

# Laboratorio 1

## Procesamiento Digital de Señales

1<sup>st</sup> Juan José Restrepo Rosero  
Pontificia Universidad Javeriana  
Cali  
Cali, Colombia  
[juanjorestrepo@javerianacali.edu.co](mailto:juanjorestrepo@javerianacali.edu.co)

2<sup>nd</sup> Juan David Vargas Mazuera  
Pontificia Universidad Javeriana  
Cali  
Cali, Colombia  
[juandavid2001@javerianacali.edu.co](mailto:juandavid2001@javerianacali.edu.co)

**Abstract**— In this laboratory report, an implementation of a layer of a convolutional network was carried out on an image of the faces of the members of the group using a Python function to calculate the 3D convolution between the color photograph of size (nh; nw; nc) and a 3D filter of fh x fw x nc. Different filters were then varied to highlight horizontal and diagonal edges of the image, to finally obtain 3 characteristic maps by means of a ReLU activation function. In addition, the SQNR of a signal for different numbers of quantization bits was statistically evaluated and analyzed.

**Keywords**— Convolution, characteristic map, kernel. Quantization.

### I. INTRODUCCIÓN

El desarrollo del laboratorio número 1 se centra en la creación de algoritmos y análisis de la función de Convolución 3D. En este caso se tiene una representación 3D de una imagen (capas RGB). Para lograr una detección de características de la imagen, como líneas horizontales, verticales, o diagonales, se implementa una función de convolución con un kernel diseñado especialmente para la detección de una o más de estas características.

Asimismo, se debe de tener en cuenta que para el desarrollo del laboratorio se evalúan los resultados del programa 3 y 6 según ABET, y para la creación de estos algoritmos, el laboratorio brinda las bases y características principales que la función debe de cumplir

### II. OBJETIVOS

1. Evaluar y analizar estadísticamente el SQNR de una señal para diferentes números de bits de cuantización.
2. Implementar una capa de una red convolucional para extraer características de una imagen.

### III. MARCO TEÓRICO

#### A. Correlación y autocorrelación

La correlación se puede definir como una operación entre dos señales que permite medir el grado de similitud que hay entre ellas, es decir, permite medir qué tan cercanas son dos señales mediante la convolución de estas. Una de las

aplicaciones más comunes de la correlación, es para obtener características de una señal por medio de otra que es conocida [1].

$$r_{xy} = x(l) * y(-l) \quad (1)$$

La ecuación que describe la relación entre la correlación de dos señales  $x$ , y su convolución. Teniendo claro el concepto de correlación, la autocorrelación se puede definir como la correlación de una señal con ella misma. La autocorrelación se usa principalmente para identificar el periodo en una señal que se puede encontrar afectada por ruido.

$$r_{xy} = x(l) * x(-l) \quad (2)$$

Un dato importante es que cuando  $l = 0$ , la correlación alcanza su máximo.

$$|r_{xy}(l)| \leq r_{xx}(0) = E_x \quad (3)$$

#### B. Cuantización

La cuantización es una de las etapas de la conversión analógica digital de una señal, encargada de representar la amplitud de las muestras de una señal analógica, en diferentes niveles, codificándolos acorde con el número de bits designados para esto, donde el primer bit es el signo. Un ejemplo de cuantización se puede ver de manera gráfica en la figura 1:

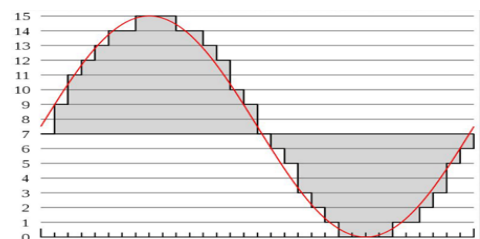


Figura. 1: Ejemplo de cuantización

#### C. Error de cuantización

El error de cuantización es la diferencia entre la señal cuantizada y la señal original, al cuantizar la señal se pierde un poco de la forma de la señal original, esto variaría dependiendo

de la cantidad de bits utilizados, con más bits, más niveles de cuantización y menor será el error de cuantización [8]. Se calcula utilizando la siguiente expresión y ver de manera ilustrativa en la figura 2:

$$e_q(n) = x(n) - x_q(n) \quad (3)$$

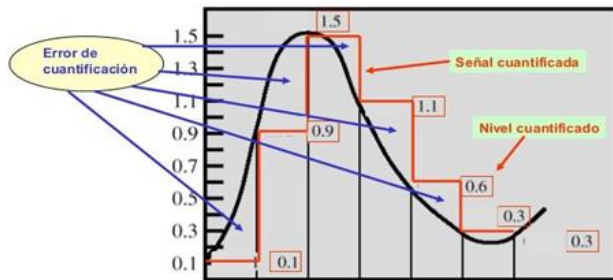


Figura. 2: Ilustración del error de cuantización

#### IV. PROCEDIMIENTO

El trabajo consiste en dos etapas. La primera consiste en procesar una imagen con el uso de filtros. La segunda consiste en evaluar estadísticamente el SQNR de una señal para diferentes números de bits de cuantización. A continuación, se mostrarán los procedimientos aplicados en cada uno de los casos.

##### A. Procesamiento de una imagen

Se busca procesar una imagen y detectar algunas de sus características, principalmente encontrar las líneas horizontales, verticales, y diagonales presentes en una imagen. Para este fin, se procesará la imagen usando un filtro para cada uno de los casos: líneas horizontales, verticales, y diagonales de 45 grados.

Inicialmente, se puede procesar una imagen usando un filtro por medio de la operación de la convolución. Para esto, primero usamos una función para encontrar la representación matricial de la imagen. Esta representación tiene códigos para cada píxel de la imagen. En este caso la imagen tiene 853 x 1280 píxeles. Además, al ser una imagen a color tiene 3 capas(R-G-B). En la figura 3 podemos ver la imagen usada:



Figura. 3. Imagen sin procesar

El filtro se centra en cada píxel y se multiplica cada valor del filtro por su valor correspondiente en la representación matricial de la imagen en cada una de sus capas. Al continuar con este proceso obtenemos una matriz que se llamará el mapa de características, el cuál tiene dos dimensiones, siendo estas

853 x 1280. Este proceso se realizará para cada uno de los 3 tipos de filtros. Además, se usarán 3 filtros distintos por cada tipo, dando un total de 9 pruebas para la realización de este trabajo. A continuación, se describirá el procedimiento usado paso a paso en el código del cuaderno de Jupyter:

1. Se obtiene la representación matricial en tres dimensiones de la imagen.
2. Se define la matriz del filtro.
3. Se redefine la matriz del filtro a una cuadrada del ancho del lado con más elementos.
4. Se redefine la matriz de la imagen, se agregan bordes dependiendo del tamaño del kernel, y se llenan con ceros para facilitar la convolución.
5. Se convoluciona la matriz redefinida con el kernel redefinido y se obtiene un mapa de características. Se obtiene la representación matricial en tres dimensiones de la imagen.
6. El mapa de características se le agrega un bias aleatorio.
7. El mapa de características se rectifica.

##### B. Evaluación del SQNR de una señal

Para el desarrollo de esta segunda actividad se desarrolló un código el cual lleva a cabo el proceso de cuantización de una señal determinada, de manera que se hizo uso del programa ScriptLab1PDS2022-Ilestudiantes.py. De esta manera, se creó una función llamada *my\_quantization* en la cual se introducen 3 parámetros de entrada los cuales son:

*A: amplitud*  
*x: señal*  
*b: número de bits a usar en la cuantización*

Dicha función retornará la variable  $X_q$  la cual representa la señal cuantizada y  $\Delta$  que hace referencia a los pasos de cuantización.

Inicialmente, la función toma los valores máximos y mínimos de la señal  $X$ , los cuales se usan para calcular el valor del paso de la cuantización:

$$\Delta = \frac{\max - \min}{2^b} \quad (3)$$

Posteriormente, se pasa a dividir la señal  $X$  entre el paso de cuantización y el resultado es redondeado a un número entero, el cual hace referencia a la codificación que le asigna un código al valor de la señal. Dicho valor pasa a ser multiplicado nuevamente por el paso de cuantización para que se decodifique y se representa el valor asociado a ese código.

```
# Escriba un programa para cuantizar la señal v con amplitudes a=1 y a= 1/2, usa
# las longitudes de palabra o número de bits b=4 y b=8.
def my_quantization(A, x, b):
    maxAmplitud = 1
    minAmplitud = -1

    pasoCuantizacion = (maxAmplitud-minAmplitud)/(2**b)

    señal_decod_Q = np.round(x/pasoCuantizacion)#Codificación
    x_q = señal_decod_Q*pasoCuantizacion #Decodificación
    Delta = pasoCuantizacion

    return x_q, Delta
```

Figura. 4: Código de la función de cuantización

Después, se procede a calcular la secuencia de error de cuantización mediante un análisis estadístico. Primero, se calculó la media y varianza por medio de las funciones mean y var de la librería Numpy. Luego, se hizo el cálculo de la densidad de probabilidad la cual se normalizó y fue representada en un histograma de 20 barras con la función hist de Matplotlib. Por último, se calculó su secuencia de autocorrelación con la función correlate de Numpy, se normalizó y graficó.

## V. RESULTADOS

### A. Procesamiento de una imagen

Teniendo en cuenta el procedimiento descrito, se probaron 3 filtros diferentes. Además, se encontraron 3 posibles kernel para cada tipo de filtro. A continuación, se mostrarán los kernel usados.

$$\begin{bmatrix} 0. & 1. & 2. \\ -1. & 0. & 1. \\ -2. & -1. & 0. \end{bmatrix} \quad \begin{bmatrix} 1. & 1. & 0. \\ 1. & 0. & -1. \\ 0. & -1. & -1. \end{bmatrix}$$

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

Figura. 5: 3 kernel para identificar las diagonales (+45) y (-45)

$$\begin{bmatrix} 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 1. & 1. & 1. & 1. & 1. & 1. & 1. & 1. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 0. & 0. & 0. & 0. \end{bmatrix} \quad \begin{bmatrix} -1. & -1. & -1. \\ 0. & 0. & 0. \\ 1. & 1. & 1. \end{bmatrix}$$

$$\begin{bmatrix} 1. & 0. & -1. \\ 2. & 0. & -2. \\ 1. & 0. & -1. \end{bmatrix}$$

Figura. 6: 3 kernel para identificar las horizontales

$$\begin{bmatrix} 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \\ 0. & 0. & 0. & 0. & 1. & 0. & 0. & 0. & 0. \end{bmatrix} \quad \begin{bmatrix} -1. & 0. & 1. \\ -1. & 0. & 1. \\ -1. & 0. & 1. \end{bmatrix}$$

$$\begin{bmatrix} 1. & 2. & 1. \\ 0. & 0. & 0. \\ -1. & -2. & -1. \end{bmatrix}$$

Figura 7. 3 kernel para identificar las verticales

De la figura 5 a 7 podemos ver los kernel usados. Estos fueron encontrados de proyectos similares. Además, podemos notar que en la figura 6 y 7 se incluyen 2 kernel que son vectores de 1. A continuación, estos kernel son convolucionados con la representación matricial de la imagen. Los resultados de esta convolución pre-rectificación se muestra a continuación (los resultados están dispuestos de forma que se asocian con los filtros colocados anteriormente en el mismo orden):

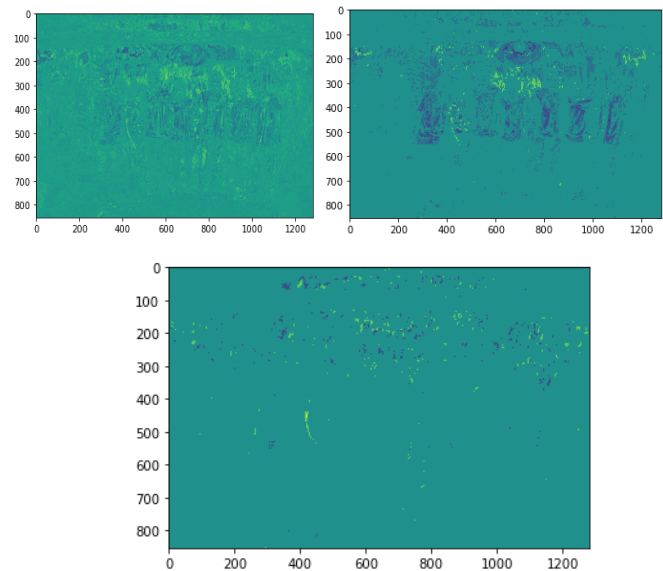


Figura 8. Imagen después de convolución para identificar las diagonales (+45) y (-45)

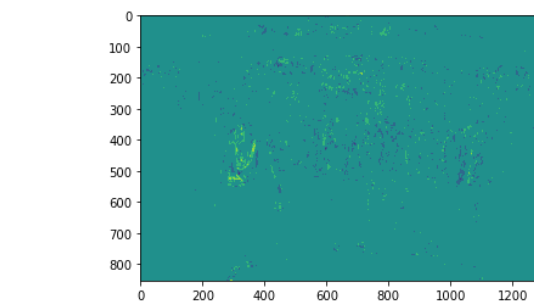
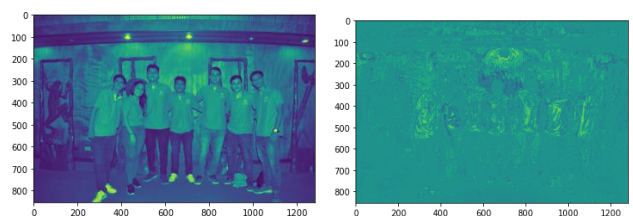


Figura 9. Imagen después de convolución para identificar las horizontales

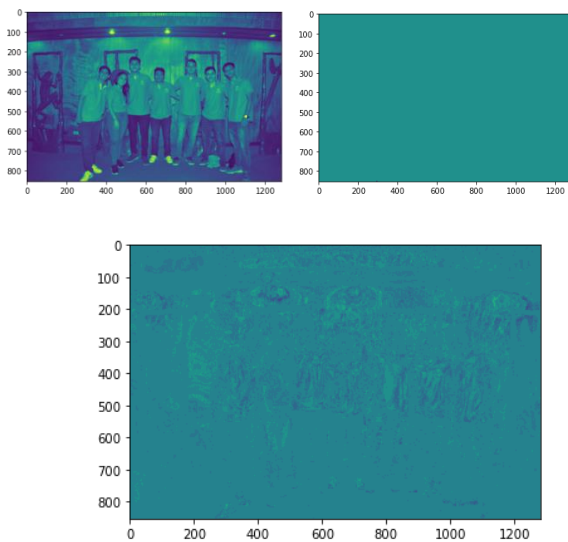


Figura 10. Imagen después de convolución para identificar las verticales

De la figura 8 a 10 podemos encontrar las imágenes después de aplicada la convolución con los filtros anteriores. En la figura 9 y 10 podemos notar como los filtros vectores no produjeron ninguna detección destacable. Además, se logra conservar partes más reconocibles de la imagen original con los segundos filtros usados tanto para diagonales como para horizontales. Esto se puede ver en la figura 8 y 9 en la imagen de la mitad.

Después de esto, se pasa la matriz de características resultante por un rectificador. Este consiste en redefinir los datos individuales de forma que los negativos se reemplazan por cero. A continuación, se muestran los resultados post rectificación:

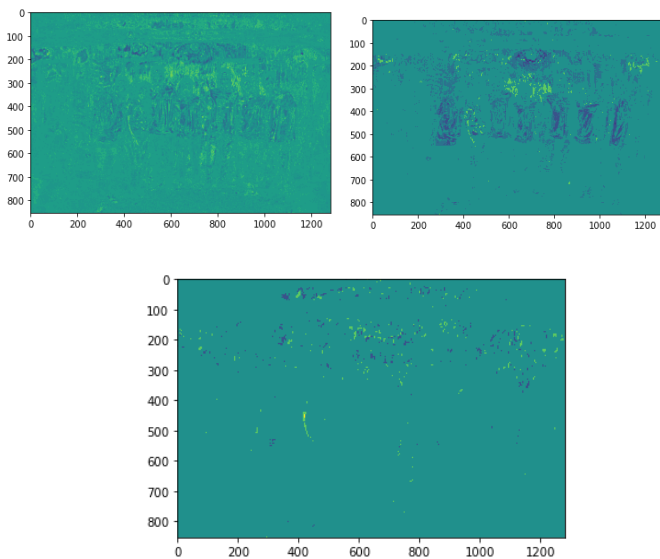


Figura 11. Imagen después de convolución y rectificación para identificar las diagonales

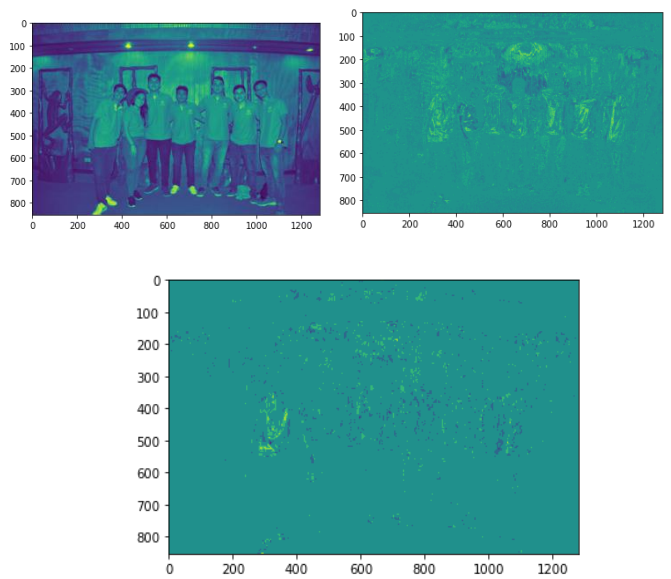


Figura 12. Imagen después de convolución y rectificación para identificar las horizontales

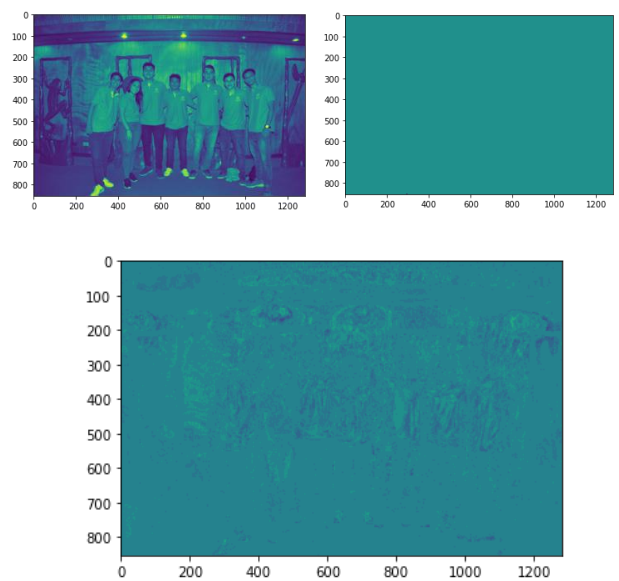


Figura 13. Imagen después de convolución y rectificación para identificar las verticales

En las figuras 11 a 13 podemos ver la representación en imagen de la matriz de características rectificada. Nuevamente observamos que al usar los filtros vectoriales no se nota una detección de las características deseadas. Por otra parte, se retienen partes importantes de la imagen original en la primera imagen de la figura 11(diagonales), segunda de la figura 12(horizontales), y tercera de la figura 13(verticales).

### B. Evaluación del SQNR de una señal

Para la cuantización de la señal, se empleó la señal dada por la siguiente función:



$$v[n] = A \sin(n\omega); \omega = \frac{2\pi}{8000} \quad (4)$$

En donde se tuvo en cuenta un rango de tiempo discreto que va de cero a ocho mil ( $n = 0:8000$ ) y se puede ver su gráfica a continuación en la figura 14:

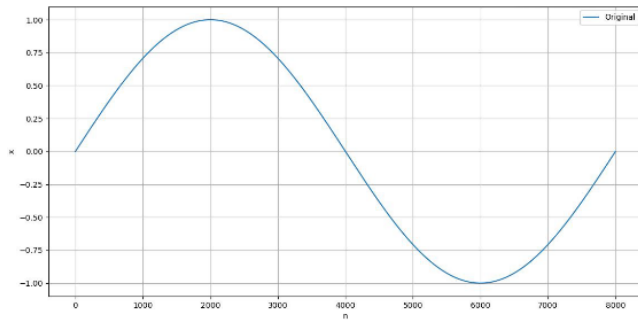


Figura. 14: Señal original a cuantizar

La señal  $v[n]$  fue pasada como parámetro a la función de *my\_quantization* con el fin de cuantizarla con dos representaciones de 4 bits y 8 bits, y amplitudes de 1 y 1/2. Además, se realizó el cálculo del error de cuantización para hacerle su debido análisis estadístico, en el cual se obtuvo su media, varianza, densidad de probabilidad y autocorrelación.

**Para  $A = 1$  y  $b = 4$**

Se realizaron los gráficos de la señal original con la cuantizada para compararlas entre sí como se puede ver en la figura 15:

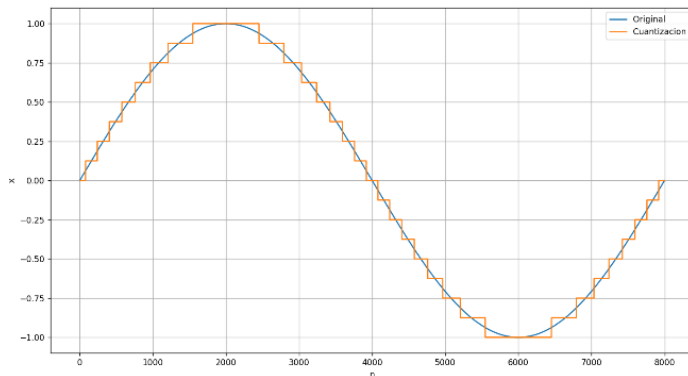


Figura. 15: Señal original y señal cuantizada

La secuencia de error de cuantización se ve reflejada en la figura 16:

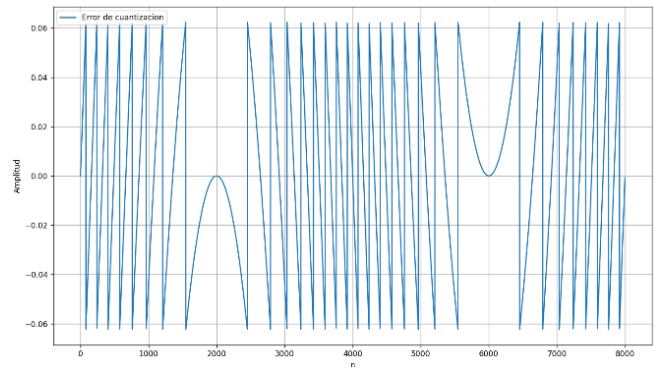


Figura. 16: Secuencia de error de cuantización usando 4 bits y amplitud de 1

Se puede apreciar que el máximo y mínimo de la figura 6 son aproximadamente 0.06 y -0.06 de manera respectiva, por lo que esta señal tendrá una distorsión debido a la diferencia entre estos valores. De igual forma, se puede ver unas variaciones bruscas del ruido antes de llegar al pico de la señal original en  $n = 2000$ , pero a medida que se acerca a este valor se vuelve más suave hasta llegar al pico en donde la diferencia es de cero debido a que en ese punto la señal original y la cuantizada son iguales.

En adición a lo anterior, se presentan los resultados del análisis estadístico del error de cuantización:

### Media y Varianza:

<b>Media</b>	4.44089e-19 $\approx 0$
<b>Varianza</b>	0.001193

La media es muy pequeña, por lo que se puede considerar casi cero y esto ocurre por tener los mismos valores positivos como negativos y la varianza muestra que el error de cuantización se mantiene en un rango reducido de valores, no varían mucho. Donde se presenta más variación lejos de los extremos de la señal original.

### C. Distribución de Probabilidad

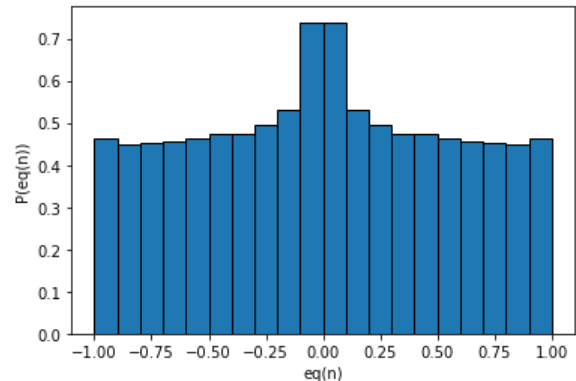


Figura. 17: Distribución de probabilidad del error de cuantización

En la figura 17 se ve el histograma de la densidad de probabilidad, en donde se aprecia que toma una forma muy uniforme lejos de los extremos de la señal original, pero a medida que se acerca a los extremos comienza a dejar de serlo pues en la barra más próxima a los costados presenta una subida en vez de tener un comportamiento decreciente.

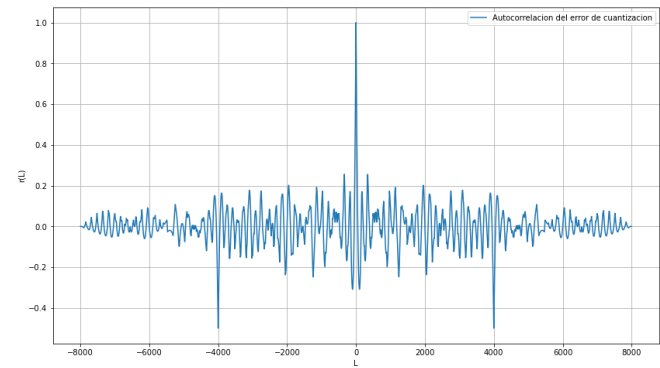


Figura. 18: Secuencia de autocorrelación del error de cuantización

En la figura 18 se muestra la gráfica de la autocorrelación del error de cuantización, en ella se ve que esta alcanza su máximo cuando  $L=0$ , no presenta variaciones muy bruscas pues se mantiene en un rango constante de valores cercanos a cero.

Para  $A = 1/2$  y  $b = 4$

De igual forma, se realizaron los gráficos de la señal original con la cuantizada con una amplitud de  $1/2$  para compararlas entre sí y también se obtuvo la gráfica de su secuencia de error de cuantización, como se puede ver en la figura 19 y 20:

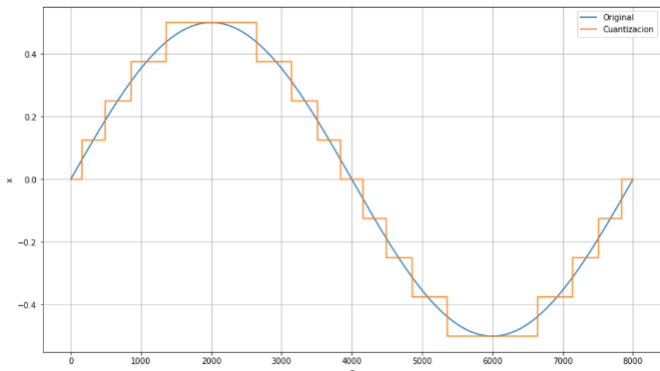


Figura. 19: Señal original y señal cuantizada

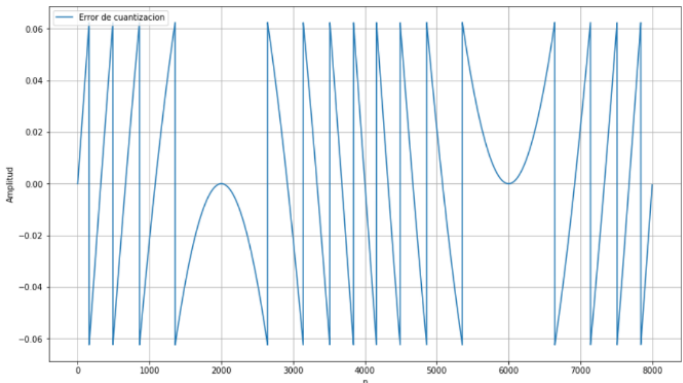


Figura. 20: Secuencia de error de cuantización usando 4 bits y amplitud de  $1/2$

En esta ocasión, en la figura 20 se aprecia que hay un máximo y un mínimo de  $0.5$  y  $-0.5$  respectivamente y por ello existe una cierta distorsión a causa de la diferencia entre los valores de estas dos señales y también debido a que, a amplitudes más bajas, el error de cuantización se vuelve dependiente de la señal de entrada. Además, se presentan unas variaciones en el ruido antes de alcanzar el pico de la señal original en  $n = 2000$  como se vió en la figura 16, solo que en este caso son más suaves.

Los resultados del análisis estadístico del error de cuantización son los siguientes:

**Media y Varianza:**

<b>Media</b>	0.0
<b>Varianza</b>	0.00114841

La media tiene un valor igual a cero, por ende se puede decir que esto pasa debido a que se cancelaran los valores positivos y negativos al ser una onda de tipo seno. De igual manera, la varianza refleja una variación mínima en el error de cuantización al permanecer en un rango reducido de valores al igual que ocurre en el caso de  $A = 1$  y  $b = 4$  bits.

Para  $A = 1$  y  $b = 8$

Ahora realizando la cuantización, pero con 8 bits, se puede ver que ambas señales tanto la original como la cuantizada terminan siendo prácticamente las mismas debido a que se incrementa la resolución cada vez que se adiciona un bit. La gráfica se puede ver en la figura 21.

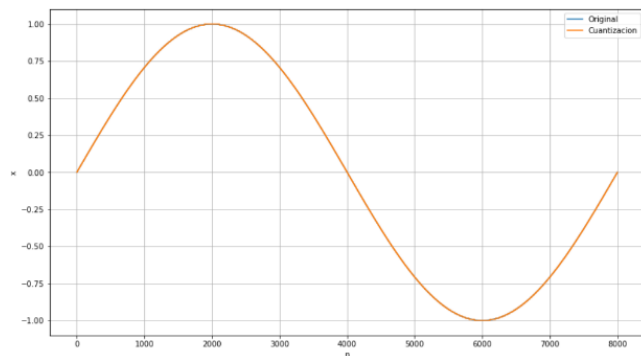


Figura. 21: Señal original y señal cuantizada

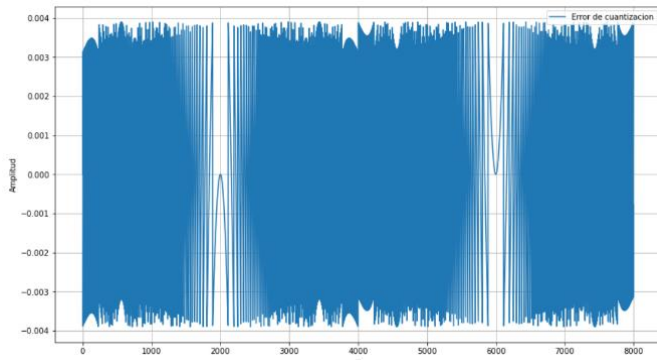


Figura. 22: Secuencia de error de cuantización usando 8 bits y amplitud de 1

En la figura 21 se puede ver el resultado de la secuencia de error de cuantización al aumentar de 4 a 8 bits el número de muestras. Podemos observar cómo hay puntos de quiebre en los máximos y mínimos relativos de la función, además, los intermedios se muestran como cotas de amplitud similar. De igual forma, se nota como el número de cotas de transición aumentan de forma significativa a medida que se aumentan el número de bits del número de muestras, esto debido a que se está aumentando la frecuencia de la señal, por lo que puntos que antes se reconstruían en líneas más “suaves”, ahora se muestran en segmentos más pequeños y esto relacionado con el delta, al tener más bits, el delta tiende a ser más pequeño haciendo que la potencia de ruido disminuya y esto hace que la relación señal a ruido (SQNR) aumente, provocando que la calidad de la señal sea mejor.

Los resultados del análisis estadístico del error de cuantización son los siguientes:

#### Media y Varianza:

<b>Media</b>	5.7245e-19 $\approx$ 0
<b>Varianza</b>	0. 0000049805

La media tiene un valor igual a cero, por ende se puede decir que esto pasa debido a que se cancelan los valores positivos y negativos al ser una onda de tipo seno. De igual manera, la varianza refleja una variación mínima en el error de cuantización al permanecer en un rango reducido de valores al igual que ocurre en el caso de  $A = 1$  y  $b = 4$  bits, solo que esta

vez es mucho más pequeño, por lo que esto se ve reflejado en el valor de la varianza del error de cuantización al ser poco, haciendo que la calidad de la señal sea mayor.

#### Para $A = 1/2$ y $b = 8$

Repitiendo el mismo procedimiento, pero con una amplitud de  $A = 1/2$  se puede ver en la figura 23 que es muy similar a la figura 21. No obstante, a amplitudes más bajas, el error de cuantificación se vuelve dependiente de la señal de entrada, lo que genera distorsión.

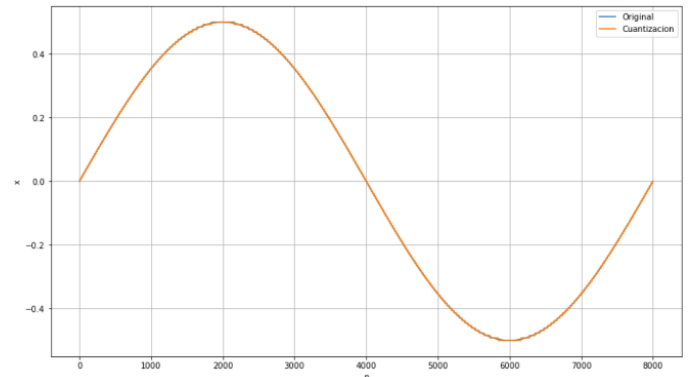


Figura. 23: Señal original y señal cuantizada

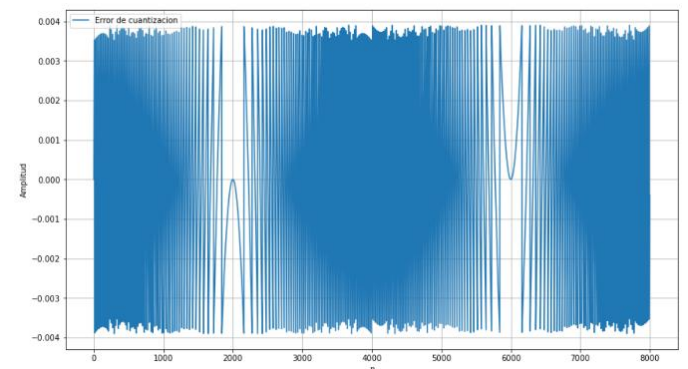


Figura. 24: Secuencia de error de cuantización usando 8 bits y amplitud de 1/2

Con base a la figura 24, se puede apreciar un comportamiento similar a cuando se tenía una amplitud del doble, es decir, de  $A = 1$ . No obstante, en el caso de  $A = 1/2$

## VI. CONCLUSIONES

### A. Procesamiento de una imagen

Se procesa una imagen para detectar 3 características: líneas horizontales, verticales, y diagonales. Se observa que existen diversos filtros aún para determinar una misma característica. Algunos de estos pueden ser vectoriales o matrices cuadradas previamente definidas y probadas. Se observa que los filtros vectoriales no tienen buenos resultados pues no se evidencia una detección. Además, se observa cómo se mejora la detección al aplicar la rectificación en algunos de las pruebas con distintos filtros. Además, la detección no es “fina”, pues detecta a grandes rasgos las características deseadas, más no delinea los detalles y particularidades de la imagen.

### B. Evaluación del SQNR de un señal

Se puede concluir que la calidad de la señal cuantizada se ve afectada en gran medida por la cantidad de bits empleada para el proceso de cuantización, donde con un mayor número de bits se tiene un menor error de cuantización. Sin embargo, el aumento de bits desmedido podría ocasionar problemas de interferencia electromagnética en la práctica. Además, se nota que el error de cuantización es muy pequeño y su media y varianza son cercanos a 0, lo cuál confirma la teoría de que la media del ruido es 0, o, en este caso, cercana a 0.

### VII. DISTRIBUCIÓN DEL TRABAJO

Cada miembro del equipo tiene experiencia en la realización de las distintas partes del laboratorio. Sin embargo, a continuación, se describirán los aportes distintivos de cada miembro:

**A. Juan José Restrepo:** Desarrolló el programa que da solución al problema del error de cuantización. Organizó el cuaderno en Jupyter. Además, se encargó de proveer los filtros usados para los primeros ejercicios.

**B. Juan David Vargas:** Desarrolló el programa que da solución al problema de convolución 3d y rectificación. Se encargó de hacer pruebas y comparar los distintos filtros encontrados.

Adicionalmente, cada miembro colaboró en el informe desde lo que realizó en Python. Los resultados fueron compartidos y en muchos casos se ejecutaron los códigos en medio de las reuniones de equipo.

### VIII. REFERENCIAS

- [1] Benítez, H., 2022. Correlación de señales.
- [2] Benítez, H., 2022. Introducción.
- [3] "The Batch | DeepLearning.AI | AI News & Insights", The Batch | DeepLearning.AI | AI News & Insights, 2022. [Online]. Available: <https://read.deeplearning.ai/the-batch/>. [Accessed: 21- Aug- 2022].
- [4] Edge detection with Sobel filters. (2022). Retrieved 22 August 2022, from <https://jameshfisher.com/2020/08/31/edge-detection-with-sobel-filters/>
- [5] Parte, Segunda. Tema 3: Filtros.
- [6] Detección de Bordes En. 2005.
- [7] Vásquez, Rubén. "Convoluciones En OpenCV Y Python." [www.youtube.com](https://www.youtube.com/watch?v=Fgp5AZcsbQI), 22 Oct. 2020, [www.youtube.com/watch?v=Fgp5AZcsbQI](https://www.youtube.com/watch?v=Fgp5AZcsbQI). [Accessed 22 Aug. 2022].
- [8] Yeh, S., Weng, C. and Lau, B., n.d. FACE DETECTION THROUGH TEMPLATE MATCHING AND COLOR SEGMENTATION. [ebook] Available at: <https://web.stanford.edu/class/ee368/Project03/Project/reports/ee368group04.pdf>
- [9] Advanced Digital Signal Processing using Python - 01 Quantization. 2021. [video] Youtube- <https://www.youtube.com/watch?v=2HudwLcYpPI>: Guitars AI.