

Mejoramiento del contraste para imágenes en escala de grises y en color mediante la ecualización del histograma

J. Orellana¹, R.D. Aponte²

1. Introducción

El procesamiento de imágenes digitales permite mejorar la calidad de una imagen o resaltar características particulares de ésta, dicho procesamiento puede ser útil posteriormente para extraer información de la imagen mediante técnicas de aprendizaje automático.

En general, si $f(x,y)$ representa una imagen de entrada en la que (x,y) es la posición de un píxel, un operador T será capaz de realizar una transformación sobre la imagen, la ecuación (1) representa matemáticamente dicha transformación:

$$g(x,y) = T[f(x,y)] \quad (1)$$

Donde $g(x,y)$ es la imagen de salida.

Las técnicas de procesamiento abarcan operaciones elementales, en las que la transformación se realiza píxel a píxel, y operaciones de filtrado espacial, donde la transformación involucra la vecindad de un píxel en particular.

En este trabajo se ha implementado la técnica de ecualización del histograma la cual es una transformación píxel a píxel cuyo objetivo es distribuir de manera uniforme los niveles de intensidad y las frecuencias a las que aparecen en los píxeles.

Para lograr lo anterior, se ha tomado de referencia la clase *exposure* de la librería *skimage* por *Python*, en particular, el método *equalize_hist()*.

La documentación de dicha librería indica que el método *equalize_hist()* mejora una imagen con bajo contraste propagando las intensidades de mayor frecuencia a lo largo de todo el histograma. La ventaja de este método es que no requiere ningún parámetro de entrada más que las operaciones realizadas sobre cada píxel, pero como desventaja puede generar imágenes de salida con tonos de color no realistas.

(Gonzalez & Woods, 2018) muestran los pasos a seguir para implementar la ecualización de histograma para imágenes en escala de grises y propone como debería ser su implementación en imágenes a color. El contexto teórico y la implementación en se presenta a continuación.

2. Descripción del problema

Se pretende mejorar el contraste de una imagen para lograr una distribución más uniforme de los niveles de intensidad de imagen, es decir, se busca corregir el rango dinámico. Para ello, se utilizará la técnica de ecualización del histograma sobre la imagen entera, usando para ello operaciones elementales sobre los píxeles. El fundamento matemático de la ecualización del histograma se presenta a continuación.

La intensidad, r de una imagen, inicialmente continua, puede ser descrita por:

$$r = 0, 1, 2 \dots L - 1$$

Donde L es el número de niveles de intensidad posibles de la imagen en los que 0 representa negro y $L - 1$ representa blanco.

Las intensidades de cada píxel de la imagen pueden ser vistas como una distribución aleatoria de valores en el intervalo $[0, L - 1]$, es posible redistribuir las probabilidades de ocurrencia de

¹ joel.orellana102@comunidadunir.net

² rubendario.aponte552@comunidadunir.net

cada intensidad de manera uniforme en el histograma de forma que el histograma se asemeje a una distribución constante, la fórmula que permite calcular la distribución acumulada de cada píxel multiplicada por la cantidad de valores posibles es:

$$s = T(r) = (L - 1) \int_0^r p_r(w) dw \quad (2)$$

Donde $T(r)$ representa la transformación o escalamiento de los pixeles originales a una redistribución más homogénea.

La integral anterior puede simplificarse considerando que para una imagen de n bits de intensidad, la probabilidad p_r que un píxel sea de una intensidad determinada viene dada por:

$$p_r(r_k) = \frac{n_k}{MN} \quad (3)$$

Donde MN es el producto de las dimensiones de la imagen y n_k es el número de pixeles que tienen la intensidad r_k .

La ecuación (3) toma a consideración que una imagen es una señal discreta de intensidades de píxel. Esta discretización es una aproximación muy buena de la imagen, pero puede introducir ciertos errores en comparación con el histograma ideal de una señal continua.

Se puede usar el conocimiento anterior para transformar la integral en una sumatoria discreta:

$$s = T(r) = (L - 1) \sum_{j=0}^k p_k(r_k) \quad (4)$$

Esta fórmula es una transformación importante conocida como **ecualización del histograma**.

Una imagen en escala de grises con una profundidad de píxel de 8 bits puede tener hasta un total de $L - 1 = 2^8 - 1 = 255$ niveles de intensidad, por lo que la fórmula anterior queda simplificada como:

$$s = T(r) = 255 \sum_{j=0}^k p_k(r_k) \quad (5)$$

En el caso de imágenes a color, (Gonzalez & Woods, 2018) propone la conversión de la imagen de RGB a HSI y la aplicación de la técnica de ecualización de histograma solo para la señal de luminancia (I). La conversión de RGB a HSI puede implementarse mediante las fórmulas (con R, G y B normalizados):

$$H = \begin{cases} \theta & \text{si } B \leq G \\ 360 - \theta & \text{si } B > G \end{cases} \text{ con } \theta = \cos^{-1} \left(\frac{\frac{1}{2} [(R - G) + (R + G)]}{[(R - G)^2 + (R - B)(G - B)]^{\frac{1}{2}}} \right)$$

$$S = 1 - \frac{3}{R + G + B} [\min(R, G, B)]$$

$$I = \frac{1}{3} (R + G + B) \quad (6)$$

También es usual ver la implementación de conversión hacia HSV. De igual manera, para conversión de RGB a HSV, donde (V) se denomina Valor (Brillo), se puede implementar (con R, G y B normalizados):

$$H = \begin{cases} 0 & \text{para } \Delta = 0 \\ 60 \left(\frac{G - B}{\Delta} \right) \bmod 6 & \text{para } C_{max} = R \\ 60 \left(\frac{B - R}{\Delta} + 2 \right) & \text{para } C_{max} = G \\ 60 \left(\frac{G - B}{\Delta} + 4 \right) & \text{para } C_{max} = B \end{cases} \quad (7)$$

$$S = \begin{cases} 0 & \text{para } C_{max} = 0 \\ \frac{\Delta}{C_{max}} & \text{para } C_{max} \neq 0 \end{cases}$$

$$V = C_{max}$$

$$C_{max} = \max(R, G, B); C_{min} = \min(R, G, B); \Delta = C_{max} - C_{min}$$

La implementación inversa de la conversión HSI y HSV se puede encontrar en lo reportado por (Kong & Ibrahim, 2008).

3. Solución propuesta

La solución propuesta para el mejoramiento del contraste de imágenes en escala de grises y a color se implementó en *Python* usando únicamente las librerías *matplotlib*, *numpy*, *pandas* y *PIL*, de manera que, la implementación fue una aplicación directa de la formulación presentada anteriormente.

El proceso inicia con la importación de la imagen RGB por medio del método *open()* de *PIL*; si la imagen se quiere procesar a escala de grises, se convierte a escala de grises con el método creado *image_rgb_to_gs()* y en caso de que se desee la imagen a color, se convierte de RGB a HSV o HSI pixel a pixel por medio de los métodos creados *rgb_to_hsv()* o *rgb_to_hsi()* y *image_rgb_to_hs()*. Seguido, se debe extraer el canal “V” o “I” de la imagen HSV o HSI (componente de brillo o luminancia) por medio de *np.array()* de *numpy*. Teniendo solo el canal a escala de grises o el canal “V” en imagen a color “V” o “I”, se procede a convertir en arreglo la intensidad de cada pixel a través del método creado *image_to_arrays()*, el cual, usa el método *np.array()* de *numpy* y devuelve un arreglo con los valores únicos de intensidad de cada pixel r_k y la frecuencia o conteo de las veces que se repite una intensidad n_k . Con estos datos, se utiliza el método creado *hist_eq_transform()* en el cual se crea un *dataframe* con la librería *pandas* para calcular la función de distribución acumulativa (CDF, por sus siglas en inglés) y posteriormente obtener las nuevas intensidades de cada píxel s_k para reemplazar los valores originales r_k . Si la imagen procesada era a color, se utiliza el método creado *hsv_to_rgb()* o *hsi_to_rgb()* para convertir la imagen nuevamente a RGB. Finalmente, se guardan las imágenes ecualizadas en *.jpg con el método creado *save_transform_image()* el cual usa los métodos *fromarray()* y *save()* de la librería *PIL*. Para comparar los resultados se usa el método creado *show_image()* que muestra las imágenes originales y ecualizadas y sus respectivos histogramas mediante los métodos *bar()* de la librería *matplotlib* usando los datos del *dataframe*. Este proceso se puede ver en la Figura 1. Se puede observar que el uso de métodos o librerías para el proceso de transformación se limitó a usar librerías preestablecidas solo para lectura inicial de imágenes y guardado de imágenes finales, mientras que en el proceso de transformación solo se usaron librerías para el trabajo con arreglos (*numpy* y *pandas*).

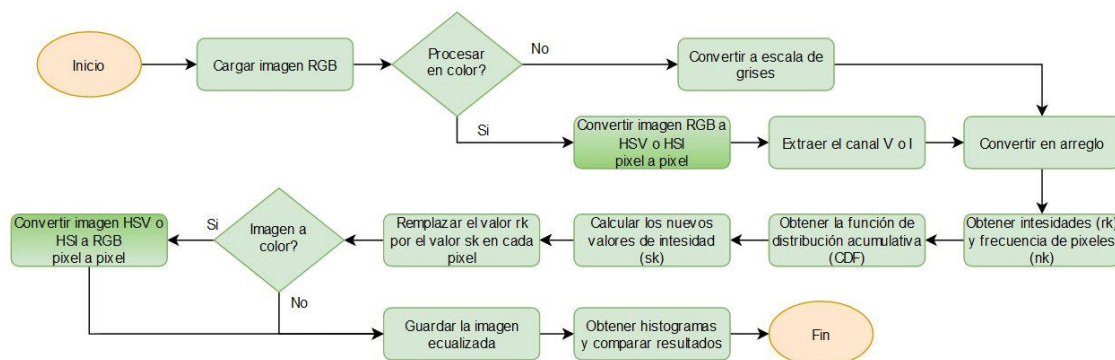


Figura 1. Diagrama de flujo del proceso de ecualización por histograma de imágenes.

Fuente. Elaboración propia.

4. Resultados

Para probar la efectividad de la solución propuesta, se utilizaron imágenes de referencia de la librería *skimage* reportadas en la documentación de la librería *equalize_hist()* y se compararon los resultados para imágenes en escala de grises.

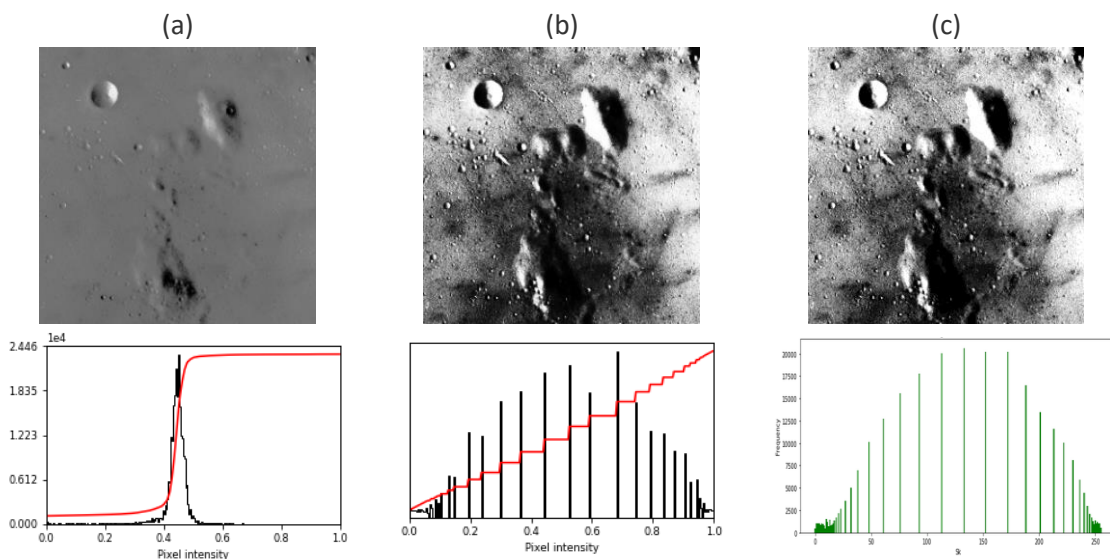


Figura 2. Comparación de ecualización del histograma. (a) Imagen original, (b) Imagen ecualización *skimage* (c) Imagen ecualización implementada.

En el caso de imágenes a color, se compararon los resultados reportados por (Lee & Tseng, 2017).

En el primer caso, se usan las imágenes de la “luna” que está en el método *data.moon()* de *skimage* y el “paisaje” en el que hace referencia hacia la biblioteca de *Wikipedia – Histogram Equalization* (ver Figura 2 y Figura 3).

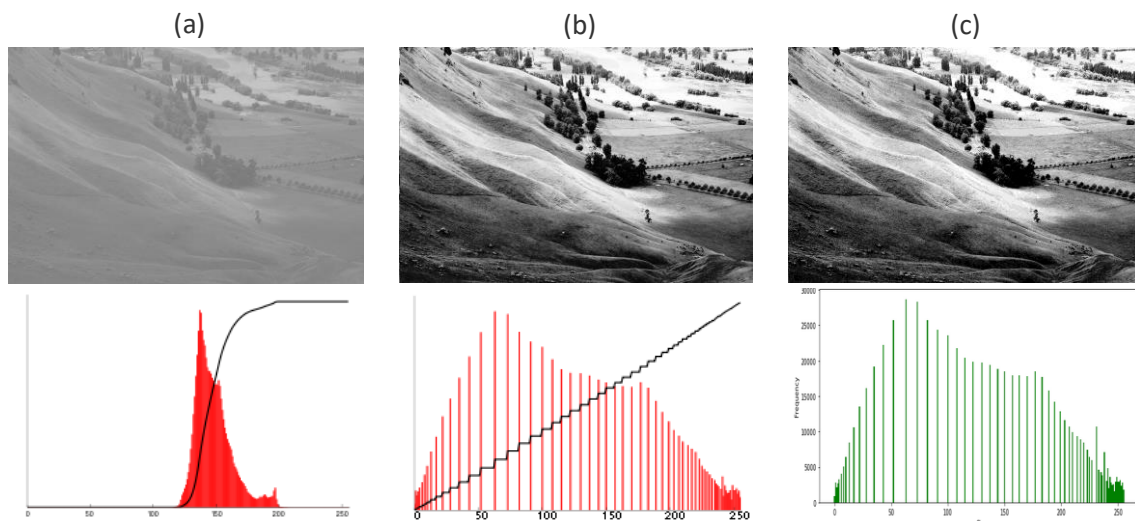


Figura 3. Comparación de ecualización del histograma. (a) Imagen original, (b) Imagen ecualización *skimage* (c) Imagen ecualización implementada.

En las imágenes a color, se tomaron como referencia la imagen del “*perro pug*” y la imagen de los “*globos*” (Figura 4 y Figura 5).

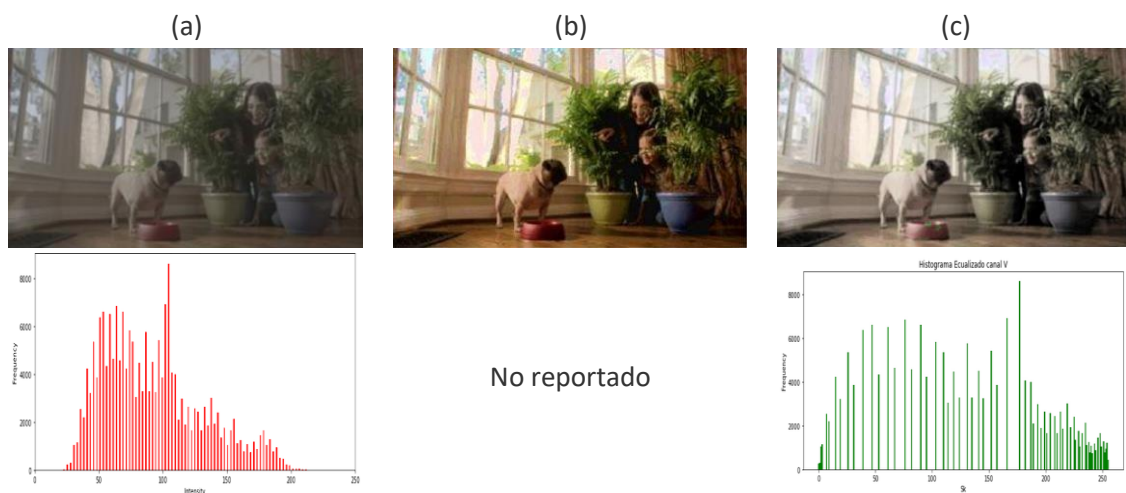


Figura 4. Comparación de ecualización del histograma. (a) Imagen original, (b) Imagen ecualización (Lee & Tseng, 2017) (c) Imagen ecualización implementada HSV.

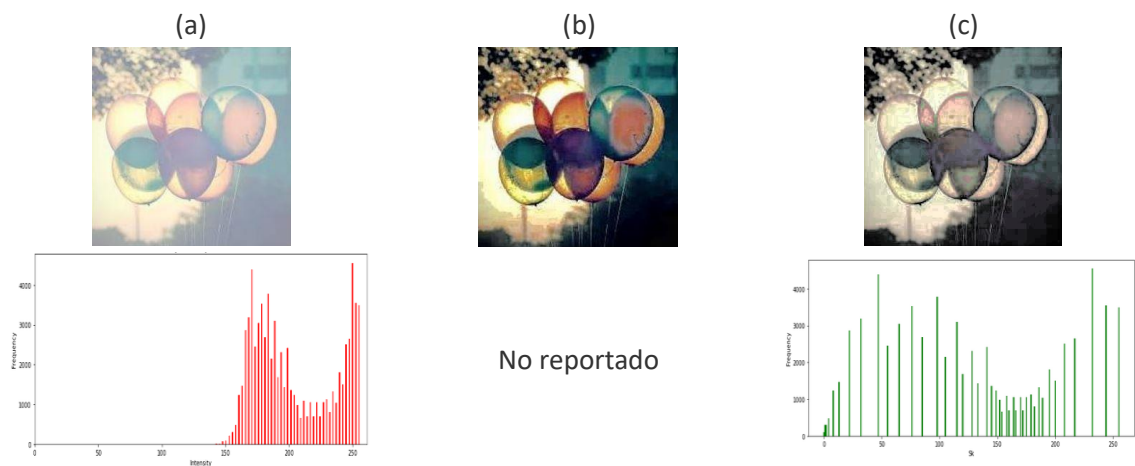


Figura 5. Comparación de ecualización del histograma. (a) Imagen original, (b) Imagen ecualización (Lee & Tseng, 2017) (c) Imagen ecualización implementada HSV.

Otras imágenes probadas se presentan a continuación:

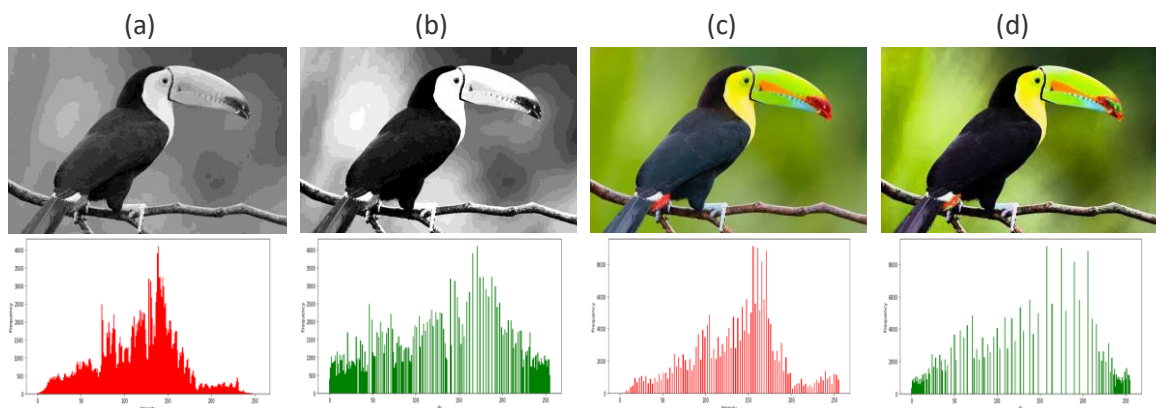


Figura 6. Comparación de ecualización del histograma. (a) Imagen escala de grises, (b) Imagen ecualización escala de grises (c) Imagen color (d) Imagen ecualización HSV.

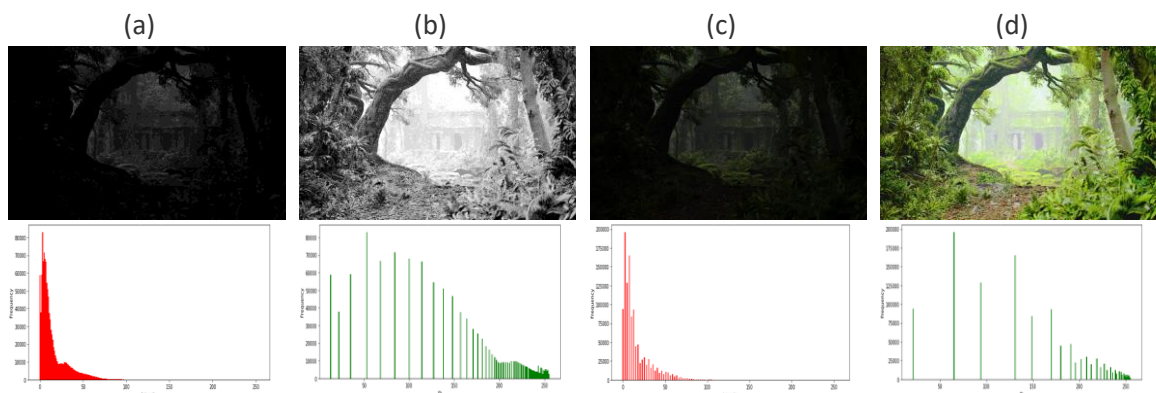


Figura 7. Comparación de ecualización del histograma. (a) Imagen escala de grises, (b) Imagen ecualización escala de grises (c) Imagen color (d) Imagen ecualización HSV.

5. Conclusiones

Los resultados anteriores confirman el análisis teórico de la técnica de ecualización del histograma, en el caso de imágenes con escala de grises se han logrado resultados muy satisfactorios que coinciden con la implementación nativa de la ecualización del histograma por *skimage*. Esto debido a que se está trabajando sobre la transformación hacia un solo canal resultado de una combinación lineal de los canales RGB ($gs = 0.2125 * r + 0.7154 * g + 0.0721 * b$).

Utilizando operaciones elementales de píxel se ha logrado una implementación robusta para mejorar el contraste de imágenes. Sin embargo, implementaciones más sofisticadas como el algoritmo CLAHE requieren ventanas de píxeles para lograr resultados aun mejores.

El mejoramiento del contraste para imágenes a color en RGB no puede producirse en cada canal ya que produce saturaciones de color no realistas, por lo que debe convertirse a otro sistema. Existen distintas formas y algoritmos que muestran cual o cuales canales deben procesarse mediante histograma (por ejemplo, HSV o HSI y trabajar con los canales V e I respectivamente) y los resultados solo pueden aplicarse a imágenes específicas, dado que en ciertas ocasiones no dan los resultados esperados, por ejemplo, cuando la imagen original tiene una distribución del histograma RGB con concentraciones bimodales de intensidad concentradas hacia tonos oscuros y muy claros. Por el contrario, cuando la distribución de intensidad de los canales RGB es similar a los canales I o V de luminancia o brillo respectivamente, se pueden lograr muy buenos resultados.

6. Referencias

- Gonzalez, R. C., & Woods, R. E. (2018). *Digital Image Processing*. Pearson.
<https://books.google.com.co/books?id=XmZvtAEACAAJ>
- Histogram Equalization — skimage v0.19.2 docs*. (s. f.). Recuperado 7 de enero de 2023, a partir de https://scikit-image.org/docs/stable/auto_examples/color_exposure/plot_equalize.html
- Histogram equalization - Wikipedia*. (s. f.). Recuperado 7 de enero de 2023, a partir de https://en.wikipedia.org/wiki/Histogram_equalization
- Histogram Equalization on Grayscale and Color Image | HYPJUDY*. (s. f.). Recuperado 7 de enero de 2023, a partir de <https://hypjudy.github.io/2017/03/19/dip-histogram-equalization/>
- Kong, N. S. P., & Ibrahim, H. (2008). Color image enhancement using brightness preserving dynamic histogram equalization. *IEEE Transactions on Consumer Electronics*, 54(4), 1962-1968.
- Lee, S.-L., & Tseng, C.-C. (2017). Color image enhancement using histogram equalization method without changing hue and saturation. *2017 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW)*, 305-306.
- Point Operations - Contrast Stretching*. (s. f.). Recuperado 7 de enero de 2023, a partir de <https://homepages.inf.ed.ac.uk/rbf/HIPR2/stretch.htm>
- RGB to grayscale — skimage v0.19.2 docs*. (s. f.). Recuperado 7 de enero de 2023, a partir de https://scikit-image.org/docs/stable/auto_examples/color_exposure/plot_rgb_to_gray.html