

T-Memoria-de-las-practicas.pdf



Ruben_Bueno_Menendez



Teledetección



4º Grado en Ingeniería Informática - Ingeniería del Software



Escuela Técnica Superior de Ingeniería Informática
Universidad de Sevilla



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.





Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

Teletelección

Rubén Bueno Menéndez
Correo: rubbuemen@alum.us.es

Memorias de las prácticas



Departamento de
Tecnología Electrónica



Índice

Práctica 1	3
Práctica 2	4
Práctica 3	8
Práctica 4	12
Práctica 5	17
Práctica 6	26
Práctica 7	29

Aviso:

- Se supondrán que todas las imágenes indicadas están dentro de una carpeta llamada "imágenes" y que el código se ejecuta en la carpeta anterior a esta.
- En algunas funciones se reutilizan funciones desarrolladas anteriormente pero no se vuelve a adjuntar el código de esta.



**KEEP
CALM
AND
ESTUDIA
UN POQUITO**

Práctica 1

Código 1.6

```
function [xord] = qsort(X)
if isempty(X)
    xord = X;
else
    x = randi(length(X)); %Generamos un número aleatorio entre 1 y el tamaño del vector
    p = X(x); %Extraemos como pivote el valor de la posición x
    X(x) = []; %Eliminamos el pivote del vector
    a = X(X <= p); %Subtraemos un vector para todos los elementos menor o igual al pivote
    b = X(X > p); %Subtraemos un vector para todos los elementos mayor al pivote
    aord = qsort(a);
    bord = qsort(b);
    xord = [aord, p, bord];
end
```

Práctica 2

Código 2.3

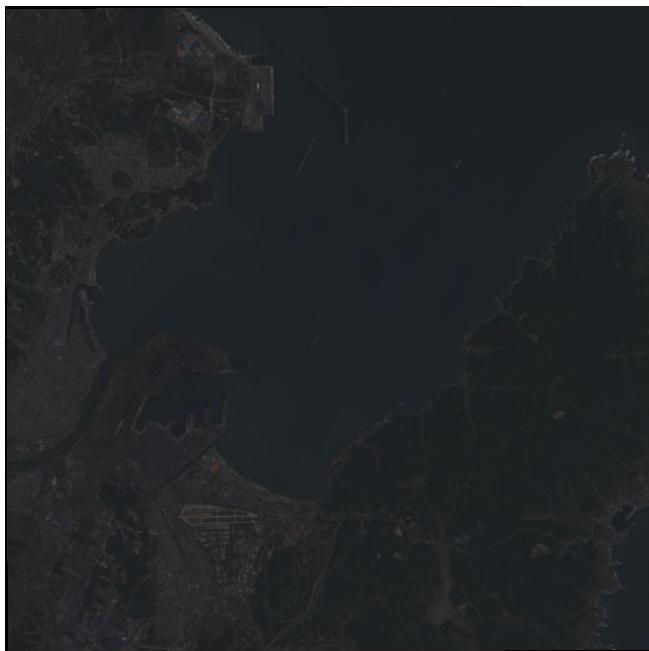
```
function im2 = corte(im1, p, b)
im1 = im1(:,:,b); %Se indica la banda a extraer para que sea monobanda
numPixels = sum(im1(:) > 0);
pixelsPercentage = numPixels * (p / 100);
h = histo(im1, 1);
ac = 0;
for v = 1:length(h)
    ac = ac + h(v);
    if ac >= pixelsPercentage
        m = v;
        break
    end
end
ac = 0;
for v = length(h):-1:1
    ac = ac + h(v);
    if ac >= pixelsPercentage
        M = v;
        break
    end
end
end
```

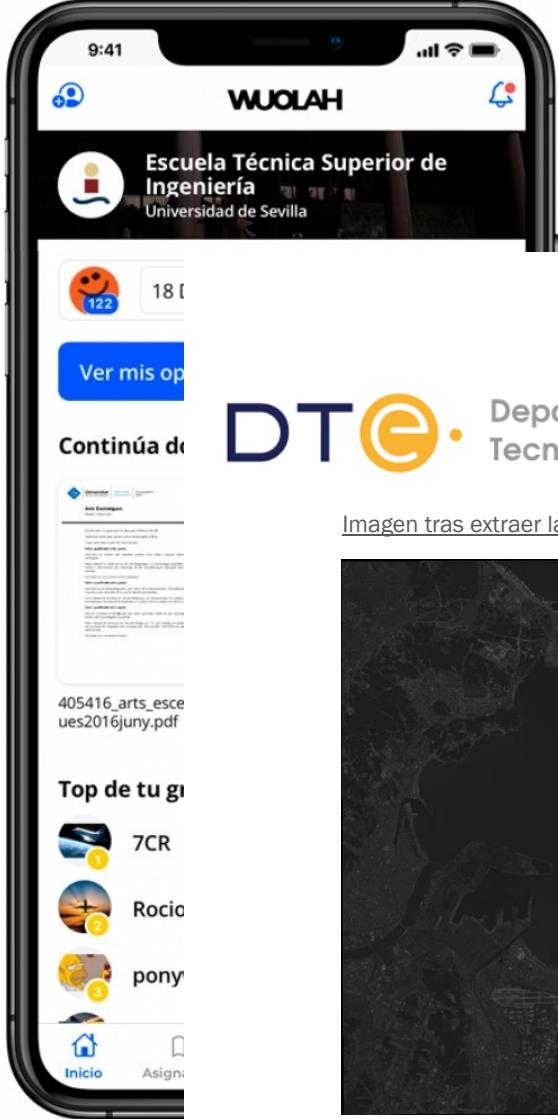
Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```
im = imread('imágenes/imagen_memoria.TIF');
corte(im, 1, 1);
```

Imagen propia tratada (antes/después)

Imagen original:





Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

DTE • Departamento de
Tecnología Electrónica

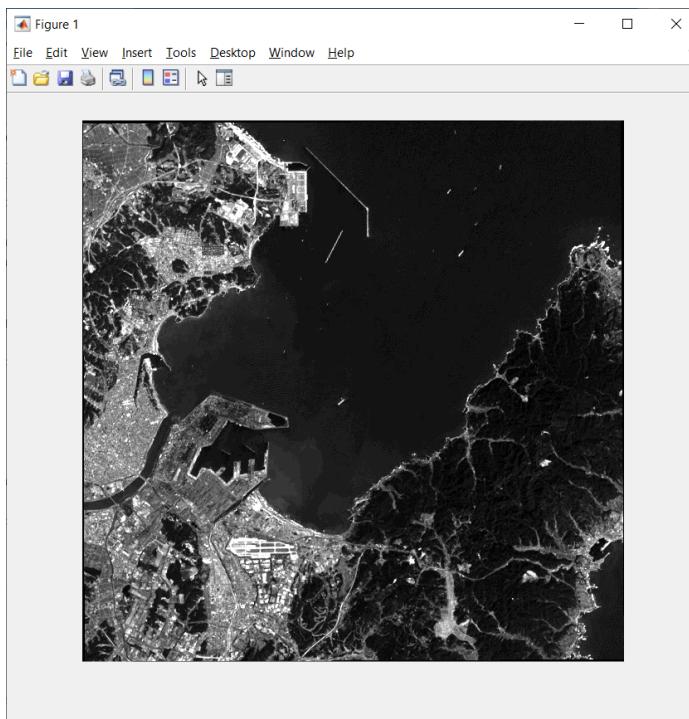


Rubén Bueno Menéndez

Imagen tras extraer la banda roja:



Imagen después de tratarla:



Reservados todos los derechos.
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Código 2.4

```
function im2 = ecual(im1, b)
im1 = im1(:,:,b); %Se indica la banda a extraer para que sea monobanda
[F, C] = size(im1);
im2 = uint8(zeros(F, C));
h = histo(im1, 1);
ac = 0;
for v = 1:length(h)
    ac = ac + h(v);
    h(v) = ac;
end
P = sum(im1(:) > 0);
FE = 255 / P;
for f = 1:F
    for c = 1:C
        nd = im1(f, c);
        if nd > 0
            ndp = round(h(nd) * FE);
            im2(f, c) = ndp;
        end
    end
end
imshow(im2);
```

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```
im = imread('imágenes/imagen_memoria.TIF');
ecual(im, 1);
```

Imagen propia tratada (antes/después)

Imagen original:

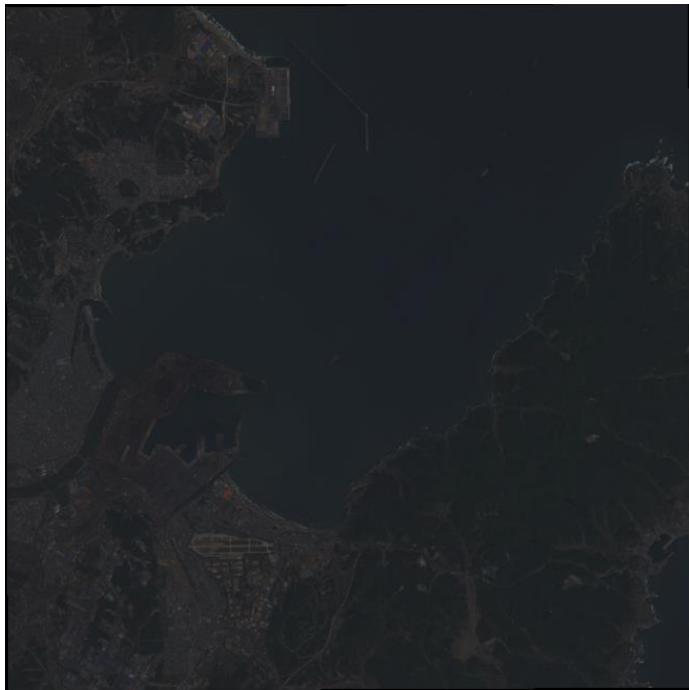


Imagen tras extraer la banda roja:

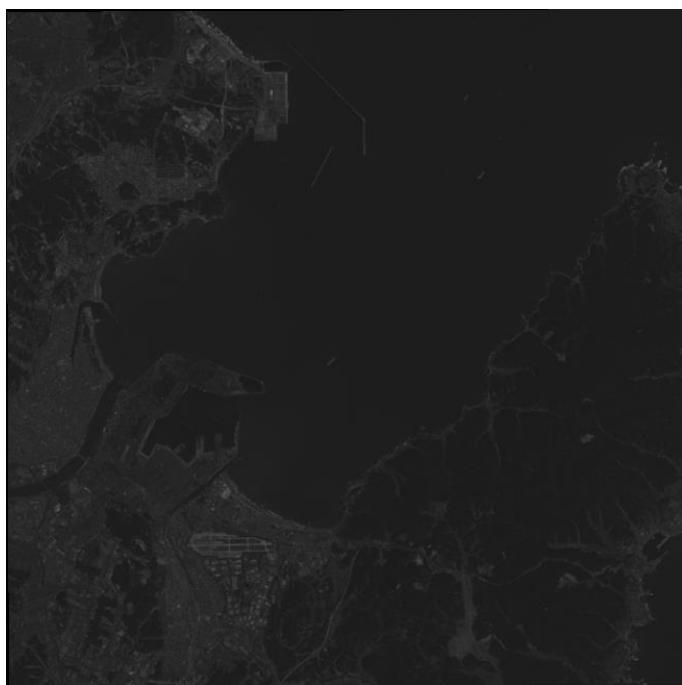
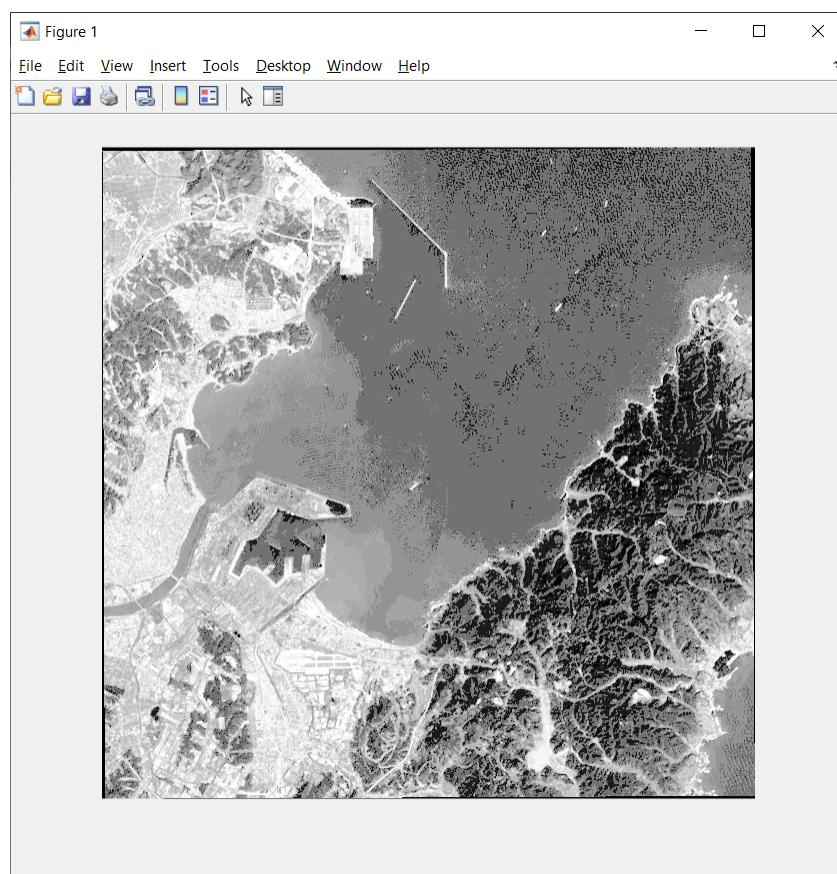


Imagen después de tratarla:



Práctica 3

Código 3.3

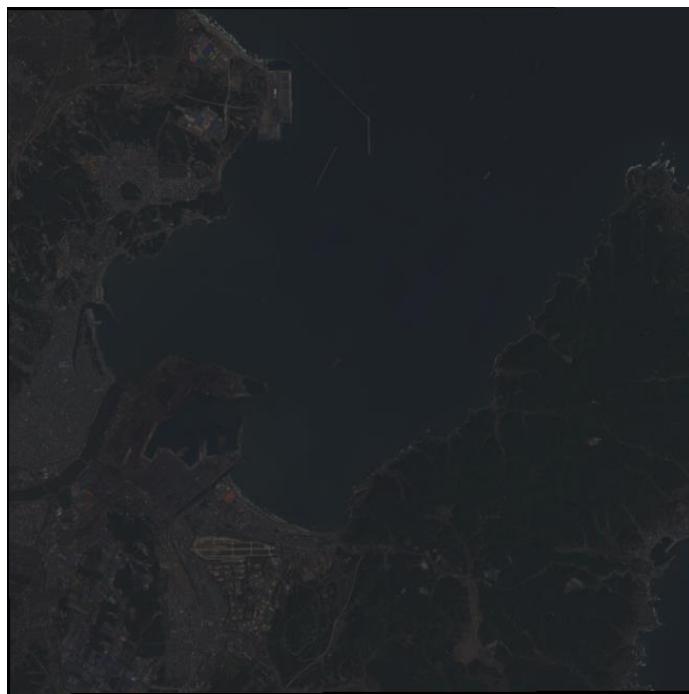
```
function im2 = filtro(im1, cf, b)
im1 = im1(:,:,b); %Se indica la banda a extraer para que sea monobanda
[F, C] = size(im1);
im2 = uint8(zeros(F, C));
cf = cf ./ sum(sum(cf)); %Se normaliza para que la suma de los coeficientes sea 1
for f = 2:F-1 %Los bordes de la imagen no se tienen en cuenta, así que se excluyen
    for c = 2:C-1
        nd = im1(f,c);
        if nd > 0
            en = double(im1(f-1:f+1, c-1:c+1));
            ndp = sum(sum(en .* cf));
            im2(f, c) = ndp;
        end
    end
end
imshow(im2);
```

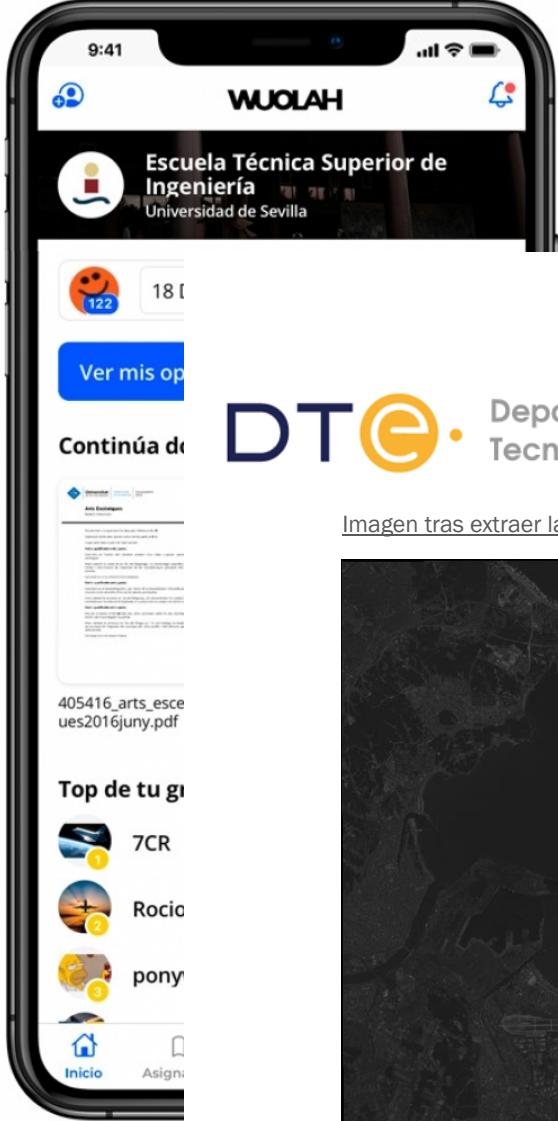
Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```
im = imread('imágenes/imagen_memoria.TIF');
cf = ones(3,3);
filtro(im, cf, 1);
```

Imagen propia tratada (antes/después)

Imagen original:





Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the App Store

GET IT ON Google Play

Imagen tras extraer la banda roja:

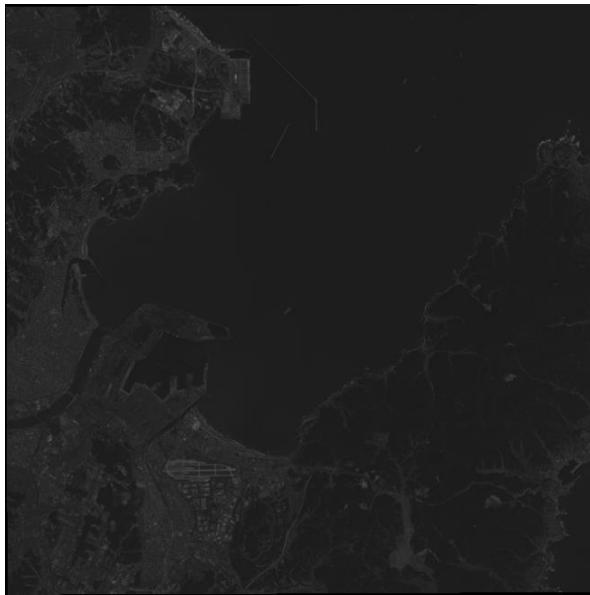
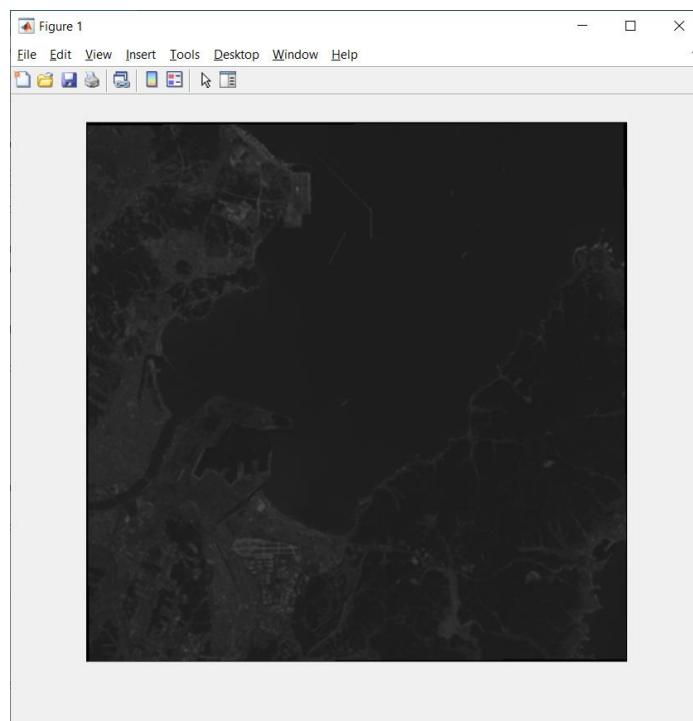


Imagen después de tratarla:



Rubén Bueno Menéndez

Reservados todos los derechos.
No se permite la explotación económica ni la transformación de esta obra. Queda permitida la impresión en su totalidad.

Código 3.4

```
function im2 = mediana(im1, b)
im1 = im1(:,:,b); %Se indica la banda a extraer para que sea monobanda
[F, C] = size(im1);
im2 = uint8(zeros(F, C));
for f = 2:F-1 %Los bordes de la imagen no se tienen en cuenta, así que se excluyen
    for c = 2:C-1
        en = double(im1(f-1:f+1, c-1:c+1));
        maximo = max(en(:));
        minimo = min(en(:));
        ndp = median(reshape(en, 1, 9));
        if ndp - minimo >= 50 || maximo - ndp >= 50
            im2(f, c) = ndp;
        else
            im2(f, c) = im1(f, c);
        end
    end
end
imshow(im2);
```

Mi mejora sigue el criterio de que haya una diferencia de al menos un valor de 50 entre la mediana y su valor máximo o mínimo del entorno.

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

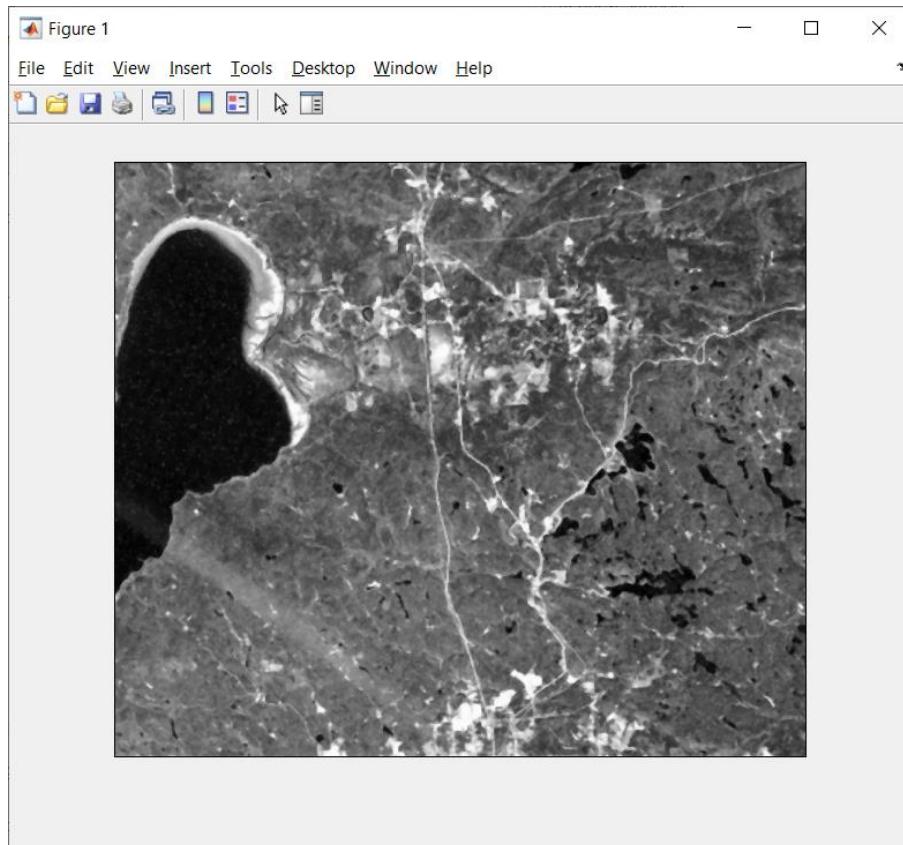
```
im = imread('imágenes/imagen_memoria_ruido_monobanda.jpg');
mediana(im, 1);
```

Imagen propia tratada (antes/después)

Imagen monobanda con ruido:



Imagen después de tratarla:



Práctica 4

Código 4.2

```

function im2 = amplia(im1, R, b)
im1 = im1(:,:,b); %Se indica la banda a extraer para que sea monobanda
[F, C] = size(im1);
if R >= 1
    Fr = F * R;
    Cr = C * R;
else
    error('El valor insertado de R no es válido, inserte uno mayor a 1');
end
im2 = zeros(Fr, Cr);
im2(1:R:Fr, 1:R:Cr) = im1;
for f = 1:R:Fr
    for c = 2:(Cr-(R-1))
        if mod(c, R) == 0
            nd1 = im2(f, c-(R-1)); %Pixel de la izquierda con valor
            nd2 = im2(f, c+1);      %Pixel de la derecha con valor
            ndp = (nd1 + nd2) / 2; %Media de ambos valores para rellenar
            for e = c-(R-2):c
                im2(f, e) = ndp;
            end
        end
    end
end
for f = 2:(Fr-(R-1))
    if mod(f, R) == 0
        f1 = im2(f -(R-1), :);
        f2 = im2(f + 1, :);
        fnd = (f1 + f2) / 2;
        for e = f-(R-2):f
            im2(e, :) = fnd;
        end
    end
end
%Finalmente rellenamos las líneas negras finales con la última línea buena:
for e = Fr:-1:Fr-(R-1)
    im2(e, :) = im2(Fr-R, :);
end
for e = Cr:-1:Cr-(R-1)
    im2(:, e) = im2(:, Cr-R);
end
im2 = uint8(im2);
imshow(im2);

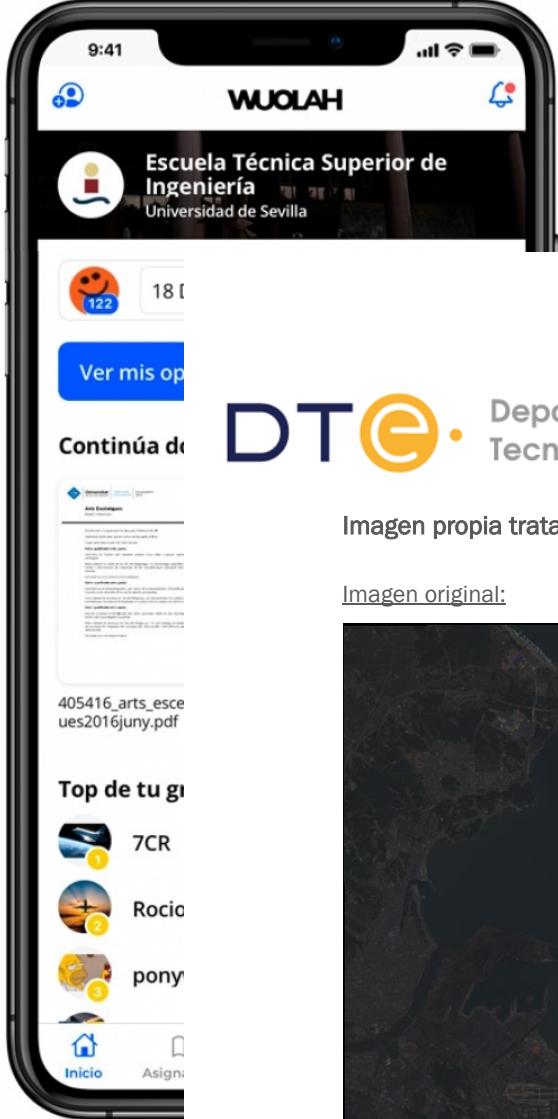
```

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```

im = imread('imágenes/imagen_memoria.TIF');
R = 2;
amplia(im, R, 1);

```



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

Rubén Bueno Menéndez

Imagen propia tratada (antes/después)

Imagen original:

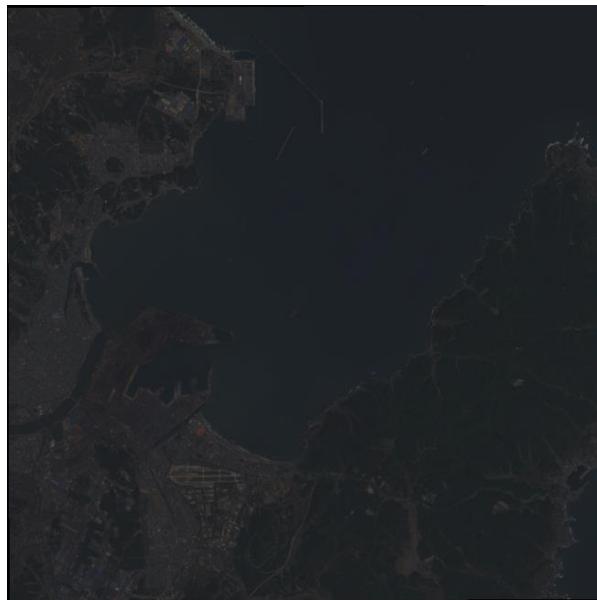
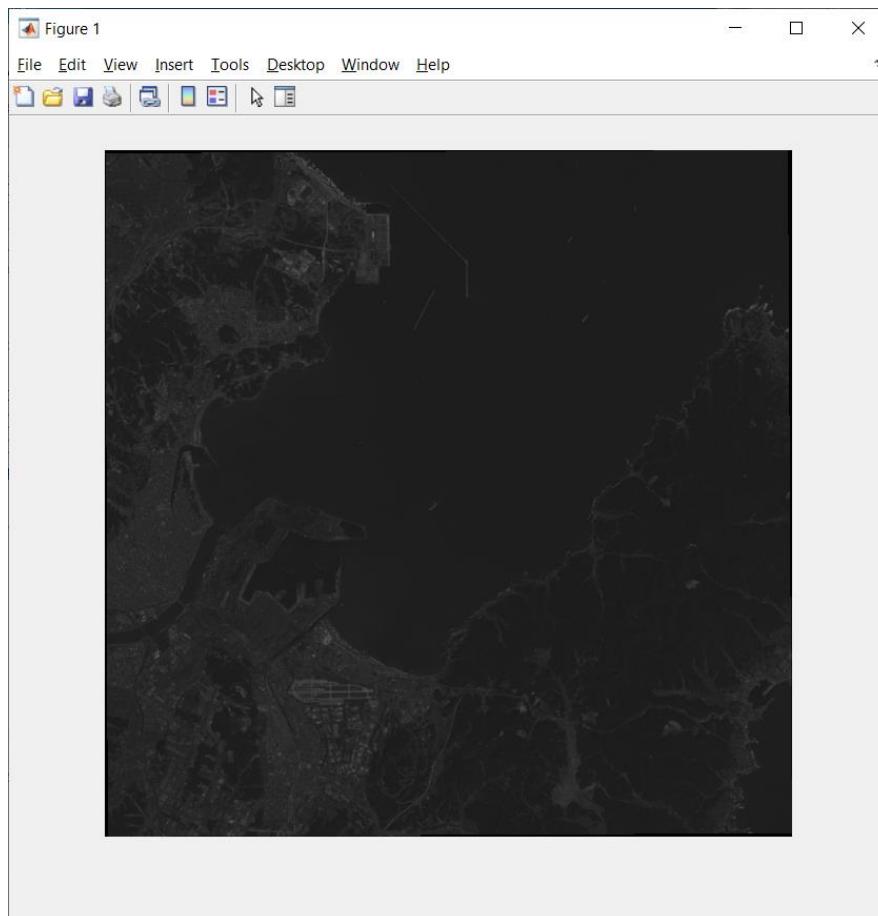


Imagen tras extraer la banda roja:



Imagen después de tratarla:



Código 4.4

```
function im2 = ampliafft(im1, R, b)
im1 = im1(:,:,b); %Se indica la banda a extraer para que sea monobanda
[F, C] = size(im1);
if R > 1
    Fr = F * R;
    Cr = C * R;
else
    error('El valor insertado de R no es válido, inserte uno mayor a 1');
end
IM1 = fft2(im1);
IM2 = zeros(Fr, Cr);
IM2(1:F/2, 1:C/2) = IM1(1:F/2, 1:C/2); %Primer cuadrante
IM2(Fr-(F/2)+1:Fr, 1:C/2) = IM1((F/2)+1:F, 1:C/2); %Segundo cuadrante
IM2(1:F/2, Cr-(C/2)+1:Cr) = IM1(1:(F/2), (C/2)+1:C); %Tercer cuadrante
IM2(Fr-(F/2)+1:Fr, Cr-(C/2)+1:Cr) = IM1((F/2)+1:F, (C/2)+1:C); %Cuarto cuadrante
im2 = real(ifft2(IM2)) * ((Fr*Cr)/(F*C)); %Recuperamos la imagen adecuando su amplitud
im2 = uint8(im2);
imshow(im2);
```

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```
im = imread('imágenes/imagen_memoria.TIF');
R = 8;
ampliafft(im, R, 1);
```

Imagen propia tratada (antes/después)

Imagen original:

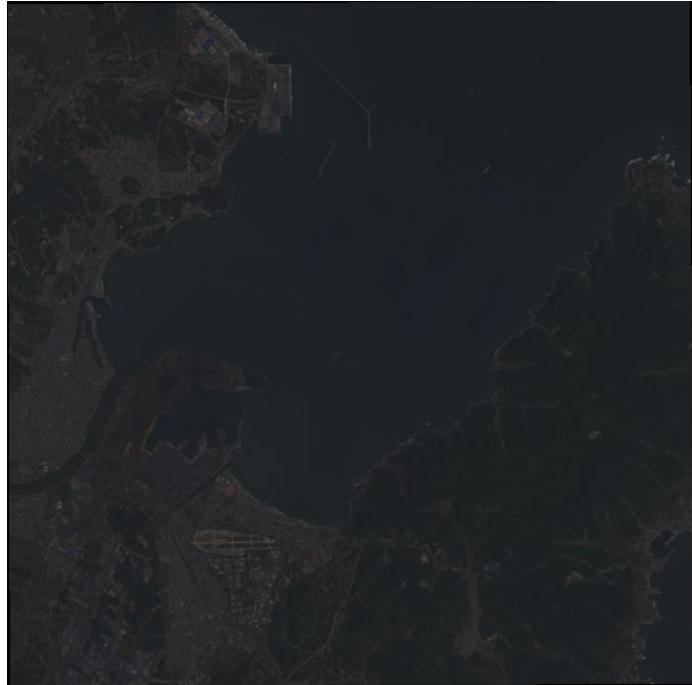


Imagen tras extraer la banda roja:

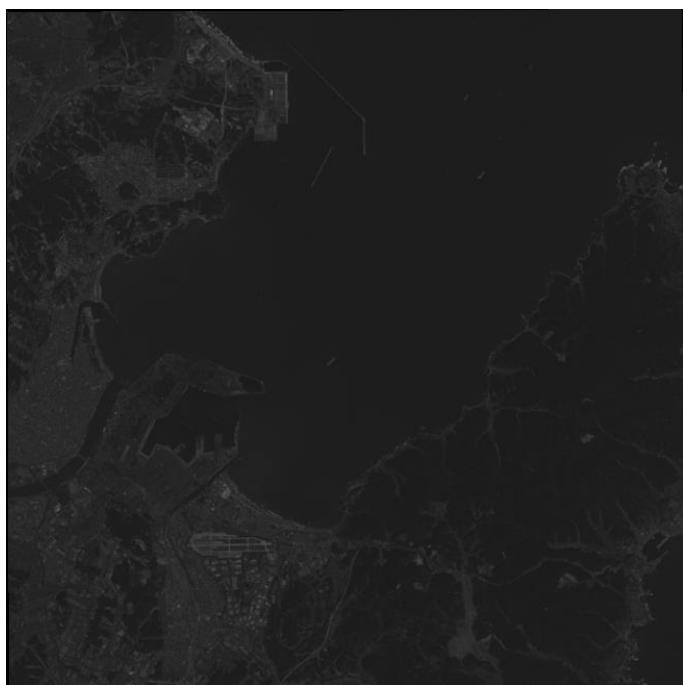
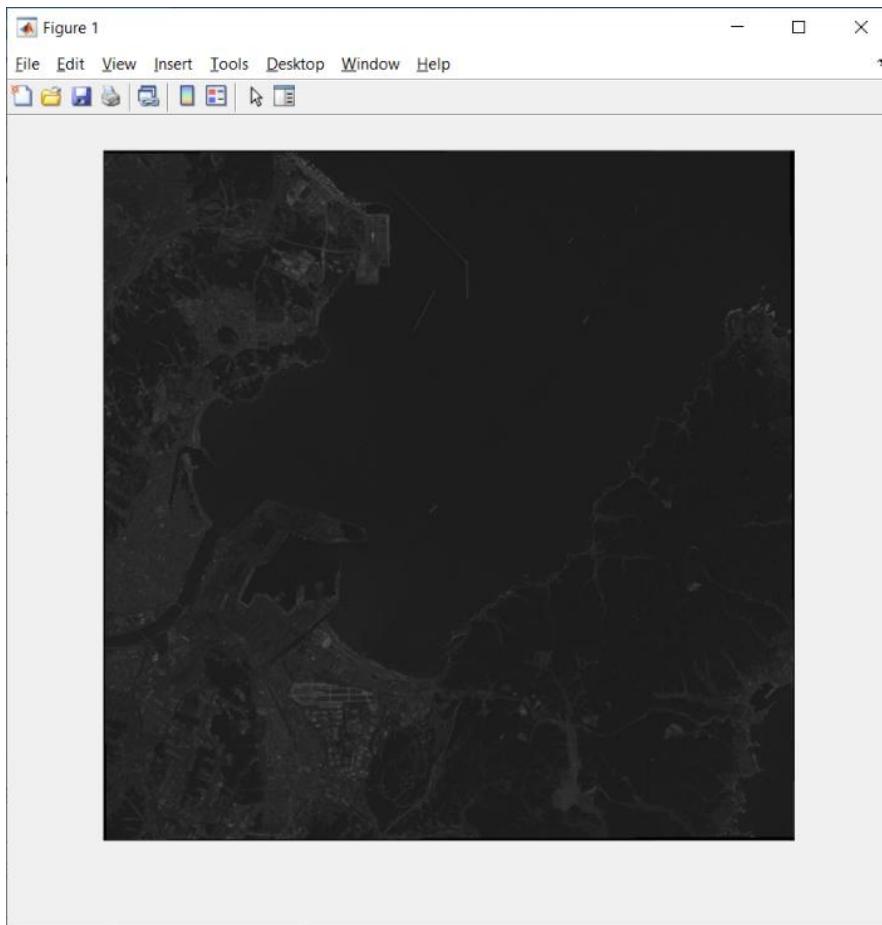


Imagen después de tratarla:





Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the App Store GET IT ON Google Play

DTE Departamento de
Tecnología Electrónica



Rubén Bueno Menéndez

Práctica 5

Código 5.3

```
function im2 = ndvi(imR, imNIR)
imR = im2double(imR);
imNIR = im2double(imNIR);
[F, C] = size(imR);
im2 = uint8(zeros(F, C));
for f = 1:F
    for c = 1:C
        ndR = imR(f,c);
        ndNIR = imNIR(f,c);
        if ndR > 0 && ndNIR > 0
            ndp = (ndNIR - ndR) / (ndNIR + ndR);
            ndp = round(((ndp + 1) / 2) * 254 + 1);
            im2(f, c) = ndp;
        end
    end
end
imshow(im2);
```

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```
imR = imread('imágenes/imagen_memoria_monobanda_R.jpg');
imNIR = imread('imágenes/imagen_memoria_monobanda_NIR.jpg');
ndvi(imR, imNIR);
```

Imagen propia tratada (antes/después)

Imagen original:

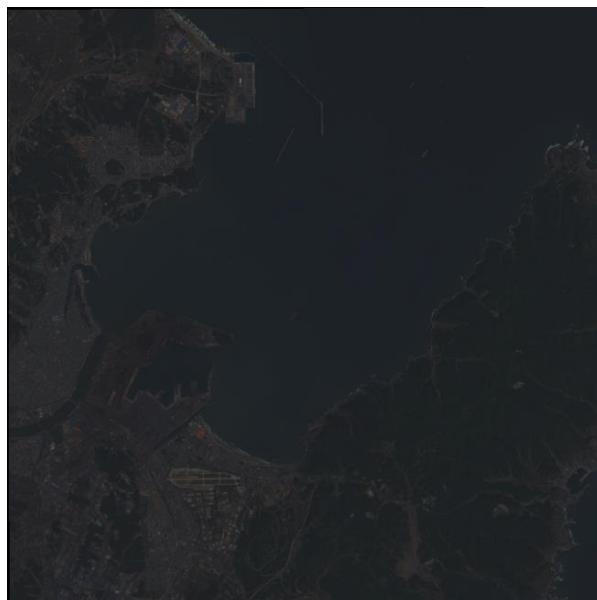


Imagen tras extraer la banda roja:

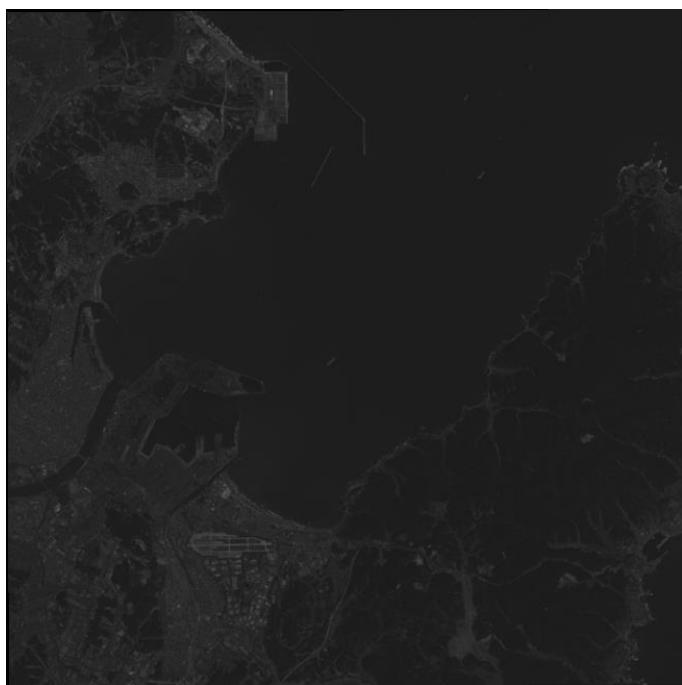


Imagen tras extraer la banda NIR:

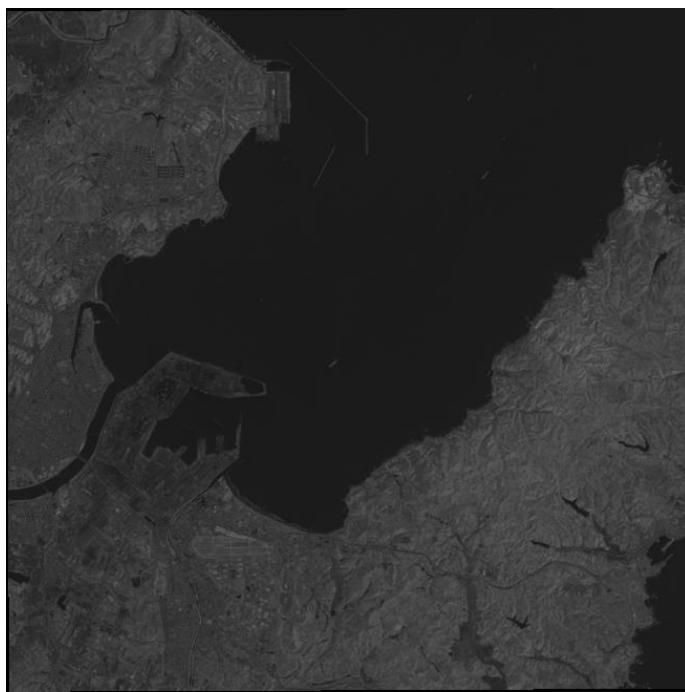
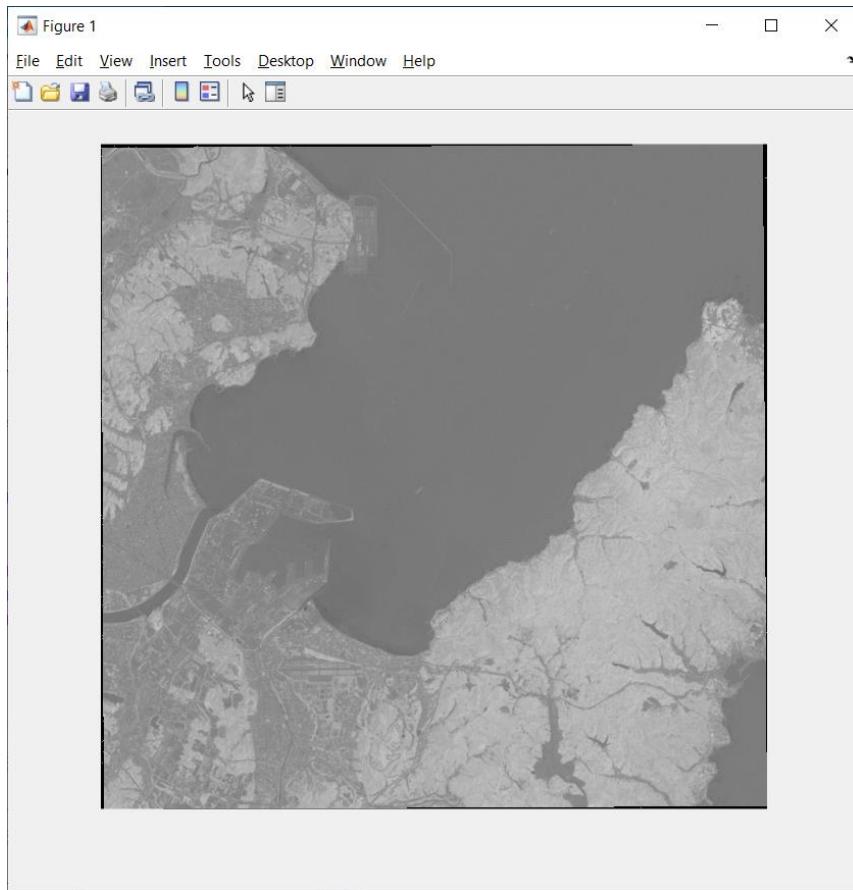


Imagen después de tratarla:



Código 5.4

```

function im2 = densi(im1, b)
im1 = im1(:,:,b); %Se indica la banda a extraer para que sea monobanda
maximoValor = find(histo(im1,1) > 1000, 1, 'last'); %He elegido 1000 como mínimo de
valor destacable
[F, C] = size(im1);
COLOR0 = [125,125,125]; %Gris
COLOR1 = [0,0,255]; %Azul
COLOR2 = [7,255,209]; %Celeste
COLOR3 = [0,255,0]; %Verde
COLOR4 = [255,255,0]; %Amarillo
COLOR5 = [244,70,17]; %Naranja
COLOR6 = [255,0,0]; %Rojo
COLOR7 = [255,255,255]; %Blanco
numColores = 7;

im2 = uint8(zeros(F, C, 3));
for f = 1:F
    for c = 1:C
        nd = im1(f,c);
        if nd > 0
            if nd < 128
                ndp = COLOR0;
            elseif nd < ((maximoValor-128)/numColores + 128)
                ndp = COLOR1;
            elseif nd < ((maximoValor-128)/numColores*(numColores - 5) + 128)
                ndp = COLOR2;
            elseif nd < ((maximoValor-128)/numColores*(numColores - 4) + 128)
                ndp = COLOR3;
            elseif nd < ((maximoValor-128)/numColores*(numColores - 3) + 128)
                ndp = COLOR4;
            elseif nd < ((maximoValor-128)/numColores*(numColores - 2) + 128)
                ndp = COLOR5;
            elseif nd < ((maximoValor-128)/numColores*(numColores - 1) + 128)
                ndp = COLOR6;
            else
                ndp = COLOR7;
            end
            im2(f, c, :) = ndp;
        end
    end
end
imshow(im2);

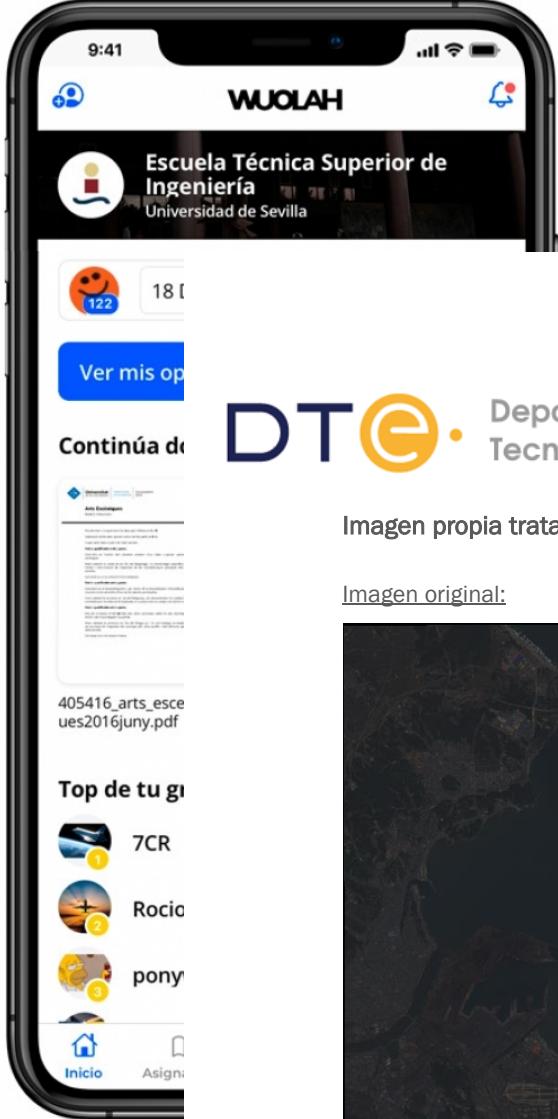
```

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```

imR = imread('imágenes/imagen_memoria_monobanda_R.jpg');
imNIR = imread('imágenes/imagen_memoria_monobanda_NIR.jpg');
densi(ndvi(imR, imNIR), 1);

```



Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the App Store GET IT ON Google Play



Rubén Bueno Menéndez

Imagen propia tratada (antes/después)

Imagen original:

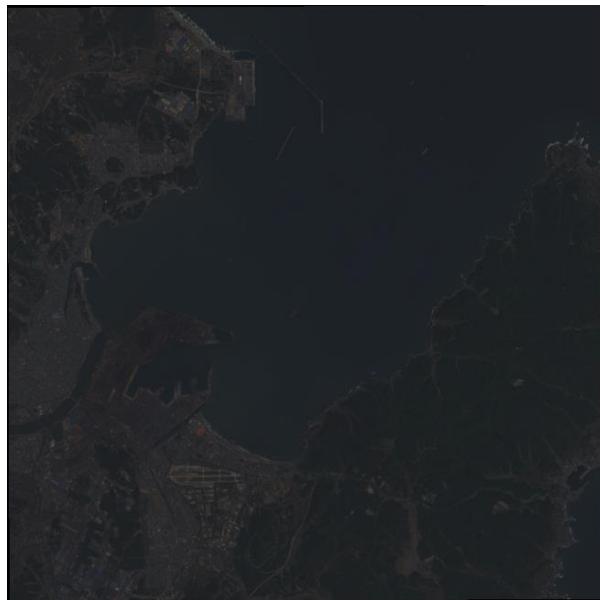


Imagen tras extraer la banda roja:



Imagen tras extraer la banda NIR:

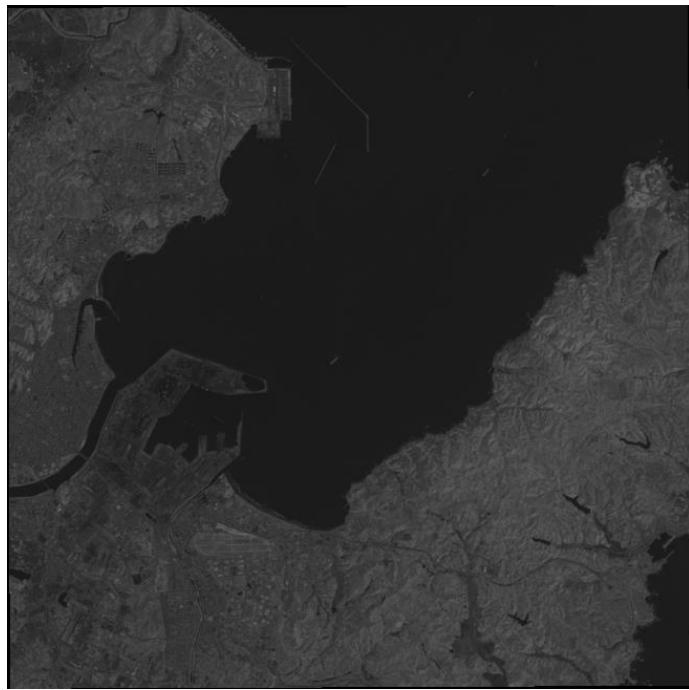
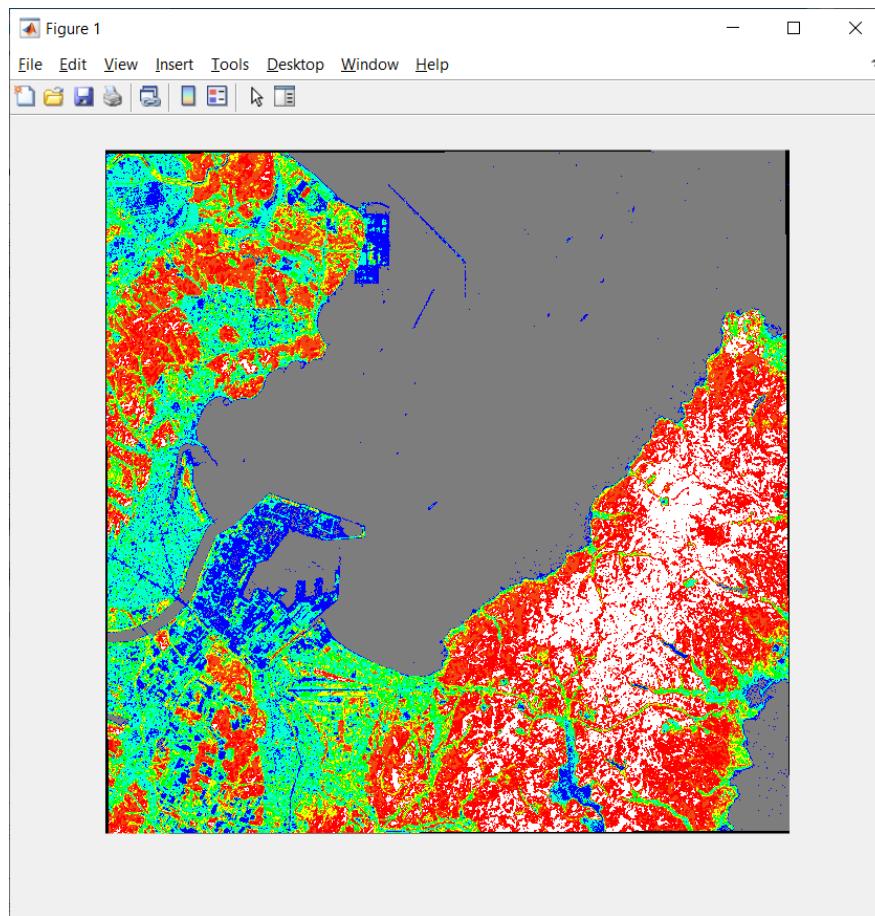


Imagen después de tratarla:



Código 5.5

```
function im2 = corteHSV(im1)
for v = 1:3
    figure(v)
    im2 = rgb2HSV(im1);
    im2 = uint8(im2 .* 255);
    if v == 1
        imTemp = im2(:, :, 2); %Para S
        im2(:, :, 2) = corte(imTemp, 1, 1); %Corte de cola al 1%
    elseif v == 2
        imTemp = im2(:, :, 3); %Para S
        im2(:, :, 3) = corte(imTemp, 1, 1); %Corte de cola al 1%
    else
        imTemp1 = im2(:, :, 2); %Para S
        imTemp2 = im2(:, :, 3); %Para V
        im2(:, :, 2) = corte(imTemp1, 1, 1); %Corte de cola al 1%
        im2(:, :, 3) = corte(imTemp2, 1, 1); %Corte de cola al 1%
    end
    im2 = double(im2) ./ 255;
    im2 = HSV2RGB(im2);
    im2 = uint8(im2 .* 255);
    imshow(im2);
end
```

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```
im = imread('imágenes/imagen_memoria.TIF');
im = im(:, :, 1:3);
corteHSV(im);
```

Imagen propia tratada (antes/después)

Imagen original:

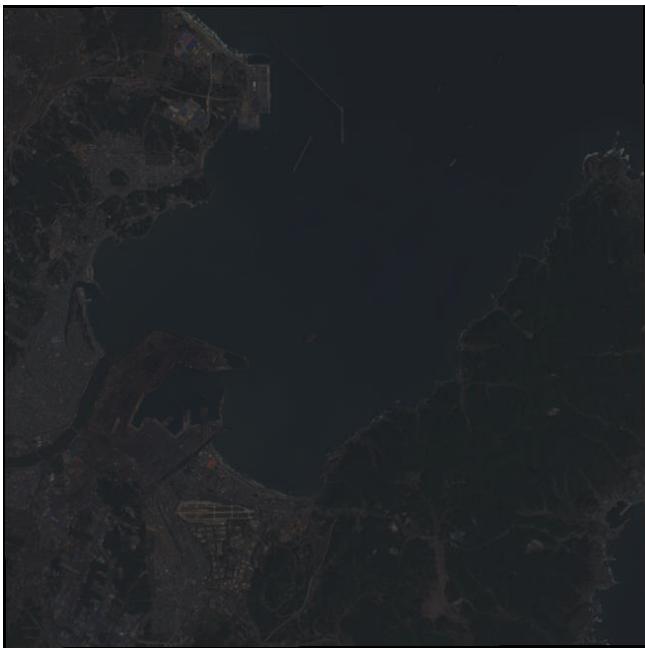


Imagen después de tratarla para el corte de la componente S:

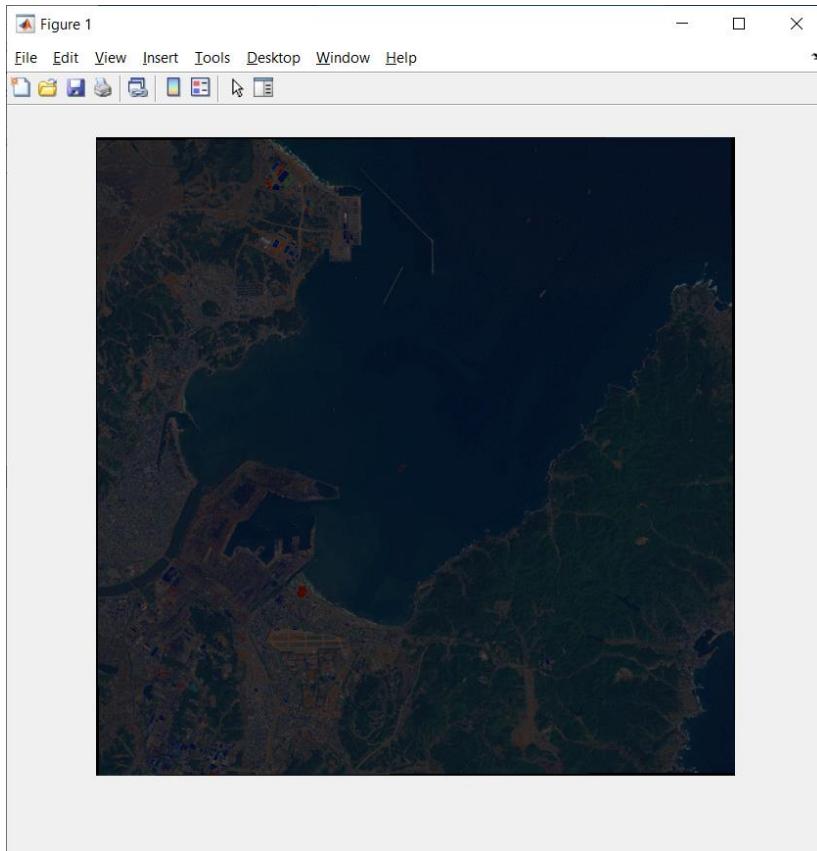
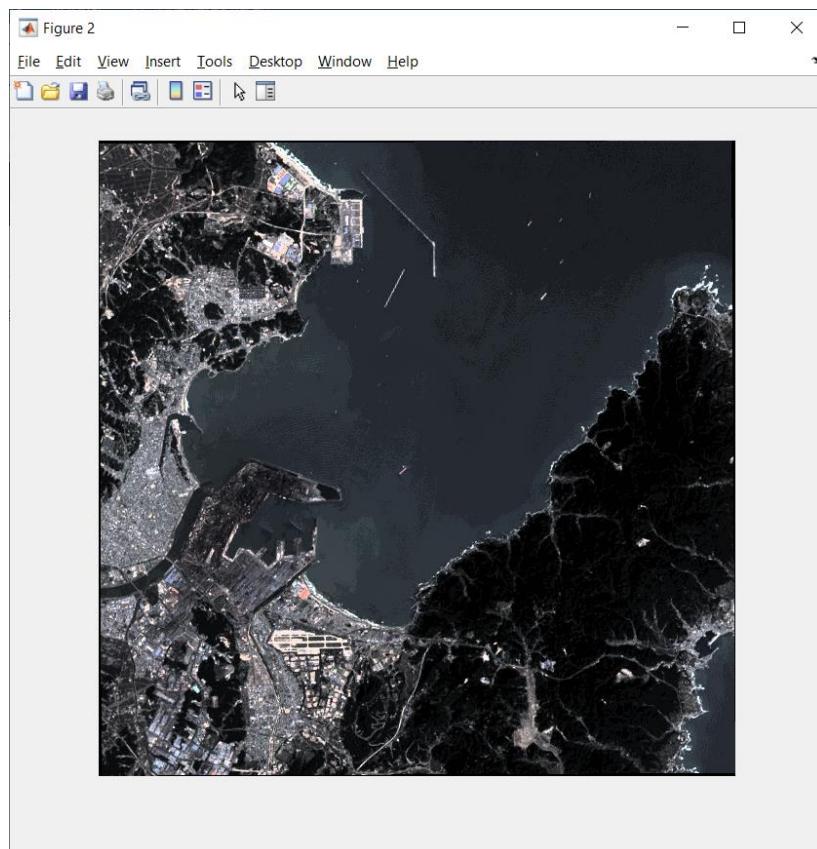
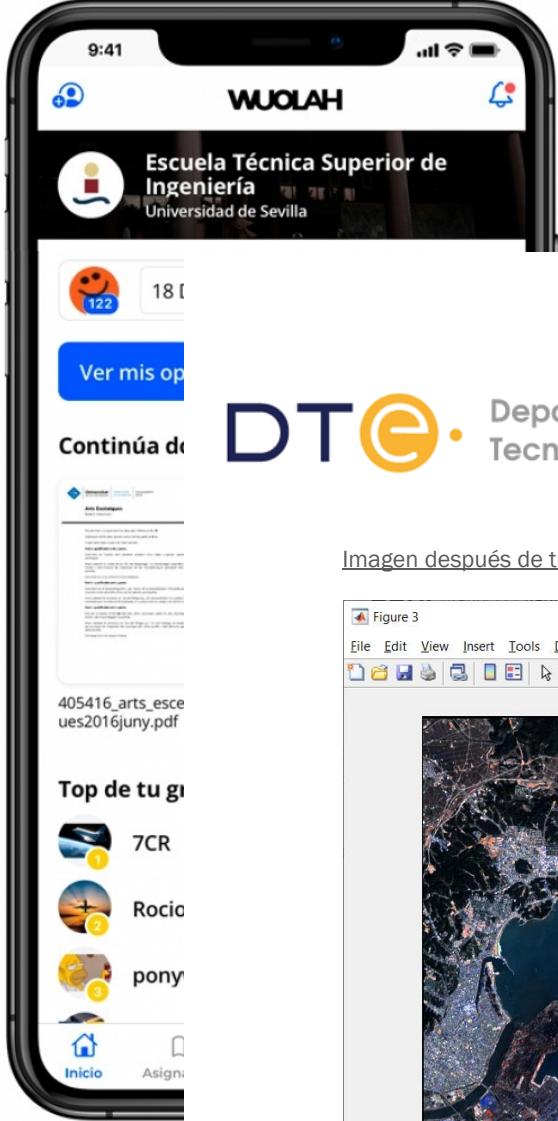


Imagen después de tratarla para el corte de la componente V:





Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the
App Store

GET IT ON
Google Play

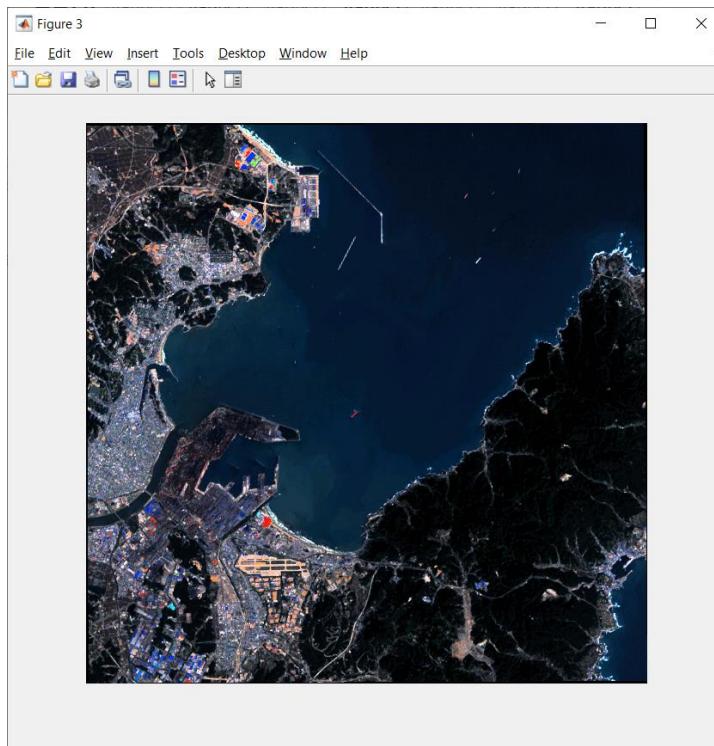
Continúa d...

DTE Departamento de
Tecnología Electrónica



Rubén Bueno Menéndez

Imagen después de tratarla para el corte de la componente S y V:



Práctica 6

Código 6.1

```

function im2 = isodata(im1, clases)
[F, C, B] = size(im1);
im2 = uint8(zeros(F, C, 3));
%Generar un color aleatorio para cada clase
cellColores = cell(1, clases);
for c = 1:clases
    cellColores{c} = [randi(255), randi(255), randi(255)];
end
%Elegir al azar los centros de las clases en el hiperespacio de bandas
b = 0;
while b == 0
    cellVectores = cell(1, clases);
    for cl = 1:clases
        y = round(rand * F);
        x = round(rand * C);
        listaBandas = zeros(1, B);
        for b = 1:B
            listaBandas(b) = im1(y, x, b);
        end
        cellVectores{cl} = double(listaBandas);
    end
    for v = 1:length(cellVectores)
        if prod(cellVectores{v}) == 0
            b = 0;
            break;
        else
            b = 1;
        end
    end
end
cellGruposTemp = cell(1, clases);
%Mientras haya algún punto que haya cambiado de grupo en una iteración
b = 0;
while b == 0
    cellGrupos = cell(1, clases);
    for f = 1:F
        for c = 1:C
            listaBandas = zeros(1, B);
            for b = 1:B
                listaBandas(b) = im1(f, c, b);
            end
            nd = double(listaBandas);
            if prod(nd) > 0
                %Calcular la distancia entre todos los pixeles y los centros elegidos
                listaDistancias = zeros(1, clases);
                for v = 1:length(cellVectores)
                    listaDistancias(v) = norm(nd - cellVectores{v});
                end
                %Asignar cada pixel al centro más próximo formando grupos
                distMenor = min(listaDistancias);
                clase = find(listaDistancias == distMenor, 1, 'last');
                im2(f, c, :) = cellColores{clase};
                cellGrupos{clase} = [cellGrupos{clase}; nd];
            end
        end
    end
    %Recalcular el centro de cada grupo como el punto medio del mismo
    for v = 1:length(cellVectores)
        cellVectores{v} = mean(cellGrupos{v});
    end
    if cellfun(@isequal, cellGrupos, cellGruposTemp)
        b = 1;
    end
    cellGruposTemp = cellGrupos;
end
imshow(im2);

```

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```
im = imread('imágenes/imagen_memoria.TIF');
im = im(:,:,1:3);
isodata(im,3);
isodata(im,5);
```

Imagen propia tratada (antes/después)

Imagen original:

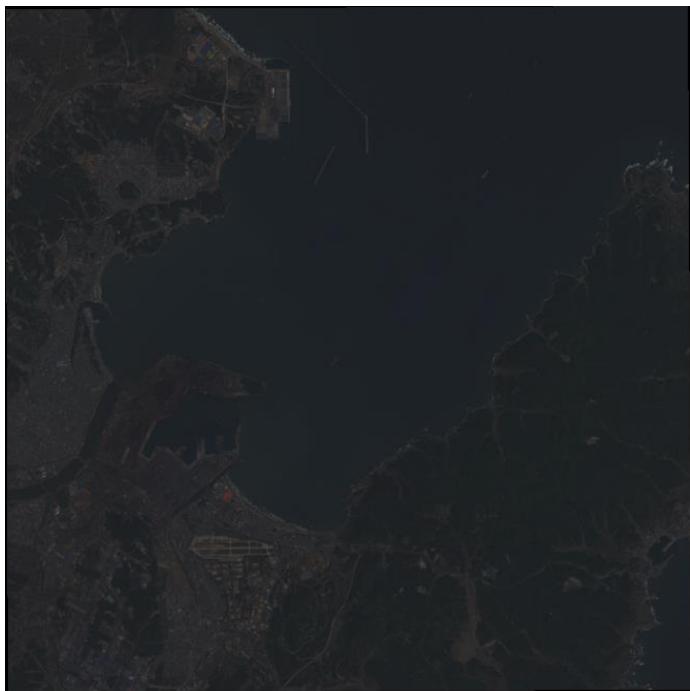


Imagen después de tratarla para una clasificación de 3 categorías:

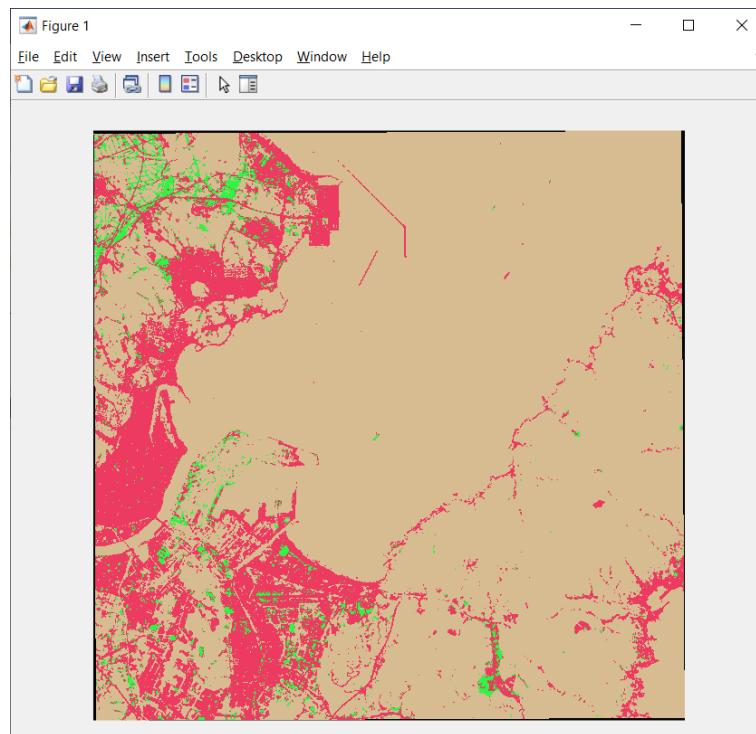
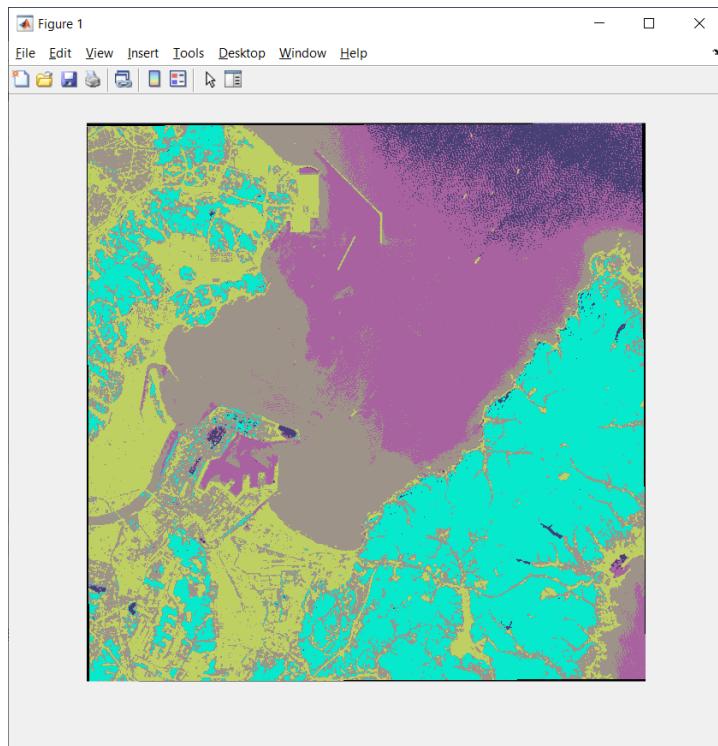


Imagen después de tratarla para una clasificación de 5 categorías:





Descarga la APP de Wuolah.
Ya disponible para el móvil y la tablet.

Available on the App Store GET IT ON Google Play



Rubén Bueno Menéndez

DTE Departamento de
Tecnología Electrónica

Práctica 7

Código 7.1

```
function im2 = gcpnoaa(im1, b)
im1 = im1(:, :, b); %Se indica la banda a extraer para que sea monobanda
xGCP = [564182.660; 619428.166; -8699.905; -1211.293; -47835.184; ...
         -31344.329; 201608.988; 289282.102; 576581.823; 702631.129; ...
         777819.735; 971756.716; 1018974.579]; %Puntos elegidos de la coordenada x
yGCP = [5189100.568; 5105847.492; 4772143.250; 4459524.870; 4276722.581; ...
         4112783.164; 4048541.418; 4001266.879; 4065272.753; 4169356.976; ...
         4296246.791; 4383322.393; 4703410.408]; %Puntos elegidos de la coordenada y
cGCP = [1259; 1294; 511; 441; 352; 327; 560; 645; 1015; 1166; 1266; 1435;
1512]; %Columnas aproximadas a las coordenadas x
fGCP = [50; 136; 333; 608; 772; 912; 1011; 1067; 1053; 984; 880; 836; 558]; %Filas
aproximadas a las coordenadas y
o = ones(13,1);
a = regress(fGCP, [o, xGCP, yGCP]);
b = regress(cGCP, [o, xGCP, yGCP]);

[F, C] = size(im1);
im2 = uint8(zeros(F+500, C+500)); %Incrementamos el tamaño de la imagen
%destino para que quepa la imagen original corregida
x = -750000; %Coordenada x de la esquina superior izquierda
y = 5400000; %Coordenada y de la esquina superior izquierda
xTemp = x;
for f = 1:F+500
    for c = 1:C+500
        fn = a(1) + a(2) * x + a(3) * y;
        cn = b(1) + b(2) * x + b(3) * y;
        x = x + 1000;
        if fn >= 1 && fn <= F && cn >= 1 && cn <= C
            im2(f,c) = im1(round(fn), round(cn));
        end
    end
    x = xTemp;
    y = y - 1000;
end
imshow(im2);
```

Tras ejecutar los siguientes comandos conseguimos la imagen resultante:

```
im = imread('imágenes/NOAA-ORBIT_4-5_LWIR.png');
gcpnoaa(im, 1);
```

Imagen propia tratada (antes/después)

Imagen original:

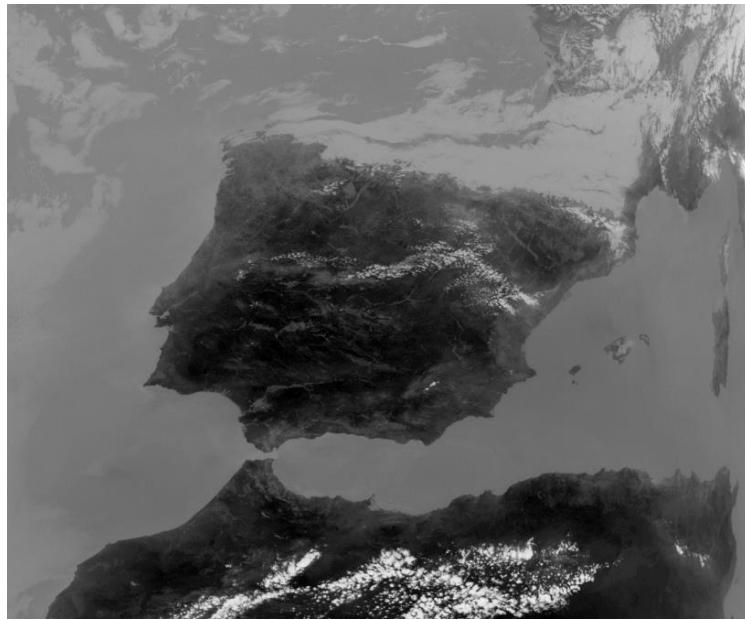


Imagen de los puntos elegidos de Geoplaner:

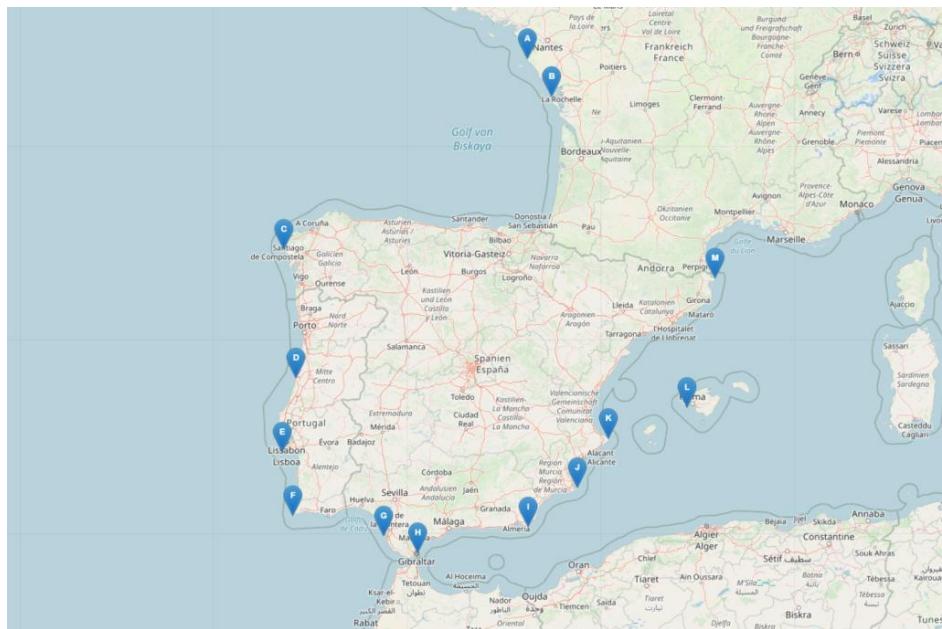


Imagen después de tratarla:

