Análisis de Compresión de Imágenes con JPEG

COMPRESIÓN MULTIMEDIA

Arturo Lorenzo Hernández | Curso 2020-2021 | 15 Enero

INDICE

Contenido

1.INTRODUCCIÓN

Este trabajo consiste en un análisis experimental del método de compresión de imágenes JPEG. Empezaremos con esta breve introducción acerca del método en cuestión y porqué es tan famoso y efectivo, después hablaremos más detalladamente sobre cómo funciona la metodología del algoritmo de compresión explicando el funcionamiento de JPEG al completo y marcando las decisiones del alumno en la práctica. Seguido de esta explicación se mostrará el análisis experimental que se ha llevado a cabo con siete imágenes en formato BMP ya que son archivos sin pérdidas. En este análisis nos centraremos en los factores de calidad utilizados para comprimir las imágenes, además de fijarnos en la relación de compresión de la imagen para cada factor de calidad y el error cuadrático medio entre la imagen comprimida y la original. Estas 3 variables serán el centro de nuestro estudio. Con lo explicado en la metodología se intentará predecir a rasgos generales el resultado de la compresión en cada imagen seleccionada y comentaremos posibles errores en las hipótesis de partida, en base a imágenes diferenciales creadas con el programa Photoshop, esto se explica en el apartado 3.

Para concluir el trabajo se hará una comparación de los compresores y descompresores implementados por el alumno en comparación con los que tiene Photoshop para exportar imágenes a formato JPEG.

Este método de compresión nace del hecho de que las imágenes naturales tienen mucha redundancia espacial entre muestras próximas, es decir, los valores que toman los píxeles de una imagen tienden a ser muy parecidos entre píxeles adyacentes.

Esto se suma al hecho de que el ojo humano es capaz de percibir mejor el brillo de los objetos que el color de los mismos, ante esta diferencia de percepción JPEG aprovecha para comprimir los espacios de color de una imagen a niveles muy altos como veremos más adelante.

El modo básico de JPEG utiliza una codificación con pérdidas que es el que estudiaremos en este trabajo, por eso se obtienen relaciones de compresión tan altas, es decir, la imagen comprimida tendrá pérdida de información con respecto a la imagen original que ya no se podrá recuperar, es más, si una imagen comprimida por JPEG se vuelve a comprimir seguirá perdiendo información, por eso para ámbitos en los que la calidad de la imagen es primordial como lo es en el campo de la medicina el uso de JPEG con pérdidas no es recomendable.

A continuación, pasaremos a explicar el funcionamiento de JPEG más en detalle.

2.METODOLOGÍA

En este apartado vamos a explicar el funcionamiento de la compresión y descompresión de imágenes con JPEG, en las prácticas de la asignatura se nos pedía desarrollar un compresor y un descompresor para cada tipo de compresión Huffman utilizada, la que utiliza tablas Huffman por defecto (estándar ITU T.81) y las tablas Huffman personalizadas para cada imagen en base a la frecuencia de las categorías que explicaremos más adelante.

A continuación, vamos a desglosar el funcionamiento del compresor implementado apoyándonos en el marco teórico del método de compresión.

2.1 CAMBIO DE ESPACIO DE COLOR

Lo que primero tenemos que hacer para comprimir una imagen JPEG es tener la imagen en el espacio de color adecuado, existen diversos espacios de color, nuestros compresores admiten imágenes BMP y JPEG con espacios de color indexado o truecolor (RGB). Aunque acepte estos espacios no quiere decir que trabaje con ellos, es más, los acepta, pero para convertirlos en el espacio de color Y CB CR que representa una imagen por medio de su luminancia (Y), la escala de azules que contiene (CB) y la escala de rojos (CR).

Foto dividida en espacio de color YCBCR

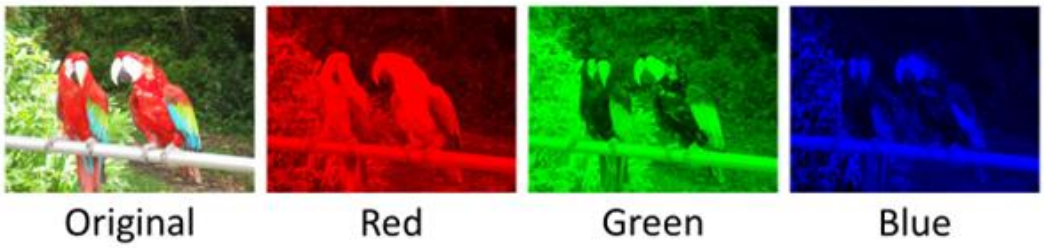


Foto dividida en espacio de color RGB

Como vemos en las imágenes anteriores una misma imagen se puede representar en los espacios de color que deseemos. El uso de Y CB CR para JPEG tiene mucho sentido ya que como se mencionó en la introducción el ojo humano distingue mucho más el

brillo que el color. YCBCR hace exactamente esto, Y es el brillo y CBCR es el color de la imagen.

En el caso de que la imagen esté representada en un espacio de color indexado lo que primero se hace es pasarlo a RGB y después a YCBCR, en el caso de que sea RGB directamente se convierte a YCBCR. Esta última conversión se consigue de manera muy sencilla con las siguientes fórmulas (1):

$$Y = 0.299R + 0.587G + 0.114B$$

$$Cb = -0.1687R - 0.3313G + 0.5B + 128$$

$$Cr = 0.5R - 0.4187G – 0.0813B + 128$$

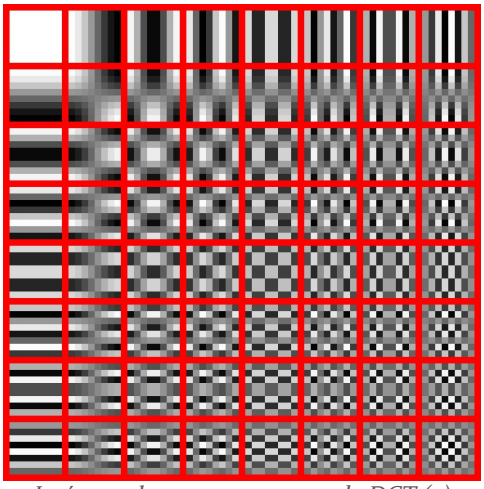
Como podemos observar en las fórmulas en el campo Cb el valor B de RGB aparece sumando al igual que el rojo en el campo Cr dándole más peso a esos colores. El rango de valores posibles en Y Cb o Cr es de [0,255] o lo que es lo mismo 1 byte.

Tras el cambio de espacio de color a YCBCR lo que hacemos es ampliar la imagen para que su tamaño sea múltiplo de 8 en caso de que no lo sea. Esto se hace para que la DCT que explicaremos a continuación pueda realizarse sin problemas.

2.2 TRANSFORMADA COSENO DISCRETA (DCT)

En este punto se divide la imagen en bloques de 8x8 píxeles, este es el motivo por el que ampliamos la imagen en el paso anterior. Los bloques son procesados de izquierda a derecha y de arriba abajo, a cada bloque de 8x8 se le aplica la DCT, que es una transformada basada en la discreta de Fourier, es una función par, compuesta por cosenos y con las propiedades de ser continua y siendo x(t) = -x(t).

Lo interesante de la DCT bidimensional es que se puede interpretar como una descomposición de una imagen en imágenes base. Como vemos a continuación.



Imágenes base que representa la DCT (3)

Por lo tanto, al aplicar la DCT a un bloque de 8x8 lo que estamos haciendo es aislar las distintas frecuencias contenidas en el bloque, siendo cada coeficiente transformado el peso de dicha frecuencia que la imagen base representa, es decir, estamos cuantificando cada coeficiente por su frecuencia. Lo que hace la DCT en los bloques de las imágenes es concentrar toda la energía en la esquina superior izquierda de la matriz de coeficientes, lo que nos confirma lo que decíamos en la introducción, que las imágenes naturales tienen mucha redundancia en el espacio, lo que se traduce en bajas frecuencias o lo que es lo mismo que los valores de los píxeles normalmente cambian de forma lenta de un punto a otro del bloque.

Por esto, al aplicar la DCT todos los coeficientes relacionados con las imágenes base de altas frecuencias tendrán pesos muy bajos y esto como veremos en el siguiente apartado nos permitirá tener una compresión de la imagen muy alta, así como los coeficientes superiores a la izquierda tendrán valores muy altos y son los que recogerán la gran mayoría de la información de la imagen a reconstruir.

En estos 2 apartados en los que hemos cambiado de espacio de color, hemos ampliada la imagen a múltiplo de 8, hemos dividido la imagen en bloques de 8x8 y les hemos aplicado la DCT a cada uno de ellos, pasando de una representación espacial a una frecuencial, no hemos perdido nada de información, todavía se podría recuperar la imagen sin ninguna pérdida.

2.3 CUANTIZACIÓN DE LOS BLOQUES

Una vez tenemos la imagen cuantificada por bloque de 8x8 lo que hacemos es eliminar toda la información que el ojo humano no es capaz de ver. Esto lo conseguimos con matrices de cuantización que tienen un tamaño 8x8 y son aquellas que se utilizan para eliminar las altas frecuencias de la imagen, el proceso es muy sencillo simplemente dividimos cada coeficiente de los bloques de 8x8 por el valor de la matriz de cuantización que esté en la misma posición que el coeficiente. (3)

$$round\left(\frac{y_{ij}}{Q_{ij}}\right)$$

Siendo Y la matriz de cuantización, Q la matriz de coeficientes y round la función que redondea al entero más próximo.

Como cada imagen JPEG está representada por 3 matrices Y CB CR, a cada matriz hay que aplicarle cuantización, lo interesante es que no le aplicamos a las 3 matrices la misma matriz de cuantización. Como el ojo humano tiene menos sensibilidad a los cambios en los colores de la imagen la matriz de cuantización para CB y CR será más restrictiva que la de Y que representa la luminancia, cuanto más restrictiva es una matriz de cuantización más altos son los valores de la misma para así con la función round convertir a cero todos aquellos coeficientes que tengan valores insignificantes. A continuación, las podemos ver y comprobarlo.

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

Luminancia

17	18	24	47	99	99	99	99
18	21	26	66	99	99	99	99
24	26	56	99	99	99	99	99
47	66	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99
99	99	99	99	99	99	99	99

Crominancia

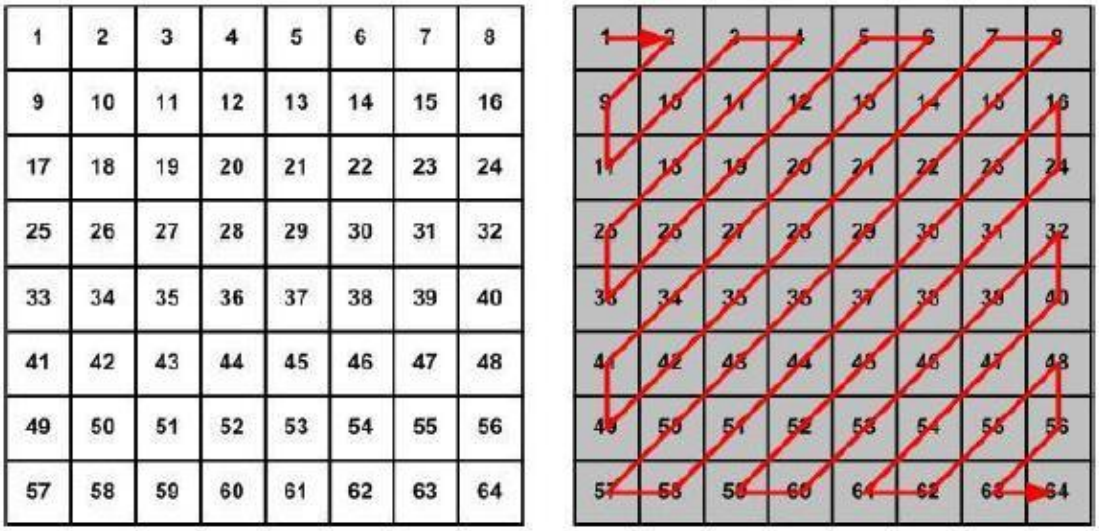
Tablas de cuantización recomendadas.

Tras este proceso de cuantización, como podemos estar comprobando si se pierde información ya que los valores de la matriz de coeficientes que al ser cuantizados tengan un valor cero o próximo a él no serán comprimidos, esto va a pasar con los coeficientes que representan imágenes base de altas frecuencias, y por lo tanto esa información, aunque sea insignificante para nuestra vista se perderá.

Y aquí es donde entran en juego los factores de calidad, JPEG nos ofrece establecer que calidad de las imágenes queremos tener con respecto a la original. Los factores de calidad lo que hacen es multiplicar las matrices de cuantización por ese factor, de esta forma cuanto más aumenta el factor de calidad más coeficientes se convertirán a cero por el hecho de que las matrices de cuantización se volverán más restrictivas (estamos multiplicando todos los valores de la matriz por una constante), esto hace que aumente la compresión ya que habrá que guardar menos coeficientes, pero la calidad empeorará debido a que habrá más perdida de información. Para nuestro estudio hemos escogido siete factores de calidad que van desde un extremo hasta el otro. Los valores son el 1, 25, 50, 100, 250, 500, 1000. Como es lógico con un factor de calidad 1 las imágenes se recuperarán muy bien con respecto a las originales, pero la compresión que obtendremos será mediocre ya que no descartaremos muchos coeficientes. Por el otro lado, con un factor de calidad 1000 tendremos una compresión altísima, pero las imágenes se recuperarán con una calidad pésima. El objetivo es encontrar un punto medio entre calidad de recuperación y una buena compresión, es por ello que el estudio se centra en el error cuadrático medio MSE, que nos dirá cuánta “calidad” se pierde con la compresión y la relación de compresión que nos mostrará el nivel de compresión de la imagen.

2.4 COMPRESIÓN DE LA IMAGEN

Una vez tenemos los coeficientes cuantizados, lo que queda es comprimir la información de la imagen que nos resulta de los pasos anteriores. Lo que primero se hace con cada bloque 8x8 es prepararlo de forma que los valores más importantes sean más accesibles, es decir, los coeficientes de la esquina superior izquierda, por ello se preparan los bloques de 8x8 para tener un acceso secuencial, pero en forma de zigzag.



Bloque 8x8 ordenado en zig-zag (5)

A continuación, vemos como se quedaría la tabla resultante a la hora de acceder a ella de forma secuencial.

1	2	9	17	10	3	4	11
18	25	33	26	19	12	5	6
13	20	27	34	41	49	42	35
28	21	14	7	8	15	22	29
36	43	50	57	58	51	44	37
30	23	16	24	31	38	45	52
59	60	53	46	39	32	40	47
54	61	62	55	48	56	63	64

Acceso secuencial al bloque en zigzag

La codificación de un bloque 8x8 de la imagen emplea codificación Run-Length, es un tipo de codificación muy sencillo que guarda secuencias de datos con el mismo valor consecutivas como un único valor más su recuento, en este caso va guardando cuantos ceros preceden a un coeficiente con valor distinto de cero. Para la codificación de un bloque 8x8 hay que distinguir 2 procesos (4):

- **Codificación de los coeficientes DC:** El primer coeficiente de la matriz se denomina el DC, es el que más información guarda ya que es el que representa frecuencia cero. Todos los coeficientes DC de una imagen se codifican por separado del resto, en una tabla que guarda las categorías y las posiciones de las diferencias entre los coeficientes DC adyacentes (Dci – Dci-1). Una categoría es el tamaño de la diferencia entre dos DCs adyacentes dada un código Hufman y las posiciones representan la amplitud de la diferencia, que nos indican la posición dentro de una categoría.
- **Codificación de los coeficientes AC:** El resto de coeficientes de un bloque 8x8 se le denominan coeficientes ACs, estos también se guardan en una tabla como los DC, para cada coeficiente AC distinto de cero se guarda una entrada en la tabla, de aquí que se use la codificación run-length. Se clasifican también en categorías y posición, la diferencia con los DC es que la categoría y posición representan una tripleta de valores: el run que es el número de ceros que le precede al coeficiente a guardar, el tamaño con el que se codifica el valor del coeficiente según un código Huffman y la amplitud que es el valor binario del AC. Además, si desde que se codifica un coeficiente todo lo que queda en la tabla son ceros, se guarda una entrada especial en la tabla EOB = [0 0] siendo categoría y posición respectivamente los valores de la tupla. De esta forma si con la cuantización hemos eliminado todas las altas frecuencias o la mayoría se tendrán que codificar pocas entradas en la tabla sobre los coeficientes AC.

Lo único que se codifica de estas tablas es la columna de categoría. La posición de cada coeficiente ira junto con su valor en binario codificada. Lo interesante también de la práctica es que estas tablas y los valores los codificamos con dos tablas de codificación distintas para ver qué método Huffman tiene mejor rendimiento, utilizamos el método Huffman por defecto que tiene unas tablas de codificación fijas para todas las imágenes y Huffman personalizado que adapta las tablas de codificación para cada imagen.

2.4.1 HUFFMAN POR DEFECTO

En la codificación Huffman por defecto utilizamos unas tablas BITS y HUFFVAL del método Huffman fijas para cada imagen, estas tablas están basadas en ITU T.81, se utilizan cuatro tablas distintas:

-Para codificar los DC de la luminancia Y

-Para codificar los AC de la luminancia Y

-Para codificar los DC de la crominancia CbCr

-Para codificar los AC de la crominancia CbCr

Estas tablas se han conseguido a través de análisis experimental por parte de investigadores de JPEG y no han sido obtenidas por el alumno, al igual que las tablas de cuantización. Estas tablas asignan códigos cortos a categorías bajas ya que es ahí donde se verán representados la mayoría de coeficientes de una imagen. El motivo de utilizar unas tablas por defecto es que el coste computacional para comprimir una imagen es menor que utilizando unas tablas a medida. Las imágenes que mejor se comprimen con este método son aquellas en las que los coeficientes tengan una amplitud pequeña para que encajen en categorías pequeñas y los códigos Huffman sean cortos, esto se traduce en un fichero comprimido más ligero.

2.4.2 HUFFMAN PERSONALIZADO

La codificación introducida por el alumno es esta, aquí no se establece ninguna tabla por defecto para codificar los valores correspondientes si no que nos basamos en las frecuencias de las categorías que obtuvimos en las tablas de codificación de los ACs y DCs para generar las tablas de codificación y esto nos ayuda a mejorar bastante la compresión con respecto a Huffman por defecto, ya que estamos eligiendo los códigos Huffman en base a la imagen y no a la generalidad, aunque no forma parte de este estudio por lógica podemos intuir que crear las tablas Huffman a medida tiene un coste computacional mayor, ya que el procesamiento de las frecuencias no se tiene que hacer en Huffman por defecto.

2.5 FICHERO HUD Y CONCLUSIÓN COMPRESIÓN

Una vez ya hemos realizado la compresión de la imagen según los pasos descritos anteriormente guardamos los 3 códigos Huffman referentes a cada una de las dimensiones del espacio de color YCBCR y toda la información relevante de las matrices de codificación para Huffman personalizada, así como el factor de calidad establecido para la compresión de la imagen. **Importante decir que este proceso de compresión no tiene pérdidas de información sobre la imagen original, el único paso con pérdidas es la cuantización de los coeficientes.**

Como conclusión de la compresión podemos afirmar que las imágenes mejor codificadas serán las que menos alternancia de colores tengan y menos cambios bruscos presenten en la imagen ya que cuanto más se mantenga un color en una zona de la imagen mejor será codificada esa zona (más coeficientes se harán cero en esa zona) , lo que esperamos de la compresión es que los contornos de los objetos que son la zona de cambio de color se codifiquen peor para factores de calidad altos al igual que las diferentes tonalidades de un color dentro de una misma zona esto último se espera que se note menos ya que no es un cambio tan brusco de color pero para factores de calidad altos si se notará bastante. Un tipo de imagen que a JPEG se espera que le cueste bastante comprimir con un nivel alto es una imagen que tenga texto pequeño ya que alterna bastante el color entre el texto y el fondo, o alguna imagen que tenga una alternancia de colores muy alta. Por ello, vamos a escoger fotos que vayan de un extremo a otro pasando por el centro y ver si esta hipótesis de partida se corresponde con la realidad. Importante recalcar que los distintos métodos de compresión tanto Huffman por defecto y Huffman personalizado tendrán el mismo MSE para una imagen ya que la información que se guarda sobre la imagen es la misma

la diferencia es que en Huffman personalizado se codifica de forma más eficiente, pero la información de la imagen codificada es la misma para ambos métodos.

A continuación, vamos a resumir como sería el proceso inverso, la descompresión de una imagen.

2.6 DESCOMPRESIÓN

Como vemos en la siguiente imagen, en esto se puede resumir el compresor que hemos estado explicando en los apartados anteriores.

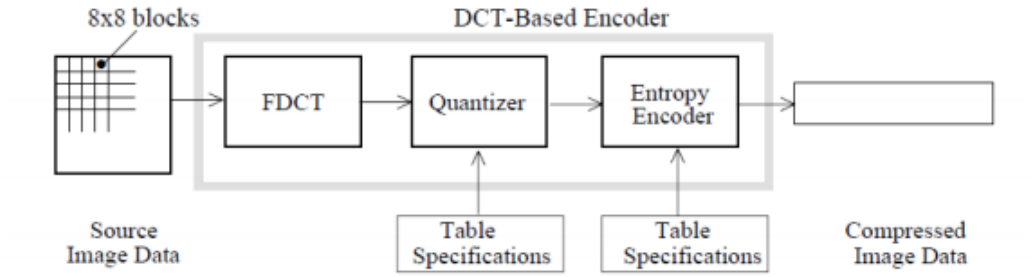


Diagrama de bloque del codificador Huffman por defecto (2)

La descompresión es básicamente el proceso inverso, no se va a entrar en detalla del proceso de descompresión entero, lo vamos a explicar apoyándonos en la siguiente imagen.

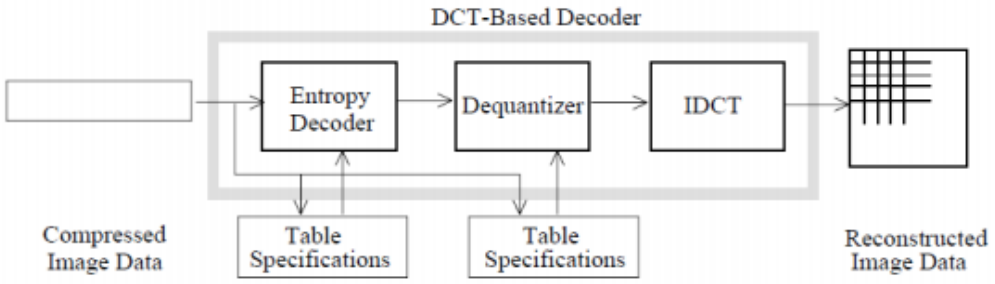


Diagrama de bloque del descodificador Huffman por defecto. (2)

Con el bitstream que tenemos en el fichero hud vamos a reconstruir las matrices de luminancia y crominancia con las mismas tablas Huffman con las que las codificamos, tras la recuperación de las matrices de coeficientes cuantizadas lo que haremos será multiplicarlas por las matrices de cuantización valor a valor de la misma forma que hicimos en la compresión. Como ya fueron redondeados los valores en la compresión todos los ceros que fueron transformados no volverán a tener sus resultados originales, habrá una pérdida de información, tras descuantizar los coeficientes distintos de cero se les aplicará a las matrices YCBCR la transformada inversa del coseno y de esta forma

tendremos de nuevo las matrices con los valores originales y tendremos otra vez la representación espacial de la imagen, es decir, tendremos la imagen lista para visualizarla.

2.7 SOFTWARE DESARROLLADO

Para realizar el análisis experimental de una manera más sencilla y rápida lo que se ha hecho es un script en Matlab que generará una gráfica para cada imagen representando en el eje X la relación de compresión y en el eje Y el error cuadrático medio de la imagen comprimida tanto para el método de compresión Huffman por defecto como Huffman personalizado, también se encargará de crear una tabla con los datos y guardarla en un archivo .xlsx para Excel. El script es muy sencillo simplemente le das un nombre de una imagen por parámetro y te ejecuta los compresores con los siete factores de calidad especificados anteriormente y los descompresores, guarda los resultados y después genera la gráfica y la tabla.

También he implementado un script que genera las tablas correspondientes de RC y MSE para comparar las imágenes originales con las imágenes comprimidas por Photoshop para cada porcentaje que he elegido, esto se verá en el apartado 4 de la práctica.

3. ANÁLISIS EXPERIMENTAL

A continuación, se va a explicar las imágenes que he escogido para hacer las hipótesis correspondientes de cómo va a resultar la aplicación de los compresores y descompresores sobre las imágenes, todas las imágenes han sido cogidas de la batería de imágenes que se nos daban en los recursos del Aula Virtual. Después de cada imagen comentaremos lo más relevante de cada una basándonos en la **imagen diferencial** entre la imagen original y la imagen comprimida con factor de calidad 1000, esto lo hacemos para ver las diferencias de luminancia y crominancia, en rojo se verán las diferencias de Cr y en azul las de la dimensión Cb, el hecho de que lo hagamos con la imagen comprimida de mayor factor de calidad es porque considero que es donde más se aprecian los fallos de JPEG. *Esta imagen diferencial la he conseguido de una forma muy sencilla con el programa Photoshop que lo utilizaremos para hacer la comparación de imágenes en el próximo apartado, simplemente he puesto en dos capas distintas las imágenes y le he puesto la opción de mostrar diferencia.*

Generalidades: Lo que creo que va a ocurrir en general para todas las imágenes es que para factores de calidad 1,25,50,100 difícilmente se note a simple vista la diferencia entre la imagen original y la comprimida. A lo mejor en 100 para alguna imagen complicada (mucha alternancia de colores) como la Img07 se nota algo, pero por lo general para factores de calidad bajos-medios JPEG respetará bien la imagen original. Lo que también espero es que para factores de calidad 250,500,1000 la imagen ya empiece a deformarse, perder color y ser bastante diferente a la original, por la gran pérdida de información que vamos a tener con JPEG eliminando coeficientes de los bloques 8x8.

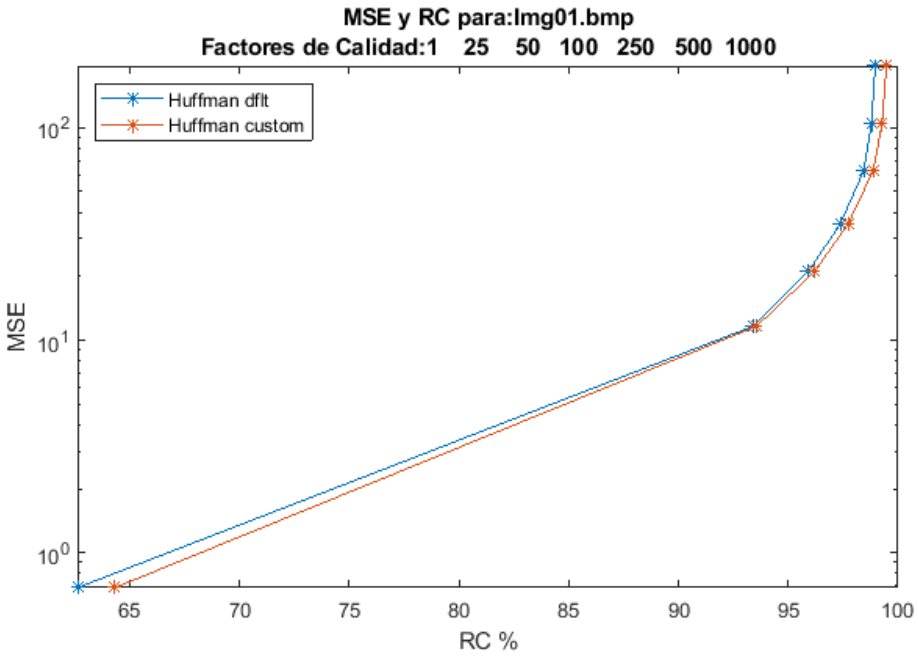
3.1 IMAGEN 01.BMP

Esta imagen podemos dividirla en 5 zonas diferentes, la pared que mantiene el color blanco con distintas tonalidades, el marco de piedra de la ventana, la zona interior de madera del marco, la bicicleta y el suelo. Lo que creo que va a pasar con la compresión de esta imagen es que la pared blanca se va a respetar bastante bien incluso para factores de calidad altos ya que es zona que tiene muy poca alternancia de colores, simplemente se puede apreciar distintos tonos de blanco, que también serán interesantes. Los bordes de la ventana estimo que no se comprimirán tan bien ya que es un cambio bastante brusco de color. La bicicleta puede ser que se comprima bien en las zonas de las ruedas ya que tienen colores "parecidos", pero donde más perderá información será en el manillar que está pegado a la pared, en la zona del sillín que está pegado al marco de la ventana y en el cuerpo ya que tiene un color blanco. Las líneas del suelo también serán interesantes ya que puede ser que vayan desapareciendo conforme aumente los factores de calidad, el suelo en general confío en que se comprima bien por mantener el color gris en todo el tramo de la imagen.

Imagen01.bmp Original

Factores	MSE_Dflt	MSE_Custom	RC_Dflt	RC_Custom
1	0,68	0,68	62,62	64,32
25	11,65	11,65	93,42	93,57
50	21,08	21,08	95,92	96,20
100	35,24	35,24	97,37	97,73
250	63,06	63,06	98,45	98,87
500	104,32	104,32	98,79	99,27
1000	195,22	195,22	98,97	99,48

Tabla para Img01.bmp



Gráfica para Img01.bmp

Imagen diferencial entre original y comprimida CaliQ 1000

Como vemos en las gráficas se confirma lo que comentábamos durante la explicación de la metodología para factor de calidad 1 la imagen comprimida no varía casi nada con respecto a la original y con factor de calidad 25 ya notamos un cambio considerable en el error y en la compresión, tienen una correlación positiva las 3 variables. También vemos que con factor de calidad 100 ya empieza a ver diferencias incluso a simple vista en el marco de la ventana, y conforme vamos aumentando el factor de calidad el error

aumenta considerablemente, llegando a perder casi la totalidad de la información del color para FC = 1000. Y cabe destacar y no lo mencionaremos más adelante que como decíamos el método Huffman personalizado comprime con un % mayor que el por defecto, aunque la diferencia es mínima, lo que quiere decir que las tablas por defecto son bastante buenas para las imágenes naturales. Centrándonos en la imagen diferencial vemos que las zonas que más sufren es la ventana, las líneas del suelo y la bicicleta, ésta última, sobre todo. Algo que no esperaba es que la parte de las ruedas sufriera pérdida de información, pero vemos que como decía al principio la parte que más pierde información es la del manillar, el cuerpo y el sillín.

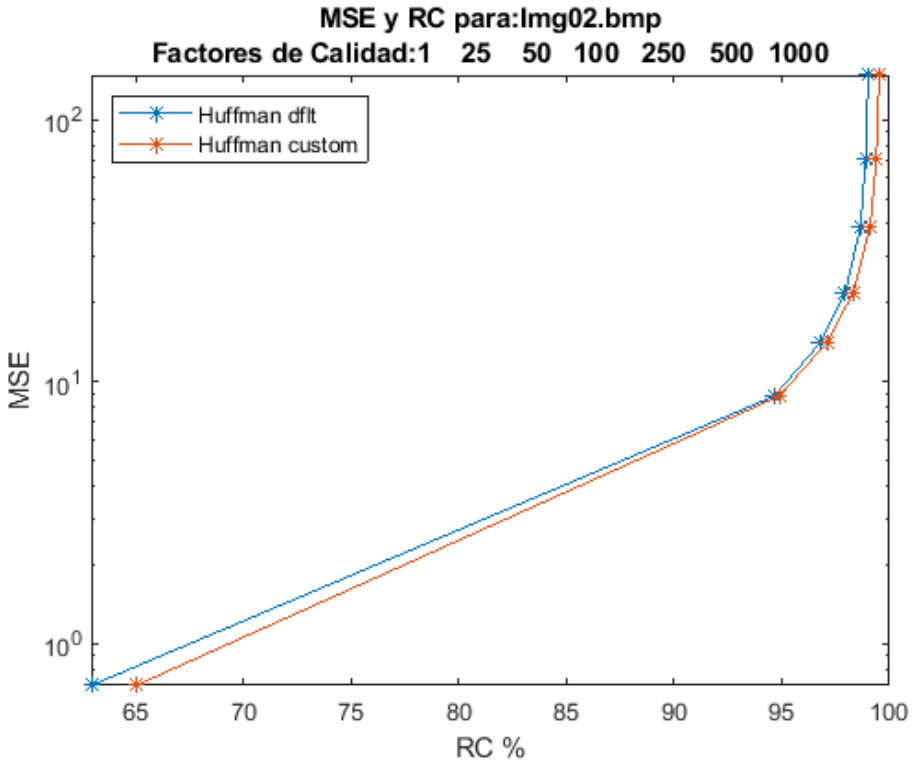
3.2 IMAGEN 02.BMP

Esta imagen es bastante interesante debido a que lo que espero no pueda comprimir bien del todo JPEG es al hombre que está utilizando la canoa en medio del río, el puente y el color del cielo que supongo que se confundirá con el color del río ya que son muy parecidos y los árboles del fondo desaparecerán. Lo que no sé muy bien es que pasará con los edificios, pero lo comentaremos en la imagen diferencial. Lo que también se va a remarcar son las ondas del río y espero se comprima bastante bien el color del agua ya que tiene una tonalidad progresiva. Creo que es una buena imagen para tener buen rendimiento con JPEG, ya que la mayoría de la misma es el cielo y el agua y son colores más o menos constantes.

Imagen02.bmp Original

Factores	MSE_Dflt	MSE_Custom	RC_Dflt	RC_Custom
1	0,69	0,69	62,93	65,03
25	8,81	8,81	94,66	94,88
50	14,16	14,16	96,81	97,11
100	21,97	21,97	97,95	98,32
250	39,21	39,21	98,66	99,11
500	70,43	70,43	98,89	99,38
1000	147,92	147,92	99,01	99,52

Tabla para Img02.bmp



Gráfica para Img02.bmp

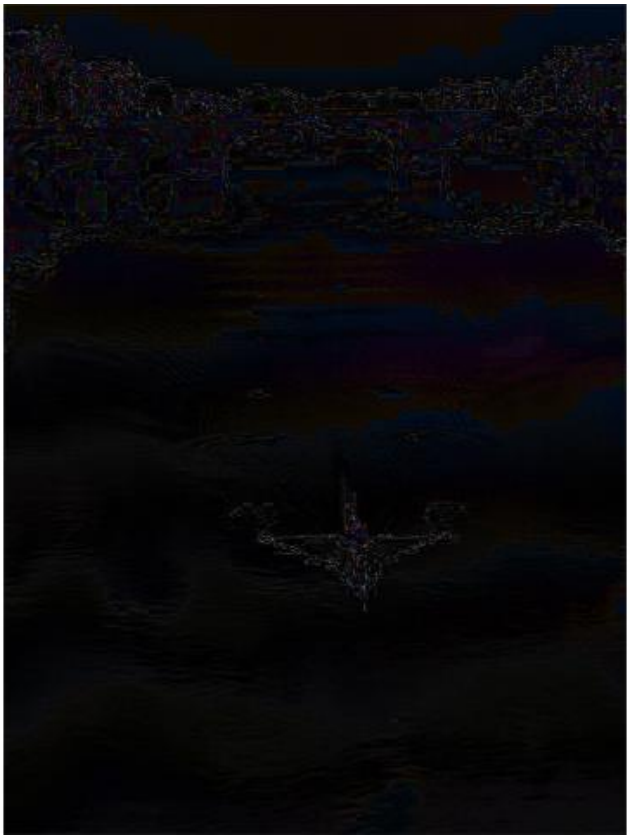


Imagen diferencial entre original y comprimida CaliQ 1000

Obtenemos buenos resultados de compresión, muy parecidos a la primera imagen incluso con menos MSE. Como intuíamos el hombre que maneja la canoa no se comprime del todo bien al igual que el contorno de los edificios y los árboles del fondo. A medida que aumenta el FC, vemos que en el agua se van marcando los distintos tonos de color que presenta y que se confunde con el cielo. También es importante destacar la pérdida de definición en las ondas del agua debido a las frecuencias altas que esto supone y son eliminadas con la cuantización de coeficientes.

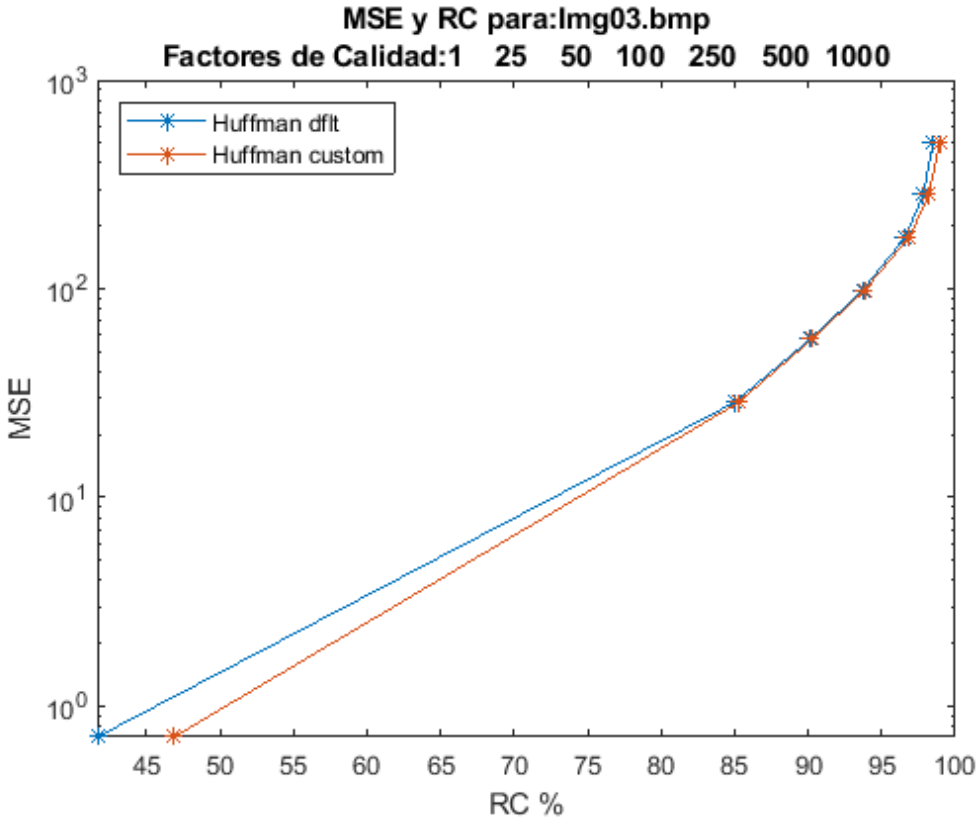
3.3 IMAGEN 03.BMP

Esta imagen es muy buena para ver como JPEG falla mucho en el cambio de objeto y la pérdida de información que se produce en los contornos. Ya que todos los candados aglomerados tienen tonalidades distintas, a pesar de que a simple vista casi todos son de color dorado, si nos fijamos todos tienen ciertas particularidades que van a hacer que JPEG no pueda comprimirlo muy bien, por ejemplo, las letras de algunos candados, la pintura con rotulador que llevan algunos, las cerraduras y los diferentes colores que presentan algunos. Para esta imagen espero bastantes errores en su compresión por todo esto que he mencionado antes que generará poca eliminación de coeficientes de altas frecuencias ya que van a tener valores altos.

Imagen03.bmp Original

Factores	MSE_Dflt	MSE_Custom	RC_Dflt	RC_Custom
1	0,71	0,71	41,66	46,87
25	28,58	28,58	85,01	85,28
50	57,31	57,31	90,13	90,27
100	98,40	98,40	93,70	93,83
250	176,95	176,95	96,61	96,89
500	282,20	282,20	97,82	98,20
1000	501,02	501,02	98,50	98,94

Tabla para Img03.bmp



Gráfica para Img03.bmp

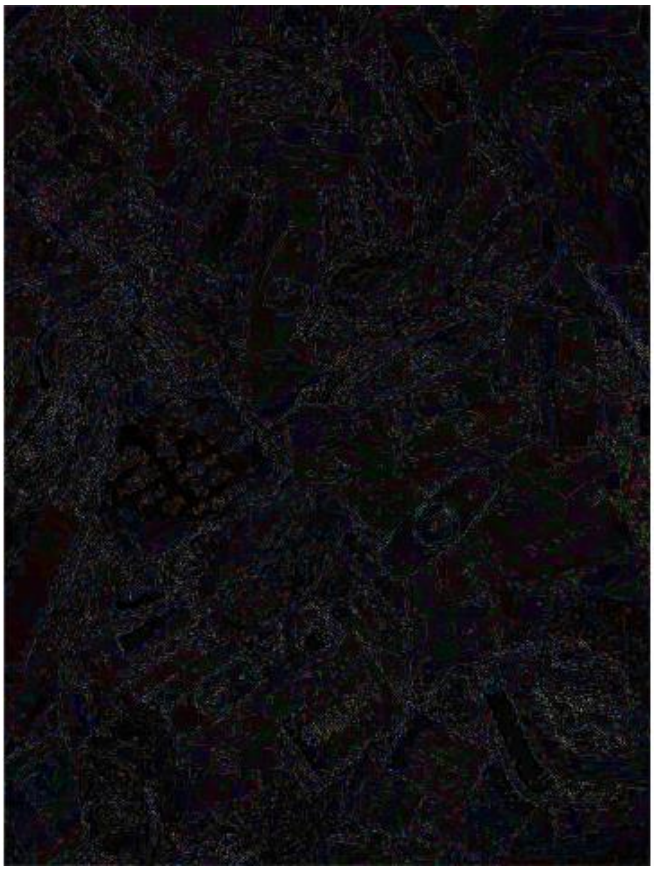


Imagen diferencial entre original y comprimida CaliQ 1000

Como observamos en la tabla y en la gráfica el error es muy alto para esta imagen, comparado con las otras dos imágenes podemos ver que es más del doble para casi todos los FC, esto se debe a lo pensado primeramente en la hipótesis de partida y viendo las imágenes nos damos cuenta de que los textos escritos a mano en los candados dejan de verse al igual que algunos agujeros de cerraduras se convierten en una parte más del cuerpo del candado, aunque a simple vista incluso con FCs altos se conserva bien la imagen. Viendo la imagen diferencial vemos fallos en la mayoría de la imagen y sobre todo en los contornos de los candados como suponíamos. En conclusión, es una imagen difícil para JPEG debido a sus valores altos en los coeficientes de altas frecuencias seguramente.

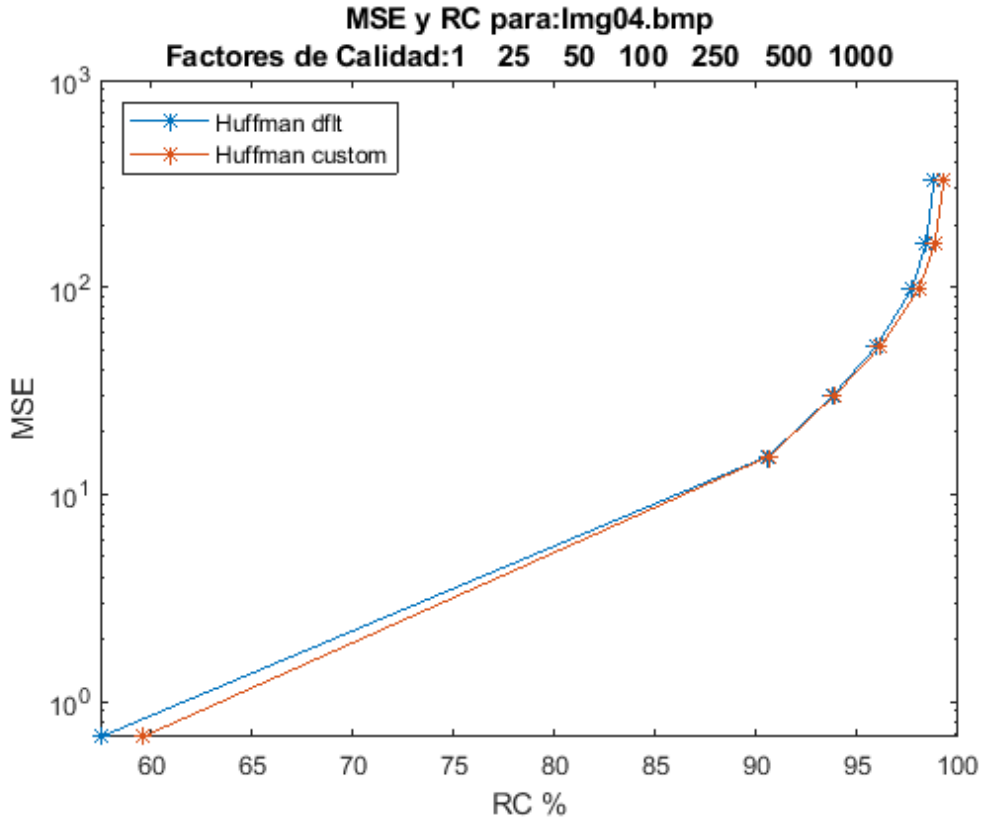
3.4 IMAGEN 04.BMP

Esta imagen es muy completa ya que tenemos el arco de piedra, además el arco es de distinto color que la pared eso generará también fallos por tener cambios bruscos de color, al igual que pasa con el contorno de la montaña el cuál no se espera que se comprima muy bien, al igual que en las líneas del suelo de la imagen de la primera imagen, aquí tenemos una línea de agua en el suelo, que se espera que vaya desapareciendo y mimetizándose con el suelo a medida que aumentemos el FC. Lo que no tengo muy claro es que pasará con las personas que hay en la escena. Lo comentaremos después.

Imagen04.bmp Original

Factores	MSE_Dflt	MSE_Custom	RC_Dflt	RC_Custom
1	0,67	0,67	57,46	59,55
25	15,27	15,27	90,57	90,63
50	30,23	30,23	93,82	93,91
100	52,24	52,24	95,96	96,18
250	98,09	98,09	97,71	98,07
500	164,23	164,23	98,41	98,84
1000	330,70	330,70	98,81	99,30

Tabla para Img04.bmp



Gráfica para Img04.bmp

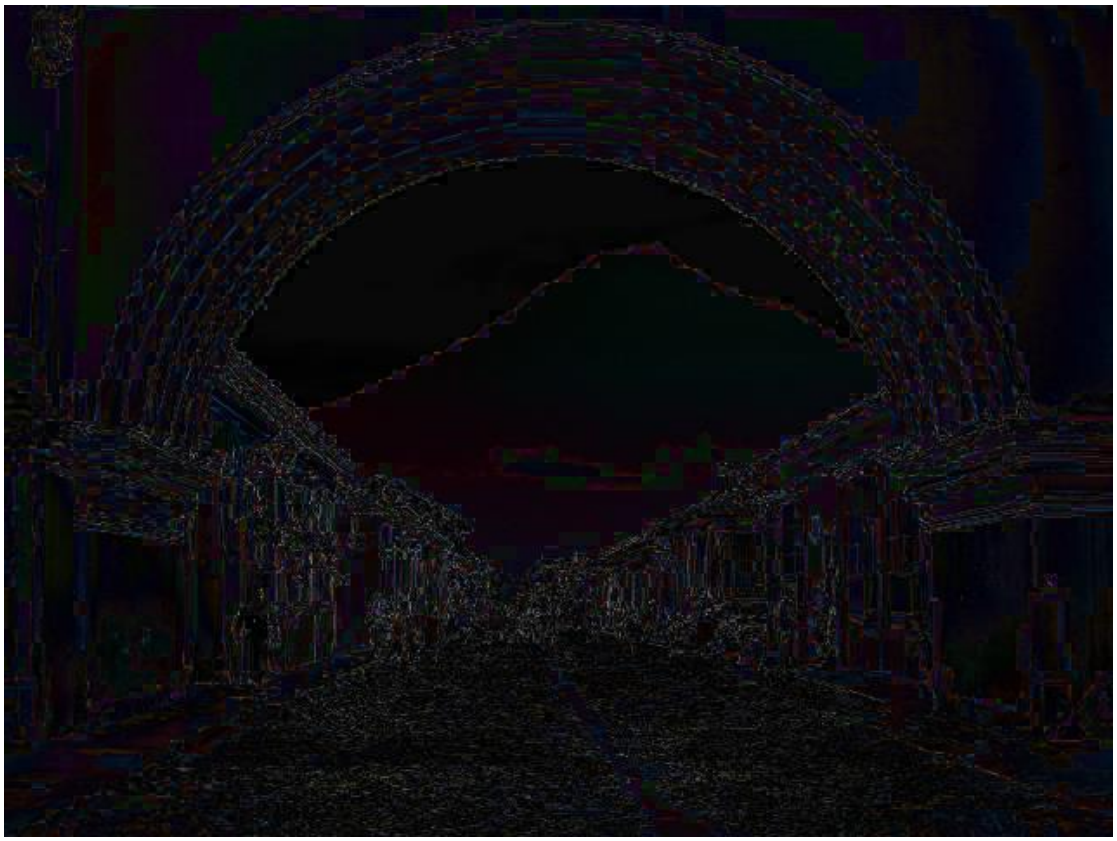


Imagen diferencial entre original y comprimida CaliQ 1000

Tenemos un error más que aceptable para todos los elementos que forman parte de esta imagen y eso es lo bueno de JPEG que aunque haya muchos elementos distintos la mayoría de esos elementos en su contenido siguen un patrón de cambio de color muy progresivo y eso es lo que hace que perdamos información de los contornos como estamos observando en la imagen diferencial que perfectamente podríamos reconstruir casi toda la silueta de los objetos de la imagen con las diferencias, pero no perdemos información del contenido de los objetos. Como era de esperar el cambio brusco de color de la pared al blanco del arco ha sido objeto de pérdida de información, ya que si nos damos cuenta entre cada línea blanca hay una línea fina amarilla y esto implica alta alternancia de colores. Por lo demás todo era igual a lo esperado.

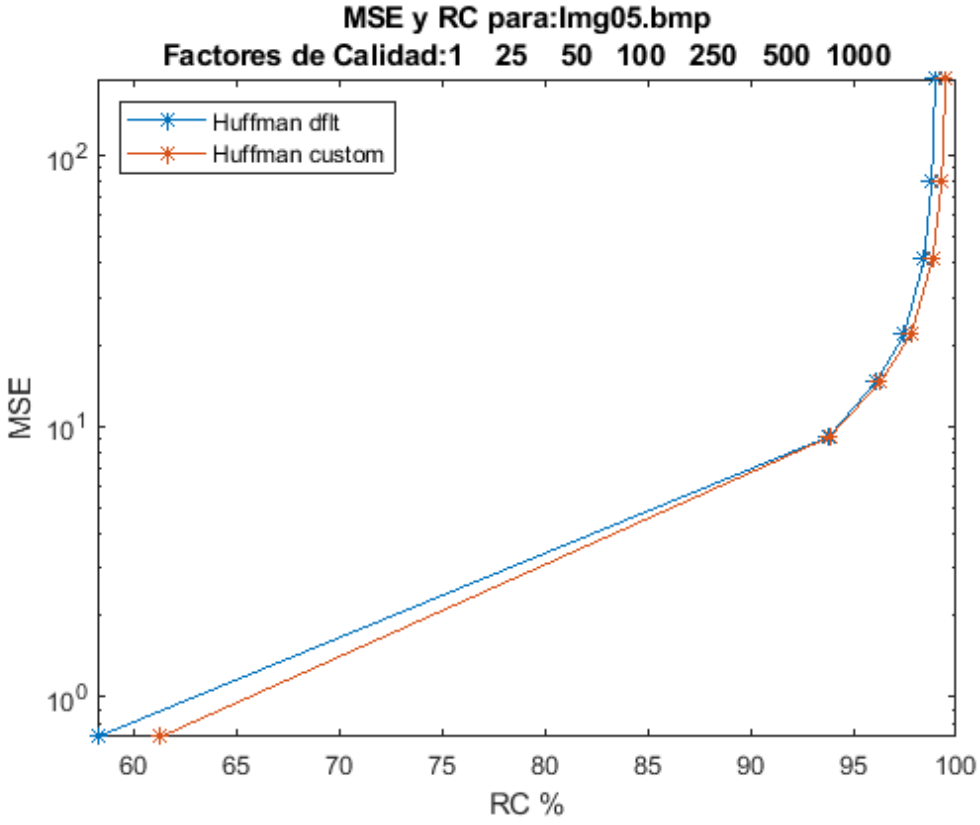
3.5 IMAGEN 05.BMP

Esta imagen ha sido escogida porque pienso que va a tener pocos errores ya que predomina mucho el motor del avión y tiene una zona de color blanco que será muy bien comprimida. De todas formas, espero que haya errores alrededor del contorno del motor y en la zona que cambia las nubes por tierra también ser verá afectada por el cambio brusco de color. Las nubes creo que será buena zona para comprimir ya que podremos quitar bastantes valores de las altas frecuencias, pero no tengo muy claro que pasará con esta última zona.

Imagen05.bmp Original

Factores	MSE_Dflt	MSE_Custom	RC_Dflt	RC_Custom
1	0,71	0,71	58,23	61,29
25	9,13	9,13	93,79	93,86
50	14,65	14,65	96,12	96,33
100	21,81	21,81	97,47	97,81
250	41,83	41,83	98,46	98,87
500	80,22	80,22	98,82	99,30
1000	191,54	191,54	98,99	99,49

Tabla para Img05.bmp



Gráfica para Img05.bmp

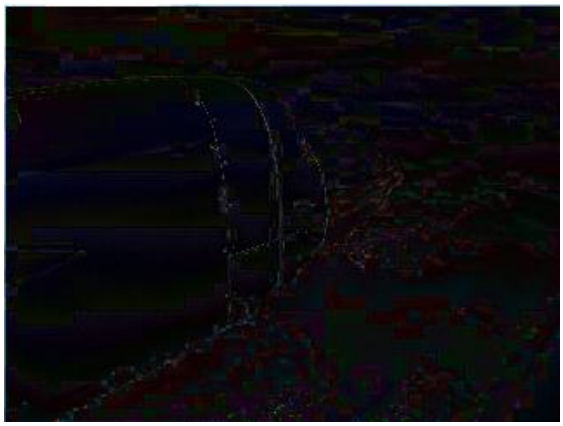


Imagen diferencial entre original y comprimida CaliQ 1000

Viendo el error, pensaba que se iba a obtener un error menor, aún que no está nada mal y podemos resaltar que las zonas más afectadas son los contornos que distinguen las partes del motor del avión se debe al cambio a color negro y una curiosidad es que al ser una imagen en la que predomine tanto el color blanco con un FC = 1000 la imagen se convierte casi entera blanca ya que elimina cualquier otro tipo de información de color con la cuantización.

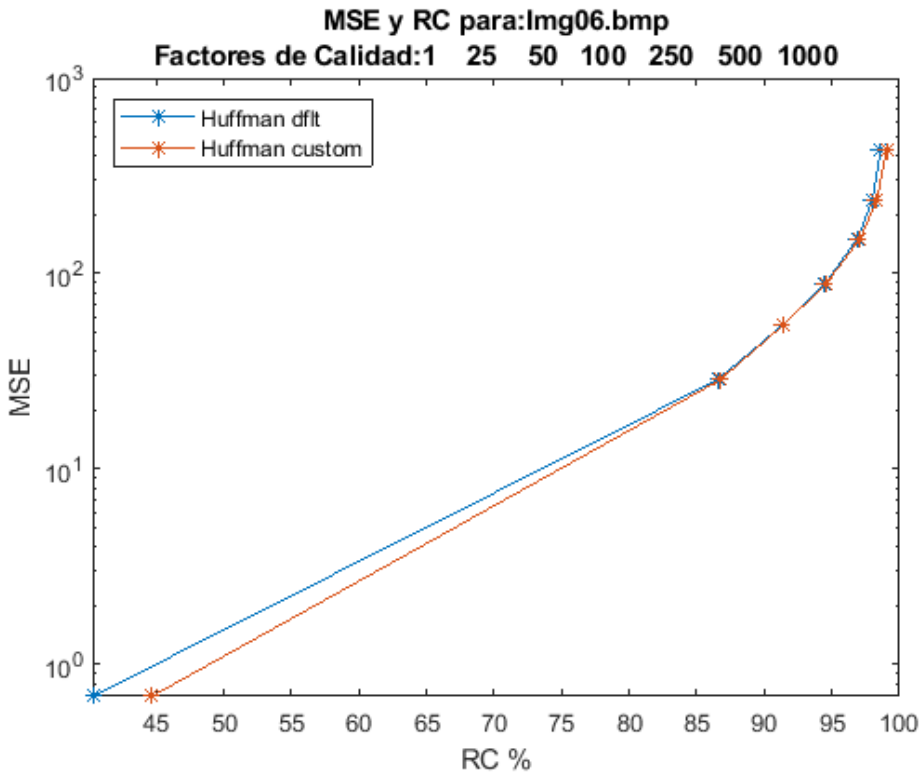
3.6 IMAGEN 06.BMP

Esta imagen espero que se comprima bastante bien y es muy buena para ver el objetivo de JPEG que es no dejar que el ojo humano aprecie la pérdida de información. Vemos que tiene muchos colores persistentes en distintas zonas locales de la imagen y eso va a hacer que se compriman bastante bien y por otro lado también vemos que los cambios de color entre el fondo azul, el hombre, el piano y la chaqueta amarilla son bastante pronunciados, esto va a hacer que perdamos bastante en los valores de la crominancia en los contornos seguramente. Por lo general, creo que se verá bastante bien la imagen comprimida.

Imagen06.bmp Original

Factores	MSE_Dflt	MSE_Custom	RC_Dflt	RC_Custom
1	0,69	0,69	40,29	44,67
25	28,70	28,70	86,56	86,76
50	54,54	54,54	91,34	91,39
100	87,78	87,78	94,43	94,50
250	151,00	151,00	96,90	97,10
500	239,47	239,47	98,00	98,34
1000	429,44	429,44	98,62	99,05

Tabla para Img06.bmp



Gráfica para Img06.bmp

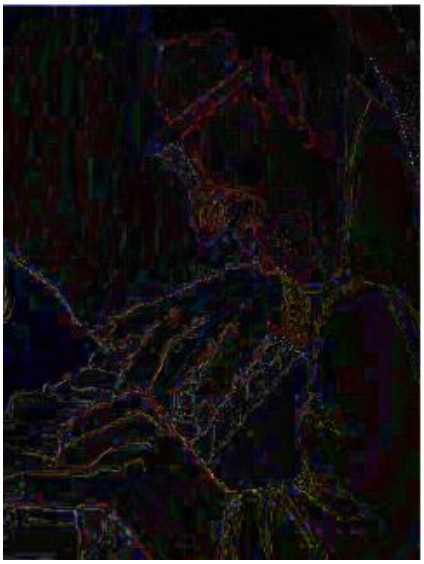


Imagen diferencial entre original y comprimida CaliQ 1000

Hemos obtenido un error razonable para los cambios que comentábamos en los contornos de los objetos de la figura, si nos fijamos vemos muchos fallos en las dimensiones Cb y Cr apreciables en la imagen diferencial y no tiene un nivel de compresión alto hasta un FC de 50, pero también hay que destacar que cómo pensábamos las imágenes se ven muy bien aun siendo comprimidas y con valores de

FC altos se sigue manteniendo la información y distinguiendo las zonas perfectamente incluso las distintas tonalidades de los colores, un muy buen trabajo de JPEG.

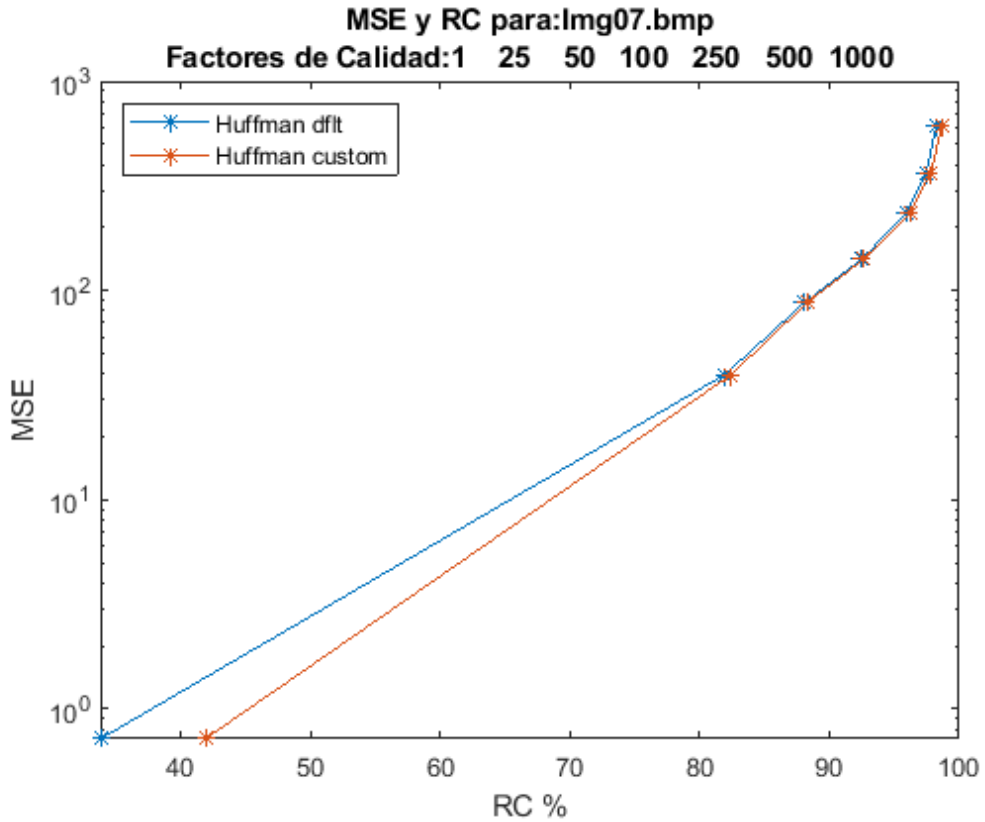
3.7 IMAGEN 07.BMP

Por último, nos queda la imagen que ha sido escogida a propósito para poner a prueba a JPEG, esta imagen como vemos es pura alternancia de colores lo que nos vamos a ver envueltos en resultados bastante nefastos en la compresión seguramente, el hecho de la diferencia tan alta en el verde y el gris y en su repetición va a hacer que la DCT deje valores muy altos en los coeficientes de frecuencias altas (esquina inferior derecha de la matriz) con lo que será realmente complejo para JPEG que incluso se vea bien la recomposición de la imagen, lo que se espera es que a medida que vaya aumentando el FC las zonas verdes se vean grises y las zonas grises se vean verdes y que se vayan mezclando los colores en los contornos.

Imagen07.bmp Original

Factores	MSE_Dflt	MSE_Custom	RC_Dflt	RC_Custom
1	0,72	0,72	33,83	41,91
25	39,62	39,62	81,94	82,44
50	87,24	87,24	88,06	88,28
100	141,15	141,15	92,53	92,66
250	235,54	235,54	95,95	96,22
500	363,20	363,20	97,45	97,81
1000	617,45	617,45	98,26	98,70

Tabla para Img07.bmp



Gráfica para Img07.bmp

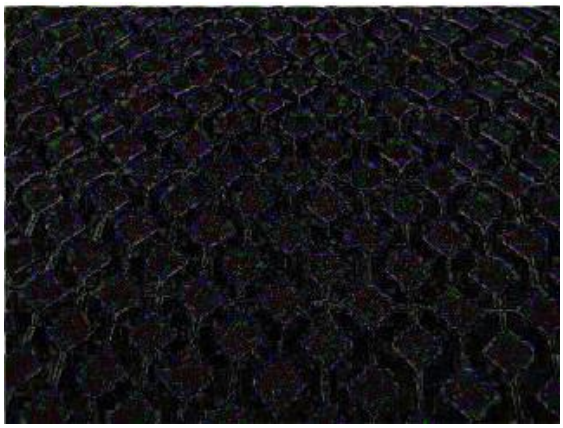


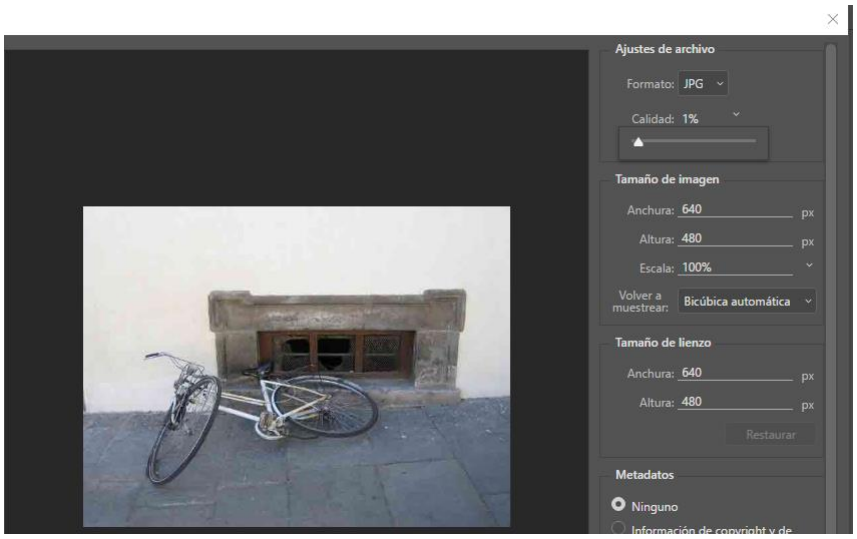
Imagen diferencial entre original y comprimida CaliQ 1000

Como esperábamos esta es la imagen con valores de MSE más altos y es que parece que está hecha a propósito para ver el mal funcionamiento de JPEG en este tipo de situaciones. Si vemos las imágenes vemos que la zona de la hierba se va poniendo cada vez más con un verde más oscuro casi negro por la mezcla de colores con el gris y también vemos que ciertas partes de las baldosas tienen un color verde claro. Una cosa que no me había percatado antes son las franjas que ahí entre las baldosas el comportamiento que tiene JPEG es el mismo que hemos visto en anteriores imágenes con las líneas que son de distinto color que el suelo en la primera imagen o como el agua del suelo que había en la cuarta imagen, al final la franjas se van alisando hasta perderse. Aun así, he de decir que para ser un ejemplo tan crítico para JPEG si vemos

las imágenes reconstruidas no se pierde en ningún momento las formas geométricas y se distinguen bastante bien las baldosas del césped.

4.COMPARACIÓN CON PHOTOSHOP

En este apartado vamos a comparar nuestro compresores y descompresores realizados en la práctica con los que tiene incorporados Photoshop. Como vemos en la siguiente imagen Photoshop nos ofrece un factor de calidad en %, siendo 0% de calidad la peor y 100% la mejor, lo que sería a la inversa que nuestros factores de calidad, vamos a ver cómo se comporta Photoshop para % calidad = {1, 50, 100} y ver si tienen coincidencias con alguno de nuestro factor de calidad. Para comparar las imágenes originales con las comprimidas, vamos a calcular el RC y MSE entre la original y la comprimida para cada porcentaje. Para esto se ha implementado un script en MatLab que calcula todas las tablas correspondientes a las 7 imágenes de este trabajo. Lo que se va a intentar ver es con que factor de calidad encaja algún porcentaje de calidad de photoshop.



Cálculo del factor de calidad en Photoshop.

4.1 IMAGEN 01.BMP

%Calidad	RC	MSE
1	98.642	51.894
50	95.426	15.054
100	73.355	1.0502

Tabla de compresión con Photoshop para Img01.bmp

Como podemos apreciar para un % de calidad máximo de la imagen no se corresponde con el MSE de factor de calidad 1 en la tabla ya que tiene un MSE un 64% mayor. Eso sí, Photoshop comprime mejor que nuestro compresor en esta comparación, y obtiene un RC casi 10 puntos mayor para este escalón debido a que el FQ = 1 no se corresponde con porcentaje 1, tiene que ser un FQ mayor. El 50% de calidad vemos que está entre nuestro 25 y 50 factor de calidad y que el 1% de calidad lo encontramos entre el 100 y

250 de nuestro factor de calidad **si miramos el MSE**. Cuando se va haciendo mayor el porcentaje de calidad no vemos mucha diferencia, por ejemplo, para un 1% de calidad tenemos 98,6% de RC y 51,9 MSE con nuestro compresor alcanzamos 98,87% de RC con un 63 de MSE con un factor de calidad de 250. Pero sí que podemos ir deduciendo que con FQ = (100,250) no conseguimos un resultado parecido al de Photoshop con el 1% de calidad, ya que tenemos una diferencia de 0,3% de RC pagando 11 puntos en el MSE, y esto se debe a que Photoshop comprime mejor que nosotros.

4.2 IMAGEN 02.BMP

%Calidad	RC	MSE
1	98.933	34.11
50	96.582	11.218
100	74.414	1.1806

Tabla de compresión con Photoshop para Img02.bmp

Aquí vemos lo mismo que pasaba en la imagen anterior, con un 100% de calidad casi se dobla el MSE que con factor de calidad 1, aunque sube 9 puntos la RC. El 50% de calidad se corresponde con un factor de calidad entre 50 y 25 como antes y con una RC de 96.6% lo que podría ser casi igual a nuestro compresor ya que nosotros obtenemos un 97.1 % de RC con factor de calidad 50 y pagando un MSE de 14,16, lo más seguro es que Photoshop esté comprimiendo mejor la imagen, con 1% de calidad de imagen se corresponde a un factor de calidad entre 100 y 250, más cercano a 250 que 100 y nosotros con FQ = 250 pagamos 39,21 de MSE pero obtenemos mejor RC que con el 1% de Photoshop por 2 décimas. Lo cual quiere decir que Photoshop sigue comprimiendo mejor ya que por 2 décimas estamos pagando 5 puntos en el MSE, si comparamos 1% de calidad frente al FQ = 250.

4.3 IMAGEN 03.BMP

%Calidad	RC	MSE
1	96.736	165.58
50	89.384	41.78
100	56.683	1.464

Tabla de compresión con Photoshop para Img03.bmp

Otra vez vemos el mismo patrón de que el 1% de calidad se asigna al intervalo FQ = (100,250) más próximo al 250 que al 100, para un 50% de calidad se le asigna el intervalo FQ = (25,50) situándolo más o menos en el medio y con 100% de calidad sería algo más que el FQ = 1. De todas formas, comparando las diferencias de MSE y viendo el RC de ambos vemos que Photoshop tiene mejores cifras.

4.4 IMAGEN 04.BMP

%Calidad	RC	MSE
1	97.893	89.694
50	93.299	21.859
100	68.998	1.1268

Tabla de compresión con Photoshop para Img04.bmp

Sigue el mismo patrón comentado previamente en las otras imágenes.

4.5 IMAGEN 05.BMP

%Calidad	RC	MSE
1	97.778	36.045
50	94.948	12.001
100	70.026	1.4876

Tabla de compresión con Photoshop para Img05.bmp

Sigue el mismo patrón que hemos comentado previamente.

4.6 IMAGEN 06.BMP

%Calidad	RC	MSE
1	96.299	155.35
50	90.425	51.01
100	54.027	1.5935

Tabla de compresión con Photoshop para Img06.bmp

Aquí vemos un ejemplo que rompe el patrón, por muy poco, pero para un 1% de calidad vemos que se corresponde con el intervalo FQ = (250,500) muy cercano a 250, pero eso no es todo, también vemos que nuestro compresor lo hace mejor ya que obtenemos un 97,1% de RC frente al 96,3% de Photoshop y además pagando menos puntos en el MSE. Y además para el 50% de calidad se acerca mucho al FQ = 50 que antes solía estar en la mitad entre FQ = (25,50). Y para el 100% de calidad ahora si que dista bastante de nuestro FQ = 1 ya que el MSE obtenido por Photoshop es más del doble del MSE obtenido por nuestro compresor. Cuando en las anteriores imágenes estaba en el intervalo entre el 50% y el 100%.

4.7 IMAGEN 07.BMP

%Calidad	RC	MSE
1	95.314	237.67
50	86.222	57.081
100	50.415	1.6316

Tabla de compresión con Photoshop para Img07.bmp

Esta imagen también supera los límites comentados en la anterior imagen, pero si nos damos cuenta tampoco son números muy alejados de lo que estábamos viendo anteriormente. Por lo que podemos afirmar que Photoshop comprime mejor en general las imágenes escogidas, con mínimas excepciones.

5.CONCLUSIÓN.

Como conclusión del trabajo podemos afirmar que JPEG es un gran método de compresión de imágenes ya que obtiene un porcentaje muy alto de compresión sin incurrir en fallo tan altos para la vista del ser humano.

El trabajo creo que ha sido muy interesante ya que hemos podido pasar a un nivel más allá de la programación, que es la investigación y el poder sacar nuestras propias conclusiones acerca de un método tan famoso e importante como éste. Además, el transcurso de las prácticas en mi opinión ha sido bastante bueno ya que poco a poco se va metiendo al alumno en el método sin que él lo sepa mucho, y se le van facilitando las herramientas necesarias para el desarrollo de la misma, así cuando se explica el último boletín ya se pueden juntar todas las piezas del puzzle y comprimir y descomprimir imágenes. A mí eso en lo particular me ha gustado mucho.

6.BIBLIOGRAFÍA

(1) Fórmula de cambio de espacio de color.
http://bibing.us.es/proyectos/abreproy/11210/fichero/Memoria+por+Cap%C3%ADtulos+%252F7+El+formato+JPEG.pdf

(2) Imágenes de compresión y descompresión JPEG

http://bibing.us.es/proyectos/abreproy/11210/fichero/Memoria+por+Cap%C3%ADtulos+%252F7+El+formato+JPEG.pdf

(3) Imagen matrices base JPEG.

https://www.freecodecamp.org/news/how-jpg-works-a4dbd2316f35/

(4) Apuntes del Aula Virtual de UM, de la asignatura Compresión Multimedia del Grado Ingeniería Informática.

https://aulavirtual.um.es/access/content/group/1919_G_2020_N_N/Teor%C3%ADa%20y%20Seminarios/Bloque%203%3A%20T%C3%A9cnicas%20de%20Compresi%C3%B3n%20Multimedia/TEMA%207.%20Compresi%C3%B3n%20de%20im%C3%A1genes/07CompresionImagenes.pdf

(5) Imagen zig-zag JPEG

https://es.wikipedia.org/wiki/Archivo:Zigzag_scanning.jpg