Relatório Projeto IVC –

Fase 3

**EDJD 2022/2023**

*Este projeto serviu de prepósito a um projeto para a cadeira de Introdução a visão por computador, em python dividido em três fases.*

# Índice

[Introdução 1](#_Introdução) [Propósitos e Objetivos 2](#_Propósitos_e_Objetivos) [Fase 1 3](#_Fase_1)  [Fase 2 4](#_Fase_2) [Fase 3 5](#_Fase_3) [Desenvolvimento da Fase 3 6](#_Desenvolvimento_da_Fase)

[Conclusão 7](#_Conclusão)

# Índice de imagens

Figura 1 – Importação da library yolov5 7

Figura 2 - ROI selection 8

Figura 3 - Frame cropping 8

Figura 4 – Conversão do frame para o formato rgb 8

Figura 5 – Análise do frame pela library yolov5 9

Figura 6 - Retângulo para representar o roi 9

Figura 7 - Inicializar o centro de massa com valor None 9

Figura 8 – Verificação com iteração por todos os objetos 10

Figura 9 - Retângulo para representar o roi quando o objeto é detetado 10

Figura 10 – Parâmetros do objeto selecionado 10

Figura 11 - Retângulo para representar a posição do objeto 11

Figura 12 Texto para identificar o objeto 11

Figura 13 – centro do objeto selecionado e detetado 11

Figura 15 - Cálculo da coordenada a devolver 12

## Introdução

Este trabalho foi proposto pelo professor José Brito, no contexto da unidade curricular Introdução à Visão por Computador, integrada no primeiro semestre do 2º ano de licenciatura de Desenvolvimento de Jogos Digitais, que visa o reforço e a aplicação dos conhecimentos adquiridos ao logo do semestre.

## Propósitos e Objetivos

No jogo, o jogador controla um paddle de forma que a bola destrua os tijolos da parede. O objetivo do trabalho é que o jogador controle o paddle através da câmara. O trabalho tem 3 fases. Em cada fase o controlo do paddle deve basear-se em diferentes técnicas de Visão por Computador. Em todas as fases deve ser apresentada a janela com o jogo e pelo menos outra janela com a imagem captada pela câmara e a visualização do resultado dos algoritmos de Visão por Computador. Poderão ser apresentadas várias janelas com resultados intermédios dos algoritmos de Visão por Computador. Em todas as fases, poderá ser usado o rato para inicializar os algoritmos usados. Na fase 1 o controlo do paddle deve ser baseado em algoritmos de segmentação. Poderá ser aplicado qualquer algoritmo de segmentação de entre os abordados na aula. O objeto a ser segmentado na imagem poderá ser um ou dois objetos de cor predefinida. Se o(s) objeto(s) a segmentar forem as mãos ou a cara do jogador, a segmentação poderá basear-se na cor da pele O controlo do paddle deve basear-se na posição do(s) objeto(s) segmentados na imagem.

### Fase 1

Na fase 1 o controlo do paddle deve ser baseado em algoritmos de segmentação. Poderá ser aplicado qualquer algoritmo de segmentação de entre os abordados na aula. O objeto a ser segmentado na imagem poderá ser um ou dois objetos de cor predefinida. Se o(s) objeto(s) a segmentar forem as mãos ou a cara do jogador, a segmentação poderá basear-se na cor da pele O controlo do paddle deve basear-se na posição do(s) objeto(s) segmentados na imagem.

### Fase 2

Na fase 2 o controlo do paddle deve ser baseado em algoritmos de deteção de movimento. Poderá ser aplicado qualquer algoritmo de deteção de movimento de entre os abordados na aula. O controlo do paddle deverá basear-se no movimento dos pixels da imagem.

### Fase 3

Na fase 3 o controlo do paddle deve ser baseado em algoritmos de deteção de objetos. Poderá ser aplicado qualquer algoritmo de deteção de objetos de entre os abordados na aula. O controlo do paddle deve basear-se na posição do(s) objeto(s) detetados na imagem.

## Desenvolvimento da Fase 3

Nesta terceira fase decidi complementar a primeira parte do projeto, complementando o código que já controlava o paddle com base na posição absoluta do objeto detetado na câmara.

Dado isto, para evitar a repetição das explicações do relatório da primeira fase, apenas focarei no desenvolvimento e integração da deteção de movimento no código já existente.

Começamos por importar a library do yolov5, que é um algoritmo de deteção de objetos que divide imagens num sistema de grelha. Cada célula na grelha é responsável por detetar objetos nela mesma.

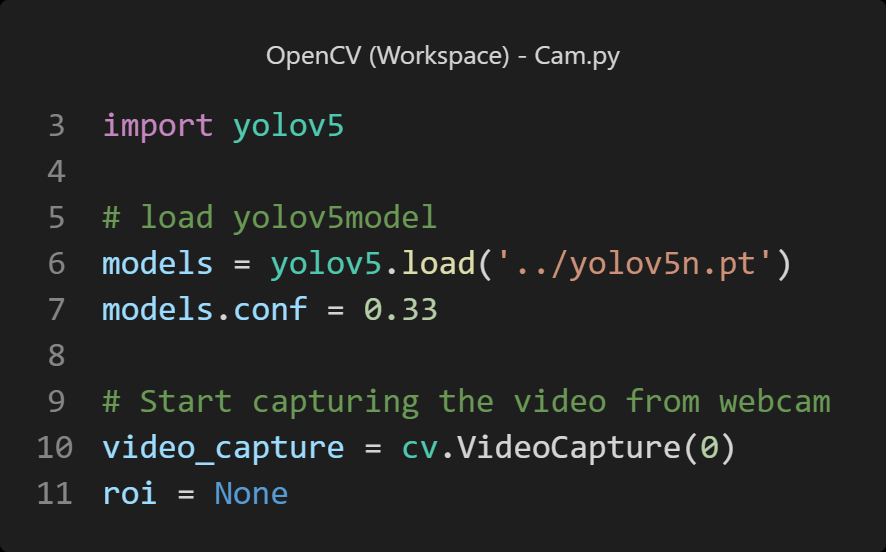


Figura 1 – Importação da library yolov5

Após lermos o frame atual do vídeo\_vapture verificamos se o roi (region of interest) é nulo, que só acontece no primeiro frame, nesse caso o utilizador pode selecionar uma região no ecrã ou cancelar a operação selecionando assim o frame na sua totalidade como roi.

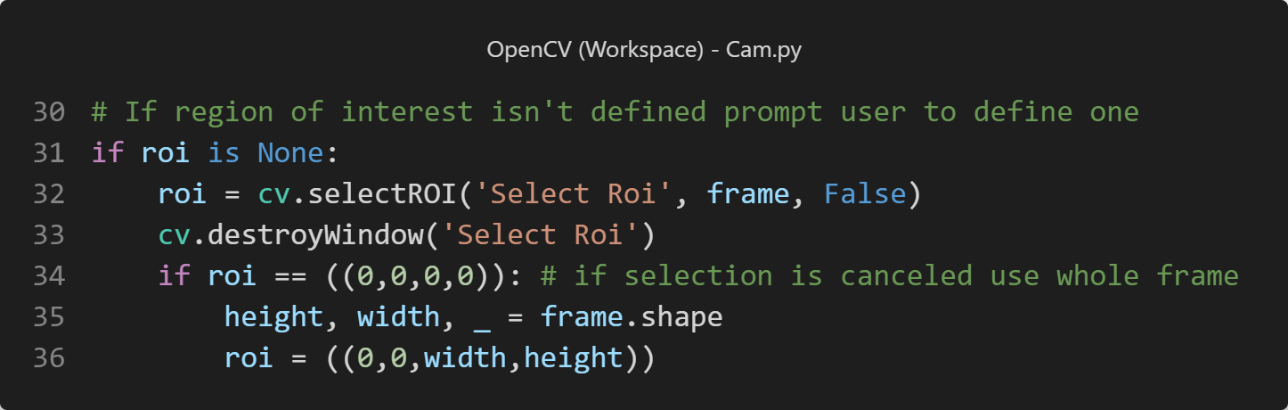


Figura 2 - ROI selection

Com o roi agora definido cortamos o frame com as suas coordenadas iniciais e dimensões obtendo assim o frame\_roi. Este permitira apenas realizar as deteções na região pretendida.

Uma imagem com texto

Descrição gerada automaticamente

Figura 3 - Frame cropping

Aplicamos de seguida a função cvtColor ao frame\_roi para alterar assim o seu formato de cor do BGR para o RGB.

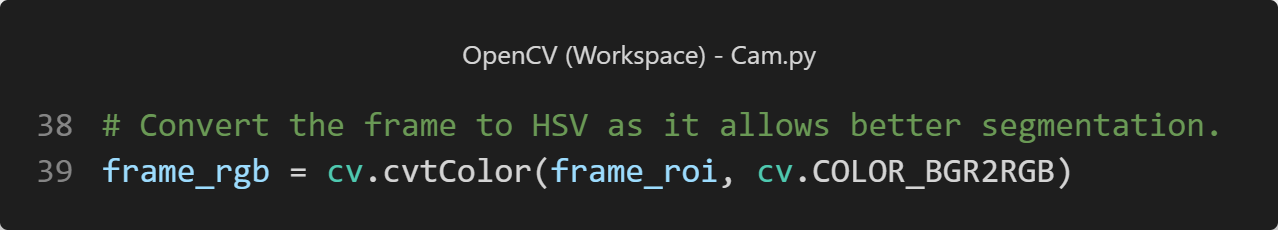


Figura 4 – Conversão do frame para o formato rgb

Analisámos o frame já em rgb com base nos modelos de yolov5 carregados, para detetar assim os objetos nestes especificados.

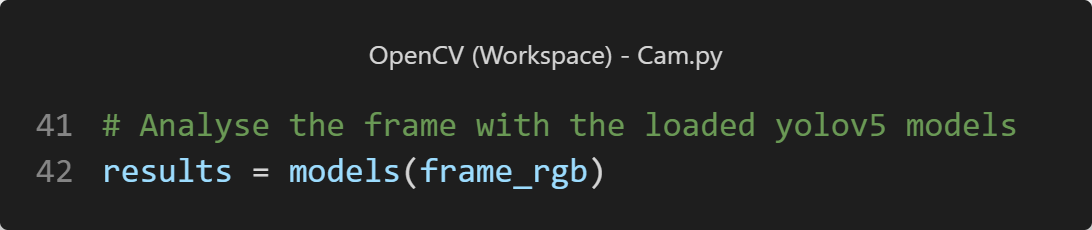


Figura 5 – Análise do frame pela library yolov5

Desta vez começamos a função draw\_contours por desenhar um retângulo azul com as dimensões do roi no frame, representando assim a área de interesse que o user selecionou.

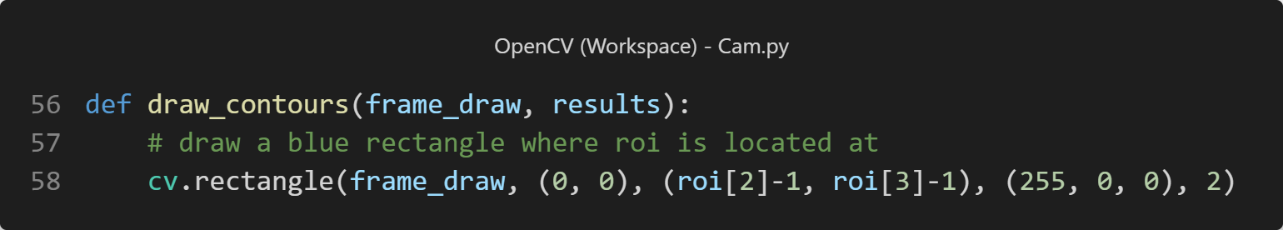


Figura 6 - Retângulo para representar o roi

Inicializamos o centro como None para o caso não detetarmos.

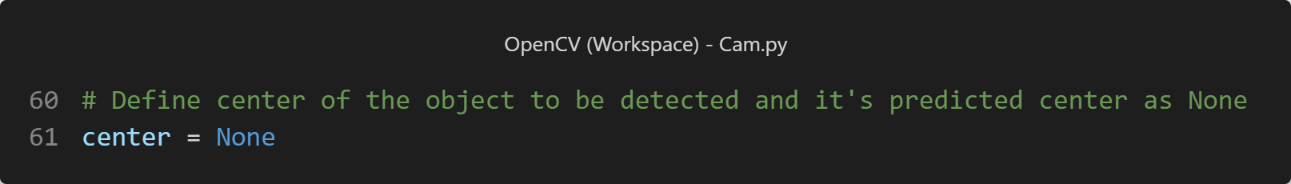


Figura 7 - Inicializar o centro de massa com valor None

Iteramos pelos objetos resultantes do modelo do yolov5 com o objetivo de encontrar o objeto pretendido através da condição que compara o nome do objeto detetado com o nome do selecionado.

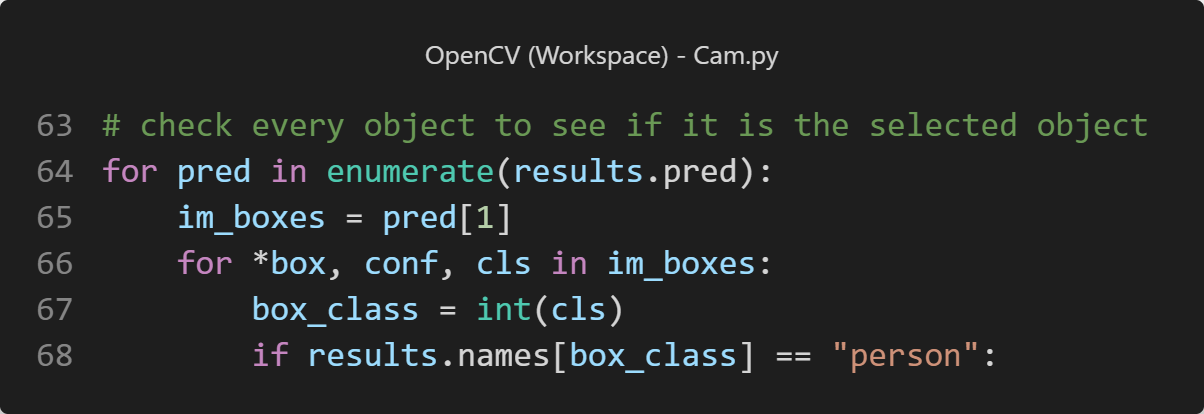


Figura 8 – Verificação com iteração por todos os objetos

Caso detetarmos o objeto selecionado nos resultados, desenhamos novamente um retângulo que representa a região de interesse, mas verde, esta mudança de cor dá um feedback ao jogador de que o programa está a detetar o objeto.

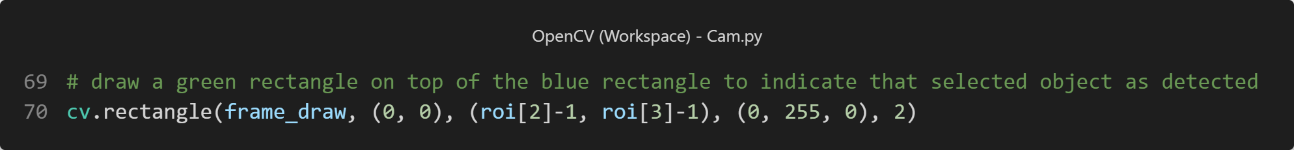


Figura 9 - Retângulo para representar o roi quando o objeto é detetado

De seguida guardamos os parâmetros do objeto selecionado em variáveis para facilitar a leitura da restante função.

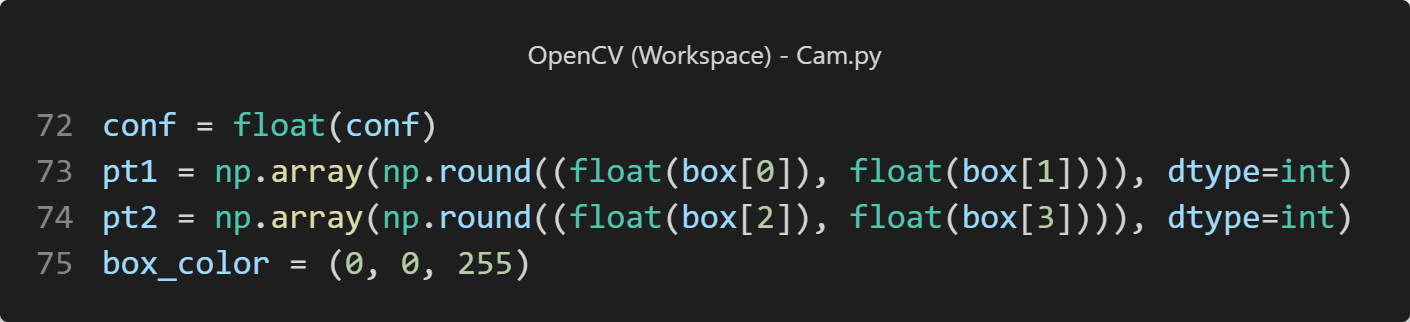


Figura 10 – Parâmetros do objeto selecionado

Com estes parâmetros começamos por desenhar um retângulo que delimita o objeto detetado.

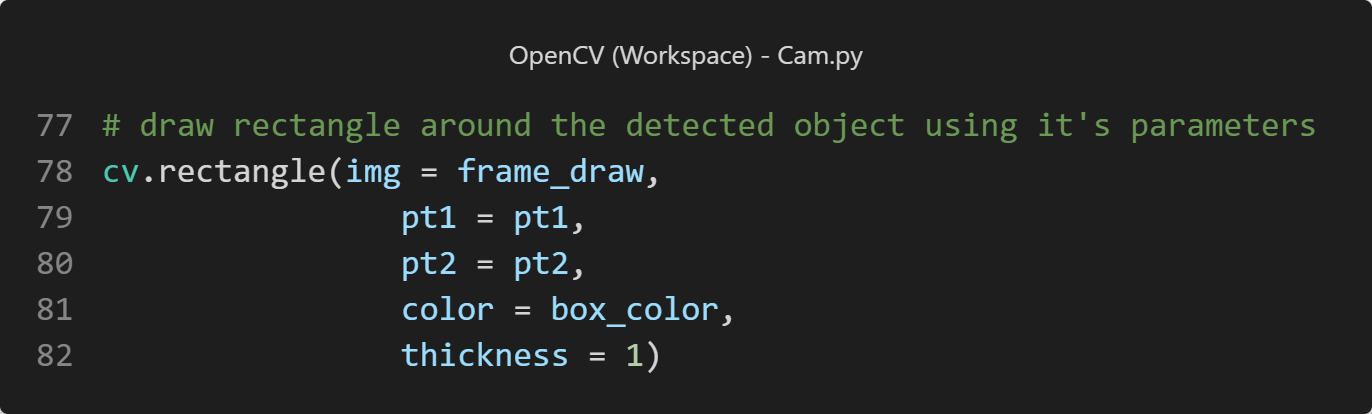


Figura 11 - Retângulo para representar a posição do objeto

E com os mesmos colocamos um texto acima da posição do retângulo que identifica o nome do modelo de objeto detetado.

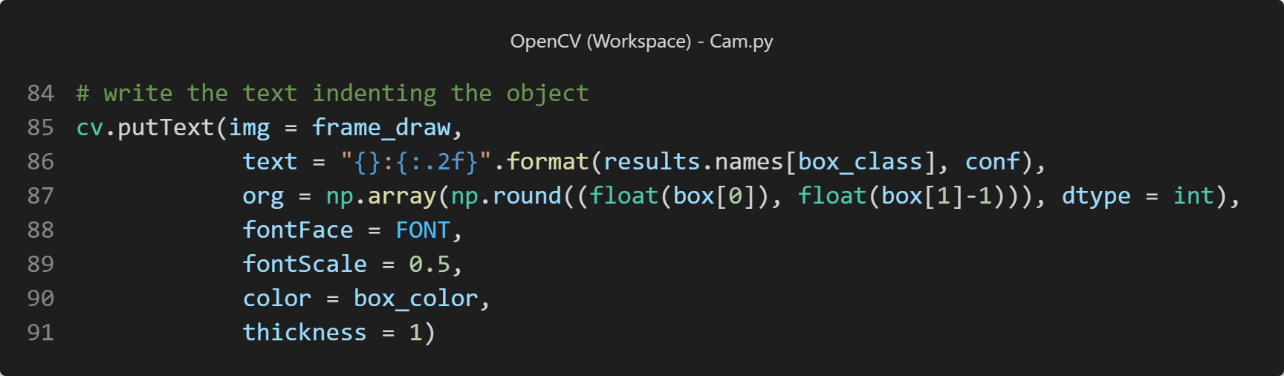


Figura 12 Texto para identificar o objeto

Guardamos o valor aproximado do centro do objeto selecionado que foi detetado para o retornar no fim da função.

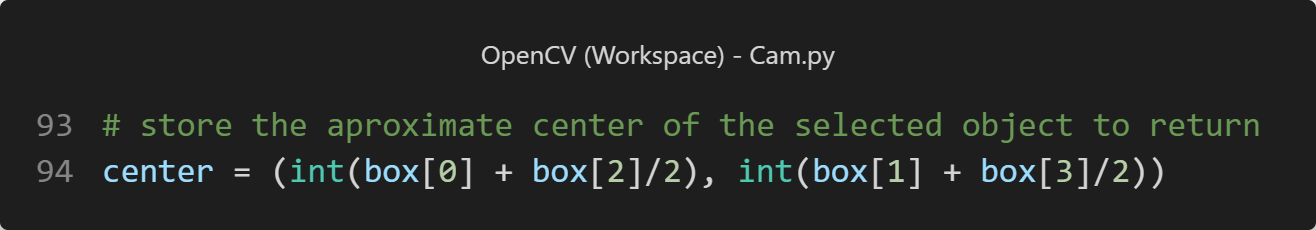


Figura 13 – centro do objeto selecionado e detetado

Devido à utilização do roi, encontrámos um obstáculo ao antigo método de movimentação do paddle, como as coordenadas do roi estavam limitadas pela sua dimensão o paddle não se movimentava para além das mesmas, fazendo com que não pudesse se movimentar no mapa todo, para resolver este problema devolvemos o resultado da multiplicação da coordenada x do centro previsto pela razão entre a largura da câmara e da largura do roi, resultando assim em coordenadas válidas para a totalidade do ecrã do breakout.

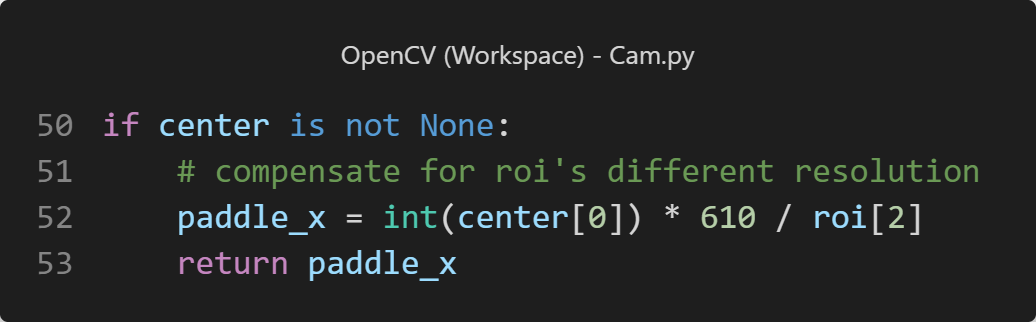


Figura 14 - Cálculo da coordenada a devolver

## Conclusão

Com esta segunda fase foi possível aprofundar mais conhecimentos acerca da deteção e rastreamento de movimentos numa imagem através de diversos métodos, e a utilização da mesma para múltiplas aplicações. Aprendi também a importância de analisar as diversas opções de soluções que estavam à minha disposição antes de resolver o problema em questão.