

# Hackathon – Fiap – POS Tech – IA para desv

Aluno: **Mcgregory D F R Pinto**

Matrícula: **rm357350**

Github: <https://github.com/McgregoryPinto/Hackathon>

Video: <https://youtu.be/bHGqpv19NE0>

(apenas após ver o vídeo percebi que não apareço, desculpem esta falha)

## Detecção de materiais cortantes

### Desafios:

O primeiro desafio foi encontrar um dataset com as imagens desejadas.

Optei por um dataset do kaggle:

<https://www.kaggle.com/datasets/feliparus/knife-scissors-and-cutter-detection-dataset/data>

Outro desafio foi selecionar qual o método/modelo usar para esta tarefa, optei pelo YOLO, pois não queria apenas repetir o que foi feito nas aulas. Para a utilização do YOLO foi necessário preparar o dataset, o que levou um tempo considerável do trabalho. Após preparar o dataset, separei as imagens que o modelo reconheceu como objetos cortantes. Neste ponto analisei as classes de objetos que o modelo fornece e separei knife, scissors e fork (faca, tesoura e garfo, respectivamente) para a detecção destes objetos nas cenas. Mas para facilitar o processo classifiquei todos como knife.

Finalizado esta etapa seguimos para o treinamento do modelo usando o dataset, como este processo demora um tempo significativo, fiz poucos ajustes nos parâmetros.

### Aplicação do modelo

Após o treinamento do modelo treinado localmente, este falhou no reconhecimento de uma faca.

Já ao utilizar o modelo pré treinado do YOLO teve um desempenho bem melhor, gerando os alarmes quando reconheceu a faca na cena. Um porém o processo é lento.

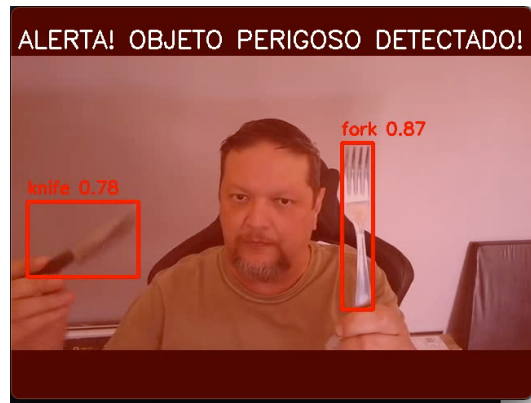
Os vídeos com essas tentativas se encontram no github com os nomes:

Original: teste.mp4

Treino local: treino\_local.mp4

Treino YOLO: treino\_yolo.mp4

Quando um objeto classificado como perigoso aparece em cena deverá aparecer um alerta “ALERTA! OBJETO PERIGOSO DETECTADO!”. Exemplo:



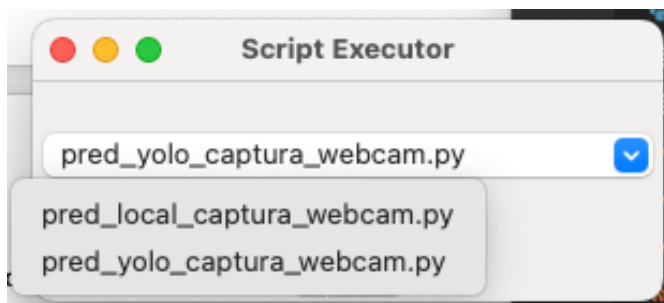
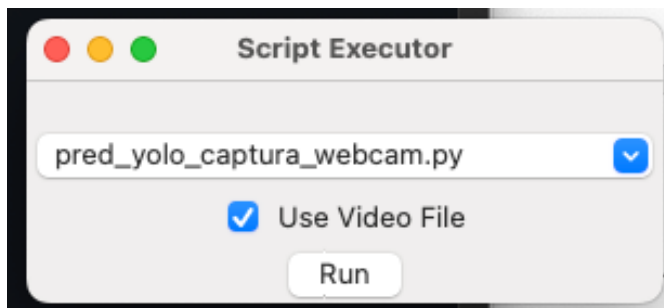
## Testando o projeto

Para testar o projeto, baixe os fontes do github. É necessário ter python instalado e instalar os pacotes que estão no arquivo requirements.txt.

No diretório do projeto execute:

```
python3 ./script_executor.py
```

Vai surgir a janela abaixo para escolher se executará o modelo treinado local ou o modelo YOLO. Se clicar em “Use vídeo file” dará a opção de usar um vídeo de seu computador, caso contrário tentará usar a sua webcam.



No final do processamento será gravado um vídeo com o resultado: pred\_local\_captura\_webcam.mp4 – se usado o treinamento local

pred\_local\_captura\_webcam.mp4 – se usado o treinamento YOLO

### **Considerações finais**

A análise frame a frame tem um custo de processamento considerável para execução local, acredito que se usar entre 20 e 30% dos frames a eficiência do modelo seria o mesmo.

A utilização dos modelos pré treinados para o proposto pelo hackathon se mostrou mais eficiente.