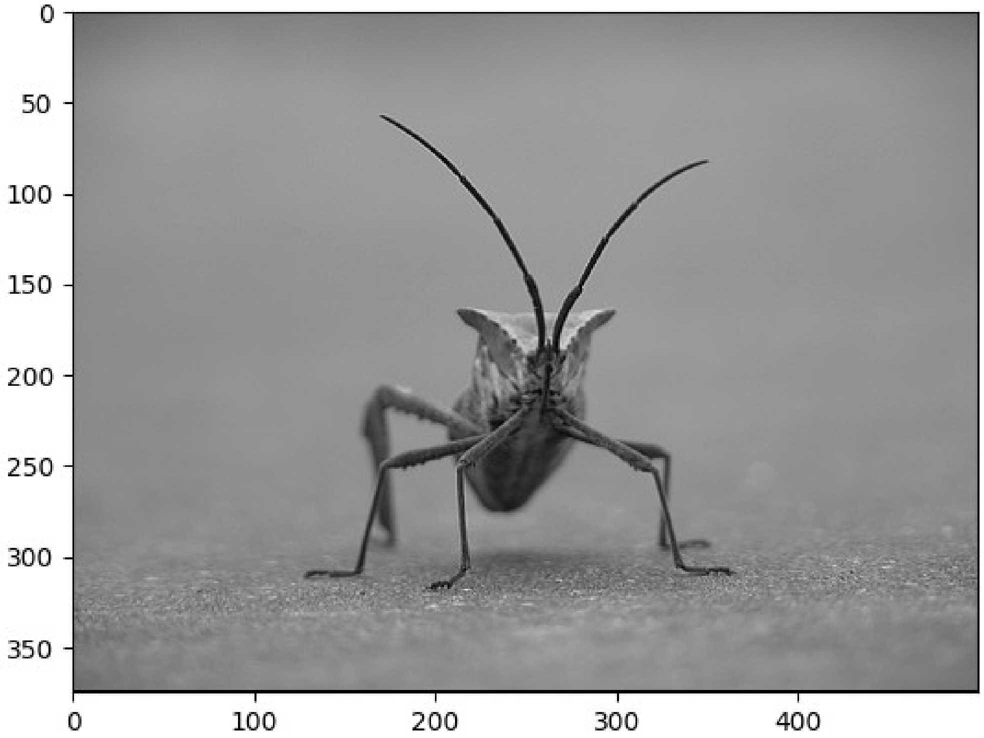
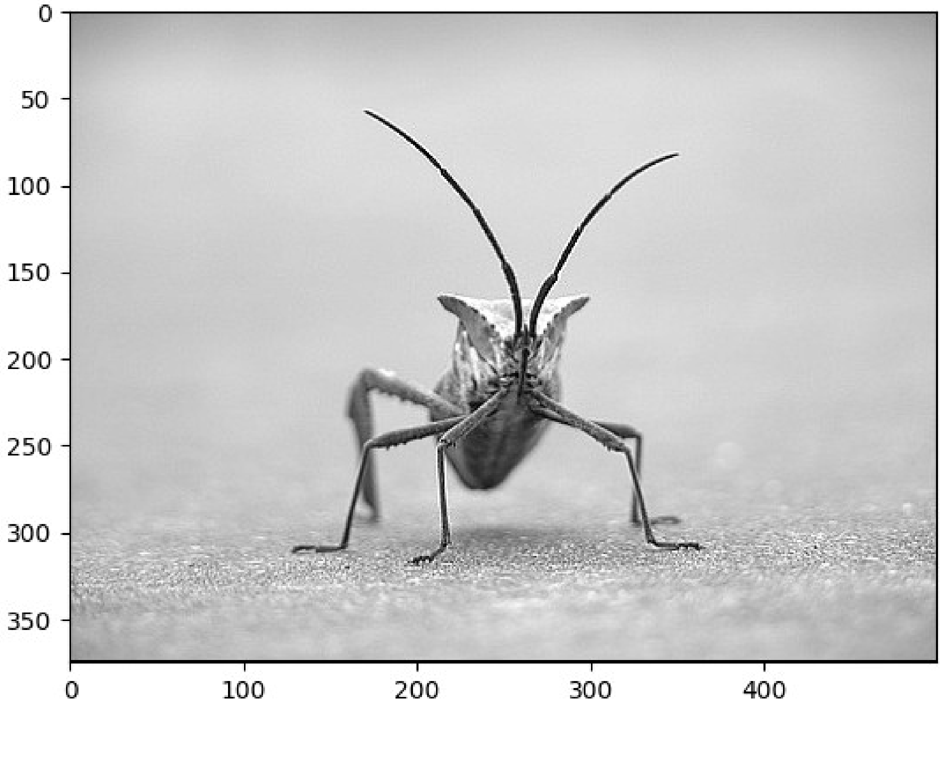
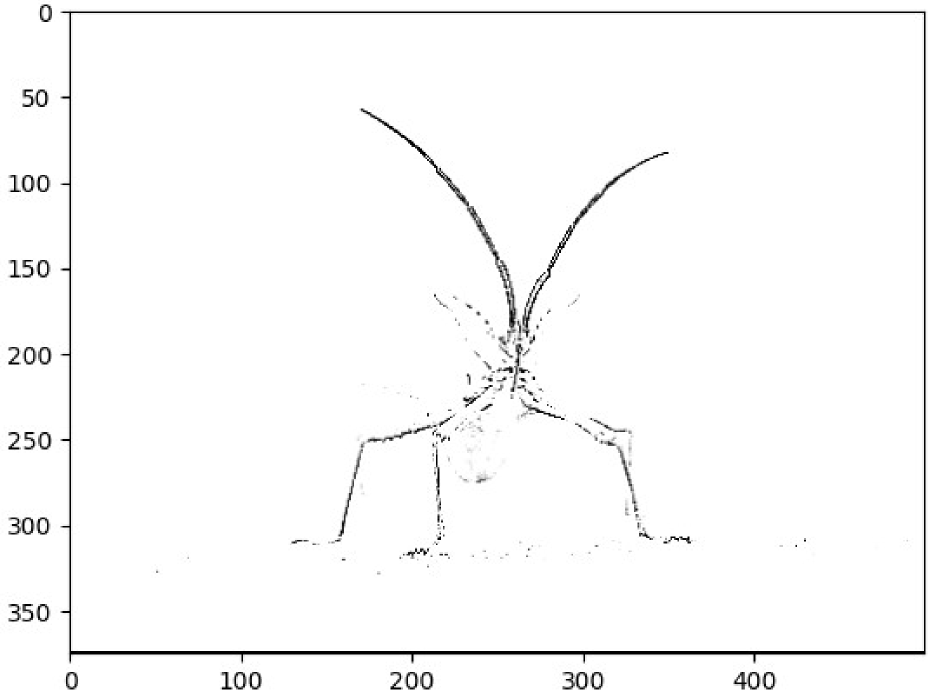
- Primeira imagem:



β = 1

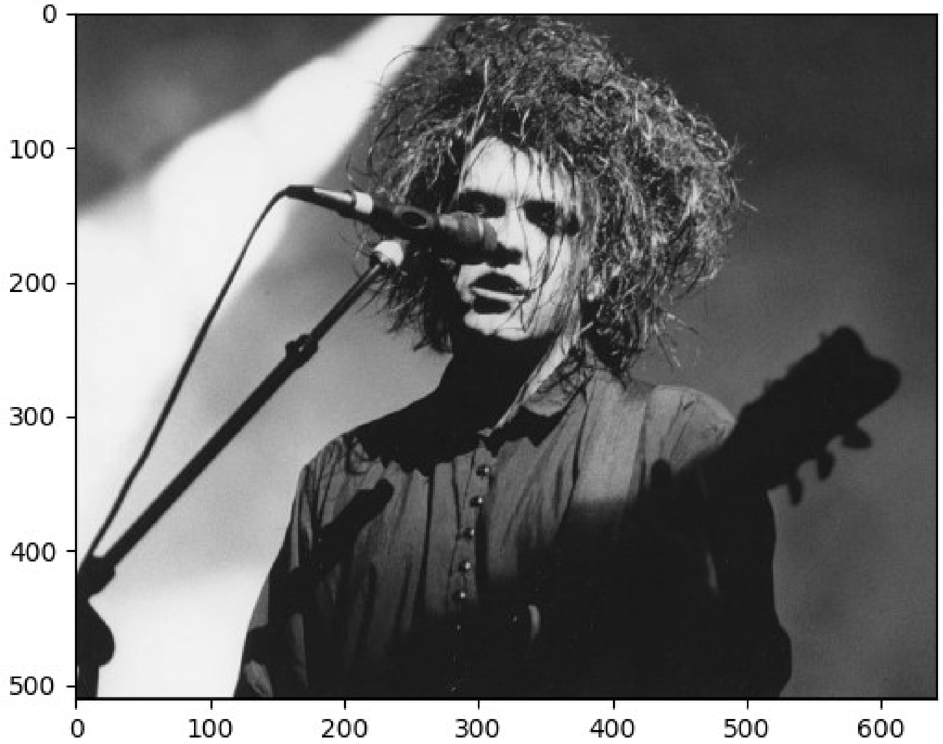


β = 5

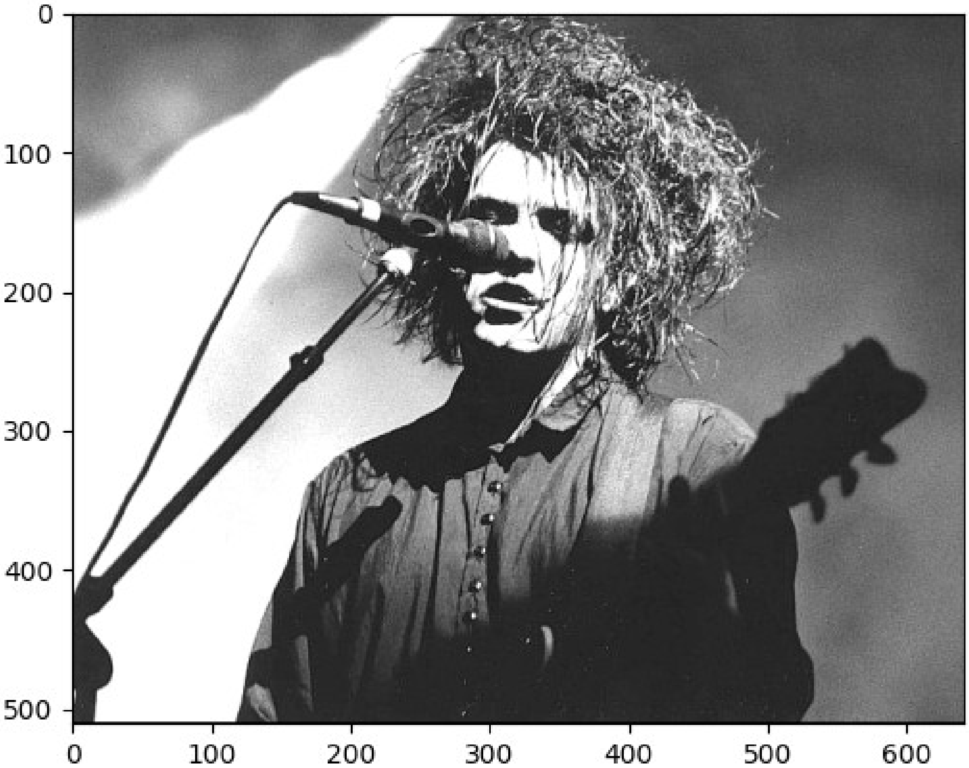


β = 50

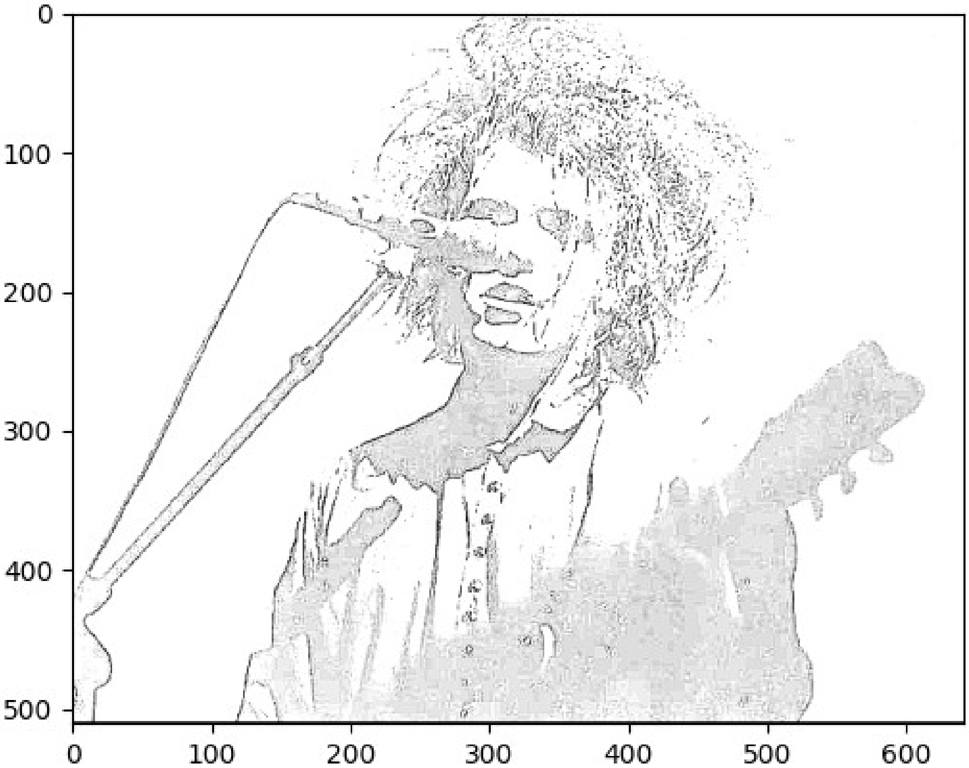
- Segunda imagem:



β = 1

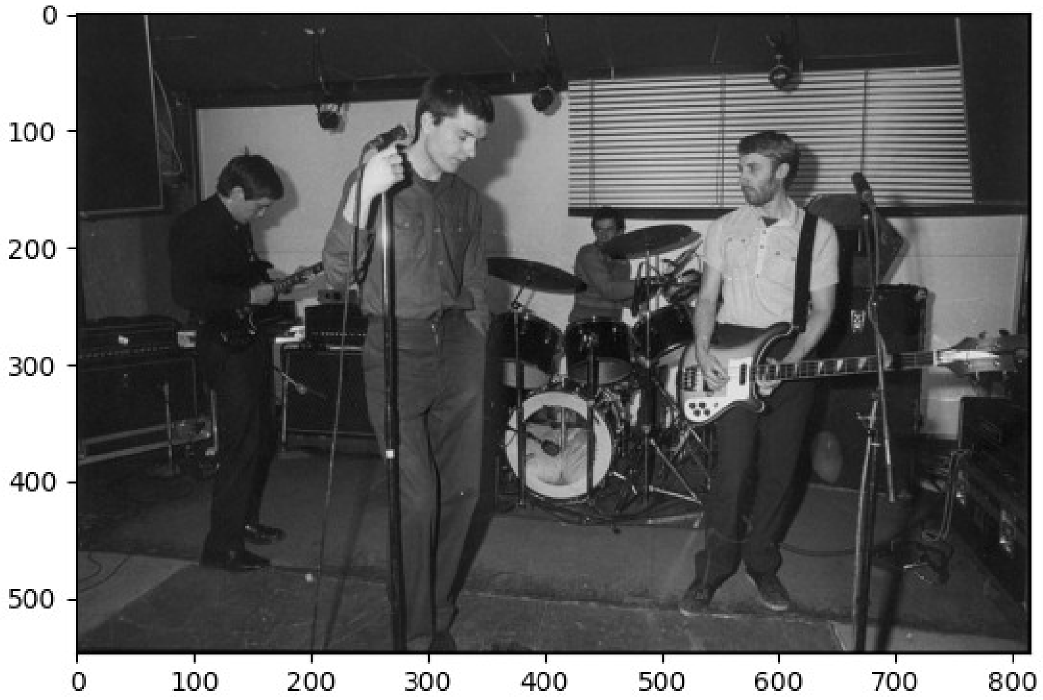

0
100
200
300
400
500
0
100
200
300
400
500
600

β = 5

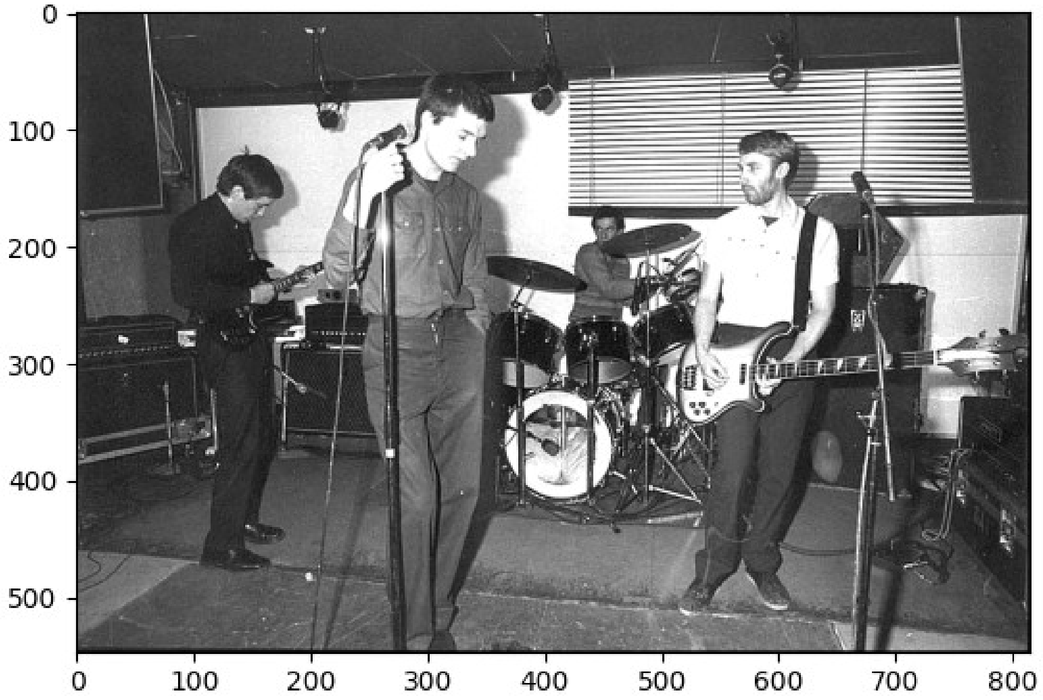

0
100
200
300
400
500
0
100
200
300
400
500
600

β = 50

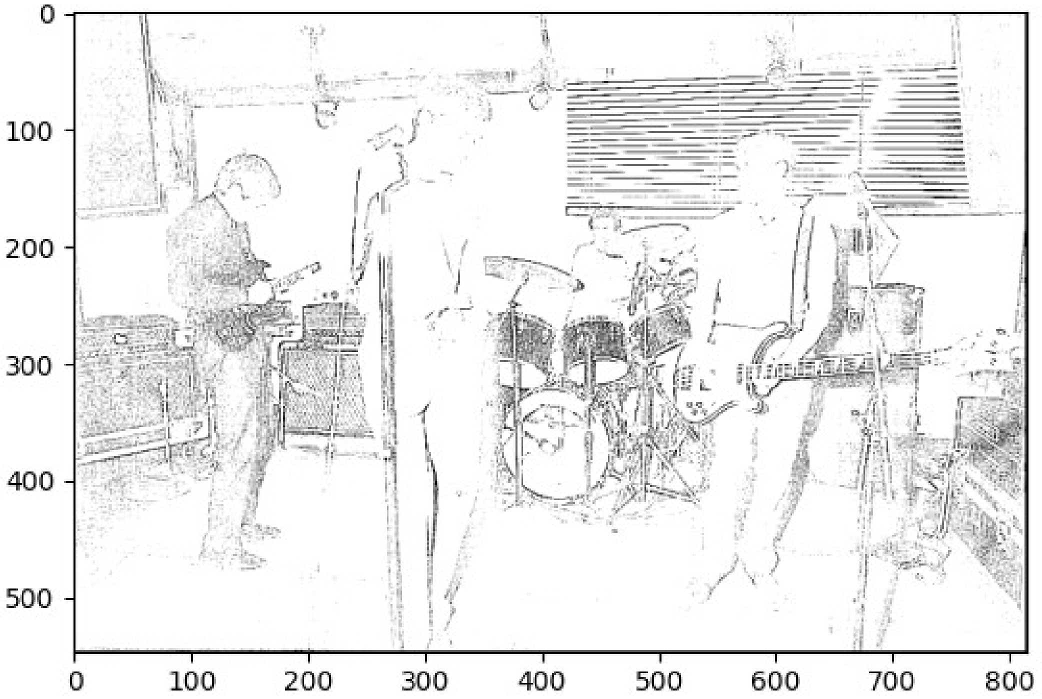
- Terceira imagem:



β = 1



β = 5



β = 50

```
#! /bin/python3

from math import e, floor, pi
from matplotlib import pyplot as plt
import cv2 as cv
import numpy as np

def sharpen(image, filter_shape, sigma, beta, file_name):
    imag = cv.imread(image)

    mdimF = floor(filter_shape/2)

    img = np.zeros([imag.shape[0]+2*mdimF, imag.shape[1]+2*mdimF], dtype =
float)

    #grayscale image conversion
    for i in range(imag.shape[0]):
        for j in range(imag.shape[1]):
            for k in range(2):
                img[i+mdimF][j+mdimF] += imag[i][j][k]
            img[i+mdimF][j+mdimF] /= 3

    #filters
    blur = np.zeros([filter_shape, filter_shape], dtype = float)
    norm = np.zeros([filter_shape, filter_shape], dtype = float)
    norm[mdimF][mdimF] = 1
    edge = np.zeros([filter_shape, filter_shape], dtype = float)
    shar = np.zeros([filter_shape, filter_shape], dtype = float)

    #gaussian blur parameters
    g1 = 1/(2*pi*sigma)
    sig2 = sigma**2
    g2 = g1*pi

    for i in range(filter_shape):
        for j in range(filter_shape):
            blur[i][j] = g1*e**(-(((i-mdimF)**2)+((j-mdimF)**2))*g2)
            edge[i][j] = norm[i][j] - blur[i][j]
            shar[i][j] = norm[i][j] + beta*edge[i][j]

    sharp = np.zeros([imag.shape[0], imag.shape[1]], dtype = float)

    #convolution
    for i in range(mdimF, imag.shape[0]-mdimF):
        for j in range(mdimF, imag.shape[1]-mdimF):
            for k in range(filter_shape):
                for l in range(filter_shape):
                    sharp[i][j] += shar[k][l] * img[i+k][j+l]
            if sharp[i][j] > 255:
                sharp[i][j] = 255
            if sharp[i][j] < 0:
                sharp[i][j] = 0

    plt.imshow(sharp, cmap = 'gray')
    plt.savefig(file_name)
    return
```