Reconhecimento de Calibre de Laranjas por Imagem

Alexandre Bruno
Eduardo Mayer
Marcio Antônio

UFPR
UNIVERSIDADE FEDERAL DO PARANÁ

HISTORICO DA LARANJA

UFPR 105 ANOS DE ORGULHO

LARANJAS

- Pós-colheita:
  - As menores são descartadas
  - As maiores recebem destino especial
  - As demais vão para o mercado

  Como selecioná-las?

  Como determinar a produção?

VISÃO COMPUTACIONAL

- OpenCV
  - HoughCircles()
  - Laranjas são aproximadamente esféricas

EXECUÇÃO DO PROGRAMA

```python
import math
import cv2
import numpy as np

#Inicializando as variáveis principais
fator = 0 #constante de conversão pixel -> mm (muda em cada imagem, pois depende da moeda de 0.25: referencial constante)
raio_medio = 0 #Soma dos raios das n laranjas medidos em mm
volume_medio = 0 #Soma dos volumes das n laranjas medidos em mL
n=0 #Total de laranjas analisadas

#Pedindo ao usuário o endereço da primeira imagem
address = input("Por favor, digite o endereco da imagem. Nao se esqueca da extensao nem de que o arquivo deve estar na mesma pasta que este programa: ")

#Verificar imagens até o usuário apertar Enter
while(address!=""):
    print() #Pula linha
    img = cv2.imread(address,0) #Guardando a imagem numa variável

    #Verificando se img guardou uma imagem com sucesso
    if isinstance(img,np.ndarray):

        #Redimensionar img de modo que a maior dimensão tenha 700 pixel. Assim, evita-se o uso de imagens muito grandes ou muito pequenas em
        #HoughCircles(), o que poderia causar erros.

        height, width = img.shape[:2]
        escala=700/max(height,width)
        img = cv2.resize(img,(round(escala*width), round(escala*height)), interpolation = cv2.INTER_AREA)

        #Tratamento de img para uso em funções futuras. A saber, HoughCircles(), circle() e imshow()
        img = cv2.medianBlur(img,5)
        cimg = cv2.cvtColor(img,cv2.COLOR_GRAY2BGR)

        #HoughCircles() procura por círculos em img e os guarda em circles. Parâmetros com Valores Padrão
        circles = cv2.HoughCircles(img,cv2.HOUGH_GRADIENT,1,100,
                                   param1=50,param2=60,minRadius=0,maxRadius=0)

        #Verificando se circles não está vazio, ou seja, se algum círculo foi encontrado
        if isinstance(circles,np.ndarray):
            #Verificando se alguma laranja foi encontrada além da moeda
```

UFPR 105 ANOS DE ORGULHO

EXECUÇÃO DO PROGRAMA

```
39          #Verificando se alguma laranja foi encontrada além da moeda
40          if len(circles[0])>1:
41
42              #Procura-se pelo círculo de menor raio. Considera-se esse círculo como uma moeda de referência de 25 centavos e 25 mm de diâmetro.
43              raio_ref = circles[0,0][0]
44              for i in circles[0,:]:
45                  if i[2]<raio_ref:
46                      raio_ref=i[2]
47              fator = 12.5/raio_ref #Uma vez que o raio da moeda é conhecido. Ela é usada para definir um fator de conversão pixel -> mm
48
49              #Aqui avaliam-se os círculos. Mostra-se para o usuário os raios e volumes estimados de cada laranja
50              #individualmente e soma-se esses valores aos totais para uso futuro.
51
52              #Também são desenhados em cimg os contornos e centros das circunferências das laranjas e da moeda.
53
54              for i in circles[0,:]:
55                  if i[2]==raio_ref:
56                      cv2.circle(cimg,(i[0],i[1]),i[2],(255,0,0),2) #Desenha a circunferência da moeda
57                      print("Moeda")
58                      print('raio: {0:.2f} mm'.format(i[2]*fator))
59                      print()
60                  else:
61                      cv2.circle(cimg,(i[0],i[1]),i[2],(0,255,0),2) #Desenha a circunferência da laranja
62                      n += 1
63                      raio_medio += i[2]*fator
64                      volume_medio += 4*math.pi*((i[2]*fator)**3)/3000
65                      print("Laranja {0}".format(n))
66                      print('raio: {0:.2f} mm'.format(i[2]*fator))
67                      print('volume: {0:.2f} mL'.format(4*math.pi*((i[2]*fator)**3)/3000))
68                      print()
69                  cv2.circle(cimg,(i[0],i[1]),2,(0,0,255),3) #Desenha o centro do círculo
70
71              #Informações resumidas de todas as n laranjas analisadas até o momento
72              print("Raio Médio: {0:.2f} mm".format(raio_medio/n))
73              print("Volume Médio: {0:.2f} mL".format(volume_medio/n))
74              #Apenas para preservar a Gramática:
75              if n==1:
76                  print("Foi analisada 1 laranja")
```

UFPR 105 ANOS DE ORGULHO

EXECUÇÃO DO PROGRAMA

```
76                  print("Foi analisada 1 laranja")
77              else:
78                  print("Foram analisadas {0} laranjas".format(n))
79              print() #Pula linha
80
81              #Se o usuário desejar, cria uma janela exibindo a imagem original tratada com as laranjas e a moeda destacadas
82              if input("Gostaria de verificar as laranjas encontradas na imagem?\n Se sim digite S, senão aperte Enter: ")=="S":
83                  cv2.imshow('Aperte Enter para Voltar',cimg)
84                  cv2.waitKey(0)
85                  cv2.destroyAllWindows()
86
87          else:print("Desculpe. Não encontrei nenhuma laranja. Tente outra imagem.") #Mensagem de Erro
88      else:print("Desculpe. Não encontrei nenhuma laranja. Tente outra imagem.") #Mensagem de Erro
89  else:print("Endereco nao encontrado. Tente novamente.") #Mensagem de Erro
90  address = input("Por favor, digite o endereco da imagem. Nao se esqueca da extensao nem de que o arquivo deve estar na mesma pasta que este programa. Aperte Enter para Sair: ")
91   #Continua o ciclo com uma nova imagem
```

UFPR 105 ANOS DE ORGULHO

CONCLUSÃO

- Desafios encontrados:
- Saber qual biblioteca utilizar.
- Transformar os valores de pixel para mm.