

Visão de Máquinas

PROJ1 – Inspeção de qualidade de vedantes automotivos

DESCRIÇÃO DAS TAREFAS

➤ **Tarefa1 (preparar a rotina de inspeção utilizando imagens com resultado conhecido)**

Implementar uma rotina em Python para inspecionar todas as imagens fornecidas nas pastas: OK (15 imagens), NOK_tamanho_forma (5 imagens) e NOK_borda (10 imagens). Como resultado, a rotina deverá criar um arquivo texto de saída contendo uma tabela com as seguintes informações:

- Nome do arquivo
- Flag de indicação do teste de integridade de borda (OK ou NOK)
- Caso peça aprovada no teste de borda, indicar o diâmetro da peça e o status de aprovação (faixa de aprovação de 1% em relação ao valor nominal de 50 milímetros)
- caso peça aprovada no teste de borda, indicar a relação entre eixos A e B e o status de aprovação e o status de aprovação (faixa de aprovação de 5%)

Segue abaixo um exemplo:

Nome do Arquivo	Teste_ borda	Diâmetro	Status Diam	Relação A/B	Status A/B
Peca_X	OK	50,1	OK	1.002	OK
Peca_Y	OK	49,8	OK	1.080	NOK
Peca_Z	NOK	-	-	-	

➤ **Tarefa2 (utilizar a rotina da Tarefa1 para contagem de peças de um vídeo)**

Implementar uma rotina em Python para contagem de peças de um vídeo (Video1_Vedacao.mp4). Como resultado, a rotina deverá criar um arquivo texto de saída contendo uma tabela com as seguintes informações:

- Identificador da peça (contador sequencial)
- Flag de indicação do teste de integridade de borda
- Caso peça aprovada no teste de borda, indicar o diâmetro da peça e o status de aprovação (faixa de aprovação de 1% em relação ao valor nominal de 50 milímetros)

- caso peça aprovada no teste de borda, indicar a relação entre eixos A e B e o status de aprovação e o status de aprovação (faixa de aprovação de 5%)

Segue abaixo um exemplo:

ID	Teste_ borda	Diâmetro	Status Diam	Relação A/B	Status A/B
1	OK	50,1	OK	1.002	OK
2	OK	49,8	OK	1.080	NOK
3	NOK	-	-	-	

➤ **Tarefa3 (incluir a inspeção de defeitos de superfície)**

Implementar uma rotina para contagem de peças de um vídeo (Video2_Vedacao.mp4). Como resultado, a rotina deverá criar um arquivo texto de saída contendo uma tabela com as seguintes informações:

- identificador da peça (contador sequencial)
- flag de indicação do teste de integridade de borda
- flag de indicação do teste de superfície
- caso peça aprovada em ambos os testes, indicar o diâmetro da peça e o status de aprovação (faixa de aprovação de 1% em relação ao valor nominal de 50 milímetros)
- caso peça aprovada em ambos os testes, indicar a relação entre eixos A e B e o status de aprovação e o status de aprovação (faixa de aprovação de 5%)

Segue abaixo um exemplo:

ID	Teste borda	Teste superf.	Diâmetro	Status diam	Relação A/B	Status A/B
1	OK	OK	50,1	OK	1.002	OK
2	OK	OK	49,8	OK	1.080	NOK
3	NOK	OK	-	-	-	
4	OK	NOK	-	-	-	-

RUBRICS DA AVALIAÇÃO:

Conceito	Descrição
C (5)	Índice de acerto de 70% ou mais das peças da tarefa1 (equivalente a 21 de 30 peças)
C+ (6)	Índice de acerto de 83% ou mais das peças da tarefa1 (equivalente a 25 de 30 peças)
B (7)	Rubric C+ e Índice de acerto de 70% ou mais das peças da tarefa2 (equiv. a 17 de 24 pçs)
B+ (8-8.9)	Rubric C+ e Índice de acerto de 83% ou mais das peças da tarefa2 (equiv. a 20 de 24 pçs)
A (9-9.5)	Rubric B+ e Índice de acerto de 70% ou mais das peças da tarefa3 ou Índice de acerto de 90% ou mais das peças das tarefas1 e 2
A+ (10)	Rubric B+ e Índice de acerto de 83% ou mais das peças da tarefa3 ou Rubric A e Interface gráfica para indicação dos valores em tempo real

DATA DA ENTREGA:

16/abril, às 23:59hs (enviar as rotinas Python via Blackboard)

OBSERVAÇÕES:

1) Utilizar os seguintes valores para calibração das imagens (conversão de pixels para milímetros):

- Nas imagens, a largura da esteira é de 65mm
- Nos vídeos, a largura da esteira é de 75,6mm

2) Sugestão: analisar as funções de processamento de contornos do OpenCV

(https://docs.opencv.org/4.x/d3/d05/tutorial_py_table_of_contents_contours.html)

3) Caso existam “furos” no interior de partículas, vc's podem utilizar a rotina fillHoles para “fechá-los”. Esta rotina tem um resultado semelhante à operação morfológica de “close”, porém mais eficiente pois deforma menos a borda do blob.

4) Segue abaixo sugestão de comando para escrita do arquivo texto:

```
f = open("Resultado.txt", "a")
...
f.write(ID, Result_Testes_Borda, Diametro, RelacaoAB)
...
f.close()
```

5) Os vídeos estão com taxa de 30 frames/segundo. Para reduzir o número de imagens a serem processadas e acelerar a varredura completa do vídeo, sugere-se rodar estes vídeos numa taxa menor (5 a 10 frames/segundo). Para isso, basta indicar o frame atual a ser lido atualizando o código abaixo

```
cap = cv2.VideoCapture('../Banco_Imagens/Video_Final/Video1_Vedacao.mp4')
num_frames = int(cap.get(cv2.CAP_PROP_FRAME_COUNT))
frame_atual = 1
while (frame_atual < num_frames-1):

    cap.set(cv2.CAP_PROP_POS_FRAMES, frame_atual)

    ...
```

ao final do while, incrementar a variável “frame_atual” de 3 a 6 para saltar frames.

```
...

frame_atual = frame_atual + 5
```