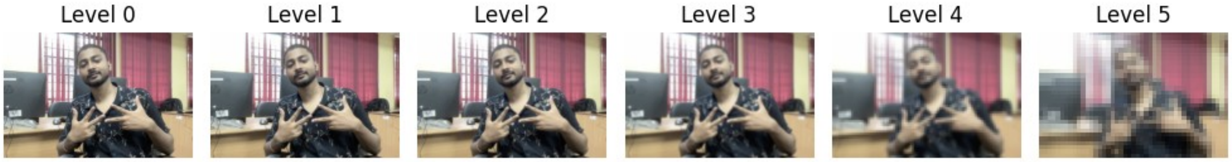
```python
import cv2
import matplotlib.pyplot as plt

# Function to display the image pyramid
def display_image_pyramid(image, levels=5):
    pyramid = [image]
    for i in range(levels):
        # Downsample the image
        image = cv2.pyrDown(image)
        pyramid.append(image)

    # Display all levels of the pyramid
    plt.figure(figsize=(10, 5))
    for i in range(levels + 1):
        plt.subplot(1, levels + 1, i + 1)
        plt.imshow(cv2.cvtColor(pyramid[i], cv2.COLOR_BGR2RGB))
        plt.title(f'Level {i}')
        plt.axis('off')

    plt.tight_layout()
    plt.show()

# Load the input image
image_path = '/content/test.jpg'  # Replace with your image path
image = cv2.imread(image_path)

# Display image pyramid
display_image_pyramid(image, levels=5)
```



```python
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Function to display the image pyramid
def display_image_pyramid(image, levels=5):
    pyramid = [image]
    for i in range(levels):
        # Downsample the image
        image = cv2.pyrDown(image)
        pyramid.append(image)

    # Display all levels of the pyramid
    plt.figure(figsize=(10, 5))
    for i in range(levels + 1):
        plt.subplot(1, levels + 1, i + 1)
        plt.imshow(cv2.cvtColor(pyramid[i], cv2.COLOR_BGR2RGB))
        plt.title(f'Level {i}')
        plt.axis('off')
    plt.tight_layout()
    plt.show()

def build_gaussian_pyramid(image, levels=5):
    pyramid = [image]
    for i in range(levels):
        image = cv2.pyrDown(image)
        pyramid.append(image)
    return pyramid

def build_laplacian_pyramid(gaussian_pyramid):
    laplacian_pyramid = []
    for i in range(len(gaussian_pyramid) - 1):
        size = (gaussian_pyramid[i].shape[1],
gaussian_pyramid[i].shape[0])
        expanded = cv2.pyrUp(gaussian_pyramid[i+1], dstsize=size)
        laplacian = cv2.subtract(gaussian_pyramid[i], expanded)
        laplacian_pyramid.append(laplacian)
    laplacian_pyramid.append(gaussian_pyramid[-1])
    return laplacian_pyramid

def reconstruct_from_laplacian(laplacian_pyramid):
    reconstructed = laplacian_pyramid[-1]
    for i in range(len(laplacian_pyramid) - 2, -1, -1):
        size = (laplacian_pyramid[i].shape[1],
laplacian_pyramid[i].shape[0])
        reconstructed = cv2.add(cv2.pyrUp(reconstructed,
dstsize=size), laplacian_pyramid[i])
    return reconstructed

def display_reconstruction(image, levels=5):
    # Build Gaussian pyramid
    gaussian_pyramid = build_gaussian_pyramid(image, levels)

    # Build Laplacian pyramid
    laplacian_pyramid = build_laplacian_pyramid(gaussian_pyramid)

    # Reconstruct images from each level
    reconstructed_images = []
    for i in range(levels, -1, -1):
        reconstructed =
reconstruct_from_laplacian(laplacian_pyramid[i:])
        reconstructed_images.append(reconstructed)

    # Display reconstructed images
    plt.figure(figsize=(15, 5))
    for i, img in enumerate(reconstructed_images):
        plt.subplot(1, levels + 1, i + 1)
        plt.imshow(cv2.cvtColor(img, cv2.COLOR_BGR2RGB))
        plt.title(f'Level {levels - i}')
        plt.axis('off')
    plt.tight_layout()
    plt.show()

# Load the input image
image_path = '/content/test.jpg'  # Replace with your image path
image = cv2.imread(image_path)

# Display reconstruction process
display_image_pyramid(image, levels=5)
display_reconstruction(image, levels=5)
```