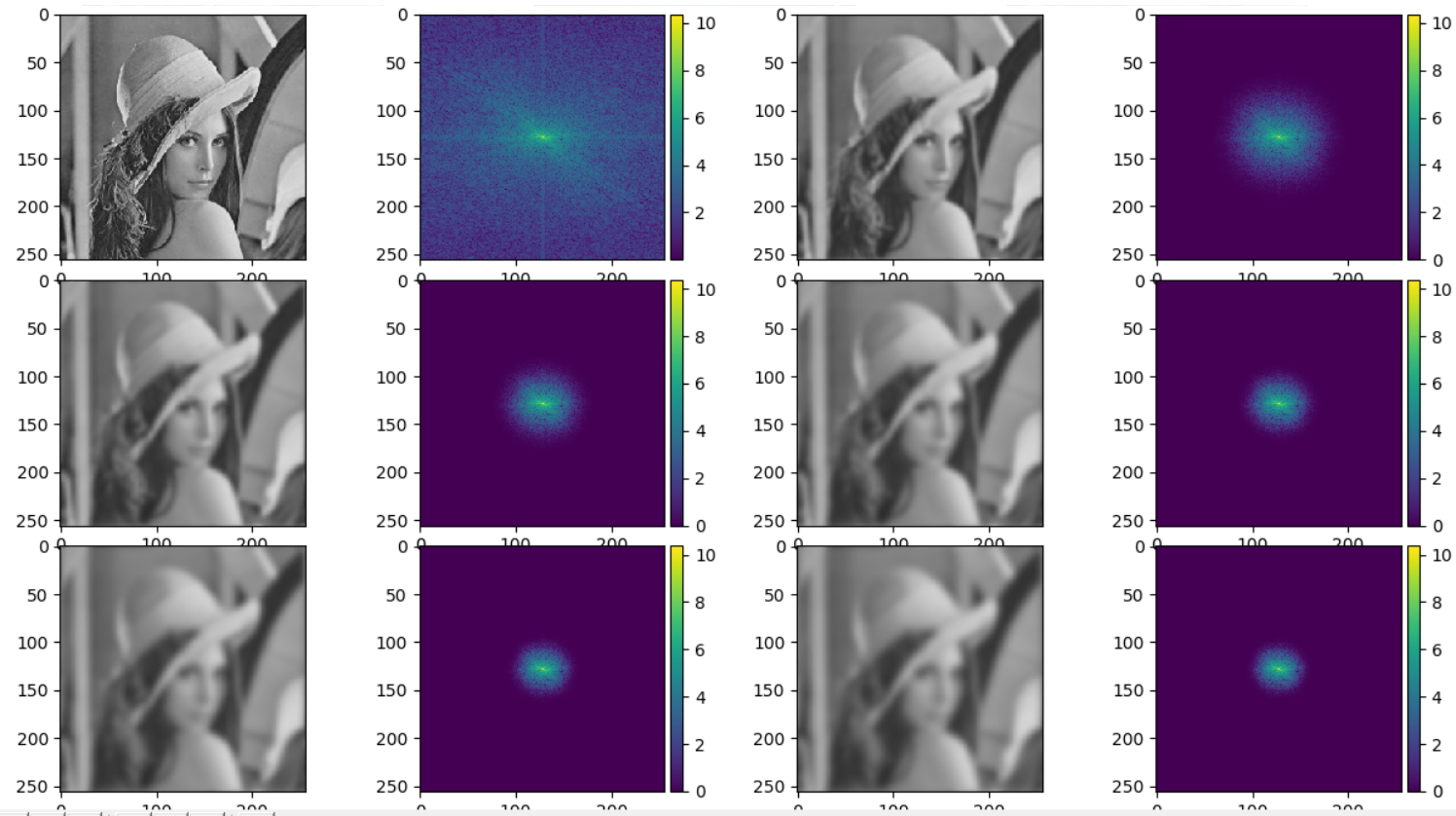
**Practical lesson №8. Gaussian and Laplace pyramids of image**

**Task №1**

Build a Gaussian pyramid of the image with at least five layers. Visualize the resulting images and the frequency amplitudes on each level of the pyramid. Make sure that the frequency range is narrowed on each layer. Build a pyramid for three different values of the Sigma of the Gaussian nucleus. For the convenience of experimentation, define a separate function (build\_gauss\_pyramid) for building a Gaussian pyramid with the parameters img (the image on which the pyramid is built), sigma (the parameter of the Gaussian kernel), n\_layers (the number of layers of the pyramid), which returns a list of required images.

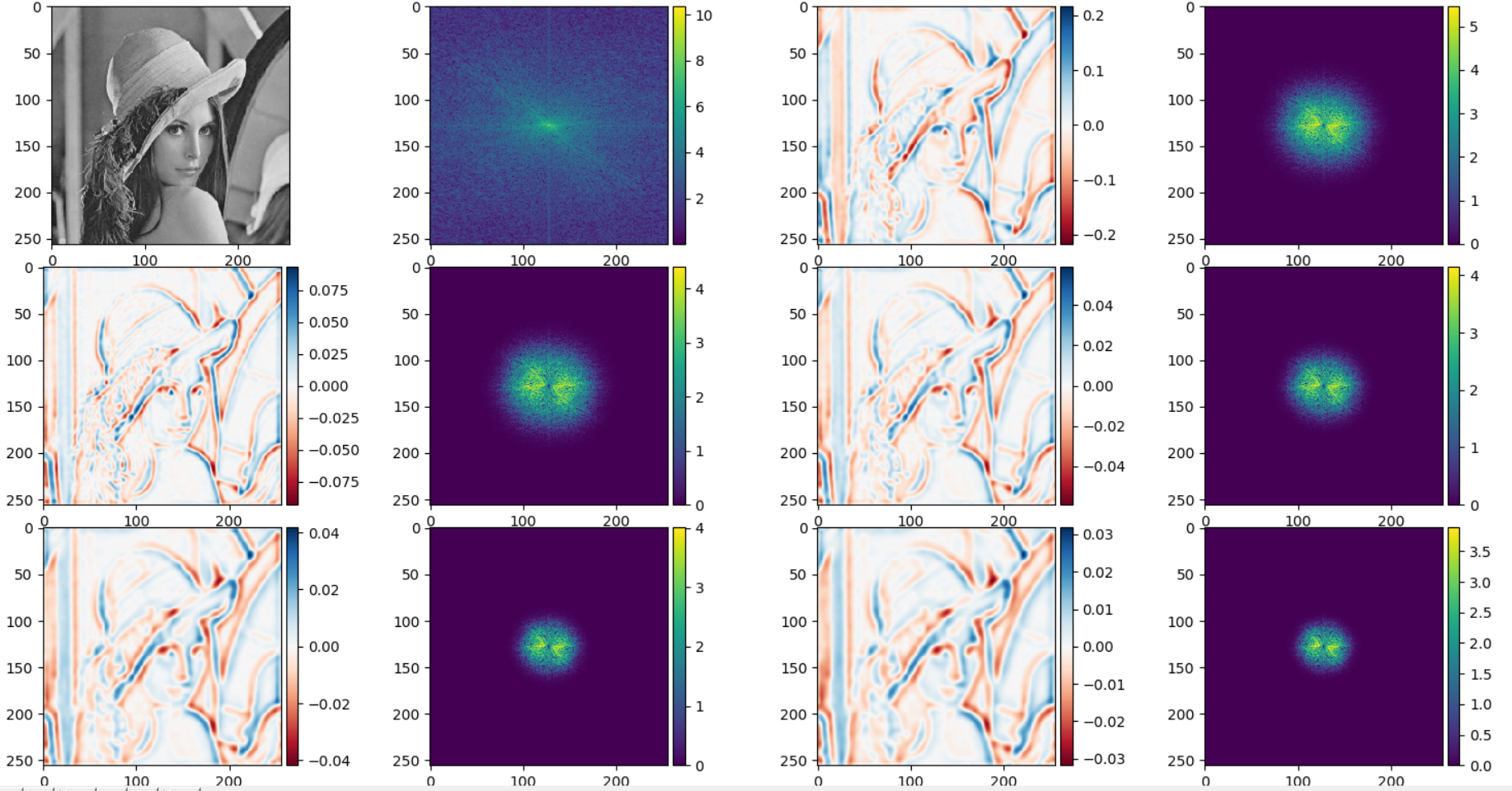
**from** skimage.io **import** imread, imshow  
**from** skimage.filters **import** gaussian  
**from** numpy.fft **import** fft2, fftshift, ifft2  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** cv2  
**from** skimage **import** img\_as\_float  
  
**def** build\_gauss\_pyramid(img, sigma, n\_layers):  
 FOURIER\_IMG = fft2(img)  
 filt\_gauss = np.zeros(img.shape)  
 filt\_gauss[img.shape[0] // 2, img.shape[1] // 2] = 1  
 IMG\_GAUSS = gaussian(filt\_gauss, sigma=sigma)  
 FOURIER\_GAUSS = fft2(IMG\_GAUSS)  
 imgs = []  
 fouries = []  
 **for** i **in** range(n\_layers):  
 FOURIER\_BLUR\_IMG = abs(FOURIER\_GAUSS)\* FOURIER\_IMG  
 FOURIER\_IMG=FOURIER\_BLUR\_IMG  
 imgs.append(ifft2(FOURIER\_BLUR\_IMG).real)  
 fouries.append(FOURIER\_BLUR\_IMG)**return** imgs, fouries  
  
**if** \_\_name\_\_ ==**"\_\_main\_\_"**:  
 src = img\_as\_float(imread(**'lena1.jpg'**, as\_gray=**True**))  
 plt.figure(figsize=(25, 17))  
 imgs, four = build\_gauss\_pyramid(src, 2, 5)  
 plt.subplot(341)  
 imshow(src)  
 plt.subplot(342)  
 FOURIER\_IMG = fft2(src)  
 imshow(np.log(1 + abs(fftshift(FOURIER\_IMG))))  
 plt.subplot(343)  
 imshow(imgs[0])  
 plt.subplot(344)  
 imshow(np.log(1 + abs(fftshift(four[0]))))  
 plt.subplot(345)  
 imshow(imgs[1])  
 plt.subplot(346)  
 imshow(np.log(1 + abs(fftshift(four[1]))))  
 plt.subplot(347)  
 imshow(imgs[2])  
 plt.subplot(348)  
 imshow(np.log(1 + abs(fftshift(four[2]))))  
 plt.subplot(349)  
 imshow(imgs[3])  
 plt.subplot(3,4,10)  
 imshow(np.log(1 + abs(fftshift(four[3]))))  
 plt.subplot(3,4,11)  
 imshow(imgs[4])  
 plt.subplot(3,4,12)  
 imshow(np.log(1 + abs(fftshift(four[4]))))  
 plt.show()



**Task №2.**

Build a Laplacian pyramid of the image with at least five layers. The function for constructing the Laplace pyramid should use the function of constructing the Gaussian pyramid and have, like the function of the Gaussian pyramid, the parameters img, sigma and n\_layers.

**from** skimage.io **import** imread, imshow  
**from** skimage.filters **import** gaussian  
**from** numpy.fft **import** fft2, fftshift, ifft2  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** cv2  
**from** skimage **import** img\_as\_float  
  
**def** build\_laplas\_pyramid(img, sigma, n\_layers):  
 FOURIER\_IMG = fft2(img)  
 filt\_gauss = np.zeros(img.shape)  
 filt\_gauss[img.shape[0] // 2, img.shape[1] // 2] = 1  
 IMG\_GAUSS = gaussian(filt\_gauss, sigma=sigma)  
 FOURIER\_GAUSS = fft2(IMG\_GAUSS)  
 imgs\_g = []  
 fouries = []  
 fouries\_lap=[]  
 imgs=[]  
 **for** i **in** range(n\_layers):  
 FOURIER\_BLUR\_IMG = abs(FOURIER\_GAUSS)\* FOURIER\_IMG  
 FOURIER\_IMG=FOURIER\_BLUR\_IMG  
 imgs\_g.append(ifft2(FOURIER\_BLUR\_IMG).real)  
 fouries.append(FOURIER\_BLUR\_IMG)  
 **for** i **in** range(n\_layers):  
 **if** (i-1)<0:  
 fouries\_lap.append(FOURIER\_IMG-fouries[0])  
 imgs.append(ifft2(fouries\_lap[0]).real)  
 **else**:  
 fouries\_lap.append(fouries[i-1]-fouries[i])  
 imgs.append(ifft2(fouries\_lap[i]).real)  
 **return** imgs, fouries\_lap  
  
**if** \_\_name\_\_ ==**"\_\_main\_\_"**:  
 src = img\_as\_float(imread(**'lena1.jpg'**, as\_gray=**True**))  
 plt.figure(figsize=(25, 15))  
 imgs, four = build\_laplas\_pyramid(src, 2, 5)  
 plt.subplot(341)  
 imshow(src)  
 plt.subplot(342)  
 FOURIER\_IMG = fft2(src)  
 imshow(np.log(1 + abs(fftshift(FOURIER\_IMG))))  
 plt.subplot(343)  
 imshow(imgs[0])  
 plt.subplot(344)  
 imshow(np.log(1 + abs(fftshift(four[0]))))  
 plt.subplot(345)  
 imshow(imgs[1])  
 plt.subplot(346)  
 imshow(np.log(1 + abs(fftshift(four[1]))))  
 plt.subplot(347)  
 imshow(imgs[2])  
 plt.subplot(348)  
 imshow(np.log(1 + abs(fftshift(four[2]))))  
 plt.subplot(349)  
 imshow(imgs[3])  
 plt.subplot(3,4,10)  
 imshow(np.log(1 + abs(fftshift(four[3]))))  
 plt.subplot(3,4,11)  
 imshow(imgs[4])  
 plt.subplot(3,4,12)  
 imshow(np.log(1 + abs(fftshift(four[4]))))  
 plt.show()



**Task №3.**

What levels of the Gaussian and Laplace pyramids need to be summed to get the original image? Find this levels, sum them and get the original image.

**from** skimage.io **import** imread, imshow  
**from** skimage.filters **import** gaussian  
**from** numpy.fft **import** fft2, fftshift, ifft2  
**import** numpy **as** np  
**import** matplotlib.pyplot **as** plt  
**import** cv2  
**from** skimage **import** img\_as\_float  
  
**def** build\_laplas\_pyramid(img, sigma, n\_layers):  
 FOURIER\_IMG = fft2(img)  
 filt\_gauss = np.zeros(img.shape)  
 filt\_gauss[img.shape[0] // 2, img.shape[1] // 2] = 1  
 IMG\_GAUSS = gaussian(filt\_gauss, sigma=sigma)  
 FOURIER\_GAUSS = fft2(IMG\_GAUSS)  
 imgs\_g = []  
 fouries = []  
 fouries\_lap=[]  
 imgs=[]  
 **for** i **in** range(n\_layers):  
 FOURIER\_BLUR\_IMG = abs(FOURIER\_GAUSS)\* FOURIER\_IMG  
 FOURIER\_IMG=FOURIER\_BLUR\_IMG  
 imgs\_g.append(ifft2(FOURIER\_BLUR\_IMG).real)  
 fouries.append(FOURIER\_BLUR\_IMG)  
 **for** i **in** range(n\_layers):  
 **if** (i-1)<0:  
 fouries\_lap.append(FOURIER\_IMG-fouries[0])  
 imgs.append(ifft2(fouries\_lap[0]).real)  
 **else**:  
 fouries\_lap.append(fouries[i-1]-fouries[i])  
 imgs.append(ifft2(fouries\_lap[i]).real)  
 original\_fories=np.zeros((src.shape[:2]))  
 **for** i **in** range(5):  
 original\_fories=original\_fories+fouries\_lap[i]+fouries[0]  
 img=ifft2(original\_fories).real  
 **return** img, original\_fories  
  
**if** \_\_name\_\_ ==**"\_\_main\_\_"**:  
 src = img\_as\_float(imread(**'lena1.jpg'**, as\_gray=**True**))  
 plt.figure(figsize=(25, 15))  
 imgs, four = build\_laplas\_pyramid(src, 2, 5)  
 plt.subplot(121)  
 imshow(np.log(1 + abs(fftshift(four))))  
 plt.subplot(122)  
 imshow(imgs)plt.show()

