



Facultad de Ingeniería
Universidad de Buenos Aires
Trabajos Final: Filtrado de Imágenes y Banco de Filtros.

2^{er} Cuatrimestre, 2019

Alejandro, Pablo Martín	98021	pabloale96@gmail.com
-------------------------	-------	----------------------

1. Introducción

El presente trabajo practico consiste en tomar la imagen patrón *lenna.bmp* (1) y aplicar diferentes filtros a las imágenes para obtener diferentes resultados, luego se utiliza un banco de filtro para mostrar las aplicaciones que tiene sobre el filtrado de las imágenes.

Se mostraran los resultados obtenidos de los diferentes filtro. La idea del filtrado de imagen consiste que a cada sub-matriz de pixeles de la imagen de $N \times N$ se la multiplique por una matriz de $N \times N$ obteniendo así una matriz con combinación de cada pixel. Matemáticamente se puede expresar como

$$\begin{bmatrix} p_1 & \dots & p_N \\ \vdots & \ddots & \vdots \\ p_N & \dots & p_N \end{bmatrix} \times \begin{bmatrix} a_1 & \dots & a_N \\ \vdots & \ddots & \vdots \\ a_N & \dots & a_N \end{bmatrix} = \begin{bmatrix} b_1 & \dots & b_N \\ \vdots & \ddots & \vdots \\ b_N & \dots & b_N \end{bmatrix} \quad (1)$$

$$\mathbf{P} \times \mathbf{A} = \mathbf{B} \quad (2)$$

en donde cada b_i sera una combinación lineal entre el pixel p_i y los coeficientes a_i . La matriz \mathbf{A} que es generada por los coeficientes a_i , termina siendo el filtro que se aplica, mientras que la matriz \mathbf{P} seria la matriz de los pixeles. En este trabajo usaremos diferentes tipos de filtros y mostraremos la imagen resultante al aplicar el filtro, ademas calcularemos el histograma de cada uno de los filtros. Para terminar, se usara un banco de filtro para mostrar aplicaciones que se pueden tener sobre los diferentes imágenes.



Figura 1: Imagen resultante en escala de grises.

2. Desarrollo

2.1. Filtros

Para simplificar las cuentas tomaremos filtros de 3×3 , usaremos matrices \mathbf{A} filtro conocidos, como entre ellos *Laplace* que transforman a la imagen en diferentes aspectos. Explicaremos con mas detalle cada filtro.

2.1.1. Blur

El filtro *Blur* consiste en un suavizado a la imagen, toma los pixeles y multiplica cada pixel por valores a_i . Generalmente los coeficientes a_i son generados por una normal multivariable de dos dimensiones por lo que la

matriz del filtro es simétrica. Debido a esto ultimo, tambien es conocido este filtro como *Gaussian Blur*. Para la imagen resultante, se usa la siguiente matriz \mathbf{A} :

$$\mathbf{A} = \begin{bmatrix} 0,0625 & 0,125 & 0,0625 \\ 0,125 & 0,25 & 0,125 \\ 0,0625 & 0,125 & 0,0625 \end{bmatrix} \quad (3)$$

al aplicar el filtro se puede obtener la imagen (2) donde se observa que la imagen es mas borrosa que la original. Se puede entender mirando la diferencia entre los histogramas de ambas imágenes, el filtro blur contiene la energía de la imagen mas distribuida entre todo el rango de los pixeles generando un histograma mas continuo, al tener la energía mas distribuida genera que la intensidad de cada pixel (altura del histograma) sea mas baja provocando un efecto borroso.



Figura 2: Comparación de las imágenes.

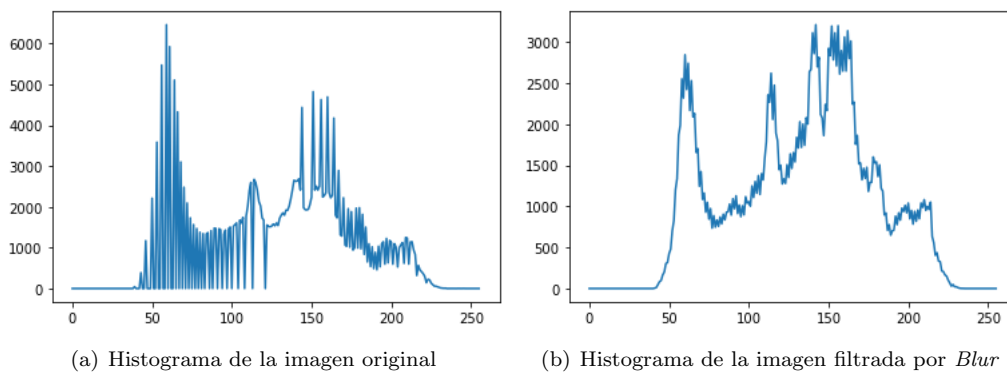


Figura 3: Comparación de los histogramas de las imágenes.

2.2. Sobel

El filtro de *Sobel* se usa principalmente para la detección de bordes. Como vimos en los trabajos prácticos, esto va a depender mucho del recorrido entonces puedo definir diferentes forma la matriz de *Sobel* para obtener los mismo resultados. Las matrices de *Sobel* mas comunes son *bottom*, *top*, *left* y *right*. La matriz de *Sobel* proviene del gradiente de la función intensidad de una imagen.

2.2.1. Bottom Sobel

El *Bottom Sobel* consiste sobre los pixeles que se le aplica el gradiente de la función de intensidad. La matriz \mathbf{A} para el *Bottom Sobel* viene dada por

$$\mathbf{A} = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (4)$$

en donde se observa que para algunos pixeles se le aplica el gradiente de la función intensidad mientras que en otros se anula la intensidad. Se considera *Bottom Sobel* sobre donde se encuentra en gradiente de la función intensidad.

2.2.2. Top Sobel

Al igual que *Bottom Sobel* consiste en donde se encuentra el gradiente obteniendo la matriz \mathbf{A}

$$\mathbf{A} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad (5)$$

se puede observar que es la matriz \mathbf{A} de *Bottom Sobel* pero multiplicada por un -1. Se debe a que se esta aplicando el mismo metodo solo que el gradiente se aplica a diferentes pixeles.

2.2.3. Left Sobel

El *Left Sobel* sigue aplicando el gradiente de la función de intensidad, solo que el gradiente se encuentra a la izquierda de la matriz \mathbf{A} como se observa

$$\mathbf{A} = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \quad (6)$$

2.2.4. Right Sobel

Por ultimo, esta el método *Right Sobel* que consiste en aplicar el gradiente a la derecha, se puede obtener la matriz \mathbf{A} multiplicando a la matriz \mathbf{A} de *Left Sobel* por -1 obteniendo

$$\mathbf{A} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad (7)$$

2.2.5. Imágenes sobre el método Sobel

En la figura (14), se puede observar las imágenes resultantes con un filtro de *Sobel*. Al mirar la diferencia entre las distintas matrices \mathbf{A} sobre todos los metodos *Sobel* se puede observar que hay contornos que se muestran mejor que otros que va dependiendo en donde se aplico el gradiente. En este método no se calculo el histograma porque obtendríamos una delta sobre los pixeles de menor valor debido a que la imagen es negra salvo por el contorno.

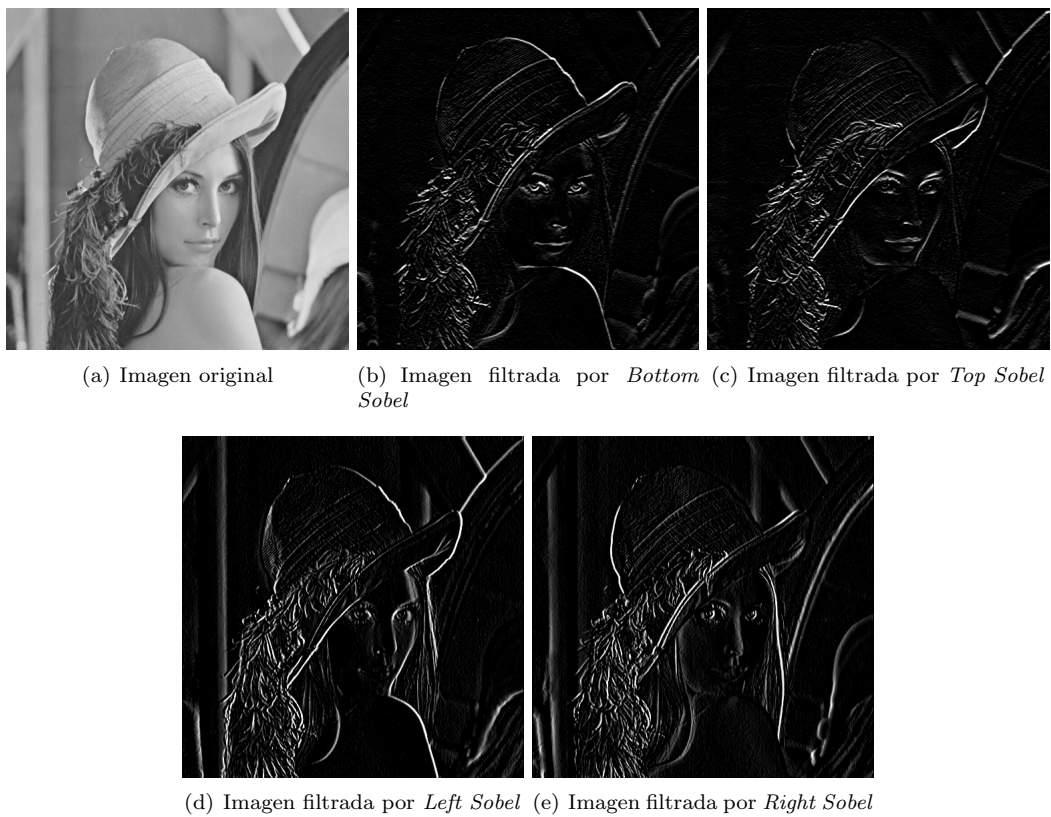


Figura 4: Comparación de las imágenes resultante.

2.3. Emboss

El filtro *Emboos* es parecido al filtro *Sobel* pero la diferencia que da ilusión de profundidad por hacer énfasis sobre los diferentes pixeles en una dirección. El filtro reemplaza cada pixel de la imagen original por un realce o una sombra sobre el pixel dependiendo de la imagen original. La matriz \mathbf{A} se obtiene a partir de

$$\mathbf{A} = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}. \quad (8)$$

La imagen resultante termina siendo los contornos de la imagen original. A diferencia de la técnica *Sobel* se tiene que los contornos se encuentra en un fondo blanco por lo que el histograma daría una delta en los pixeles mas altos por lo que no se calculo al igual que en el *Sobel*. La imagen resultante termina siendo

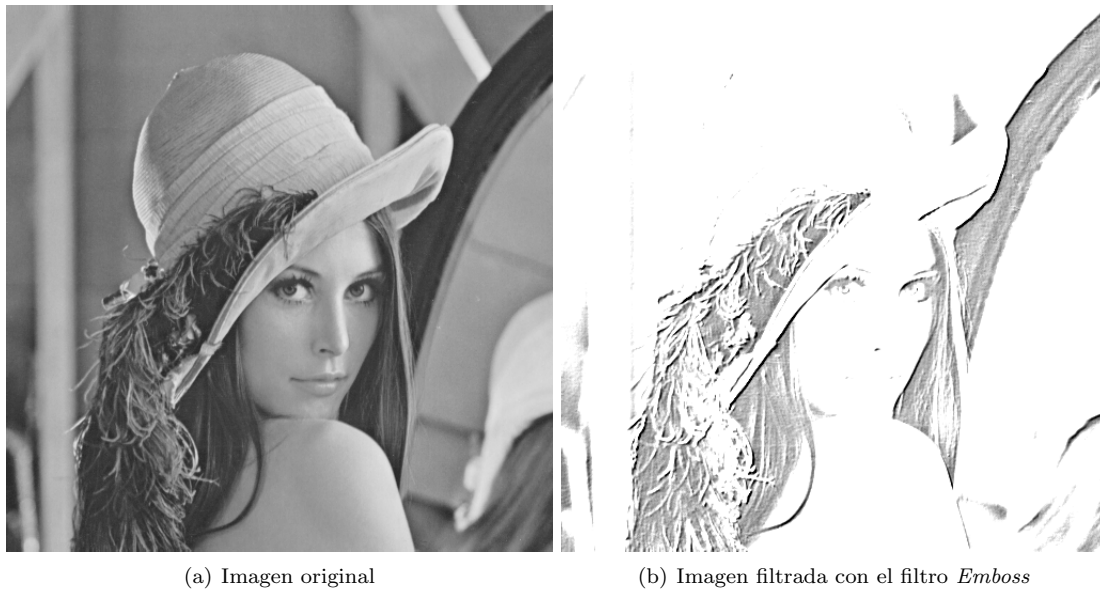


Figura 5: Comparación de las imágenes.

2.4. Outline

Un método para realce de la diferencia de los pixeles se termina usando el filtro *Outline*. Consiste en ver los pixeles vecinos con lo mas cercanos con la misma intensidad. Obtiene una imagen de apariencia negra porque toma diferencia entre pixeles (valor demasiado chico salvo en un contorno) por lo que no se calcula el histograma. La matriz de **A** termina siendo

$$\mathbf{A} = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}. \quad (9)$$

donde se observa que se toma la diferencia entre los pixeles vecinos. El valor de 8 en el pixel del medio viene dado que estoy tomando los 8 pixeles vecinos a ese pixel. Entonces el resultado obtenido es



Figura 6: Comparación de las imágenes.

2.5. Laplace

El metodo de *Laplace* es parecido al metodo de *Outline*, vuelve a tomar la diferencia entre los pixeles vecinos pero con la diferencia que toma los pixeles que se encuentran arriba, abajo, derecha e izquierda del pixel, poniendo la diagonales con un valor de intensidad de 0. Representando eso en una matriz se obtendria la matriz \mathbf{A} como

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (10)$$

donde el valor de 4 corresponde a los 4 vecinos que estoy tomando sobre el pixel central. Entonces, la imagen obtenida sera parecida a la imagen conseguida con *Outline* pero con menos detalles por la cantidad de direcciones que se tomaron, obteniendo



(a) Imagen original

(b) Imagen filtrada con el filtro *Laplace*

Figura 7: Comparación de las imágenes.

2.6. Sharpen

Al igual que *Laplace* el metodo de *Sharpen* toma la diferencia entre los valores adyacente y no los diagonales, la diferencia entre *Laplace* viene dada por la intensidad que se toma del pixel central, se tuma una cantidad mayor dando una imagen mas viva que en *Laplace*, esto se podria entender a que estas tomando un poco mas de energia en el pixel central por las diferencias que estas tomando. La matriz \mathbf{A} se calcula como

$$\mathbf{A} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}. \quad (11)$$

este metodo se hizo para demostrar que cambiando el valor del pixel central del filtro se puede tener otra imagen muy diferente. En la figura (8), se observa la diferencia entre la imagen original y la imagen generada por *Sharpen* notandose que esta ultima es mas viva a comparacion de la original, esto se debe que el histograma de sharpen tiene una distribución diferente que la original esto se observa en la figura (9)



Figura 8: Comparación de las imágenes.

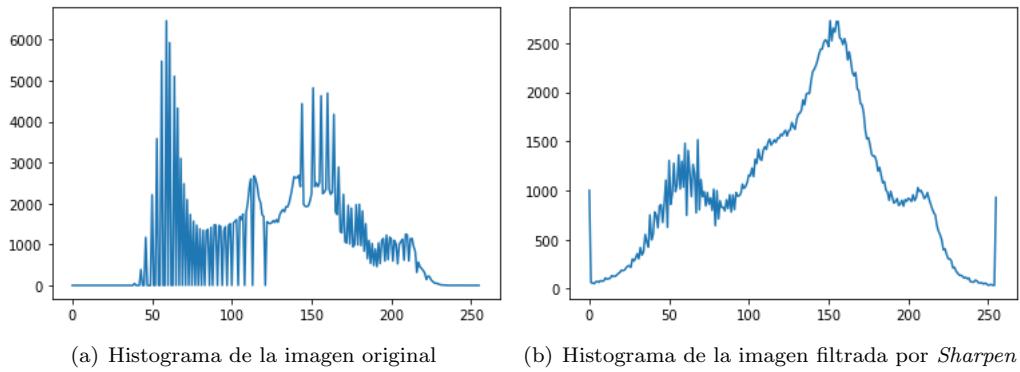


Figura 9: Comparación de los histogramas de las imágenes.

2.7. Kernel

Por ultimo, el metodo de *Kernel* se usa para difocar la imagen, tomando la suma de los pixeles vecinos pero dividiendo la intensidad por un numero mayor a 1 asi se reduce la intensidad de cada pixel. En forma de matriz \mathbf{A} se expresa como

$$\mathbf{A} = \frac{1}{25} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}. \quad (12)$$

mostrando que cada pixel es dividido por un numero mayor a 1, logrando la apariencia de difocado. La imagen resultante comparada con la imagen original se puede observar en la figura (10). El difocado se observa en el histograma como la reducción de la intensidad del mismo. Los resultados se observan a continuación



Figura 10: Comparación de las imágenes.

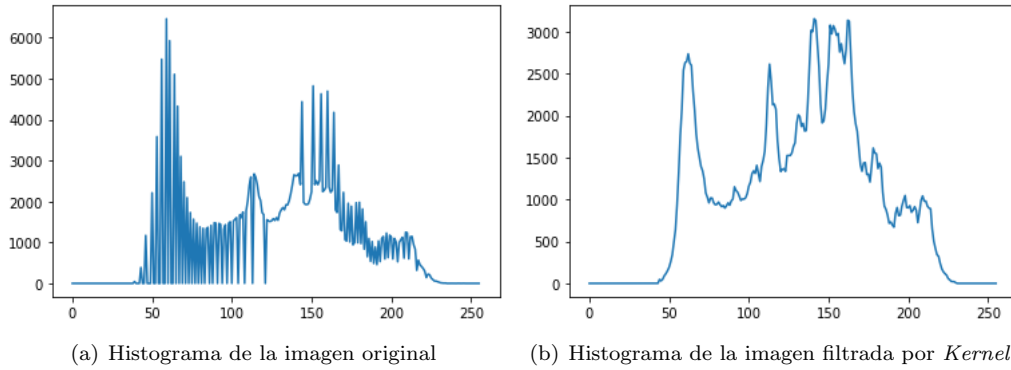


Figura 11: Comparación de los histogramas de las imágenes.

3. Banco de Filtro

En la figura (12), se observa el sistema tradicional del banco de filtro, en donde se aplica a una señal $x[n]$ se le aplica N filtros. Los filtros H sirve para quedarnos con los $M \times M$ pixel de la imagen, por lo que termina siendo una matriz \mathbf{H} igual

$$H = \begin{bmatrix} 1 & \dots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \dots & 1 \end{bmatrix} \quad (13)$$

de $M \times M$, y cada filtro se le aplica a una parte diferente de la imagen, esto seria parecido a un zoom aplicado anteriormente. Por ende, para realizar un zoom a una imagen con banco de filtro, consiste en ir aplicando banco de filtros consecutivos hasta obtener el zoom deseado ya que me va *Downsapling* la señal filtrada. En una imagen, se tendría que la imagen original de 512×512 se divide en 4 imágenes de 256×256 (quedándose con el zoom de cada región) y luego se vuelve a dividir en 4 dando 8 imágenes de 128×128 (banco de filtro consecutivo) y así hasta obtener el zoom deseado.

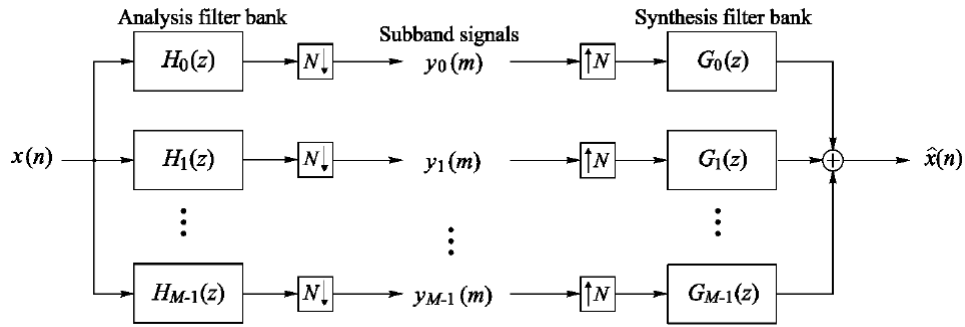


Figura 12: Sistema del banco de filtro.

En este trabajo se realizó una división de sub bandas, tomadas de a 2 para mostrar la continuidad de aplicar banco de filtros. También, se usó un banco de filtro simple con $N = 4$ y se le aplicó a cada sub señal un filtro diferente para mostrar las aplicaciones de banco de filtro.

3.0.1. Sub-Bandas

La figura (13), muestra el sistema de sub bandas usando banco de filtro, en donde se observa que se va dividiendo la imagen en regiones de 2 hasta un límite K . La idea de realizar subbanding es mostrar el proceso que realiza bancos de filtros consecutivos. La imagen (14(a)) muestra la imagen original mientras que (14(b)) muestra la división hecha con los anchos de filtros consecutivos. Por último, se tienen las imágenes (14(c)) y (14(d)) corresponden a la imagen aplicando el banco de filtro de la figura (13) pero con una codificación de 0.2 bits por pixel 0.1 bits por pixel.

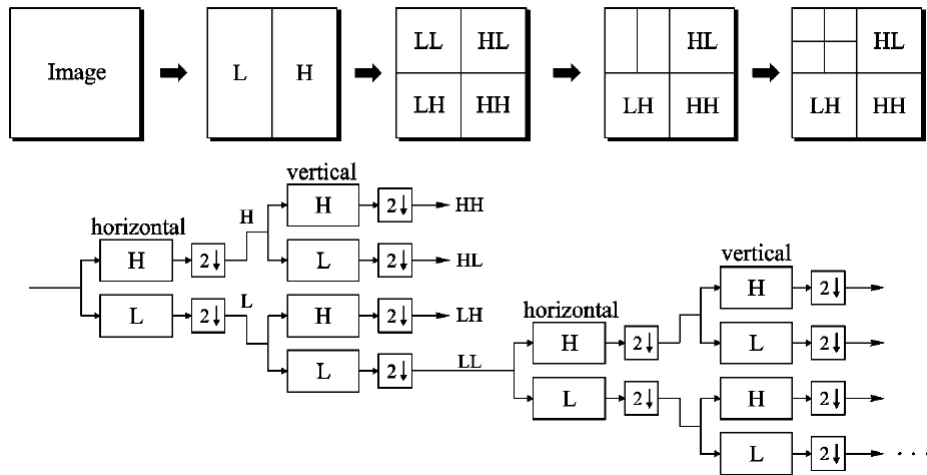


Figura 13: Sistema usado para la división de sub bandas..



Figura 14: Comparación de las imágenes resultante.

4. Filtrado por partes con banco de filtro

En esta sección, mostraremos el resultado de usar el banco de filtro en una aplicación concreta como en el filtrado de una imagen. Como vimos antes, un banco de filtro me permite trabajar con bandas de la señal de entrada permitiendo aplicar diferentes filtros en diferentes información de la señal de entrada. La figura (15) realiza esto mismo, divido la imagen original en cuatro bandas permitiendo filtrar cada sub imagen con un filtro distinto de los mencionados anteriormente. En la figura, la parte superior izquierda corresponde al filtrado con el método *Outline*, la parte superior derecha corresponde al método *Sharpen*, la parte inferior izquierda corresponde al método *Kernel* y la parte inferior derecha corresponde al método *Right Sobel*.



Figura 15: Imagen resultante de haber aplicado 4 filtro distintos con banco de filtros.

5. Conclusiones

Al realizar el trabajo se tubo que tener en cuenta tanta la imagen usada en formato con su respectivo header como los filtros a aplicar. Es importante destacar que cada filtro tiene su función particular para poder modificar la imagen permitiendo diferentes resultados.

Como se menciona es importante destacar los valores de la matrices \mathbf{A} ya que si fueran cualquier otro numero podría dar un resultado totalmente diferente.

Por ultimo, mencionar la uso de los banco de filtros para realizar diferentes aplicaciones sobre la señal de entrada, tanto como estimar como para realizar operaciones sobre ellas (anular frecuencias, borrar datos de poco interés, etc) permitiendo trabajar sobre diferentes bandas de la señal.