Aluno: André Luis Francisco Junior
PIM - Trabalho final

- Utilizei o ffmpeg para transformar o vídeo em frames e os frames em vídeo novamente;

- Também utilizei as seguintes bibliotecas:
    - cv2
    - numpy
    - matplotlib
    - os

- Deixei um arquivo requirements.txt com todas as versões e bibliotecas instaladas, se der algum erro na execução use pip install -r requirements.txt

- Dividi o trabalho em 2 arquivos .py:
    - imagensTeste.py que compara os algoritmos em uma dada cena para ajudar a escolher o melhor método.
    - main.py que executa o algoritmo de matching em si.

- Consegui fazer o vídeo com o matching tranquilamente, mas tive problemas pra fazer o video das respostas do cvTemplateMatching (a resposta era uma imagem preta), por isso utilizei o plt e salvei as figuras como imagem (por isso o algoritmo demora pra rodar, se comentar as linhas do plt roda bem rápido), mas assim é possível ver o vídeo e as respostas lado a lado

- Os vídeos de entrada são (vídeo / imagem a ser procurada):
    - gato1.mp4 / gato1.png
    - gato2.mp4 / gato2.png
    - carro.mp4 / carro.png
    - mouse.mp4 / mouse.png

- As imagens para teste do método são (imagem / imagem a ser procurada):
    - Capturar1.png / gato1.png
    - Capturar2.png / gato2.png
    - Capturar3.png / carro.png
    - Capturar4.png / mouse.png

- Para testar o código:
    - No arquivo imagensTeste.py não precisa editar nada, apenas rodar o código e as imagens serão geradas e salvas na pasta testes
    - No arquivo main.py, no final do arquivo, existe 4 chamadas de funções:
        - #matching("gato1.mp4", "gato1.png", cv.TM_SQDIFF_NORMED)
        - #matching("gato2.mp4", "gato2.png", cv.TM_SQDIFF_NORMED)
        - #matching("carro.mp4", "carro.png", cv.TM_SQDIFF_NORMED)
        - #matching("mouse.mp4", "mouse.png", cv.TM_CCOEFF)
    - Cada uma dessas chamadas de funções executa o algoritmo para uma entrada;

    - Basta descomentar a linha da entrada desejada e executar.

- Os resultados serão salvos na pasta ‘./’, o nome de saída é:
    - “out_” + nome do vídeo de entrada;
    - “out_res_” + nome do vídeo de entrada para o resultado do cv.matchTemplate;
    - Também é salvo os frames do vídeo e os plots do matplotlib, ficam na pasta que leva o nome do vídeo (carro, gato1, gato2 mouse) (não consegui enviar junto no moodle por ser maior do que o limite determinado pelo moodle mas deixei no drive, link: https://drive.google.com/drive/folders/1EGGGrLWQpQG3AA6a4yk4Z0XdpF5RRESz?usp=sharing)

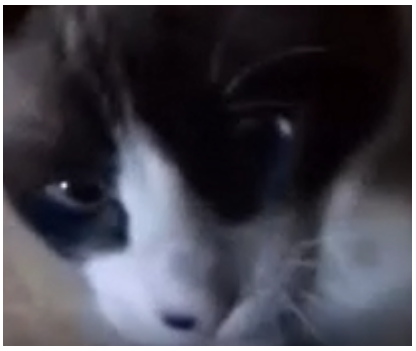
- Entradas utilizadas:

Entrada:

Imagem a ser procurada:

Entrada:

Imagem a ser procurada:



Entrada:

Imagem a ser procurada:

Entrada:

Imagem a ser procurada:

Referências:
FFMPEG.ORG. ffmpeg documentation. Disponível em: https://www.ffmpeg.org/documentation.html
OpenCV. Open Source Computer Vision: Template Matching. Disponível em: https://docs.opencv.org/4.5.2/d4/dc6/tutorial_py_template_matching.html