

UNIVERSIDAD NACIONAL DEL SUR

INTRODUCCIÓN AL PROCESAMIENTO DIGITAL DE IMÁGENES

SEGUNDO CUATRIMESTRE 2021

TRABAJO PRÁCTICO INTEGRADOR



Profesor:

Dr. Delrieux, Claudio

Autor:

Parma Paternostro, Nicolás Daniel

15 de febrero de 2022

1. Introducción

En el presente trabajo se desarrollará sobre las herramientas utilizadas para poder detectar la frontera de un cultivo de células de levadura (*Saccharomyces cerevisiae*) y posteriormente calcular su área. Esta problemática a resolver resulta de gran interés ya que permitiría obtener la ley de control sobre el crecimiento de células frente a la exposición ante distintos fluidos, pudiendo así acelerar o desacelerar el crecimiento de la célula según sea necesario. El trabajo fue planteado como una pequeña problemática a resolver dentro de un proyecto multidisciplinario de investigación en la que se encuentra trabajando el LCI - UNS en conjunto con grupos de distintas universidades, entre ellas UTN y UBA.

La tecnología utilizada para capturar las imágenes se llama Lab on a chip. En dicho chip se encuentran distintos pocillos con microbombas individuales que inyectan distintos fluidos, permitiendo realizar múltiples experimentos en simultáneo.

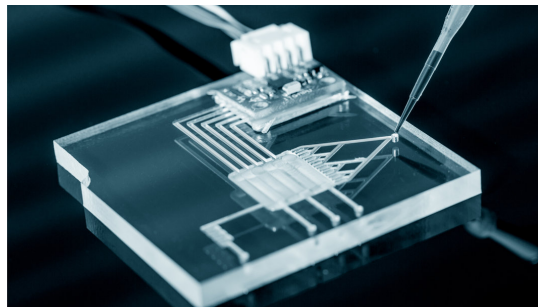


Figura 1. Lab on a Chip.

Las imágenes a partir de las que se trabajaron para obtener una primera aproximación a la resolución del problema de la segmentación del cultivo son las siguientes.

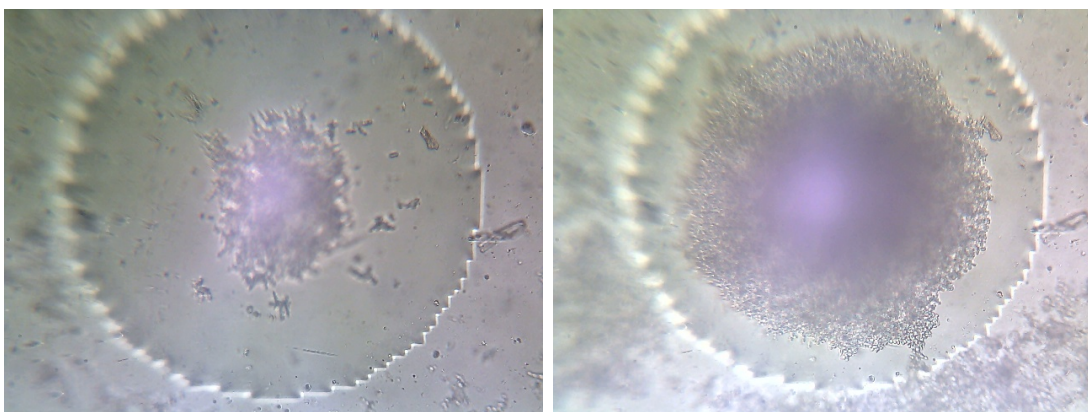


Figura 2. Imágenes obtenidas a partir del ensayo.

Donde el diámetro del pocillo es de $700\mu m$, este dato será utilizado al momento de calcular el área del cultivo de células. Las imágenes se toman cada 30 minutos, con el fin de obtener un ensayo con múltiples imágenes en el cual se pueda ir observando la evolución en cuanto a superficie de la célula a estudiar.

2. Desarrollo

2.1. Segmentación de imágenes

El primer paso para poder computar el área del cultivo es detectar correctamente su frontera. Esto se realiza utilizando un algoritmo de segmentación que permita agrupar un conjunto de píxeles que tienen características similares. Para poder llevar esto a cabo, se utilizó el método de segmentación contorno activo (Snakes). El mismo funciona minimizando la energía, deformando la curva generada inicialmente por el usuario (es un método semi-supervisado), influenciada por restricciones y fuerzas tanto dentro de la imagen que la fuerzan hacia los contornos como fuerzas internas que resisten a dicha deformación.

Para la implementación de dicho algoritmo, se utilizó la librería scikit-image. La función utilizada tiene los siguientes parámetros:

```
skimage.segmentation.active_contour(image, snake, alpha=0.01, beta=0.1, w_line=0,
w_edge=1, gamma=0.01, max_px_move=1.0, max_num_iter=2500, convergence=0.1, *,
boundary_condition='periodic', coordinates='rc')
```

Donde los parámetros principales a utilizar son los siguientes:

- `image`: Imagen de entrada a segmentar.
- `snake`: Coordenadas de la superficie del contorno inicial (la cantidad de puntos utilizados es arbitraria).
- `alpha`: Parámetro del largo del contorno, valores mas altos implican que el contorno se contraiga más rápido.
- `beta`: Parámetro de la suavidad del contorno, valores más altos conllevan que el mismo sea más suave.
- `gamma`: Parámetro explícito sobre el tiempo entre iteraciones.

Los demás parámetros presentes no fueron modificados, por lo que la función utiliza los valores por defecto. En cuanto a los parámetros modificados, los valores de `alpha`, `beta` y `gamma` serán detallados según la imagen a segmentar. Para el parámetro `snake` (contorno inicial) se utilizó una circunferencia conformada por 800 puntos, externa a la frontera de la levadura e interna al pocillo y levemente desfasada del centro de la imagen para que coincida en mayor medida con el centro de la probeta. La función *active_contour* devuelve un arreglo de (N,2) donde N corresponde al número de puntos utilizados para la circunferencia inicial a ajustar (en este caso 800), mientras que el dos corresponde al par de coordenadas de cada punto del contorno una vez optimizado.

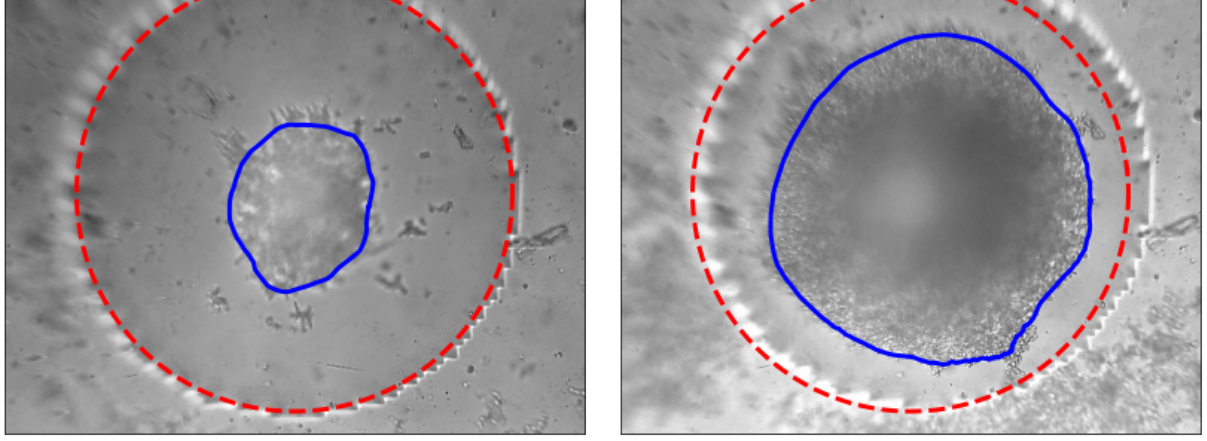


Figura 3. Imágenes segmentadas.

Para el caso de la primer imagen se utilizó un alpha de 0.7 (debido a que la superficie es menor), un beta de 2 y un gamma de 0.005, mientras que para la segunda imagen se utilizó un alpha de 0.45, mismo beta y mismo gamma. Para las siguientes secciones, se aplicará un leve filtrado Gaussiano en las imágenes para suavizar las mismas y tener menor variación entre imagen e imagen.

2.2. Cálculo del área

Para poder resolver el problema del cálculo del área, primero se debe conocer la cantidad de píxeles a los que corresponde la superficie del pocillo, ya que el mismo tiene un diámetro conocido y es posible extrapolar dicha información para la superficie del cultivo de células. En primer lugar, como fue mencionado durante la introducción, se sabe que el diámetro del pocillo es de $700\mu m$, por lo que se corresponde a un área de $2\pi(350)^2\mu m^2 = 384845,1\mu m^2$.

Una vez obtenido a cuantos píxeles corresponde el área de la probeta, se obtiene nuevamente el área de la probeta pero esta vez en píxeles, para poder obtener el área del cultivo en μm^2 a partir de su valor en píxeles. Para obtener el área en píxeles de la probeta y posteriormente el contorno, se utilizó la fórmula de Shoelace, la cual permite calcular el área de un polígono a partir de un conjunto de coordenadas. Luego, mediante una simple extrapolación es posible determinar el área del contorno en micrómetros cuadrados.

La fórmula de Shoelace está definida como:

$$A = \frac{1}{2} \left| \sum_{i=1}^{n-1} x_i y_{i+1} + x_n y_1 - \sum_{i=1}^{n-1} x_{i+1} y_i - x_1 y_n \right|$$

y puede aplicarse de manera sencilla en código mediante el uso de productos puntos. Para Python el código utilizado para obtener el área en píxeles es la siguiente

```
A = 0.5*np.abs(np.dot(x,np.roll(y,1))-np.dot(y,np.roll(x,1)))
```

Donde la función $roll(arreglo, n)$ traslada los últimos n elementos dentro del arreglo y los coloca al comienzo. En este caso se cambia de lugar únicamente el último par coordenado ya que a la función se le ingresa un uno. Esto se realiza así ya que es una condición necesaria al momento de realizar el cómputo del área que se comience operando sobre el último par coordenado.

Como fue mencionado anteriormente, una vez obtenida el área en píxeles del pocillo, basta con realizar una extrapolación para obtener el área en μm^2 del cultivo, la línea de código utilizado es la siguiente

$$MareaSnake = areaSnake * MareaTestTube / areaTestTube$$

Donde el prefijo M corresponde al área en μm^2 , areaSnake al área del contorno del cultivo y areaTestTube corresponde a la superficie del pocillo.

Además de obtener el valor del área, es deseable computar la circunferencia equivalente del contorno detectado, tanto para verificar que la detección del contorno detectado sea óptima como para comprobar que el cómputo del área es correcta. Para ello, una vez calculada el área de la curva en píxeles, simplemente se despeja el radio utilizando la fórmula del área de un círculo. La expresión utilizada es la siguiente $Radio_{equiv} = \sqrt{area_{pix}/\pi}$, donde el centro de la circunferencia utilizado es el mismo que se uso para la segmentación. Luego de obtener los distintos valores de áreas, se procedió a graficar las distintas circunferencias calculadas, obteniendo así la siguiente imagen.

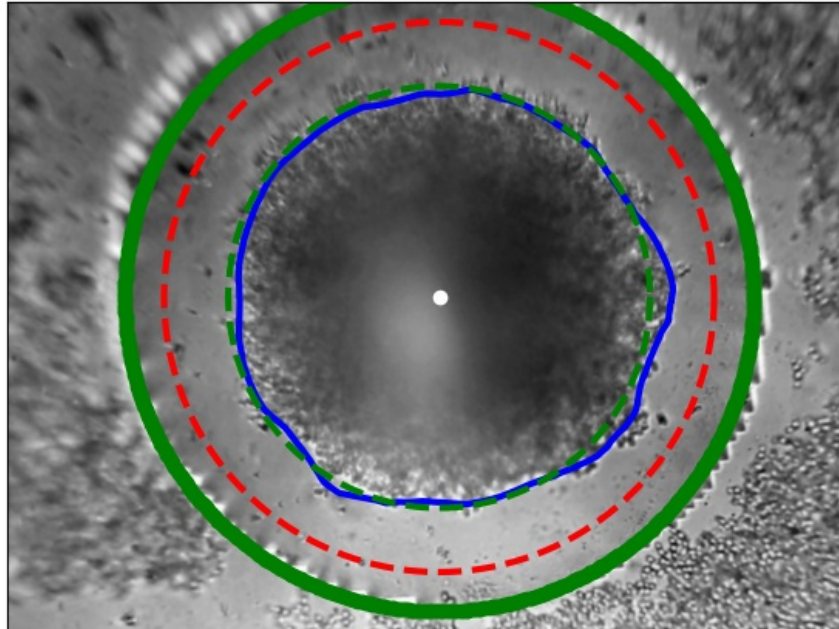


Figura 4. Imagen con las distintas circunferencias.

Donde el punto blanco corresponde al centro del cultivo, el trazo en azul continuo el contorno detectado, el trazo en verde discontinuo la circunferencia equivalente, el trazo rojo discontinuo el

contorno con el que se inicializó el algoritmo y el trazo verde continuo la circunferencia del pocillo utilizado para los cálculos de área.

2.3. Cómputo en lotes (Batch)

Cada ensayo consta de alrededor de 100 imágenes, por lo que es deseable poder calcular el área de todas las imágenes mediante un único script. Para poder llevar esto a cabo, se desarrolló un programa que permitiese abrir un conjunto de imágenes dentro de una carpeta y posteriormente realizar la segmentación y cálculo de las distintas áreas mencionados en las subsecciones anteriores. Además, es deseable disponer de los resultados obtenidos en cuanto a área en micrones cuadrados y radio equivalente del cultivo celular en un planilla de cálculos para poder realizar gráficos y obtener las tendencias de crecimiento del mismo a lo largo del ensayo. Como fue mencionado anteriormnete, esto es deseable ya que permite obtener una ley de control frente a la exposición de distintos fluidos.

Para poder llevar a cabo esta tarea, se utilizó la librería glob, la cual permite obtener una lista con todos los archivos dentro de un directorio, por lo que mediante un simple ciclo for es posible realizar la segmentación y cálculo del área para cada imagen. Luego, para poder escribir los resultados requeridos en una planilla de cálculos, se utilizó la librería pandas. En primer lugar, se creo una lista vacía para cada resultado que se desee almacenar y en cada iteración del ciclo for se utilizó el método *append()* para ir agregando los resultados deseados.

	A	B	C	D
1		Nombre	Área	radio eq
2	0	2_ensayotENSAYO1_1.jpg	39480.414	112.10266
3	1	2_ensayotENSAYO1_2.jpg	40592.159	113.67007
4	2	2_ensayotENSAYO1_3.jpg	90885.894	170.08786
5	3	2_ensayotENSAYO1_4.jpg	1671.6512	23.06736
6	4	2_ensayotENSAYO1_5.jpg	314.0013	9.9974857
7	5	2_ensayotENSAYO1_6.jpg	76229.952	155.77146
8	6	2_ensayotENSAYO1_7.jpg	34724.263	105.13361
9	7	2_ensayotENSAYO1_8.jpg	40492.933	113.53106
10	8	2_ensayotENSAYO1_9.jpg	27382.305	93.359833
11	9	2_ensayotENSAYO1_10.jpg	35544.174	106.36758
12	10	2_ensayotENSAYO1_11.jpg	81385.093	160.95241
13	11	2_ensayotENSAYO1_12.jpg	88913.078	168.23172
14	12	2_ensayotENSAYO1_13.jpg	106599.69	184.20569
15	13	2_ensayotENSAYO1_14.jpg	118432.4	194.16025
16	14	2_ensayotENSAYO1_15.jpg	138162.36	209.71038
17	15	2_ensayotENSAYO1_16.jpg	155861.47	222.75236
18	16	2_ensayotENSAYO1_17.jpg	166405.31	230.14877
19	17	2_ensayotENSAYO1_18.jpg	166126.84	229.95611
20	18	2_ensayotENSAYO1_19.jpg	167039.03	230.58659
21	19	2_ensayotENSAYO1_20.jpg	167095.9	230.62584
22	20	2_ensayotENSAYO1_21.jpg	168294.12	231.45125
23	21	2_ensayotENSAYO1_22.jpg	164806.22	229.04028
24	22	2_ensayotENSAYO1_23.jpg	164746.2	228.99857
25	23	2_ensayotENSAYO1_24.jpg	166029.96	229.88905
26	24	2_ensayotENSAYO1_25.jpg	169842.16	232.51331
27	25	2_ensayotENSAYO1_26.jpg	166814.22	230.43137
28	26	2_ensayotENSAYO1_27.jpg	169438.93	232.23713
29	27	2_ensayotENSAYO1_28.jpg	163794.86	232.48093
30	28	2_ensayotENSAYO1_29.jpg	169944.32	232.58323
31	29	2_ensayotENSAYO1_30.jpg	172520.49	234.33945
32	30	2_ensayotENSAYO1_31.jpg	173219.53	234.81373
33	31	2_ensayotENSAYO1_32.jpg	174776.35	235.86657
34	32	2_ensayotENSAYO1_33.jpg	173270.92	234.84856
35	33	2_ensayotENSAYO1_34.jpg	172937.46	234.62247
36	34	2_ensayotENSAYO1_35.jpg	171551.29	233.68027
37	35	2_ensayotENSAYO1_36.jpg	172727.41	234.47993

Figura 5. Hoja de cálculos con los resultados.

A su vez, se guardaron todas las imágenes obtenidas en un directorio aparte para poder comparar las segmentaciones de cada imagen y evaluar los resultados obtenidos. Cabe mencionar que para las primeras 10 imágenes se utilizó un α de 0.65 y para las restantes un α de 0.4 (los demás parámetros permanecieron constantes). Esto se realizó con el fin de proveer segmentaciones decentes para todo el batch, con un cierto grado de error pero con una mayor simplicidad ya que se podría haber definido una base de más valores de α o bien realizar una automatización para la obtención del valor, pero ya comenzaría a ser un proceso muy complejo que no es deseable para esta instancia del proyecto.

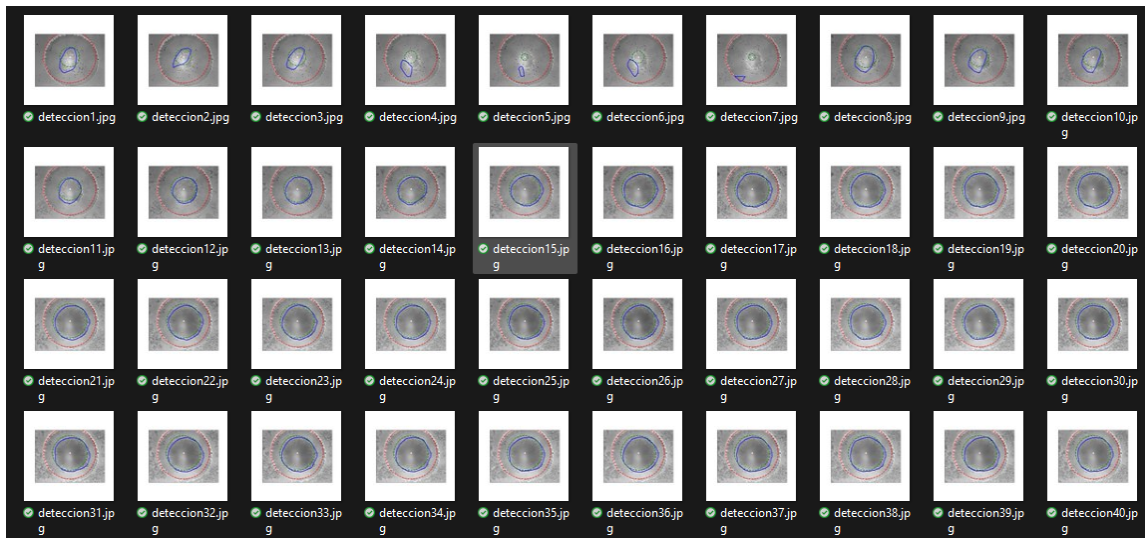


Figura 6. Imágenes obtenidas.

2.4. Mejoras implementadas

Cómo se puede observar de la Figura 5, las detecciones no son del todo correctas para algunas de las primeras imágenes del batch, esto se debe por varios motivos:

- Si se fija un parámetros de α , β y γ para todo el lote, los resultados obtenidos no son acordes para todos los casos.
- Cambios en el enfoque o luminancia de las imágenes provocan cambios en la frontera del cultivo, ocasionando detecciones incorrectas del mismo.
- Es necesario el ajuste de parámetros entre lote y lote.

Para tratar de solucionar dichos inconvenientes se plantearon una serie de estrategias. En primera instancia se probó con distintos filtrados (Laplaciano, Sobel, inverse Gaussian filter) sin lograr mejoras visibles en los resultados en primera instancia. Luego, se utilizó el filtrado Gaussiano mejorando la detección ya que se reduce la alta frecuencia provista por las células aisladas. El sigma utilizado para el filtrado está entre 0 y 1 dependiendo del batch a segmentar.

Además, se establecieron una serie de condiciones con el fin de evitar detecciones incorrectas. Se estableció que el área mínima es de $3000\mu m^2$ ya que como se observa en las figuras anteriores, la segmentación de las imágenes 5 y 7 es incorrecta con un área prácticamente nula. En caso que el área sea menor al mínimo, se realiza nuevamente la segmentación pero restándole 0.1 a alpha ya que esto ocasiona que el contorno se contraiga en menor medida. Otro valor de corrección de alpha con resultados aceptables es multiplicar al mismo por un factor de $2/3$.

Además, es deseable evitar cambios bruscos de la superficie del contorno entre imagen e imagen debido a la naturaleza del ensayo. Es decir, si el área computada resulta ser menor al 50% de la imagen previa, nuevamente se realiza una corrección de alpha, multiplicándolo por $2/3$ para que el parámetro de parada sea menor y así obtener una segmentación más acorde. De la misma manera, si el nuevo área resulta ser mayor al 150%, se multiplica al parámetro alpha por $3/2$ con el fin de tener un mayor ajuste del contorno.

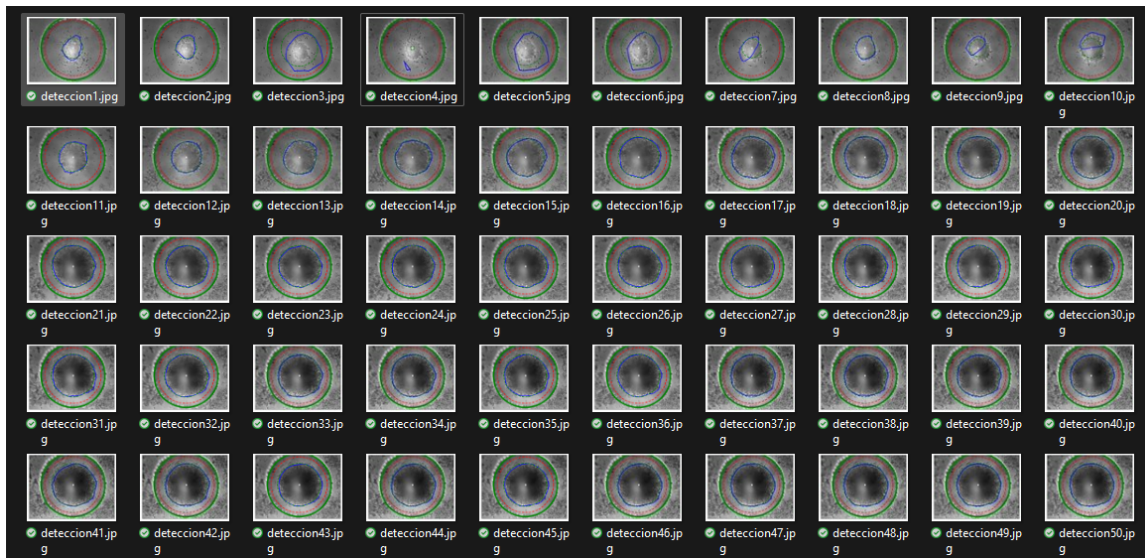


Figura 7. Imágenes resultantes al aplicar las distintas mejoras.

2.5. Aspectos a mejorar

Como se puede observar, los resultados para las primeras imágenes no son del todo correcto aun habiendo aplicado distintos métodos para tratar de mejorar las mismas. Se intuye que la solución óptima implica algún tipo de filtrado más acorde que los utilizados y/o realizar alguna operación sobre las imágenes con el fin de solo retener el cultivo celular y poder segmentar la misma de manera óptima independientemente de la luminosidad de la imagen, las células "sueltas" sobre el pocillo que introducen alta frecuencia y las diferencias que existen entre lote y lote.

3. Conclusiones

En el presente trabajo se pudo desarrollar un método de detección de fronteras y cálculo de área de un cultivo celular cuya segmentación no termina de ser óptimo para todas las imágenes del ensayo. La continuación y optimización del mismo quedará a cargo del LCI - UNS. Sin embargo, cabe destacar que dejando de lado algunas de las primeras imágenes, los resultados son mas que aceptables y los cálculos de áreas correctos. Esto último se pudo comprobar gracias a la obtención y graficación de la circunferencia equivalente al contorno detectado.

Por otro lado, se pudo conocer e implementar distintas técnicas introductorias para la segmentación de imágenes, que con sus debidos ajustes y modificaciones serán útiles como primera aproximación para el desarrollo de proyectos o tecnologías que impliquen la detección de objetos deseados en distintos ambientes. Cabe destacar la importancia de esto último ya que distintas tecnologías innovadoras de alto impacto como lo son la conducción autónoma, monitoreo de los elementos de seguridad en industrias o semaforización inteligente entre otros, hacen uso de la segmentación de imágenes.

4. Referencias

- Contornos activos: https://scikit-image.org/docs/dev/auto_examples/edges/plot_active_contours.html
- Cálculo área polígono: <https://stackoverflow.com/questions/24467972/calculate-area-of-polygon-given-x-y-coordinates>
- Fórmula de Shoelace: https://en.wikipedia.org/wiki/Shoelace_formula
- Numpy Roll: <https://numpy.org/doc/stable/reference/generated/numpy.roll.html>
- Procesar imágenes por lotes: <https://www.youtube.com/watch?v=QxzxLVzNfbI>
- Escribir variables a hoja de cálculos: <https://www.geeksforgeeks.org/exporting-a-pandas-dataframe-to-an-excel-file/>
- Ordenar batch: <https://www.py4u.net/discuss/17227>