UFPB-CI-DSC
Disciplina: Visão Computacional
Prof. Augusto de Holanda B. M. Tavares
25 de Junho de 2024

4ª Atividade

Transformações de intensidade, dilatação e erosão

O código abaixo mostra como utilizar a biblioteca OpenCV para abrir um vídeo, exibí-lo em uma janela e salvar o mesmo vídeo em um arquivo:

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

%matplotlib inline

cap = cv.VideoCapture('corgi_race.mp4')

frame_width = int(cap.get(3))
frame_height = int(cap.get(4))
size = (frame_width, frame_height)
result = cv.VideoWriter('filename.avi',
                        cv.VideoWriter_fourcc(*'MJPG'),
                        30, size)

while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        print("Can't receive frame (stream end?). Exiting ...")
        break

    result.write(frame)
    cv.imshow('frame', frame)

    if cv.waitKey(30) == ord('q'):
        break

# Release everything if job is finished
cap.release()
cv.destroyAllWindows()
```

O código acima carrega algumas bibliotecas e define que o objeto de captura *cap* vem de um arquivo .mp4. Em seguida, é estabelecido um *loop* que determina que enquanto a fonte de vídeo estiver aberta (reproduzindo) ou o usuário não der uma entrada para sair (neste

caso a tecla 'q') algum processo será executado. Este processo verifica que a fonte de fato está fornecendo um quadro de imagem e, em seguida, o exibe. Ao final do processo a fonte de captura de vídeo é liberada e quaisquer janelas abertas são fechadas. O resultado do processo também é salvo em um arquivo .avi.

Utilizando o código acima como base, assim como o que foi explorado nas atividades anteriores, escreva um código que realize as tarefas abaixo. Em todos os casos, mantenha o vídeo em dimensão 640×480 (ou menos) para evitar um custo computacional excessivo (caso esteja salvando o vídeo, lembre-se de ajustar as dimensões do arquivo de destino!).

1. Implemente uma transformação de intensidade de limiarização, ou seja, em que todos os *pixels* abaixo de um limiar são definidos em 0 e acima do limiar são definidos em $L-1$. Aplique esta transformação nos canais de imagem de um vídeo colorido. Exiba o resultado e compare com o vídeo original. Escreva um curto parágrafo descrevendo o que é observado como resultado da transformação.

2. Implemente uma transformação de intensidade de negativo. Aplique esta transformação nos canais de imagem de um vídeo colorido. Exiba e compare os resultados. Escreva um curto parágrafo descrevendo o que é observado como resultado da transformação.

3. Implemente uma transformação de intensidade logarítimica nos canais de imagem de um vídeo colorido. Compare o impacto na imagem para diferentes valores da constante c, incluindo o caso de normalização pelo valor máximo do logaritmo da imagem. Escreva um curto parágrafo descrevendo o que é observado como resultado da transformação.

4. Implemente uma transformação de intensidade exponencial nos canais de imagem de um vídeo colorido. Compare o impacto na imagem para diferentes valores das constantes c e γ. Escreva um curto parágrafo descrevendo o que é observado como resultado da transformação.

5. Implemente uma transformação de intensidade de equalização de histograma no canal vermelho um vídeo colorido. Compare o resultado da sua implementação com a função cv.equalizeHist(). Escreva um curto parágrafo descrevendo o que é observado como resultado da transformação.

6. Converta um vídeo para escala de cinza e aplique uma transformação de limiarização em torno da média de intensidade de cada quadro. Em seguida, aplique um processo de erosão e um processo de dilatação utilizando um núcleo de tamanho suficiente para que os efeitos sejam perceptíveis. Escreva um curto parágrafo descrevendo os resultados observados quando comparados ao vídeo limiarizado original.

Funções de referência

Algumas funções de referência necessárias para implementar as tarefas acima são descritas abaixo:

```
bins = range(min_bin, max_bin, interval)
hist,bins = np.histogram(array,bins)
```

- Retorna o histograma e a lista da categorias do histograma. O argumento *bins* especifica o menor e o maior valor da lista de potes e o seu espaçamento.

```
cv.convertScaleAbs(frame)
```

- Modifica a escala e devolve uma versão da entrada em valores absolutos de 8 *bits*.

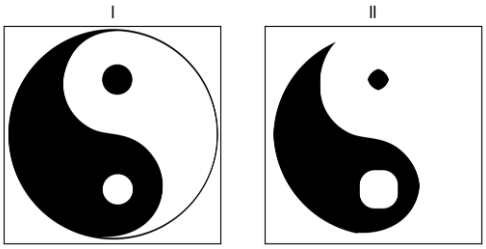
```
kernel = np.ones((odd_integer, odd_integer), "uint8")
cv.dilate(frame,kernel)
#or
cv.erode(frame,kernel)
```

- Retorna uma imagem dilatada ou erodida com grau de intensidade definido pelo tamanho da máscara *kernel*, que é uma matriz de uns de dimensões ímpares.

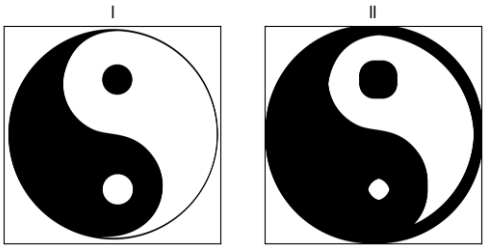
Dilatação e erosão

A **erosão** e a **dilatação** são as **operações morfológicas** mais elementares aplicadas a uma imagem.

Na dilatação uma máscara de formato quadrado ou circular é convoluída com a imagem original, sendo definido um ponto de ancoragem. Para cada *pixel* o máximo valor encoberto pela máscara é tomado como o valor do *pixel* correspondente na imagem resultante. Isto resultará nas regiões mais claras se dilatando na imagem.



A erosão opera de maneira análoga, mas tomando o mínimo valor, de modo que as áreas mais claras são erodidas (pode-se dizer também que neste caso as zonas escuras são dilatadas).



As principais aplicações deste processo são tratar o ruído através da erosão, dilatar elementos de pequenas dimensões para formar áreas de intensidade uniforme contínuas e localizar pontos de alta ou baixa intensidade isolados (picos ou buracos na imagem).