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 procesamiento de imágenes y visión por computadora. (scikit-image, 2023)

BIBLIOTECA NUMPY: Esta se centra en vectores y matrices, proporciona matrices multidimensionales, objetos 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)

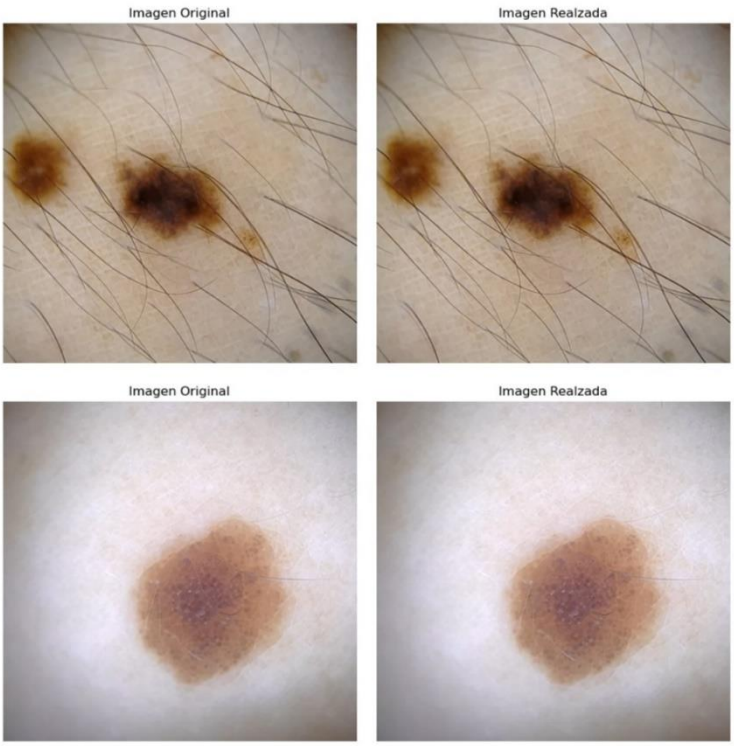
FUNCIONES CON FILTROS AUTOMATICOS

1. BIBLIOTECA PILLOW

1.1 Realce de características (Detalles): La función “ImageFilter.DETAIL” es un filtro automático que utiliza algoritmos de procesamiento de imágenes para resaltar las características como detalles y bordes de la imagen, lo cual ayuda a visualizar los detalles finos y mejorar la calidad de la imagen.

Figura 1.

Respuesta del código “Realce de características PILLOW” con la función “DETAIL”



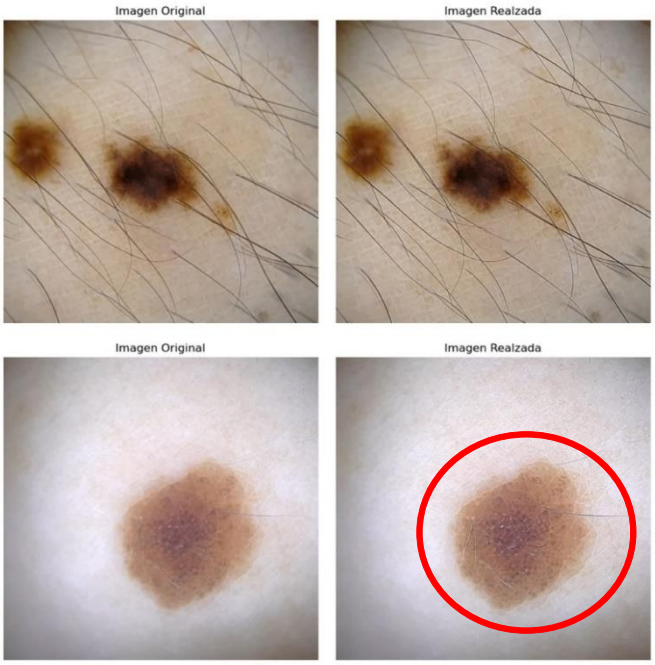
Nota. Muestras originales y realzadas utilizando el filtro “ImageFilter.DETAIL”. Creado por los autores.

RESULTADOS: Al aplicar el filtro “ImageFilter.DETAIL” y comparar la respuesta con la imagen original, no se percibe un cambio significativo en sus detalles y bordes, sin embargo, si mejora la iluminación de la zona central.

1.2 Realce de características (bordes): La función "ImageFilter.EDGE_ENHANCE" 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.

Respuesta del código "Realce de características PILLOW" con la función "EDGE_ENAHANCE"



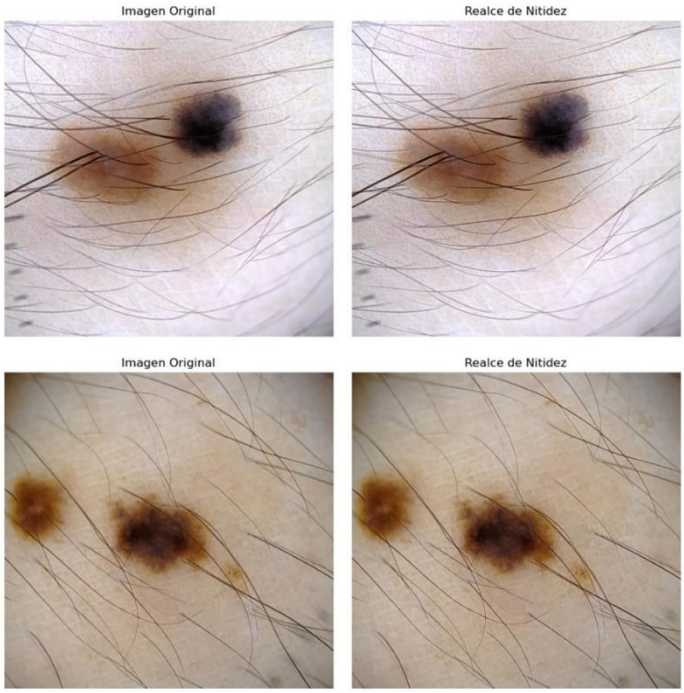
Nota: Muestras originales y realzadas utilizando el filtro "ImageFilter.EDGE_ENHANCE". Creado por los autores.

RESULTADOS: Al implementar este filtro se mejoran los bordes y la pigmentación de la imagen. Sin embargo, existen estructuras de color blanco en la imagen dos de la figura 2, que se pierden al resaltar los bordes de la lesión.

1.3 Mejora de nitidez: La función “ImageFilter.SHARPEN”, es un filtro espacial que utiliza el método de convolución para aumentar la nitidez de una imagen. Este proceso realza los bordes y detalles de la imagen al incrementar las diferencias de intensidad entre píxeles adyacentes. (PythonExamples.org, 2023).

Figura 3.

Respuesta del código “NITIDEZ” con la función “SHARPEN”.



Nota: Muestras originales y realzadas utilizando el filtro “ImageFilter.SHARPEN”.

Creado por los autores.

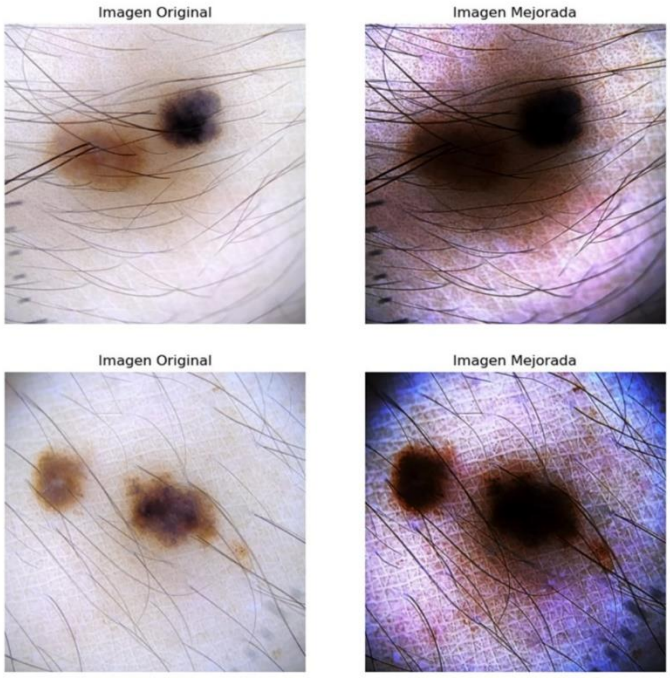
RESULTADOS: En la figura 3, no se observan cambios significativos en comparación con las imágenes originales, sin embargo, hay perdida de algunos detalles, por ejemplo, en áreas donde existe transición de colores en la pigmentación de la lesión.

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, se redistribuyen los valores de los píxeles en el histograma de tal manera que se obtiene una distribución uniforme y detalles más claros.(scikit-image, 2023).

Figura 4.

Respuesta del código “Mejora de contraste SKIMAGE” con la función “Exposure.qualize_hist”



Nota: Muestras originales y después de utilizar el filtro “exposure.equalize_hist()”. Creado por los autores.

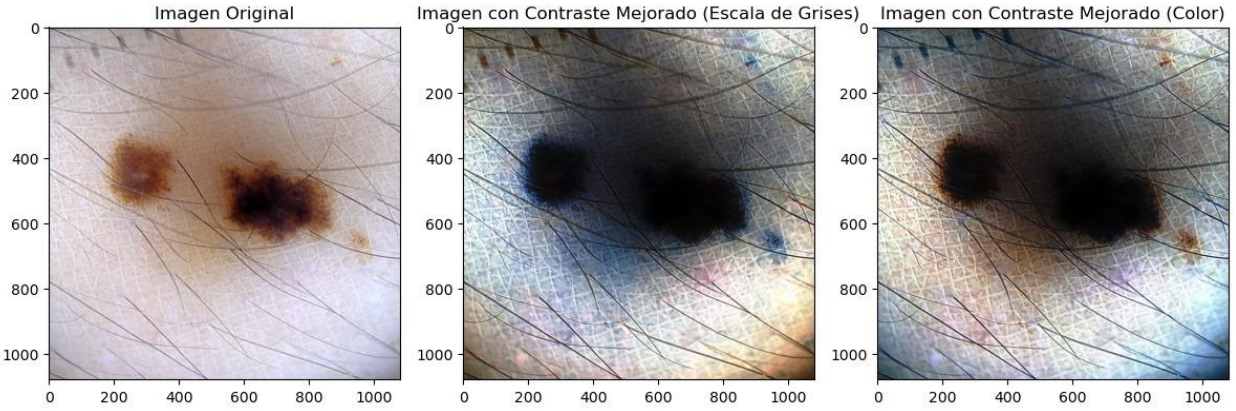
RESULTADOS: Al aplicar este filtro la imagen pierde su tono inicial, esto se debe a que la imagen original exhibe áreas con niveles de brillo y oscuridad variados, provocando que la imagen resultante pierda su pigmentación y color. Este inconveniente resalta la complejidad de abordar el contraste en imágenes con diferencias significativas en la iluminación.

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. (Smith, 2023)

Figura 5.

Respuesta del código “Mejora de contraste OPENCV” con la función “cv2.merge”



Nota: Muestras originales, con contraste mejorado en escala de grises y color, utilizando el filtro “cv2.merge()”. Creado por los autores.

RESULTADO: Este filtro separa la imagen en los 3 canales RGB, inicialmente hace la modificación desde la escala de grises, sin embargo, las sombras toman tonalidades azules y el exterior o borde de la imagen las sombras se ven de un color 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. (OpenCV, 2023)

Figura 6.

Respuesta del código "Filtro de realce laplaciano OPENCV" con la función "cv2.Laplacian"

Nota: Muestra original y con realce de bordes, utilizando el filtro "cv2.Laplacian()". Creado por los autores.

RESULTADO: Al aplicar esta función los tonos de la piel cambian 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 útil para una función que elimine 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 pixeles 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. (Jan Erik Solem, 2012).

Figura 7.

*Respuesta del código "CONTRASTE" con la función "*cv2.cvtColor*"*

Nota: Muestra original y Ecualización de Histograma, utilizando el filtro "cv2.cvtColor()". Creado por los autores.

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. (Bradski & Kaehler, 2008)

Figura 8.

*Respuesta del código "Filtro de realce laplaciano OPENCV" con la función "*cv2.Sobel*".*

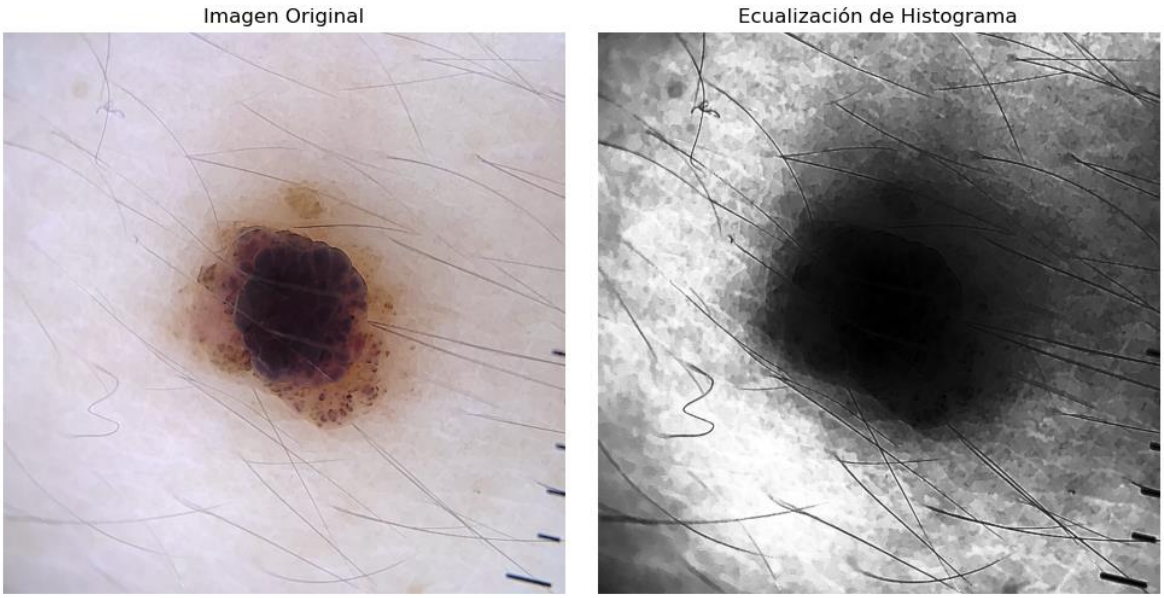
Nota: Imagen original y procesada por el filtro cv2**.**Sobel(), creado por los autores.

RESPUESTA: Este filtro realza bordes y texturas de la piel en escala de grises, una aplicación de este filtro es crear mascaras 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. (Jan Erik Solem, 2012).

Figura 9.

*Respuesta del código "MEJORA DE CONTRASTE" con la función "*cv2.equalizeHist*"*



Nota: Imagen Orignal y procesada por ecualización de histograma por medio del filtro cv2.equalizeHist()". Creado por los autores.

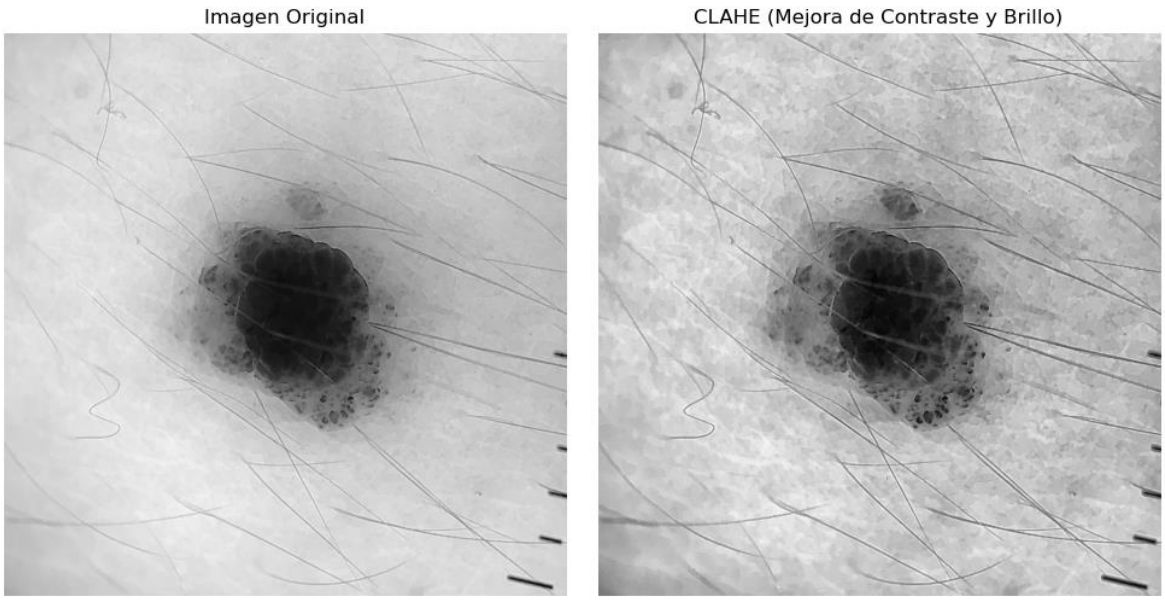
RESULTADO: Este filtro trabaja en escala de grises, lo cual genera que se disminuya en su gran parte su tonalidad o la pigmentación de la lesión y se difumine su forma. A su vez, es posible evidenciar que se aprecian con más intensidad algunos detalles como sucede con los vellos.

3.6 **Brillo y contraste:** 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. (Szeliski, 2010).

Figura 10.

*Respuesta del código "BRILLO Y CONTRASTE" con la función "*cv2**.**createCLAHE*"*



Nota: Imagen original en escala de grises y procesada por el filtro cv2**.**createCLAHE()". Creado por los autores.

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. (pillow documentation, 2023).

Figura 11.

Respuesta del código “Compesación de Luces con Pillow” con la función “enhancer.enhance” utilizando un parámetro de 0.8.

Nota: Imagen original y modificada por el filtro enhancer.enhance(0.8). Creado por los autores.

RESULTADO: Al utilizar el método enhance con un factor de 0.8 se disminuye el brillo y la lesión se oscurece en su parte central.

Figura 12.

Respuesta del código "Compesación de Luces con Pillow" con la función "enhancer.enhance" utilizando un parámetro de 1.3.

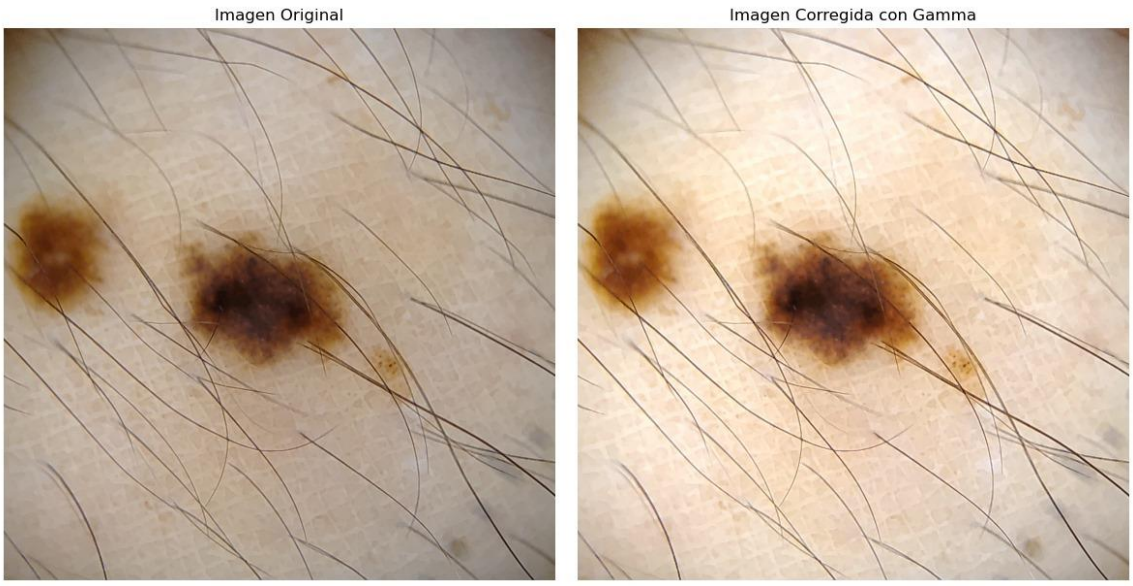
Nota: Imagen original y modificada por el filtro enhancer.enhance(1.3). Creado por los autores.

RESULTADO: Al utilizar el método enhance con un factor de 1.3 se aumenta el brillo, lo que hace que el tono de piel pierde su color característico.

1.2 **Corrección Gamma**: la función principal es ImageEnhance.Brightness(gamma) el cual pertenece al módulo ImageEnhance de la biblioteca Pillow, se utiliza para crear un objeto que modifica, aumenta o disminuye el brillo de una imagen. Un factor mayor a 1 aumentará el brillo, mientras que un factor menor a 1 lo disminuirá. (McBride, 2020).

Figura 13.

Respuesta del código "Correción Gamma con Pillow" con la función "ImageEnhance.Brightness"



Nota: Imagen original y con el filtro "ImageEnhance.Brightness()", con un factor gamma = 1.3. Creado por los autores.

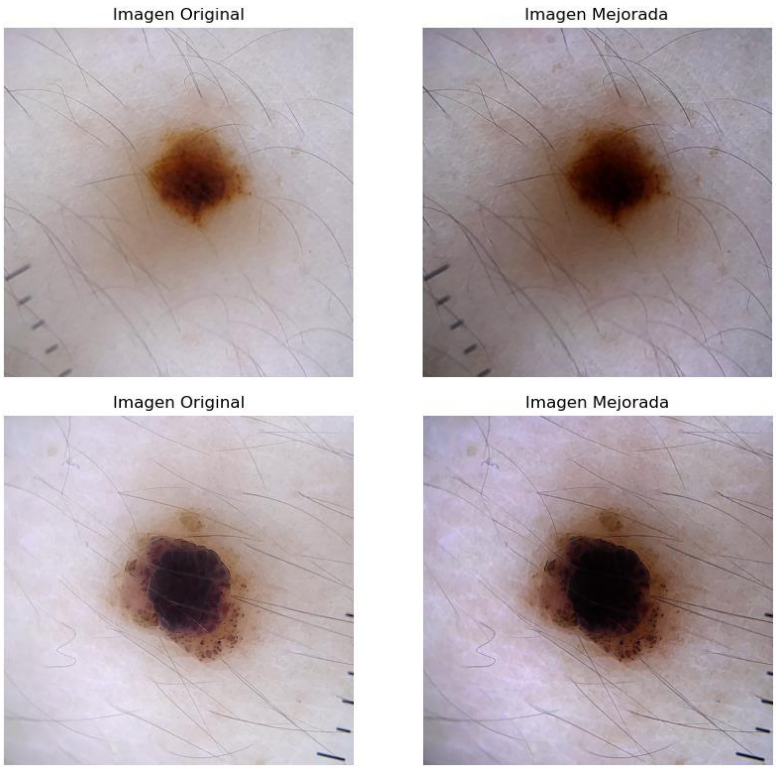
RESULTADO: El factor de corrección gamma=1.3, logra ajustar el brillo de una imagen al elevar los valores de intensidad de los píxeles, dicha característica se ve reflejada en colores cálidos que toma la imagen, y se puede observar con claridad la estructura interna de la lesión, mostrada en la figura 13.

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, 2023).

Figura 14.

Respuesta del código “Función gamma Skimage” con la función “exposure.adjust_gamma”



Nota: Imagen original y con el filtro exposure.adjust_gamma(), utilizando un parámetro gamma=1.8. Creado por los autores.

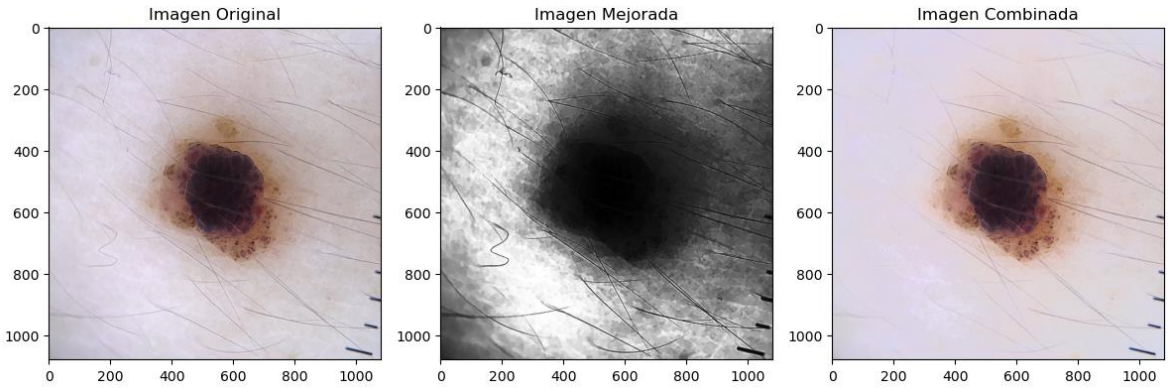
RESULTADO: Al variar el parámetro con valores menores a 1 no se observan cambios notorios. Sin embargo, para valores superiores a 1 los colores toman un tono intenso, por lo que no se puede detallar muy bien la pigmentación. Al ajustar el parámetro gamma en

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. (scikit-image, 2023)

Figura 15.

Respuesta del código "Función gamma Skimage" con la función "exposure.equalize_hist"



Nota: Imagen Original, escala de grises para ecualizar y la combinación de estas dos imágenes ponderando cada una con un factor Alpha=1.2, Creado por los autores.

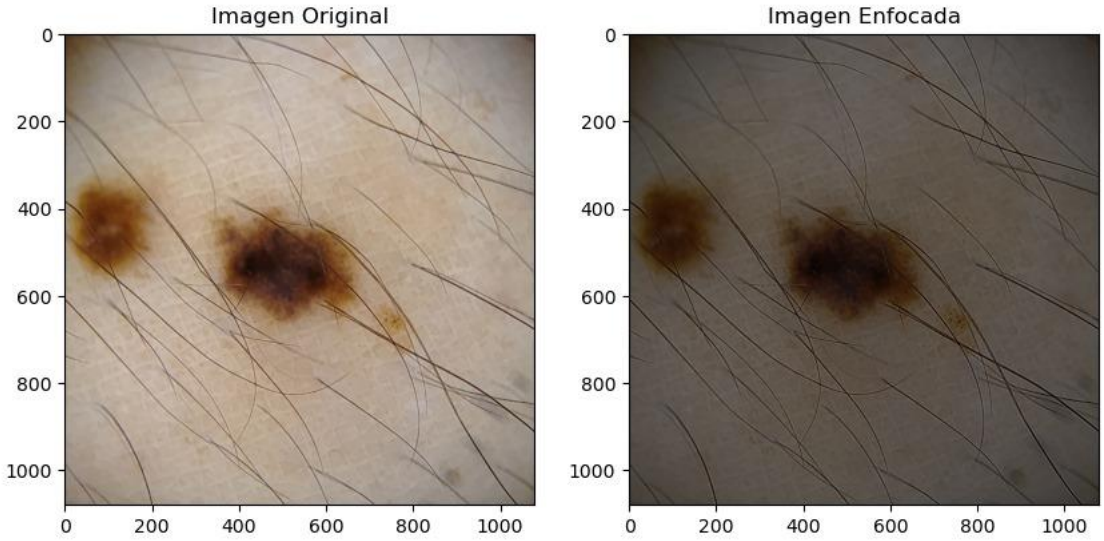
RESULTADO: Al observar la imagen final se resaltan los colores oscuros, sin embargo, la estructura de la piel y vellos se ven borrosos, como se aprecia en la parte izquierda de la imagen combinada, esto puede ser semejante al efecto del maquillaje.

3. BIBLIOTECA 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 sintetizar 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. (OpenCV, 2023).

Figura 16.

*Respuesta del código "Mascara de desenfoque" con la función "*cv2.GaussianBlur*" utilizando un parámetro de desviación estándar en 1*.

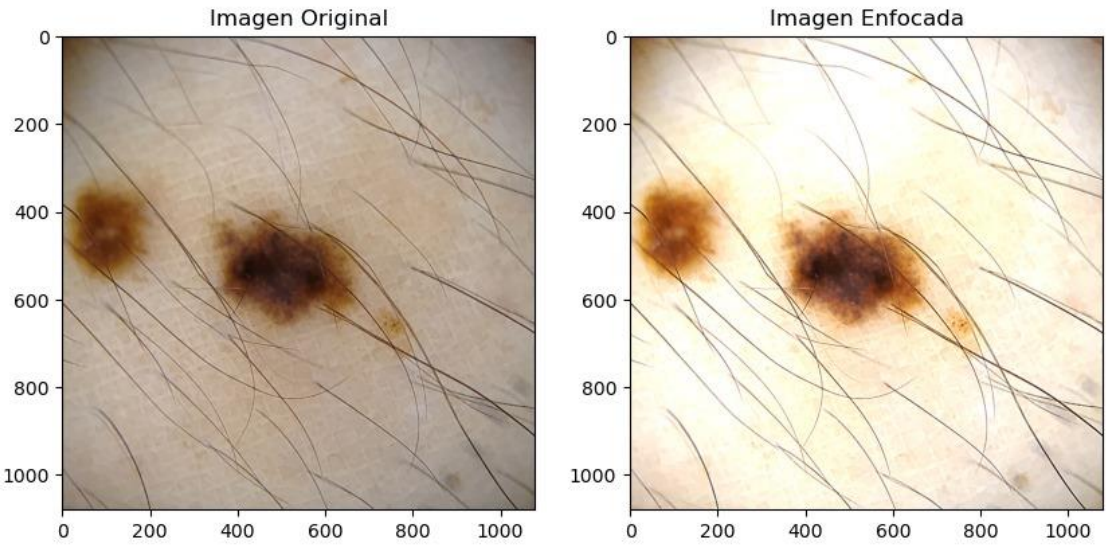


Nota: Imagen Original, y enfocada con una desviación estándar en 1, creado por los autores.

RESULTADO: El parámetro afecta el brillo de la imagen original, con una desviación estándar de 1, toda la imagen se oscurece y se pierden los detalles de la lesión, ya que no se perciben las diferentes tonalidades de pigmentación en la piel.

Figura 17.

*Respuesta del código "Mascara de desenfoque" con la función "*cv2.GaussianBlur*" utilizando un parámetro de desviación estándar en 2.*



Nota: Imagen Original, y enfocada con una desviación estándar en 2. Creado por los autores.

RESULTADO: Con un parámetro de 2 el brillo de la imagen aumenta, por lo cual el color original de la piel se altera, tomando un tono más pálido.

3.2 **Corrección de exposición y balance de color "COVERTSCALEABS**": "cv2.convertScaleAbs(imagen, alpha, beta)" Aplica una transformación lineal a cada uno de los píxeles de la imagen, escalando y ajustando los valores de pixeles mediante el

contraste (para un valor de variable **Alpha** mayor a 1 aumenta, mientras que en un rango de [0-1]disminuye) y el brillo (para un valor de variable **beta** positivo aumenta brillo, y para un valor negativo lo disminuye). (OpenCV, 2023)

Figura 18.

Respuesta del código "Corrección de Exposición y Balance de Color" con la función "cv2.convertScaleAbs".

Nota: Imagen Original, y corregida con la combinación de parámetros Alpha=0.1 y Beta=0.8, Creado por los autores.

RESULTADO: Se variaron los parámetros, para determinar que los valores apropiados era un Alpha=0.1 beta=0.8, ya que el valor de 0.1 aumenta el brillo ligeramente de la imagen y con un valor de 0.8 esta presenta un aumento significativo en la diferencia entre los tonos.

Sin embargo, la imagen obtenida aplicando estos parámetros pierde por completo las características de la lesión.

4. BIBLIOTECA OPEN CV

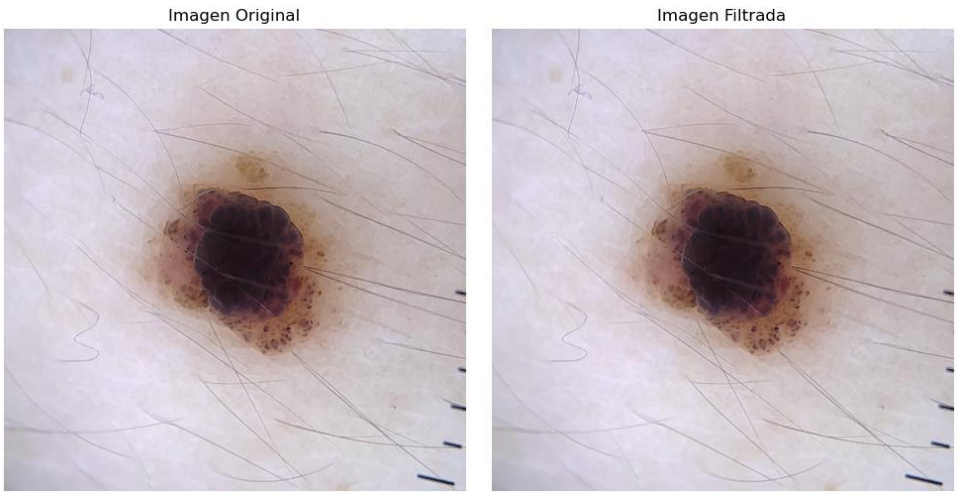
4.1. Filtro bilateral: La función “cv2.bilateralFilter” se utiliza para suavizar una imagen mientras se mantiene los bordes. El filtrado bilateral es el resultado de la combinación entre el dominio y rango, es decir, reemplaza el valor de píxel con un promedio de valores de píxel similares y cercanos. (Biteral filtering, 2023)

sigma_color: Indica el valor de desviación estándar para los componentes y espacio de color. Este parámetro afecta la similitud de color, ya que, al tener un valor alto, el rango de pixeles considerados como similares será mayor, razón por la cual se mezclarán más colores dentro de la vecindad de píxeles y dará como resultado áreas más grandes de color equivalente. (OpenCV, 2023)

sigma_space: Este parámetro afecta la similitud espacial o espacio de coordenadas, un valor mayor en este parámetro significa que los píxeles más alejados se influirán entre sí siempre y cuando sus colores estén lo suficientemente cerca. (OpenCV, 2023).

Figura 19 .

Respuesta del código “Filtro bilateral” con la función “cv2.bilateralFilter”.



Nota: Imagen Original, y con aplicación de filtro bilateral. Creado por autores.

RESULTADO: Al aplicar el filtro bilateral no se observan cambios en la nitidez y color de la imagen, pero se puede apreciar que la zona media de la lesión se aclara. Aun así la selección de parámetros (sigma_space y sigma_color) no cumplió con las expectativas esperadas.

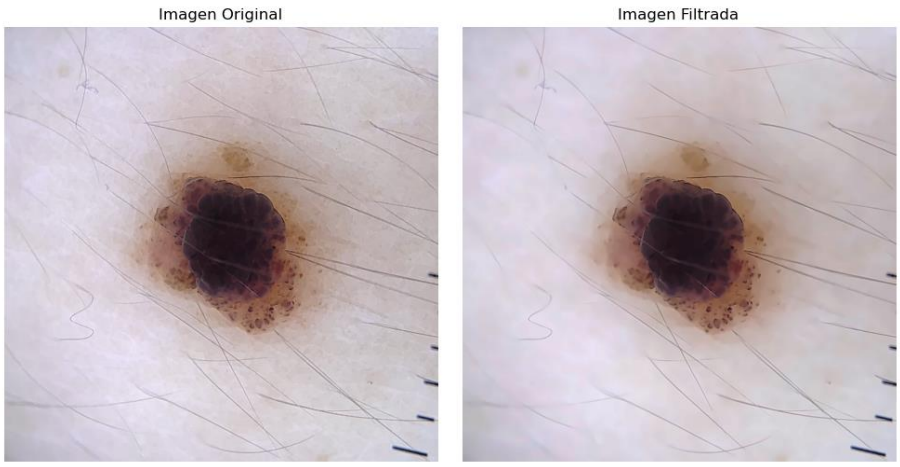
Este filtro es útil en situaciones donde otros métodos de filtrado pueden suavizar demasiado los detalles.

4.2. Filtro no local

"cv2.fastNlMeansDenoisingColored(src, None, h, hColor)": Es utilizado para eliminar el ruido de una imagen a color mediante un método no local, por lo que se mantiene los detalles y los bordes. Lo que hace esta función es el promedio ponderado del píxel en función de la similitud de su vecindario. A diferencia de algunos métodos de suavizado que solo consideran la proximidad espacial, este método también tiene en cuenta la similitud de color. (OpenCV, 2023).

Figura 20.

Respuesta del código "Filtro no local" con la función "cv2.fastNlMeansDenoisingColored"



Nota: Imagen original y con filtro no local. Creado por los autores.

RESULTADO: Al implementar el filtro no local se observa que existe alteraciones en la estructura de la piel, ya que al intentar eliminar ruido hace que se pierda nitidez en los detalles de la imagen como se evidencia en la figura 20.

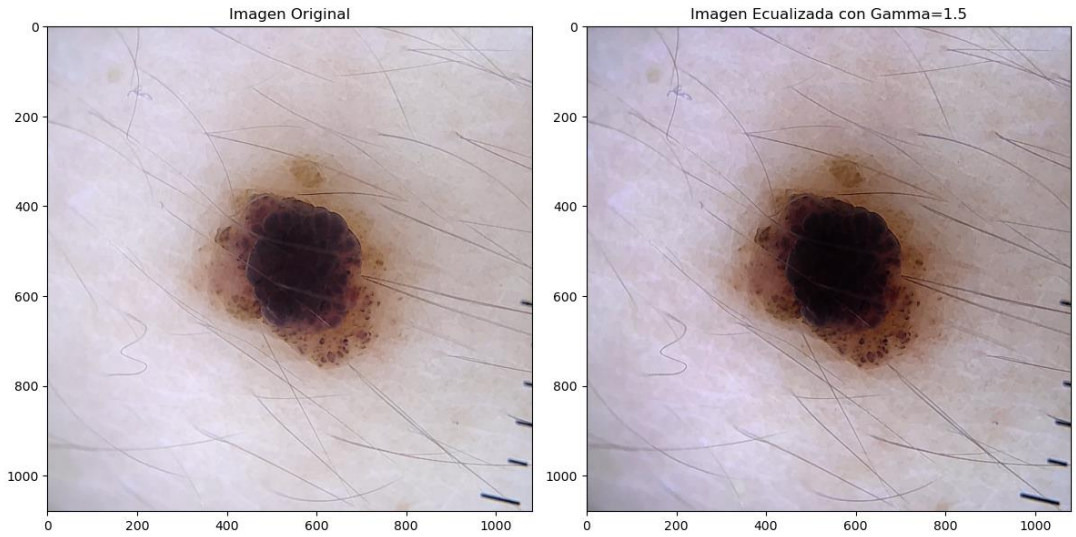
4.3 **Ecualización con parámetro gamma**

El filtro “ecualizar_con_gamma(imagen, gamma)” se basa en la corrección gamma, la cual es una operación no lineal que se utiliza para ajustar el brillo y el contraste de la imagen.

El parámetro gamma es el que controla la corrección, para valores de gamma mayores a 1 aumentan el contraste y la saturación, lo opuesto sucede con valores menores a 1. (Gonzales R., 2018).

Figura 21 .

*Respuesta del código "Ecualización con parametro gramma con Numpy y OpenCV" con la función "ecualizar_con_gamma"***.**



Nota: Ecualización con parámetros Gamma=1,5. Creado por los autores.

RESULTADO: Tras implementar el filtro de ecualización con parámetro Gamma se percibe un incremento de tono en las áreas oscuras, lo que hace perder la estructura interna de la lesión e intensificar las sombras o ruido de la imagen. A pesar de ello, se avivan los colores, lo que permite que los bordes se resalten.

4.4 FILTROS CON KERNELLS

FILTER 2D: El filtro cv2.filter2D en OpenCV realiza una convolución 2D en la imagen de entrada con un kernel específico. La convolución es un proceso matemático que combina dos funciones para producir una tercera, que generalmente representa una modificación de la primera función. Cada píxel en la imagen de salida es calculado como una combinación lineal de los píxeles vecinos en la imagen original, ponderados por los valores del kernel.

En el contexto del procesamiento de imágenes, la convolución se utiliza comúnmente para aplicar efectos como realce de características, suavizado, detección de bordes y agudización. (OpenCV, 2023).

4.4.1 AGUDIZACIÓN: La función "cv2.filter2D(imagen_rgb, -1, kernel_agudizacion)" aplica el filtro de agudización a la imagen utilizando la convolución 2D. El parámetro -1 indica que la salida debe tener la misma profundidad que la imagen de entrada.

El kernel seleccionado para la convolución es el siguiente:

Figura 22.

Filtro Sharpening

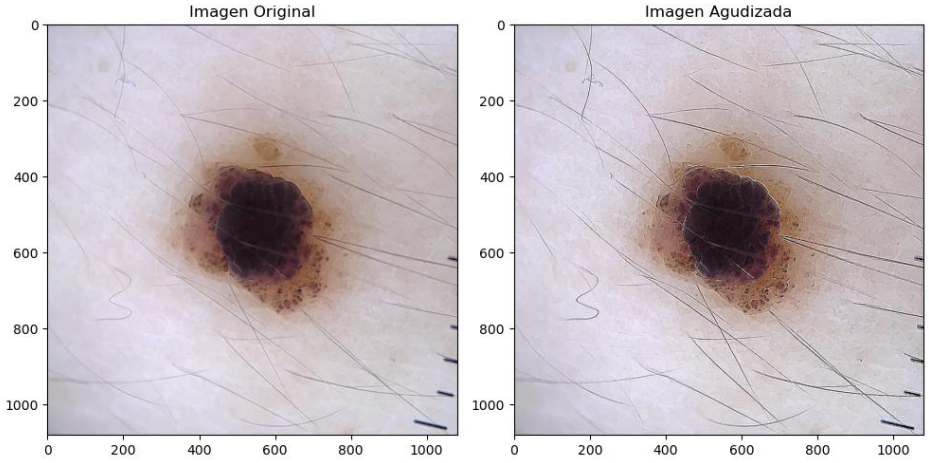
$$\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Nota: Tomada de OpenCV-3-computer visión with Python, (Chapter 2: Detecting Edges and Applying Image Filters, Pag.62), por (Prateek, 2015) .

El filtro de agudización utiliza el kernel mostrado en la figura 22, resalta los detalles y bordes en la imagen, haciendo que estos elementos sean más pronunciados y nítidos, ya que se les asigna un mayor peso en comparación con los píxeles circundantes. Este tipo de filtro es comúnmente utilizado para mejorar la definición de los objetos en una imagen.

Figura 23 .

Respuesta del código "AGUDIZACIÓN" con kernell o filtro sharpening



Nota: Imagen original y con un Kernel de agudización. Creado por los autores.

RESULTADO: En la aplicación de este filtro se puede apreciar una mejora en la imagen, ya que revela características que no eran perceptibles a simple vista en la versión original. Por ejemplo, aparece un borde blanco en la zona superior de la lesión.

4.4.2 REALCE CARACTERISTICAS

El kernel seleccionado para realzar características.

Figura 24.

Filtro Realce de características

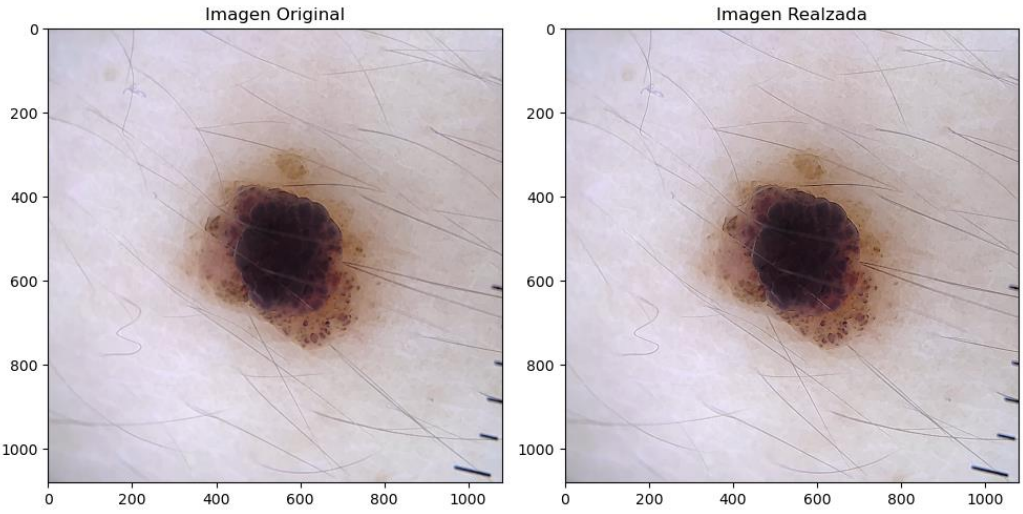
$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Nota: Tomada de *Aplicación de la convolución de matrices al filtrado de imágenes* (convolución de filtros, p.101), por (Gimenez et al., 2016) .

El filtro de realce de características utiliza el kernel mostrado en la figura 24, el cual se utiliza para realzar las características de la imagen, destacando ciertos detalles y bordes. En este caso, el kernel tiene un valor central alto en (5) y valores negativos alrededor, lo que enfatizará las características y resaltará detalles importantes de la imagen.

Figura 25 .

Respuesta del código "REALCE DE CARACTERISTICAS" con kernell o filtro específico para realzar características.



Nota: Imagen original y con un Kernel de realzado. Creado por los autores.

RESULTADO: A través del uso de este filtro, se logra ver un cambio en la tonalidad de la imagen, donde aumenta la nitidez y se realzan las características de la imagen principalmente en sus bordes y texturas.

4.4.3 SUAVIZADO:

El kernel seleccionado para suavizado es:

Figura 26.

Filtro Blurring

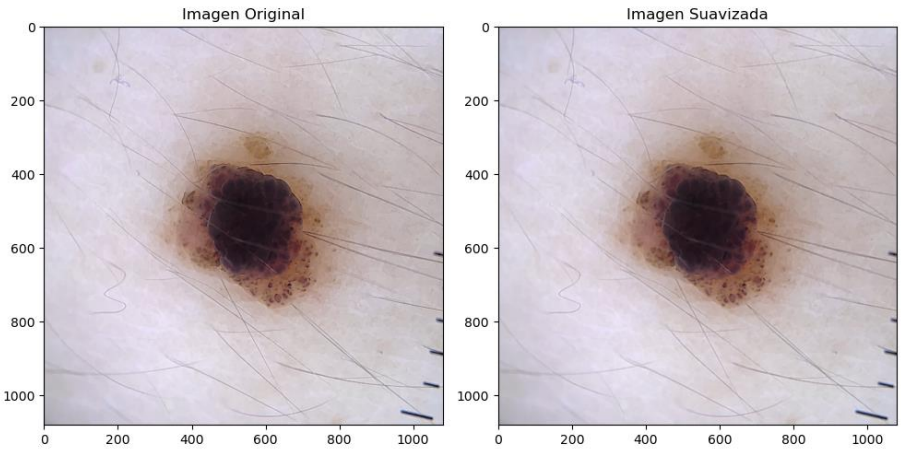
$$1/9 \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Nota: Tomada de OpenCV-3-computer visión with Python, (Chapter 2: Detecting Edges and Applying Image Filters, Pag.54), por (Prateek, 2015) .

Cada elemento del kernel tiene un valor de 1, y luego se normaliza dividiendo por 9, de modo que la suma de los elementos sea 1. El filtro de suavizado utilizado es un filtro de promedio, que tiene el efecto de suavizar la imagen al promediar los valores de los píxeles en una región, lo que hace que se reduzca el ruido.

Figura 27.

Respuesta del código "SUAVIZADO" con kernell o filtro Blurring



Nota: Imagen original y con un Kernel de suavizado. Creado por los autores.

RESULTADO: Mediante el empleo del filtro con kernel de suavizado, la imagen se difumina uniformemente sin alterar su color o bordes, esto es debido a que se promedian los pixeles y se obtiene un leve suavizado en los detalles de la lesión.