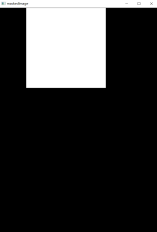
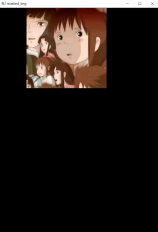
*'''  
2021.1.31FromIvicxDS:openCV;E6：直方图  
'''***'''  
读取灰度图，根据相应灰度值的像素点数绘制直方图  
cv2.calcHist(images,channels,mask,histSize,ranges)  
\*所有参数传入要用中括号[images]  
images:  
channels:BGR[0],[1],[2]  
mask:指定图像的一个区域进行统计  
histSize:指定取值的宽度ex0-10画一条，默认256  
range:256  
'''**import cv2*#读取格式为BGR*import numpy  
from matplotlib import pyplot as plt  
  
def showPicture(name,picture):  
 *#图像的显示,也可以显示多窗口* cv2.imshow(name,picture)  
 *#在键盘中按任意键退出显示并向后执行语句  
 #cv2.waitKey(1000)表示只显示1秒* cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
img=cv2.imread(**"image/C2.jpg"**)  
color=(**"b"**,**"g"**,**"r"**)  
for i,col in enumerate(color):  
 histr=cv2.calcHist([img],[i],None,[256],[0,256])  
 plt.plot(histr,color=col)  
 plt.xlim([0,256])  
  
plt.show()

*'''  
2021.1.31FromIvicxDS:openCV;E6：直方图,掩码的创建  
'''***'''  
读取灰度图，根据相应灰度值的像素点数绘制直方图  
cv2.calcHist(images,channels,mask,histSize,ranges)  
\*所有参数传入要用中括号[images]  
images:  
channels:BGR[0],[1],[2]  
mask:指定图像的一个区域进行统计  
histSize:指定取值的宽度ex0-10画一条，默认256  
range:256  
'''**import cv2*#读取格式为BGR*import numpy  
  
def showPicture(name,picture):  
 *#图像的显示,也可以显示多窗口* cv2.imshow(name,picture)  
 *#在键盘中按任意键退出显示并向后执行语句  
 #cv2.waitKey(1000)表示只显示1秒* cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
img=cv2.imread(**"image/C2.jpg"**)*#灰度图*print(img.shape)  
  
*#构建一个MASK掩码>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>*mask=numpy.zeros(img.shape[:2],numpy.uint8)*#八位灰度图*print(mask.shape)  
mask[0:300,100:400]=255*#要保留的位置设为白色*showPicture(**"maskedImage"**,mask)  
  
*#用与操作使得掩码生效>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>*masked\_img=cv2.bitwise\_and(img,img,mask=mask)  
showPicture(**"masked\_img"**,masked\_img)

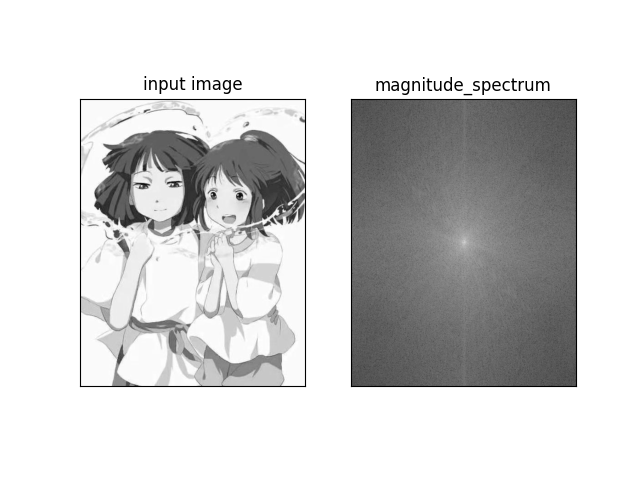




*'''  
2021.1.31FromIvicxDS:openCV;E6：直方图,直方图均衡化  
'''***'''  
灰度值->像素个数->概率->累计概率->根据函数映射后的灰度值->取整  
'''**import cv2*#读取格式为BGR*import numpy  
  
def showPicture(name,picture):  
 *#图像的显示,也可以显示多窗口* cv2.imshow(name,picture)  
 *#在键盘中按任意键退出显示并向后执行语句  
 #cv2.waitKey(1000)表示只显示1秒* cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
img=cv2.imread(**"image/93401a6eb97e5b304361025f759006d.jpg"**,0)  
equ=cv2.equalizeHist(img)  
res=numpy.hstack((img,equ))  
  
showPicture(**"equalizeHist"**,res)



*'''  
2021.1.31FromIvicxDS:openCV;E6：直方图,直方图均衡化  
'''***'''  
将图片分成小格分别做均衡化防止图片中细节丢失  
【涉及对格子的边界处理】  
'''**import cv2*#读取格式为BGR*import numpy  
  
def showPicture(name,picture):  
 *#图像的显示,也可以显示多窗口* cv2.imshow(name,picture)  
 *#在键盘中按任意键退出显示并向后执行语句  
 #cv2.waitKey(1000)表示只显示1秒* cv2.waitKey(0)  
 cv2.destroyAllWindows()  
  
img=cv2.imread(**"image/x1.jpg"**,0)  
  
clahe=cv2.createCLAHE(clipLimit=2.0,tileGridSize=(8,8))  
  
equ\_img=cv2.equalizeHist(img)  
clahe\_img=clahe.apply(img)  
res=numpy.hstack((img,equ\_img,clahe\_img))  
  
showPicture(**"equalizeHist"**,res)  


*'''  
2021.1.31FromIvicxDS:openCV;E6：傅里叶变换和滤波器  
'''***'''  
高频：变化剧烈的灰度值【边界】  
低频：  
滤波：  
 低通滤波器：只保留低频，会使图像模糊  
 高通滤波器：只保留高频，会使图像细节增强  
openCV操作  
 cv2.dft(),cv2.idft正逆操作，图像要转换为np.float32格式  
 默认结果低频在左上角，通过shift变换拉到中心位置  
 cv2.dft()的返回结果是复数，要转换为图像格式才能展示(0,255)  
'''**import cv2*#读取格式为BGR*import numpy  
from matplotlib import pyplot as plt  
  
img=cv2.imread(**"image/x1.jpg"**,0)  
  
img\_float32=numpy.float32(img)  
  
dft=cv2.dft(img\_float32,flags=cv2.DFT\_COMPLEX\_OUTPUT)  
dft\_shift=numpy.fft.fftshift(dft)  
  
magnitude\_spectrum=20\*numpy.log(cv2.magnitude(dft\_shift[:,:,0],dft\_shift[:,:,1]))  
  
plt.subplot(121)  
plt.imshow(img,cmap=**"gray"**)  
plt.title(**"input image"**)  
plt.xticks(())  
plt.yticks(())  
  
plt.subplot(122)  
plt.imshow(magnitude\_spectrum,cmap=**"gray"**)  
plt.title(**"magnitude\_spectrum"**)  
plt.xticks(())  
plt.yticks(())  
  
plt.show()  
  
  
  


*'''  
2021.1.31FromIvicxDS:openCV;E6：傅里叶变换和滤波器  
'''*import cv2*#读取格式为BGR*import numpy  
from matplotlib import pyplot as plt  
  
img=cv2.imread(**"image/x1.jpg"**,0)  
  
img\_float32=numpy.float32(img)  
  
dft=cv2.dft(img\_float32,flags=cv2.DFT\_COMPLEX\_OUTPUT)  
dft\_shift=numpy.fft.fftshift(dft)  
  
rows,cols=img.shape  
crow,ccol=int(rows/2),int(cols/2)*#确定中心位置*mask=numpy.zeros((rows,cols,2),numpy.uint8)*#低通滤波【取中心正方形】*mask[crow-30:crow+30,ccol-30:ccol+30]=1  
  
fshift=dft\_shift\*mask  
f\_ishift=numpy.fft.ifftshift(fshift)*#逆运算一步步返回为图像*img\_back=cv2.idft(f\_ishift)  
img\_back=cv2.magnitude(img\_back[:,:,0],img\_back[:,:,1])  
  
plt.subplot(121)*#表格*plt.imshow(img,cmap=**"gray"**)  
plt.title(**"input image"**)  
plt.xticks(())  
plt.yticks(())  
  
plt.subplot(122)*#表格*plt.imshow(img\_back,cmap=**"gray"**)  
plt.title(**"output image"**)  
plt.xticks(())  
plt.yticks(())  
  
plt.show()

*'''  
2021.1.31FromIvicxDS:openCV;E6：傅里叶变换和滤波器  
'''*import cv2*#读取格式为BGR*import numpy  
from matplotlib import pyplot as plt  
  
img=cv2.imread(**"image/x1.jpg"**,0)  
  
img\_float32=numpy.float32(img)  
  
dft=cv2.dft(img\_float32,flags=cv2.DFT\_COMPLEX\_OUTPUT)  
dft\_shift=numpy.fft.fftshift(dft)  
  
rows,cols=img.shape  
crow,ccol=int(rows/2),int(cols/2)*#确定中心位置*mask=numpy.ones((rows,cols,2),numpy.uint8)*#低通滤波【取中心正方形】*mask[crow-30:crow+30,ccol-30:ccol+30]=0  
  
fshift=dft\_shift\*mask  
f\_ishift=numpy.fft.ifftshift(fshift)*#逆运算一步步返回为图像*img\_back=cv2.idft(f\_ishift)  
img\_back=cv2.magnitude(img\_back[:,:,0],img\_back[:,:,1])  
  
plt.subplot(121)*#表格*plt.imshow(img,cmap=**"gray"**)  
plt.title(**"input image"**)  
plt.xticks(())  
plt.yticks(())  
  
plt.subplot(122)*#表格*plt.imshow(img\_back,cmap=**"gray"**)  
plt.title(**"output image"**)  
plt.xticks(())  
plt.yticks(())  
  
plt.show()

