AMRITA
VISHWA VIDYAPEETHAM
श्रद्धावान् लभते ज्ञानम्
DEEMED TO BE UNIVERSITY UNDER SECTION 3 OF UGC ACT, 1956

19ECE457 Wavelets and Applications
Project:Image Fusion

TEAM 2

Amirdhavarshini T	CB.EN.U4CCE21005
K Pooja	CB.EN.U4CCE21027
Supraja Gomathi K	CB.EN.U4ECE21153

Introduction and tools used

- In image fusion, wavelet transform is used to extract high-frequency details (edges, textures) and low-frequency components (overall structure) from input images.
- The goal is to combine these components from multiple images into one, preserving the most significant details.
- Tools/Libraries used:
  - Tools:Jupyter Notebook

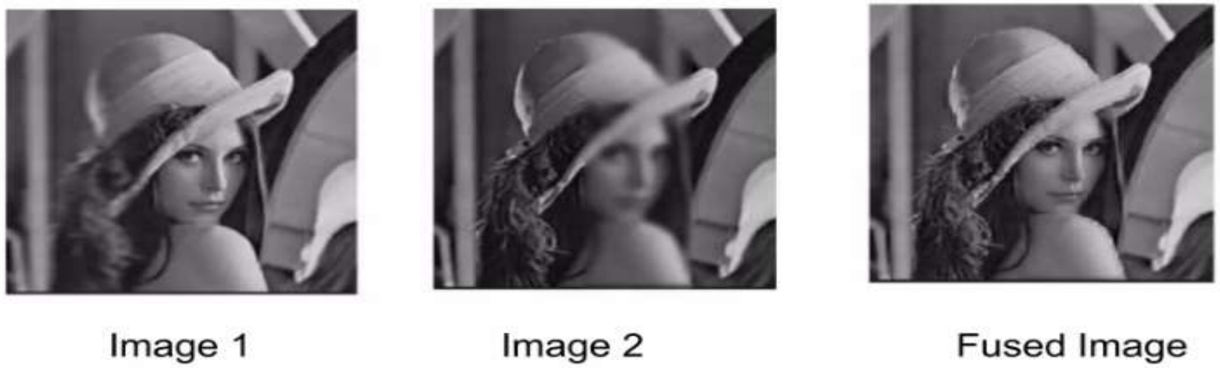


Image 1

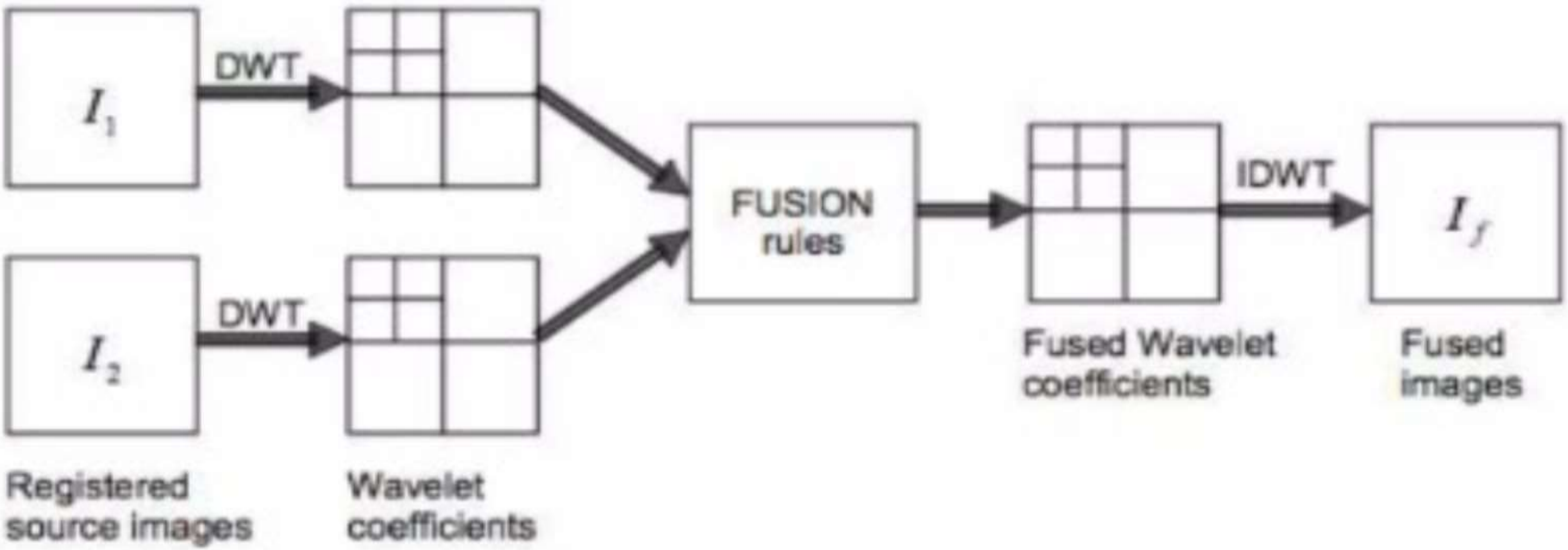
Image 2

Fused Image

Image 1 is blurred in the left and Image 2 in the middle. The fused image is of better quality than both the input images. Image is decomposed using Biorthogonal Wavelets at for levels. The Approximation image merged by taking maximum and detailed structures by taking minimum.

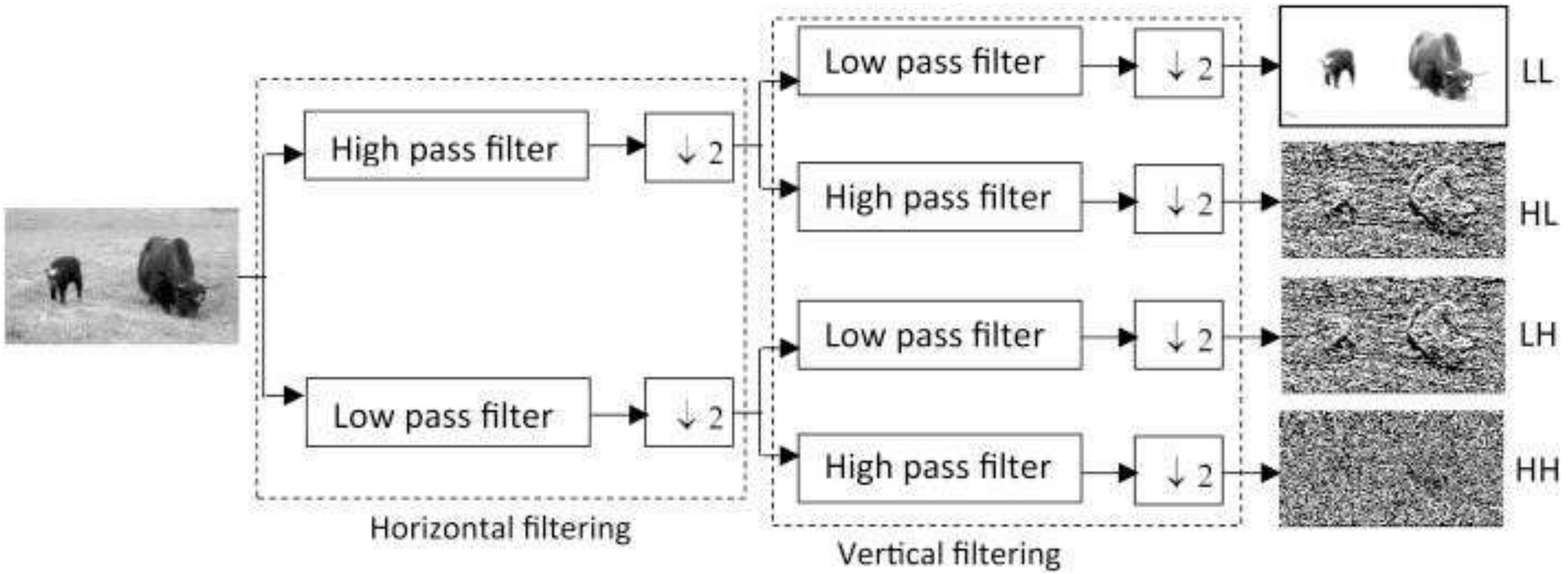
Methodology

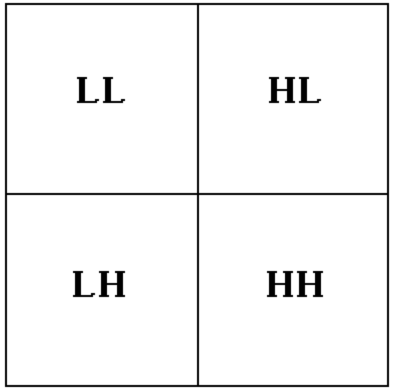
1) Get the images to be fused**(OpenCV: To read the images.)**
2) Apply the wavelet transform on both the images through chosen wavelet at the desired level**(PyWavelets: To perform the wavelet transform)**
3) Get the approximation and detail coefficients for both the images**(PyWavelets: The wavedec function returns the approximation and detail coefficients.)**
4) Merge the coefficients by desired fusion rule**(NumPy: To perform operations on the coefficients.)**
5) Apply Inverse discrete wavelet transform on the merged coefficients and get the fused image**(PyWavelets: To perform the inverse wavelet transform)**



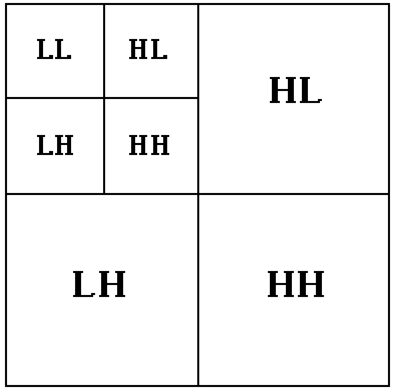
I_1
DWT
I_2
DWT
FUSION rules
IDWT
I_f
Registered source images
Wavelet coefficients
Fused Wavelet coefficients
Fused images

DECOMPOSITION OF IMAGE

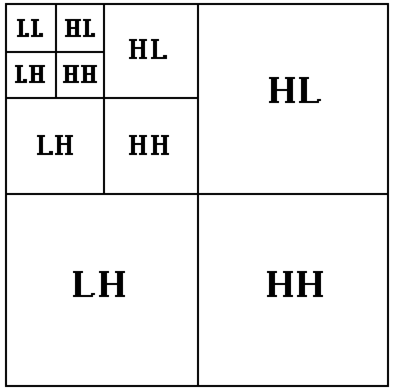




(a) Single Level Decomposition

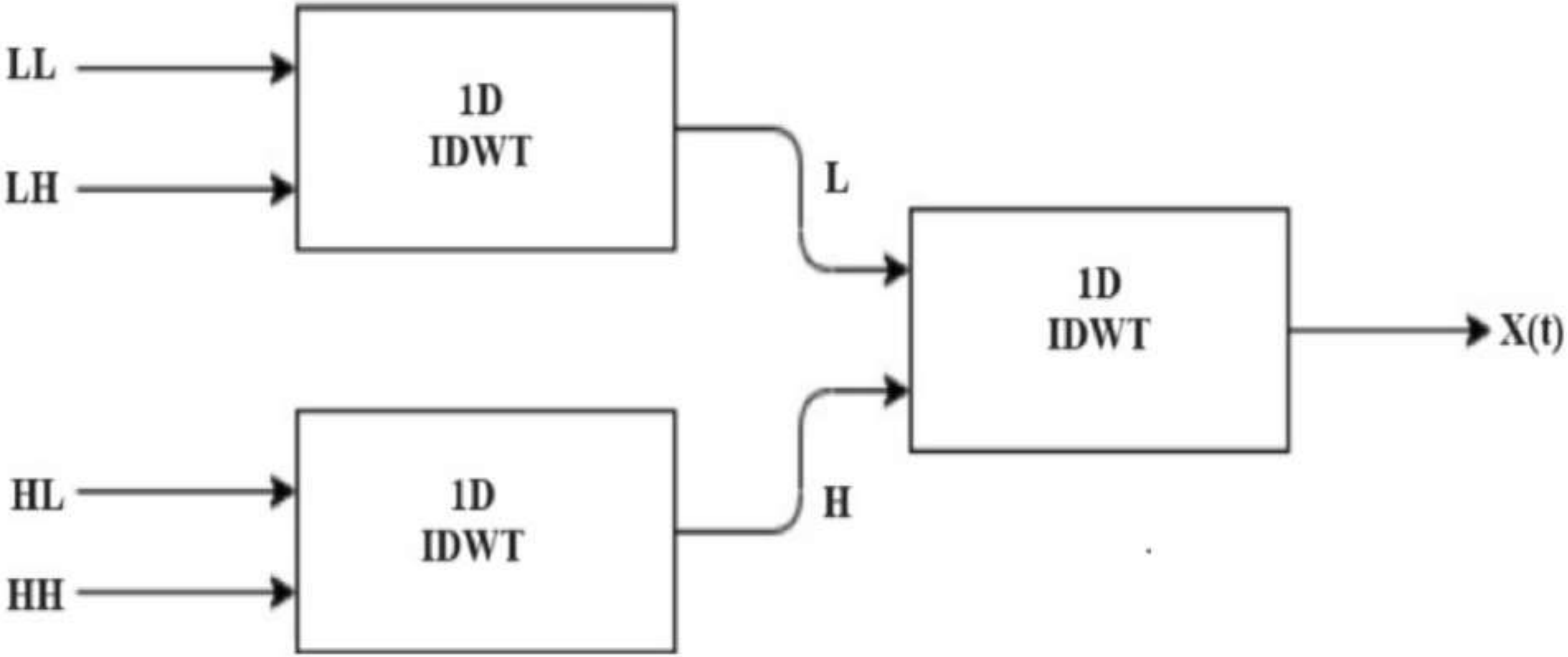


(b) Two Level Decomposition

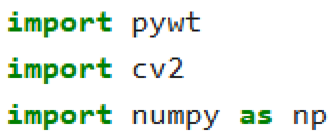


(c) Three Level Decomposition

RECONSTRUCTION OF IMAGE

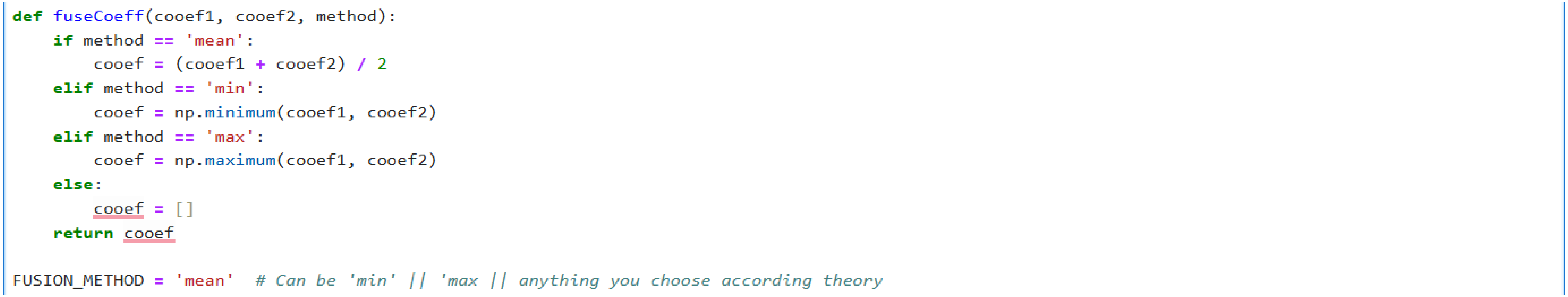


CODE IMPLEMENTATION



```python
import pywt
import cv2
import numpy as np
```

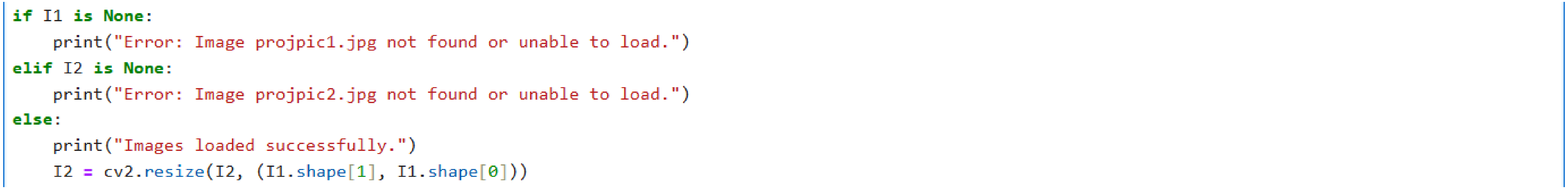
Imports the required libraries



```python
def fuseCoeff(cooef1, cooef2, method):
    if method == 'mean':
        cooef = (cooef1 + cooef2) / 2
    elif method == 'min':
        cooef = np.minimum(cooef1, cooef2)
    elif method == 'max':
        cooef = np.maximum(cooef1, cooef2)
    else:
        cooef = []
    return cooef

FUSION_METHOD = 'mean'  # Can be 'min' || 'max || anything you choose according theory
```

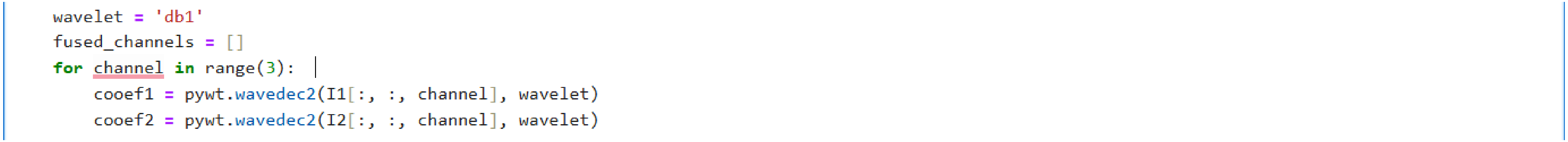
Fuses two wavelet coefficients based on the specified fusion method (mean, min, or max).



```python
if I1 is None:
    print("Error: Image projpic1.jpg not found or unable to load.")
elif I2 is None:
    print("Error: Image projpic2.jpg not found or unable to load.")
else:
    print("Images loaded successfully.")
    I2 = cv2.resize(I2, (I1.shape[1], I1.shape[0]))
```

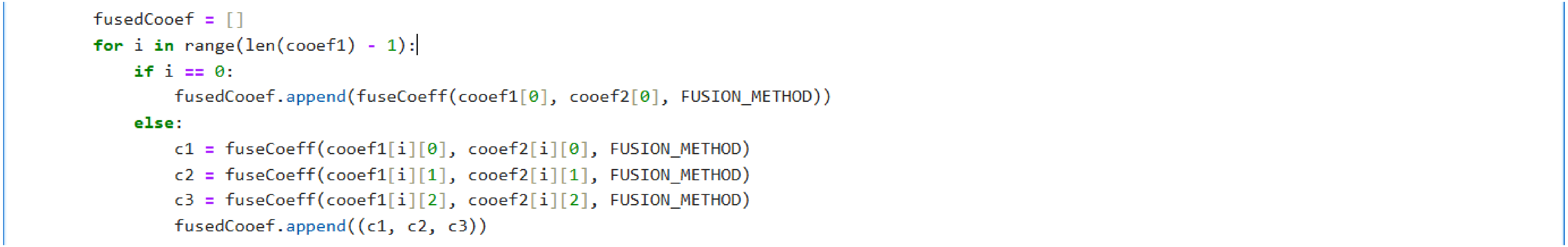
Resizes the second image to match the dimensions of the first image.

Contd..



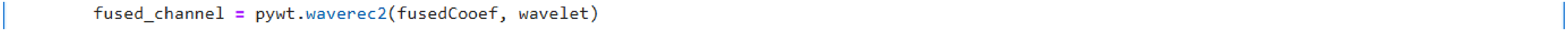
```python
wavelet = 'db1'
fused_channels = []
for channel in range(3):
    cooef1 = pywt.wavedec2(I1[:, :, channel], wavelet)
    cooef2 = pywt.wavedec2(I2[:, :, channel], wavelet)
```

Fuses the approximation coefficients from the wavelet decomposition of both images' color channels.



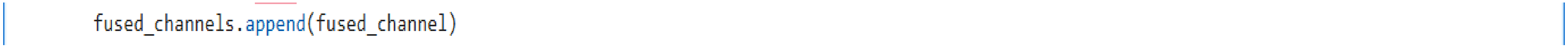
```python
fusedCooef = []
for i in range(len(cooef1) - 1):
    if i == 0:
        fusedCooef.append(fuseCoeff(cooef1[0], cooef2[0], FUSION_METHOD))
    else:
        c1 = fuseCoeff(cooef1[i][0], cooef2[i][0], FUSION_METHOD)
        c2 = fuseCoeff(cooef1[i][1], cooef2[i][1], FUSION_METHOD)
        c3 = fuseCoeff(cooef1[i][2], cooef2[i][2], FUSION_METHOD)
        fusedCooef.append((c1, c2, c3))
```

Fuses the detail coefficients (horizontal, vertical, and diagonal) from both images' color channels for each decomposition level.



```python
fused_channel = pywt.waverec2(fusedCooef, wavelet)
```

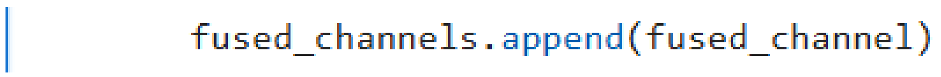
Reconstructs the fused color channel image from the fused wavelet coefficients.



```python
fused_channels.append(fused_channel)
```

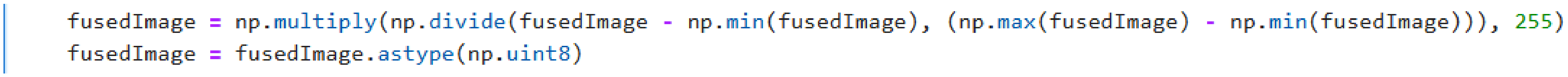
Merges the fused color channels back into a single RGB image.

Contd..



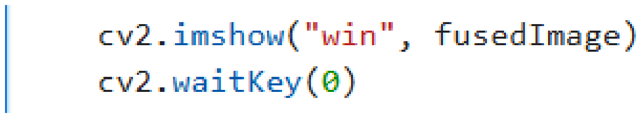
```
        fused_channels.append(fused_channel)
```

Merges the fused color channels back into a single RGB image.



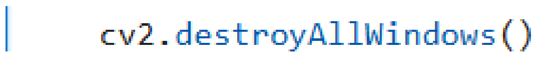
```
    fusedImage = np.multiply(np.divide(fusedImage - np.min(fusedImage), (np.max(fusedImage) - np.min(fusedImage))), 255)
    fusedImage = fusedImage.astype(np.uint8)
```

Normalizes the fused image to the range [0, 255] and converts the normalized fused image to an unsigned 8-bit integer format for display.



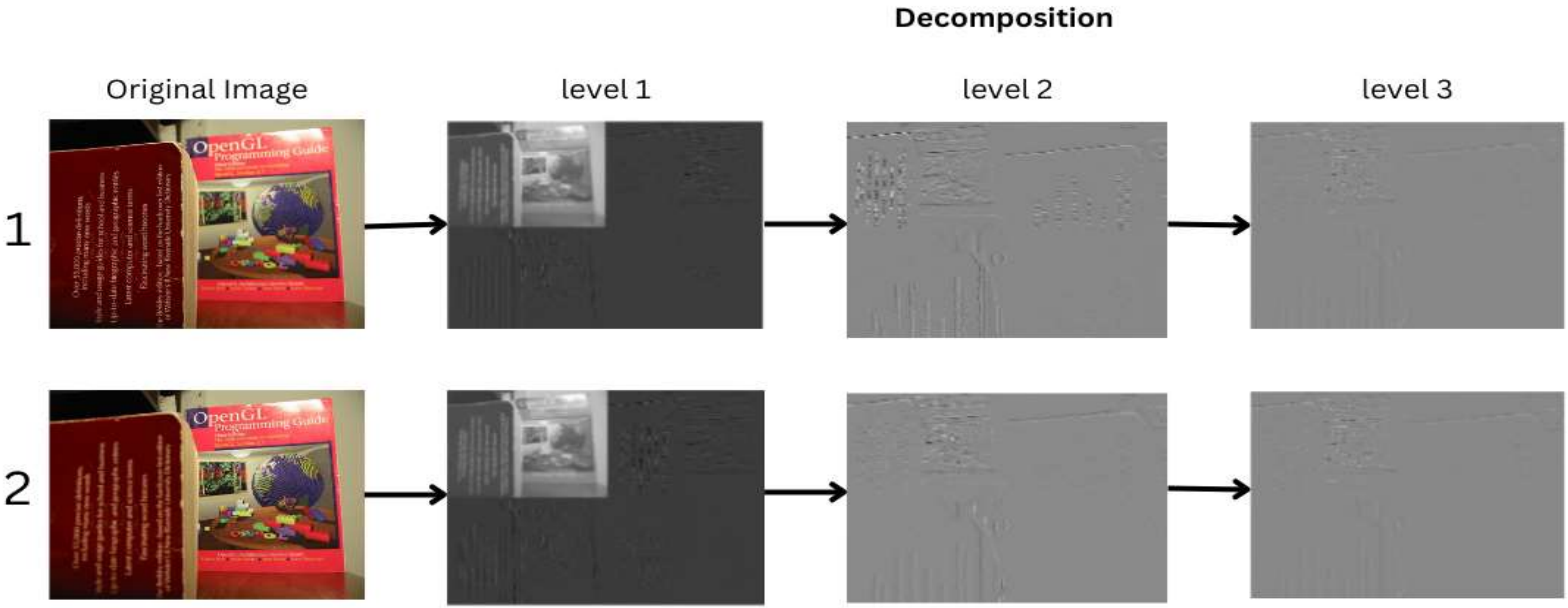
```
    cv2.imshow("win", fusedImage)
    cv2.waitKey(0)
```

Displays the fused image in a window,Waits for a key press to close the displayed image window.



```
    cv2.destroyAllWindows()
```

Closes all OpenCV windows.



Decomposition

Original Image

level 1

level 2

level 3

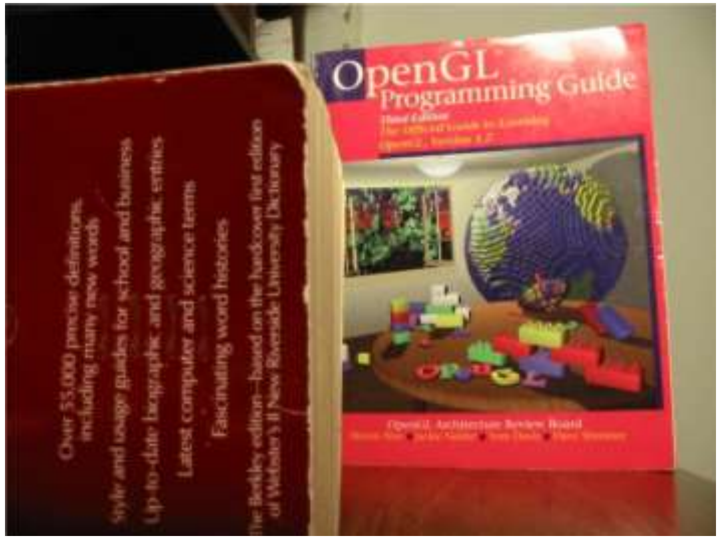
1

2

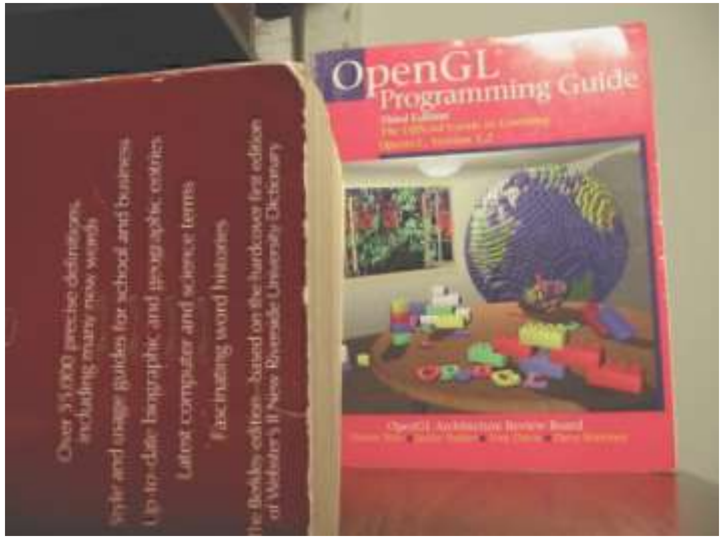
Reconstruction of fused image

FINAL OUTPUTS

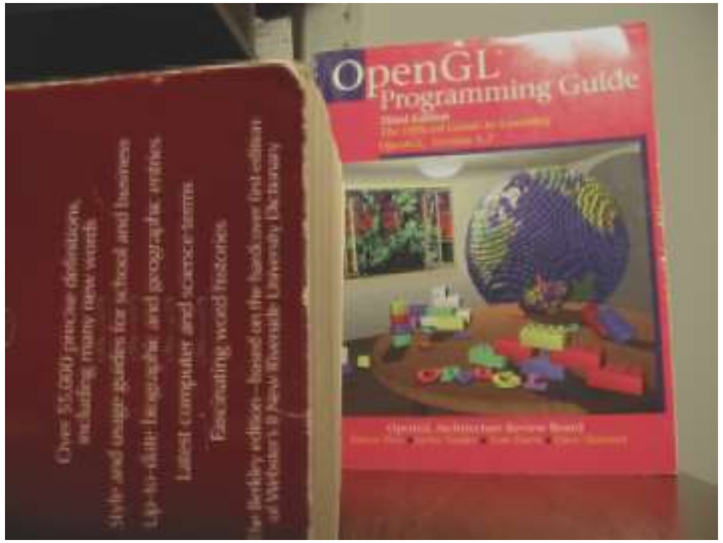
Using mean fusion rule


OpenGL Programming Guide
Third Edition
Over 55,000 precise definitions, including many new words
Style and usage guides for school and business
Up-to-date biographic and geographic entries
Latest computer and science terms
Fascinating word histories
The Berkley edition—based on the hardcover first edition of Webster's II New Riverside University Dictionary
OpenGL Architecture Review Board

Using min fusion rule


OpenGL Programming Guide
Third Edition
Over 55,000 precise definitions, including many new words
Style and usage guides for school and business
Up-to-date biographic and geographic entries
Latest computer and science terms
Fascinating word histories
The Berkley edition—based on the hardcover first edition of Webster's II New Riverside University Dictionary
OpenGL Architecture Review Board

Using max fusion rule


OpenGL Programming Guide
Third Edition
Over 55,000 precise definitions, including many new words
Style and usage guides for school and business
Up-to-date biographic and geographic entries
Latest computer and science terms
Fascinating word histories
The Berkley edition—based on the hardcover first edition of Webster's II New Riverside University Dictionary
OpenGL Architecture Review Board