**Hackaton - TC5 - FIAP IA para Devs**

**Thais Santos Barbosa RM 357883**

A solução desenvolvida integra ferramentas como YOLOv8, OpenCV e Mailtrap e tem como objetivo a detecção de objetos cortantes em vídeos de diferentes fontes e emitindo alertas em tempo real. A seguir, algumas etapas do código serão explicadas em maiores detalhes.

**1. Configuração do Ambiente**

O primeiro passo foi configurar o ambiente de desenvolvimento, garantindo que todas as bibliotecas necessárias fossem instaladas. As bibliotecas utilizadas incluem:

* ***Ultralytics***: Para trabalhar com o modelo YOLOv8.
* ***Roboflow***: Para gerenciar datasets e treinar o modelo.
* ***OpenCV***: Para manipulação de vídeos e imagens.
* ***yt-dlp***: Para baixar vídeos do YouTube.
* ***Requests***: Para envio de alertas por e-mail.

**2. Definição de Configurações**

Algumas variáveis foram configuradas para controlar o comportamento do sistema:

* ***forcar\_retreinar\_modelo***: Define se o modelo YOLOv8 deve ser treinado novamente ou se será apenas carregado do Google Drive.
* ***nome\_video\_output***: Nome que será dado ao arquivo de vídeo de saída com as detecções.
* ***opcao***: Permite escolher a origem do vídeo (upload, YouTube ou caminho local).

**3. Entrada de Vídeo**

O sistema oferece três opções para fornecer o vídeo de entrada:

1. ***Upload de um vídeo:*** Utilizando a biblioteca google.colab.files.
2. ***Download de um vídeo do YouTube***: Utilizando o yt-dlp.
3. ***Caminho local para o vídeo***: Especificado diretamente no código.

O código utiliza a estrutura match-case para selecionar a origem do vídeo com base na variável opcao.

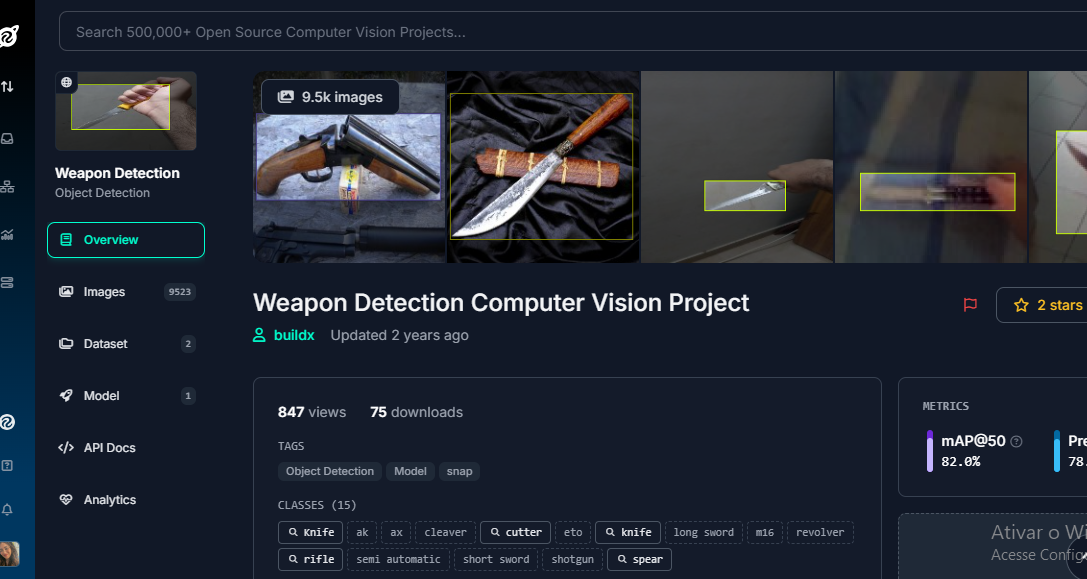
**4. Treinamento do Modelo YOLOv8**

Caso a variável forcar\_retreinar\_modelo esteja ativada, o modelo YOLOv8 será treinado utilizando um dataset fornecido pelo Roboflow. O fluxo inclui:

* Configuração da API Key do Roboflow.
* Download do dataset.
* Treinamento do modelo com parâmetros como número de épocas, tamanho das imagens e cache.

O modelo treinado é salvo no Google Drive para reutilização futura.

Neste caso, foi utilizado o ***dataset weapon-detection-7kro8***, que contém 9523 imagens de objetos cortantes (espadas, facas, estiletes etc.) em diversas posições e iluminações. Link: <https://universe.roboflow.com/buildx/weapon-detection-7kro8>



O ***modelo pré-treinado escolhido foi o yolov8m.pt***, que possui maior capacidade de aprendizado que as versões nano e small, porém exige menos recursos computacionais que a versão large.

O ***treinamento ocorreu em 30 épocas***, um valor que permitiu uma significativa redução do loss ao longo das etapas.

**5. Carregamento do Modelo Treinado**

Se o modelo já estiver treinado, ele é carregado diretamente para realizar as detecções. O arquivo do modelo (***best.pt***) é recuperado do Google Drive.

**6. Processamento do Vídeo**

O vídeo de entrada é processado frame a frame utilizando o OpenCV. Para cada frame:

1. O modelo YOLOv8 realiza a detecção de objetos.
2. Se um objeto cortante for detectado e seu score ***(nível de confiança) foi maior que o threshold definido***, um retângulo é desenhado ao redor do objeto.
3. Um alerta é emitido via e-mail com uma imagem capturada do frame.

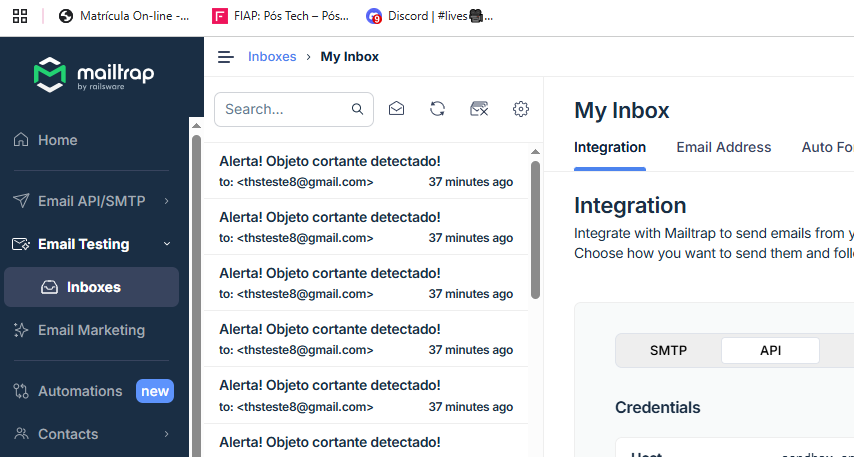
O total de objetos detectados é contabilizado e exibido ao final do processamento.

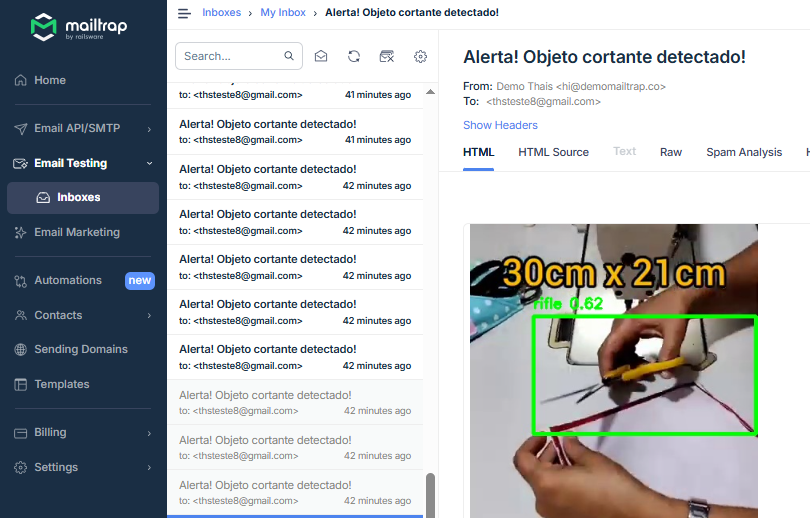
**7. Envio de Alertas por E-mail**

Quando um objeto cortante é detectado, ***um alerta é enviado por e-mail utilizando a API do Mailtrap***. O e-mail inclui:

* Um título de alerta.
* Uma mensagem informativa.
* Uma imagem capturada do objeto detectado.

O frame é convertido para base64 e incluído no corpo do e-mail como uma imagem.





**8. Vídeo de Saída**

O vídeo processado, com as detecções marcadas, é salvo no caminho especificado pela variável ***nome\_video\_output***. O vídeo pode ser baixado para análise posterior.