Centro de Investigación en Matemáticas (CIMAT).
Unidad Monterrey

Compresión de imágenes con métodos de ML

Armando Salinas Lorenzana

2 de septiembre de 2025

1. Problema

Este ejercicio trata sobre compresión de imágenes a color. Considera una imagen en color como la que se muestra en la Figura 1.1 (derecha). Tu objetivo es reducir el tamaño (en Kb) de la imagen, tratando de mantener un balance entre tamaño y calidad de la misma, y para esto, utilizarás dos métodos.

a) **k-means**: En este caso, se simplifica la imagen identificando k grupos de colores en los píxeles de la imagen, y posteriormente, se asigna cada píxel a su grupo de color correspondiente. Para esto, considera cada píxel $\mathbf{x}_i \in \mathbb{N}^3$, es decir, como un punto en el espacio RGB. Tu matriz de datos será entonces $\mathbf{X} \in \mathbb{N}^{(h\times w)\times 3}$, donde w y h son el ancho y la altura de la imagen, respectivamente. Los pasos se describen en el Algoritmo 1.

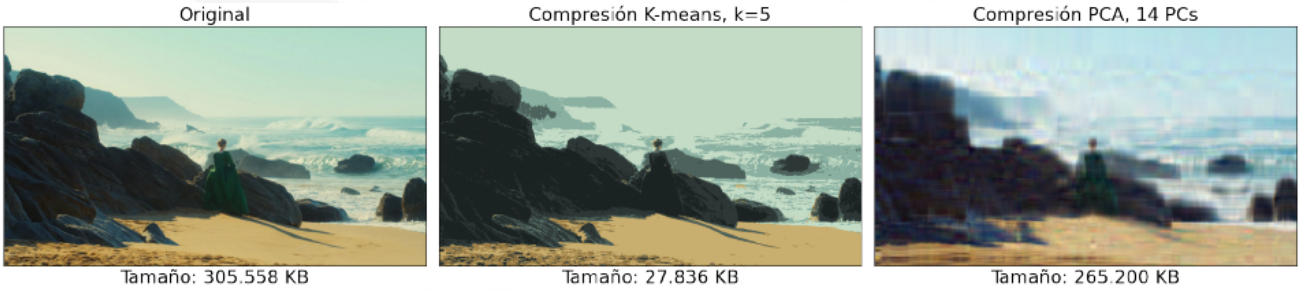

Figura 1.1: Compresión de imágenes. Izquierda: imagen original. Centro: compresión obtenida con k-means usando $k = 5$ colores. Derecha: compresión obtenida con PCA usando 14 componentes principales.

Algorithm 1 Compresión de imágenes 1

1: **Input:** imagen RGB $\mathbf{I}_{h\times w}$, número de clústers k
2: Obtén la matriz de datos $\mathbf{X} \in \mathbb{N}^{(h\times w)\times 3}$
3: Realizar k-means y obtener la asignación del clúster $C(i)$ para cada píxel $\mathbf{x}_i$, así como los centroides $\boldsymbol{\mu}_k$ de cada clúster.
4: Asignar los valores RGB de cada píxel, de acuerdo a su clúster correspondiente, es decir: $x_i^{(k)}$
5: Reconstruir la imagen comprimida $\tilde{\mathbf{I}}$ con su tamaño original

b) **Componentes principales**: En este método, realizarás la compresión simplificando la imagen mediante algunos componentes principales obtenidos en cada canal de color de la imagen. En este caso, considera representaciones de la forma $X_{\text{channel}} \in \mathbb{N}^{h\times w}$, y el procedimiento se describe en el Algoritmo 2.

Algorithm 2 Compresión de imágenes 2

1: **Input:** imagen RGB $\mathbf{I}_{h\times w}$, número de componentes p o varianza acumulada v.
2: Obtener las matrices de datos por cada canal X_R, X_G y X_B cada una del tamaño de la imagen original.
3: Realizar PCA en cada canal, para reducir la dimensión de cada canal de la imagen según los p primeros componentes principales, o equivalentemente, según los p componentes principales que acumulan una varianza v.
4: Sea $\tilde{X}h \times p$ los scores obtenidos con PCA, $Vw \times p$ los componentes principales, y $\boldsymbol{\mu}_w$ el vector de promedios de la imagen original. Reconstruir cada canal de la imagen con su descomposición PCA correspondiente mediante
$$\tilde{\mathbf{I}}_{\text{channel}} = \tilde{X}V' + \mathbf{M},$$
donde $\mathbf{M}$ contiene el vector de medias $\boldsymbol{\mu}$ por renglones.
5: Obtener la imagen final comprimida mediante $\tilde{\mathbf{I}} = [\tilde{\mathbf{I}}_R\, \tilde{\mathbf{I}}_G\, \tilde{\mathbf{I}}_B]$.

Para cada inciso, verifica el resultado con distintos valores de k y p, reportando también el tamaño en Kb de la imagen. Escribe tus conclusiones de este ejercicio donde consideres el balance entre compresión y calidad de la imagen. ¿Qué método prefieres y por qué? ¿Qué nivel de compresión te parece adecuado? ¿Qué criterio (cuantitativo) se te ocurre para evaluar la compresión de la imagen original?
Detalles de implementación:
En el Ejercicio 4, puedes calcular el tamaño de la imagen con la siguiente función:

```
from io import BytesIO
def imageSize(img):
img_file = BytesIO()
image = Image.fromarray(np.uint8(img))
image.save(img_file, 'png')
return img_file.tell()/1024
```

donde `img` es una imagen RGB en forma de tensor, cuyas entradas son enteros en un rango de 0 a 255.
Todos los elementos para la reconstrucción de la imagen del paso 4 del Algoritmo 2, puedes obtenerlos de la clase PCA del módulo `sklearn.decomposition`. Alternativamente, puedes usar el método `inverse_transform()` del mismo módulo.

1.1. Solución

En este ejercicio se realizó la comparación de la compresión de imagen usando dos algortimos distintos, usando k-means y PCA, la imagen con la que se realizó la comparación fue la siguiente

Figura 1.2: Figura de prueba. Dimensiones:640 x 420. Peso: 72.7 kb

a) Primero se realizó el método de k-means, se realizó para distintos valores de grupos que se le debía indicar al algoritmo y en base a esto pudimos obtener los siguientes resultados.

Figura 1.3: Comparación usando el método k-means con distintos valores de agrupación.

Podemos observar importantes diferencias, la prueba se realizó para cuatro distintos número de grupos, primeramente se realizó el grupo de dos elementos para observar qué clusterizaba, podemos notar que solo hay dos colores por la misma razón y que la imagen pesa 33.89 kb, es decir es la imagen con menos peso, sin embargo no es la mejor compresión ya que practicamente la gama de colores es muy baja, posteriormente se realizó con 15 grupos, en la imagen podemos observar que ahora se distinguen muchos más colores y la imagen se asemeja más a la original, aunque podemos notar que

aún no es lo suficiente, el peso de la imagen claramente incremento a 43.46 kb es decir 10 kb más. Luego se probó con 50 grupos, y podemos observar en la imagen que ya es aceptable la compresión aunque aún podemos distinguir un pequeño matiz que no es natural en una fotografía, pero en terminos generales es aceptable, el peso de la imagen incremento a 43.95 kb, un poco incremento con el de 15 grupos, por último se realizó la prueba con 100 grupos, y aquí ya podemos aceptar la compresión de imagen dado que no podemows hacer distinción entre la original y esta nueva,a demás el peso de la imagen disminuyó a 42.61 kb, con una mayor calidad, aunque no es mucho el la disminución del peso, ocurrió este efecto, en comparación de la iamgen original se pudieron ahorrar 30.18 kb de la imagen sin apenas perder calidad, asi que nos quedamos con esta última opción, para valores de grupos mayores a 100, ya no podemos notar cambios significativos en la calidad. Algo a destacar es que el tiempo de procesamiento de este algoritmo es tardado, por lo que es un punto a considerar.

b) Ahora se mostraremos los resultados obtenidos al implementar el algoritmo de pca para la compresión de la misma figura

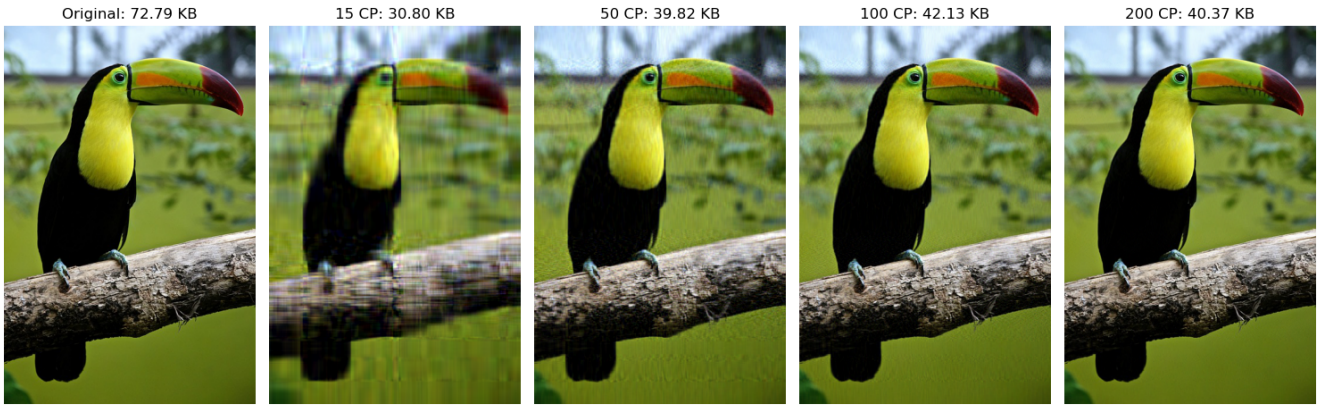

Figura 1.4: Comparación usando el método PCA con distintos valores de componentes principales.

Podemos observar las diferencias que existen al implementar este método. Primeramente para el uso de 15 componentes podemos observar una imagen bastante borrosa, realmente calidad es mala, esto tiene todo el sentido puesto que solo se usan 15 componentes y la información que se recupera es poca, por esta razón el peso de la imagen es de 30.80 kb, luego para 50 componentes podemos observar una imagene más nítida en comparación con la de 15 componenetes, aunque podemos llegar a observar que aún tiene cierto grado de borrosidad, el incremento del peso de la imagen es hasta 39.82 kb, para 100 componentes podemos observar una imagen bastante nítida, una iamgen con una calidad bastante aceptable, a comparación de 50 componentes, este solo incrementó 3 kb a 42.13 kb, por lo que no es mucha la diferencia, sin embargo al usar 200 componentes podemos notar una imagen de muy buena calidad con un peso de 40.37 kb, un efecto que también sucedió con k-means, que el peso de la imagen dismiuyó, por tanto esta es la mejor opción puesto que se observa una muy imagen de gran calidad con una disminución del 37.42kb, es decir se ahorró 7 kb más en comparación con k-means.

Discusión:

Podemos notar que ambos métodos hacen un buen trabajo al comprimir las imagenes y hacerlas menos pesadas, sin embargo existen diferencias entres estos dos métodos, la principal diferencia es que parece ser que el método de PCA suele ser mejor o más sencible al degrado de los colores, como fue en este caso, la calidad se mantiene mucho mejor en este método en cada componente, caso contrario con k-means que se logra distinguir distintas tonalidades en la imagen cuando existen estos degradados de colores, sin embargo podemos pensar que k-means puede realizar un mejor trabajo cuando se trata de imagenes como dibujos o pinturas, donde no existen gradientes de colores y los colores se pueden distinguir claramente en la imagen original, en este caso k-means es superior. Por tanto podría surgir la pregunta, ¿cuál es mejor? creo que la respuesta depende del enfoque, si se está trabajando con imágenes donde las regiones son claramente definidas y no hay muchas transiciones suaves, K-Means puede ser una opción simple y efectiva. Si se está trabajando con imágenes más complejas, con gradientes suaves o transiciones sutiles, PCA puede ser una opción más robusta, ya que captura la estructura global de la variación en la imagen.