

Universidad de Murcia

Curso 2020/21

Grado de Ingeniería Informática, 4º curso

Facultad de Informática

**Compresión Multimedia**

Juan José Morell Fernández

juanjose.morellf@gmail.com

24418612H

Índice

[1. Resumen 3](#_Toc60933558)

[2. Introducción 4](#_Toc60933559)

[3. Metodología 5](#_Toc60933560)

[3.1 Procedimiento de compresión y descompresión 5](#_Toc60933561)

[3.2 Tipos de compresión y Factor de calidad 10](#_Toc60933562)

[3.3 Imágenes y Factores de calidad seleccionados 11](#_Toc60933563)

[3.4 Manual de usuario 14](#_Toc60933564)

[4. Resultados Experimentales 15](#_Toc60933565)

[4.1 Imagen 1 17](#_Toc60933566)

[4.2 Imagen 2 20](#_Toc60933567)

[4.3 Imagen 3 22](#_Toc60933568)

[4.4 Imagen 4 24](#_Toc60933569)

[4.5 Imagen 5 26](#_Toc60933570)

[4.6 Imagen 6 29](#_Toc60933571)

[5. Otro Software. Photoshop 31](#_Toc60933572)

[5.1 Imagen 1 32](#_Toc60933573)

[5.2 Imagen 2 33](#_Toc60933574)

[5.3 Imagen 3 35](#_Toc60933575)

[5.4 Imagen 4 36](#_Toc60933576)

[5.5 Imagen 5 37](#_Toc60933577)

[5.6 Imagen 6 39](#_Toc60933578)

[6. Conclusiones 40](#_Toc60933579)

[7. Bibliografía 41](#_Toc60933580)

# 1. Resumen

En esta práctica se tratará de estudiar la compresión de imágenes, intentando discernir el procedimiento de este de forma teórica y posteriormente implementarlo en el programa Matlab.

Constará de dos tipos de compresiones, que serán probadas sobre seis imágenes. Finalmente, se compararán los datos obtenidos en el experimento con otro software, Photoshop.

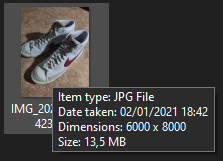
# 2. Introducción

Una imagen normalmente está representada por una matriz, donde cada uno de sus valores corresponde a cada pixel y simboliza la intensidad de estos. Si el valor es alto significa que ese pixel tiene un nivel superior de brillo, mientras que si es un valor pequeño el nivel de brillo también lo será. Asimismo, las imágenes con color presentan diferentes canales por cada uno de los componentes de color existente. Los componentes más típicos son el rojo, verde y azul; formando la representación más conocida de las imágenes, RGB.



*Figura 1 – Espacio de color RGB*

Aunque esta representación de las imágenes es la más común, no es como se almacenan en un disco duro. Por ejemplo, almacenar una imagen con color de 48 megapíxeles conlleva guardar un total de 48 millones de valores por cada uno de los canales, sumando un total de 144 millones de valores. Esto dependiendo del método de almacenamiento puede llegar a ocupar un total de 144 MB. Pero en la realidad, una imagen de esas características llega a ocupar menos de 14 MB. ¿Cómo se consigue lograr esto?



*Figura 2 – Imagen de ejemplo, tomada con una cámara de 48 MP.*

Justamente esta es la pregunta que hay que conseguir responder en este proyecto. Cuál es el proceso que lleva a una imagen, que supuestamente tendría que ocupar una gran cantidad de megabytes, pueda ocupar tan poco en disco.

La respuesta a esta pregunta es la compresión de imágenes, en este caso JPEG. Esta es una de las extensiones más conocidas mundialmente en las imágenes, y es un formato de compresión de imágenes.

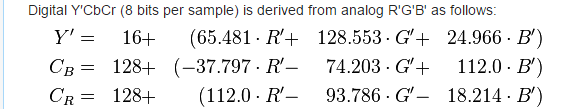
# 3. Metodología

## 3.1 Procedimiento de compresión y descompresión

JPEG es un método de compresión con pérdida, lo que significa que la imagen final que se almacena en nuestro ordenador no tiene toda la información original. Cuanta más información se descarte en el proceso de compresión peor calidad tendrá la imagen comprimida. Este trato entre la calidad y la ocupación de la imagen en disco será uno de los puntos importantes dentro de este proyecto, y será comentado junto a las imágenes que se mostraran en el apartado 3.2.

En general, JPEG consigue una buena relación tamaño/calidad, ya que los datos que va a despreciar van a ser aquellos que no son realmente visibles por la vista humana.

El primer paso de la compresión es la conversión del espacio de color. Para no almacenar la imagen con un tamaño de 114 MB, se cambia el espacio de color RGB, a uno llamado YCbCr. Este se caracteriza por tener un canal que representa la intensidad de luz, y otros dos representan los colores, el azul y el rojo. Esto permite la separación entre la luminancia (Y) y la crominancia (Cb, Cr).

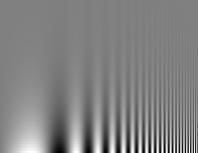


*Figura 3 – Transformación de espacio RGB a YCbCr.*

La transformación del espacio RGB a YCbCr se trata de una transformación lineal, que puede ser expresada mediante una multiplicación de matrices.

Las imágenes actuales son mucho más sensibles a los cambios de brillo que al de colores, por lo que se reducen los tamaños de los canales de color. Gracias al espacio YCbCr, se puede hacer esto sin problemas mediante el proceso de *downsample*, o también conocido como *chroma subsampling*.

Además de este cambio, otro factor importante a la hora de reducir el tamaño de las imágenes son las características de la vista humana. La idea reside en que los objetos o detalles pequeños que están representado en la imagen son más propensos a pasar desapercibidos que objetos grandes, a la vista humana. Este efecto se denomina *sensibilidad de contraste dependiente de la frecuencia.*



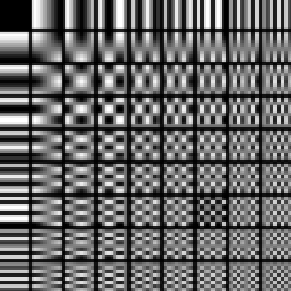
*Figura 4 – Efecto de la frecuencia espacial*

Un buen ejemplo de este fenómeno sería una imagen con líneas negras donde cada vez se van juntando más hasta que parece que no existe separación, de izquierda a derecha. Mientras que el contraste va incrementando desde abajo hacia arriba de la imagen. Con esto se puede apreciar una curva en donde se desvanecen las líneas, y se debe a que nuestro ojo es más sensible a las variaciones donde el brillo sea mayor.

Se puede apreciar en la parte superior de la figura 4 que las líneas verticales casi no son visibles. Esto se debe a que se trata de la parte que presenta menor contraste y mayor frecuencia. Estos fenómenos nos dan espacio para poder comprimir más una imagen, ya que la vista humana no detecta tanta información.

JPEG lleva a cabo esto mediante la división de la imagen en bloques de 8 x 8 y cuantizandolos posteriormente, de forma individual, en una representación del dominio de frecuencias. Cada uno de estos bloques es comparado con un conjunto de 64 patrones de frecuencias donde la frecuencia espacial aumenta desde la esquina superior izquierda hasta la esquina inferior derecha.

Este proceso descompone la imagen en componentes de frecuencia, convirtiendo cada bloque de 8x8, que representaba el nivel de brillo de la imagen, en otro bloque de 8 x 8 que representa la presencia de un componente de frecuencia en particular. Este método se conoce como *discrete cosine transform* o *DCT*.

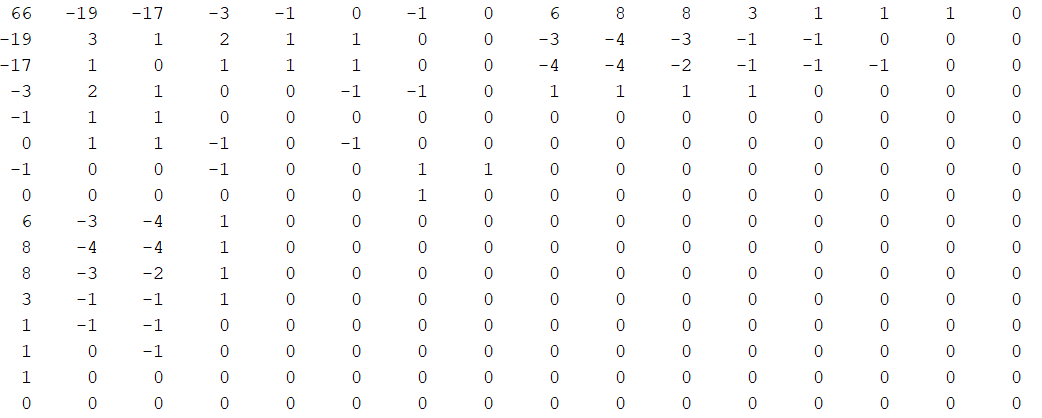


*Figura 5 – Conjunto de 64 patrones de frecuencias DCT*.

Mediante esta representación se pueden comprimir las frecuencias que son menos visibles al ojo humano, dividiendo la matriz del bloque 8 x 8 entre una matriz constante. Estas frecuencias menos visibles son divididas por constantes más grandes, al contrario que en las frecuencias más visibles que son divididas por constantes menores.

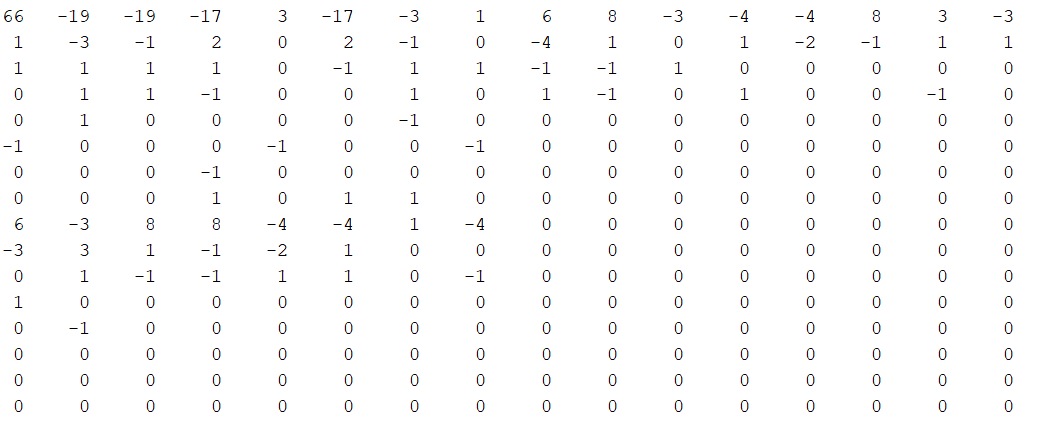
En nuestro caso, se utilizan las matrices de cuantización de JPEG por defecto, tanto para luminancia como para crominancia. También son conocidas por el nombre de ITU T.81.

El siguiente paso es la cuantización, que trata de redondear el resultado de la división de las matrices a los enteros más cercanos. Mediante el uso de números grandes el resultado de su división se acercará más a cero, lo que resulta en un ratio de compresión mayor, pero reduce también la calidad de la imagen. En nuestro proyecto, se podrá controlar mediante una variable de entrada llamada factor de calidad.



*Figura 6 – Matriz tras la cuantización*

Tras la cuantización, obtenemos unas matrices donde existen una gran cantidad de ceros en las frecuencias altas. Pero como se puede apreciar en la figura 6, los ceros están intercalados con diferentes números. Para ordenar el contenido de la matriz y poder almacenarlo de una forma más eficiente se reordenan los coeficientes mediante un recorrido en zigzag desde la esquina superior izquierda a la esquina inferior derecha.



*Figura 7 – Matriz tras recorrido en zigzag*.

Una vez juntados los ceros, en vez de almacenar la matriz así se puede almacenar su valor junto al número de veces que aparece consecutivamente. Un ejemplo es que, si aparecen cinco ceros seguidos, en vez de almacenar los cinco, se almacena un 5 junto a un 0, representando que existen cinco ceros consecutivos. A esta técnica se le conoce como *Run-Length encoding*.

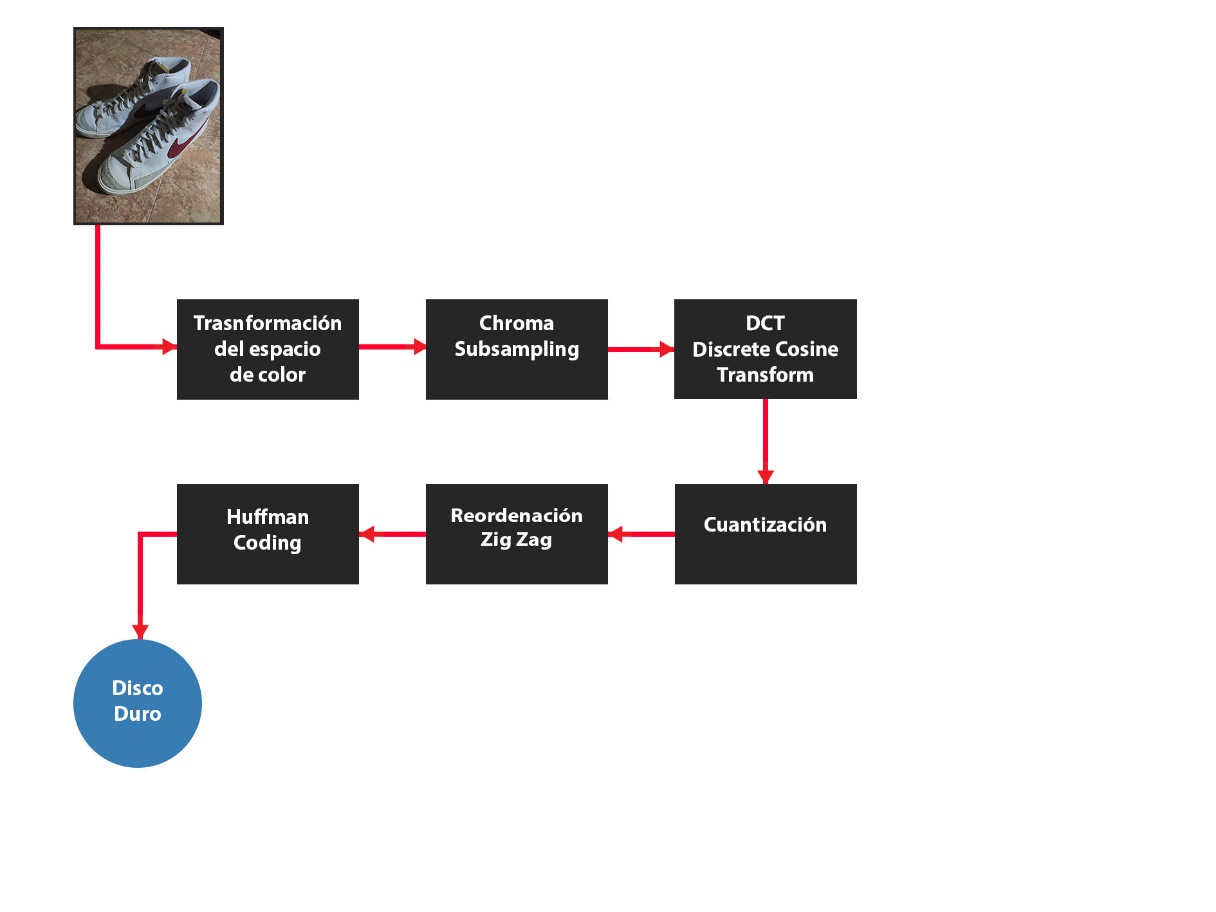
Otra forma de comprimir esto es teniendo en cuenta la frecuencia de aparición de cada elemento en la matriz, almacenando con más bits aquellos valores con más presencia y menor cantidad al resto. Esto consigue reducir el número de bits por símbolo. Esta técnica es conocida por *Entropy coding* o *Huffman coding*.

Ambas técnicas se recogen en el grupo de codificación sin perdidas, ya que no se pierde nada de la información original a la hora de comprimir. Se centran en almacenar los datos de una manera eficiente aprovechando la redundancia en los datos. Asimismo, estas técnicas también son utilizadas en otros ámbitos, como texto, audio o video.

Esto se basa en la idea de la entropía, específicamente en la entropía de Shannon, que trata de calcular la cantidad de incertidumbre existente en una imagen asociándolo en una distribución de probabilidad. En otras palabras, sirve como una medida de desorden. Si esta medida aumenta, la entropía también lo hará y convertirá a la información de la imagen menos predecible.

Todo este ejemplo se ha realizado mediante el programa creado tras el desarrollo de la práctica y una imagen en formato *bmp* de tamaño 10 x 10.

En definitiva, el proceso de compresión de una imagen toma el siguiente camino:



*Figura 8 – Esquema de compresión de una imagen*

Para la decodificación de la misma imagen comprimida sería el proceso inverso. Debido a que parte de la información original se ha perdido tras los procesos de subsampling o cuantización, la imagen decodificada no va a ser idéntica a la original. Aun así, si la comprensión de la imagen se ha realizado con un correcto factor de calidad no se debería apreciar ninguna diferencia. Cuanto más alto sea dicho factor más notorios serán los efectos de la compresión.



*Figura 9 – Comparación entre diferentes compresiones*

En esta figura no se podrá apreciar del todo bien el cambio entre imágenes, ya que dicha imagen esta también comprimida en formato JPEG. Si se desean ver las imágenes a su calidad total, se facilitarán las copias originales junto a esta documentación.

## 3.2 Tipos de compresión y Factor de calidad

Dentro del proyecto se desarrollan dos tipos de compresión / descompresión que se nombraran como: default y custom. Las diferencias entre estos son la forma de aplicar *Huffman encoding* durante el proceso de compresión.

El primero de ellos, default, realiza un Huffman encoding guiado por un valor constante, sin tener en cuenta los valores que va a codificar. Mientras que, Huffman encoding custom, trata de mejorar esto teniendo en cuenta la frecuencia de aparición de los datos que se van a codificar.

Por tanto, teóricamente el segundo de los compresores va a ser más eficiente. Esto se tratará en el apartado 4, junto a los datos reales de las pruebas.

Otro aspecto importante es el factor de calidad, parámetro principal de ambas compresiones implementadas, ya que especifica cuanto se va a comprimir una imagen o cuanta calidad se va a perder. Durante el proceso de experimentación se realizarán pruebas con varios valores para este parámetro para conocer cuál es el más eficiente dentro del ratio compresión / calidad.

## 3.3 Imágenes y Factores de calidad seleccionados

Todas las imágenes que se utilizan en esta práctica provienen de un [repositorio](https://data.mendeley.com/datasets/sp4g8h7v8k/1)[[1]](#footnote-1) de imágenes con formato bmp sobre naturaleza y aves.



*Figura 10 – Imagen 1*

Imagen 1: Esta imagen presenta a un pájaro junto a un fondo de color verde, y puede presentar un comportamiento interesante a la hora de comprimir un color solido de fondo y un objeto que resalta bastante. Otro fenómeno que tratar sería el cambio brusco entre los colores del pájaro y fondo.



*Figura 11 – Imagen 2*

Imagen 2: Esta segunda imagen presenta un escenario parecido a la anterior, pero con la particularidad de que el fondo no es homogéneo, sino todo lo contrario. El objetivo con esta imagen es estudiar el comportamiento del proceso de compresión en una imagen donde los elementos tienen un color parecido, pero con gran cantidad de detalles.



*Figura 12 – Imagen 3*

Imagen 3: La característica principal de esta imagen es la simetría y estudiar cómo se comporta dentro del proceso de compresión, si se trata de algún factor clave a la hora de eliminar redundancia o no.



*Figura 13 – Imagen 4*

Imagen 4: Al igual que la anterior, se intenta seguir estudiando el fenómeno de la simetría. Pero en este caso, al existir cierta semejanza con los patrones de frecuencia presentes en DCT, el objetivo es averiguar que tan bien funciona la compresión con patrón de franjas y diferentes colores.



*Figura 14 – Imagen 5*

Imagen 5: El objetivo con esta imagen es estudiar el comportamiento del compresor cuando existe un color dominante, pero con detalles.



*Figura 15 – Imagen 6*

Imagen 6: En esta imagen tiene como objetivo estudiar el comportamiento de la compresión de una imagen que contenga texto, y como trata los contornos de este. Además, presenta vegetación en la parte inferior que añade ruido a la imagen.

## 3.4 Manual de usuario

Para el desarrollo del proyecto se ha realizado uso de un script propio, que permite realizar la comprensión y descompresión, tanto para la versión *default* como *custom*, de una imagen para distintos factores de calidad.

Con esto se facilita el proceso de obtención de las variables MSE y RC, y las gráficas que se van a estudiar en el apartado 4.

# 4. Resultados Experimentales

En este apartado se va a realizar el estudio de los resultados obtenidos después de aplicar la compresión y descompresión sobre las imágenes. En este se reflejarán los resultados producidos por los dos tipos de compresión: default y custom, aplicados juntamente con una serie de factores de calidad.

Dichos factores de calidad han sido seleccionados de forma que se pueda estudiar con facilidad los cambios de las relaciones de compresión y el error cuadrático medio. Teniendo en cuenta que un factor de calidad igual a 1 es la mínima compresión posible y 1000 multiplica por 10 la compresión por defecto, se han seleccionado los siguientes factores de calidad para realizar el estudio:

[1, 25, 50, 100, 250, 500, 1000]

Los factores de calidad se pueden dividir en tres grupos:

* FC < 100: Un factor de calidad menor a 100 se caracteriza por una compresión de mayor calidad. Por consiguiente, la relación de compresión que se obtiene no es muy alta.
* FC = 100: Simboliza la calidad de compresión estándar.
* FC > 100: Con valores superiores a 100 se obtiene una calidad de compresión cada vez menor. Por eso, se alcanzan relaciones de compresión superiores.

Los parámetros que se van a tener en cuenta para el estudio, como se ha comentado anteriormente, son: la relación de compresión y el error cuadrático medio.

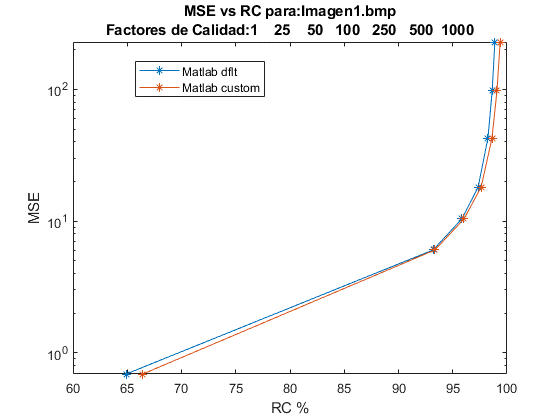
La relación de compresión es una medida que representa cuanto es capaz de comprimir un algoritmo de compresión. La definición que se usa en este proyecto es la siguiente:

Donde TO es el tamaño original de la imagen y TC es el tamaño del archivo comprimido. Esta medida se expresa en porcentaje (%).

Asimismo, el error cuadrático medio, MSE, mide la cantidad de error que existe entre dos conjuntos de datos, en este caso imágenes. Para obtener esto se comparan los pixeles de la imagen original con los de la imagen comprimida.

En cada apartado se mostrará una gráfica con la relación entre relación de compresión y MSE, además se añadirá una tabla donde se reflejen los valores de cada uno de ellos en cada factor de calidad. Y, por último, la evolución de las imágenes y la diferencia entre la imagen original y la comprimida con factor de calidad de 1000.

## 4.1 Imagen 1



*Figura 16 – Gráfica Imagen 1*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FC | MSE | DEFAULT | MSE | CUSTOM | RC | DEFAULT | RC | CUSTOM |
| **1** | 0.6902 | 0.6902 | 64.9002 | 66.4337 |
| **25** | 6.0353 | 6.0353 | 93.1874 | 93.3494 |
| **50** | 10.3787 | 10.3787 | 95.8372 | 96.0345 |
| **100** | 18.116 | 18.116 | 97.3429 | 97.6133 |
| **250** | 42.6655 | 42.665 | 98.2421 | 98.6184 |
| **500** | 97.7043 | 97.7043 | 98.6356 | 99.0764 |
| **1000** | 229.0251 | 229.0251 | 98.8801 | 99.3711 |

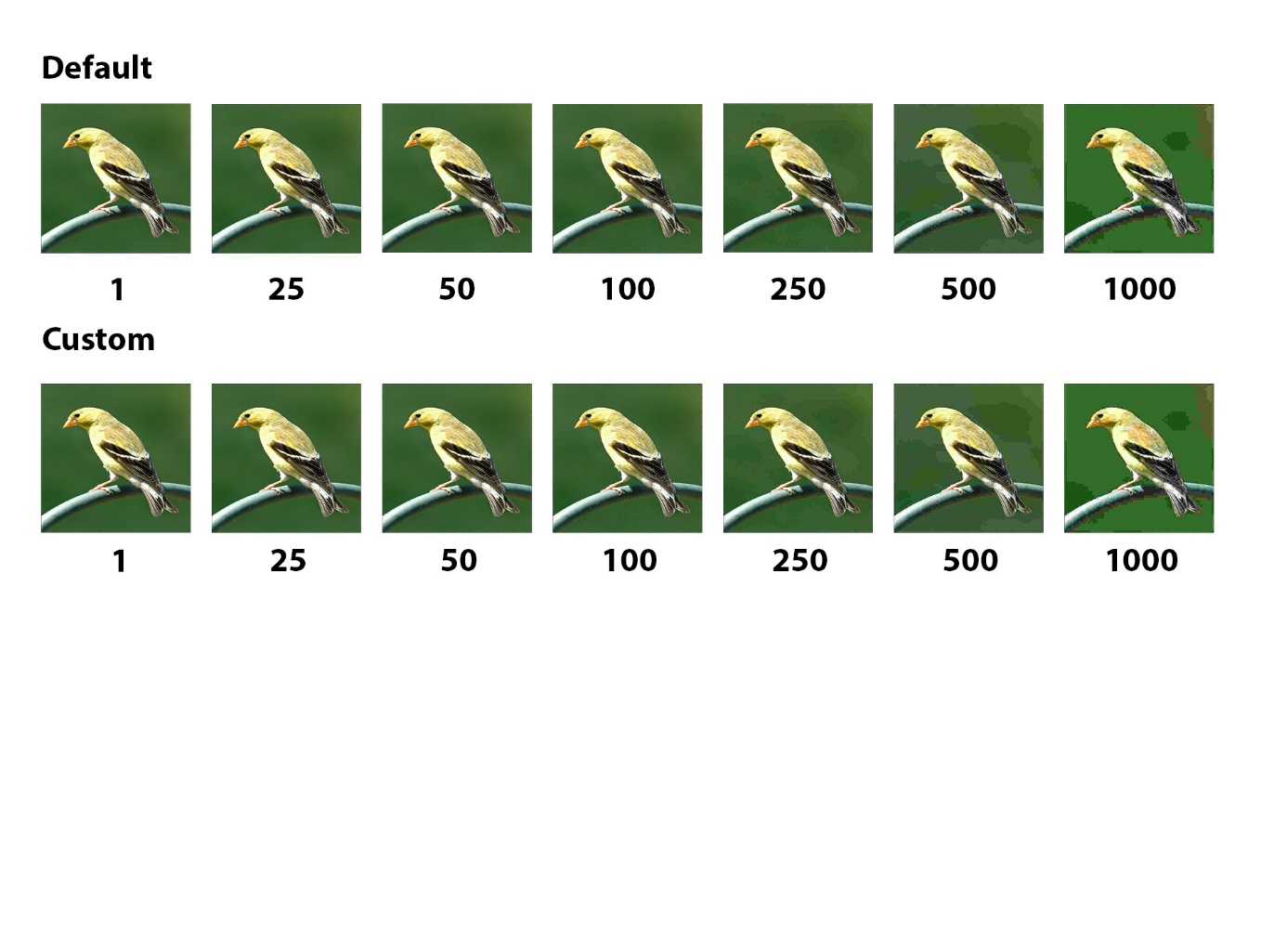
Podemos apreciar que la imagen empeora según aumenta el factor de calidad, como se ha explicado en el apartado anterior. Lo interesante es que el error cuadrático medio de todos los resultados es idéntico, y esto se debe a que la imagen que se obtiene como resultado de comprimir y descomprimir la imagen original es idéntica tanto en la compresión default como en compresión custom. Esto se debe a que la diferencia entre ambos compresores es la forma de realizar *Huffman encoding*, parte del proceso en la que no se pierde información.

Asimismo, los valores obtenidos en la relación de compresión siempre son superior para la compresión custom que para el tipo de compresión default.

El resultado obtenido con el factor de calidad a 1 es una imagen que es casi idéntica a la original, ya que el error cuadrático medio es menor a 1 (0.6902). Y en los factores de calidad superior aumenta exponencialmente, destacando que para un valor de factor de calidad por defecto se obtiene un MSE de 18 y consiguiendo una relación de compresión, semejante para ambos métodos, de 97%.

Una vez superado el factor de calidad 100, el MSE aumenta muchísimo en cada paso, alcanzando un máximo de 229. Mientras que la relación de compresión se mantiene con un crecimiento estable, solamente aumenta un 2% respecto con los valores obtenidos con un factor de calidad 100.

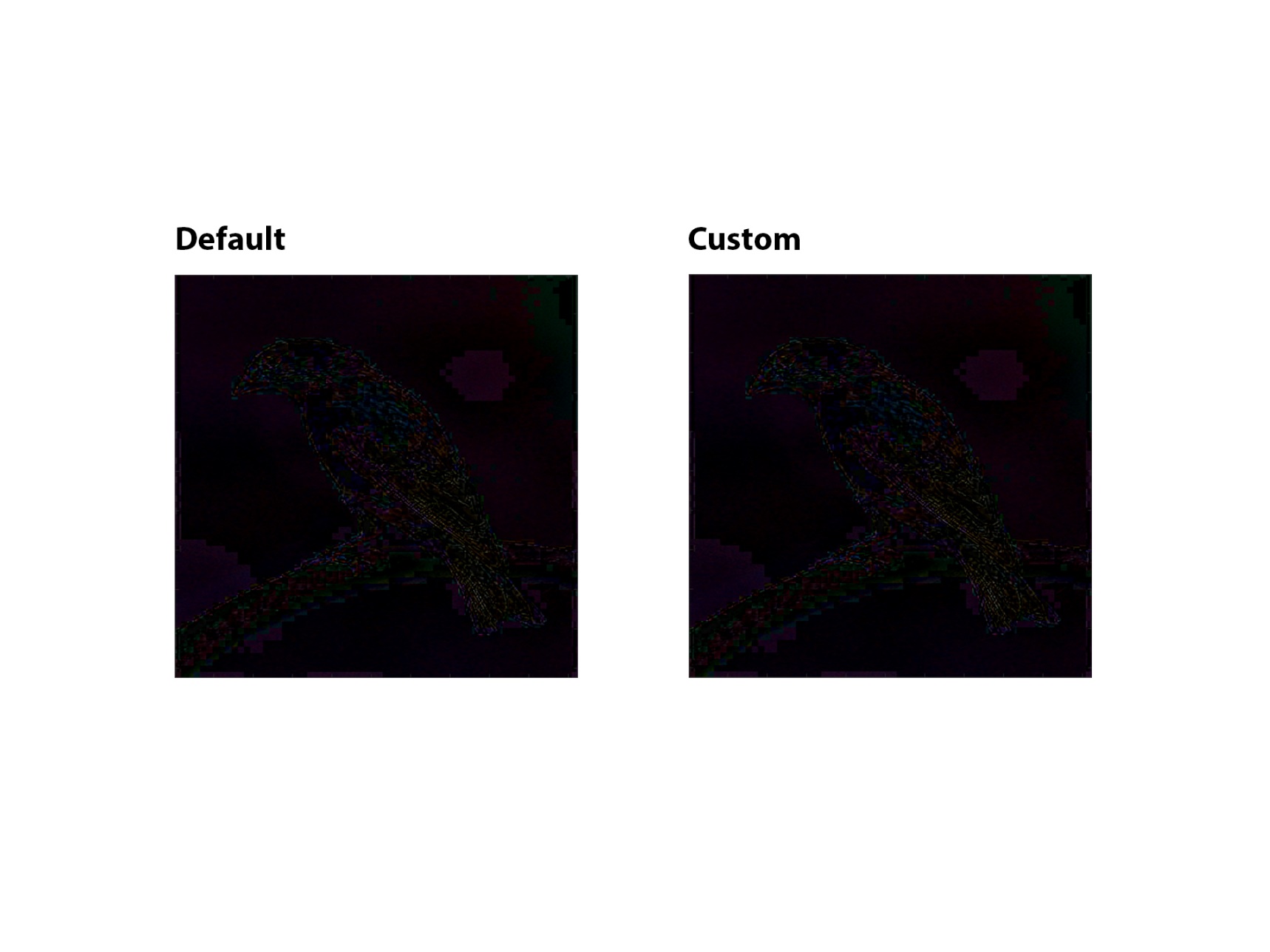
Para facilitar el entendimiento de los resultados, se disponen las imágenes resultantes de ambos compresores:



*Figura 17 – Evolución de la compresión Imagen 1*

Se puede apreciar que para los factores de calidad entre 1 y 100 se asemejan visualmente a la imagen original. Mientras que, para factores de calidad superior a 100 se empieza a notar como en el fondo se crea un color verde sólido y unos contrastes entre zonas bastante visible.

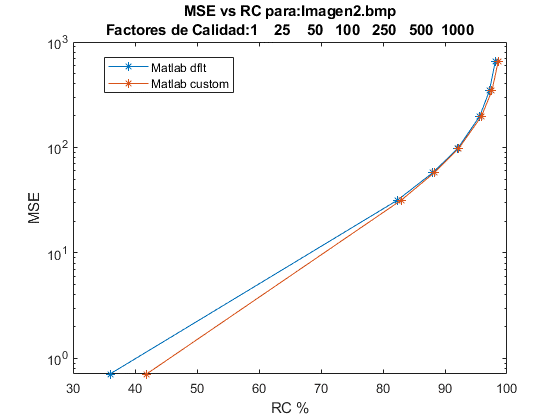
Igualmente, estos saltos se pueden apreciar más fácilmente observando las imágenes de diferencia entre la original y la imagen que ha sido comprimida, tanto por default como por custom.



*Figura 18 – Imagen diferencia entre Imagen original y comprimida con FC 1000*

Por tanto, se puede concluir que una imagen donde hay un color de fondo semejante y saltos bruscos de color se va a obtener la mayoría del error en los bordes de estos cambios, como se aprecia en la figura. Y se obtiene un resultado similar para ambos tipos de compresores.

## 4.2 Imagen 2



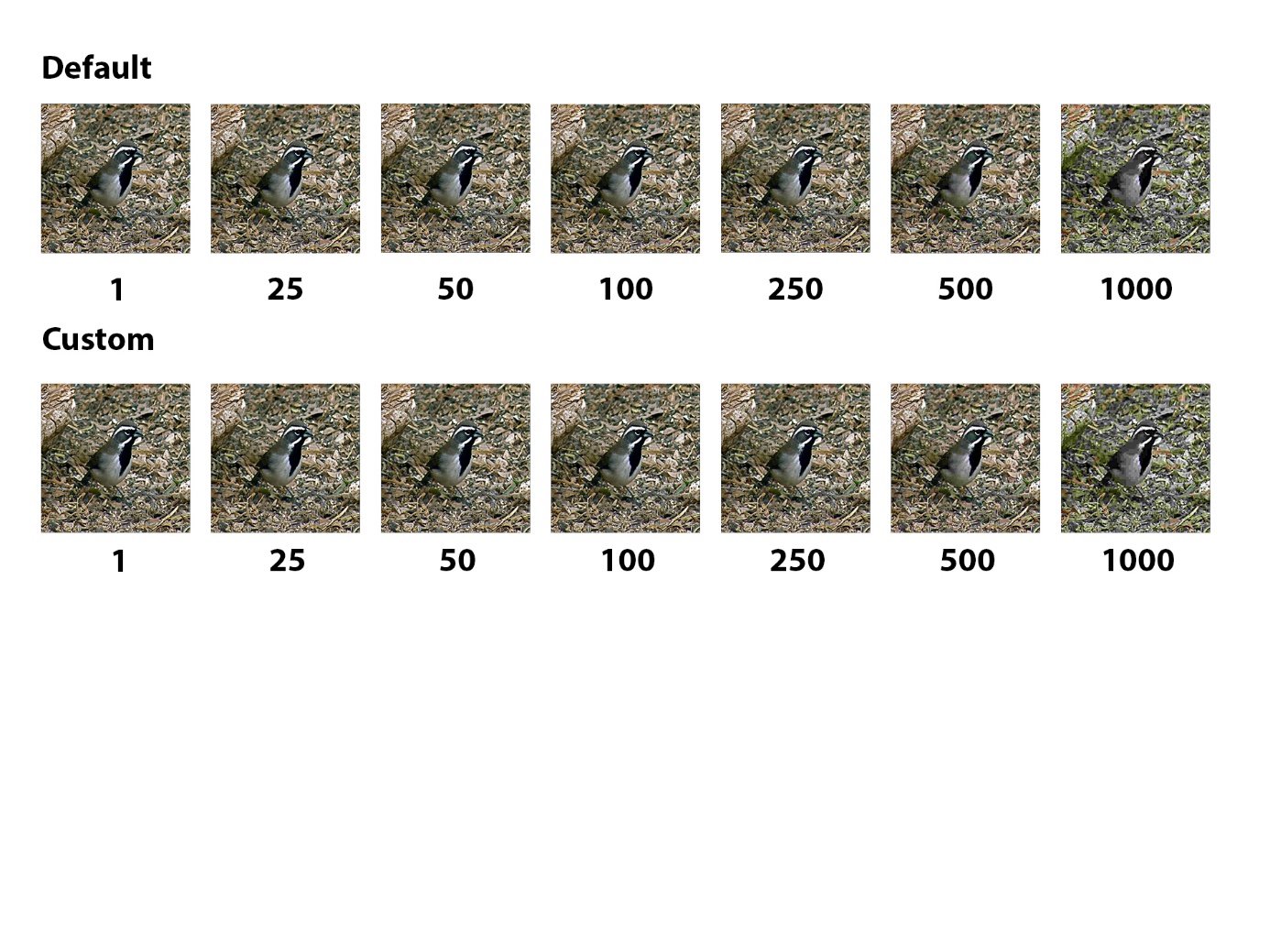
*Figura 19 – Gráfica Imagen 2*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FC | MSE | DEFAULT | MSE | CUSTOM | RC | DEFAULT | RC | CUSTOM |
| 1 | 0.7111 | 0.7111 | 35.9714 | 41.8698 |
| 25 | 31.7015 | 31.7015 | 82.3525 | 82.9575 |
| 50 | 57.7807 | 57.7807 | 88.0428 | 88.2856 |
| 100 | 97.6127 | 97.6127 | 92.0297 | 92.1962 |
| 250 | 198.3503 | 198.3503 | 95.6089 | 95.8995 |
| 500 | 346.0193 | 346.0193 | 97.1755 | 97.5721 |
| 1000 | 655.3356 | 655.3356 | 98.1176 | 98.5897 |

La imagen 2 se caracteriza por un fondo con mucho ruido y vegetación con detalle. Observando la gráfica, se aprecia que los resultados con factores de calidad mayores a 1 tienen un tipo de crecimiento exponencial.

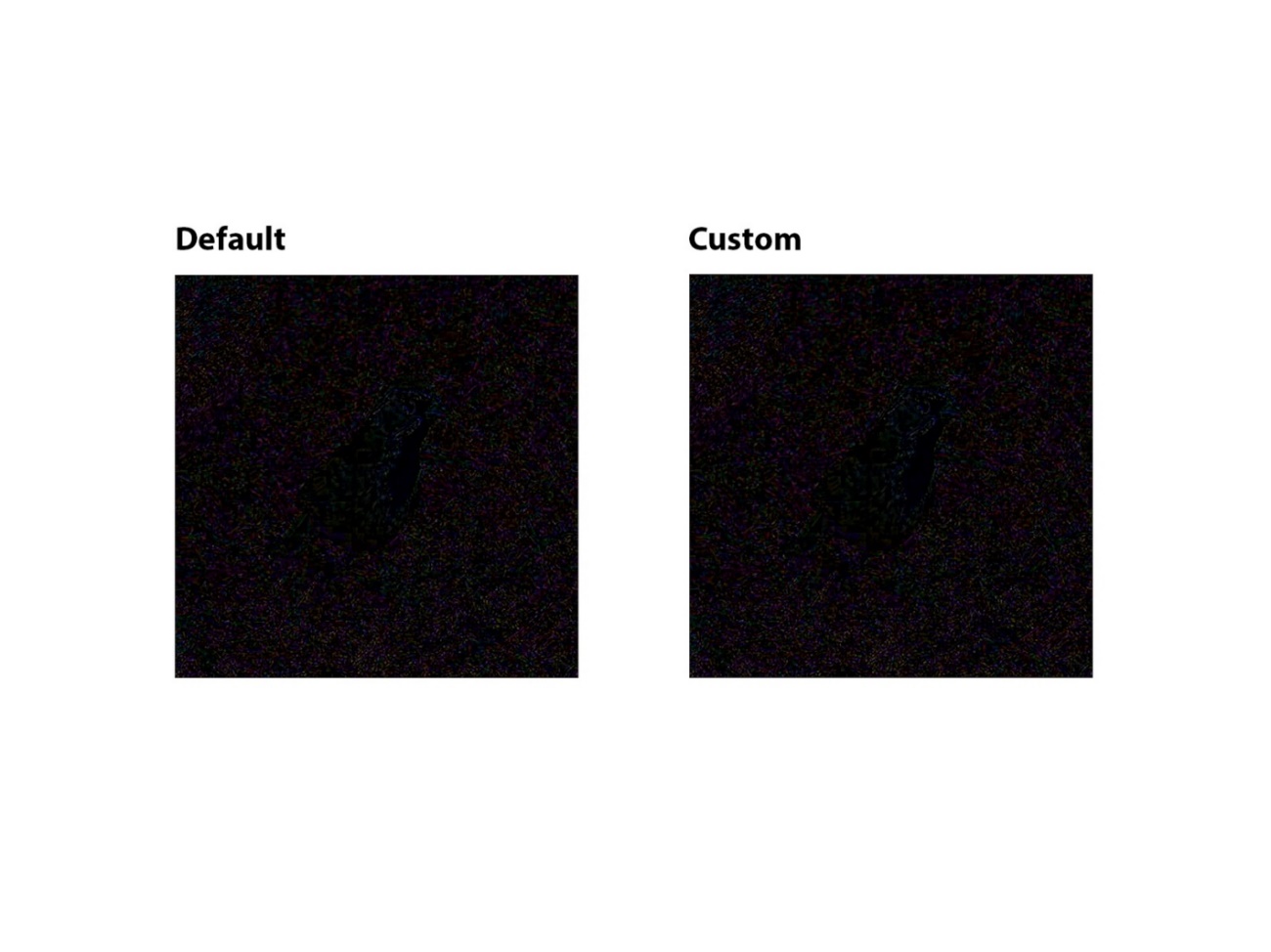
En primer lugar, con los factores de calidad del primer grupo, < 100, no se sobrepasa un MSE de 58%, aunque la relación de compresión que se logra no es muy elevada. Empieza a superar el rango del 90%, con un factor de calidad igual al de por defecto, 100. Pero, con el inconveniente de que el error cuadrático medio aumenta juntamente con la relación de compresión.

Una particularidad de esta imagen con respecto a la anterior es que se obtiene un mayor valor de error cuadrático medio. Esto se debe a la gran cantidad de información existente en el fondo de la imagen. Para observar este fenómeno se usan las siguientes imágenes, donde se expone la evolución de las imágenes obtenidas de los compresores y la diferencia entre la imagen original y la imagen obtenida a través de una compresión con un factor de calidad de 1000.



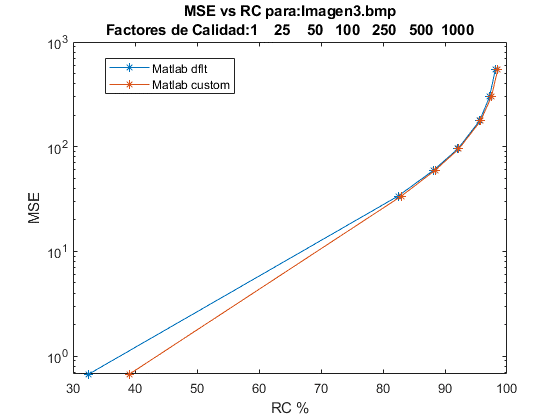
*Figura 20 - Evolución de la compresión Imagen 2*

Se puede apreciar como a partir de la imagen 100 el fondo va cambiando de forma y color, hasta alcanzar en la última imagen un fondo totalmente diferente al de la imagen original. Este fondo se convierte como en un “bosque verde pixelado”, y se puede apreciar mejor en las imágenes de diferencia, donde existe un granulado general en todo el fondo.



*Figura 21 - Imagen diferencia entre Imagen original y comprimida con FC 1000*

## 4.3 Imagen 3



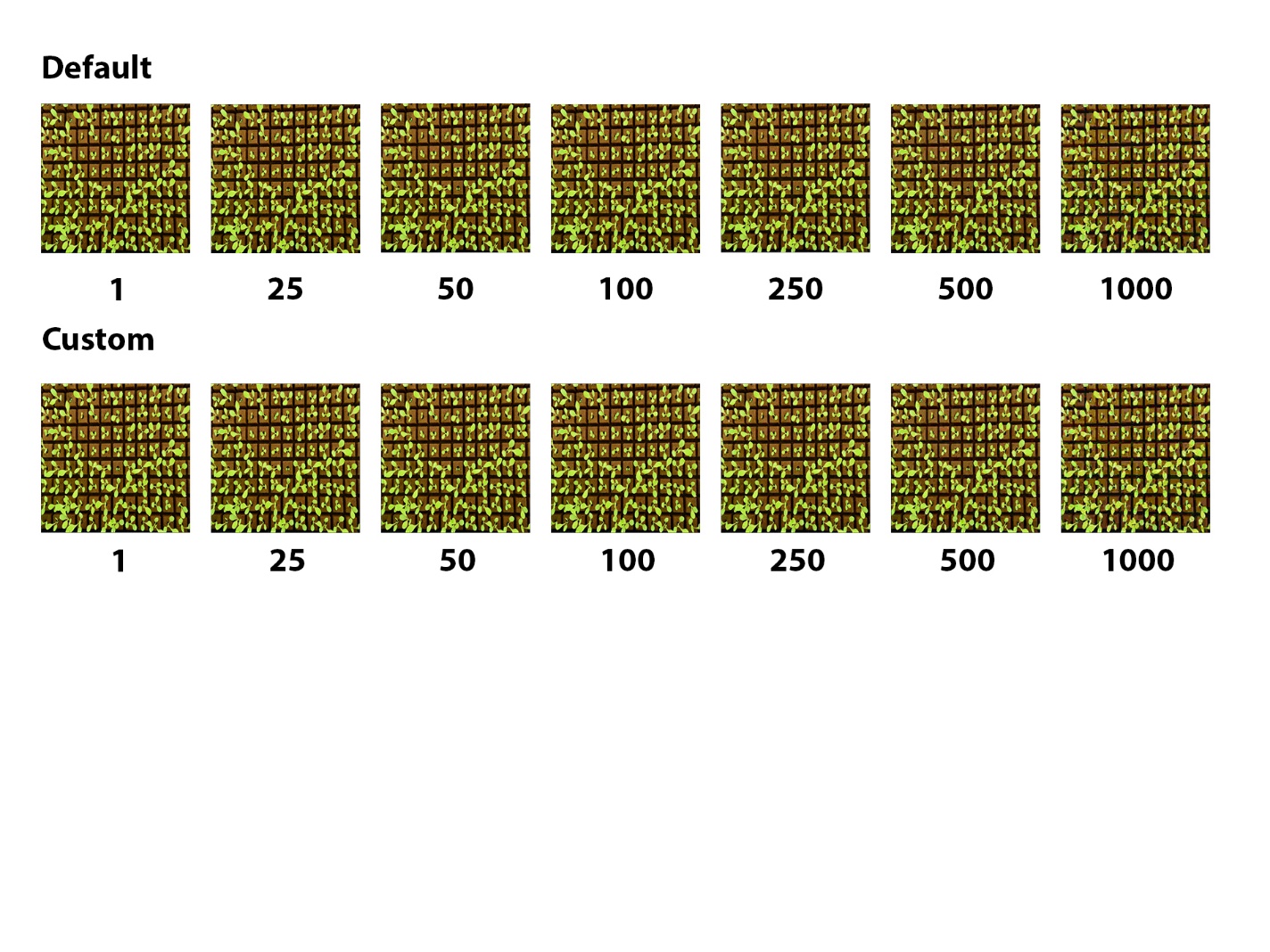
*Figura 22 – Gráfica Imagen 3*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FC | MSE | DEFAULT | MSE | CUSTOM | RC | DEFAULT | RC | CUSTOM |
| 1 | 0.6735 | 0.6735 | 32.5640 | 39.1390 |
| 25 | 33.7043 | 33.7043 | 82.4212 | 82.96 |
| 50 | 59.4832 | 59.4832 | 88.1384 | 88.3939 |
| 100 | 95.1046 | 95.1046 | 92.0545 | 92.2226 |
| 250 | 176.2319 | 176.2319 | 95.5596 | 95.7405 |
| 500 | 298.8728 | 298.8728 | 97.1813 | 97.4738 |
| 1000 | 549.6785 | 549.6785 | 98.1355 | 98.525 |

Los resultados obtenidos en esta imagen son semejantes a los de la imagen anterior, alcanzando un porcentaje superior al 90% a partir del valor por defecto del factor de calidad, 100. Mientras que, para valores superiores se obtiene un error cuadrático medio bastante elevado sin poder mejorar demasiado la relación de compresión obtenida para el valor por defecto.

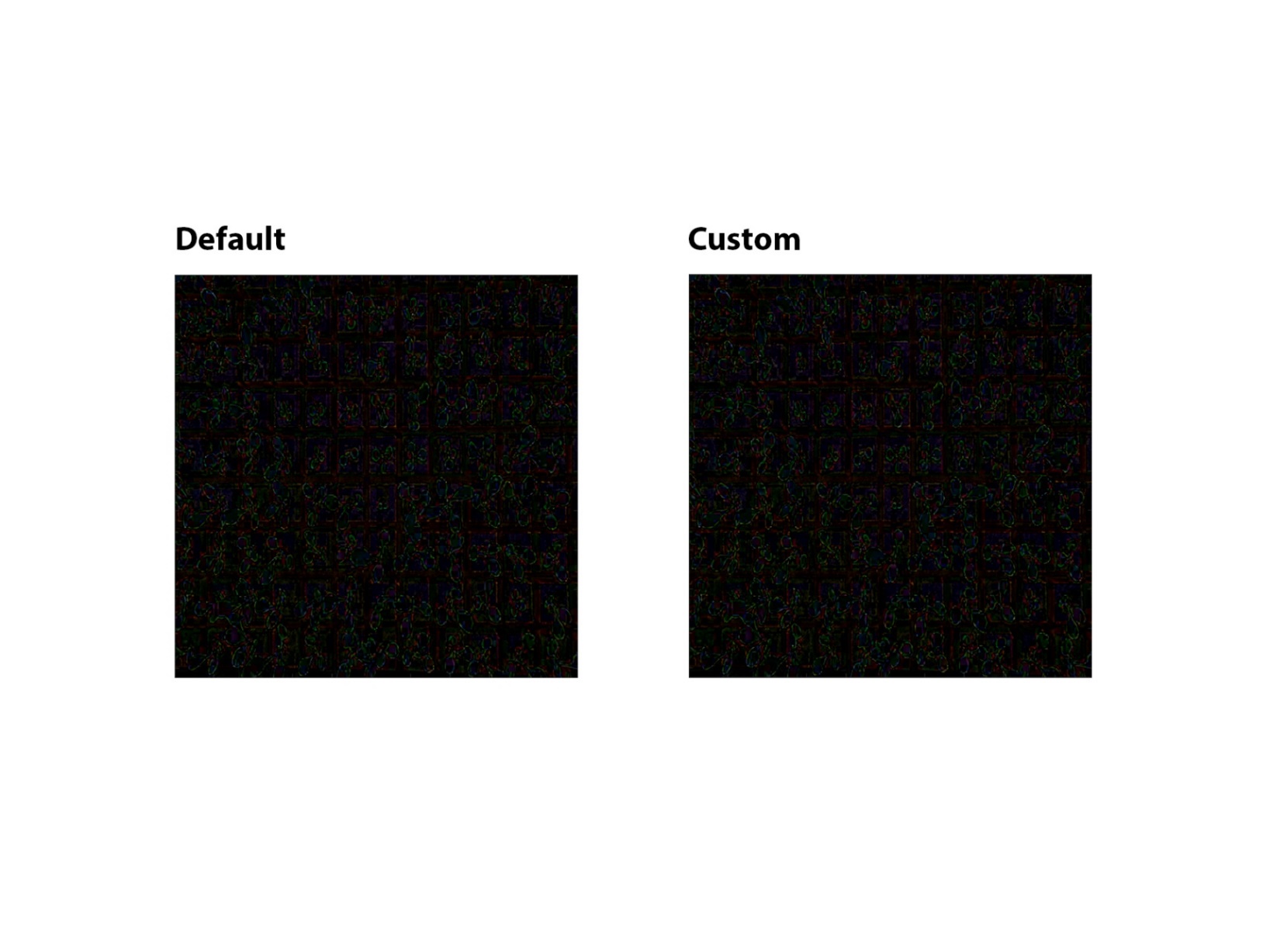
En cuanto a los valores menores a 100, cabe destacar el elevado error cuadrático medio que empieza a aparecer después del factor de calidad 1. Indicando que la imagen a comprimir tiene mucha información diferente en ella y es complicada de comprimir.

Con respecto a las imágenes obtenidas, no se nota gran cambio visualmente. Se aprecia como la calidad de estas desciende cuanta más compresión se realiza, pero sin destacar grandes cambios bruscos entre zonas.



*Figura 23 - Evolución de la compresión Imagen 3*

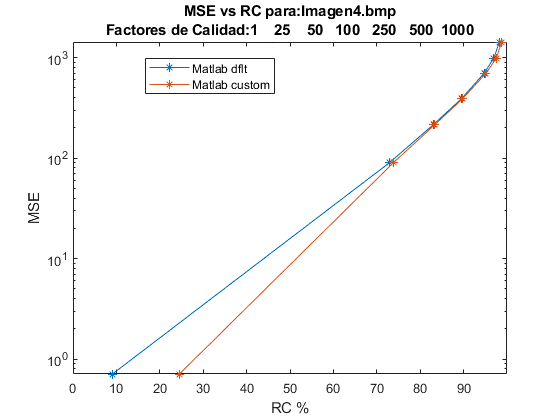
Si nos fijamos en las diferencias obtenidas entre la imagen original y la imagen con la compresión más alta, se pueden observar las formas de la imagen a la perfección. Esto nos indica que hay una gran pérdida de datos durante la compresión.



*Figura 24 - Imagen diferencia entre Imagen original y comprimida con FC 1000*

Sintetizando, la compresión de una imagen no tiene que ser mejor en casos de simétrica, incluso podría ser lo contrario como demuestra esta imagen. Por tanto, la idea inicial de aprovechar la entropía y la repetición de las formas para que el compresor pudiese manejar la redundancia a su favor es errónea para este caso.

## 4.4 Imagen 4



*Figura 25 – Gráfica Imagen 4*

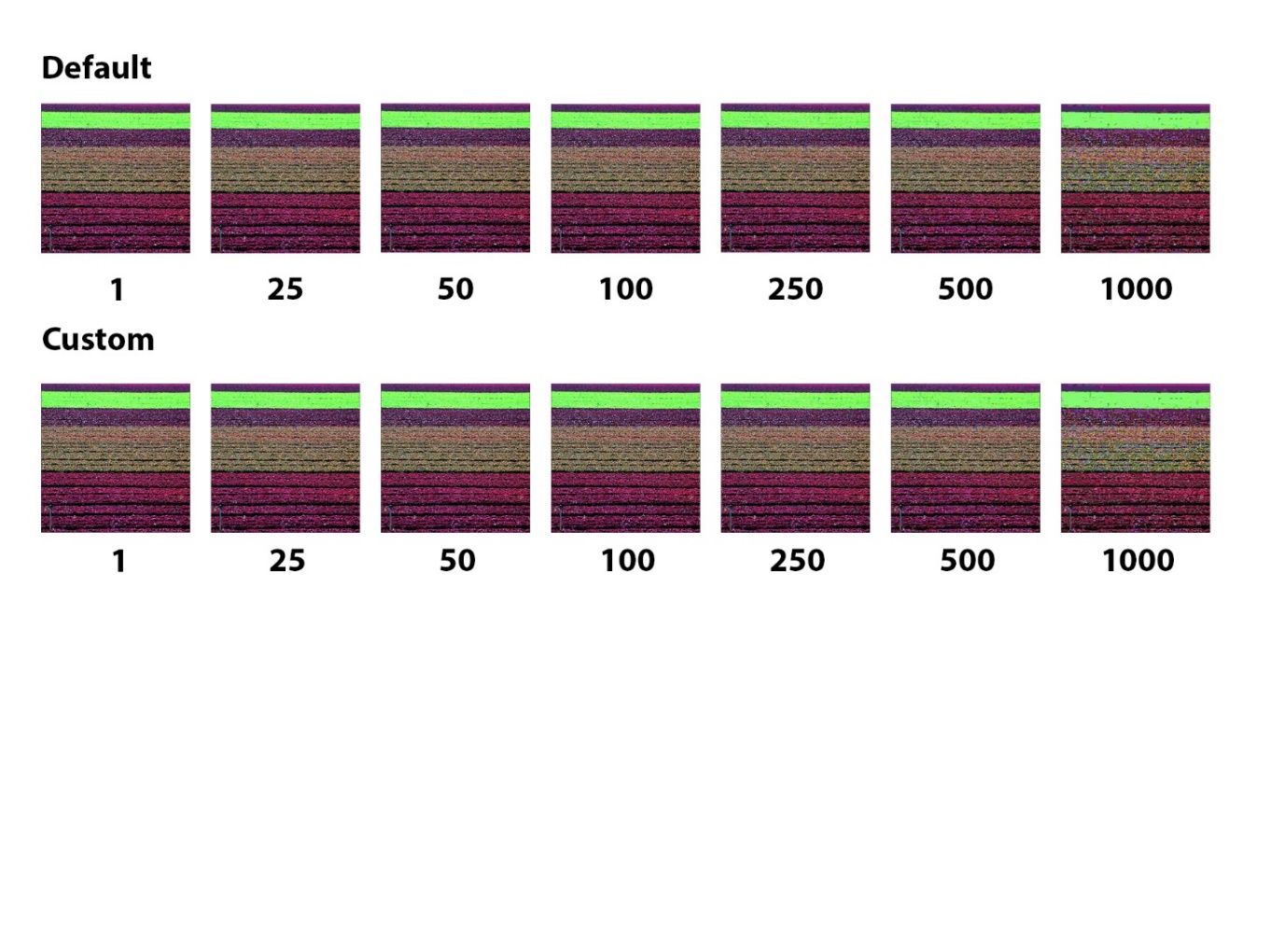
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FC | MSE | DEFAULT | MSE | CUSTOM | RC | DEFAULT | RC | CUSTOM |
| 1 | 0.7126 | 0.7126 | 9.0890 | 24.5258 |
| 25 | 89.5999 | 89.5999 | 72.8602 | 73.7744 |
| 50 | 213.6284 | 213.6284 | 82.9538 | 83.2954 |
| 100 | 388.3704 | 388.3703 | 89.4751 | 89.7534 |
| 250 | 697.9605 | 697.9605 | 94.7237 | 95.1610 |
| 500 | 983.8421 | 983.8421 | 96.9412 | 97.5145 |
| 1000 | 1437.11 | 1437.11 | 98.2586 | 98.8088 |

Al igual que la imagen anterior, en esta se intenta aprovechar los colores lisos, aunque con detalle, y la simétrica de la imagen para conseguir una mejor compresión. En cambio, el resultado que obtenemos vuelve a ser totalmente lo contrario.

Ya en los valores de factor de calidad menores a 100 y diferentes a 1, se empieza a observar un valor de error cuadrático medio bastante alto, a diferencia de imágenes anteriores, teniendo en cuanta que la relación de compresión aun no es muy alta.

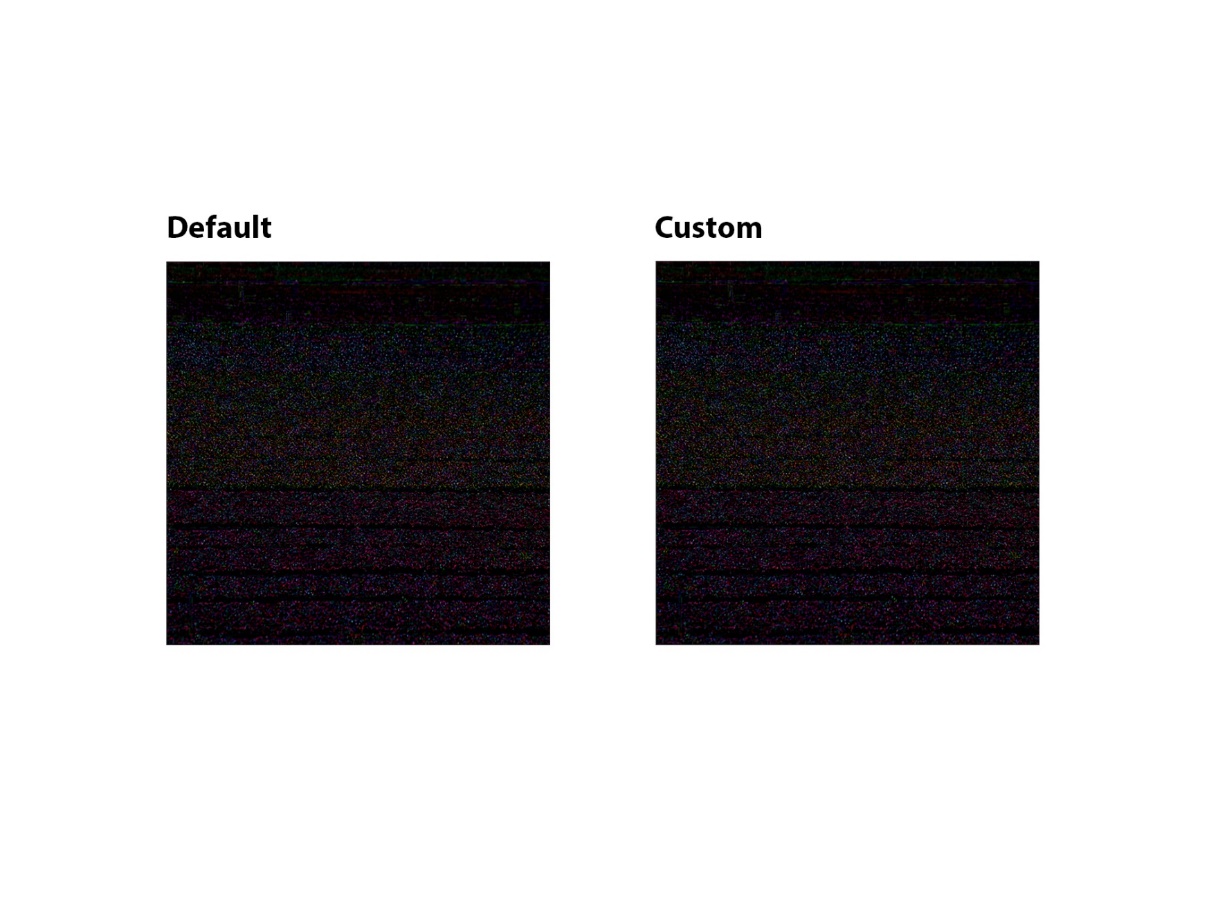
Esto se sigue repitiendo para el valor por defecto y se confirma finalmente en los valores superiores a 100, donde se alcanza un error cuadrático medio de 1437. Lo que nos quiere indica, que la idea inicial esta errónea.

A continuación, se muestra la evolución de las imágenes comprimidas:



*Figura 26 - Evolución de la compresión Imagen 4*

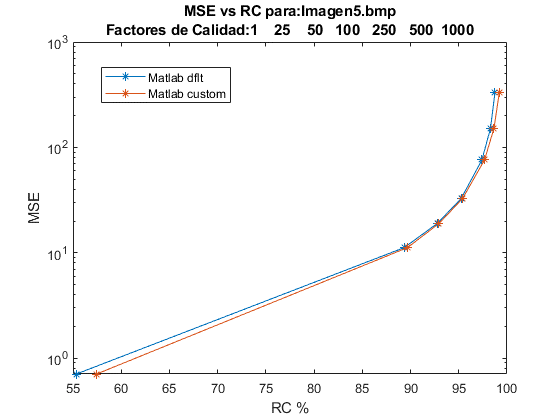
En estas imágenes se puede apreciar la bajada de calidad existente cuanto más vamos aumentando el valor de factor de calidad. Cabe destacar que las franjas superiores del campo se van transformando en un color sólido y sin detalles. Mientras que, el resto de las franjas se convierten en un amasijo de pixeles que mezclan la información de la imagen original. Esto se puede apreciar con mayor facilidad con las imágenes de diferencia:



*Figura 27 - Imagen diferencia entre Imagen original y comprimida con FC 1000*

Como conclusión, reforzados la idea de que la simetría en una imagen no siempre beneficia al compresor y lo que realmente le ayuda es la existencia de colores semejantes en franjas de la imagen. Esto último queda reflejado en esta imagen en las dos franjas superiores del campo.

## 4.5 Imagen 5



*Figura 28 – Gráfica Imagen 5*

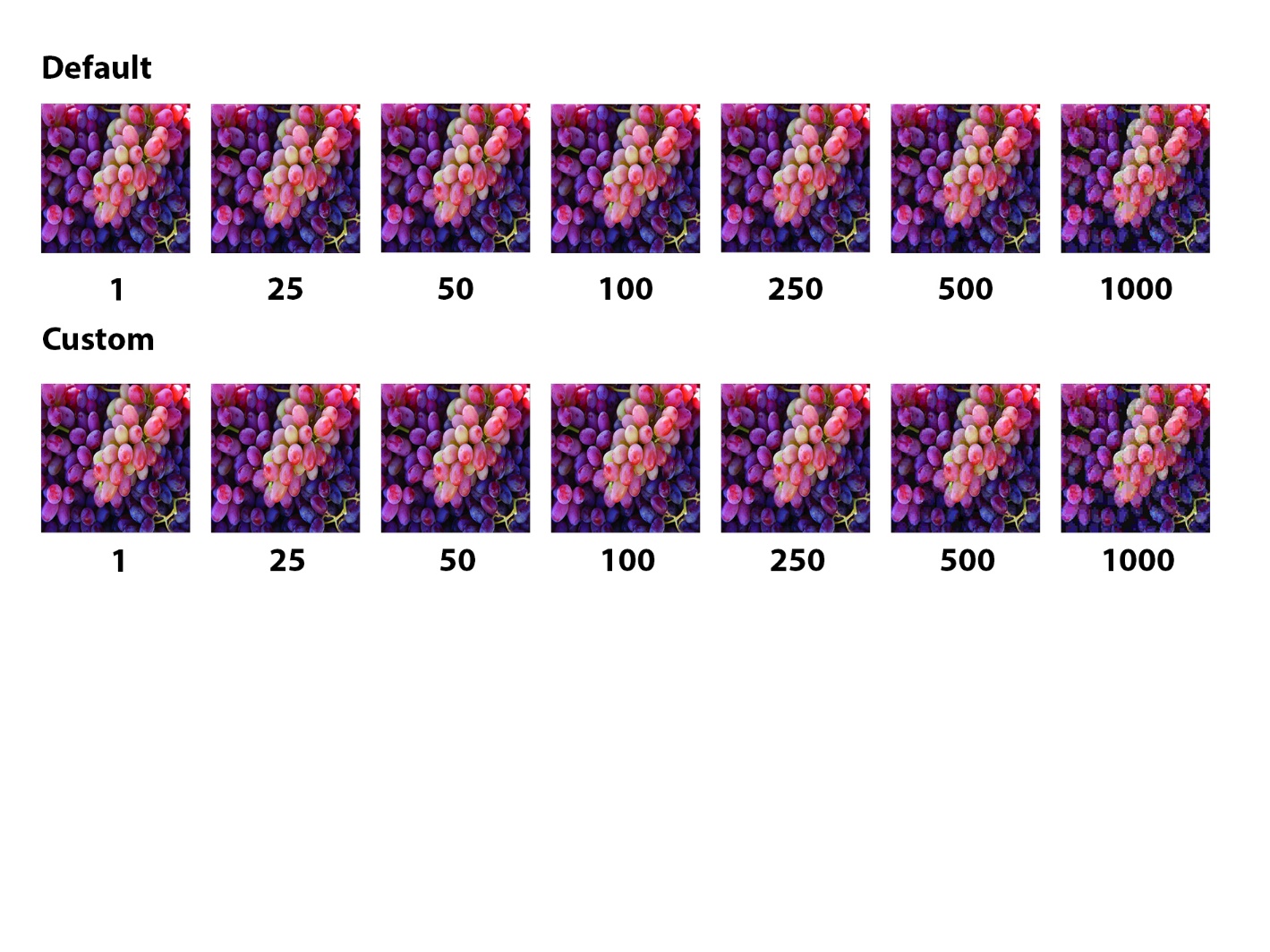
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FC | MSE | DEFAULT | MSE | CUSTOM | RC | DEFAULT | RC | CUSTOM |
| 1 | 0.7067 | 0.7067 | 55.3466 | 57.4507 |
| 25 | 11.1792 | 11.1792 | 89.3327 | 89.6772 |
| 50 | 18.9244 | 18.9244 | 92.7570 | 92.9402 |
| 100 | 32.6870 | 32.6870 | 95.2429 | 95.3936 |
| 250 | 76.0380 | 76.0380 | 97.4025 | 97.6717 |
| 500 | 151.9227 | 151.9227 | 98.2911 | 98.6517 |
| 1000 | 332.5231 | 332.5231 | 98.7543 | 99.2047 |

Los valores del factor de calidad inferiores a 100 obtienen buenos resultados con respecto al error cuadrático medio, indicando que los compresores devuelven imágenes casi idénticas a la original. Aunque, se sigue manteniendo una relación de compresión bastante baja. A diferencia que las demás imágenes, esta es la primera que alcanza un 90% de relación de compresión antes del valor 100 del factor de calidad (50).

El valor por defecto, 100, obtiene una muy buena relación MSE / relación de compresión logrando alcanzar un 95% de compresión con un error cuadrático medio menor a 33%. Esto se debe a que la imagen seleccionada para este experimento tiene un color dominante y se puede aprovechar adecuadamente la idea de la redundancia de datos.

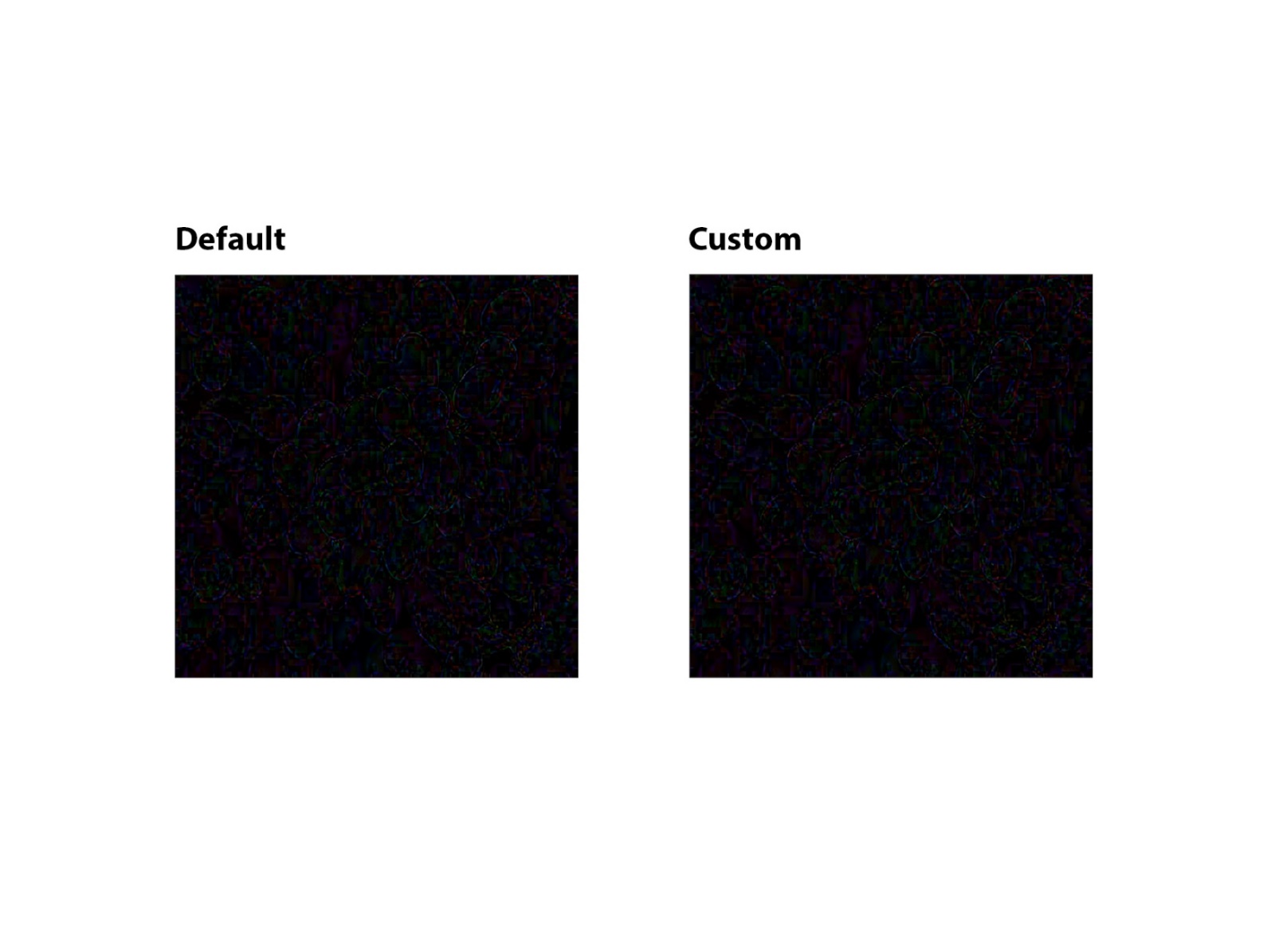
Cabe destacar, que al igual que en las imágenes anteriores los valores obtenidos para cada uno de los tipos de compresor son similares. Los valores de MSE son idénticos para default y custom, mientras que los valores de la relación de compresión se mantienes semejantes, un poco superiores para el compresor custom (esta diferencia se nota sobre todo en el factor de calidad 1).

En las imágenes obtenidas, figura 29, se puede observar como el cambio entre ellas no es demasiado grande, incluso obteniendo una imagen bastante semejante a la original con valores superiores a 100.



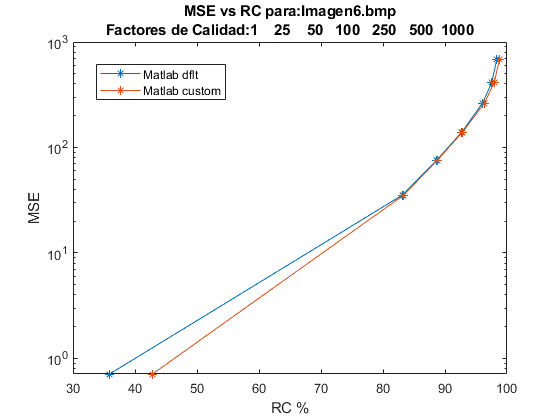
*Figura 29 - Evolución de la compresión Imagen 5*

En cuanto a las imágenes de diferencia, demuestran que el error cuadrático medio proviene de los contornos de las uvas, volviendo a confirmar la teoría de que la existencia de un color dominante en la imagen ayuda a la compresión a eliminar redundancia y a aprovechar la entropía de la imagen.



*Figura 30 - Imagen diferencia entre Imagen original y comprimida con FC 1000*

## 4.6 Imagen 6



*Figura 31 – Gráfica Imagen 6*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| FC | MSE | DEFAULT | MSE | CUSTOM | RC | DEFAULT | RC | CUSTOM |
| 1 | 0.7119 | 0.7119 | 35.9031 | 42.8234 |
| 25 | 35.3456 | 35.3456 | 83.1041 | 83.3303 |
| 50 | 75.6001 | 75.6001 | 88.6372 | 88.7547 |
| 100 | 138.7864 | 138.7864 | 92.6629 | 92.7883 |
| 250 | 263.8015 | 263.8015 | 96.0627 | 96.3578 |
| 500 | 409.4427 | 409.4427 | 97.5164 | 97.9139 |
| 1000 | 687.5086 | 687.5086 | 98.3653 | 98.8039 |

Esta imagen obtiene resultados similares a los de la imagen 2, ya que esta tiene mucha vegetación por la parte inferior, lo que añade ruido a la imagen y que sea difícil de comprimir. Pero lo interesante es cómo se comporta el texto que aparece en ella.

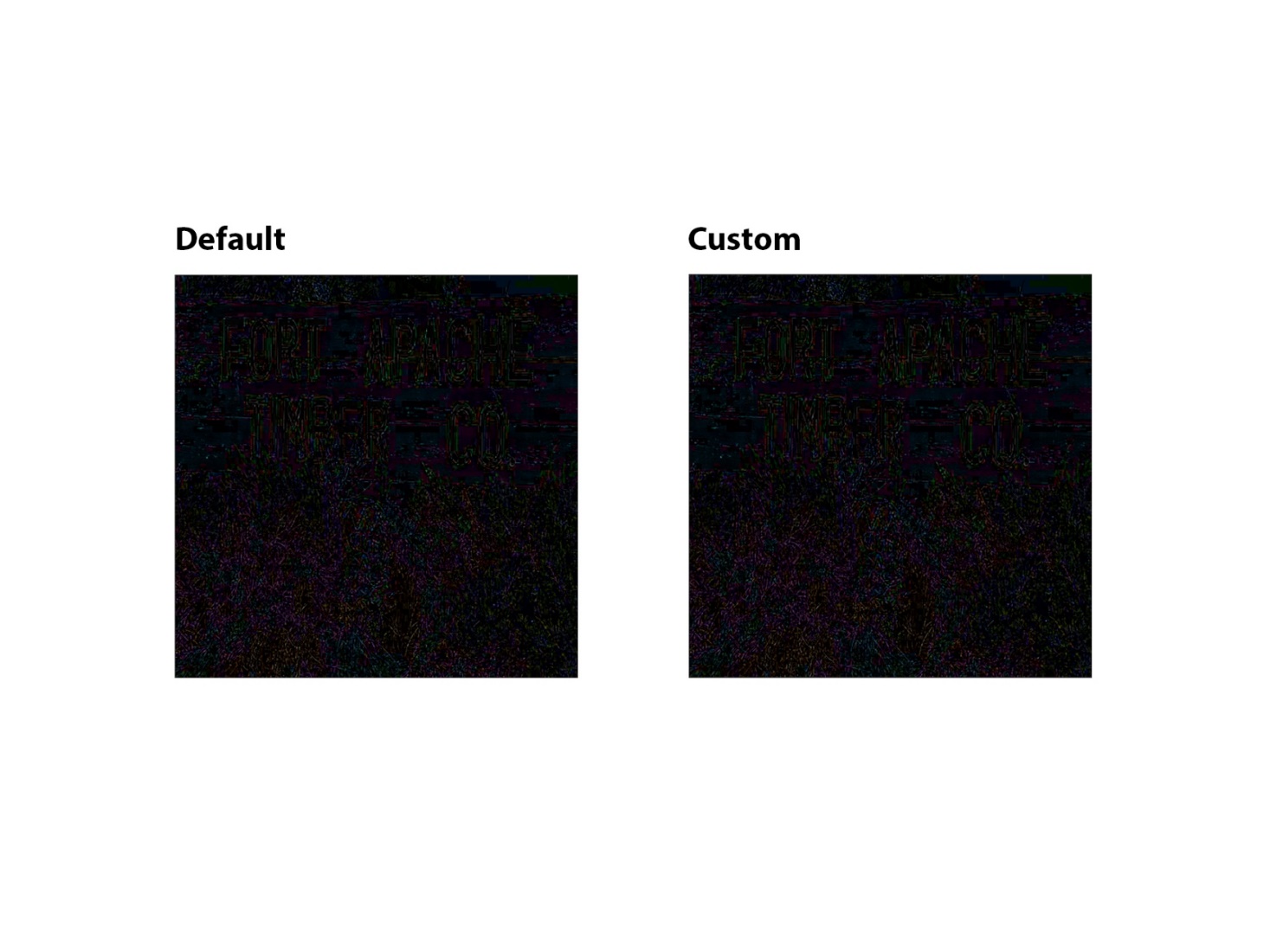
Se puede ver como los resultados obtenidos con los factores de calidad menores de 100 son bastante buenos, a pesar de su baja relación de compresión. Y a partir de 100, el error cuadrático medio crece de forma exponencial sin mejorar demasiado en la relación de compresión.

Observando las imágenes obtenidas, se puede ver que la evolución de la vegetación tanto en color como en definición de los detalles. Pero en cuanto al texto, se mantiene, pero se nota el cambio de color, al igual que en la imagen 1.



*Figura 32 - Evolución de la compresión Imagen 6*

Esta idea se reafirma en las imágenes de diferencia entre la imagen original y la imagen comprimida con el factor de calidad de 1000. Se aprecia como los bordes de las letras generan error, pero el compresor consigue mantener la forma. Mientras que en la parte de vegetación aparece mucha granulación, que se traduce a mucho error.



*Figura 33 - Imagen diferencia entre Imagen original y comprimida con FC 1000*

En conclusión, una imagen con texto consigue comprimirse con buen resultado si además este mantiene un color estable, como en la imagen. En cuanto al resto de la imagen, se repite la misma idea que en las anteriores. Cuanto más detalle exista mayor error se genera.

# 5. Otro Software. Photoshop

En este apartado se van a comparar los resultados obtenidos con los compresores implementados en Matlab con el compresor implementado en Photoshop, con el objetivo de ver que tan eficaz es el nuestro.

Para esto, primero se van a convertir los valores utilizados para el estudio de nuestros compresores (factor de calidad) en las medidas usadas en Photoshop. Recordamos que los valores de factor de calidad usados eran los siguientes:

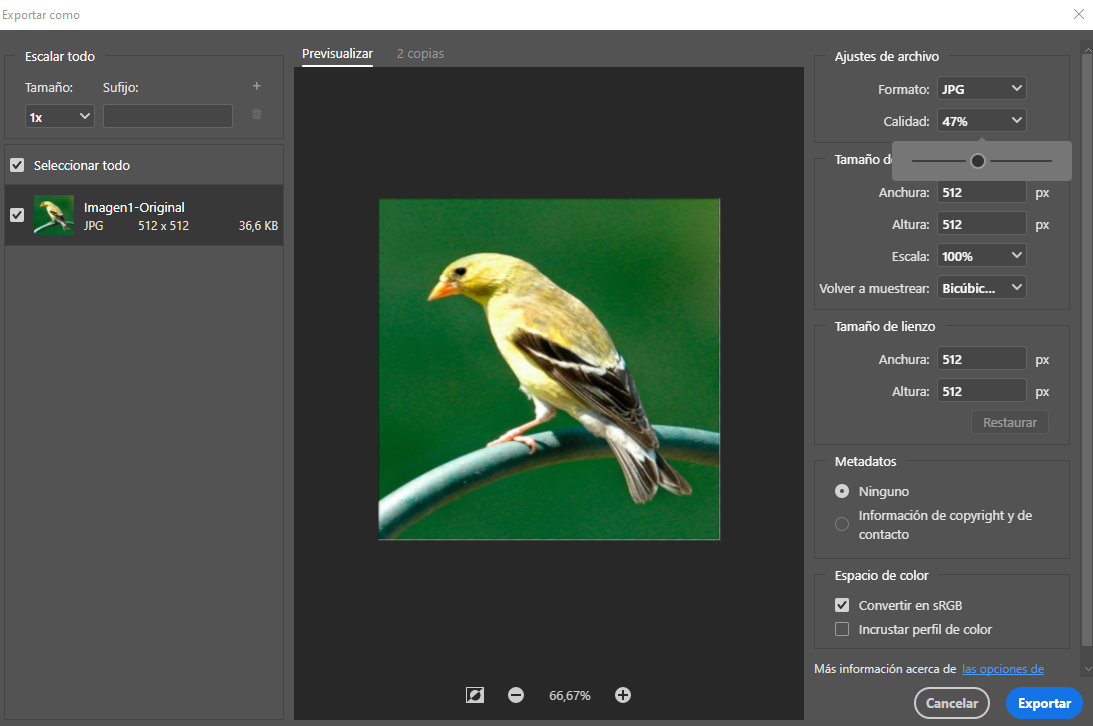
[1, 25, 50, 100, 250, 500, 1000]

Y los valores de compresión que se utilizan en Photoshop van por porcentaje entre 1 a 100%, por ello los utilizados serán:

[1, 3, 5, 10, 25, 50, 100]

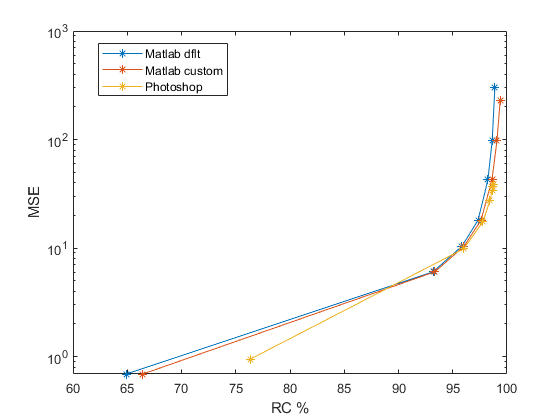
Hay que tener en cuenta que los valores de compresión de Photoshop funcionan al revés que en nuestro proyecto. El factor de calidad 1 genera la imagen con la mayor relación de compresión, y el factor de calidad 100 el menor.

Para generar dichas imágenes, se abre cada una de ellas y con la herramienta exportar (figura 34) se guardan en formato JPEG, realizando un proceso similar al que desarrollamos en Matlab. Y posteriormente, mediante una función creada en Matlab, comparamos el JPEG obtenido en Photoshop con las imágenes comprimidas que obteníamos con los compresores, default y custom, implementados en Matlab.



*Figura 34 – Herramienta exporta imagen en Photoshop*

## 5.1 Imagen 1



*Figura 35 - Gráfica Imagen 1 en Photoshop*

|  |  |  |
| --- | --- | --- |
|  | **MSE** | **RC** |
| **1** | 0.9456 | 76.33 |
| **3** | 9.817 | 95.99 |
| **5** | 17.46 | 97.78 |
| **10** | 27.18 | 98.42 |
| **25** | 34.24 | 98.64 |
| **50** | 36.68 | 98.7 |
| **100** | 38.17 | 98.76 |

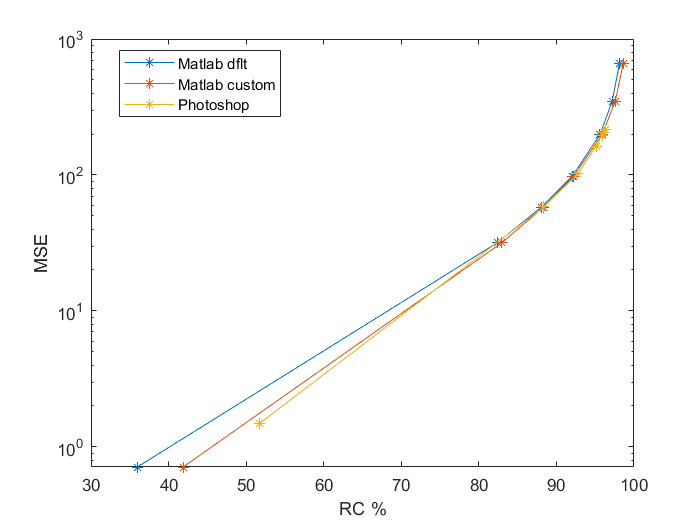
Se puede apreciar en la gráfica que los valores obtenidos mediante la compresión de Photoshop son mejores. En cambio, para valores de factor de calidad 1 (para nuestro compresor) se obtiene un mejor MSE, pero con una relación de compresión menor.

Para compararlo de una forma más sencilla, se va a comparar los resultados de aplicar el valor por defecto de compresión, 100. En Photoshop el equivalente es 10.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FC** | **MSE | Default** | **MSE | Custom** | **RC | Default** | **RC | Custom** |
| **100** | 18.116 | 18.116 | 97.3429 | 97.6133 |

Para la compresión de Matlab se obtiene un MSE de 18, mientras que para Photoshop se obtiene un 27. En cuanto a relación de calidad se obtiene 97.5 y 98, respectivamente. Esto se traduce a que los resultados para la imagen 1 son semejante entre los dos programas, sin una diferencia notoria. Pero para valores superiores a 100 / 10 es mejor Photoshop.

## 5.2 Imagen 2



*Figura 36 - Gráfica Imagen 2 en Photoshop*

|  |  |  |
| --- | --- | --- |
|  | **MSE** | **RC** |
| **1** | 1.486 | 51.71 |
| **3** | 58.24 | 88.38 |
| **5** | 101.8 | 92.74 |
| **10** | 162.6 | 95.11 |
| **25** | 195.6 | 95.9 |
| **50** | 205.1 | 96.11 |
| **100** | 214.8 | 96.3 |

Los resultados obtenidos para la imagen en los compresores de Matlab para el valor de factor de calidad 1 es mejor que para el primer valor de Photoshop, pero con una relación de compresión menor.

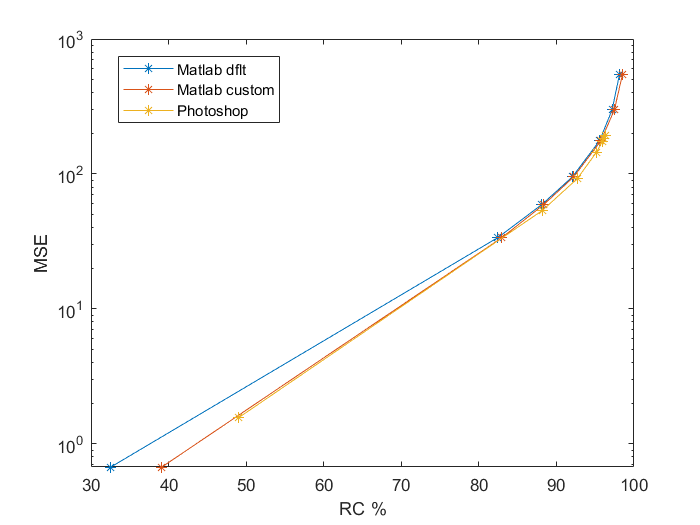
Para el resto de los datos se obtienen datos semejantes en ambos programas. Se puede comprobar de una forma más específica comparando los valores obtenidos mediante la compresión por defecto: factor de calidad 100 y calidad de compresión 10.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FC** | **MSE | Default** | **MSE | Custom** | **RC | Default** | **RC | Custom** |
| **100** | 97.6127 | 97.6127 | 92.0297 | 92.1962 |

Los valores en ambos son semejantes, siendo mejor Photoshop en relación de compresión, pero siendo Matlab más eficiente en error cuadrático medio.

En conclusión, estos datos nos dan a conocer que los procesos de compresión de ambos programas, tanto Photoshop como el desarrollado en Matlab, son semejantes.

## 5.3 Imagen 3

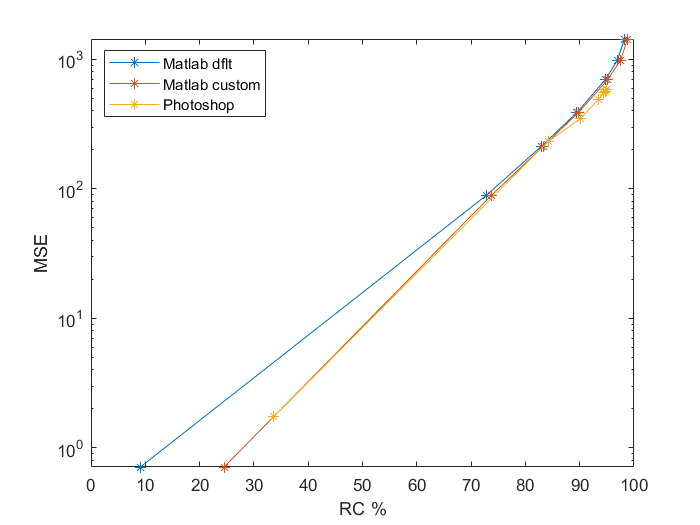


*Figura 37 - Gráfica Imagen 3 en Photoshop*

|  |  |  |
| --- | --- | --- |
|  | **MSE** | **RC** |
| **1** | 1.56 | 49.01 |
| **3** | 53.7 | 88.2 |
| **5** | 92.27 | 92.69 |
| **10** | 144.4 | 95.14 |
| **25** | 173.8 | 95.92 |
| **50** | 182.7 | 96.13 |
| **100** | 191.3 | 96.32 |

Los resultados obtenidos en estas imágenes son semejantes en ambos programas como se puede observar en la gráfica. Como en la última imagen, se puede comprobar que se repite los mejores resultados para Matlab en el primer factor de calidad, y posteriormente se igualan. Consiguiendo un mejor resultado de relación de compresión / error cuadrático medio en valores de factor de calidad superiores.

## 5.4 Imagen 4



*Figura 38 - Gráfica Imagen 4 en Photoshop*

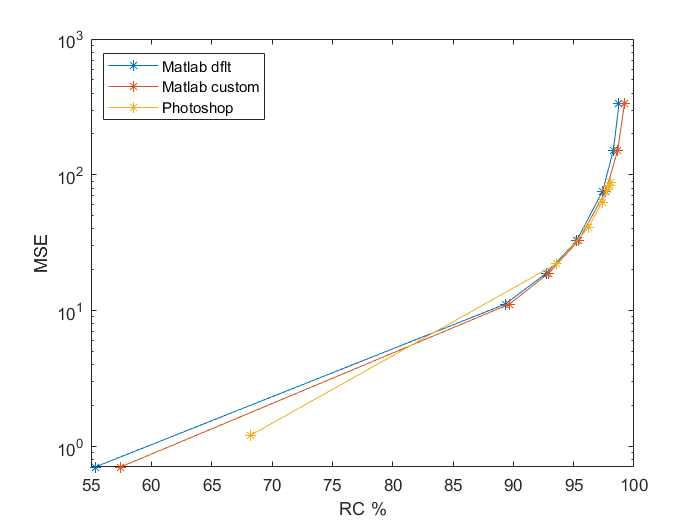
|  |  |  |
| --- | --- | --- |
|  | **MSE** | **RC** |
| **1** | 1.742 | 33.66 |
| **3** | 232.1 | 84.17 |
| **5** | 348.4 | 90.11 |
| **10** | 489.8 | 93.41 |
| **25** | 551 | 94.41 |
| **50** | 566.9 | 94.61 |
| **100** | 582.7 | 94.9 |

Tras la ejecución de la compresión de Photoshop se puede observar que esta imagen también da problemas. Al tener muchos detalles, el error cuadrático medio aumenta mucho, y también se ve reflejado en este programa. Aunque el resultado sea semejante al de Matlab, Photoshop lo mejora un poco.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FC** | **MSE | Default** | **MSE | Custom** | **RC | Default** | **RC | Custom** |
| **250** | 697.9605 | 697.9605 | 94.7237 | 95.1610 |

El resultado que se obtiene con Photoshop en la calidad de compresión mínima es semejante al que se obtiene en Matlab con un factor de calidad del 250. Concluyendo en que el primero usa un método de compresión un poco más eficiente.

## 5.5 Imagen 5



*Figura 39 - Gráfica Imagen 5 en Photoshop*

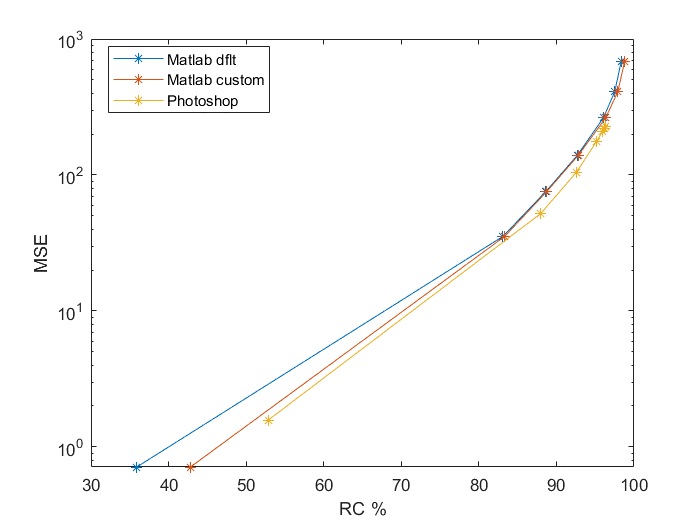
|  |  |  |
| --- | --- | --- |
|  | **MSE** | **RC** |
| **1** | 1.211 | 68.25 |
| **3** | 22.21 | 93.6 |
| **5** | 40.75 | 96.21 |
| **10** | 62.95 | 97.36 |
| **25** | 77.35 | 97.8 |
| **50** | 82.87 | 97.93 |
| **100** | 88.2 | 98.84 |

Los resultados obtenidos en estas imágenes son semejantes en ambos programas como se puede observar en la gráfica. Al igual que todas las imágenes anteriores, se repiten los mejores resultados para Matlab en el primer factor de calidad, y posteriormente se igualan. Consiguiendo un mejor resultado las compresiones de Photoshop en relación de compresión / error cuadrático medio en valores de factor de calidad superiores.

Esto se puede ver más claramente comparando los últimos datos obtenidos de cada programa en los últimos factores de calidad. El más similar en Matlab al último de Photoshop es el factor de calidad 250.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **FC** | **MSE | Default** | **MSE | Custom** | **RC | Default** | **RC | Custom** |
| **250** | 76.0380 | 76.0380 | 97.4025 | 97.6717 |

## 5.6 Imagen 6



*Figura 40 - Gráfica Imagen 6 en Photoshop*

|  |  |  |
| --- | --- | --- |
|  | **MSE** | **RC** |
| **1** | 1.574 | 52.84 |
| **3** | 52.1 | 87.98 |
| **5** | 104.6 | 92.58 |
| **10** | 176.3 | 95.17 |
| **25** | 209.5 | 95.96 |
| **50** | 218.4 | 96.17 |
| **100** | 226.7 | 96.35 |

En esta última imagen se repiten los resultados, los mejores resultados para Matlab son con el primer factor de calidad, y posteriormente se igualan. Se consigue un mejor resultado en las compresiones de Photoshop cuando los valores de factor de calidad son superiores.

De la misma manera, el último valor obtenido para Photoshop se asemeja con el valor obtenido en Matlab por una compresión con el factor de calidad de 250.

# 6. Conclusiones

Tras experimentar con los compresores que se han desarrollado en la práctica se llegan a las siguientes conclusiones.

Gracias a la primera y la quinta imagen, se puede concluir en que los colores lisos o dominantes en una imagen ayudan al compresor a obtener unos mejores resultados, ya que consiguen aprovecharse de la redundancia existente en ellas.

Por el contrario, cuando existe muchas diferencias de colores en las imágenes, el compresor obtiene unos resultados con gran diferencia entre la imagen original y la comprimida. Esto se debe a que no se puede beneficiar de la redundancia, antes comentada. Del mismo modo, se consigue demostrar que la simetría de una imagen tampoco logra ayudar al compresor con la tercera y cuarta imagen.

Además, el texto en imágenes funciona de una forma similar. Si este texto esta representado mediante un color liso, como en el caso de la sexta imagen que se representa con color blanco, se consigue uno valores de compresión bastante buenos. Pero si este estuviese representado con diferentes colores / detalles, complicaría su compresión alcanzando valores de error superiores.

Todos estos resultados obtenidos de los compresores implementados en Matlab se comparan con la compresión que realiza Photoshop para estudiar su veracidad. Logrando mejores resultados en este segundo en orden general, pero siendo similares a los obtenidos en los compresores de Matlab.

En definitiva, la práctica nos permite entender el proceso que se toma para comprimir una imagen siguiendo la metodología de JPEG. Tanto de forma teórica, con la explicación de términos como entropía o DCT, y de forma práctica, comprimiendo unas imágenes seleccionadas para poder razonar ideas planteadas con los resultados que se han obtenido.

# 7. Bibliografía

[1] G. Strang, "*The Discrete Cosine Transform*", SIAM Review, Vol. 41, N.1, pp. 135’147, 1999. [En línea]. Disponible en: <https://epubs.siam.org/doi/pdf/10.1137/S0036144598336745>

[2] Computer Science. *YCbCr and RGB Colour.* (23 de noviembre de 2019). Acceso: 26 de diciembre de 2020. [Video en línea]. Disponible en: <https://youtu.be/3dET-EoIMM8>

[3] 3Blue1Brown. *But what is the Fourier Transform? A visual introduction.* (26 de junio de 2018). Acceso: 27 de diciembre de 2020. [Video en línea]. Disponible en: <https://youtu.be/spUNpyF58BY>

[4] Barry Van Veen. *The Two-Dimensional Discrete Cosine Transform.* (22 de junio de 2020). Acceso: 27 de diciembre de 2020. [Video en línea]. Disponible en: <https://youtu.be/DS8N8cFVd-E>

[5 Leo Isikdogan. *How Image Compression Works.* (26 de junio de 2018). Acceso: 3 de enero de 2021. [Video en línea]. Disponible en: <https://youtu.be/Ba89cI9eIg8>

[6] Computerphile. *Colourspaces (JPEG Pt0) – Computerphile*. (10 de abril de 2015). Acceso: 23 de diciembre de 2020. [Video en línea]. Disponible en: <https://youtu.be/LFXN9PiOGtY>

[7] Computerphile. *JPEG ‘files’ & Colour (JPEG Pt1) – Computerphile*. (21 de abril de 2015). Acceso: 3 de enero de 2021. [Video en línea]. Disponible en: <https://youtu.be/n_uNPbdenRs>

[8] Computerphile. *JPEG DCT, Discrete Cosine Transform (JPEG Pt2) – Computerphile*. (22 de mayo de 2015). Acceso: 3 de enero de 2021. [Video en línea]. Disponible en: <https://youtu.be/Q2aEzeMDHMA>

[9] Computerphile. *The Problem with JPEG – Computerphile*. (9 de junio de 2015). Acceso: 6 de enero de 2021. [Video en línea]. Disponible en: <https://youtu.be/yBX8GFqt6GA>

1. https://data.mendeley.com/datasets/sp4g8h7v8k/1 [↑](#footnote-ref-1)