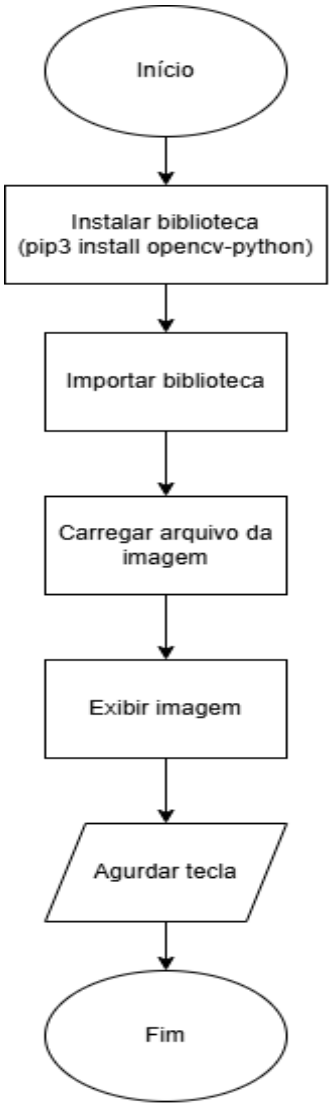
Relatório de progresso: Reconhecimento de Display 7 segmentos com OpenCV

06/06/2025

Visão geral

Instalação da biblioteca e primeiros passos para leitura do arquivo e exibição de imagem em linguagem Python.

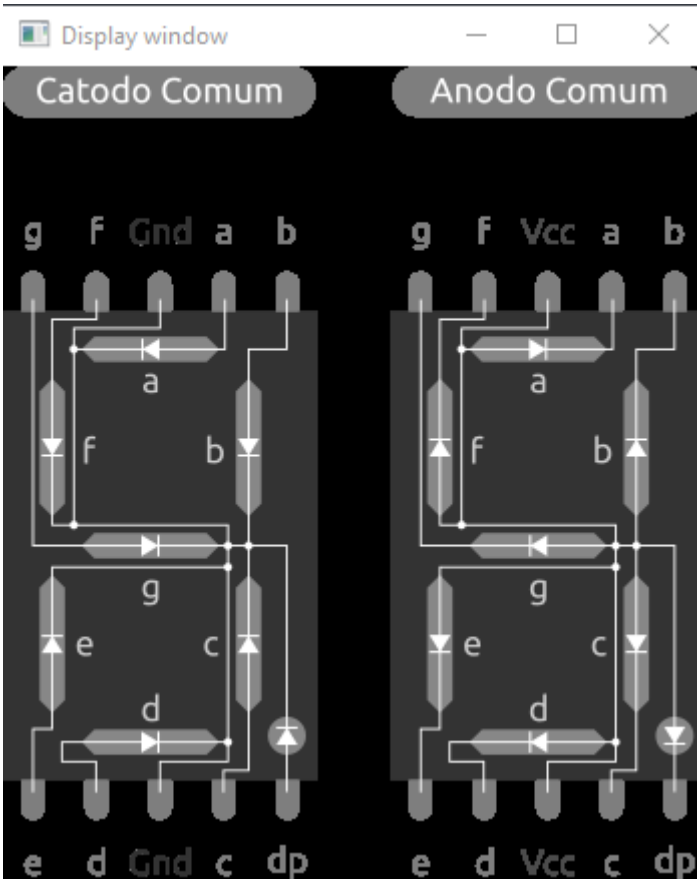
Fluxograma



Código

```python
import cv2 as cv

img = cv.imread("img/display.png", cv.IMREAD_GRAYSCALE)
cv.imshow("Display window", img)
k = cv.waitKey(0)
```

Saída



24/06/2025

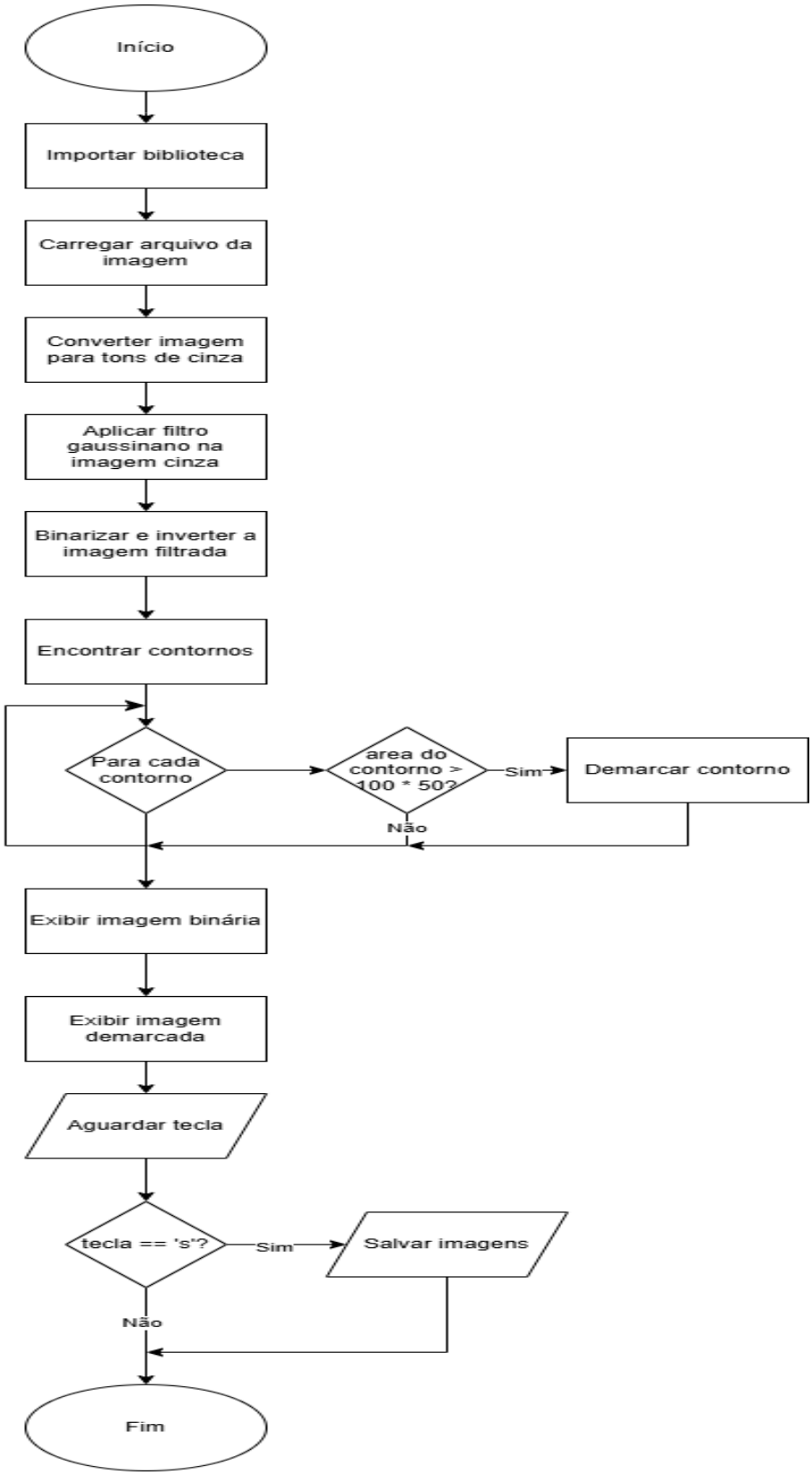
Visão Geral

Testes básicos de aplicação de filtros de cor e binarização da imagem para extração de dados.

Uma imagem amostral de um multímetro foi escolhida para a realização do procedimento. As etapas foram:

- Converter a imagem para tons de cinza
- Aplicar filtro gaussiano para “borrar” levemente a imagem e reduzir ruídos nas bordas
- Binarizar a imagem apenas para tons pixels pretos ou brancos utilizando um valor limiar de intensidade
- Inverter a imagem binária
- Encontrar contornos utilizando os métodos internos do OpenCV
- Filtrar os contornos encontrados (carece de ajuste)
- Demarcar contornos

Fluxograma

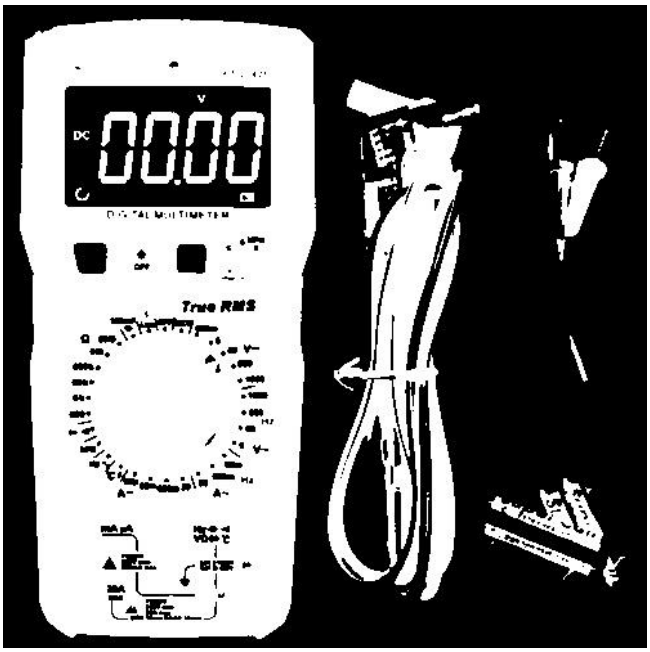

Início
Importar biblioteca
Carregar arquivo da imagem
Converter imagem para tons de cinza
Aplicar filtro gaussinano na imagem cinza
Binarizar e inverter a imagem filtrada
Encontrar contornos
Para cada contorno
area do contorno > 100 * 50?
Sim
Demarcar contorno
Não
Exibir imagem binária
Exibir imagem demarcada
Aguardar tecla
tecla == 's'?
Sim
Salvar imagens
Não
Fim

Código

```python
import cv2 as cv

# Carregar e redimensionar imagem
img_dir = r'.\img'
img = cv.imread(f"{img_dir}\\multimetro.jpg")
img = cv.resize(img, (500, 500))

# Converter imagem para preto e branco e aplicar filtro gaussiano
gray_img = cv.cvtColor(img, cv.COLOR_BGR2GRAY)
blurred_img = cv.GaussianBlur(gray_img, (3, 3), 0)

# Binarizar imagem
_, thresh = cv.threshold(blurred_img, 120, 255, cv.THRESH_BINARY_INV)

# Encontrar contornos
contours, _ = cv.findContours(thresh, cv.RETR_EXTERNAL, cv.CHAIN_APPROX_SIMPLE)

# Filtrar contornos através de uma aréa mínima para evitar pequenos ruídos (carece de ajustes)
for cnt in contours:
  area = cv.contourArea(cnt)
  if area > 100 * 50:
      x, y, w, h = cv.boundingRect(cnt)
      cv.rectangle(img, (x, y), (x + w, y + h), (0, 255, 0), 2)

cv.imshow("Imagem binarizada", thresh)
cv.imshow("Imagem com contorno", img)
k = cv.waitKey(0)

if chr(k) == 's':
  cv.imwrite(f'{img_dir}\\multimetro_binario.jpg', thresh)
  cv.imwrite(f'{img_dir}\\multimetro_demarcado.jpg', img)
```

Entrada

Saída




ET-2042F
DC
00.00
DIGITAL MULTIMETER
HOLD
OFF
True RMS
mA µA
20A
MAX
COM
TIANQIU