Nome: Fernando Bastos Arbache
RA: 31806211

Nome: Hugo Shiniti Nishimoto
RA: 32048416

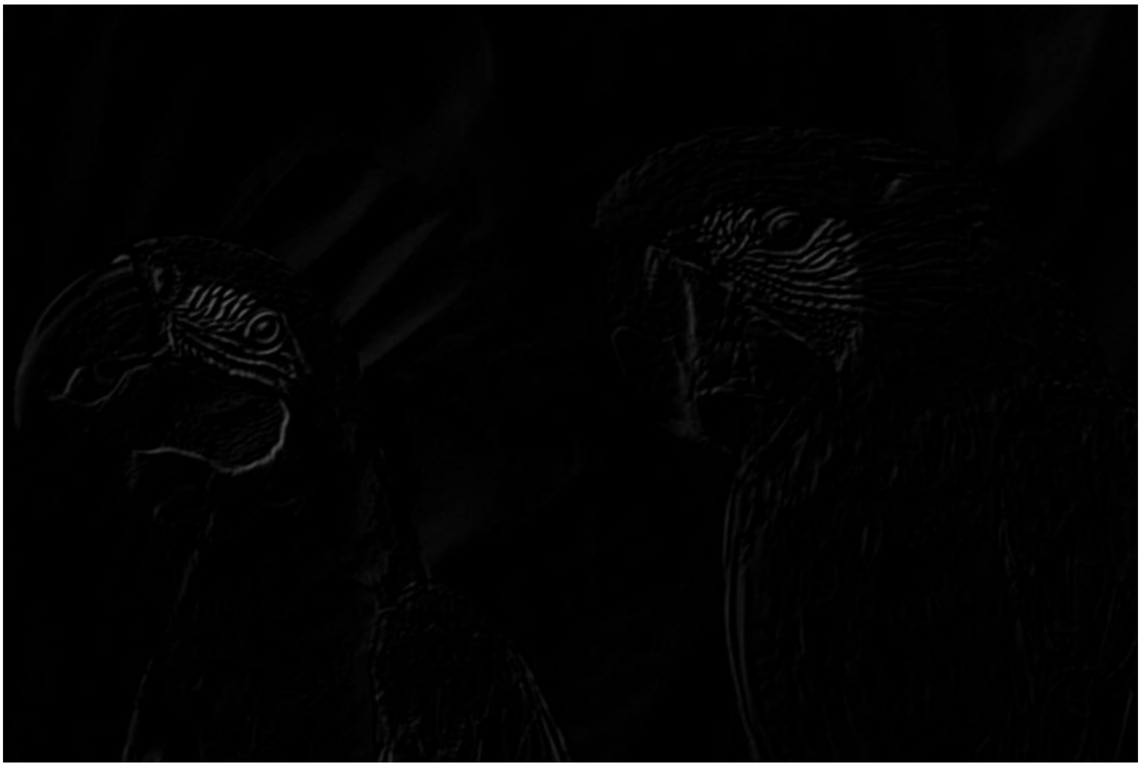
Nome: Lucas Souza Silva
RA: 32058179

Nome: Vinicius Rodrigues Figueira de Faria
RA: 32013639

Atividade Semana 10 – Computação Visual

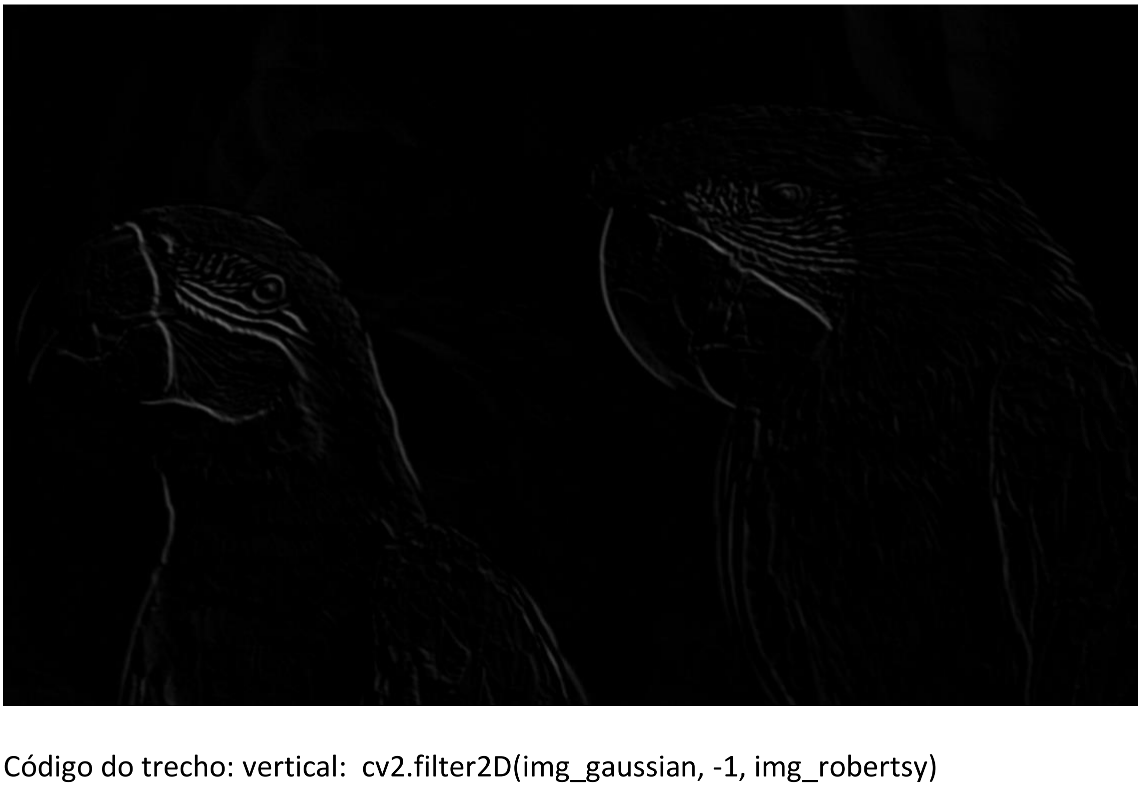
Resolução:

Roberts – Eixo Horizontal



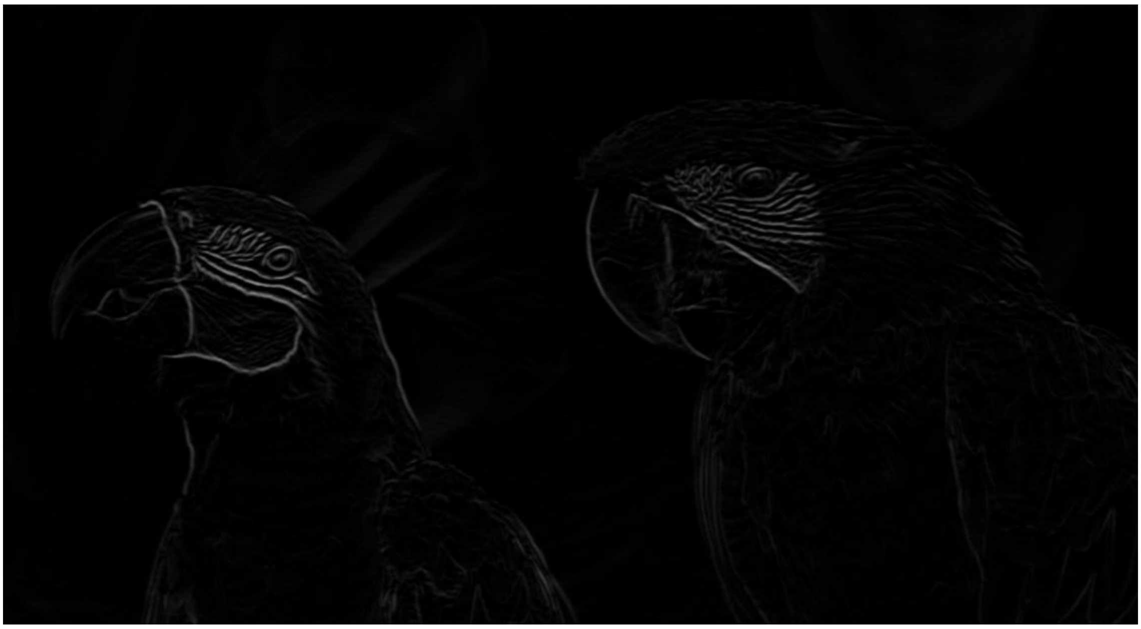
Código do trecho: horizontal: cv2.filter2D(img_gaussian, -1, img_robertsx)

Roberts – Eixo Vertical



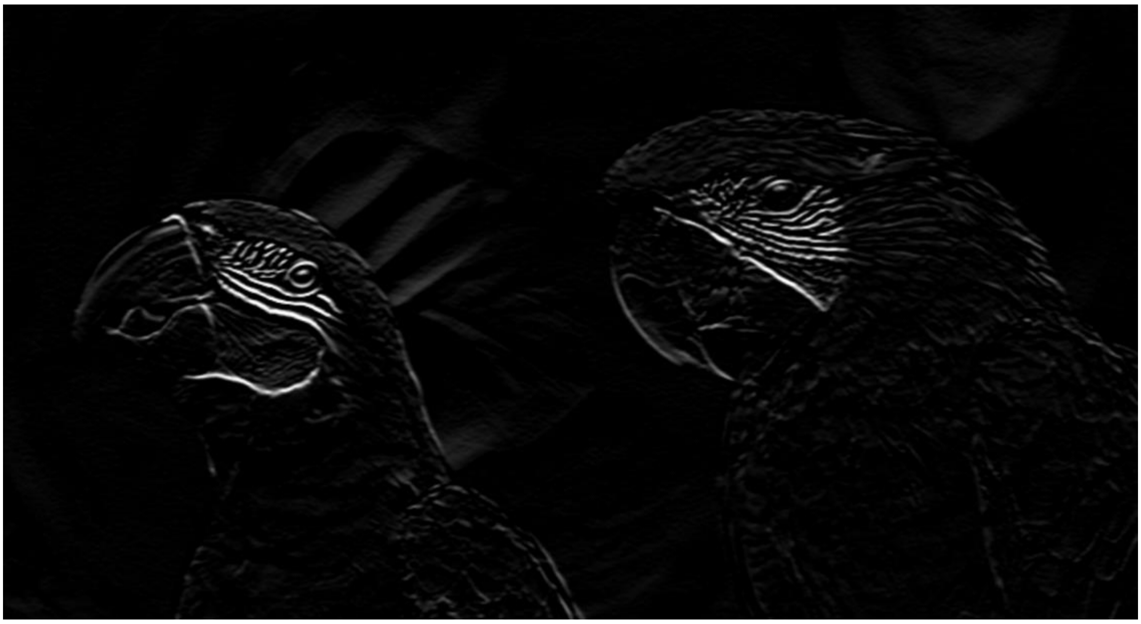
Código do trecho: vertical: cv2.filter2D(img_gaussian, -1, img_robertsy)

Roberts - Resultante



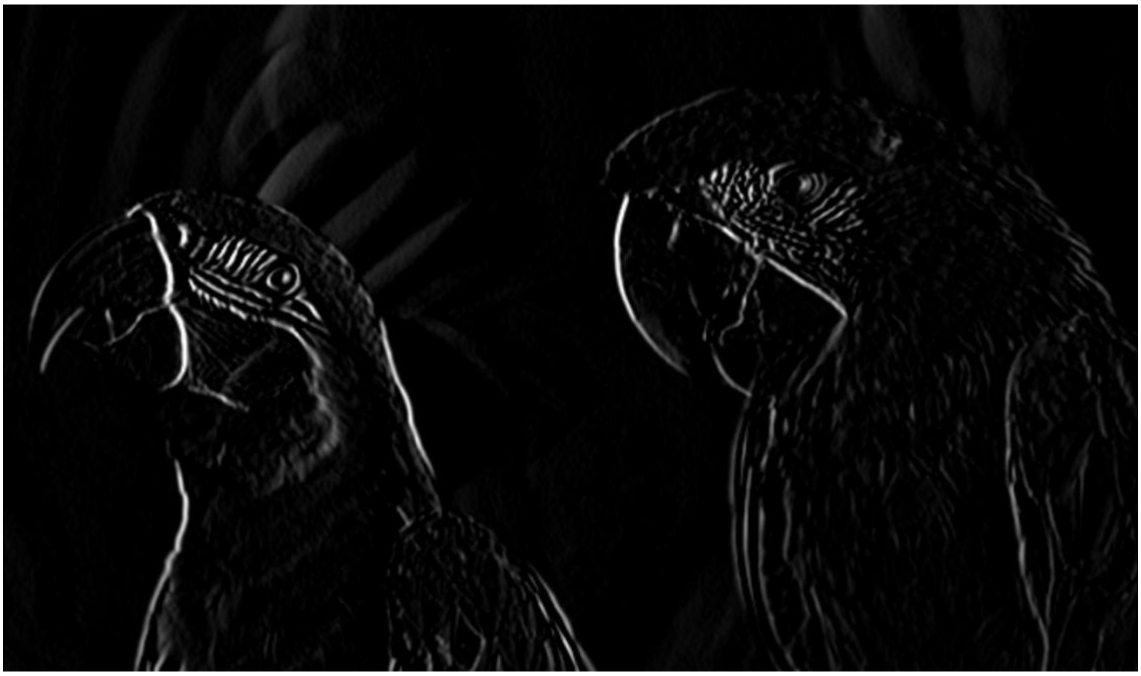
Código do trecho: img_roberts = horizontal + vertical

Prewitt – Eixo Horizontal



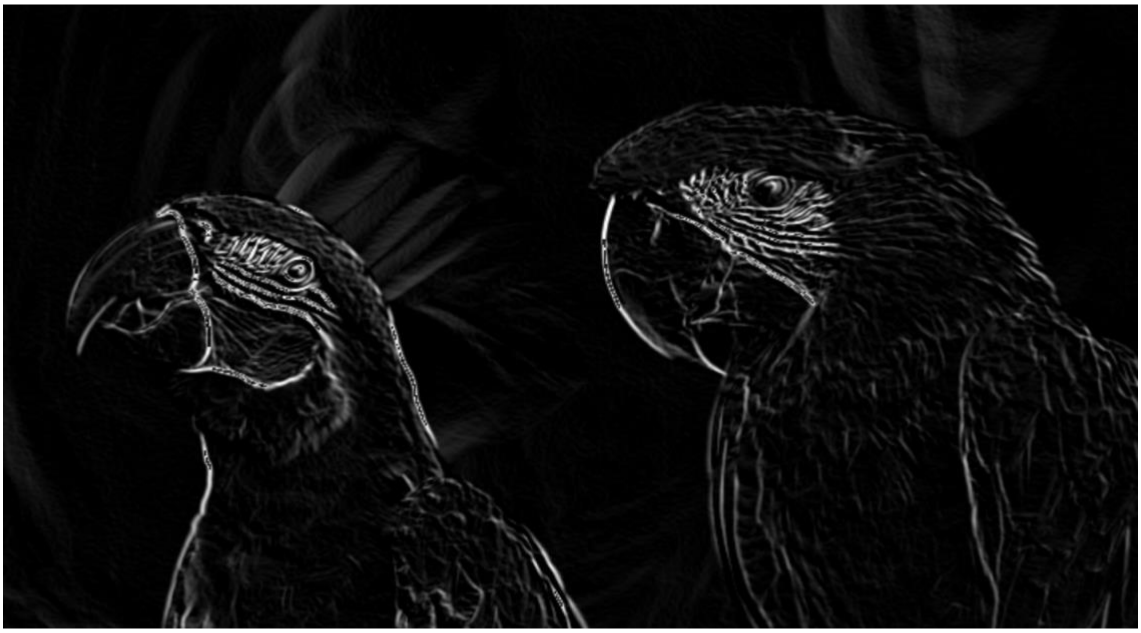
Código do trecho: img_prewittx = cv2.filter2D(img_gaussian, -1, kernelx)

Prewitt – Eixo Vertical



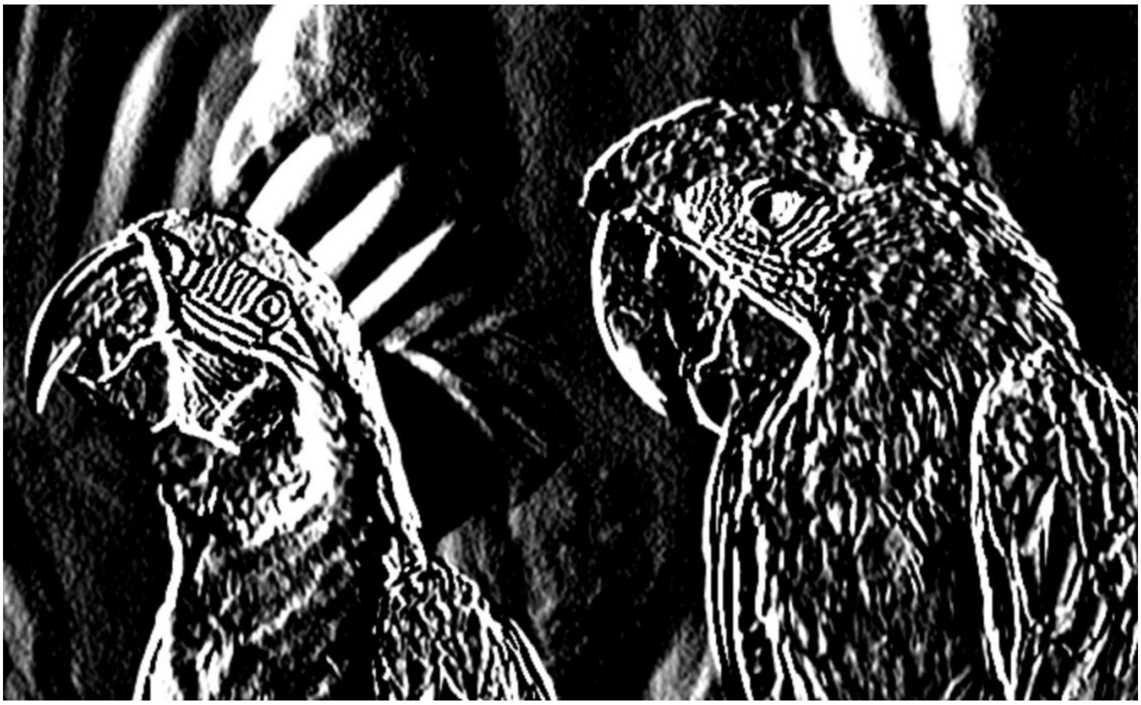
Código do trecho: img_prewitty = cv2.filter2D(img_gaussian, -1, kernely)

Prewitt – Resultante



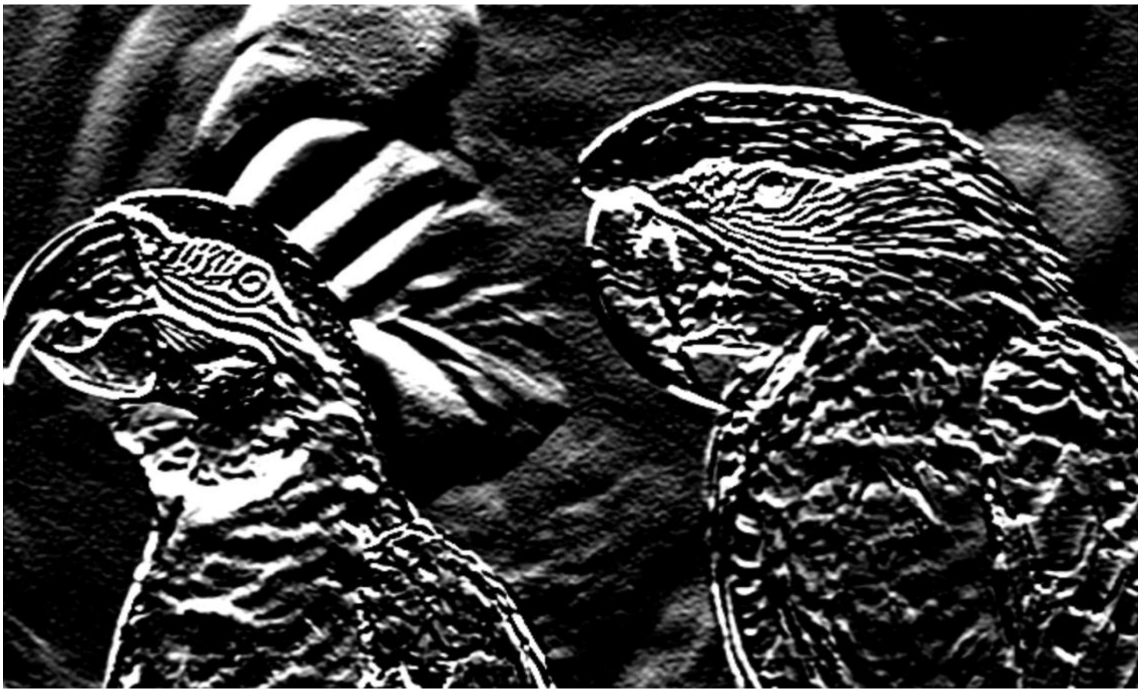
Trecho do código: img_prewitt = img_prewittx + img_prewitty

Sobel – Eixo Horizontal



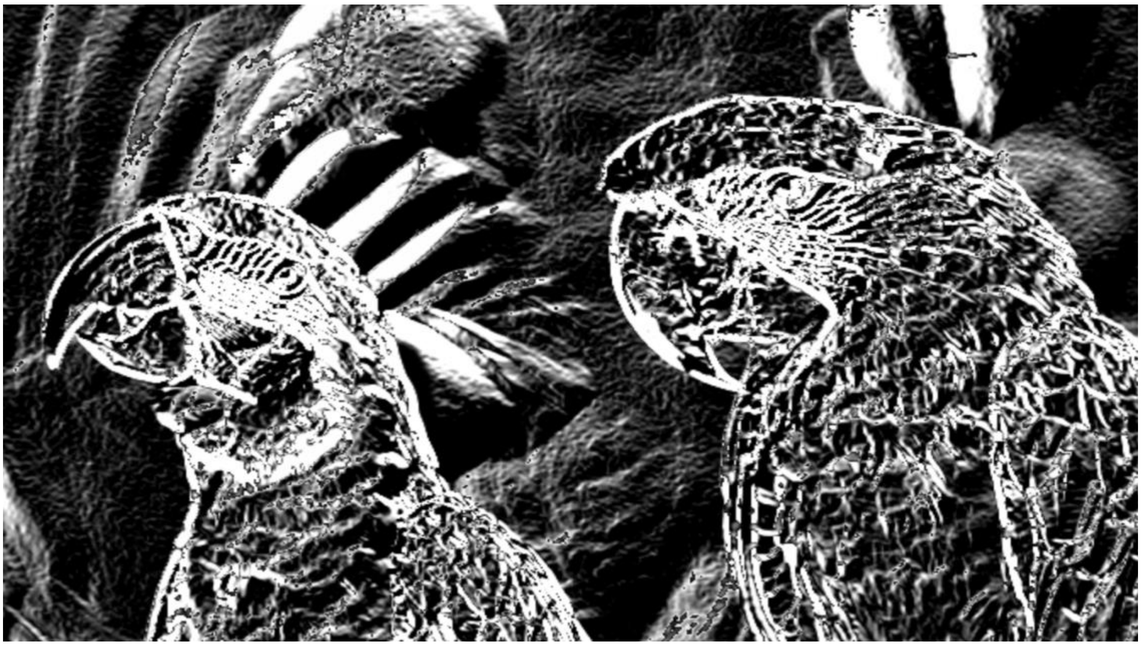
Trecho do código: horizontal = cv2.filter2D(img_gaussian, -1, img_robertsx)

Sobel – Eixo Vertical



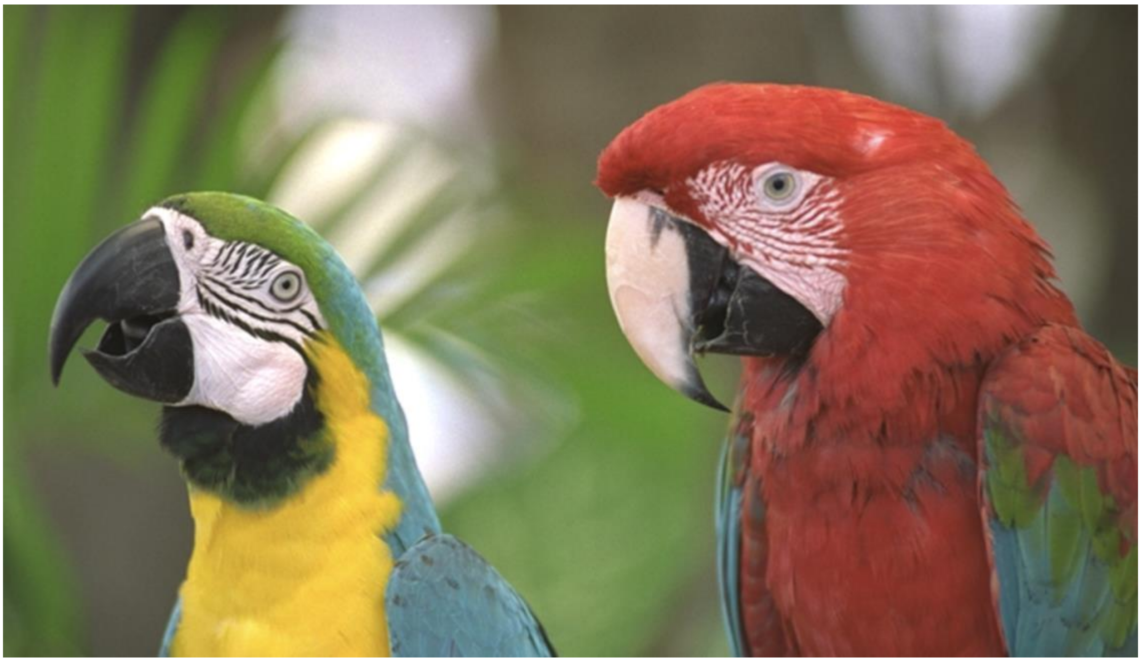
Trecho do código: vertical = cv2.filter2D(img_gaussian, -1, img_robertsy)

Sobel – Resultante

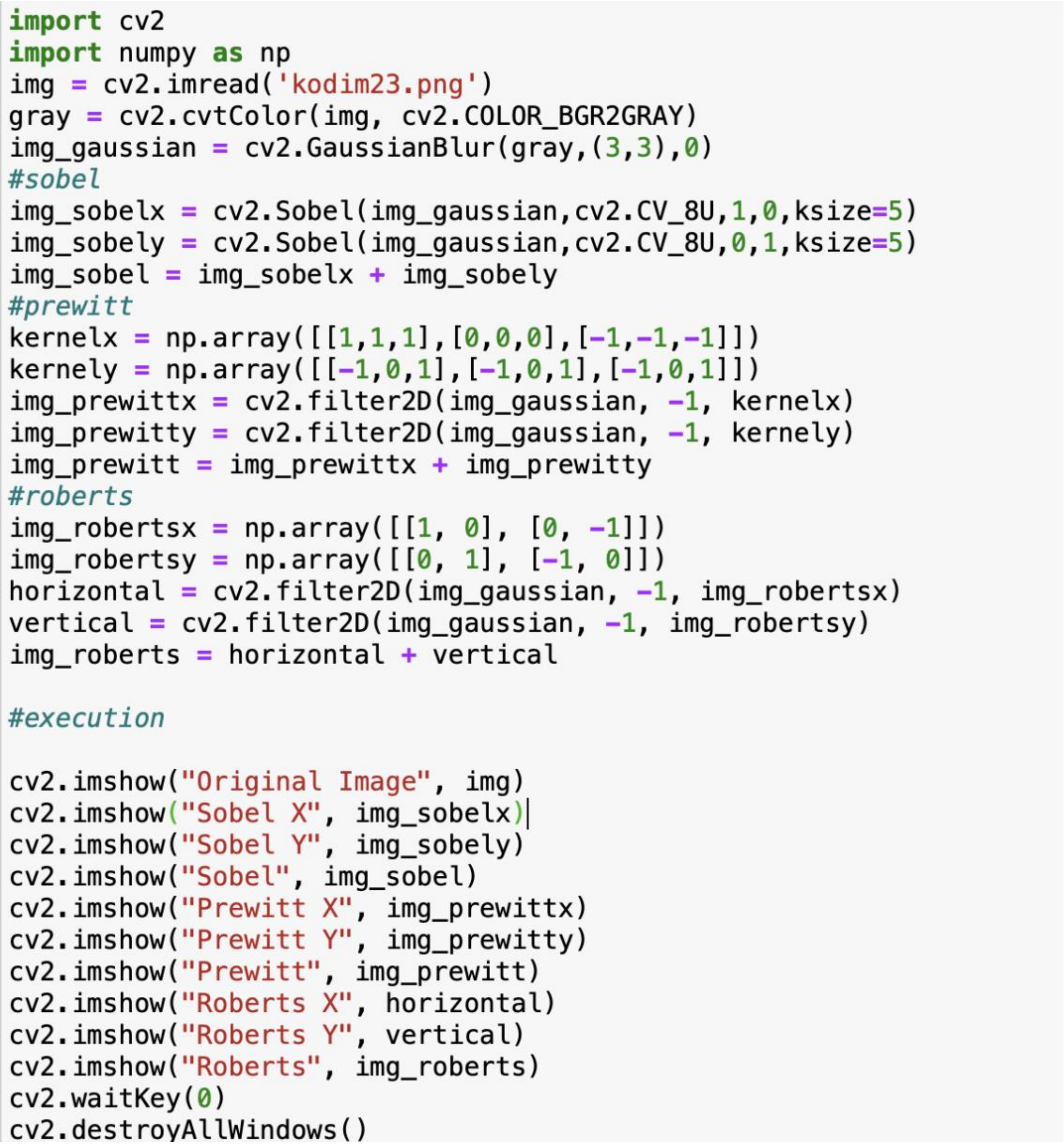


Trecho do código: img_roberts = horizontal + vertical

Imagem Original (Roberts, Prewitt, Sobel)



Código fonte:



```
import cv2
import numpy as np
img = cv2.imread('kodim23.png')
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
img_gaussian = cv2.GaussianBlur(gray,(3,3),0)
#sobel
img_sobelx = cv2.Sobel(img_gaussian,cv2.CV_8U,1,0,ksize=5)
img_sobely = cv2.Sobel(img_gaussian,cv2.CV_8U,0,1,ksize=5)
img_sobel = img_sobelx + img_sobely
#prewitt
kernelx = np.array([[1,1,1],[0,0,0],[-1,-1,-1]])
kernely = np.array([[-1,0,1],[-1,0,1],[-1,0,1]])
img_prewittx = cv2.filter2D(img_gaussian, -1, kernelx)
img_prewitty = cv2.filter2D(img_gaussian, -1, kernely)
img_prewitt = img_prewittx + img_prewitty
#roberts
img_robertsx = np.array([[1, 0], [0, -1]])
img_robertsy = np.array([[0, 1], [-1, 0]])
horizontal = cv2.filter2D(img_gaussian, -1, img_robertsx)
vertical = cv2.filter2D(img_gaussian, -1, img_robertsy)
img_roberts = horizontal + vertical

#execution

cv2.imshow("Original Image", img)
cv2.imshow("Sobel X", img_sobelx)
cv2.imshow("Sobel Y", img_sobely)
cv2.imshow("Sobel", img_sobel)
cv2.imshow("Prewitt X", img_prewittx)
cv2.imshow("Prewitt Y", img_prewitty)
cv2.imshow("Prewitt", img_prewitt)
cv2.imshow("Roberts X", horizontal)
cv2.imshow("Roberts Y", vertical)
cv2.imshow("Roberts", img_roberts)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

Referências:

- https://gist.github.com/rahit/c078cabc0a48f2570028bff397a9e154

- https://scikit-image.org/docs/stable/auto_examples/edges/plot_edge_filter.html

- B. N, M. Afsar, K. K. Khaitan, Rahul and C. Gururaj, "Optimized FPGA Implementation and Synthesis of Image Segmentation Techniques," *2021 IEEE Mysore Sub Section International Conference (MysuruCon)*, Hassan, India, 2021, pp. 191-196, doi: 10.1109/MysuruCon52639.2021.9641613.