E9 222 Signal Processing in Practice
Discrete Cosine Transform (DCT)

Instructor: Chandra Sekhar Seelamantula

Due: 21 January 2026

Learning Objectives

- Construct DCT basis functions (1D and 2D),

- Interpret DCT coefficients as energy in frequency-like modes,

- Implement a simplified JPEG-style compression pipeline and study energy compaction.

Notation. For a length-N signal $x[n]$, $n = 0, \ldots, N-1$, the (orthonormal) DCT-II basis can be written as:

$$\phi_k[n] = \alpha_k \cos\left(\frac{\pi}{N}\left(n + \tfrac{1}{2}\right)k\right), \quad \alpha_k = \begin{cases} \sqrt{\frac{1}{N}}, & k = 0, \\ \sqrt{\frac{2}{N}}, & k = 1, 2, \ldots, N-1. \end{cases}$$

Then $\hat{x}[k] = \sum_{n=0}^{N-1} x[n]\phi_k[n]$ and (perfect reconstruction) $x[n] = \sum_{k=0}^{N-1} \hat{x}[k]\phi_k[n]$.

---

1 Part 1: Plot 1D DCT basis functions

(a) Construct the basis

Let $N = 32$ and define the orthonormal DCT-II basis functions $\{\phi_k[n]\}_{k=0}^{N-1}$ using the formula above.

a) Implement a function that returns the $N \times N$ DCT matrix D with entries $D_{k,n} = \phi_k[n]$.

b) Verify numerically that D is orthonormal by computing $\|DD^\top - I\|_F$ (Frobenius norm). Report the value.

(b) Visualize basis functions

Create a single figure showing the first **eight** basis functions $k = 0, 1, \ldots, 7$ (e.g., in a 2×4 grid). Each subplot must include: the curve $\phi_k[n]$ vs. n, and the label k.

2 Part 2: Plot 2D DCT basis functions

(a) Separable 2D basis

Let $M = N = 8$. Define a 2D DCT basis image (basis *pattern*) for indices (u, v) as:

$$\Phi_{u,v}[m, n] = \phi_u[m]\phi_v[n], \quad m, n \in \{0, \ldots, 7\}.$$

Using your 1D basis $\phi_k[\cdot]$, implement a function to generate $\Phi_{u,v}$ for any (u, v).

(b) Visualize the full 8×8 set

Make a figure that shows the entire set of 64 basis patterns $\{\Phi_{u,v}\}$ arranged as an 8×8 grid (rows indexed by u, columns by v).

(c) 2D DCT on image blocks

Answer the following:

i) Perform a 2D DCT on an 8×8 image block by projecting onto the 2D basis functions. You can crop any grayscale image to 8×8 for this purpose.

ii) Reconstruct the image block from its DCT coefficients. Verify perfect reconstruction.

iii) Comment on the distribution of energy in the DCT coefficients for typical image blocks.

3 Part 3: Implement JPEG-style compression and investigate energy compaction]

In this part, you will implement a simplified JPEG-like pipeline:

image $\rightarrow$ (8×8 blocks) $\rightarrow$ 2D DCT $\rightarrow$ quantization $\rightarrow$ inverse DCT $\rightarrow$ reconstructed image.

You will also investigate the **energy compaction** property of the DCT.

(a) Block DCT and reconstruction (no quantization)

a) Divide the image into 8×8 blocks.

b) For each block, compute the 2D DCT coefficients.

c) Reconstruct the image using inverse 2D DCT *without* quantization.

d) Report the maximum absolute reconstruction error (should be near numerical precision).

(b) Quantization (JPEG idea)

Implement quantization per 8×8 block:

$$\hat{x}_Q[u,v] = \text{round}\left(\frac{\hat{x}[u,v]}{Q[u,v]}\right), \qquad \widetilde{x}[u,v] = \hat{x}_Q[u,v] \cdot Q[u,v].$$

Use a parametric quantizer $Q[u,v] = 1 + s(u+v)$ for a scalar $s > 0$ (you pick s values to sweep). Then reconstruct the image using $\widetilde{x}$ and inverse DCT.

(d) Quality vs compression study

For at least 5 different compression strengths, do the following:

a) Show reconstructed images

b) Compute and report PSNR and the sparsity (fraction of zero coefficients after quantization)

c) Plot PSNR and zero-fraction vs. compression strength.

(e) Energy compaction (core concept)

For each of the blocks, compute:

$$E_{\text{total}} = \sum_{u,v} |\hat{x}[u,v]|^2, \qquad E_K = \sum_{(u,v) \in \mathcal{S}_K} |\hat{x}[u,v]|^2$$

where $\mathcal{S}_K$ contains the top-left $K \times K$ coefficients (low-frequency square).

a) Plot the average energy fraction E_K / E_{total} vs. K for $K = 1, \ldots, 8$.

b) Interpret the plot: how quickly does energy accumulate as K increases?

c) Repeat the same experiment for a **high-frequency** square (e.g., bottom-right $K \times K$) and compare.