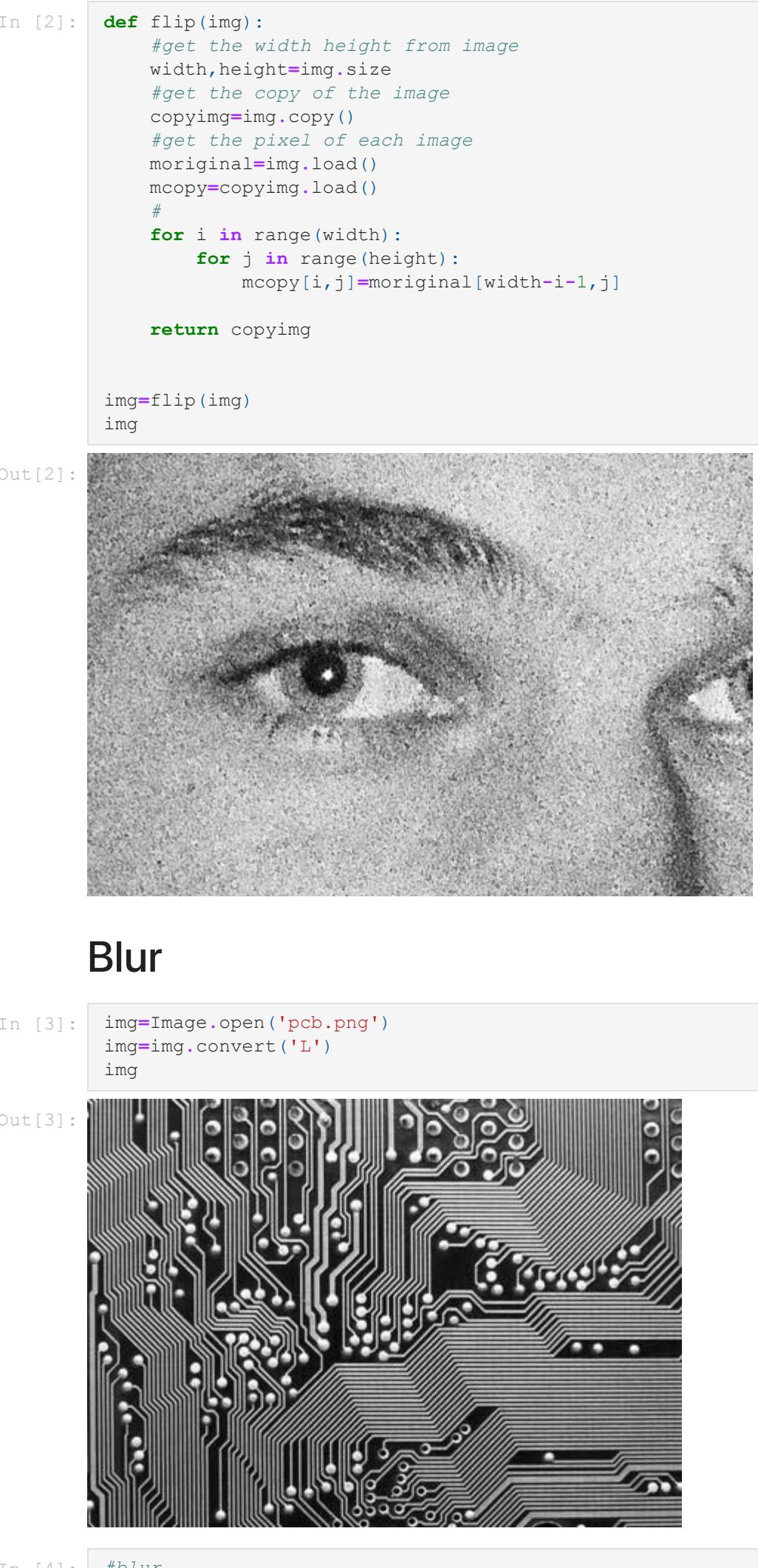


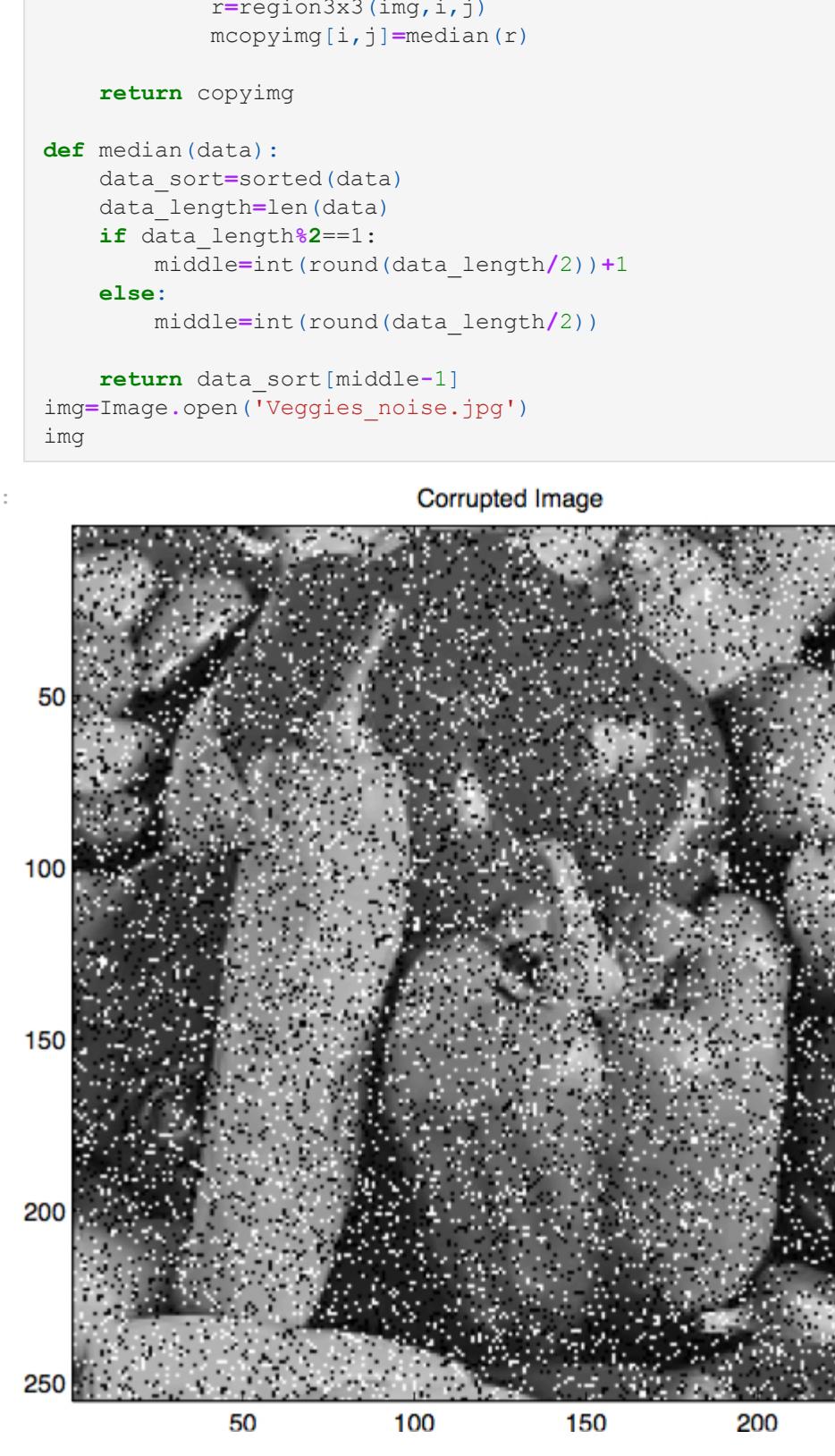
View

```
In [1]: from PIL import Image  
img=Image.open('eye.png')  
img.convert('L')  
img
```



Flip

```
In [2]: def flip(img):  
    #get the width height from image  
    width,height=img.size  
    #get the copy of the image  
    copyimg=copy()  
    #get the pixel of each image  
    moriginal=img.load()  
    mcopy=copyimg.load()  
    #  
    for i in range(width):  
        for j in range(height):  
            mcopy[i,j]=moriginal[width-i-1,j]  
  
    return copyimg  
  
img=flip(img)  
img
```

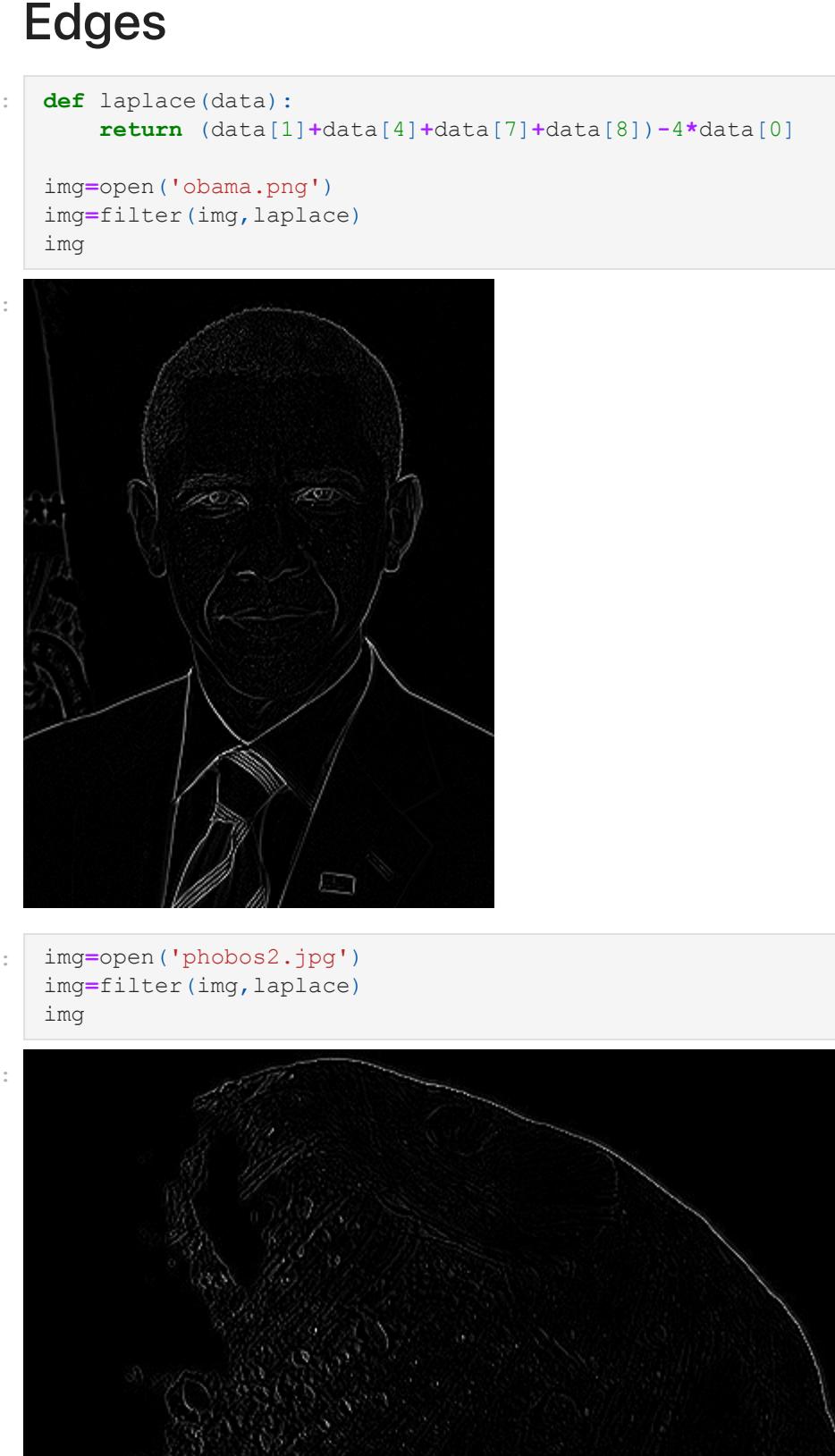


Blur

```
In [3]: img=Image.open('pcb.png')  
img=img.convert('L')  
img
```

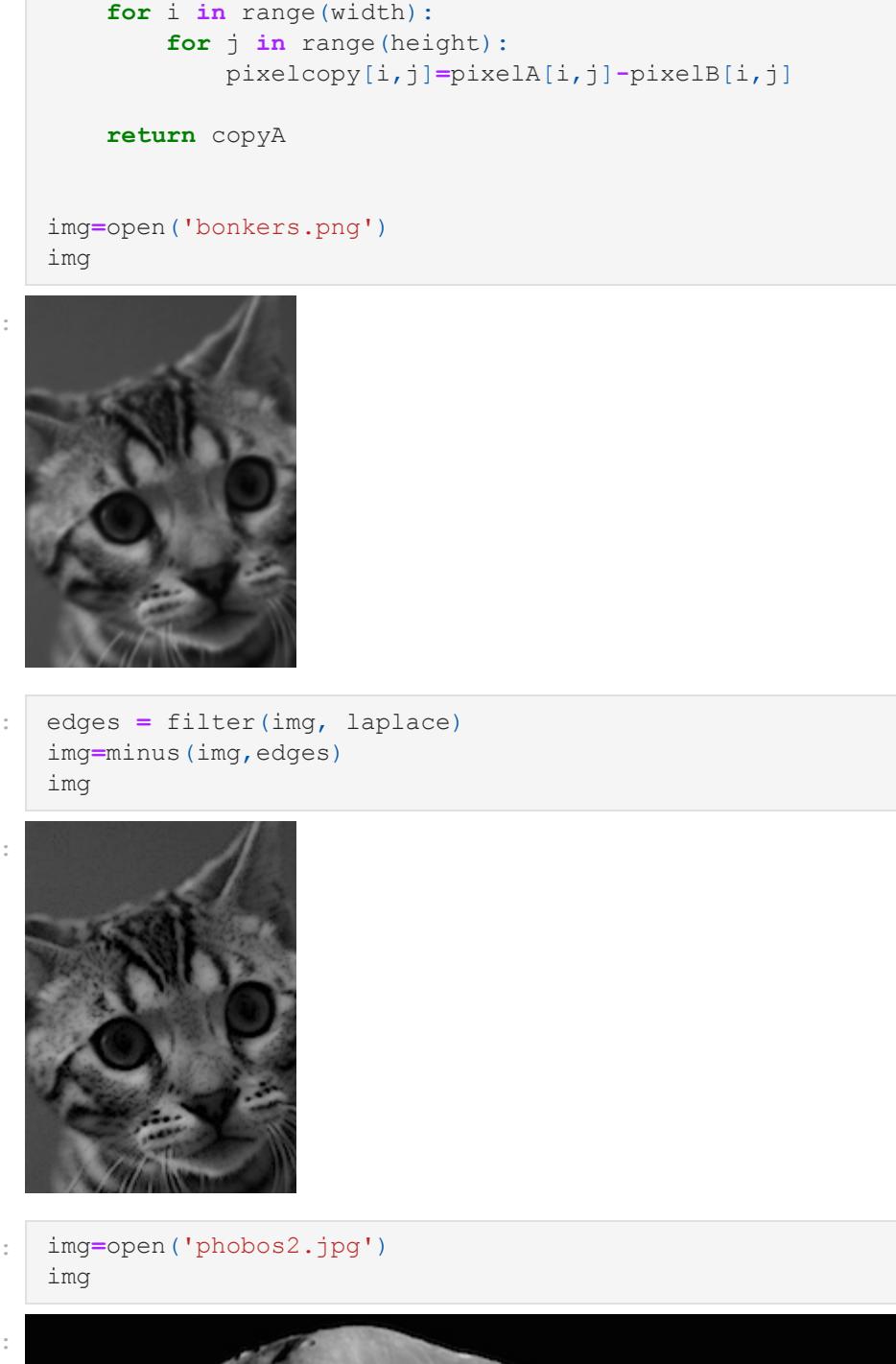


```
In [4]: #blur  
def blur(img):  
    #get the width of the image  
    width,height=img.size  
    copyimg=copy()  
    mcopying=copyimg.load()  
  
    for i in range(width):  
        for j in range(height):  
            r=region3x3(img,i,j)  
            mcopying[i,j]=avg(r)  
  
    return copyimg  
  
def avg(data):  
    return int(sum(data)/len(data))  
  
def region3x3(img,x,y):  
    center=img[x,y]  
    ccenter=region3x3(img, x, y)  
    N = getpixel(img, x, y - 1)  
    NW=getpixel(img, x+1, y - 1)  
    NWW=getpixel(img, x-1, y - 1)  
    S = getpixel(img, x, y + 1)  
    SW=getpixel(img, x+1, y + 1)  
    SWW=getpixel(img, x-1, y + 1)  
    Ww=getpixel(img, x-1, y)  
    Ew=getpixel(img, x+1, y)  
  
    return [center,N,NE,NW,S,SE,SW,W,E]  
  
def getpixel(img,x,y):  
    width,height=img.size  
    pixel=img.load()  
  
    if x<0:  
        x=0  
    elif x>=width:  
        x=width-1  
  
    if y<0:  
        y=0  
    elif y>=height:  
        y=height-1  
  
    return pixel[x,y]
```



Denoise

```
In [16]: def denoise(img):  
    #get the width of the image  
    width,height=img.size  
    copyimg=copy()  
    mcopying=copyimg.load()
```



Generic Filter

```
In [7]: def open(fname):  
    img = Image.open(fname)  
    im = img.convert("L")  
    return im  
  
def filter(image, f):  
    #get the width of the image  
    width,height=image.size  
    copyimg=copy()  
    mcopying=copyimg.load()  
  
    for i in range(width):  
        for j in range(height):  
            r=region3x3(image,i,j)  
            mcopying[i,j]=f(r)  
  
    return copyimg
```



Blur refactored

```
In [8]: img=open('pcb.png')  
img
```



```
Out[8]:
```



```
In [9]: img=filter(img,avg)  
img
```


Denoise refactored

```
In [10]: img=open('guesswho.png')  
img
```



```
In [11]: img=filter(img,median)  
img
```


Edges

```
In [12]: def laplace(data):  
    return (data[11]*data[4]+data[7]*data[8])-4*data[0]
```



```
img=open('obama.png')  
img=filter(img,laplace)
```


Sharpen

```
In [20]: def minus(A,B):  
    pixelA=A.load()  
    pixelB=B.load()  
    copyA=A.copy()  
    pixelcopy=copyA.load()
```



```
img=open('bonkers.png')  
img
```


Blur

```
In [21]: edges = filter(img, laplace)
```



```
img=minus(img,edges)
```


Edges

```
In [22]: img=open('phobos2.jpg')  
img
```



```
edges = filter(img, laplace)
```


Sharpen

```
In [20]: def minus(A,B):  
    pixelA=A.load()  
    pixelB=B.load()  
    copyA=A.copy()  
    pixelcopy=copyA.load()
```



```
img=open('bonkers.png')  
img
```

