

 <p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA SÃO PAULO Campus Birigui</p>	<p>INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA Campus Birigui</p> <p>Bacharelado em Engenharia de Computação</p>	
<p>Disciplina: Processamento Digital de Imagens</p>	<p>Desafio Stanford</p>	
<p>Professor: Prof. Dr. Murilo Vargas da Silva</p>	<p>Data: 14/09/2023</p>	
<p>Nome do aluno: Leonardo Reneres dos Santos</p>	<p>Prontuário: BI3009131</p>	

Questões:

Objetivo: Implementar códigos que utilizam operações básicas combinando duas imagens.

- Verificação de defeitos em placas: Basicamente realizando uma operação de subtração entre uma imagem de uma placa sem defeito com uma placa com defeito é possível encontrar defeitos no processo de fabricação:

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

- Detecção de movimento: A partir de um vídeo, ao realizar a subtração do fundo da cena sem nenhuma pessoa é possível detectar movimentos:

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

Resposta:

Código:

1.

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

```
# Load test images

origImg = cv2.imread('pcbCropped.png', cv2.IMREAD_GRAYSCALE) / 255.0

defectImg = cv2.imread('pcbCroppedTranslatedDefected.png',
cv2.IMREAD_GRAYSCALE) / 255.0


# Perform shift

rows, cols = origImg.shape

xShift = 10

yShift = 10

registImg = np.zeros(defectImg.shape)

registImg[yShift:rows, xShift:cols] = defectImg[0:rows-yShift,
0:cols-xShift]


# Show difference images

diffImg1 = abs(origImg - defectImg)

plt.subplot(1, 3, 1), plt.imshow(diffImg1, cmap='gray'),
plt.title('Unaligned Difference Image')

diffImg2 = 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((height, width))

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')

plt.show()

# Save images

cv2.imwrite('Defect_Detection_diff.png', diffImg1 * 255)

cv2.imwrite('Defect_Detection_diffRegistered.png', diffImg2 * 255)

cv2.imwrite('Defect_Detection_bw.png', bwImg * 255)

```

2.

```

import cv2

# Crie um objeto VideoCapture para ler o vídeo.

cap = cv2.VideoCapture('surveillance.avi')

# Verifique se o vídeo foi aberto corretamente

if not cap.isOpened():

    print("Erro ao abrir o vídeo")


frame_width = int(cap.get(3))

frame_height = int(cap.get(4))

frame_rate = 30

# Perform background accumulation and subtraction

```

```
# background = background.astype(float)

# Defina o codec de vídeo e crie um objeto VideoWriter
fourcc = cv2.VideoWriter_fourcc(*'MJPG')

out = cv2.VideoWriter('output1.avi', fourcc, frame_rate, (frame_width,
frame_height), 0)

alpha = 0.95

theta = 0.1

ret, frame = cap.read()

background = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

while cap.isOpened():

    # Leia o próximo frame do vídeo

    ret, frame = cap.read()

    if ret:

        # Converta o frame para escala de cinza

        imgcur = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)

        # Converta o frame para float

        #imgcur = imgcur.astype(float)

        # Atualize o background

        background = alpha * background + (1 - alpha) * imgcur

        # Calcule a diferença entre o frame atual e o background

        diffImg = abs(imgcur - background)
```

```
# Aplique o limiar

ret, threshImg = cv2.threshold(diffImg, theta, 255,
cv2.THRESH_BINARY)

# Escreva o frame no arquivo de saída


# Exiba o frame na janela

cv2.imshow('Frame', frame)

cv2.imshow('Background', background.astype('uint8'))

cv2.imshow('Difference', diffImg.astype('uint8'))

cv2.imshow('Thresholded Difference', threshImg.astype('uint8'))

frame2 = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)


out.write(diffImg.astype('uint8'))

# Saia se a tecla 'q' for pressionada

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

    break

else:

    break


# Libere os recursos e feche as janelas

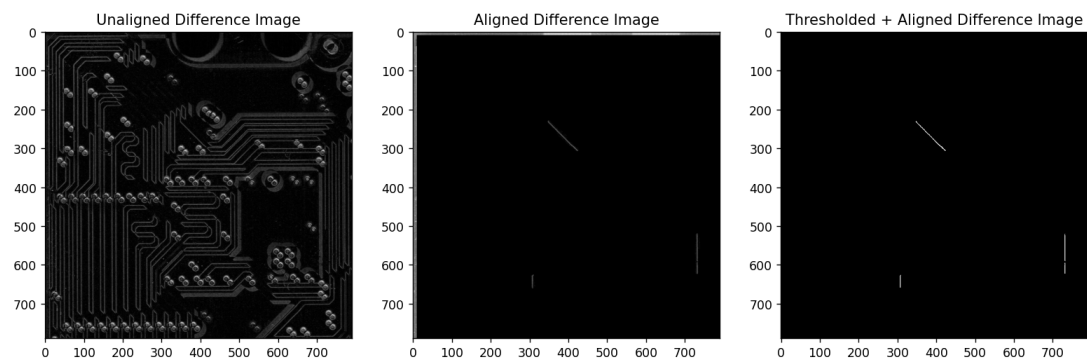
cap.release()

out.release()

cv2.destroyAllWindows()
```

Resultados:

1.



2.

O arquivo salvo é o video plotado em difference:

