Agrupamiento

Práctica 7

César Borja Moreno

NIP: 800675

20/05/2022

Aprendizaje Automático

Índice

1. Objetivo

Implementar el algoritmo K-means y utilizarlo para cuantificar el color de imágenes.

2. Implementación del algoritmo K-means

2.1. Inicialización de centroides iniciales

Para la inicialización de los centroides iniciales (el número de centroides K se discutirá en la sección **??**) se ha utilizado la heurística "furthest", que consiste en lo siguiente:

1. Se elige una muestra (pixel) aleatoria como el primer centroide.
2. El segundo centroide es la será la muestra cuya distancia euclídea sea mayor con respecto al centroide inicial.
3. Cualquier nuevo centroide j será la muestra que esté a más distancia de su centroide más cercano.

Este método de inicialización garantiza que los centroides iniciales están distribuidos correctamente y por ello el algoritmo K-means necesita muy pocas iteraciones para converger. Además, garantiza que los centroides no van a ser colocados en muestras del mismo color.

En las figuras 1 y 2 se muestra la comparación de las imágenes reconstruidas utilizando el algoritmo K-means inicializado con la heurística "furthest" y de manera aleatoria. Comparándolas con la imagen original 3, se aprecia como la paleta de colores de la inicialización aleatoria no es fiel a la de la imagen original.

2.2. Implementación de la actualización de los clusters

Una vez inicializados los centroides, se ejecutan los siguientes pasos:

1. Se etiqueta cada muestra con su centroide más cercano.
2. Se recalculan los centroides.
3. Se vuelven a calcular las etiquetas. ¿Han cambiado? Paso 1.
4. Fin.

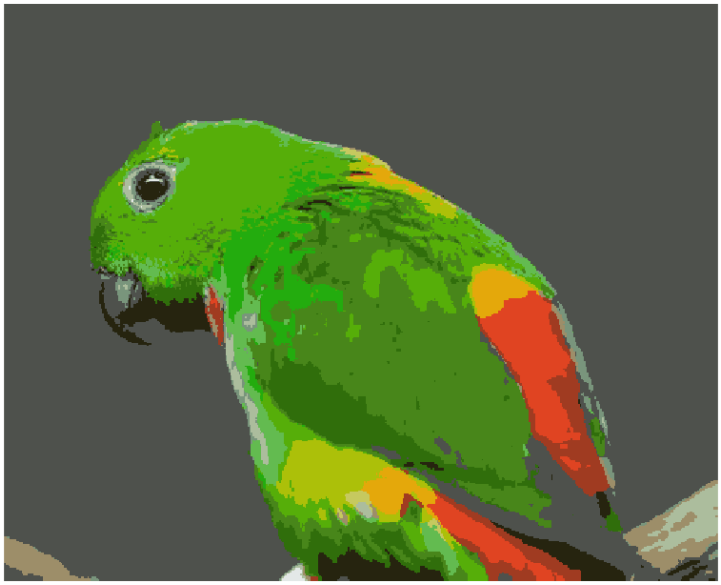

Figura 1: Smallparrot con K = 16 inicializado con "furthest"

Figura 2: Smallparrot con K = 16 inicializado con centroides aleatorios

Para calcular las etiquetas a cada muestra se ha utilizado la siguiente función de MATLAB:

```
Z = dsearchn(mu,D);
```

Esta función devuelve para cada muestra en D el índice del centroide de mu más cercano, es decir, etiqueta las muestras de D con los centroides de mu.

Por otro lado, para recalcular los centroides se calcula la media de los valores de las muestras de cada centroide. Esta media será la posición del nuevo centroide. Esto se ha implementado con el siguiente código:

```
for i=1:k
    v = D(c==i,:);
    media = mean(v);
    munew = [munew;media];
end
```

3. Pruebas y resultados

Para comprobar el funcionamiento del algoritmo aplicado a la cuantificación del color de imágenes se han probado tres imágenes distintas. Además se ha utilizado la técnica del codo para estimar cuál es el número óptimo de clusters para cada imagen.

3.1. Comparación de imágenes resultantes

A continuación se muestran varias ejecuciones del algoritmo para tres imágenes distintas con valores de K (número de clusters):

$$K \in \{2, 4, 8, 16, 32\}$$

.

Figura 3: Smallparrot original

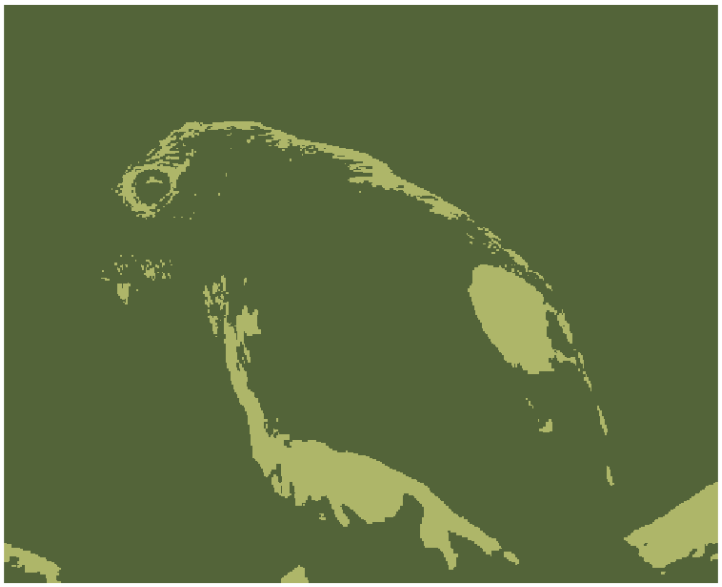

Figura 4: Smallparrot con K = 2

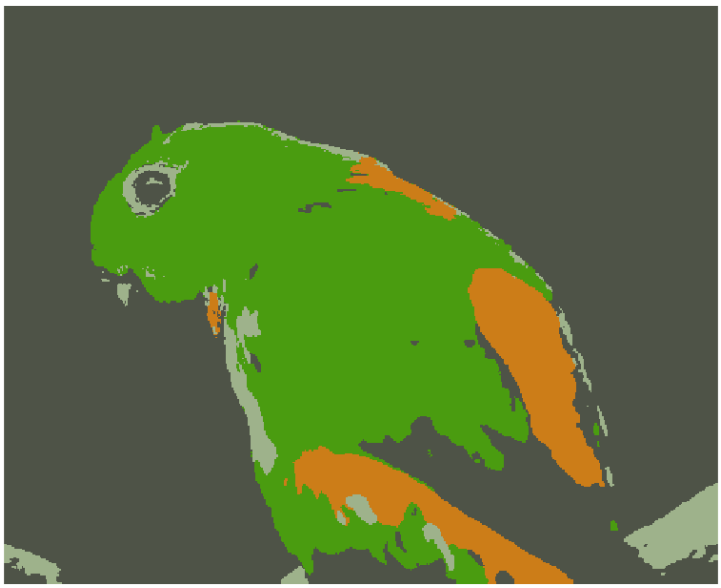

Figura 5: Smallparrot con K = 4

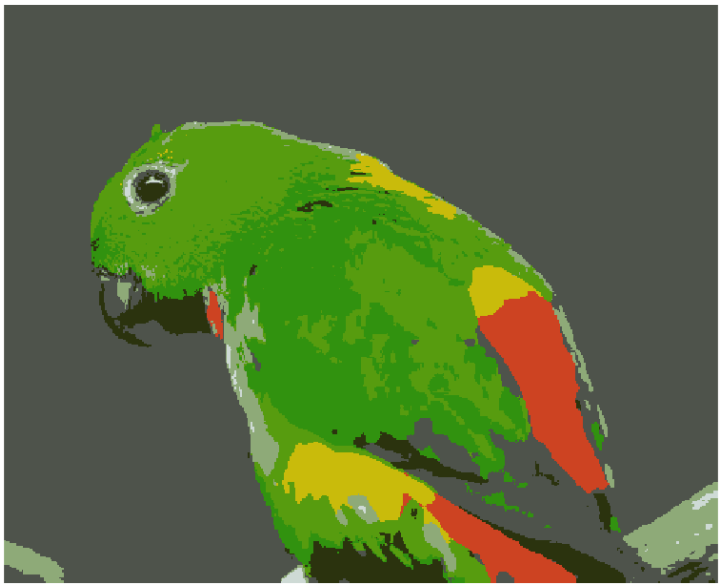

Figura 6: Smallparrot con K = 8

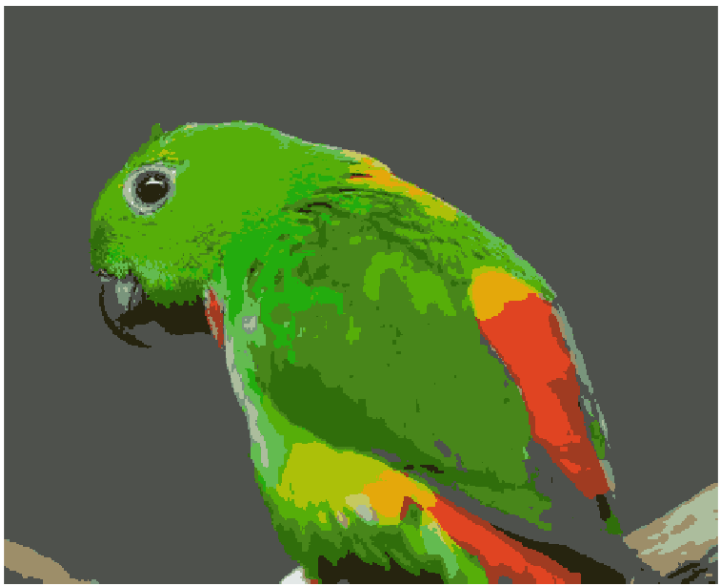

Figura 7: Smallparrot con K = 16

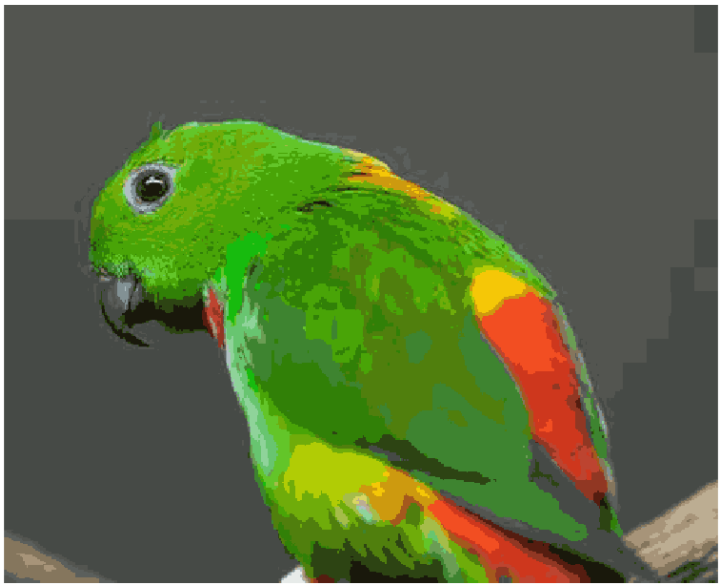

Figura 8: Smallparrot con K = 32

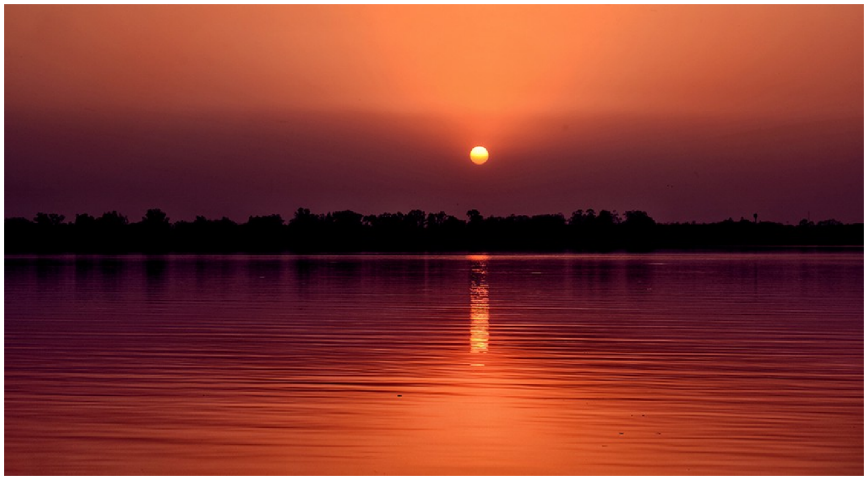

Figura 9: Atardecer original

Figura 10: Atardecer con K = 2

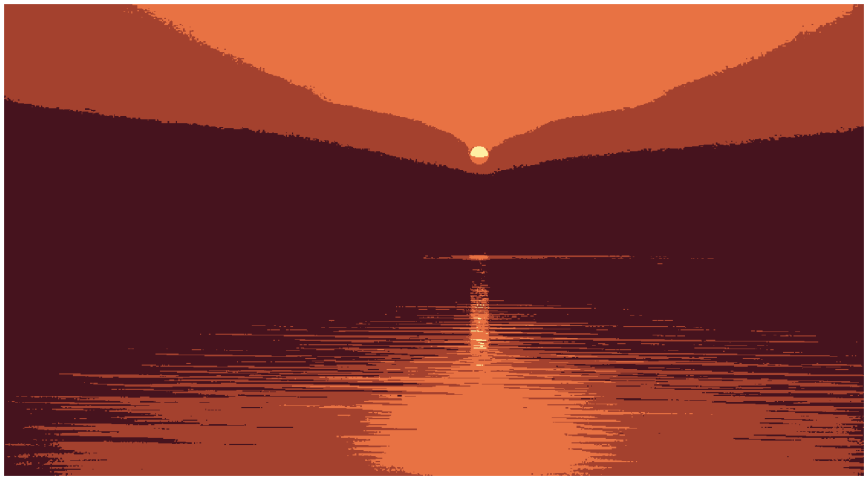

Figura 11: Atardecer con K = 4

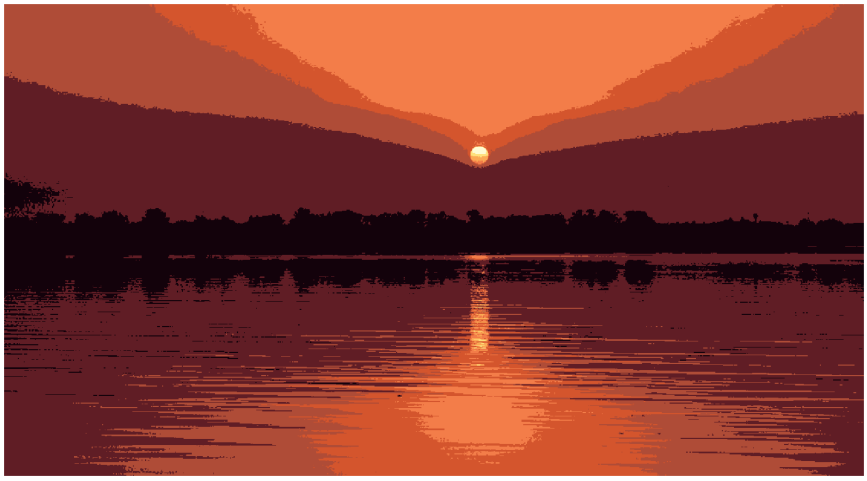

Figura 12: Atardecer con K = 8

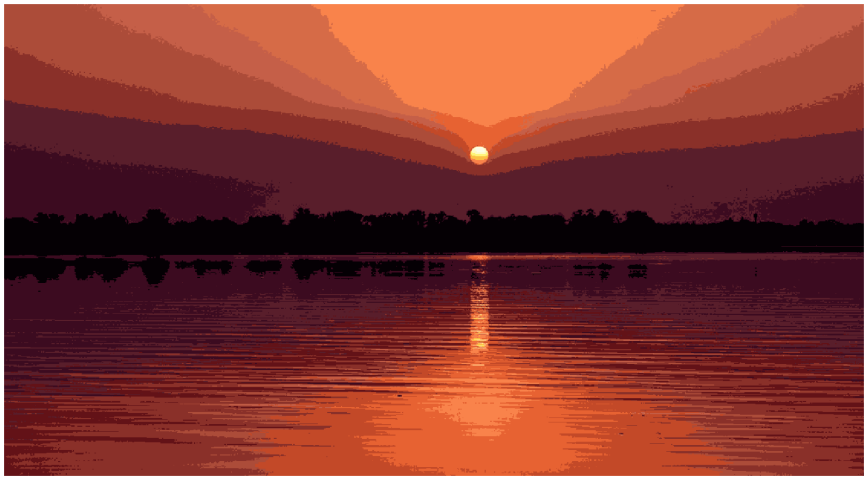

Figura 13: Atardecer con K = 16

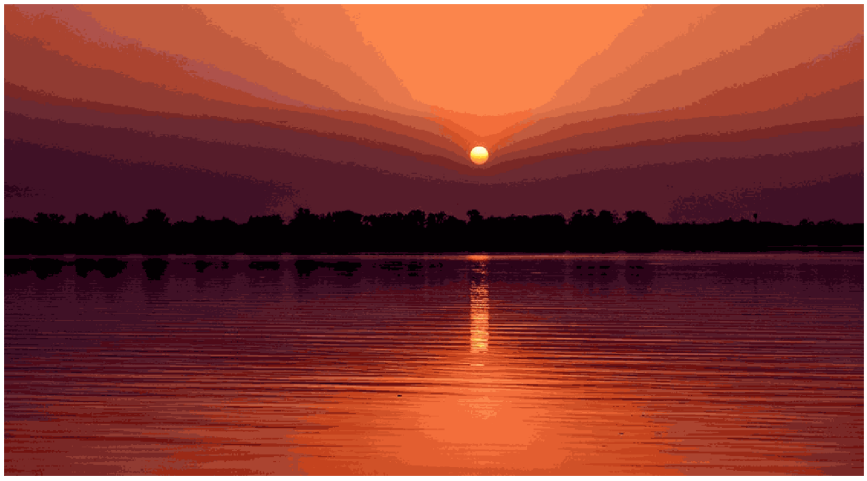

Figura 14: Atardecer con K = 32

Figura 15: El Color original

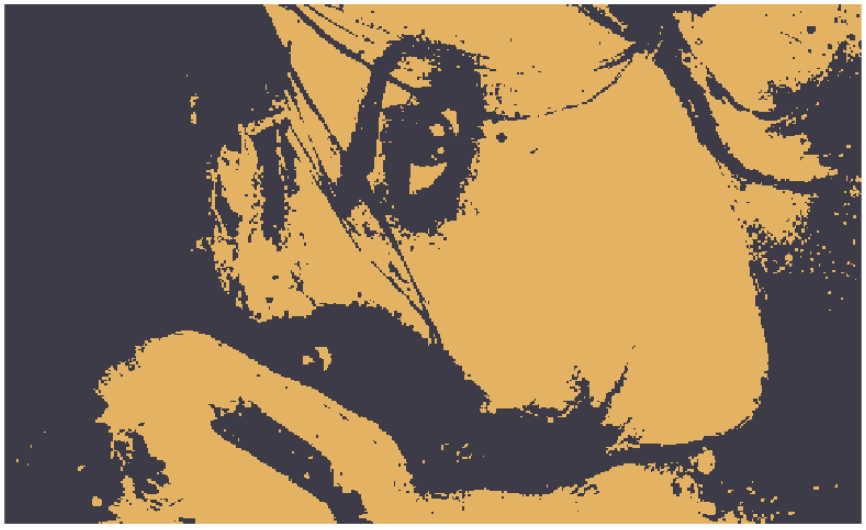

Figura 16: El Color con K = 2

Figura 17: El Color con K = 4

Figura 18: El Color con K = 8

Figura 19: El Color con K = 16

Figura 20: El Color con K = 32

3.2. Elección del número de clusters K óptimo para cada clase

Para elegir cuántos clusters utilizar se utiliza el método del "codo", que consiste en obtener la gráfica de la función de distorsión (J) con respecto al número de clusters (K), buscando el punto en el que se genera el "codo". Este punto es en el cual la reducción de J empieza a ser menor. Esto se hace para conseguir un balance entre el error que se producen en las imágenes reconstruidas y el tiempo de ejecución del algoritmo.

En las figuras 21, 22 y 24 se muestran dichas gráficas para la imagen "Smallparrot", "Atardecer" y "El Color" respectivamente.

La función de distorsión J se define de la siguiente manera:

$$\frac{1}{m}\sum_{i=1}^{m}\|x^{(i)} - \mu_{c^{(i)}}\|^2 \tag{1}$$

Es decir, es el promedio de las distancias de cada muestra a su correspondiente centroide, elevadas al cuadrado.

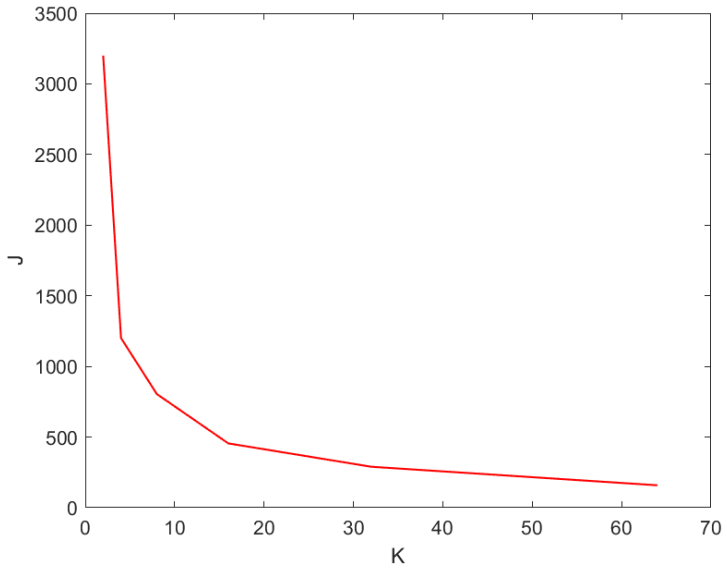

Figura 21: Función de distorsión en función de K para la imagen "Smallparrot". El codo se encuentra en K = 16

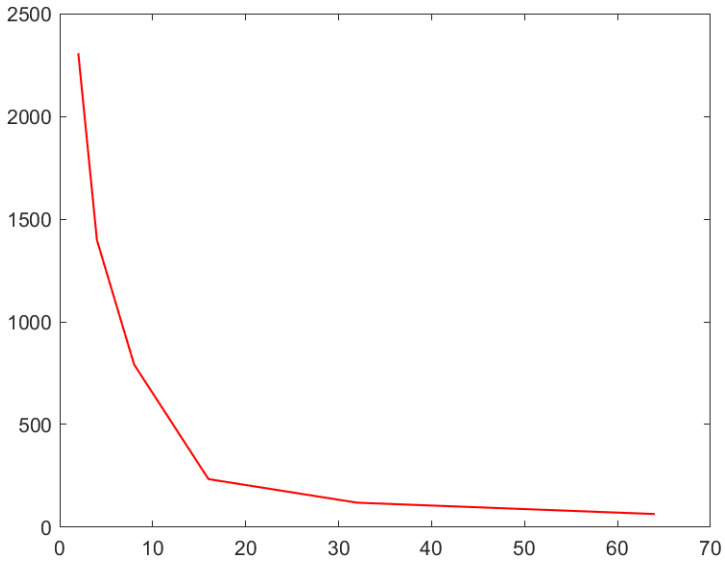

Figura 22: Función de distorsión en función de K para la imagen "Atardecer". El codo se encuentra en K = 16

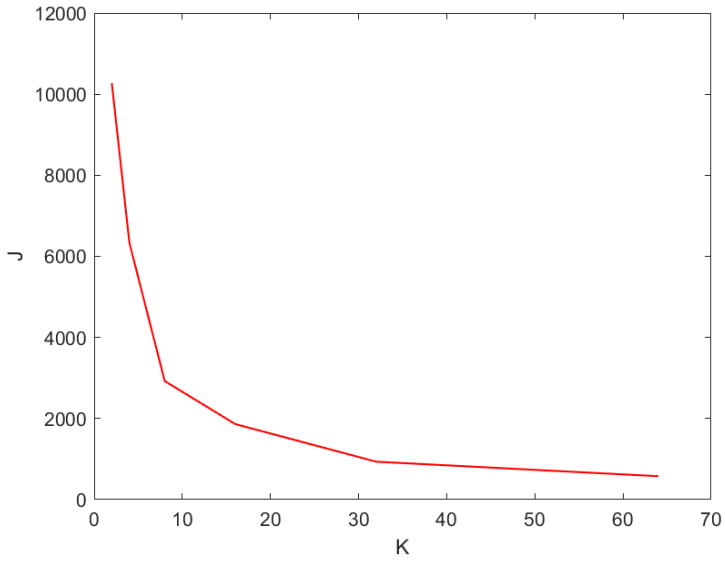

Figura 23: Función de distorsión en función de K para la imagen "El Color". El codo se encuentra en K = 8

3.3. Ganancia en compresión y error en reconstrucción

Una vez seleccionado el número de clusters, se va a estudiar la ganancia en compresión conseguida en relación al error de reconstrucción de la imagen. Para esto se va a coger como ejemplo la imagen "Smallparrot", que tal y como se ha visto en el apartado anterior, se van a escoger K = 16.

La imagen "Smallparrot" tiene un total de 186240 píxeles con 3 canales de color, lo que resulta en $186240 * 3 = 558720$ componentes. Tras ejecutar K-means, el tamaño se reduce al tamaño de 16 clusters + 186240 etiquetas, lo que resulta en $16 * 3 + 186240 = 186288$ componentes. Esto supone un ahorro de espacio del 66,66. %

En cuanto al error de reconstrucción de la imagen es de J = 453,18. Tal y como se estima en la Figura 21. Se puede observar como es un error bastante pequeño.

Por último cabe destacar que, además de la ganancia en compresión y el error de reconstrucción, el tiempo de ejecución para obtener la nueva imagen es también relevante. De echo, la ganancia en compresión es muy favorable y no va a variar demasiado, siempre que se cumpla que $K \ll m$. La razón por la que no se escoge un valor de K mayor, no es otra que el tiempo de ejecución.

En el cuadro 1 se muestra, además de la ganancia en compresión y el error de reconstrucción, el tiempo de ejecución del algoritmo de K-means para la imagen "Smallparrot" con diferentes valores de K. En la Figura **??** se muestra la evolución del tiempo de ejecución con respecto a K. Como se puede ver, el tiempo de ejecución aumenta linealmente con el valor de K.

4. Conclusiones

En esta práctica se ha comprobado que el algoritmo K-means inicializado con la heurísitca "furthest" aplicado en el contexto de la quantificación de color produce imágenes muy similares a las originales (como no ocurre en la inicialización de centroides aleatoria), ahorrando más del 65 % de espacio de almacenamiento.

K	Compresión (%)	Error reconstrucción	T (s)
2	66,67	3198,54	6,06
4	66,66	1205,73	8,44
8	66,66	678,88	13,74
16	66,66	453,18	24,17
32	66,65	269,03	45,19
64	66,63	166,51	90,95

Cuadro 1: Porcentaje de compresión, error de reconstrucción y tiempo de ejecución de la imagen resultante de aplicar K-means a la imagen "Smallparrot"

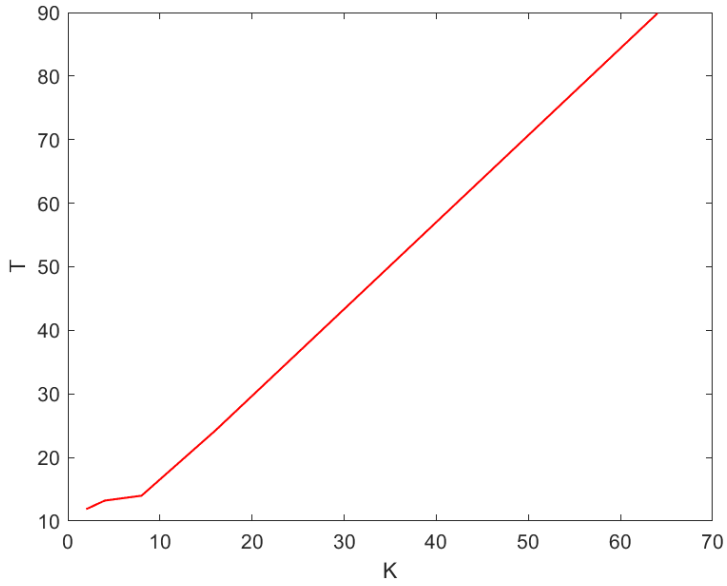

Figura 24: Tiempo de ejecución de K-means en función del valor de K