

แบบจำลองหุ่นยนต์สำหรับพัฒนาทักษะการเขียนโปรแกรมในเด็กประถม โดย<sup>†</sup>  
ใช้ตัวต่อที่สร้างด้วยกระดาษเป็นชุดคำสั่ง

**ROBOT SIMULATION FOR LEARN PROGRAMMING IN  
PRIMARY SCHOOL USING A PHYSICAL JIGSAW PUZZLE AS  
COMMAND**

โดย  
เฉลิมวงศ์ จัํหนونงโพธิ<sup>‡</sup>

**CHALERMWONG JUMNONGPHO**  
นนทกร มาคีรี  
**NONTHAKORN MARKEEREE**

ปริญญาบัณฑิตนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต<sup>†</sup>  
สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ

คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ภาคเรียนที่ 1 ปีการศึกษา 2563

แบบจำลองหุ่นยนต์สำหรับพัฒนาทักษะการเขียนโปรแกรมในเด็กประถม โดย  
ใช้ตัวต่อที่สร้างด้วยกระดาษเป็นชุดคำสั่ง

**ROBOT SIMULATION FOR LEARN PROGRAMMING IN PRIMARY  
SCHOOL USING A PHYSICAL JIGSAW PUZZLE AS COMMAND**

โดย  
เฉลิมวงศ์ จัํหนอนโพธิ์  
**CHALERMWONG JUMNONGPHO**  
นนทกร มากีรี  
**NONTHAKORN MARKEEREE**

อาจารย์ที่ปรึกษา  
ดร.ชยานนท์ ทรัพย์อาภา

ปริญญาบัณฑิตนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต  
สาขาวิชาวิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ  
คณะเทคโนโลยีสารสนเทศ  
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ภาคเรียนที่ 1 ปีการศึกษา 2563

**ROBOT SIMULATION FOR LEARN PROGRAMMING IN PRIMARY  
SCHOOL USING A PHYSICAL JIGSAW PUZZLE AS COMMAND**

**CHALERMWONG JUMNONGPHO  
NONTHAKORN MARKEEREE**

**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT  
OF THE REQUIREMENT FOR THE DEGREE OF  
BACHELOR OF SCIENCE PROGRAM IN DATA SCIENCE AND  
BUSINESS ANALYTICS  
FACULTY OF INFORMATION TECHNOLOGY  
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

**COPYRIGHT 2020**

**FACULTY OF INFORMATION TECHNOLOGY**

**KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

# ใบรับรองปริญานิพนธ์ ประจำปีการศึกษา 2563

คณะเทคโนโลยีสารสนเทศ

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง แบบจำลองหุ่นยนต์สำหรับพัฒนาทักษะการเขียนโปรแกรมในเด็ก  
ประถมโดยใช้ตัวต่อที่สร้างด้วยกระดาษเป็นชุดคำสั่ง

**ROBOT SIMULATION FOR LEARN PROGRAMMING IN  
PRIMARY SCHOOL USING A PHYSICAL JIGSAW PUZZLE  
AS COMMAND**

ผู้จัดทำ

1.นายเฉลิมวงศ์ จำหนองโพธิ รหัสนักศึกษา 60070131

2.นายนนทกร มากิริ รหัสนักศึกษา 60070144



.....อาจารย์ที่ปรึกษา  
(ดร.ชยานนท์ ทรัพย์อาภา)

## ใบรับรองโครงการ (PROJECT)

เรื่อง

แบบจำลองหุ่นยนต์สำหรับพัฒนาทักษะการเขียนโปรแกรมในเด็กประถม โดย<sup>ใช้ตัวต่อที่สร้างด้วยกระดาษเป็นชุดคำสั่ง</sup>

**ROBOT SIMULATION FOR LEARN PROGRAMMING IN PRIMARY  
SCHOOL USING A PHYSICAL JIGSAW PUZZLE AS COMMAND**

นายเฉลิมวงศ์ จำหนองโพธิ์

รหัสนักศึกษา 60070131

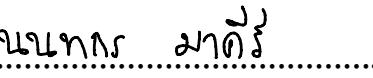
นายนนทกร มากิริ

รหัสนักศึกษา 60070144

ขอรับรองว่ารายงานฉบับนี้ ข้าพเจ้าไม่ได้คัดลอกมาจากที่ใด  
รายงานฉบับนี้ได้รับการตรวจสอบและอนุมัติให้เป็นส่วนหนึ่งของ  
การศึกษาวิชาโครงการ หลักสูตรวิทยาศาสตรบัณฑิต (วิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ)  
ภาคเรียนที่ 1 ปีการศึกษา 2563

  
.....

(นายเฉลิมวงศ์ จำหนองโพธิ์)

  
.....

(นายนนทกร มากิริ)

|                         |   |                       |
|-------------------------|---|-----------------------|
| <b>หัวข้อโครงการ</b>    | แบบจำลองหุ่นยนต์สำหรับพัฒนาทักษะการเรียน โปรแกรมในเด็กประถม โดยใช้ตัดต่อที่สร้างด้วยกระดาษเป็นชุดคำสั่ง |                       |
| <b>นักศึกษา</b>         | นายณัลิมวงศ์ จ้าหนองโพธิ์   | รหัสนักศึกษา 60070131 |
|                         | นายนนทกร มาครี  | รหัสนักศึกษา 60070144 |
| <b>ปริญญา</b>           | วิทยาศาสตรบัณฑิต  |                       |
| <b>สาขาวิชา</b>         | วิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ   |                       |
| <b>ปีการศึกษา</b>       | 2563  |                       |
| <b>อาจารย์ที่ปรึกษา</b> | ดร.ชยานนท์ ทรัพย์อภา  |                       |

## บทคัดย่อ

กิจกรรมอันบล็อก(Unplug) เป็นกิจกรรมการเรียนการสอน โปรแกรมมิ่งที่ไม่ใช้คอมพิวเตอร์ เพื่อให้เกิดความเข้าใจในหลักการพื้นฐานของคอมพิวเตอร์และตรรกศาสตร์ แต่ด้วยข้อจำกัดในการใช้งานคอมพิวเตอร์ ทำให้กิจกรรมการเรียนรู้แบบ Unplug สามารถตอบสนองผู้เรียน ในระดับ ประถมศึกษา ได้ดีกว่าการเรียนรู้โดยใช้คอมพิวเตอร์จริง โดยการเรียนรู้ผ่านกิจกรรมแบบลงมือทำ และการแก้ปัญหาของผู้เรียน โดยในแต่ละกิจกรรมจะมีจุดประสงค์ย่อยที่แตกต่างกัน แต่ยังคงทำให้ ผู้เล่น ได้ฝึกทักษะการแก้ปัญหา การคิดอย่างเป็นระบบ และมีทักษะในการสื่อสารที่ดี โดยทาง ผู้จัดทำจะทำการสร้างเครื่องมือที่ใช้ช่วยสนับสนุนการเรียนการสอน โปรแกรมมิ่ง ที่ไม่ต้องใช้ คอมพิวเตอร์ (Unplugged Programming) ซึ่งจะเน้นให้เด็กมีกิจกรรมร่วมกัน โดยการต่อจิ๊กซอว์ และ ใช้กล้องถ่ายภาพชุดคำสั่งจากจิ๊กซอว์เพื่อเป็นสั่งคำสั่งในการทำโปรแกรมมิ่ง และแสดงผลการ ทำงานของคำสั่งบนแอปพลิเคชั่น ทำให้เด็กได้รับความรู้และความเข้าใจในหลักการทำางานพื้นฐาน ของคอมพิวเตอร์

|                      |  |            |          |
|----------------------|--|------------|----------|
| <b>Project Title</b> | ROBOT SIMULATION FOR LEARN PROGRAMMING IN PRIMARY SCHOOL USING A PHYSICAL JIGSAW PUZZLE AS COMMAND |            |          |
| <b>Student</b>       | Mr. Chalermwong Jumnongpho   | Student ID | 60070131 |
|                      | Mr. Nonthakorn Markeeree   | Student ID | 60070144 |
| <b>Degree</b>        | Bachelor of Science  |            |          |
| <b>Program</b>       | Data Science and Business Analytic   |            |          |
| <b>Academic Year</b> | 2020   |            |          |
| <b>Advisor</b>       | Dr. Chayanon Subrpa  |            |          |

## ABSTRACT

Unplug is an activity that does not use computer to program for gain an understanding of the basic principle of computer and logic. But due to limitations in computer usage. Therefore, learning unplugs activities is more responsive to primary school students than learning by using real computers by learning through hands-on activities and problem solving of primary school students. Each unplug activity has a different purpose. But still help participants to practice problem solving skills, systematic thinking and have good communication skills. In which the researcher will create tools to support teaching and learning programming. That does not require a computer (Unplugged Programming). Which will focus on primary school students to do activities together by connecting the jigsaw and use the camera to take a set of instructions from a jigsaw to order commands in programming and display the results of the command on the application. Helping students to gain knowledge and understanding of basic working principles of computers

## **กิตติกรรมประกาศ**

วิทยานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยความกรุณาจาก ดร.ชยานนท์ ทรัพย์อากา อาจารย์ที่ปรึกษา วิทยานิพนธ์ที่ได้ให้คำเสนอแนะและแนวคิด ตลอดจนแก้ไขข้อบกพร่องเพื่อให้ วิทยานิพนธ์ ฉบับนี้มีความสมบูรณ์ ขอรบกวนพระคุณ คุณพ่อ คุณแม่ ที่เคยให้กำลังใจกันเสมอมา ขอบคุณอาจารย์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังที่ให้คำปรึกษา และ สุดท้ายนี้ ขอบคุณ พี่อนนกศึกษาคนละเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่เคยช่วยให้คำแนะนำดี ๆ และเป็นกำลังใจให้โดยตลอด

เฉลิมวงศ์ จ้านองโพธิ์

นนทกร มากีริ

# สารบัญ

หน้า

|  |     |
|--|-----|
| บทคัดย่อภาษาไทย .....                              | I   |
| บทคัดย่อภาษาอังกฤษ .....                           | II  |
| กิตติกรรมประกาศ.....                               | III |
| สารบัญ .....                                       | IV  |
| สารบัญรูป .....                                    | VI  |
| สารบัญตาราง .....                                  | IX  |
| บทที่  |     |
| 1. บทนำ.....                                       | 1   |
| 1.1 ความเป็นมาและความสำคัญของปัญหา.....            | 1   |
| 1.2 ความมุ่งหมายและวัตถุประสงค์ในการศึกษา.....     | 2   |
| 1.3 ขอบเขตของโครงงาน .....                         | 2   |
| 1.4 ขั้นตอนการศึกษา.....                           | 2   |
| 1.5 ประโยชน์ที่คาดว่าจะได้รับ.....                 | 3   |
| 2. การทบทวนวรรณกรรมที่เกี่ยวข้อง .....             | 4   |
| 2.1 ทฤษฎีบันท .....                                | 4   |
| 2.1.1 กิจกรรมการเรียนรู้แบบอันปลั๊ก (Unplug) ..... | 4   |
| 2.1.2 กิจกรรมที่เกี่ยวข้องกับ CS Unplugged.....    | 5   |
| 2.1.3 การตรวจจับวัตถุ (Object Detection) .....     | 8   |
| 2.1.4 YOLO (You Only Look Once) Version 3 .....    | 12  |
| 2.1.5 YOLO (You Only Look Once) Version 4 .....    | 12  |
| 2.2 เทคโนโลยีที่เกี่ยวข้อง .....                   | 14  |
| 2.2.1 Python .....                                 | 14  |
| 2.2.2 Pygame .....                                 | 14  |
| 2.2.3 OpenCV .....                                 | 15  |
| 2.2.4 Tensorflow .....                             | 15  |
| 2.2.5 Webcam .....                                 | 15  |
| 2.2.6 Darknet .....                                | 16  |

# สารบัญ(ต่อ)

หน้า

|   |    |
|---|----|
| 3. การออกแบบและการพัฒนา .....                         | 17 |
| 3.1 การออกแบบแพนกາพยสเคส.....                         | 17 |
| 3.2 ออกแบบส่วนของระบบ .....                           | 23 |
| 3.3 ออกแบบรูปภาพจีกซอว์ .....                         | 26 |
| 3.4 ติดตั้ง YOLO (You only look once) .....           | 28 |
| 3.5 สร้าง Label Image.....                            | 30 |
| 4. ผลดำเนินงานระบบแบบจำลองหุ่นยนต์ .....              | 32 |
| 4.1 ผลการทดลองตรวจรูปภาพ.....                         | 32 |
| 4.2 การทดลองของระบบ.....                              | 35 |
| 4.3 ข้อจำกัดในการใช้งานระบบ .....                     | 39 |
| 4.4 สรุปขั้นตอนการทำงานของระบบ .....                  | 39 |
| 5. สรุปผลการวิจัย อุปสรรค และการพัฒนาต่อในอนาคต ..... | 40 |
| 5.1 สรุปผลการดำเนินงานวิจัย .....                     | 40 |
| 5.2 อุปสรรคระหว่างดำเนินงานวิจัย .....                | 40 |
| 5.3 สิ่งที่จะพัฒนาต่อยอดในอนาคต .....                 | 41 |
| บรรณานุกรม .....                                      | 42 |
| ภาคผนวก .....   | 45 |
| ภาคผนวก ก .....                                       | 46 |
| ภาคผนวก ข .....                                       | 52 |

# สารบัญ

| รูปที่  | หน้า |
|---|------|
| 2.1 ตำแหน่งของ Bot และบ้านบนตาราง .....                             | 6    |
| 2.2 ตำแหน่งของ Bot และบ้านบนตารางเมื่อออกรคำสั่ง .....              | 6    |
| 2.3 ตำแหน่งของ Bot และบ้านบนตารางเมื่อออกรคำสั่งถูกต้อง.....        | 7    |
| 2.4 แสดงความยากโดยเพิ่มสิ่งกีดขวาง .....                            | 8    |
| 2.5 ตัวอย่าง Object Detection .....                                 | 8    |
| 2.6 ขั้นตอนการทำงานของ R-CNN .....                                  | 9    |
| 2.7 ขั้นตอนการทำงานของ Fast R-CNN .....                             | 10   |
| 2.8 ขั้นตอนการทำงานของ Faster R-CNN .....                           | 11   |
| 2.9 การทำงานของ RPN .....   | 11   |
| 2.10 เปรียบเทียบความเร็วของ R-CNN, Faster R-CNN, Faster R-CNN ..... | 11   |
| 2.11 Bounding Box.....  | 12   |
| 2.12 แสดงการเปรียบเทียบ YOLO v4 ,YOLO v3 และ EfficientDet .....     | 13   |
| 2.13 Python.....  | 14   |
| 2.14 Pygame .....   | 14   |
| 2.15 OpenCV .....   | 15   |
| 2.16 Tensorflow.....  | 15   |
| 2.17 Darknet .....  | 16   |
| 3.1 ภาพ Use case diagram ของระบบแบบจำลองหุ่นยนต์ .....              | 17   |
| 3.2 ระบบแบบจำลองหุ่นยนต์ .....                                      | 23   |
| 3.3 อธิบายรูปทรงจิ๊กซอว์.....                                       | 24   |
| 3.4 ตัวอย่างไฟล์ JSON .....   | 25   |
| 3.5 ตัวอย่างของค่าที่แสดงผลจากไฟล์ JSON.....                        | 25   |
| 3.6 ตัวอย่างการใส่คำสั่งเพื่อ Detect รูปภาพ .....                   | 28   |
| 3.7 แสดง Class และค่า Accuracy จากรูปภาพที่ Detect .....            | 29   |
| 3.8 แสดงรูปภาพผลลัพธ์จากการ Detect.....                             | 29   |
| 3.9 การทำ Label Image .....   | 30   |
| 3.10 ไฟล์ Text ที่ได้หลังจากการทำ Label Image .....                 | 30   |

# สารบัญรูป(ต่อ)

| รูปที่  | หน้า |
|---|------|
| 3.11 ความหมายของคำใน Label .....  | 31   |
| 3.12 ขั้นตอนการดำเนินการทดลอง Object detection โดยใช้ YOLO.....           | 31   |
| 4.1 กราฟแสดงค่า Average loss ในระหว่างการ Train ของ YOLO v3.....          | 33   |
| 4.2 กราฟแสดงค่า Average loss ในระหว่างการ Train ของ YOLO v4.....          | 33   |
| 4.3 ผลลัพธ์จากการ Detect รูปภาพจีกซอร์ เทคนด้วย YOLO v3.....              | 34   |
| 4.4 ผลลัพธ์จากการ Detect รูปภาพจีกซอร์ เทคนด้วย YOLO v4.....              | 34   |
| 4.5 ตัวอย่างการแสดงตำแหน่งของ Bounding Box ของรูปภาพที่ทำการ Detect ..... | 35   |
| 4.6 แสดงคำอธิบายรูปทรงจีกซอร์ในระบบ.....                                  | 35   |
| 4.7 ตัวอย่างระบบในด้านที่ 1 .....   | 36   |
| 4.8 แสดงข้อความ “SUCCESSFUL” เมื่อ Robot เดินถึงเป้าหมาย.....             | 36   |
| 4.9 แสดงข้อความ “UNSUCCESSFUL” เมื่อ Robot เดินไม่ถึงเป้าหมาย .....       | 37   |
| 4.10 แสดงข้อความ “TRY AGAIN” เมื่อไม่เจอกีกซอร์ในรูปภาพ .....             | 47   |
| 4.11 การต่อจีกซอร์เพื่อแก้โจทย์วิธีปกติ .....                             | 38   |
| 4.12 การต่อจีกซอร์เพื่อแก้โจทย์วิธี Loop .....                            | 38   |
| 4.13 กระบวนการทำงานของระบบ .....  | 39   |
| ก.1 ตัวอย่างไฟล์รูปภาพและไฟล์ Label ทั้งหมด .....                         | 47   |
| ก.2 ไฟล์ obj.names .....  | 48   |
| ก.3 ไฟล์ obj.data.....  | 48   |
| ก.4 ไฟล์ train.txt.....   | 49   |
| ก.5 ไฟล์ test.txt .....   | 49   |
| ก.6 ไฟล์ yolov4-custom.cfg .....  | 50   |
| ก.7 กราฟแสดงค่า avg loss ในระหว่างการเทรน .....                           | 51   |
| ข.1 เลือก Command Prompt ในช่องค้นหา.....                                 | 53   |
| ข.2 pip install virtualenv .....  | 53   |
| ข.3 คัดลอก path ไฟล์เดอร์ระบบ.....  | 54   |
| ข.4 เข้าไปที่ไฟล์เดอร์ระบบด้วยคำสั่ง cd (folder path).....                | 54   |
| ข.5 สร้าง environment ด้วยคำสั่ง virtualenv env_name .....                | 54   |

## สารบัญ(ต่อ)

| รูปที่  | หน้า |
|---|------|
| ข.6 สร้าง environment แบบกำหนด path กรณีมี python หลาย version..... | 55   |
| ข.7 คำสั่งการเข้าใช้งาน environment.....                            | 55   |
| ข.8 แสดงชื่อ environment เมื่อเข้าใช้งาน .....                      | 55   |
| ข.9 คำสั่งในการติดตั้ง library ใน requirement .....                 | 55   |
| ข.10 คำสั่งในการติดตั้ง library ใน requirement-gpu.....             | 56   |
| ข.11 การใส่คำสั่งในการ โหลดและเชฟ Model ใน Yolo .....               | 56   |
| ข.12 คำสั่งเพื่อเปิดระบบ .....                                      | 56   |

## สารบัญตาราง

| ตารางที่   | หน้า   |
|--|--|
| 3.1 รายละเอียดข้อมูลในกราฟเชิงเส้นที่ต้องการรับจาก API .....<br>3.2 รายละเอียดข้อมูลที่ต้องการส่งไปยัง API .....<br>3.3 รายละเอียดข้อมูลที่ต้องการรับจาก API .....<br>3.4 รายละเอียดข้อมูลที่ต้องการส่งไปยัง API .....<br>3.5 รายละเอียดข้อมูลที่ต้องการรับจาก API .....<br>3.6 รายละเอียดข้อมูลที่ต้องการรับจาก API .....<br>3.7 รายละเอียดข้อมูลที่ต้องการส่งไปยัง API .....<br>4.1 รายละเอียดข้อมูลที่ต้องการรับจาก API ..... | 18<br>19<br>20<br>21<br>22<br>26<br>26<br>32 |

## บทที่ 1

### บทนำ

ในบทนี้ทางเราจะกล่าวถึงความเป็นมาและความสำคัญของปัญหา สถานการณ์ปัจจุบันในการเรียนโปรแกรมมิ่งของเด็กระดับประถมศึกษา วัตถุประสงค์ในการศึกษาและพัฒนา รวมไปถึงการนำเทคโนโลยีเข้ามาร่วมในการพัฒนา

#### 1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากปัจจุบันภาครัฐต้องการให้เกิดการเรียนโปรแกรมมิ่งในเด็กระดับประถมศึกษา คุณหญิงกัลยา โสภณพนิช รัฐมนตรีช่วยว่าการกระทรวงศึกษาธิการ ได้แต่งนโยบายด้านการศึกษา เพื่อพัฒนาคนสู่ศตวรรษที่ 21 [1] สนับสนุนให้เด็กไทยได้เรียน Coding ภาษาคอมพิวเตอร์ พร้อม พัฒนาหลักสูตรรองรับโลกยุคดิจิทัล จัดโดยสถาบันส่งเสริมการสอนวิทยาศาสตร์และเทคโนโลยี (สวทช) คุณหญิงกัลยา โสภณพนิช กล่าวว่า การศึกษาเป็นหัวใจของการพัฒนาคนให้กับประเทศไทย หนึ่งในนโยบายที่รัฐบาลแต่งต่อรัฐสภา คือการพัฒนาทรัพยากรมนุษย์ด้วยภาษาคอมพิวเตอร์ หรือ Coding เพื่อเตรียมคนไทยสู่ศตวรรษที่ 21 ให้เท่าทันพลวัตรของการเปลี่ยนแปลงอย่างต่อเนื่อง โดย มีเป้าหมายเริ่มอบรมครูที่จะสอน Coding จำนวน 1,000 คน ในเดือนตุลาคม 2562 นี้ เพื่อจะ ได้ สอน นักเรียนชั้นประถมศึกษาปีที่ 1-3 ทันเปิดภาคเรียนที่ 2 ในเดือนพฤษจิกายน ซึ่งระยะแรก จะเรียนโดย ไม่ต้องใช้คอมพิวเตอร์ (Unplugged Coding) เพื่อให้มีพื้นฐานตระกะ การคิดแบบ Coding ก่อน จากนั้น จึงจะสามารถเรียนการสื่อสารกับคอมพิวเตอร์ในระดับชั้นต่อไป ในปัจจุบันการเรียนโดย ใช้คอมพิวเตอร์ในระดับประถมศึกษาเป็นไปได้ยาก เนื่องจากการเรียนโดยใช้คอมพิวเตอร์และ ให้เด็ก ในระดับประถมศึกษาเขียนโปรแกรมคอมพิวเตอร์เอง จะทำให้เด็กในระดับ ประถมศึกษาไม่สนใจในการเรียนนั้นหรือใช้คอมพิวเตอร์ในการเล่นเกมและอาจทำให้เด็กติดเกม ทำให้ในการเรียนครั้งต่อไปเด็กอาจจะไม่สนใจเรียน ซึ่งครูผู้สอนก็ไม่สามารถควบคุมเด็กได้ ทั้งหมดในระหว่างการเรียนการสอน ที่ไม่ต้องใช้คอมพิวเตอร์ หรือที่เรียกว่าอันปลักโปรแกรมมิ่ง (Unplugged programming) โดยที่เน้นให้เด็กมีกิจกรรมร่วมกัน โดยการต่อจิ๊กซอว์และใช้กล้องถ่ายภาพชุดคำสั่งจากจิ๊กซอว์เพื่อเป็นสั่งคำสั่งในการแก้ปัญหาโจทย์โปรแกรมมิ่ง โดยแสดงผลการทำงานผ่านแอปพลิเคชัน ทำให้เด็กมีความสนใจและเข้าถึงการเรียนได้ง่าย ครูผู้สอนก็สามารถควบคุมเด็กได้ดีขึ้น เพราะเด็กจะอยู่ด้วยกันเป็นกลุ่ม เนื่องการทำกิจกรรมร่วมกัน

## 1.2 ความมุ่งหมายและวัตถุประสงค์ในการศึกษา

ผู้จัดทำได้กำหนดวัตถุประสงค์ได้เป็น 2 ข้อ ดังนี้

- 1.2.1 เป็นการสร้างเครื่องมือเพื่อช่วยสนับสนุนการเรียนการสอนในรูปแบบของจีกซอฟ์ หรือตัวต่อที่สามารถแปลงเป็นคำสั่งโปรแกรมมิ่งได้
- 1.2.2 เพื่อให้เด็กในระดับประถมศึกษาได้ช่วยกันมีส่วนร่วมในการต่อจีกซอฟ์เพื่อแก้ปัญหาโจทย์ตามที่ผู้สอนออกแบบไว้

## 1.3 ขอบเขตของโครงการ

ผู้จัดทำด้องให้ความสำคัญต่อการกำหนดขอบเขตของทำโครงการเพื่อให้ได้ผลการศึกษาที่น่าเชื่อถือ ซึ่งได้แก่ การออกแบบจีกซอฟ์ สร้างระบบ ทำโน้ตเดลคำสั่ง โดยมีรายละเอียดดังต่อไปนี้

- 1.3.1 อออกแบบจีกซอฟ์สำหรับการเรียนรู้ของเด็กประถม โดยที่รองรับการทำงานแบบ Iterator (for loop)
- 1.3.2 ทำระบบแบบจำลองหุ่นยนต์ ที่สามารถรับคำสั่งจากรูปภาพ (จีกซอฟ์ที่ออกแบบไว้) และแสดงผลการทำงานคำสั่งนั้น ๆ
- 1.3.3 ศึกษาและทำโน้ตเดลเพื่อแปลงจีกซอฟ์ให้เป็นคำสั่งที่ทำงานกับแบบจำลองหุ่นยนต์
- 1.3.4 นำคำสั่งที่ได้จากการทำโน้ตเดลในการแปลงจีกซอฟ์ให้เป็นคำสั่ง สามารถทำงานบน ระบบแบบจำลองหุ่นยนต์ที่สามารถรับคำสั่งจากรูปภาพ(จีกซอฟ์ที่ออกแบบไว้) และแสดงผลการทำงานคำสั่งนั้น ๆ ได้

## 1.4 ขั้นตอนการศึกษา

เป็นการศึกษาว่าผู้จัดทำมีลำดับขั้นตอนการทำโครงการอย่างไร ซึ่งได้แก่

- 1.4.1 อออกแบบรูปแบบการเรียนรู้แบบอันปลดล็อก (Unplugged) ที่ใช้จีกซอฟ์ในการทำโปรแกรมมิ่ง ซึ่งจีกซอฟ์จะมีสีและรูปทรงที่แตกต่างตามคำสั่งของภาษาโปรแกรม มิ่ง
- 1.4.2 ศึกษาและทดลองใช้งานโน้ตเดลตรวจจับวัตถุจากรูปภาพ (Object Detection) ที่มีอยู่แล้ว เช่น YoLo(You only look once) เพื่อแปลงจีกซอฟ์ให้เป็นคำสั่งที่ใช้งานกับแบบจำลองหุ่นยนต์
- 1.4.3 ทำชุดข้อมูลสำหรับการฝึก (Training Dataset) จากรูปจีกซอฟ์ที่ออกแบบไว้ และทำการ re-training หรือการฝึกอีกรอบ เพื่อเพิ่มประสิทธิภาพของโน้ตเดลที่ตรวจจับวัตถุจากรูปภาพ (Object Detection)
- 1.4.4 อออกแบบและพัฒนาแบบจำลองหุ่นยนต์กับการเรียนอันปลดล็อก โปรแกรมมิ่ง

(Unplugged programming) ที่รองรับ if, else และ for loop

## 1.5 ประโยชน์ที่คาดว่าจะได้รับ

- 1.5.1 เครื่องมือที่ใช้สนับสนุนการเรียนการสอนแบบอันปลั๊ก(Unplugged)
- 1.5.2 ทำให้เด็กประณีตจากการตั้งสมมติฐาน มีการคิดคำนวณ และรู้จักสังเกตการทำงานของหุ่นยนต์ได้
- 1.5.3 สามารถนำไปต่อยอดและพัฒนาในระดับที่สูงขึ้น เช่น เพิ่มคำสั่ง การทำงานแบบมีฟังก์ชัน

## บทที่ 2

# การทบทวนวรรณกรรมที่เกี่ยวข้อง

### 2.1 ทฤษฎี

#### 2.1.1 กิจกรรมการเรียนรู้แบบอันปลั๊ก (Unplug) [2]

เป็นแนวคิดการจัดการเรียนการสอนทางด้านคอมพิวเตอร์เพื่อให้เกิดความเข้าใจในหลักการพื้นฐานของคอมพิวเตอร์และตรรกศาสตร์ โดยผ่านกิจกรรมต่าง ๆ ที่โดยไม่ใช้คอมพิวเตอร์ โดยกระตุนให้ผู้เรียนได้มีการเรียนรู้อย่างสนุกสนาน และสามารถฝึกทักษะการแก้ปัญหา การใช้ความคิดสร้างสรรค์ การคิดอย่างเป็นระบบ และมีทักษะการสื่อสาร ซึ่งเป็นพื้นฐานที่เกี่ยวข้องกับคอมพิวเตอร์ เช่น การเขียนคำสั่ง การเขียนโปรแกรม ลำดับการทำงาน

กิจกรรมการเรียนรู้แบบ Unplug ได้รับความสนใจอย่างมากในทั่วโลก เพราะด้วยการเรียนรู้ที่เปลี่ยนไป และอย่างให้ผู้เรียนได้เรียนรู้หลักการทำงานในด้านตรรกศาสตร์ของคอมพิวเตอร์ แต่ด้วยข้อจำกัดในการใช้งานคอมพิวเตอร์ เช่น คอมพิวเตอร์ไม่พอใช้ในบางโรงเรียน หรือไม่มีอินเทอร์เน็ต ทำให้กิจกรรมการเรียนรู้แบบ Unplug ได้ตอบสนองผู้เรียนในระดับประถมศึกษาได้ดีกว่าการเรียนรู้โดยใช้คอมพิวเตอร์จริง โดยการเรียนรู้ผ่านกิจกรรมแบบลงมือทำและการแก้ปัญหาของผู้เรียน

ในวันศุกร์ที่ 9 สิงหาคม 2562 เวลา 14.00 น. คุณหญิงกัลยา โสภณพนิช รัฐมนตรีช่วยว่าการกระทรวงศึกษาธิการ แถลงนโยบายด้านการศึกษาเพื่อพัฒนาคนสู่ศตวรรษที่ 21 ที่เสนอการเรียนคอมพิวเตอร์โดยไม่ใช้คอมพิวเตอร์ สำหรับเด็กระดับชั้นประถมศึกษาปีที่ 1-3 ทำให้มีการนำแนวคิดที่เรียกว่า Computer Science Unplugged (CS Unplugged) ที่เกิดมาจากการกลุ่มนักวิทยาศาสตร์ด้านคอมพิวเตอร์ชาวนิวซีแลนด์ 3 คนคือ Tim Bell กับ Ian H. Witten และ Mike Fellows ที่ร่วมกันเขียนหนังสือชื่อ “Computer Science Unplugged off-line activities and games for all ages [3]” ในปี 1998 ที่มีเนื้อหาเกี่ยวกับกิจกรรมและเกมการเรียนรู้แบบอффไลน์โดยไม่ใช้คอมพิวเตอร์สำหรับทุกเพศทุกวัย จากนั้นในปี 2015 ได้มีการนำแนวคิด CS Unplugged ของ Tim Bell กับ Ian H. Witten และ Mike Fellows มาทำเป็นบทความในเรื่อง “CS UNPLUGGED An enrichment and extension programme for primary-aged students [4]” โดยบทความนี้จะมีเนื้อหาที่รวมกิจกรรมการเรียนรู้แบบ Unplug มากมายสำหรับเด็กในระดับประถมศึกษาและในแต่ละกิจกรรมจะมีการอธิบายการเล่น อยุหงส์เด็กที่ เหมาะสม การสรุปผลการทำกิจกรรม ได้มีบุคลากรที่ก่อตั้งในไทยของคุณหญิงกัลยา โสภณพนิช รัฐมนตรีช่วยว่าการกระทรวงศึกษาธิการ[5] และได้นำเสนอแนวคิดกิจกรรมชื่อ CS Unplugged ที่จะนำมาใช้ร่วมกับการเรียนรู้ใหม่ด้านคอมพิวเตอร์และตรรกศาสตร์แก่เยาวชน ซึ่งสามารถต่อยอดไปถึงศาสตร์อื่นๆ เช่น อัลกอริทึม

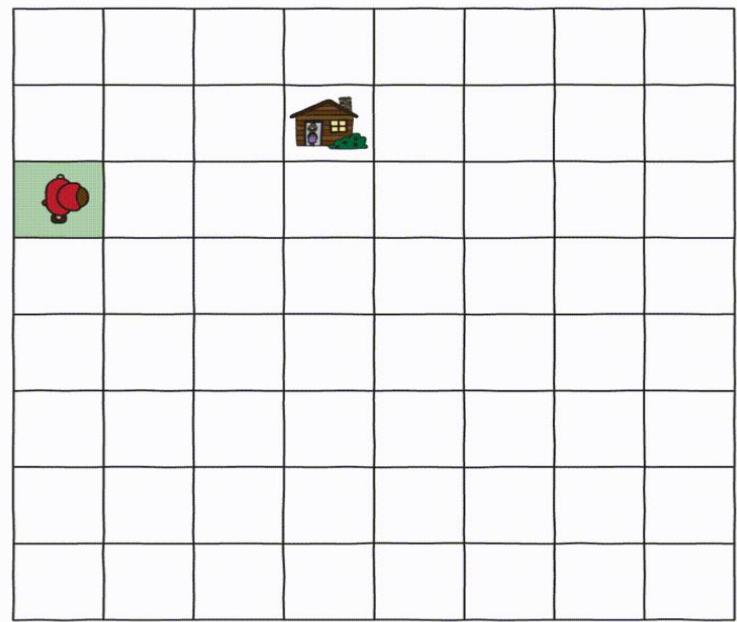
การเขียนโปรแกรม การเขียนโค้ด ตระกูลศาสตร์เป็นต้น โดยแนวคิดนี้หมายความว่าสำหรับการเรียนคอมพิวเตอร์สำหรับเด็กอนุบาลไปจนถึงชั้นประถมปลาย โดยการทำกิจกรรม เช่น “ไฟ” ปริศนา เกมกระดาน เป็นต้น เกมหรือกิจกรรมมักจะเน้นให้เด็กคิดเป็น คิดอย่างเป็นระบบ มีขั้นเชิงกลยุทธ์ ซึ่งทั้งหมดล้วนเป็นพื้นฐานในการต่อยอดการศึกษาต่อในศาสตร์อื่น ๆ ที่เกี่ยวกับคอมพิวเตอร์ โดยแนวคิดนี้เชื่อว่าพัฒนาของเด็กในระดับประถมศึกษานั้นจะตอบสนองการเรียนรู้ภาคปฏิบัติได้กว่าภาคทฤษฎีในห้องเรียน

ในปัจจุบัน แนวคิด CS Unplugged ได้รับความนิยมเพิ่มขึ้นอย่างมาก และมีจุดประสงค์ย่อยที่แตกต่างกัน แต่ยังคงไว้ซึ่งการเรียนคอมพิวเตอร์โดยไม่ใช้คอมพิวเตอร์ที่ได้ทำกิจกรรมที่สนุก และพัฒนาตระกูลศาสตร์ของเด็ก ได้เป็นอย่างดี

### **2.1.2 กิจกรรมที่เกี่ยวข้องกับ CS Unplugged [6]**

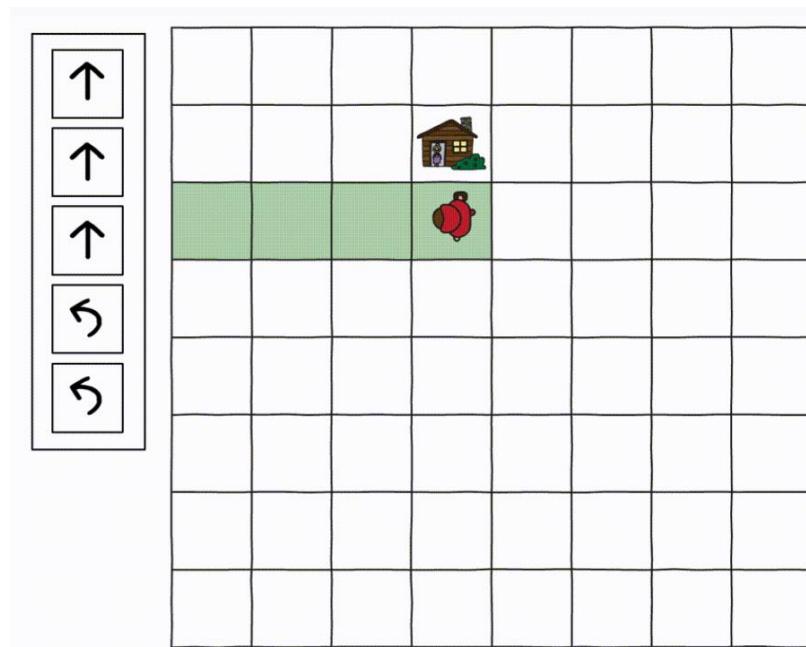
#### **กิจกรรม Rescue Mission**

- เป็นกิจกรรมให้ผู้เล่นเดินบนตารางไปสู่เป้าหมายที่กำหนดโดยใช้คำสั่งโปรแกรมมิ่ง โดยจะมีผู้เล่น 3 คนบาท คือ Developer, Tester และ Bot ซึ่ง Developer จะออกแบบเงื่อนไขของกิจกรรม ส่วน Tester จะเป็นผู้ออกแบบคำสั่งและ Bot จะรับคำสั่งจาก Tester และทำการคำสั่งโดยมีคำสั่งทั้งหมด 3 คำสั่ง คือ Move forward, Turn left , Turn right ใน การเล่นอาจจะมีลูกศรเพื่อกำหนดทิศทางการเดิน คำสั่ง Move forward คือการเดินไปข้างตามทิศทางที่ลูกศรชี้อยู่ คำสั่ง Turn left คือการหันทิศทางไปทางซ้าย 90 องศา แต่ไม่มีการเดิน เริ่มต้นผู้เล่นที่เป็น Bot จะยืนอยู่ในตาราง ดังแสดงตามรูปที่ 2.1 คือผู้หญิงใส่ชุดสีแดงและเป้าหมาย คือ บ้าน สิ่งเหล่านี้จะถูกออกแบบโดย Developer คำสั่งที่ Tester ออกให้ Bot นั้น จะถูกทำงาน 1 ครั้ง เช่น Tester ตั้ง “Move forward” Bot จะเดินไปทิศทางข้างหน้า 1 ช่อง



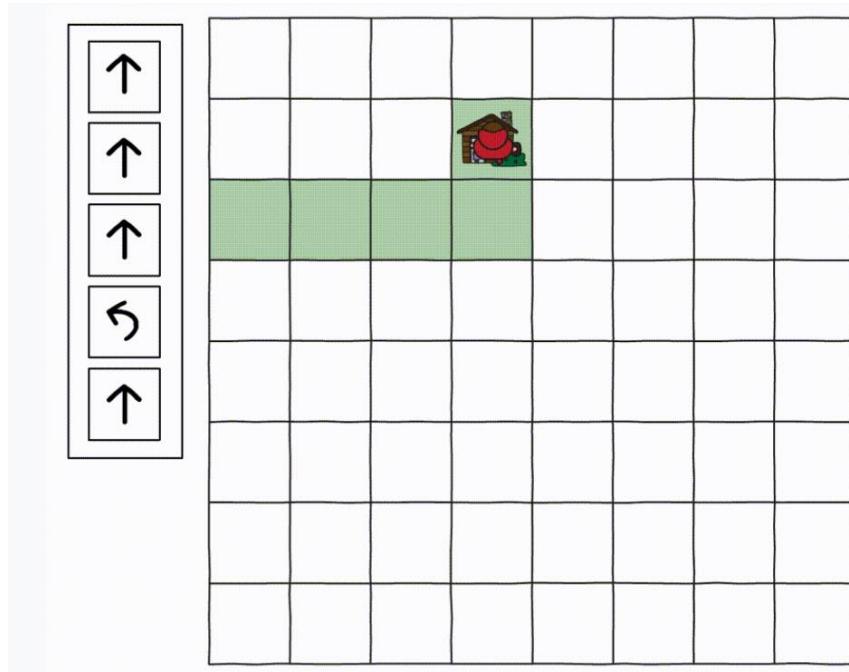
รูปที่ 2.1 ตำแหน่งของ Bot และบ้านบนตาราง

2. จะเป็นการเดินของผู้เล่นที่เป็น Bot และทำการรับคำสั่งจาก Tester ในกรณี Bot ได้หันหน้าไปทางขวาอยู่แล้ว และรับคำสั่งว่า “Move forward , Move forward, Move forward, Turn left, Turn left” พนว่า Bot ยังเดินไม่ถึงจุดและหันหน้าไม่ตรงกับบ้าน ดังแสดงตามรูปที่ 2.2



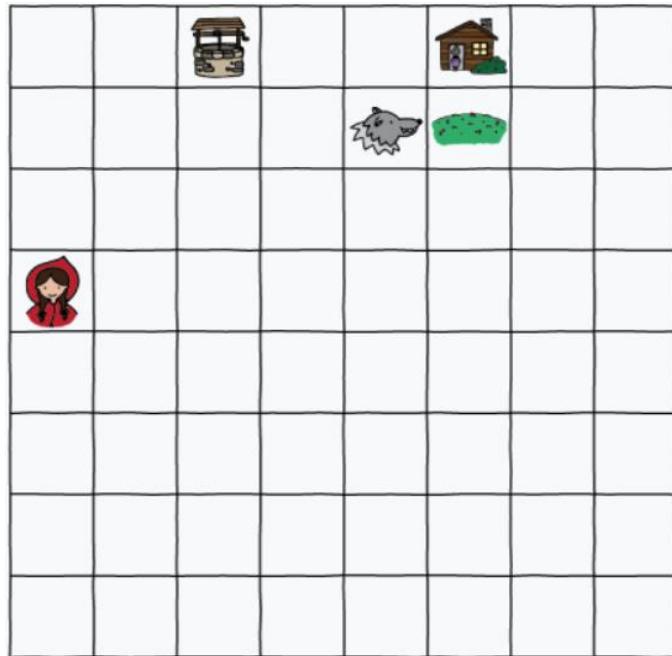
รูปที่ 2.2 ตำแหน่งของ Bot และบ้านบนตารางเมื่อออคำสั่ง

3. ให้ Tester ทำการบอกคำสั่งอีกรอบ “Move forward , Move forward, Move forward, Turn left, Move forward” ครั้งนี้ Bot สามารถ รับคำสั่งและทำตามจนถึงบ้านได้สำเร็จ ดังแสดงตามรูปที่ 2.3 ก็จะจบกิจกรรมในรอบนี้



รูปที่ 2.3 ตำแหน่งของ Bot และบ้านบนตารางเมื่อออกคำสั่งถูกต้อง

4. ในรอบถัดไป Developer สามารถออกแบบให้เพิ่มสิ่งกีดขวางเพื่อให้เพิ่มความยาก ให้กับ Tester ในการออกคำสั่งให้ Bot ดังแสดงตามรูปที่ 2.4 หรือลดคำสั่งเหลือเพียง Move forward และ Turn left



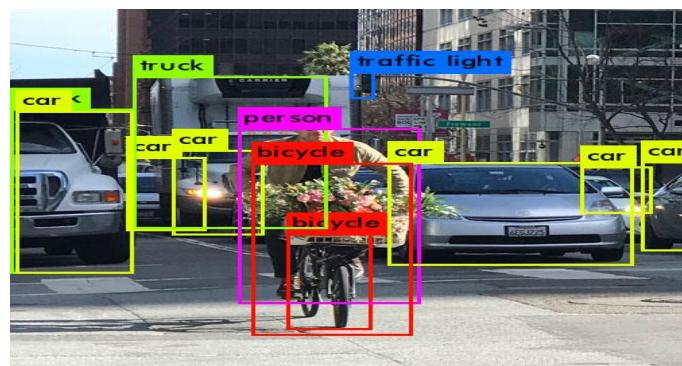
รูปที่ 2.4 แสดงความยากโดยเพิ่มสี่กีดขวาง

ในตัวอย่างของกิจกรรม CS Unplugged ผู้จัดทำจะใช้กิจกรรมนี้เป็นตัวต้นแบบในการทำงาน

### 2.1.3 การตรวจจับวัตถุ (Object Detection) [7]

การตรวจจับวัตถุ คือ เทคโนโลยีทางคอมพิวเตอร์ ที่ใช้สำหรับตรวจจับวัตถุที่สามารถกำหนดเองได้ เช่น มนุษย์ รถยนต์ สุนัข เป็นต้น

Object Detection สามารถเจาะลึกลงไปได้อีกหลายแบบ เช่น การทำ Face Detection [8] Pedestrian Detection [9] และสามารถประยุกต์ใช้ได้หลากหลายงาน เช่น ใช้ในงานสำหรับรักษาความปลอดภัยตรวจจับบุคคลที่ต้องสงสัย หรือการทำให้รถยนต์สามารถขับเคลื่อนได้ โดยไม่มีคนขับ



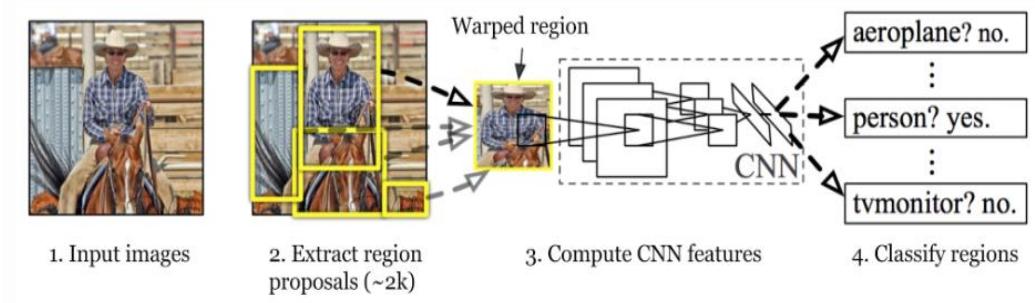
รูปที่ 2.5 ตัวอย่าง Object Detection

(ที่มา: [https://en.wikipedia.org/wiki/Object\\_detection](https://en.wikipedia.org/wiki/Object_detection))

## Algorithm ของ Object Detection

### 1. R-CNN [10]

เป็นหนึ่งในขั้นตอนของ Object detection ที่ถูกเขียนโดย Ross Girshick ซึ่งมีความแม่นยำในการตรวจจับวัตถุในงาน Pascal voc 2010 (กิจกรรมที่เกิดขึ้นเพื่อแบ่งขั้นการตรวจจับวัตถุด้วยคอมพิวเตอร์) สูงถึง 53.7% ซึ่งในปัจจุบัน มีการกำหนด Class ในการตรวจจับทั้งหมด 20 Class และในงาน ILSVRC2013 [11] ได้มีการกำหนดเป็น 200 Class แต่ R-CNN มีความแม่นยำในการตรวจจับวัตถุถึง 31.4% ซึ่งถือว่ามีความแม่นยำสูง ในระดับนี้ R-CNN จะนำพื้นที่บางส่วนของวัตถุ (R - region) มาสกัดโดยใช้ CNN (Convolution Neural Network) [12] คือ การจำลองการมองเห็นของมนุษย์ที่ทำการมองรูปภาพเป็นพื้นที่ย่อย ๆ จากนั้นทำการนำพื้นที่ย่อย ๆ เหล่านั้นมาพسانกันเพื่อดูว่าสิ่งที่มองเห็นอยู่คืออะไร



รูปที่ 2.6 ขั้นตอนการทำงานของ R-CNN

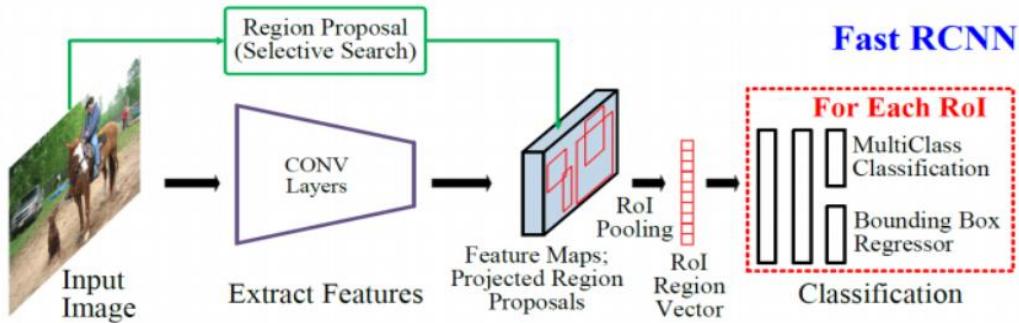
จากรูป 2.6 มีขั้นตอนดังนี้

1. นำเข้ารูปภาพที่ต้องการ Detect
2. จะเป็นการสกัดบริเวณที่คิดว่า “น่าจะเป็นวัตถุ” เรียกว่า Selective search ขึ้นมาโดยทำไม่ต่ำกว่า 2000 ครั้ง
3. จะนำสิ่งที่สิ่งสกัดได้มาคำนวณ โดยใช้ CNN
4. จำแนกแต่ละพื้นที่ในวัตถุโดยใช้ class-specific linear SVMs และได้ผลลัพธ์ออกมาว่าเป็น Class อะไร

โดยปัญหาของ R-CNN คือ ข้าเพราะต้องทำการคำนวณ CNN ทุกรอบตามจำนวนที่ได้จากการสกัด โดยใช้ Selective search เช่น ถ้าพื้นที่ถูกสกัดมา 2,000 อัน นั่นคือต้องคำนวณ CNN ใหม่ทั้งหมด 2,000 รอบ

## 2. Fast R-CNN [13]

Fast R-CNN ถูกปรับปรุงขึ้นเนื่องจาก R-CNN ยังมีประสิทธิภาพในการประมวลผลที่ช้า ซึ่ง Fast R-CNN ประมวลผลเร็วกว่า R-CNN ถึง 10 เท่าและยังมีความแม่นยำสูงกว่า R-CNN



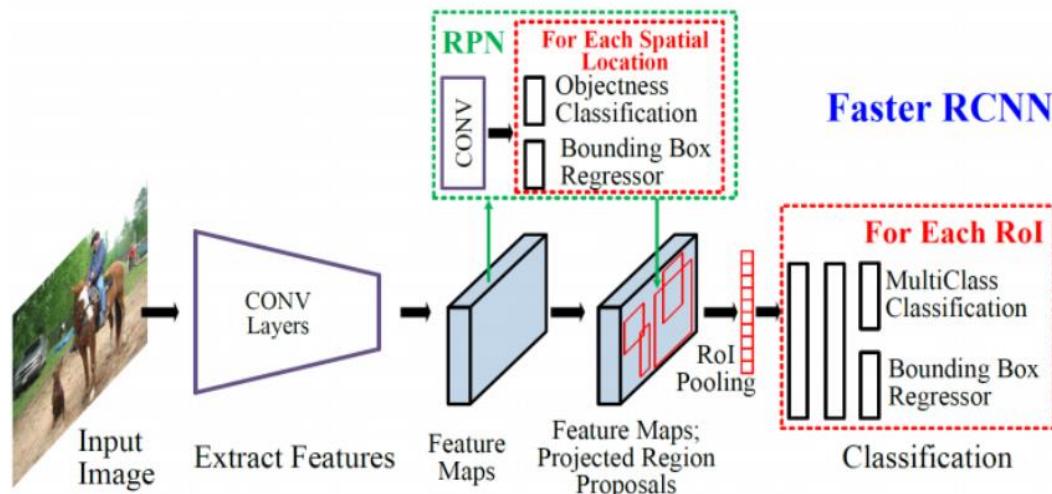
รูปที่ 2.7 ขั้นตอนการทำงานของ Fast R-CNN

(ที่มา: <https://arxiv.org/pdf/1809.02165.pdf>)

จากรูป 2.7 มีขั้นตอนการทำงานคือ จะใช้การคำนวณ CNN ก่อนหลังจากนั้นจะใช้วิธี Selective search เพื่อเลือกบริเวณที่คิดว่า “น่าจะเป็น” วัตถุและเรียกส่วนที่ได้จาก Selective search ว่า ROI และหลังจากนั้นทำการนำแต่ละ ROI ที่ได้จะไปรวมกันเรียกว่า ROI pooling layer ซึ่งจะไป Map กับ ROI feature vector โดยใช้ fully connected layers (FCs) ทำหน้าที่ในการเรียนรู้และจำแนกประเภทของวัตถุโดยผลลัพธ์ที่ได้จะออกมากเป็น 2 แบบ คือ MultiClass Classification (softmax) เป็นส่วนของพื้นหลังของภาพวัตถุ และ Bounding Box Regressor คือกล่องที่บอกรอบเขตของ Object ที่สนใจ

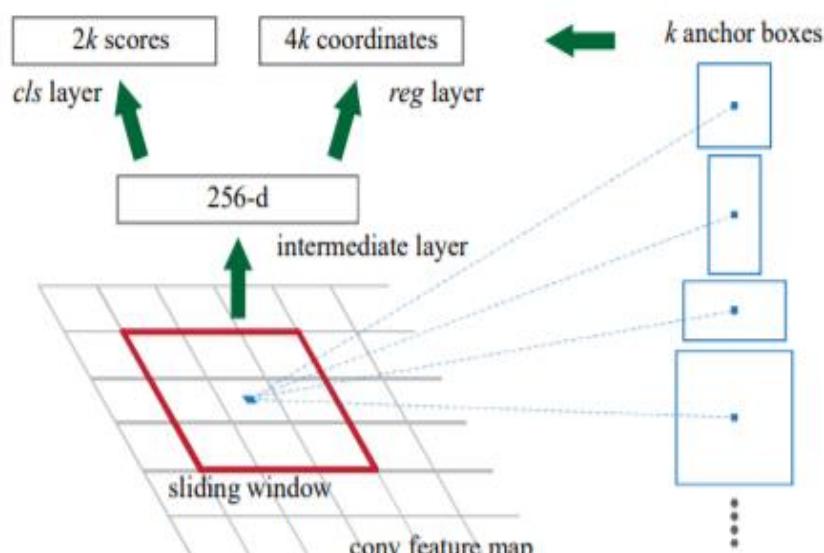
## 3. Faster R-CNN [14]

การทำงานของ Fast R - CNN นั้นพบข้อเสียหนึ่งอย่างคือเกิดอาการครอบคลุมในการ map ROI จึงทำให้เกิด Faster R-CNN โดยมีการนำ Region Proposal Network (RPN) [15] โดยการทำงานของ RPN คือการสร้าง Anchor box โดย Anchor box นั้นจะเป็นกรอบที่จำนวน  $\text{Width} \times \text{Height} \times 9$  Anchors โดย RPN จะแสดงความน่าจะเป็นว่าตำแหน่งและขนาดของ Anchor box นั้น เป็นกรอบของวัตถุที่ต้องการหรือไม่ ดังแสดงตามรูปที่ 2.9 โดยจะเป็นการทำงานบน GPU (Graphics Processing Unit) หลังจากนั้นนำเอา Feature Maps และ Project Region Proposals มาทำการ RoI Pooling เพื่อบอกผลลัพธ์ว่า บริเวณใดที่ของภาพที่มีวัตถุอยู่ ดังแสดงตามรูปที่ 2.8



รูปที่ 2.8 ขั้นตอนการทำงานของ Faster R-CNN

(ที่มา: <https://arxiv.org/pdf/1809.02165.pdf>)



รูปที่ 2.9 การทำงานของ RPN

|   | <b>R-CNN</b> | <b>Fast R-CNN</b> | <b>Faster R-CNN</b> |
|---|--------------|-------------------|---------------------|
| Test time per image<br>(with proposals) | 50 seconds   | 2 seconds         | <b>0.2 seconds</b>  |
| (Speedup)                               | 1x           | 25x               | <b>250x</b>         |
| mAP (VOC 2007)                          | 66.0         | <b>66.9</b>       | <b>66.9</b>         |

รูปที่ 2.10 เปรียบเทียบความเร็วของ R-CNN, Fast R-CNN, Faster R-CNN

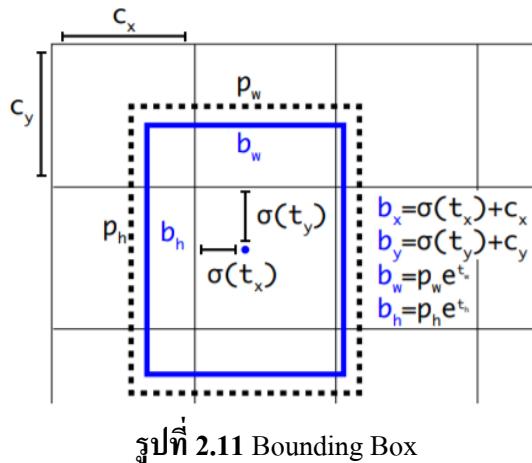
(ที่มา: <https://dzone.com/articles/from-r-cnn-to-faster-r-cnn-the-evolution-of-object>)

จากรูปที่ 2.10 จะเห็นได้ว่า Faster R-CNN เร็วกว่า R-CNN มาก จึงสามารถใช้สำหรับ ทำ Object Detection แบบ Realtime ได้ และมีความเร็วถึง 0.2 วินาที แต่ยังมีวิธีที่ดีกว่า Faster R-CNN คือ YOLO ในปัจจุบันคือ Version 3 มีความรวดเร็วและแม่นยำมาก

#### 2.1.4 YOLO (You Only Look Once) Version 3 [16]

เป็นวิธีในการทำ Object Detection ได้เร็วกว่า R-CNN ซึ่งสาเหตุที่ R-CNN ข้างต้นคือเป็นเพราะว่าแต่ละส่วนของ RoI ยังมีการ Train แยกกันอยู่ แต่การใช้ YOLO จะทำนายได้โดยว่าอะไรคือ Object และอยู่ตรงไหนของรูปภาพ หลักการทำงานของ YOLO

1. Bounding Box Prediction คือกรอบที่ล้อมรอบ Object ที่ได้ทำการ Detect ประกอบไปด้วย 4 ค่า คือ  $b_x$  = ตำแหน่งของ x ,  $b_y$  = ตำแหน่งของ y ,  $b_w$  = ความกว้างของกรอบ ,  $b_h$  = ความยาวของกรอบ ดังแสดงตามรูปที่ 2.11

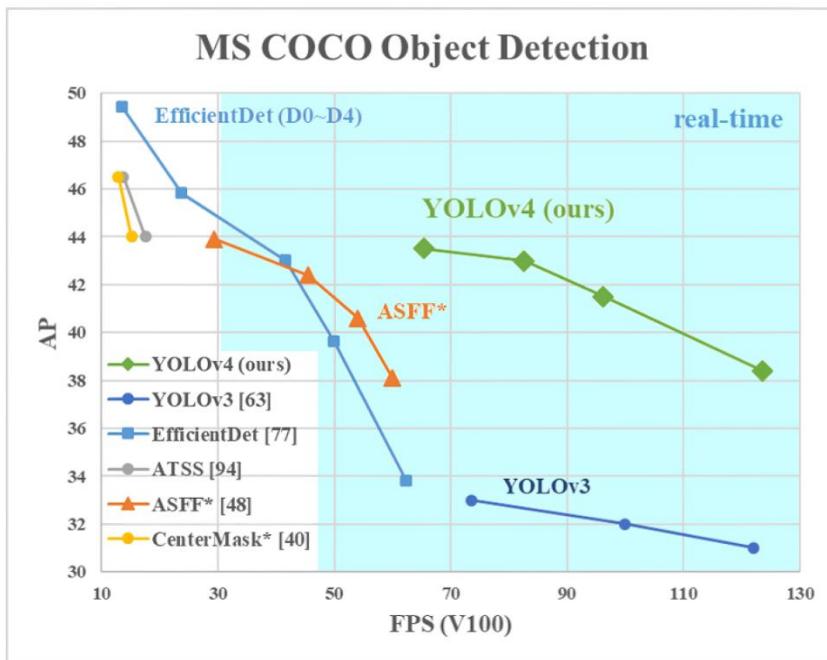


2. Class Prediction คือ ในช่วงแรกมีการใช้ Softmax function แต่การทำวิธีนี้ไม่สามารถแยก Label ที่ซ้อนทับกันได้ ทำให้มีการใช้ Logistic Classifiers ขึ้นมา โดยวิธีนี้จะสามารถทำการ Detect ภาพที่มีการซ้อนทับกันของ Label ได้ดี
3. YOLO Version 3 มีการนำ Feature Extraction ที่มีประสิทธิภาพมากขึ้น โดยใช้ Neural Network ที่มีความลึกมากกว่าเดิมคือ Darknet53 ที่มี 53 Convolutional layers

#### 2.1.5 YOLO (You Only Look Once) Version 4 [17]

ในช่วงเดือนพฤษภาคม ปี ค.ศ.2020 ได้มีการเปิดตัว YOLO Version 4 ที่สามารถ tren ข้อมูลบน GPU เครื่องเดียวที่มีขนาด Mini-batch size ที่เล็กลง เป็นการคำนวณแบบคู่ขนาน ซึ่งถ้า

Mini-batch size มีขนาดที่ใหญ่เกินไป อาจจะทำให้ข้อมูลมีขนาดใหญ่กว่า Memory ของเครื่องที่จะรับได้ ซึ่งทำให้การ เทคนข้อมูลมีความรวดเร็วและแม่นยำขึ้น และมีการปรับปรุงจำนวน Convolution layer ซึ่งใน YOLO Version 3 มี 74 Convolutional layers แต่ใน YOLO Version 4 ได้เพิ่มความลึกมาเป็น 137 Convolutional layers ซึ่งทำให้มีความแม่นยำที่เพิ่มขึ้นเมื่อทำการทดสอบ



รูปที่ 2.12 แสดงการเปรียบเทียบ YOLO v4 ,YOLO v3 และ EfficientDet

จากรูปที่ 2.12 เห็นได้ว่า YOLO Version 4 ทำงานได้เร็วขึ้น 2 เท่าเมื่อเทียบกับ EfficientDet แต่ประสิทธิภาพยังเทียบเท่ากัน และปรับปรุงค่า AP (Average Precision) ที่สูงขึ้นกว่า YOLO Version 3 ค่อนข้างมาก

ผู้จัดทำได้เลือกใช้ YOLO Version 4 พร้อมกับมีไฟล์ Weight ที่ทางผู้สร้างได้ทำการ Train จาก COCO Dataset [18] จนได้ค่า Average Loss ที่ต่ำมาก โดยมีจำนวน Class มากถึง 80 Class ที่สามารถ Detect ได้ และ YOLO สามารถ Train และ Test โดยใช้ Custom Dataset เพื่อใช้สำหรับ Detect รูปภาพที่ต้องการ ได้ สามารถ Set จำนวนรอบในการ Train และจำนวนรูปภาพในการ Train ในแต่ละรอบ ได้ รองรับการ Detect ได้ทั้ง Webcam (Realtime), Picture และ Video โดยระยะเวลาในการ Train ขึ้นอยู่กับ Hardware , GPU และจำนวน Class ยิ่งมีมากทำให้รูปภาพในการ Train ยิ่งเยอะ ดังผลให้เวลาในการ Train มีมากขึ้น ความแม่นยำในการ Detect ขึ้นอยู่กับจำนวน รูปภาพในแต่ละ Class และการกำหนดจำนวนรูปที่นำเข้า Train ในแต่ละรอบ ยิ่งจำนวนรูปเยอะยิ่ง ทำให้มีความแม่นยำ

## 2.2 เทคโนโลยีที่เกี่ยวข้อง

### 2.2.1 Python [19]



รูปที่ 2.13 Python

(ที่มา: [https://en.wikipedia.org/wiki/Python\\_\(programming\\_language\)](https://en.wikipedia.org/wiki/Python_(programming_language)))

ภาษาโปรแกรม Python คือภาษาโปรแกรมคอมพิวเตอร์ระดับสูง เป็นภาษายอดนิยม ถูกสร้างขึ้นโดย Guido van Rossum และมีการเปิดตัวในปี 1991 เป็นภาษาที่มีความน่าสนใจอย่างมากในการพัฒนาแอปพลิเคชันอย่างรวดเร็ว

Python เป็นภาษาที่ค่อนข้างง่ายในการเข้าใจและการเขียน เนื่องจากมีไวยากรณ์ที่ง่าย ทำให้สามารถสร้างแอปพลิเคชันที่สวยงามหรือทำการวิเคราะห์ข้อมูลได้ เช่น Classification , Clustering , Regression เป็นต้น ภาษา Python มีการทำงานแบบ Interpreter คือเป็นการแปลงคำสั่งทีละบรรทัด ซึ่งทำให้สามารถแก้ไขผิดพลาดได้ง่ายและมีประสิทธิภาพมากขึ้น

### 2.2.2 Pygame [20]

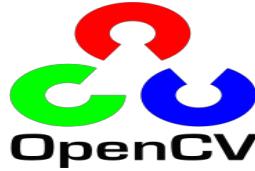


รูปที่ 2.14 Pygame

(ที่มา: <https://en.wikipedia.org/wiki/Pygame>)

Pygame เป็น Library หนึ่งของภาษา Python ที่ออกแบบมาสำหรับการเขียนวิดีโอเกม มีคำสั่งและ Function ต่างๆ ที่สามารถนำไปพัฒนาเกม ตั้งแต่เกม Tetris ซึ่งเป็นเกมแก้ปัญหาจัดเรียง ตัวบล็อกที่หล่นลงมาแล้วจัดเรียงให้เป็นแท่ง ไปจนถึงเกม Civilization 4 หรือ 3D Shooters โดย Pygame สามารถจัดการกับกราฟิก 2 มิติ, เพลง, การตรวจสอบการชน, กราฟิก 3 มิติ (ร่วมกับ OpenGL), จอยสติ๊ก, ภาพยนตร์ เป็นต้น

### 2.2.3 OpenCV [21]



รูปที่ 2.15 OpenCV

(ที่มา: <https://en.m.wikipedia.org/wiki/OpenCV>)

OpenCV (Open Source Computer Vision) ถูกเขียนขึ้นด้วยภาษา C++ โดยมีการรองรับภาษา Python, Java และ MATLAB เป็น Library ฟังก์ชันการเขียนโปรแกรม ส่วนใหญ่จะเน้นไปที่การแสดงผลด้วยคอมพิวเตอร์แบบเรียลไทม์และ OpenCV ยังมีการสนับสนุนเฟรมเวิร์กการเรียนรู้เชิงลึก (Deep Learning) ซึ่งได้แก่ Tensorflow, Pytorch และ Caffe

### 2.2.4 Tensorflow [21]



รูปที่ 2.16 Tensorflow

Tensorflow เป็น Library ที่ใช้ในการพัฒนาด้าน Machine Learning เป็นของ บริษัท Google มีการท างานโดยใช้ภาษา Python รองรับการทำงานทั้งบน CPU และ GPU รองรับระบบปฏิบัติการ Windows, macOS, Linux และ Andriod

### 2.2.5 Web Camera (Webcam)

เป็นอุปกรณ์ที่ใช้สำหรับถ่ายภาพหรือจับภาพเคลื่อนไหวพร้อมกับแสดงผลบนคอมพิวเตอร์ โดยทางผู้จัดทำได้นำมาต่อเข้ากับคอมพิวเตอร์เพื่อทำการถ่ายรูปภาพจิ๊กซอว์และนำภาพไปประมวล

### 2.2.6 Darknet [23]



รูปที่ 2.17 Darknet  
(ที่มา: <https://pjreddie.com/darknet>)

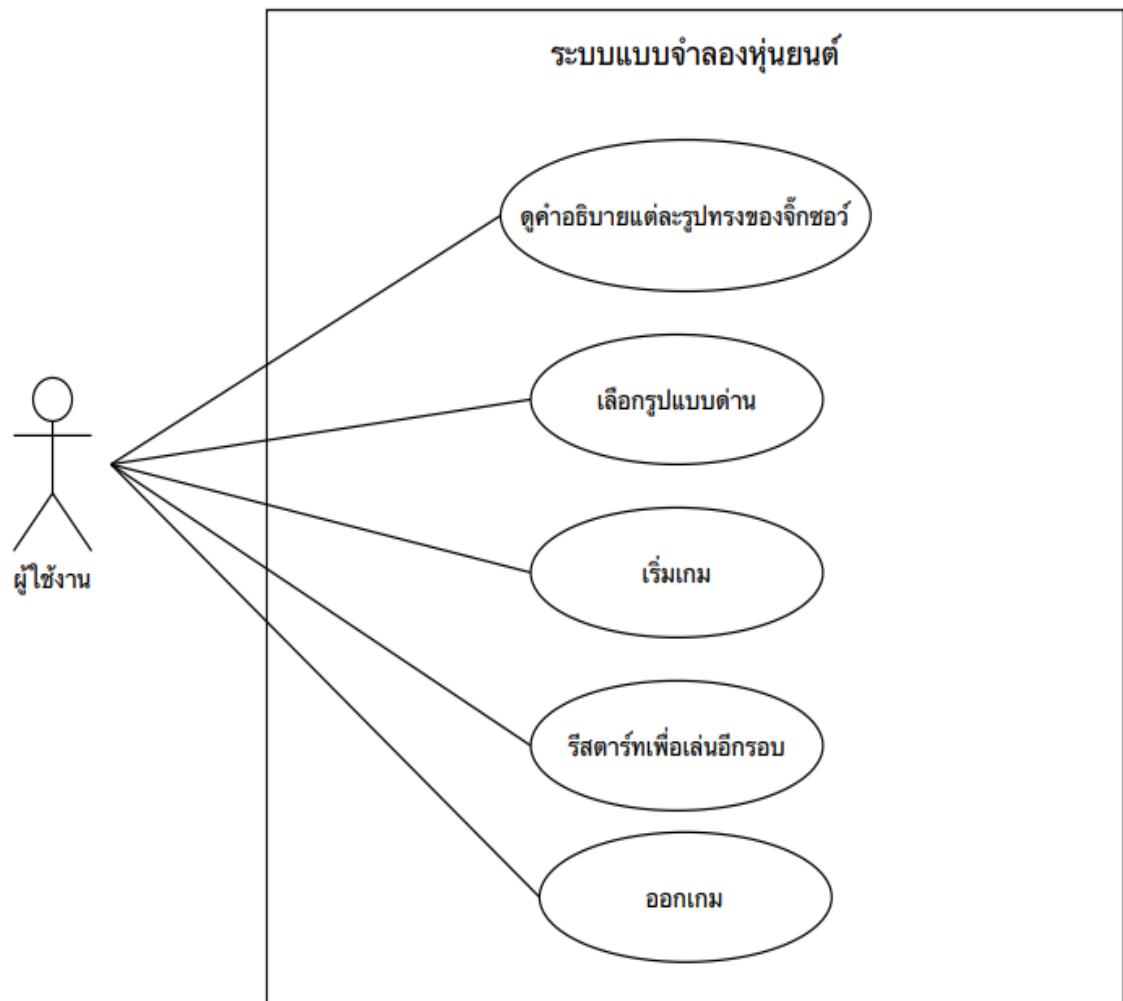
Darknet เป็น Neural Network Framework แบบ Open source ที่เขียนด้วยภาษา C และ Cuda (Compute Unified Device Architecture) คือ แพลตฟอร์มสำหรับการประมวลผลแบบคุ่นนาน พัฒนาโดยบริษัท Nvidia โดย Darknet สามารถใช้งานได้ทั้งบน CPU และ GPU ทำให้มีการประมวลผลที่รวดเร็ว โดยทางผู้จัดทำ Darknet มาใช้กับ YOLO เพื่อทำการ Train และ Detect รูปภาพที่ได้ทำการเตรียมไว้

## บทที่ 3

# การออกแบบและการพัฒนา

### 3.1 การออกแบบแพนภาคูสเกส

ทางผู้จัดทำได้ทำการออกแบบระบบโดยใช้ Use case diagram เพื่ออธิบายหลักการทำงานของระบบว่ามีการทำงานอย่างไร มีความเกี่ยวข้องกันในส่วนใด ดังนี้



รูปที่ 3.1 ภาพ Use case diagram ของระบบแบบจำลองหุ่นยนต์

จากรูปที่ 3.1 และดงยูสเคสต่าง ๆ ของระบบ ซึ่งจะอธิบายรายละเอียดยูสเคสเพิ่มเติมได้ดังตารางที่ 3.1 – 3.5

### ตารางที่ 3.1 รายละเอียดยูสเคสในการคุยกับชีบายแต่ละรูปทรงของจิ๊กซอว์

|                                     |   |  |
|-------------------------------------|---|--|
| ชื่อยูสเคส                          | ดูคำอธิบายแต่ละรูปทรงของจิ๊กซอว์  |  |
| หมายเลยยูสเคส                       | 1   |  |
| เหตุการณ์                           | ผู้ใช้อ่านคำอธิบายคำสั่งโปรแกรมมิ่งที่แทนด้วยจิ๊กซอว์แต่ละทรงและหน้าที่ของ Object ต่าง ๆ ที่แสดงในด้าน                                      |  |
| เหตุการณ์ที่นำໄไปสู่การใช้งานยูสเคส | ผู้ใช้รับรู้ถึงคำสั่งโปรแกรมมิ่ง สามารถนำໄไปต่อจิ๊กซอว์ เพื่อเก็บปัญหาและเข้าใจความหมายของ Object ในด้าน                                    |  |
| รายละเอียดโดยย่อ                    | บอกให้ผู้ที่เพิ่งเข้ามาใช้งาน ได้รู้ถึงรูปทรงของจิ๊กซอว์แต่ละรูปทรงว่าให้แทนด้วยคำสั่งโปรแกรมมิ่งอะไรหน้าที่ของ Object ต่าง ๆ ที่แสดงในด้าน |  |
| ผู้ที่เกี่ยวข้อง                    | ผู้ใช้งานระบบ   |  |
| ยูสเคสที่เกี่ยวข้อง                 | -   |  |
| ผู้ที่มีส่วนเกี่ยวข้อง              | -   |  |
| เงื่อนไขที่ต้องทำก่อน               | -   |  |
| เงื่อนไขหลังจากทำยูสเคส             | -   |  |
| หลักการทำงาน                        | ผู้ใช้งาน   | ระบบ   |
|                                     | 1. ผู้ใช้งาน กดเข้าสู่หน้า อธิบายรูปทรงจิ๊กซอว์ และตัวอย่างการต่อจิ๊กซอว์   | 2. แสดงรูปภาพอธิบายรูปทรง จิ๊กซอว์แต่ละทรง เพื่อบอกถึง การแทนคำสั่งโปรแกรมมิ่งและ หน้าที่ของ Object ที่แสดงใน ด้านต่าง ๆ |
| เงื่อนไขที่ถูกยกเว้น                | -   |  |

**ตารางที่ 3.2 รายละเอียดยูสเคสเดือกรูปแบบค่า**

| ชื่อยูสเคส                          | เดือกรูปแบบค่า   |           |      |                                  |  |
|-------------------------------------|--|-----------|------|----------------------------------|--|
| หมายเลยยูสเคส                       | 2  |           |      |                                  |  |
| เหตุการณ์                           | ผู้ใช้เลือกรูปแบบค่าที่ต้องการ   |           |      |                                  |  |
| เหตุการณ์ที่นำໄไปสู่การใช้งานยูสเคส | ผู้ใช้ต้องการเลือกรูปแบบค่าตามความต้องการ  |           |      |                                  |  |
| รายละเอียดโดยย่อ                    | ผู้ใช้สามารถเลือกค่า โดยในแต่ละค่าจะมีรูปแบบความยากและง่ายแตกต่างกัน   |           |      |                                  |  |
| ผู้ที่เกี่ยวข้อง                    | ผู้ใช้งานระบบ  |           |      |                                  |  |
| ยูสเคสที่เกี่ยวข้อง                 | -  |           |      |                                  |  |
| ผู้ที่มีส่วนเกี่ยวข้อง              | -  |           |      |                                  |  |
| เงื่อนไขที่ต้องทำก่อน               | -  |           |      |                                  |  |
| เงื่อนไขหลังจากทำยูสเคส             | -  |           |      |                                  |  |
| หลักการทำงาน                        | <table border="1"> <thead> <tr> <th>ผู้ใช้งาน</th> <th>ระบบ</th> </tr> </thead> <tbody> <tr> <td>1. ผู้ใช้กดเลือกค่าตามที่ต้องการ</td> <td>2. แสดงค่าตามที่ผู้ใช้งานได้กดเลือก และ โชว์ชื่อของค่าที่เลือกกำกับไว้</td> </tr> </tbody> </table> | ผู้ใช้งาน | ระบบ | 1. ผู้ใช้กดเลือกค่าตามที่ต้องการ | 2. แสดงค่าตามที่ผู้ใช้งานได้กดเลือก และ โชว์ชื่อของค่าที่เลือกกำกับไว้ |
| ผู้ใช้งาน                           | ระบบ   |           |      |                                  |  |
| 1. ผู้ใช้กดเลือกค่าตามที่ต้องการ    | 2. แสดงค่าตามที่ผู้ใช้งานได้กดเลือก และ โชว์ชื่อของค่าที่เลือกกำกับไว้   |           |      |                                  |  |
| เงื่อนไขที่ถูกยกเว้น                | -  |           |      |                                  |  |

### ตารางที่ 3.3 รายละเอียดคุณลักษณะเริ่มเกม

|  |  |   |
|--|--|---|
| ชื่อคุณลักษณะ                          | เริ่มเกม   |   |
| หมายเลขอุปกรณ์                         | 3  |   |
| เหตุการณ์                              | ผู้ใช้ทำการเริ่มเล่นเกม  |   |
| เหตุการณ์ที่นำໄไปสู่การใช้งานคุณลักษณะ | ผู้ใช้ต้องการที่จะเล่นเกมเพื่อที่แก้ไขปัญหาจากโจทย์  |   |
| รายละเอียดโดยย่อ                       | เมื่อผู้ใช้ทำการต่อจิกซอว์เสร็จ กดปุ่ม Start ระบบจะทำการถ่ายรูปภาพจิกซอว์จากกล้องเว็บแคม จากนั้นหุ่นยนต์จะเคลื่อนที่ตามคำสั่งโปรแกรมมิ่งที่ได้จากการต่อจิกซอว์โดยผู้ใช้งาน |   |
| ผู้ที่เกี่ยวข้อง                       | ผู้ใช้งานระบบ  |   |
| คุณลักษณะที่เกี่ยวข้อง                 | -  |   |
| ผู้ที่มีส่วนเกี่ยวข้อง                 | -  |   |
| เงื่อนไขที่ต้องทำก่อน                  | 1. ดูคำอธิบายแต่ละรูปทรงของจิกซอว์ และตัวอย่างการต่อจิกซอว์ในกรณีที่ผู้ใช้งานไม่ทราบคำสั่งในรูปทรงจิกซอว์<br>2. เลือกรูปแบบค่า   |   |
| เงื่อนไขหลังจากทำคุณลักษณะ             | -  |   |
| หลักการทำงาน                           | ผู้ใช้งาน  | ระบบ  |
|  | 1. ผู้ใช้ทำการต่อจิกซอว์ และกดปุ่ม Start เพื่อเริ่มเกม   | 2. ถ่ายรูปภาพจิกซอว์ที่ได้ทำการต่อและแปลงเป็นคำสั่งให้วางในระบบ จากนั้นหุ่นยนต์จะเคลื่อนที่ตามคำสั่งโปรแกรมมิ่งที่ได้จากการต่อจิกซอว์โดยผู้ใช้งาน |
| เงื่อนไขที่ถูกยกเว้น                   | -  |   |

### ตารางที่ 3.4 รายละเอียดัญญาสเคต์ฟิล์มสตาร์ท

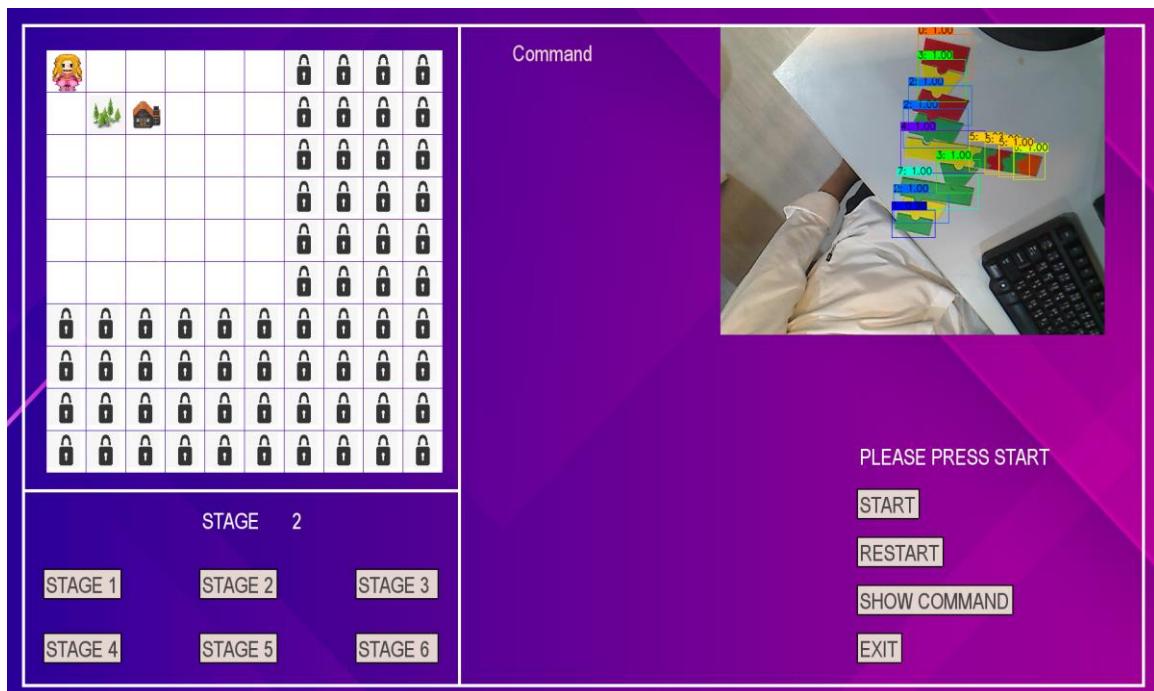
|  |  |   |
|--|--|---|
| ชื่อัญญาสเคต์                          | ฟิล์มสตาร์ท  |   |
| หมายเหยียดัญญาสเคต์                    | 4  |   |
| เหตุการณ์                              | ผู้ใช้ทำการรีสตาร์ทเพื่อเคลียร์คำสั่ง และรีเซ็ตค่าต้นใหม่ หุ่นยนต์กลับไปยังจุดเริ่มต้น เพื่อทำการเริ่มเล่นอีกครั้ง                   |   |
| เหตุการณ์ที่นำໄไปสู่การใช้งานัญญาสเคต์ | ผู้ใช้ต้องการที่จะเคลียร์คำสั่งและรีเซ็ตค่าต้นเพื่อเริ่มเล่นอีกครั้ง   |   |
| รายละเอียดโดยย่อ                       | หากผู้ใช้งานต้องการลบคำสั่งที่ได้แปลงออกมายัง ต้องการรีเซ็ตค่าต้นไปยังจุดเริ่มต้น สามารถกดปุ่ม Restart เพื่อทำการเริ่มเล่นใหม่อีกรอบ |   |
| ผู้ที่เกี่ยวข้อง                       | ผู้ใช้งานระบบ  |   |
| ัญญาสเคต์ที่เกี่ยวข้อง                 | -  |   |
| ผู้ที่มีส่วนเกี่ยวข้อง                 | -  |   |
| เงื่อนไขที่ต้องทำก่อน                  | -  |   |
| เงื่อนไขหลังจากทำัญญาสเคต์             | เริ่มเกม   |   |
| หลักการทำงาน                           | ผู้ใช้งาน  | ระบบ  |
|  | 1. กดปุ่ม Restart เพื่อทำการเคลียร์คำสั่งที่แปลงออกมายัง   | 2. ระบบจะเปลี่ยนจากคำสั่งเก่าที่แสดงไว้เปลี่ยนเป็นคำว่า Pleased Start หมายถึง ไม่มีคำสั่งให้แสดงผล สามารถเดินเริ่มเกม และหุ่นยนต์กลับไปสู่จุดเริ่มต้นของค่านั้น |
| เงื่อนไขที่อยู่ยกเว้น                  | -  |   |

### ตารางที่ 3.5 รายละเอียดยูสเคสออกแบบ

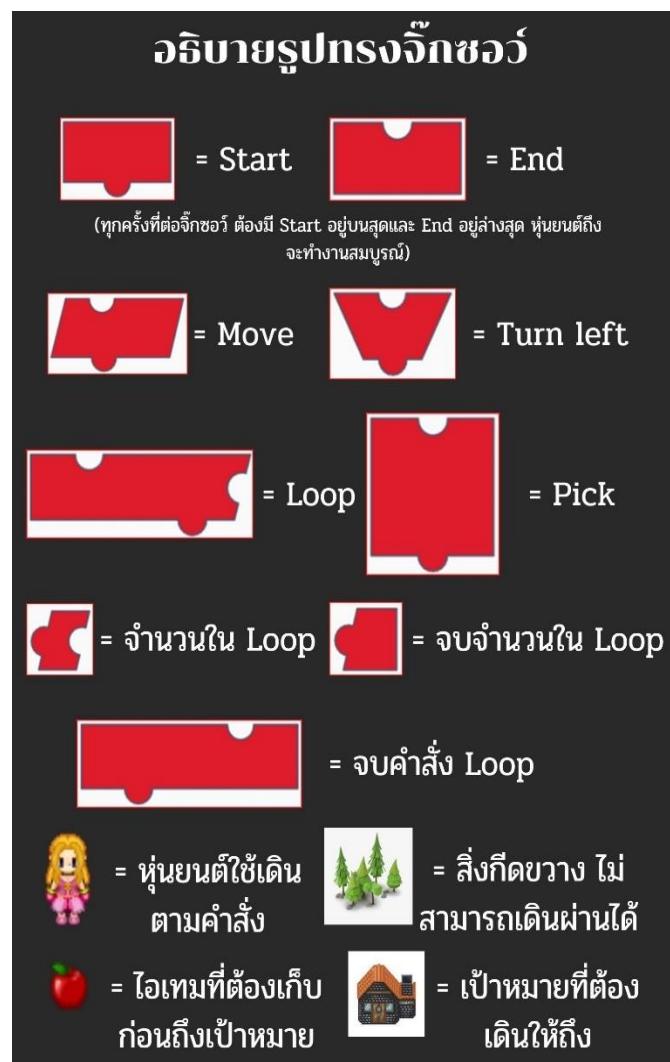
|   |   |  |
|---|---|--|
| <b>ชื่อยูสเคส</b>                         | ออกแบบ  |  |
| <b>หมายเลขอุสเคส</b>                      | 5   |  |
| <b>เหตุการณ์</b>                          | ผู้ใช้ทำการออกจากเครื่องระบบ เป็นการสิ้นสุดการทำงาน                           |  |
| <b>เหตุการณ์ที่นำไปสู่การใช้งานยูสเคส</b> | ผู้ใช้ต้องการที่จะออกจากเครื่องระบบ   |  |
| <b>รายละเอียดโดยย่อ</b>                   | หากผู้ใช้ต้องการที่จะออกในหน้าจอของระบบ สามารถกดปุ่ม Exit เพื่อเป็นการปิดระบบ |  |
| <b>ผู้ที่เกี่ยวข้อง</b>                   | ผู้ใช้งานระบบ   |  |
| <b>ยูสเคสที่เกี่ยวข้อง</b>                | -   |  |
| <b>ผู้ที่มีส่วนเกี่ยวข้อง</b>             | -   |  |
| <b>เงื่อนไขที่ต้องทำก่อน</b>              | -   |  |
| <b>เงื่อนไขหลังจากทำยูสเคส</b>            | เริ่มเกม  |  |
| <b>หลักการทำงาน</b>                       | <b>ผู้ใช้งาน</b>  | <b>ระบบ</b>                                  |
|   | 1. กดปุ่ม Exit เพื่อทำการออกจากระบบ<br>แบบจำลองหุ่นยนต์                       | 2. ระบบจะทำการปิดหน้าต่าง<br>ของตัวระบบทันที |
| <b>เงื่อนไขที่ถูกยกเว้น</b>               | -   |  |

### 3.2 ออกแบบส่วนของระบบ

- ผู้จัดทำได้ทำการออกแบบส่วนของระบบเปียนด้วยภาษา Python ดังแสดงตามรูปที่ 3.2 โดยมีหน้าจอแสดงผลของค่าที่มีตัว Robot , Objective และสิ่งกีดขวาง มีปุ่ม Stage คือทำการเลือกค่าต่าง ๆ ปุ่ม Show Command คือแสดงคำอธิบายรูปทรงจิ๊กซอว์ว่าแทนค่าด้วยคำสั่งโปรแกรมมิ่งอะไรและหน้าที่ของ Object ต่าง ๆ ที่แสดงในค่า ดังแสดงตามรูปที่ 3.3 ปุ่ม Start คือการกดเพื่อให้ Robot เดินตามคำสั่งจากรูปจิ๊กซอว์เพื่อให้ไปถึง Objective ปุ่ม Restart คือการเริ่มเล่นใหม่ ส่วนสุดท้ายคือการแสดงผล Command หรือคำสั่งที่ได้จากการตรวจจับจิ๊กซอว์จากรูปภาพและหน้าจอแสดงผลผ่านกล้องเว็บแคม โดยมีการตรวจจับผ่านกล้องแบบเรียลไทม์ซึ่งใช้ Tensorflow ในการทำให้คลอนระบบได้



รูปที่ 3.2 ระบบแบบจำลองหุ่นยนต์



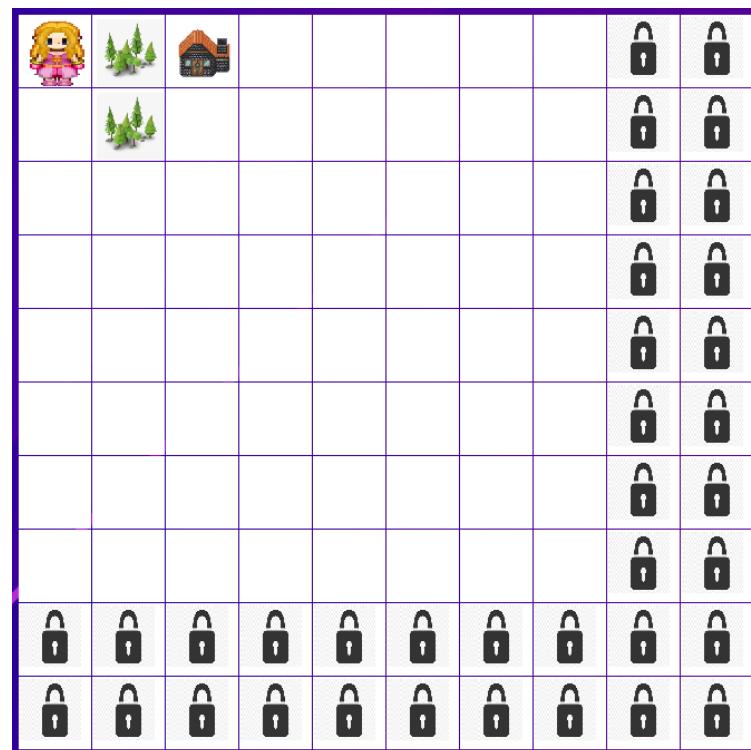
ຮູບທີ 3.3 ອົບຍາຍຮູບກຣອງຈິກຂອວ

- ສ່ວນຂອງການເລືອກດ່ານ ຜູ້ຈັດທ່າໄດ້ທ່ານການເພີ້ນຕົວແປຣແລະເຫັນຄ່າຂອງແຕ່ລະດ່ານເປັນໄຟລ໌  
ນາມສຸກຸລ JSON ຈາກນີ້ໃນສ່ວນຂອງຮະບນຈະທໍາການ Config ໄຟລ໌ JSON ຂອງດ່ານຕາມທີ່ເລືອກ  
ເພື່ອມາແສດງພລດັ່ງແສດງຕາມຮູບທີ່ 3.4 ແລະ ຕ້ວອຍ່າງຂອງດ່ານດັ່ງແສດງຕາມຮູບທີ່ 3.5

```

    "grid": [
        {
            "stage":4,
            "column":7,
            "row":7,
            "char_x":0,
            "char_y":0,
            "task_x":2,
            "task_y":0,
            "box_x": [1,1],
            "box_y": [0,1],
            "item" : true,
            "item_x" : 0,
            "item_y" : 0
        }
    ]
  
```

รูปที่ 3.4 ตัวอย่างไฟล์ JSON



รูปที่ 3.5 ตัวอย่างของค่าที่แสดงผลจากไฟล์ JSON

โดยมีการอธิบายแต่ละตัวแปร ดังแสดงตามตารางที่ 3.6

ตารางที่ 3.6 อธิบายตัวแปรของค่าในไฟล์ JSON

| ชื่อตัวแปร | Data type | คำอธิบาย  |
|------------|-----------|---|
| stage      | Integer   | ชื่อของค่า  |
| column     | Integer   | จำนวนคอลัมน์  |
| row        | Integer   | จำนวนแถว  |
| char_x     | Integer   | ตำแหน่งของ Robot ในแกน x  |
| char_y     | Integer   | ตำแหน่งของ Robot ในแกน y  |
| task_x     | Integer   | ตำแหน่งของเป้าหมายในแกน x   |
| task_y     | Integer   | ตำแหน่งของเป้าหมายในแกน y   |
| box_x      | List      | ตำแหน่งของสิ่งกีดขวางในแกน x                                      |
| box_y      | List      | ตำแหน่งของสิ่งกีดขวางในแกน y                                      |
| Item       | Boolean   | ใน Stage นี้มีสิ่งของให้หยอดหรือไม่ True คือ มี , False คือ ไม่มี |
| Item_x     | Integer   | ตำแหน่งของสิ่งของในแกน x  |
| Item_y     | Integer   | ตำแหน่งของสิ่งของในแกน y  |

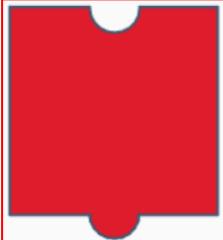
### 3.3 ออกแบบรูปภาพจิ๊กซอว์

ผู้จัดทำได้ทำการออกแบบจิ๊กซอว์ จำนวน 15 รูปทรงและใช้เครื่อง 3D Printer ปรินต์จิ๊กซอว์ ซึ่งในแต่ละรูปทรงจะมีคำสั่งในการใช้งานแตกต่างกัน ดังแสดงตามตารางที่ 3.7

ตารางที่ 3.7 แสดงรูปทรงและคำสั่งของจิ๊กซอว์

| รูปทรงจิ๊กซอว์  | Class | คำสั่ง |
|---|-------|--------|
|  | 0     | Start  |
|  | 1     | End    |

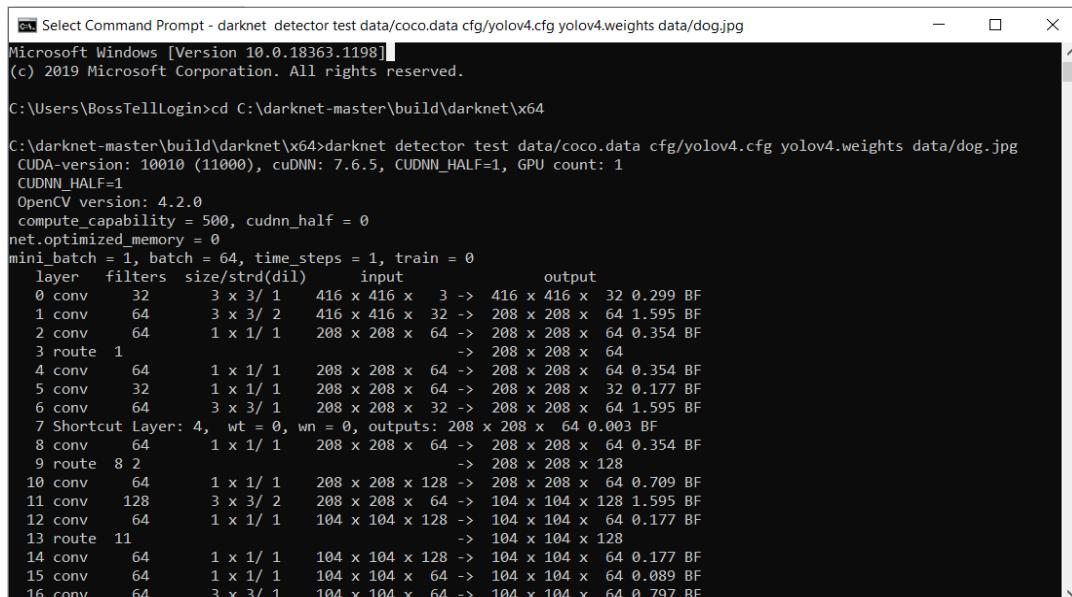
|   |    |                       |
|---|----|-----------------------|
|    | 2  | Move                  |
|    | 3  | Turn left             |
|    | 4  | Loop                  |
|    | 5  | จำนวนใน Loop          |
|    | 6  | จบจำนวนใน loop หรือ : |
|   | 7  | จบคำสั่ง Loop         |
|  | 8  | Command               |
|  | 9  | If                    |
|  | 10 | Condition ที่ 1       |
|  | 11 | Condition ที่ 2       |
|  | 12 | Condition ที่ 3       |

|   |    |      |
|---|----|------|
|  | 13 | Pick |
|  | 14 | Else |

### 3.4 ติดตั้ง YOLO (You only look once)

- ผู้จัดทำทำการติดตั้ง YOLO โดยตามวิธีการใน Github ของ AlexeyAB [24] ซึ่งได้ทำการติดตั้งบน Windows เพื่อทำการ Train และ Detect รูปภาพจีกซอร์ที่ได้ทำการถ่ายไว้และได้ทำการดาวน์โหลด yolov4.weights คือค่าน้ำหนักที่ได้จากการ Train ทั้งหมดของ YOLO Version 4 ที่มี 80 Class จากนั้นทำการเขียนคำสั่งเพื่อคุณติดตั้งสำเร็จและใช้งานได้ ดังแสดงตามรูปที่

3.6



```
darknet Select Command Prompt - darknet detector test data/coco.data cfg/yolov4.cfg yolov4.weights data/dog.jpg
Microsoft Windows [Version 10.0.18363.1198]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\BossTelLogin>cd C:\darknet-master\build\darknet\x64

C:\darknet-master\build\darknet\x64>darknet detector test data/coco.data cfg/yolov4.cfg yolov4.weights data/dog.jpg
CUDA-version: 10010 (11000), cuDNN: 7.6.5, CUDNN_HALF=1, GPU count: 1
CUDNN_HALF=1
OpenCV version: 4.2.0
compute_capability = 500, cudnn_half = 0
net.optimized_memory = 0
mini_batch = 1, batch = 64, time_steps = 1, train = 0
layer filters size/strd(dil) input output
0 conv 32 3 x 3 / 1 416 x 416 x 3 -> 416 x 416 x 32 0.299 BF
1 conv 64 3 x 3 / 2 416 x 416 x 32 -> 208 x 208 x 64 1.595 BF
2 conv 64 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 64 0.354 BF
3 route 1 -> 208 x 208 x 64
4 conv 64 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 64 0.354 BF
5 conv 32 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 32 0.177 BF
6 conv 64 3 x 3 / 1 208 x 208 x 32 -> 208 x 208 x 64 1.595 BF
7 Shortcut Layer: 4, wt = 0, wn = 0, outputs: 208 x 208 x 64 0.003 BF
8 conv 64 1 x 1 / 1 208 x 208 x 64 -> 208 x 208 x 64 0.354 BF
9 route 8 2 -> 208 x 208 x 128
10 conv 64 1 x 1 / 1 208 x 208 x 128 -> 208 x 208 x 64 0.709 BF
11 conv 128 3 x 3 / 2 208 x 208 x 64 -> 104 x 104 x 128 1.595 BF
12 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64 0.177 BF
13 route 11 -> 104 x 104 x 128
14 conv 64 1 x 1 / 1 104 x 104 x 128 -> 104 x 104 x 64 0.177 BF
15 conv 64 1 x 1 / 1 104 x 104 x 64 -> 104 x 104 x 64 0.089 BF
16 conv 64 3 x 3 / 1 104 x 104 x 64 -> 104 x 104 x 64 0.797 BF
```

รูปที่ 3.6 ตัวอย่างการใส่คำสั่งเพื่อ Detect รูปภาพ

- เมื่อทำการ Detect เสร็จ จะทำการแสดงค่า Accuracy ออกมา โดยนำยอดมาเป็น 4 Class คือ bicycle 99% , dog 99% , car 27% , truck 91% ดังแสดงตามรูปที่ 3.7

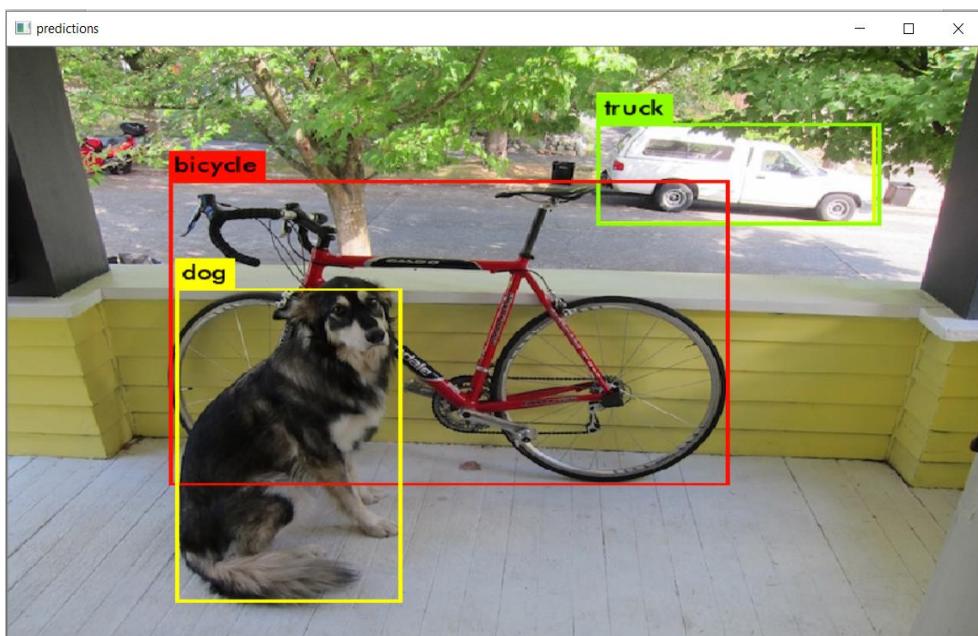
```

Select Command Prompt - darknet detector test data/coco.data cfg/yolov4.cfg yolov4.weights data/dog.jpg
Total BFLOPS 60.137
avg_outputs = 500162
Allocate additional workspace_size = 26.22 MB
Loading weights from yolov4.weights...
seen 64, trained: 32032 K-images (500 Kilo-batches_64)
Done! Loaded 162 layers from weights-file
data/dog.jpg: Predicted in 219.585000 milli-seconds.
bicycle: 99%
dog: 99%
car: 27%
truck: 91%

```

รูปที่ 3.7 แสดง Class และค่า Accuracy จากรูปภาพที่ Detect

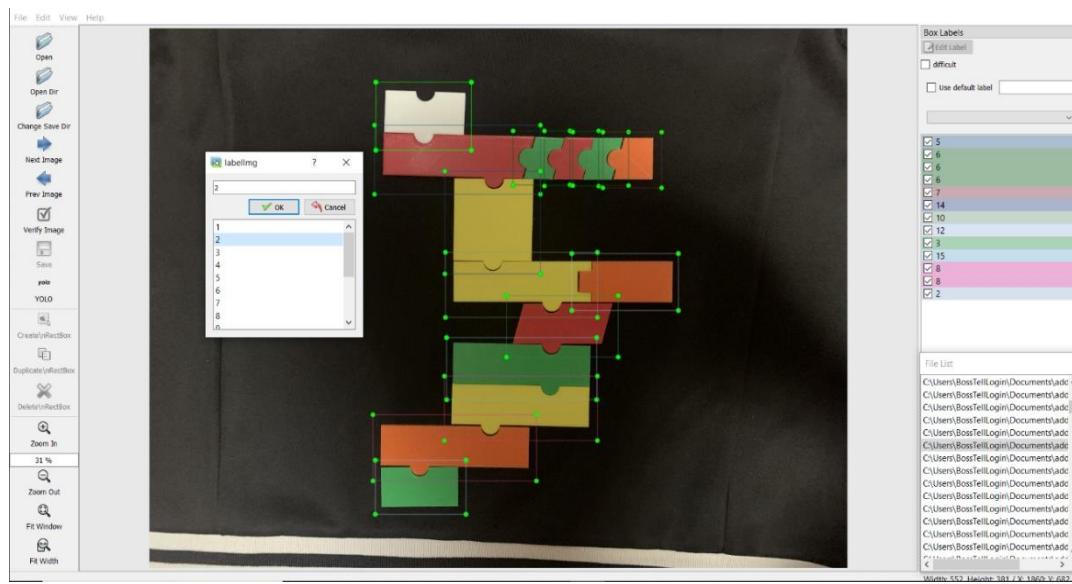
3. แสดงผลลัพธ์ที่ได้จากการ Detect โดยที่มี Bounding Box ล้อมรอบ Object ที่ตรวจจับเจอดัง แสดงตามรูปที่ 3.8



รูปที่ 3.8 แสดงรูปภาพผลลัพธ์จากการ Detect

### 3.5 สร้าง Label Image

- ผู้จัดทำการสร้าง Label Image เพื่อบอกตำแหน่งของ Object ที่อยู่ในรูปภาพ เพราะว่าในการนำรูปภาพมา Train ใน YOLO ต้องมีไฟล์ Text ที่บอกค่าของตำแหน่ง Object ของรูปภาพ โดยตามวิธีใน Github ของ tzutalin [25] จะนั่นทำการเปิด Directory ของรูปภาพจัดซอร์ที่ได้เตรียมไว้เพื่อทำการ Label Image ดังแสดงตามรูปที่ 3.9

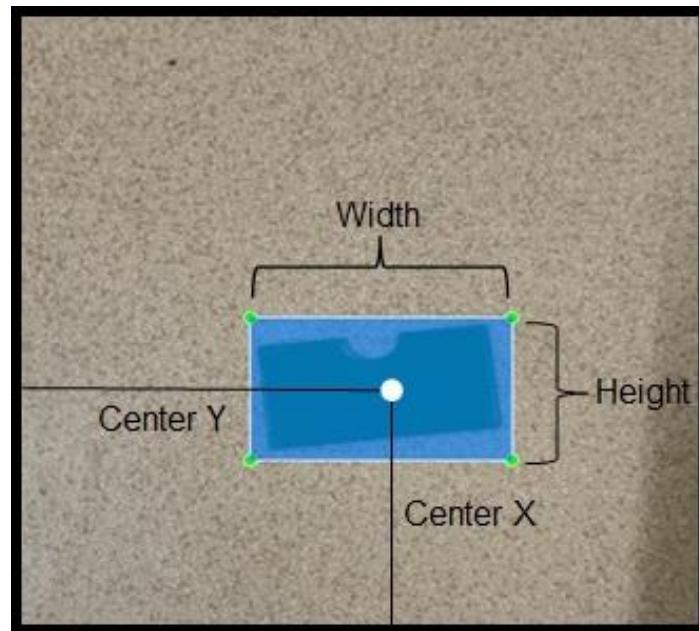


รูปที่ 3.9 การทำ Label Image

- เมื่อทำการสร้าง Label Image เสร็จแล้ว จะได้ไฟล์ Text ที่มีชื่อไฟล์เดียวกับชื่อของรูปภาพนั้น โดยในไฟล์จะแสดงอักษรที่มีค่า คือ Class ID , Center x , Center y , Width , Height ดังแสดงตามรูปที่ 3.10 และค่าทั้ง 5 ค่ามีข้อความดังแสดงตามรูปที่ 3.11

```
IMG_0546.txt - Notepad
File Edit Format View Help
4 0.441592 0.243552 0.237847 0.127976
5 0.561880 0.240575 0.081597 0.100198
5 0.604787 0.241567 0.077629 0.102183
5 0.646825 0.242063 0.079861 0.099206
6 0.692088 0.244213 0.084077 0.103505
13 0.492063 0.359954 0.136905 0.190146
9 0.533234 0.475694 0.217758 0.120701
11 0.681548 0.470403 0.152778 0.105489
2 0.591642 0.552414 0.159970 0.114087
14 0.533606 0.630952 0.215030 0.115079
7 0.532490 0.704530 0.219246 0.119378
7 0.437624 0.777612 0.234375 0.122685
1 0.388765 0.850694 0.130208 0.100198
```

รูปที่ 3.10 ไฟล์ Text ที่ได้หลังจากการทำ Label Image



รูปที่ 3.11 ความหมายของค่าใน Label

สามารถสรุปผลกระบวนการทดลอง Train และ Detect รูปภาพโดยใช้ YOLO ได้ดังแสดงตามรูปที่ 3.11



รูปที่ 3.12 ขั้นตอนการดำเนินการทดลอง Object detection โดยใช้ YOLO

## บทที่ 4

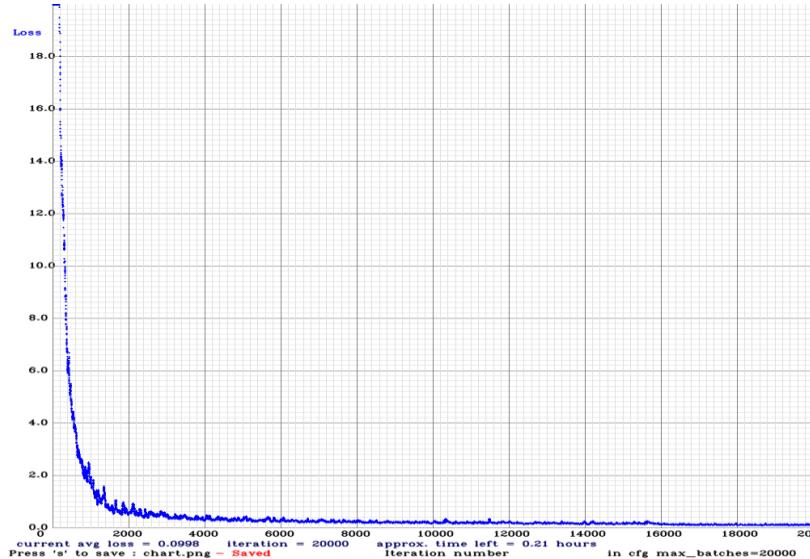
# ผลการดำเนินงานระบบแบบจำลองหุ่นยนต์

### 4.1 ผลการทดลองตรวจจับรูปภาพ

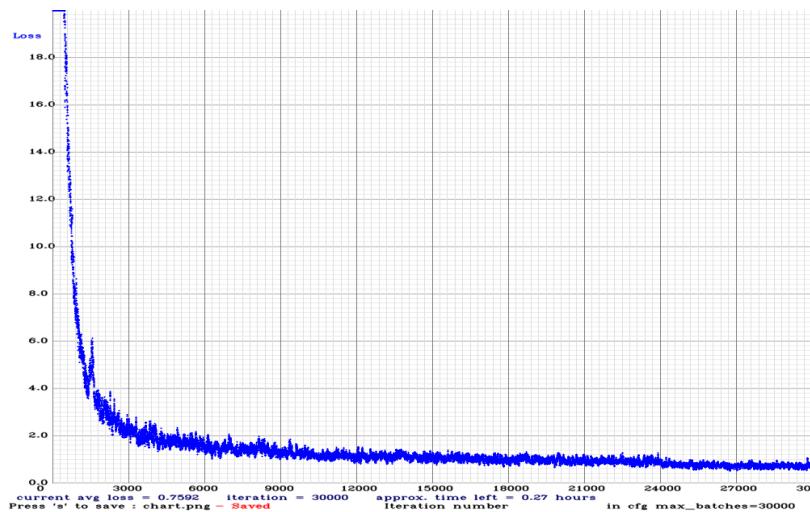
1. จากการที่ผู้จัดทำได้ทำการเทรนรูปภาพจีกซอว์โดยใช้ทั้ง YOLO v3 และ YOLO v4 โดยมีตัวแปรในการเทรนรูปภาพดังแสดงตามตารางที่ 4.1 โดยใน YOLO v3 ได้ทำการถ่ายรูปจีกซอว์โดยใช้กล้องจากโทรศัพท์ซึ่งมีความคมชัด แต่ในการใช้งานจริง ทางผู้จัดทำต้องใช้งานด้วยกล้องเว็บแคมซึ่งความคมชัดจะมีน้อยกว่ากล้องโทรศัพท์ ทำให้มีการเพิ่มรูปภาพที่ถ่ายด้วยกล้องเว็บแคมเข้าไปเทรนใน YOLO v4 แทน เมื่อเทรนสำเร็จจะได้กราฟที่แสดงค่า Average loss คือ กราฟที่บ่งบอกว่ามีความผิดพลาดมากน้อยแค่ไหนในการ Detect ยิ่งค่า Average loss น้อยแสดงว่าการ Detect มีความผิดพลาดน้อยหรือสามารถตรวจจับรูปภาพได้ดี ดังแสดงตามรูปที่ 4.1 และ 4.2 จะสังเกตได้ว่าในการเทรนรอบแรก ๆ ค่า Average loss ยังได้ผลลัพธ์ที่สูงมากเนื่องจากรูปภาพที่นำมาใช้ใน Train มีน้อย แต่เมื่อเพิ่มรอบของการเทรน ค่า Average loss จะค่อยลดลงเรื่อย ๆ ใน YOLO v4 ค่า Average loss สูงกว่า YOLO v3 แต่เมื่อนำมาใช้งานจริงค่อนข้างยืดหยุ่นกว่าและแม่นยำกว่า เพราะมี Dataset จำกัดลักษณะทั้ง 2 รูปแบบ และมีความลึกของ Layer ในการเทรนมากกว่า YOLO v3

ตารางที่ 4.1 อธิบายตัวแปรในการเทรนรูปภาพ

| ชื่อตัวแปร                           | จำนวนค่าที่ใช้ |
|--------------------------------------|----------------|
| จำนวนคลาส                            | 15 คลาส        |
| จำนวนรูปที่ใช้เทรนใน YOLO v3         | 2,658 รูป      |
| จำนวนรูปที่ใช้เทรนใน YOLO v4         | 3,398 รูป      |
| จำนวนรูปที่ใช้ทดสอบใน YOLO v3        | 266 รูป        |
| จำนวนรูปที่ใช้ทดสอบใน YOLO v4        | 340 รูป        |
| จำนวนรอบที่เทรนใน YOLO v3            | 20,000 รอบ     |
| จำนวนรอบที่เทรนใน YOLO v4            | 30,000 รอบ     |
| จำนวนรูปภาพที่สุ่มเข้าไปเทรนใน 1 รอบ | 64 รูป         |

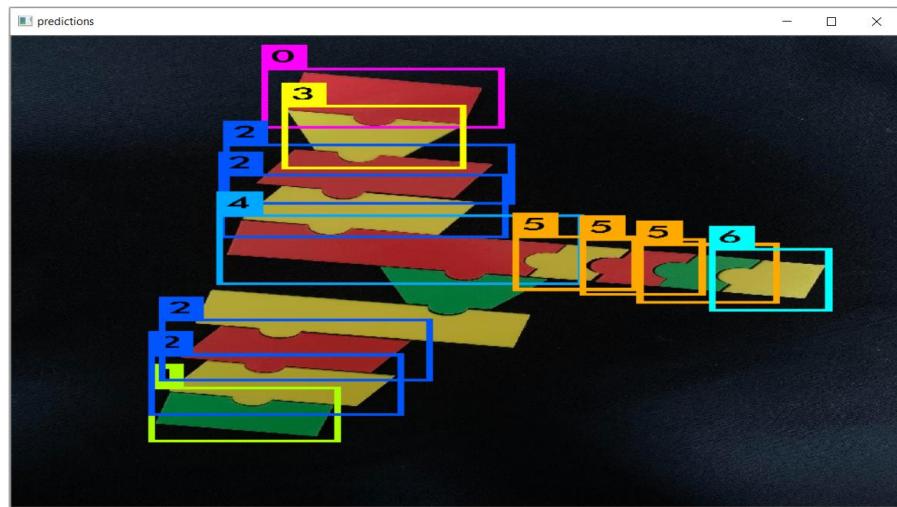


รูปที่ 4.1 กราฟแสดงค่า Average loss ในระหว่างการ Train ของ YOLO v3

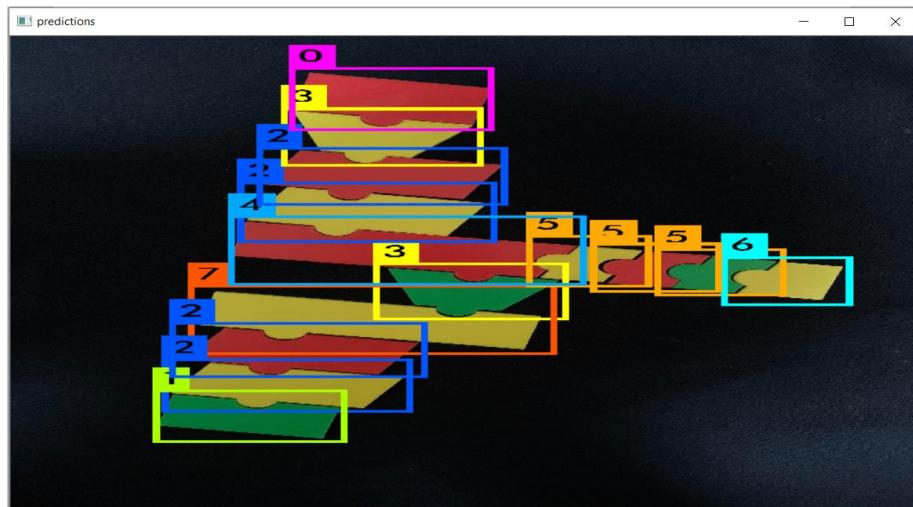


รูปที่ 4.2 กราฟแสดงค่า Average loss ในระหว่างการ Train ของ YOLO v4

2. ทางผู้จัดทำได้นำ Weight ที่ได้ทำการเทรนใน YOLO v3 และ YOLO v4 มาทำการเปรียบเทียบผลลัพธ์ โดยใช้รูปภาพเดียวกัน ดังแสดงตามรูปที่ 4.3 และ 4.4 เพื่อให้เห็นว่า YOLO v3 มีจิ๊กซอว์ 2 ชิ้นที่ตรวจจับไม่เจอกันขณะที่ YOLO v4 สามารถตรวจจับได้ครบถ้วนและถูกต้องทั้งหมด จึงได้นำ Weight ที่เทรนใน YOLO v4 มาใช้ในการตรวจจับรูปภาพจิ๊กซอว์



รูปที่ 4.3 ผลลัพธ์จากการ Detect รูปภาพจิ๊กซอว์ เทคนิคด้วย YOLO v3



รูปที่ 4.4 ผลลัพธ์จากการ Detect รูปภาพจิ๊กซอว์ เทคนิคด้วย YOLO v4

จากรูปที่ 4.3 และ 4.4 ได้ผลการทดลองตรวจจับรูปภาพ ได้ค่าเฉลี่ย Accuracy ของ YOLO v3 อยู่ที่ประมาณ 88.41% และ YOLO v4 อยู่ที่ 96%

3. ผู้จัดทำได้ทำการออกแบบตำแหน่งของ Bounding Box ของรูปภาพที่ได้ทำการ Detect โดยใช้ OpenCV เพื่อทำการเรียงลำดับตำแหน่งจากบนลงมาล่าง ตามการทำงานแบบ Flowchart โดยเมื่อ Detect ภาพ ออกมานะจะมีการนองกว่าแต่ละรูปทรงมีตำแหน่งอยู่ในส่วนใดของรูปภาพ มี 3 ค่า คือ Center x และ y ค่าสุดท้ายคือ Class ที่ได้ทำการเรียงลำดับจากบนลงล่าง ดังแสดงตามรูปที่ 4.5

In [10]: `sorted(lst, key=lambda x: x[-2])`

Out[10]: [[188, 39, '0'],  
 [182, 90, '3'],  
 [166, 143, '2'],  
 [154, 189, '2'],  
 [148, 232, '4'],  
 [348, 256, '5'],  
 [393, 266, '5'],  
 [435, 274, '5'],  
 [481, 285, '6'],  
 [245, 294, '3'],  
 [120, 323, '7'],  
 [106, 371, '2'],  
 [101, 417, '2'],  
 [95, 459, '1']]

รูปที่ 4.5 ตัวอย่างการแสดงตำแหน่งของ Bounding Box ของรูปภาพที่ทำการ Detect

## 4.2 การทดลองของระบบ

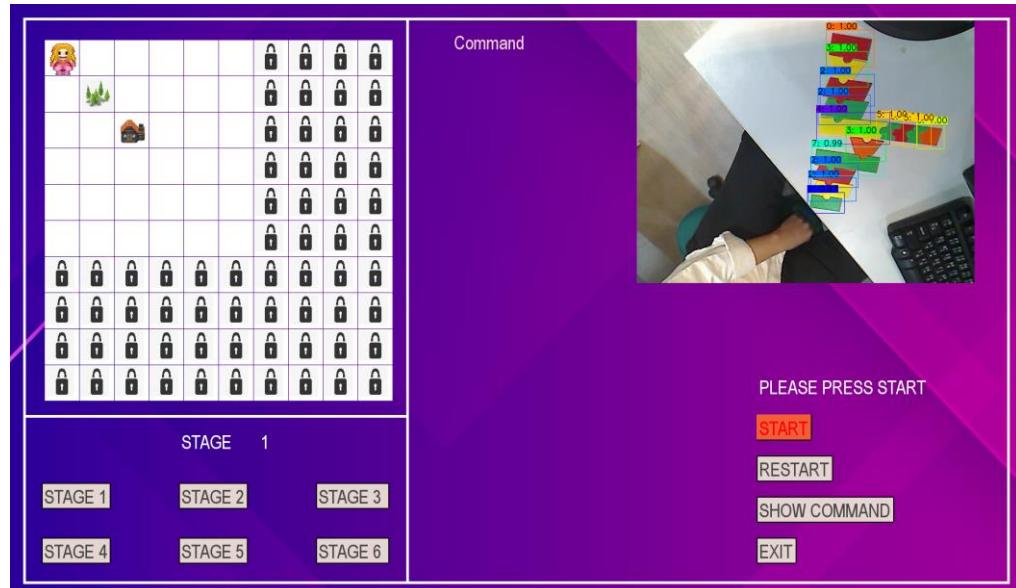
จากการที่ผู้จัดทำได้ทำการเทรนรูปภาพจิ๊กซอว์ครบถ้วนทรง ขึ้นตอนต่อไปคือการนำผลที่ได้จากการ Detect รูปทรงจิ๊กซอว์มาใช้งานร่วมกับระบบ โดยมีขั้นตอนการทำงานดังนี้

- เมื่อเปิดระบบขึ้นมาสามารถเปิดดูคำอธิบายรูปทรงจิ๊กซอว์เพื่อทำให้สามารถต่อได้ถูกวิธีและหน้าที่ของ Object ที่แสดงในด้าน โดยการกดไปที่ปุ่ม Show Command ดังแสดงตามรูปที่ 4.6



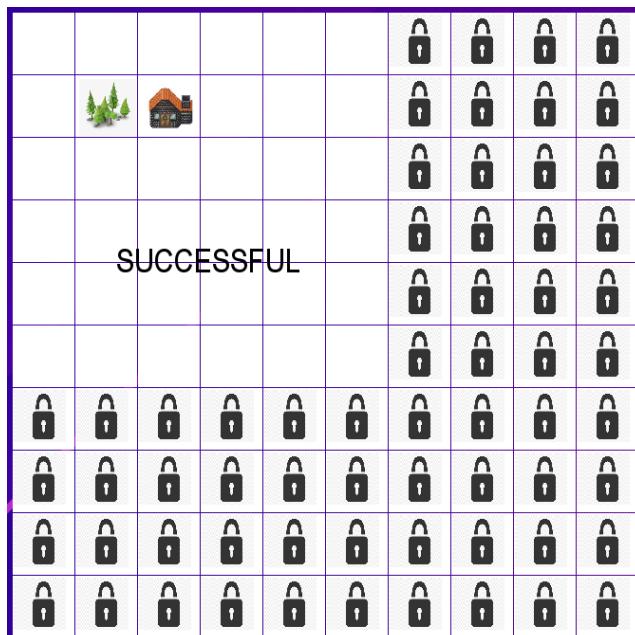
รูปที่ 4.6 แสดงคำอธิบายรูปทรงจิ๊กซอว์ในระบบ

- สามารถเลือกค่าด้านโดยกดไปที่ปุ่ม Stage ต่าง ๆ โดยที่แต่ละค่าจะมีความยากและง่ายแตกต่างกัน แต่ในตัวอย่างผู้จัดทำได้นำค่าด้านที่ 1 หรือ Stage 1 มาเป็นตัวอย่าง ดังแสดงตามรูปที่ 4.7

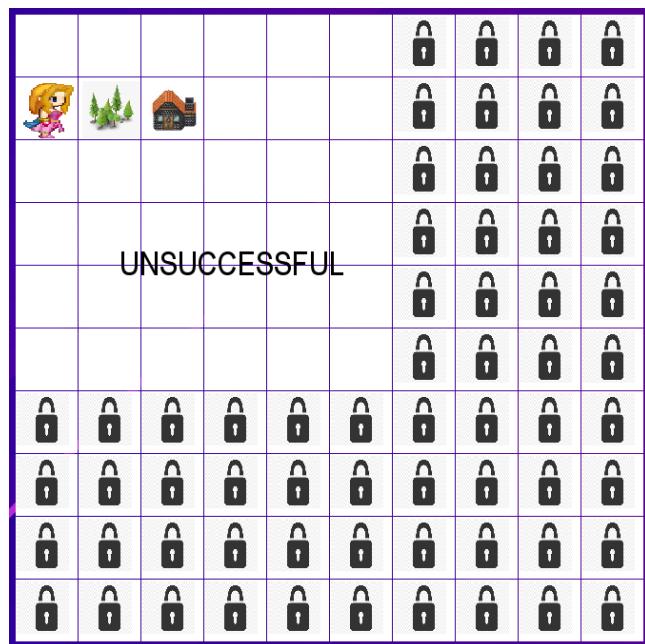


รูปที่ 4.7 ตัวอย่างระบบในด่านที่ 1

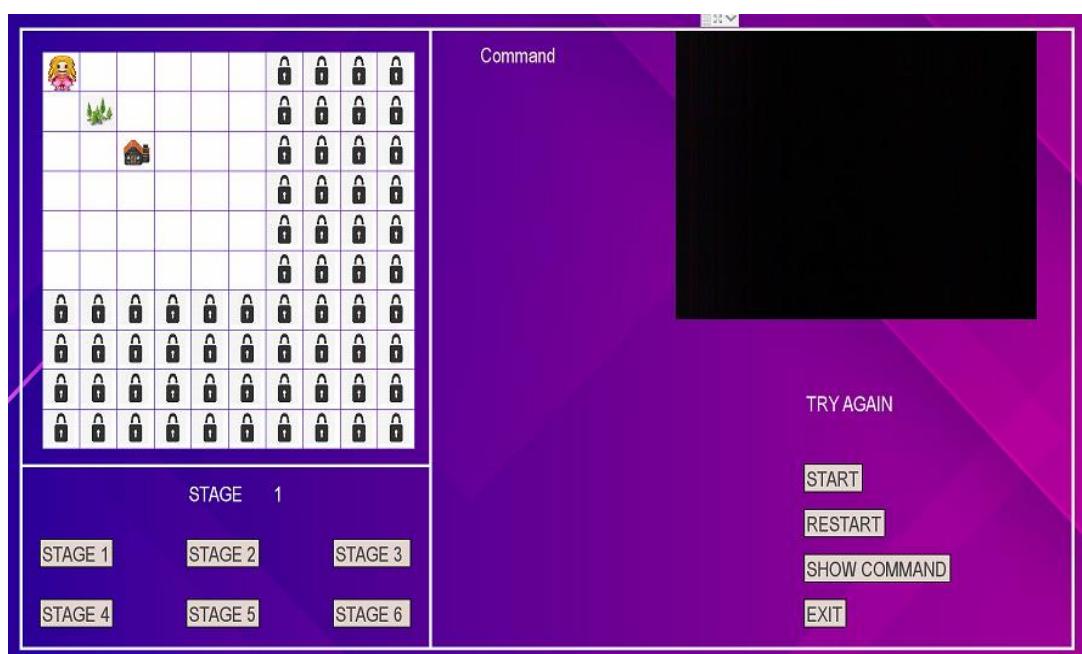
3. เมื่อกดปุ่ม Start จะเป็นการถ่ายรูปภาพจิ้กซอว์ที่ได้ทำการต่อไว้และแสดงผลบนระบบที่เป็นกล้องเว็บแคม จากนั้นจะนำภาพเข้าไปตรวจจับจิ้กซอว์เพื่อแปลงเป็นคำสั่ง ทำให้ Robot เดินตามคำสั่ง เมื่อ Robot เดินถึงเป้าหมายสำเร็จจะแสดงข้อความ “SUCCESSFUL” แต่หากเดินไม่ถึงเป้าหมายจะแสดงข้อความ “UNSUCCESSFUL” ถ้ากดปุ่ม Start โดยที่ไม่ได้นำจิ้กซอว์ที่ต่อไว้มาระบุลงกล้องเว็บแคม จะขึ้นข้อความว่า “TRY AGAIN” หมายความว่า ไม่เจอจิ้กซอว์ในรูปภาพ ดังแสดงตามรูปที่ 4.8 , 4.9 และ 4.10



รูปที่ 4.8 แสดงข้อความ “SUCCESSFUL” เมื่อ Robot เดินถึงเป้าหมาย

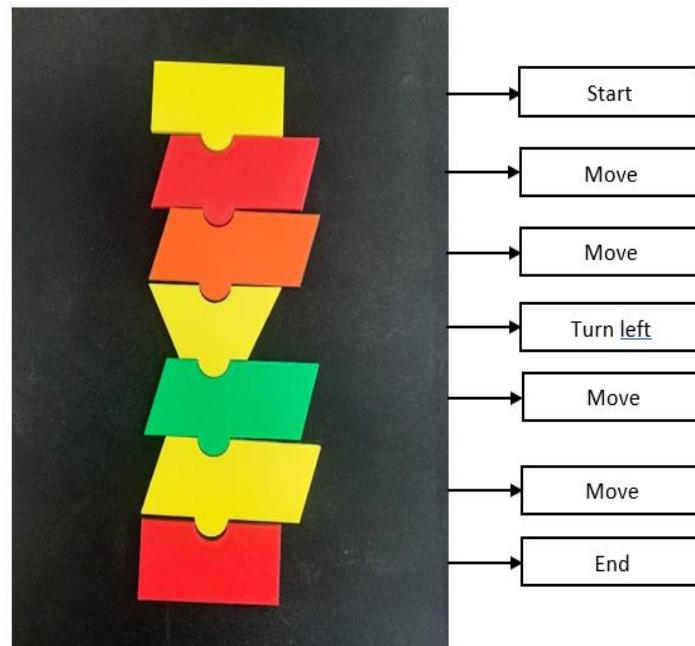


รูปที่ 4.9 แสดงข้อความ “ UNSUCCESSFUL ” เมื่อ Robot เดิน ไม่ถึงเป้าหมาย

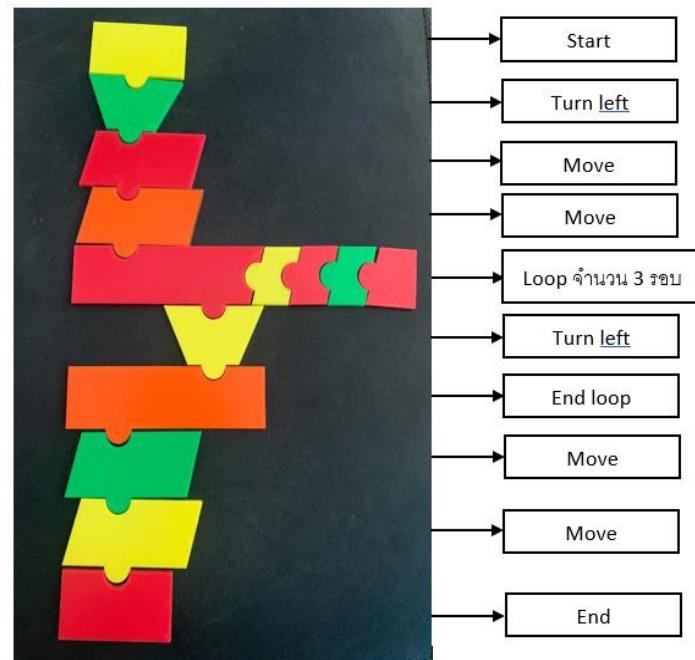


รูปที่ 4.10 แสดงข้อความ “TRY AGAIN” เมื่อไม่เจอจีกซอว์ในรูปภาพ

4. ทางผู้จัดทำได้ทำการทดลองต่อจีกซอว์แก้โจทย์ 2 วิธี ใน Stage 1 เพื่อให้เห็นรูปภาพการทำงาน จากระบบที่ได้ออกแบบ ดังแสดงตามรูปที่ 4.11 และ 4.12



รูปที่ 4.11 การต่อจิ๊กซอว์เพื่อแก้โจทย์ วิธีปกติ



รูปที่ 4.12 การต่อจิ๊กซอว์เพื่อแก้โจทย์ วิธี Loop

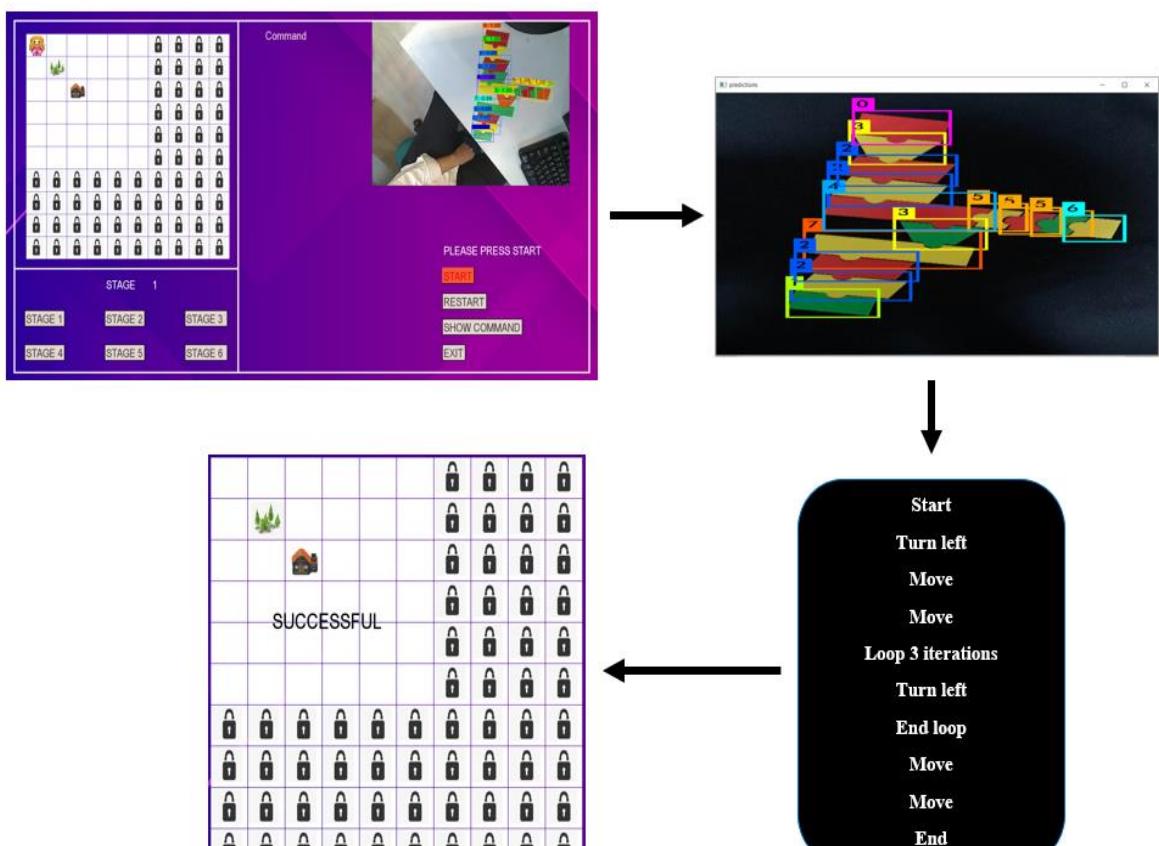
5. ผู้ใช้งานสามารถกดคปุ่ม “Restart” เพื่อเป็นการเคลียร์คำสั่งก่อนหน้าและเป็นการรีเซ็ตค่า ให้ Robot กลับไปยังจุดเริ่มต้นของค่าอนันน์ ๆ
6. ผู้ใช้งานสามารถกดคปุ่ม “EXIT” เพื่อเป็นการออกจากระบบ

### 4.3 ข้อจำกัดในการใช้งานระบบ

1. ในการต่อจิ๊กซอว์ ความมีผ้าหรือกระดาษแข็งที่เป็นสีขาวหรือสีดำในการรองเป็นฐานต่อจิ๊กซอว์ ไม่ควรมีฐานรองเป็นรูปทรงที่คล้ายกับจิ๊กซอว์ เพราะอาจจะทำให้ตรวจจับผิดพลาด
2. ระยะมุมสูงของกล้องการพอดีกับการต่อจิ๊กซอว์ใน 1 ชุดคำสั่ง ไม่ควรสูงเกินไป
3. เมื่อต่อจิ๊กซอว์เสร็จ ควรจะเคลียร์จิ๊กซอว์ที่ไม่ได้ใช้ออกจากกล้องเว็บแคมก่อน เพื่อไม่ให้เกิดการตรวจจับจิ๊กซอว์ขึ้นที่ไม่ต้องการ
4. การเดินครัวอยู่ในที่ ๆ มีแสงไม่สว่างหรือมีดเกินไป เพราะถ้าสว่างไป จิ๊กซอว์สีอ่อนจะกลืนไป กับแสงทำให้ตรวจจับไม่เจอ แต่ถ้ามีดเกินไปก็ทำให้ตรวจจับไม่ชัดหรือไม่ครบหุ่ง

### 4.4 สรุปขั้นตอนการทำงานของระบบ

จากการที่ผู้จัดทำได้ทำการทดลองใช้ระบบแบบจำลองหุ่นยนต์ ได้สรุปขั้นตอนกระบวนการทำงานดังแสดงตามรูปที่ 4.13



รูปที่ 4.13 กระบวนการทำงานของระบบ

## บทที่ 5

# สรุปผลการวิจัย อุปสรรค และการพัฒนาต่อในอนาคต

ในบทนี้ผู้จัดทำได้ทำการสรุปผลการดำเนินงานที่ผ่านมาในภาคเรียนที่ 1 และภาคเรียนที่ 2 โดยในภาคเรียนที่ 1 นั้นผู้จัดทำได้ทำการศึกษาโมเดลที่ใช้ตรวจจับวัตถุจากรูปภาพ , วางแผน , ทดลอง และออกแบบระบบเมื่อต้น และในภาคเรียนที่ 2 ก็ได้นำผลลัพธ์จากการเทรนรูปภาพจิกซ์อว์มาร่วมกับระบบเพื่อให้ทำงานได้ตามที่ต้องการ ทั้งนี้ได้รวมถึงอุปสรรคและการพัฒนาต่อไปในอนาคตอีกด้วย

## 5.1 สรุปผลการดำเนินงานวิจัย

ระบบแบบจำลองหุ่นยนต์ที่สามารถเดินตามคำสั่งด้วยการตรวจจับจิกซ์อว์ในรูปภาพนี้ ผู้จัดทำได้ทำการศึกษาโมเดลในการตรวจจับรูปภาพหรือ Object Detection เพื่อหาโมเดลที่คิดว่า สามารถนำมาใช้งานได้ในระยะเวลาที่กำหนด มีการเปรียบเทียบผลลัพธ์ ส่วนระบบผู้จัดทำได้ พัฒนาให้ใช้งานร่วมกับโมเดลที่ได้เลือกมา โดยมีวัตถุประสงค์ให้สามารถนำไปช่วยให้การเรียน โปรแกรมมิ่งในเด็กระดับประถมศึกษามีความน่าสนใจมากขึ้น

ซึ่งระบบที่ถูกพัฒนาแล้วนั้น สามารถใช้งานเลือกโจทย์เพิ่มโจทย์ และออกแบบโจทย์เอง ได้ เมื่อใช้งานเวลาด้วยจิกซ์อว์จะสามารถเช็คคำสั่งหรือ Command ได้ สามารถต่อจิกซ์อว์ไปแล้วเช็คคำสั่งไปด้วยได้เพื่อทำให้เกิดลองผิบลองถูก เห็นภาพในหัว ทำให้เมื่อเจอโจทย์ที่ยากขึ้น จะสามารถ ประมวลผล ได้ เนื่องจากการทำงานของคำสั่ง โดยดูจากการที่ Robot เดินไปให้ถึงเป้าหมาย ส่วนรูปภาพ Robot , Objective หรือสิ่งกีดขวางต่าง ๆ สามารถเปลี่ยนรูปได้หากนำไปใช้งานจริง ในส่วนของโมเดล Object Detection ได้เลือกนำ YOLO v4 มาใช้งานเนื่องจากมีความสามารถในการ ตรวจจับที่แม่นยำกว่า YOLO v3 ซึ่งเป็นที่นิยมกันมาก ผู้จัดทำได้ทำการเปรียบเทียบทั้ง 2 โมเดล เพิ่มรูปภาพเข้าไปเทรนเพื่อเพิ่มความแม่นยำและไม่ทำให้เกิด Overfitting มีออกแบบรูปทรงจิกซ์อว์ ที่ยังไม่ได้นำมาใช้เพื่อและเข้าไป เตรนร่วมกันเพื่อที่ในอนาคตมีการเพิ่มคำสั่งจะ ได้ไม่ต้องเสียเวลา ออกแบบและเทรนใหม่ เมื่อนำทั้ง 2 ส่วนมาร่วมกันสามารถนำไปใช้ประโยชน์ในการศึกษา และต่อยอด ได้มากมายในอนาคต

## 5.2 อุปสรรคระหว่างดำเนินงานวิจัย

1. ในการทำจิกซ์อว์ช่วงแรกผู้จัดทำได้ทำเป็นการตัดกระดาษ แต่พอมาใช้งานจริงแล้ว ไม่ดี เท่าที่ควร จึงได้ศึกษาและใช้งานเครื่อง 3D Printer ซึ่งค่อนข้างที่จะดี แต่การซื้อเครื่องและเวลา ที่ใช้ในการทำจิกซ์อว์ค่อนข้างนาน

2. รูปทรงจีกซอร์ที่ได้ออกแบบขึ้นมาเองและมีทั้งหมด 15 รูปทรง ทำให้การที่จะจัดเตรียมข้อมูลทั้งการออกแบบ , ถ่ายรูป และทำ Label ค่อนข้างที่จะใช้เวลานานพอสมควร
3. การใช้งานโมเดล YOLO v3 ในภาคเรียนที่ 1 ยังไม่สามารถตรวจจับได้ครบถ้วนรูปทรงเนื่องจากข้อมูลใน Dataset เป็นรูปถ่ายจากกล้องโทรศัพท์ เมื่อมาใช้งานจริงต้องใช้กล้องเว็บแคมซึ่งมีความคมชัดน้อยกว่า ทำให้เกิด Overfitting ผู้จัดทำจึงทำการศึกษาโมเดล YOLO v4 ซึ่งเปิดตัวมาช่วงกลางปี ค.ศ.2020 และเพิ่มข้อมูลใน Dataset ในมีรูปที่ถ่ายจากกล้องเว็บแคมเข้าไปด้วยทำให้เกิดการเทรนเพิ่มขึ้นและใช้เวลานานกว่าเดิม

### 5.3 สิ่งที่จะพัฒนาต่อยอดในอนาคต

1. เพิ่มคำสั่งให้กับจีกซอร์ที่เหลืออยู่และเพิ่มรูปแบบโจทย์ ทำให้มีการต่อด้วยรูปแบบที่หลากหลายขึ้น
2. อออกแบบระบบให้มีหน้าตาที่น่าใช้งานมากขึ้น ดึงดูดเด็กในระดับประถมศึกษาสนใจกับระบบนี้

## บรรณานุกรม

- [1] ณริรัตน์ บุญหลัง, ยุทธพงศ์ เลือกกลั่นดี และนวัตตน์ รามสูตร (2019). “Coding” สร้างคน สู่  
ศตวรรษที่21. [ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/3cmz6wn>
- [2] นัตรพงศ์ ชูแสงนิล (2019). เกมการเรียนรู้แบบ Unplug. [ออนไลน์] เข้าถึงได้จาก:  
<https://www.scimath.org/article-technology/item/10631-unplug>
- [3] Tim Bell, Ian H. Witten and Mike Fellows (1998). **Computer Science Unplugged . . . off-line activities and games for all ages.** [ออนไลน์] เข้าถึงได้จาก:  
<https://bit.ly/3cg0Llh>
- [4] Robyn Adams and Jane McKenzie (2015). **An enrichment and extension programme for primary-aged students.** [ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/2Ld5tBA>
- [5] กันต์ อุ่ยมอินทร (2019). **Computer Science Unplugged** เรียนคอมฯแบบไม่ใช้คอมฯ.  
[ออนไลน์] เข้าถึงได้จาก: <https://www.bangkokbiznews.com/blog/detail/647968>
- [6] CS Unplugged. **Rescue Mission.** [ออนไลน์] เข้าถึงได้จาก:  
<https://csunplugged.org/en/topics/kidbots/unit-plan/rescue-mission/>
- [7] Keng Surapong (2020). **Object Detection คืออะไร บทความสอน AI ตรวจจับวัตถุ.** [ออนไลน์]  
เข้าถึงได้จาก: <https://www.bualabs.com/archives/3453>
- [8] Faen Zhang, Xinyu Fan, Guo Ai, Jianfei Song, Yongqiang Qin, Jiahong Wu (2019).  
**Accurate Face Detection for High Performance.** [ออนไลน์] เข้าถึงได้จาก:  
<https://arxiv.org/pdf/1905.01585v3.pdf>
- [9] Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, Ling Shao (2020). **Pedestrian Detection: The Elephant In The Room.** [ออนไลน์] เข้าถึงได้จาก:  
<https://arxiv.org/pdf/2003.08799v6.pdf>
- [10] Ross Girshick Jeff Donahue Trevor Darrell Jitendra Malik (2014). **Rich feature hierarchies for accurate object detection and semantic segmentation Tech report (v5)**  
[ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/1311.2524.pdf>

- [11] Olga Russakovsky, Jia Deng, et al. (2015). **ImageNet Large Scale Visual Recognition Challenge.** [ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/1409.0575.pdf>
- [12] Pasu Jitnamporn (2019). **CNN คืออะไร ...** [ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/2A2fkb9>
- [13] Ross Girshick (2015). **Fast R-CNN.** [ออนไลน์] เข้าถึงได้จาก:  
<https://arxiv.org/pdf/1504.08083.pdf>
- [14] Shaoqing Ren , Kaiming He , Ross Girshick , Jian Sun (2016). **Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks.** [ออนไลน์]  
เข้าถึงได้จาก: <https://arxiv.org/pdf/1506.01497.pdf>
- [15] Tanay Karmarkar (2018). **Region Proposal Network (RPN) - Backbone of Faster R-CNN.**  
[ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/2VUucjp>
- [16] Joseph Redmon , Ali Farhadi (2018). **YOLOv3: An Incremental Improvement.**  
[ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/1804.02767.pdf>
- [17] Alexey Bochkovskiy , Chien-Yao Wang , Mark Liao (2020). **YOLOv4: Optimal Speed and Accuracy of Object Detection.** [ออนไลน์] เข้าถึงได้จาก:  
<https://arxiv.org/pdf/2004.10934.pdf>
- [18] Tsung-Yi Lin, Michael Maire, Serge Belongie., et al. (2015). **Microsoft COCO: Common Objects in Context.** [ออนไลน์] เข้าถึงได้จาก:  
<https://arxiv.org/pdf/1405.0312.pdf>
- [19] Sarayut Nonsiri , PhD. **ภาษาโปรแกรม Python คืออะไร ?.** [ออนไลน์] เข้าถึงได้จาก:  
<https://www.9experttraining.com/articles/python-คืออะไร>
- [20] Nuttakan Chuntra (2018). **มาสร้างแอปด้วย Pygame กันเถอะ (ขั้นตอนการติดตั้ง).**  
[ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/3dgtiVd>
- [21] **Tensorflow (เทนเซอร์ฟล)** คืออะไร. [ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/35H5JlC>
- [22] Nuttakan Chuntra (2018). **OpenCV คืออะไร?.** [ออนไลน์] เข้าถึงได้จาก:  
<https://bit.ly/3cZo5C1>
- [23] Joseph Redmon (2013 - 2016). **Darknet: Open Source Neural Networks in C.** [ออนไลน์]  
เข้าถึงได้จาก: <https://pjreddie.com/darknet/>

[24] AlexeyAB. **AlexeyAB/darknet**. [ออนไลน์] เข้าถึงได้จาก:

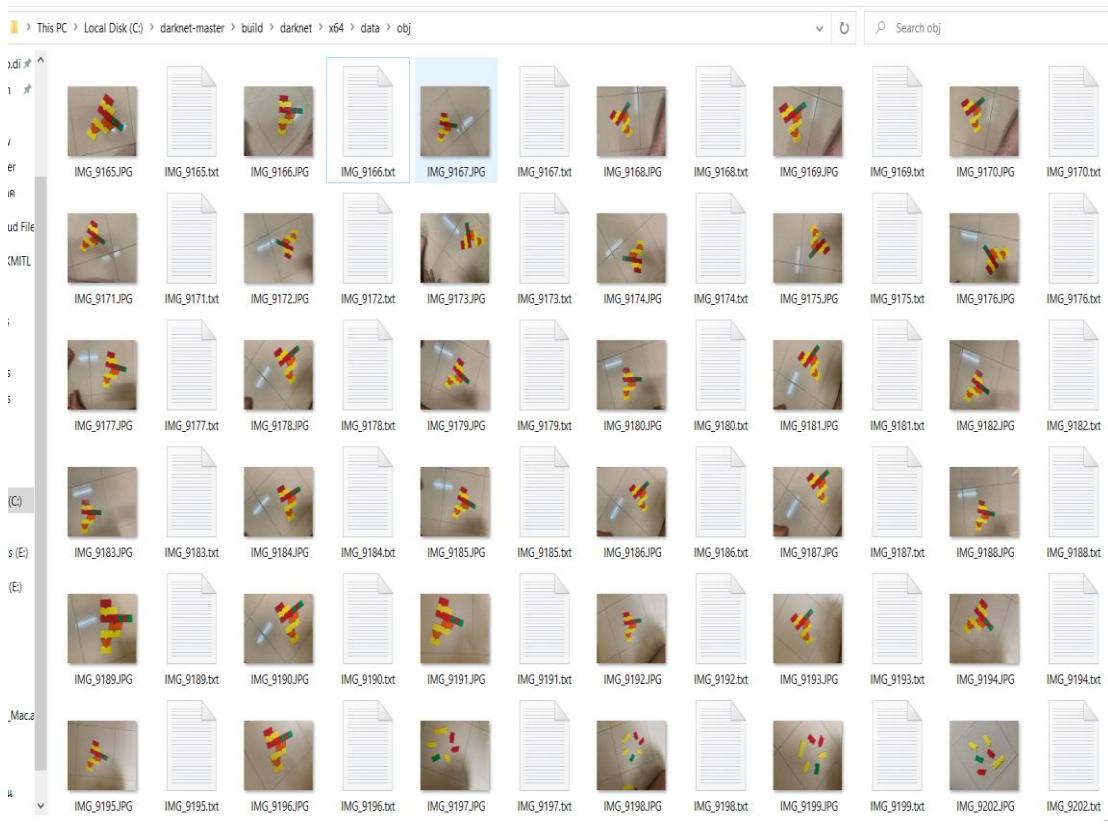
<https://github.com/AlexeyAB/darknet>

[25] tzutalin. **tzutalin / labelImg**. [ออนไลน์] เข้าถึงได้จาก: <https://github.com/tzutalin/labelImg>

## ភាគធនវក

ภาคผนวก ก.  
คู่มือการเทrnโนมเดลใน YOLO

1. ทำการถ่ายรูปสิ่งของที่ต้องการนำมาทำ Object Detection จากนั้นนำรูปภาพทั้งหมดมาทำ Label เพื่อเป็นการบอกตำแหน่งของวัตถุว่าอยู่ตรงไหนของรูปภาพ เพื่อที่เวลานำเข้าไปเทรน โมเดล จะได้รู้ว่าตำแหน่งนี้คือวัตถุชิ้นนี้ เมื่อทำเสร็จจะได้ไฟล์มา 2 นามสกุลคือ jpg (รูปที่ถ่าย) และ txt (คำตำแหน่งของ Label) ถือเป็นการเตรียมข้อมูลก่อนนำไปเทรน
2. ทำการติดตั้ง YOLO วิธีของ AlexeyAB จาก <https://github.com/AlexeyAB/darknet> !เมื่อติดตั้งเสร็จนำรูปและ Label ที่ได้นำใส่โฟลเดอร์ C:\darknet-master\build\darknet\x64\data\obj



รูปที่ ก.1 ตัวอย่างไฟล์รูปภาพและไฟล์ Label ทั้งหมด

### 3. ทำการเตรียมไฟล์ Config

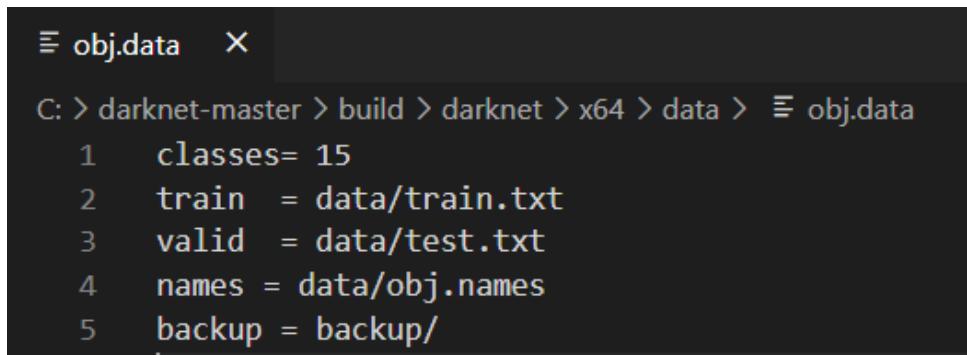
- 3.1 ทำการสร้างไฟล์ obj.names เพื่อทำหน้าที่บอกชื่อ Class ว่ามีอะไรบ้าง ในที่นี้ผู้จัดทำได้ตั้งให้แต่ละรูปทรงเป็น Class ตั้งแต่ 0 – 14



```
0
1
2
3
4
5
6
7
8
9
10
11
12
13
14
```

รูปที่ ก.2 ไฟล์ obj.names

3.2 สร้างไฟล์ obj.data เพื่อเป็นการบอก Path ต่าง ๆ ในการเข้าไปดึงข้อมูลมาเทรน

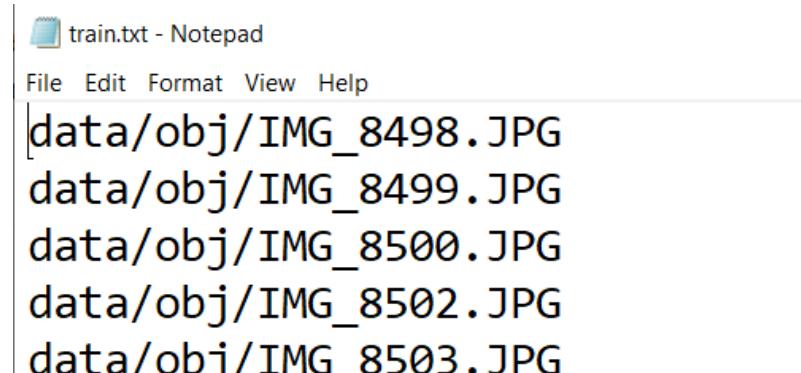


```
≡ obj.data  ×
C: > darknet-master > build > darknet > x64 > data > ≡ obj.data
1   classes= 15
2   train   = data/train.txt
3   valid   = data/test.txt
4   names   = data/obj.names
5   backup  = backup/
|
```

รูปที่ ก.3 ไฟล์ obj.data

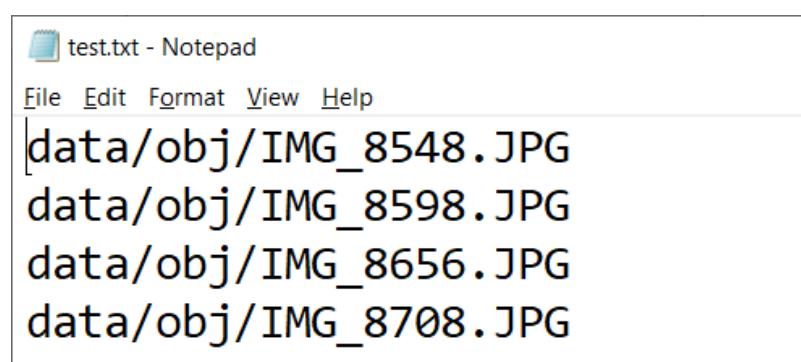
classes = จำนวนคลาสหรือประเภทของวัตถุในรูปภาพ ในที่นี้ 15 คือจีกซอร์ทั้งหมด 15 รูปทรง  
 train = เป็นการบอก Path ที่เชื่อมไปในตอนที่ train ข้อมูล  
 test = เป็นการบอก Path ที่เชื่อมไปในตอนที่ test ข้อมูล  
 names = เป็นการบอก Path เพื่อทำการบอกว่ามีคลาสชื่ออะไรบ้าง  
 backup = ในระหว่างที่เทรน จะเก็บไฟล์ weights ไว้ใน Path นี้

3.3 ทำการแบ่งข้อมูล train test จะได้เป็นไฟล์ train.txt และ test.txt โดยในนั้นจะมี Path ของรูปภาพแต่ละรูปที่ได้ทำการแบ่งข้อมูลไว้แล้ว โดยควรจะแบ่งข้อมูลไว้ test ประมาณ 10% - 15% จากข้อมูลทั้งหมด



```
train.txt - Notepad
File Edit Format View Help
data/obj/IMG_8498.JPG
data/obj/IMG_8499.JPG
data/obj/IMG_8500.JPG
data/obj/IMG_8502.JPG
data/obj/IMG_8503.JPG
```

รูปที่ ก.4 ไฟล์ trian.txt



```
test.txt - Notepad
File Edit Format View Help
data/obj/IMG_8548.JPG
data/obj/IMG_8598.JPG
data/obj/IMG_8656.JPG
data/obj/IMG_8708.JPG
```

รูปที่ ก.5 ไฟล์ test.txt

3.4 ปรับค่า Config โดย copy ไฟล์จาก yolov4.cfg มาเปลี่ยนเป็น yolov4-custom.cfg และนำไปไว้ใน Path C:\darknet-master\build\darknet\x64 以便นั้นปรับค่าดังนี้

- บรรทัดที่ 2 ค่า batch คือการแบ่งข้อมูลทั้งหมดออกเป็นก้อน ๆ ตามที่ได้กำหนดค่า เพื่อลดเวลาในการเทรนและทำให้เครื่องไม่เกิน Memory หากเกินไป ในส่วนนี้การกำหนดค่าต้องดูที่ Memory ของแต่ละเครื่องว่า ได้ถึงแค่ไหน โดยสามารถกำหนดได้ ตั้งแต่ 16, 32, 64
- บรรทัดที่ 7, 8 ปรับค่า width และ height จาก 608 ให้เป็น 416
- บรรทัดที่ 9 channels หากข้อมูลรูปภาพเป็นสี RGB กำหนดให้เป็น 3 เมื่อเดิม แต่ หากข้อมูลรูปเป็น Grayscale ให้กำหนด channels เป็น 1
- บรรทัดที่ 19 max\_batches คือการกำหนดจำนวนรอบในการเทรนทั้งหมด ยิ่งข้อมูลมี คลาสเยอะ ควรจะตั้งจำนวนรอบให้เยอะขึ้นตาม
- บรรทัดที่ 21 steps คือการแบ่งจำนวนรอบในการเทรน มี 2 ค่า ควรกำหนดให้เป็น 80% และ 90% ของ max\_batches
- บรรทัดที่ 968, 1,056 และ 1,144 หรือจะกดคันหาคำว่า yolo แล้วจะเจอคำว่า classes

- ให้เปลี่ยนค่านี้ทั้ง 3 บรรทัดให้เป็นจำนวนคลาสทั้งหมดของข้อมูลเรา
- เมื่อใส่ตรง classes เมื่อเลื่อนขึ้นไปจะเห็นคำว่า filters ของทั้ง 3 บรรทัด ให้เปลี่ยนค่า  
นี้ โดยใช้สูตร  $(\text{classes} + 5) * 3$

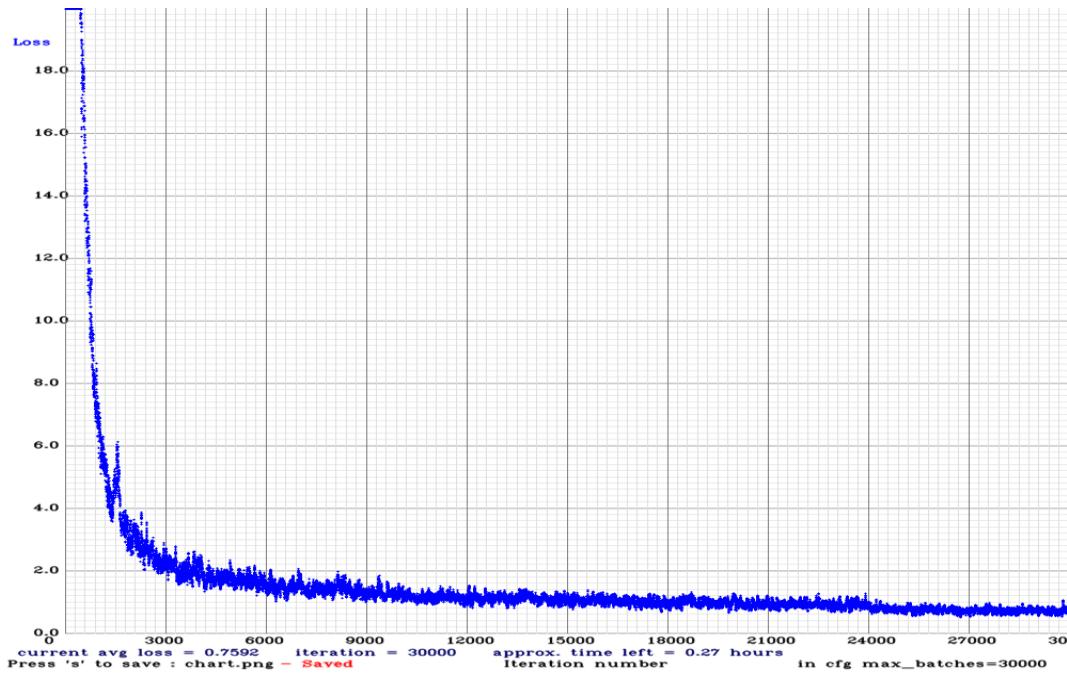
```

958 [convolutional]
959 size=1
960 stride=1
961 pad=1
962 filters=60
963 activation=linear
964
965 [yolo]
966 mask = 0,1,2
967 anchors = 12, 16, 19, 36, 40, 28, 36, 75, 76, 55, 72, 146, 142, 110, 192, 243, 459, 401
968 classes=15
969 num=9
970 jitter=.3
971 ignore_thresh = .7
972 truth_thresh = 1
973 scale_x_y = 1.2
974 iou_thresh=0.213
975 cls_normalizer=1.0
976 iou_normalizer=0.07
977 iou_loss=ciou
978 nms_kind=greedyNMS
979 beta_nms=0.6
980

```

รูปที่ ก.6 ไฟล์ yolov4-custom.cfg

4. เมื่อทำการเซ็ตค่า Config เสร็จ ให้ทำการดาวน์โหลด pre-trained weights-file ของ YOLO v4 ([https://github.com/AlexeyAB/darknet/releases/download/darknet\\_yolo\\_v3\\_optimal/yolov4.conv.137](https://github.com/AlexeyAB/darknet/releases/download/darknet_yolo_v3_optimal/yolov4.conv.137))
5. เริ่มทำการเทรนโดยเปิด command line เข้าไปที่ Path C:\darknet-master\build\darknet\x64 จากนั้นให้ทำการใส่คำสั่ง darknet.exe detector train data/obj.data yolov4-custom.cfg yolov4.conv.137
6. ในระหว่างการเทรน จะมีกราฟแสดงค่า avg loss ซึ่งสามารถเช็คคุณว่า model สามารถทำงานได้ดี แค่ไหน และໄວ่สำหรับการนำไฟล์ weights มาทำการ test ความแม่นยำได้โดยคุณจากกราฟว่า รอบไหนควรนำมา test

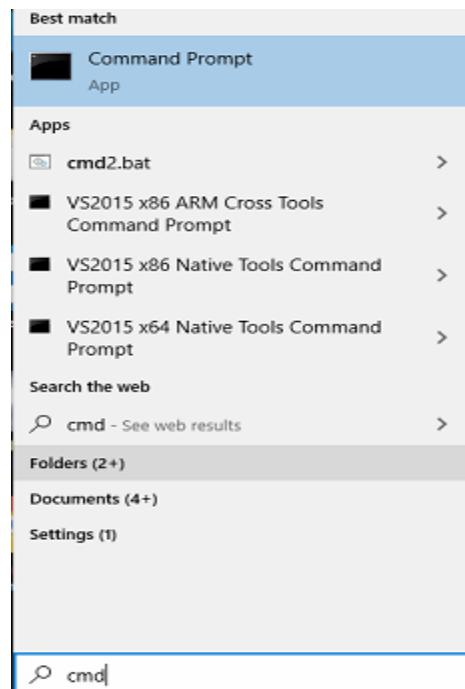


รูปที่ ก.7 กราฟแสดงค่า avg loss ในระหว่างการเทรน

7. เมื่อทำการเทรนเสร็จ จะได้ไฟล์ weights คล้ายไฟล์ดังแต่รอบที่ 100 จนถึงรอบสุดท้าย อยู่ในไฟล์เดอร์ backup ซึ่งสามารถนำรูปมาทำการ test โดยใส่คำสั่ง darknet detector test data/obj.data yolov4-custom.cfg backup/yolov4-custom\_final.weights data/<image file>
8. หากต้องการตรวจจับจากวิดีโอให้ใส่คำสั่ง darknet detector demo data/obj.data yolov3-custom.cfg backup/yolov3-custom\_final.weights <video file>
9. หากต้องจากตรวจจับแบบ Realtime จากกล้อง webcam ให้ใส่คำสั่ง darknet detector demo data/obj.data yolov3-custom.cfg backup/yolov3-custom\_final.weights

ภาคผนวก ข.  
คู่มือการติดตั้งระบบ

1. ติดตั้ง python version 3.7.8 ( <https://www.python.org/downloads/release/python-378/> )
2. ติดตั้ง virtualenv เพื่อสร้าง environment เปิด Command Prompt พิมพ์ pip install virtualenv แล้วกด Enter เพื่อติดตั้ง environment เมื่อติดตั้งเสร็จแล้วให้ไปที่โฟลเดอร์ของระบบ และคัดลอก path ไว้ ในหน้า Command Prompt พิมพ์ cd เว้นวรรค และตามด้วย path ที่ได้คัดลอกไว้ เพื่อเข้าไปที่โฟลเดอร์ของระบบ

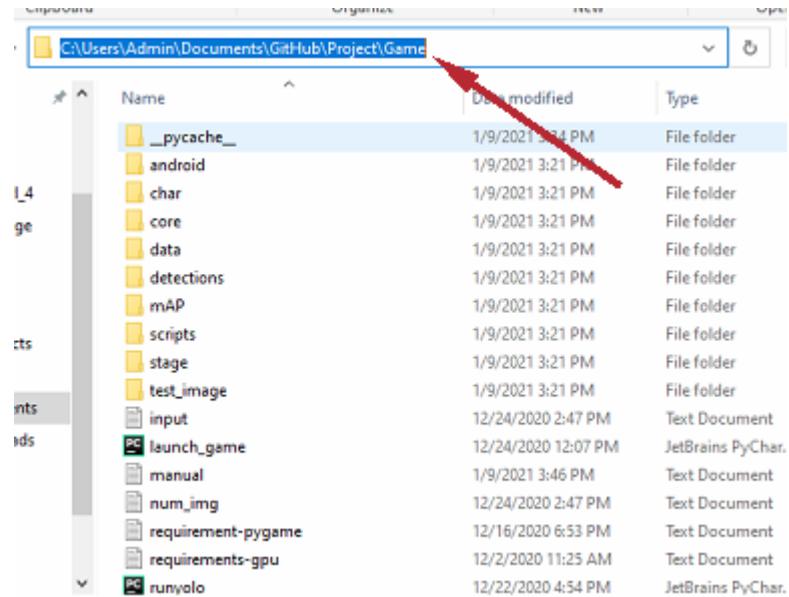


รูปที่ ๑.๑ เลือก Command Prompt ในช่องค้นหา

```
Command Prompt
Microsoft Windows [Version 10.0.18363.1256]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\Admin>pip install virtualenv
```

รูปที่ ๑.๒ pip install virtualenv



|                    | Name                | Date modified     | Type |
|--------------------|---------------------|-------------------|------|
| __pycache__        | 1/9/2021 3:21 PM    | File folder       |      |
| android            | 1/9/2021 3:21 PM    | File folder       |      |
| char               | 1/9/2021 3:21 PM    | File folder       |      |
| core               | 1/9/2021 3:21 PM    | File folder       |      |
| data               | 1/9/2021 3:21 PM    | File folder       |      |
| detections         | 1/9/2021 3:21 PM    | File folder       |      |
| mAP                | 1/9/2021 3:21 PM    | File folder       |      |
| scripts            | 1/9/2021 3:21 PM    | File folder       |      |
| stage              | 1/9/2021 3:21 PM    | File folder       |      |
| test_image         | 1/9/2021 3:21 PM    | File folder       |      |
| input              | 12/24/2020 2:47 PM  | Text Document     |      |
| launch_game        | 12/24/2020 12:07 PM | JetBrains PyChar. |      |
| manual             | 1/9/2021 3:46 PM    | Text Document     |      |
| num_img            | 12/24/2020 2:47 PM  | Text Document     |      |
| requirement-pygame | 12/16/2020 6:53 PM  | Text Document     |      |
| requirements-gpu   | 12/2/2020 11:25 AM  | Text Document     |      |
| runyolo            | 12/22/2020 4:54 PM  | JetBrains PyChar. |      |

รูปที่ ข.3 กดลอก path ไฟล์เดอร์ระบบ

```
C:\Users\Admin>cd C:\Users\Admin\Documents\GitHub\Project\Game
```

รูปที่ ข.4 เข้าไปที่ไฟล์เดอร์ระบบด้วยคำสั่ง cd (folder path)

3. สร้าง environment สำหรับระบบ ซึ่งจะใช้ virtualenv ที่ได้ติดตั้งไว้แล้วบนไฟล์เดอร์ของระบบ โดยใช้ คำสั่ง virtualenv env\_name หากมี version python มากกว่าหนึ่ง version ให้ใช้คำสั่ง virtualenv -p ( python path ) env\_name

```
C:\Users\Admin\Documents\GitHub\Project\Game>virtualenv env_name
```

รูปที่ ข.5 สร้าง environment ด้วยคำสั่ง virtualenv env\_name

```
C:\Users\Admin\Documents\GitHub\Project\Game>virtualenv -p C:\Users\Admin\AppData\Local\Programs\Python\Python37\Scripts
\ env_name
```

รูปที่ ข.6 สร้าง environment และกำหนด path กรณีมี python หลาย version

4. เข้าใช้งานใน environment โดยการพิมพ์คำสั่ง env\_name\Scripts\activate หลังจากเข้ามาใน environment ที่สร้างแล้ว จะต้องติดตั้ง library ที่จำเป็นของระบบ โดยพิมพ์ pip install -r requirement-pygame.txt และรอสักครู่ หลังจากนั้น ให้ติดตั้ง library อีก 1 ชุด โดยพิมพ์ pip install -r requirements-gpu.txt

```
C:\Users\Admin\Documents\GitHub\Project\Game>env_name\Scripts\activate
```

รูปที่ ข.7 คำสั่งการเข้าใช้งาน environment

```
(env_name) C:\Users\Admin\Documents\GitHub\Project\Game>
```

รูปที่ ข.8 แสดงชื่อ environment เมื่อเข้าใช้งาน

```
[env_name] C:\Users\Admin\Documents\GitHub\Project\Game>pip install -r requirement-pygame.txt
```

รูปที่ ข.9 คำสั่งในการติดตั้ง library ใน requirement

```
(env_name) C:\Users\Admin\Documents\GitHub\Project\Game>pip install -r requirements-gpu.txt
```

รูปที่ ข.10 คำสั่งในการติดตั้ง library ใน requirement-gpu

5. เมื่อติดตั้ง library ครบแล้วในขั้นตอนต่อไป จะเป็นการนำค่า weights ที่ได้จากการเทรน ไปเข้า Model ใน Yolo เพื่อทำการ โหลดและเชฟ Model โดยพิมพ์คำสั่ง python save\_model.py --weights ./data/yolov4-custom3\_final.weights --output ./checkpoints/yolov4-416 --input\_size 416 --model yolov4 เมื่อใช้คำสั่งข้างต้นสำเร็จแล้ว สามารถเข้าตัวระบบ ได้โดยการพิมพ์ py launch\_game.py

```
(env_name) C:\Users\Admin\Documents\GitHub\Project\Game>python save_model.py --weights ./data/yolov4-custom3_final.weights --output ./checkpoints/yolov4-416 --input_size 416 --model yolov4
```

รูปที่ ข.11 การใส่คำสั่งในการ โหลดและเชฟ Model ใน Yolo

```
(env_name) C:\Users\Admin\Documents\GitHub\Project\Game>py launch_game.py
```

รูปที่ ข.12 คำสั่งเพื่อเปิดระบบ

## ประวัติผู้เขียน

**ชื่อ – นามสกุล** นาย เนตริมวงศ์ จ้าหอนองโพธิ์

**รหัสนักศึกษา** 60070131

**วัน เดือน ปีเกิด** 9 สิงหาคม 2541

**ประวัติการศึกษา**

วุฒิ ม.6 น้ำมินตราชินุทิศ เตรียมอุดมศึกษาพัฒนาการ



นักศึกษาชั้นปีที่ 4 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**ภูมิลำเนา** 119/68 หมู่ 2 ต.บางเสาธง อ.บางเสาธง จ.สมุทรปราการ

**เบอร์โทรศัพท์** 087-6814631 **E-Mail** chalerimwong.j@gmail.com

**สาขาที่จบ** วิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ **รุ่นที่** 1 ปีการศึกษาที่จบ 2564

**ชื่อ – นามสกุล** นาย นนทกร มาศรี

**รหัสนักศึกษา** 60070144

**วัน เดือน ปีเกิด** 9 กันยายน 2541

**ประวัติการศึกษา**

วุฒิ ม.6 โรงเรียนบดินทรเดชา (สิงห์ สิงหนาท) ๒



นักศึกษาชั้นปีที่ 4 คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

**ภูมิลำเนา** 126/239 ช.รามอินทรา 40 แยก 8 แขวงนวลดันทร์ เขตบึงกุ่ม กรุงเทพมหานคร 10230

**เบอร์โทรศัพท์** 083-7722253 **E-Mail** 60070144@it.kmitl.ac.th

**สาขาที่จบ** วิทยาการข้อมูลและการวิเคราะห์เชิงธุรกิจ **รุ่นที่** 1 ปีการศึกษาที่จบ 2564

# แบบจำลองหุ่นยนต์สำหรับพัฒนาทักษะการเขียนโปรแกรมในเด็ก

## ประถม โดยใช้ตัวต่อที่สร้างด้วยกระดาษเป็นชุดคำสั่ง

เฉลิมวงศ์ จ้าหหนองโพธิ์<sup>1</sup> และ หนทกร มาศรี<sup>2</sup>

<sup>1</sup> คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

<sup>2</sup> คณะเทคโนโลยีสารสนเทศ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง กรุงเทพฯ

Emails: 60070131@it.kmitl.ac.th, 60070144@it.kmitl.ac.th

### บทคัดย่อ

ในปัจจุบันการเรียนเขียนโปรแกรมในเด็กระดับประถมศึกษาที่สามารถทำให้เด็กมีส่วนรวมในการเรียนเป็นไปได้ค่อนข้างยากเนื่องจากการที่ให้เด็กนักเรียนนั่งเรียนหน้าคอมพิวเตอร์โดยที่อาจารย์สอนหน้าห้องอาจจะทำให้อาจารย์ดูแลหรือสอนเด็กได้ไม่ทั่วถึงรวมถึงเด็กอาจจะไม่ได้สนใจกับการสอนของอาจารย์ แต่กลับสนใจอินเทอร์เน็ตหรือเล่นเกมแทน ผู้วิจัยได้มองเห็นปัญหาและได้ศึกษา กิจกรรมอันปลึ้ก (*Unplug*) เป็นการสอนเกี่ยวกับเรื่องโปรแกรมมิ่งโดยไม่ต้องใช้คอมพิวเตอร์ แต่เป็นการใช้อุปกรณ์ที่หาได้ทั่วไปที่สามารถนำมาเล่นกิจกรรมได้และยังทำให้เด็กเริ่มต้นเข้าใจในโปรแกรมมิ่งไปในตัวด้วย

ทางผู้วิจัยจึงได้สร้างระบบแบบจำลองหุ่นยนต์นี้ขึ้นโดยมีวัตถุประสงค์เพื่อเป็นการสนับสนุนการเรียนรู้แบบอันปลึ้กในเด็กระดับประถมศึกษา เน้นให้เด็กมีกิจกรรมร่วมกันโดยการที่ต่อจิ๊กซอว์ที่เป็นคำสั่งโปรแกรมมิ่งและใช้กล้องถ่ายภาพดูจิ๊กซอวนั้นเพื่อแปลงเป็นคำสั่งและแสดงผลการทำงานบนระบบ ทำให้เด็กได้แลกเปลี่ยนความคิด มีกิจกรรมร่วมกันและได้เข้าใจหลักการทำงานของคอมพิวเตอร์ โปรแกรมมิ่งไปในตัว

คำสำคัญ – โปรแกรมมิ่ง; กิจกรรมอันปลึ้ก (*Unplug*); ประถมศึกษา; จิ๊กซอว์; คอมพิวเตอร์

### 1. บทนำ

ปัจจุบันการเรียนรู้แบบอันปลึ้ก[1] มีประโยชน์มากในการสอนเด็ก ในระดับประถมศึกษาให้มีกระบวนการคิด เข้าใจหลักการพื้นฐาน ของตรรกศาสตร์และคอมพิวเตอร์ อีกทั้งสามารถเล่นได้ทุกเวลา มีเพียงอุปกรณ์ที่หาได้ทั่วไป เช่น กระดาษ แก้วน้ำ ดินสอ เป็นต้น ซึ่งจะสนับสนุนการเรียนโปรแกรมมิ่งในเด็กประถม การที่เด็กต้องเขียนโปรแกรมคอมพิวเตอร์ด้วยตัวเองอาจทำให้เด็กเบื่อหรือไม่ สอนใจในการเรียน จึงมีแนวคิดที่จะสร้างเครื่องมือที่ใช้ช่วยสนับสนุนการเรียนการสอนโปรแกรมมิ่ง ที่ให้เด็กร่วมกันต่อจิ๊กซอว์เพื่อแก้ปัญหาจากโจทย์ที่ได้ออกแบบมา ทำงานผ่านการถ่ายรูปจิ๊กซอว์ และแสดงผลในระบบบนคอมพิวเตอร์

### 2. ทฤษฎีบท และเทคโนโลยีที่เกี่ยวข้อง

#### 2.1 ทฤษฎีบท

##### 2.1.1 กิจกรรมการเรียนรู้แบบอันปลึ้ก (*Unplug*)

เป็นการเรียนรู้โปรแกรมมิ่งและหลักการทำงานในคอมพิวเตอร์ โดยไม่ใช้คอมพิวเตอร์ แต่จะเป็นการใช้กิจกรรมต่าง ๆ เป็นสื่อกลางให้เด็กหรือผู้เล่นได้มีการคิดอย่างเป็นระบบ วิเคราะห์การแก้ไขปัญหา ฝึกทักษะการสื่อสาร จึงทำให้การเรียนรู้แบบอัน

ปลึ้ก(*Unplug*) ตอบสนองผู้เรียนในระดับประถมศึกษาได้ดีกว่าการเรียนรู้โดยใช้คอมพิวเตอร์จริง

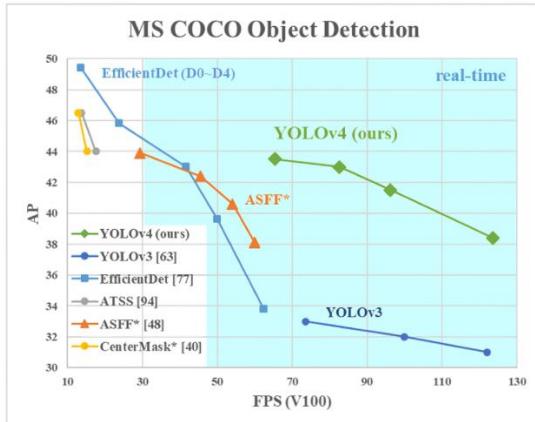
##### 2.1.2 การตรวจจับวัตถุ (*Object Detection*) [2]

เป็นเทคโนโลยีทางคอมพิวเตอร์ ใช้สำหรับการตรวจจับวัตถุจากรูปภาพโดยสามารถกำหนดวัตถุเองได้ เช่น มนุษย์ รถยนต์ สุนัข แมว เป็นต้น ซึ่ง Object Detection นั้นยังสามารถแยกไปได้อีกหลายแขนง เช่น Face Detection [3] เป็นการตรวจจับใบหน้าจากรูปหรือวิดีโอ หรือ Pedestrian Detection [4] เป็นการตรวจจับวัตถุที่อยู่หน้ารถยนต์เพื่อเตือนคนขับให้เบรกก่อนที่จะชน และสามารถนำไปประยุกต์ใช้ในหลากหลายงาน เช่น ใช้สำหรับรักษาความปลอดภัย เพื่อตรวจจับบุคคลที่ต้องสงสัย หรือการสั่งให้รถยนต์สามารถซับเคลื่อนเองได้โดยไม่ต้องใช้คนขับ เป็นต้น

##### 2.1.3 YOLO (*You only look once*) [5]

เป็นเทคโนโลยีในการตรวจจับวัตถุ (*Object Detection*) แบบเรียลไทม์ (realtime) ที่จำนวนภาพในวิดีโอด้วย 30 ภาพต่อ 1 วินาที (30 FPS) ซึ่งมีความแม่นยำ 57.9% ในการทดสอบของ COCO test-dev [6] การทำงานของ YOLO คือจะตรวจจับวัตถุที่เราต้องการซึ่งอยู่ใน dataset ที่สร้างขึ้นและนำไปเทรน (train)

จนมีความแม่นยำมากพอ และเมื่อใส่ input ที่เป็นรูปภาพให้และ YOLO จะทำการตรวจสอบ ตีกรอบวัตถุและบอกว่าวัตถุนั้นคืออะไร ตามชื่อของ Class ที่ได้ทำการใส่ใน Dataset ในช่วงแรกของการทำโครงการผู้วิจัยได้ใช้ YOLO v3 [7] ต่อมาในช่วงเดือน พฤษภาคม ค.ศ.2020 ได้มีการเปิดตัว YOLO v4 [8] ที่สามารถคำนวณได้แบบคุ่นหาน ซึ่งจะเพิ่มความแม่นยำกว่า YOLO v3



รูปที่ 1. แสดงการเปรียบเทียบ YOLO v4 ,YOLO v3 และ EfficientDet ผู้วิจัยจึงเลือกใช้ YOLO v4 ที่มีความแม่นยำในการตรวจจับวัตถุ สูงกว่า YOLO v3

## 2.2 เทคโนโลยีที่เกี่ยวข้อง

### 2.2.1 Python [9]

เป็นภาษาทางคอมพิวเตอร์ระดับสูง ถูกเปิดตัวในปี ค.ศ.1991 Python เป็นภาษาที่ค่อนข้างง่ายในการเข้าใจ มี Library ให้ใช้งาน ได้หลากหลายและครอบคลุมการทำงาน โดยสามารถสร้างแอปพลิเคชัน สร้าง Gui (Graphic user interface) สร้างเกม หรือเขียนเพื่อทำการวิเคราะห์ข้อมูลในด้านData Science ได้ เช่น Classification, Regression, Clustering เป็นต้น

### 2.2.2 Pygame [10]

เป็น Library หนึ่งในภาษา Python มีฟังก์ชันต่างๆในการเขียนและพัฒนาเกม สามารถสร้างเกมตั้งแต่เกมแก้ปัญหาการวางแผนล็อกต่างๆไปจนถึงเกมแนว Civilization หรือ 3D Shooters ซึ่งมีกราฟิกที่สวยงาม

### 2.2.3 OpenCV (Open Source Computer Vision) [11]

เป็น Library ใน Machine Learning ที่สามารถใช้งานได้บนภาษา Python จะเน้นไปทางการเรียนรู้จากรูปภาพโดยผ่านคอมพิวเตอร์ และแสดงผลแบบเรียลไทม์ จึงทำให้สามารถประมวลผลบน GPU ได้

### 2.2.4 Web Camera (Webcam, เว็บแคม)

เป็นอุปกรณ์ถ่ายภาพหรือภาพเคลื่อนไหวพร้อมกับแสดงผลบนคอมพิวเตอร์

### 2.2.5 Darknet [12]

เป็นเฟรมเวิร์ค Neural Network ถูกพัฒนาโดยบริษัท Nvidia มีการติดตั้งที่ง่าย สามารถใช้งานได้ทั้งบน CPU และ GPU ซึ่งทำให้มีการประมวลผลที่รวดเร็ว มีการนำมาใช้งานทางด้าน Object Detection และ Image Classification ที่สามารถรองรับการทำงานที่หลากหลาย

## 3. การออกแบบและการพัฒนา

### 3.1 การออกแบบส่วนของระบบ

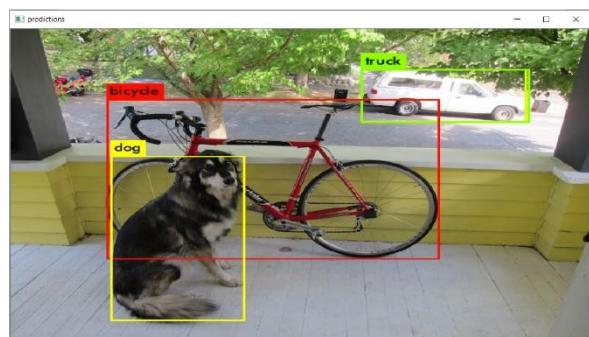
ผู้วิจัยได้ออกแบบระบบด้วยภาษา python โดยใช้ library Pygame เพื่อสร้างเป็นเกมให้ดูน่าสนใจ และแสดงดังรูปที่ 2 ขอแสดงผลการทำงานจะอยู่ทางซ้าย ซึ่งมีตัว robot, objective และสิ่งกีดขวาง ตรงกลางคำว่า Command คือการทำงานของ input ที่ได้รับจากรูปภาพ คำว่า SHOW COMMAND คือสามารถกดเพื่อดูคำอธิบายรูปทรงของจีกซอฟ์ เมื่อผู้ใช้งานกดปุ่ม START จะเริ่มการทำงานทั้งหมดของระบบ การกด RESTART จะเริ่มการทำงานของระบบใหม่ทั้งหมด กด EXIT เพื่อเป็นการออกจากระบบ และทางซ้ายล่างจะเป็นด้านต่างๆที่เลือกได้

### 3.2 ออกแบบรูปทรงจีกซอฟ์

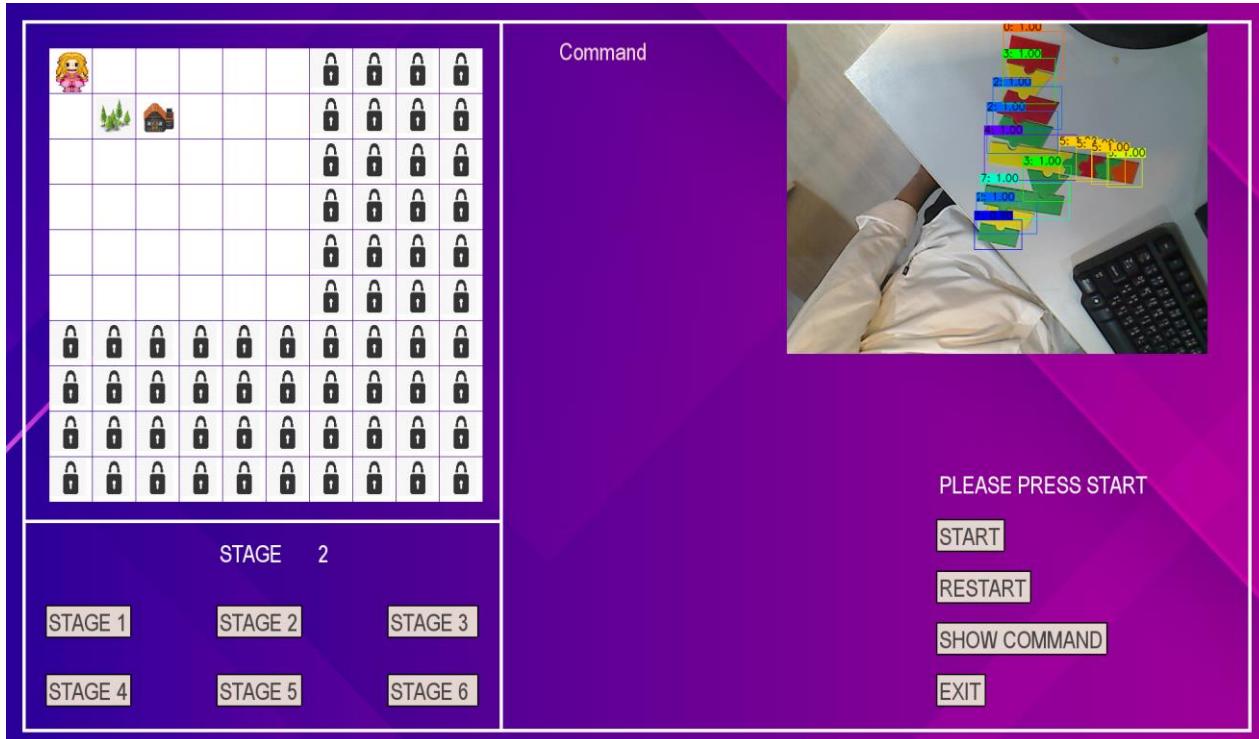
ผู้วิจัยได้ทำการออกแบบรูปทรงจีกซอฟ์ทั้งหมด 15 รูปทรง โดยที่แต่ละรูปทรงจะให้แทนคำสั่งทางโปรแกรมมิ่งที่แตกต่างกันออกไป

### 3.3 ติดตั้ง YOLO (You only look once)

ผู้วิจัยได้ทำการติดตั้ง YOLO ที่มี Darknet เป็นเฟรมเวิร์คในการเทรน โดยทำการติดตั้งบน Windows และได้ทดลองใช้จาก COCO Dataset ที่ได้ทำการเทรนไว้ 80 Class ใช้ Weight ของ YOLO v4 ที่ได้จากการトレน เพื่อทดสอบว่าติดตั้งสำเร็จ ดังแสดงตามรูปที่ 3



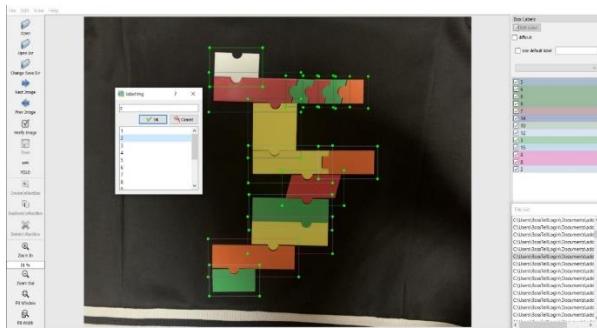
รูปที่ 3. ผลลัพธ์ที่ได้จากการทดสอบ YOLO



รูปที่ 2. ระบบแบบจำลองหุ่นยนต์

### 3.3 ทำการสร้าง Label Image

Label Image ทำขึ้นไว้เพื่อบอกตำแหน่งของ Object หรือจิ๊กซอว์ ในรูปภาพที่ได้ทำการถ่ายว่าอยู่ในตำแหน่งใดของรูปภาพโดยจะต้องทำการครอบจิ๊กซอว์ซึ่งเรียกว่า Bounding box ดังแสดงตามรูปที่ 4 จากนั้นจะได้ค่ามา 5 ค่าคือ Class ID, Center x, Center y, Width, Height ใช้สำหรับการนำเข้าไปเทรนใน YOLO ควบคู่กับรูปภาพ



รูปที่ 4. ตัวอย่างการทำ Label Image

## 4. ผลการดำเนินงาน

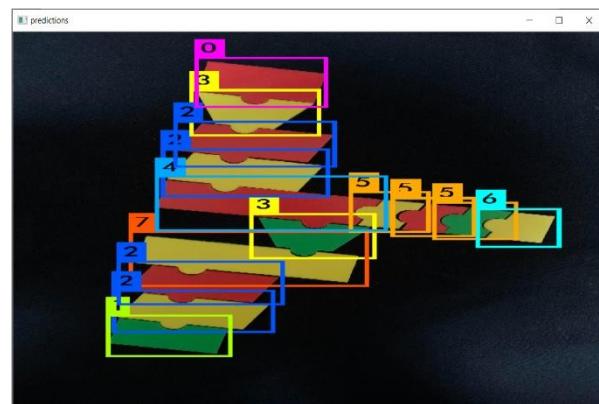
### 4.1 ผลการทดลองตรวจจับรูปภาพ

ผู้วิจัยได้ทำการเทรนข้อมูลรูปภาพจิ๊กซอว์ โดยใช้ทั้ง YOLO v3 และ YOLO v4 โดยอธิบายตัวแปรในการเทรนรูปภาพ ดังแสดงตามตารางที่ 1

ตารางที่ 1. อธิบายตัวแปรในการเทรนรูปภาพ

| ชื่อตัวแปร                          | จำนวนค่าที่ใช้ |
|-------------------------------------|----------------|
| จำนวนคลาส                           | 15 คลาส        |
| จำนวนรูปที่ใช้เทรนใน YOLO v3        | 2,658 รูป      |
| จำนวนรูปที่ใช้เทรนใน YOLO v4        | 3,398 รูป      |
| จำนวนรอบที่เทรนใน YOLO v3           | 20,000 รอบ     |
| จำนวนรอบที่เทรนใน YOLO v4           | 30,000 รอบ     |
| จำนวนรูปภาพที่สุมเข้าไปเทรนใน 1 รอบ | 64 รูป         |

เมื่อผู้วิจัยทำการเทรนสำเร็จ ได้ลองทดสอบความถูกต้องในการตรวจจับรูปภาพจิ๊กซอว์โดยเปรียบเทียบระหว่าง YOLO v3 และ YOLO v4 จึงได้เห็นผลลัพธ์ว่า YOLO v4 สามารถตรวจจับรูปจิ๊กซอว์ได้แม่นยำกว่า



รูปที่ 5. ผลลัพธ์จากการตรวจจับรูปภาพจิ๊กซอว์โดยใช้ YOLO v4

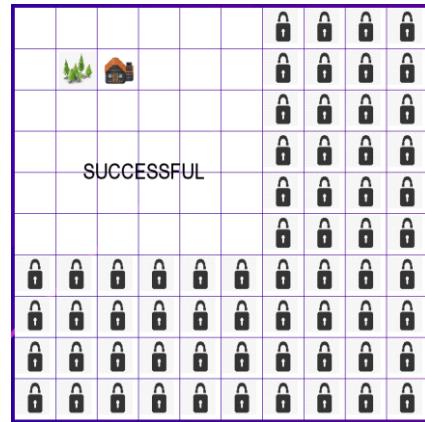
#### 4.2 การทดลองของระบบ

หลังจากผู้ใช้ได้เทอร์มินัลฯ ทั้งหมดแล้ว ขั้นตอนต่อไปคือ การนำผลที่ได้ไปใช้งานร่วมกับระบบโดยมีขั้นตอนการทำงานคือ 1) เปิดระบบขึ้นมาโดยผู้ใช้งานสามารถกดปุ่ม SHOW COMMAND เพื่อทำการดูรูปภาพคำอธิบายรูปทรงจิ๊กซอว์ ดังแสดงตามรูปที่ 5 2) ทำการเลือกด่านเพื่อใช้สำหรับแก้โจทย์โดยในแต่ละด่านมีความยากและง่ายแตกต่างกัน 3) ทำการต่อจิ๊กซอว์ เพื่อแก้ปัญหาด้านที่ทำการเลือก 4) กดปุ่ม START เพื่อทำการถ่ายรูปจิ๊กซอว์ที่ได้ทำการต่อ จากนั้นทำการตรวจสอบจิ๊กซอว์ที่อยู่ในรูปเพื่อแปลงเป็นคำสั่งและแสดงผลออกมาให้ Robot เดินตามคำสั่งที่ได้ 5) ผู้ใช้งานสามารถกดปุ่ม RESTART เพื่อทำการลบคำสั่งที่ได้ก่อนหน้าและรีเซ็ตด่านให้ Robot กลับไปยังจุดเริ่มต้น



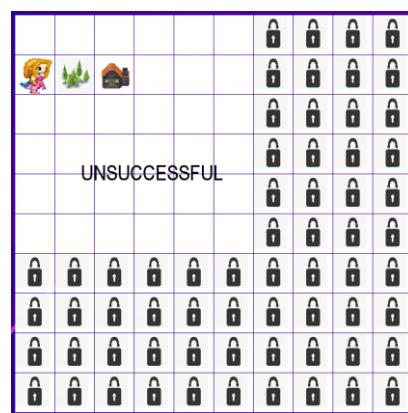
รูปที่ 6. อธิบายรูปทรงจิ๊กซอว์ในระบบ

หากทำการต่อจิ๊กซอว์ได้ถูกต้อง เมื่อกดปุ่ม START ระบบจะทำการแสดงผลสามารถทำให้ Robot เดินถึงเป้าหมายได้และจะแสดงข้อความ “SUCCESSFUL” เพื่อเป็นการบอกว่าผ่านด่าน ดังแสดงตามรูปที่ 7



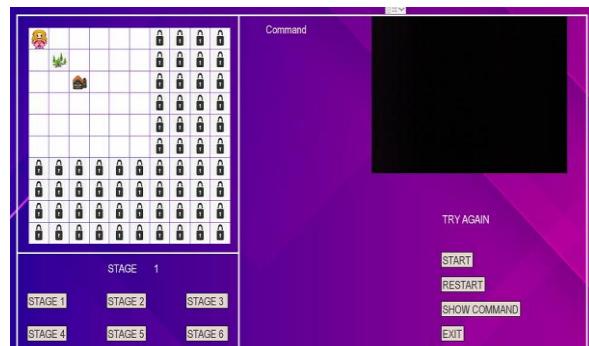
รูปที่ 7. แสดงข้อความ “SUCCESSFUL” เมื่อ Robot เดินถึงเป้าหมาย

แต่ถ้าหากต่อจิ๊กซอว์ผิด ทำให้ Robot เดินไม่ถึง Objective จะแสดงข้อความ “UNSUCCESSFUL” เพื่อบอกว่ายังไม่ผ่านด่านให้ทำการต่อจิ๊กซอว์ใหม่ ดังแสดงตามรูปที่ 8

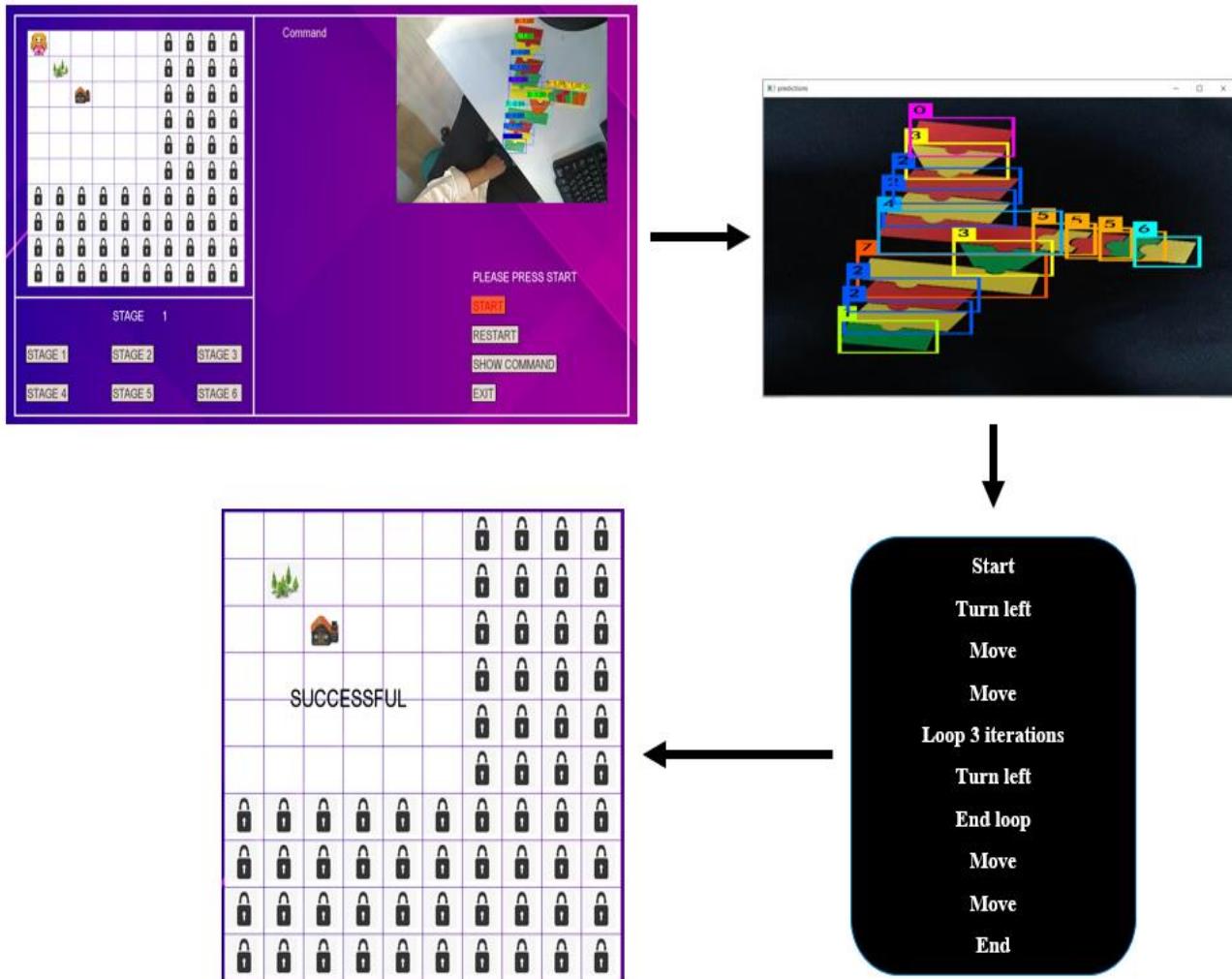


รูปที่ 8. แสดงข้อความ “UNSUCCESSFUL” เมื่อ Robot เดินไม่ถึงเป้าหมาย

และหากไม่มีจิ๊กซอว์ที่ต้องการไว้ตั้งกล่อง เมื่อทำการกด START จะแสดงข้อความ “TRY AGAIN” ตรงช่อง Command โดยข้อความนี้จะหยุดแสดงก์ต่อเมื่อ นำจิ๊กซอว์ที่ได้ทำการต่อไว้มาวางไว้ที่กล่องเว็บแคม ดังแสดงตามรูปที่ 9



รูปที่ 9. แสดงข้อความ “TRY AGAIN” เมื่อไม่พบจิ๊กซอว์ในกล่อง



รูปที่ 10. สรุปกระบวนการทำงานของระบบ

จากรุปที่ 10 ผู้จัดได้นำแต่ละขั้นตอนในการทำงานของระบบ มาอธิบายเป็นรูปภาพ เรียนเรียงการทำงานให้เห็นภาพง่ายขึ้น

## 5. บทสรุป

### 5.1 สรุปผลการดำเนินงาน

ผู้จัดได้ทำการศึกษาโมเดลในการตรวจรูปภาพหรือ Object Detection เพื่อหาโมเดลที่คิดว่าสามารถนำมาใช้งานได้ในระยะเวลาที่กำหนด มีการเปรียบเทียบผลลัพธ์เพื่อใช้โมเดลที่มีความแม่นยำที่สุด

ในส่วนของตัวระบบที่ผู้จัดได้ทำการพัฒนาให้สามารถใช้งานร่วมกับโมเดลที่ได้เลือกมาโดยมีวัตถุประสงค์คือสามารถนำไปช่วยให้การเรียนโปรแกรมมีง่ายขึ้น ซึ่งระบบที่ถูกพัฒนาขึ้นนี้สามารถเลือกโจทย์เพิ่มโจทย์ ออกแบบและแก้ไขโจทย์เองได้ เมื่อใช้งานระบบจะสามารถเช็คคำสั่งหรือ Command ได้ สามารถต่อจิ๊กซอว์ไป เช็คคำสั่งไปด้วยได้เพื่อให้เด็กได้ลองผิดลองถูก มีการตรวจจับจิ๊กซอว์แบบเรียลไทม์แสดงผ่านระบบ เมื่อกดปุ่ม START

จะเห็นการทำงานของจิ๊กซอว์ที่ต่อไว้ เพื่อให้เด็กเข้าใจการทำงานของคำสั่งในการโปรแกรมมิ่ง ส่วนของรูปภาพ Robot, Objective หรือสิ่งกีดขวางสามารถเปลี่ยนรูปได้หากนำไปใช้งานจริง

ในส่วนของโมเดล Object Detection ได้เลือกนำ YOLO v4 มาใช้งาน เนื่องจากมีความสามารถในการตรวจจับวัตถุจากรูปภาพ วิดีโอ หรือแบบเรียลไทม์ ได้แม่นยำกว่าใน YOLO v3 มาก ผู้จัดได้ทำการเทรนและประยุกต์ผลลัพธ์ของทั้ง 2 โมเดล มีการเพิ่มรูปภาพเข้าไปเทรนเพิ่มเพื่อเพิ่มความแม่นยำ และลดการเกิด Overfitting

ผู้จัดได้ทำการนำจิ๊กซอว์ในรูปทรงที่ยังไม่ได้ใช้งานนำไปเทรนร่วมกับรูปทรงที่นำมาใช้งาน เพื่อที่ในอนาคตสามารถเพิ่มคำสั่งได้โดยที่ไม่ต้องเสียเวลาในการออกแบบและเทรนข้อมูลใหม่

## 6 สิ่งที่จะพัฒนาต่ออยอดในอนาคต

ทำการเพิ่มคำสั่งให้กับรูปทรงจิ๊กซอว์ที่เหลืออยู่และเพิ่มรูปแบบของโจทย์ให้มีความหลากหลายมากขึ้นและออกแบบระบบ

ให้สามารถเพื่อให้เหมาะสมกับการใช้งานของเด็ก

## เอกสารอ้างอิง

- [1] ฉัตรพงศ์ ชูแสงนิล (2019). เกมการเรียนรู้แบบ Unplug. [ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/36V6dHb>
- [2] Keng Surapong (2020). Object Detection คืออะไร บทความสอน AI ตรวจจับวัตถุ. [ออนไลน์] เข้าถึงได้จาก: <https://www.bualabs.com/archives/3453>
- [3] Faen Zhang, Xinyu Fan, Guo Ai, Jianfei Song, Yongqiang Qin, Jiahong Wu (2019). Accurate Face Detection for High Performance. [ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/1905.01585v3.pdf>
- [4] Irtiza Hasan, Shengcai Liao, Jinpeng Li, Saad Ullah Akram, Ling Shao (2020). Pedestrian Detection: The Elephant In The Room. [ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/2003.08799v6.pdf>
- [5] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi (2016). You Only Look Once: Unified, Real-Time Object Detection [ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/1506.02640.pdf>
- [6] Tsung-Yi Lin, Michael Maire, Serge Belongie., et al. (2015). Microsoft COCO: Common Objects in Context. [ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/1405.0312.pdf>
- [7] Joseph Redmon, Ali Farhadi (2018). YOLOv3: An Incremental Improvement. [ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/1804.02767.pdf>
- [8] Alexey Bochkovskiy , Chien-Yao Wang , Mark Liao (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. [ออนไลน์] เข้าถึงได้จาก: <https://arxiv.org/pdf/2004.10934.pdf>
- [9] Sarayut Nonsiri, PhD. ภาษาโปรแกรม Python คืออะไร?. [ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/34cOJ7F>
- [10] Nuttakan Chuntra (2018). มาสร้างแอปด้วย Pygame กันเถอะ (ขั้นตอนการติดตั้ง). [ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/3dgTiVd>
- [11] Nuttakan Chuntra (2018). OpenCV คืออะไร?. [ออนไลน์] เข้าถึงได้จาก: <https://bit.ly/3cZo5C1>
- [12] Joseph Redmon (2013 - 2016). Darknet: Open Source Neural Networks in C. [ออนไลน์] เข้าถึงได้จาก: <https://pjreddie.com/darknet/>