

# Operaciones morfológicas y bordes

Gustavo Bermúdez, Juan Sebastian Olarte

## Metodología –

Cargar una imagen médica.

Corríjalo, mejore el nivel de gris y el contraste y elimine el ruido si corresponde.

### I. MARCO TEÓRICO

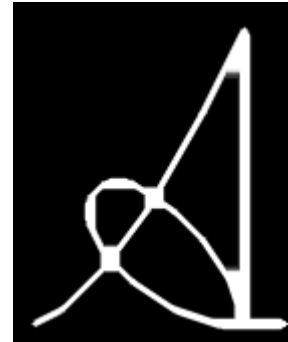
Las transformaciones morfológicas son algunas operaciones simples basadas en la forma de la imagen, que normalmente se aplican a imágenes binarias. Necesita dos entradas, una es nuestra imagen original, la segunda se llama elemento estructurante o núcleo (kernel) que decide la naturaleza de la operación. Dos operadores morfológicos básicos son Erosión y Dilatación. Luego, sus formas variantes como Apertura, Cierre, Gradiente, etc, también entran en juego. A continuación se verán algunas de estas transformaciones, apoyándonos en la siguiente imagen binaria:



#### Erosión:

Similar a la convulsión 2D, en el proceso de erosionado un kernel se desliza a través de la imagen. Un píxel de la imagen original (1 ó 0) sólo se considerará 1 si todos los píxeles que caen dentro de la ventana del kernel son 1, de lo contrario se erosiona (se hace a cero). Por tanto, todos los píxeles cerca de los bordes de los objetos en la imagen serán descartados dependiendo del tamaño del kernel. Como consecuencia, el grosor o el tamaño de los objetos en primer plano disminuye o, en otras palabras, la región blanca disminuye en la imagen. Este procedimiento es útil para eliminar pequeños ruidos blancos, separar dos objetos conectados, etc. A continuación un ejemplo donde se utiliza un kernel de 7×7 formado por unos:

```
1 | import cv2
2 | import numpy as np
3 |
4 |
5 | img = cv2.imread('A.png',0)
6 | kernel = np.ones((7,7),np.uint8)
   | erosion = cv2.erode(img,kernel,iterations = 1)
```



#### Dilatación:

El proceso de dilatación es justo lo opuesto a la erosión. Aquí, un elemento de píxel es '1' si al menos un píxel de la imagen de los que caen dentro de la ventana del kernel es '1'. Por lo tanto, la dilatación aumenta el tamaño de los objetos de primer plano, es decir, la región blanca. Normalmente, en casos como la eliminación del ruido, la erosión es seguida de dilatación. La razón para esto es que aunque la erosión elimina los ruidos blancos también encoge los objetos. Por tanto, para recuperar el tamaño inicial, este se dilata. La transformación de dilatación también es útil para unir partes rotas de un objeto. A continuación un ejemplo de cómo la dilatación funciona:

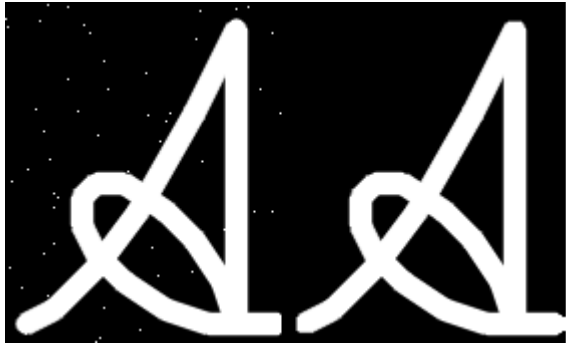
```
1 | dilatacion = cv2.dilate(img,kernel,iterations = 1)
```



#### Apertura:

La apertura es simplemente otro nombre para erosión seguida de dilatación. Como se explicó anteriormente, es útil para eliminar el ruido. En este caso se utiliza la función, **cv2.morphologyEx()**. Vea el ejemplo a continuación:

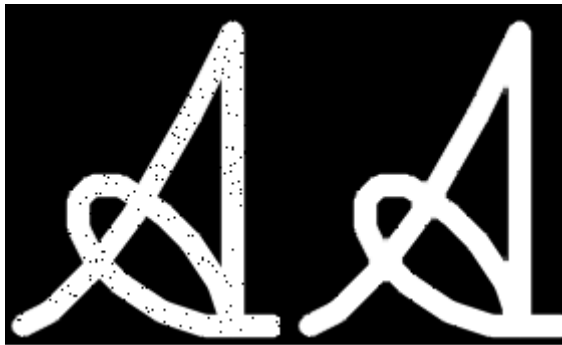
```
1 | apertura = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
```



### Cierre:

El cierre es el opuesto de Apertura, es decir, dilatación seguida de erosión. Es útil para cerrar pequeños agujeros dentro de los objetos de primer plano, o pequeños puntos negros en el objeto. Vea un ejemplo a continuación:

```
1 | cierre = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
```



### Gradiente morfológico:

Es la diferencia entre la dilatación y la erosión de una imagen. El resultado se verá como el contorno del objeto.

```
1 | gradiente = cv2.morphologyEx(img, cv2.MORPH_GRADIENT, kernel)
```



### Segmentación de Imágenes Utilizando Watershed:

La segmentación subdivide una imagen en sus partes constituyentes u objetos, con el fin de separar las partes de interés del resto de la imagen, por lo tanto el nivel al que se lleva a cabo esta subdivisión depende del problema a resolver. En el proceso de detectar las partes en una imagen se identifican bordes de la imagen, o se segmenta está en regiones, líneas o

curvas, etc.

## II. Resultados

### PROCEDIMIENTO SUGERIDO.

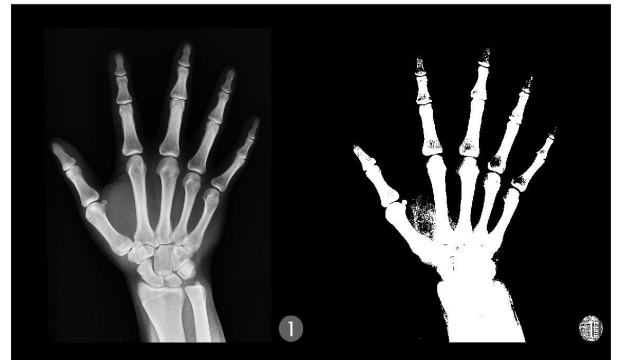
Carga una imagen y se transforma en una escala de grises si corresponde.

1. Usar un método Binarize OTSU y guardar los resultados.

### Código:

```
I = imread('medi.jpg');  
I = rgb2gray(I);  
level = graythresh(I);  
BW = imbinarize(I,level);  
imshowpair(I,BW,'primer punto')
```

### Resultado:

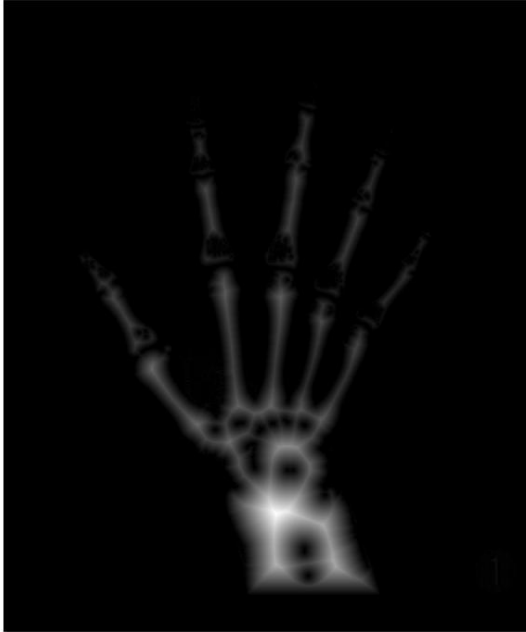


2. Usar un método watershed Binarize y guardar el resultado.

### Código:

```
%% parte 2  
B = bwdist(~BW);  
imshow(B,[],[])
```

### Resultado:



3. Calcule la envolvente de la lesión médica y muéstrela (segmentación y bordes).

Para este punto se uso la lesión de una persona con cancer pulmonar para que asi sea mas fácil observar sus diferentes lesiones y cambios respecto al codigo.

Código:

```
I= shadow_adapthisteq; %Lectura de la Imagen Mejorada
figure(1),imshow(I); %Visualización Imagen
figure(2),imhist(I); %Visualización Histograma

n=imhist(I); %Se guarda la imagen en una variable
N=sum(n); %Se suman las magnitudes del Histograma
max=0; %Inicialización de la variable Maxima

%Conteo de magnitudes del Histograma de 0 a 256
for i=1:256
    P(i)=n(i)/N;
end

%Algoritmo Binarización por OTSU
for T=2:255
    w0=sum(P(1:T)); %Se suman los vectores desde Fila 1 hasta T
    w1=sum(P(T+1:256)); %Suma desde T+1 hasta su extremo
    u0=dot([0:T-1],P(1:T))/w0; %Producto punto de los vectores
    u1=dot([T:255],P(T+1:256))/w1; %Producto punto de los vectores
    sigma=w0*w1*((u1-u0)^2); %Umbral
    if sigma>max
        max=sigma;
        threshold=T-1; %Límite del umbral
    end
end
```

Binarización Código

```
% Ahora estamos empezando a llegar a alguna parte. A continuación,
% calcule la cuenca transformada de [D].
ld = watershed(D);
figure
imshow(label2rgb(ld))
title("Imagen 4")

% Las líneas de la cresta de la cuenca, en blanco, corresponden a
% [ld == 0]. Usamos estas líneas de cresta para segmentar la imagen
% binaria cambiando el de píxeles correspondientes en el fondo.

bw2 = bw;
bw2(ld == 0) = 0;
figure
imshow(bw2)
title("Imagen 5")
```

Cuenca código

```
% La siguiente llamada a | imextendedmin | Idealmente debería producir
% pequeñas de manchas que se encuentran aproximadamente en el medio de
% las celdas que se van a segmentar. Enfermo use | imshoupair | para
% superponer la máscara a la imagen original.

mask = imextendedmin(D,2);
figure
imshowpair(bw,mask,'blend')
title("Imagen 6")

% Se modifique la transformación de distancia para que solo tenga mínimos
% en de las ubicaciones deseadas, y luego repita los pasos de la cuenca
% hidrográfica anteriores.

D2 = [imposemin(D,mask);
ld2 = watershed(D2);
bw3 = bw;
bw3(ld2 == 0) = 0;
figure
imshow(bw3)
title("Imagen 7")
```

Código manchas

Resultado:



Imagen de la radiografía pulmonar sin ningun cambio

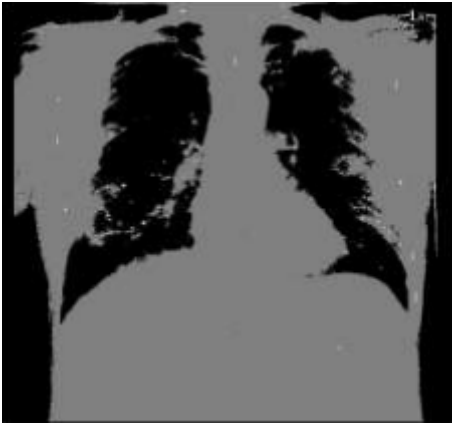


Imagen de radiografía aplicándose manchas e imposición de mínimos.

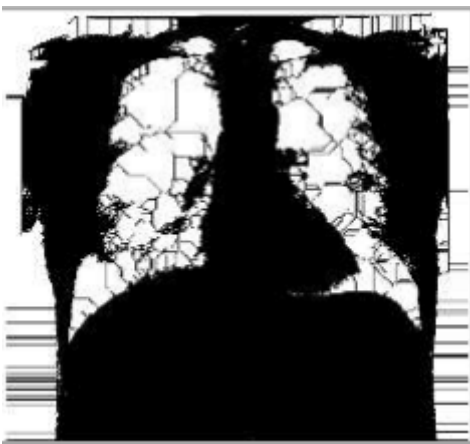


Imagen de radiografía aplicándose envolventes de zona mediante segmentación de imágenes,

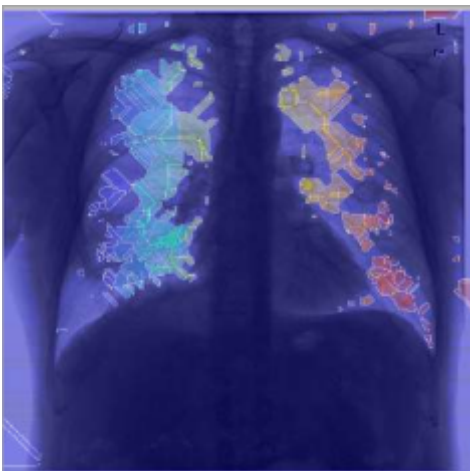


Imagen segmentada final aplicando todos los métodos de mejora como binarización, contraste , luminosidad..

## V. CONCLUSIONES

El uso de los métodos de Binarización de imágenes como

OTSU o Watershed se basan en algoritmos matriciales al igual que vectorización de mínimos, máximos y extendidos, es decir, transformación de píxeles. Es por ello que se concluye que para las imágenes médicas se pueden observar con mayor claridad cuáles son las zonas afectadas por ejemplo; el cáncer de pulmón, las zonas limpias sin ruido o imperfecciones de contraste, estos resultados los pudimos observar mediante la aplicación y el desarrollo de este laboratorio.