

Práctica 11. Alejandro Ruiz & David Morais.....	1
Binarización usando la técnica del promedio de vecinos	2
Etiquetado de imágenes.....	3
Ejercicio. Encontrar defectos binarizando, etiquetando y analizando sus propiedades	6

Práctica 11. Alejandro Ruiz & David Morais

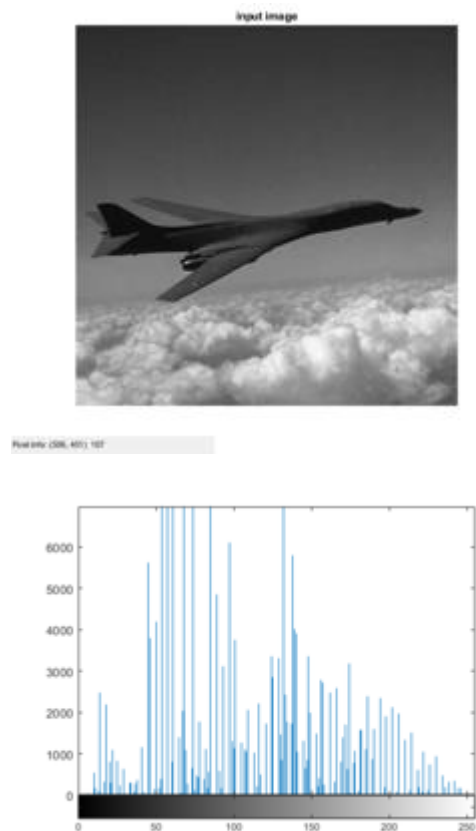
```

im = imread('airplane.tif');
figure, imshow(im), title('input image'), impixelinfo;
% Analizamos la imagen para ver si existen dos regiones para poder
% binarizar
figure, imhist(im);
% Binarizar comparando con un umbral suele dar malos resultados en función
% de la imagen y la iluminación
bin=im>70;
figure, imshow(bin), title('fallo');

% Recortamos la imagen para intentar binarizar la parte que nos interesa
%troc=imcrop(im);
%bin=troc>70;
%figure, imshow(troc), title('imagen recortada');

% Umbral de Otsu para binarización de imágenes
th=graythresh(im);
imbw=imbinarize(im,th);
figure,imshow(imbw), title('binarizacion con Otsu');

```



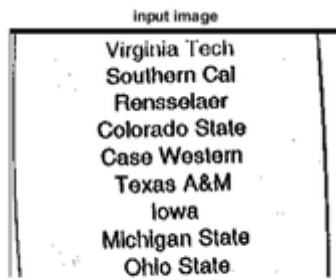


Binarización usando la técnica del promedio de vecinos

```
im=imread("textsheet.jpg");
figure,imshow(im),title('input image');

kernel = ones(50)/50/50;
im3=imfilter(double(im),kernel,'replicate');
%figure,imshow(im3,[]),title('imagen filtrada')
im4=im>(im3-20);
figure,imshow(im4),title('input image');
```





Etiquetado de imágenes

```

im=imread("arros.tif");
figure,imshow(im),title('Input image');

% Primero la filtramos
ee=strel('disk',20,0);
grans=imtophat(im,ee);
figure,imshow(grans),title('arroses');

% Ahora es fácilmente binarizable
imbw=im2bw(grans,graythresh(grans));
figure,imshow(imbw),title('imagen binarizada');

% Los objetos que están en el contorno de la imagen no los debo procesar
% porque me darán falsos resultados
marker=imbw;
marker(2:end-1,2:end-1)=0;
rec=imreconstruct(marker,imbw);
figure,imshow(rec),title('Arroses de los bordes');
arrozCentro=imsubtract(imbw,rec);
figure,imshow(arrozCentro),title('Arroses del centro');

% Labelling con conectividad 4
label=bwlabel(arrozCentro,4);
figure,imshow(label,[]),title('imagen etiquetada'), colormap colorcube;
max(label(:))

datos=regionprops(label,'all');
areas=[datos.Area];
figure,hist(areas,20),title('Areas');

clear;

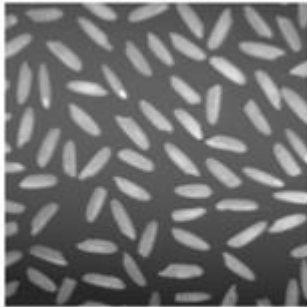
```

ans =

binarización con Otsu



Input image



arroz

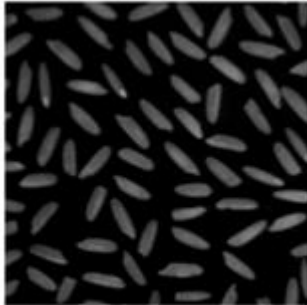


imagen binarizada



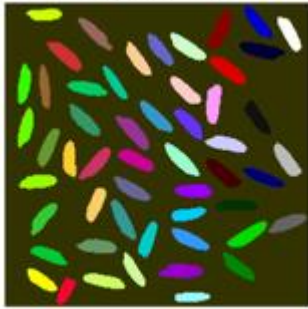
Arroces de los bordes



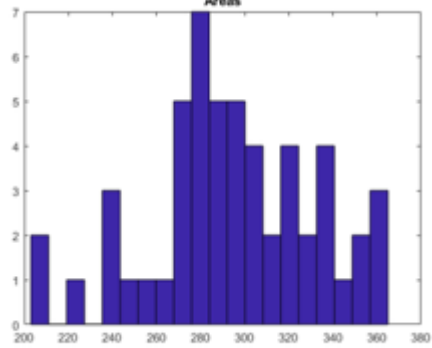
Arroces del centro



imagen etiquetada

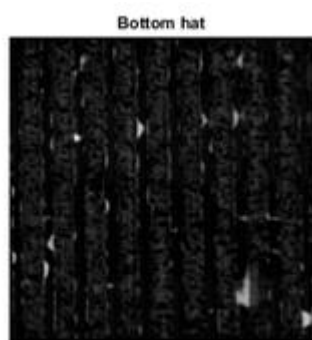
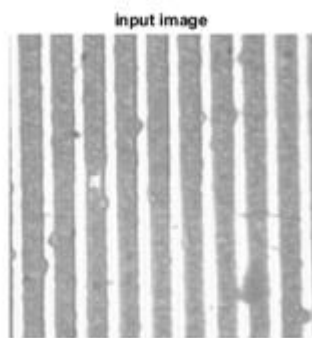


Areas

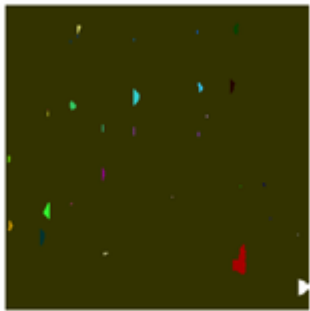
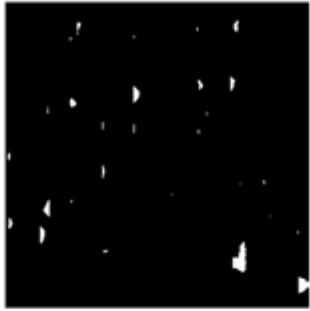


Ejercicio. Encontrar defectos binarizando, etiquetando y analizando sus propiedades

```
im=imread('r4x2_256.tif');
figure,imshow(im),title('input image');
% Para eliminar pequeñas estructuras negras
ee=strel('rectangle', [50,2]);
bh=imbothat(im,ee);
figure,imshow(bh,[]), title('Bottom hat'),impixelinfo;
% Miramos los valores de la imagen, para ver que threshold usar para
% binarizar la imagen. En esta imagen se puede hacer de esta manera poque
% existen regiones diferenciadas por el procesado anterior.
imbw=bh>50;
figure,imshow(imbw);
% Etiquetamos los diferentes defectos
label=bwlabel(imbw,8);
figure, imshow(label,[]), colormap colorcube;
% Finalmente obtenemos las propiedades de los diferentes defectos presentes
% en la PCB
props=regionprops(label,'Area');
area=[props.Area];
```



Pixel info: (X, Y) Pixel Value



Published with MATLAB® R2024a