

Lab05 - Image Processing and Analysis

19110522 – Bùi Thị Thanh Xuân

```
Entrée [1]: import numpy as np
import pandas as pd
import cv2
from matplotlib import pyplot as plt
import matplotlib.cm as cm
import matplotlib.gridspec as gridspec
from pylab import imread
from skimage.color import rgb2gray

Entrée [2]: def ShowImage(ImageList, nRows = 1, nCols = 2, WidthSpace = 0.00, HeightSpace = 0.00):
    from matplotlib import pyplot as plt
    import matplotlib.gridspec as gridspec

    gs = gridspec.GridSpec(nRows, nCols)
    gs.update(wspace=WidthSpace, hspace=HeightSpace) # set the spacing between axes.
    plt.figure(figsize=(20,10))
    for i in range(len(ImageList)):
        ax1 = plt.subplot(gs[i])
        ax1.set_xticklabels([])
        ax1.set_yticklabels([])
        ax1.set_aspect('equal')

        plt.subplot(nRows, nCols, i+1)

        image = ImageList[i].copy()
        if (len(image.shape) < 3):
            plt.imshow(image, plt.cm.gray)
        else:
            plt.imshow(image)
        plt.title("Image " + str(i))
        plt.axis('off')

    plt.show()
```

1/ Kiểm một ảnh trên internet và thực hiện tất cả các biến đổi wavelet 'db5', 'sym5', 'coif5', 'bior1.3', 'haar' ở mức 1 và 2 (two level decomposition)

```
Entrée [3]: import pywt
import pywt.data
from pywt import dwt2, idwt2

Entrée [4]: # Read Image
image_color = imread("church.jpg")
# Convert Image into Gray
image_gray = cv2.cvtColor(image_color, cv2.COLOR_RGB2GRAY)

# Display Image
ShowImage([image_color, image_gray], 1, 2)
```



db5

Entrée [7]: `from pywt._doc_utils import wavedec2_keys, draw_2d_wp_basis`

```
x = image_gray.astype(np.float32)
shape = x.shape

max_leve = 3 # how many levels of decomposition to draw
label_levels = 3 # how many levels to explicitly label on the plots

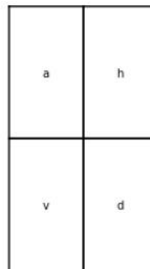
fig, axes = plt.subplots(2, 3, figsize=[14, 8])
for level in range(0, max_leve):
    if level == 0:
        # show the original image before decomposition
        axes[0, 0].set_axis_off()
        axes[1, 0].imshow(x, cmap=plt.cm.gray)
        axes[1, 0].set_title('Image')
        axes[1, 0].set_axis_off()
        continue

    # plot subband boundaries of a standard DWT basis
    draw_2d_wp_basis(shape, wavedec2_keys(level), ax=axes[0, level],
                     label_levels=label_levels)
    axes[0, level].set_title('{} level\ndecomposition'.format(level))

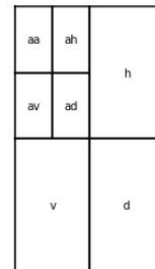
    # compute the 2D DWT
    c = pywt.wavedec2(x, 'db5', mode='periodization', level=level)
    # normalize each coefficient array independently for better visibility
    c[0] /= np.abs(c[0]).max()
    for detail_level in range(level):
        c[detail_level + 1] = [d/np.abs(d).max() for d in c[detail_level + 1]]
    # show the normalized coefficients
    arr, slices = pywt.coeffs_to_array(c)
    axes[1, level].imshow(arr, cmap=plt.cm.gray)
    axes[1, level].set_title('Coefficients\n({} level)'.format(level))
    axes[1, level].set_axis_off()

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

1 level
decomposition



2 level
decomposition



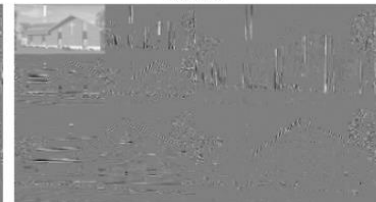
Image



Coefficients
(1 level)



Coefficients
(2 level)



sym5

```
Entrée [8]: x = image_gray.astype(np.float32)
            shape = x.shape

max_level = 3      # how many levels of decomposition to draw
label_levels = 3    # how many levels to explicitly label on the plots

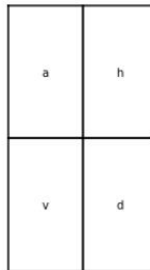
fig, axes = plt.subplots(2, 3, figsize=[14, 8])
for level in range(0, max_level):
    if level == 0:
        # show the original image before decomposition
        axes[0, 0].set_axis_off()
        axes[1, 0].imshow(x, cmap=plt.cm.gray)
        axes[1, 0].set_title('Image')
        axes[1, 0].set_axis_off()
        continue

    # plot subband boundaries of a standard DWT basis
    draw_2d_wp_basis(shape, wavedec2_keys(level), ax=axes[0, level],
                     label_levels=label_levels)
    axes[0, level].set_title('{} level\ndecomposition'.format(level))

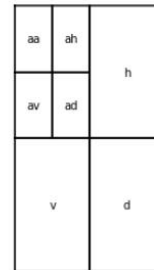
    # compute the 2D DWT
    c = pywt.wavedec2(x, 'sym5', mode='periodization', level=level)
    # normalize each coefficient array independently for better visibility
    c[0] /= np.abs(c[0]).max()
    for detail_level in range(level):
        c[detail_level + 1] = [d/np.abs(d).max() for d in c[detail_level + 1]]
    # show the normalized coefficients
    arr, slices = pywt.coeffs_to_array(c)
    axes[1, level].imshow(arr, cmap=plt.cm.gray)
    axes[1, level].set_title('Coefficients\n({} level)'.format(level))
    axes[1, level].set_axis_off()

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

1 level
decomposition



2 level
decomposition



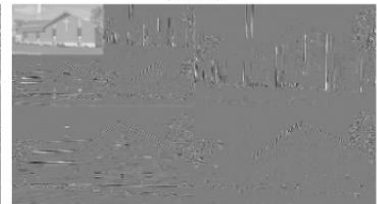
Image



Coefficients
(1 level)



Coefficients
(2 level)



coif5

```
Entrée [9]: x = image_gray.astype(np.float32)
shape = x.shape

max Lev = 3      # how many levels of decomposition to draw
label_levels = 3 # how many levels to explicitly label on the plots

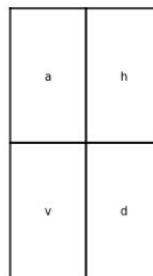
fig, axes = plt.subplots(2, 3, figsize=[14, 8])
for level in range(0, max_lev):
    if level == 0:
        # show the original image before decomposition
        axes[0, 0].set_axis_off()
        axes[1, 0].imshow(x, cmap=plt.cm.gray)
        axes[1, 0].set_title('Image')
        axes[1, 0].set_axis_off()
        continue

    # plot subband boundaries of a standard DWT basis
    draw_2d_wp_basis(shape, wavedec2_keys(level), ax=axes[0, level],
                    label_levels=label_levels)
    axes[0, level].set_title('{} level\ndecomposition'.format(level))

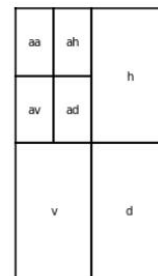
    # compute the 2D DWT
    c = pywt.wavedec2(x, 'coif5', mode='periodization', level=level)
    # normalize each coefficient array independently for better visibility
    c[0] /= np.abs(c[0]).max()
    for detail_level in range(level):
        c[detail_level + 1] = [d/np.abs(d).max() for d in c[detail_level + 1]]
    # show the normalized coefficients
    arr, slices = pywt.coeffs_to_array(c)
    axes[1, level].imshow(arr, cmap=plt.cm.gray)
    axes[1, level].set_title('Coefficients\n({} level)'.format(level))
    axes[1, level].set_axis_off()

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

1 level
decomposition



2 level
decomposition



Image



Coefficients
(1 level)



Coefficients
(2 level)



bior1.3

```
Entrée [10]: x = image_gray.astype(np.float32)
shape = x.shape

max_level = 3      # how many levels of decomposition to draw
label_levels = 3    # how many levels to explicitly label on the plots

fig, axes = plt.subplots(2, 3, figsize=[14, 8])
for level in range(0, max_level):
    if level == 0:
        # show the original image before decomposition
        axes[0, 0].set_axis_off()
        axes[1, 0].imshow(x, cmap=plt.cm.gray)
        axes[1, 0].set_title('Image')
        axes[1, 0].set_axis_off()
        continue

    # plot subband boundaries of a standard DWT basis
    draw_2d_wp_basis(shape, wavedec2_keys(level), ax=axes[0, level],
                     label_levels=label_levels)
    axes[0, level].set_title('{} level\ndecomposition'.format(level))

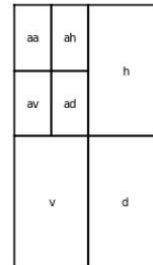
    # compute the 2D DWT
    c = pywt.wavedec2(x, 'bior1.3', mode='periodization', level=level)
    # normalize each coefficient array independently for better visibility
    c[0] /= np.abs(c[0]).max()
    for detail_level in range(level):
        c[detail_level + 1] = [d/np.abs(d).max() for d in c[detail_level + 1]]
    # show the normalized coefficients
    arr, slices = pywt.coeffs_to_array(c)
    axes[1, level].imshow(arr, cmap=plt.cm.gray)
    axes[1, level].set_title('Coefficients\n({} level)'.format(level))
    axes[1, level].set_axis_off()

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

1 level
decomposition



2 level
decomposition



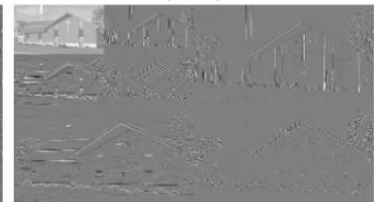
Image



Coefficients
(1 level)



Coefficients
(2 level)



haar

```
Entrée [11]: x = image_gray.astype(np.float32)
             shape = x.shape

max_level = 3 # how many levels of decomposition to draw
label_levels = 3 # how many levels to explicitly label on the plots

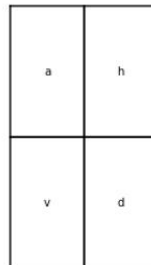
fig, axes = plt.subplots(2, 3, figsize=[14, 8])
for level in range(0, max_level):
    if level == 0:
        # show the original image before decomposition
        axes[0, 0].set_axis_off()
        axes[1, 0].imshow(x, cmap=plt.cm.gray)
        axes[1, 0].set_title('Image')
        axes[1, 0].set_axis_off()
        continue

    # plot subband boundaries of a standard DWT basis
    draw_2d_wp_basis(shape, wavedec2_keys(level), ax=axes[0, level],
                     label_levels=label_levels)
    axes[0, level].set_title('{} level\ndecomposition'.format(level))

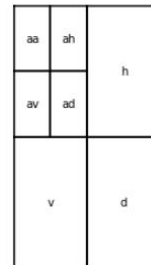
    # compute the 2D DWT
    c = pywt.wavedec2(x, 'bior1.3', level=level)
    # normalize each coefficient array independently for better visibility
    c[0] /= np.abs(c[0]).max()
    for detail_level in range(level):
        c[detail_level + 1] = [d/np.abs(d).max() for d in c[detail_level + 1]]
    # show the normalized coefficients
    arr, slices = pywt.coeffs_to_array(c)
    axes[1, level].imshow(arr, cmap=plt.cm.gray)
    axes[1, level].set_title('Coefficients\n({} level)'.format(level))
    axes[1, level].set_axis_off()

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

1 level
decomposition



2 level
decomposition



Image



Coefficients
(1 level)



Coefficients
(2 level)



2/ **Viết một hàm trong đó truyền các tham số là hình ảnh cần biến đổi, function wavelet dùng ('db5', 'sym5', 'coif5', 'bior1.3', 'haar') và level biến đổi. Giá trị trả về là ảnh xấp xỉ và các ảnh detail. Kiểm tra kết quả với một bức ảnh kiểm từ internet**

```

:entrée [15]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
              continuous_wavelets = ['mexh', 'morl', 'cgau5', 'gaus5']

              list_list_wavelets = [discrete_wavelets, continuous_wavelets]
              list_funcs = [pywt.Wavelet, pywt.ContinuousWavelet]

              fig, axarr = plt.subplots(nrows=2, ncols=5, figsize=(16,8))
              for ii, list_wavelets in enumerate(list_list_wavelets):
                  func = list_funcs[ii]
                  row_no = ii
                  for col_no, waveletname in enumerate(list_wavelets):
                      wavelet = func(waveletname)
                      family_name = wavelet.family_name
                      biorthogonal = wavelet.biorthogonal
                      orthogonal = wavelet.orthogonal
                      symmetry = wavelet.symmetry
                      if ii == 0:
                          _ = wavelet.wavefun()
                          wavelet_function = _[0]
                          x_values = _[-1]
                      else:
                          wavelet_function, x_values = wavelet.wavefun()
                      if col_no == 0 and ii == 0:
                          axarr[row_no, col_no].set_ylabel("Discrete Wavelets", fontsize=16)
                      if col_no == 0 and ii == 1:
                          axarr[row_no, col_no].set_ylabel("Continuous Wavelets", fontsize=16)
                      axarr[row_no, col_no].set_title("{}".format(family_name), fontsize=16)
                      axarr[row_no, col_no].plot(x_values, wavelet_function)
                      axarr[row_no, col_no].set_yticks([])
                      axarr[row_no, col_no].set_yticklabels([])

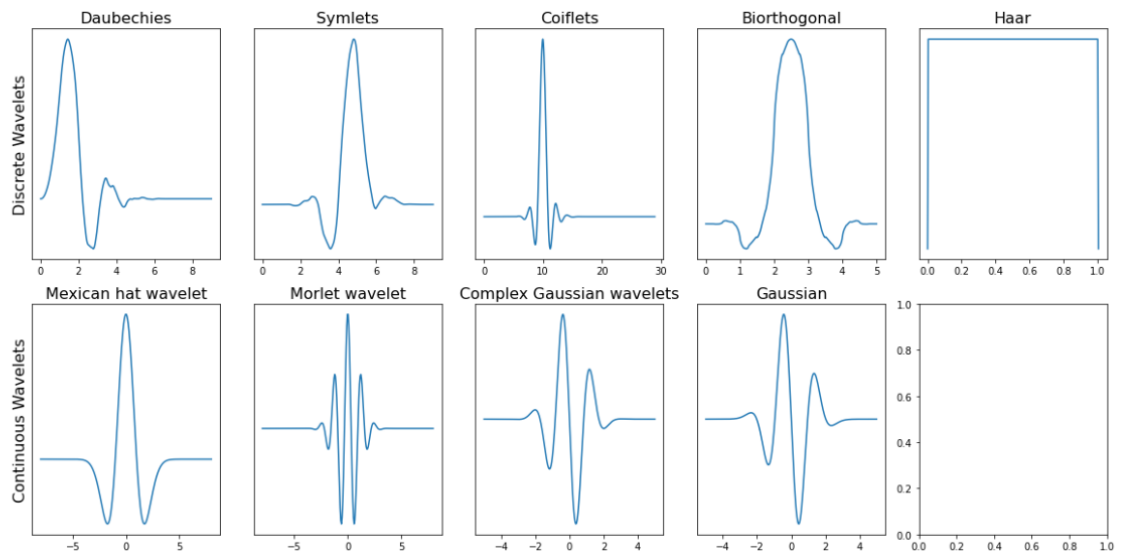
```

```

plt.tight_layout()
plt.show()

```

C:\Users\ADMIN\anaconda3\lib\site-packages\numpy\core_asarray.py:102: ComplexWarning: Casting complex values to real discards the imaginary part
return array(a, dtype, copy=False, order=order)



```
Entrée [13]: # Read Image
image_color = imread("Sample2.jpg")
# Convert Image into Gray
image_gray = cv2.cvtColor(image_color, cv2.COLOR_RGB2GRAY)

# Display Image
ShowImage([image_color, image_gray], 1, 2)
```

Image 0



Image 1



```
Entrée [13]: import pywt
import pywt.data
from pywt import dwt2, idwt2

# Wavelet transform of image, and plot approximation and details
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[3])
LL, (LH, HL, HH) = coeffs2

# Display Image
ShowImage([LL, LH, HL, HH], 2, 2)
```

Image 0



Image 1



Image 2



Image 3




```
Entrée [14]: cmaplist = [cm.gray, cm.jet, cm.rainbow, cm.viridis, cm.cubehelix, cm.RdBu]
fig = plt.figure(figsize=(20,10))
for i, a in enumerate([LL, LH, HL, HH]):
    ax = fig.add_subplot(2, 2, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=cmaplist[5])
    ax.set_title(titles[i], fontsize=10)
    ax.set_xticks([])
    ax.set_yticks([])

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

Approximation



Horizontal detail



Vertical detail



Diagonal detail



3/ Kiểm một ảnh từ internet. Thực hiện biến đổi wavelet để phân rã thành ảnh xấp xỉ và ảnh detail. Sau đó tiến hành các bước sau

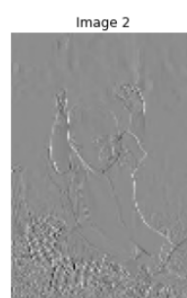
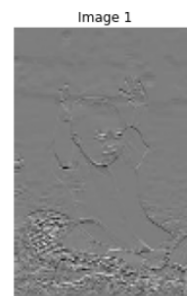
```
Entrée [21]: # Read Image
image_color = imread("sam3.jpg")
# Convert Image into Gray
image_gray = cv2.cvtColor(image_color, cv2.COLOR_RGB2GRAY)

# Display Image
ShowImage([image_color, image_gray], 1, 2)
```



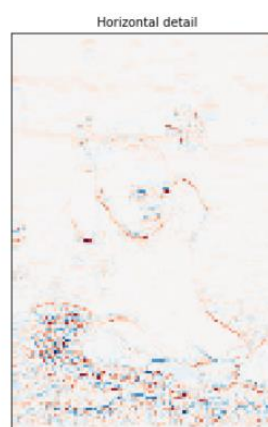
```
Entrée [19]: # Wavelet transform of image, and plot approximation and details
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[3])
LL, (LH, HL, HH) = coeffs2

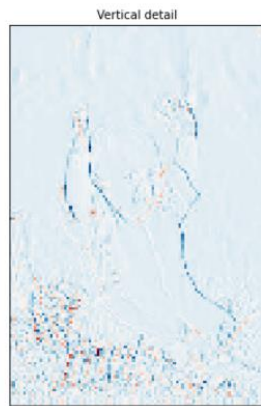
# Display Image
ShowImage([LL, LH, HL, HH], 2, 2)
```



```
Entrée [20]: cmaplist = [cm.gray, cm.jet, cm.rainbow, cm.viridis, cm.cubehelix, cm.RdBu]
fig = plt.figure(figsize=(20,10))
for i, a in enumerate([LL, LH, HL, HH]):
    ax = fig.add_subplot(2, 2, i + 1)
    ax.imshow(a, interpolation="nearest", cmap=cmaplist[5])
    ax.set_title(titles[i], fontsize=10)
    ax.set_xticks([])
    ax.set_yticks([])

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





Khôi phục ảnh gốc từ ảnh detail và ảnh approximation

Db5

Entrée [18]:

```
discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[0])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```



sym5

Entrée [19]:

```
discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[1])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```



coif5

```
Entrée [20]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[2])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```

Image 0



bior1.3

```
Entrée [21]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[3])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```

Image 0



haar

```
Entrée [22]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[4])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```

Image 0



Xóa thông tin ảnh horizontal detail và khôi phục lại ảnh từ ảnh approximation và các ảnh detail còn lại

db5

```
Entrée [24]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[0])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```

Image 0



sym5

```
Entrée [25]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[1])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```

Image 0



coif5

```
Entrée [26]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[2])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```

Image 0



bior1.3

```
Entrée [27]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[3])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```

Image 0



haar

```
Entrée [28]: discrete_wavelets = ['db5', 'sym5', 'coif5', 'bior1.3', 'haar']
titles = ['Approximation', 'Horizontal detail',
          'Vertical detail', 'Diagonal detail']
coeffs2 = dwt2(image_gray, discrete_wavelets[4])
LL, (LH, HL, HH) = coeffs2
# Display Image
ShowImage([LL+LH+HL], 2, 2)
```

Image 0



Chưa làm được:

Xóa thông tin ảnh vertical detail và khôi phục lại ảnh từ ảnh approximation và các ảnh detail còn lại

Entrée []:

Xóa thông tin ảnh diagonal detail và khôi phục lại ảnh từ ảnh approximation và các ảnh detail còn lại

Entrée []: