الجامعة الإسلامية بالمدينة المنورة
ISLAMIC UNIVERSITY OF MADINAH
كلية الحاسب الآلي ونظم المعلومات
Faculty of Computer and Information Systems

An Intelligent Weapon Detection System for Surveillance Cameras

Graduation Project II (Midterm Presentation)

Team Members

Name	ID
Mohamed Nasser Hashem	372029063
Mustafa Ahmed Abdulsami	372029353
Husen Muhammad Kalaepeh	352072163

Supervised By

Dr. Emad Nabil

Agenda

Introduction

- Importance
- Problem statement
- Aim & Objectives
- Methodology
- Timeline

Background

- Object Detection
- Algorithms
- Datasets

Analysis & Design

- Requirement
- Actors
- Graphical User Interfaces
- Unified Modelling Language

Implementation

- Tools
- Screenshots
- Demo Code

Introduction

Introduction

Importance & Problem Statement

- Security threats have become common and real in this time, humans are no longer able to protect valuables.
- On the other hand, the presence of humans has become more intense in public places. With so many criminals impossible to discern them by visual ability,
- For this, the developers and the police are working to secure it both technically and realistically. Because security alone is unable to deal with such matters

Valuables Insurance

Social Insurance

Teams

Introduction

Aim

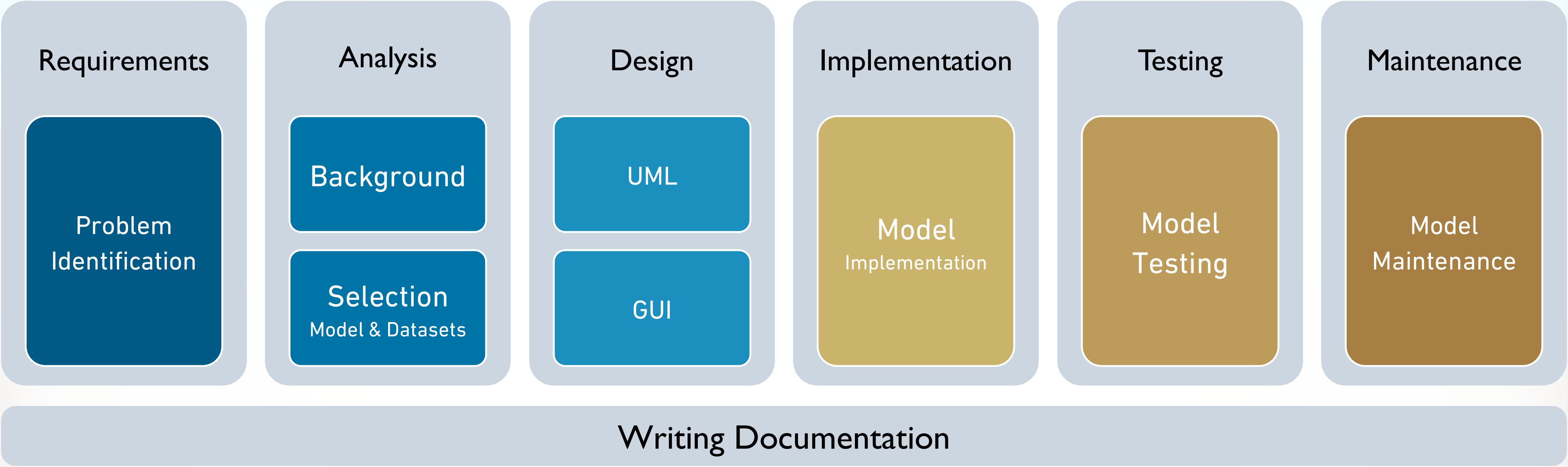
Create a system connected to a camera that can identify the criminal who enters public places, by identifying if he is carrying a gun, trifle, or a knife. and to give a warning to all those present in public place and warn them.

Objectives

1. Create model(s) for threat/Weapon Detection
2. Train the model(s) using relevant data sets
3. Measure the performance of the model(s).
4. Enhance the system until it reaches an accepted accuracy rate.

Introduction

Methodology



Requirements

Problem Identification

Analysis

Background

Selection
Model & Datasets

Design

UML

GUI

Implementation

Model
Implementation

Testing

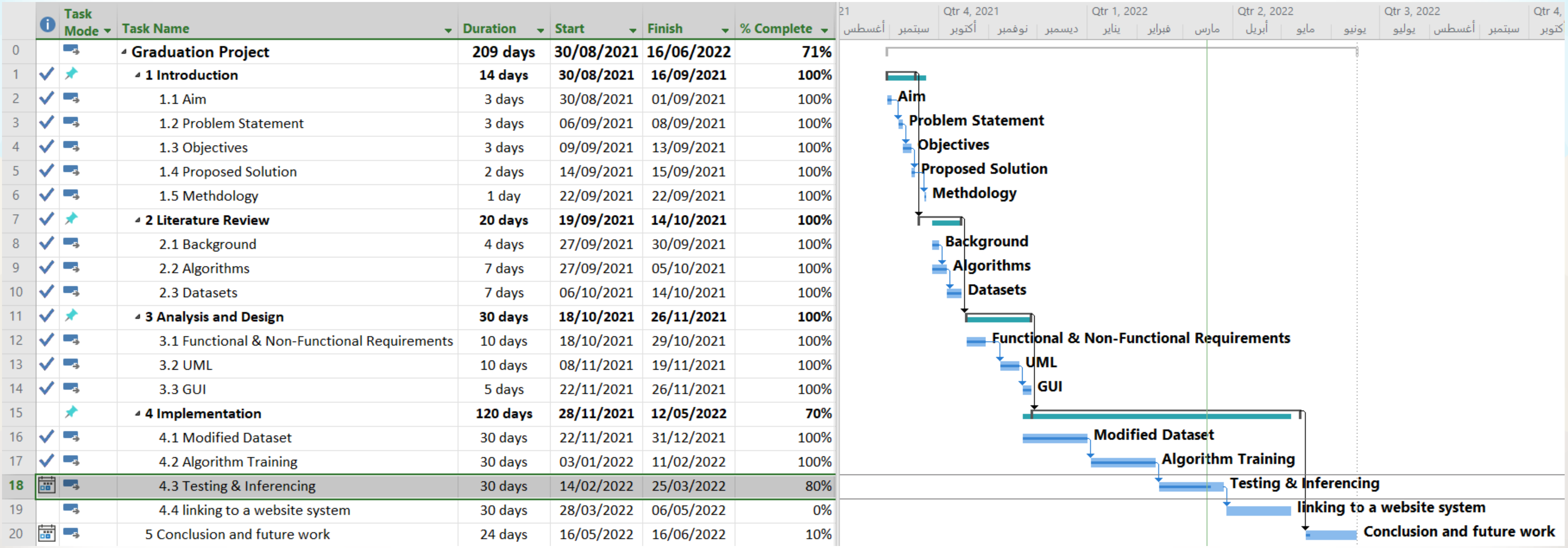
Model
Testing

Maintenance

Model
Maintenance

Writing Documentation

Introduction | Timeline



	ⓘ	Task Mode	Task Name	Duration	Start	Finish	% Complete
0		⇒	Graduation Project	209 days	30/08/2021	16/06/2022	71%
1	✓	📌	1 Introduction	14 days	30/08/2021	16/09/2021	100%
2	✓	⇒	1.1 Aim	3 days	30/08/2021	01/09/2021	100%
3	✓	⇒	1.2 Problem Statement	3 days	06/09/2021	08/09/2021	100%
4	✓	⇒	1.3 Objectives	3 days	09/09/2021	13/09/2021	100%
5	✓	⇒	1.4 Proposed Solution	2 days	14/09/2021	15/09/2021	100%
6	✓	⇒	1.5 Methdology	1 day	22/09/2021	22/09/2021	100%
7	✓	📌	2 Literature Review	20 days	19/09/2021	14/10/2021	100%
8	✓	⇒	2.1 Background	4 days	27/09/2021	30/09/2021	100%
9	✓	⇒	2.2 Algorithms	7 days	27/09/2021	05/10/2021	100%
10	✓	⇒	2.3 Datasets	7 days	06/10/2021	14/10/2021	100%
11	✓	📌	3 Analysis and Design	30 days	18/10/2021	26/11/2021	100%
12	✓	⇒	3.1 Functional & Non-Functional Requirements	10 days	18/10/2021	29/10/2021	100%
13	✓	⇒	3.2 UML	10 days	08/11/2021	19/11/2021	100%
14	✓	⇒	3.3 GUI	5 days	22/11/2021	26/11/2021	100%
15		📌	4 Implementation	120 days	28/11/2021	12/05/2022	70%
16	✓	⇒	4.1 Modified Dataset	30 days	22/11/2021	31/12/2021	100%
17	✓	⇒	4.2 Algorithm Training	30 days	03/01/2022	11/02/2022	100%
18	📅	⇒	4.3 Testing & Inferencing	30 days	14/02/2022	25/03/2022	80%
19		⇒	4.4 linking to a website system	30 days	28/03/2022	06/05/2022	0%
20	📅	⇒	5 Conclusion and future work	24 days	16/05/2022	16/06/2022	10%

Background

Background

Object Detection

Define:

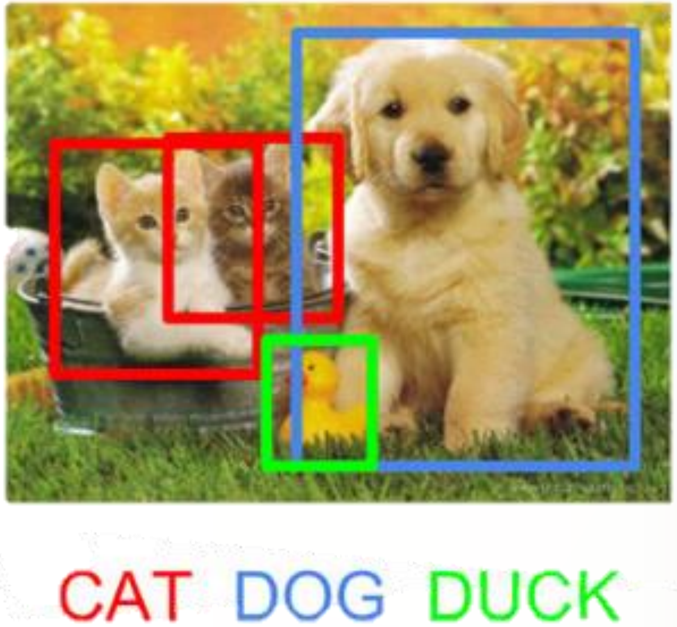
Using a bounding box, determine the presence of things in an image and the types or classes of the objects discovered

Input

an image with one or more things, such as a photograph, is utilized

Output

One or more bounding boxes (e.g., specified by a point, width, and height)



Background

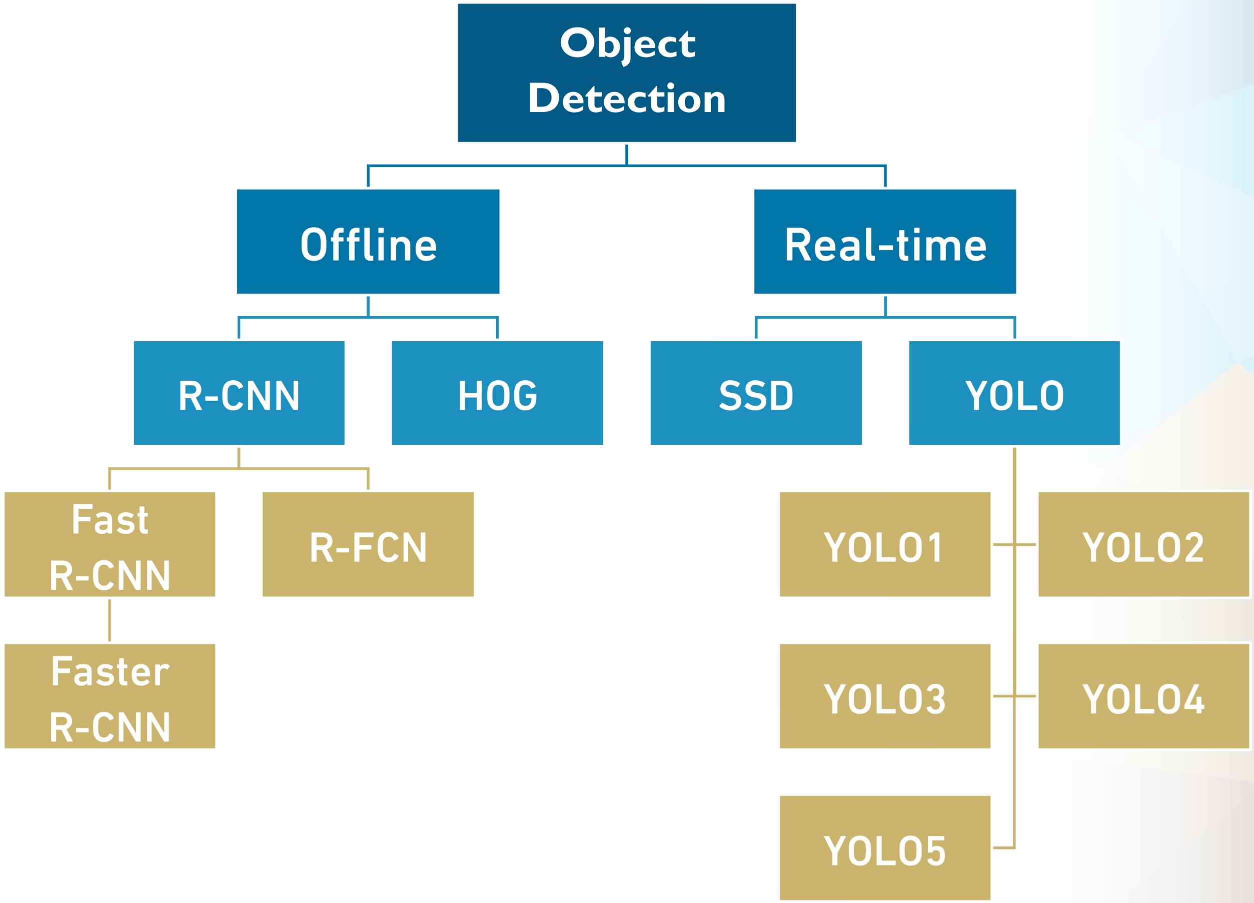
Object Detection: Types

Real-time

Task of doing object detection in real-time with fast inference while maintaining a base level of accuracy

Offline

The task of doing the detection of objects, mainly depends on the strength of performance

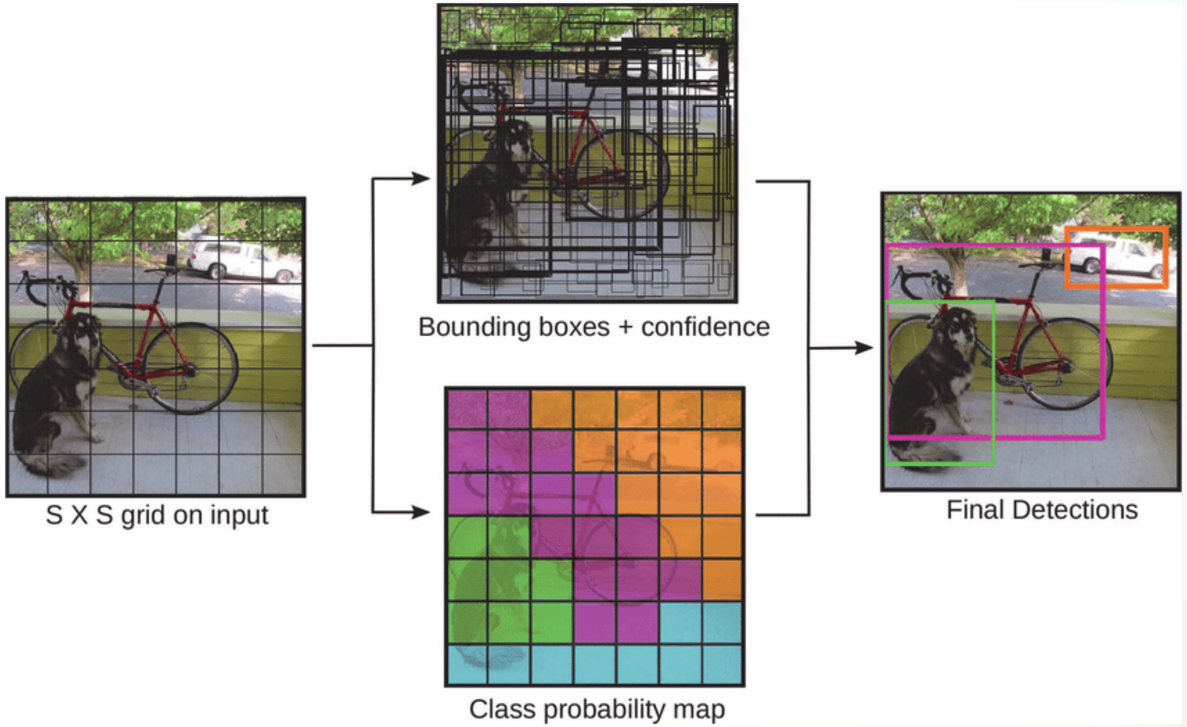


Background

Algorithms

You Just Look Once (YOLO): 2019

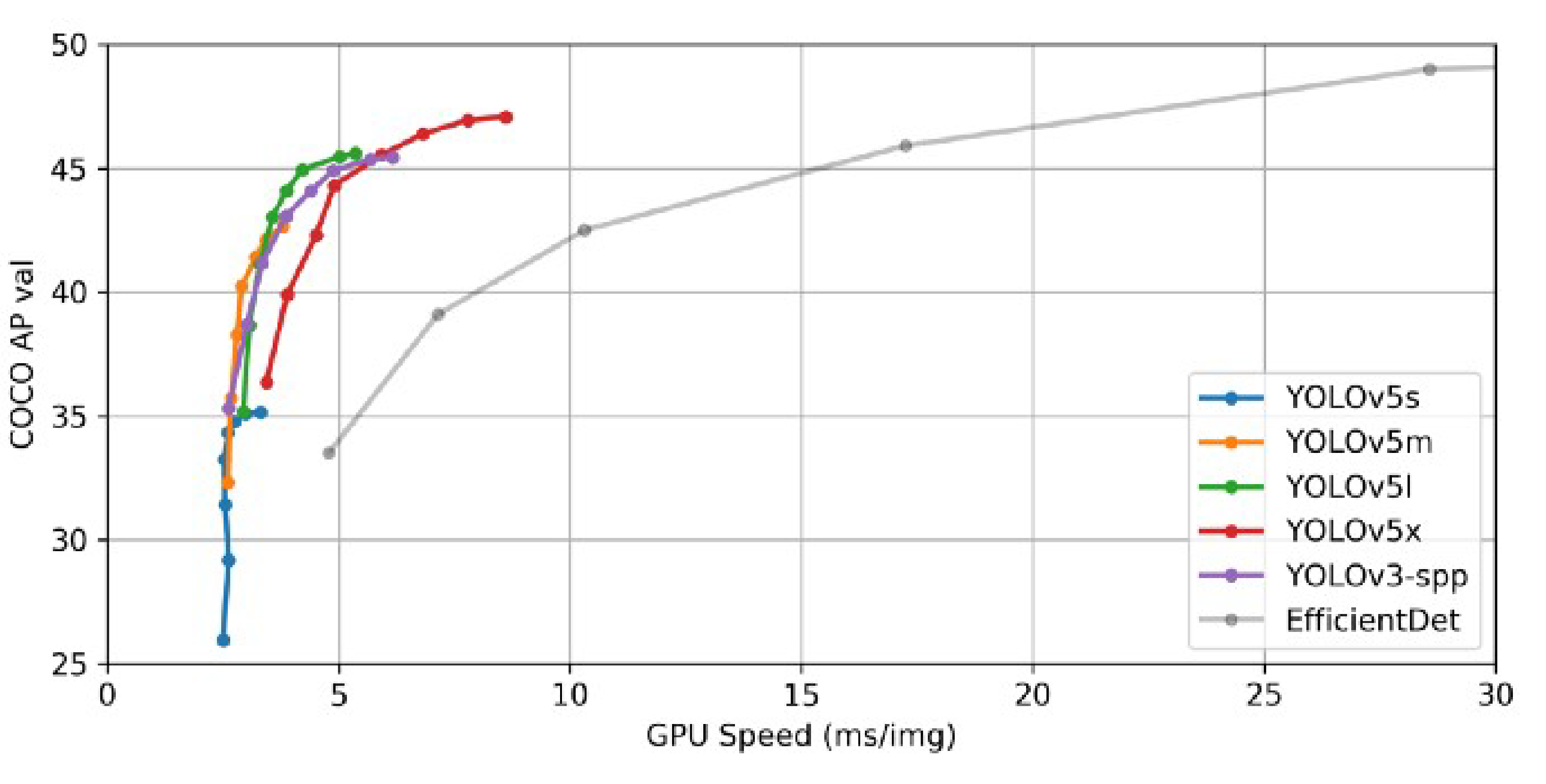
A single neural network is used in an object detection method. In contrast to several other object detection algorithms that do a bit-by-bit scan of the picture.



Background

Algorithms

YOLO models comparison



Background

Algorithms

YOLOv5: 2020

The YOLOv5 model is the most current addition to the YOLO family of models. It was created and is maintained using the Darknet framework. YOLOv5 is the first YOLO model to be written in the PyTorch framework, making it significantly lighter and easier to use.

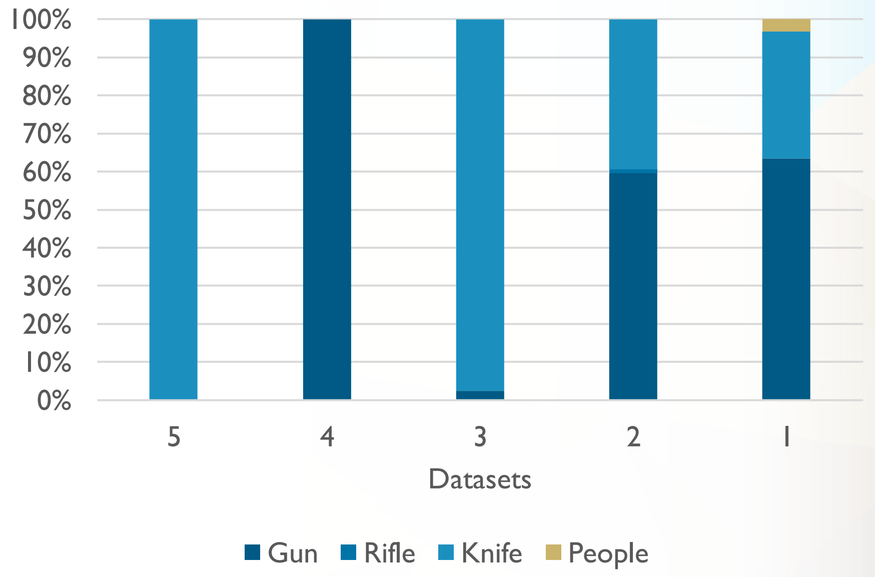
PYTORCH

Deep Learning with PyTorch

Background

Datasets

Dataset	Gun	Rifle	Knife	People	Total
Crime Detection	2000	-	1050	100	**3150**
Weapon detection	10770	187	7140	-	**18097**
Weapon Classification	315	-	12900	-	**13215**
Handgun Dataset	1900	-	-	-	**1900**
Knife Dataset	-	-	500	-	**500**
Total Datasets					**36862**



Analysis & Design

Analysis & Design

Function Requirement

1. Login
2. Create account
3. Identify a person holding a weapon
4. Create an alarm if a weapon detected
5. Display camera stream
6. Logout

Non-Function Requirement

1. The system security
2. Visibility all 24/7
3. Accuracy in recognizing OD
4. Speed in recognizing OD
5. Preserve the privacy of the place

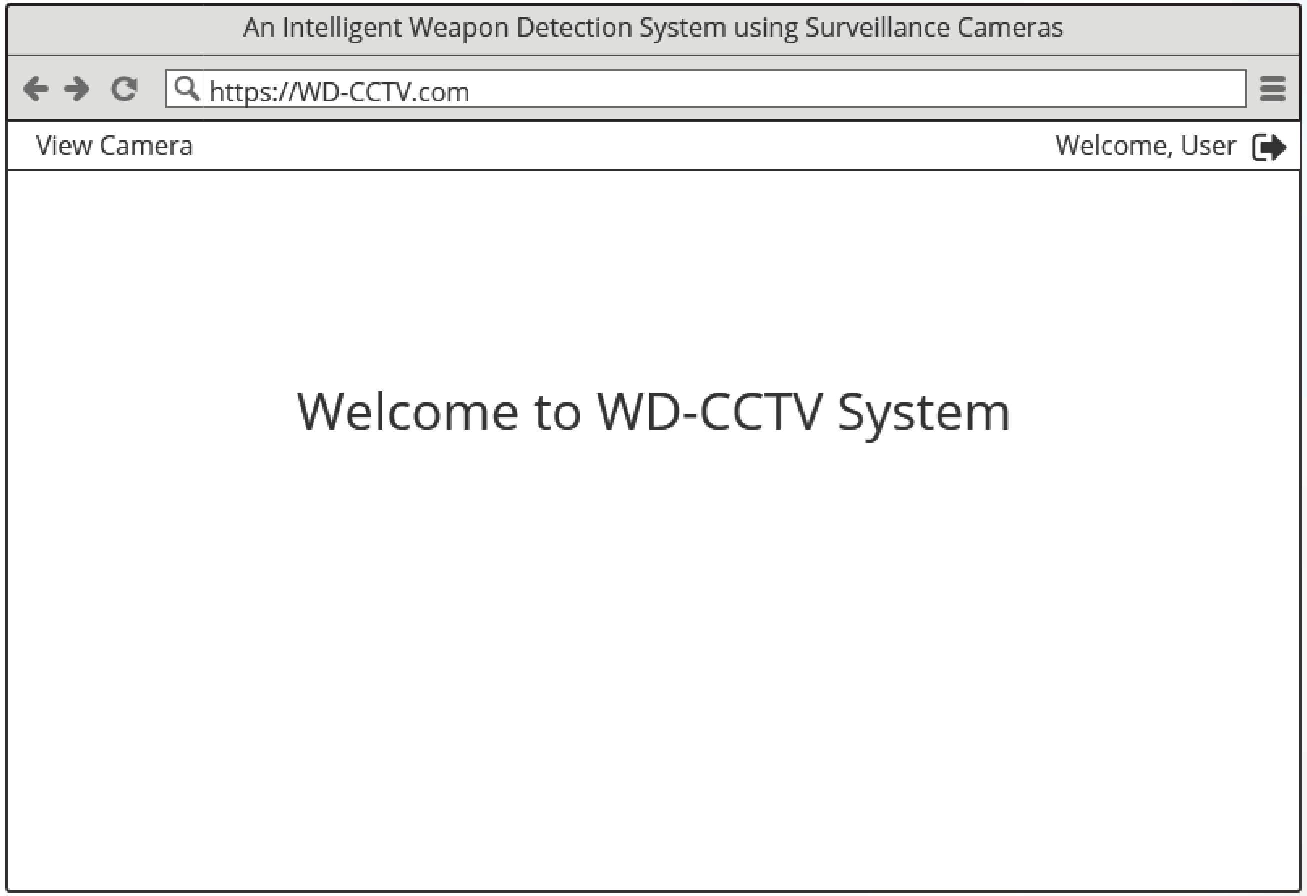
Analysis & Design

Actors

- System
- Admin
- Users

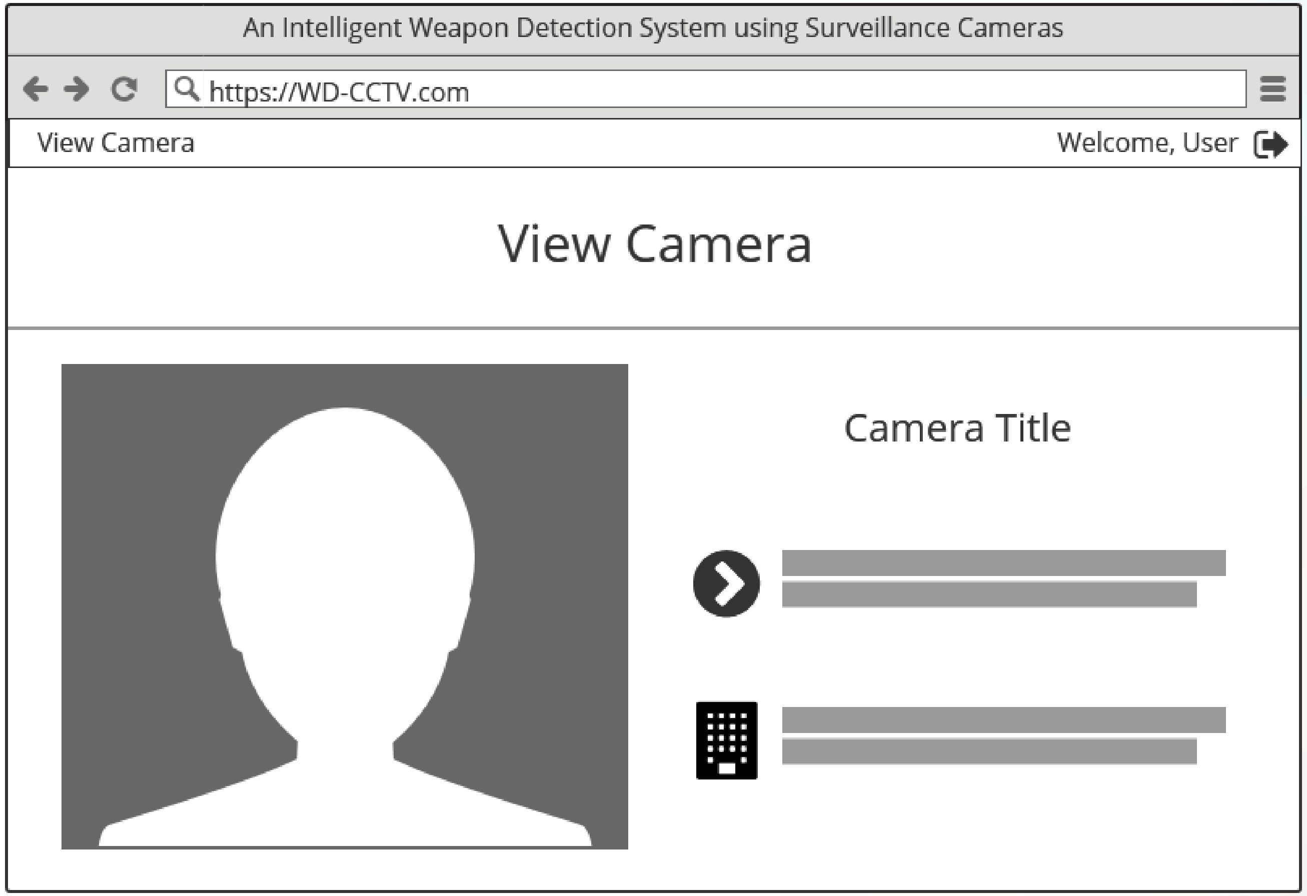
Analysis & Design

GUI | Main page (User)


An Intelligent Weapon Detection System using Surveillance Cameras
https://WD-CCTV.com
View Camera
Welcome, User
Welcome to WD-CCTV System

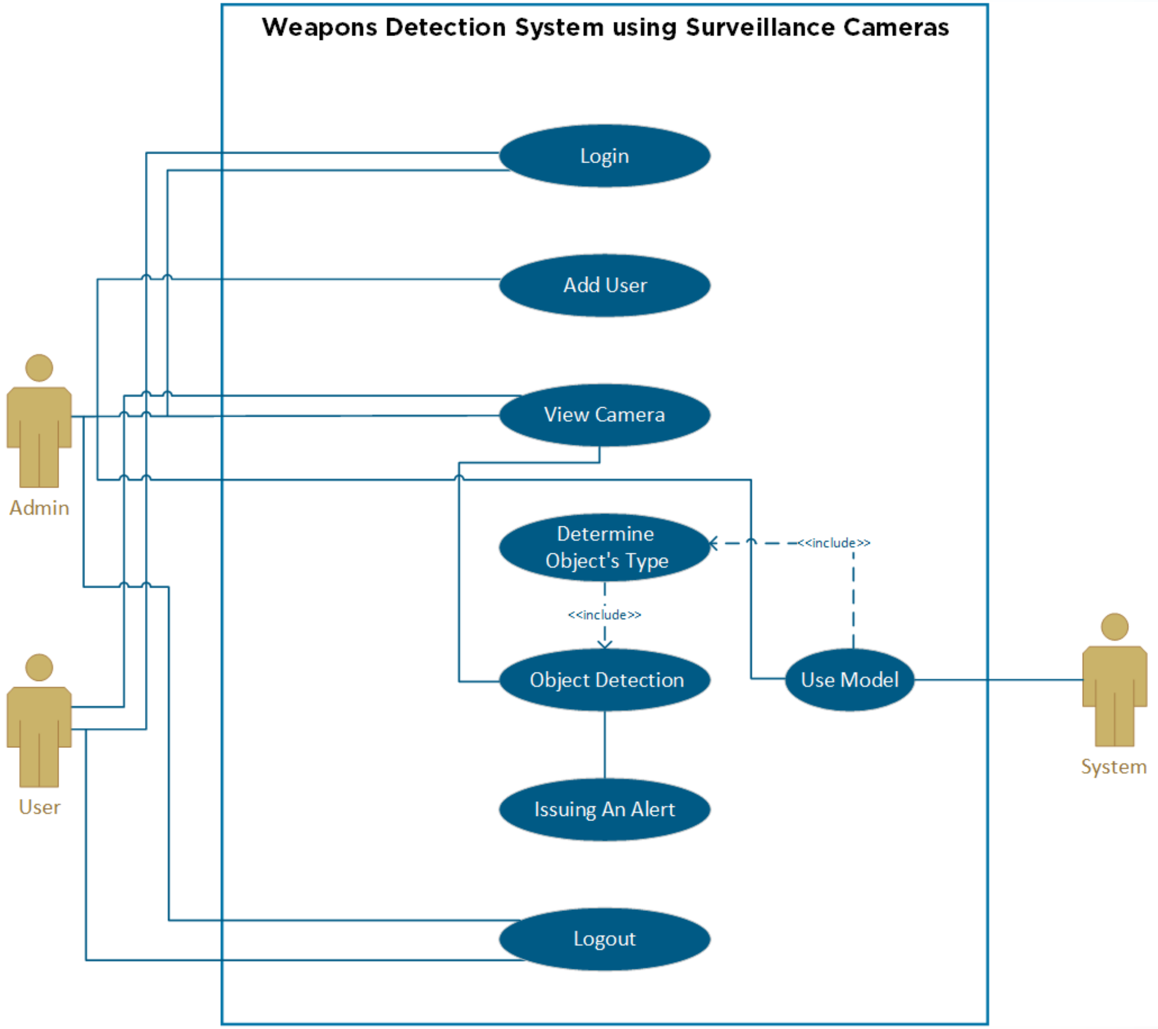
Analysis & Design

GUI | View Camera (User)



Analysis & Design

UML | Use-Case



Implementation

Implementation

Modified Dataset

Dataset type	Ful datasets	1st sample
Knife	21500	2570
Gun	15000	3240
Riffle)	190	190
Total Datasets	**37000**	**6000**

Types of Object detection annotation

1. Pascal VOC (.XML)
2. CreateML (.JSON)
3. YOLO (.TXT)

Implementation

YOLO Dataset annotation format

An Object detection annotation format, that having a text file per each picture (containing the annotations and a numeric representation of the label) and a label map (which translates the numeric IDs to human readable strings) are included in this format. The annotations are normalized to lie between 0 and 1, making them easier to deal with even after resizing or extending the photos .

Img.txt	0 0.716797 0.395833 0.216406 0.147222
	1 0.687109 0.379167 0.255469 0.158333
	2 0.420312 0.395833 0.140625 0.166667

Implementation

How to get YOLO format

1. **XML → YOLO**

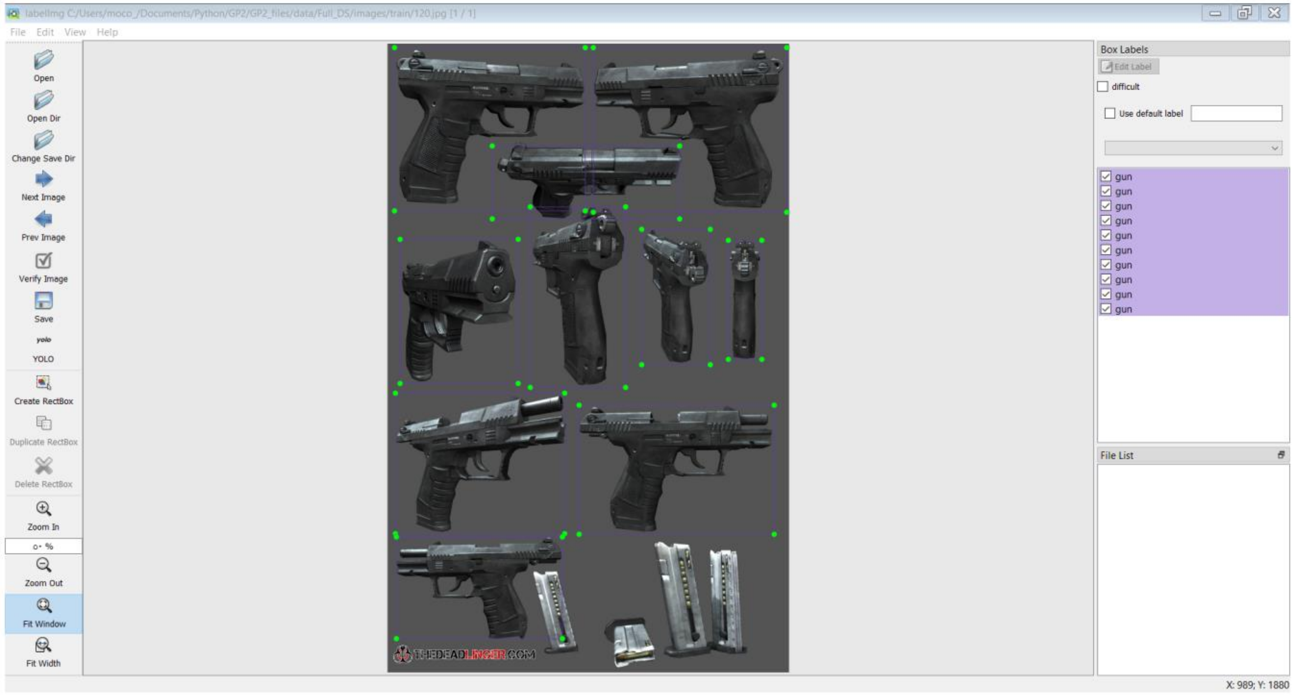
   In fact, XML is the most used format to annotate dataset. The 1st way to prepare it for our project.

2. **Generate YOLO format**

   The second way to get annotate dataset is by generate it with software called LabelImg. that is a graphical image annotation tool. saves annotation as XML, CreateML, and YOLO formats.

Implementation

How to get YOLO format | Generate YOLO format



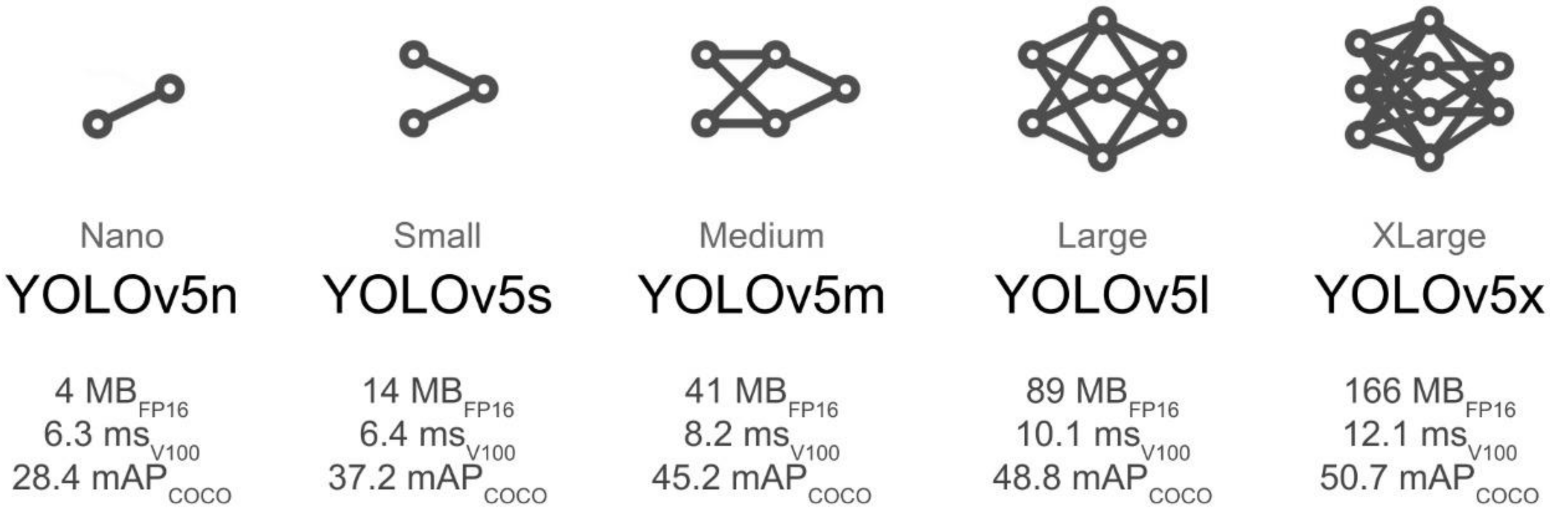
□ difficult

□ Use default label

☑ gun
☑ gun
☑ gun
☑ gun
☑ gun
☑ gun
☑ gun
☑ gun
☑ gun
☑ gun

Implementation

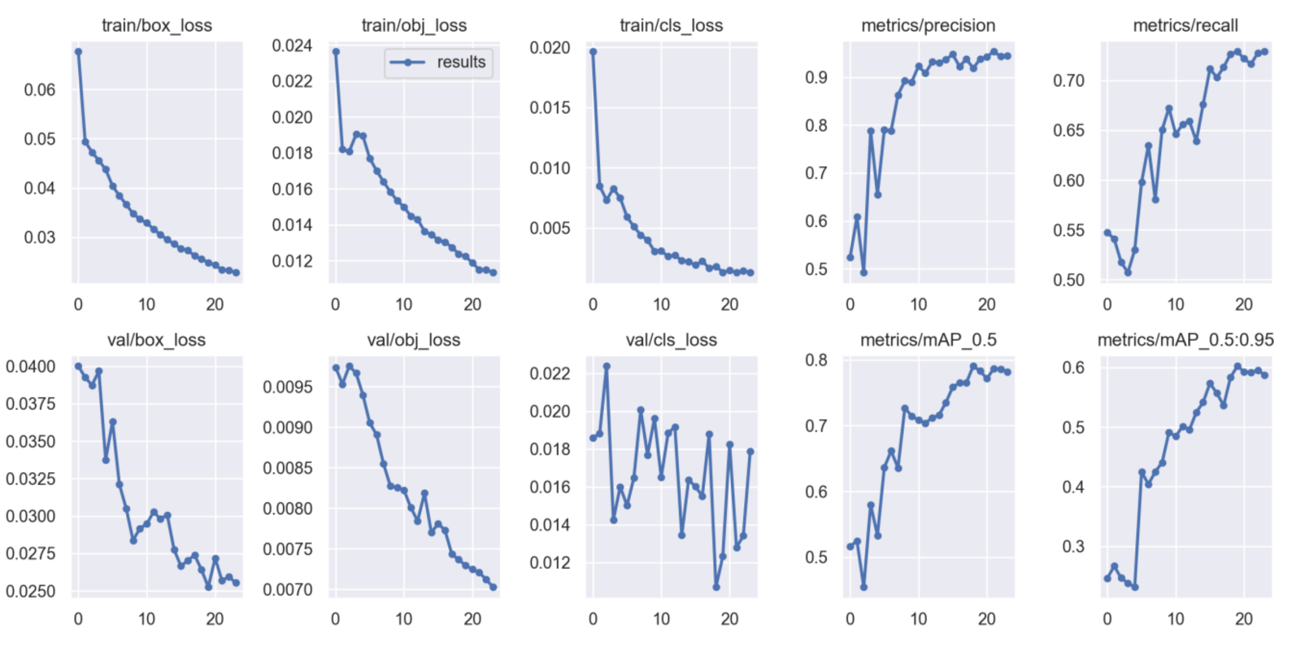
Algorithm Training



Nano	Small	Medium	Large	XLarge
YOLOv5n	YOLOv5s	YOLOv5m	YOLOv5l	YOLOv5x
4 MB_{FP16} 6.3 ms_{V100} 28.4 mAP_{COCO}	14 MB_{FP16} 6.4 ms_{V100} 37.2 mAP_{COCO}	41 MB_{FP16} 8.2 ms_{V100} 45.2 mAP_{COCO}	89 MB_{FP16} 10.1 ms_{V100} 48.8 mAP_{COCO}	166 MB_{FP16} 12.1 ms_{V100} 50.7 mAP_{COCO}

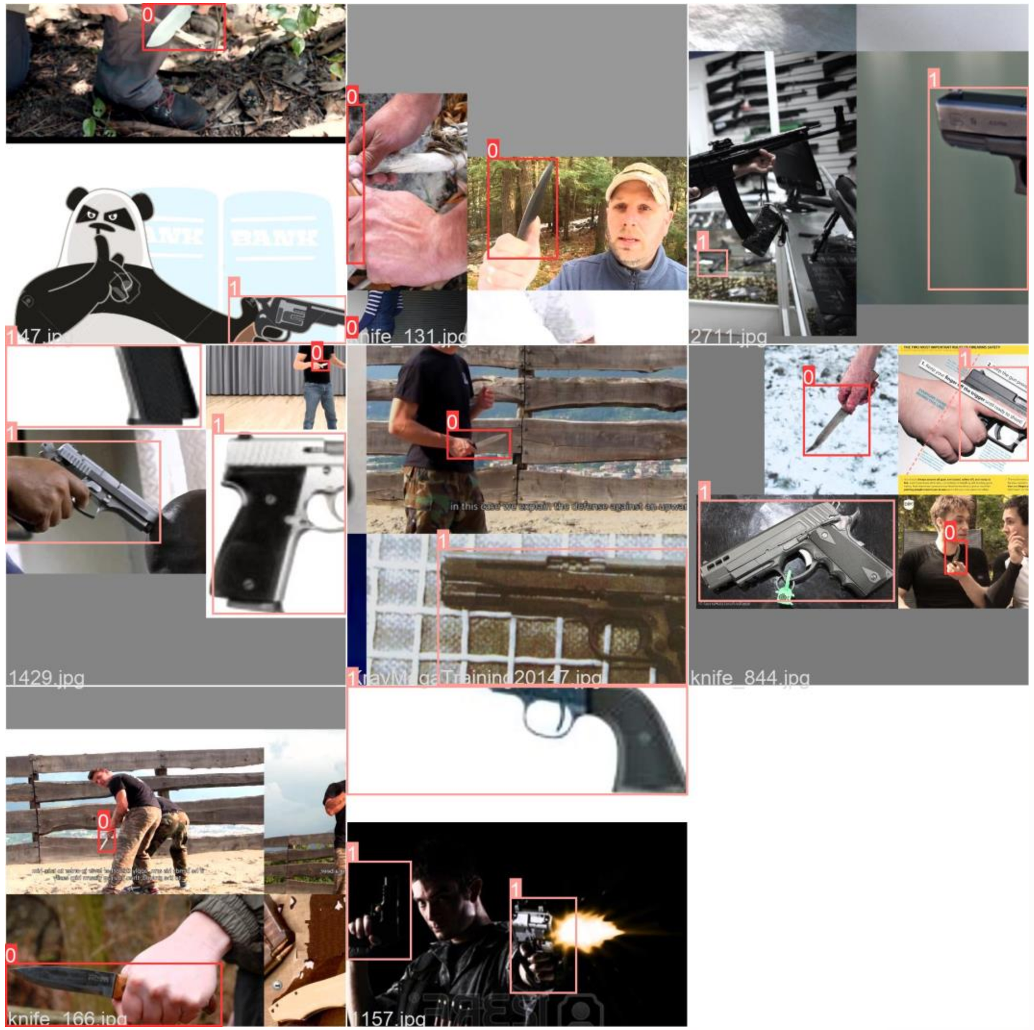
Implementation

1st Training: YOLOv5m



Implementation

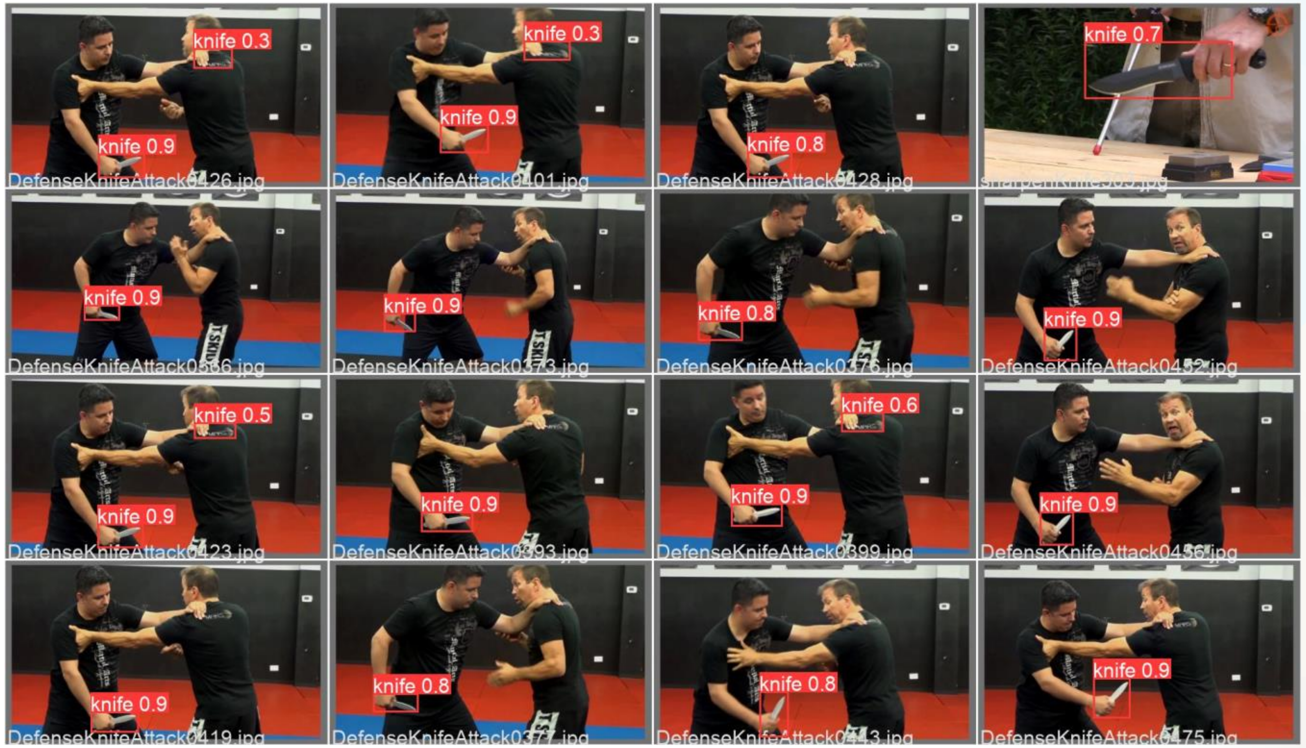
1st Training: YOLOv5m



Trained batch of trained model

Implementation

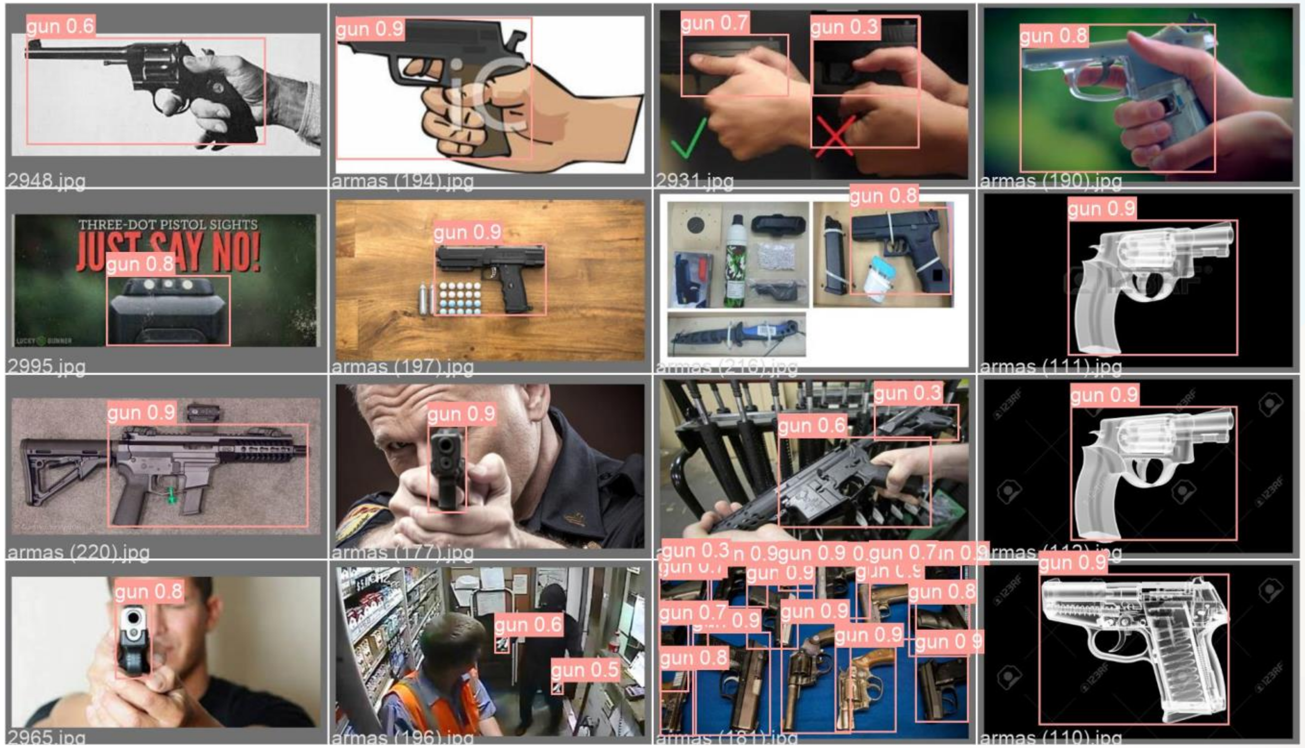
1st Training: YOLOv5m


knife 0.3
knife 0.9
DefenseKnifeAttack0426.jpg
knife 0.3
knife 0.9
DefenseKnifeAttack0401.jpg
knife 0.8
DefenseKnifeAttack0428.jpg
knife 0.7
knife 0.9
DefenseKnifeAttack0366.jpg
knife 0.9
DefenseKnifeAttack0373.jpg
knife 0.8
DefenseKnifeAttack0376.jpg
knife 0.9
DefenseKnifeAttack0452.jpg
knife 0.5
knife 0.9
DefenseKnifeAttack0423.jpg
knife 0.9
DefenseKnifeAttack0393.jpg
knife 0.6
knife 0.9
DefenseKnifeAttack0399.jpg
knife 0.9
DefenseKnifeAttack0456.jpg
knife 0.9
DefenseKnifeAttack0419.jpg
knife 0.8
DefenseKnifeAttack0377.jpg
knife 0.8
DefenseKnifeAttack0443.jpg
knife 0.9
DefenseKnifeAttack0475.jpg

Validation batch of trained model (knifes)

Implementation

1st Training: YOLOv5m



Validation batch of trained model (guns)

Implementation

Tools

Prog. Language

- Python

Library

- PyTorch & LabelImg

Database

- MySQL

Code Editor

- PyCharm & Visual Studio Code

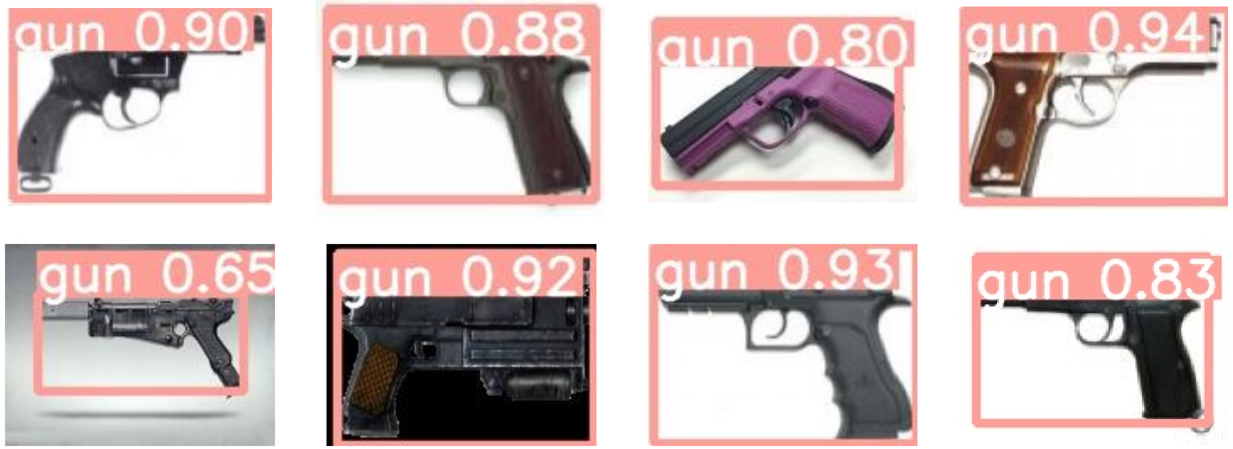
Code Hosting

- Microsoft GitHub

Implementation

Testing & Inferencing

Model Inferencing



Implementation

Trained Model

Implementation

WebstieDemo

Thanks! For Listening