

LABORATORIO DE OPERACIONES BÁSICAS EN IMAGEN DIGITAL

Cristofher Solís, José David Soto
cristofhersj@gmail.com, josedavidsz@hotmail.com
Área académica de Ingeniería Mecatrónica
Instituto Tecnológico de Costa Rica

Resumen

En el presente informe se presenta el desarrollo del "LABORATORIO DE OPERACIONES BÁSICAS EN IMAGEN DIGITAL", el cual plantea cuatro ejercicios por realizar y analizar referentes a imágenes digitales. Se documentan asimismo las decisiones tomadas con su respectiva justificación basándose en la teoría del curso MT9008 Sistemas de Visión.

Palabras clave

1.Brillo , 2.Contraste, 3.Imagen, 4.Reflexión, 5.Visión Por Computadora

I. INTRODUCCIÓN

El presente laboratorio propone la realización de cuatro ejercicios, se plantea utilizar como herramienta de desarrollo Python, para el tratamiento de las imágenes (operaciones de segmentación por ejemplo) se emplea la librería OpenCV la cuál consiste en una librería de libre uso que contiene funciones enfocadas en visión por computadora. Cabe destacar que para cada ejercicio se explica la metodología empleada para la resolución.

II. EJERCICIO 1: DIAGNÓSTICO DE LA CALIDAD DE IMAGEN

- En este ejercicio van a usar ustedes las 3 imágenes denominadas "figura-03-13A – X", las cuales se pueden apreciar en las figuras 1, 2 y 3.

Las tres imágenes muestran un entorno similar, no obstante se nota una importante diferencia en cuanto a la calidad de las imágenes, tanto en brillo como en contraste estas diferencias se van a justificar más a profundidad en la siguiente sección.



Figura 1: Imagen "figura-03-13a".



Figura 2: Imagen "figura-03-13a-v2".

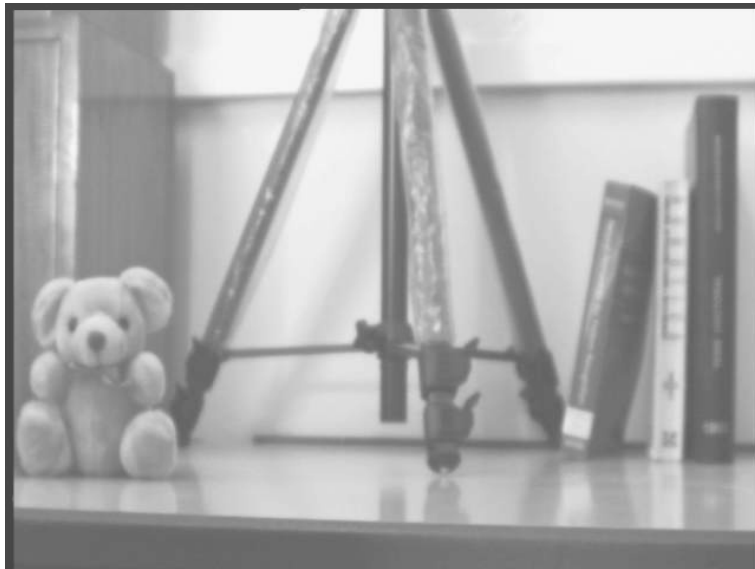


Figura 3: Imagen "figura-03-13a-v3".

- ¿Qué pueden decir sobre la calidad de cada una de ellas? Justifiquen sus respuestas
A simple vista se pueden notar diferencias tanto en brillo como en contraste, con el fin de determinar estas diferencias con un criterio más sólido, se obtienen los histogramas correspondientes a las tres imágenes en cuestión.

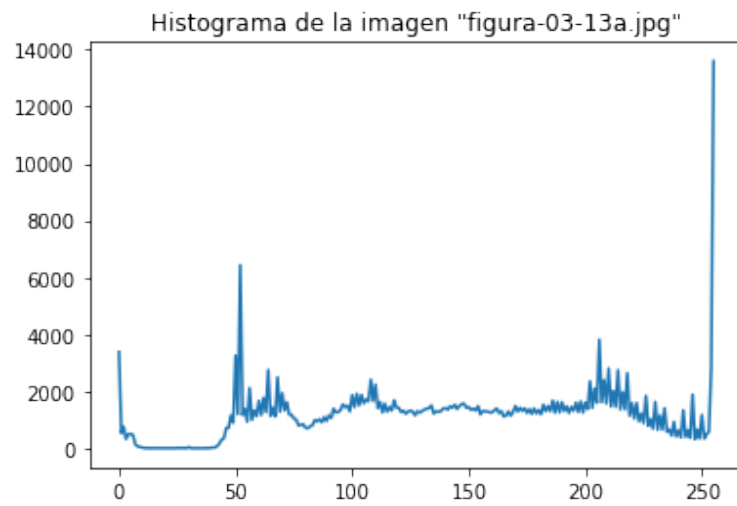


Figura 4: Histograma de la imagen "figura-03-13a".

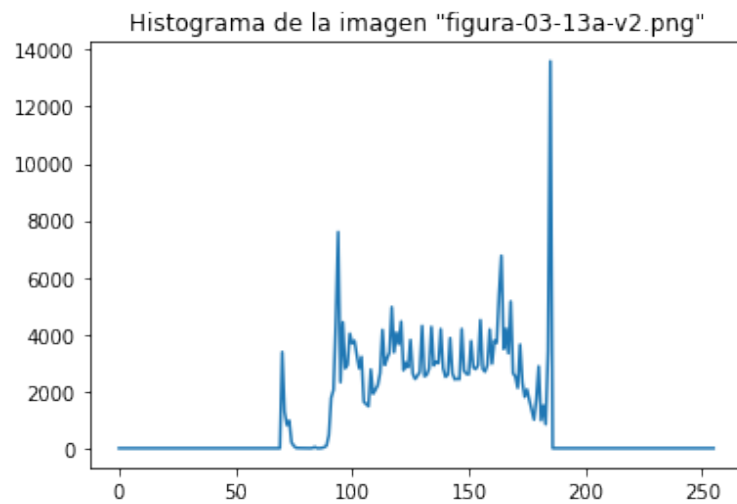


Figura 5: Histograma de la imagen "figura-03-13a-v2".

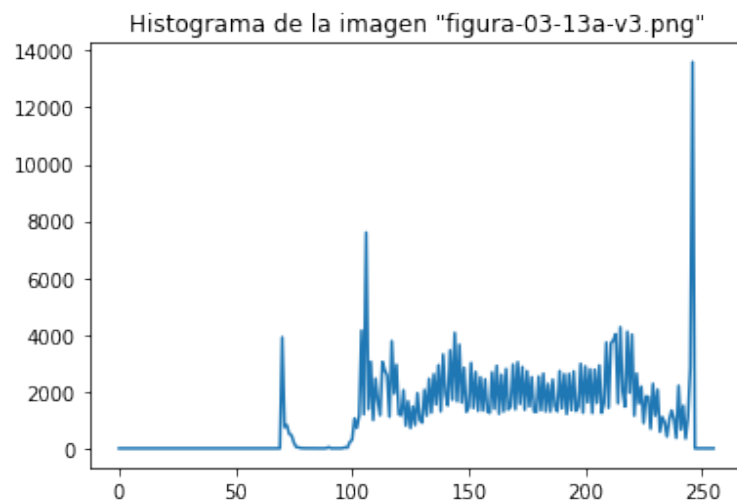


Figura 6: Histograma de la imagen "figura-03-13a-v3".

Al analizar los histogramas, se observa que el histograma de la figura 4 tiene un contraste elevado, no obstante no llega a estar sobre-saturada, en el caso del histograma en la figura 5 se observa una alta saturación en los valores de grises comprendidos en el rango de tonalidades 90 a 190 así como un bajo contraste. Finalmente en el histograma de la figura 6 se aprecia un elevado brillo y alta saturación principalmente en las zonas comprendidas entre las tonalidades de 100 y 250.

- Transformen las imágenes v2 y v3 de manera que se aproximen a la “original”. Una vez transformadas ¿Qué pueden afirmar sobre las diferencias entre ellas? Recuerden que todas sus afirmaciones han de justificarse adecuadamente. Con el fin de transformar las imágenes a la original, se busca el lograr “estirar” y ajustar los histogramas para que estos abarquen un mayor rango de tonalidades y ajustar el contraste para obtener la forma lo más similar posible a la observada en la figura 4

Inicialmente se intentó lograr un efecto similar utilizando la función de ecualizador presente en la librería de cv2 con la figura 2, obteniendo así las figuras 7 y 8. No obstante, como se puede apreciar en la figura 8 el histograma, si bien alarga el rango de tonos presentes en la imagen, la cantidad de píxeles en las diferentes tonalidades no se asemeja a lo visto en el histograma original, por lo que no se tomó como un resultado aceptable y se probó una metodología distinta.



Figura 7: Imagen “figura-03-13a-v2” al aplicar un ecualizador.

Histograma de la imagen “figura-03-13a-v2.png” al aumentar el contraste

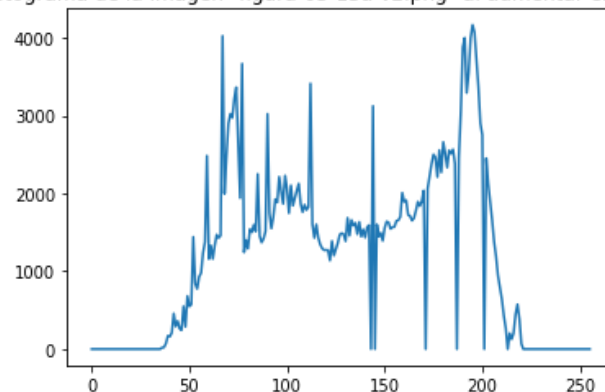


Figura 8: Histograma de la imagen “figura-03-13a-v2” al aplicar un ecualizador.

Se desarrolló la ecuación 1, con la cual se intenta dar forma semejante al histograma de las imágenes (figuras 2 y 3) respecto al de la imagen de referencia (3). ¿Cómo se construyó esta función y en qué consiste? Considerando como base la figura 4, correspondiente al histograma de la imagen base (figura 1) se observa que existe un pico inicial justamente en el valor de intensidad 0. Si se compara este pico con el resto de histogramas, se puede apreciar que presentan un pico inicial, sin embargo, este no se encuentra en el valor de intensidad 0. Por lo tanto, primeramente se decide que la función matemática

debe desplazar ese primer pico al 0 de intensidad en el histograma. Para lograr esto, se decide que se debe realizar una resta a todos los valores de la matriz que representa la imagen por corregir y que el valor del sustraendo debe ser el valor mínimo de intensidad encontrado en dicha matriz.

Posteriormente a dicho desplazamiento, se considera un factor que permita "estirar" el histograma y con ello incrementar el contraste. Para esto se decide emplear un factor normalizado y entero correspondiente a doscientos cincuenta y cinco dividido entre la diferencia entre el valor mínimo de los píxeles en la imagen y el valor máximo presente de los píxeles. Dicho factor entonces ubicará la intensidad de los píxeles en el rango de 0 a 255, contemplando los valores actuales de los píxeles de la imagen.

$$\text{Imagen Corregida} = (\text{Imagen actual} - \text{Mín}(\text{Imagen actual})) \cdot \text{Entero}\left(\frac{255}{\text{Máx}(\text{Imagen actual}) - \text{Mín}(\text{Imagen actual})}\right) \quad (1)$$

Al aplicar la ecuación 1, se obtuvieron las figuras 9 y 11 cuyos histogramas se representan en las figuras 10 y 12. En ambos casos se logra con éxito aumentar el contraste puesto que se denota una mayor variedad de tonalidades presentes en los píxeles, así como un nivel de saturación similar al deseado, en las imágenes se notan pequeñas diferencias sobre todo en las tonalidades de las sombras las cuales no llegan a ser idénticas a la referencia.

Nótese que los histogramas corregidos no presentan una forma completamente continua, esto se debe en parte a la forma original de estos y también al mapeo realizado mediante la función, esta al "estirar" el histograma deja espacios de tonos vacíos pues la naturaleza matemática del factor de ampliación implica que existirá un espacio entre intensidades de los píxeles (por cuestiones de multiplicación y redondeo) en especial si el histograma original ya contemplaba espacios en el eje de intensidad.

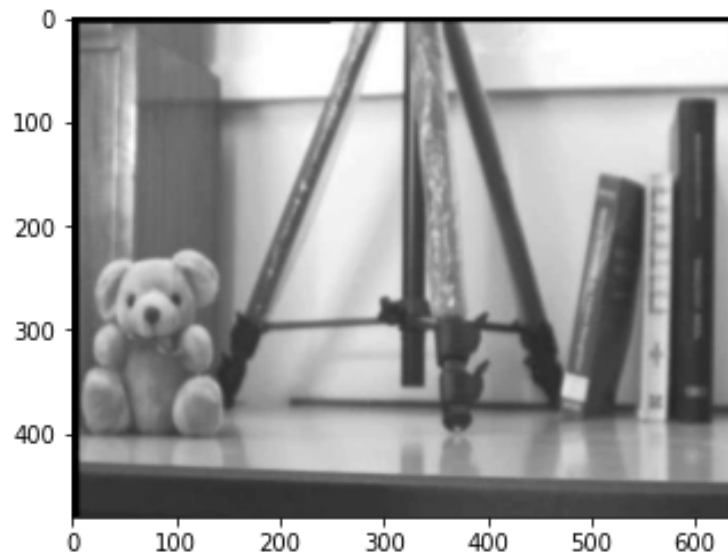


Figura 9: Imagen "figura-03-13a-v2" al aplicar la ecuación 1.

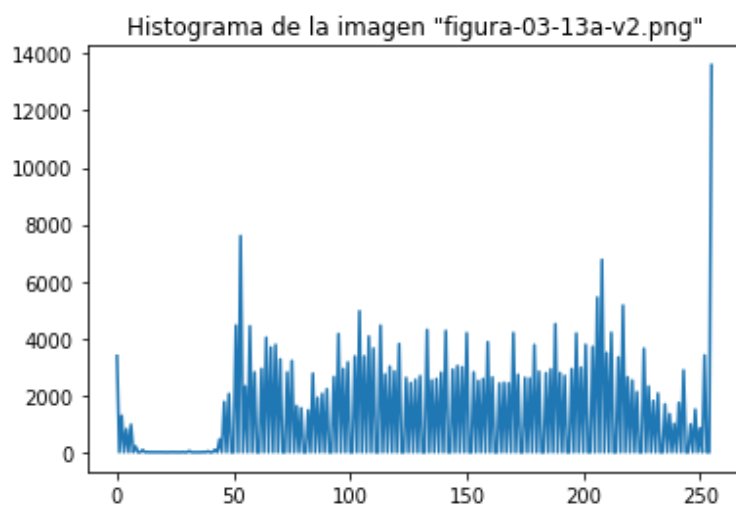


Figura 10: Histograma de la imagen "figura-03-13a-v2" al aplicar la ecuación 1.

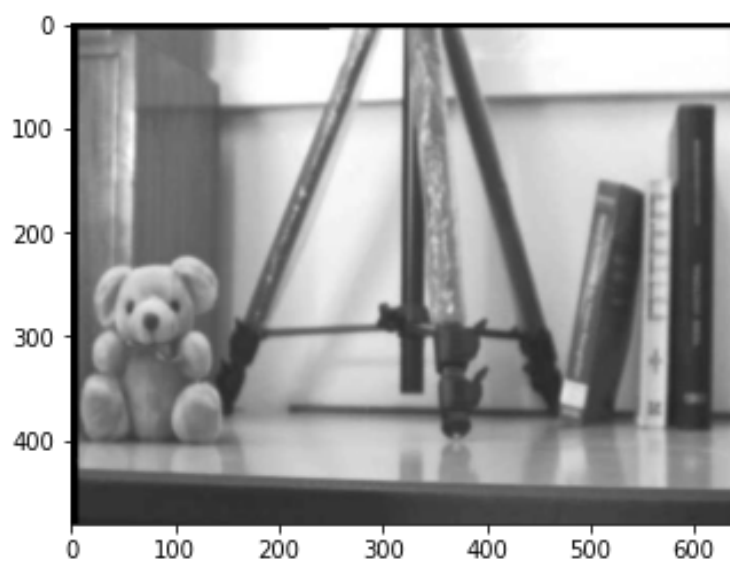


Figura 11: Imagen "figura-03-13a-v3" al aplicar la ecuación 1.

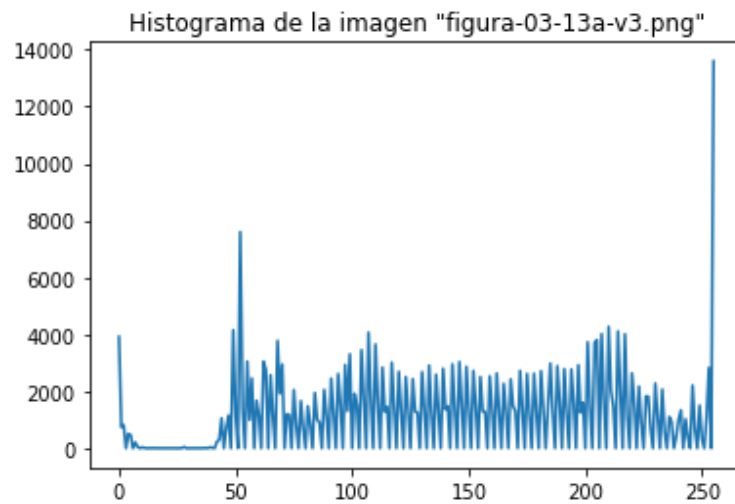


Figura 12: Histograma de la imagen "figura-03-13a-v3" al aplicar la ecuación 1.

III. EJERCICIO 2: BINARIZACIÓN

- En este ejercicio van a usar ustedes la imagen denominada "figura-07-2A, la cual se encuentra en la figura 13.



Figura 13: Imagen "figura-07-02a".

- Se nos dice que queremos binarizar dicha imagen. Justifiquen el análisis y las decisiones. Presenten los resultados obtenidos
El primer paso a seguir para binarizar la imagen en cuestión, es el analizar el histograma correspondiente a la misma, por ende, se obtiene el histograma de la figura 14.

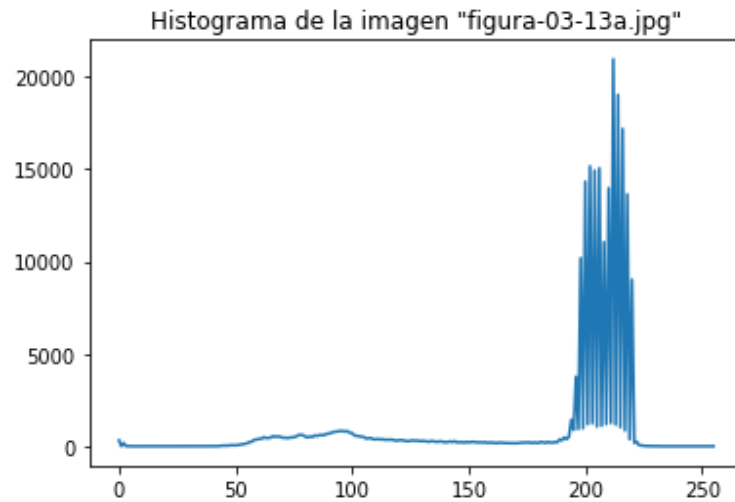


Figura 14: Histograma de la imagen "figura-07-02a".

Como se puede apreciar, se notan picos en las zonas cercanas a las tonalidades entre 50 y 100 y a su vez en las zonas entre 170 y 230. Esta primer elevación se espera que responda a las llaves y monedas presentes en la imagen, en el segundo pico se espera que se componga principalmente por el fondo y los reflejos en las llaves. Dado este análisis, se escoge binarizar utilizando un valor de umbral de 170 como referencia con el fin de separar las monedas y llaves del fondo, obteniendo de esta manera la figura 15. Cabe destacar que no se realizan cambios en el contraste de la imagen puesto que el histograma ya da indicios sobre puntos que pueden utilizarse como umbrales, al existir una división entre el rango dado para objeto y fondo, realmente no se tiene una ganancia al alterar el contraste de la imagen.

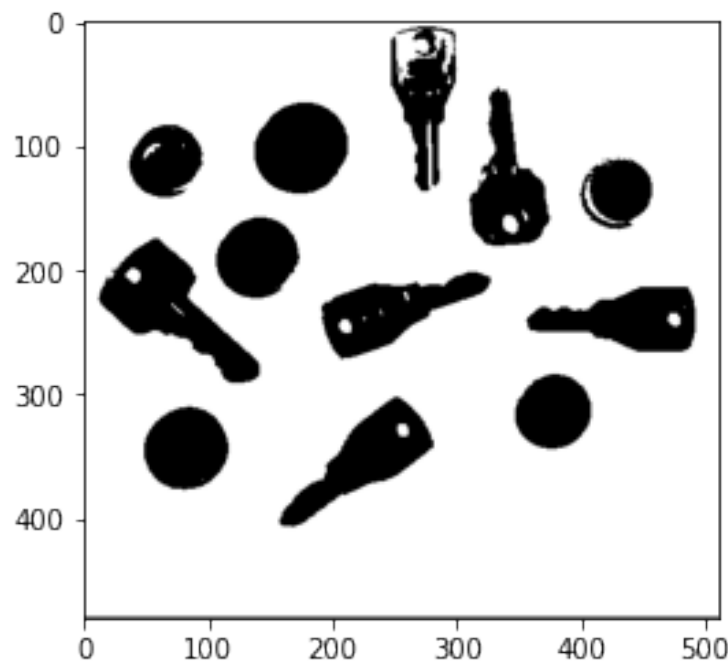


Figura 15: Imagen binarizada "figura-07-02a" empleando un umbral de 170.

$$Imagen\ binarizada(x, y) = \begin{cases} 1 & \text{si } Imagen(x, y) \in R \\ 0 & \text{en caso contrario} \end{cases} \quad (2)$$

Se emplea un método de binarización de la librería OpenCV de Python, el cual utiliza la ecuación 2 (donde R corresponde a la región determinada por el umbral) [2]. Se puede apreciar que la figura 15 presenta efectivamente una diferenciación entre los objetos (llaves y monedas) y el fondo. Cabe destacar que algunos objetos, como por ejemplo la llave que se encuentra en la sección superior central, no quedan completamente negros puesto que sufren del efecto reflectivo, por lo cual el interior cuenta

con "manchas" identificadas como fondo. Pese a esto se aprecia que efectivamente los objetos son identificados. Asimismo, se experimentó con otros valores de umbral para identificar y analizar las diferencias apreciables al aumentar o disminuir dicho valor (se realiza el ejercicio añadiendo/sustrayendo 20 del valor empleado originalmente).

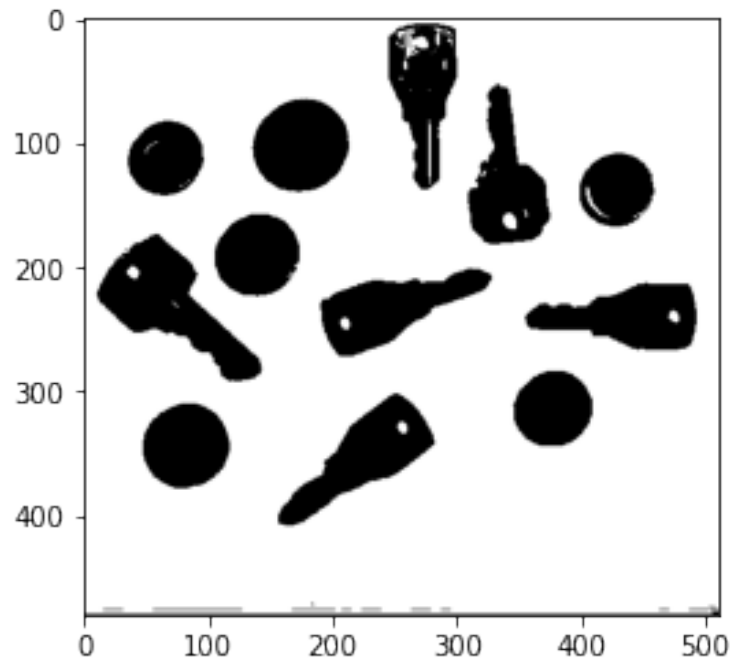


Figura 16: Imagen binarizada "figura-07-02a" empleando un umbral de 190.

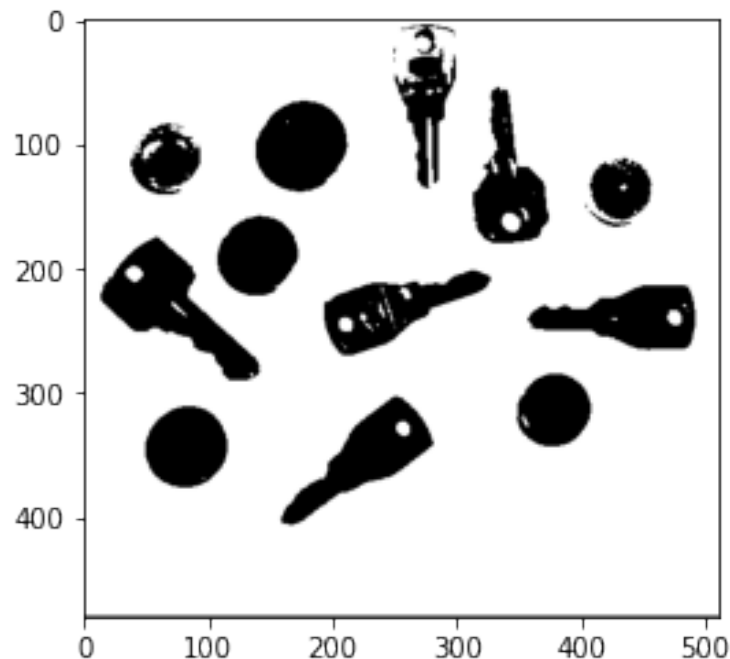


Figura 17: Imagen binarizada "figura-07-02a" empleando un umbral de 150.

Se puede apreciar que la figura 16, la cual emplea un umbral de 190, logra disminuir el efecto de la reflexión, obteniendo objetos más "rellenos" e identificables. Sin embargo, cabe destacar que respecto a la binarización empleando 170 (figura 15), los agujeros de las llaves son ligeramente más pequeños. Por el contrario, la figura presenta peores resultados en cuanto a la binarización entre objeto y fondo, puesto que existen más manchas de fondo en los objetos. Lo anterior tiene sentido puesto que al incrementar el umbral, según el método empleado y la imagen utilizada, se expandiría la frecuencia con la cual se

asigna el valor máximo al resultado contribuyendo a mayor cantidad de negro. Para el caso contrario ocurre al revés, lo cual implica que se incluye dentro del mapeo a 0 aquellos valores correspondientes al brillo en los objetos.

IV. EJERCICIO 3: ANÁLISIS DE SEGMENTACIÓN POR NIVEL

- En este ejercicio van a usar ustedes la imagen denominada “figura-03-19a”

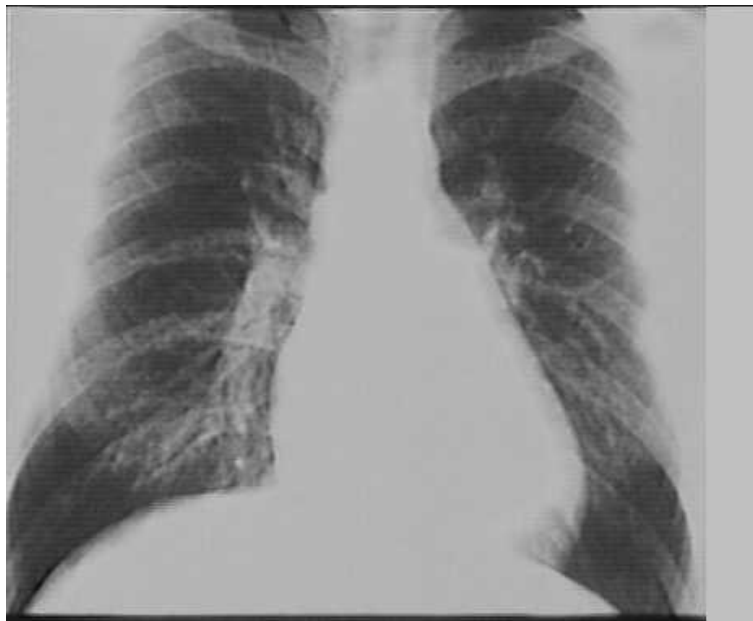


Figura 18: Imagen “figura-03-19a”.

Se puede apreciar que la figura 18 parece corresponder a una radiografía de tórax.

- Se nos dice que queremos segmentar por medio de técnicas basadas en nivel (valor) las costillas ¿podemos hacerlo? Justifiquen el análisis y las decisiones. Presenten los resultados obtenidos.

Para poder dar una respuesta adecuada a la pregunta, primeramente se aplicarán técnicas de segmentación por nivel a la figura 18 para justificar si realmente se puede llegar a dividir la región de las costillas o no mediante técnicas basadas en valor. A continuación se presentarán técnicas basadas en umbralización, considerando pertinentemente el histograma de la imagen (figura 19)

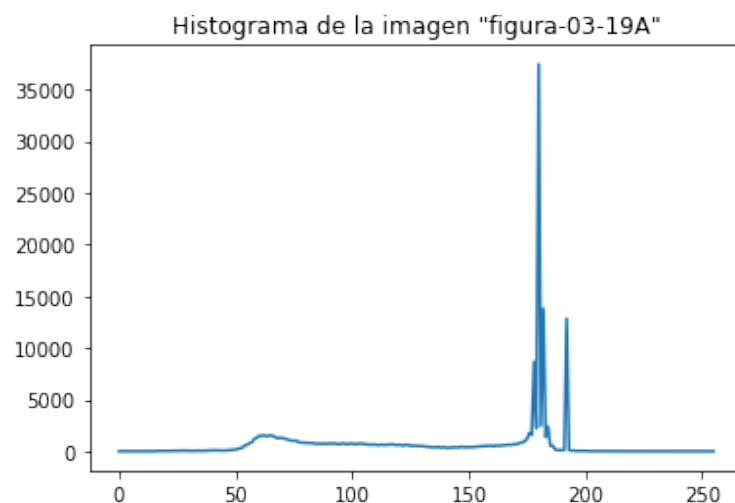


Figura 19: Histograma de “figura-03-19a”.

IV-A. Umbralización de banda

De forma semejante al ejercicio de binarización, se emplea el método descrito por la ecuación 2 para segmentar en dos posibles valores la imagen: 0 o 1, en función de un único valor de umbral. Considerando los picos observados en la

figura alrededor de 60 -los cuales se sugieren que pueden corresponder a los pulmones, puesto que son los elementos más oscuros en la radiografía- y alrededor de 180, se puede definir un umbral tentativo entre 60 y 180. Asimismo existe un tercer pico considerable alrededor de 195. Por tanto se considera realizar la binarización de la imagen con dos umbrales: 150 y 190.

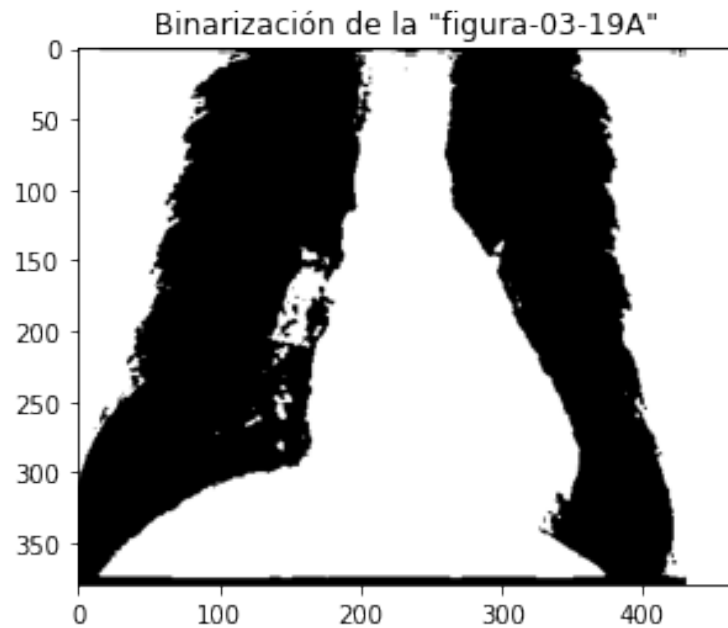


Figura 20: Umbralización de banda con umbral de 150 para "figura-03-19a".

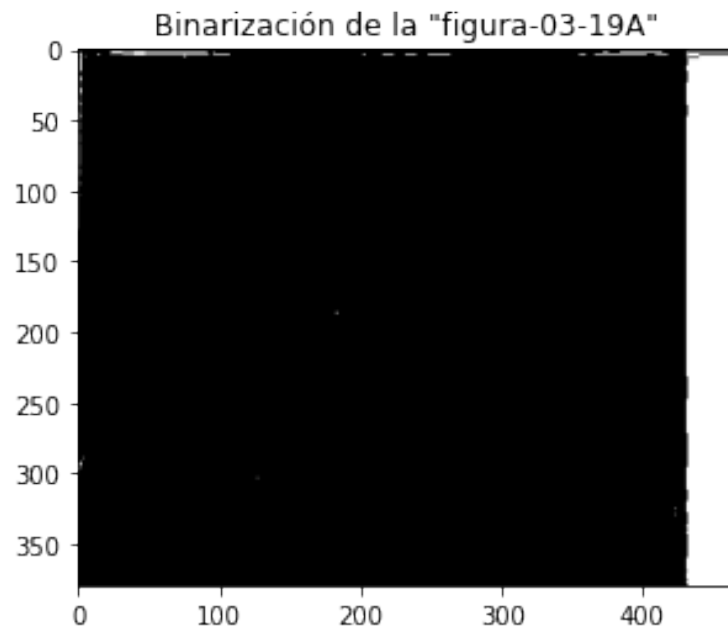


Figura 21: Umbralización de banda con umbral de 190 para "figura-03-19a".

Se puede apreciar que empleando solamente un único umbral resulta complicado obtener una distinción de las costillas. Por tanto este tipo de acercamiento -solamente aplicar un umbral sobre la imagen- no brinda resultados satisfactorios para la segmentación de las costillas. Esto se justifica en parte ya que al utilizar el histograma como medio para definir umbrales para distinguir secciones de una imagen no es del todo preciso, ya que el histograma no brinda especificaciones espaciales referentes a la imagen.

IV-B. Multiumbralización

Para el caso de la multiumbralización, se trataron distintos valores de umbrales, así como la cantidad de estos para intentar segmentar las costillas, la descripción matemática de lo realizado se encuentra en la ecuación 3. El primer intento que se realiza, basándose en la ubicación de pocas cantidades de píxeles en el histograma (preferiblemente "valles"), coloca una cantidad de umbrales igual a 4, con los valores de [5, 70, 180, 190]. El resultado se aprecia en la figura 22.

$$Imagen_s(x, y) = \begin{cases} 1 & \text{si } Imagen(x, y) \in R_1 \\ 2 & \text{si } Imagen(x, y) \in R_2 \\ \vdots & \\ \vdots & \\ \vdots & \\ 0 & \text{en caso contrario} \end{cases} \quad (3)$$

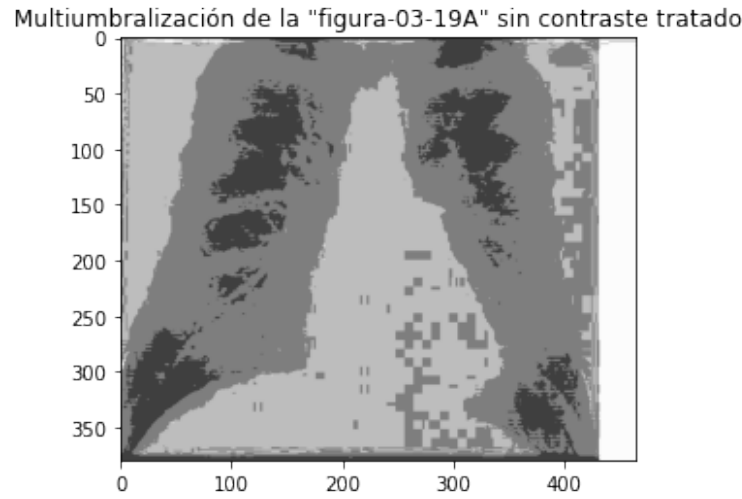


Figura 22: Multiumbralización con los siguiente umbrales:5, 70, 180, 190 para "figura-03-19a".

Como segundo intento se utilizan 4 valores de umbrales, alterándolo los valores empleados ligeramente. El resultado se aprecia en la figura 23.

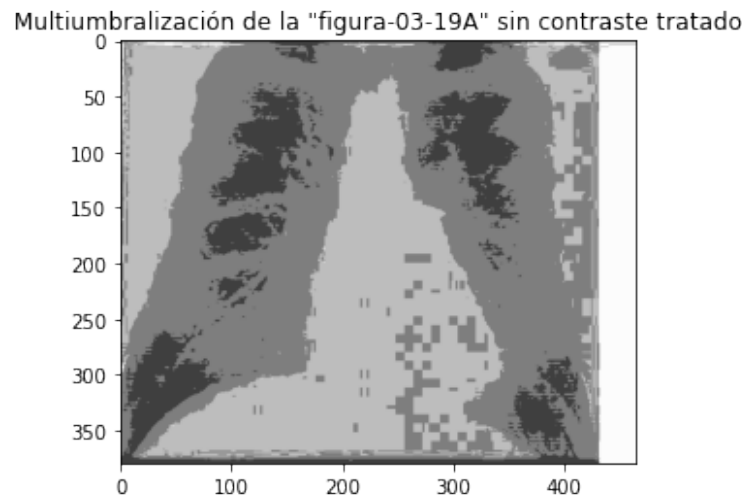


Figura 23: Multiumbralización con los siguiente umbrales:5, 70, 180, 190 para "figura-03-19a".

Asimismo se decidió incluir un tercer objeto de estudio empleando 5 valores de umbrales, tomando como criterio el valor del histograma, el resultado de esta iteración se aprecia en la figura 24.

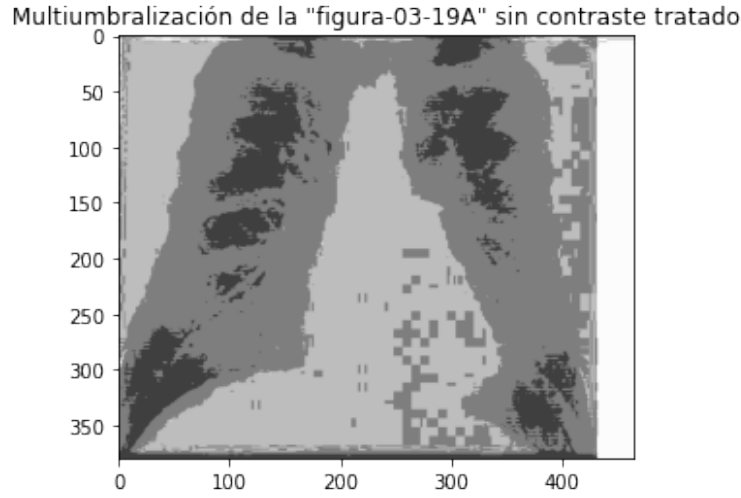


Figura 24: Multiumbralización con los siguiente umbrales:5, 70, 180, 190 para "figura-03-19a".

Se puede apreciar que esta técnica aproxima ligeramente la forma de las costillas, sin embargo para ninguna variante se alcanzan resultados que permitan segmentar por completo a las costillas. Cabe destacar que la región pulmonar queda claramente delimitada, pero las costillas no; una observación que se tiene respecto a estas delimitaciones es que a simple vista, no es posible demarcar por completo las regiones que pertenecen a las costillas. Esto en parte a que la intensidad de los píxeles en las costillas no es homogénea y los niveles de umbralización escogidos no hacen distinción de la ubicación en el espacio, son mapeadores de intensidad y en caso de que otra región que no corresponda a una costilla, presente un nivel de gris semejante, van a ser considerados como lo mismo en el momento de segmentar por niveles.

IV-C. Semiumbralización

Este proceso es semejante al empleado en la umbralización de banda, se rige bajo la ecuación 4, y por tanto bajo la misma lógica se emplean los mismos valores de umbrales. Los resultados se observan en la figura 25 y la figura 26.

$$Imagen_s(x,y) = \begin{cases} Imagen(x,y) & \text{si } Imagen(x,y) \in R \\ 0 & \text{en caso contrario} \end{cases} \quad (4)$$

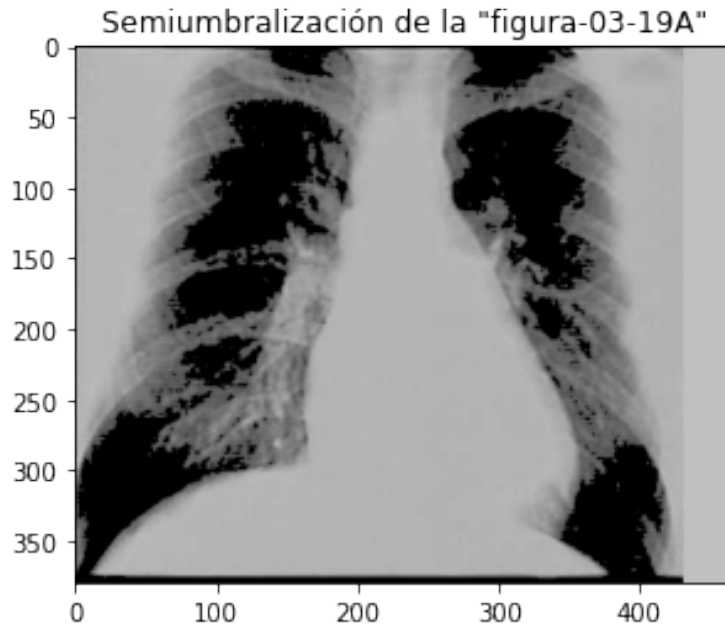


Figura 25: Semiumbralización con un umbral de 150 para "figura-03-19a".

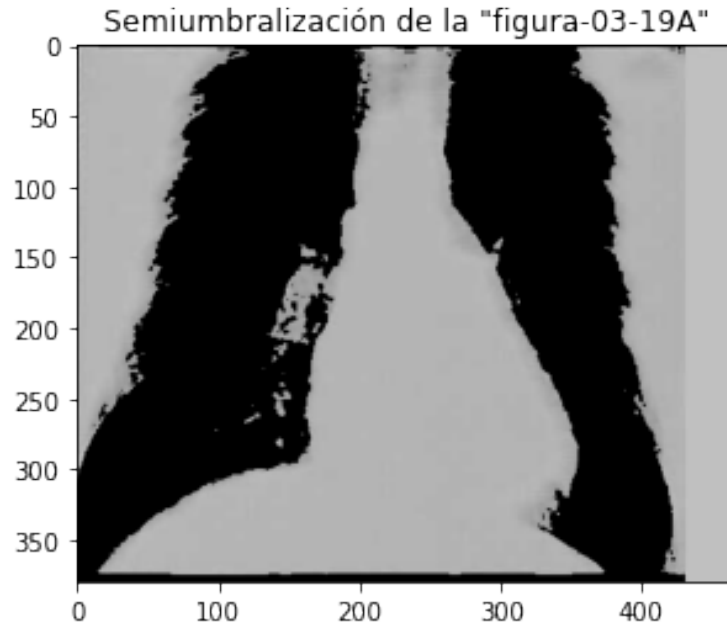


Figura 26: Semiumbralización con un umbral de 80 para "figura-03-19a".

Se aprecia que un solo valor de umbral no incide de forma significativa en la segmentación de las costillas. Si bien es cierto, a diferencia de mapear todos los valores basados en un umbral hacia dos únicos valores (como en umbralización de banda), se retienen los valores de la imagen que cumplan con cierto nivel por encima del umbral, el resultado no determina la región de las costillas. Se puede apreciar en la figura 25 que desde el uso de un umbral de 80 se pierde información referente a las costillas, mientras que para la figura 26 se pierde por completo el rastro de las costillas. En ambos casos se imposibilita aislar las costillas.

IV-D. Umbralización adaptativa

En esta subsección se discuten los resultados al emplear la técnica de umbralización adaptativa, no sin antes definir qué se empleó para la tarea y el por qué de las decisiones tomadas. La biblioteca OpenCV emplea la función descrita en la ecuación 5 para definir la umbralización adaptativa de una imagen original I_{or} , dando como resultado I_{res} . Nótese que $maxValue$ corresponde a un parámetro de entrada.

$$I_{res}(x, y) = \begin{cases} maxValue & \text{si } I_{or}(x, y) \in R \\ 0 & \text{en caso contrario} \end{cases} \quad (5)$$

Donde $T(x, y)$ representa un umbral que se calcula individualmente para cada píxel, según un algoritmo. Para este laboratorio se emplea *ADAPTIVE_THRESH_GAUSSIAN_C* el cual consiste en una suma ponderada (correlación cruzada con una ventana gaussiana) de la vecindad $blockSize \times blockSize$ de (x, y) menos C (esta constante es un parámetro de entrada). Cabe destacar que el $blockSize$ consiste en el tamaño de la vecindad, siendo siempre un número impar.

Los resultados obtenidos se muestran en las figuras 27 y 27. Se puede apreciar que el método adaptativo no logra separar adecuadamente las costillas con los parámetros introducidos, sin embargo se observan los bordes de los elementos de forma más definida. (Sin consistir en un algoritmo para explícitamente describir bordes en una imagen).

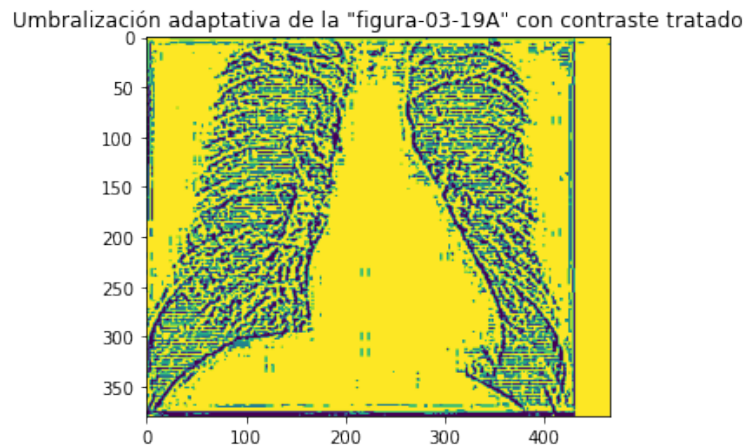


Figura 27: Semiumbralización con un umbral de 80 para "figura-03-19a".

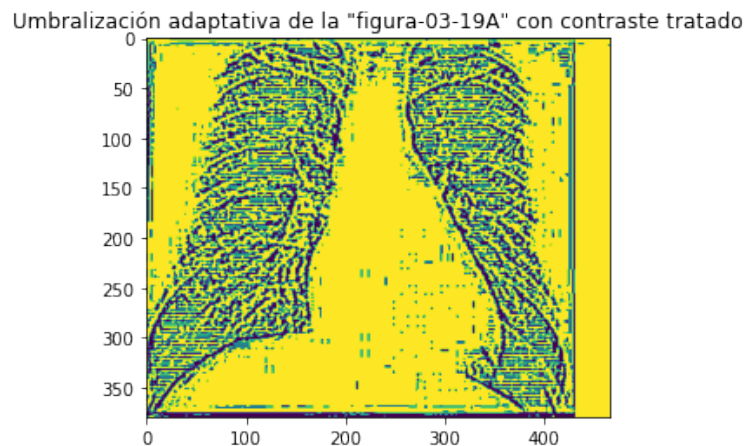


Figura 28: Semiumbralización con un umbral de 80 para "figura-03-19a".

IV-E. Síntesis ejercicio 3

Se puede apreciar que los resultados obtenidos indican que en cierta medida se pueden llegar a segmentar algunos elementos de la imagen como lo serían los pulmones. Sin embargo las costillas suponen un reto al emplear la segmentación por nivel, puesto que los valores de intensidad lumínica en estos elementos es semejante al valor de intensidad en otras regiones. Esto no puede claramente distinguirse en el histograma, puesto a que como se mencionó previamente, no es una herramienta espacial, los píxeles que tienen un mismo valor no necesariamente son vecinos entre sí. Esto complica la tarea de definir umbrales que permitan filtrar el resto de contenido sin deshacerse en el camino de las costillas. Parcialmente podría definirse un área de interés en donde las costillas podrían estar ubicadas, pero estaría lejos de ser una aproximación precisa.

V. EJERCICIO 4: GENERACIÓN DE INSUMOS PARA SEGMENTACIÓN POR CONTENIDO

- En este ejercicio van a usar ustedes las imágenes denominadas "figura-13- 4x"



Figura 29: Imagen "figura-13-04b".



Figura 30: Imagen "figura-13-04c".

- En muchas aplicaciones de visión se necesita realizar acciones previas a las que van a constituir nuestro objetivo
- En este caso, se quieren usar las dos imágenes para realizar una segmentación por movimiento. Para ello, lo primero que debemos hacer es obtener una imagen binaria con los píxeles que serían "útiles" en una segmentación por movimiento (la cual no vamos a hacer)
- Deben ustedes, en primer lugar, plantear una solución para obtener dichos píxeles y, posteriormente, explorar si algún tipo de nivelación por nivel (valor) mejoraría el resultado de dicha operación. Recuerden que se trata de analizar opciones EN BASE A CRITERIO y no quedarse con la "prueba y error" al azar.

La solución planteada para realizar dicha segmentación es el obtener la resta entre ambas imágenes para así detallar el movimiento que se dio entre las dos figuras, puesto que la diferencia entre estas es equivalente al cambio de posición del objeto. Una vez logrado esto, se busca separar el objeto del fondo (en este caso el conejo del fondo).

Con la finalidad de obtener la diferencia entre ambas imágenes y descartar el fondo, se utiliza la función de la librería de opencv llamada `absdiff()` [2], la cual se encarga de restar dos arrays de información y devuelve el resultado en valor absoluto con el fin de evitar valores negativos, al mismo tiempo, antes de realizar dicha sustracción, se transforma el tipo de datos a uno mejor adaptado para realizar la operación sin recurrir a errores. Esta función produce el resultado que se aprecia en la figura 31, en el cual se aprecian las diferencias sin el fondo.

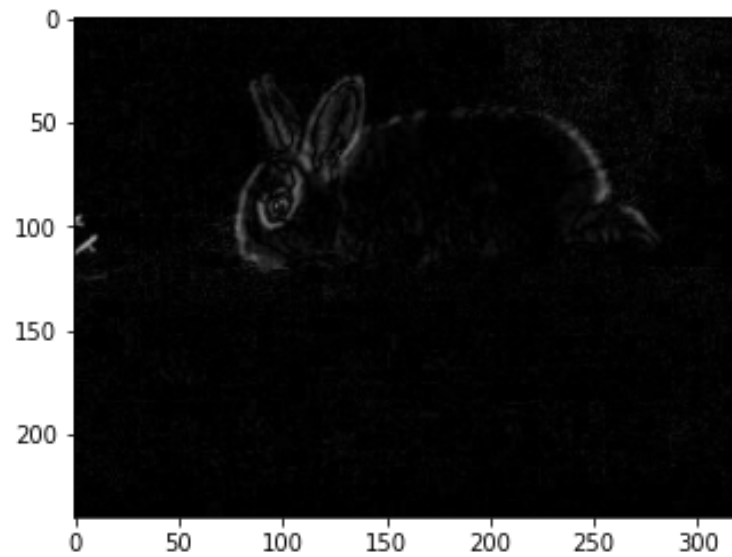


Figura 31: Diferencia entre las imágenes "figura-13-04b" y "figura-13-04b".

El siguiente paso a seguir es el binarizar la imagen obtenida para poder trabajar con esta de forma más eficiente, por ende se obtiene el histograma de la figura 31 el cual se visualiza en la figura 32.

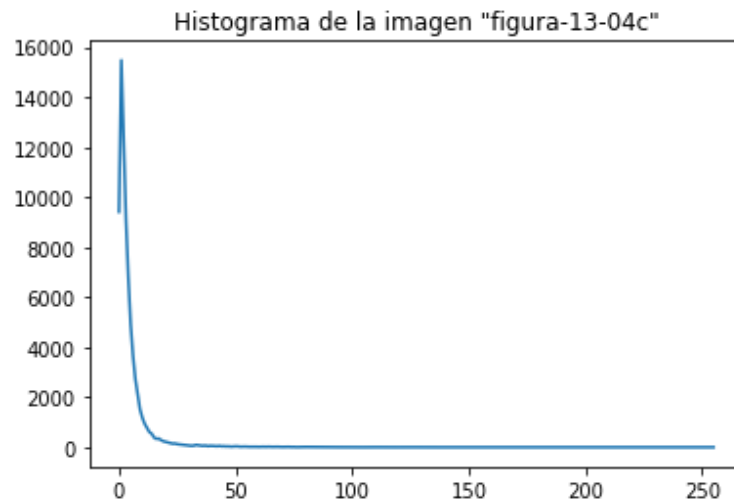


Figura 32: Histograma de la diferencia entre las imágenes "figura-13-04b" y "figura-13-04b".

Para realizar la binarización o bien umbralización de banda, se toma en cuenta el histograma de la figura 32 el cual muestra una importante concentración de tonalidades oscuras, por ende se utiliza un valor de threshold que sea capaz de excluir los píxeles importantes, se probaron dos valores, el de una tonalidad de 25, presente en la figura 33, no obstante se encuentra un ruido similar al ruido sal y pimienta, por lo que se selecciona un valor ligeramente más restrictivo (40), el cual se aprecia en la figura 34

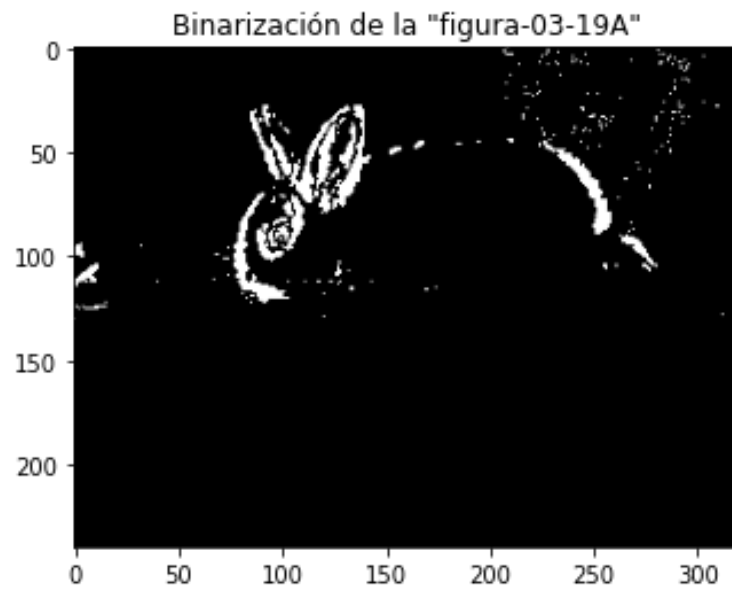


Figura 33: Imagen binarizada con un valor de threshold de 25.

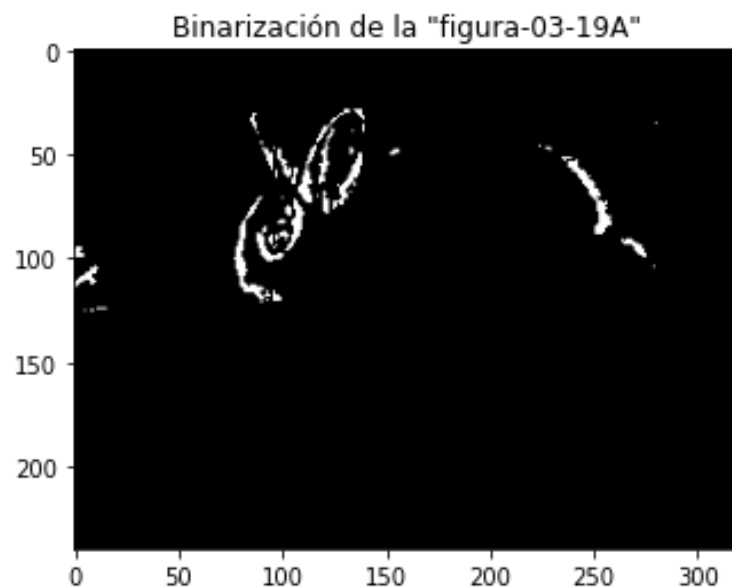


Figura 34: Imagen binarizada con un valor de threshold de 40.

Si bien el segundo caso reduce considerablemente el ruido apreciado en la primer instancia, se pierden varios píxeles que podrían llegar a ser importantes, por lo que se puede intentar corregir este error por medio de la utilización de un filtro mediano o morfológico para evitar esta pérdida de datos.

La figura 34 presenta un contorno de partes importantes del objeto, no obstante para lograr una segmentación por movimiento exitosa, se requiere definir de forma más efectiva. Por ende se procede a realizar una practica distinta con el fin de obtener las partes del contorno faltantes, primeramente se optó por binarizar las imágenes iniciales con un valor de threshold de 150 basándose en el histograma presente en las figuras 35 y 36 (correspondientes a las figuras 13-04b y 13-04c), este valor fue escogido puesto que se aprecian pocos picos alrededor de dicha cifra, considerando que en 75 hay una gran concentración de píxeles (dado que se repite ese valor de intensidad considerablemente en las imágenes) mientras que los picos encontrados más a la derecha pueden representar el blanco del conejo, dado el brillo elevado de este en comparación con el resto de las imágenes.

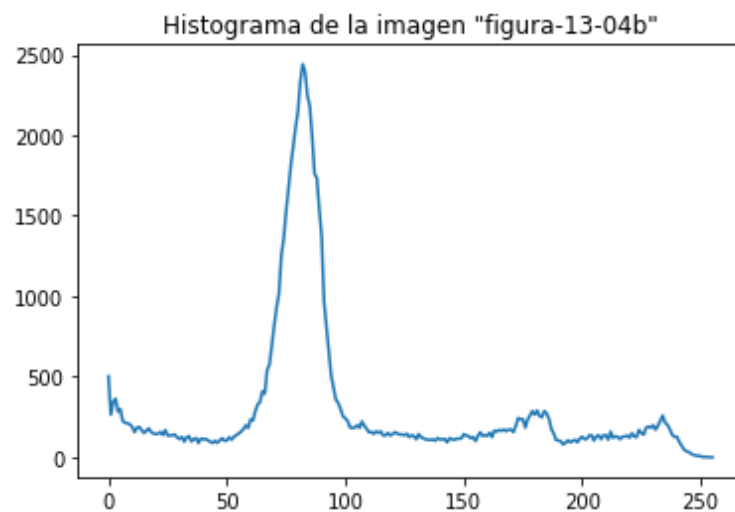


Figura 35: Histograma de la imagen original (luego del movimiento).

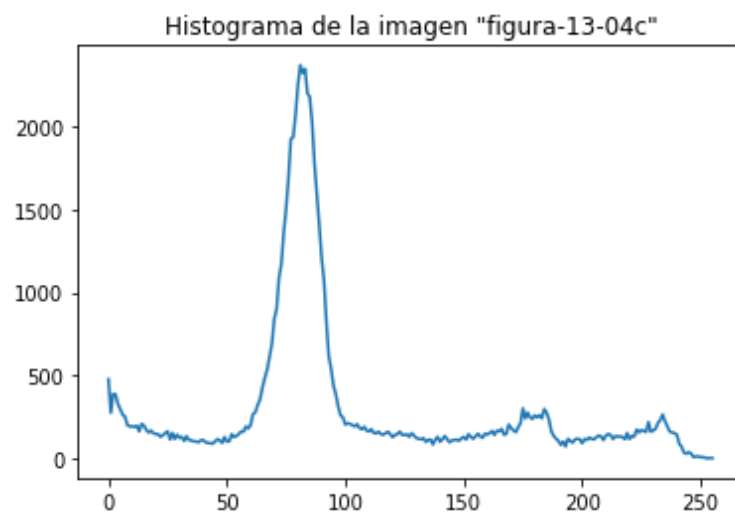


Figura 36: Histograma de la Segunda imagen (luego del movimiento).

El resultado obtenido por medio de la binarización planteada anteriormente se observa en las figuras 37 y 38 en las cuales se resaltan partes del cuerpo del conejo que en secciones anteriores del ejercicio fueron omitidas.

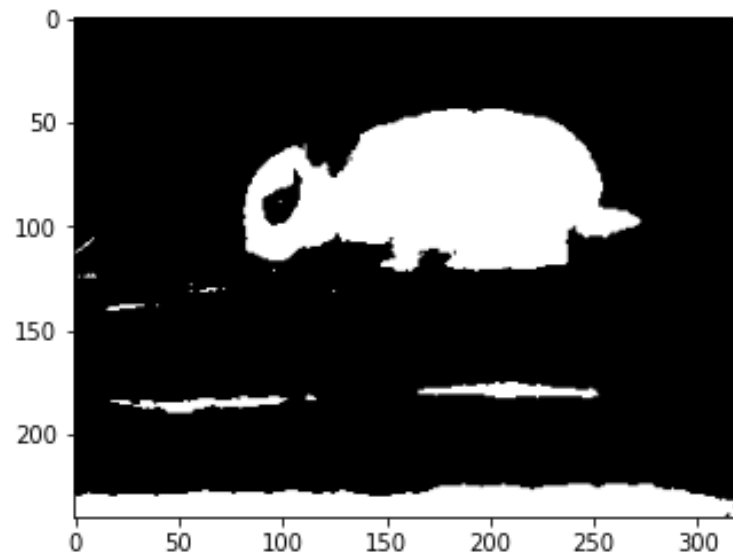


Figura 37: Imagen original binarizada con un valor de threshold de 150.

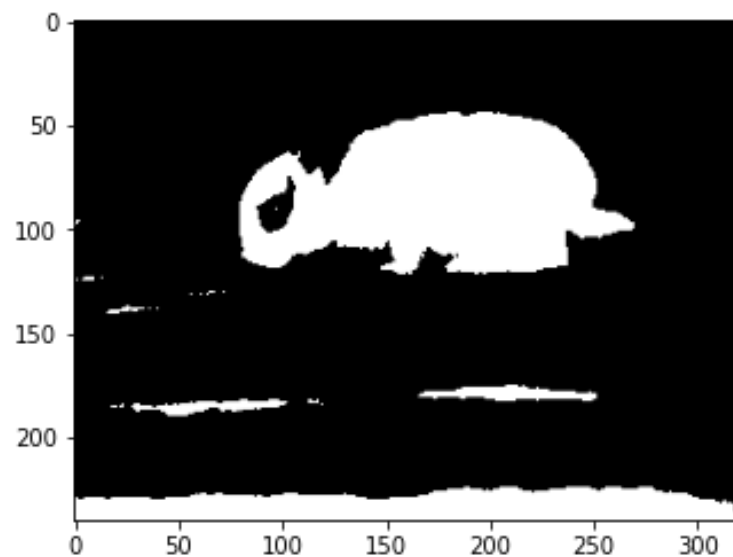


Figura 38: Segunda imagen (luego del movimiento) binarizada con un valor de threshold de 150.

Seguidamente se decide realizar la misma operación de diferencia absoluta entre imágenes, esto con el fin de obtener los contornos que pueden ser de utilidad para la segmentación por movimiento. El resultado de esta diferencia absoluta se aprecia en la figura 39, en la cual se ve un contorno más definido para las patas y la espalda del conejo que en la figura 34, sin embargo, no se aprecian las orejas puesto que en la binarización se suprimieron estos elementos al escoger el umbral de 150, ya que a diferencia del conejo, las orejas tienen menos brillo (son más grises).

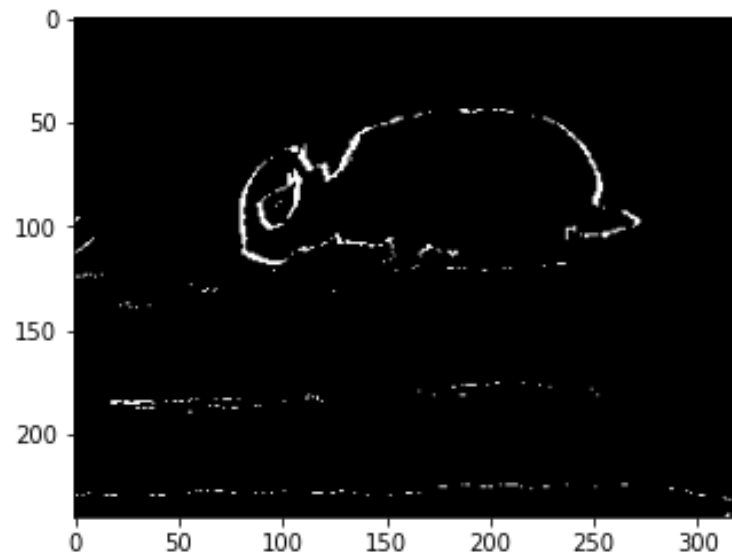


Figura 39: Contorno obtenido con la diferencia entre ambas imágenes binarizadas.

Esta diferencia llega a representar una oportunidad de mejora puesto que las imágenes obtenidas pueden llegar a complementarse entre sí, por tanto, se decide realizar una operación lógica OR entre la figura 39 y la figura 34, con la cual se obtiene la figura 40. Se puede apreciar una mejor definición del contorno del conejo producto del movimiento de este. Aún así se incorpora ruido en la alfombra de la imagen. Se decide añadir una imagen adicional al reporte para observar qué ocurre al colocar "por encima" (empleando un OR) el contorno obtenido con la primera imagen que compone el movimiento, resultando en la figura 41.

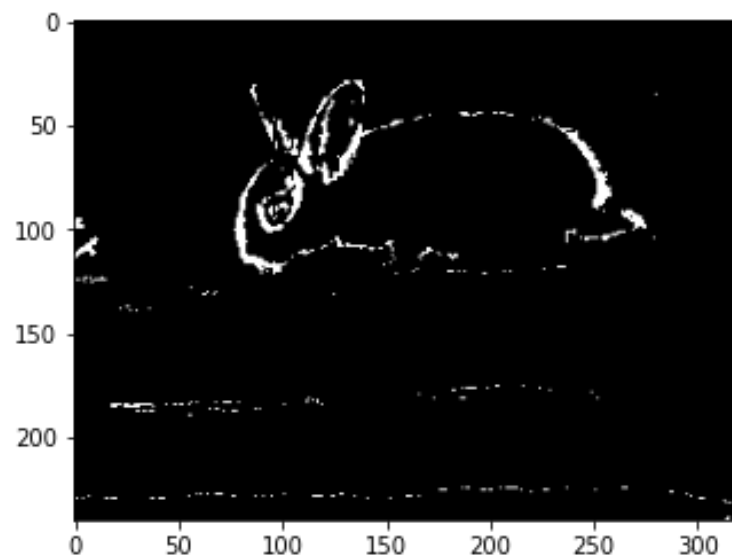


Figura 40: Contorno completo obtenido a través de unir ambos procesos.

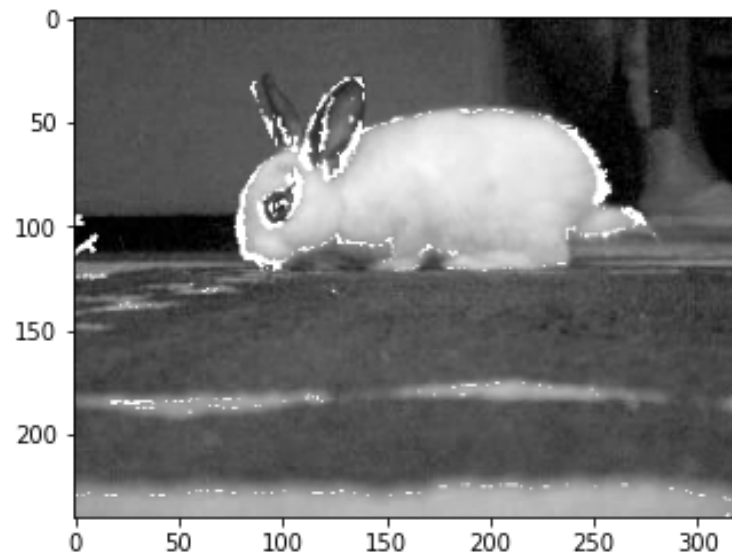


Figura 41: Comparación de la imagen original con el contorno obtenido final.

De este proceso se puede deducir que la binarización ayuda a producir en conjunto con otras técnicas lógicas -aplicadas a las imágenes- un insumo de mayor calidad para la segmentación por movimiento. Es posible comparar este resultado con las imágenes que describen el movimiento y apreciar los elementos en movimiento. Cabe destacar que esta efectividad de la técnica para este caso específico tiene relación con que el brillo de uno de los elementos en movimiento destaca por encima del resto de píxeles que tienen un valor inferior. Asimismo, el uso de la diferencia absoluta entre imágenes también brinda resultados satisfactorios en cuanto a la obtención de insumos para movimiento, pero no precisamente va a englobar por completo un elemento.

Para la presente aplicación no se utiliza otro tipo de nivelación por nivel por múltiples motivos, para el caso de la multiumbralización, no resulta ser una estrategia efectiva porque la diferencia en las posiciones del objetivo se pueden representar de forma efectiva únicamente con un único tono, por lo que esta aplicación no se ve beneficiada del uso de dicha técnica.

Por otra parte la semiumbralización no genera un gran aporte puesto que esta consiste en conservar el valor original en el objeto y tratar únicamente el fondo, no obstante, al apreciar la figura 31 se puede apreciar una clara diferencia entre el fondo y el objeto, por lo que el mantener el tono original en el objeto no genera ganancias para la solución del problema.

La umbralización adaptativa de manera similar al caso de la multiumbralización no genera mayor ventajas para la solución del problema puesto que originalmente se distinguen claramente el objeto del fondo y al generar la diferencia entre las figuras, esta distinción se mantiene, además de que existe un único objeto en la figura, por ende no es necesario la utilización de esta técnica.

VI. CONCLUSIONES

- El histograma es una herramienta útil que permite el análisis de imágenes y el diagnóstico de la calidad de imagen de forma numérica.
- Se pueden cambiar propiedades de una imagen, basándose en el histograma, como por ejemplo el brillo y el contraste dado, al analizar tanto la forma como la distribución de los datos en el mismo.
- La binarización permite destacar uno o varios elementos que cumplan con un umbral de intensidad lumínica. Resulta especialmente útil cuando se tiene un fondo y objetos definidos bajo distintos umbrales. El histograma resulta útil como referencia en estos casos.
- Las técnicas de segmentación por nivel permiten un análisis profundo de imágenes, así como detectar detalles que no se notan a simple vista en la imagen inicial, son especialmente útiles con elementos con contrastes definidos y diferentes entre sí. Si existe bajo contraste y se comparten valores de intensidad entre objetos, puede ser complicado de segmentar.
- Un histograma NO brinda información espacial sobre una imagen, los píxeles con una misma intensidad no necesariamente se encuentran contiguos unos de los otros.
- Las técnicas de segmentación por nivel pueden dar lugar a ruido de diferentes formas en la imagen. Este puede reducirse de distintos métodos como la implementación de filtros.
- Es posible emplear dos imágenes tomadas en tiempos distintos para la generación de insumos para la segmentación por movimiento por medio de la eliminación de fondos y obtención de los contornos del objeto que se desee analizar su movimiento.

REFERENCIAS

- [1] Mark S. Nixon and Alberto S. Aguado, “Feature Extraction Image Processing for Computer Vision”. Academic Press, 2012
- [2] S. Gollapudi, *Learn computer vision using OpenCV*. Berkeley, CA: Apress, 2019.

VII. ANEXOS

```
# -*- coding: utf-8 -*-
"""VisionLab1.ipynb

Automatically generated by Colaboratory.

Original file is located at
    https://colab.research.google.com/drive/1WW4f1EAWs0Shewmvo1DIby6oyKCBO1uC

# Laboratorio 1
### *Sistemas de Visión*
"""

#Se importan las librerías requeridas para el funcionamiento del programa
import cv2
import numpy as np
import matplotlib.pyplot as plt
from google.colab.patches import cv2_imshow

"""# **Ejercicio 1 -----**"""

#Se carga la imagen con la que se va a trabajar inicialmente
img = plt.imread('figura-03-13a.jpg',0)
# Se calcula el histograma al clasificar los píxeles en valores de 0 a 255
histr = cv2.calcHist([img],[0],None,[256],[0,256])
histr_or = histr
# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-03-13a.jpg"')
plt.show()
plt.imshow(img)

#Mismo procedimiento para la segunda imagen
#Se carga la imagen con la que se va a trabajar inicialmente
img = plt.imread('figura-03-13a-v2.png',0)
# Se calcula el histograma al clasificar los píxeles en valores de 0 a 255
histr = cv2.calcHist([img],[0],None,[256],[0,256])

# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-03-13a-v2.png"')
plt.show()
plt.imshow(img)

#Mismo procedimiento para la tercer imagen
#Se carga la imagen con la que se va a trabajar inicialmente
img = plt.imread('figura-03-13a-v3.png',0)
# Se calcula el histograma al clasificar los píxeles en valores de 0 a 255
histr = cv2.calcHist([img],[0],None,[256],[0,256])

# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-03-13a-v3.png"')
```

```

plt.show()
plt.imshow(img)

#Con m todo YUV
img = plt.imread('figura-03-13a-v2.png',0)
# converting to LAB color space
lab= cv2.cvtColor(img, cv2.COLOR_BGR2LAB)
l_channel, a, b = cv2.split(lab)
# Applying CLAHE to L-channel
# feel free to try different values for the limit and grid size:
clahe = cv2.createCLAHE(clipLimit=2.0, tileGridSize=(8,8))
cl = clahe.apply(l_channel)

# merge the CLAHE enhanced L-channel with the a and b channel
limg = cv2.merge((cl,a,b))

# Converting image from LAB Color model to BGR color spcae
enhanced_img = cv2.cvtColor(limg, cv2.COLOR_LAB2BGR)

# Stacking the original image with the enhanced image
result = np.hstack((img, enhanced_img))
cv2.imshow(result)
histr = cv2.calcHist([enhanced_img],[0],None,[256],[0,256])
plt.plot(histr)
plt.title('Histograma de la imagen "figura-03-13a-v2.png" al aumentar el contraste')
plt.show()

#Con ecualizador
img = cv2.imread('figura-03-13a-v2.png', cv2.IMREAD_GRAYSCALE)
image_enhanced = cv2.equalizeHist(img)
plt.imshow(image_enhanced, cmap='gray'), plt.axis("off")
plt.show()
histr = cv2.calcHist([enhanced_img],[0],None,[256],[0,256])
plt.plot(histr)
plt.title('Histograma de la imagen "figura-03-13a-v2.png" al aumentar el contraste')
plt.show()

#Correci n de contraste para figura-03-13a-v2.png
img = plt.imread('figura-03-13a-v2.png',0)

img_correction = ((img-img.min()*(255/(img.max()-img.min()))).astype(np.uint8)
histr2_v2 = cv2.calcHist([img_correction],[0],None,[256],[0,256])
plt.plot(histr2_v2)
plt.title('Histograma de la imagen "figura-03-13a-v2.png"')
plt.show()
plt.imshow(img_correction)

#Correci n de contraste para figura-03-13a-v3.png
img = plt.imread('figura-03-13a-v3.png',0)

img_correction = ((img-img.min()*(255/(img.max()-img.min()))).astype(np.uint8)
histr2_v3 = cv2.calcHist([img_correction],[0],None,[256],[0,256])
plt.plot(histr2_v3)
plt.title('Histograma de la imagen "figura-03-13a-v3.png"')
plt.show()
plt.imshow(img_correction)

```

"""# **Ejercicio 2 -----**"""


```

#Se carga la imagen con la que se va a trabajar inicialmente
img2 = plt.imread('figura-07-02a.jpg',0)
# Se calcula el histograma al clasificar los pixeles en valores de 0 a 255
histr = cv2.calcHist([img2],[0],None,[256],[0,256])

# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-03-13a.jpg"')
plt.show()
plt.imshow(img2)

#Empleando 128 (la mitad de 256)
#CON CV2
th, img2_bin = cv2.threshold(img2, 128, 255, cv2.THRESH_TOZERO)
plt.imshow(img2_bin)

#Empleando el promedio
#CON CV2
th, img2_bin = cv2.threshold(img2, img2.mean(), 255, cv2.THRESH_TOZERO)
plt.imshow(img2_bin)

#Empleando la lógica viendo el histograma
#CON CV2
th, img2_bin = cv2.threshold(img2, 170, 255, cv2.THRESH_TOZERO)
plt.imshow(img2_bin)

#Empleando la lógica viendo el histograma
#CON CV2
th, img2_bin = cv2.threshold(img2, 190, 255, cv2.THRESH_TOZERO)
plt.imshow(img2_bin)

#Empleando la lógica viendo el histograma
#CON CV2
#Explicar por qu
th, img2_bin = cv2.threshold(img2, 170, 255, cv2.THRESH_BINARY)
plt.imshow(img2_bin)

#Empleando la lógica viendo el histograma
#CON CV2
#Explicar por qu
th, img2_bin = cv2.threshold(img2, 190, 255, cv2.THRESH_BINARY)
plt.imshow(img2_bin)

#Empleando la lógica viendo el histograma
#CON CV2
#Explicar por qu
th, img2_bin = cv2.threshold(img2, 150, 255, cv2.THRESH_BINARY)
plt.imshow(img2_bin)

#Notese que no se cambi el contraste, puesto que la binarizaci n
#es posible sin esta, ya que imagen solo trata fondo y objeto (llaves)

"""# **Ejercicio** **3** **-----**"""

#Se carga la imagen con la que se va a trabajar inicialmente
img3 = plt.imread('figura-03-19a.jpg',cv2.IMREAD_GRAYSCALE)
# Se calcula el histograma al clasificar los pixeles en valores de 0 a 255

```

```

histr = cv2.calcHist([img3],[0],None,[256],[0,256])

# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-03-19A"')
plt.show()
plt.imshow(img3)

#Se trata la imagen primero, para aumentar el rango dinámico
# Qu se pretende? Para obtener mayor cantidad de intensidades y con ello
#se pretende poder expandir posibilidad de segmentación
#Corrección de contraste para figura-03-19A

img_correction_3 = ((img3-img3.min()*(255/(img3.max()-img3.min()))).astype(np.uint8))
histr3 = cv2.calcHist([img_correction_3],[0],None,[256],[0,256])
plt.plot(histr3)
plt.title('Histograma de la imagen "figura-03-19A"')
plt.show()
plt.imshow(img_correction_3)

#Resultado: no se obtienen mejoras significativas en el contraste
#Puesto que no se logra diferenciar algo significativamente distinto
#de la imagen sin tratar.

#No parece oportuna su utilización para binarización

#Umbralización fija: de banda, con imagen tratada
th, img3_bin = cv2.threshold(img3, 150, 255, cv2.THRESH_BINARY)
plt.imshow(img3_bin)
plt.title('Binarización de la "figura-03-19A"')
#Conclusión: el resultado no mejora significativamente empleando mismo umbral

#Umbralización fija: de banda, con imagen tratada
th, img3_bin = cv2.threshold(img3, 190, 255, cv2.THRESH_BINARY)
plt.imshow(img3_bin)
plt.title('Binarización de la "figura-03-19A"')
#Conclusión: el resultado no mejora significativamente empleando mismo umbral

#Umbralización fija: Multiumbralización (sigue siendo manual)
#k regiones
#T límites: lista de límites [] (k-1)
T = [5, 70, 180, 190]
out = np.digitize(img3, bins = T)*int(255/len(T))
plt.imshow(out)
plt.title('Multiumbralización de la "figura-03-19A" sin contraste tratado')
#Se pueden tratar distintos umbrales para alcanzar a ver distintos elementos
#se requiere analizar picos del histograma para una consideración Gaussiana
#PERO recordar que histograma no da pistas espaciales, nese el dilema
#de los huesos.

#Umbralización fija: Multiumbralización (sigue siendo manual)
#k regiones
#T límites: lista de límites [] (k-1)
T = [5, 90, 180, 190]
out = np.digitize(img3, bins = T)*int(255/len(T))
plt.imshow(out)
plt.title('Multiumbralización de la "figura-03-19A" sin contraste tratado')
#Se pueden tratar distintos umbrales para alcanzar a ver distintos elementos

```

*#se requiere analizar picos del histograma para una consideración Gaussiana
#PERO recordar que histograma no da pistas espaciales, n tese el dilema
#de los huesos.*

*#Umbralización fija: Multiumbralización (sigue siendo manual)
#k regiones
#T límites: lista de límites (k-1)
T = [5, 70, 90, 180, 190]
out = np.digitize(img3, bins = T)*int(255/len(T))
plt.imshow(out)
plt.title('Multiumbralización de la "figura-03-19A" sin contraste tratado')
#Se pueden tratar distintos umbrales para alcanzar a ver distintos elementos
#se requiere analizar picos del histograma para una consideración Gaussiana
#PERO recordar que histograma no da pistas espaciales, n tese el dilema
#de los huesos.*

*#Umbralización Fija: Semiumbralización
#Empleando la lógica viendo el histograma
#CON CV2
#BINARIZACIÓN en 150, observando histograma
th, img3_bin = cv2.threshold(img3, 150, 255, cv2.THRESH_TOZERO)
plt.imshow(img3_bin)
plt.title('Semiumbralización de la "figura-03-19A"')
#Conclusión: se puede llegar a identificar una parte de los pulmones
#pero no todo, puesto que el corazón obstruye parte del pulmón
#asimismo, estructura ósea no se puede ver.*

*#Umbralización fija: Semiumbralización con imagen tratada
th, img3_bin = cv2.threshold(img3, 80, 255, cv2.THRESH_TOZERO)
plt.imshow(img3_bin)
plt.title('Semiumbralización de la "figura-03-19A"')
#Conclusión: el resultado no mejora significativamente empleando mismo umbral*

*#Umbralización adaptativa: sin contraste modificado
gray = cv2.cvtColor(img3, cv2.COLOR_BGR2GRAY)
img3_adapt = cv2.adaptiveThreshold(gray, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY)
plt.imshow(img3_adapt)
plt.title('Umbralización adaptativa de la "figura-03-19A" con contraste tratado')*

*#Umbralización adaptativa: CON contraste modificado
gray_mod = cv2.cvtColor(img_correction_3, cv2.COLOR_BGR2GRAY)
img3_adapt = cv2.adaptiveThreshold(gray_mod, 255, cv2.ADAPTIVE_THRESH_GAUSSIAN_C, cv2.THRESH_BINARY)
plt.imshow(img3_adapt)
plt.title('Umbralización adaptativa de la "figura-03-19A" con contraste tratado')*

*"""# **Ejercicio 4 -----**"""*

*#
#Se carga la imagen con la que se va a trabajar inicialmente
img4_1 = plt.imread('figura-13-04b.jpg', cv2.IMREAD_GRAYSCALE)
Se calcula el histograma al clasificar los píxeles en valores de 0 a 255
histr = cv2.calcHist([img4_1], [0], None, [256], [0, 256])*

*# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-13-04b"')
plt.show()
plt.imshow(img4_1)*

```

#
#Se carga la imagen con la que se va a trabajar inicialmente
img4_2 = plt.imread('figura-13-04c.jpg',cv2.IMREAD_GRAYSCALE)
# Se calcula el histograma al clasificar los pixeles en valores de 0 a 255
histr = cv2.calcHist([img4_2],[0],None,[256],[0,256])

# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-13-04c"')
plt.show()
plt.imshow(img4_2)

#Se determina la diferencia para conocer el movimiento
#del conejo
resta_img4 = cv2.absdiff(img4_1, img4_2)
histr = cv2.calcHist([resta_img4],[0],None,[256],[0,256])

# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-13-04c"')
plt.show()
plt.imshow(resta_img4)

th, bin_resta_img4 = cv2.threshold(resta_img4, 40, 255, cv2.THRESH_BINARY)
bin_resta_primera = bin_resta_img4
plt.imshow(bin_resta_img4)
plt.title('Binarización de la "figura-03-19A"')
#Conclusión: el resultado no mejora significativamente empleando mismo umbral

th, bin_resta_img4 = cv2.threshold(resta_img4, 25, 255, cv2.THRESH_BINARY)
bin_resta_primera = bin_resta_img4
plt.imshow(bin_resta_img4)
plt.title('Binarización de la "figura-03-19A"')
#Conclusión: el resultado no mejora significativamente empleando mismo umbral

th, bin_resta_img4 = cv2.threshold(resta_img4, 40, 255, cv2.THRESH_BINARY_INV)
plt.imshow(bin_resta_img4)
plt.title('Binarización de la "figura-03-19A"')
#Conclusión: el resultado no mejora significativamente empleando mismo umbral

#Extracción del conejo
img4_restainvert = cv2.bitwise_not(resta_img4)
plt.imshow(img4_restainvert)
plt.title('Inversión de la "figura-03-19A" con contraste tratado')

#primero binarizar
th,img4_1_bin = cv2.threshold(img4_1, 150, 255, cv2.THRESH_BINARY)
plt.imshow(img4_1_bin)

th,img4_2_bin = cv2.threshold(img4_2, 150, 255, cv2.THRESH_BINARY)
plt.imshow(img4_2_bin)

#Se determina la diferencia para conocer el movimiento
#del conejo
resta_img4 = cv2.absdiff(img4_1_bin, img4_2_bin)
histr = cv2.calcHist([resta_img4],[0],None,[256],[0,256])

```

```
# Grafica el histograma generado anteriormente
plt.plot(histr)
plt.title('Histograma de la imagen "figura-13-04c"')
plt.show()
plt.imshow(resta_img4)

#Extracción del conejo
img4_resta_invert = cv2.bitwise_not(resta_img4)
plt.imshow(img4_resta_invert)

bordes = cv2.bitwise_or(resta_img4, bin_resta_primera)
plt.imshow(bordes)

plt.imshow(cv2.bitwise_and(img4_1, cv2.bitwise_not(bordes)))

plt.imshow(cv2.bitwise_and(img4_1, (bordes)))

plt.imshow(cv2.bitwise_or(img4_1, cv2.bitwise_not(bordes)))

plt.imshow(cv2.bitwise_or(img4_1, (bordes)))
```