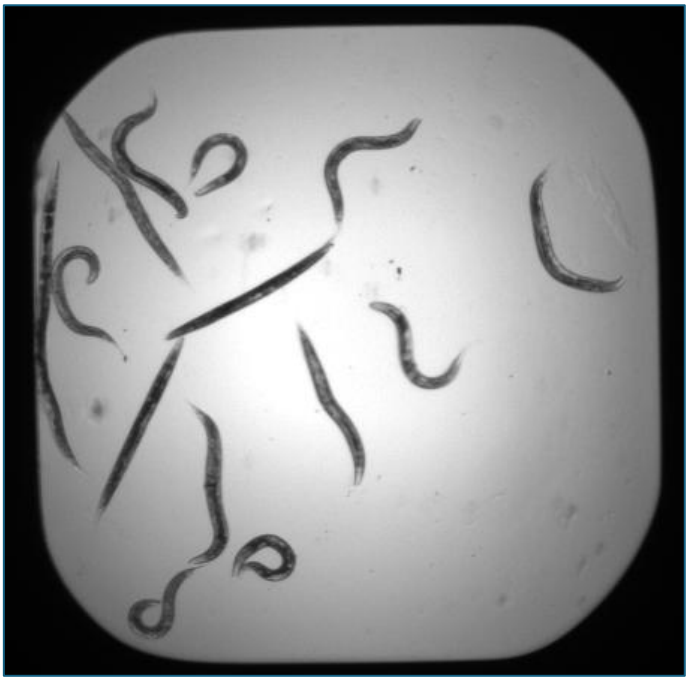
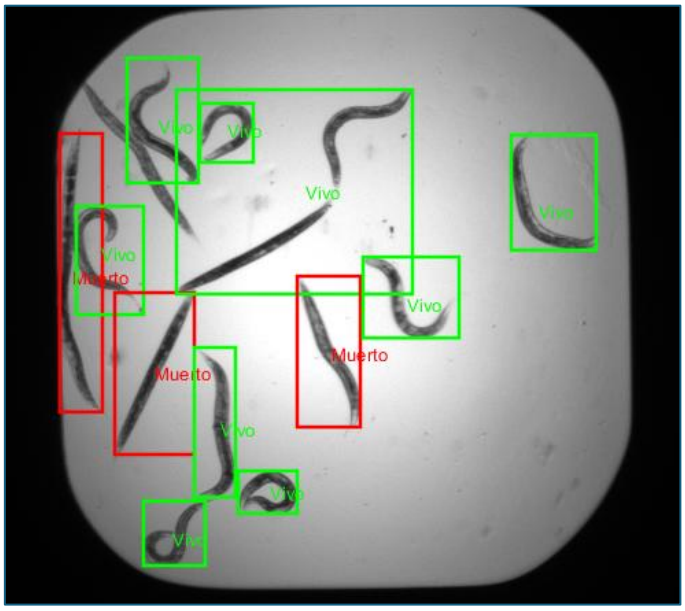
IDENTIFICACIÓN DE GUSANOS





Iván García Pallarés

Especificaciones

En este proyecto se desarrollará un programa capaz de clasificar imágenes obtenidas mediante un microscopio, con el objetivo de identificar y contar gusanos intestinales vivos y muertos. El reto principal consiste en analizar la morfología de los gusanos presentes en cada imagen, distinguiendo entre los vivos, los cuales presentan una forma curvilínea y los muertos, que tienen una forma rectilínea. A partir de esta información, el sistema deberá determinar a qué categoría pertenece la imagen en su conjunto, según la mayoría de gusanos detectados.

Además, el programa deberá evaluar su precisión comparando los resultados obtenidos con un archivo de referencia (.csv), que indica la clasificación correcta y el número real de gusanos vivos y muertos por imagen. También se requerirá generar una versión anotada de cada imagen, en la que se destaquen visualmente los gusanos identificados y su clasificación. Este proyecto plantea un caso práctico de aplicación de técnicas de procesamiento de imágenes y clasificación automática, con potencial utilidad en el ámbito biomédico y diagnóstico asistido por computador.

Diseño de la solución

Para resolver el problema de clasificación e identificación de gusanos vivos y muertos en imágenes microscópicas, hemos desarrollado un algoritmo en MATLAB basado en procesamiento de imagen y análisis morfológico. A continuación, se detalla el diseño y razonamiento de la solución adoptada.

Esquema:

1. Leer carpeta con imágenes y archivo CSV con los datos reales.
2. Para cada imagen:
   a. Crear una máscara que aísle el área útil (interior del cristal).
   b. Binarizar adaptativamente para detectar gusanos (objetos oscuros).
   c. Eliminar objetos tocando los bordes y zonas externas.
   d. Aplicar morfología (aperturas, cierres, relleno) para mejorar las detecciones.
   e. Detectar componentes conectados y calcular su excentricidad.
   f. Clasificar cada objeto como "vivo" (forma curva) o "muerto" (forma recta).
   g. Dibujar la detección y guardar la imagen anotada.
   h. Escribir los resultados en un archivo CSV de salida.
3. Evaluar la precisión comparando con los datos reales del archivo CSV.

Decisiones y técnicas aplicadas

- **Lectura de archivos**: Se ha decido primeramente declarar las variables principales de archivos para facilitar la faena siendo más versátil.

```
% Parámetros ajustables
folder_path    = 'WormImages';
output_folder = 'Resultado';
image_output_folder = fullfile(output_folder, 'Imagenes');
ground_truth_file = fullfile('WormDataA.csv');
```

Seguido de esto eliminamos la estructura de resultados previos eliminanado la carpeta de almacenamiento de estos ("Resultados/"). I volvemos a crear toda la estructura de nuevo.

```
if exist(output_folder,'dir')
    rmdir(output_folder, 's');
end
    mkdir(output_folder);
    mkdir(image_output_folder);
    fid = fopen(csv_file, 'w');
    fprintf(fid, 'Nombre_fichero;Status;Muertos;Vivos\n');
    fclose(fid);
```

Como el archivo 'WormDataA.csv' separa la información con dos separadores (',' y ';') lo cambiamos todo a ';' para más tarde poder comparar la columna de Status más fácilmente.

```
% Estructurar datos reales
if isfile(ground_truth_file)
    % Leer archivo como texto y reemplazar comas por punto y coma
    raw1 = fileread(ground_truth_file);
    raw1 = strrep(raw1, ',', ';');

    % Escribir a archivo temporal limpio
    fid = fopen(ground_truth_file, 'w');
    fwrite(fid, raw1);
    fclose(fid);
end
```

- **Máscara del área útil (interior del cristal)**: Se ha creado mediante binarización inicial, inversión y limpieza morfológica. Esta máscara restringe el análisis solo a las zonas relevantes.
  Donde primero binarizamos para quedarnos con las zona clara de cristal, invirtiendo la mascara para quedarnos con el interior blanco eliminando posibles objetos externos de 120 px.

```
% Binarizar para obtener máscara gruesa de cristal
MASK   = imbinarize(I, 0.1);        % fondo claro = 1, cristal = 0
MASK   = imcomplement(MASK);           % interior = 1
MASK   = bwareaopen(MASK, 120);        % Eliminar regiones
```

  Una vez tenemos una mascara básica toca mejorar la precisión ya que hay una serie de imágenes oscuras que con este método no es suficiente ya que recortamos zonas oscuras como si fueran del contorno. Por lo que primero nos interesa aislar el fondo.

```
% Aislar interior: fondo exterior a blanco
interior = I;
interior(~MASK) = 255;
```

  Y con la mascara aplicada generamos una segunda binarización con el umbral de Otsu invirtiendo la binarización y limpiando otra vez con zonas de 120 px.

```
% Segunda binarización de la máscara
level  = graythresh(interior);
MKTemp = imbinarize(interior, level);   % umbral de Otsu
MKTemp = imcomplement(MKTemp);
MKTemp(~MASK) = 0;
MKFinal = bwareaopen(MKTemp, 120);      % elimina regiones pequeñas
MKFinal = bwareaopen(~MKFinal, 120);    % elimina huecos pequeños
MASK  = MKFinal;
```

- **Segmentación adaptativa**: Se ha utilizado un umbral adaptativo (adaptthresh de 11 pixeles) para adaptarse a variaciones locales de iluminación, lo que mejora la detección de gusanos con contraste variable aplicado con la máscara previamente calculada. Completamente necesario para imágenes con zonas oscuras donde se destaquen los gusanos pero por su diferencia con el fondo no porque sean más oscuros.

```
%% Detectar objetos oscuros con umbral adaptativo
% Prepara la imagen para binarizar solo dentro de la máscara
grayMasked = I;
grayMasked(~MASK) = 255;          % forzamos fuera del cristal a blanco
T  = adaptthresh(grayMasked, 0.6, 'NeighborhoodSize', 11);
BW0 = imbinarize(grayMasked, T);    % 1 = claro
BW = ~BW0;                          % gusanos (oscuros) → 1
BW(~MASK) = 0;                      % zona fuera = 0

% Forzamos TODO lo que esté fuera de la máscara a 0 (negro)
BW(~MASK) = 0;
```

  Una vez tratado nos genera una imagen con el contenido a tratar para encontrar los gusanos pero nos queda una marco de la máscara necesario para eliminar.

```
%% Eliminar Marco Mascara
se = strel('disk',1);                       % prueba radio 3-5px
innerMask = imerode(MASK, se);
% Limpiar cualquier componente que toque el borde de la imagen
BW_noframe = imclearborder(BW);
% Aplicar el interior recortado para quitar resto de marco
BW_noframe(~innerMask) = 0;
```

- **Filtrado morfológico**: Para mejorar la segmentación, se aplican operaciones como bwareaopen, imopen, imclose y imfill, que permiten eliminar ruido, suavizar bordes y cerrar huecos. En este caso se ha ido a prueba y error despacio para tartar de contornear lo mejor posible los gusanos y separarlos de otros que estén cerca y/o evitar que se junten con otros.

```
%% Filtrado morfológico
BW1 = bwareaopen(BW_noframe, 150);
se = strel('disk', 1);  % disco pequeño para suavizado ligero
BW1 = imopen(BW1, se); % Eliminar bordes
BW1 = bwareaopen(BW1, 150); % Eliminar Bloques
BW1 = imfill(BW1, 'holes'); % Cerrar huecos
BW1 = bwareaopen(BW1, 250); % Eliminar Bloques
BW1 = imopen(BW1, se); % Eliminar bordes
BW1 = imclose(BW1, strel('disk', 1));
```

- **Clasificación por excentricidad y visualización**: Se ha elegido la medida de **excentricidad** como criterio principal de clasificación:
    - Excentricidad cercana a **1** indica forma **recta** → *gusano muerto*.
    - Excentricidad significativamente menor indica forma **curva** → *gusano vivo*.

  Se generan imágenes de salida con las detecciones marcadas en rojo (muertos) y verde (vivos).

```
vivos = 0;
muertos = 0;

stats = regionprops(BW1, 'BoundingBox', 'Eccentricity', 'Centroid');
L = bwlabel(BW1);

% Mostrar y guardar imagen con anotaciones
fig = figure('Visible', 'off');
imshow(I), title('Gusanos detectados'); hold on;

for i = 1:length(stats)
    bb = stats(i).BoundingBox;
    centroid = stats(i).Centroid;
    ecc = stats(i).Eccentricity;

    if ecc > 0.99
        rectangle('Position', bb, 'EdgeColor', 'r', 'LineWidth', 1.5);
        text(centroid(1), centroid(2), 'Muerto', 'Color', 'r', 'FontSize', 8);
        muertos = muertos + 1;
    else
        rectangle('Position', bb, 'EdgeColor', 'g', 'LineWidth', 1.5);
        text(centroid(1), centroid(2), 'Vivo', 'Color', 'g', 'FontSize', 8);
        vivos = vivos + 1;
    end
end
```

- **Guardado y evaluación**: Guardamos los resultados (archivo, status, muertos, vivos) en un archivo CSV y las imágenes con los marcos en una carpeta.

```
if muertos < vivos
    status = "alive";
else
    status = "dead";
end
saveas(fig, fullfile(image_output_folder, [name(1:end-4) '_detect.png']));
close(fig);
```

Una vez tenemos todo toca comparar con los valores reales que previamente hemos tratado para facilitar sobre todo esta faena y obtener por la terminar el porcentaje de aciertos.

```
%% Contador de porcentaje sobre el numero de aciertos
T1 = readtable(ground_truth_file);
T2 = readtable(csv_file);

% Comparar columna 'Status' (segunda columna)
status1 = strtrim(string(T1{:,2}));
status2 = strtrim(string(T2{:,2}));

coinciden = strcmpi(status1, status2);
porcentaje = 100 * sum(coinciden) / numel(coinciden);
fprintf('Coincidencia en la columna "Status": %.2f%%\n', porcentaje);
```

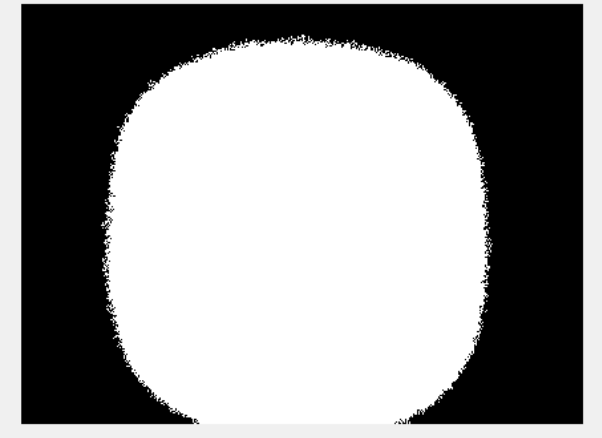
Problemas encontrados y soluciones

1. **Detección errónea en fondos oscuros:**
   En algunas imágenes, especialmente aquellas con fondo oscuro o contraste irregular, el umbral global de Otsu no era efectivo y generaba falsos positivos.
   Solución: Se sustituyó el umbral de Otsu por un umbral adaptativo (adaptthresh) con vecindarios pequeños, lo que permitió una segmentación más precisa ajustada a la iluminación local.
2. **Detección incorrecta de bordes o marcos del portaobjetos:**
   Parte del borde de la imagen o el cristal se detectaba como objeto válido.
   Solución: Se aplicó una máscara para aislar el área útil del cristal mediante erosión morfológica y se recortó visualmente la imagen ignorando los píxeles cercanos al borde.
3. **Detectar gusanos cruzados como objetos únicos:**
   Cuando dos gusanos se cruzaban o se tocaban, eran detectados como un solo componente conectado.
   Solución tentativa: Se optimizó el filtrado morfológico ajustando los tamaños de apertura y cierre, pera que los gusanos se destacaran lo mejor posible para evitar que se juntaran y se curazaran individualizándolos lo máximo posible siendo exitoso en casos de acercamientos, pero falla cuando los gusanos se cruzan enteramente.
4. **Detección de múltiples gusanos como una única región:**
   Algunas regiones agrupaban más de un gusano, especialmente si había huecos internos o solapamientos.
   Solución: Se aplicó relleno de agujeros (imfill) y eliminación de secciones con pocos píxeles, para separar componentes individuales y descartar zonas pequeñas no válidas.

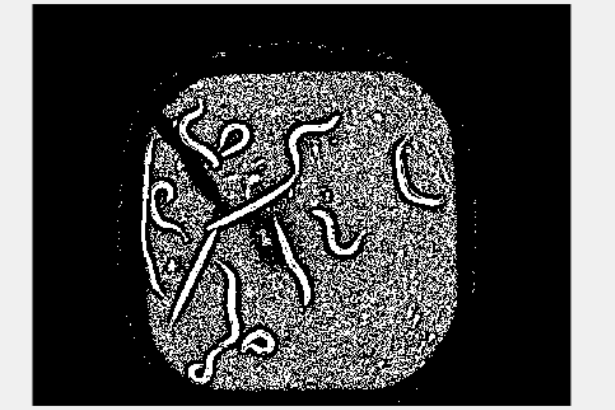
Evolución de una imagen

Para poner un caso como ejemplo se ha decidió poner el caso de la imagen 1:

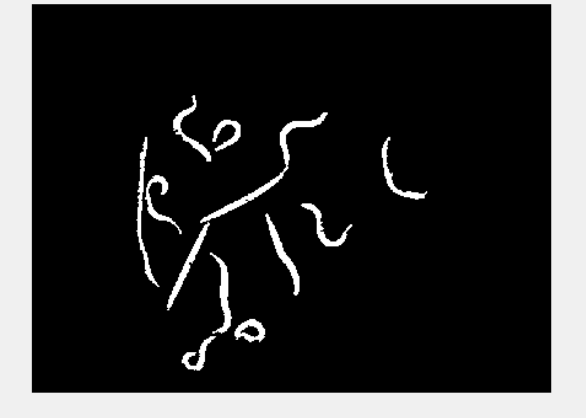
1. Mascara de la zona de los gusanos:



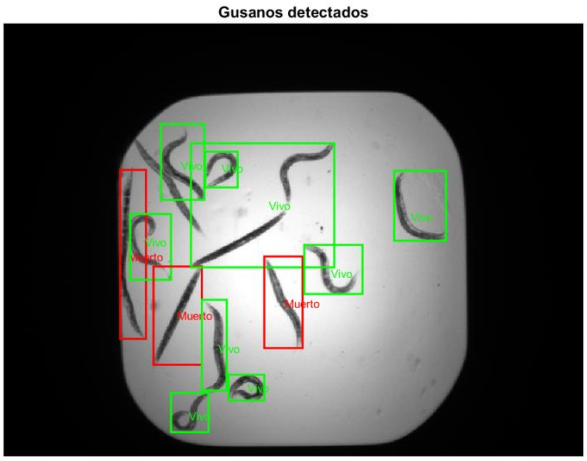
2. Aplicación de la máscara sobre el fondo junto con el umbral adaptativo :



3. Análisis morfológico de limpieza y separación de gusanos:



4. Análisis de los gusanos:



5. Guardado de datos en .csv

	A	B	C	D
1	Nombre_fiche	Status	Muertos	Vivos
2	wormA01.tif	alive	3	9

Juego de pruebas

La evaluación del sistema se ha llevado a cabo mediante un conjunto de 24 imágenes proporcionadas junto a un archivo .csv que contiene la clasificación esperada (basada en el número de gusanos vivos y muertos) para cada una de ellas. La validación se ha realizado de manera manual y visual, comparando los resultados obtenidos por el algoritmo con la información del fichero de referencia, las imágenes que se han tenido más relevancia a la hora del análisis han sido la 1 por ser bastante estándar y por tener un gusano pegado a la pared y la 12 por su baja iluminación.

Para cada imagen, se ha inspeccionado la versión generada con anotaciones visuales (gusanos vivos en verde, muertos en rojo), y se ha comprobado si la clasificación global de la imagen determinada por la mayoría de gusanos identificados como vivos o muertos coincidía con la anotación proporcionada.

En este proceso se ha observado que el algoritmo ha clasificado correctamente 22 de las 24 imágenes, fallando en las imágenes número 9 y 10. En ese caso particular, se produjo una discrepancia entre la proporción de gusanos detectados como vivos y muertos respecto a la referencia, posiblemente causada por una detección incorrecta de la forma en gusanos solapados o parcialmente visibles.

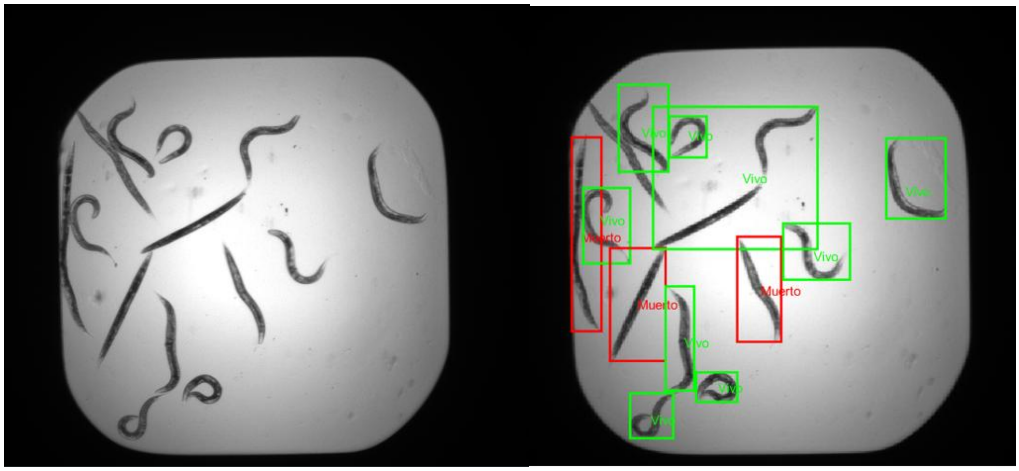
Este resultado refleja una precisión del 91,67%, lo cual es muy satisfactorio teniendo en cuenta la complejidad del problema, las diferencias de iluminación, los solapamientos entre gusanos y la variabilidad morfológica. La combinación de técnicas de binarización adaptativa, filtrado morfológico y clasificación por excentricidad ha demostrado ser eficaz para la mayoría de los casos.

Este análisis refuerza la fiabilidad del sistema en contextos similares, aunque también sugiere que podría mejorarse aún más mediante técnicas adicionales, como separación de componentes solapados o redes neuronales para refinar la clasificación morfológica en casos ambiguos.

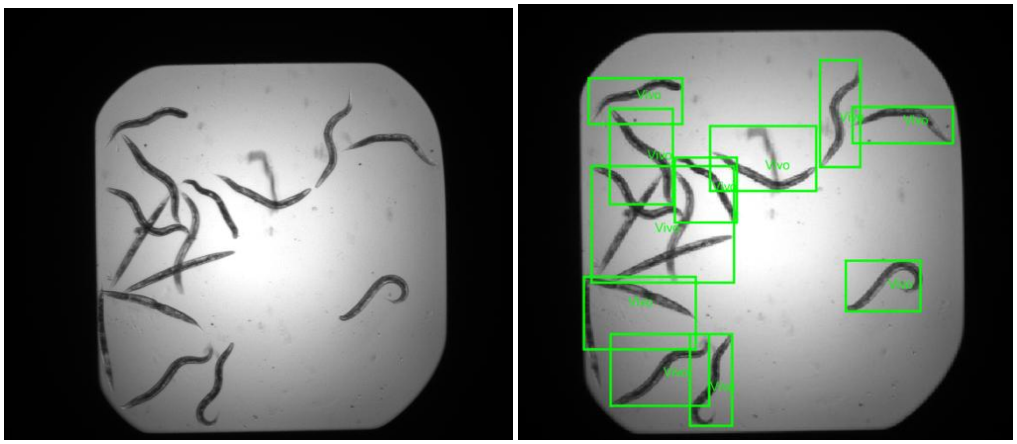
	A	B	C	D
1	Nombre_fiche	Status	Muertos	Vivos
2	wormA01.tif	alive	3	9
3	wormA02.tif	alive	0	11
4	wormA03.tif	alive	0	3
5	wormA04.tif	alive	2	7
6	wormA05.tif	alive	5	8
7	wormA06.tif	alive	4	7
8	wormA07.tif	alive	3	4
9	wormA08.tif	alive	2	10
10	wormA09.tif	dead	9	7
11	wormA10.tif	dead	10	6
12	wormA11.tif	alive	5	12
13	wormA12.tif	alive	2	12
14	wormA13.tif	dead	12	6
15	wormA14.tif	dead	15	5
16	wormA15.tif	dead	11	5
17	wormA16.tif	dead	7	6
18	wormA17.tif	dead	9	3
19	wormA18.tif	dead	15	1
20	wormA19.tif	dead	7	6
21	wormA20.tif	dead	14	2
22	wormA21.tif	dead	9	4
23	wormA22.tif	dead	14	1
24	wormA23.tif	dead	10	1
25	wormA24.tif	dead	10	3

```
>> Analisis_gusanos
Warning: Column headers from the file were mod
headers are saved in the VariableDescriptions
Set 'VariableNamingRule' to 'preserve' to use
Coincidencia en la columna "Status": 91.67%
```

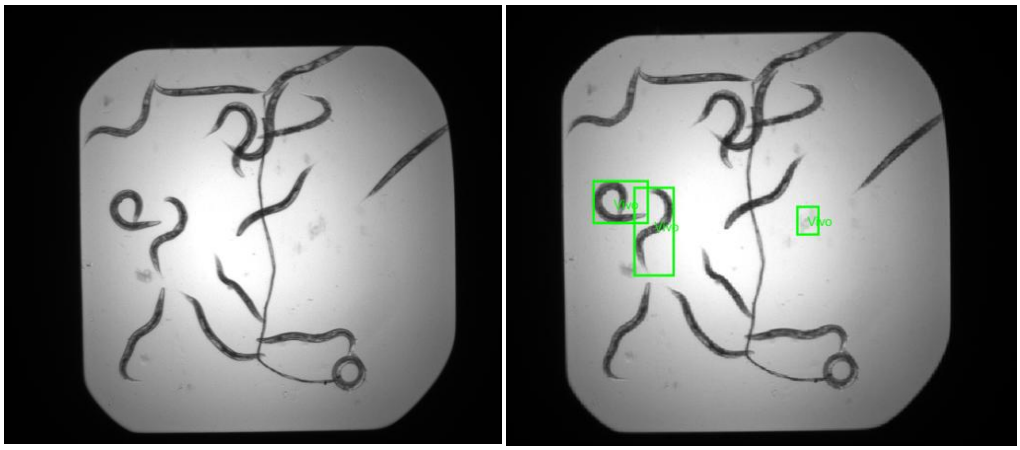
Análisis Imagen 1:



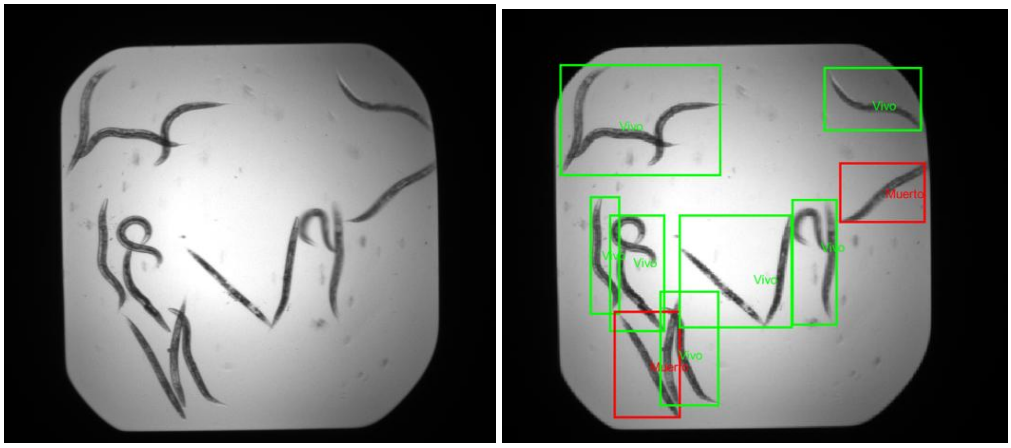
Análisis Imagen 2:



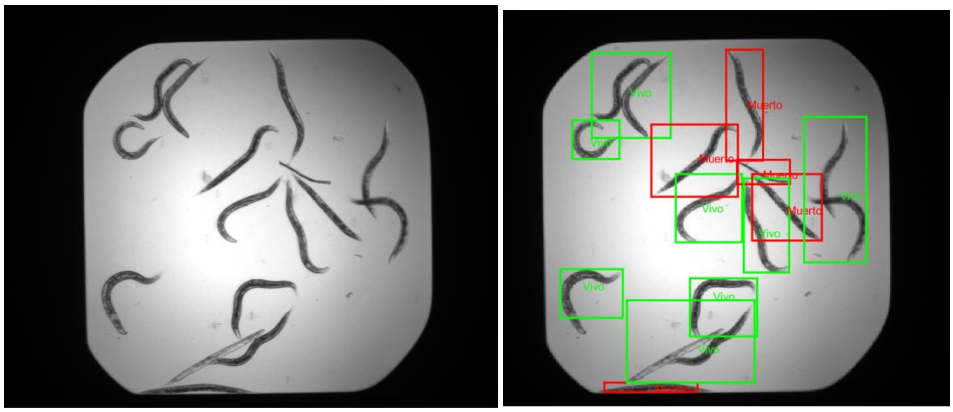
Análisis Imagen 3:



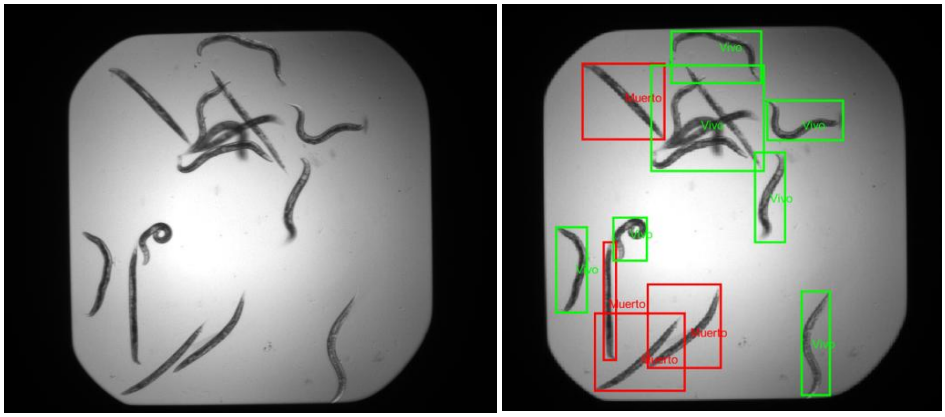
Análisis Imagen 4:



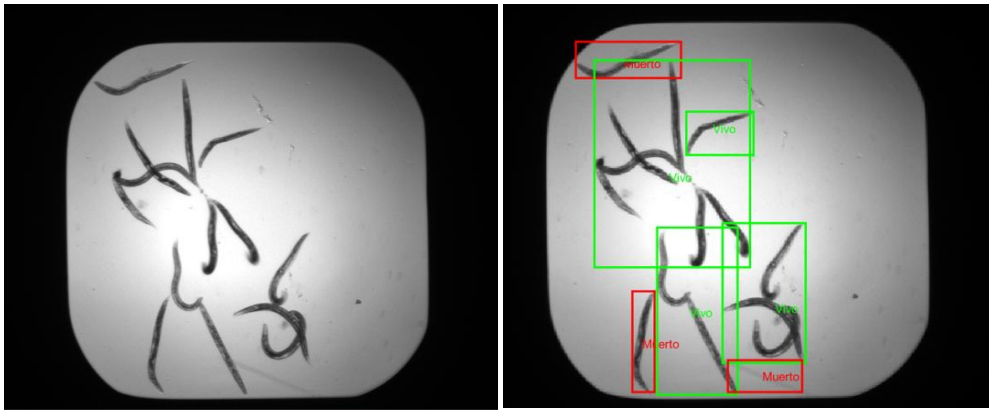
Análisis Imagen 5:



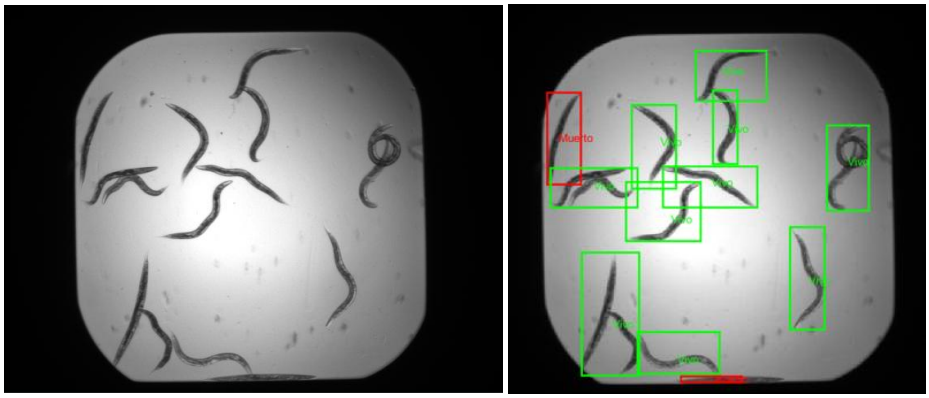
Análisis Imagen 6:



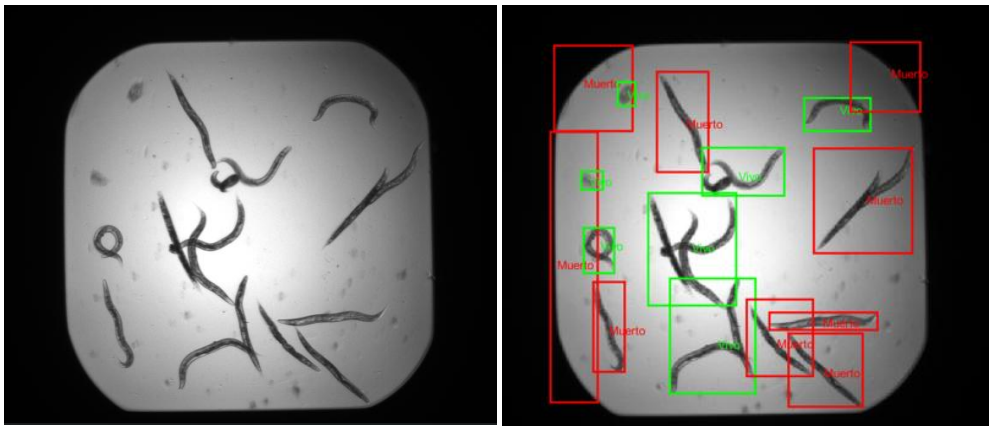
Análisis Imagen 7:



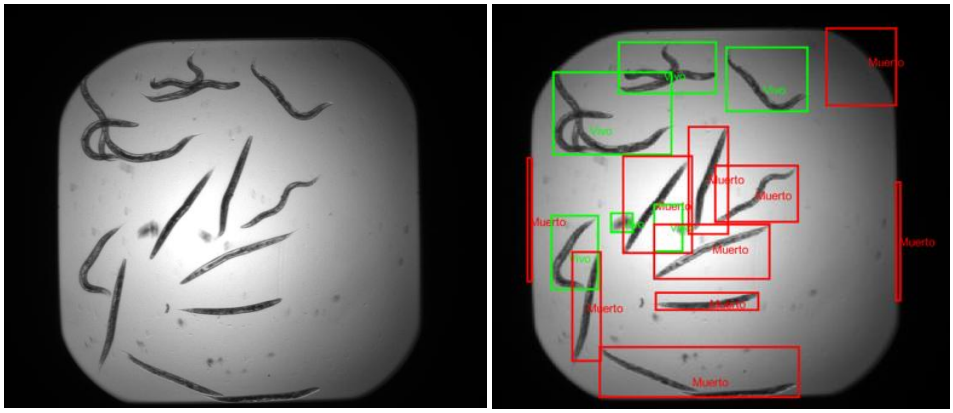
Análisis Imagen 8:



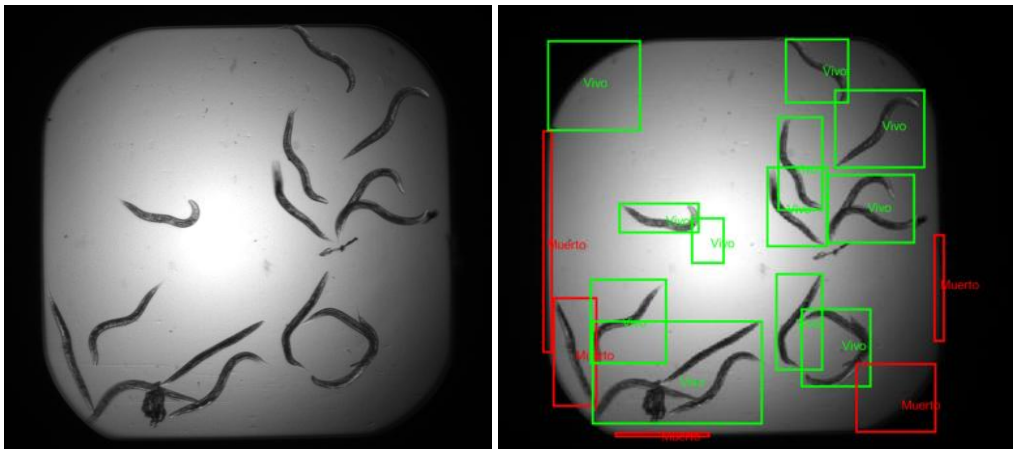
Análisis Imagen 9:



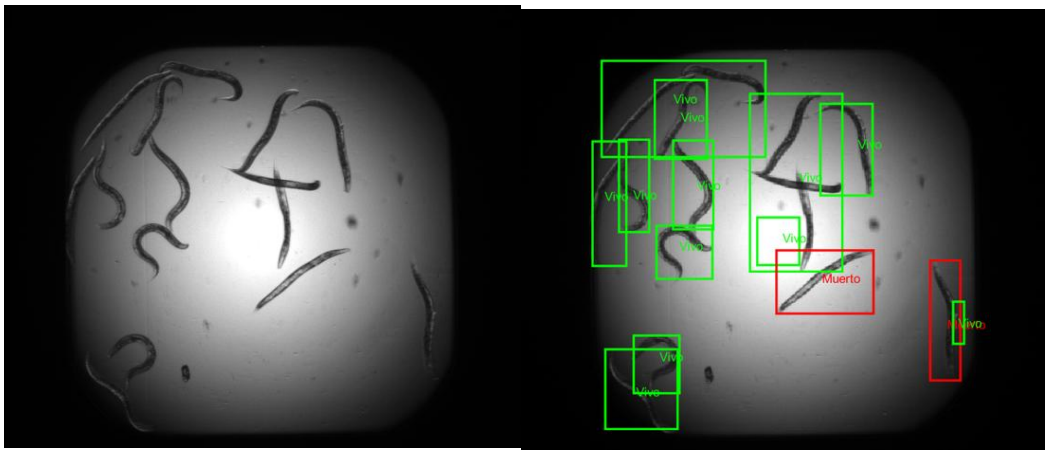
Análisis Imagen 10:



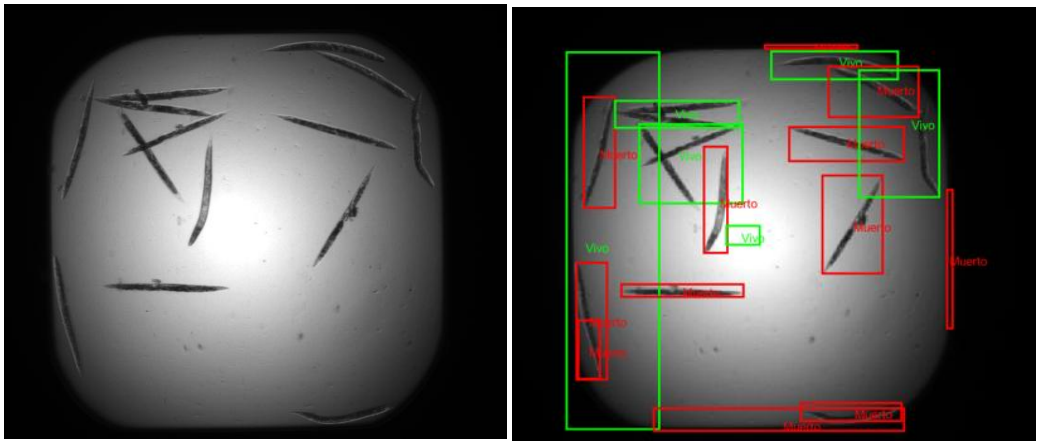
Análisis Imagen 11:



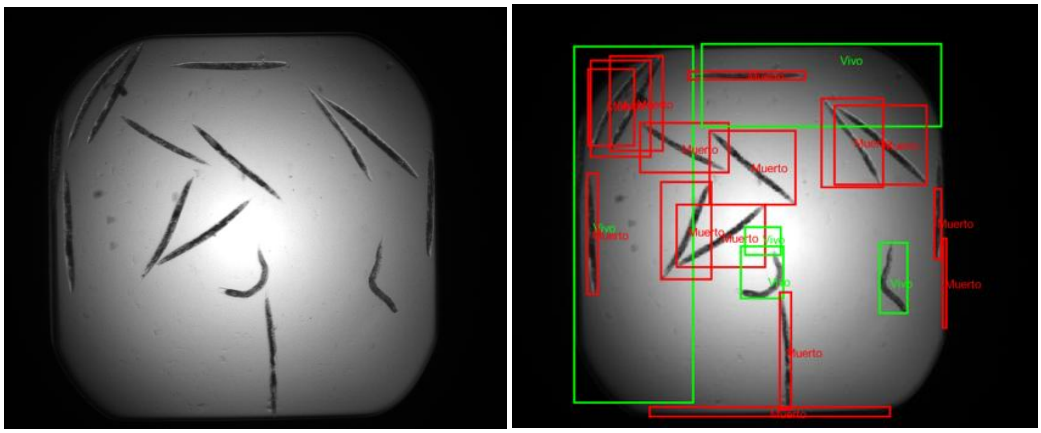
Análisis Imagen 12:



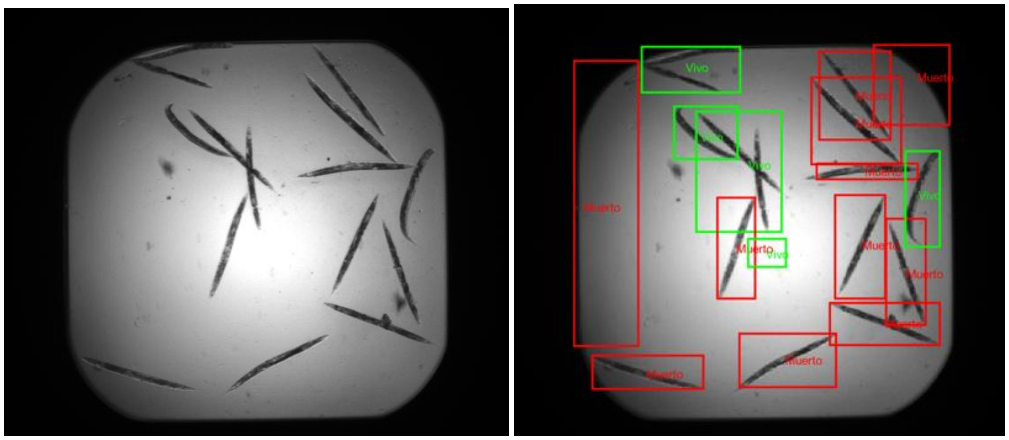
Análisis Imagen 13:



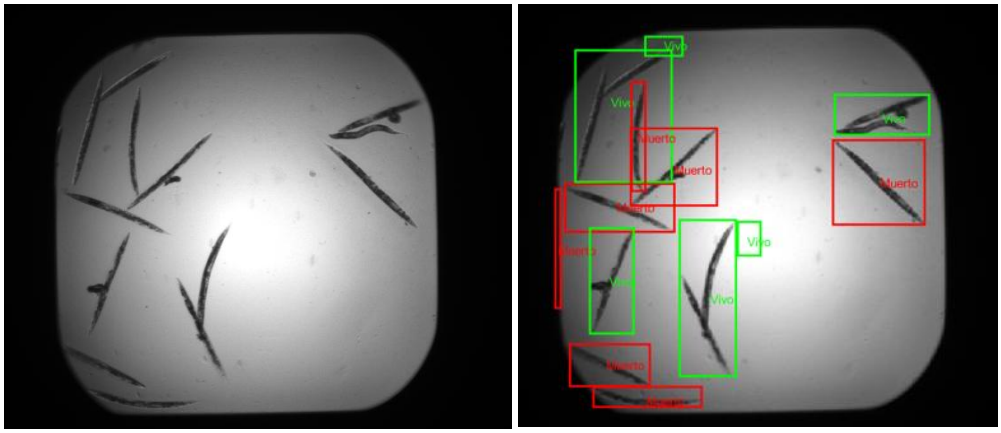
Análisis Imagen 14:



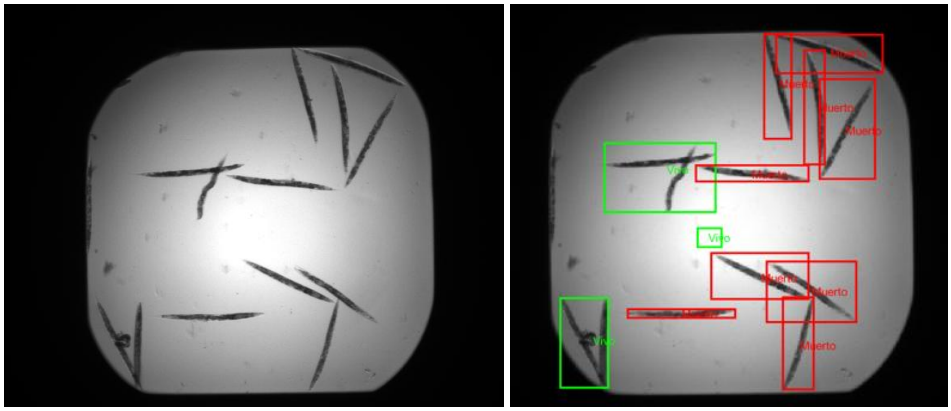
Análisis Imagen 15:



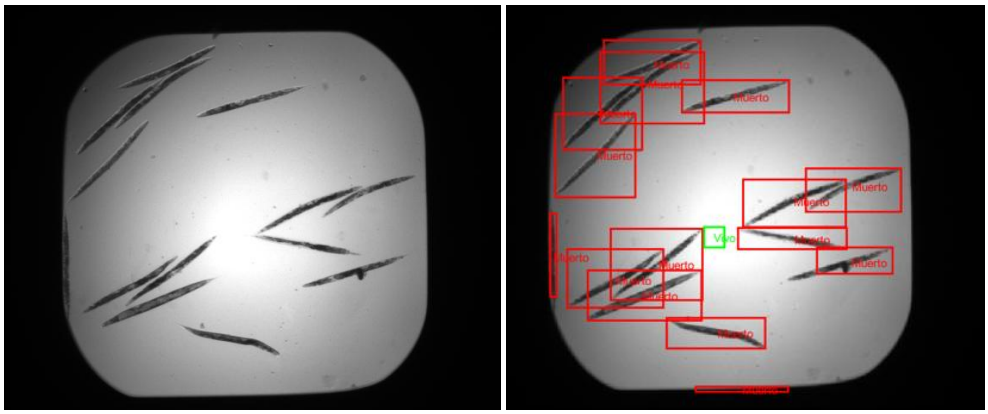
Análisis Imagen 16:



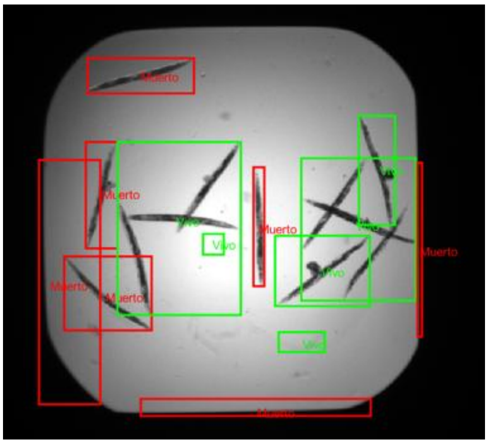
Análisis Imagen 17:



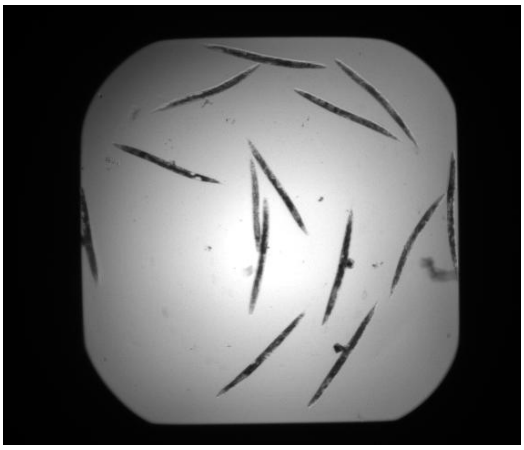
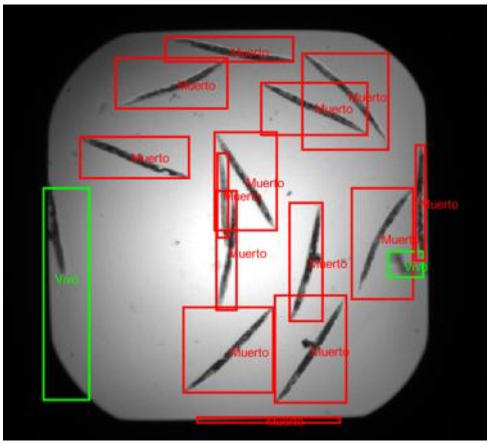
Análisis Imagen 18:



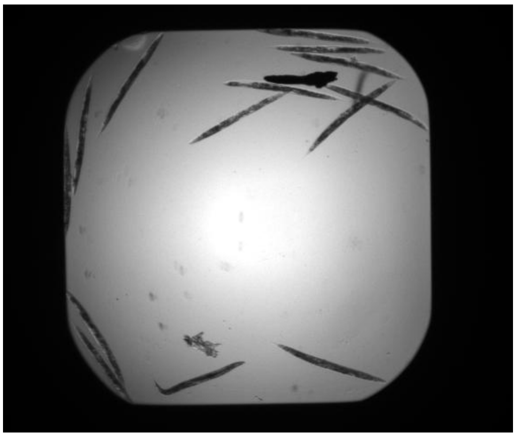
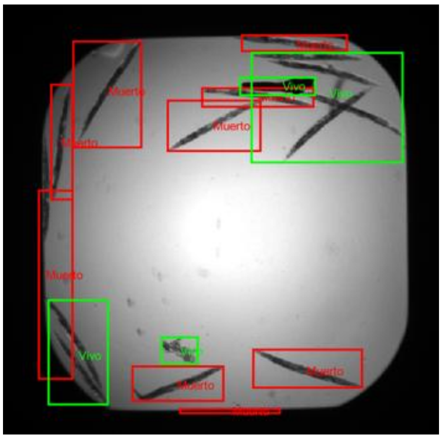
Análisis Imagen 19:

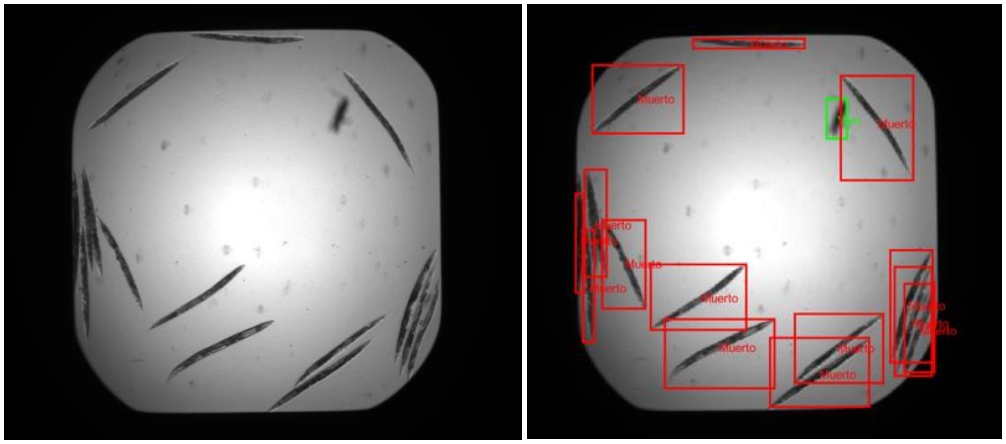
Análisis Imagen 20:

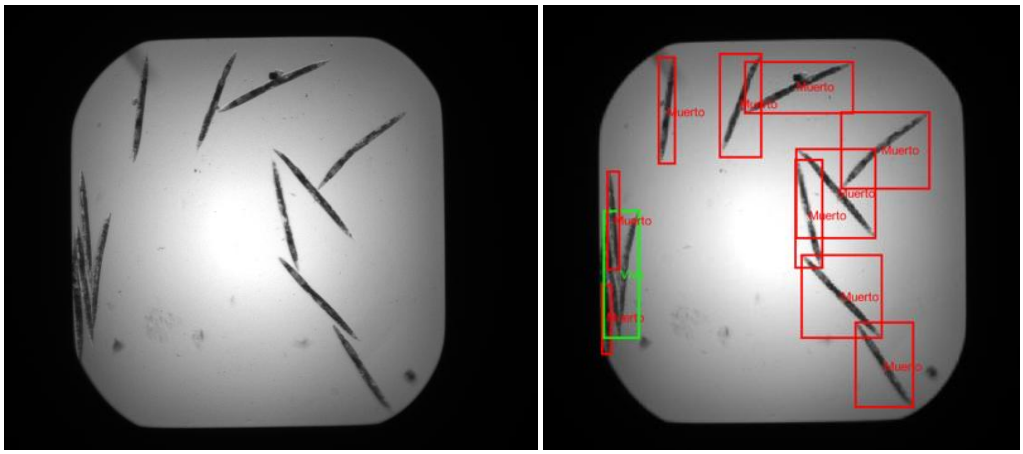
Análisis Imagen 21:

Análisis Imagen 22:



Análisis Imagen 23:



Análisis Imagen 24: