

# Testing Document

Version	Description	Author	Date (DD/MM/YYYY)
0.0	First draft of the Testing Document	AKHTAR KURNIAWAN	24/09/20
0.1	Separate test cases for User Story 2	AKHTAR KURNIAWAN	27/09/20
1.0	Test cases for Sprint 1	AKHTAR KURNIAWAN	02/10/20

## 1. Introduction

### 1.1 Proposal

The purpose of this document is to define and present the test cases for project Robomaster AI Challenge by team RM-Koala, covering the test cases for the system use cases.

### 1.2 Target Users

This document is mainly designed for those responsible for executing the test cases in this project, namely the team members, client, and COMP90082 teaching team.

### 1.3 Conventions, terms and abbreviations

This section explains the concept of some important terms that will be used throughout this document. These terms are described in the following table, presented in alphabetical order.

Term	Description
Accurate armour identification	The program correctly identifies the type of nearest armour.
Accurate location prediction	Bounding box covers both lights of the nearest armour. It should be small enough so that it does not include the light of another armour.

## 2. Covered Requirements

This section lists the system requirements covered in the test cases.

User Story Id	User Story
1	As a participant in the competition, I want my robot to locate the position of the opponent robot's armour in the pictures so that my robot is informed about where to shoot.
2	As a participant in the competition, I want my robot to identify what is the type of the armour pad the enemy is showing so that my robot is informed about how many points it can score if it successfully shoots to the identified armour.

## 3. Test Cases

### 3.1 User Story 1

#### 3.1.1 TC001: Armour Correctly Located

Test Type: Functional	Execution Type: Manual
Objective: Verify if armour is correctly located.	

**Setup:**

- The tester needs to have a *google* account.
- Prepare 10-20 test images of robots.

**Procedure:**

1. Sign in to google colab <https://colab.research.google.com/>
2. Upload the relevant notebook (red or blue) to google colab
3. Run all the cells from top
4. When reaching *test your own image* section, upload your image
5. Copy the given code into a new block, replacing the image file name. Run it.
6. Manually check the result accuracy.

**Notes:**

[1] Both team numbers (1 or 2) should be represented in the test set.

[1] Both armour light colours (red and blue) should be represented in the test set.

[2] All 3 robot poses should be represented in the test set.

**Expected outcome:**

- For each image, a bounding box that surrounds the nearest armour from the camera will be shown, as well as a label for colours.
- The processing time for each image will also be shown.
- Returns an accurate location prediction for at least 70% of images.
- Returns an accurate armour colour identification for at least 70% of images.

**Time constraint:**

Displayed processing time for each image should not exceed 0.1 s.

### 3.1.2 TC002: No Armour Bounding Box is Drawn

<b>Test Type:</b>	<b>Execution Type:</b>
Functional	Manual
<b>Objective:</b>	
Verify that program does not draw a bounding box when no target robot is visible in the test image.	
<b>Setup:</b>	
<ul style="list-style-type: none"> <li>• The tester needs to have a google account.</li> <li>• Tests are done separately for red and blue robots.</li> <li>• Prepare 10-20 images test images that do not show target robots or armours.</li> </ul>	
<b>Procedure:</b>	
Same as TC001	
<b>Notes:</b>	
[1] Preferably, test images show the environment similar to the Robomaster arena to simulate the real match condition	
<b>Expected outcome:</b>	
<ul style="list-style-type: none"> <li>• No bounding box is drawn</li> </ul>	
<b>Time constraint:</b>	
Displayed processing time for each image should not exceed 0.1 s.	

## 3.2 User Story 2

### 3.2.1 TC003: Robot Pose Correctly Identified

<b>Test Type:</b>	<b>Execution Type:</b>
Functional	Manual
<b>Objective:</b> Verify if the pose is correctly identified.	
<b>Setup:</b> <ul style="list-style-type: none"> <li>The tester needs to have a <i>google</i> account.</li> <li>One test covers both robot colours.</li> </ul>	
<b>Procedure:</b> Same as TC001, but skip step 2.	
<b>Notes:</b> [1] Both armour light colours (red and blue) should be represented in the test set. [2] Both team numbers (1 or 2) should be represented in the test set. [3] All 3 robot poses should be represented in the test set.	
<b>Expected outcome:</b> <ul style="list-style-type: none"> <li>For each image, a bounding box that surrounds the whole robot will be shown, as well as a label (front, side, or back).</li> <li>The processing time for each image will also be shown.</li> <li>Returns an accurate pose identification for at least 70% of images.</li> </ul>	
<b>Time constraint:</b> Displayed processing time for each image should not exceed 0.1 s.	

### 3.1.2 TC004: No Robot Bounding Box is Drawn

<b>Test Type:</b>	<b>Execution Type:</b>
Functional	Manual
<b>Objective:</b> Verify that program does not draw a bounding box when no robot is visible.	
<b>Setup:</b> <ul style="list-style-type: none"> <li>The tester needs to have a google account.</li> <li>Prepare 10-20 images test images that do not show any robots.</li> </ul>	
<b>Procedure:</b> Same as TC003	
<b>Notes:</b> [1] Preferably, test images show the environment similar to the Robomaster arena to simulate the real match condition	
<b>Expected outcome:</b> <ul style="list-style-type: none"> <li>No robot bounding box is drawn</li> </ul>	
<b>Time constraint:</b> Displayed processing time for each image should not exceed 0.1 s.	