

Image Processing

Image Processing Fundamentals: Edge Detection

Original Image



Edge Detected Image



Basic Image Processing

- Image = 2D matrix of pixel values
- Each pixel represents brightness/color
- Example grayscale image:

$$X = \begin{bmatrix} 10 & 10 & 80 & 10 & 10 & 10 \\ 10 & 10 & 80 & 10 & 10 & 10 \\ 10 & 10 & 80 & 10 & 10 & 10 \\ 10 & 10 & 80 & 10 & 10 & 10 \\ 10 & 10 & 80 & 10 & 10 & 10 \\ 10 & 10 & 80 & 10 & 10 & 10 \end{bmatrix}$$

57	153	174	168	150	152	129	151	172	161	155	156
55	182	163	74	75	62	83	17	110	210	180	154
80	180	50	14	34	6	10	83	48	106	159	181
106	109	5	124	131	111	120	204	166	15	56	180
94	68	137	251	237	239	239	228	227	87	71	201
72	106	207	253	233	214	220	239	228	98	74	206
88	68	179	209	185	215	211	158	139	75	20	160
89	97	165	84	10	168	134	11	31	62	22	148
99	168	191	193	158	227	178	143	182	106	36	190
106	174	155	252	236	231	149	178	228	43	95	234
90	216	116	149	236	187	85	150	79	38	218	241
90	224	147	108	227	210	127	102	36	101	255	224
90	214	173	66	103	143	96	50	2	109	249	215
87	196	235	75	1	81	47	0	6	217	255	211
83	202	237	145	0	0	12	108	209	138	243	238
96	206	123	207	177	121	123	209	175	13	96	218

Convolution: Spatial Domain

- Slide kernel over image
- Multiply and sum values
- Example kernel (vertical edge detection):
- [Demo](#)

$$K = \begin{bmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{bmatrix}$$

Convolution is Complicated 🤪

Example:

Suppose we have an image X and a kernel K as follows:

$$X = [X_1 \quad X_2 \quad X_3 \quad X_4 \quad X_5 \quad X_6]$$
$$K = [K_1 \quad K_2 \quad K_3]$$

The convolution is given by

$$X * K = \sum_{i=1}^6 X_i K_{7-i}$$

Or equivalently,

$$X * K = [X_1 K_3 + X_2 K_2 + X_3 K_1 \quad X_2 K_3 + X_3 K_2 + X_4 K_1 \quad X_3 K_3 + X_4 K_2 + X_5 K_1 \quad X_4 K_3 + X_5 K_2 + X_6 K_1]$$

Let's make it simpler using the convolution theorem!

What is the convolution theorem?

Suppose two functions f and g and their Fourier transforms F and G . Then,


$$\underbrace{(f * g)}_{\text{convolution}} \leftrightarrow \underbrace{(F \cdot G)}_{\text{multiplication}}$$

The Fourier transform is a one-to-one mapping between f and F (and g and G).

But what is the Fourier transform 🤔?

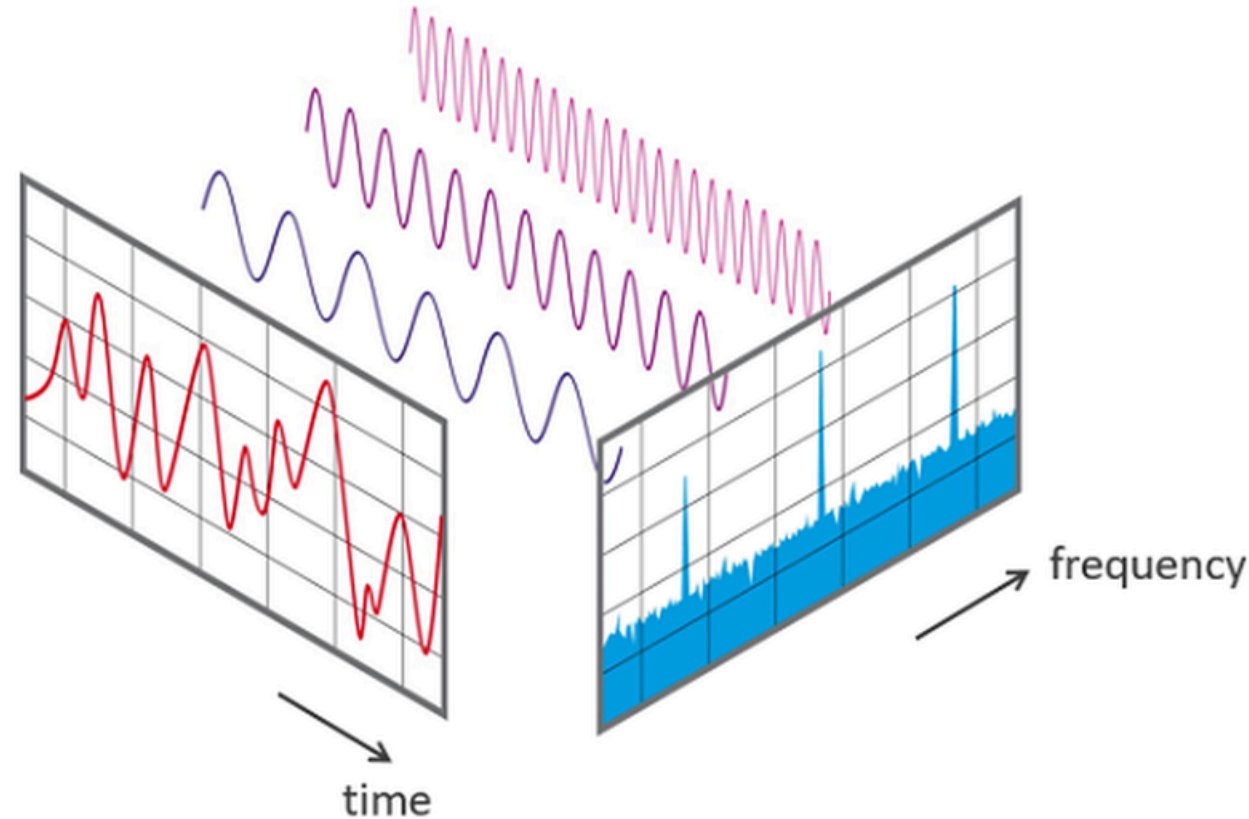
Fourier Transform: The Basics

Transform a signal from:

- Time/Space domain  Frequency domain

Key Concept:

- Any signal can be decomposed into sum of sine and cosine waves
- Each wave has specific frequency and amplitude

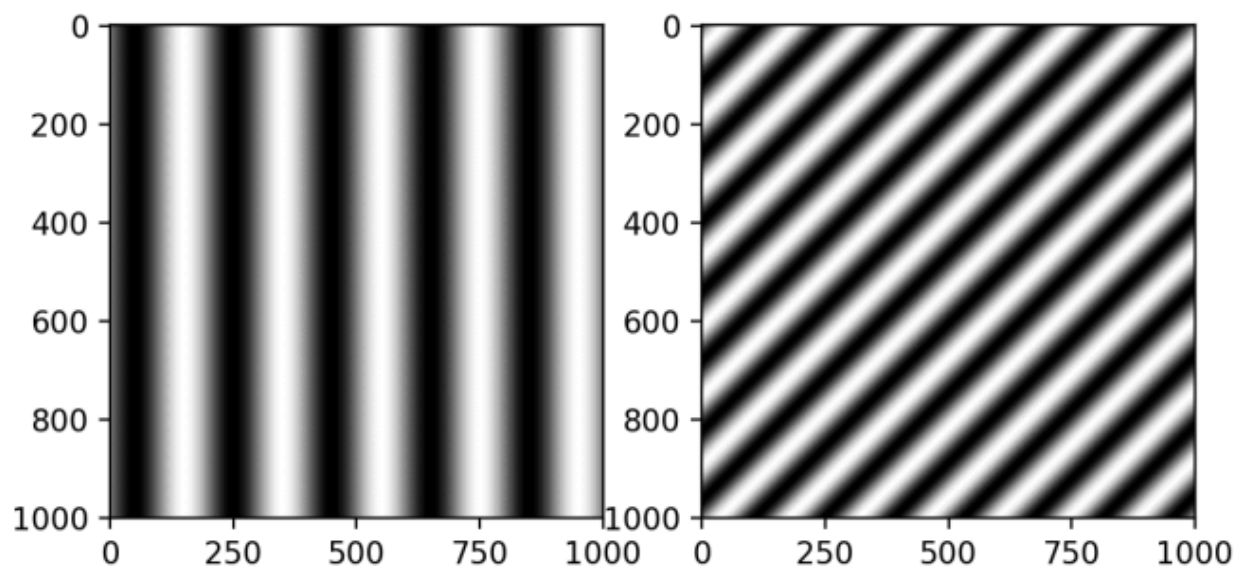


2D Fourier Transform

2D Fourier Transform decomposes image into sum of $2D$ waves.

$$\mathcal{F}(X)[h, w] = \sum_{k=0}^{H-1} \sum_{\ell=0}^{W-1} X[k, \ell] \cdot \underbrace{e^{-2\pi i \left(\frac{hk}{H} + \frac{w\ell}{W} \right)}}_{2D \text{ wave}}$$

For image X with size $H \times W$.



Edge Detection in Frequency Domain

Original Image ➡ Fourier Transform ➡ Apply Filter ➡ Inverse Transform

```
# Convert to frequency domain
FX = np.fft.fft2(img_gray) # Image
FK = np.fft.fft2(kernel_padded) # Kernel

# Multiply
filtered = FX * FK

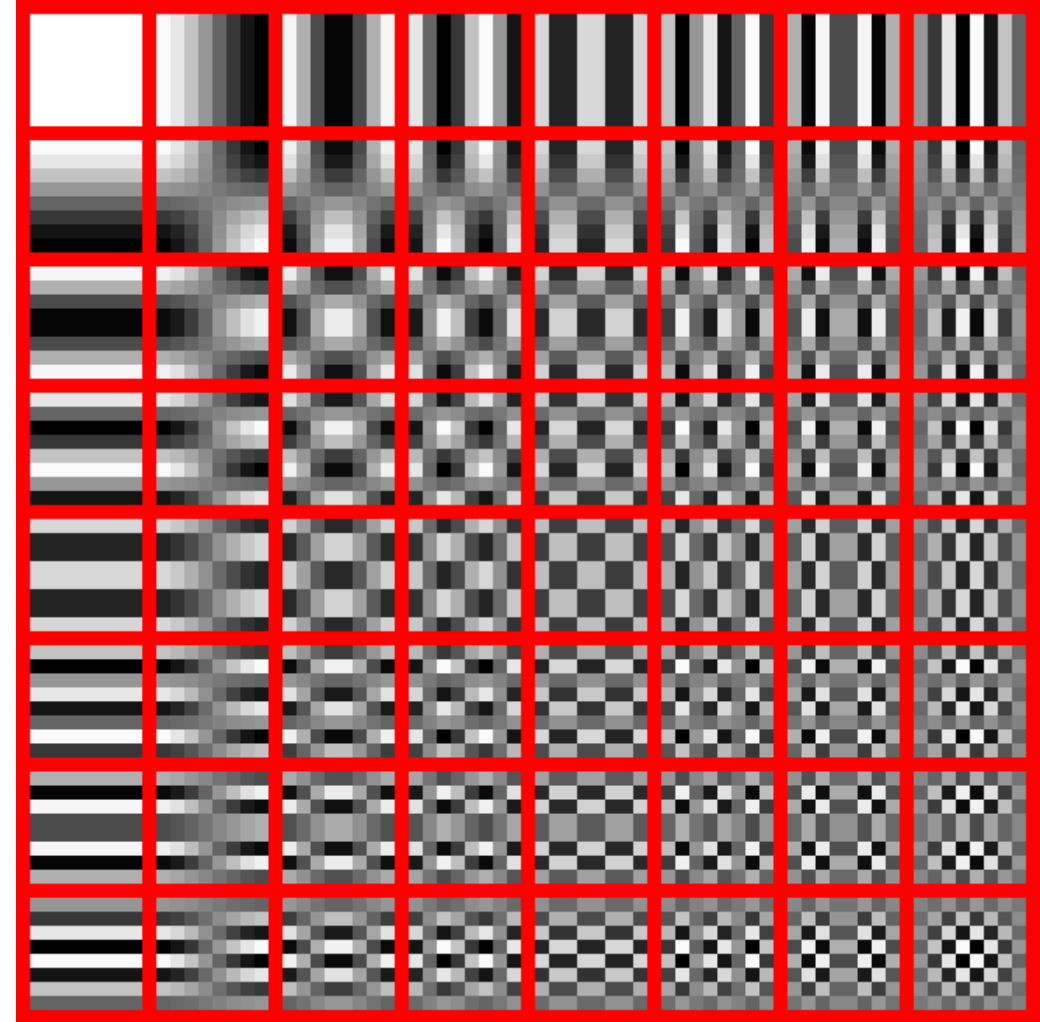
# Convert back
result = np.real(np.fft.ifft2(filtered))
```

JPEG Compression 📷

1. Divide image into 8x8 blocks
2. Apply Discrete Cosine Transform (similar to Fourier)
3. Quantize frequencies
 - Keep low frequencies
 - Discard high frequencies
4. Encode efficiently

Benefits:

- Smaller file size
- Maintains visual quality
- Exploits human visual perception



Coding Exercise

Coding Exercise