

Desafio de Stanford

Detecção de variações em PCBs

Código:

```
import cv2
import numpy as np

placa_sem defeito = cv2.imread('pcbCroppedTranslated.png')
placa_com defeito =
cv2.imread('pcbCroppedTranslatedDefected.png')

if placa_sem defeito is None or placa_com defeito is None:
    print("Erro ao carregar as imagens.")
else:

    if placa_sem defeito.shape == placa_com defeito.shape:

        diferenca = cv2.absdiff(placa_sem defeito,
placa_com defeito)

        diferenca_em_escala_de_cinza = cv2.cvtColor(diferenca,
cv2.COLOR_BGR2GRAY)

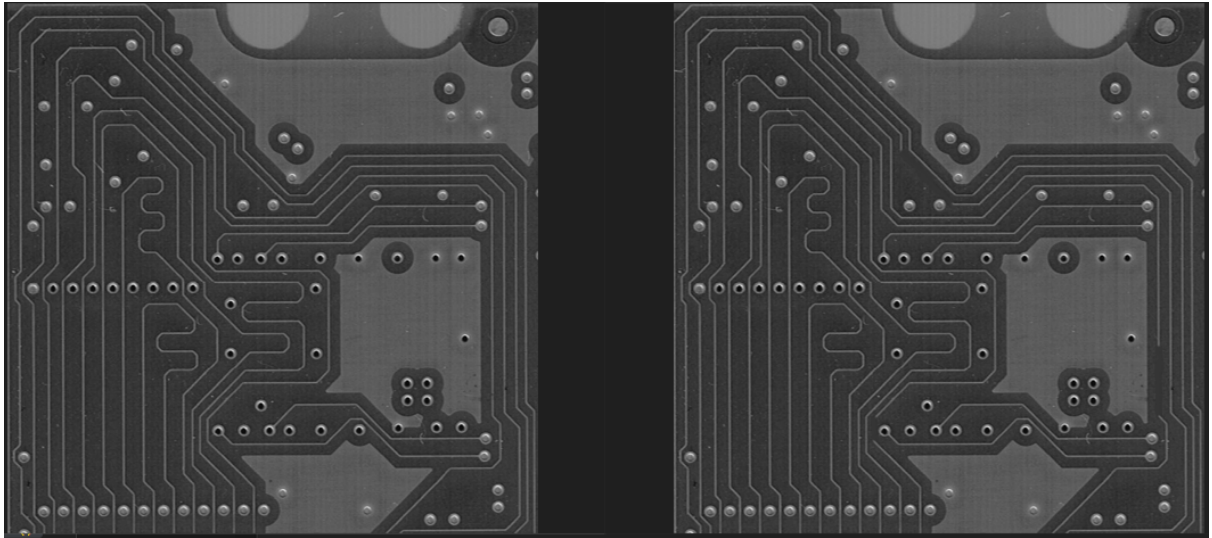
        _, thresholded =
cv2.threshold(diferenca_em_escala_de_cinza, 30, 255,
cv2.THRESH_BINARY)

        contornos, _ = cv2.findContours(thresholded,
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        resultado = cv2.drawContours(placa_com defeito.copy(),
contornos, -1, (0, 0, 255), 2)

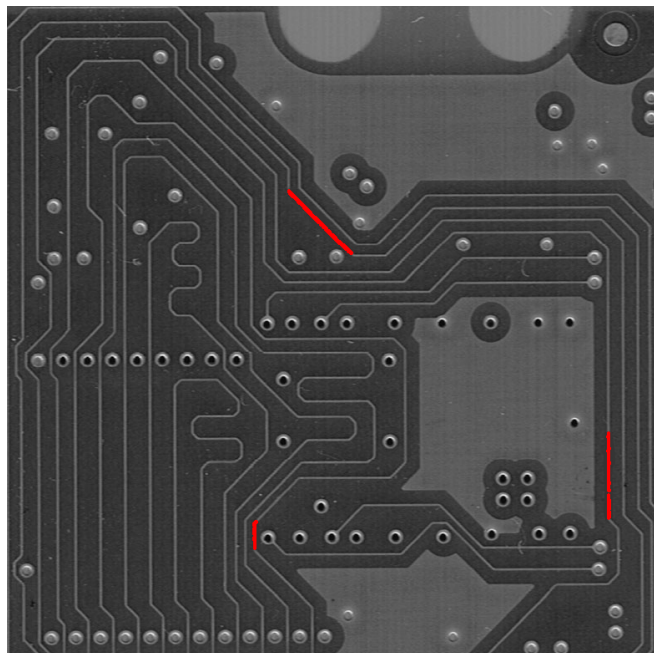
        cv2.imshow('Resultado', resultado)
        cv2.waitKey(0)
        cv2.destroyAllWindows()
    else:
```

```
print("As imagens têm dimensões diferentes.")
```

Imagens a serem comparadas:



Resultado:



Detecção de variações em vídeos

```
import cv2

def calcular_diferenca(frame_anterior, frame_atual):

    gray_anterior = cv2.cvtColor(frame_anterior, cv2.COLOR_BGR2GRAY)
    gray_atual = cv2.cvtColor(frame_atual, cv2.COLOR_BGR2GRAY)

    diff_frame = cv2.absdiff(gray_anterior, gray_atual)

    _, thresh = cv2.threshold(diff_frame, 30, 255, cv2.THRESH_BINARY)

    return thresh

video_capture = cv2.VideoCapture('output.avi')

ret, frame_anterior = video_capture.read()

while True:

    ret, frame_atual = video_capture.read()

    if not ret:
        break

    diff_frame = calcular_diferenca(frame_anterior, frame_atual)

    cv2.imshow('Detecção de Movimento', diff_frame)

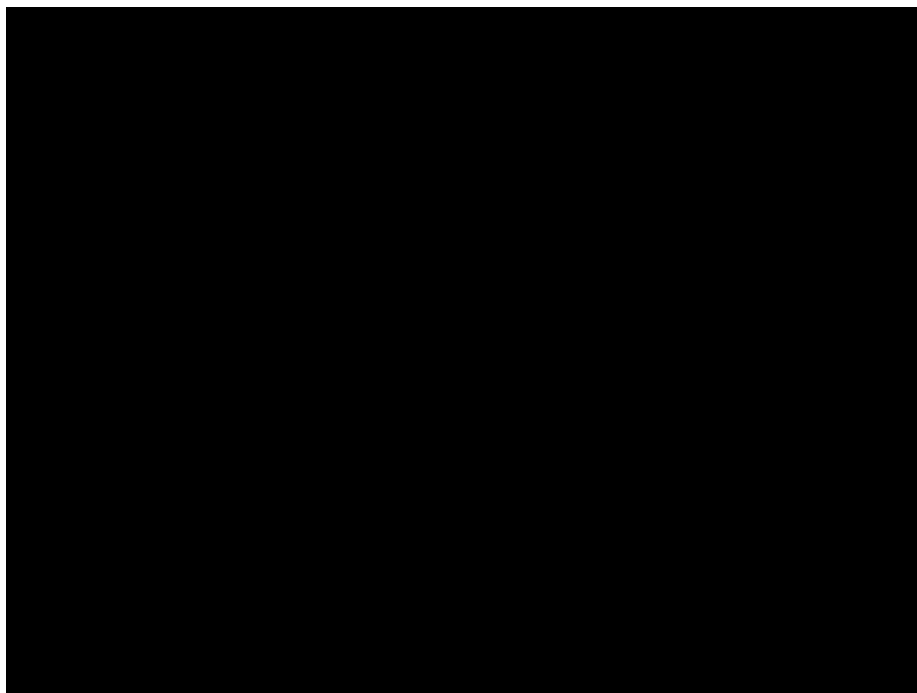
    frame_anterior = frame_atual

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

video_capture.release()
cv2.destroyAllWindows()
```

Resultado:

O vídeo sem movimento fica completamente preto:



Quando o movimento é detectado, as bordas ficam realçadas

