



COMP90082 – Software Project RoboMaster AI Challenge

Object-detection Algorithm
Training/Evaluation Tool

Developer/Presenter: RM-Team-Koala

PRESENTATION OUTLINE

01

02

03

04

05

06

INTRODUCTION

What you need to know about this project

FINAL SCOPE

What we accomplished in this project

DEMO SESSION

To see is to believe

PRODUCT QUALITY

How we tested our software

HANDOVER

Our final delivery package

FEEDBACK TIME

MEET THE DEV TEAM



**SEJIN
KIM**

Position:
Team Lead

Duty:
UI Development
Support

Confluence
Maintenance



**JIA
YIN**

Position:
UX designer

Duty:
UI Development Lead
Diagram and Model



**CHE-HAO
CHANG**

Position:
Product Owner

Duty:
Algorithm
Development
Lead

External
Communication



**ISAAC
PEDROZA**

Position:
Scrum Master

Duty:
UI Development
Support

Git Management



**AKHTAR
KURNIAWAN**

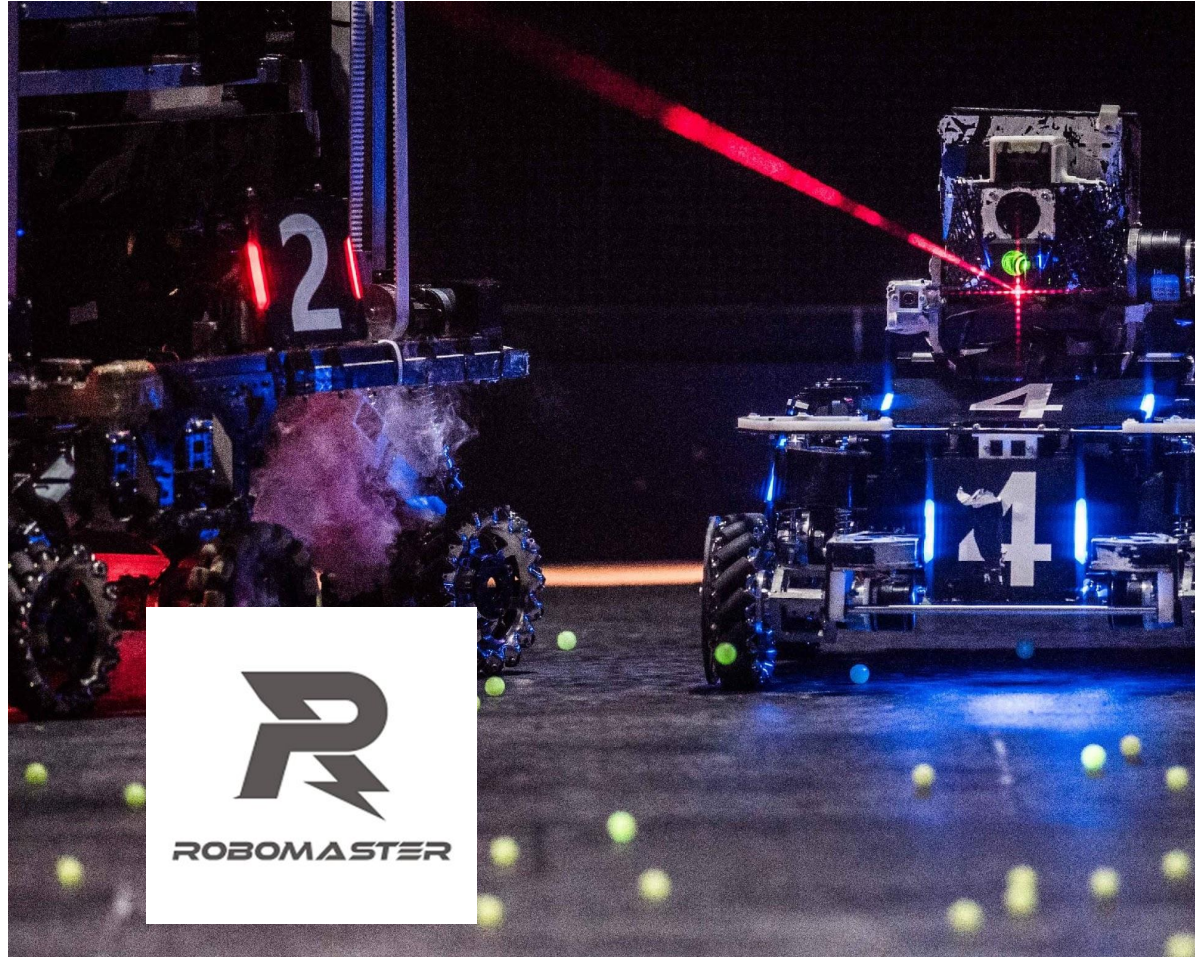
Position:
Quality Assurance

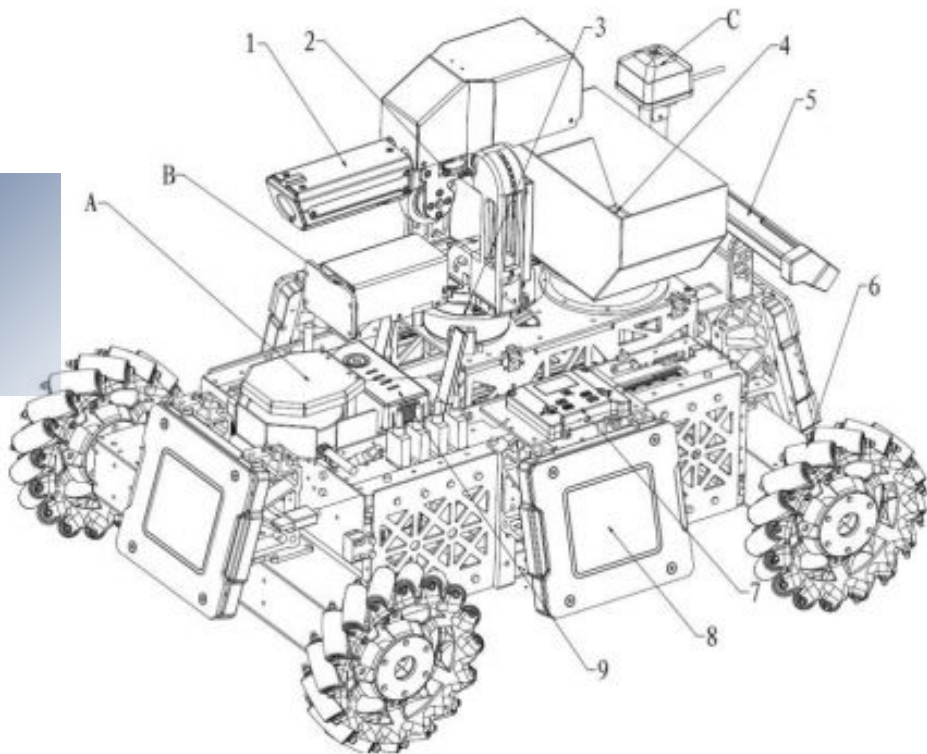
Duty:
Algorithm
Development Support

Software Tester

What is RoboMaster AI Challenge?

- 2 on 2 robotic battle within an 8m x 4.5m arena.
- Rover-like omni-directional robot with sensors and a gun turret.
- Projectiles that land on armour pads will deduct HP (Hit Points). Once HP is down to zero, the robot is out.
- Robots have different armour pads. While being attacked, different HP deduction will be applied.





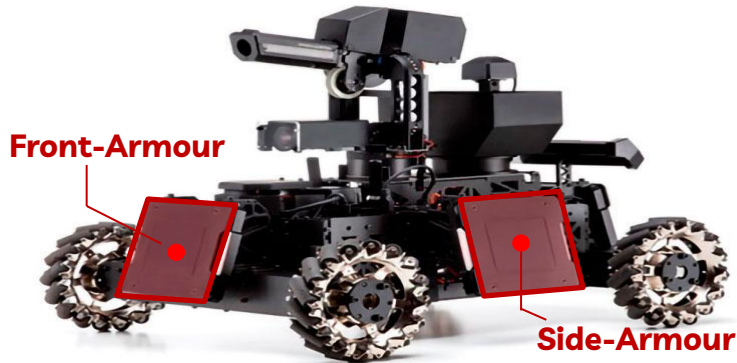
ABOUT THE PROJECT

1. Developing a computer vision algorithm that can help the battle robots in the competition.
2. Developing a visualisation tool that can help our client evaluate the performance of the algorithm.

PROJECT SPECIFICATION

Task 1 – Armour Localisation

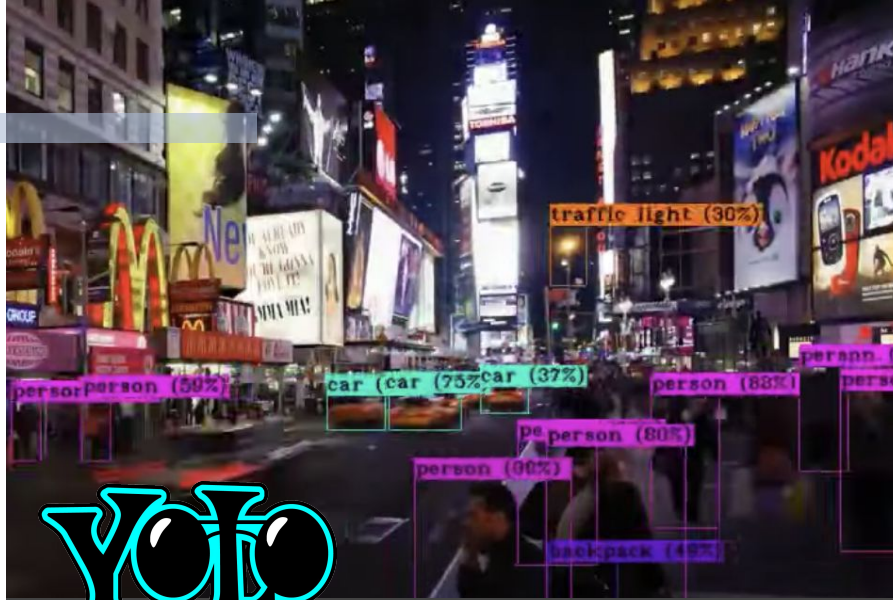
Position the armour pads in the image and return the coordination of it, so the robot can arrange attacks.



Task 2 – Armour Identification

Identify the the armour type for best attacking strategy .

CHOICE OF ALGORITHM



Real-time object-detection YOLOv4-tiny:

- Came out May 2020
- SOTA performance on speed and accuracy
- Popular – Many available frameworks
- Smaller size – Faster inference

Tech Stack

- Darknet: Neural Network Framework
- CVAT: Image Labelling Tool
- Python:
 - OpenCV: Algorithm
 - Tkinter: GUI

DELIVERED

- User Story 1 & 2:
 - Core Algorithm Implementation
- User Story 3:
 - Visualisation tool GUI
- User Story 4:
 - Algorithms Integration
- User Story 7 & 8:
 - Advanced functions

NOT DELIVERED

- User Story 5 & 6:
 - Implement Second Algorithm

NOT COMMITTED

- User Story 1:
 - Image Preprocessing
 - Training Sample Augmentation

FINAL SCOPE

User Story	Importance	Story Points	Sprint	Status
1	Must	39	Sprint1	C/D
2	Must	30	Sprint1	C/D
3	Desirable	36	Sprint1/2	C/D
4	Desirable	16	Sprint2	C/D
5	Optional	24	Sprint2	C
6	Optional	24	Sprint2	C
7	Optional	30	Sprint2	C/D
8	Optional	30	Sprint2	C/D

ALGORITHM PERFORMANCE TEST



Localisation ☐
Identification ☐



Localisation ☐
Identification ☐



Localisation ☐
Identification ☐

Parameter	Acceptance Criteria	Achieved
Localisation Accuracy	$\geq 70\%$	90%
Identification Accuracy	$\geq 70\%$	85%
Processing time per image	≤ 100 ms	~5 ms (w/ Google Colab GPU)

DEMO TIME

Scenario 1

One Robot and
One Image.

Scenario 2

One Robot and
Multiple Images.

Scenario 3


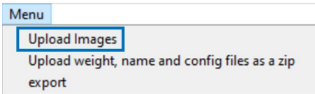
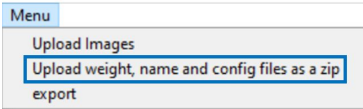
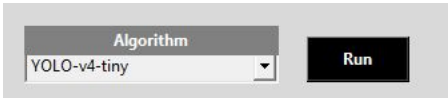
No Robot.

Scenario 4

Multiple Robots
and One Image.


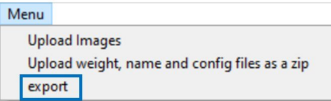
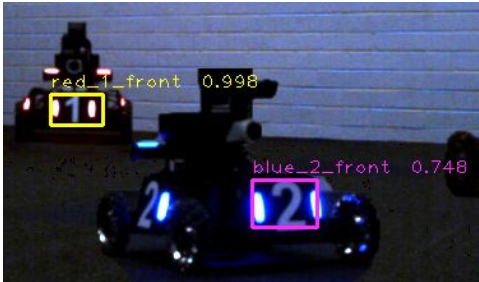
SOFTWARE QUALITY TEST CASE



Test Case	User Story	What is Tested	Expected Outcome	Image
TC001-2	1	Correct localisation	70% accuracy, 0.1 s processing time/image	 A photograph of a small robot with blue lights. A green bounding box is drawn around the robot, and the text 'blue_2_side 0.982' is overlaid in green.
TC003-4	2	Correct identification	70% accuracy, 0.1 s processing time/image	
TC005	3	Uploading image(s)	<ul style="list-style-type: none">Image is displayed on GUI frame.User can navigate using next/prev buttons	 A screenshot of a software menu. The 'Menu' tab is selected. Under it, 'Upload Images' is highlighted with a blue box. Below it, the text 'Upload weight, name and config files as a zip export' is visible.
TC006	3	Uploading weight, name, and config files	The zip file is only accepted if it contains all the required files.	 A screenshot of a software menu. The 'Menu' tab is selected. Under it, 'Upload weight, name and config files as a zip export' is highlighted with a blue box.
TC007	3	Run button	<ul style="list-style-type: none">Bounding box and label are drawn on each robot's nearest armour.Output board displays the prediction output.	 A screenshot of a software interface. It shows a dropdown menu labeled 'Algorithm' with 'YOLO-v4-tiny' selected. To the right of the dropdown is a black button with the word 'Run' in white.

SOFTWARE QUALITY TEST CASE (2)



Test Case	User Story	What is Tested	Expected Outcome	Image
TC008	3	Slider	Displayed images change accordingly as user drags the slider.	
TC009	3	Export button	Image annotations are saved as txt files.	
TC010	4	Consistency between armour localisation and identification	The located armour should be correctly labelled.	
TC011	7	Localisation on multiple robots	Bounding box is drawn on each robot's nearest armour.	
TC012	8	Identification on multiple robots	Label is shown on each robot's bounding box.	

HANDOVER PACKAGE




GitHub Repo Link:

https://github.com/cchia790411/rm_ai_challenge_2020s2_koala

THANK YOU FOR LISTENING

Feedbacks & Questions
are welcomed!

A gold trophy with a stylized 'RM' logo on top, sitting on a white pedestal. The word 'ROBOMASTERS' is printed on the front of the pedestal. The background is dark with blue and red light streaks.

ROBOMASTERS