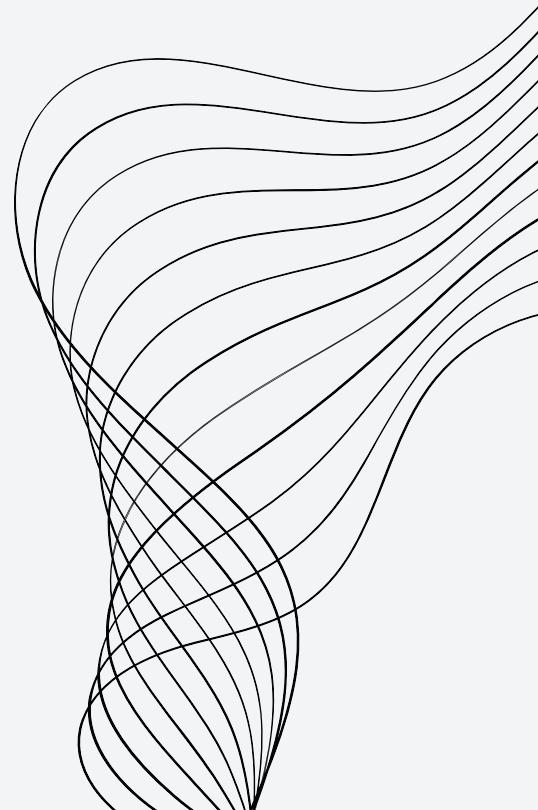


DÉPLOIEMENT D'UNE APPLICATION DE DÉTECTION D'ARMES À FEU



PLAN

- 
- 01** INTRODUCTION
 - 02** CONCEPTION
 - 03** IMPLÉMENTATION
 - 04** TESTS
 - 05** OBSTACLES
 - 06** CONCLUSION

01 INTRODUCTION

APERÇU GÉNÉRAL



Motivation du projet



Description du projet

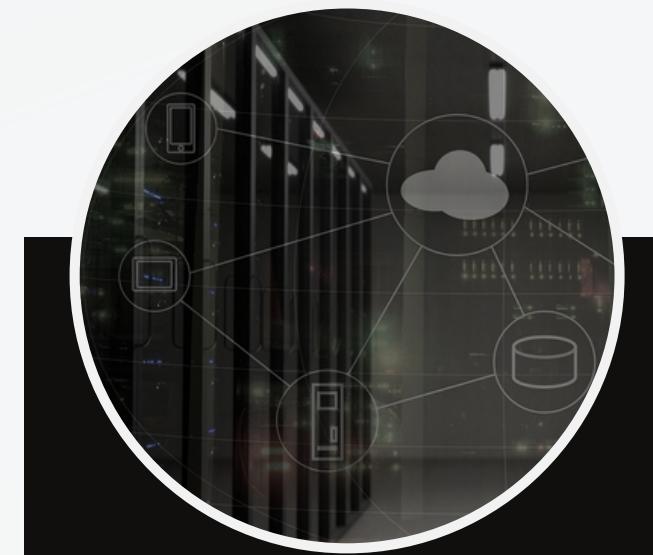


02 CONCEPTION

SYSTÈME



Modèle



API



Application
Web

OUTILS EXISTANTS

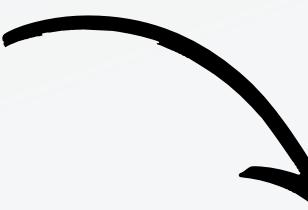
- Coût
- Scalabilité
- Praticabilité

ON-PREMISE

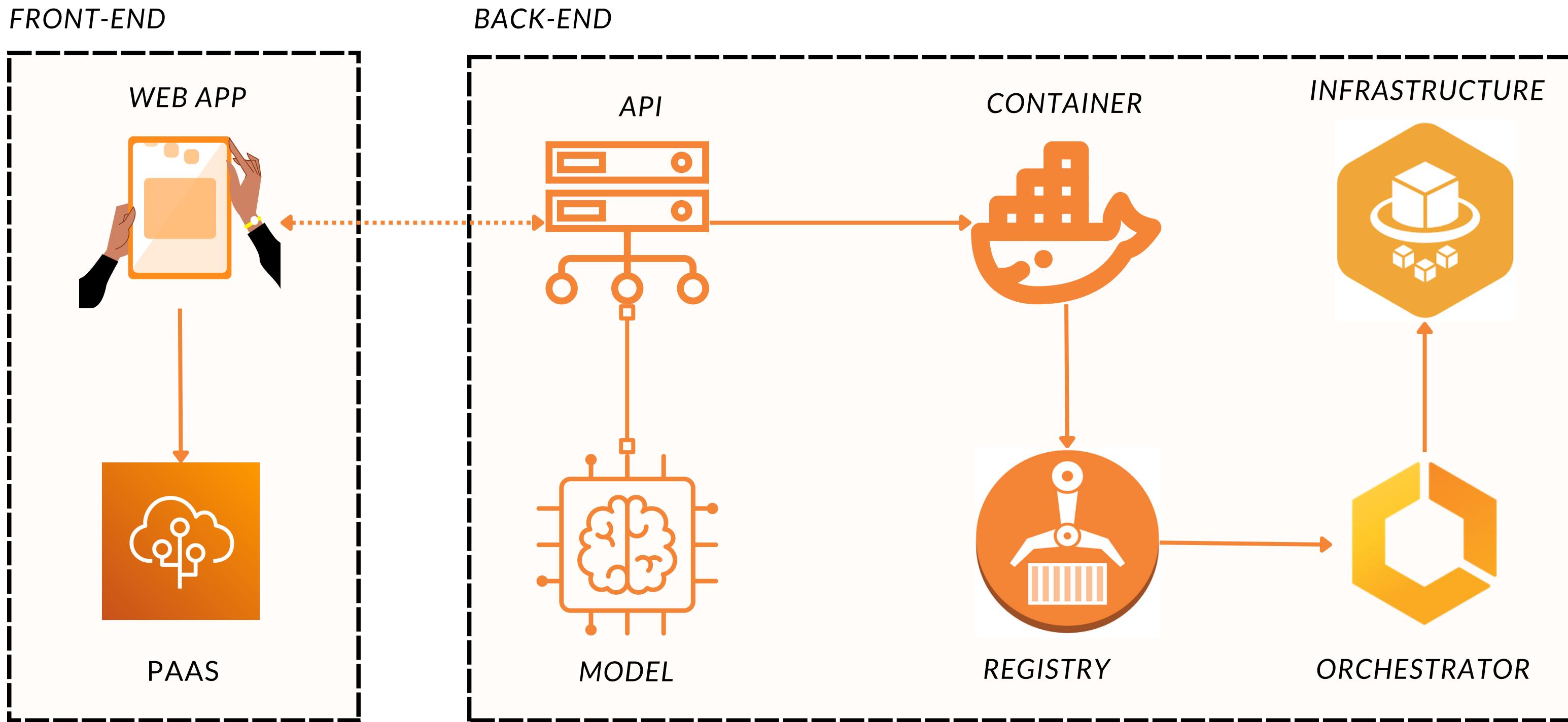
SERVERLESS

PAAS

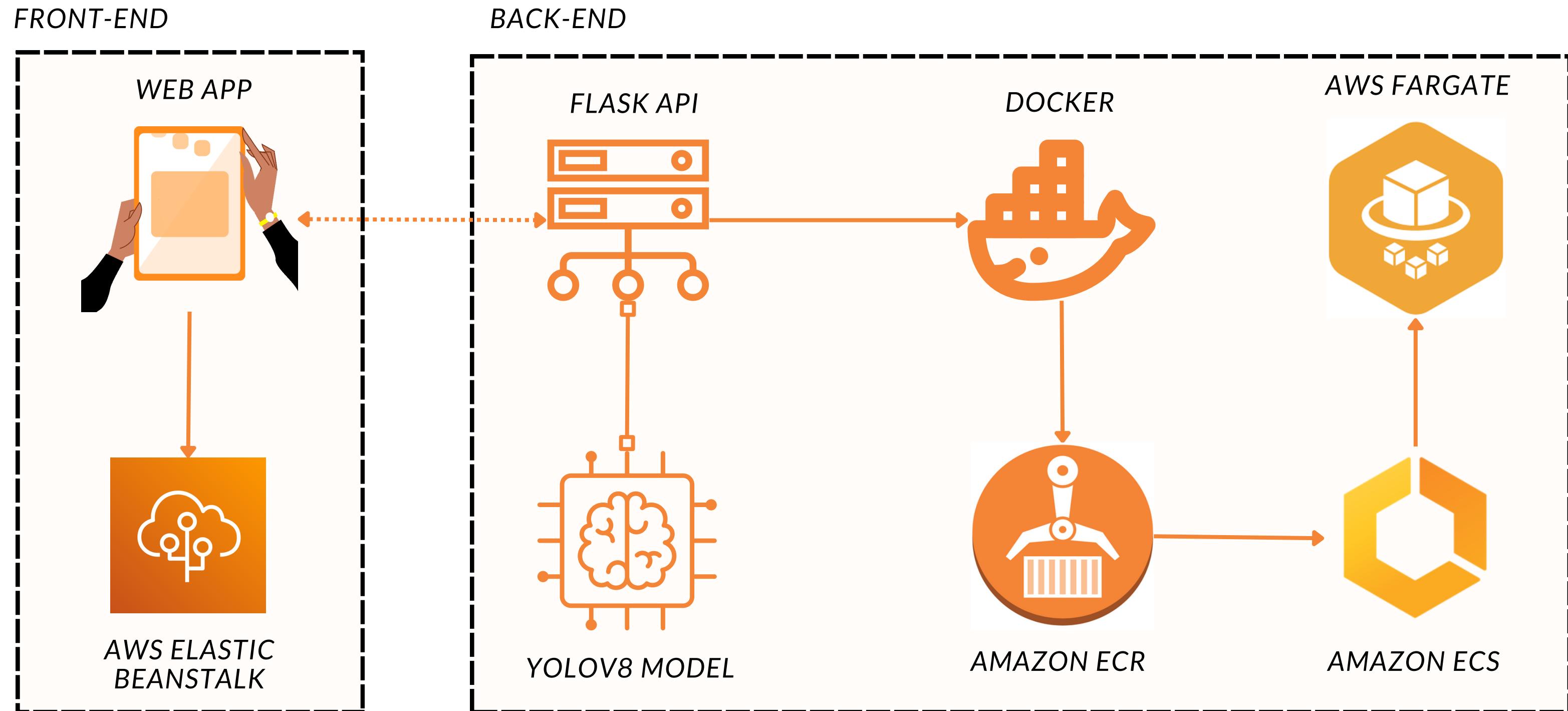
CONTENEURS



ARCHITECTURE DU SYSTÈME



OUTILS UTILISÉS



03 IMPLÉMENTATION

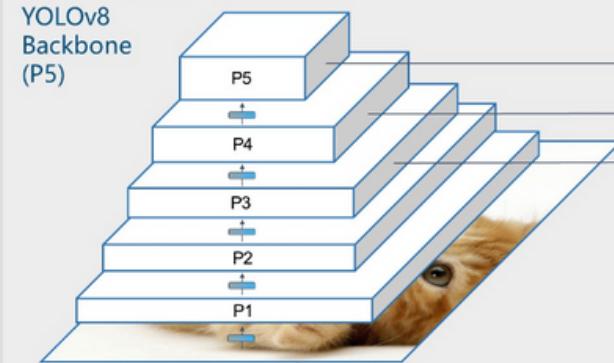
BACK-END

LE MODÈLE



YOLOv8

Backbone

YOLOv8
Backbone
(P5)

Conv
 $k=3, s=2, p=1$
0
P1

Conv
 $k=3, s=2, p=1$
1
P2

C2f
shortcut=True, $n=3 \times d$
2
P3

Conv
 $k=3, s=2, p=1$
3
P3

C2f
shortcut=True, $n=6 \times d$
4
Stride=8

Conv
 $k=3, s=2, p=1$
5
P4

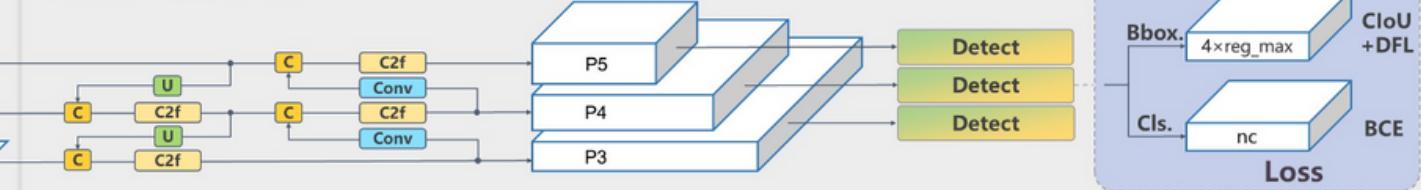
C2f
shortcut=True, $n=6 \times d$
6
Stride=16

Conv
 $k=3, s=2, p=1$
7
P5

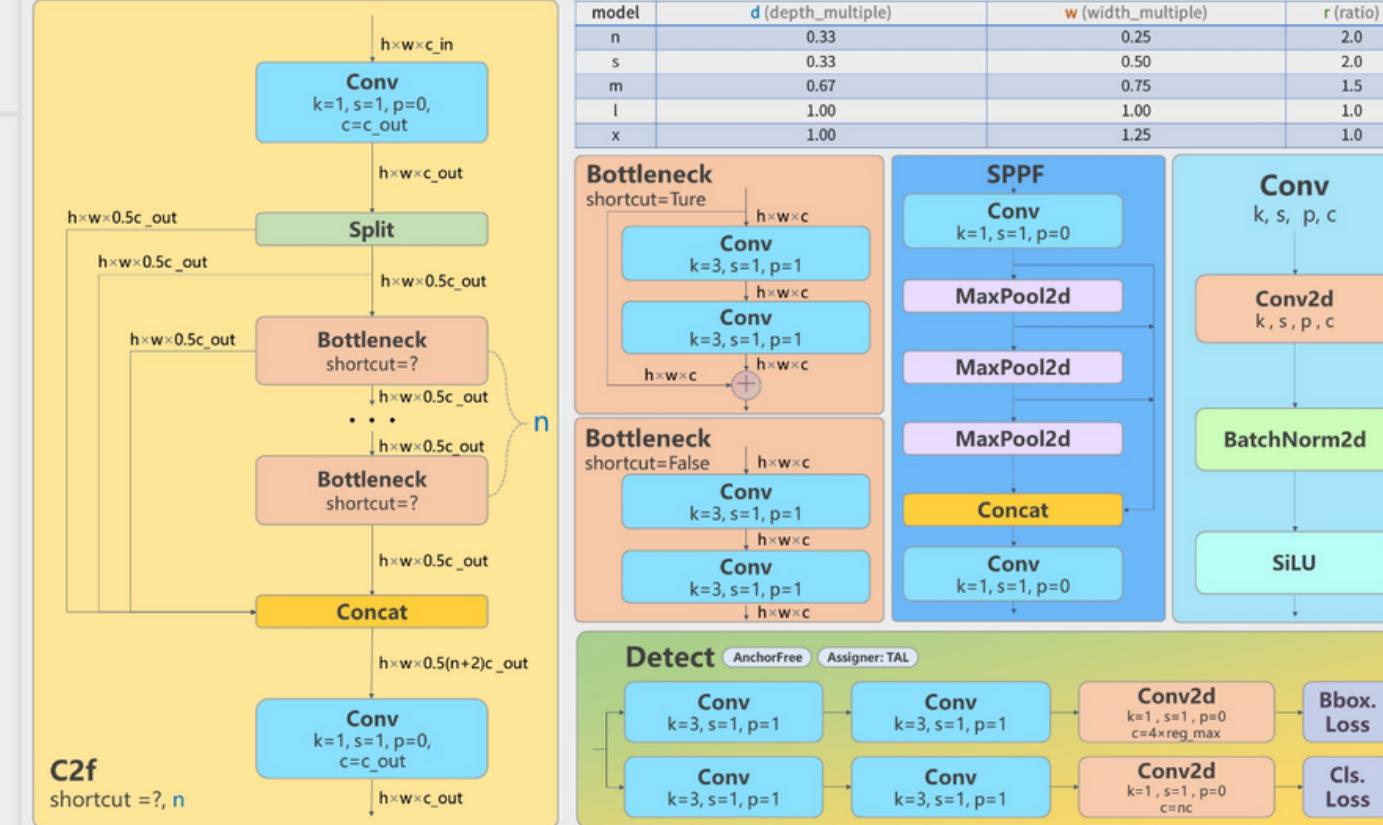
C2f
shortcut=True, $n=3 \times d$
8
P5

SPPF
Stride=32
9

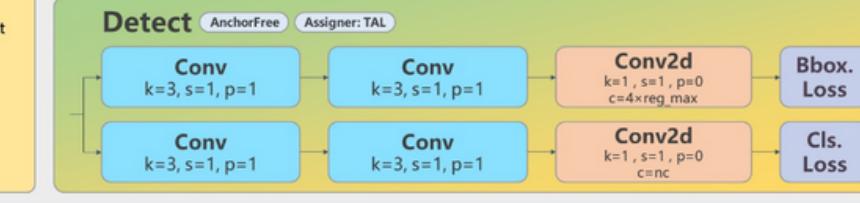
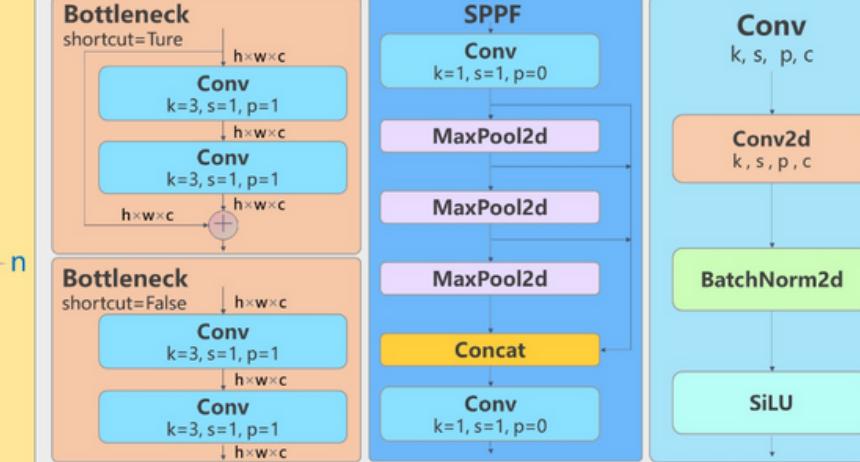
Head YOLOv8Head



Details



model	d (depth_multiple)	w (width_multiple)	r (ratio)
n	0.33	0.25	2.0
s	0.33	0.50	2.0
m	0.67	0.75	1.5
l	1.00	1.00	1.0
x	1.00	1.25	1.0



80x80x256xw → Detect

80x80x768xw → Concat → 80x80x256xw → Detect

80x80x512xw → Upsample → 40x40x512xw → C2f → 40x40x512xw → Concat → 40x40x768xw → Detect

40x40x512xw → C2f → 40x40x512xw → Concat → 40x40x768xw → Detect

40x40x512xw → C2f → 40x40x512xw → Concat → 40x40x768xw → Detect

40x40x512xw → C2f → 40x40x512xw → Detect

20x20x512wxr → Upsample → 20x20x512wxr → C2f → 20x20x512wxr → Concat → 20x20x512wxr → C2f → 20x20x512wxr → Detect

Backbone

Note:
height×width×channel

Head

L'API FLASK

```
@app.route('/api')
def api_home():
    return 'API is running'

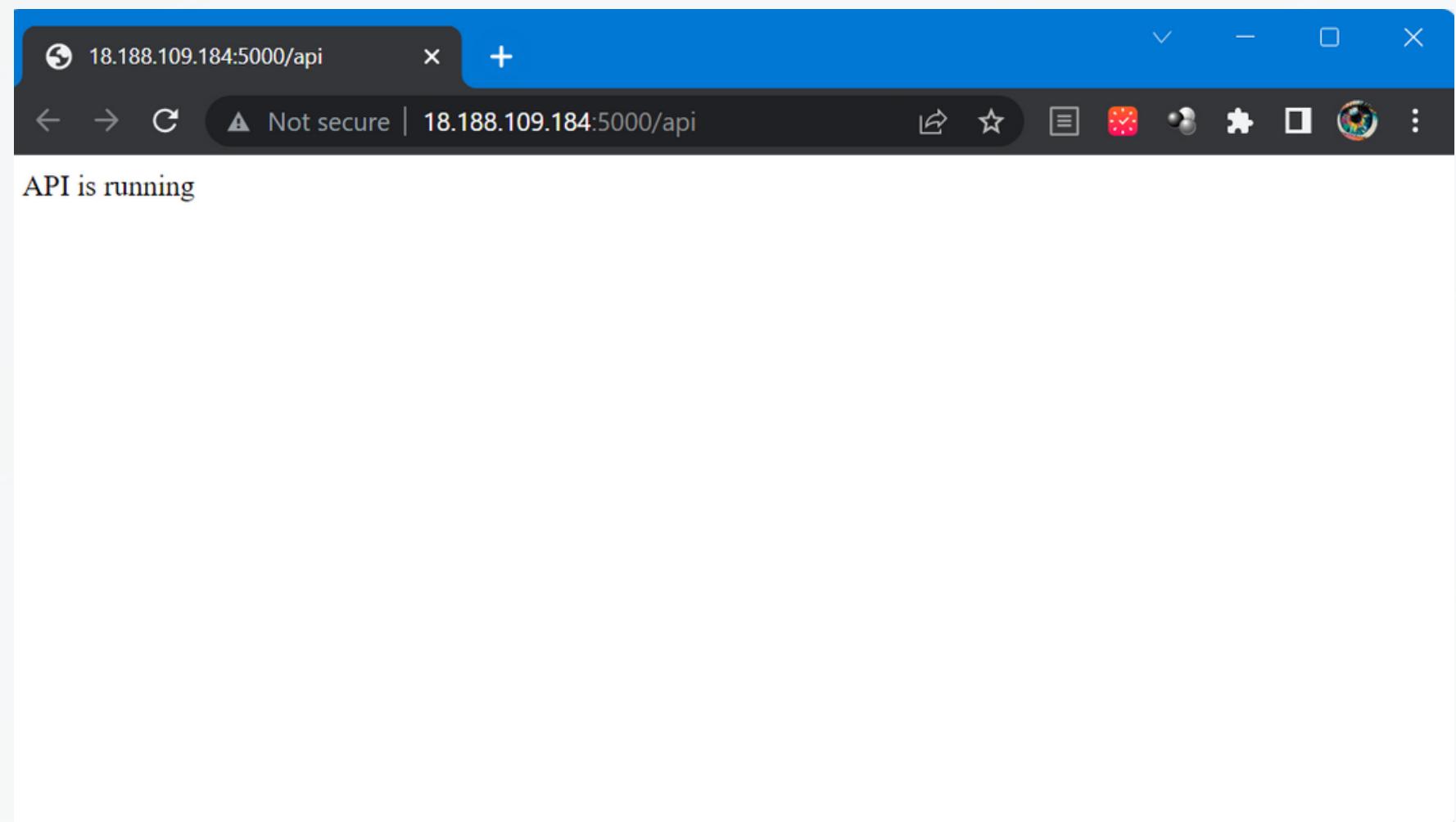
@app.route('/api/predict', methods=['POST'])
def api_predict():
    """
    Process the uploaded file and return the prediction as a base64 image
    """
    # Get the file from the request
    file = request.files['file']

    # Read the image and preprocess it
    image = processor.process_image(file)
    cv2.imwrite('img.png', image)

    # Make a prediction
    predicted_image = model.predict_image(image)

    # Convert predicted image to Base64
    encoded_image_data = processor.image_to_base64(predicted_image)

    return str(encoded_image_data)
```



LE CONTENEUR

```
1 # Choose a lightweight version of Python
2 FROM python:3.11.3-slim-buster
3
4 # Set Working directory
5 WORKDIR /app
6
7 # Copy requirements
8 COPY requirements.txt /app/
9
10 # OpenCV Dependencies
11 RUN apt-get update && apt-get install -y libgl1-mesa-glx libglib2.0-0
12
13 # No caching to keep docker image small
14 RUN pip3 install --no-cache-dir -r requirements.txt
15
16 # Copy files into the app folder
17 COPY . /app
18
19 # Expose the port where the Flask app will run
20 EXPOSE 5000
21
22 # Run the appdoc
23 CMD [ "python3", "-m" , "flask", "run", "--host=0.0.0.0"]
```



Amazon ECR > Repositories		
	Private	Public
Private repositories (2)		
<input type="text"/> Find repositories		
	Repository name	URI
<input type="checkbox"/>	bmi-api	934857846971.dkr.ecr.us-east-2.amazonaws.com/bmi-api
<input type="checkbox"/>	gun-detection-api	934857846971.dkr.ecr.us-east-2.amazonaws.com/gun-detection-api

LE CLUSTER

Amazon Elastic Container Service > Clusters > gun-detection-cluster > Services > gun-detection-service > Health

gun-detection-service [Info](#)

[C](#) [Update service](#) [Delete service](#)

[Health and metrics](#) [Logs](#) [Configuration and tasks](#) [Deployments and events](#) [Networking](#) [Tags](#)

Status [Info](#)

ARN	Status	Tasks	Deployments current state
gun-detection-cluster/gun-detection-service	Active	0 Pending, 1 Running / 1 Desired	1 Completed

Health

1h 3h 12h 1d 3d 1w Custom [C](#) Add to dashboard

CPU utilization
Percent
CPUUtilization Minimum (blue), CPUUtilization Maximum (orange), CPUUtilization Average (green)
04:00 05:00 06:00

Memory utilization
Percent
MemoryUtilization Minimum (blue), MemoryUtilization Maximum (orange), MemoryUtilization Average (green)
04:00 05:00 06:00



Tasks (1/1)

Filter tasks by property or value [Running tasks](#) All launch types

Task	Last status	Desired status	Task definition
f7696932bd5645a0ab5e8177c6a4bb72	Running	Running	gun-detection-task

Containers for task f7696932bd5645a0ab5e8177c6a4bb72

Containers (1)

Container name	Container runtime ID	Image URI
gun-detection-api	f7696932bd5645a0ab5e8177c6a4bb72	934857846971.dkr.ecr.us-east-2.amazonaws.com/gun-detection-api:latest

L'INFRASTRUCTURE



AMAZON EC2



Service configuration Info

Service ARN
gun-detection-cluster/gun-detection-service

Task definition: revision
[gun-detection-task:2](#)

Service type
REPLICA

► Service discovery

Service Connect Info
[Configure](#)

Capacity provider
FARGATE

Created by
arn:aws:iam::934857846971:root

Auto Scaling

Desired tasks
1

Min tasks
-

(i) No Auto Scaling resources configured for this service.

gun-detection-service [Info](#)

Update service

Delete service

Health and metrics

Logs

Configuration and tasks

Deployments and events

Networking

Tags

Deployment configuration [Info](#)

View pipelines

Deployment status

Completed

Deployment type

ECS

Platform version

LATEST

Min and max running tasks

100% min and 200% max

▶ Deployment failure detection

▼ Task placement strategy and constraints

Task placement strategy

Task placement constraint

Deployments (1) [Info](#)

Filter deployments

< 1 >

Start date

Status

Failed tasks

Tasks

Version

Task definition

Revision

Last deployment

4/23/2023, 2:55:20 AM

Primary

100%

0

1 Running | 0 Pending | 1 Desired

1.4.0

gun-detection-task

2

Completed

Events (6)

Filter events by value

< 1 >

Started at

Message

Event ID

4/23/2023, 9:01:27 PM

service gun-detection-service has reached a steady state.

26da137b-b522-40e3-9dd2-1f707fe78ffa

4/23/2023, 3:00:59 PM

service gun-detection-service has reached a steady state.

f276cbac-957f-49da-80c4-04a8a6aedd74

4/23/2023, 9:00:56 AM

service gun-detection-service has reached a steady state.

789efa63-2aed-4c9a-a983-7a024ce06e86

4/23/2023, 3:00:27 AM

service gun-detection-service has reached a steady state.

02ec08a2-4525-422f-88bd-3fe2963850ec

4/23/2023, 3:00:27 AM

service gun-detection-service deployment ecs-svc/4147796273202384607 deployment completed.

3e99343d-a6af-4db7-8d38-5f60826a4a07

4/23/2023, 2:55:21 AM

service gun-detection-service has started 1 tasks: task f7696932bd5645a0ab5e8177c6a4bb72.

5ff32cfa-6a3a-4990-810f-d17a08a9cae

FRONT-END

L'INTERFACE

```
const fileInput = document.getElementById('file-input');
const submitBtn = document.getElementById('submit-btn');
const imageOutput = document.getElementById('image-output');

API_URL = 'http://18.188.109.184:5000/api'

submitBtn.addEventListener('click', async (event) => {
    // prevent the page from reloading
    event.preventDefault();

    // get the file and prepare it for sending
    const file = fileInput.files[0];
    const formData = new FormData();
    formData.append('file', file);

    postImage(formData)

});
```



Home

 **Smart Eye** Deep Demographics 24/4/2023

Guns Detection with YOLOv8

Choose File No file chosen **Submit**

Predictions will be shown below...

ELASTIC BEANSTALK

Elastic Beanstalk > Applications

Info
Your application is being deleted.

All applications

Filter results matching the display values

Application name	Environments	Date created	Last modified	ARN
Test	Test-env	2023-04-24 01:34:24 UTC-0400	2023-04-24 01:34:24 UTC-0400	arn:aws:elasticbeanstalk:us-east-2:934857846971:application/Test
gun-detection-webapp		2023-04-24 00:20:37 UTC-0400	2023-04-24 00:20:37 UTC-0400	arn:aws:elasticbeanstalk:us-east-2:934857846971:application/gun-detection-webapp



Home

 **Smart Eye** Deep Demographics 24/4/2023

Guns Detection with YOLOv8

Choose File No file chosen **Submit**

Predictions will be shown below...

04 TESTS

IMAGES

Home

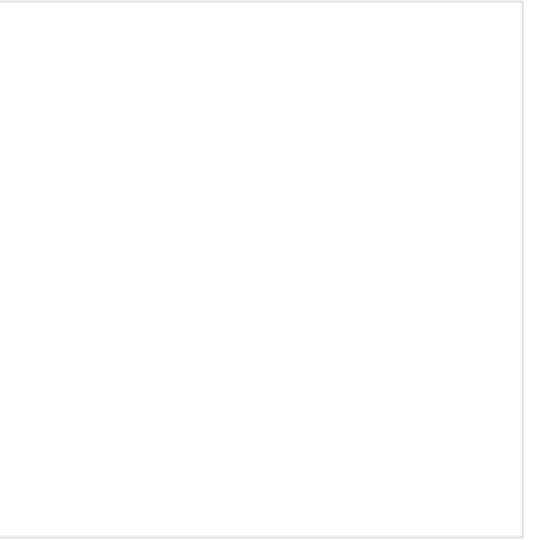
 **Smart Eye**
Deep Demographics

24/4/2023

Guns Detection with YOLOv8

No file chosen

Predictions will be shown below...



Home

 **Smart Eye**
Deep Demographics

24/4/2023

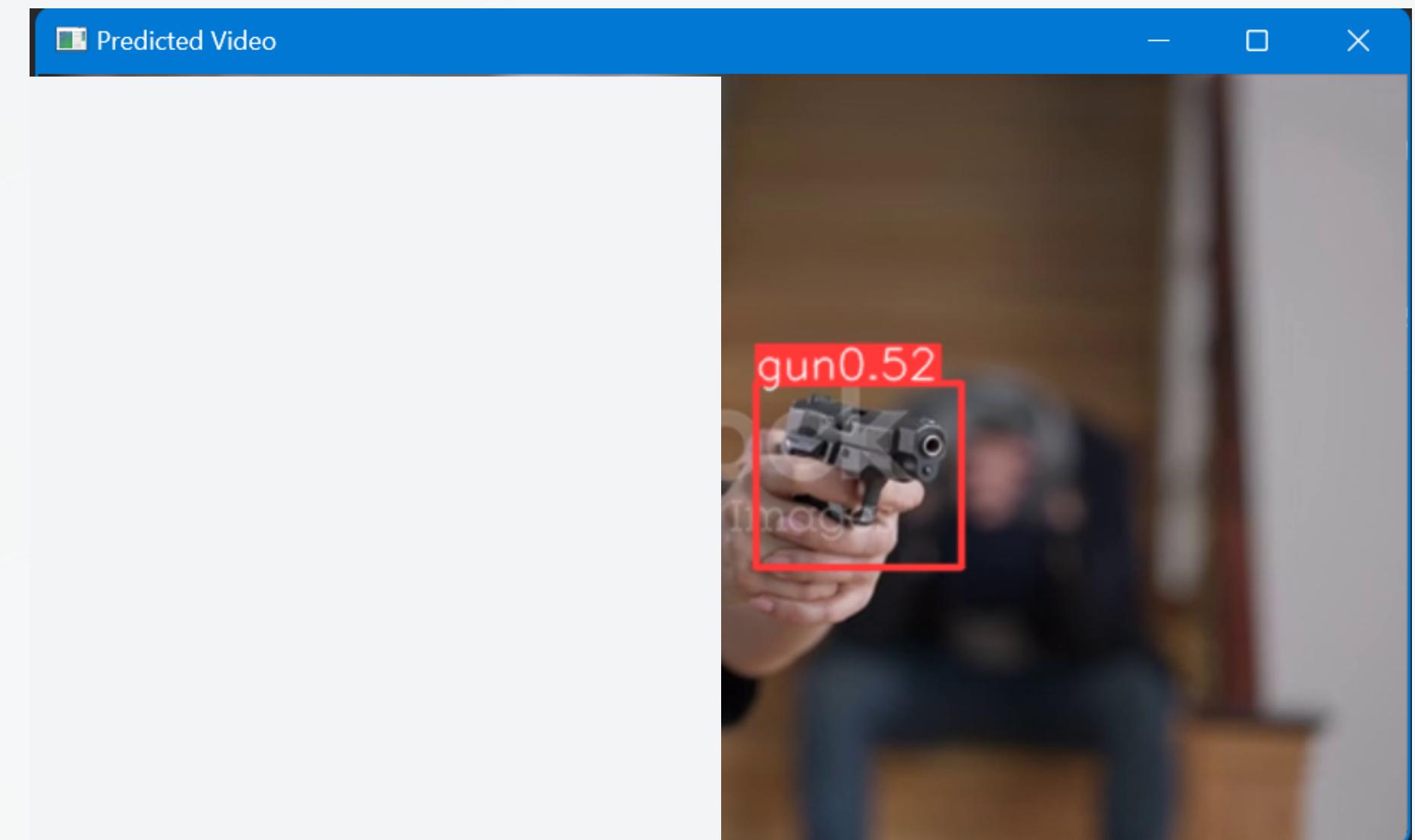
Guns Detection with YOLOv8

4529570706...f70f6fc_o.jpg



VIDÉOS

VOIR DÉMO



05 OBSTACLES

ELASTIC BEANSTALK

Elastic Beanstalk > Environments > Test-env

Test-env
Test-env.eba-uvrverb4.us-east-2.elasticbeanstalk.com (e-njnnptug7i)
Application name: Test

Health **Running version** **Platform**

Degraded **Sample Application** Node.js 18 running on 64bit
[Upload and deploy](#) Amazon Linux 2/5.8.0
[Causes](#) [Change](#)

Recent events [Show all](#)

Time	Type	Details
2023-04-24 02:31:49 UTC-0400	WARN	Environment health has transitioned from Info to Degraded. Command failed on all instances. Incorrect application version found on all instances. Expected version "Sample Application" (deployment 1). Application update is aborting. 1 out of 1 instance completed (running for 52 seconds).
2023-04-24 02:30:56 UTC-0400	ERROR	During an aborted deployment, some instances may have deployed the new application version. To ensure all instances are running the same version, re-deploy the appropriate application version.
2023-04-24 02:30:56 UTC-0400	ERROR	Failed to deploy application.
2023-04-24 02:30:56 UTC-0400	ERROR	Unsuccessful command execution on instance id(s) 'i-03e01c99c31292db3'. Aborting the operation.
2023-04-24 02:30:56 UTC-0400	INFO	Command execution completed on all instances. Summary: [Successful: 0, Failed: 1].

06 CONCLUSION



01

APPRENTEISSAGES

02

LESSONS

03

PERSPECTIVES

MERCI
DE VOTRE ATTENTION ;)

