786432	uint8

```
>> imshow(A(1:256,:,:))
```



```
>> imshow(A(257:413,:,:))
```



```
>> imshow(A(414:768,:,:))
```



Utilización de funciones del paquete de imágenes

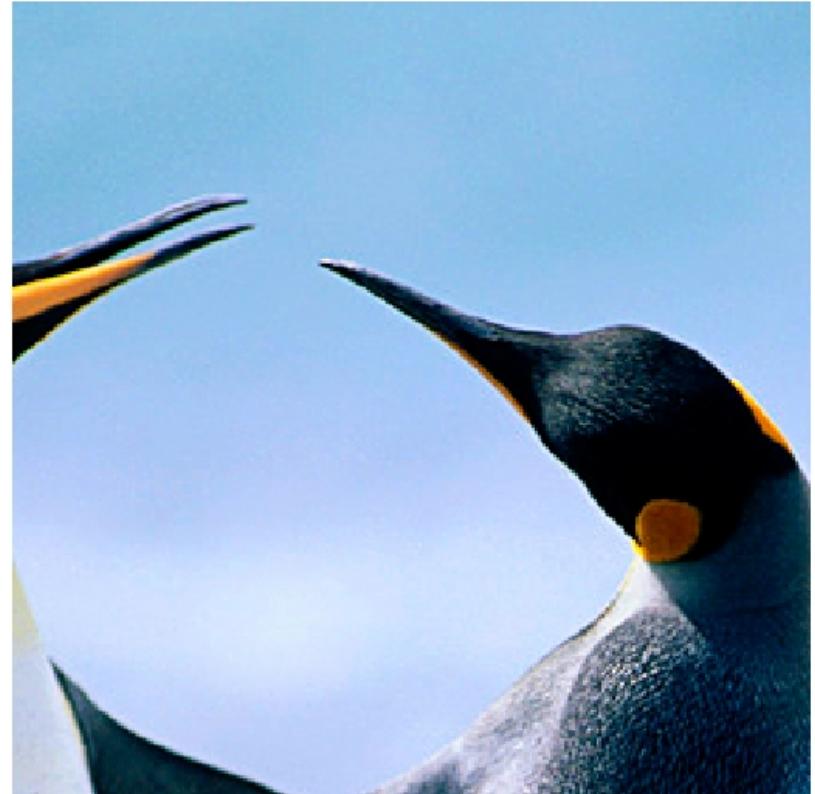
```
>> pkg load image
```

Función imcrop

```
>>d1=imcrop(A,[0 0 300 300]);  
>>imshow(d1)
```



```
>>d2=imcrop(A,[301 0 300 300]);  
>>imshow(d2)
```



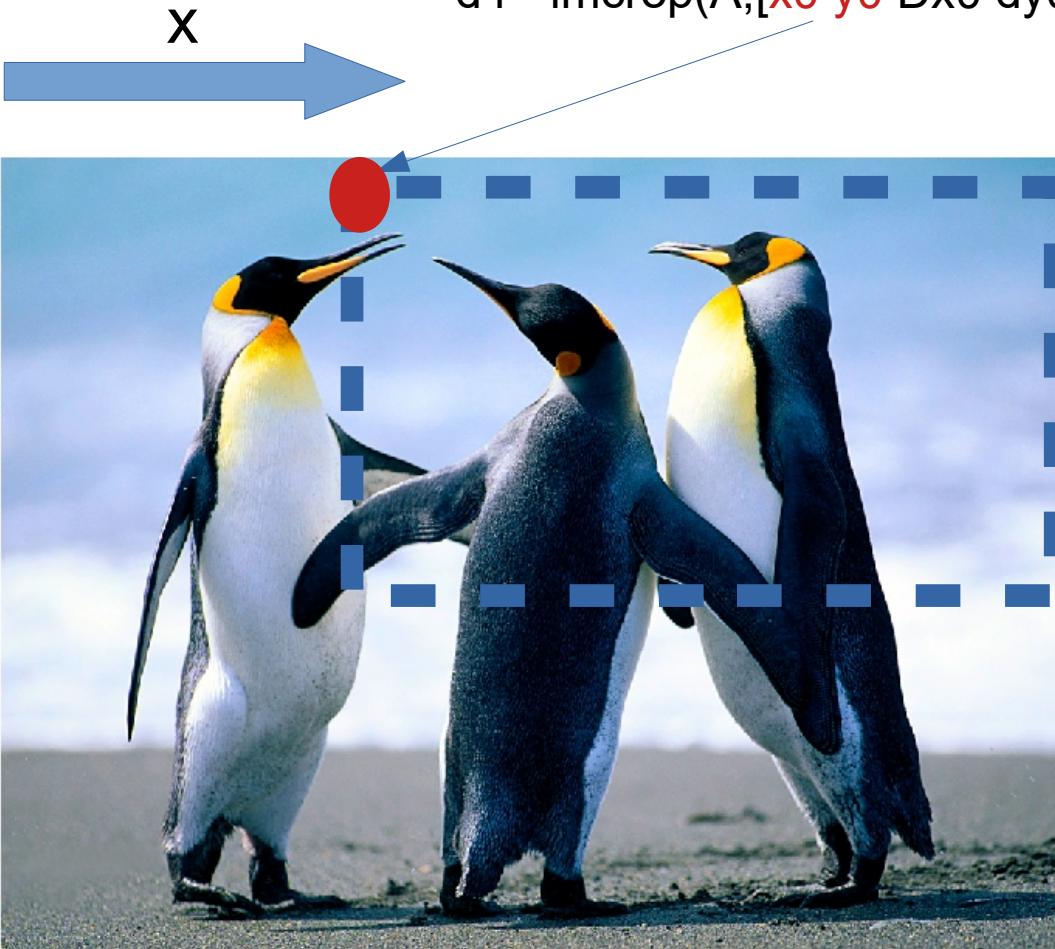
```
>> d3=imcrop(A,[0 0 1024 300]);  
>> imshow(d3)
```



Variables in the current scope:

Attr Name	Size	Bytes	Class
A	768x1024x3	2359296	uint8
d3	300x1024x3	921600	uint8

$d4 = \text{imcrop}(A, [x0\ y0\ Dx0\ dy0])$



Name	Size
=====	=====
A	768x1024x3



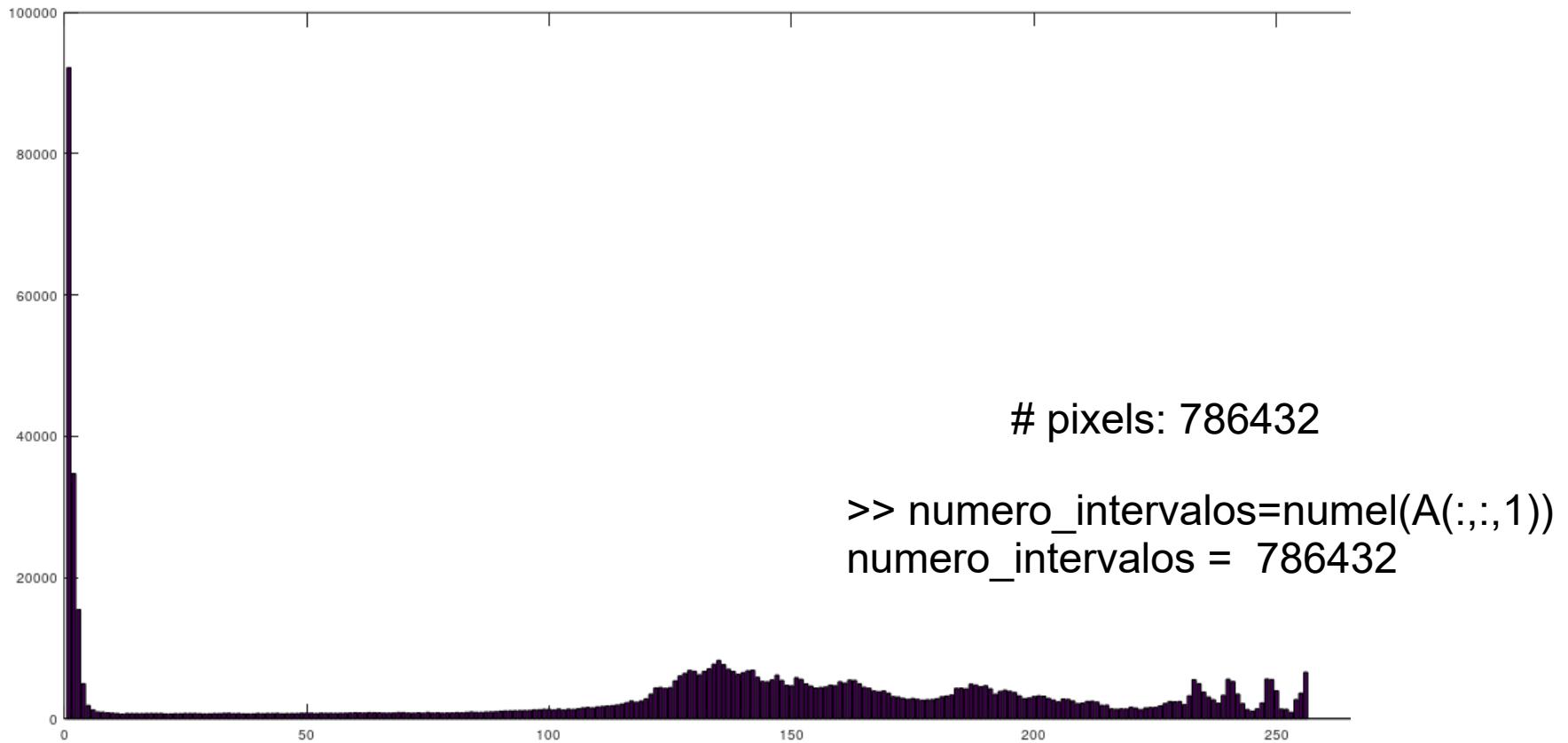
```
d4=imcrop(A,[301 50 723 350]);  
>> figure(4)  
>> imshow(d4)
```

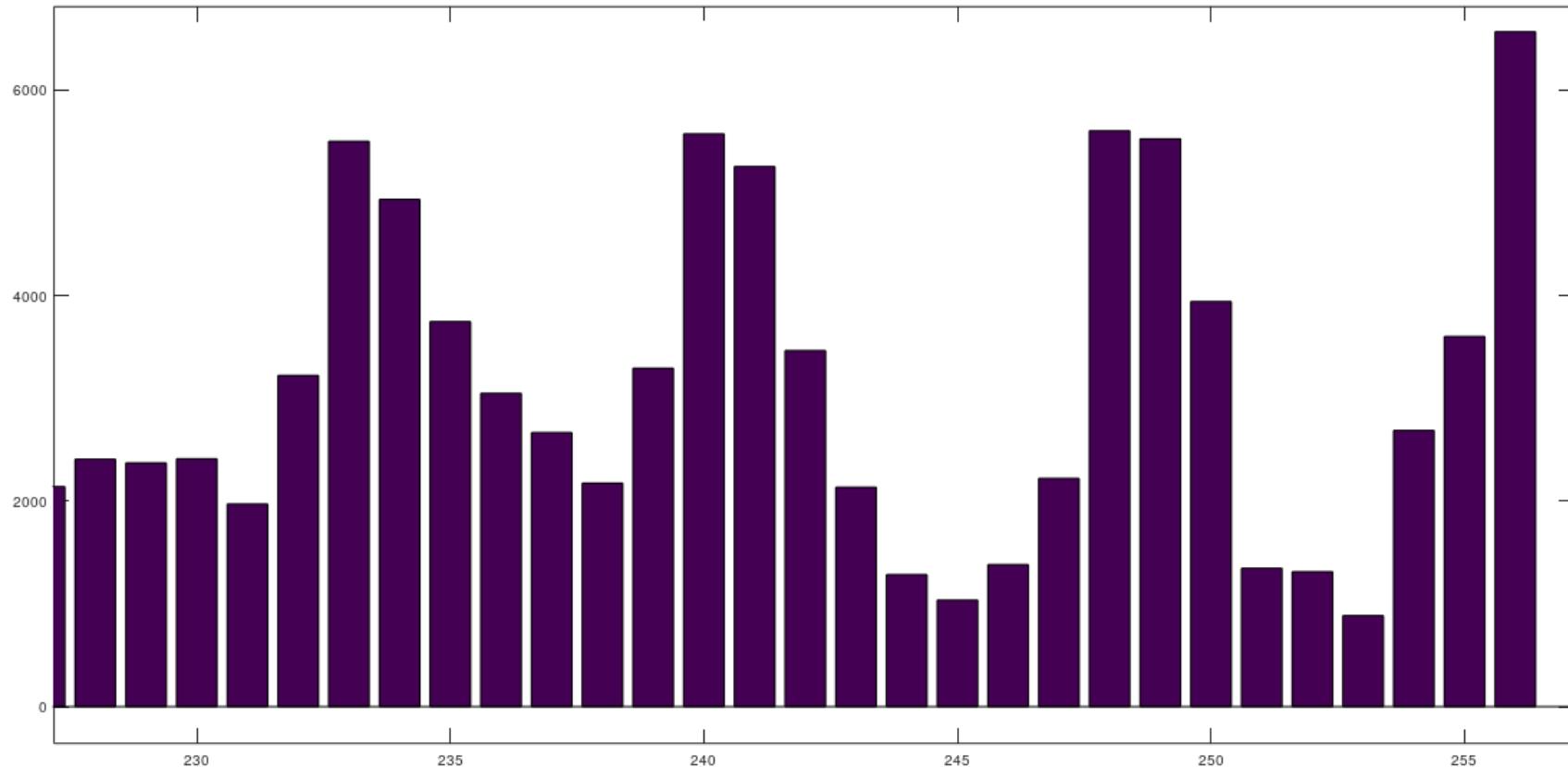
Variables in the current scope:

Attr Name	Size	Bytes	Class
=====	=====	=====	=====
A	768x1024x3	2359296	uint8
d4	351x724x3	762372	uint8

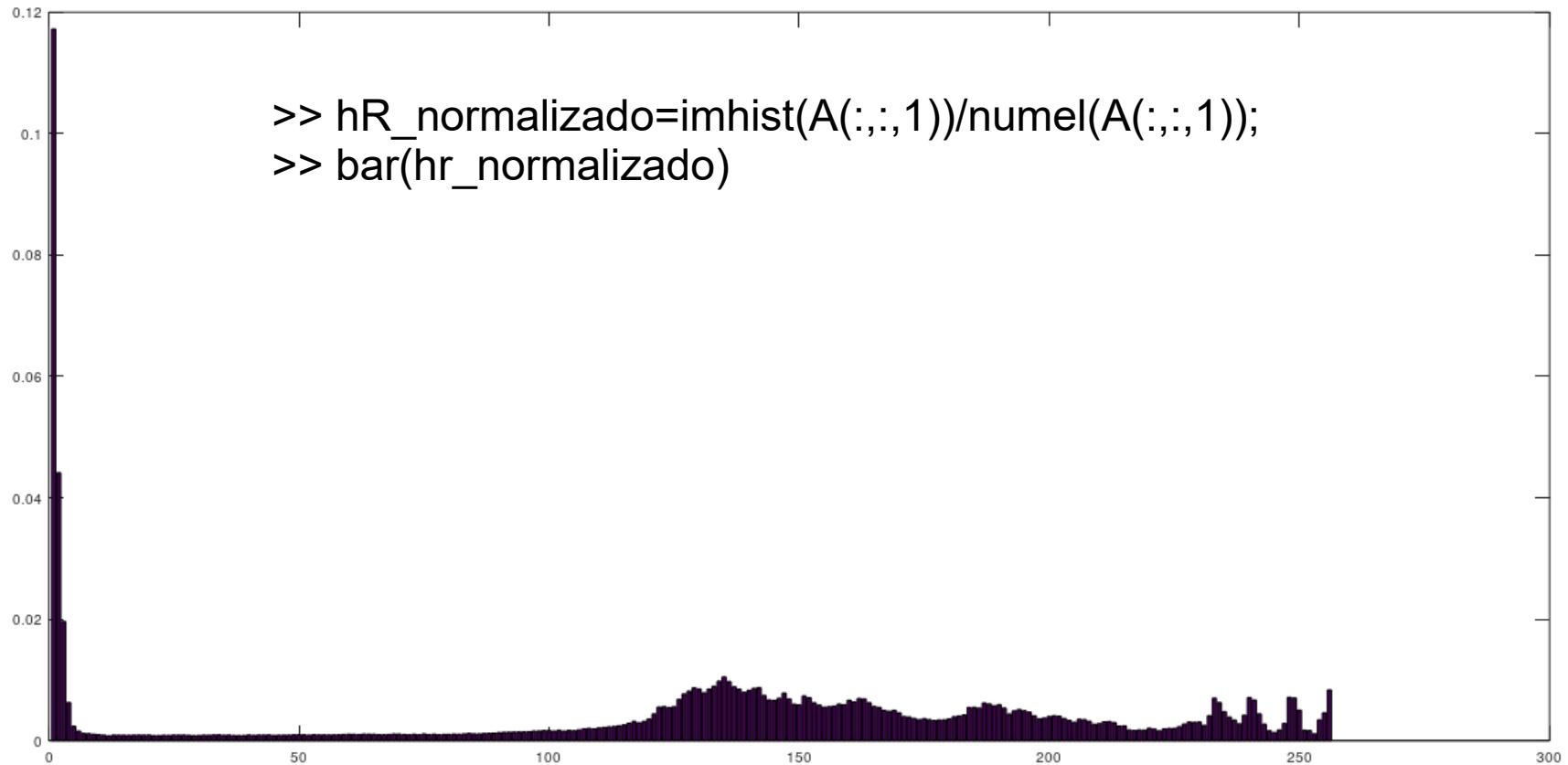
Función imhist

```
>> hR=imhist(A(:,:,1));  
>> bar(hR)
```





Histograma normalizado



Transformaciones de intensidad imadjust

Amodificada=imadjust(A,[low_in high_in],[low_out high_out],gamma)

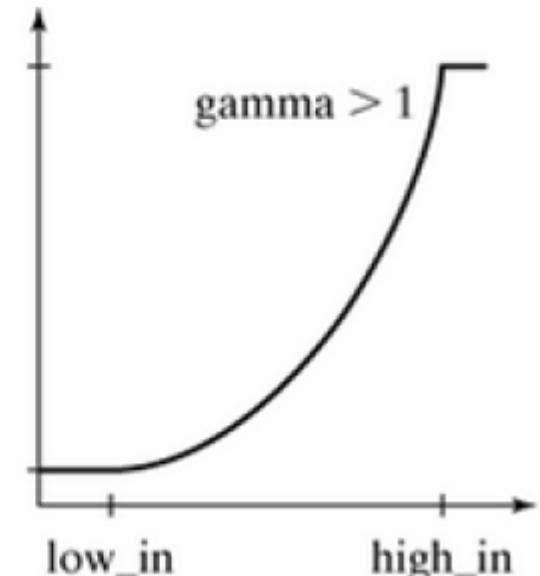
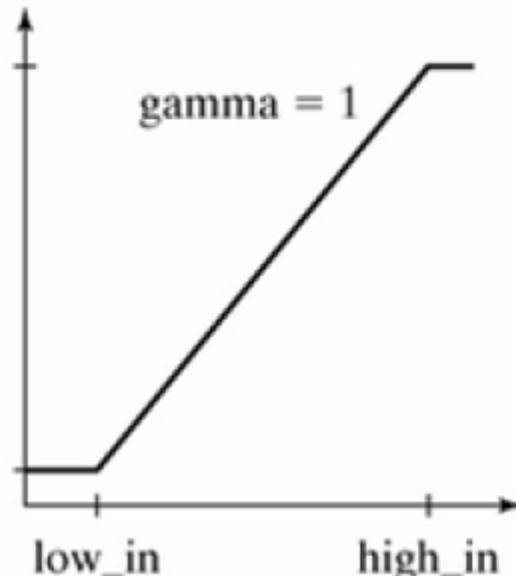
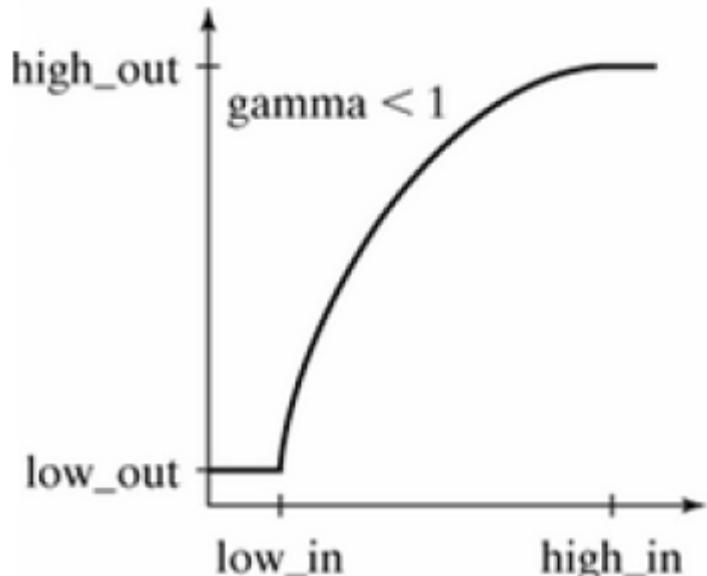


Imagen original



Imagen Transformada

>> Amodificada=imadjust(A,[0.2 1],[0 1],1);

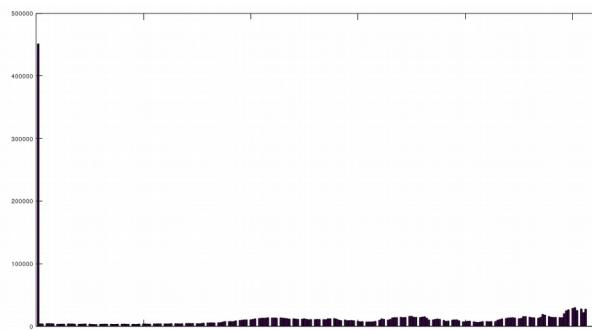
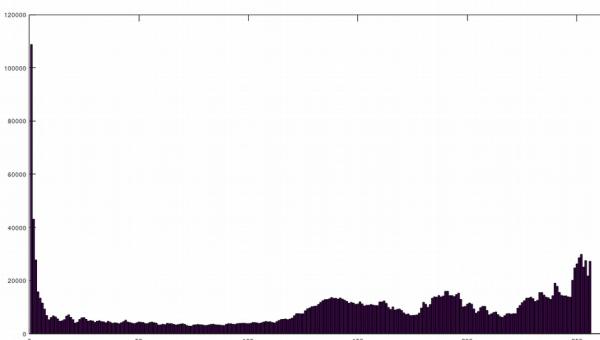
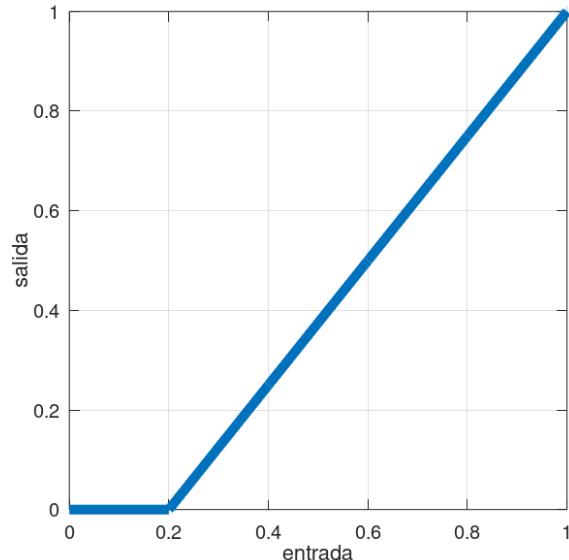


Imagen original

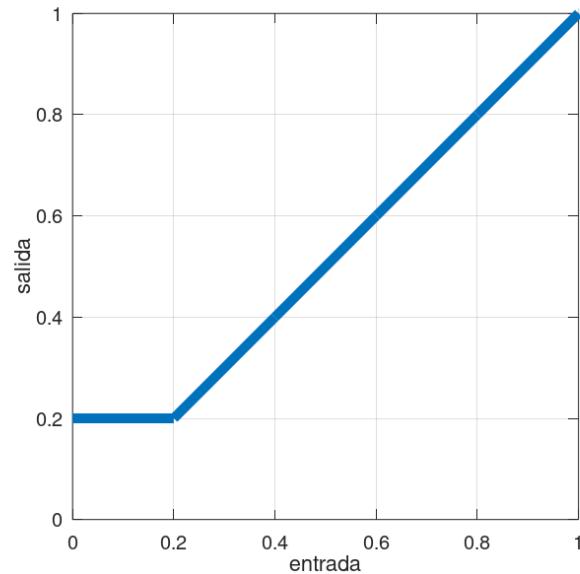


Imagen Transformda

```
>> Amodificada2=imadjust(A,[0.2 1],[0.2 1],1);
```

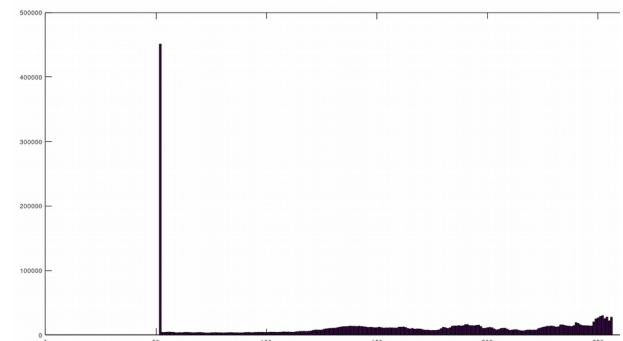
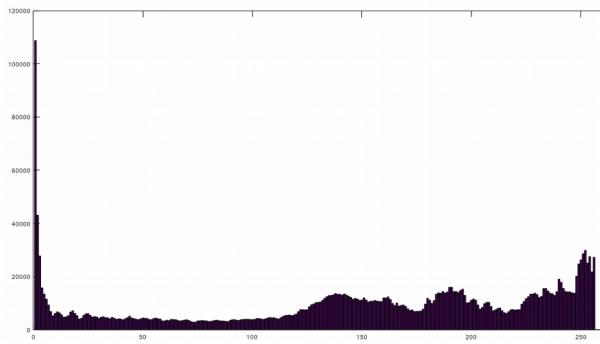


Imagen original



Imagen Transformada

```
>> Amodificada3=imadjust(A,[0.4 0.6],[0 1],1);
```

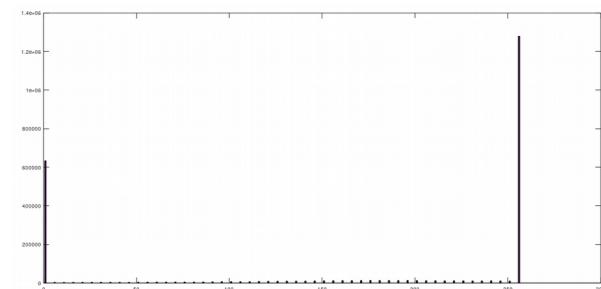
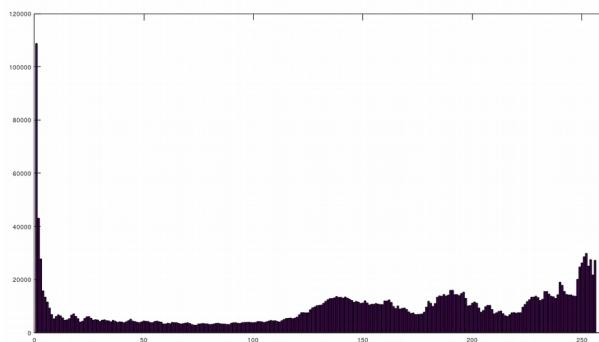
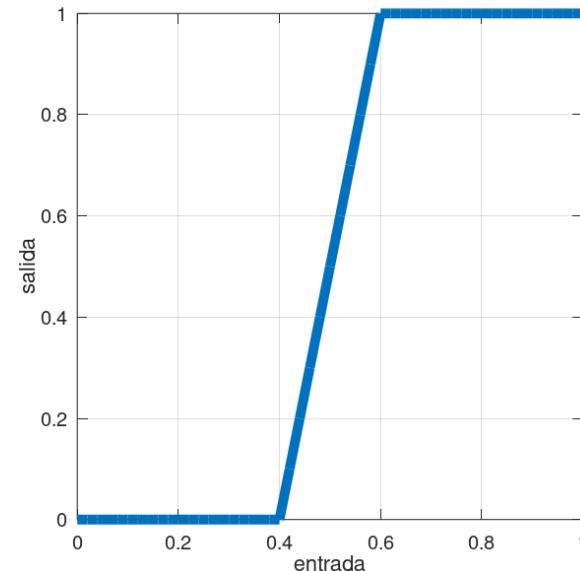


Imagen original



```
>>imadjust(A,[0 1],[0 1],0.2);
```



```
>>imadjust(A,[0 1],[0 1],2);
```

Imagen original

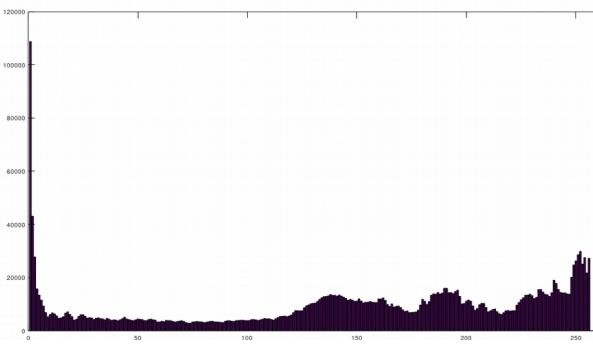
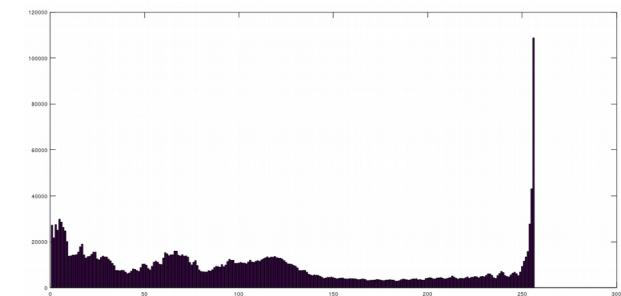
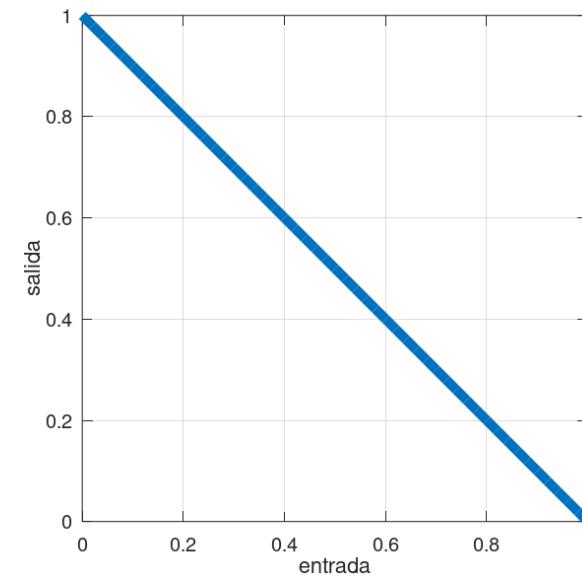


Imagen Transformada

`>>imadjust(A,[0 1],[1 0],1);`



Función padarray



Variables in the current scope:

Attr Name	Size	Bytes	Class
====	=====	=====	=====
A	768x1024x3	2359296	uint8
A2	1168x1224x3	4288896	uint8

```
>> A2=padarray(A,[200 100]);  
>> imshow(A2)
```

Función imrotate



Variables in the current scope:

Attr Name	Size	Bytes	Class
=====	=====	=====	=====
A	768x1024x3	2359296	uint8
R	1267x1267x3	4815867	uint8

```
>> AR=imrotate(A,45);  
>> imshow(AR)
```

Función fliplr



```
>> Af=fliplr(A);  
>> imshow(Af)
```

Función imresize



```
>> Ae=imresize(A,0.2);  
>> imshow(Ae)
```

Attr Name	Size
=====	=====
A	768x1024x3
Ae	154x205x3

Bytes	Class
=====	=====
2359296	uint8
94710	uint8

Attr Name	Size	Bytes	Class
=====	=====	=====	=====
A	768x1024x3	2359296	uint8
Ae	154x205x3	94710	uint8

$$\text{Nuevo Tamaño} \approx \text{Tamaño original} * L^2 : L \in (0,1]$$

$$\text{Nuevo Tamaño} \approx 2359296 * 0.2^2 \approx 94372$$

$$768 * 0.2 \approx 154$$

$$1024 * 0.2 \approx 205$$

Función imfilter

AFiltrada = imfilter(f, w, 'replicate')

- f es la imagen de entrada
- w es la máscara
- 'replicate': el tamaño de la imagen se aumenta duplicando los valores del borde.

$w = \text{ones}(3,3)/9$



$w = \text{ones}(5,5)/25$



$w = \text{ones}(10,10)/100$



$w = \text{ones}(15,15)/(15*15)$



$w = \text{ones}(30,30)/900$



Filtros especiales

```
>> wp=fspecial('prewitt')  
wp =
```

```
1 1 1  
0 0 0  
-1 -1 -1
```



```
>> wk=fspecial('kirsch')  
wk =
```

```
3 3 3  
3 0 3  
-5 -5 -5
```



```
>> ws=fspecial('sobel')  
ws =
```

```
1 2 1  
0 0 0  
-1 -2 -1
```



```
>> wg=fspecial('gaussian',5,0.6)  
wg =
```

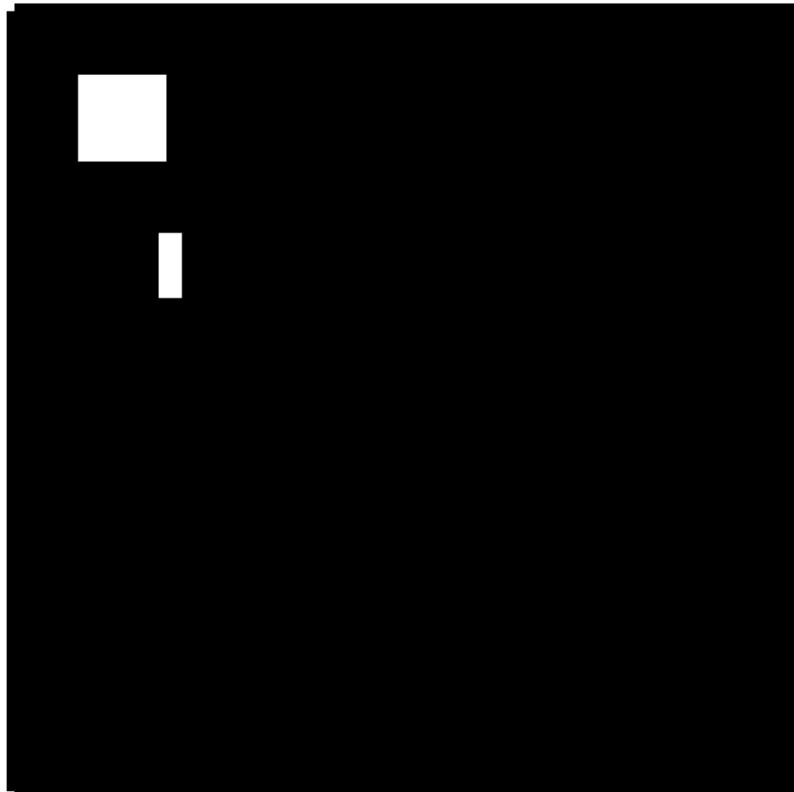
```
0.0000065857 0.0004247805 0.0017035362 0.0004247805 0.0000065857  
0.0004247805 0.0273983822 0.1098782413 0.0273983822 0.0004247805  
0.0017035362 0.1098782413 0.4406547744 0.1098782413 0.0017035362  
0.0004247805 0.0273983822 0.1098782413 0.0273983822 0.0004247805  
0.0000065857 0.0004247805 0.0017035362 0.0004247805 0.0000065857
```



Descripción y representación

Función bwlabel

Matriz de ceros y unos



Attr	Name	Size	Bytes	Class
=====	=====	==	=====	=====
A		100x100	80000	double

[L,n]=bwlabel(A);

Variables in the current scope:

Attr	Name	Size	Bytes	Class
=====	=====	=====	=====	=====
A		100x100	80000	double
L		100x100	80000	double
n		1x1	8	double

>> n

n = 6

>> L(1,1)

ans = 1

>> L(100,100)

ans = 6

Función regionprops

```
>> D=regionprops(L)
```

```
D =
```

6x1 struct array containing the fields:

Area

BoundingBox

Centroid

```
>> D(1).Area
```

```
ans = 1
```

```
>> D(3).Area
```

```
Ans = 121
```

```
>>D(4).Area
```

```
Ans=24
```

```
>> D(4).Centroid
```

```
ans =
```

21.000 33.500

```
>> box=D(4).BoundingBox
```

```
box =
```

19.5000 29.5000 3.0000 8.0000

```
>> L(29:29+8+1,19:19+3+1)
```

```
ans =
```

0 0 0 0 0

0 4 4 4 0

0 4 4 4 0

0 4 4 4 0

0 4 4 4 0

0 4 4 4 0

0 4 4 4 0

0 4 4 4 0

0 4 4 4 0

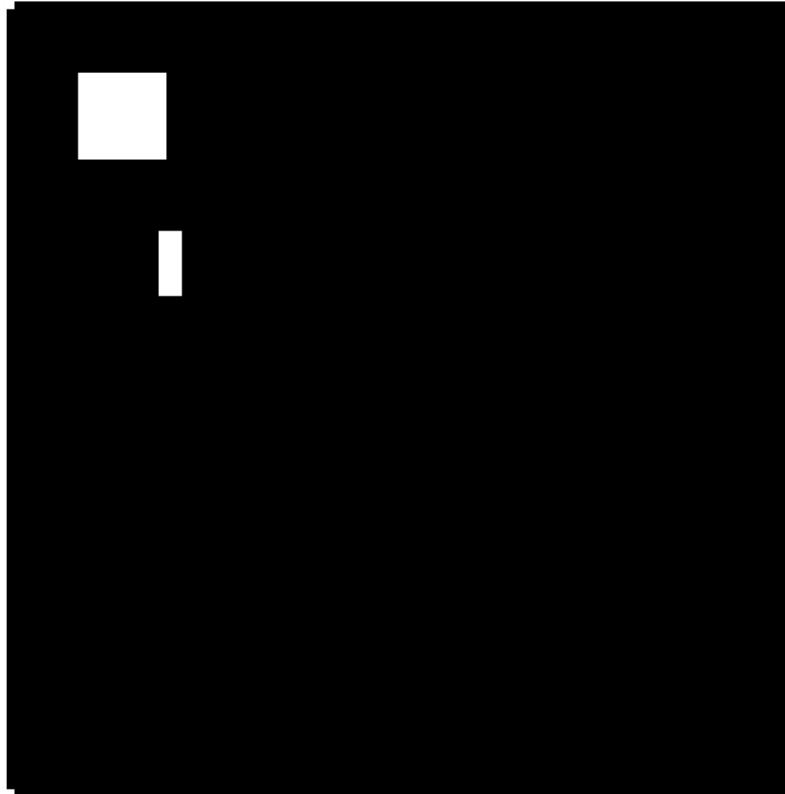
0 0 0 0 0

```
>> D=regionprops(L,'all')
```

```
D =
```

6x1 struct array containing the fields:

- Area
- BoundingBox
- Centroid
- EulerNumber
- Extent
- FilledArea
- FilledImage
- Image
- PixelIdxList
- PixelList
- SubarrayIdx
- ConvexArea
- ConvexHull
- ConvexImage
- Eccentricity
- EquivDiameter
- Extrema
- MajorAxisLength
- MinorAxisLength
- Orientation
- Perimeter



```
>> D(4).Orientation
```

```
ans = 90
```

```
>> D(3).Orientation
```

```
ans = 0
```

Imagen original



Imagen negativa

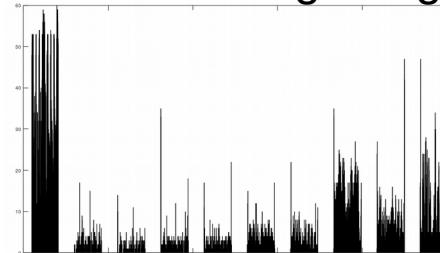


```
imadjust(A,[0 1],[1 0]);
```

Imagen en escala de grises



```
Agris=rgb2gray(A);
```



```
>> t=graythresh(Agris)  
t = 0.42353
```

```
hist(Agris)
```

Imagen binaria



```
Abin=im2bw(Agris,t)
```

Morfología

Imagen binaria



```
>> se = strel('rectangle',[3 3])  
se =  
Flat STREL object with 9 neighbors
```

Neighborhood:

```
1 1 1  
1 1 1  
1 1 1
```

Imagen dilatada



```
Abin_dilate=imdilate(Abin,se);
```

Imagen erosionada



```
Abin_rode=imerode(Abin_dilate,se);
```



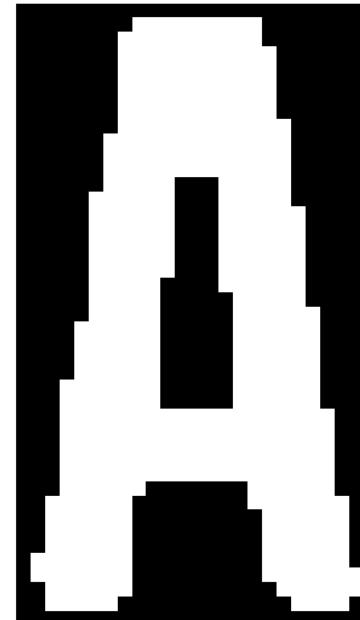
ABC 1234

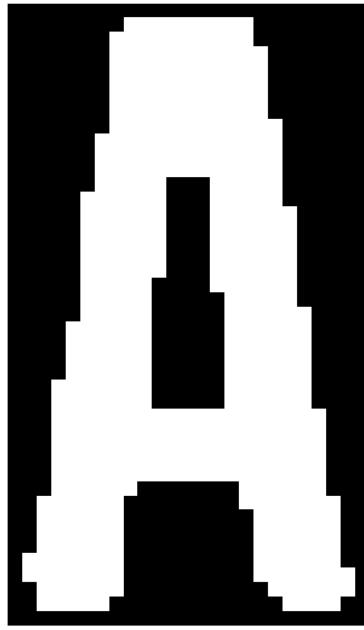
```
>> [L,n]=bwlabel(Abin_rode);
>> D=regionprops(L);
>> D
D =
```

8x1 struct array containing the fields:

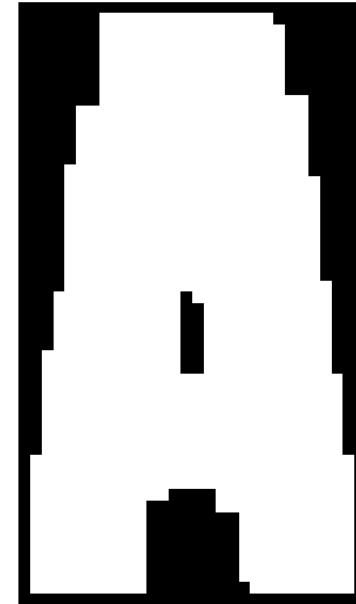
- Area
- BoundingBox
- Centroid

```
Box=D(1).BoundingBox;
x=Box(1)-0.5;
y=Box(2)-0.5;
dx=Box(3);
dy=Box(4);
imshow(Abin_rode(y:y+dy+1,x:x+dx+1))
```





```
A1=Abin_rode(y:y+dy+1,x:x+dx+1)
```



```
A2=imresize(A1,[52 30])
```

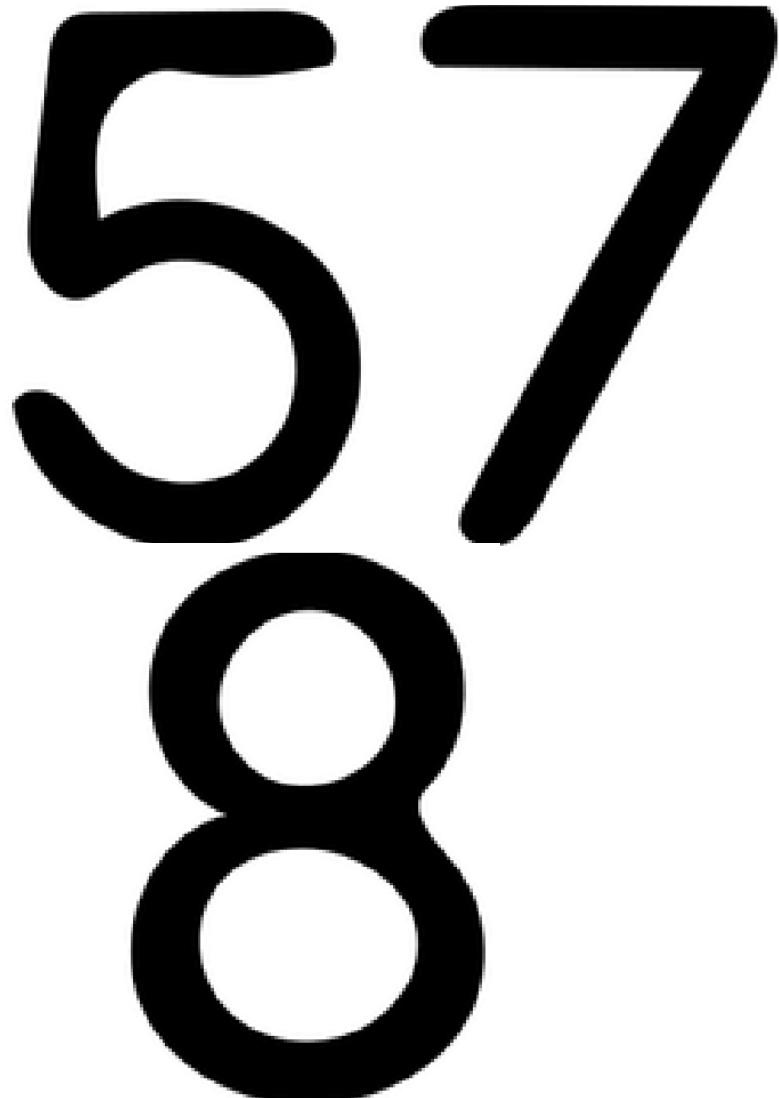
Variables in the current scope:

Attr Name	Size	Bytes	Class
A1	43x25	1075	logical
A2	52x30	1560	logical

```
>> A5=imread('number5.png');
>> A8=imread('number8.png');
>> A7=imread('number7.png');
>> whos
Variables in the current scope:
```

Attr	Name	Size	Bytes	Class
=====	=====	=====	=====	=====
	A5	179x127x3	68199	uint8
	A7	180x124x3	66960	uint8
	A8	183x119x3	65331	uint8

Total is 200490 elements using 200490 bytes



5

7

8

```
>> A5m=imresize(A5,[180 124]);  
>> A7m=imresize(A7,[180 124]);  
>> A8m=imresize(A8,[180 124]);  
>> whos
```

Variables in the current scope:

Attr	Name	Size	Bytes	Class
		=====	=====	=====
	A5	179x127x3	68199	uint8
	A5m	180x124x3	66960	uint8
	A7	180x124x3	66960	uint8
	A7m	180x124x3	66960	uint8
	A8	183x119x3	65331	uint8
	A8m	180x124x3	66960	uint8

```
>> corr2(A5m,A5m)  
ans = 1.0000  
>> corr2(A5m,A7m)  
ans = 0.086218  
>> corr2(A5m,A8m)  
ans = 0.60644  
>> corr2(A7m,A8m)  
ans = 0.14581
```