

VISIÓ PER COMPUTADOR

PRÁCTICA

Clasificación de imágenes

Alumnos: Anna Gracia Colmenarejo

Asignatura: Visión por computador

Grado: Ingeniería Informática

Profesora: Susana Alvarez Fernande

Data: 06/06/2022

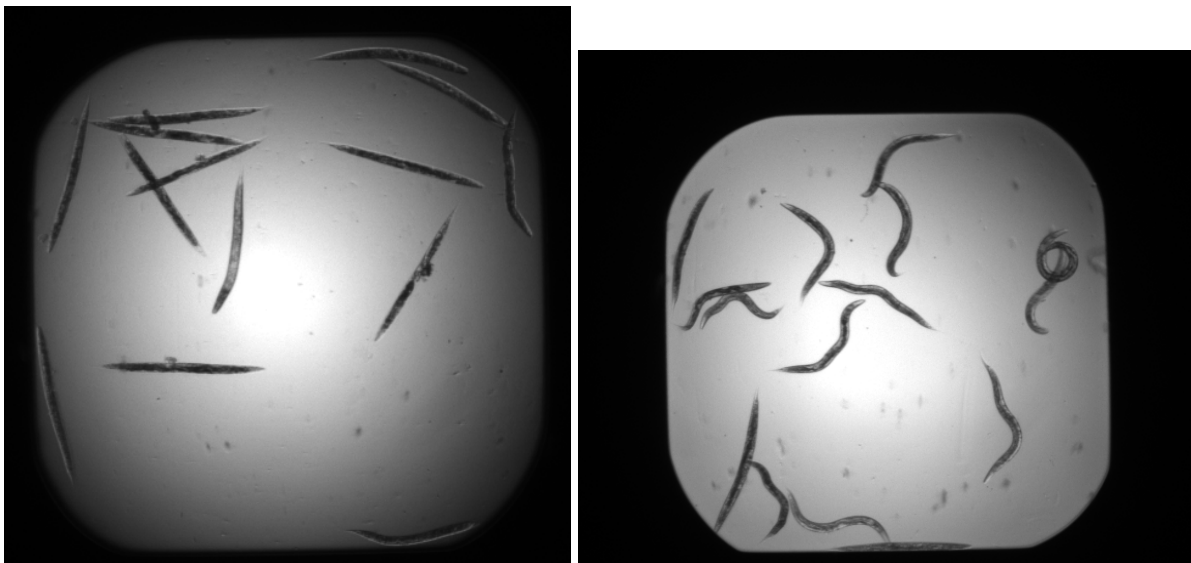
ÍNDEx

Especificaciones	3
Esquema de diseño	3
Pre-procesado	4
Segmentación	4
Obtención de las características necesarias y clasificación	5
Cálculo de precisión	7
Código	8
Evolución de las imágenes	10
ORIGINAL	10
MEJORA CONTRASTE	11
BINARIZACIÓN	12
EROSIÓN	13
ELIMINAR MANCHAS	14
Juego de pruebas	14
Bibliografía y fuentes de información	16

1. Especificaciones

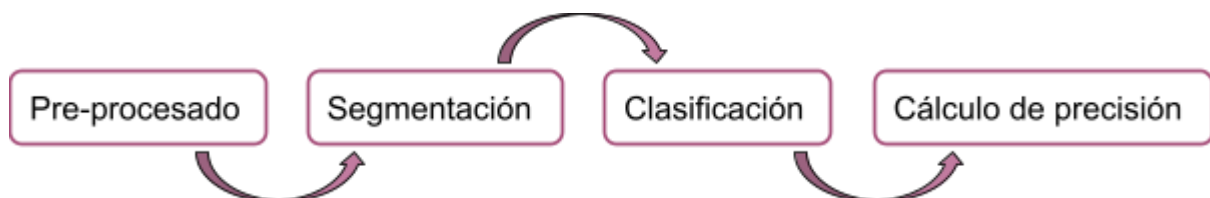
El siguiente programa realiza una clasificación de imágenes tomadas por un microscopio en las que se han capturado gusanos intestinales. La clasificación a realizar será por si estos gusanos están vivos o muertos. Estos gusanos tienen una forma curvilínea cuando están vivos mientras que cuando están muertos son rectilíneos. Las imágenes a clasificar pueden contener gusanos que estén todos vivos, todos muertos o que estén mezclados vivos o muertos. Para realizar esta clasificación se ha focalizado estudiar si son rectilíneos o curvilíneos.

A continuación se muestran unas imágenes de ejemplo.



2. Esquema de diseño

La resolución de este problema se ha estructurado en cuatro fases ordenadas: preprocesamiento, segmentación, clasificación y cálculo de precisión.



Pre-procesado

Inicialmente realizaremos un pre-procesamiento, es decir, un conjunto de técnicas para mejorar la calidad de la imagen y así obtener luego una mejor clasificación.

Una de las técnicas principales sería la eliminación de ruido. El ruido aparece en la fotografía digital cuando el color y la luz de algunos píxeles se ve alterado. Pero en este caso las imágenes realizadas por el microscopio no tenían esta característica por lo que no se han utilizado métodos para su eliminación.

La primera técnica utilizada es una mejora del contraste de la imagen. El contraste es la diferencia relativa en la intensidad entre un punto de la imagen, es decir, el efecto que se produce al destacar un elemento visual en comparación a otro en la misma imagen. En imágenes donde el contraste es muy bajo sería más complicado identificar los elementos de esta, por lo que a las imágenes que vamos a tratar queremos mejorar este contraste y poder identificar mejor los gusanos. La función de MATLAB que se ha utilizado es `adapthisteq(img)` que realiza una ecualización adaptativa del histograma de la imagen por contraste *CLAHE*. *CLAHE* es una variante de la ecualización adaptativa de histograma(AHE) que se encarga de la sobre-amplificación del contraste. CLAHE opera en pequeñas regiones de la imagen, llamadas mosaicos, en lugar de en la imagen completa. Luego, los mosaicos vecinos se combinan mediante interpolación bilineal para eliminar los límites artificiales.

```
function img=preprocesado(img)

    % MEJORA DEL CONTRASTE %
    img=adapthisteq(img);
end
```

Segmentación

Una vez hemos mejorado la calidad de la imagen el siguiente paso es segmentarla. La segmentación es la subdivisión de la imagen en sus partes constituyentes u objetos con el fin de separar las partes de interés del resto.

Primeramente, se binarizará la imagen, esta es una técnica que reduce la escala de grises de la imagen a dos únicos valores 1 y 0, blanco y negro. En la práctica se ha realizado mediante la función de MATLAB `imbinarize(img)`. Esta función crea

una imagen binaria a partir de una imagen 2D o 3D en escala de grises. Reemplaza todos los valores que se encuentran por encima de un umbral determinado por 1 y reemplazando todos los demás valores por 0.

Necesitamos invertir la imagen para obtener un fondo blanco para el siguiente paso. A continuación, se realiza una erosión de la imagen. La erosión es una operación morfológica que se concibe como una reducción de la imagen. En este caso la erosión nos ayuda a separar gusanos que pueden encontrarse muy juntos y que por ese motivo se detectan como uno solo. La función de matlab `imerode(img,se)` erosiona la imagen utilizando un elemento estructurado *SE*. En este caso el elemento estructurado es en forma de diamante al que se le especifica la distancia desde el origen del SE hasta los puntos del diamante. Y por último las imágenes de los gusanos contienen pequeñas manchas las cuales queremos eliminar y que no cuenten como un gusano más. Para esto MATLAB tiene una función que elimina todos los componentes conectados que tienen menos píxeles que el mínimo indicado. En este caso hemos comprobado que con que se eliminen elementos con menos de 70 píxeles no se está eliminando ningún gusano ni información necesaria.

```
function img=segmentacion(img)

    % BINARIZACIÓN DE LA IMAGEN %
    img=imbinarize(img);

    % INVERTIR IMAGEN %
    img=imcomplement(img);

    % EROSIÓN DE LA IMAGEN %
    se=strel('diamond',2);
    img=imerode(img,se);

    % ELIMINAR MANCHAS QUE NO SON GUSANOS %
    img=bwareaopen(img,70);
end
```

Obtención de las características necesarias y clasificación

Para poder clasificar en vivos o muertos se ha estudiado la característica de excentricidad mediante la función `regionprops` de MATLAB. La excentricidad es un parámetro que determina el grado de desviación de una sección cónica con respecto a una circunferencia.

La función mencionada anteriormente devuelve las medidas para el conjunto de propiedades para cada objeto en una imagen binaria. En la práctica hemos llamado a esta función de la siguiente manera:

```
regionprops('table',img,'Eccentricity','Area','Image');
```

donde el parámetro 'table' hace que la función nos devuelva una tabla con un número de filas igual al de objetos en la imagen. 'img' es la imagen binaria a analizar, 'Eccentricity' es la característica a estudiar en este caso la excentricidad, 'Area' el número de píxeles en la región y 'Image' es una imagen del mismo tamaño que el cuadro delimitador de la región, devuelta como una matriz binaria. Los píxeles activados corresponden a la región y todos los demás píxeles están desactivados.

```
function estado=clasificacion(img)
    cont_muertos=0;
    cont_vivos=0;

    stat = regionprops('table',img,'Eccentricity','Area','Image');
    for j = (1:size(stat,1))
        if(stat.Area(j) < 100 || stat.Area(j) > 1000 )
            stat.Area(j) = 0;
            stat.Eccentricity(j) = 0;
        end
        %          EXCENTRICIDAD          %

        if(stat.Eccentricity(j) < 0.99)
            cont_vivos=cont_vivos+1;
        else
            cont_muertos=cont_muertos+1;
        end
    end
    if cont_vivos > cont_muertos
        estado=1;
    else
        estado=0;
    end
end
```

La excentricidad devuelve valores cercanos a 1 cuando un objeto tiene poca excentricidad. Cuando la excentricidad es 0 estamos hablando de un círculo. Por lo que según los valores devueltos por la excentricidad clasificábamos el objeto como gusano vivo o muerto, cuando los valores sean 0.99 o 1 quiere decir que estamos tratando con un gusano muerto.

Cálculo de precisión

Para calcular si estamos acertando con la clasificación se han comparado los resultados con un excel en el que se indicaba por cada fotografía si los gusanos estaban vivos o muertos.

Se comparaban los resultados obtenidos con los del excel dado y se calculaba el porcentaje de aciertos.

3. Código

```
clear;

% PATH DE LA CARPETA ON ES TROBEN LES IMATGES DE CUCS %
path='/MATLAB/AnnaGracia_WormImages/WormImages/';

% ***** OBTENCIÓ D'UNA IMATGE A TRACTAR ***** %
imatges_cucs = dir(fullfile(path, '*.tif'));
tam=size(imatges_cucs);
noms_imatges = {imatges_cucs.name};

% VECTOR PARA GUARDAR EL ESTADO DE LOS GUSANOS
vec_clasficacion=zeros(tam);
vec_clasficacion=string(vec_clasficacion);

for i=1:tam
    img = imread(strcat(path,noms_imatges{i}));

    % PRE PROCESADO %
    img=preprocesado(img);

    % SEGMENTACIÓN %
    img=segmentacion(img);

    % CLASIFICACIÓN %
    estado=clasificacion(img);
    if estado == 1
        vec_clasficacion(i)="alive";
    else
        vec_clasficacion(i)="dead";
    end
end

% ESCRIBIR RESULTADOS EN CSV %
writematrix(vec_clasficacion,"clasificacion.csv");

% CÁLCULO DE PRECISION %
vect_original=readtable("WormData.csv");
vect_original = table2array(vect_original(:,2));

aciertos=0;
for i=1:tam
    if strcmp(vec_clasficacion(i),vect_original(i)) == 1
        aciertos = aciertos+1;
    end
end
precision=(aciertos./24)*100;
precision
```



```

function img=preprocesado(img)

    % MEJORA DEL CONTRASTE %
    img=adapthisteq(img);
end

function img=segmentacion(img)

    % BINARIZACIÓN DE LA IMAGEN %
    img=imbinarize(img);

    % INVERTIR IMAGEN %
    img=imcomplement(img);

    % EROSIÓN DE LA IMAGEN %
    se=strel('diamond',2);
    img=imerode(img,se);

    % ELIMINAR MANCHAS QUE NO SON GUSANOS %
    img=bwareaopen(img,70);
end

```

```

function estado=clasificacion(img)
    cont_muertos=0;
    cont_vivos=0;

    stat = regionprops('table',img,'Eccentricity','Area','Image');
    for j = (1:size(stat,1))
        if(stat.Area(j) < 100 || stat.Area(j) > 1000 )
            stat.Area(j) = 0;
            stat.Eccentricity(j) = 0;
        end
        %          EXCENTRICIDAD          %

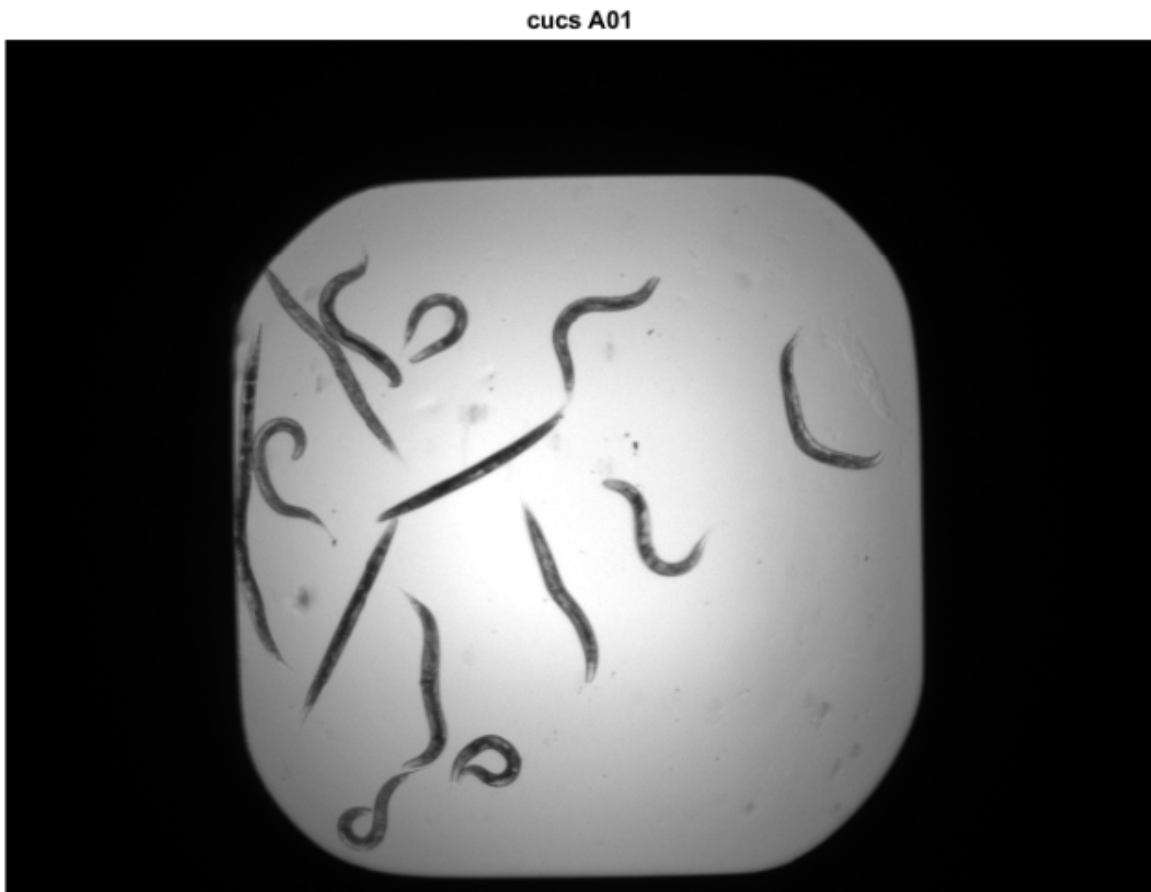
        if(stat.Eccentricity(j) < 0.99)
            cont_vivos=cont_vivos+1;
        else
            cont_muertos=cont_muertos+1;
        end
    end
    if cont_vivos > cont_muertos
        estado=1;
    else
        estado=0;
    end
end

```

4. Evolución de las imágenes

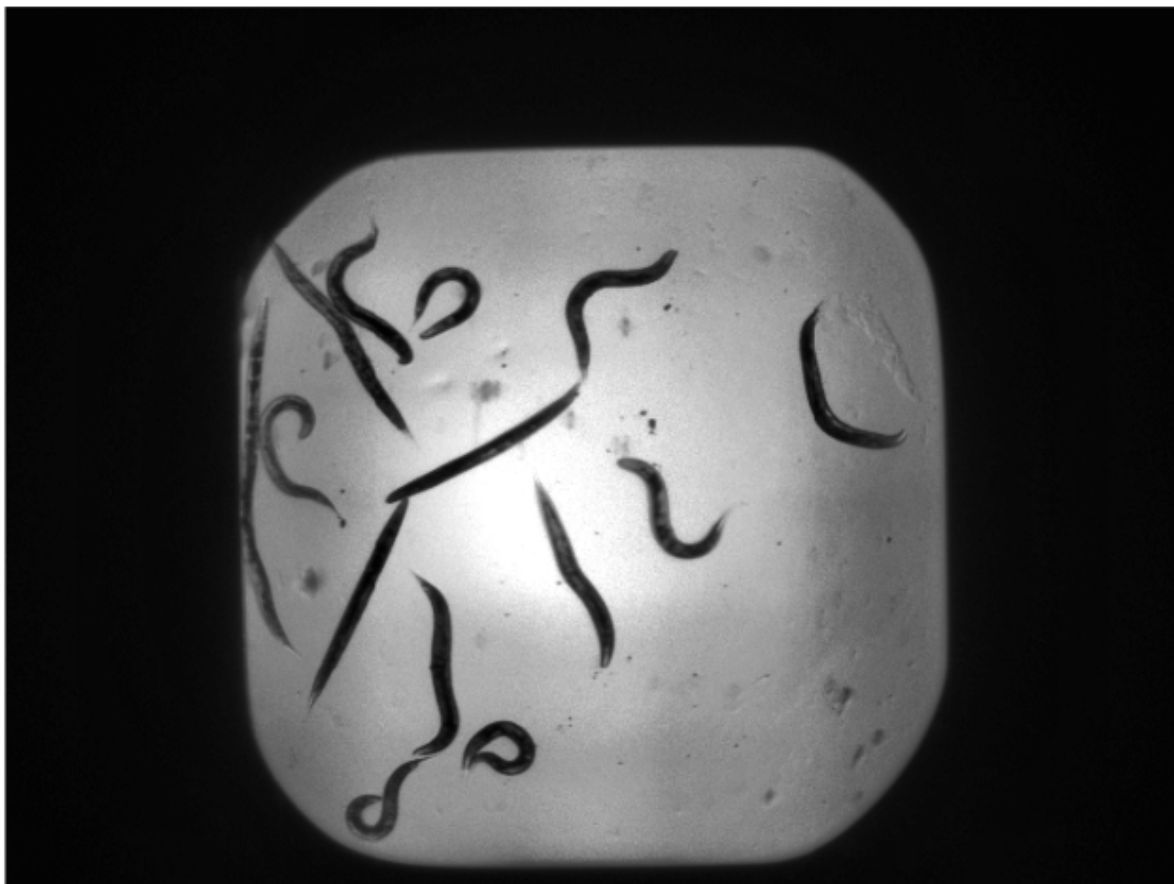
A continuación, se muestra la evolución de dos imágenes a lo largo de todo el proceso de clasificación.

ORIGINAL



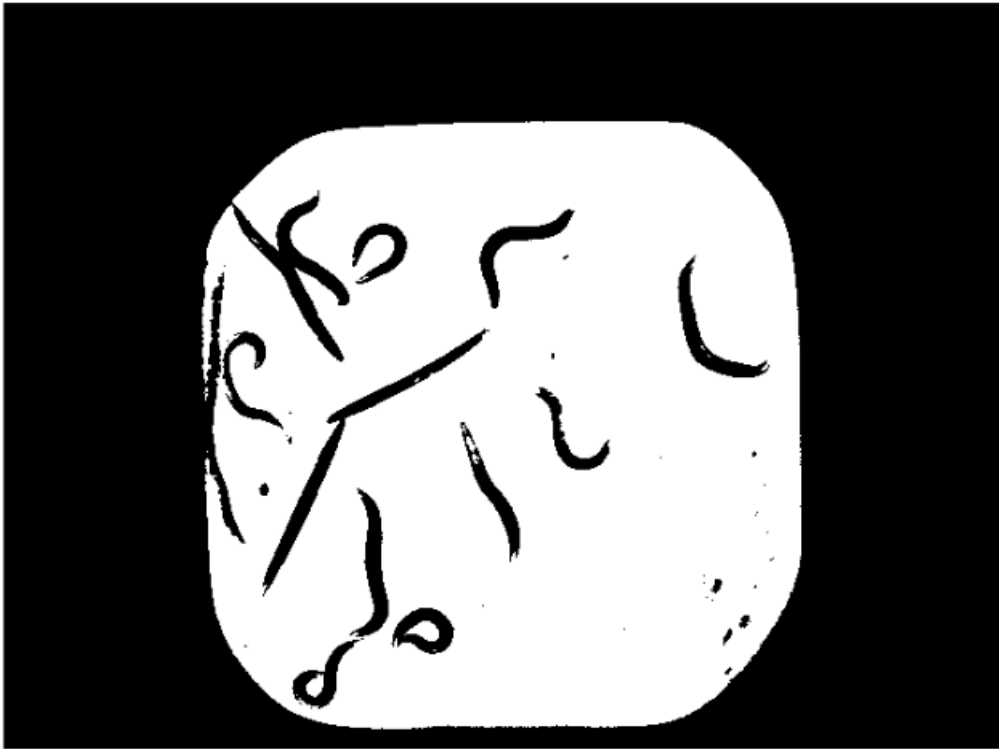
MEJORA CONTRASTE

cucs A01



BINARIZACIÓN

cucs A01



INVERTIR IMAGEN

cucs A01



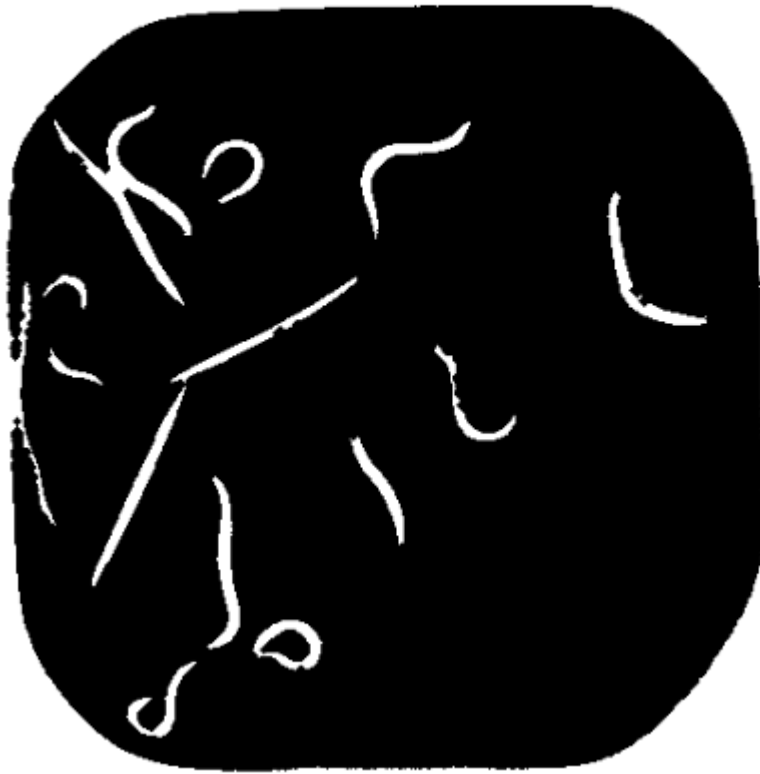
EROSIÓN

cucs A01



ELIMINAR MANCHAS

cucs A01



5. Juego de pruebas

Como juego de pruebas se ha realizado el análisis de las 24 fotografías dadas.

En la siguientes imágenes se pueden ver los resultados, donde se ha obtenido una precisión de un 96% aproximadamente.

aciertos	23
estado	0
i	24
imatges_cucs	24x1 struct
img	520x696 logical
noms_imatges	1x24 cell
path	'/MATLAB/AnnaGrac...
precision	95.8333
tam	[24,1]
vec_clasficacion	24x1 string
vect_original	24x1 cell

1	File	Status
2	wormA01.tif	alive
3	wormA02.tif	alive
4	wormA03.tif	alive
5	wormA04.tif	alive
6	wormA05.tif	alive
7	wormA06.tif	alive
8	wormA07.tif	alive
9	wormA08.tif	alive
10	wormA09.tif	alive
11	wormA10.tif	alive
12	wormA11.tif	alive
13	wormA12.tif	alive
14	wormA13.tif	dead
15	wormA14.tif	dead
16	wormA15.tif	dead
17	wormA16.tif	dead
18	wormA17.tif	dead
19	wormA18.tif	dead
20	wormA19.tif	dead
21	wormA20.tif	dead
22	wormA21.tif	dead
23	wormA22.tif	dead
24	wormA23.tif	dead
25	wormA24.tif	dead

6. Bibliografía y fuentes de información

Documentación MATLAB:

adaphistectq
imbinarize
imcomplement
imerode
bwareaopen
regioprops
writematrix
readtable
table2array

https://en.wikipedia.org/wiki/Image_noise

<https://es.wikipedia.org/wiki/Contraste>

<https://alphauniverse-latin.com/notas/que-es-el-contraste-y-como-puedo-aprovecharlo-en-mis-fotos#:~:text=El%20contraste%20no%20es%20otra,o%20cualquier%20otro%20elemento%20visual>

https://en.wikipedia.org/wiki/Adaptive_histogram_equalization#Contrast_Limited_AHE

<https://es.acervolima.com/ecualizacion-de-histograma-clahe-opencv/>

[https://es.wikipedia.org/wiki/Segmentaci%C3%B3n_\(procesamiento_de_im%C3%A1genes\)#:~:text=La%20segmentaci%C3%B3n%20es%20uno%20de,grupos%20de%20p%C3%ADxeles\)%20denominadas%20segmentos](https://es.wikipedia.org/wiki/Segmentaci%C3%B3n_(procesamiento_de_im%C3%A1genes)#:~:text=La%20segmentaci%C3%B3n%20es%20uno%20de,grupos%20de%20p%C3%ADxeles)%20denominadas%20segmentos)

[https://es.wikipedia.org/wiki/Erosi%C3%B3n_\(morfolog%C3%ADa\)](https://es.wikipedia.org/wiki/Erosi%C3%B3n_(morfolog%C3%ADa))