JPEG[1]

- Divide image into 8×8 blocks.
- Center gray values (i.e., subtract 128).
- Compute the DCT of each block.
- Quantize DCT coefficients using quantization table.
- Compute differences of quantized DCT coefficients along zig-zag path.
- Run-length encode differences.
- Huffman-code run-lengths.
- Huffman-code labels.

[1] Joint Photographic Experts Group

Why it Works

Let's consider a discrete one-dimensional signal. Compute the covariance matrix for values inside of a moving 1×8 window:

$$\mathbf{C} = \frac{1}{N}\sum_{n=0}^{N-1} \left[\, f_{n+0} \cdots \;\; f_{n+7} \,\right]^{\mathrm{T}} \left[\, f_{n+0} \;\; \cdots \;\; f_{n+7} \,\right]$$

$$= \begin{bmatrix} \langle f_{n+0} f_{n+0} \rangle & \langle f_{n+0} f_{n+1} \rangle & \cdots & \langle f_{n+0} f_{n+7} \rangle \\ \langle f_{n+1} f_{n+0} \rangle & \langle f_{n+1} f_{n+1} \rangle & \cdots & \langle f_{n+1} f_{n+7} \rangle \\ \vdots & \vdots & \ddots & \vdots \\ \langle f_{n+7} f_{n+0} \rangle & \langle f_{n+7} f_{n+1} \rangle & \cdots & \langle f_{n+7} f_{n+7} \rangle \end{bmatrix}.$$

We observe that:

$$\begin{aligned} \langle f_{n+i} f_{n+j} \rangle &= \langle f_{m+i} f_{m+j} \rangle \\ &= \left\langle f_{n+(i-j)} f_{n+0} \right\rangle. \end{aligned}$$

Why it Works (contd.)

- Because **C** is circulant, its eigenvectors are sampled harmonic signals.
- Consequently, the DCT is really the KL transform of the window in disguise!
- The quantization table tells us which eigenvectors are most important perceptually.
- Observation: JPEG compresses by dimensionality reduction.