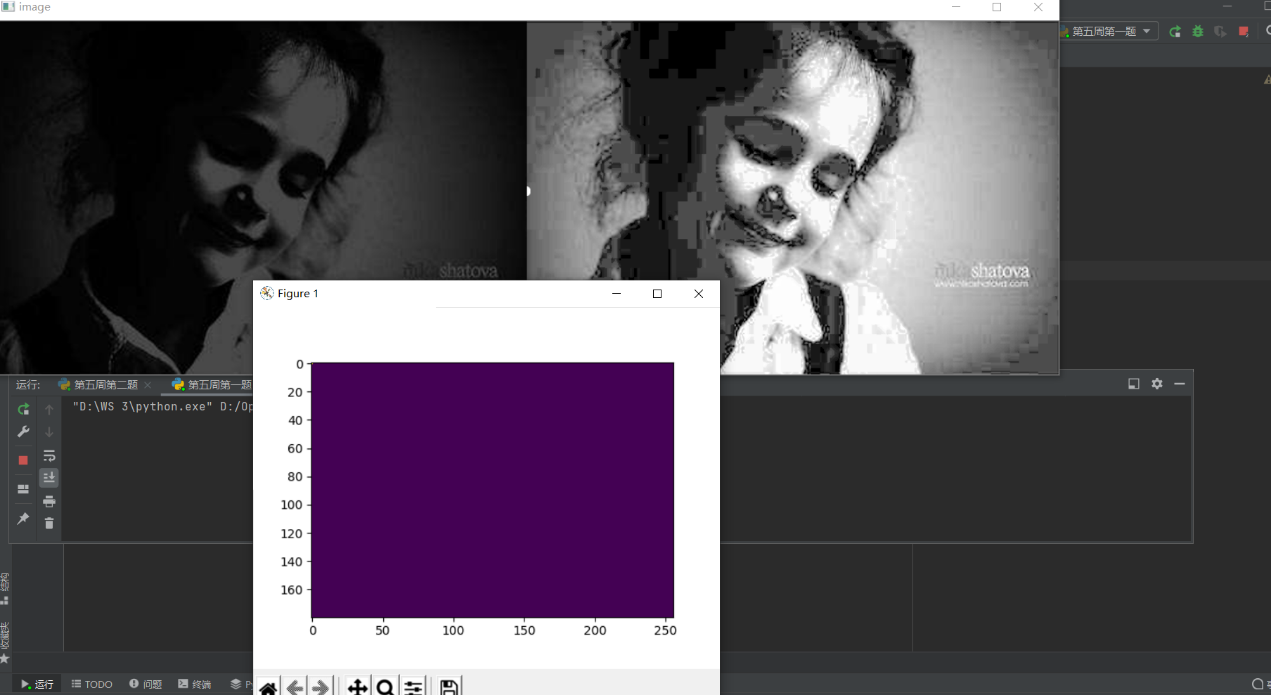
第一题

1.代码：

import cv2  
import numpy as np  
from matplotlib import pyplot as plt  
  
img=cv2.imread(&apos;girl.png&apos;,0)  
equ=cv2.equalizeHist(img)#（对灰度图）均衡化  
res=np.hstack((img,equ))#并排堆放（把矩阵进行行连接）  
cv2.imwrite(&apos;res.png&apos;,res)#将图像保存到指定的文件  
cv2.imshow(&apos;image&apos;,res)  
  
img = cv2.imread(&apos;girl.png&apos;)  
hsv = cv2.cvtColor(img,cv2.COLOR\_BGR2HSV)  
hist = cv2.calcHist([hsv], [0, 1], None, [180, 256], [0, 180, 0, 256])#获取直方图数值  
                                                                      #channels=[0 ，1] 因为我们需要同时处理 H 和 S 两个通道（颜色和饱和度）  
                                                                      # bins=[180 ，256]H 通道为 180，S 通道为 256  
                                                                      # range=[0 ，180 ，0 ，256]H 的取值范围在 0 到 180，S 的取值范围在 0 到 256  
plt.imshow(hist,interpolation = &apos;nearest&apos;)  
plt.show()  
  
cv2.waitKey(0)  
cv2.destroyAllWindows()

2.效果图：



第二题

1.代码：

import numpy as np  
import cv2  
from matplotlib import pyplot as plt  
  
img=cv2.imread(&apos;house.jpg&apos;,0)  
dft=cv2.dft(np.float32(img),flags=cv2.DFT\_COMPLEX\_OUTPUT)#实现傅里叶变换。cv2.dft(原始图像，转换标识) 这里的原始图像必须是np.float32格式。  
                      # 所以，我们首先需要使用cv2.float32 ()函数将图像转换。 而转换标识的值通常为cv2.DFT\_COMPLEX\_OUTPUT，用来输出一个复数阵列  
dft\_shift=np.fft.fftshift(dft)  
  
cows,cols=img.shape  
cow,col=int(cows/2),int(cols/2)  
mask=np.ones((cows,cols,2),np.uint8)  
mask[:,col-15:col+15]=255  
fshift=dft\_shift\*mask  
f\_ishif=np.fft.ifftshift(fshift)  
img\_back=cv2.idft(f\_ishif)  
img\_back=cv2.magnitude(img\_back[:,:,0],img\_back[:,:,1])  
img\_back=cv2.medianBlur(img\_back,5)  
  
plt.subplot(121),plt.imshow(img,cmap=&apos;gray&apos;)  
plt.title(&apos;1&apos;),plt.xticks([]),plt.yticks([])  
plt.subplot(122),plt.imshow(img\_back,cmap=&apos;gray&apos;)  
plt.title(&apos;2&apos;),plt.xticks([]),plt.yticks([])  
plt.show()

2.效果图：

