

1. Guia de Instalação

Para instalar o projeto siga um dos seguintes passos:

Use o instalador de pacotes do python “pip” no terminal ou cmd e digite:

1.

- **pip install opencv-contrib-python**
- **pip install scipy**
- **pip install matplotlib**

2.

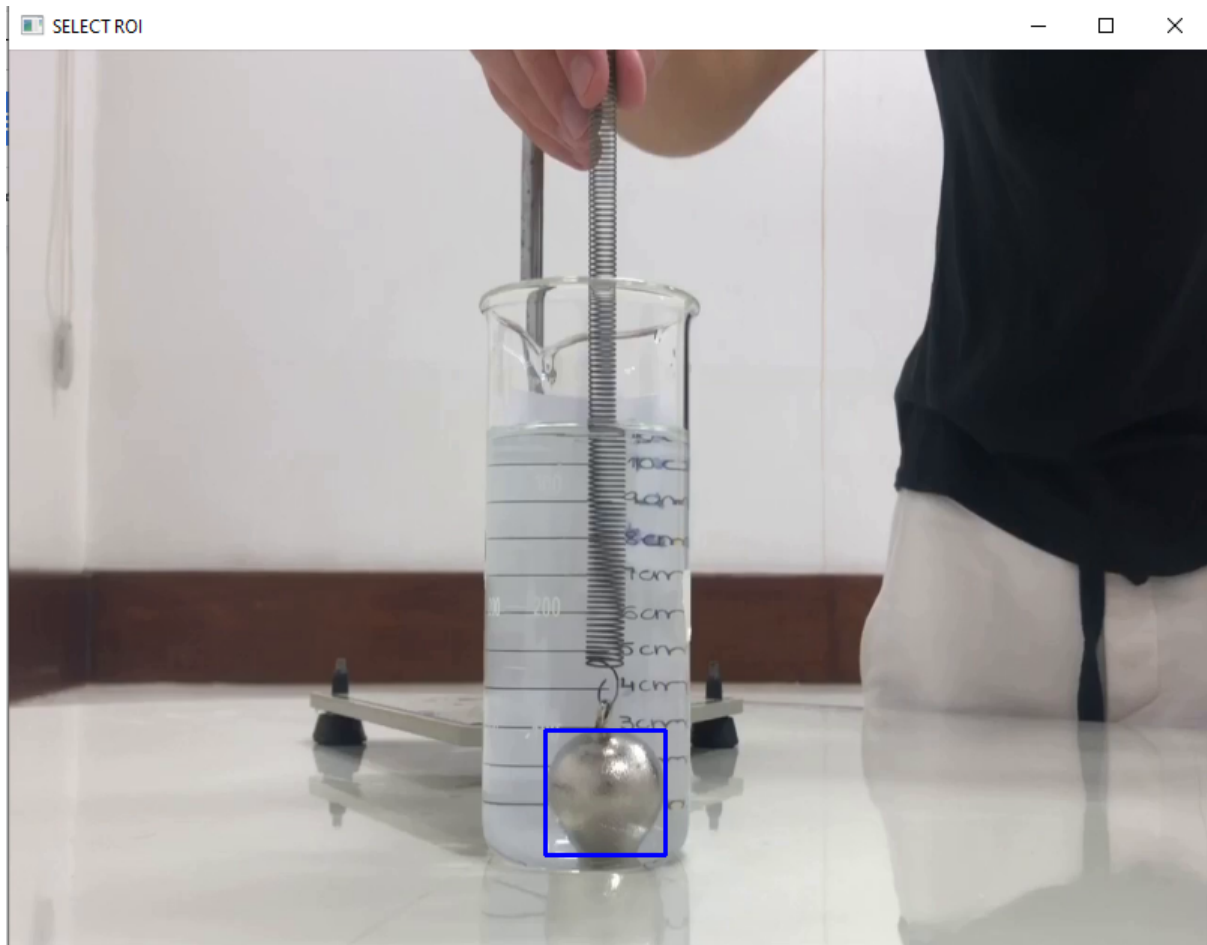
- **pip install -r requirements.txt**

2. Como usar

Antes de usar, é necessário entender que existem três rotinas que são utilizadas para detectar objetos presentes no módulo DetectarObjetos.py, e portanto, atendem necessidades diferentes.

1. Tracking

É responsável por detectar um único objeto, é possível selecionar uma região de interesse utilizando o primeiro frame do vídeo em questão:



Também é possível calibrar manualmente a região de interesse, que pode ser útil em situações nas quais o Tracking apresenta instabilidade em muitas posições, dando as coordenadas de X, Y e de tamanho do retângulo. Para tanto, dentro da rotina de Tracking, no `DetectarObjetos.py`, comente a linha 26 e descomente a linha 27, dessa maneira:

```
26      #box = selectROIfromFrame(first_frame) # Tracking
27      box = (557, 140, 20, 28) # Define manualmente a área a ser analisada
```

Note que existem sugestões de áreas a serem analisadas para os casos no qual há instabilidade no Tracking:

```
# Salto com vara: (353, 410, 30, 60)
# Basquete: (557, 140, 20, 28)
```

Outra forma de calibrar a região de interesse é por meio de uma imagem, essa é bem menos recomendada, mas para usá-la, certifique-se de descomentar as linhas 18 e 25, e comentar as linhas 26 e 27.

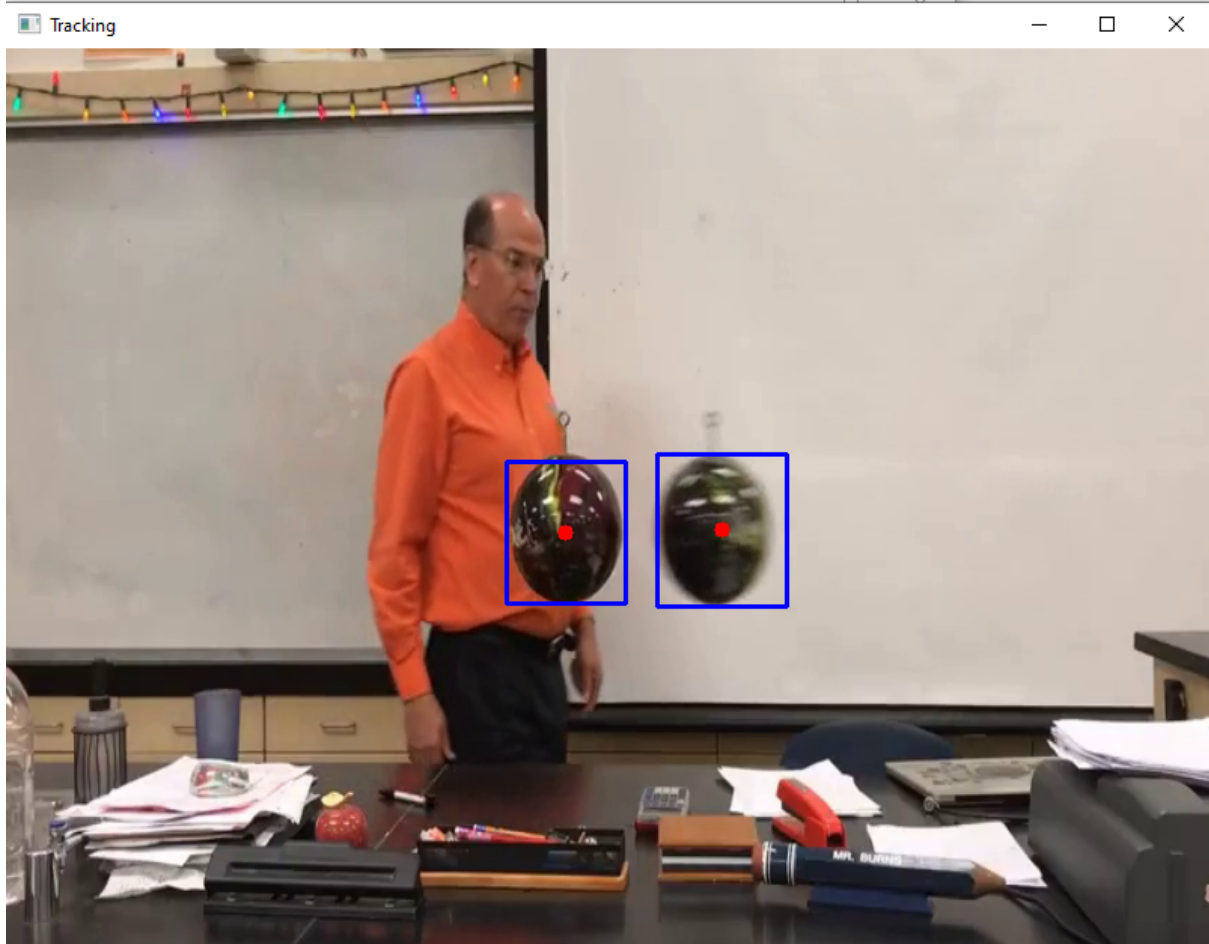
```

18  img_object = cv2.imread('objects\Ball.png') # Se for imagem
19  # Salto com vara: (353, 410, 30, 60)
20  # Basquete: (557, 140, 20, 28)
21  ret, first_frame = cap.read()
22
23  first_frame = cv2.resize(first_frame, (800, 600)) # Se for frame
24
25  box = selectROIfromFrame(img_object)
26  #box = selectROIfromFrame(first_frame) # Tracking
27  #box = (557, 140, 20, 28) # Define manualmente a área a ser analisada
28

```

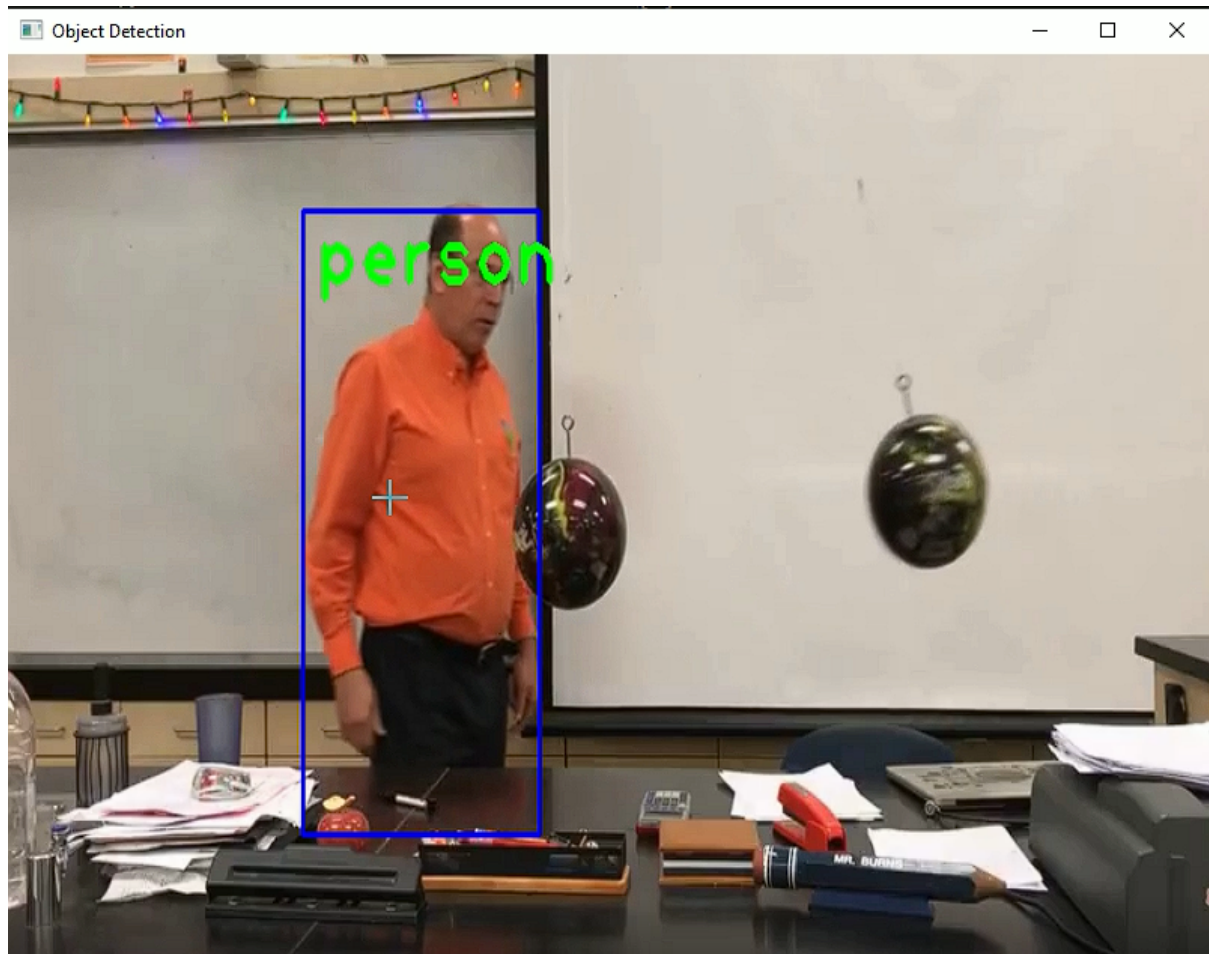
2. MultiTracking

A diferença efetiva dele para o Tracking é que aqui é possível definir mais de uma região de interesse, e portanto capturar o movimento de vários objetos simultaneamente vide exemplo:



3. Detecção de objetos

Este utiliza uma abordagem diferente do (1) e (2), é baseada em um modelo pré-treinado de Inteligência Artificial utilizando TensorFlow, a desvantagem dele é que ele não tem uma taxa de detecção tão aguçada para pequenos objetos em movimento, a vantagem é que ele permite classificar objetos e não precisa de uma região de interesse, ele detecta o tipo do objeto, se é um carro, uma bola de golfe, basquete e etc...



O arquivo principal é o “AnalisaVideo.py”, é por onde chamamos as funções criadas nos módulos, as funções Tracking e MultiTracking retornam X, Y e Tempo, enquanto ObjectDetectionAI não retorna nada, é só chamá-la. Note o exemplo de arquivo “AnalisaVideo.py”:

OBS.: O MultiTracking precisa do número de objetos como parâmetro

```
1 import cv2
2 import numpy as np
3 import matplotlib.pyplot as plt
4 from DetectarObjetos import ObjectDetectionAI, MultiTracking, Tracking
5 from AnalisaDados import HarmonicOscAmort, ProjectileMotion, PendCollision, MinQuadrados
6
7 X, Y, Time = MultiTracking(2, 'samples\Collision_1.mp4')
8
9 PendCollision(X, Y, Time, 2)
```

Foi criado um módulo relativo à análise de dados, denominado “AnalisaDados.py”, que auxilia e entrega algumas coisas prontas dependendo do experimento analisado. Conta com funções que analisam Colisão Pendular, Lançamento Oblíquo(da mesma maneira funciona para os retilíneos) e Movimento Harmônico Amortecido(deve funcionar com o harmônico simples também). É só chamar elas e fornecer os dados.