

## **Anexo C:**

### **“PRUEBAS CON FILTROS AUTOMATICOS Y MANUALES”**

#### **LIBRERÍAS DE PYTHON**

**BIBLIOTECA PILLOW:** Esta biblioteca proporciona una amplia compatibilidad con formatos de archivo (GIF, JPEG y PNG), además cuenta con capacidad de procesamiento de imágenes eficientes. La biblioteca de imágenes principal está diseñada para un acceso rápido a los datos almacenados en unos pocos formatos de píxeles básicos (Rojo, 2021).

**BIBLIOTECA SKIMAGE:** Es una colección de algoritmos para imágenes procesamiento y visión por computadora.

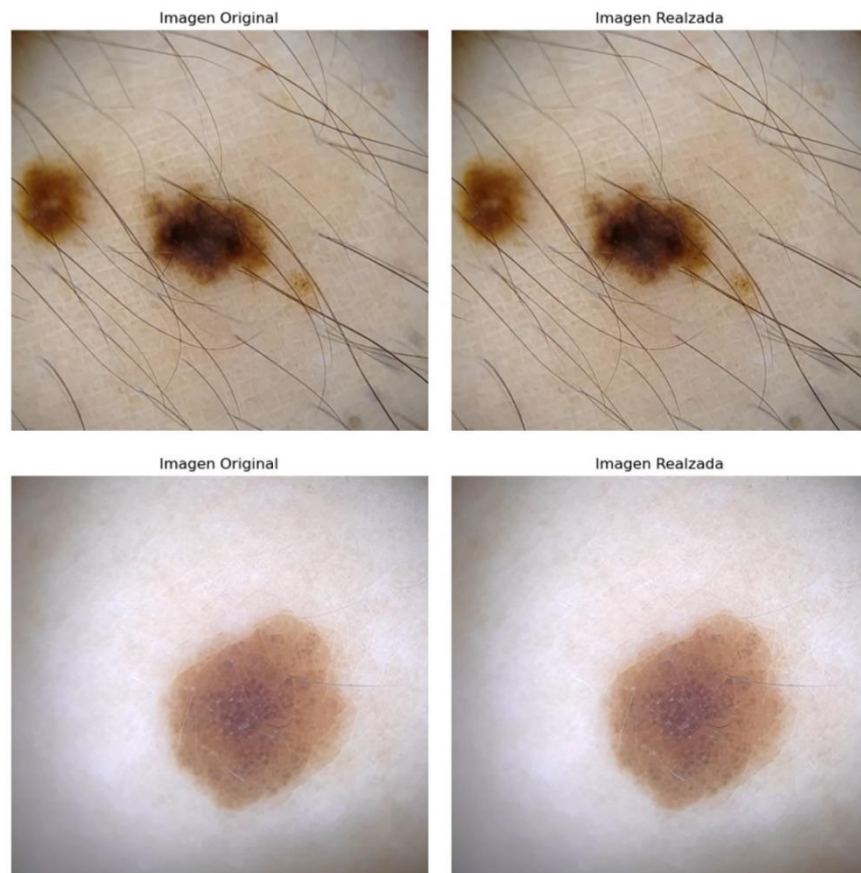
**BIBLIOTECA NUMPY:** Esta se centra en vectores y matrices, proporciona matrices multidimensional objeto de alto rendimiento y herramientas para trabajar con arreglos. (Moreno, 2020)

**BIBLIOTECA OPENCV:** Es una biblioteca de código abierto. Proporciona una infraestructura común para aplicaciones de visión por computadora. Cuenta con algoritmos optimizados, que incluyen aprendizaje automático. (Moreno, 2020).

## FUNCIONES CON FILTROS AUTOMATICOS

### 1. BIBLIOTECA PILLOW

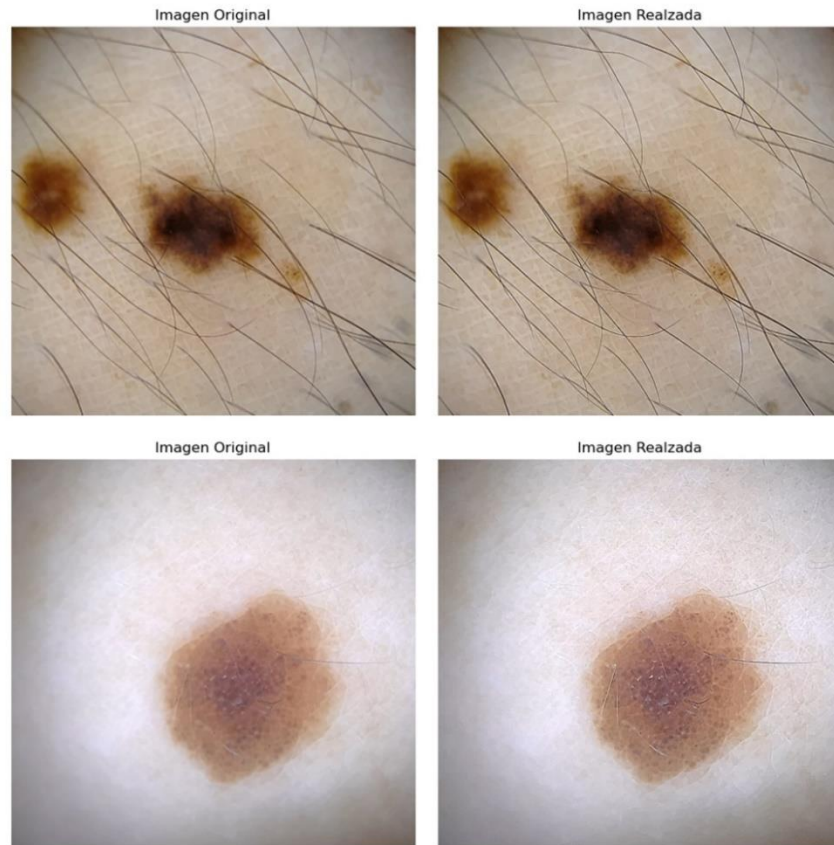
**1.1 Realce de características (Detalles):** La función “ImageFilter.DETAIL” permite resaltar los detalles y los bordes de la imagen, mejorando su nitidez y contraste. Ayuda a visualizar los detalles finos y mejorar la calidad de la imagen. Este filtro automático utiliza algoritmos de procesamiento de imágenes para resaltar las características en la imagen.



**Figura 1.** Muestras originales y realzadas utilizando el filtro “ImageFilter.DETAIL”

**RESULTADOS:** Al comparar la imagen original con la del filtro “ImageFilter.DETAIL”, no se percibe un cambio significativo en los parámetros, sin embargo, si mejora la nitidez de esta.

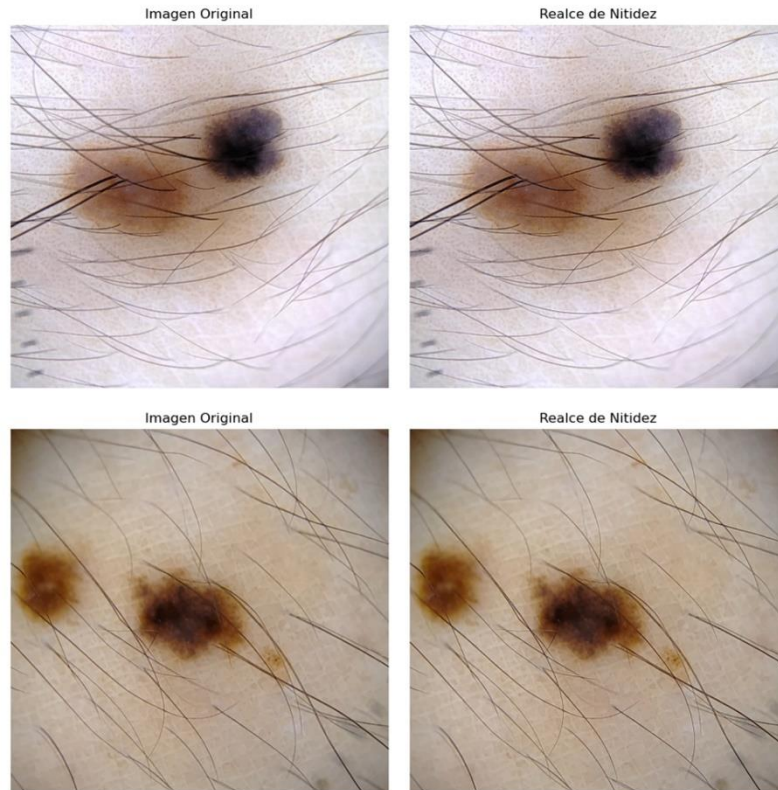
**1.2 Realce de características (bordes):** La función “ImageFilter.EDGE\_ENHANCE” se utiliza para mejorar los bordes de una imagen, destaca los cambios bruscos en la intensidad en los píxeles vecinos. Al aplicar este filtro a una imagen, se resaltan los bordes y mejora la nitidez. Esto puede ser útil en diversas aplicaciones, como la detección de contornos o la mejora de la calidad de una imagen (Suárez L.A., 2014).



**Figura 2.** Muestras originales y realzadas utilizando el filtro “ImageFilter.EDGE\_ENHANCE”

**RESULTADOS:** Al implementar este filtro, se mejora la definición y la pigmentación de la piel, destacando su estructura característica. Sin embargo, es importante tener presente que este filtro no puede resaltar ciertas áreas blancas presentes en la imagen dos de la figura 2.

**1.3 Mejora de nitidez:** La función “ImageFilter.SHARPEN”, es un filtro espacial que utiliza el método de convolución para incrementar la nitidez de una imagen. Este proceso realza los bordes y detalles al intensificar las diferencias de intensidad entre píxeles adyacentes, logrando así un aumento en los detalles y bordes de la imagen. <https://pythontic.com/image-processing/pillow/sharpen-filter>

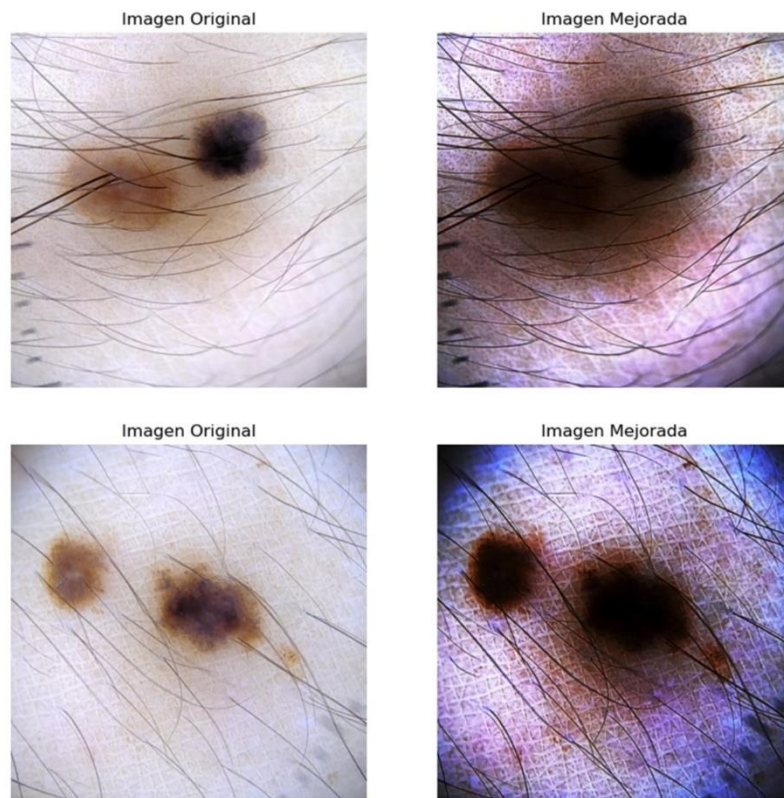


**Figura 3.** Muestras originales y realzadas utilizando el filtro “ImageFilter.SHARPEN”

**RESULTADOS:** La mejora de la nitidez es un proceso subjetivo, depende del tipo de imagen. En la figura 3, no se observan cambios significativos en comparación con las imágenes originales, sin embargo, hay pérdida de ciertos detalles finos, especialmente en área de transición gradual.

## 2. BIBLIOTECA SKIMAGE

**2.1 Mejora de contraste Skimage:** La función “`exposure.equalize_hist()`” permite visualizar el contraste de la imagen después de la ecualización del histograma, lo que hace mejorar la distribución de los niveles de intensidad en una imagen. Al aplicar la ecualización del histograma, se redistribuyen los valores de los píxeles en el histograma de tal manera que se obtiene una distribución uniforme y detalles más claros. <https://scikit-image.org/docs/stable/api/skimage.exposure.html>



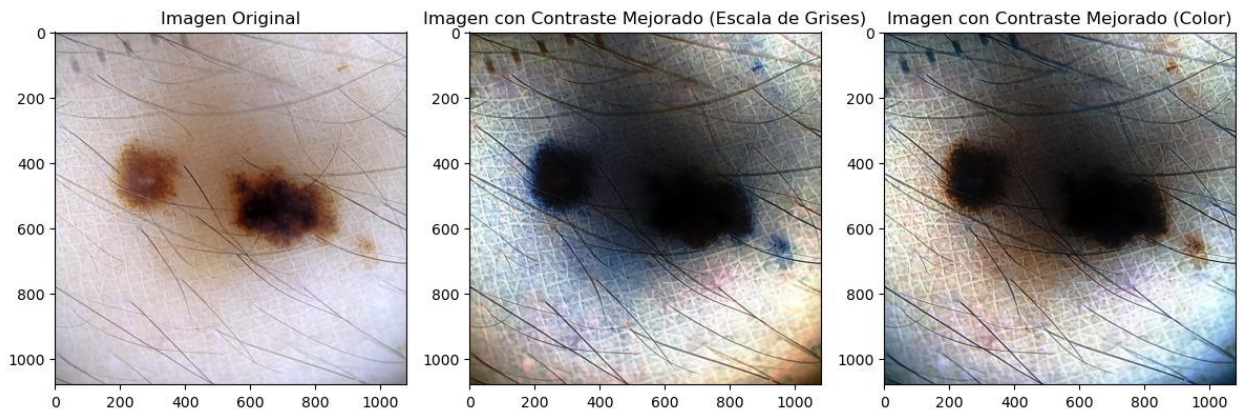
**Figura 4.** Muestras originales y después de utilizar el filtro “`exposure.equalize_hist()`”

**RESULTADOS:** En este caso la imagen pierde por completo la pigmentación de la piel, ya que este filtro hace que aumente el brillo y oscuridad, es decir las zonas con sombras leves las intensifica, lo que hace saturar el color y perder los detalles.



### 3. BIBLIOTECA OPENCV

**3.1 Mejora de contraste OPENCV:** la función “cv2.merge()” se utiliza para combinar varios canales de color en una única imagen, este aplica la ecualización a cada canal, y luego une nuevamente los canales. Cuando la imagen es en color, cada canal (rojo, verde, azul) se almacena por separado.



**Figura 5.** Muestras originales, con contraste mejorado en escala de grises y color, utilizando el filtro “cv2.merge()”

**RESULTADO:** Este filtro separa la imagen en los 3 canales RGB, inicialmente hace la modificación desde la escala de grises, sin embargo, en este estado las sombras toman tonalidades azules y el exterior o borde de la imagen las sombras de ven de una pigmentación café. Por otra parte, al tener el resultado final de la imagen en color, las sombras del centro toman tonalidades cafés que no corresponden a la lesión y las sombras de los bordes exteriores tienen tonalidades azules.

**3.2 filtro de realce laplaciano:** La función “cv2.Laplacian()” se utiliza para realizar la detección de detalles mediante el operador Laplaciano. Este operador calcula la segunda derivada de la imagen y resalta las áreas de la imagen donde hay cambios bruscos en la intensidad, indicando así la presencia de bordes y detalles.

[https://docs.opencv.org/4.x/d4/d86/group\\_imgproc\\_filter.html#gac342a1bb6eabf6f55c803b09268e36dc](https://docs.opencv.org/4.x/d4/d86/group_imgproc_filter.html#gac342a1bb6eabf6f55c803b09268e36dc)

Libro: "Learning OpenCV 3" de Adrian Kaehler y Gary Bradski; Capítulo 6: Image Processing; Sección 6.3: Gradients and Sobel Derivatives; Sección 6.4: Laplacian



**Figura 6.** Muestra original y con realce de bordes, utilizando el filtro “cv2.Laplacian()”

**RESULTADO:** Al aplicar esta función los tonos de la piel pasan a color negro y los vellos a color gris, este filtro se enfoca en los detalles superficiales y anula por completo cualquier color o textura de la piel, esto podría ser de utilidad para una función de eliminar vellos.

**3.3 Contraste:** La función “cv2.equalizeHist” aplica la ecuación de histograma al canal de luminosidad, lo cual permite mejorar el contraste al redistribuir los valores de píxeles en el canal de luminosidad. La ecualización del histograma expande los rangos de intensidad de píxeles en las regiones de baja luminosidad y comprime los rangos en las regiones de alta luminosidad, lo que resulta en una imagen con un mejor equilibrio de intensidades.

**Libro: "Programming Computer Vision with Python" de Jan Erik Solem**

**Capítulo 1: Basics; Sección 1.4.1: Histogram Equalization**



**Figura 7.** Muestra original y Ecualización de Histograma, utilizando el filtro “cv2.cvtColor()”

**RESPUESTA:** Todas las tonalidades presentes en la imagen se convierten en formato sepia, donde se realza la estructura de la piel y los vellos, pero se pierde por completo la forma, color y bordes del lunar.



3.4 **Filtro realce laplaciano sobel:** La función “cv2.Sobel()” calcula las derivadas de la primera, segunda, tercera o imagen mixta utilizando un operador de Sobel extendido. Para realizar la detección de bordes mediante el operador, se calcula la primera derivada de la imagen y resalta las áreas donde hay cambios bruscos en la intensidad, indicando la presencia de bordes.

Libro: "Learning OpenCV 3" de Adrian Kaehler y Gary Bradski; Capítulo 6: Image Processing; Sección 6.3: Gradients and Sobel Derivatives

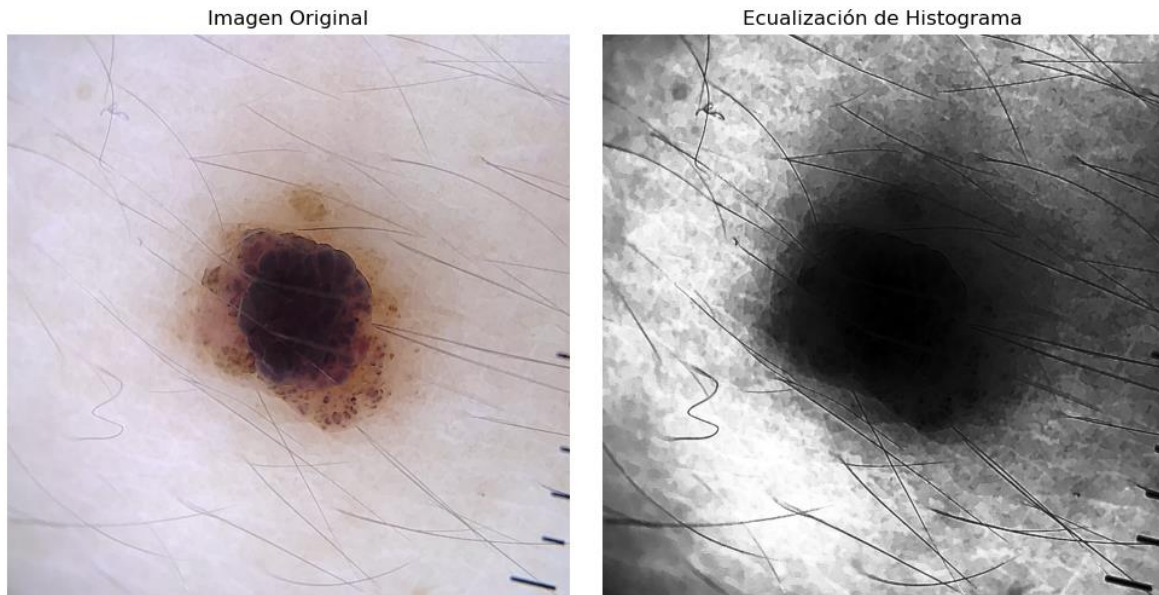


**Figura 8.** Imagen original y procesada por el filtro cv2.Sobel()

**RESPUESTA:** Este filtro realza bordes y texturas de la piel con un color blanco o gris, los demás colores los vuelve negros, una aplicación de este filtro es crear mascarar para modificar imágenes a partir de las estructuras resaltadas.

3.5 **Mejora contraste:** La función “cv2.equalizeHist()” se utiliza para realizar la ecualización del histograma en una imagen. Este proceso mejora el contraste de la imagen al redistribuir los niveles de intensidad de píxeles en todo el rango disponible.

Libro: "Programming Computer Vision with Python" de Jan Erik Solem; Capítulo 1: Basics; Sección 1.4.1: Histogram Equalization

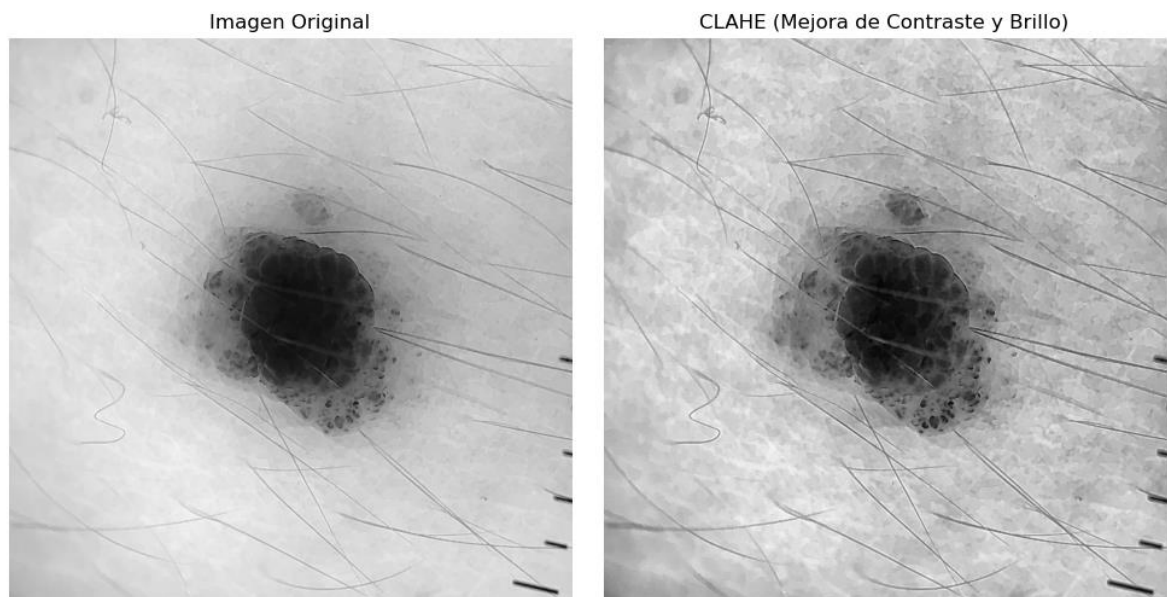


**Figura 9.** Imagen Original y procesada por ecualización de histograma por medio del filtro cv2.equalizeHist()”

**RESULTADO:** Al aplicar este tipo de filtro se pierde por completo la forma y color de la lesión pigmentada, se aprecia una intensidad en los detalles superficiales como es el caso de los vellos. Este filtro identifica las sombras o ruido como bordes.

**3.6 Brillo y contraste la función:** La función “cv2.createCLAHE()” contiene un objeto CLAHE (Contrast Limited Adaptive Histogram Equalization), que es una técnica de ecualización de histograma adaptativa con limitación de contraste. El CLAHE es una versión mejorada de la ecualización de histograma estándar. A diferencia de la ecualización de histograma global, este divide la imagen en pequeñas regiones llamadas "parches" y aplica la ecualización de histograma por separado a cada parche, lo que evita el ruido en áreas de bajo contraste.

Libro: "Computer Vision: Algorithms and Applications" de Richard Szeliski;  
Capítulo 3: Image Processing; Sección 3.4.1: Histogram Equalization



**Figura 10.** Imagen original en escala de grises y procesada por el filtro cv2.createCLAHE()”

### **RESULTADO:**

Al comparar la imagen original con la del filtro CLAHE observamos que mejora el contraste, incluso se puede ver en detalle la estructura de la piel.

Al implementar este filtro no se afecta globalmente la distribución de intensidades, lo que permite que las regiones de baja luminosidad o colores oscuros, puedan resaltar su textura, detalles locales y los bordes.

## FUNCIONES MANUALES (VARIANDO PARAMETROS)

### 1. BIBLIOTECA PILLOW

1.1 **Compensación de luces:** La función “`enhancer.enhance()`” crea un objeto `enhancer` para ajustar el brillo de la imagen y atributos visuales. La función `enhance()` toma un parámetro que controla la intensidad del ajuste. Al variar dicho parámetro en un factor de 1 no realiza ningún cambio, mientras que valores mayores a 1 aumentarán la intensidad del brillo, y valores entre 0 y 1 disminuirán la intensidad.

#### Con un parámetro 0.8:



**Figura 11.** Imagen original y modificada por el filtro `enhancer.enhance(0.8)`

Tomando como referencia la imagen original de la figura 11, se comprueba que el rango de  $[0-1]$  se opaca la imagen y entre  $[1-2]$  se aumenta la intensidad de brillo, el rango que observamos adecuado para la modificar de forma adecuada las imágenes es entre  $[1-1.5]$ .

**RESULTADO:** Al utilizar el método `enhance` con un factor de 0.8 se disminuye el brillo en un 20%, al disminuir el brillo la lesión se oscurece en su parte central.



**Con un parámetro de 1.3:**

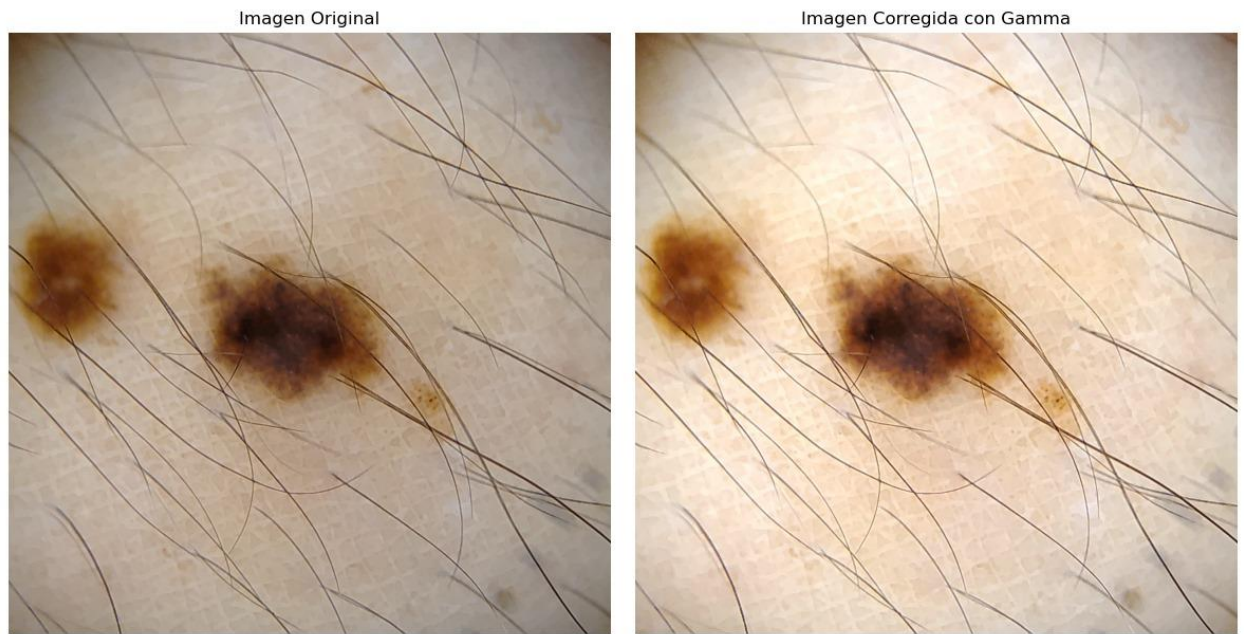


Figura 12. Imagen original y modificada por el filtro `enhance(1.3)`

**RESULTADO:** Al utilizar el método `enhance` con un factor de 1.3 se aumenta el brillo en un 30%, lo que hace que el borde exterior de la imagen pierda el color característico de la piel en dicha área.

1.2 **Corrección Gamma:** la función principal es `ImageEnhance.Brightness()` el cual pertenece al módulo `ImageEnhance` de la biblioteca `Pillow` y se utiliza para crear un objeto que modifica el brillo de una imagen.

Puede utilizarse para aumentar o disminuir el brillo de una imagen a través del método `enhance(factor)`. Un factor mayor a 1 aumentará el brillo, mientras que un factor menor a 1 lo disminuirá.



**Figura 13.** Imagen original y con el filtro “ImageEnhance.Brightness()”, con un factor gamma = 1.3

El rango que utiliza el parámetro gamma es de [0-2], donde el rango de 0 a 1 oscurece la imagen y de 1 a 2 aumenta la intensidad.

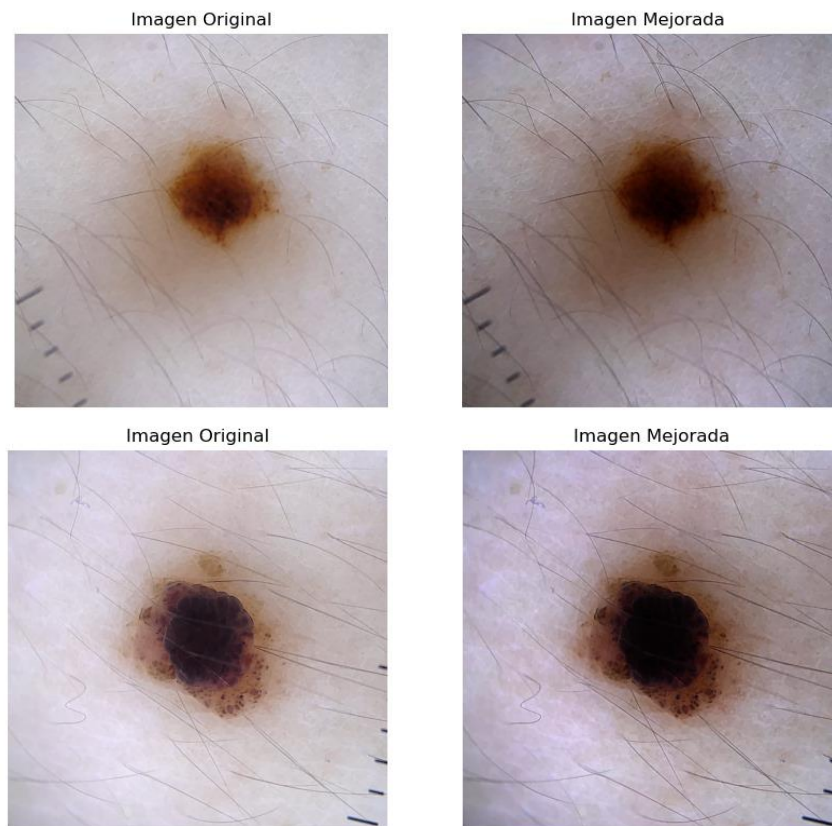
**RESULTADO:** El factor de corrección gamma=1.3, logra ajustar la luminancia de una imagen al elevar los valores de intensidad de los píxeles.

Al aumentar la intensidad de luz los colores toman un color cálido, y se puede observar con más claridad la estructura interna del lesión.

## 2. BIBLIOTECA SKIMAGE

2.1 **FUNCIÓN GAMMA:** La función “`exposure.adjust_gamma()`” permite aumentar el contraste de la imagen, realzando así los detalles y haciendo que los colores de la imagen sean más intensos. Además se puede ajustar el contraste de una imagen mediante la variación del parámetro gamma, donde un valor mayor que 1 aumenta el contraste, y un valor menor que 1 lo disminuye.

scikit-image: Documentación de `exposure.adjust_gamma()`.

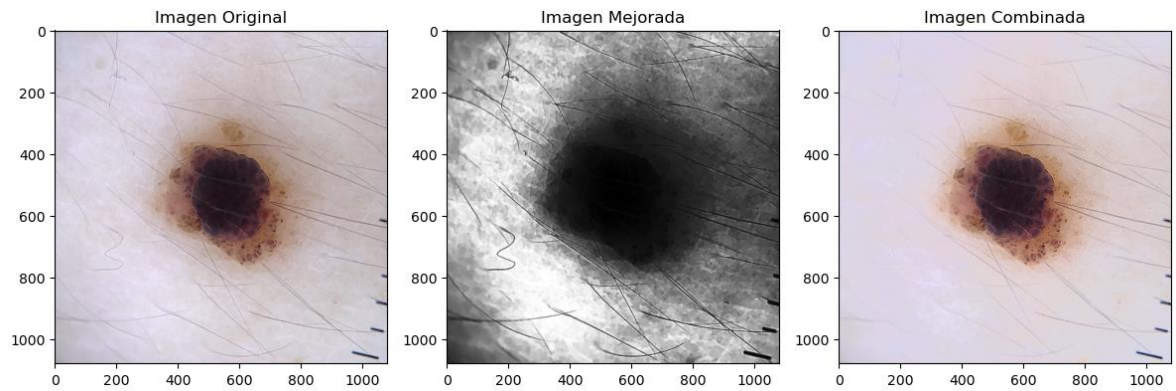


**Figura 14.** Imagen original y con el filtro `exposure.adjust_gamma()`, utilizando un parámetro  $\text{gamma}=1.8$

**RESULTADO:** Al variar el parámetro con valores menores a 1 no se observa un cambio significativo, sin embargo, para valores superiores a este los colores toman un valor intenso, por lo que no se puede detallar muy bien la pigmentación, al ajustar el parámetro gamma en 1.8 se aumenta el contraste, este filtro es apropiado para situaciones donde las condiciones de iluminación original afectan la visibilidad de detalles importantes.

2.2 **FUNCIÓN GAMMA:** Con la función `exposure.equalize_hist()` se mejora el contraste mediante la ecualización del histograma, especialmente en áreas donde la distribución original de intensidades es limitada.

Este código toma la imagen original la pasa a una escala de grises como entrada y aplica la ecualización del histograma.



**Figura 15.** Imagen Original, escala de grises para ecualizar y la combinación de estas dos imágenes ponderando cada una con un factor  $\text{Alpha}=1.2$

**RESULTADO:** se resaltan los colores oscuros, sin embargo, la estructura de la piel y vellos se ven borrosos, como si tuviera un efecto de maquillaje, que hace que se desvanezcan.



### 3. BIBLIOTECA OPENCV **OPENCV Y NUMPY**

3.1 MASCARA DE DESENFOQUE “GAUSSIANBLUR”: Este filtro permite suavizar, reducir el ruido y las transiciones abruptas de píxeles.

La instrucción “**cv2.addWeighted(imagen, 1, cv2.GaussianBlur(imagen, (0, 0), 2), -0.5)**” realiza un proceso de enfoque mediante el uso de desenfoque gaussiano y la aplicación de una máscara. La expresión “**cv2.addWeighted()**” se utiliza para encapsular la función principal “**cv2.GaussianBlur()**”, la cual cuenta con una desviación estándar para crear una versión desenfocada de la imagen original. Posteriormente se combina la imagen original y la versión desenfocada con ponderaciones específicas.

Desenfoca una imagen usando un filtro gaussiano. La función convoluciona la imagen fuente con el núcleo gaussiano especificado. Se admite el filtrado local.

[https://docs.opencv.org/4.x/d4/d86/group\\_imgproc\\_filter.html#gaabe8c836e97159a9193fb0b11ac52cf1](https://docs.opencv.org/4.x/d4/d86/group_imgproc_filter.html#gaabe8c836e97159a9193fb0b11ac52cf1)

el filtrado gaussiano de paso bajo calcula un promedio ponderado de los valores de píxeles en la vecindad, en el que los pesos disminuyen con la distancia desde el centro de la vecindad

[https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL\\_COPIES/MANDUCHI1/Bilateral\\_Filtering.html](https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html)

Se aplica un suavizado gaussiano en la imagen de origen de entrada

<https://idus.us.es/bitstream/handle/11441/102090/TFG-2877-MORENO%20FERNANDEZ.pdf?sequence=1&isAllowed=y>

parámetro de desviación estándar en 1

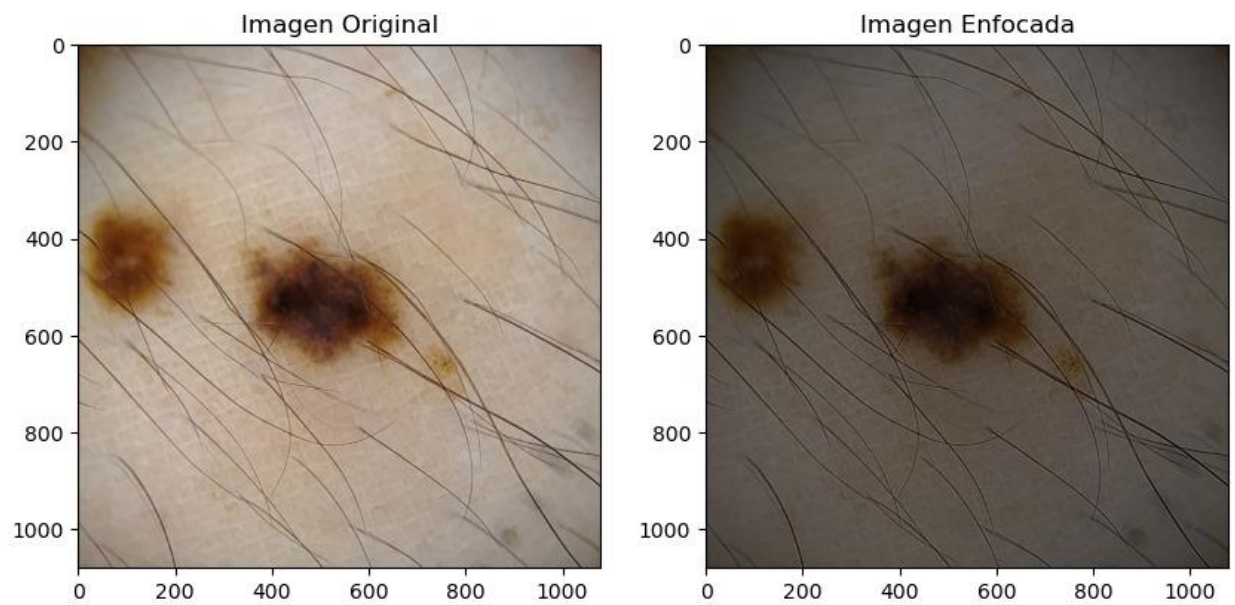


Figura 16

## RESULTADO

parámetro de desviación estándar en 2

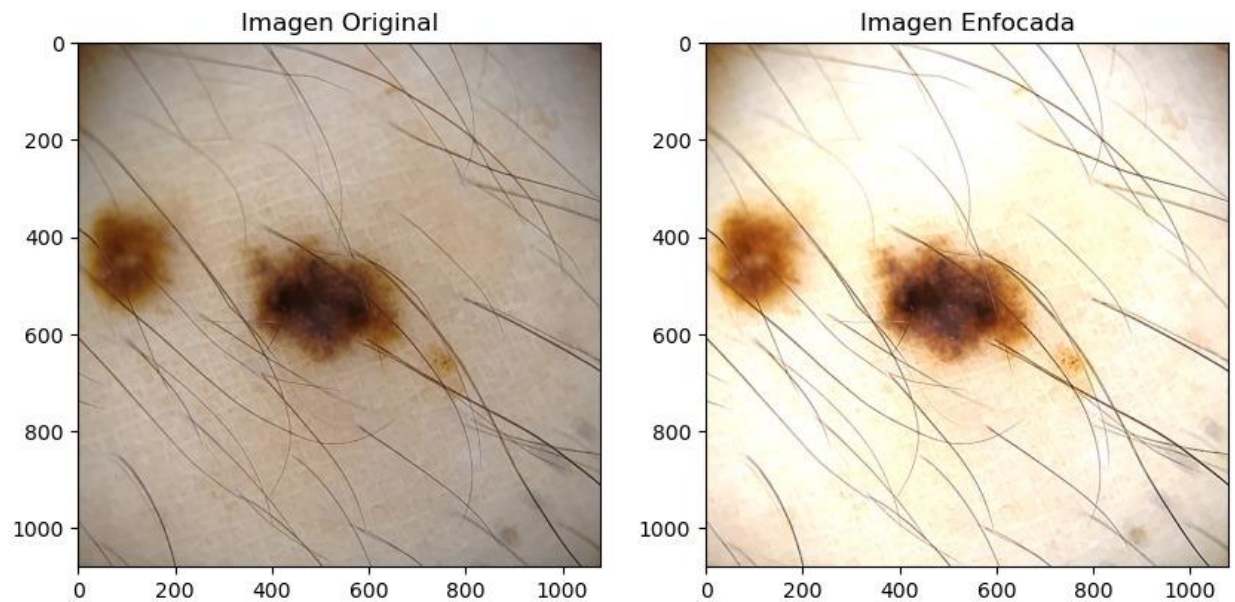


Figura 17

## RESULTADO

se ve saturado la imagen

### 3.2 CORRECCIÓN DE EXPOSICIÓN Y BALANCE DE COLOR

“COVERTSCALEABS”: “cv2.convertScaleAbs(imagen, alpha=alpha, beta=beta)”

Aplica una transformación lineal a cada uno de los píxeles de la imagen, escalando y ajustando los valores de píxeles mediante el contraste (para un valor de variable **alpha** >1 aumenta, mientras entre [0-1] disminuye) y el brillo (para un valor de variable **beta** positivo aumenta brillo, valor negativo lo disminuye).

La función permite ajustar el brillo y el contraste de la imagen, ya que al utilizar "Abs" se calcula el valor absoluto, asegurando que los resultados sean valores enteros no negativos.



Figura 18

RESULTADO: En este caso se varían dos parámetros Alpha=0.1 y beta=0.8, con valores mayores la imagen se pierde por completo, al aumentar el valor.

Inicialmente se quería conservar el contraste y que el brillo no fuera elevado para poder observar a detalle las características internas de la lesión.