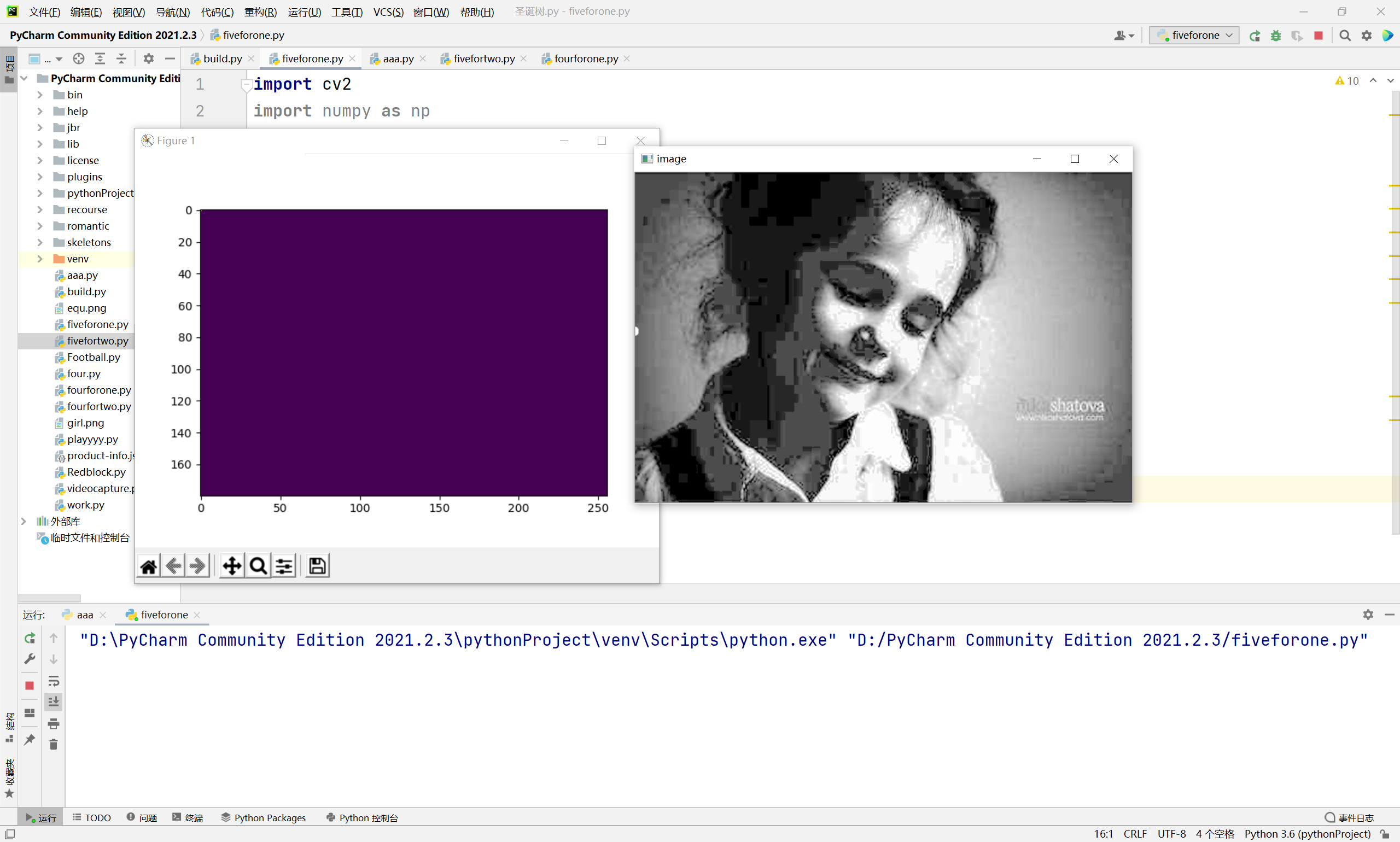
***1***

**import** cv2  
**import** numpy **as** np  
**from** matplotlib **import** pyplot **as** plt  
  
img = cv2.imread(**'./recourse/girl.png'**, 0)  
equ = cv2.equalizeHist(img) *# 并排堆叠图像*cv2.imshow(**'image'**, equ) *# 显示均衡化后的图*img = cv2.imread(**'./recourse/girl.png'**) *# 重新读取图片*hsv = cv2.cvtColor(img, cv2.COLOR\_BGR2HSV) *# 画2D直方图*hist = cv2.calcHist([hsv], [0, 1], **None**, [180, 256], [0, 180, 0, 256])  
plt.imshow(hist, interpolation=**'nearest'**)  
plt.show()  
  
cv2.waitKey(0) *# 按任意键关闭窗口*cv2.destroyAllWindows()



***2***

**import** cv2  
**import** numpy **as** np  
**from** matplotlib **import** pyplot **as** plt  
  
img = cv2.imread(**'./recourse/house.jpg'**, 0)  
dft = cv2.dft(np.float32(img), flags=cv2.DFT\_COMPLEX\_OUTPUT)  
dft\_shift = np.fft.fftshift(dft) *# 将低频分量转移到频谱中心*magnitude\_spectrum = 20 \* np.log(cv2.magnitude(dft\_shift[:, :, 0], dft\_shift[:, :, 1])) *# 构建振幅图*rows, cols = img.shape  
crow, ccol = rows / 2, cols / 2 *# 获取中心位置*mask = np.ones((rows, cols, 2), np.uint8) *# 高通滤波*mask[int(crow - 10):int(crow + 10), int(ccol - 114):int(ccol - 4)] = 0  
mask[int(crow - 10):int(crow + 10), int(ccol + 4):int(ccol + 114)] = 0  
fshift = dft\_shift \* mask *# 将掩码与当前得到的结果结合在一起*f\_ishift = np.fft.ifftshift(fshift) *# 将低频还原到左上角*img\_back = cv2.idft(f\_ishift) *# 傅里叶逆变换*img\_back = cv2.magnitude(img\_back[:, :, 1], img\_back[:, :, 0]) *# 将实部与虚部进行处理*plt.imshow(img\_back, cmap=**'gray'**)  
plt.title(**'result'**), plt.xticks([]), plt.yticks([])  
plt.show()

