

1. Primeira imagem:



Imagem original

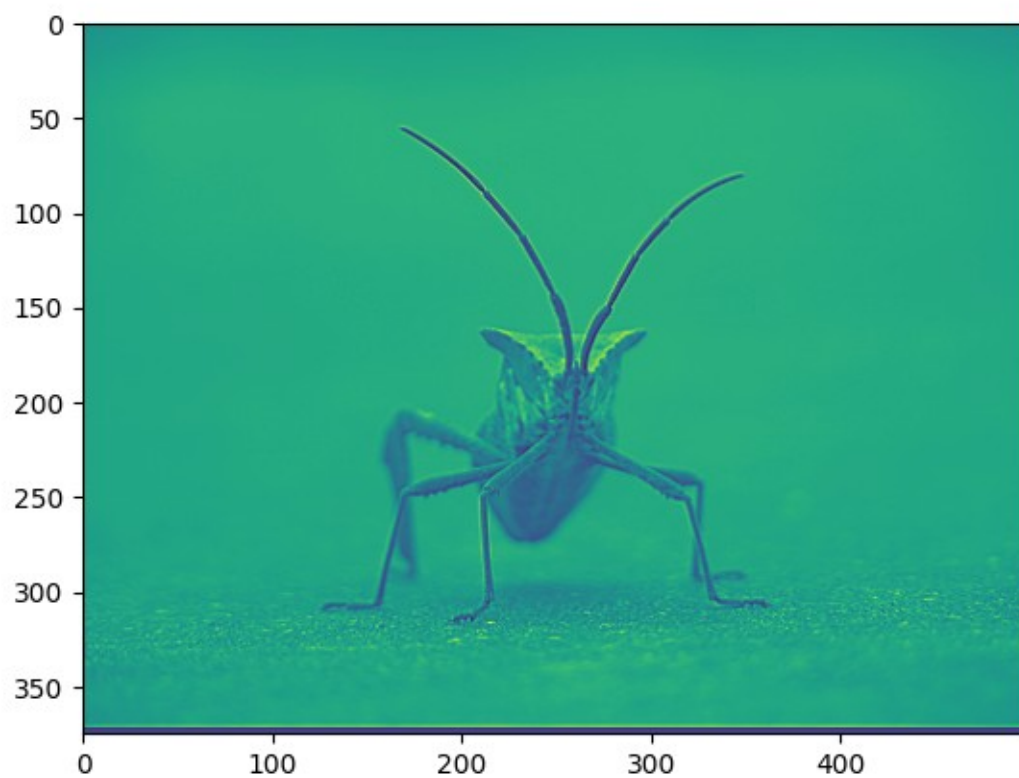


Imagem com filtro de nitidez

2. Segunda imagem:



Imagem original

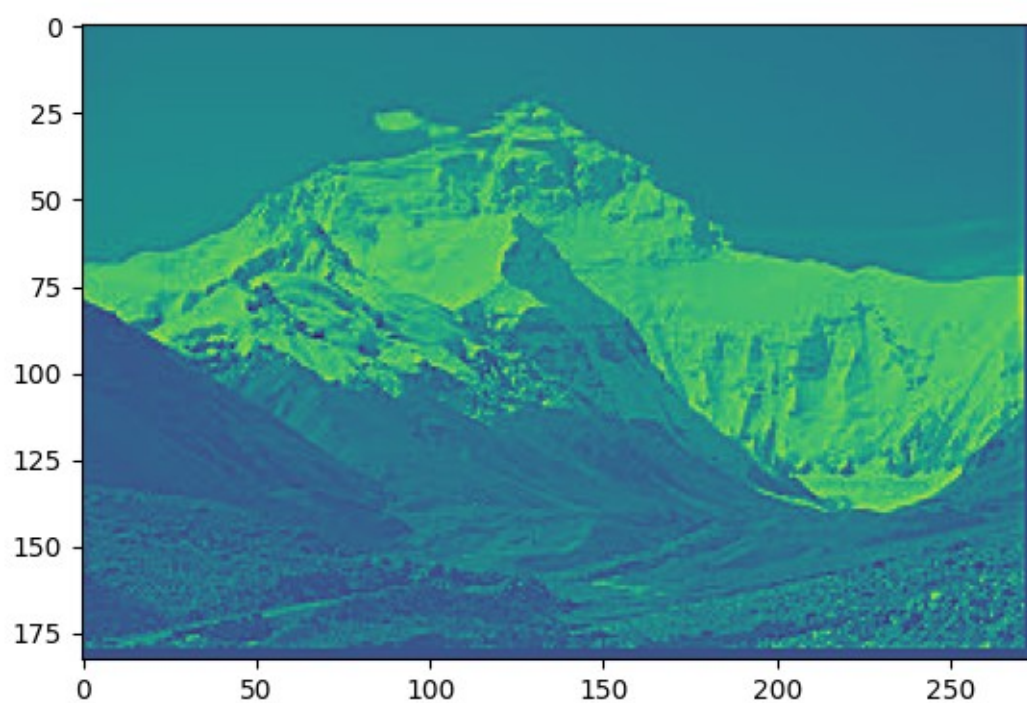


Imagem com filtro de nitidez

3. Terceira imagem:



Imagem original

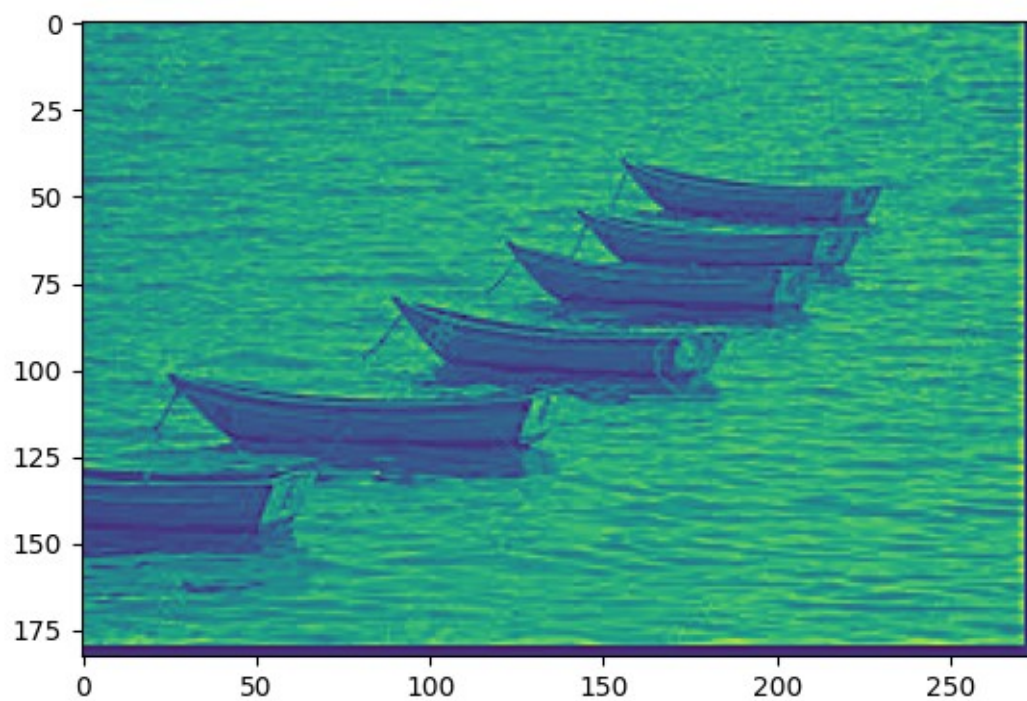


Imagem com filtro de nitidez

```

import cv2 as cv
from matplotlib import pyplot as plt
from matplotlib import image as im
import numpy as np
from math import pi
from math import e
from math import ceil

image = cv.imread("diretorio e nome do arquivo de imagem")

jpim = np.zeros([image.shape[0], image.shape[1]], dtype = float)

for i in range( len(image)-1):
    for j in range(len(image[0])-1):
        jpim[i][j] = (image[i][j][0]/3) + (image[i][j][1]/3) +
(image[i][j][2]/3)

#parametros da mascara
dimF = 5
mdimF = ceil(dimF/2)
filt = np.zeros([dimF, dimF], dtype = float)
sigma = 2
sig2 = sigma**2

gauss1 = 1/(2*pi*sig2)
gauss2 = gauss1*pi

blur = np.zeros([dimF, dimF], dtype = float)
shar = np.zeros([dimF, dimF], dtype = float)
norm = np.zeros([dimF, dimF], dtype = float)

beta = 10

#sharpening mask
for i in range(dimF):
    for j in range(dimF):
        if i == (mdimF-1) and j == (mdimF-1):
            norm[i][j] = 1
            blur[i][j] = gauss1*e**(-(((i-mdimF)**2)+((j-mdimF)**2))*gauss2)
            shar[i][j] = (beta+1)*norm[i][j] - beta*(blur[i][j])

sharp = np.zeros([len(jpim), len(jpim[0])], dtype = float)

#convolucao
for i in range(len(jpim)):
    for j in range(len(jpim[0])):
        for k in range(len(filt)):
            for l in range(len(filt)):
                if (i+k) > (len(jpim)-1) or (j+l) > (len(jpim[0])-1):
                    sharp[i][j] += 0
                else:
                    sharp[i][j] += shar[k][l] * jpim[i+k][j+l]

plt.imshow(sharp)
plt.show()

```