FACULTAD DE INGENIERÍA

UNIVERSIDAD DE BUENOS AIRES

DEPARTAMENTO DE ELECTRÓNICA

86.09 - PROCESOS ESTOCÁSTICOS

1° CUATRIMESTRE 2023

Trabajo Práctico N° 1
Compresión de Imágenes

Autores:
Bruno Berlatzky (99489) Maximiliano Torre (93679) Alexis Joel Romero Espinoza (103942) Erik Schatz (96470)

Índice

1. Introducción **1**

2. Desarrollo **1**

2.1. Ejercicio 1:Correlación 1
2.2. Ejercicio 2: Compresión 2
2.3. Ejercicio 3: Descompresión 2
2.4. Ejercicio 4: Error cuadrático medio 4

3. Conclusiones **7**

1. Introducción

En el presente trabajo se busca comprimir imágenes para que ocupen un menor espacio de almacenamiento, teniendo en cuenta que las imágenes presentan redundancias que pueden ser reducidas sin afectar la calidad de la imagen final. El objetivo se puede lograr aplicando la transformada de Karhunen-Loève(KLT), que consta en eliminar las redundancias comparando los pixeles con sus vecinos y eliminando los que presenten menor cantidad de cambios, y de esa forma al guardar los datos se puede reemplazar la redundancia por un dato con menor cantidad de pixeles, ocupando menos memoria, sin perder información importante.

2. Desarrollo

Para simplificar la compresión, las imágenes dadas las transformamos en imágenes representadas en una matriz de dos dimensiones, donde cada elemento contiene un valor que representa la intensidad en una escala de grises.
Se busca que la imagen procesada ocupe un menor espacio sin alterar significativamente a la calidad de la imagen resultante. Para llevar a cabo esto, se aplica la compresión PCA.

2.1. Ejercicio 1:Correlación

Para este ejercicio, utilizamos las imagenes 1 y 2,descomponiendo en matrices de 2x1 de pixeles contiguos para poder observar la correlación que hay entre ellos. Para esta descomposición, se asumió que las filas de la matriz de la imagen tiene un nuevo par de filas. También se busca calcular el coeficiente de correlación, el cual se calcula como:

$$\rho(X,Y) = \frac{Cov(X,Y)}{\sigma_X \sigma_Y} \tag{1}$$

Para la imagen 1, la matriz resultante al aplicar la función corrcoef(x,y) es:

$$\begin{bmatrix} 1{,}0000 & 0{,}9792 \\ 0{,}9792 & 1{,}0000 \end{bmatrix} \tag{2}$$

El valor del coeficiente de correlación que nos interesa es $\rho_1 = 0{,}9792$

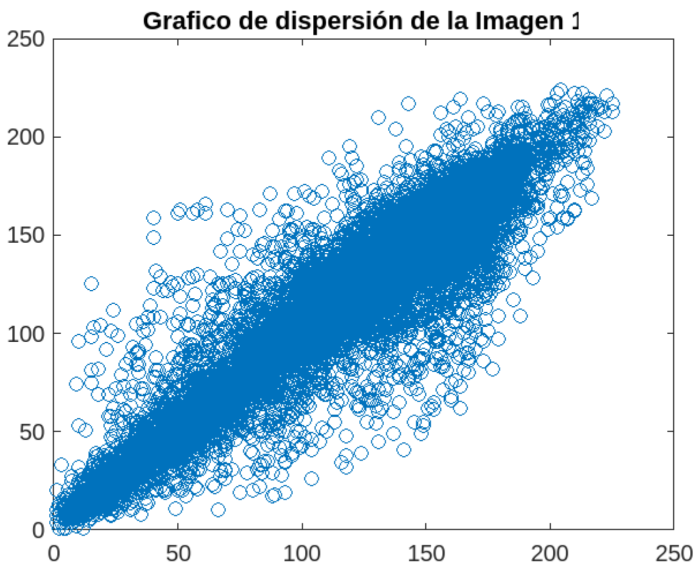


Figura 1: Gráfico de dispersión imagen 1

Para la imagen 2:

$$\begin{bmatrix} 1{,}0000 & 0{,}1459 \\ 0{,}1459 & 1{,}0000 \end{bmatrix} \tag{3}$$

El valor del coeficiente de correlación resulta $\rho_2 = 0{,}1459$

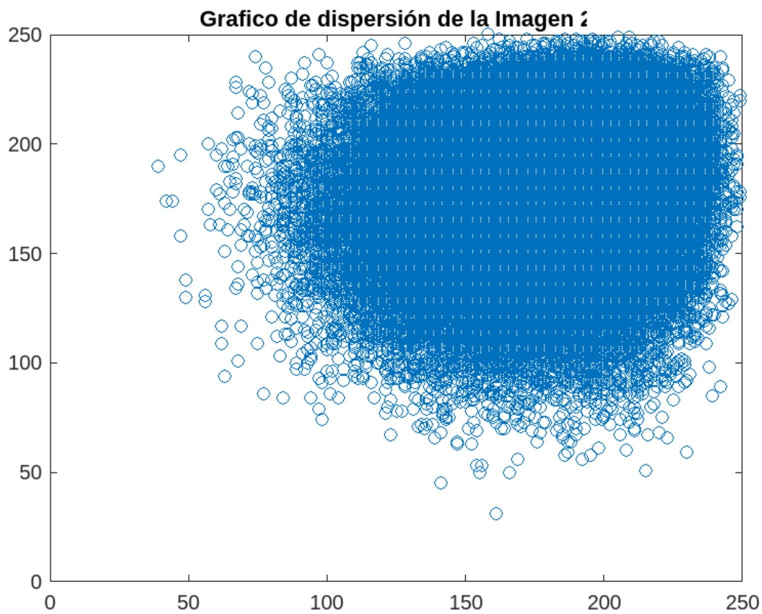


Figura 2: Gráfico de dispersión imagen 2

2.2. Ejercicio 2: Compresión

Para este proceso, se segmentan las matrices de las imágenes en bloques de 8x8, en el caso de que las filas y/o columnas de la imagen no sean múltiplo de 8, completamos con ceros en los bordes de la imagen, los cuales no afectarán a los cálculos posteriores.

En este ejercicio, se estimará la matriz de covarianza C_X y la media $\hat{\mu_X}$ para la descomposición de la imagen 3.

La estimación de estos parámetros se calculará de la siguiente forma:

- Media: $\frac{1}{n}\sum_{k=1}^{n}\mathbf{X}_k$
- Matriz de covarianza: $\hat{C}_X = \frac{1}{n}\sum_{k=1}^{n}(\mathbf{X}_k - \hat{\mu}_{\mathbf{X}})(\mathbf{X}_k - \hat{\mu}_{\mathbf{X}})^T$

Luego se descompone la matriz de covarianza de 64x64 en su matriz de autovectores V, y la matriz de autovalores D. Para poder comprimir la imagen, se utilizaran los autovectores con mayor varianza, es decir los que mas información contienen de la imagen. En este caso se utiliza una tasa de compresión del 20 %, reduciendo la matriz de autovectores a 64x12. Al proyectar los vectores x_i en el subespacio generado por los autovectores, obtenemos las proyecciones y_i, una matriz de 12x2640.

2.3. Ejercicio 3: Descompresión

Para descomprimir las imágenes, se aplica la transformación inversa, utilizando las proyecciones antes calculadas y la matriz de autovectores reducidos y luego hay que sumarle la media, obteniendo un bloque del tamaño original pero con la perdida de información ya que se suprimieron autovalores.

Para este trabajo práctico, también se realizó el algoritmo de procesamiento de imágenes para RGB, entendiéndolo como simplemente la extensión del mismo algoritmo para escala de grises, aplicado para cada canal de color de RGB.

(a) Imagen original en escala de grises (b) CR=20 %

Figura 3: Resultado de compresión y descompresión en escala de grises

(a) Imagen original RGB (b) CR=20 %

Figura 4: Resultado de compresión y descompresión en RGB

Se observa en ambos procedimientos que para un CR=20 % la calidad de la imagen se ve reducida, pero los rasgos principales de la imagen se reconocen perfectamente.

2.4. Ejercicio 4: Error cuadrático medio

Figura 5: Imagen 4 original en escala de grises.

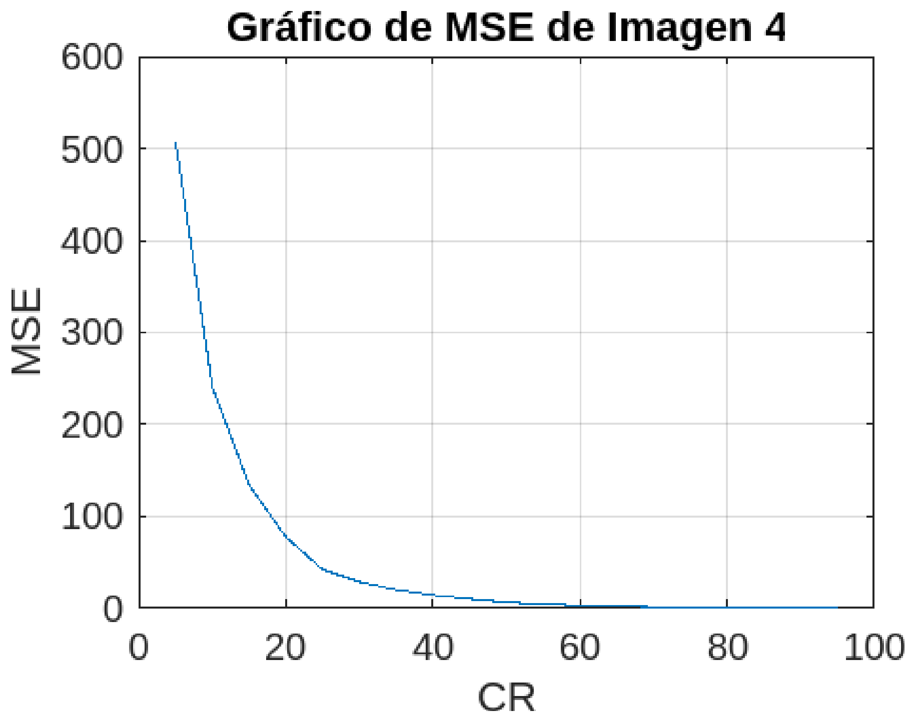


Figura 6: Gráfico del MSE en función de la tasa de compresión.

Para la realización de este ejercicio se eligió encontrar un algoritmo que permitiera definir automaticamente la cantidad de dimensiones removidas del subespacio de autovectores la matriz C_x. Para ello simplemente se definió la ecuacion que las vincula, y el algoritmo solamente debe redondear al entero mas cercano posible para el valor de CR requerido.

$$CR = \frac{\text{datos comprimidos}}{\text{datos totales}} \tag{4}$$

$$\text{datos totales} = m * n; \ \text{datos comprimidos} = size(Y_x) + size(U_c) + size(\mu_x) + 4 \tag{5}$$

Donde m y n son las filas y columnas de la imagen, Y_c es la matriz con las proyecciones de los bloques originales sobre el subespacio reducido generado por los autovectores U, μ_x es la media de los bloques originales y por ultimo el valor 4 surge de requerir guardar el tamaño de las filas y columnas de la imagen original, y las filas y columnas de los bloques. Para simplificar se define K como el tamaño del bloque entero, y por ultimo se define L como la cantidad de bloques que tiene la imagen original.

$$size(Y_x) = (K - \lambda)L \tag{6}$$

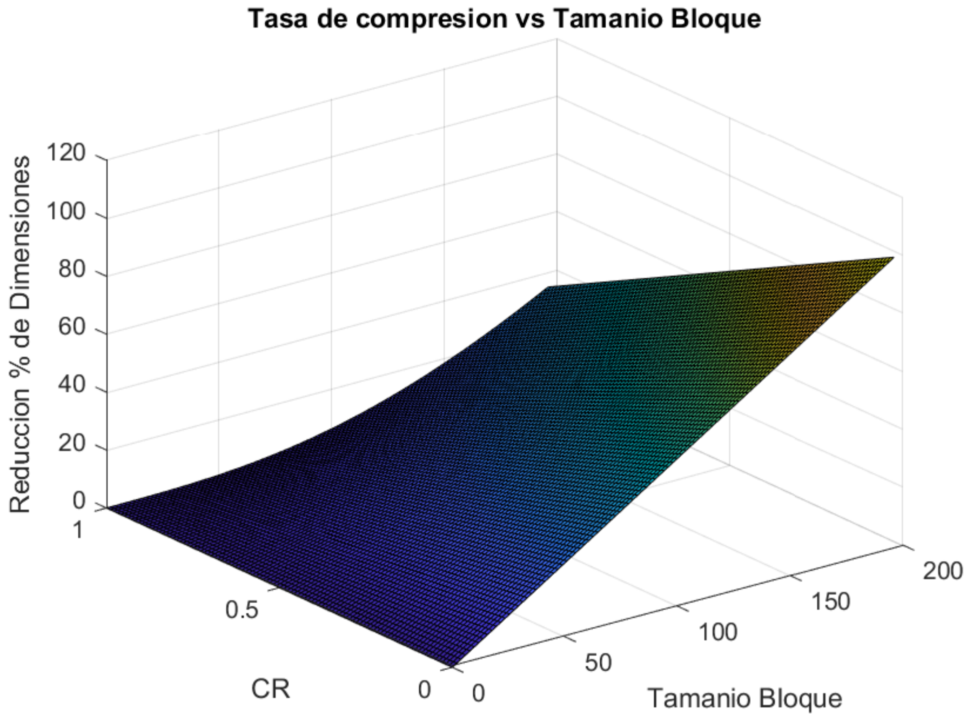
$$size(U) = K(K - \lambda) \tag{7}$$

$$size(\mu_x) = K \tag{8}$$

Se llega entonces a la ecuación:

$$\lambda = k - \frac{mnC_R - k - 4}{\frac{mn}{k} + k} \tag{9}$$

Se grafica a continuación el porcentaje de reducción de dimensiones del subespacio original de la diagonalizacion de C_x, en función del CR y del tamaño del bloque.



Este gráfico indica que para valores de CR entre el 5 % y 25 % aumentar el tamaño de los bloques reduce significativamente la cantidad de dimensiones del subespacio de U. Esto se traduce en una perdida significativa de información de la imagen, como se puede visualizar a continuación entre los siguientes conjuntos de imágenes:

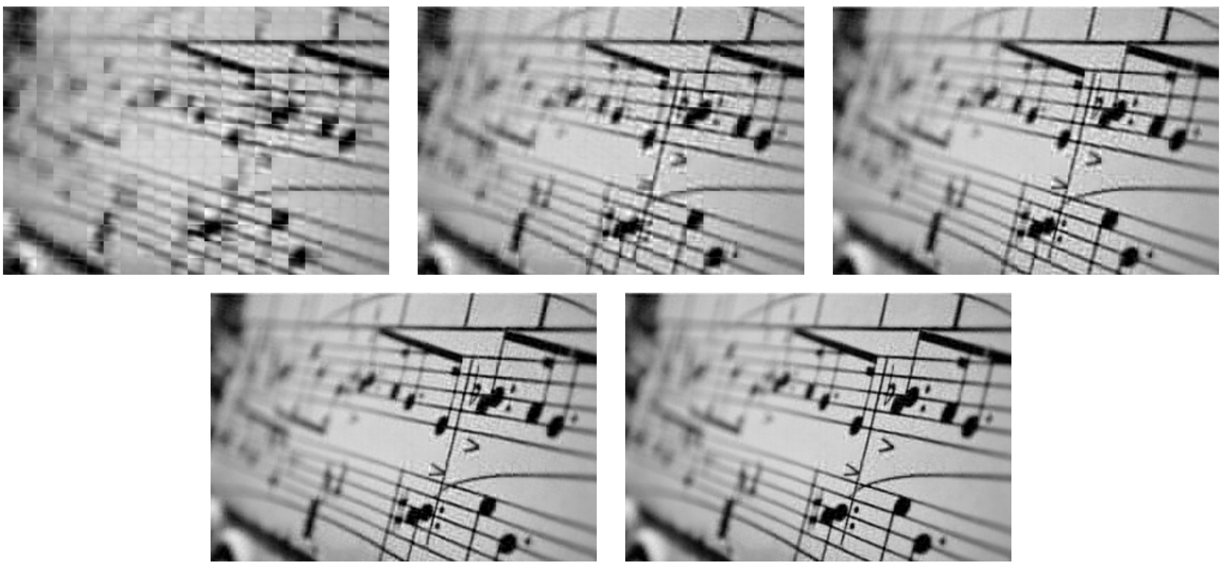


Figura 7: Progresión de la compresión de la imagen 4 para un bloque de 14x14

Este primer conjunto corresponde a la progresión de la compresión de la imagen 4 desde un CR de 5 % al 25 %, donde se puede apreciar que ya para un 10 % con un bloque de 14x14 la imagen es bastante semejante visualmente a la original.

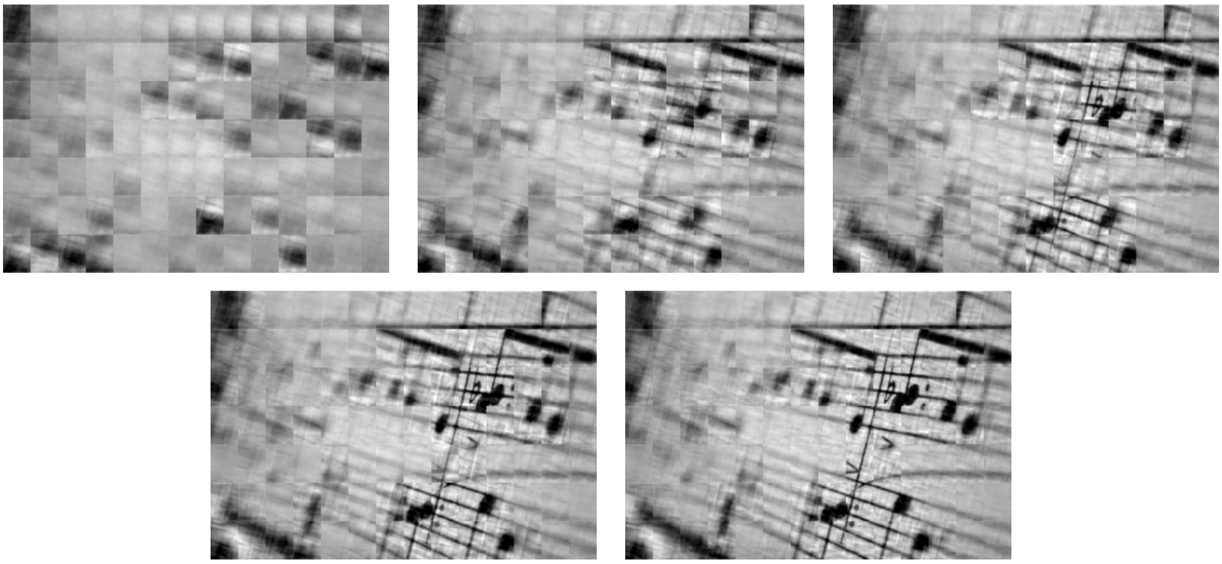


Figura 8: Progresión de la compresión de la imagen 4 para un bloque de 32x23

En este caso, para un bloque de 32x23, recién al 25 % la imagen tiene rasgos distinguibles y comparables de la imagen original.

Se concluye que variar el tamaño de los bloques es un juego de tradeoff entre el costo computacional de los cálculos de correlación, la cantidad de memoria reducida, y la calidad de la imagen.

3. Conclusiones

A lo largo de este trabajo, pudimos entender y aplicar el metodo de compresion mediante PCA (Análisis de componentes principales), del cual podemos destacar la gran utilidad que este aporta si se quiere reducir el almacenamiento y el procesamiento, eliminando redundancias al considerar los autovalores mas importantes, es decir los que tienen mas importancia en la imagen ,pero puede traer inconvenientes a la hora de eliminar información ya que depende de que tasa de compresión se utilice, pudiendo afectar a la calidad de la imagen resultante en comparación con la imagen original. Por ultimo se puede concluir que este método, a pesar de tener una buena tasa de compresión con un MSE bueno, es computacionalmente muy caro por el costo de recorrer con bloques, calcular la descomposición den valores singulares, y proyectar matrices.