

The Strengths and Weaknesses of Different Image Compression Methods

Samuel Teare and Brady Jacobson

Lossy vs Lossless

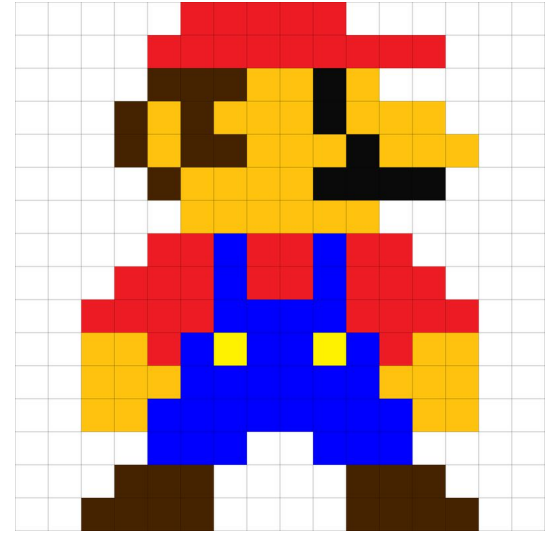
Lossy compression reduces a file size by permanently removing parts of the data that may be redundant or not as noticeable.

Lossless compression guarantees the original data can be recovered or decompressed from the compressed file.

PNG Compression

PNG Compression consists of three parts:

1. Filtering
 2. LZ77 Compression
 3. Huffman Coding
- } Deflate Compression



Filtering

Five types of Filters:

1. None - No filter
2. Sub - difference between this byte and the byte to its left
 - a. $\text{Sub}(x) = \text{Original}(x) - \text{Original}(x - \text{bpp})$
3. Up - difference between this byte and the byte above it
 - a. $\text{Up}(x) = \text{Original}(x) - \text{Above}(x)$
4. Average - difference between this byte and the average of the byte to the left and the byte above.
 - a. $\text{Avg}(x) = \text{Original}(x) - (\text{Original}(x - \text{bpp}) + \text{Above}(x)) / 2$
5. Paeth - Uses the byte to the left, above, and above left.
 - a. The nearest of the left, above, or above left to the estimate is the Paeth Predictor
 - b. $\text{Paeth}(x) = \text{Original}(x) - \text{Paeth Predictor}(x)$

Paeth Algorithm

Estimate = left + above - above left

Distance to left = Absolute(estimate - left)

Distance to above = Absolute(estimate - above)

Distance to above left = Absolute(estimate - above left)

The byte with the smallest distance is the Paeth Predictor

LZ77 Compression

LZ77 Compression looks for sequences in the data that are repeated.



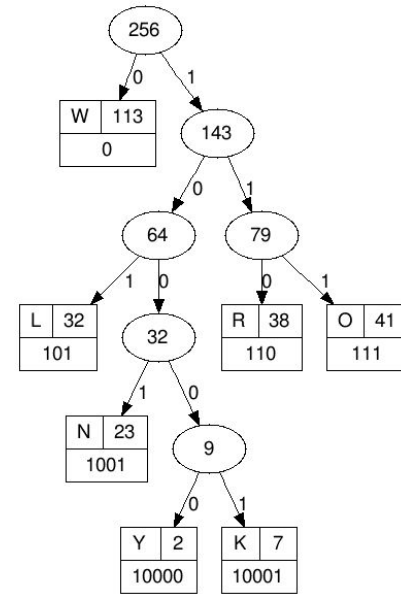
LZ77 uses a *sliding window* to keep track of previous bytes. This is then used to compress a group of bytes that exhibit the same sequence as previous bytes. The compression takes the form of the bytes leading up to the repeated sequence, the distance behind the current byte that the sequence starts, and the length (number) of repeated bytes.

Example: White [D=1, L=4] Red [D=1, L=4] [D=10, L=5]

Huffman Coding

Huffman coding looks at the frequency of literals (characters or pixels).

The frequency of the literals is organized into a Huffman tree. This allows the bytes of the literals to be represented by only a few bits. The more frequent the literal the fewer number of bits used to represent that literal.



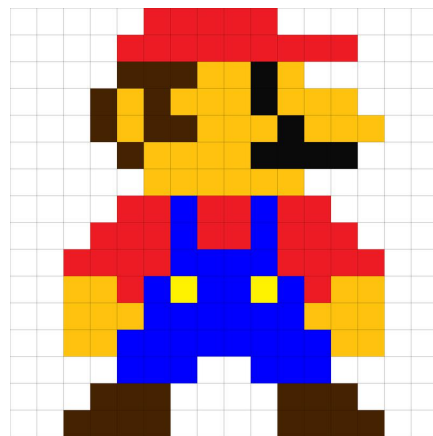
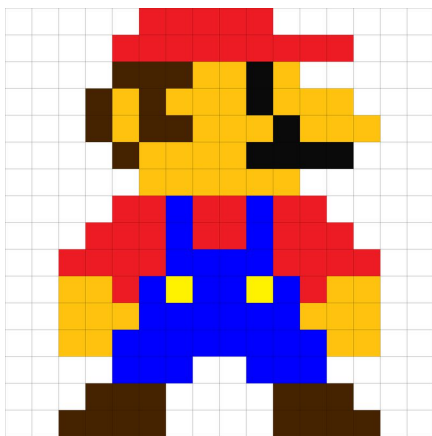
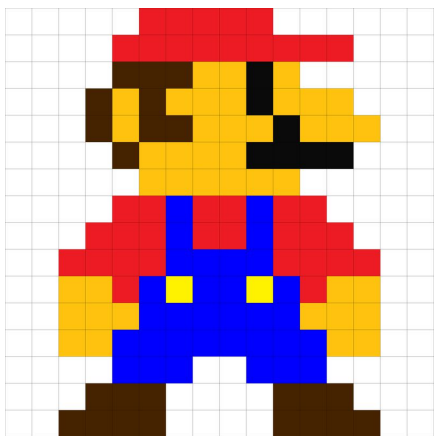
Size Comparison

8-bit Mario: (800x800)

PNG: 35,358 (35KB)

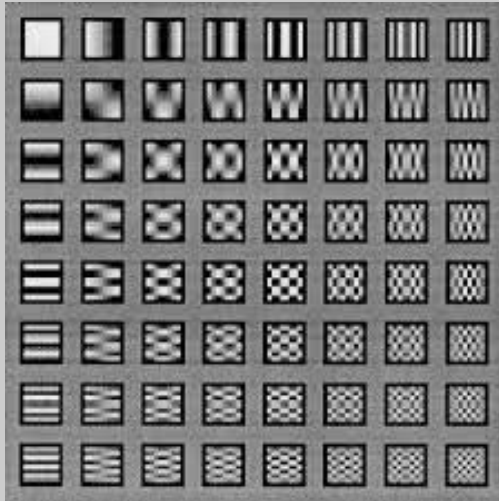
TIFF: 125,274 (125KB)

JPG: 167,046 (167KB)



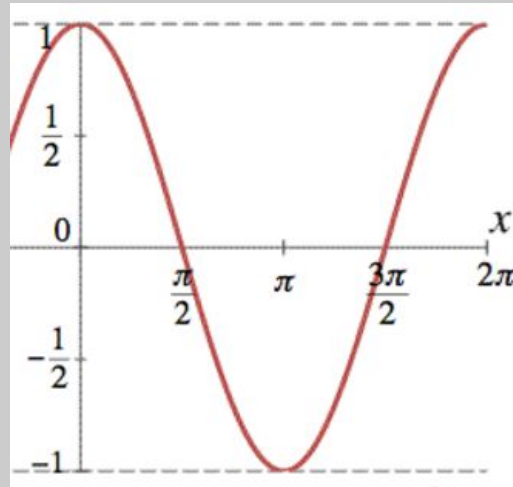
JPEG

JPEG is type of image compression that uses Discrete Cosine Transformations. The degree of compression can be changed by altering the quantization table.

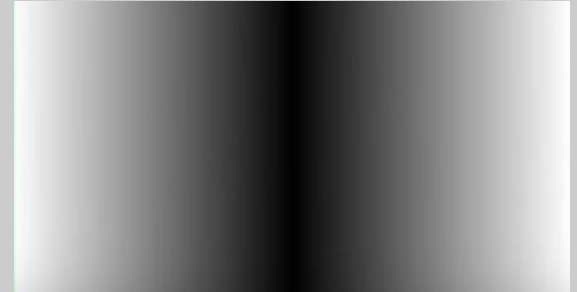


Discrete Cosine Transform

A type of Lossy transformation. For JPEG, this method takes a collection of cosine lines, each representing a different image. By adding these cosine lines together, we can recreate the original image.



=



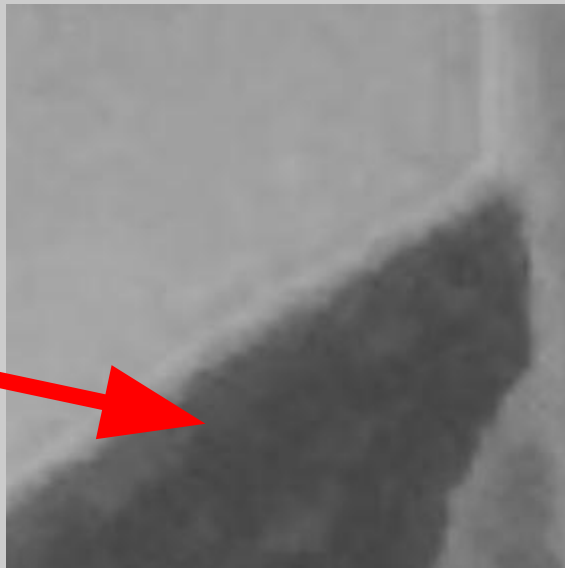
JPEG Compression consists of the following steps:

1. Make sure the color format is YCbCr.
2. Separate the image into groups of 8 by 8 pixels.
3. Perform two-dimensional Discrete Cosine Transform on the values.
4. Perform Quantization.
5. Perform Huffmans Coding.

Original Image

8 by 8 pixel image

Input table (Random Example)



106	104	101	106	108	106	110	126
106	106	104	104	106	106	106	115
99	101	101	101	101	106	110	115
99	95	101	101	106	119	164	126
95	97	101	101	160	169	173	133
95	95	142	169	178	176	178	126
106	160	171	176	176	180	171	146
162	173	173	178	167	180	135	144

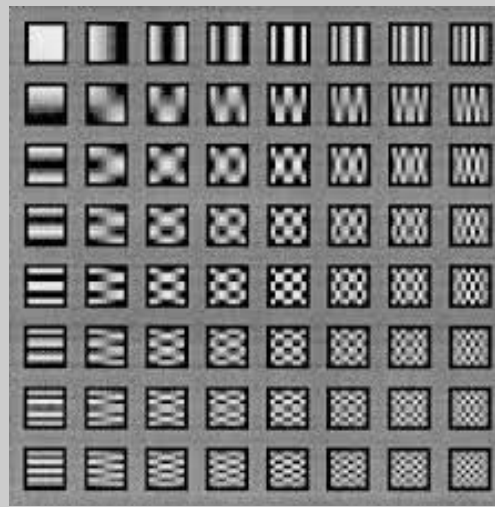
First, convert the original image from RGB to YCbCr. The Y component of this colorspace represents brightness, which is more useful for compressing than the hue or color of an image.

Secondly, we divide the image into groups of 8 by 8 pixels. From here we take the brightness of each and place it in a chart.

The brightness is between 0 and 255. We now center the list around 0 instead of 128. This would match a Cosine line. We take these values and perform two dimensional DCT-II on them.

$$G_{u,v} = \frac{1}{4} \alpha(u) \alpha(v) \sum_{x=0}^7 \sum_{y=0}^7 g_{x,y} \cos \left[\frac{(2x+1)u\pi}{16} \right] \cos \left[\frac{(2y+1)v\pi}{16} \right]$$

The results are the coefficients, which match up with the chart of possible patterns. This is an example of the chart. Each value represents the influence of its respective pattern. (Values near the top left, such as A, usually have more influence than values at the bottom right, such as BL.)



A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	AA	AB	AC	AD	AE	AF
AG	AH	AI	AJ	AK	AL	AM	AN
AO	AP	AQ	AR	AS	AT	AU	AV
AW	AX	AY	AZ	BA	BB	BC	BD
BE	BF	BG	BH	BI	BJ	BK	BL

This is an example of a Quantization table. This differs between different methods, which can allow for different compression results. In most cases, values at the top left are smaller than values on the bottom left.

a	b	c	d	e	f	g	h
i	j	k	l	m	n	o	p
q	r	s	t	u	v	w	x
y	z	aa	ab	ac	ad	ae	af
ag	ah	ai	aj	ak	al	am	an
ao	ap	aq	ar	as	at	au	av
aw	ax	ay	az	ba	bb	bc	bd
be	bf	bg	bh	bi	bj	bk	bl

Coefficient table

A	B	C	D	E	F	G	H
I	J	K	L	M	N	O	P
Q	R	S	T	U	V	W	X
Y	Z	AA	AB	AC	AD	AE	AF
AG	AH	AI	AJ	AK	AL	AM	AN
AO	AP	AQ	AR	AS	AT	AU	AV
AW	AX	AY	AZ	BA	BB	BC	BD
BE	BF	BG	BH	BI	BJ	BK	BL

Quantization table

a	b	c	d	e	f	g	h
i	j	k	l	m	n	o	p
q	r	s	t	u	v	w	x
y	z	aa	ab	ac	ad	ae	af
ag	ah	ai	aj	ak	al	am	an
ao	ap	aq	ar	as	at	au	av
aw	ax	ay	az	ba	bb	bc	bd
be	bf	bg	bh	bi	bj	bk	bl

Quantized table

Z	Y	X	W	0	0	0	0
V	U	I	0	0	0	0	0
S	R	0	0	0	0	0	0
0	Q	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

The Quantization step is the only lossy operation in the entire method. Some frequencies of brightness are unnecessary. To get rid of unnecessary values, we take each coefficient and divide it by its respective value from the quantization table, and round each to the nearest whole number. This allows us to get rid of smaller influences, mostly values like BL at the bottom right. With A/a , the resulting Z will usually remain large. More importantly, for equations like BL/bl , the result will be rounded to 0.

Original Image



Compressed Image



JPEG vs PNG

JPEG: 2.8MB



PNG: 5.9MB

