INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO, CÂMPUS BIRIGUI - SP
BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

ISADORA DISPOSTI BUENO DOS SANTOS

EXERCÍCIOS – DESAFIO STANFORD

BIRIGUI - SP
2023

- https://web.stanford.edu/class/ee368/Handouts/Lectures/Examples/3-Combining-Images/Defect_Detection/
    - Código:

```
import cv2
import numpy as np
import matplotlib.pyplot as plt

# Load test images
origImg = cv2.imread('./imagem/pcbCropped.png',
cv2.IMREAD_GRAYSCALE).astype(np.float64) / 255.0
defectImg =
cv2.imread('./imagem/pcbCroppedTranslatedDefected.p
ng', cv2.IMREAD_GRAYSCALE).astype(np.float64) /
255.0

# Perform shift
xShift = 10
yShift = 10
row, col = origImg.shape
registImg = np.zeros_like(defectImg)
registImg[yShift:row, xShift:col] = defectImg[0:row
- yShift, 0:col - xShift]

# Show difference images
diffImg1 = np.abs(origImg - defectImg)
plt.subplot(1, 3, 1), plt.imshow(diffImg1,
cmap='gray'), plt.title('Unaligned Difference
Image')

diffImg2 = np.abs(origImg - registImg)
plt.subplot(1, 3, 2), plt.imshow(diffImg2,
cmap='gray'), plt.title('Aligned Difference Image')

bwImg = diffImg2 > 0.15
```

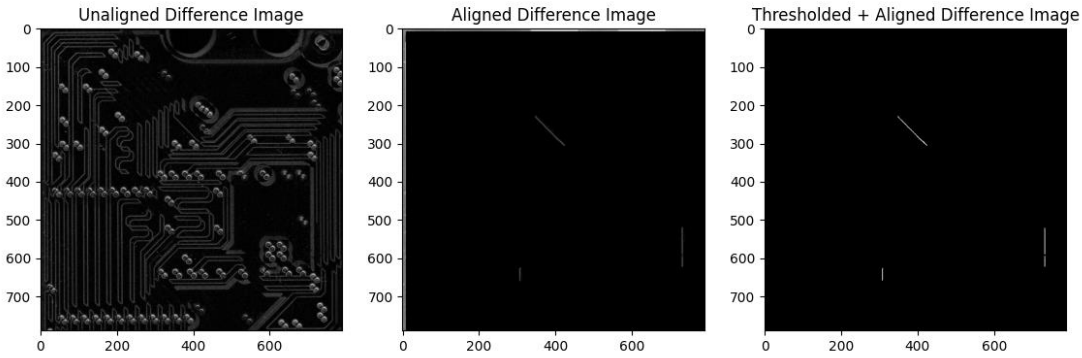
```
height, width = bwImg.shape
border = round(0.05 * width)
borderMask = np.zeros_like(bwImg)
borderMask[border:height-border, border:width-
border] = 1
bwImg = bwImg * borderMask

plt.subplot(1, 3, 3), plt.imshow(bwImg,
cmap='gray'), plt.title('Thresholded + Aligned
Difference Image')

# Save images
cv2.imwrite('Defect_Detection_diff.png', diffImg1 *
255.0)
cv2.imwrite('Defect_Detection_diffRegisted.png',
diffImg2 * 255.0)
cv2.imwrite('Defect_Detection_bw.png', bwImg *
255.0)

plt.show()
```

- Demonstração:



- https://web.stanford.edu/class/ee368/Handouts/Lectures/Examples/3-Combining-Images/Background_Subtraction/
  - Código:

```
import cv2
import numpy as np

# Read video
vrObj =
cv2.VideoCapture('./imagem/surveillance.mpg')
frame_width = int(vrObj.get(3))
frame_height = int(vrObj.get(4))
vwObj =
cv2.VideoWriter('Background_Subtraction.avi',
cv2.VideoWriter_fourcc(*'MJPG'), 30, (frame_width,
frame_height))

nFrames = int(vrObj.get(cv2.CAP_PROP_FRAME_COUNT))

# Perform background accumulation and subtraction
alpha = 0.95
theta = 0.1

_, background = vrObj.read()
background = cv2.cvtColor(background,
cv2.COLOR_BGR2GRAY).astype(np.float64) / 255.0

while True:
    ret, frame = vrObj.read()
    if not ret:
        break

    currImg = cv2.cvtColor(frame,
cv2.COLOR_BGR2GRAY).astype(np.float64) / 255.0
    background = alpha * background + (1 - alpha) *
currImg
    diffImg = np.abs(currImg - background)
```

```
    threshImg = (diffImg > theta).astype(np.uint8)
* 255

    cv2.imshow('New frame', currImg)
    cv2.imshow('Background frame', background)
    cv2.imshow('Difference image', diffImg)
    cv2.imshow('Thresholded difference image',
threshImg)

    vwObj.write(cv2.cvtColor(frame,
cv2.COLOR_BGR2RGB))

    if cv2.waitKey(1) & 0xFF == ord('q'):
        break

vrObj.release()
vwObj.release()
cv2.destroyAllWindows()

# Save images
cv2.imwrite('Background_Subtraction_curr.png',
currImg * 255.0)
cv2.imwrite('Background_Subtraction_background.png'
, background * 255.0)
cv2.imwrite('Background_Subtraction_thresh.png',
threshImg)
```

- Demonstração: