

1)

Añade el efecto sal y pimienta a la imagen seleccionando aleatoriamente y reemplazando 2,5% de los píxeles negros y 2,5% píxeles blancos. Este comando solo funciona con imágenes de 8 bits.

En este caso tenemos 550x413 píxeles que son un total de = 227150 píxeles . Si un 2,5% de los píxeles negros se ven afectados entonces $(2,5\% \times 227150 \text{ píxeles}) / 100\% = 5.678,75$ píxeles negros , y 5.678,75 píxeles blancos.

2)

Pienso que el más adecuado para eliminar el ruido sal y pimienta de la foto es la mediana porque dado que el tamaño del kernel siempre es impar en la mediana (3x3, 5x5) el resultado de ésta siempre será directamente el valor de uno de los píxeles del conjunto de datos por lo que no será necesario sumar dos valores centrales y dividirlos que es lo que pasaría si fueran pares.

La ventaja de este filtro viene dada por su característica de desprestigiar valores extremos, ya que tenemos que ordenar nuestros datos de menos a mayor y cogemos el dato de enmedio, lo cual para el caso de una imagen con ruido de sal y pimienta (0 y 255) que son extremos de escala se pueden eliminar mediante este procedimiento. Y así sustituir los valores de cada píxel por la mediana de los vecinos.

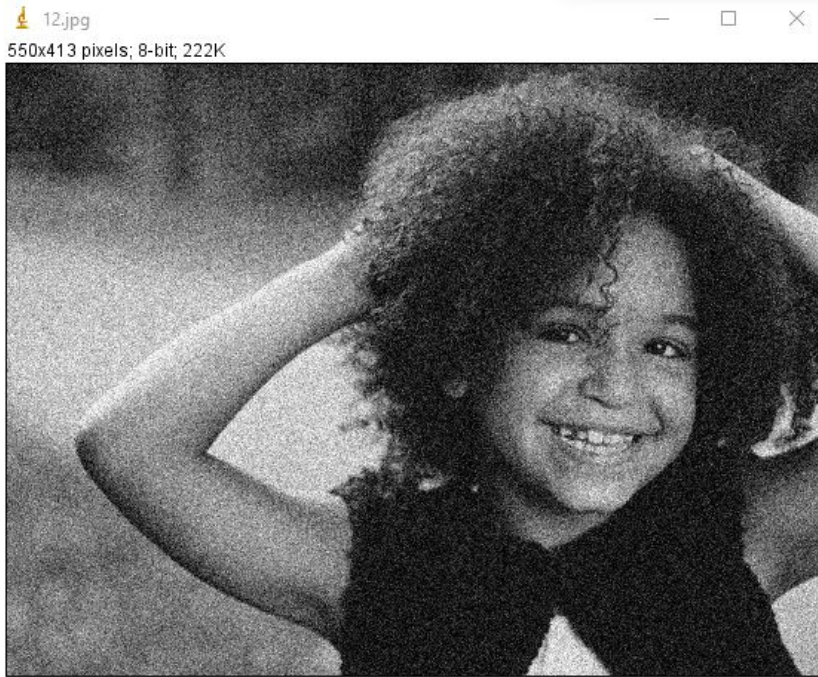
En cambio si hubiéramos cogido la media del resultado de suavizar los valores de los vecinos de cada píxel no hubiera sido tan bueno ya que valores como el 255 o el 0 los hubiera cogido para realizar la media y por tanto eliminaría peor el efecto de ruido de sal y pimienta.

El filtro Gaussiano es similar al de la media también coge los valores 255 y 0 , y suaviza más la imagen. Según el valor de su constante sigma si la aumento , el máximo de la gráfica de la campana de Gauss bajará por lo que son ms iguales todas las probabilidades que antes eran muy poco probables de ocurrir, ocurren. Sobre todos en los extremos

Tampoco con el efecto de máximo y mínimo no hubiéramos quitado el ruido , porque son ideales para quitar el ruido pimienta(máximo) y el ruido sal(mínimo) , respectivamente.

Como podemos apreciar en las siguientes imagenes el filtro de la mediana quita mejor el ruido que el filtro de Gauss , aunque éste último las emborrona más.

Imagen con ruido sal y pimienta:



Resultado de aplicar máscaras 3x3 y 5x5 con el filtro de la mediana respectivamente:

12.jpg



550x413 pixels; 8-bit; 222K



12.jpg



550x413 pixels; 8-bit; 222K



Resultado de aplicar $\sigma=3$ y $\sigma=5$ al filtro de Gauss , respectivamente:



2)

Al realizar la función de Laplaciano no se reescalan los valores a $[0,255]$, puesto que como entrada la hemos tratado la profundidad con 16 bits de enteros con signo para evitar el desbordamiento , por tanto los valores que eran menores que cero , los hemos tratado como números negativos. Lo que pasará con esto es que la imagen que obtendremos estará en gris completamente , osea no veremos nada. Porque lo que hemos hecho es calcular el Laplaciano de

toda la imagen en 16 bits con entero , y luego transformarla a 8 bits sin signo(0..255) , es decir uint8 , para que nos muestre el cambio que ha realizado la imagen y así tratar los valores negativos y positivos que se pasen de -255(y menores) y de +255, como valores absolutos,consiguiendo así que estos valores se reescalen a 255, osea que la mayor parte de la imagen será en negro si aplicamos máscaras grandes. Con esto lo que conseguimos es un negativo de la imagen.

Imagen en escala de grises.

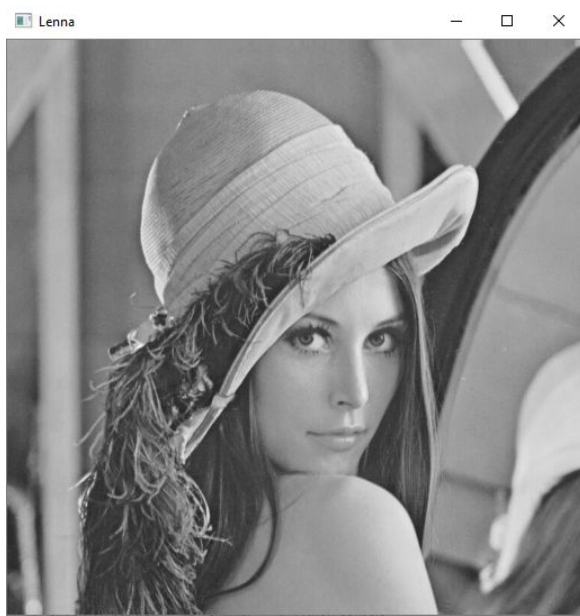


Imagen al aplicarle un kernel = 3 en el algoritmo Laplaciano.

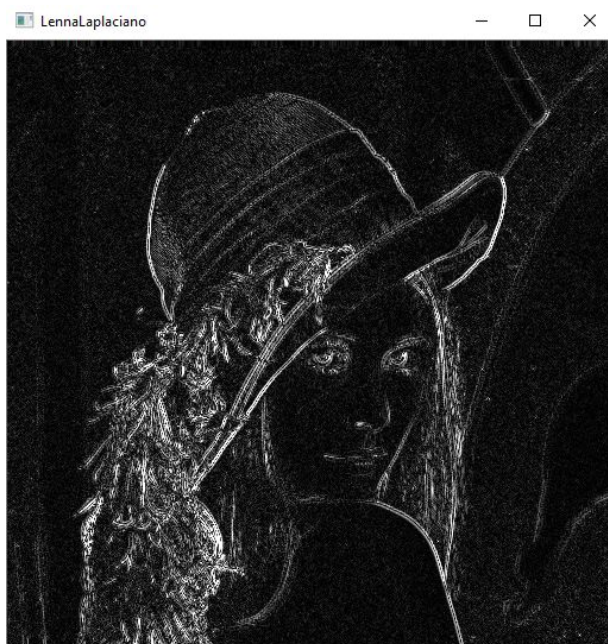


Imagen el aplicarle un kernel = 5 en el algoritmo Laplaciano.

