

CamPython

CamPython é um repositório do *GitHub* criado para colocar programas feitos em *Python* que foram desenvolvidos durante os estudos relacionados à manipulação de imagens e ao uso de câmera.

O repositório pode ser encontrado no perfil do *GitHub* [@brenoCardoso2002](#).

Introdução

Durante os estudos, foram encontradas bibliotecas *Python* que são úteis no desenvolvimento de sistemas para manipulação de imagens, interação com câmeras e criação de detectores.

Essas bibliotecas fornecem recursos para edição, processamento e análise de imagens, além de permitirem a interação com dispositivos de câmera para a captura de imagens em tempo real.

Elas também oferecem métodos para criar detectores capazes de identificar características específicas em imagens, como rostos e mãos.

Algumas das bibliotecas utilizadas incluem *cv2*, *cvzone* e *face_recognition*.

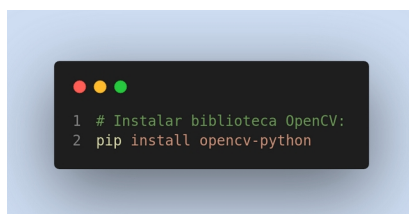
OpenCV (cv2)

A biblioteca *OpenCV* é uma biblioteca de código aberto muito utilizada para o processamento de imagens, visão computacional e aprendizado de máquina.

Com essa biblioteca é possível realizar uma variedade de tarefas relacionadas a imagens, tal como carregar, visualizar, manipular, processar, detectar objetos, rastrear movimentos, realizar reconhecimento facial, entre outras funcionalidades.

Essa biblioteca é uma ferramenta muito robusta para se trabalhar com imagens e realizar análises visuais avançadas.

Abaixo há duas imagens onde nelas são possíveis ver como instala a biblioteca *OpenCV* e como realiza a importação dela para a utilização no código *Python*.



```
1 # Instalar biblioteca OpenCV:
2 pip install opencv-python
```



```
1 # Importar a biblioteca OpenCV:
2 import cv2
```

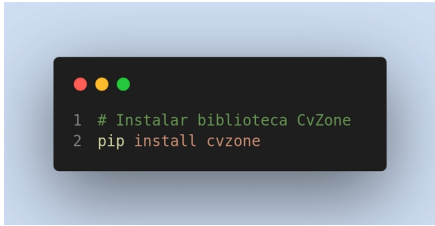
CvZone

A biblioteca *CvZone* é uma poderosa ferramenta de visão computacional, ela foi desenvolvida para simplificar o processo durante a criação de sistemas em *Python* nessa área.

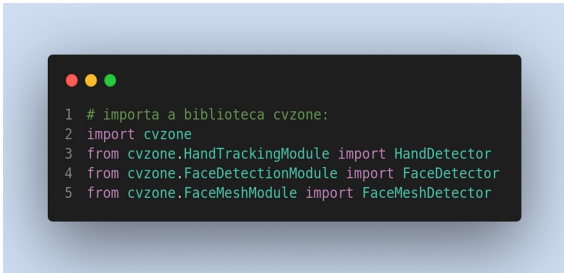
Construída com base na biblioteca *OpenCV*, ela oferece recursos avançados, como a detecção de objetos, rastreamento e reconhecimento facial, entre outros.

Essa biblioteca oferece uma abordagem mais acessível e simplificada para a realização de tarefas mais complexas de visão computacional, aproveitando a base estabelecida pelo *OpenCV*.

Com ela, os desenvolvedores podem aproveitar essas funcionalidades avançadas de maneira a conseguir acelerar o processo de desenvolvimento de sistemas de visão computacional.



```
1 # Instalar biblioteca CvZone
2 pip install cvzone
```



```
1 # importa a biblioteca cvzone:
2 import cvzone
3 from cvzone.HandTrackingModule import HandDetector
4 from cvzone.FaceDetectionModule import FaceDetector
5 from cvzone.FaceMeshModule import FaceMeshDetector
```


face_recognition

A biblioteca *face_recognition* é uma biblioteca voltada à visão computacional, ela oferece recursos para o reconhecimento facial em imagens. Ela é construída sobre a biblioteca *dlib* e utiliza diversas técnicas de aprendizado de máquina para a identificação de rostos em imagens.


Com ela é possível realizar muitas tarefas, como detecção de rostos, extração de características faciais e comparação entre rostos, entre outras tarefas possíveis.

Essa biblioteca permite que seja criado sistemas de reconhecimento facial e detecção de sentimentos por exemplo.

Uma das principais vantagens do uso dessa biblioteca é a facilidade de seu uso, além de que ela fornecer resultados precisos. Isso faz ela ser uma escolha popular para projetos que envolvem reconhecimento facial, pois facilita e agiliza o desenvolvimento e garante resultados que podem ser considerados confiáveis.



```
1 # Instalar biblioteca face_recognition:
2 pip install face_recognition
```



```
1 # Importa a biblioteca face_recognition:
2 import face_recognition
```

Sobre o repositório

Como dito anteriormente o repositório CamPython é hospedado no perfil do [GitHub @brenoCardoso2002](#).

Ele é organizado por partes que estão divididas por pastas, onde cada uma representa uma parte dos estudos. No próximo tópico está informado como foi feita essa divisão, explicando o conteúdo encontrado em cada uma das pastas.

Divisão por partes/pastas

Pasta 'Parte I':

A seção chamada 'Parte I' do repositório consiste em códigos que abordam conceitos básicos sobre a biblioteca *OpenCV*, focando em operações essenciais como a abertura e a manipulação de imagens.

Os arquivos que estão incluídos nessa pasta são os seguintes:

- AbrirImagem.py:

Arquivo feito em *Python*, que permite a abertura e exibição de uma imagem.

- GirarImagem.py:

Arquivo feito em *Python*, onde é possível girar uma imagem e exibir ela na orientação desejada.

- RecortarImagem.py:

Arquivo feito em *Python*, onde é realizado o corte uma imagem em uma área específica e no fim ela é exibida.

- RedimensionarImagem.py:

Arquivo feito em *Python*, que a imagem é redimensionada e exibida.

É importante salientar que, com exceção do arquivo denominado *AbreImagem.py*, todos os códigos possuem duas versões. A primeira versão realiza a exibição da imagem manipulada e a segunda versão realiza o download da imagem.

Pasta 'Parte 2':

A seção denominada 'Parte 2' do repositório contém códigos que abordam conceitos relacionados à abertura da câmera e visualização do vídeo da mesma.

Os arquivos que estão incluídos nessa pasta são os seguintes:

- AbrirCamera.py:

Arquivo feito em *Python*, nele a câmera é aberta e é exibido o vídeo em tempo real.

- SalvarCamera.py:

Arquivo feito em *Python*, onde é possível salvar uma foto da câmera quando o usuário pressiona a tecla 's'.

Esses arquivos fornecem funcionalidades para interagir com a imagem da câmera do dispositivo e é possível manipular as imagens em tempo real mesclando coisas da 'Parte 1' e 'Parte 2'.

Pasta 'Parte 3':

A seção chamada 'Parte 3' do repositório trata sobre conceitos relacionados à detecção de mãos utilizando a biblioteca *CvZone*.

Os arquivos que estão incluídos nessa pasta são os seguintes:

- DetectarMao.py:

Arquivo feito em *Python*, onde é detectado uma mão utilizando a câmera.

- RetanguloPersonalizado.py:

Arquivo feito em *Python*, no qual é feito um retângulo personalizado ao redor da mão identificada.

- ContadorDedos.py:

Arquivo feito em *Python*, aqui o código conta e informa a quantidade de dedos levantados na mão identificada.

Esses arquivos fornecem funcionalidade para identificar e rastrear a posição de uma mão em tempo real, bem como realizar tarefas específicas como contagens de dedos. Esses conceitos são aplicados por meio da biblioteca *CvZone*.

Pasta 'Parte 4':

A seção denominada 'Parte 4' do repositório aborda conceitos relacionados à detecção de rostos utilizando a biblioteca *CvZone*.

Os arquivos incluídos nessa pasta são os seguintes:

- DetectaRosto.py:

Arquivo feito em *Python*, onde é realizado a detecção de um rosto utilizando a câmera.

- RetanguloPersonalizado.py:

Arquivo feito em *Python*, aqui é criado um retângulo personalizado ao redor do rosto identificado.

Esses arquivos fornecem funcionalidades para identificar e rastrear a presença de rostos na visão da câmera.

A detecção de rostos é realizada por meio da biblioteca *CvZone*, permitindo que seja informado se há ou não um rosto.

Pasta 'Parte 5':

A seção denominada 'Parte 5' do repositório é dedicada à exploração de conceitos relacionados à detecção de malhas faciais.

O arquivo incluído nessa seção é o seguinte:

- DetectarFace.py:

Este código é responsável por detectar a presença de uma face na imagem ou no vídeo. Ele utiliza técnicas avançadas de detecção de faces fornecidas pela biblioteca *CvZone*.

Com o uso do arquivo *DetectaFace.py*, é possível a análise e rastreamento de pontos específicos da face, como olhos, nariz, boca e contornos faciais.

Essa seção oferece uma introdução aos conceitos básicos de detecção de faces e serve como ponto de partida para a exploração de recursos mais avançados disponíveis no *CvZone*.

Pasta 'Parte 6':

A seção chamada "Parte 6" do repositório é dedicada à exploração de conceitos relacionados ao reconhecimento facial utilizando a biblioteca *face_recognition*.

O objetivo principal é realizar comparações entre dois rostos para determinar se eles pertencem à mesma pessoa.

Os arquivos incluídos nessa seção são os seguintes:

- CompararFace.py:

Este código permite comparar dois rostos e determinar se eles correspondem à mesma pessoa.

Ele exibe uma porcentagem de correspondência entre os dois rostos, fornecendo uma medida de similaridade.

- CompararFace2.py:

Este código permite comparar dois rostos e determinar se eles correspondem à mesma pessoa.

Ele exibe uma mensagem indicando se os rostos pertencem ou não à mesma pessoa, fornecendo uma resposta mais simplificada.

Esses arquivos permitem explorar e aplicar os conceitos fundamentais do reconhecimento facial, utilizando a biblioteca *face_recognition*.

Ao comparar dois rostos, é possível obter uma medida de similaridade ou uma resposta clara sobre a correspondência entre eles.

Essa seção é útil para o desenvolvimento de aplicativos ou sistemas que envolvam o reconhecimento de rostos e a verificação de identidade.

Pasta 'Parte Bônus':

A seção chamada "Pasta bônus" do repositório contém um arquivo especial que realiza o download de uma imagem aleatória de um site sempre que é executado.

Esse recurso é um bônus adicional, para poder ter uma imagem para realizar testes.

O arquivo incluído nessa pasta é o seguinte:

- DownloadImagem.py:

Este código é responsável por fazer o download de uma imagem aleatória de um site sempre que é executado.

Ele utiliza técnicas para acessar o site, baixar a imagem e salvá-la no diretório atual.

Esse arquivo é uma adição especial ao repositório, proporcionando um recurso divertido e interessante.

Ao executar o código, uma nova imagem aleatória será baixada a cada execução, permitindo a obtenção de imagens variadas e surpreendentes para poder usar nos testes.