# LABSHEET 4

# IMAGE TRANSFORMATIONS

---

---

**Discrete Cosine Transformation of Images using Python**

With the help of **scipy.fft.dct()** method, we can compute the discrete cosine transform by selecting different types of sequences and return the transformed array by using this method.

**scipy.fft.dct(*x, type=2, n=None, axis=-1, norm=None, overwrite_x=False, workers=None, orthogonalize=None*)**

**x*array_like***

The input array.

**type*{1, 2, 3, 4}, optional***

Type of the DCT . Default type is 2.

**n*int, optional***

Length of the transform. If n < x.shape[axis], *x* is truncated. If n > x.shape[axis], *x* is zero-padded. The default results n = x.shape[axis].

**axis*int, optional***

Axis along which the dct is computed; the default is over the last axis (i.e., axis=-1).

**norm*{"backward", "ortho", "forward"}, optional***

Normalization mode (see Notes). Default is "backward".

**overwrite_x*bool, optional***

If True, the contents of *x* can be destroyed; the default is False.

**workers*int, optional***

Maximum number of workers to use for parallel computation. If negative, the value wraps around from os.cpu_count(). See **fft** for more details.

**orthogonalize*bool, optional***

Whether to use the orthogonalized DCT variant (see Notes). Defaults to True when norm="ortho" and False otherwise.

**QN 1:** Read a gray scale image and apply 2D DCT to the image. Plot the result.

**QN 2:** Implement 2D DCT without in-built functions. Compare the result with that of QN 1.

---

**Wavelet Transform Analysis of Images using Python**

PyWavelets is open source wavelet transform software for Python. Its dependencies are numpy, scipy and matplotlib.

For more information : https://pywavelets.readthedocs.io/en/latest/

**Explaining dwt2() and idwt2():** Single level wavelet decomposition and reconstruction of 2D signals.

C = pywt.dwt2(X, 'Wname')

x = pywt.dwt2(C, 'Wname')

where,

x: Input Image

C: Output [Wavelet coefficients in the form of tuple (cA, (cH, cV, cD))

Wname : name of the wavelet used

**Sample Code**

```
import numpy as np
import matplotlib.pyplot as plt
import pywt
import pywt.data
# Load image
---------------------------- #read the image using imread
# Wavelet transform of image, and plot approximation and details
titles = ['Approximation', ' Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = pywt.dwt2(original, 'bior1.3')
LL, (LH, HL, HH) = coeffs2
fig = plt.figure(figsize=(12, 3))
for i, a in enumerate([LL, LH, HL, HH]):
    ax = fig.add_subplot(1, 4, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=plt.cm.gray)
    ax.set_title(titles[i], fontsize=10)
    ax.set_xticks([])
    ax.set_yticks([])

fig.tight_layout()
plt.show()
```

Answer the following questions:

1) Which is the approximation and detail subbands of the image? What do they represent?
2) How many sub-bands is the image getting divided into?
3) What are the dimensions of these subbands?
4) If you want to apply wavelet transformation in level 1, which subband is taken as input.
5) Perform level 1 and level 2 wavelet decomposition. Answer Qn 1 to 4 for the input image you considered here.