COCOROBO

CocoPi

AI for STEM Competition

Asia-Pacific STEAM_AI Technology Innovation Challenge

Custom Model Recognition

CocoRobo

目錄

CONTENTS

Chapter 1

Custom Model Training

Chapter 2

Achieve Custom Model Recognition

Chapter 3

Recognize Traffic Signs

CHAPTER

P

O

W

E

ONE.

Custom Model Training

CHAPTER

Custom Model Training

How do machines distinguish between "cats" and "dogs"?
How do you differentiate them?

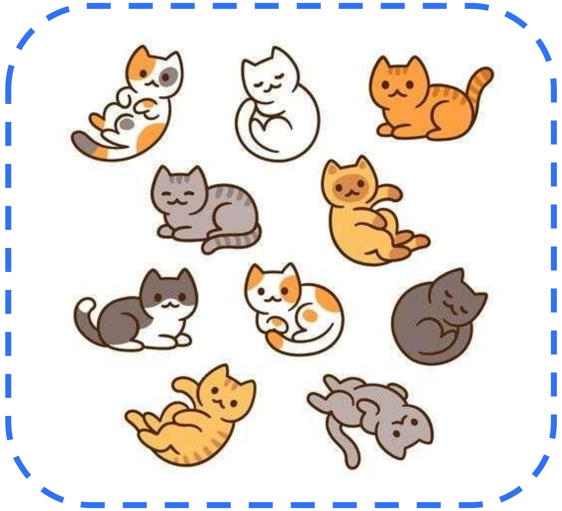
→ **cat**

→ **dog**

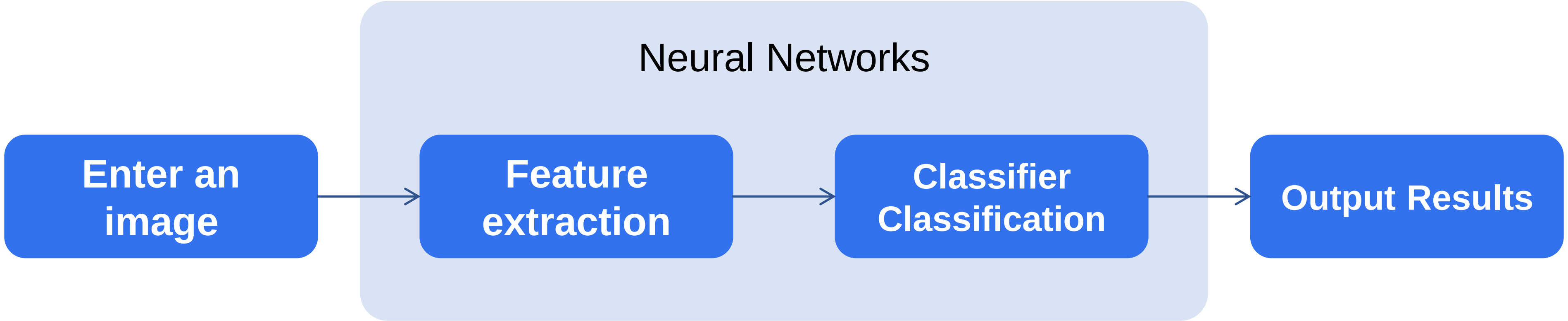
Custom Model Training

- Size and Volume
- Fur and Skin
- Facial Features
- Color and Patterns
- ……

features



Custom Model Training



Neural Networks

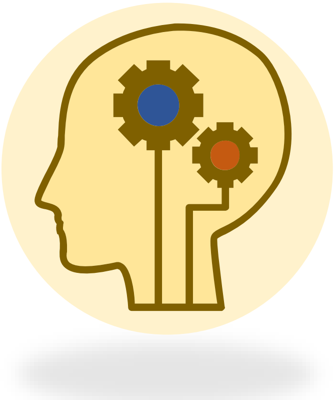
Enter an image

Feature extraction

Classifier Classification

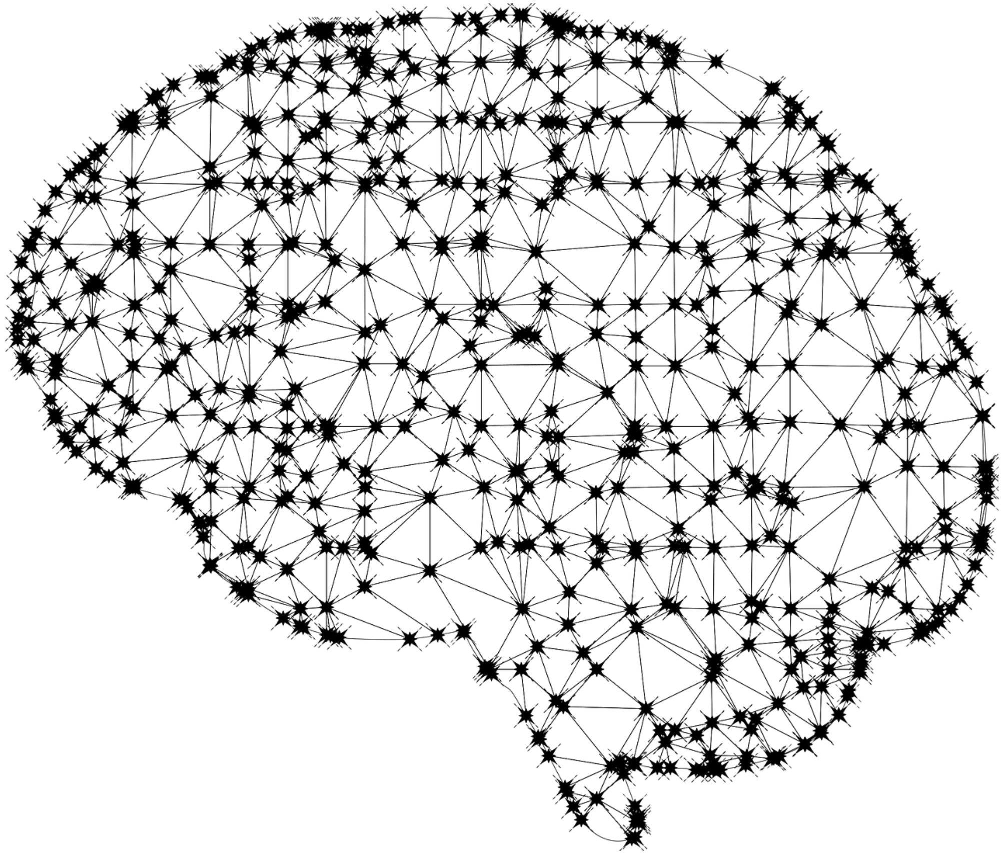
Output Results

model

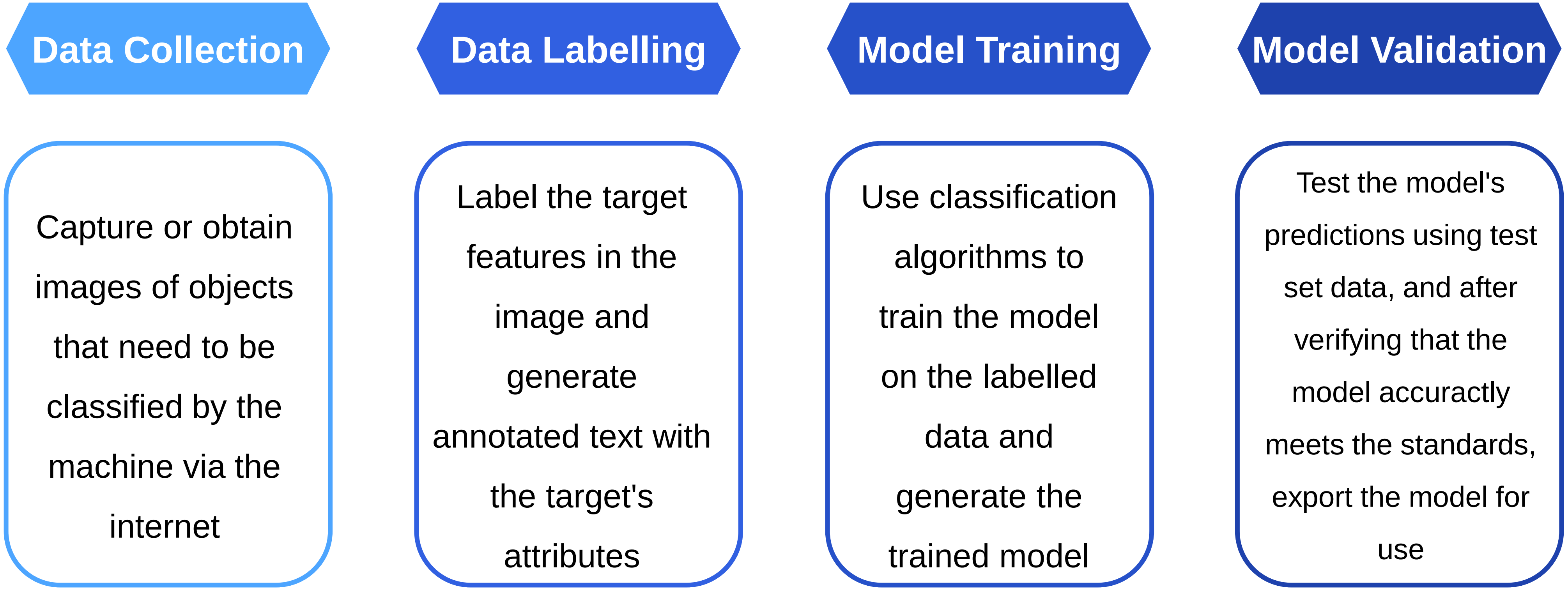


Custom Model Training

What steps are involved in custom model training?



Custom Model Training



Data Collection

Capture or obtain images of objects that need to be classified by the machine via the internet

Data Labelling

Label the target features in the image and generate annotated text with the target's attributes

Model Training

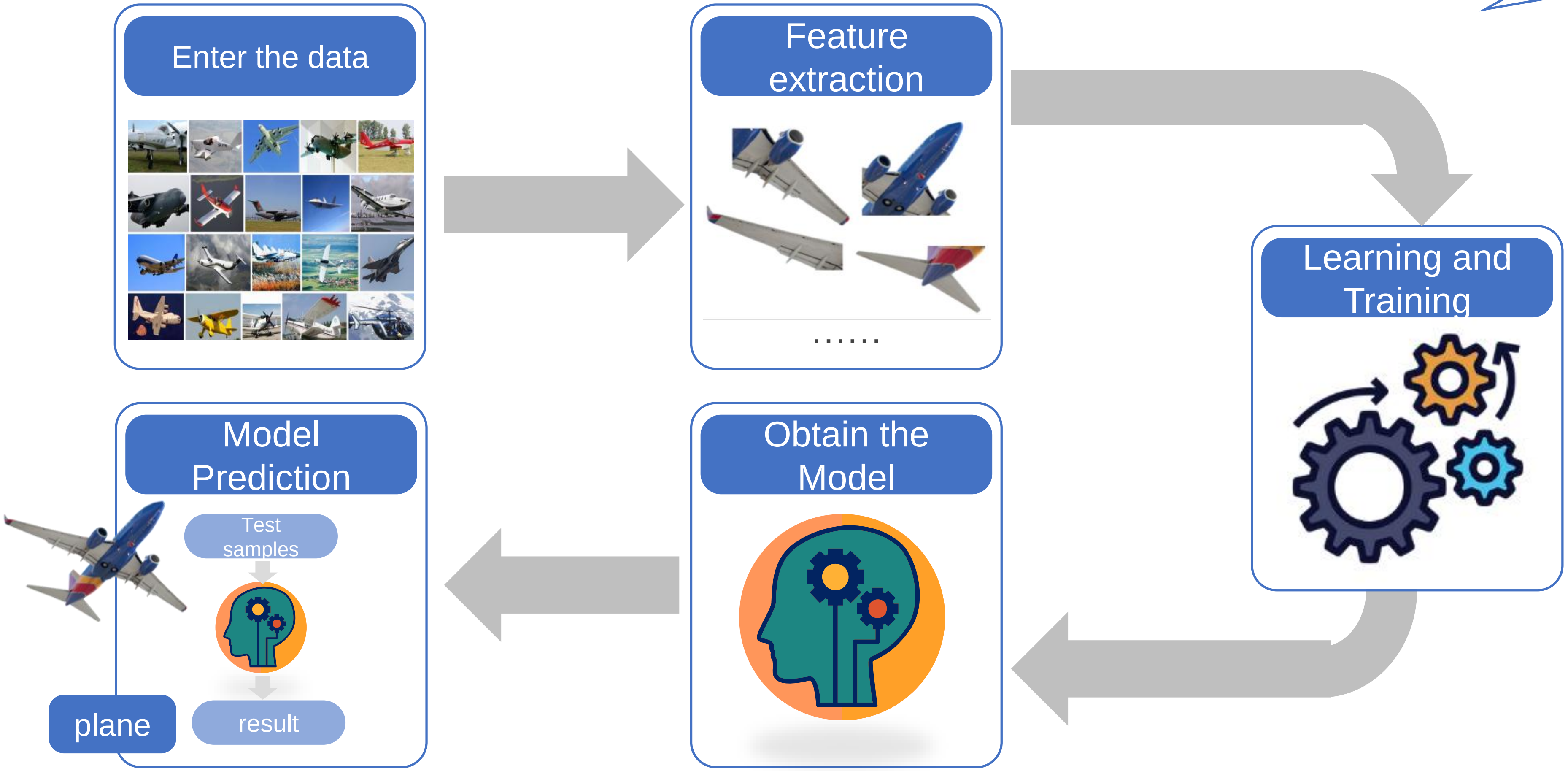
Use classification algorithms to train the model on the labelled data and generate the trained model

Model Validation

Test the model's predictions using test set data, and after verifying that the model accuractly meets the standards, export the model for use

Custom Model Training

Take, for example, the identification of aircraft



Enter the data

Feature extraction

……

Learning and Training

Obtain the Model

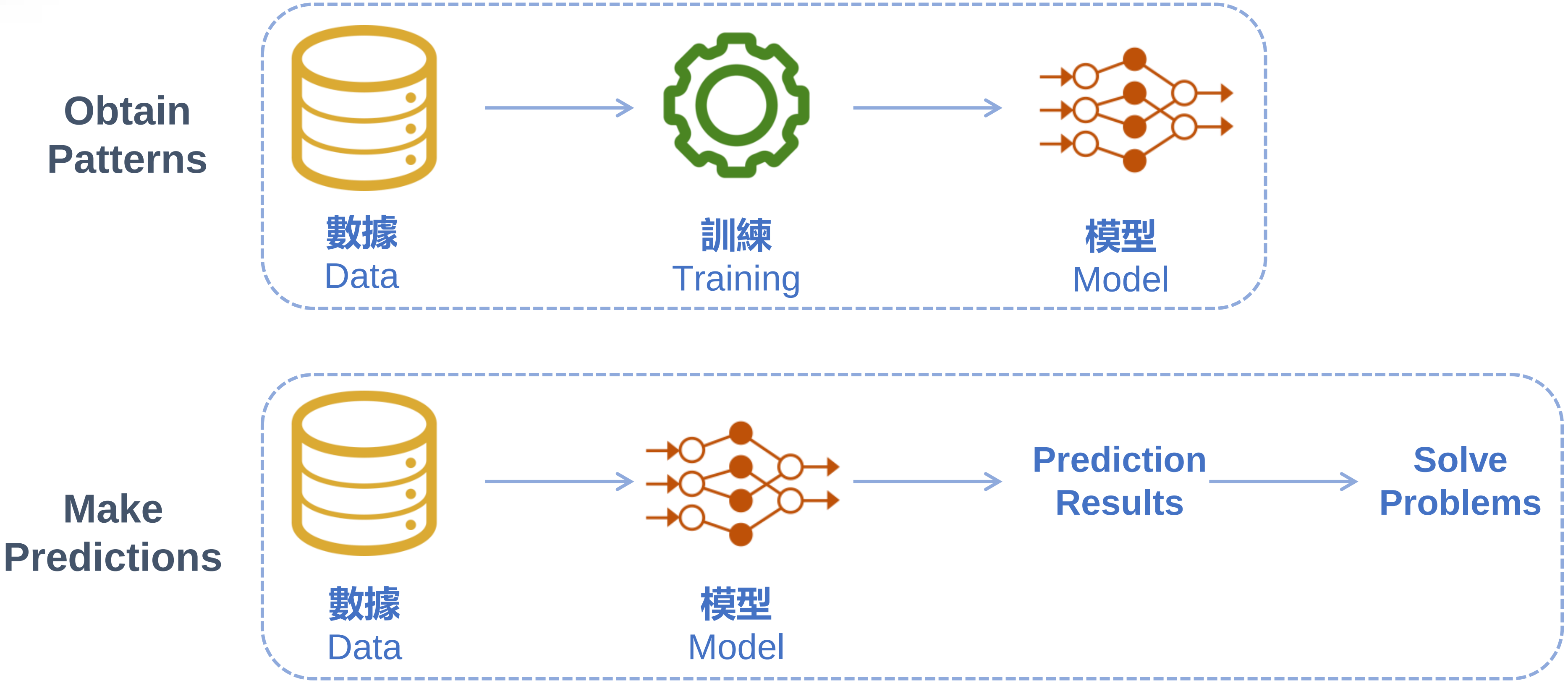
Model Prediction

Test samples

result

plane

Custom Model Training



CHAPTER

P

O

W

E

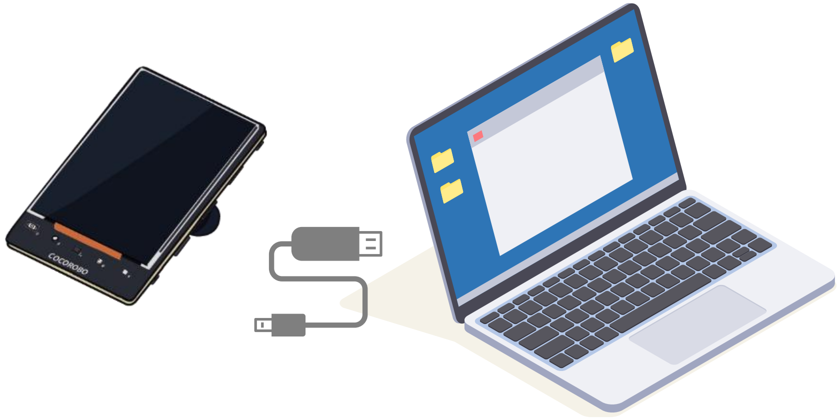
TWO.

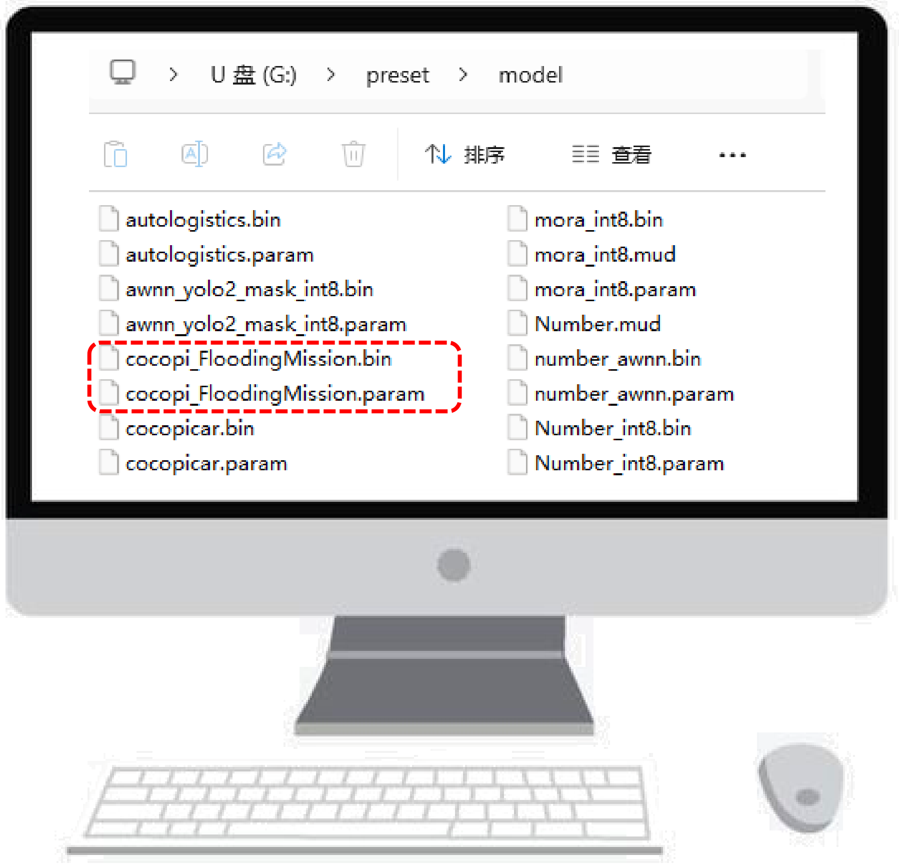
Achieve Custom Model Recognition

CHAPTER

Achieve Custom Model Recognition

Model Import





U 盘 (G:) › preset › model

autologistics.bin
autologistics.param
awnn_yolo2_mask_int8.bin
awnn_yolo2_mask_int8.param
cocopi_FloodingMission.bin
cocopi_FloodingMission.param
cocopicar.bin
cocopicar.param
mora_int8.bin
mora_int8.mud
mora_int8.param
Number.mud
number_awnn.bin
number_awnn.param
Number_int8.bin
Number_int8.param

Connect the cocopi to the computer， move the following files

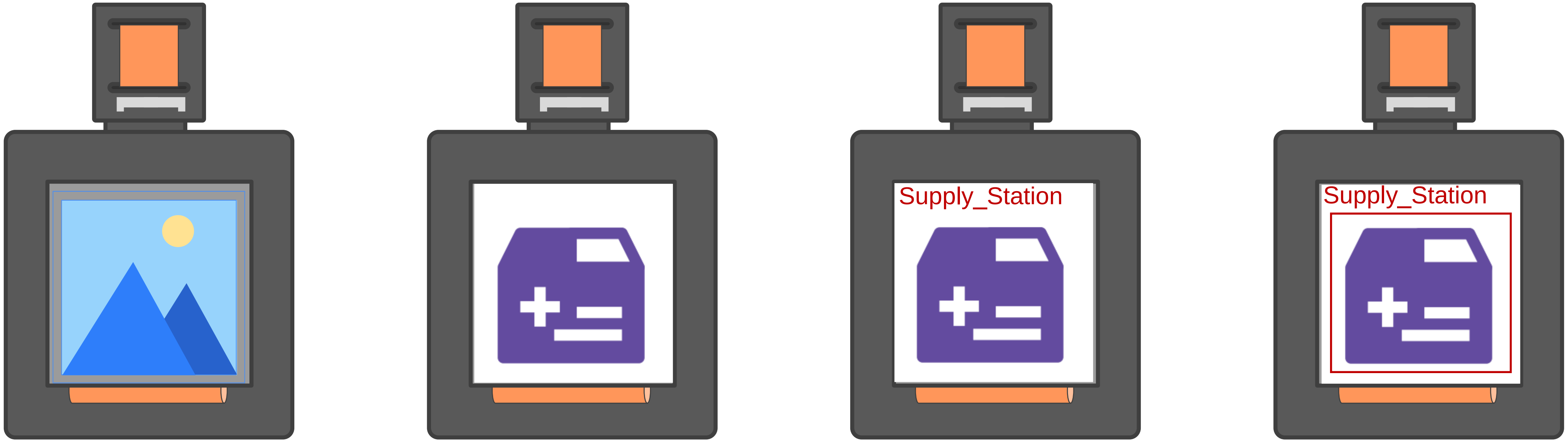
cocopi_FloodingMission.bin

cocopi_FloodingMission.param

to **/root/preset/model**

Achieve Custom Model Recognition

Implementation process



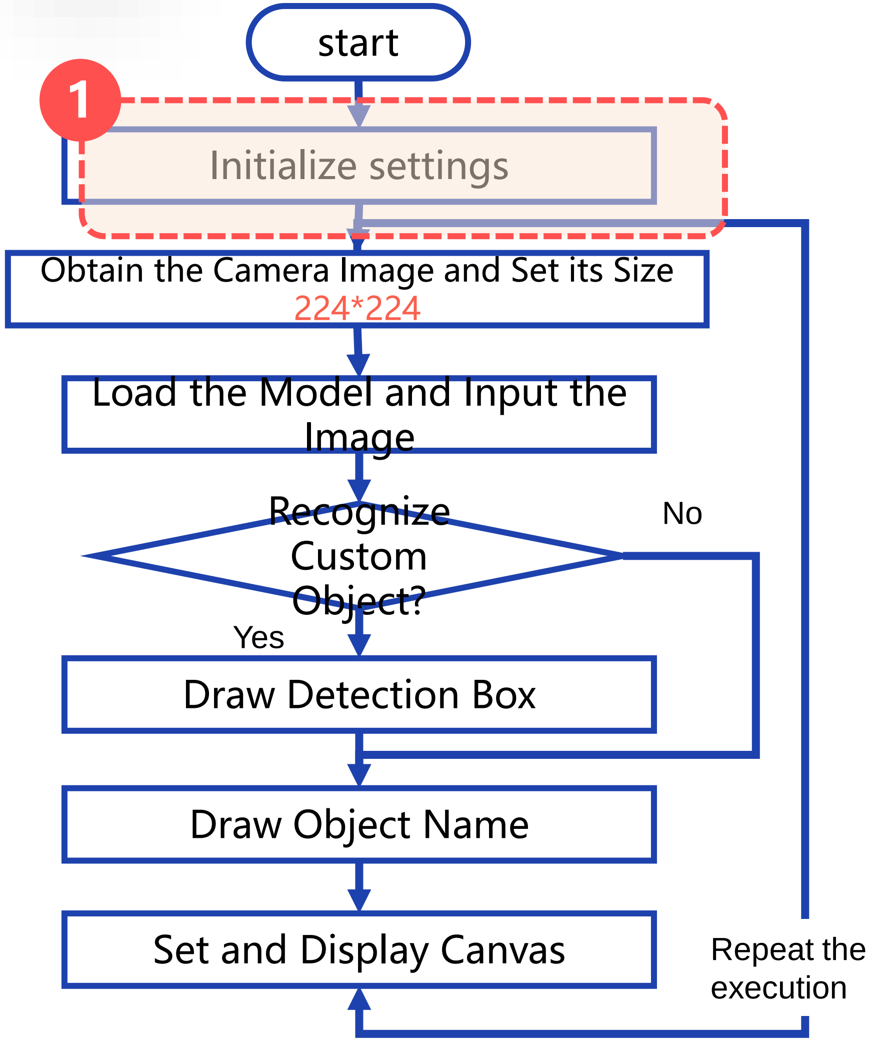
1 Obtain Camera Image

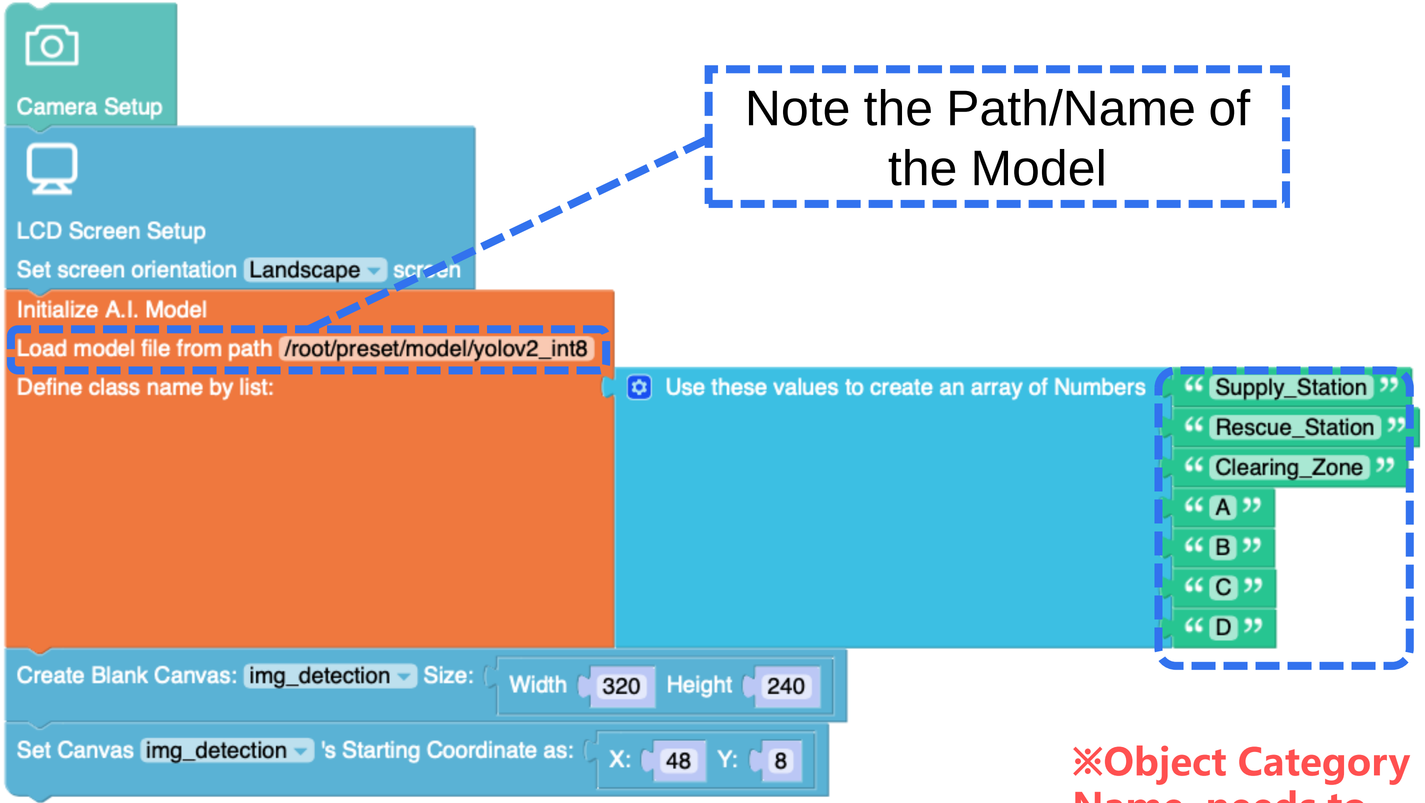
2 Custom Model Recognition

3 Display the Name of the Recognized Object

4 Draw a Box Around the Recognized Object

Achieve Custom Model Recognition




Camera Setup

LCD Screen Setup
Set screen orientation Landscape screen

Initialize A.I. Model
Load model file from path /root/preset/model/yolov2_int8
Define class name by list: Use these values to create an array of Numbers "Supply_Station" "Rescue_Station" "Clearing_Zone" "A" "B" "C" "D"

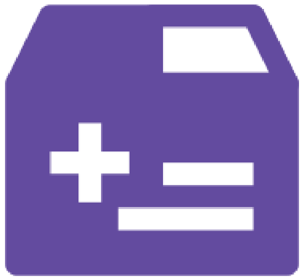
Create Blank Canvas: img_detection Size: Width 320 Height 240

Set Canvas img_detection 's Starting Coordinate as: X: 48 Y: 8

Note the Path/Name of the Model

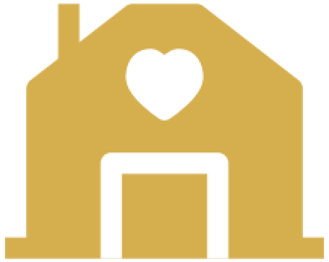
※Object Category Name, needs to correspond with the model

Achieve Custom Model Recognition



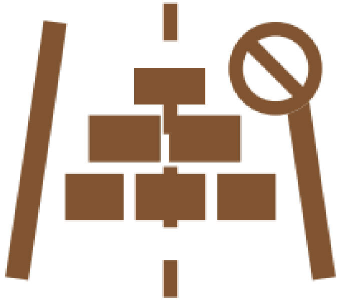
Supply_Station

補給站



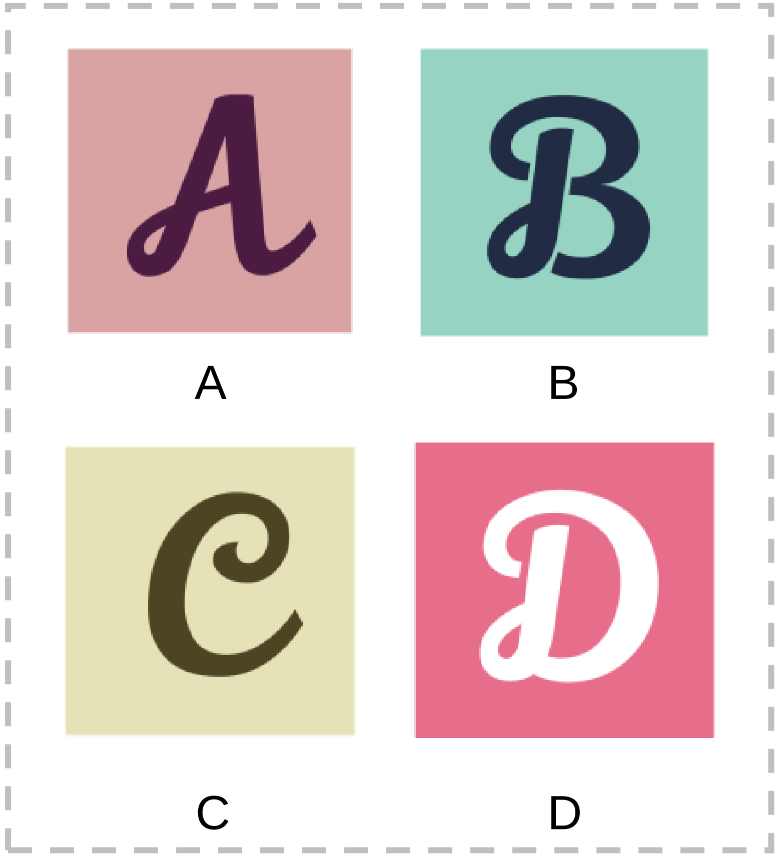
Rescue_Station

救助站



Clearing_Zone

清理區



A

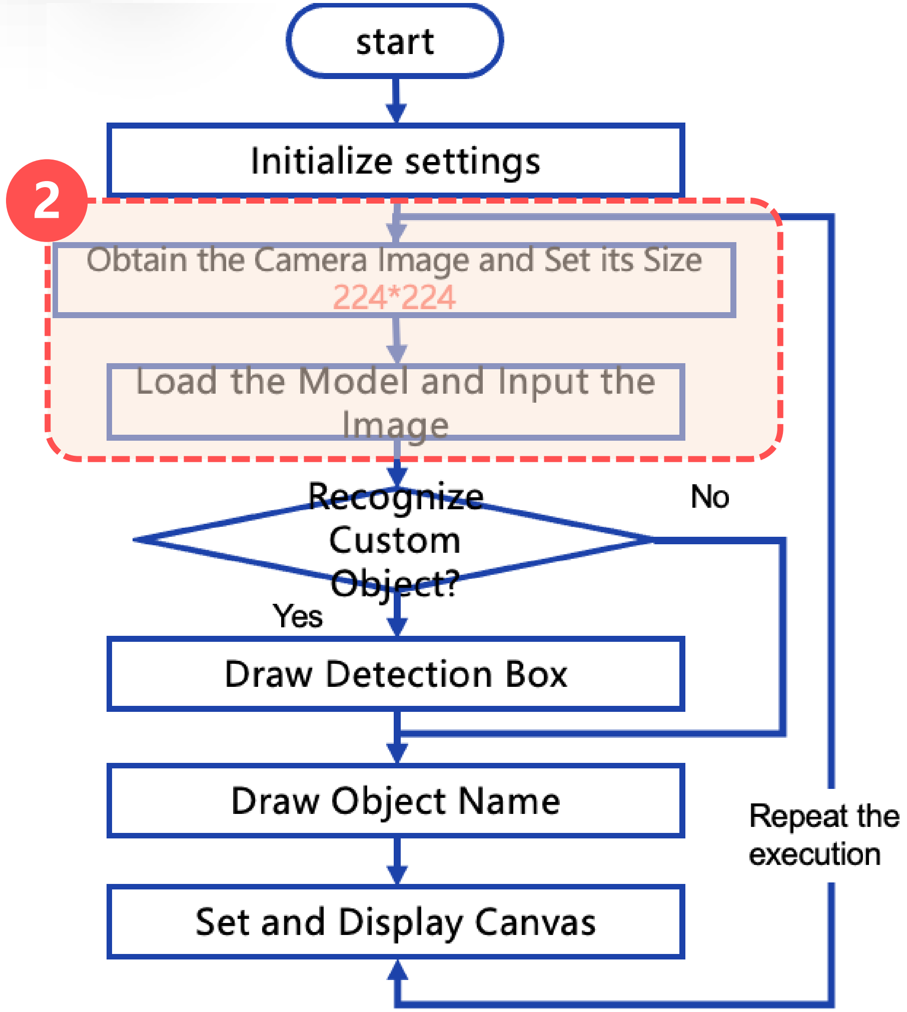
B

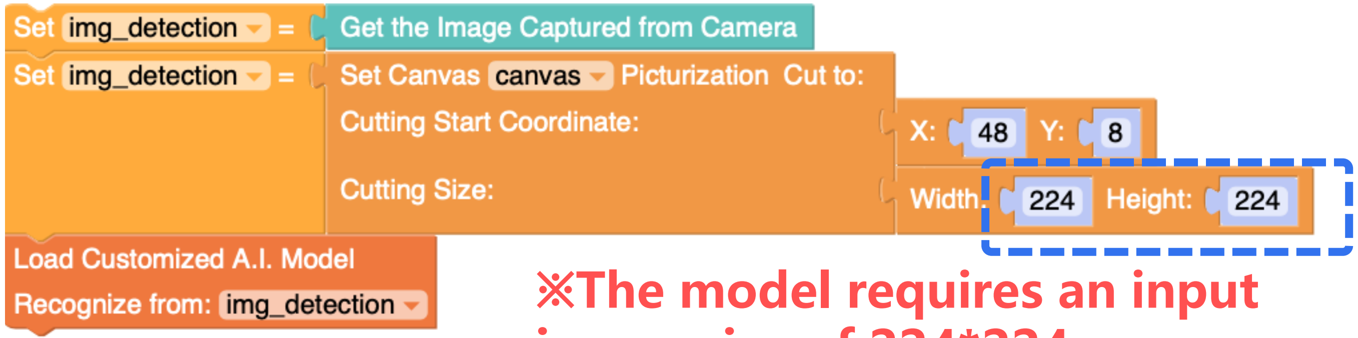
C

D

Route Card

Achieve Custom Model Recognition

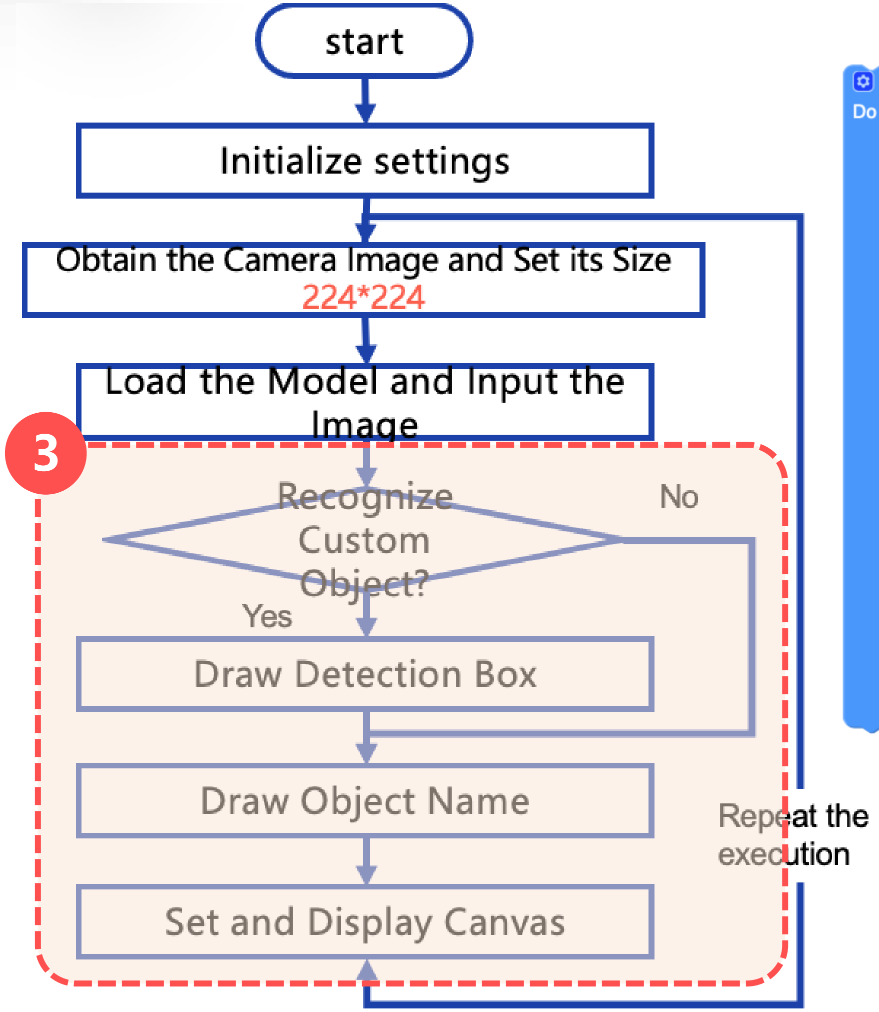




```
Set img_detection = Get the Image Captured from Camera
Set img_detection = Set Canvas canvas Picturization Cut to:
    Cutting Start Coordinate: X: 48 Y: 8
    Cutting Size: Width: 224 Height: 224
Load Customized A.I. Model
Recognize from: img_detection
```

※The model requires an input image size of 224*224

Achieve Custom Model Recognition



start

Initialize settings

Obtain the Camera Image and Set its Size 224*224

Load the Model and Input the Image

3

Recognize Custom Object?

No

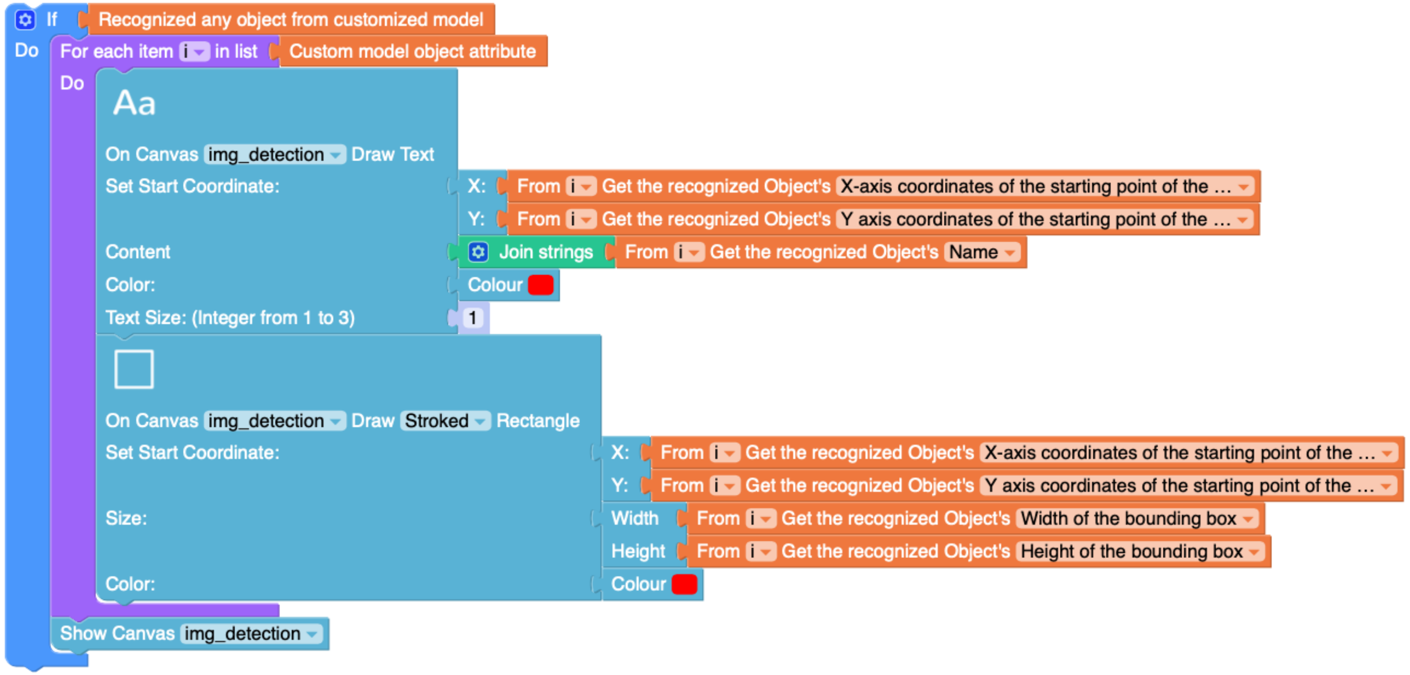
Yes

Draw Detection Box

Draw Object Name

Set and Display Canvas

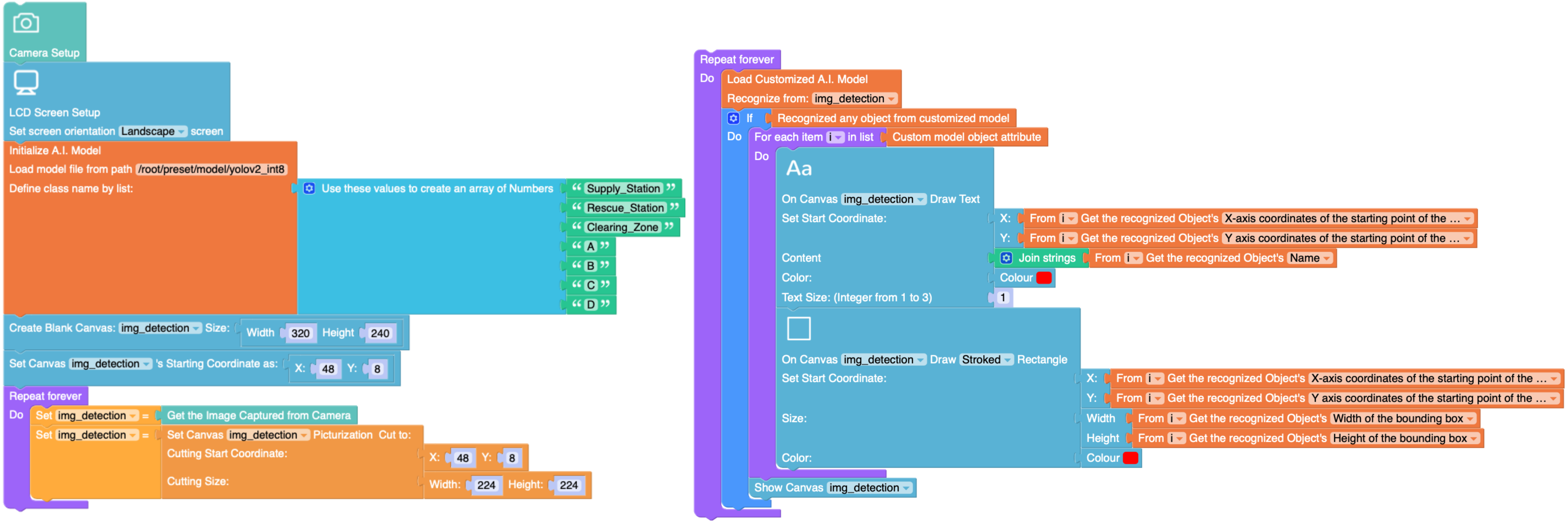
Repeat the execution



```
If  Recognized any object from customized model
Do  For each item i in list  Custom model object attribute
    Do  Aa
        On Canvas img_detection Draw Text
        Set Start Coordinate:  X: From i Get the recognized Object's X-axis coordinates of the starting point of the ...
                               Y: From i Get the recognized Object's Y axis coordinates of the starting point of the ...
        Content                Join strings  From i Get the recognized Object's Name
        Color:                 Colour 🟥
        Text Size: (Integer from 1 to 3)  1
        □
        On Canvas img_detection Draw Stroked Rectangle
        Set Start Coordinate:  X: From i Get the recognized Object's X-axis coordinates of the starting point of the ...
                               Y: From i Get the recognized Object's Y axis coordinates of the starting point of the ...
        Size:                  Width  From i Get the recognized Object's Width of the bounding box
                               Height From i Get the recognized Object's Height of the bounding box
        Color:                 Colour 🟥
    Show Canvas img_detection
```

Achieve Custom Model Recognition

Reference Progam



```
Camera Setup
LCD Screen Setup
Set screen orientation [Landscape] screen
Initialize A.I. Model
Load model file from path [/root/preset/model/yolov2_int8]
Define class name by list:  Use these values to create an array of Numbers
    " Supply_Station "
    " Rescue_Station "
    " Clearing_Zone "
    " A "
    " B "
    " C "
    " D "
Create Blank Canvas: [img_detection] Size: Width [320] Height [240]
Set Canvas [img_detection] 's Starting Coordinate as: X: [48] Y: [8]
Repeat forever
Do  Set [img_detection] = Get the Image Captured from Camera
    Set [img_detection] = Set Canvas [img_detection] Picturization  Cut to:
                          Cutting Start Coordinate:  X: [48] Y: [8]
                          Cutting Size:  Width: [224] Height: [224]
```

```
Repeat forever
Do  Load Customized A.I. Model
    Recognize from: [img_detection]
    If  Recognized any object from customized model
    Do  For each item [i] in list  Custom model object attribute
        Do  Aa
            On Canvas [img_detection] Draw Text
            Set Start Coordinate:  X: From [i] Get the recognized Object's [X-axis coordinates of the starting point of the ...]
                                   Y: From [i] Get the recognized Object's [Y axis coordinates of the starting point of the ...]
            Content  Join strings  From [i] Get the recognized Object's [Name]
            Color:  Colour 🟥
            Text Size: (Integer from 1 to 3)  [1]
            □
            On Canvas [img_detection] Draw [Stroked] Rectangle
            Set Start Coordinate:  X: From [i] Get the recognized Object's [X-axis coordinates of the starting point of the ...]
                                   Y: From [i] Get the recognized Object's [Y axis coordinates of the starting point of the ...]
            Size:  Width  From [i] Get the recognized Object's [Width of the bounding box]
                   Height  From [i] Get the recognized Object's [Height of the bounding box]
            Color:  Colour 🟥
        Show Canvas [img_detection]
```

COCOROBO

CHAPTER

P O W E

THREE.

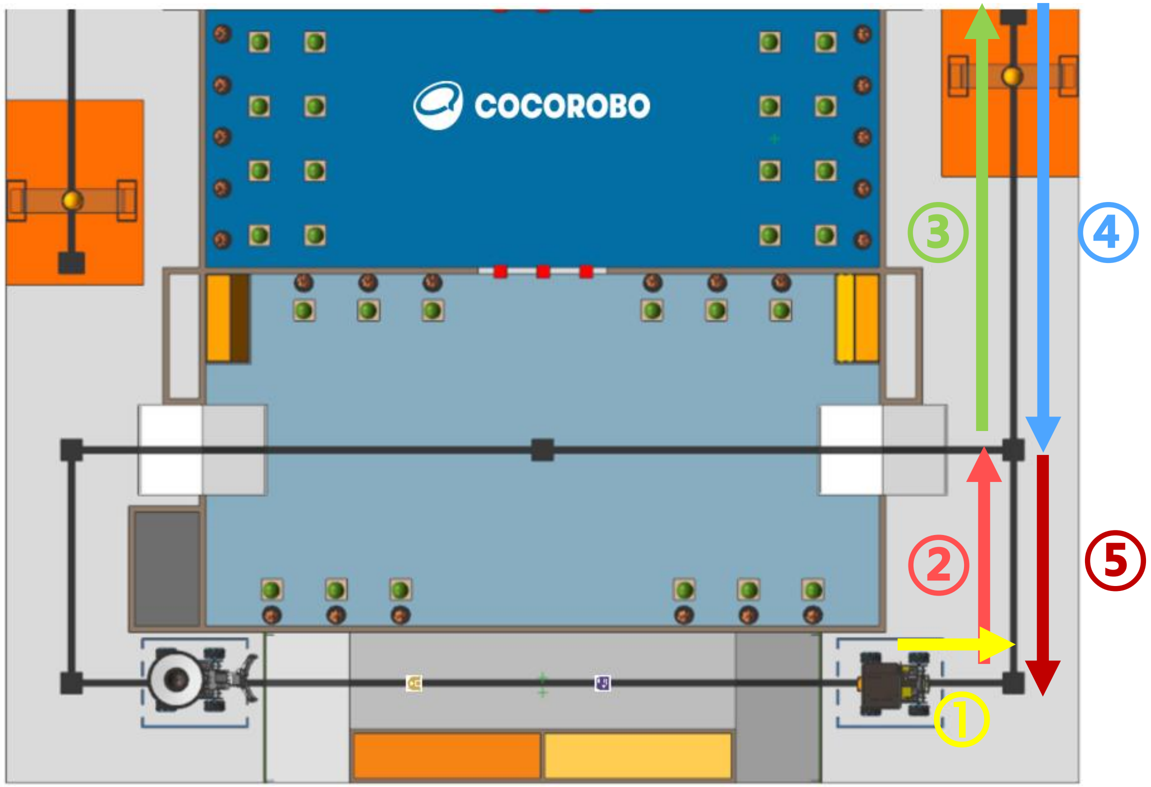
Recognize Traffic Signs

CHAPTER

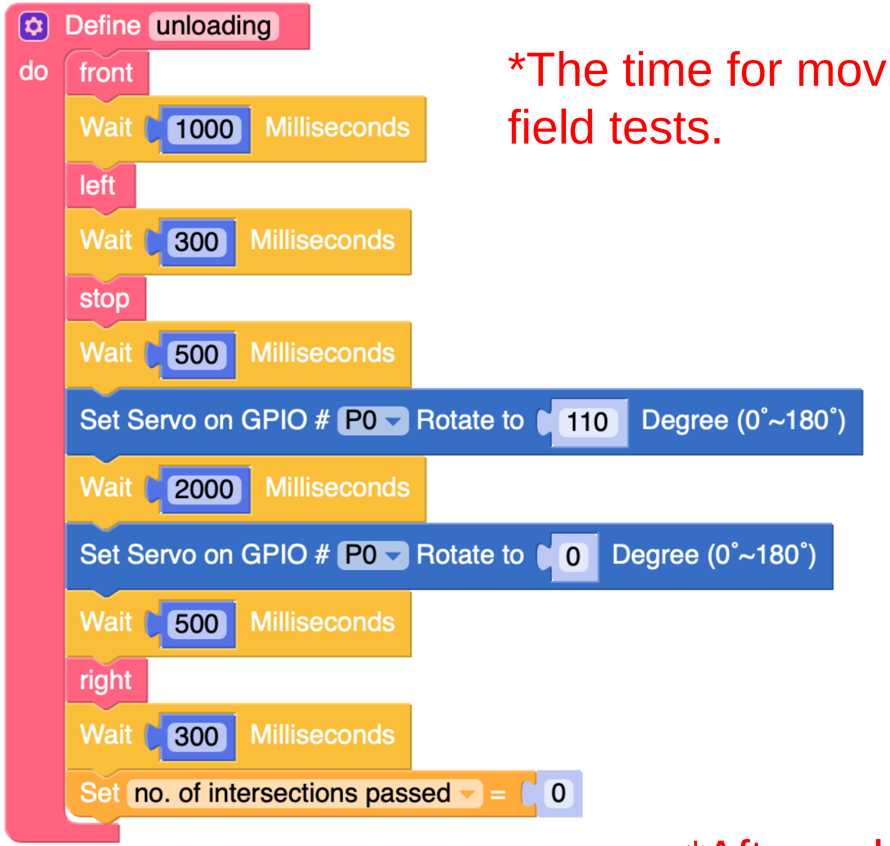
Recognize Traffic Signs

Task 1：

Let the car automatically follow the "material" route. When the "Supply Station" sign is recognized, unload the materials at the appropriate location.



Recognize Traffic Signs



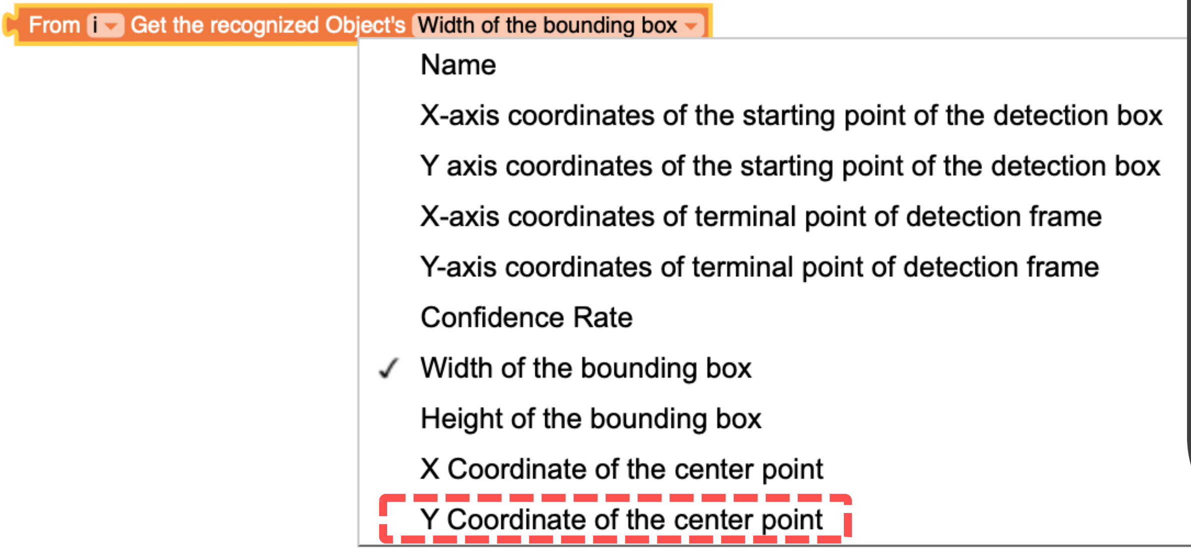
```
Define unloading
do  front
    Wait 1000 Milliseconds
    left
    Wait 300 Milliseconds
    stop
    Wait 500 Milliseconds
    Set Servo on GPIO # P0 Rotate to 110 Degree (0˚~180˚)
    Wait 2000 Milliseconds
    Set Servo on GPIO # P0 Rotate to 0 Degree (0˚~180˚)
    Wait 500 Milliseconds
    right
    Wait 300 Milliseconds
    Set no. of intersections passed = 0
```

*The time for moving forward needs to be adjusted and optimized based on real field tests.

*After unloading, reset the "number of intersections passed," effectively returning the status to its initial state.

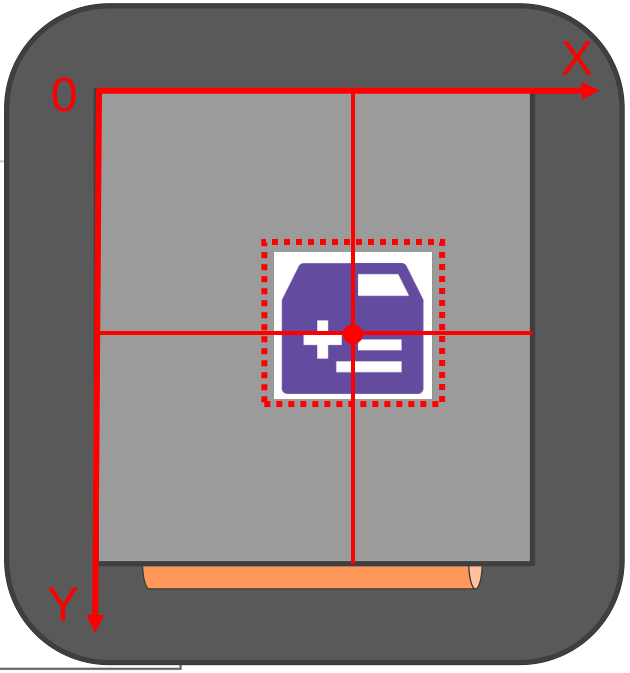
Recognize Traffic Signs

"The car recognizes the sign" is a fixed result. So, when should the car stop to unload the materials?



From i Get the recognized Object's Width of the bounding box

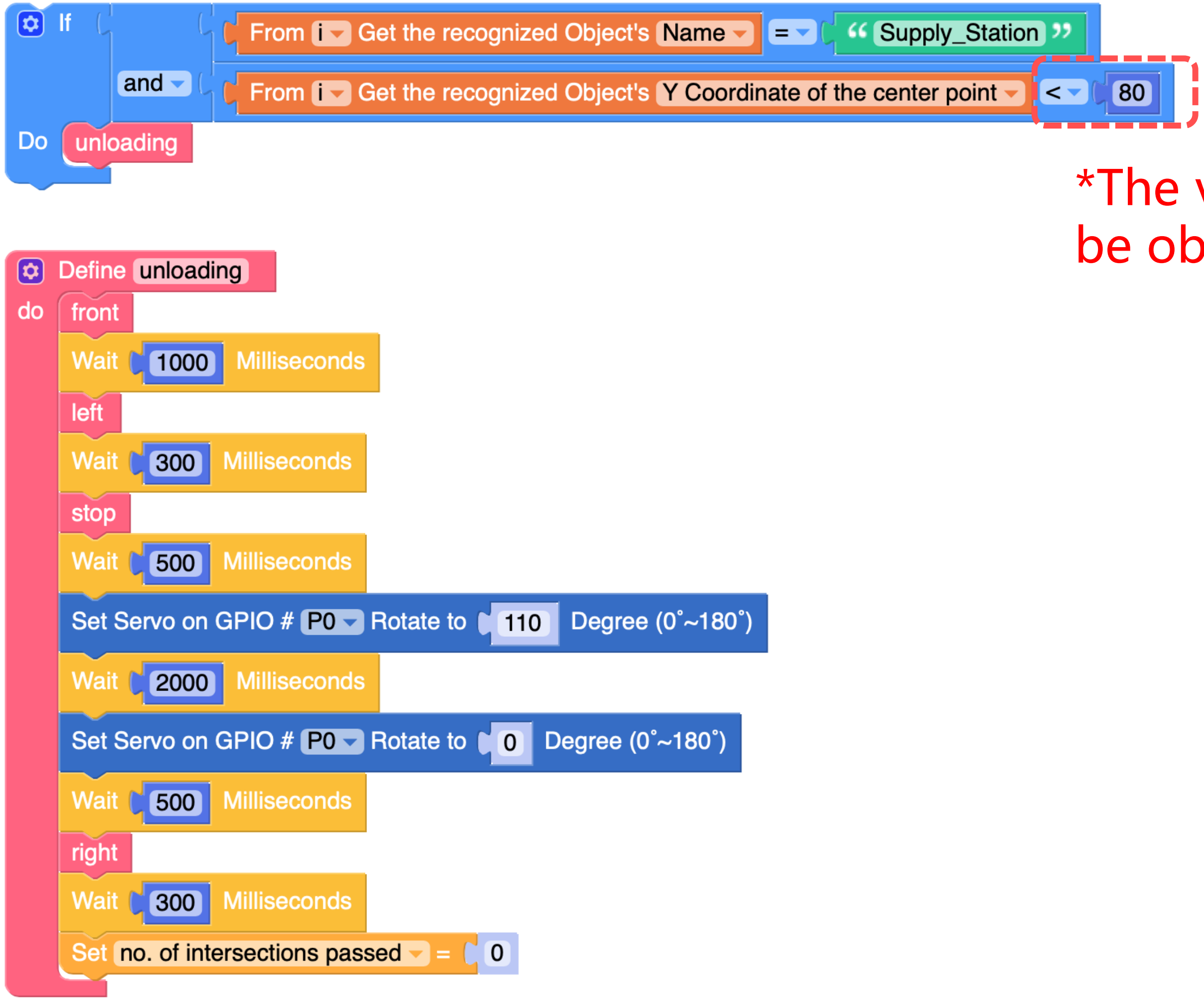
- Name
- X-axis coordinates of the starting point of the detection box
- Y axis coordinates of the starting point of the detection box
- X-axis coordinates of terminal point of detection frame
- Y-axis coordinates of terminal point of detection frame
- Confidence Rate
- ✓ Width of the bounding box
- Height of the bounding box
- X Coordinate of the center point
- Y Coordinate of the center point



The car needs to stop at the right location to unload the materials.

By detecting the Y-axis coordinate of the detection box's centre point, the distance between the car and the sign is determined.

Recognize Traffic Signs



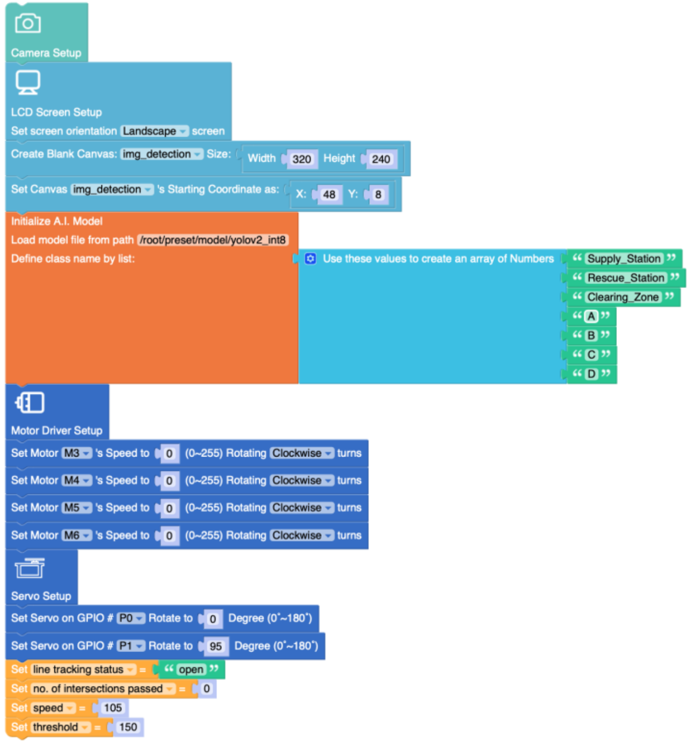
```
If    From i Get the recognized Object's Name = " Supply_Station "
  and From i Get the recognized Object's Y Coordinate of the center point < 80
Do  unloading

Define unloading
do  front
    Wait 1000 Milliseconds
    left
    Wait 300 Milliseconds
    stop
    Wait 500 Milliseconds
    Set Servo on GPIO # P0 Rotate to 110 Degree (0˚~180˚)
    Wait 2000 Milliseconds
    Set Servo on GPIO # P0 Rotate to 0 Degree (0˚~180˚)
    Wait 500 Milliseconds
    right
    Wait 300 Milliseconds
    Set no. of intersections passed = 0
```

*The value of the Y-axis coordinate needs to be obtained through actual testing.

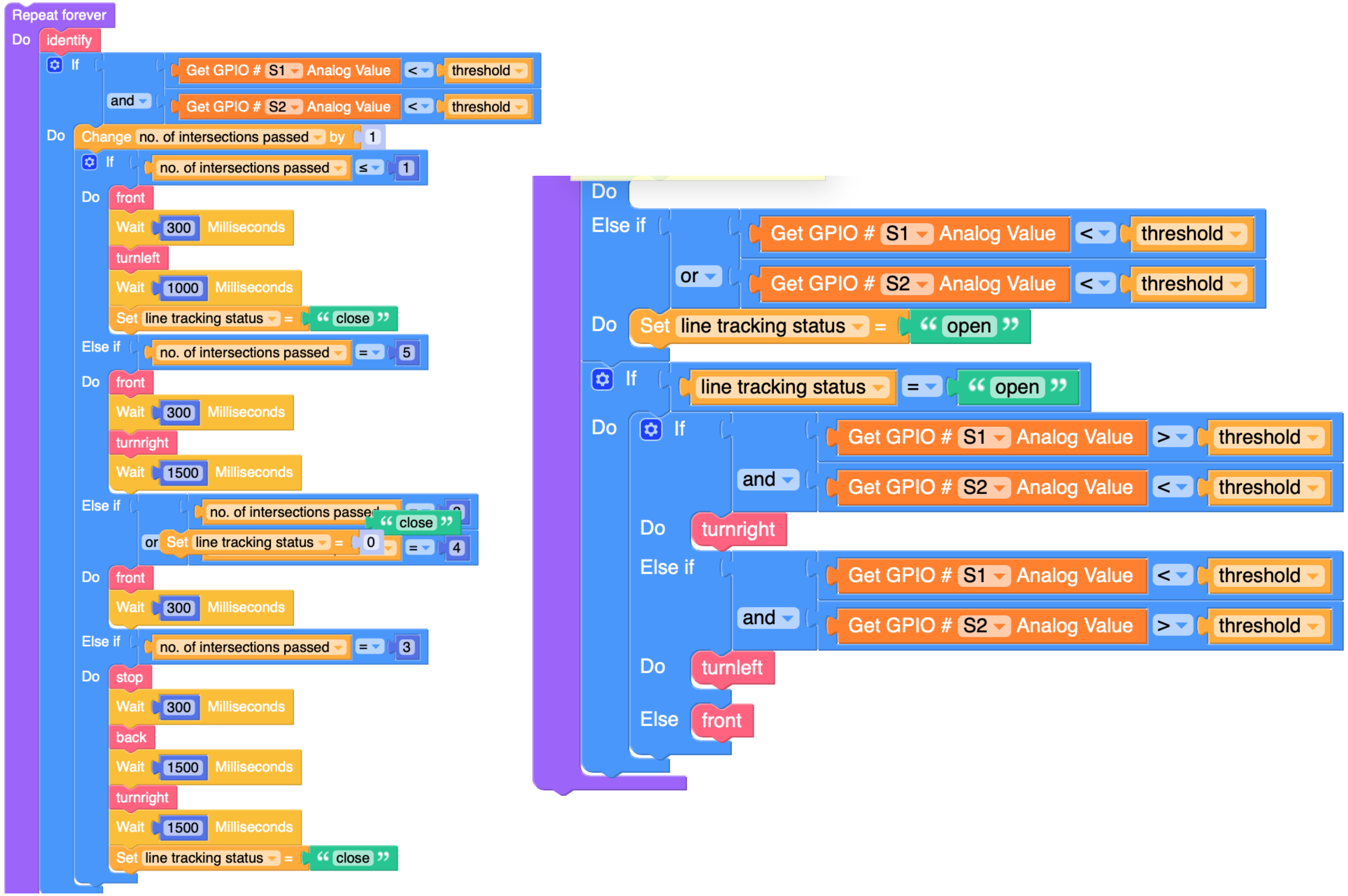
Recognize Traffic Signs

*This page contains the main program (functions can be found on the next page).

Reference Program



```
Camera Setup
LCD Screen Setup
Set screen orientation Landscape screen
Create Blank Canvas: img_detection Size: Width 320 Height 240
Set Canvas img_detection 's Starting Coordinate as: X: 48 Y: 8
Initialize A.I. Model
Load model file from path /root/preset/model/yolov2_int8
Define class name by list: Use these values to create an array of Numbers
    "Supply_Station"
    "Rescue_Station"
    "Clearing_Zone"
    "A"
    "B"
    "C"
    "D"
Motor Driver Setup
Set Motor M3 's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M4 's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M5 's Speed to 0 (0~255) Rotating Clockwise turns
Set Motor M6 's Speed to 0 (0~255) Rotating Clockwise turns
Servo Setup
Set Servo on GPIO # P0 Rotate to 0 Degree (0˚~180˚)
Set Servo on GPIO # P1 Rotate to 95 Degree (0˚~180˚)
Set line tracking status = "open"
Set no. of intersections passed = 0
Set speed = 105
Set threshold = 150
```



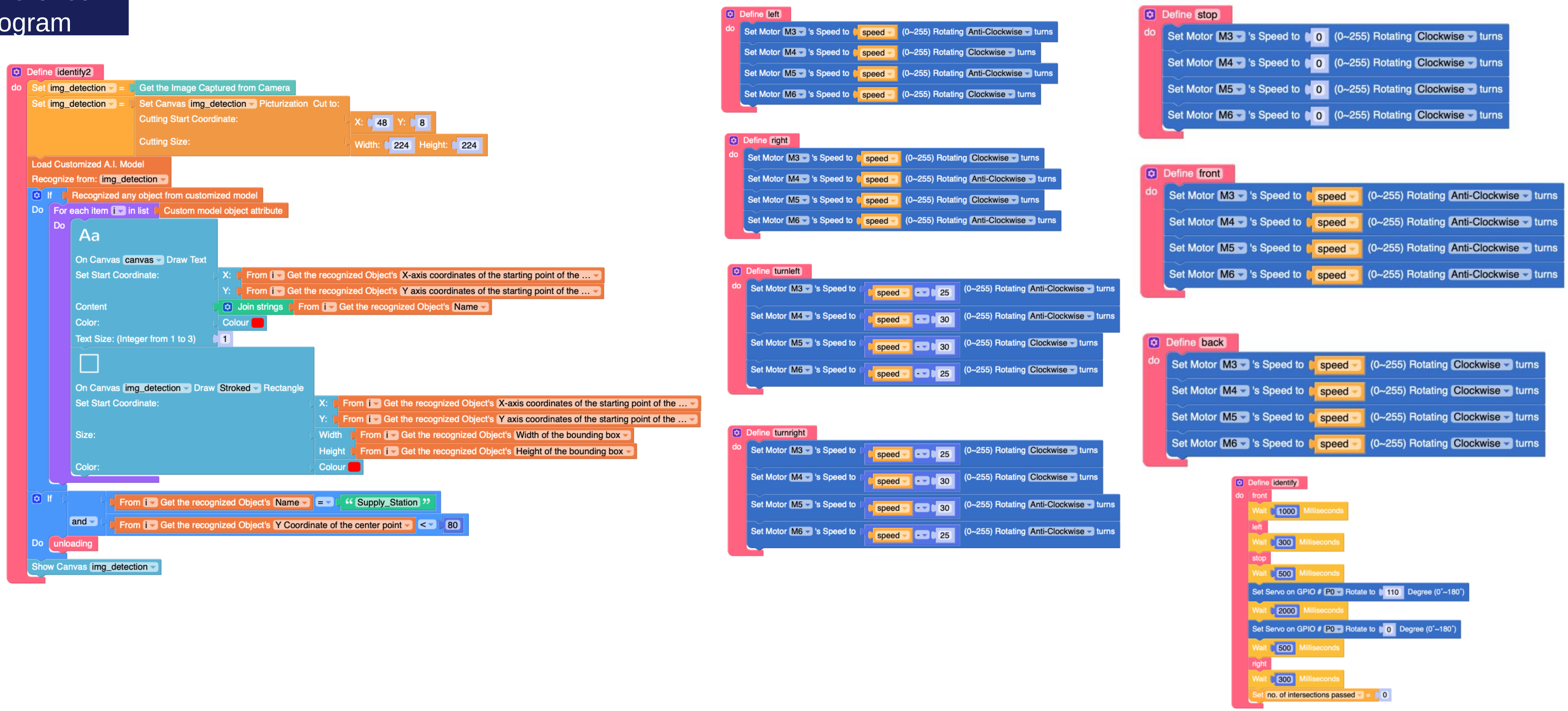
```
Repeat forever
Do  identify
    If      Get GPIO # S1 Analog Value < threshold
        and Get GPIO # S2 Analog Value < threshold
    Do  Change no. of intersections passed by 1
        If no. of intersections passed ≤ 1
        Do  front
            Wait 300 Milliseconds
            turnleft
            Wait 1000 Milliseconds
            Set line tracking status = "close"
        Else if no. of intersections passed = 5
        Do  front
            Wait 300 Milliseconds
            turnright
            Wait 1500 Milliseconds
        Else if no. of intersections passe "close"
             or Set line tracking status = 0  = 4
        Do  front
            Wait 300 Milliseconds
        Else if no. of intersections passed = 3
        Do  stop
            Wait 300 Milliseconds
            back
            Wait 1500 Milliseconds
            turnright
            Wait 1500 Milliseconds
            Set line tracking status = "close"
```

```
Do
Else if     Get GPIO # S1 Analog Value < threshold
        or  Get GPIO # S2 Analog Value < threshold
Do  Set line tracking status = "open"
If line tracking status = "open"
Do  If      Get GPIO # S1 Analog Value > threshold
        and Get GPIO # S2 Analog Value < threshold
    Do  turnright
    Else if     Get GPIO # S1 Analog Value < threshold
            and Get GPIO # S2 Analog Value > threshold
    Do  turnleft
    Else front
```

Recognize Traffic Signs

*This page contains the main program (functions can be found on the next page).

Reference Program



```
Define identify2
do  Set img_detection = Get the Image Captured from Camera
    Set img_detection = Set Canvas img_detection Picturization  Cut to:
                        Cutting Start Coordinate:  X: 48  Y: 8
                        Cutting Size:  Width: 224  Height: 224
    Load Customized A.I. Model
    Recognize from: img_detection
    If  Recognized any object from customized model
    Do  For each item i in list  Custom model object attribute
        Do  Aa
            On Canvas canvas Draw Text
            Set Start Coordinate:  X: From i Get the recognized Object's X-axis coordinates of the starting point of the ...
                                   Y: From i Get the recognized Object's Y axis coordinates of the starting point of the ...
            Content  Join strings  From i Get the recognized Object's Name
            Color:  Colour 🟥
            Text Size: (Integer from 1 to 3)  1
            □
            On Canvas img_detection Draw Stroked Rectangle
            Set Start Coordinate:  X: From i Get the recognized Object's X-axis coordinates of the starting point of the ...
                                   Y: From i Get the recognized Object's Y axis coordinates of the starting point of the ...
            Size:  Width  From i Get the recognized Object's Width of the bounding box
                   Height From i Get the recognized Object's Height of the bounding box
            Color:  Colour 🟥
    If  From i Get the recognized Object's Name = " Supply_Station "
        and  From i Get the recognized Object's Y Coordinate of the center point < 80
    Do  unloading
    Show Canvas img_detection
```

```
Define left
do  Set Motor M3 's Speed to speed (0~255) Rotating Anti-Clockwise turns
    Set Motor M4 's Speed to speed (0~255) Rotating Clockwise turns
    Set Motor M5 's Speed to speed (0~255) Rotating Anti-Clockwise turns
    Set Motor M6 's Speed to speed (0~255) Rotating Clockwise turns
```

```
Define right
do  Set Motor M3 's Speed to speed (0~255) Rotating Clockwise turns
    Set Motor M4 's Speed to speed (0~255) Rotating Anti-Clockwise turns
    Set Motor M5 's Speed to speed (0~255) Rotating Clockwise turns
    Set Motor M6 's Speed to speed (0~255) Rotating Anti-Clockwise turns
```

```
Define turnleft
do  Set Motor M3 's Speed to speed - 25 (0~255) Rotating Anti-Clockwise turns
    Set Motor M4 's Speed to speed - 30 (0~255) Rotating Anti-Clockwise turns
    Set Motor M5 's Speed to speed - 30 (0~255) Rotating Clockwise turns
    Set Motor M6 's Speed to speed - 25 (0~255) Rotating Clockwise turns
```

```
Define turnright
do  Set Motor M3 's Speed to speed - 25 (0~255) Rotating Clockwise turns
    Set Motor M4 's Speed to speed - 30 (0~255) Rotating Clockwise turns
    Set Motor M5 's Speed to speed - 30 (0~255) Rotating Anti-Clockwise turns
    Set Motor M6 's Speed to speed - 25 (0~255) Rotating Anti-Clockwise turns
```

```
Define stop
do  Set Motor M3 's Speed to 0 (0~255) Rotating Clockwise turns
    Set Motor M4 's Speed to 0 (0~255) Rotating Clockwise turns
    Set Motor M5 's Speed to 0 (0~255) Rotating Clockwise turns
    Set Motor M6 's Speed to 0 (0~255) Rotating Clockwise turns
```

```
Define front
do  Set Motor M3 's Speed to speed (0~255) Rotating Anti-Clockwise turns
    Set Motor M4 's Speed to speed (0~255) Rotating Anti-Clockwise turns
    Set Motor M5 's Speed to speed (0~255) Rotating Anti-Clockwise turns
    Set Motor M6 's Speed to speed (0~255) Rotating Anti-Clockwise turns
```

```
Define back
do  Set Motor M3 's Speed to speed (0~255) Rotating Clockwise turns
    Set Motor M4 's Speed to speed (0~255) Rotating Clockwise turns
    Set Motor M5 's Speed to speed (0~255) Rotating Clockwise turns
    Set Motor M6 's Speed to speed (0~255) Rotating Clockwise turns
```

```
Define identify
do  front
    Wait 1000 Milliseconds
    left
    Wait 300 Milliseconds
    stop
    Wait 500 Milliseconds
    Set Servo on GPIO # P0 Rotate to 110 Degree (0˚~180˚)
    Wait 2000 Milliseconds
    Set Servo on GPIO # P0 Rotate to 0 Degree (0˚~180˚)
    Wait 500 Milliseconds
    right
    Wait 300 Milliseconds
    Set no. of intersections passed = 0
```

P O W

E

See you in the next class!

THANK YOU

P

JUST LEAVE PRESENTATION TO ORI N

COCOROBO