การพัฒนาระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อน

สิริทัศน์ เลิศตระกูลถาวร

สารนิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต
บัณฑิตวิทยาลัย สาขาวิชาเทคโนโลยีสารสนเทศ
สถาบันเทคโนโลยีไทย-ญี่ปุ่น
ปีการศึกษา 2563

DEVELOPMENT OF A NEST COUNTING SYSTEM WITH A YOLO OBJECT DETECTION VIA A THERMAL IMAGING CAMERA

Siritat Lerdtrakultavorn

A Term Paper Submitted in Partial Fulfillment of the Requirements
for the Degree of Master of Science Program in Information Technology
Graduate School
Thai-Nichi Institute of Technology
Academic Year 2020

หัวข้อสารนิพนธ์	การพัฒนาระบบนับจำนวนนกแอ่นกินรับด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อน
โดย	สิริทัศน์ เลิศตระกูลถาวร
สาขาวิชา	เทคโนโลยีสารสนเทศ
อาจารย์ที่ปรึกษาสารนิพนธ์	รองศาสตราจารย์ ดร. รัตติกร วรากูลศิริพันธุ์

บัณฑิตวิทยาลัย สถาบันเทคโนโลยีไทย-ญี่ปุ่น อนุมัติให้นับสารนิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญามหาบัณฑิต

.....คณบดีบัณฑิตวิทยาลัย

(รองศาสตราจารย์ ดร. พิชิต สุขเจริญพงษ์)

วันที่.....เดือน.....พ.ศ.....

คณะกรรมการสอบสารนิพนธ์

..... ประธานกรรมการ

(รองศาสตราจารย์ ดร. อรรณพ หมั่นสกุล)

..... กรรมการ

(ดร. สรมย์พร เจริญพิทย์)

..... อาจารย์ที่ปรึกษาสารนิพนธ์

(รองศาสตราจารย์ ดร. รัตติกร วรากูลศิริพันธุ์)

สิริทัศน์ เลิศตระกูลถาวร: การพัฒนาระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อน. อาจารย์ที่ปรึกษา : รองศาสตราจารย์ ดร. รัตติกร วรากูลศิริพันธุ์, 59 หน้า.

รังนกถือเป็นอาหารที่ได้รับความนิยมมาอย่างยาวนาน โดยเฉพาะอย่างยิ่งชาวจีน เนื่องจากมีคุณค่าทางสารอาหารที่ได้ผ่านการวิจัยมาแล้ว อาทิเช่น Glycoprotein (ไกลโคโปรตีน) ถือเป็นส่วนประกอบสำคัญในโครงสร้างร่างกายมนุษย์ ทั้งในเนื้อเยื่อและกระดูก ทำหน้าที่เสมือนสารหล่อลื่นของเซลล์, อีพิเดอร์มอล โกรท แฟคเตอร์ Epidermal Growth Factor (EGF) เป็นปัจจัยที่สำคัญมากในการเจริญเติบโตของร่างกายมนุษย์ ซึ่งมีหน้าที่กระตุ้นการพัฒนาแบ่งเซลล์ และเยื่อบุต่างๆ ของมนุษย์ได้อย่างมีประสิทธิภาพ เป็นต้น ในแต่ละปีประเทศจีนมีมูลค่าการนำเข้ารังนกเป็นจำนวนกว่าพันล้านบาท โดยนำเข้าจากประเทศอินโดนีเซีย มาเลเซีย และไทย เป็นหลัก ส่งผลให้ธุรกิจฟาร์มนกแอ่นกินรังมีการเติบโตและกระจ่ายตัวอยู่ทั่วประเทศไทยโดยเฉพาะภาคใต้ ซึ่งมีอาคารหรือบ้านนกแอ่นกินรังกว่า 10,000 หลัง ในการประกอบธุรกิจฟาร์มนกแอ่นกินรังมีความจำเป็นต้องควบคุมปัจจัยต่างๆ ไม่ว่าจะเป็นอุณหภูมิ ความชื่น ความสะอาด เชื้อโรค หรือศัตรูตามธรรมชาติ ทางผู้วิจัยได้นำเสนอการพัฒนาระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อน เพื่อเป็นส่วนช่วย หรือเครื่องมืออำนวยความสะดวกให้แก่เกษตรกรฟาร์มนกแอ่นกินรัง ในการตรวจนับนกแอ่นกินรังภายในอาคารฟาร์มนกแอ่นกินรัง ซึ่งเป็นระบบปิด มีแสงสว่างน้อย ใช้สายตาในการตรวจนับค่อยข้างลำบาก และขาดความแม่นยำ ทางผู้วิจัยได้นำเทคโนโลยี IoT โดยนำบอร์ด Raspberry Pi มาทำงานร่วมกับกล่องถ่ายภาพความร้อน FLIR Lepton 3.5 และนำภาพที่ถ่ายได้มาผ่านกระบวนการ Image Detection และอาศัยอัลกอริทึม YOLOv3 เพื่อตรวจนับจำนวนนกบนภาพ ซึ่งผู้วิจัยได้จำลองการถ่ายภาพนกหงส์หยกในกรงด้วยระยะ 50 ซม. 1 เมตร 1.5 เมตร 2 เมตร และ 2.5 เมตรตามลำดับ โดยผลสรุปจากการทดลองระยะ 1 เมตรเป็นระยะถ่ายภาพที่เหมาะสมที่สุด เพราะระบบตรวจนับจำนวนนกด้วยกล้องถ่ายภาพความร้อนสามารถทำการ Detection และตรวจนับจำนวนนกได้แม่นยำที่สุด จากนั้นจะได้ส่งข้อมูลการตรวจนับไปยังแอพพลิเคชั่นไลน์ เพื่อทำการแจ้งเตือนและได้ทำการบันทึกข้อมูลดังกล่าว พร้อมนำมาข้อมูลมาทำรายงานในรูปแบบวิชวลไลเซชันเพื่อให้เกษตรกรง่ายต่อการดูข้อมูลที่เกิดขึ้น

บัณฑิตวิทยาลัย ลายมือชื่อนักศึกษา .....
สาขาวิชา เทคโนโลยีสารสนเทศ ลายมือชื่ออาจารย์ที่ปรึกษา .....
ปีการศึกษา 2563

SIRITAT LERDTRAKULTAVORN: DEVELOPMENT OF A NEST COUNTING SYSTEM WITH A YOLO OBJECT DETECTION VIA A THERMAL IMAGING CAMERA. ADVISOR: ASSOC. PROF. DR. RUTTIKORN VARAKULSIRIPUNTH, 59 PP.

Bird's nest has long been a popular food, especially in the Chinese, because of its well-researched digestive products, such as Glycoprotein it is an essential ingredient in both human bodies and bones. Epidermal Growth Factor (EGF) is a very important root in body growth which is useful for development, formulation, and growth. Various human beings effectively. Each year, China has a value of billions of bird's nest imports. By imported from Indonesia In Malaysia and Thailand, the birds nesting business has grown and spread throughout Thailand, especially in the South. Which has more than 10,000 buildings or nesting birdhouses? In operating the nesting bird farm business, it is necessary to control various factors. Whether it's temperature, humidity, cleanliness, germs, or natural enemies. The researcher himself has proposed the development of a YOLO Object Detection system through a thermal imaging camera. To help or tools to facilitate farmers, bird-eating farms, nests in the counting of nest-eating birds in the building of a bird's nest Which is a closed system Low light Difficult to use the eyes to count the researcher has applied IoT technology by using a Raspberry Pi board to work with a Lepton 3.5 thermal imaging box. And taken pictures taken through the Image Detection process using the YOLO V.3 algorithm to count the number of birds on the image. The researcher modeled the photograph of budgies in a cage at a distance of 50 cm., 1 m, 1.5 m, 2 m, and 2.5 m respectively. Because the bird counting system with a thermal imaging camera can detect and count birds with the most accuracy. Then sent the counting data to the LINE application to notify and record such information and the data was used to make a report in a visualization format to make it easier for farmers to view the information.

Graduate School Student's Signature.....

Field of Study Information Technology Advisor's Signature.....

Academic Year 2020

กิตติกรรมประกาศ

การทำสารนิพนธ์นี้จะสำเร็จลุล่วงไปได้ด้วยดีนั้น ผู้วิจัยขอกราบขอบพระคุณในความกรุณาของรองศาสตราจารย์ ดร. อรรณพ หมั่นสกุล ซึ่งได้สละเวลาในการให้คำแนะนำ คำปรึกษา การสนับสนุน รวมถึงตรวจสอบข้อบกพร่องทุกขั้นตอนและกำลังใจเต็มเปี่ยมตลอดระยะเวลาในการทำสารนิพนธ์ฉบับนี้

ทั้งนี้ผู้วิจัยขอกราบขอบพระคุณคณะกรรมการสอบทุกท่าน ได้แก่ ดร. สรมย์พร เจริญพิทย์, ดร. โสภณ มงคลลักษมี ที่ได้กรุณาให้ข้อเสนอแนะที่เป็นประโยชน์และช่วยตรวจสอบชี้ข้อบกพร่องของสารนิพนธ์ฉบับนี้ด้วยความเอาใจใส่ จนทำให้สารนิพนธ์ฉบับนี้สำเร็จและมีความสมบูรณ์

เหนือสิ่งอื่นใดผู้วิจัยขอขอบคุณครอบครัวของผู้วิจัยที่คอยให้กำลังใจ และให้การสนับสนุนในทุกๆ ด้านอย่างที่สุดเสมอมา

ผู้วิจัยหวังเป็นอย่างยิ่งว่า สารนิพนธ์ฉบับนี้จะเกิดประโยชน์ต่อผู้ที่มีความสนใจ สามารถนำไปต่อยอด องค์ความรู้เพื่อประยุกต์ใช้งานกับระบบการแนะนำการศึกษาเพื่อให้เกิดคุณค่าและสามารถใช้งานได้จริง

สิริทัศน์ เลิศตระกูลถาวร

สารบัญ

สารบัญ (ต่อ)

สารบัญตาราง

สารบัญรูป

สารบัญรูป (ต่อ)

นบทที่ 1
บทนำ

1.1 สภาวะความเป็นมา แนวทางเหตุผล และปัญหา

“รังนกนางแอ่น” ถือเป็นอาหารที่ได้รับการยกย่องมาอย่างยาวนานว่าเป็นอาหารเสริมที่ดีต่อสุขภาพ เป็นที่นิยมอย่างยิ่งโดยเฉพาะชาวจีน ซึ่งจากการวิจัยสารอาหารในรังนกนางแอ่นกินรังมีสาร Glycoprotein (ไกลโคโปรตีน) ช่วยเพิ่มภูมิคุ้มกันให้กับร่างกาย โดยช่วยกระตุ้นการแบ่งเซลล์ของเม็ดเลือดขาว และกระตุ้นการหลั่งสารภูมิต้านทานต่างๆ ได้แก่ อิมมูโนโกลบูลินเอ (IgA) ซึ่งเป็นแอนติบอดีหลักบริเวณเยื่อบุทางเดินอาหารและทางเดินหายใจ และ มีสารอีพิเดอร์มอล โกรท แฟคเตอร์ Epidermal Growth Factor (EGF) ซึ่งช่วยสร้างเซลล์ผิวใหม่ทดแทนเซลล์เก่าที่เสื่อมสลาย ช่วยซ่อมแซมเซลล์ผิวที่ถูกทำลาย ช่วยให้ผิวพรรณสดใส ดูอ่อนกว่าวัย [1]

ประเทศจีนเป็นตลาดนำเข้ารังนกที่ใหญ่ที่สุดของประเทศไทย ในปี 2563 ประเทศจีนได้นำเข้ารังนก จำนวน 214,018 กิโลกรัม มูลค่ากว่าหมื่นล้านบาท และสมาคมธุรกิจรังนกมณฑลกวางตุ้ง คาดการณ์ว่า ปี 2564 จีนจะมีปริมาณการนำเข้ารังนกราว 450 ตัน

ปี	มูลค่า (USD)	ปริมาณ (KG)
2560	145,390,935	77,899
2561	222,813,492	176,549
2562	320,482,640	214,922
2563 (ม.ค.-ก.ย.)	**331,710,229**	**214,018**

รูปที่ 1.1 มูลค่าการนำเข้ารังนกของประเทศจีน [2]

แนวโน้มดังกล่าวสอดคล้องกับการที่ผู้ประกอบการธุรกิจรังนกแอ่นกินรังในไทยที่มีปริมาณเพิ่มขึ้นเป็นลำดับ ปัจจุบันมีบ้านนกแอ่นทั่วประเทศกว่า 17,800 หลัง พื้นที่ที่มีบ้านนกแอ่นมากที่สุด คือ ภาคใต้ 10,000 หลัง ภาคตะวันออก 4,000 หลัง ขณะที่ ภาคกลาง ภาคตะวันออกเฉียงเหนือ และภาคเหนือ เริ่มมีแนวโน้มพบธุรกิจบ้านนกแอ่นมากขึ้นเช่นกัน โดยคาดการณ์ว่า บ้านนกแอ่นกินรังกว่า 3 หมื่นหลัง สามารถผลิตรังนกได้อยู่ที่ 100 – 150 ตันต่อปี [3] ซึ่งราคาประมูลรังนกดิบอยู่ประมาณ 25,000 บาทต่อกิโลกรัม ภาพรวมมีเม็ดเงินผ่านตลาดประมูล 3,000 กว่าล้านบาท/ปี ที่ผ่านมามีการส่งออกไปจีนได้ราคาประมาณ 80,000 บาทต่อกิโลกรัม แต่ชาวจีนรับไปขายต่อเพิ่มเป็น

300,000-400,000 บาทต่อกิโลกรัม ทำให้ส่วนภาพรวมตลาดรังนกทั่วประเทศมีมูลค่ารวมกว่า 15,000-16,000 ล้านบาท (รวมตลาดรังนกพร้อมดื่ม) กอปรกับขณะนี้ พ.ร.บ.สงวนและคุ้มครองสัตว์ป่า พ.ศ. 2562 ฉบับแก้ไขใหม่ ได้ประกาศในราชกิจจานุเบกษา ในมาตรา 14 ระบุให้ผู้เลี้ยงรังนกบ้านสามารถไปยื่นขอ “ใบอนุญาต” เก็บและครอบครองรังนกบ้าน หรือรังนกคอนโดได้อย่างถูกต้องตามกฎหมาย ซึ่งเป็นผลดีทำให้สามารถสร้างมูลค่าและผลประโยชน์ทางเศรษฐกิจให้กับประเทศไทยได้ไม่น้อย

รูปที่ 1.2 ปริมาณบ้านนกแอ่นกินรังในประเทศไทย [3]

อินเทอร์เน็ตสำหรับทุกสรรพสิ่ง (Internet Of Things) หมายถึง อุปกรณ์อิเล็กทรอนิกส์ต่างๆ ที่สามารถเชื่อมต่อกับอินเตอร์เน็ตได้ ไม่ว่าจะแบบมีสายหรือไร้สาย โดยทำการรวบรวมและส่งข้อมูลที่ได้รับจากเซนเซอร์ประเภทต่าง ๆ เช่น เซ็นเซอร์ตรวจวัดความชื้น เซ็นเซอร์ตรวจวัดอุณหภูมิ เซ็นเซอร์ตรวจวัดความสั่นสะเทือน เซ็นเซอร์ตรวจจับการเคลื่อนไหว ผ่านโครงข่ายอินเตอร์เน็ต ซึ่งสะดวก และรวดเร็วมากยิ่งขึ้น ทำให้เราสามารถสั่งการ และควบคุมการใช้งานอุปกรณ์อิเล็กทรอนิกส์ต่างๆ เหล่านั้นได้อีกด้วย

ในงานวิจัยนี้ได้นำเสนอ “การพัฒนาระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อน” โดยนำเทคโนโลยีอินเทอร์เน็ตสำหรับทุกสรรพสิ่ง (Internet of Things) เข้ามาช่วยในการตรวจนับจำนวนรังนกแอ่นกินรังที่ได้เข้ามาทำรังในอาคารหรือบ้านนก ด้วยการนำกล้องถ่ายภาพความร้อน มาเป็นเครื่องมือช่วยในการตรวจนับจำนวนนกแอ่น

กินรัง และนำข้อมูลจากการตรวจนับมาเป็นส่วนหนึ่งในการประเมินผลการดำเนินธุรกิจฟาร์มนกนางแอ่นกินรัง

1.2 วัตถุประสงค์ของการวิจัย

1.2.1 เพื่อออกแบบและใช้งานการตรวจนับจำนวนนกนางแอ่นกินรังจากเทคโนโลยีอินเทอร์เน็ตของสรรพสิ่ง และกล้องถ่ายภาพความร้อน

1.2.2 เพื่อนำข้อมูลจากกล้องถ่ายภาพความร้อน มาบ่งชี้นกนางแอ่นกินรัง

1.2.3 เพื่อผู้ประกอบการสามารถรับรู้จำนวนนกนางแอ่นกินรังผ่านอุปกรณ์คอมพิวเตอร์หรืออุปกรณ์พกพาได้สะดวกทุกที่ทุกเวลา

1.3 ขอบเขตของการวิจัย

1.3.1 ออกแบบอุปกรณ์สำหรับถ่ายภาพด้วยกล้องความร้อนซึ่งสามารถถ่ายภาพขนาด 160x120 พิกเซล และทำงานได้ในช่วงอุณหภูมิ -10 ถึง +80 องศาเซลเซียส

1.3.2 จำลองการถ่ายภาพด้วยกล้องตรวจจับความร้อนจากนกหงส์หยก 4 ตัว ในกรงขนาดกว้าง 60 เซนติเมตร ยาว 160 เซนติเมตร สูง 66 เซนติเมตร ในสภาพแวดล้อมใกล้เคียงกับในฟาร์มนกแอ่นกินรัง

1.3.3 นำภาพถ่ายจากกล้องถ่ายภาพความร้อนมาตรวจนับจำนวนนกได้ถูกต้องด้วยระยะการถ่ายภาพที่ 50 เซนติเมตร 1 เมตร 1.5 เมตร 2 เมตร และ 2.5 เมตร

1.3.4 นำข้อมูลจากการตรวจนับจำนวนนกมาแสดงในรูปแบบวิชวลไลเซชัน

1.4 ประโยชน์ที่คาดว่าจะได้รับ

1.4.1 สามารถนำข้อมูลจากกล้องถ่ายภายความร้อนมาใช้ในการบ่งชี้และนับจำนวนนกนางแอ่นกินรังได้

1.4.2 สามารถลดระยะเวลาในการตรวจนับจำนวนนกนางแอ่นกินรังและเพิ่มความถูกต้องแม่นยำในการตรวจนับได้

1.4.3 สามารถนำข้อมูลจำนวนนกนางแอ่นกินรังที่รวบรวมมาเป็นข้อมูลส่วนหนึ่งในการพิจารณาและประเมินผลการดำเนินงานขอธุระกิจฟาร์มนกนางแอ่นกินรังได้

1.5 นิยามศัพท์เฉพาะ

1.5.1 IoT (Internet of Things) คือ อุปกรณ์อิเล็กทรอนิกส์ที่มีความชาญฉลาดสามารถเชื่อมต่อกับเครือข่ายสามารถส่งข้อมูลที่มีประโยชน์จากเซ็นเซอร์ชนิดต่างๆ ซึ่งสามารถนำไปใช้งานให้เกิดประโยชน์หลากหลาย เช่น บ้านอัจฉริยะ อุปกรณ์สวมใส่อัจฉริยะ เมืองอัจฉริยะ ยานพาหนะอัจริยะ ใช้ในธุรกิจประเภทหรืออุตสาหกรรมประเภทต่างๆ และอื่นๆ อีกมากมาย

1.5.2 กล้องถ่ายภาพความร้อน คือ กล้องที่ใช้เซ็นเซอร์วัดอุณหภูมิที่ผิวของวัตถุ ทำงานโดยอาศัยหลักการจับพลังงานรังสีอินฟราเรด (IR) ที่ถ่ายทอดออกมาจากวัตถุไปสู่สิ่งแวดล้อมและสร้างภาพแถบสีที่วัตถุที่ร้อนกว่าจะแสดงสีสว่างและวัตถุที่เย็นกว่าจะแสดงสีมืดกว่า ซึ่งเป็นการวัดแบบไม่สัมผัสและไม่ทำลายวัตถุ และเป็นการวัดอุณหภูมิแบบพื้นที่

1.5.3 Google Colab หรือ Google Colaboratory คือ IDE ออนไลน์ที่สามารถใช้ในการเขียนโปรแกรมภาษาไพทอน หรือภาษาอื่นๆ โดยที่ไม่ต้องติดตั้งโปรแกรมใดๆ เพิ่มเติม เพียงแค่มีบัญชีของกูเกิล

1.5.4 Google Sheets (กูเกิล ชีท) คือ แอปพลิเคชันในกลุ่มของ Google Drive (กูเกิลไดรฟ์) ซึ่งเป็นนวัตกรรมของ Google (กูเกิล) มีลักษณะการทำงานคล้ายกันกับ Microsoft Excel (ไมโครซอฟท์ เอ็กเซล) คือสามารถสร้าง Column, Row สามารถใส่ข้อมูลต่างๆ ลงไปใน Cell (เซลล์) ได้ และ คำนวณสูตรต่างๆได้

1.5.5 LabelIMG คือ เป็นเครื่องมือสำหรับทำคำอธิบายประกอบภาพกราฟิก ซึ่งเขียนด้วยภาษาไพทอนโดยคำอธิบายประกอบภาพจะถูกบันทึกเป็นไฟล์ XML ในรูปแบบ PASCAL VOC ซึ่งเป็นรูปแบบที่ ImageNet ใช้ และยังรองรับรูปแบบ YOLO และ CreateML

1.5.6 Visualization คือ เป็นการนำข้อมูลมาสร้างภาพ แผนผัง หรือ ภาพเคลื่อนไหว เพื่อใช้ในการสื่อสารแทนข้อมูลตัวเลข หรือข้อความ ให้ชัดเจนและเข้าใจง่ายกว่าเดิม

1.5.7 Power BI คือ โปรแกรมที่ใช้ในการวิเคราะห์ผลทางธุรกิจและนำเสนอออกมาในรูปแบบ Interactive ทั้งแบบตาราง กราฟ ชาร์ท แผนที่ และรูปร่างต่างๆ โดยสามารถแชร์รายงานให้บุคคลอื่นๆ ได้ โดยสามารถแสดงผลผ่านเว็บไซต์ อุปกรณ์มือถือ และแท็บเล็ตได้อีกด้วย

1.5.8 Python คือ ภาษาโปรแกรมระดับสูงภาษาหนึ่ง ถูกออกแบบให้โค้ดสามารถ อ่านได้ง่าย โดยการใช้ “ช่องไฟ” หรือ whitespace เป็นตัวแบ่งว่าโค้ดแต่ละบรรทัดอยู่ภายใต้ Block ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม เป็นภาษา Opensource ซึ่งใครก็สามารถโหลดไปใช้ได้ฟรี

1.5.9 Raspberry Pi คือ บอร์ดคอมพิวเตอร์ขนาดเล็ก สามารถเชื่อมต่อกับจอมอนิเตอร์ คีย์บอร์ด และเมาส์ได้ รองรับระบบปฏิบัติการลินุกซ์ มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย

1.5.9 Line Notification คือ บริการที่ทางแอปพลิเคชันไลน์ ให้สามารถส่งข้อความ หรือภาพ ให้แจ้งเตือนอัตโนมัติ ไม่ว่าจะส่งเข้าผ่านกลุ่ม หรือบัญชีส่วนตัว ผ่าน API ของ LINE โดยตรง

บทที่ 2
แนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้อง

ในการศึกษาเรื่องระบบตรวจนับจำนวนรังนกแอ่นกินรัง เพื่อประเมินผลความสำเร็จธุรกิจฟาร์มนี้ ผู้วิจัยได้สืบค้นข้อมูล ทฤษฎีและงานวิจัยที่เกี่ยวข้อง เพื่อเป็นแนวทางในการศึกษา โดยมีหัวข้อดังต่อไปนี้

2.1 พฤติกรรมการสืบพันธ์ของนกแอ่นกินรัง

2.2 อินเทอร์เน็ตของสรรพสิ่ง (Internet of things)

2.3 การตรวจจับวัตถุ Object Detection

2.4 TensorFlow

2.5 Open-Source Computer Vision (OpenCV)

2.6 อัลกอริทึม YOLO (You Only Look Once)

2.7 โปรแกรม Microsoft Power BI

2.8 งานวิจัยที่เกี่ยวข้อง

2.1 พฤติกรรมการสืบพันธ์ของนกแอ่นกินรัง

โดยเฉลี่ยนกแอ่นกินรังใช้ระยะเวลาในการสร้างรังประมาณ 48 วัน ซึ่งนกตัวผู้และนกตัวเมียจะช่วยกันสร้างรังนก โดยนกแอ่นกินรังใช้ระยะเวลาหลังและก่อนออกหาอาหาร ซึ่งส่วนมากรังนกแอ่นกินรังจะสร้างจากน้ำลายนกเพียงอย่างเดียว หรือมีขนนกปะปนอยู่บ้าง โดยนกนางแอ่นกินรังจะใช้น้ำลายลากวนไปวนมา เป็นแนวโค้งครึ่งวงกลม คล้ายถ้วยฝ่าซีก ติดกับผนัง หลังจากสร้างรังได้ 25 ถึง 30 วันนกแอ่นกินรังจะเริ่มผสมพันธ์กันครั้งแรก

รูปที่ 2.1 พฤติกรรมการเลี้ยงดูลูกนกแอ่นกินรัง [4]

การวางไข่ นกนางแอ่นกินรังจะวางไข่ครั้งละ 2 ฟอง โดยแต่ละฟองจะห่างกันโดยเฉลี่ย 0 ถึง 4 วัน และใช้ระยะเวลาในการฟักไข่เฉลี่ยอยู่ที่ระหว่าง 22 ถึง 27 วัน ซึ่งฟองแรกและฟองที่สองเฉลี่ยฟักห่างกันไม่เกิน 2 วัน โดยพ่อแม่นก จะมีพฤติกรรมช่วยกันฟักไข่โดยผลัดหรือสลับกันฟักไข่ในช่วง 4 ถึง 5 วันแรกหลังจากฟักไข่แล้ว พ่อแม่นกจะให้ความอบอุ่นอยู่ตลอดเวลา โดยการให้ความอบอุ่น มี 2 แบบคือ นอนบนรังนกทั้งสองตัว หรือนอนบนรังนกเพียงตัวเดียวและอีกตัวเกาะห้อยอยู่กับข้างรังนก เมื่อลูกนกแอ่นกินรังอายุ 20 ถึง 29 วัน จะมีขนปกคลุมทั่วตัว ทำให้พ่อนกและแม่นกจะต้องเปลี่ยนมานอนโดยการเกาะแขวนตัวห้อยอยู่กับรังนกแทน และลูกนกนอนเบียดอยู่บนรัง เมื่อลูกนกมีอายุประมาณ 39 ถึง 52 วัน ก็จะบินออกจากรัง [4]

รูปที่ 2.2 การเจริญเติบโตและพฤติกรรมของลูกนกแอ่นกินรัง [4]

2.2 อินเทอร์เน็ตของสรรพสิ่ง (Internet of things)

อินเทอร์เน็ตของสรรพสิ่ง คือการที่อุปกรณ์ชนิดต่าง ๆ ที่สามารถเชื่อมต่อเข้ากับเครือข่ายโดยสามารถส่งข้อมูลถึงกัน ซึ่งข้อมูลที่ได้รับอาจมาจากอุปกรณ์เซ็นเซอร์ประเภทต่าง ๆ ซึ่งการเชื่อมต่อนี้ทำให้เราสามารถควบคุม ตรวจสอบ จัดการ อุปกรณ์ต่างๆ ได้ทั้งแบบอัตโนมัติ หรือกึ่งอัตโนมัติได้ รวมไปถึงเมื่อได้เชื่อมต่อกับการใช้งานประเภทอื่นๆ เพื่อนำไปใช้ประโยชน์ในด้านต่างๆ จนเกิดเป็นนวัตกรรมต่างๆ อาทิเช่น Smart Home Smart Intelligent Transport Smart Device Smart Network Smart Office Smart Grid เป็นต้น ซึ่งการเชื่อมโยงนั้น จะสามารถเก็บและรวบรวมข้อมูลได้อย่างเป็นระบบ นอกจากนี้แล้ว ยังมีระบบคลาวด์ที่จัดเก็บและประมวลผลข้อมูลผ่าน

ออนไลน์ โดยที่เราสามารถควบคุมหรือกำหนดความเป็นส่วนตัวและสามารถเข้าถึงข้อมูลได้ตลอดเวลา

รูปที่ 2.3 SMART CITY [5]

2.2.1 Raspberry Pi

Raspberry Pi คือ บอร์ดคอมพิวเตอร์ขนาดเล็กที่มีขนาดใกล้เคียงกับบัตรเครดิต พัฒนาขึ้นโดย Raspberry Pi Foundation ในสหราชอาณาจักร เป็นคอมพิวเตอร์ประสิทธิภาพสูง ราคาประหยัด ที่มีซีพียู แรม รอม รวมอยู่ไว้บนบอร์ดเดียวกัน สามารถต่อจอ คีย์บอร์ด เมาส์ ใช้งานได้เหมือนคอมพิวเตอร์ทั่วไป รองรับระบบปฏิบัติการลินุกซ์ (Linux Operating System) ได้หลายระบบ เช่น Raspbian NOOBS Raspberry Pi Desktop (for PC and Mac) หรือแม้แต่ Third Party Operating System Images เช่น Ubuntu MATE,OSMC, Libre ELES เป็นต้น โดยติดตั้งบน SD Card บอร์ด Raspberry Pi นี้ถูกออกแบบมาให้มี CPU GPU และ RAM อยู่ภายในชิปเดียวกัน มีจุดเชื่อมต่อ GPIO ให้ผู้ใช้สามารถนำไปใช้ร่วมกับอุปกรณ์อิเล็กทรอนิกส์อื่นๆ ได้อีกด้วย โดยในงานวิจัยนี้ได้ใช้บอร์ด Raspberry Pi เป็นรุ่น Pi 3 B+ โดยมี spec

การประมวลผล : Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz

RAM : 1GB LPDDR2 SDRAM

การเชื่อมต่อ : 2.4GHz and 5GHz IEEE 802.11.b/g/n/ac wireless LAN, Bluetooth 4.2, BLE, Gigabit Ethernet over USB 2.0 (maximum throughput 300 Mbps)

Extended 40-pin GPIO header

Full-size HDMI

4 USB 2.0 ports

CSI camera port for connecting a Raspberry Pi camera

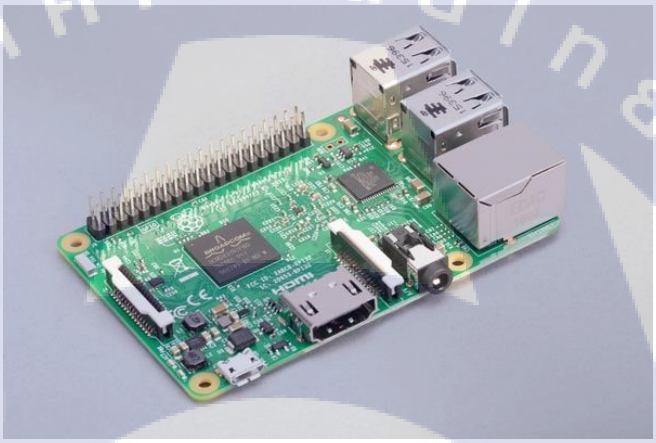
DSI display port for connecting a Raspberry Pi touchscreen display

4-pole stereo output and composite video port

Micro SD port for loading your operating system and storing data

5V/2.5A DC power input

Power-over-Ethernet (PoE) support (requires separate PoE HAT) [6]



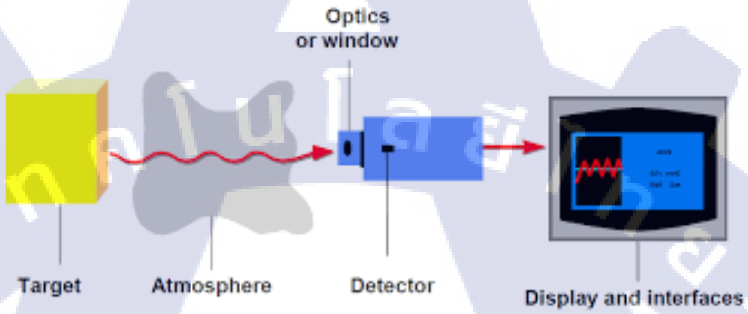
รูปที่ 2.4 บอร์ด Raspberry Pi 3 B+ [6]

2.2.2 เซ็นเซอร์ (Sensor)

เซ็นเซอร์ คือ เป็นอุปกรณ์ซึ่งทำหน้าที่เป็นตัวตรวจจับปริมาณหรือสัญญาณทางฟิสิกส์ต่างๆ เช่น อุณหภูมิ เสียง แสง แรงกด ความดันบรรยากาศ ระยะทาง ความเร็ว อัตราเร่ง ระดับ ของเหลว และอัตราการไหล อาศัยหลักการทำงานที่แตกต่างกันขึ้นอยู่กับชนิดของเซนเซอร์ โดยการแปลงสัญญาณทางด้านอินพุตซึ่งเป็นคุณสมบัติทางฟิสิกส์ให้เป็นสัญญาณทางด้านเอาต์พุตซึ่งเป็นคุณสมบัติทางไฟฟ้าซึ่งสามารถนำไปประมวลผลต่อได้ โดยสามารถแบ่งเป็นอนาล็อกเซ็นเซอร์และดิจิทัลเซ็นเซอร์

2.2.3 เซ็นเซอร์กล้องถ่ายภาพความร้อน (Thermal Camera Sensor)

เซ็นเซอร์กล้องถ่ายภาพความร้อน (Thermal Camera Sensor) หรือ กล้องอินฟราเรด (Infrared Thermography) เป็นเซ็นเซอร์ที่วัดอุณหภูมิที่ผิวของวัตถุ เป็นการวัดแบบไม่สัมผัสและไม่ทำลายวัตถุ โดยอาศัยตรวจจับรังสีอินฟราเรด (Infrared) ที่แผ่ออกจากวัตถุ แล้วแปลงรังสีเหล่านั้นในอยู่ในรูปของสัญญาณไฟฟ้า และวงจรอิเล็กทรานิกส์บนเซ็นเซอร์จะทำหน้าที่แปลงให้อยู่ในรูปของตัวเลข สี หรือกราฟ [7]



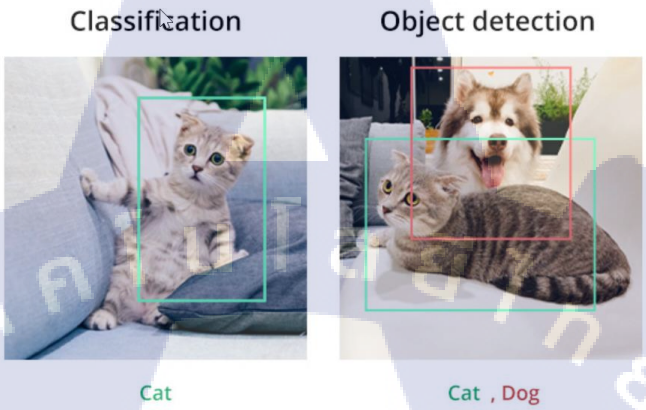
รูปที่ 2.5 การทำงานของ infrared thermography [7]

Lepton Version 3.5 เป็นโมดูลกล้องถ่ายภาพอินฟราเรดคลื่นยาว (LWIR) โดยอาศัยการจับการสะท้อนรังสีอินฟราเรดในแถบความยาวคลื่น (ตั้งแต่ 8 ไมครอนถึง 14 ไมครอน) โดยสามารถวัดอุณหภูมิ ได้ตั้งแต่ -10 °C ถึง 80 °C มีความแม่นยำ + - 5° C โดยโมดูลกล้องถ่ายภาพอินฟราเรด Lepton Version 3.5 มีความละเอียดของภาพถ่ายที่ 160h x 120v พิกเซล อัตราเฟรม 8.7 เฮิร์ตซ์ และใช้พลังงานไฟฟ้า 2.8 โวลต์ ถึง 3.1 โวลต์

2.3 การตรวจจับวัตถุ Object Detection

"เทคโนโลยีตรวจจับวัตถุ" (Object Detection) คือ หนึ่งในฟีเจอร์หลักของ AI (Artificial Intelligence) ที่ใช้กับกล้องหรือกล้องวิดีโอ ซึ่งสามารถค้นหาสิ่งของโดยนำ AI มาวิเคราะห์ข้อมูลจากการมองเห็นของคอมพิวเตอร์ (Computer Vision) และการประมวลผลภาพ (Image Processing) เพื่อตรวจจับวัตถุที่อยู่ในรูปหรือวิดีโอ เช่น มนุษย์ สัตว์ สิ่งของ รถยนต์ อาคาร และวัตถุอื่น ๆ ที่อยู่ในรูปภาพ หรือวิดีโอ โดยตามหลักก่อนที่จะพัฒนามาเป็นเทคโนโลยีตรวจจับวัตถุ (Object Detection) จะต้องผ่านการจัดหมวดของวัตถุ (Object Classification) มาก่อน โดยที่การจัดหมวดของวัตถุจะเป็นการจัดหมวดหมู่ของรูปภาพว่า รูปภาพนั้นคือภาพอะไร แต่เทคโนโลยีตรวจจับวัตถุจะเป็นการระบุเลยว่า ในรูปภาพนั้นมีวัตถุอะไรบ้าง ซึ่งจุดนี้จะต้องอาศัยการทำงานของ AI เข้ามาช่วยใน

การวิเคราะห์ข้อมูลด้วยเช่นกัน เช่น การจัดหมวดของวัตถุจะสามารถระบุได้ว่าวัตถุที่อยู่ในภาพ คือ แมว ในขณะที่เทคโนโลยีตรวจจับวัตถุจะระบุได้ว่าวัตถุที่อยู่ในภาพ มีแมว และสุนัข โดยหลักการสามารถทำได้หลายวิธี การทำมาร์กพื้นที่ที่นิยมได้แก่ วาดกล่องรอบวัตถุ (Bounding Box) หรือ ถมสีให้ทุก Pixel ของวัตถุนั้น (เรียกว่า Segmentation)



รูปที่ 2.6 Object Classification และ Object detection [8]

2.4 TensorFlow

TensorFlow คือ Deep Learning Library ที่เป็นไลบรารี่ที่ใช้ในการพัฒนา Machine Learning ได้รับการพัฒนาโดยบริษัทกูเกิล ได้ทำการเปิดตัวเมื่อวันที่ 11 กุมภาพันธ์ 2017 ซึ่ง TensorFlow นั้นจะเป็น Open Source ที่จะใช้ไพทอนในการเขียน รองรับเวอร์ชันทั้ง Python2 และ Python3 โดย TensorFlow สามารถทำงานบน CPU และ GPUs รองรับระบบปฏิบัติการ Linux macOS Windows และ Android Google ได้เพิ่มประสิทธิภาพให้กับผลิตภัณฑ์มากมายด้วย TensorFlow ไม่ว่าจะเป็น เครื่องมือค้นหา (Search Engine) การแปลภาษา (Translation) คำบรรยายภาพ (Image Captioning) และ เครื่องมือช่วยการเสนอแนะ (Recommendations) เพื่อช่วยให้เห็นภาพมากขึ้น Google นำ AI มาช่วยให้พัฒนาประสบการณ์ของผู้ใช้ ทั้งในแง่ความเร็วของผลลัพธ์ และในแง่ผลลัพธ์ที่ถูกต้องแม่นยำมากขึ้น อย่างเช่น ถ้าเราลองพิมพ์คำอะไรลงไปในช่องค้นหา ล่ะก็กูเกิลจะสามารถแนะนำคำต่อไป หรือคำที่สมบูรณ์ให้เราได้ทันทีเลย

2.5 Open Source Computer Vision (OpenCV)

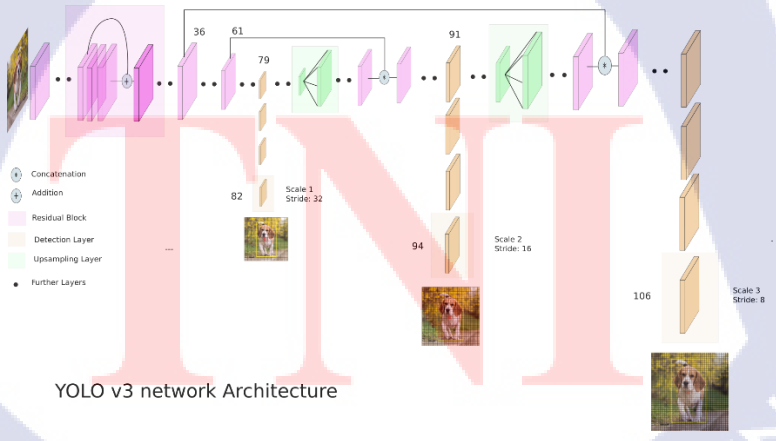
OpenCV เป็นไลบรารี่ที่รวบรวมฟังก์ชันต่างๆ สำหรับการประมวลผลภาพและคอมพิวเตอร์วิทัศนศาสตร์เอาไว้เป็นจำนวนมาก โดยไลบรารี่นี้อยู่ภายใต้ใบอนุญาต BSD ซึ่งสามารถใช้ได้ฟรีทั้งทางด้านการศึกษาและทางการค้า นอกจากนั้น OpenCV ยังมีอินเตอร์เฟสที่หลากหลายรองรับการ

พัฒนาโปรแกรมบนภาษาโปรแกรมต่างๆ เช่นในภาษา C/C++, Python, Java เป็นต้น และ OpenCV ยังสามารถรันได้ทั้งบน Window, Linux, Android, และ Mac

OpenCV ถูกพัฒนาขึ้นมาเพื่อใช้ในการช่วยในการเขียนโปรแกรมเกี่ยวกับกล้องและหุ่นยนต์ ซึ่งถูกพัฒนาขึ้นโดยบริษัทอินเทล เพื่อให้การพัฒนาโปรแกรมทางด้าน การมองเห็นของคอมพิวเตอร์ (Computer Vision) สามารถประมวลผลภาพดิจิตอลได้ทั้งภาพนิ่ง และภาพเคลื่อนไหว เช่น ภาพจากกล้องวีดีโอ หรือ วีดีโอไฟล์ เป็นไปได้อย่างสะดวก มีฟังก์ชันสำเร็จรูปสำหรับจัดการข้อมูลด้านข้อมูลภาพ และการประมวลผลภาพพื้นฐาน เช่น การหาขอบภาพ การกรองข้อมูลภาพ ฟังก์ชันต่างๆ ของ OpenCV จะสามารถเรียกใช้งานได้จะต้องมีการเรียก ไฟล์ส่วนหัว (Header File) และลิงค์ (Link) ไลบรารี่ต่างๆ รวมถึง DLL (Dynamic Link Library)

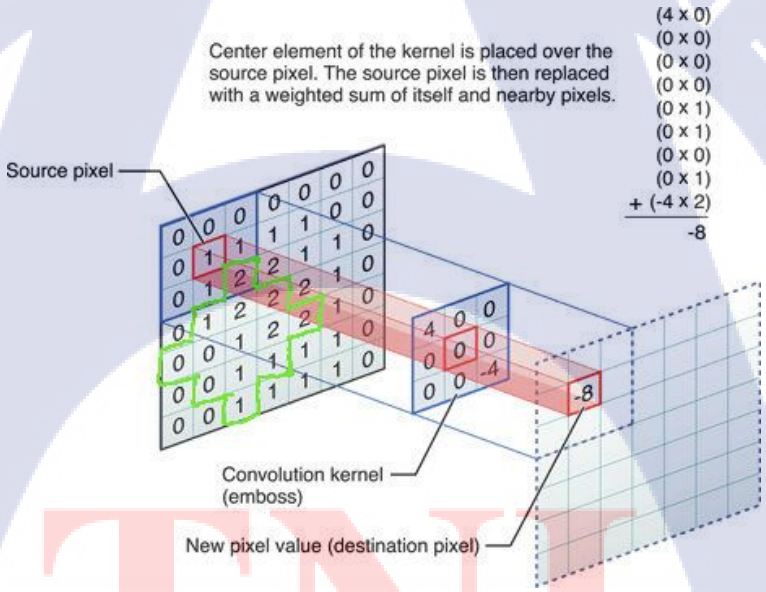
2.6 อัลกอริทึม You only look once (YOLO)

YOLO ย่อมาจาก You Only Look Once เป็นระบบเทคนิคการตรวจจับวัตถุในภาพ (Modern Convolutional Detection) ที่ถูกพัฒนาการด้วยโปรแกรม C++ สำหรับงานปัญญาประดิษฐ์แบบโครงข่ายใยประสาท (Neural Network) ด้วยความสามารถที่ทำงานบนหน่วยประมวลผล CUDA ของ GPU Card ทำให้สามารถประมวลผลภาพแบบ Real Time ได้อย่างแม่นยำและรวดเร็ว ภายใต้โครงข่ายใยประสาทเพียงชุดเดียว YOLO สามารถทำนายชนิดของวัตถุสิ่งของและตำแหน่งได้พร้อมๆ กัน สามารถแยกแยะภาพวาด หรือภาพถ่าย สามารถแยกแยะวัตถุจากภาพ Background คาดเดาวัตถุที่กระจัดกระจายในภาพได้เป็นอย่างดี



รูปที่ 2.7 YOLOv3 Network Architecture [9]

YOLO แต่ละเวอร์ชันจะมีโครงสร้าง Convolution Box ที่แตกต่างกัน YOLO จะทำการกำหนดขนาดของภาพกับ Model ภาพในฐานข้อมูล การคำนวณแต่ละรูปแบบของ Convolution Box YOLO ได้นำเอา Neural Network แบบต่าง ๆ เช่น Feed Forward, Back Propagation มาต่อออกจาก Convolution Kernel นั้นเอง ซึ่งเป็นลักษณะของ Learning Rule ที่แตกต่างกัน โดย YOLO มีขั้นตอนในการตรวจจับวัตถุในภาพ คือ แบ่งภาพออกเป็น Grid Cell ขนาดเล็ก และนำ Grid Cell ไปทดสอบความเหมือนกับลำดับโมเดลภาพที่ต้องการเปรียบเทียบ โดยสร้างเป็น Array of Grid Cell ค่าใน Grid จะเป็น Probability ของแต่ละ Model ในฐานข้อมูล ซึ่งจะสอดคล้องกับ Location ของภาพ ณ ตำแหน่งนั้น ๆ จากนั้นจะลดขนาด และทำการวนทดสอบความเหมือนกับลำดับโมเดลภาพใน State หลายๆ รอบ โดยเรียกขั้นตอนทั้งสองขั้นตอนว่า การทำ Convolution หรือเรียกว่าการทำ Filter และใช้ Max Pool ในการรวมกลุ่มของ Grid Cell เล็กๆ ที่เหมือนกันออก เพื่อลดขนาดในการคำนวณ



รูปที่ 2.8 Convolution Layer [10]

2.7 โปรแกรม Microsoft Power BI

เป็นผลิตภัณฑ์ของบริษัทไมโครซอฟโดยระบบ Power BI เป็นส่วนหนึ่งของบริการซอฟต์แวร์ แอปพลิเคชัน ซึ่งสามารถเชื่อมต่อกับแหล่งข้อมูลที่ไม่เกี่ยวข้องกัน ให้เป็นข้อมูลเชิงลึกที่สอดคล้อง สามารถแสดงผลข้อมูล และโต้ตอบได้ ข้อมูลอาจเป็นสเปรดชีต Excel ข้อมูลบนระบบ Cloud หรือคลังข้อมูลแบบไฮบริด Power BI ยังสามารถแชร์สิ่งเหล่านั้นกับบุคคลหรือทุกคนที่

ต้องการได้อย่างง่ายดาย ไม่ว่าจะเป็นแพลตฟอร์มสำหรับอุปกรณ์ Windows, iOS และ Android อีกทั้งยังมีเวอร์ชันฟรีที่ให้ใช้งานได้อีกด้วย

รูปที่ 2.9 โปรแกรม Power BI ที่สามารถแสดงผลได้หลายแพลตฟอร์ม [11]

2.8 งานวิจัยที่เกี่ยวข้อง

งานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อนดังนี้

2.8.1 ชีววิทยาการสืบพันธุ์ของนกแอ่นกินรัง Aerodramus Germani OUSTALET ที่วัดสุทธิวาตวราราม อำเภอเมือง จังหวัดสมุทรสาคร

จากงานวิจัยของ ยุทธพงค์ ดำศรีสุข วิจักขณ์ ฉิมโฉม และนันทชัย พงษ์พัฒนานุรักษ์ ได้ทำการศึกษาในหัวข้อ ชีววิทยาการสืบพันธ์ของนกแอ่นกินรัง Aerodramus Germani OUSTALET ที่วัดสุทธิวาตวราราม อำเภอเมือง จังหวัดสมุทรสาคร ได้ทำการศึกษาเกี่ยวกับพฤติกรรมการทำรัง ระยะเวลา รูปแบบของรังของนกนางแอ่นกินรัง การสืบพันธ์ ระยะเวลาการฟักไข่ ตลอดจนการเลี้ยงดู การให้อาหารลูกนก ที่ลูกนกต้องอาศัยอยู่ภายในรังเดียวกันกับพ่อแม่นก เมื่อลูกนกโตขึ้นพ่อหรือแม่จะอยู่บนรังเพียงตัวเดียว ส่วนอีกตัวจะเกาะด้านข้างของรัง หรือ ทั้งพ่อและแม่เกาะด้านข้างของรังทั้งคู่ [4]

รูปที่ 2.10 พฤติกรรมการสร้างรังของนกแอ่นกินรัง [4]

2.8.2 สถาปัตยกรรมเพื่อการอยู่ร่วมกันระหว่างคนกับนก

จากงานวิจัยของ ร่มเกล้า ช่วยดู ได้ทำการศึกษาในหัวข้อสถาปัตยกรรมเพื่อการอยู่ร่วมกันระหว่างคนกับนก ได้นำเสนอรูปแบบสถาปัตยกรรมสิ่งก่อสร้างในชุมชนทับเที่ยงจังหวัดตรัง ให้มีความกลมกลืนกับรูปแบบวัฒนธรรมการดำเนินชีวิตของคนในพื้นที่ และยังสามารถใช้พื้นที่ให้นกแอ่นกินรังสามารถเข้ามาทำรัง ในสภาพแวดล้อมที่เหมาะสมอีกด้วย เพื่อเป็นช่องทางการสร้างรายได้ให้กับชุมชมอีกช่องทางหนึ่ง และในงานวิจัยนี้ยังได้กล่าวถึงระดับความสำเร็จของธุรกิจฟาร์มนกแอ่นกินรังไว้อีกด้วย คือ

ระดับที่ 1 ฟาร์มนกแอ่นที่ล้มเหลว ไม่ประสบผลสำเร็จ คือ มีรังนกแอ่นกินรังน้อยกว่า 5 รังในระยะเวลา 6 เดือน มีนกแอ่นกินรังน้อยกว่า 50 ตัว อาศัยอยู่

ระดับที่ 2 ฟาร์มนกแอ่นที่เกือบผ่านแต่ไม่ผ่าน 100% คือ มีรังนกแอ่นกินรังน้อยกว่า 10 รังในระยะเวลา 6 เดือน มีนกแอ่นกินรังน้อยกว่า 100 ตัว อาศัยอยู่

ระดับที่ 3 ฟาร์มนกแอ่นที่ประสบความสำเร็จ คือ มีรังนกแอ่นกินรังมากกว่า 15 รัง ในระยะเวลา 6 เดือน มีนกเข้าอาศัยมากกว่า 150 ตัว [12]

รูปที่ 2.11 Model การจำลองสถาปัตยกรรม [12]

2.8.3 Implementation of a Low-Cost Real-Time People Counting System on Raspberry Pi Based on Applied Tiny YOLOv3

จากงานวิจัย ของ ปราโมทย์ ปัญญาโต และ นลิน สีดาห้าว ได้ทำการศึกษาในหัวข้อ Implementation of a Low-Cost Real-Time People Counting System on Raspberry Pi Based on Applied Tiny YOLOv3 โดยทางทีมวิจัยได้นำ Tiny YOLOv3 มาทำการ Classification นับบุคคลที่ทำการเดินเข้าออกสถานที่ต่าง ๆ ซึ่งผลของการวิจัยความแม่นยำขึ้นอยู่กับหลายปัจจัย

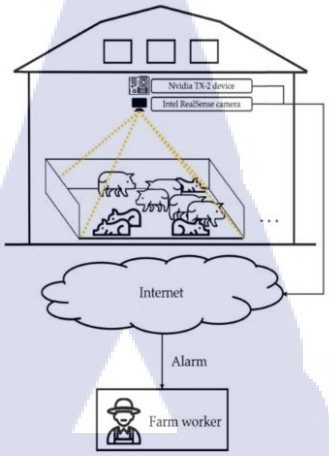
เช่น แสง มุมของกล้อง ความเร็วของคนเดิน การกำหนดค่าเกณฑ์ และดาต้าเซตที่ได้ทำการเทรนโมเดลที่ใช้กับอัลกอริทึม YOLO [13]

Bright-ness (%)	Threshold (%)	Total number of people	System counting	Counting accuracy (%)
100	80	24	16	67
	60		24	100
90	80		16	67
	60		24	100
80	80		15	62
	60		22	91
70	80		6	25
	60		12	50
60	80		0	0
	60		5	21
50	80		0	0
	60		0	0

รูปที่ 2.12 รายละเอียดประสิทธิภาพการตรวจนับบุคคล ระหว่างการกำหนดเกณฑ์ต่ำสุดของอัลกอริทึม TinyYOLOv3 ในการตรวจจับวัตถุที่ 80% และ 60% [13]

2.8.4 Practical Monitoring of Undergrown Pigs for IoT-Based Large-Scale Smart Farm

จากงานวิจัยของ Sungju Lee, Hanse Ahn, Jihyun Seo, Yongwha Chung, Daihee Park, Sungbum Pan ได้ทำการศึกษาในหัวข้อ Practical Monitoring of Undergrown Pigs for IoT-Based Large-Scale Smart Farm โดยใช้ อุปกรณ์ IoT รวมกับกล้องตรวจจับการเคลื่อนไหว เพื่อตรวจจับสุกรที่เป็นโรคจากกลุ่มสุกรที่ถูกเลี้ยงไว้ในห้องเดียวกันโดยใช้เทคนิค Gaussian Mixture Model เปรียบเทียบกันระหว่าง อัลกอริทึม Mask R-CNN กับ Tiny YOLO รวมกับอุปกรณ์ NVIDIA TX-2 และ Intel RealSense Camera ปรากฏว่า อัลกอริทึม Tiny YOLO สามารถทำการแจ้งเตือนเมื่อพบสุกรที่เป็นโรคไปยังผู้เลี้ยงได้ดีกว่าอัลกอริทึม Mask R-CNN ซึ่งไม่สามารถทำการแยกสุกรที่เป็นโรคได้ [14]



รูปที่ 2.13 ขั้นตอนการทำงานของ Monitoring System [14]

2.8.5 Development Of A Thermographic Image Instrument Using The Raspberry Pi Embedded System For The Study Of The Diabetic Foo

เป็นงานวิจัยของ Rafael Bayareh, Arturo Vera, Lorenzo Leija, J. Gutierrez-Martínez ในประเทศแม็กซิโก ได้นำเสนออุปกรณ์สำหรับช่วยตรวจโรคเบาหวานที่เท้าต้นทุนต่ำ ด้วยการนำกล้องตรวจจับความร้อนของ FLIR รุ่น Lepton Dev Kit มาใช้งานร่วมกับบอร์ด Raspberry Pi 2 โดยนำภาพถ่ายความร้อนที่ได้มาทำการวินิจฉัยโรคเบาหวานที่เท้า ซึ่งผู้ที่ป่วยเป็นโรคเบาหวานที่บริเวณเท้า ผิวหนังบริเวณที่มีความผิดปกติ โดยจะมีอุณหภูมิที่สูงกว่าที่ผิวหนังที่เท้าบริเวณอื่น และการตรวจด้วยวิธีนี้ยังช่วยลดผลข้างเคียงจากการตรวจด้วยวิธีการเจาะเลือด โดยทางทีมวิจัยได้สรุปผลว่างานวิจัยชิ้นนี้สามารถใช้งานได้จริง และสามารถลดต้นทุนของอุปกรณ์ในตลาดที่คุณสมบัติความละเอียดของกล้องถ่ายภาพความร้อนที่เท่ากันได้ถึง 110 ดอลลาร์สหรัฐ [15]

รูปที่ 2.14 อุปกรณ์ที่ใช้ในงานวิจัย [15]

2.8.6 Deep Learning-Based Real-Time Multiple-Person Action Recognition System

เป็นงานวิจัยของ Jen-Kai Tsai, Chen-Chien Hsu , Wei-Yen Wang and Shao-Kang Huang โดยร่วมกันพัฒนาฟังก์ชัน “Zoom In” ซึ่งเป็นฟังก์ชันที่ช่วยให้การทำ “Action Recognition” ให้มีความแม่นยำมากยิ่งขึ้น ซึ่งการทำงานของงานวิจัยชิ้นนี้ จะนำภาพวิดีโอที่ขนาด 1440 x 1080 พิกเซล มาลดขนาดลงเหลือ 640 x 480 พิกเซล แล้วทำการตรวจสอบใบหน้า และกริยาท่าทางของภาพคนในวิดีโอ ด้วยอัลกอริทึม YOLOv3 ร่วมกับไลบรารี่ Facenet โดยเมื่อถ้าพบภาพมนุษย์อยู่ห่างออกไป ก็จะเรียกใช้ฟังก์ชัน Zoom In เพื่อให้เห็นหน้ามนุษย์ได้ชัดเจนมากยิ่งขึ้น ซึ่งส่งผลให้อัลกอริทึม YOLOv3 และไลบรารี่ Facenet สามารถทำการตรวจสอบใบหน้าได้แม่นยำมากยิ่งขึ้น [16]

	Average Accuracy	Short-Distance Accuracy	Long-Distance Accuracy
With zoom-in	90.79%	91.49%	89.29%
Without zoom-in	69.74%		32.14%

รูปที่ 2.15 ผลการทดลองเปรียบเทียบการใช้ฟังก์ชัน Zoom-In [16]

2.8.7 Unmanned Aerial Vehicle and Artificial Intelligence for Thermal Target Detection in Search and Rescue Applications

เป็นงานวิจัยของ Joseph McGee , Sajith J Mathew , Felipe Gonzalez โดยงานวิจัยชิ้นนี้ทางทีมวิจัยได้ทำการเปรียบเทียบผลการเทรนโมเดลเข้าอัลกอริทึม YOLOv3 โดยระบุอัตราการเรียนรู้เริ่มต้น (Learning Rate) ที่แตกต่างกันคือ รอบที่ 1 ระบุเป็น 0.0001 และ 0.01 ในรอบที่ 2 ส่วนพารามิเตอร์อื่นกำหนดไว้ และคอมพิวเตอร์ที่ทำการใช้งานก็เหมือนกัน โดยนำภาพมาจากกล้องถ่ายภาพความร้อนของ FLIR รุ่น Tau 2 640 ติดกับโดรนรุ่น Phantom 3 Pro และทำการถ่ายภาพบริเวณ Red Beach, South Bribie Island และ QLD ในประเทศออสเตรเลีย ที่ความสูง 26 และ 40 เมตร จำนวน 10,380 ภาพ มาทำการเทรนเข้าอัลกอริทึม YOLOv3 ตามเงื่อนไขที่ระบุไว้ในตอนต้น ผลจากการทดลองที่ได้คือการเทรนโมเดลด้วยการระบุอัตราการเรียนรู้เริ่มต้น (Learning Rate) ที่มาก จะมีผลกับการคำนวณค่าของโมเดลซึ่งจะเปลี่ยนไปมาก Loss หรือค่าความสูญเสียก็จะเปลี่ยนแปลงไปมากเช่นกัน ระยะเวลาในการประมวลผลของเครื่องคอมพิวเตอร์ก็จะนานตามไปด้วย แต่ถ้าระบุอัตราการเรียนรู้เริ่มต้น (Learning Rate) น้อย การคำนวณค่าของโมเดลก็จะเปลี่ยนไปน้อย Loss หรือค่าความสูญเสียก็จะไม่ค่อยเปลี่ยนแปลง ระยะเวลาการประมวลผลก็จะน้อยตามไปด้วย [17]

TABLE I. SET 1 TRAINING TABLE

Image Number	Initial learning rate	Epochs	Loss	mAP (%)	Training Time (HH:MM)	GPU	Mem (GB)	CPUs	Batch Size
10380	0.0001	23	6	96.92	14:19	P100	40	12	10
10380	0.0001	20	5.31	97.66	12:25	P100	40	12	10
10380	0.0001	100	3.81	98.6	61:12	P100	40	12	10
Total		**143**	**3.81**	**98.6**	**87:56**				

TABLE II. SET 2 TRAINING TABLE

Image Number	Initial learning rate	Epochs	Loss	mAP (%)	Training Time (HH:MM)	GPU	Mem (GB)	CPUs	Batch Size
10380	0.01	50	6.676	91.4	30.56	P100	40	12	10
10380	0.01	250	4.73	97.7	153	P100	40	12	10
Total		**300**	**4.73**	**97.7**	**183:56**				

รูปที่ 2.16 ผลการทดลองเปรียบเทียบการอัตราการเรียนรู้เริ่มต้น (learning rate) [17]

จากข้อมูลของงานวิจัยข้างต้นเห็นได้ว่าบอร์ด Raspberry Pi สามารถทำงานร่วมกับกล้องถ่ายภาพความร้อนของ FLIR ได้เป็นอย่างดี สามารถนำภาพถ่ายที่ได้มาใช้งานได้โดยตรง หรือสามารถนำมาทำเป็นภาพต้นแบบในการทำดาต้าโมเดล สำหรับอัลกอริทึม YOLOv3 ได้ ซึ่งต้องเลือกใช้กล้องให้มีคุณสมบัติสอดคล้องเหมาะสมกับวัตถุประสงค์ในการใช้งาน ในส่วนของอัลกอริทึม YOLOv3 เป็น Framework ที่มีความสามารถในการทำงานด้าน Image Detection, Image Recognition ที่หลายงานวิจัยได้นำมาใช้ในงานวิจัยของตนเอง ทั้งยังสามารถเขียนฟังก์ชันเพิ่มเติมเพื่อเพิ่มประสิทธิภาพในการทำงาน หรือสามารถที่จะสร้างโมเดลเป็นของตนเองเพื่อใช้งานได้อีกด้วย โดยในการสร้างหรือการเทรนโมเดลจะต้องคำนึงถึงการกำหนดค่าให้เหมาะสมกับการเทรนโมเดล และประสิทธิภาพของเครื่องคอมพิวเตอร์ที่ใช้ โดยทางผู้วิจัยเห็นว่ามีความเป็นไปได้ในงานวิจัยที่จะทำการถ่ายภาพนกแอ่นกินรังจากกล้องถ่ายภาพความร้อน และนำภาพมาทำการตรวจนับด้วยอัลกอริทึม YOLOv3 ซึ่งสามารถทำการตรวจพบวัตถุต่างๆ ได้เป็นอย่างดีโดยศึกษาจากงานวิจัยที่ได้นำกล้องถ่ายภาพความร้อนติดโดรนแล้วทำการตรวจพบวัตถุต่างๆ อีกทั้งผลจากการตรวจนับจำนวนนกนางแอ่นกินรัง ยังเป็นประโยชน์ด้านสถิติสำหรับผู้ประกอบการว่ามีประสิทธิภาพจากการดำเนินงานหรือไม่ โดยสามารถเทียบวัดกับเกณฑ์ในงานวิจัยข้างต้นได้อีกด้วย

บทที่ 3
วิธีการดำเนินงาน

จากการศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องเพื่อเป็นความรู้สำหรับการศึกษาที่เกี่ยวข้องกับเรื่องการพัฒนาระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อนนี้ ผู้ศึกษาได้นำความรู้ที่ได้รับมาดำเนินงานตามขั้นตอนดังนี้

ขั้นตอนที่ 1 การวิเคราะห์ข้อมูลปัญหาของงานวิจัย

ขั้นตอนที่ 2 การศึกษาข้อมูลและเทคโนโลยีที่นำมาใช้

ขั้นตอนที่ 3 ออกแบบการทดลองและดำเนินการทดลอง

ขั้นตอนที่ 4 วิเคราะห์ผลการทดลอง

ขั้นตอนที่ 5 การสรุปผลการวิจัย

โดยแผนการดำเนินงานวิจัยมีทั้งหมด 3 ขั้นตอน ดังตารางที่ 3.1

1. ขั้นตอนเบื้องต้น เป็นขั้นตอนในการหาหัวข้องานวิจัย ผู้วิจัยจึงได้สืบค้นข้อมูล ทฤษฎี และงานวิจัยที่เกี่ยวข้องกับเทคนิคการตรวจนับนกแอ่นกินรังด้วยกล้องตรวจจับความร้อน
2. ขั้นตอนการดำเนินการ เป็นขั้นตอนในการดำเนินงานวิจัยได้เตรียมอุปกรณ์ที่ใช้สำหรับการบันทึกภาพนกแอ่นกินรังในฟาร์มนกนางแอ่นกินรังด้วยกล้องถ่ายภาพความร้อนและโปรแกรมสำหรับตรวจจับและนับจำนวนนกแอ่นกินรัง
3. ขั้นตอนการสรุป เป็นขั้นตอนสุดท้ายในการทำงานวิจัยการสรุปผลงานวิจัยทั้งหมดที่ได้ทำมา ทั้งขั้นตอนเบื้องต้นและขั้นตอนการดำเนินการ เพื่อนำไปเขียนรูปเล่มสารนิพนธ์

ตารางที่ 3.1 แผนการดำเนินงานวิจัย

การดำเนินงาน	ระยะเวลา							
	ปี 2563				ปี 2564			
	พ.ค.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
ขั้นเบื้องต้น								
- การหาหัวข้อ								
- การทบทวนวรรณกรรม								
- ทำการทดลองเบื้องต้น								
- ร่างเค้าโครงสารนิพนธ์								
- นำเสนอเค้าโครงสารนิพนธ์								
ขั้นตอนการดำเนินงาน								
- การเตรียมอุปกรณ์								
- การเก็บข้อมูลและเตรียมข้อมูล								
- การประมวลผล								
ขั้นตอนการสรุป								
- สรุปและตีพิมพ์งานวิจัย								
- เขียนสารนิพนธ์เล่มสมบูรณ์								
- สอบป้องกันสารนิพนธ์								

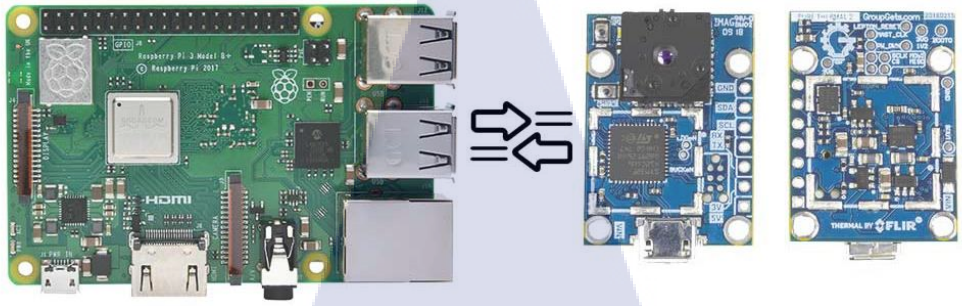
3.1 ขั้นตอนที่ 1 การวิเคราะห์ข้อมูลปัญหาของงานวิจัย

ผู้วิจัยมีความสนใจในเทคโนโลยีอินเตอร์เน็ตทุกสรรพสิ่ง (IoT) อยู่แล้วจึงได้ขอคำปรึกษากับอาจารย์ที่ปรึกษาเพื่อหาหัวข้อที่สามารถนำเทคโนโลยี IoT เข้ามาแก้ไขปัญหาที่เกิดขึ้น และสามารถทำได้จริง จึงเล็งเห็นว่าด้วยการเติบโตของธุรกิจฟาร์มนกแอ่นกินรัง ซึ่งเป็นธุรกิจที่เกิดขึ้นมานาน มีมูลค่าการส่งออกและเป็นที่ต้องการของตลาดเป็นอย่างมาก ทั้งยังปัจจุบันกฎหมายยังได้เปิดให้สามารถจัดเก็บรังนกที่เกิดจากการเลี้ยงได้อย่างถูกกฎหมาย ทำให้มีผู้ประกอบการที่สนใจในธุรกิจฟาร์มนกแอ่นกิจรังไม่น้อย แต่ในการลงทุนทำฟาร์มนกแอ่นกินรัง จำเป็นต้องใช้เงินลงทุนจำนวนไม่น้อย ทั้งสถานที่ อาคาร หรือแม้แต่อุปกรณ์ต่างๆ รวมถึงเทคโนโลยีที่ใช้ในฟาร์มนกแอ่นกินรัง ในการทำธุรกิจฟาร์มนกแอ่นกินรัง เพื่อให้ได้คุณภาพและปริมาณที่เหมาะสม จำต้องมีการควบคุม บริหารจัดการอาคารโรงเรือนนกแอ่นกินรัง โดยจัดทำเป็นโรงเรือนหรืออาคารปิดมีการควมคุมการเข้าออก ควบคุมปัจจัยต่างๆ ไม่ว่าจะเป็นอุณหภูมิที่เหมาะสมซึ่งควรอยู่ที่ 27-29 องศาเซลเซียส ความชื่นที่เหมาะสมอยู่ที่ 80-95 เปอร์เซ็นต์ [18] เชื้อโรค มด แมลงรบกวน หรือแม้แต่สัตว์ที่เป็นอันตรายต่อนกแอ่นกินรังเอง เช่น นกผู้ล่า ค้างคาว ตุ๊กแก หนู ดังนั้น ผู้วิจัยจึงนำเสนอเทคนิคการตรวจนับนกแอ่นกินรังภายในอาคารฟาร์มนกแอ่นกินรัง เพื่อเป็นส่วนช่วยในการทำฟาร์มนกแอ่นกินรัง ว่า ณ ขณะนั้นมีจำนวนนกแอ่นกินรังในอาคารฟาร์มนกมากน้อยเท่าไร เพียงพอเหมาะสมหรือไม่ หรือว่าจำเป็นต้องดำเนินการแก้ไขอย่างไร เพื่อให้นกแอ่นกินรังเข้ามาทำรังเพิ่มมากยิ่งขึ้น

3.2 ขั้นตอนที่ 2 การศึกษาข้อมูลและเทคโนโลยีที่นำมาใช้

ผู้วิจัยได้ทำการศึกษาจากบทความและงานวิจัยต่างๆ ที่ใกล้เคียง เพื่อนำข้อมูลที่ได้มาปรับใช้ในงานวิจัยชิ้นนี้ ไม่ว่าจะเป็นอุปกรณ์ทางด้านฮาร์ดแวร์ หรือซอฟต์แวร์ที่เกี่ยวข้อง รวมถึงอัลกอริทึมที่เหมาะสม ซึ่งในงานวิจัยชิ้นนี้ทางผู้วิจัยได้เลือกใช้อุปกรณ์เป็นบอร์ด Raspberry Pi 3 B+ ซึ่งมีคุณสมบัติเป็นเสมือนคอมพิวเตอร์ขนาดเล็ก ที่สามารถเชื่อมต่อกับอุปกรณ์เซนเซ็นต์ สามารติดตั้งโปรแกรมอัลกอริทึมในการตรวจจับ และตรวจนับจำนวนนกแอ่นกินรังได้อีกด้วย ใช้งานร่วมกับกล้องตรวจจับอุณหภูมิความร้อน FILR Lepton 3.5 เพราะเป็นกล้องตรวจจับอุณหภูมิความร้อนที่มีขนาดเล็กเพียง 10.50 × 12.7 × 7.14 มล. ทำงานได้ในช่วงอุณหภูมิ -10 ถึง +80 องศาเซลเซียส ขนาดภาพถ่ายที่ถ่ายออกมามีขนาด 120 x 160 พิกเซล เพื่อใช้ในการทำต้นแบบสำหรับงานวิจัยนี้

สำหรับอัลกอริทึมที่เลือกใช้ผู้วิจัยได้เลือกใช้อัลกอริทึม YOLYv3 เพราะมีความแม่นยำสูง ประมวลผลข้อมูลได้ในระยะเวลาอันสั้น เมื่อเทียบกับอัลกอริทึมอื่นๆ โดยศึกษาจากงานวิจัยหลายๆ งานวิจัย ทั้งยังมีไลบรารี่สำหรับใช้ในการพัฒนาโปรแกรมอย่างแพร่หลายให้ศึกษา และสามารถนำมาใช้ได้อีกด้วย

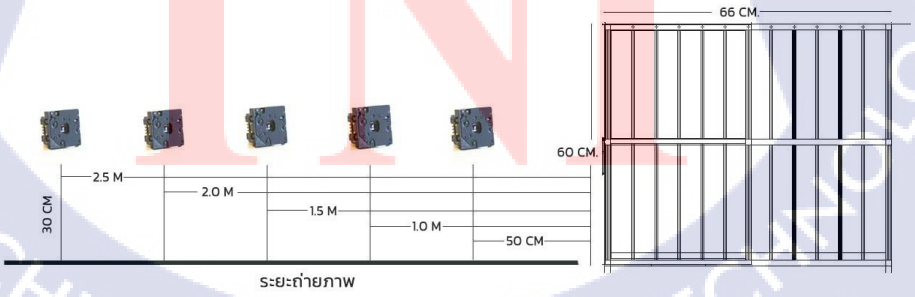


รูปที่ 3.1 บอร์ด Raspberry pi 3 B+ และ Lepton 3.5 Pure thermal v.2 [6, 19]

3.3 ขั้นตอนที่ 3 ออกแบบการทดลองและดำเนินการทดลอง

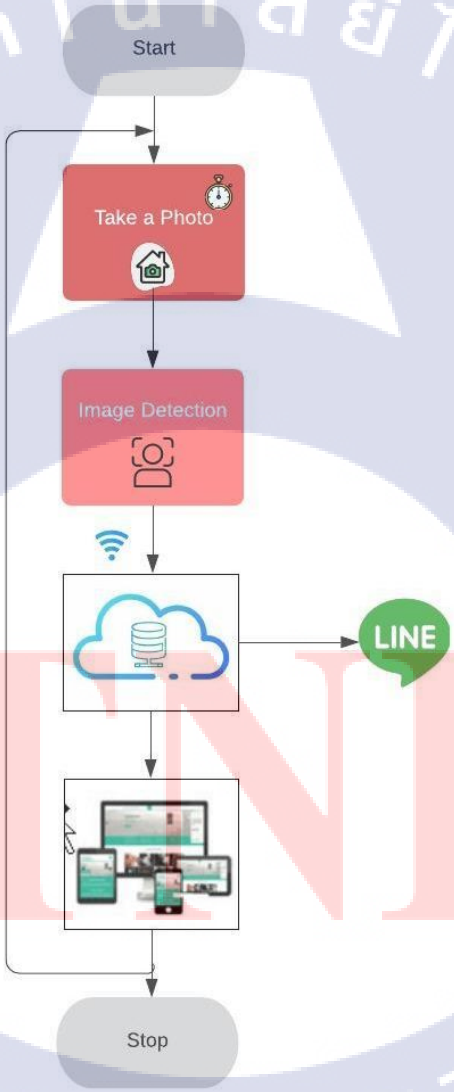
ผู้วิจัยได้ทำการออกแบบขั้นตอนของการทำงานของเทคนิคการตรวจนับนกนางแอ่นกินรัง โดย ได้ศึกษาการติดตั้ง การเชื่อมต่อบอร์ด Raspberry pi 3 B+ ร่วมกับ Lepton 3.5 บนบอร์ด Pure thermal V.2 และได้ติดตั้งซอฟต์แวร์ต่างๆ ที่เกี่ยวข้อง เพื่อสามารถที่จะนำซอฟต์แวร์ต่างๆ ที่เกี่ยวข้องมาทำการทดสอบการใช้งาน และปรับปรุงให้สามารถใช้งานได้ตามวัตถุประสงค์ที่ว่างไว้

หลังจากผู้วิจัยได้พัฒนาอุปกรณ์ข้างต้น และได้นำอุปกรณ์ดังกล่าวมาทำการทดสอบ ด้วยการจำลองการถ่ายภาพนกหงส์หยกอายุ 3 – 4 เดือนซึ่งมีขนาดใกล้เคียงกับนกแอ่นกินรัง จำนวน 4 ตัว ภายในกรงนกขนาด 120 x 60 x 66 เซนติเมตร ควบคุมแสงให้มีสภาพใกล้เคียงกับฟาร์มนกนางแอ่นกินรัง โดยการนำผ้าปิดรอบกรง และปิดไฟทั้งหมด เพื่อให้มีแสงน้อยที่สุด จากนั้นทำการถ่ายคลิปวิดีโอด้วยอุปกรณ์ที่ได้ทำขึ้นเพื่อนำไปทำโมเดลสำหรับนำมาใช้ในงานวิจัย จากนั้นทำการทดลองโดยการตั้งกล้องถ่ายภาพความร้อนด้วยระยะ 50 เซนติเมตร 1 เมตร 1.5 เมตร 2 เมตร และ 2.5 เมตร สูงจากพื้น 30 เซนติเมตร เพื่อทดสอบการทำงานของอุปกรณ์ และความถูกต้องแม่นยำของอัลกอริทึมที่ได้จัดทำไว้



รูปที่ 3.2 การจำลองการทดลอง

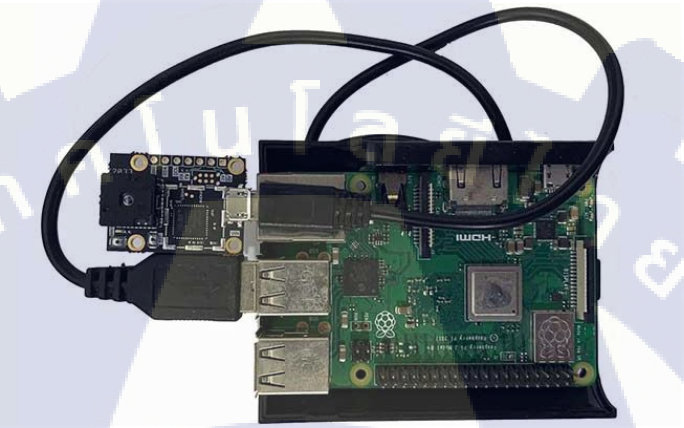
ขั้นตอนการทำการงานของระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อน เมื่อระบบเริ่มทำงาน ระบบจะทำการถ่ายภาพนกด้วยกล้องถ่ายภาพความร้อนตามเวลาที่ได้ตั้งเวลาไว้ จากนั้นจะนำภาพที่ได้ถ่ายได้เข้าอัลกอรึทึม YOLOv3 เพื่อทำการตรวจนับจำนวนนกที่พบด้วยโมเดลที่ได้จัดทำขึ้น เมื่อได้ข้อมูลแล้วจะทำการส่งข้อมูลบันทึกเก็บไว้ที่ Google Sheets พร้อมทั้งระบบจะทำการส่งข้อมูลที่นับได้ กับภาพถ่ายที่ได้ทำการตีกรอบนกที่ตรวจพบไปทาง Line Notification ระบบนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อนยังได้นำข้อมูลที่ได้จัดเก็บไว้มาแสดงรายงานในรูปแบบวิชวลไลเซชันด้วยโปรแกรม Power Bi โดยแสดงผังงานของระบบตามรูปที่ 3.3



รูปที่ 3.3 ขั้นตอนการทำงานของเทคนิคการตรวจนับนกแอ่นกินรัง

3.3.1 อุปกรณ์ที่ใช้ภายในเทคนิคการตรวจนับจำนวนแอ่นกินรัง

- บอร์ด Raspberry Pi 3 Model B+
- บอร์ด Pure Thermal V.2, กล้องตรวจจับความร้อน Lepton 3.5
- โปรแกรมไพทอน
- กูเกิล ชีท (Google Sheets)
- Power Bi



รูปที่ 3.4 Raspberry pi 3 B+ ต่อกับกล้องตรวจจับความร้อน Lepton 3.5

3.3.2 ติดตั้งระบบปฏิบัติการสำหรับ Raspberry Pi

สำหรับ Raspberry Pi 3 B+ จะทำการติดตั้งระบบปฏิบัติการเป็น Raspbian บน Micro SD การ์ด โดย Micro SD การ์ดควรมีพื้นที่ขั้นต่ำ 32 กิกะไบต์ หลังจากทำการติดตั้งระบบปฏิบัติการแล้ว ควรทำการอัพเดทและอัพเกรดระบบปฏิบัติการด้วยคำสั่ง

```
$ sudo apt-get update
$ sudo apt-get full-upgrade
```

3.3.3 ติดตั้งไดร์เวอร์ และโปรแกรมสำหรับใช้งานกล้องตรวจจับความร้อน

ดาวน์โหลดโค้ดและทำดำเนินการติดตั้ง ตามขั้นตอนที่คู่มือของทางผู้ผลิตแนะนำเพื่อให้สามารถเขียนโปรแกรมไพทอนเรียกใช้อุปกรณ์ตัวกล้องตรวจจับความร้อนได้ จากบอร์ด Raspberry Pi

```
$ sudo raspi-config
    Select "Advanced Options"
    Select "GL Driver"
    Select "GL (Full KMS) OpenGL desktop driver with full KMS"
    Select "ok"
    Select "Finish"
$ sudo reboot
```

รูปที่ 3.5 การตั้งค่าในบอร์ด Raspberry Pi ให้ใช้งาน GL Driver

```
$ sudo apt-get install qt5-default qtmultimedia5-dev qtdeclarative5-dev \
    qml-module-qtquick-controls2 qml-module-qtmultimedia \
    qml-module-qtquick-layouts qml-module-qtquick-window2 \
    qml-module-qtquick-templates2 qml-module-qtgraphicaleffects \
    libusb-1.0-0-dev cmake git
```

รูปที่ 3.6 การติดตั้งซอฟต์แวร์อื่นๆ ที่จำเป็นในการเรียกใช้กล้องตรวจจับความร้อน

```
$ sudo git clone https://github.com/groupgets/GetThermal.git
$ sudo cd GetThermal
$ sudo git submodule update --init
$ sudo cd libuvc
$ sudo mkdir build
$ sudo cd build
$ sudo cmake ..
$ sudo make
$ sudo cd ../..
$ sudo mkdir build
$ sudo cd build
$ sudo qmake ..
$ sudo make
```

รูปที่ 3.7 การดาวน์โหลดและติดตั้งไดร์เวอร์สำหรับกล้องตัวจับความร้อน

```
$ sudo sudo sh -c "echo 'SUBSYSTEMS==\"usb\", ATTRS{idVendor}==\"1 e4 e\",
ATTRS{idProduct}==\"0100\", SYMLINK+=\"pt1\", GROUP=\"usb\", MODE=\"666\"' >
/etc/udev/rules.d/99-pt1.rules"
$ sudo GetThermal -platform eglfs
```

รูปที่ 3.8 การดาวน์โหลดและติดตั้งไดร์เวอร์สำหรับกล้องตัวจับความร้อน (ต่อ)

3.3.4 เตรียมข้อมูลสำหรับเทรนให้โมเดลสำหรับอัลกอริทึม YOLOv3

หลังจากได้เตรียมอุปกรณ์ด้านฮาร์ดแวร์เรียบร้อยแล้ว ต่อมาเป็นการจัดเตรียมข้อมูลเพื่อนำมาสร้างโมเดล โดยการนำภาพถ่ายวีดีโอคลิปจากนกหงส์หยกภายในกรงที่ได้เตรียมไว้ โดยมีการความคุมแสงบริเวณโดยรอบให้มืดใกล้เคียงกลับฟาร์มนกแอ่นกินรัง ด้วยการนำผ้าปิดรอบกรงนก และปิดแสงไฟทั้งหมด ทำการบันทึกวีดีโอคลิปในด้านต่าง ๆ เพื่อที่จะได้นำภาพวีดีโอคลิปดังกล่าวมาสร้างโมเดลที่มีความถูกต้องแม่นยำ

ทำการแปลงวีดีโอคลิปเป็นภาพที่ละเฟรมโดยใช้โค้ดจากโปรแกรมไพทอน ให้ทำงานในสภาพแวดล้อมเสมือนบนแพลตฟอร์ม Google Colab ซึ่งการทำงานจำเป็นต้องติดตั้งไลบรารีจำนวนหลายตัว เพื่อให้โค้ดไพรทอรนสามารถงานได้อย่างสมบูรณ์ เช่น TensorFlow เป็นไลบรารี่ที่พัฒนาด้วยบริษัทกูเกิลซึ่งใช้สำหรับการพัฒนาด้าน Machine Learning โดยเฉพาะ, OPENCV-Python เป็นไลบรารี่ที่ใช้ทำงานกับรูปภาพและการทำงานแบบเรียวทาม, Matplotlib เป็นไลบรารี่ที่ใช้ในการสร้างกราฟ

```
pip install tensorflow==2.4.0
pip install keras==2.4.3 numpy==1.19.3 pillow==7.0.0 scipy==1.4.1 h5py==2.10.0
matplotlib==3.3.2 opencv-python keras-resnet==0.2.0
pip install opencv-python
```

รูปที่ 3.9 การติดตั้งไลบรารี่ด้วยโค้ดด้านล่างใน Google Colab

```
import cv2
import os
listing = os.listdir(r'/content/drive/MyDrive/Count_bird/Clip')
count=1
for vid in listing:
    vid = r"/content/drive/MyDrive/Count_bird/Clip/"+vid
    vidcap = cv2.VideoCapture(vid)
    def getFrame(sec):
        vidcap.set(cv2.CAP_PROP_POS_MSEC,sec*500)
        hasFrames,image = vidcap.read()
        if hasFrames:
            cv2.imwrite("/content/drive/MyDrive/Count_bird/Frame2/"+str(count)+".jp
g", image) # Save frame as JPG file
        return hasFrames
    sec = 0
    frameRate = 1 # Change this number to 1 for each 1 second

    success = getFrame(sec)
    while success:
        count = count + 1
```

รูปที่ 3.10 โค้ดสำหรับแบ่งภาพเป็นเฟรมจากวีดีโอคลิป

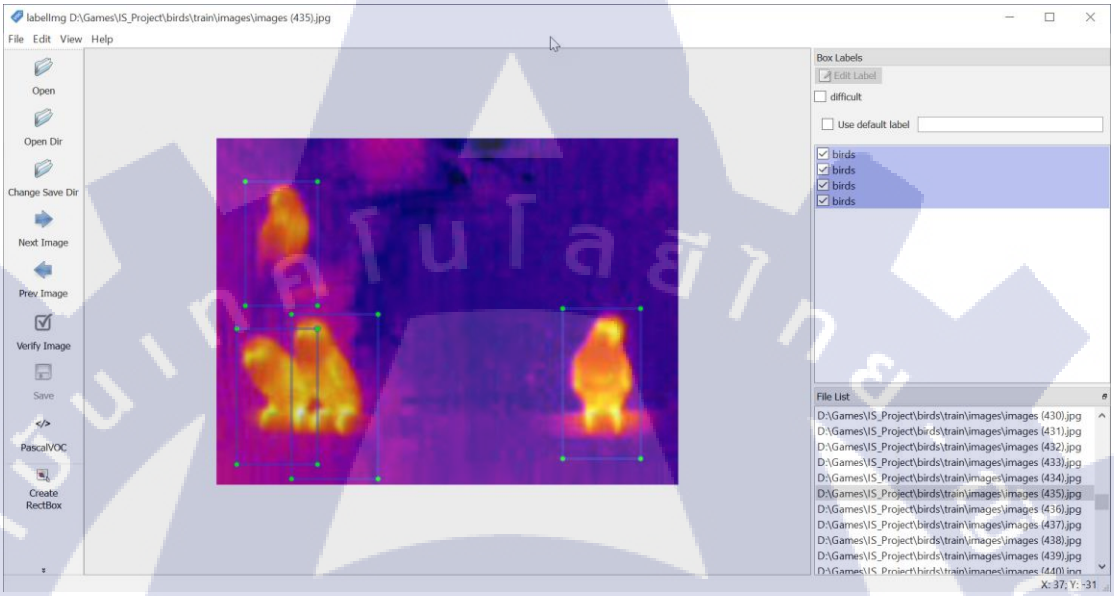
หลังจากที่โค้ดทำงานเรียบร้อยแล้วจะได้ภาพสำหรับเทรนในการทำโมเดล จำนวนภาพจะมากหรือน้อยอยู่ที่ความยาวของวีดีโอคลิปและจำนวนคลิปที่ได้ถ่ายไว้

3.3.5 การทำ Label Image

เมื่อได้ภาพที่ต้องการเทรนโมเดลแล้วจำเป็นต้องทำการระบุตำแหน่งของ Object ในภาพเพื่อใช้ในการเทรนภาพเข้าอัลกอริทึม โดยทางผู้วิจัยได้ใช้โปรแกรม LabelIMG เพื่ออำนวยความสะดวกในการทำงานด้านการระบุตำแหน่งของ Object สำหรับจำนวนภาพที่ใช้ในการเทรนข้อมูลยิ่งมีจำนวนภาพมากความแม่นยำของอัลกอริทึมย่อมมีมากไปด้วย โดยภาพที่ทำ Label Image แล้ว ทางผู้วิจัยจะใช้หลักของ Pareto [20] โดยจะแบ่งจำนวน 80 เปอร์เซ็นต์เพื่อใช้ในการเทรนโมเดล และจำนวนที่เหลืออีก 20 เปอร์เซ็นต์ ใช้สำหรับทำการตรวจสอบหรือการ Validation โดยทางผู้วิจัยได้ใช้ภาพจำนวน 609 ภาพในการเทรนโมเดล แบ่งเป็นภาพสำหรับเทรนโมเดลจำนวน 480 ภาพ และอีก 209 ภาพใช้สำหรับตรวจสอบ จากจำนวนภาพทั้งหมด 956 ภาพซึ่งบางภาพมีความเบลอไม่ชัดเจน

บางภาพไม่มีภาพนกในภาพ หรือบางภาพมีภาพนกใกล้กล้องจนเกินไป จึงไม่สามารถนำมาใช้ได้ จำนวน 347 ภาพ

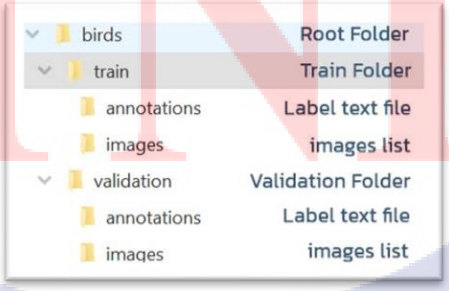
โปรแกรม LabelIMG จะเป็นโปรแกรมที่ช่วยกำหนดตำแหน่งวัตถุบนภาพในรูปแบบแกน X และแกน Y โดยการตีกรอบบนภาพ



รูปที่ 3.11 หน้าจอโปรแกรม LabelIMG

3.3.6 <u>การเทรนภาพเข้าอัลกอริทึม YOLOv3</u>

นำภาพที่ได้หลังจากทำการตีกรอบ Object รูปภาพด้วยโปรแกรม LabelIMG แล้วมาแบ่งในสัดส่วนที่กำหนด โดยนำภาพและเท็กซ์ไฟล์มาจัดให้อยู่ในโฟลเดอร์ตามรูปที่ 3.6



รูปที่ 3.12 โครงสร้างโฟลเดอร์หลังจากLabelIMG

อัพโหลดภาพทั้งหมดที่ได้เตรียมไว้ทั้งหมดขึ้นกูเกิลไดรฟ์เพื่อใช้ในการเทรนโมเดลด้วย Google Colab (Google Colaboratory) ซึ่งเป็นบริการของบริษัทกูเกิล โดยได้ทำการติดตั้งไลบรารี่ ImageAI [21] เพิ่มเติม

ImageAI เป็นไลบรารี่ถูกพัฒนาโดย MOSES OLAFENWA ด้วยโปรแกรมไพทอนโดยนำไลบรารี่ TensorFlow มาใช้งานในส่วนของ Machine Learning สามารถตรวจจับหรือทำนายภาพจากคลิปวิดีโอหรือภาพด้วยชุดข้อมูลจาก COCO Dataset [22] ด้วยอัลกอริทึม RetinaNet, YOLOv3 และ TinyYOLOv3 ได้อย่างง่ายด้วยโปรแกรมมิ่งไม่กี่บรรทัด ทั้งยังสามารถสร้างโมเดลเพื่อใช้งานได้เองอีกด้วย

```
pip install imageai --upgrade
```

รูปที่ 3.13 ติดตั้งไลบรารี่ ImageAI

```
from imageai.Detection.Custom import DetectionModelTrainer

trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="/content/drive/MyDrive/birds")
trainer.setTrainConfig(object_names_array=["birds"], batch_size=4,
num_experiments=200,
train_from_pretrained_model="/content/drive/MyDrive/birds/pretrained-
yolov3.h5")
trainer.trainModel()
```

รูปที่ 3.14 โค้ดสำหรับการเทรนโมเดล

- setModelTypeAsYOLOv3 เป็นโมเดลที่เลือกใช้ในการเทรน
- data_directory ข้อมูลที่ทำการอัพโหลดขึ้นกูเกิลไดรฟ์
- batch_size จำนวนตัวอย่างในการฝึกอบรม ปกติค่าตั้งต้นจะเป็น 4 สามารถปรับเปลี่ยนเป็น 6, 8, 12 ได้ แต่ควรให้สัมพันธ์กับ GPU ที่เลือกใช้
- num_experiments จำนวนรอบที่ให้ระบบทำการเทรนโมเดล

ในขั้นตอนนี้อาจจะใช้เวลาพอสมควรขึ้นอยู่กับจำนวนภาพและ num_experiments ที่ได้ระบุไว้ ควรเตรียมอุปกรณ์คอมพิวเตอร์ให้พร้อม ควรตั้งค่า Power & Sleep ให้เป็น Never ป้องกันเครื่องคอมพิวเตอร์ดับขณะทำงาน หลังจากทำการเทรนโมเดลเรียบร้อยแล้ว ต้องทำการประเมินความแม่นยำของโมเดลที่ได้แต่ละชุดว่าชุดใดมีความแม่นยำมากที่สุดเพื่อสามารถจะได้นำไปใช้ได้อย่างแม่นยำ

```
from imageai.Detection.Custom import DetectionModelTrainer

trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="/content/drive/MyDrive/birds")
metrics =
trainer.evaluateModel(model_path="/content/drive/MyDrive/birds/models",
json_path="/content/drive/MyDrive/birds/json/detection_config.json",
iou_threshold=0.5, object_threshold=0.3, nms_threshold=0.5)
print(metrics)
```

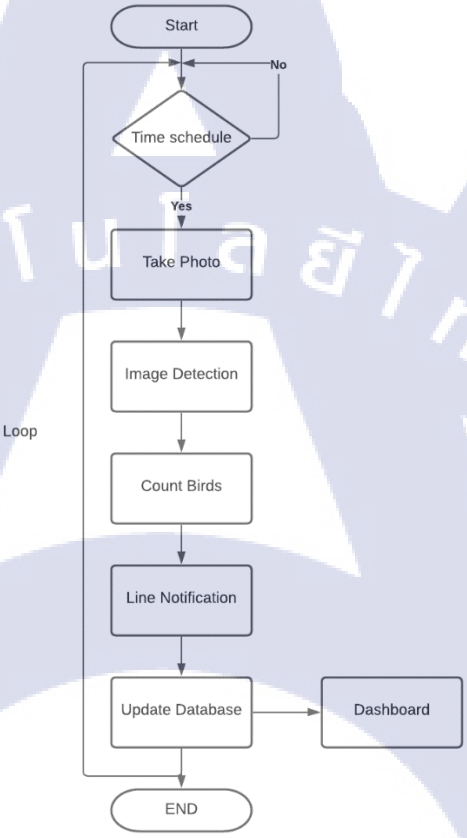
รูปที่ 3.15 โค้ดการประเมินความแม่นยำของชุดโมเดลที่ได้จากการเทรนรูปภาพ

3.3.7 พัฒนาโปรแกรมเทคนิคการตรวจนับ

ในการพัฒนาแอพพลิเคชั่นได้ใช้โปรแกรมไพทอนในการพัฒนาเนื่องจากเป็นภาษาระดับสูง สามารถทำงานได้บนหลายระบบปฏิบัติการ สนับสนุนแนวคิดแบบ Object-Oriented Programming มีไลบรารี่ด้าน Machine Learning รองรับมีความสามารถในการประมวลผลได้อย่างรวดเร็ว เป็นภาษาที่เข้าใจง่าย สามารถทำงานร่วมกับบอร์ด Raspberry Pi และสามารถเขียนคำสั่งควบคุมอุปกรณ์เซนเซอร์ที่ได้เชื่อมต่อไว้ได้อีกด้วย

ขั้นตอนการทำการงานของโปรแกรมตรวจนับจำนวนนกแอ่นกินรังด้วย YOLO Object Detection ผ่านกล้องถ่ายภาพความร้อน เมื่ออุปกรณ์เริ่มทำงานจะให้ถ่ายภาพนกด้วยกล้องถ่ายภาพความร้อนตามเวลาที่ได้ตั้งเวลาไว้ด้วยคำสั่ง Crontab -e ใน Raspberry Pi จากนั้นจะนำภาพที่ได้จากการถ่ายภาพ เข้าอัลกอริทึม YOLOv3 เพื่อทำการตรวจนับจำนวนนก โดยการนับตามจำนวนการกรอบที่ตีล้อมภาพนกที่พบ ทำการเก็บข้อมูลที่นับไว้ที่ตัวแปรที่กำหนด เมื่อได้ตรวจนับครบแล้วโปรแกรมจะทำการส่งข้อมูลที่นับได้ กับภาพถ่ายที่ได้ทำการตีกรอบนกที่ตรวจพบไปทางโปรแกรมไลน์ที่ได้ลงทะเบียนไว้ หลังจากที่ส่งข้อมูผ่านทางโปรแกรมไลน์ โปรแกรมจะทำการบันทึกข้อมูลไว้ที่กูเกิล

ชีท (Google Sheets) โดยในกูเกิล ชีท (Google Sheets) จะเก็บข้อมูลวันเดือนปี เวลา และจำนวนที่นับได้ นำข้อมูลที่ได้จัดเก็บไว้มาแสดงรายงานในรูปแบบวิชวลไลเซชัน ด้วยโปแกรม Power Bi จากนั้นโปรแกรมจะรอจนถึงช่วงเวลารอบถัดไปเพื่อทำงานซ้ำต่อไป โดยแสดงผังการทำงานของโปรแกรมตามรูปที่ 3.7



รูปที่ 3.16 แผนผังการทำงานของโปรแกรม

ผู้วิจัยได้พัฒนาโค้ดด้วยไพทอนเวอร์ชัน 3.7.3 ในตัว Raspberry Pi โดยแบ่งการทำงานของเป็น 4 ส่วนคือ ส่วนแรก เรียกโมดูลกล้องให้ทำการถ่ายภาพ เมื่อถึงเวลาที่ระบุไว้ และบันทึกชื่อไฟล์เป็นวันและเวลาปัจจุบัน ส่วนที่สองหลังจากถ่ายภาพแล้วให้ทำการ Detection รูปนกบนภาพถ่ายด้วยโมเดลที่ได้จัดเตรียมไว้ในข้างต้น และทำการนับจำนวนนก พร้อมทั้งตีกรอบรูปนกบนภาพ จากนั้นทำการบันทึกภาพไว้ที่โฟลเดอร์ที่กำหนด ส่วนที่สามทำการส่งภาพและจำนวนที่นกนับไปยังโปรแกรมไลน์ที่ลงทะเบียนไว้ ส่วนที่สี่นำข้อมูล วันเวลา จำนวนนก ลำดับชื่ออุปกรณ์ และภาพส่งก็บบันทึกไว้ที่กูเกิลชีท เพื่อนำข้อมูลที่ได้ไปใช้ในการแสดงผลด้วยโปรแกรม Power BI

```python
# import library
import cv2
import time
import os

# set path save images
execution_path = os.getcwd()+"/ImgTake"
# define a video capture object
vid = cv2.VideoCapture(0)
result = True
# set image name with current date time
img_name = time.strftime("%Y%m%d%H%M%S")
while(result):
    # Capture the video frame
    ret, frame = vid.read()
    cv2.imwrite(os.path.join(execution_path , img_name +'.jpg'),frame)
    result = False
# After the loop release the cap object
vid.release()
# Destroy all the windows
cv2.destroyAllWindows()
print("Take Image Complete")
```

รูปที่ 3.17 โค้ดการถ่ายภาพ

```
#--- Detection Image ------
from imageai.Detection.Custom import CustomObjectDetection

input_image= "/home/pi/Desktop/Count/ImgTake/"+img_name+".jpg"
output_image= "/home/pi/Desktop/Count/ImgOutput/"+img_name+".jpg"

detector = CustomObjectDetection()
detector.setModelTypeAsYOLOv3()
detector.setModelPath("/home/pi/Count/Models/detection_model-ex-017--loss-
0011.473.h5")
detector.setJsonPath("/home/pi/Count/json/detection_config.json")
detector.loadModel()
detections = detector.detectObjectsFromImage(input_image, output_image)

count = 0
for detection in detections:
    print(detection["name"], " : ", detection["percentage_probability"], " : ", detectio
n["box_points"])
    count = count+1

print(output_image)
print("Detection Image Complete",count)
```

รูปที่ 3.18 โค้ดการทำ Image Detection

```
#--- Send line -----
import requests
import urllib.parse
import sys

LINE_ACCESS_TOKEN="PgQrWJ0WXR655eTcgvEmGic9V4aUMmE5ls5TWAlr56Y"
url = "https://notify-api.line.me/api/notify"
file = {'imageFile':open(output_image,'rb')}
data = ({
        'message': count
    })
LINE_HEADERS = {"Authorization":"Bearer "+LINE_ACCESS_TOKEN}
session = requests.Session()
r=session.post(url, headers=LINE_HEADERS, files=file, data=data)
print(r.text)
print("Send line Complete")
```

รูปที่ 3.19 โค้ดการส่งข้อมูลทางไลน์

3.3.8 พัฒนา Dashboard สำหรับแสดงข้อมูล

การพัฒนา Dashboard เพื่อแสดงผลข้อมูลเราจะใช้โปรแกรม Power BI สำหรับนำข้อมูลมาแสดงผลเนื่องสามารถใช้งานได้สะดวก สามารถติดต่อกับฐานข้อมูลได้หลากหลาย ทั้งยังสามารถแสดงผลผ่านทางเว็บเบราว์เซอร์หรือหน้าจอมือถือ และมีเวอร์ชันที่ไม่เสียค่าบริการสำหรับใช้งาน ในส่วนของการจัดเก็บข้อมูลได้จัดเก็บในรูปแบบตารางข้อมูลในกูเกิลชีทซึ่งมีความสะดวกในการใช้งาน ทั้งผู้วิจัยเห็นว่ายังสามารถนำข้อมูลไปใช้ได้เลยโดยไม่ต้องมีความรู้ทางด้านเครื่องมือต่างๆ ที่ซับซ้อนมากกนัก โดยการจัดเก็บข้อมูลจากอุปกรณ์ Raspberry Pi ที่ได้ส่งข้อมูลเวลา จำนวนนกที่ตรวจพบ มาเก็บไว้

โดยในส่วนของกูเกิลชีทที่ได้สร้างขึ้นเพื่อจัดเก็บข้อมูลจากโปรแกรมตรวจนับจำนวนนก โดยแบ่งเป็นจำนวน 3 คอลัมน์ ประกอบด้วย Data เก็บข้อมูล วัน เดือน ปี ในการตรวจนับและส่งข้อมูล TIME คือเวลาในการตรวจนับและส่งข้อมูล และ Amount คือจำนวนที่โปรแกรมได้ตรวจพบและนับจำนวนนกได้ส่งมาบันทึก โดยจะบันทึกเรียงต่อลงมาเรื่อยๆ

	A	B	C
1	Date	TIME	Amount
2	2021-02-01	18:30:51	87
3	2021-02-01	19:30:04	70
4	2021-02-01	20:30:05	85
5	2021-02-02	18:30:51	72
6	2021-02-02	19:30:04	80
7	2021-02-02	20:30:05	79
8	2021-02-03	18:30:51	69
9	2021-02-03	19:30:04	82
10	2021-02-03	20:30:05	96

รูปที่ 3.20 หน้าจอข้อมูลที่จัดเก็บในกูเกิลชีท

ในการบันทึกข้อมูลในกูเกิลชีท ที่ถูกส่งข้อมูลมาจาก Raspberry Pi จำเป็นต้องทำการเปิด API เพื่อทำการเชื่อมต่อระหว่างโค้ดโปรแกรมไพทอนใน Raspberry Pi กับทางกูเกิลชีทที่ได้สร้างไว้

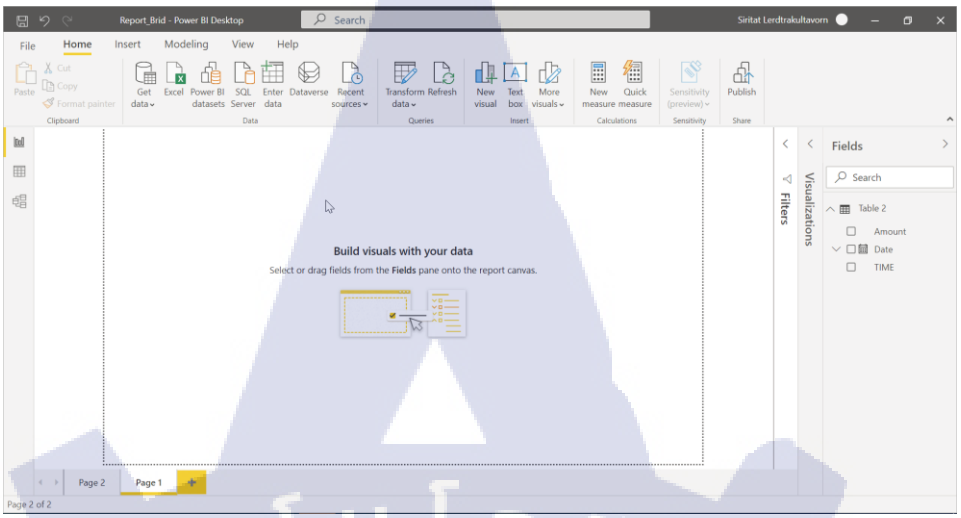
```
#--- Insert Data to Google Sheet ------
import gspread
from datetime import datetime
import time
from oauth2client.service_account import ServiceAccountCredentials

# define the scope
scope = ["https://spreadsheets.google.com/feeds",'https://www.googleapis.com/auth/spreadsheets',"https://www.googleapis.com/auth/drive.file","https://www.googleapis.com/auth/drive"]

# add credentials to the account
creds = ServiceAccountCredentials.from_json_keyfile_name('/content/drive/MyDrive/Colab Notebooks/PythonAPI.json', scope)
```

รูปที่ 3.21 โค้ดการบันทึกข้อมูลขึ้นเก็บที่กูเกิลชีท

```
# authorize the clientsheet
client = gspread.authorize(creds)
# get the instance of the Spreadsheet
sheet = client.open('Birds_Count').get_worksheet(0)
data = sheet.get_all_records()
data = sheet.row_count
os.environ['TZ'] = 'Asia/Bangkok'
time.tzset()
datestring = datetime.strftime(datetime.now(), '%Y-%m-%d')
t = time.localtime()
current_time = time.strftime("%H:%M:%S", t)
row = [datestring,current_time,count]
index = sheet.row_count
sheet.insert_row(row, index)

print("Save Data Complete")
```

รูปที่ 3.22 โค้ดการบันทึกข้อมูลขึ้นเก็บที่กูเกิลชีท (ต่อ)

หลังจากได้บันทึกข้อมูลแล้วจะนำข้อมูลจากกูเกิลชีทมาจัดทำรายงานโดยแสดงผลในรูปแบบของวิชวลไลเซชันด้วยโปรแกรม Power BI ซึ่งจะนำข้อมูลมาแสดงในรูปแบบกราฟที่ง่ายต่อการดูข้อมูลปัจจุบัน หรือข้อมูลย้อนหลัง ผู้ใช้งานสามารถเลือกช่วงเวลาในการดูจำนวนนกที่ได้ทำการบันทึกข้อมูลไว้ด้วยตัวเอง ทั้งยังสามารถดูข้อมูลผ่านอุปกรณ์ต่างๆ เช่น คอมพิวเตอร์ มือถือ หรือแท็บเล็ตเป็นต้น



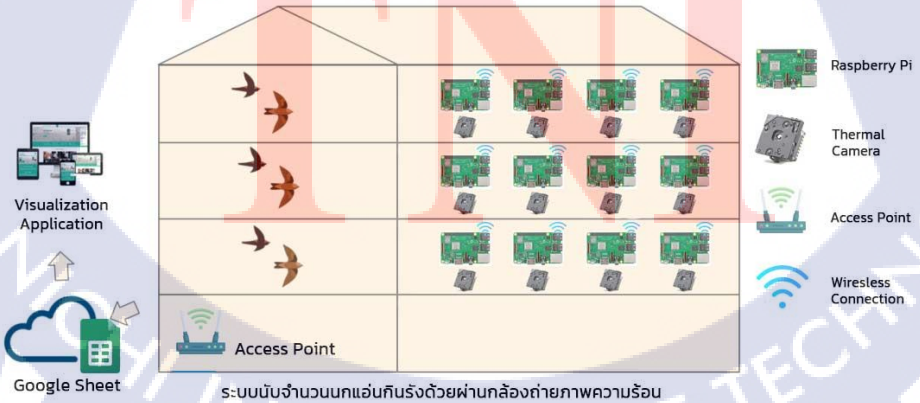
รูปที่ 3.23 หน้าจอโปรแกรม Power Pi

3.4 ขั้นตอนที่ 4 วิเคราะห์ผลการทดลอง

ผลลัพธ์จากการทดลองในขั้นตอนที่ 3 ผู้วิจัยจะนำข้อมูลมาทำการตรวจสอบความถูกต้องของจำนวนนกในภาพถ่าย ที่ถ่ายด้วยกล้องตรวจจับความร้อน ความแม่นยำในการตั้งกล้องถ่ายภาพในระยะต่าง ๆ ตลอดจนการทำงานในการส่งข้อมูลเข้าจัดเก็บในกูเกิลชีท ออกรายงานในรูปแบบวิชวลไลเซชันและส่งข้อมูลแจ้งเตือนผ่านทางโปรแกรมไลน์

3.5 ขั้นตอนที่ 5 การสรุปผลการวิจัย

ส่วนของการสรุปผลการวิจัย จะกล่าวถึงความแม่นยำในการตรวจนับจำนวนนก ตลอดจนข้อจำกัดที่เกิดจากการทดลอง



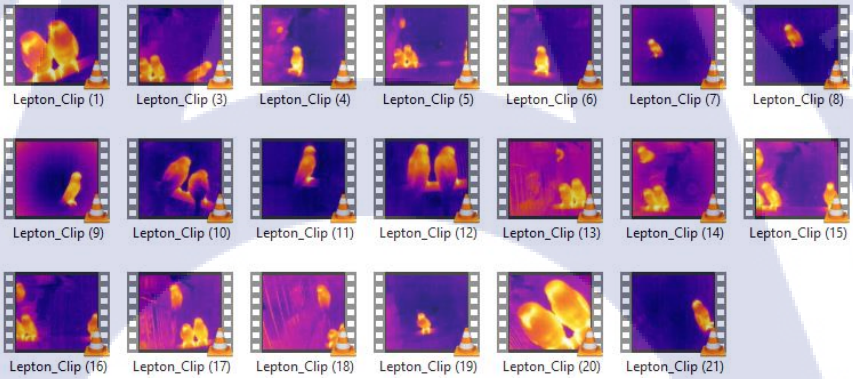
รูปที่ 3.24 ออกแบบการระบบตรวจนับจำนวนนกแอ่นกินรัง

บทที่ 4
ผลการวิจัย

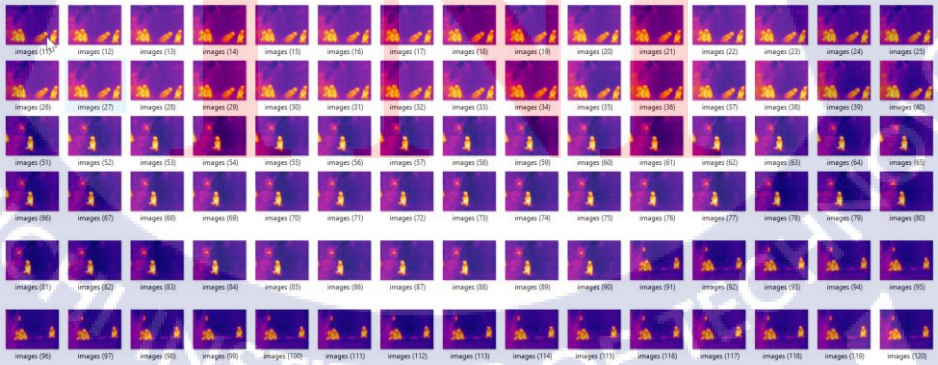
จากการศึกษาแนวคิด ทฤษฎี และงานวิจัยที่เกี่ยวข้องเพื่อเป็นความรู้สำหรับการศึกษาในการวิจัยครั้งนี้ ผู้วิจัยได้จำลองการวิจัยเพื่อศึกษาเกี่ยวกับเทคนิคการตรวจนับนกนางแอ่นกินรังด้วยการนำโมเดลที่ผู้วิจัยได้ทำการเทรนโมเดลขึ้น มาทำการทดลองความแม่นยำจากระยะการถ่ายภาพนกหงส์หยกจำนวน 4 ตัว ในกรงขนาด 120 x 60 x 66 เซนติเมตร

4.1 ผลการถ่ายภาพด้วยกล้องถ่ายภาพความร้อน

ผลจากการเตรียมภาพถ่ายนก ด้วยการแบ่งเฟรมภาพจากคลิปวิดีโอจำนวน 21 คลิป ออกมาเป็นภาพ และทำการคัดเลือกภาพที่มีความชัดเจนจำนวน 609 ภาพจากภาพทั้งหมด 956 ภาพเพื่อนำมาทำการเทรนโมเดล



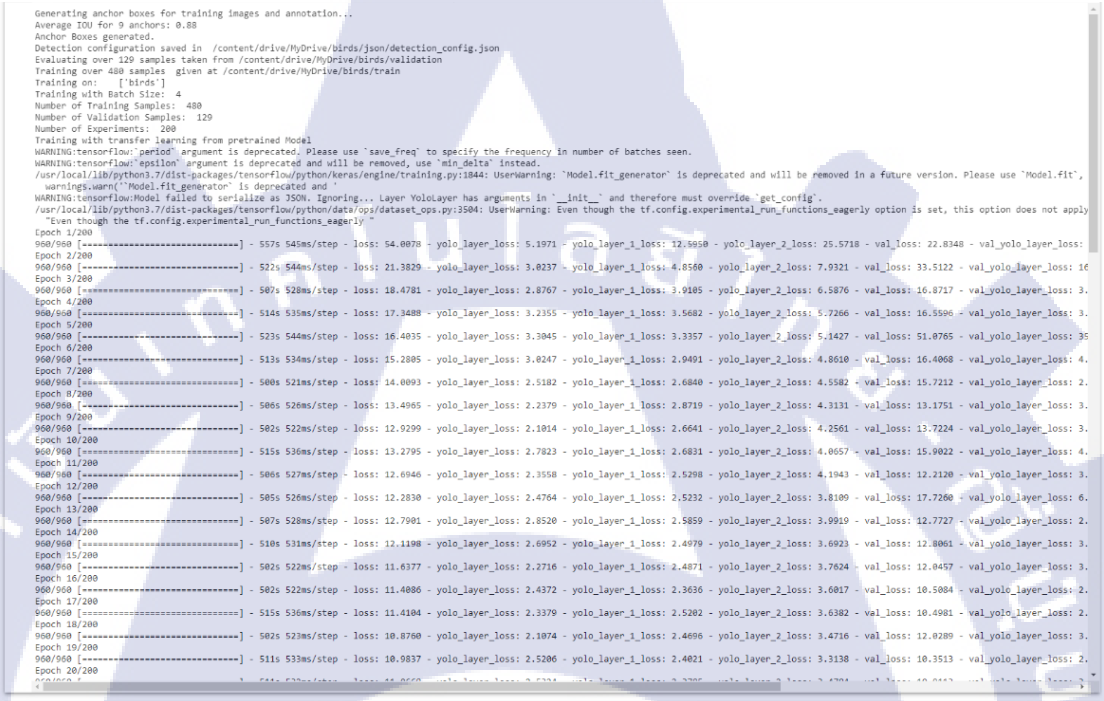
รูปที่ 4.1 ภาพคลิปวิดีโอ



รูปที่ 4.2 ภาพหลังจากที่ได้แบ่งเฟรมจากวีดีโอคลิป

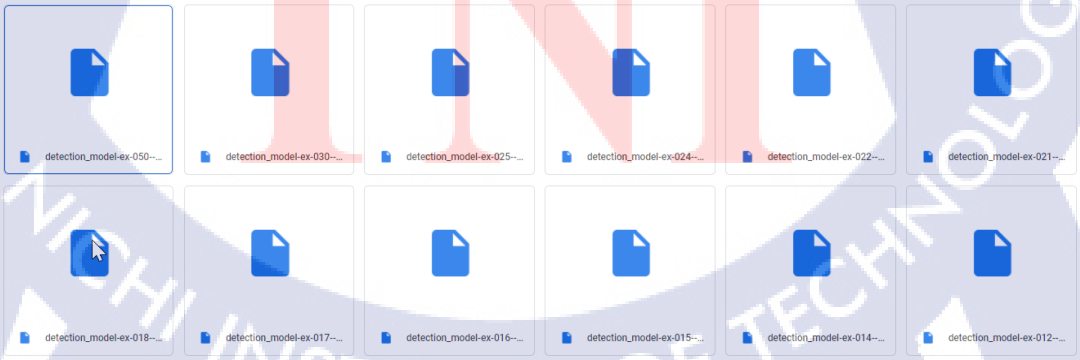
4.2 ผลการเทรนนิ่งภาพที่ถ่ายด้วยกล้องถ่ายภาพความร้อน

ภาพที่ได้มาทำ Labeling และเทรนโมเดล ด้วยอัลกอริทึม Yolov3 หลังจากเทรนโมเดลแล้ว จะได้โมเดลต้นแบบจำนวนหลายโมเดล ต้องทำการประเมินความแม่นยำเพื่อหาโมเดลที่แม่นยำที่สุด เพื่อนำมาใช้ในการทำ Detection



รูปที่ 4.3 แสดงการเทรนนิ่งโมเดล

ชุดโมเดลที่ได้หลังจากการเทรนโมเดลด้วยภาพถ่ายที่ได้เตรียมไว้



รูปที่ 4.4 ชุดโมเดลหลังจากทำการเทรน

ผลที่ได้จากการตรวจสอบความแม่นยำของโมเดล โดยจะเลือกชุดโมเดลที่มีค่า mAP (Mean Average Precision) มากที่สุดจากชุดโมเดลทั้งหมดที่ได้เทรนออกมา [23]

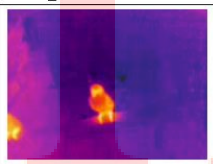
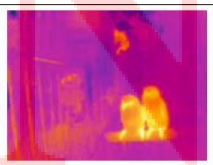
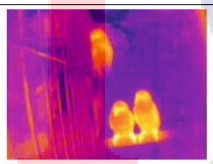
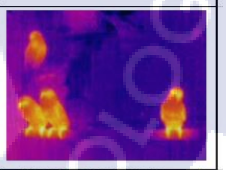
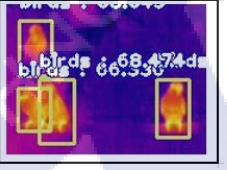
```
WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
Model File:  /content/drive/MyDrive/birds/models/detection_model-ex-001--loss-0033.955.h5

Evaluation samples:  129
Using IoU:  0.5
Using Object Threshold:  0.3
Using Non-Maximum Suppression:  0.5
birds: 0.5054
mAP: 0.5054
==============================
WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
Model File:  /content/drive/MyDrive/birds/models/detection_model-ex-002--loss-0020.331.h5

Evaluation samples:  129
Using IoU:  0.5
Using Object Threshold:  0.3
Using Non-Maximum Suppression:  0.5
birds: 0.5951
mAP: 0.5951
==============================
WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
Model File:  /content/drive/MyDrive/birds/models/detection_model-ex-003--loss-0017.841.h5

Evaluation samples:  129
Using IoU:  0.5
Using Object Threshold:  0.3
Using Non-Maximum Suppression:  0.5
birds: 0.9231
mAP: 0.9231
==============================
WARNING:tensorflow:No training configuration found in the save file, so the model was *not* compiled. Compile it manually.
Model File:  /content/drive/MyDrive/birds/models/detection_model-ex-004--loss-0016.713.h5
```

รูปที่ 4.5 ผลลัพธ์จากการตรวจสอบความแม่นยำของโมเดล

4.3 ผลการนำโมเดลมาใช้งานจริงในการถ่ายภาพ

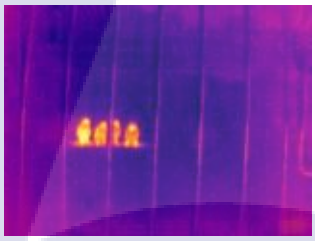
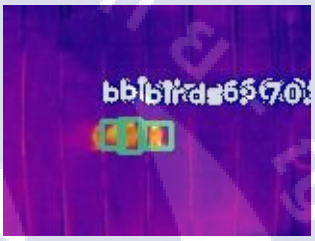
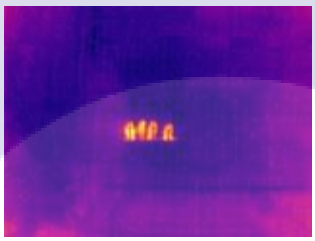
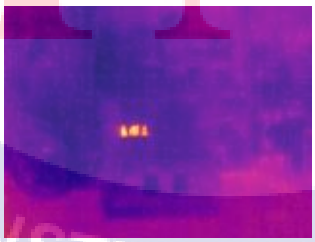
นำโมเดลที่ได้จากการเทรนมาทดสอบถ่ายภาพนกหงส์หยกเพื่อทดสอบความแม่นยำของอัลกอริทึมที่ได้จัดทำขึ้น โดยการถ่ายภาพนกหงส์หยก 1 ตัว 2 ตัว 3 ตัว และ 4 ตัว ตามลำดับ

ตารางที่ 4.1 แสดงภาพถ่ายกับภาพที่ทำการ Detection

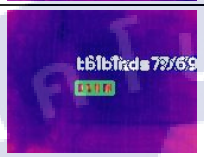
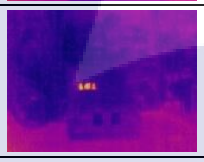
No	1	2	3	4
ภาพถ่าย				
ภาพหลังทำ Detection				

ผลการถ่ายภาพนกในกรงด้วยระยะที่แตกต่างกันที่ 0.5 เมตร 1 เมตร 1.5 เมตร 2.0 เมตร และ 2.5 เมตร โดยนำมาเปรียบเทียบในตารางที่ 4.1 และตารางที่ 4.2

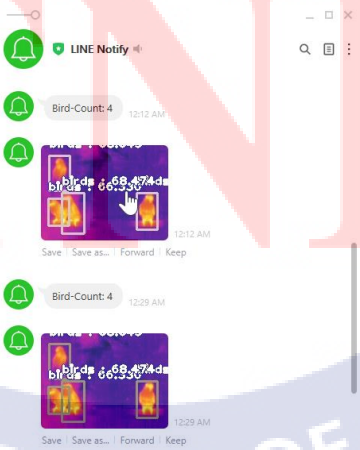
ตารางที่ 4.2 แสดงภาพเปรียบเทียบภาพถ่ายกับภาพที่ทำการ Detection

ระยะถ่ายภาพ (เมตร)	ภาพถ่าย	ภาพหลังทำ Detection
0.5		
1		
1.5		
2		
2.5		

ตารางที่ 4.3 แสดงภาพและผลการทำ Detection

ระยะถ่ายภาพ (เมตร)	ภาพที่ทำ Detection	Output จากการทำ Detection
0.5		return 1. / (1. + np.exp(-x)) birds : 74.24904704093933 : [12, 5, 37, 42] birds : 58.45000743865967 : [0, 7, 20, 44] /home/pi/Desktop/Count/Output/0.5M1.jpg Detection Image Comple 2 >>> %Run Imgdetect.py
1		"Even though the tf.config.experimental_run_functions_ea birds : 61.40317916870117 : [51, 60, 62, 74] birds : 66.85764193534851 : [60, 59, 71, 76] birds : 70.07520794868469 : [73, 61, 87, 74] /home/pi/Desktop/Count/Output/1M1.jpg Detection Image Complete 3 >>> %Run Imgdetect.py
1.5		"Even though the tf.config.experimental_run_functions_ea birds : 62.35275864601135 : [58, 60, 68, 75] birds : 64.98007178306658 : [67, 58, 77, 76] birds : 69.41078901290894 : [76, 60, 89, 75] birds : 75.51071643829346 : [83, 62, 97, 77] /home/pi/Desktop/Count/Output/1.5M.jpg Detection Image Complete 4
2		data functions. tf.data functions are still traced and execu "Even though the tf.config.experimental_run_functions_eage birds : 59.67540740966797 : [63, 59, 80, 67] /home/pi/Desktop/Count/Output/2M.jpg Detection Image Complete 1 >>> %Run Imgdetect.py
2.5		/usr/local/lib/python3.7/dist-packages/tensorflow/python/ though the tf.config.experimental_run_functions_eagerly o data functions. tf.data functions are still traced and ex "Even though the tf.config.experimental_run_functions_e /home/pi/Desktop/Count/Output/2.5M.jpg Detection Image Complete 0 >>>

ผลการส่งข้อความและภาพที่ได้ทำการตรวจพบและนับจำนวนนก ผ่านทางโปแกรมแอปพลิเคชันไลน์ โดยจะเป็นข้อความผลการตรวจนับจำนวนนก 1 บรรทัดและภาพที่ได้ทำการตีกรอบที่ตรวจพบจำนวนนก อีกจำนวน 1 ภาพ



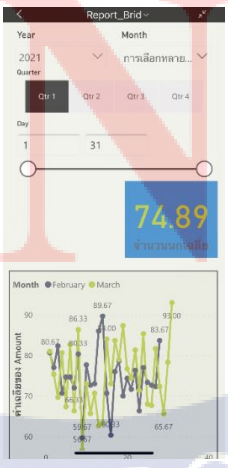
รูปที่ 4.6 การส่งข้อมูลจำนวนนก และภาพทางโปแกรมแอปพลิเคชันไลน์

4.4 ผลการนำข้อมูลการตรวจพบจำนวนนกมาแสดงในรูปแบบของวิชวลไลเซชัน

ผลการนำข้อมูลที่ได้บันทึกไว้ในกูเกิล ชีท (Google Sheets) มาทำรายงานโดยแสดงผลในรูปแบบของวิชวลไลเซชันด้วยโปรแกรม Power BI ซึ่งรูปแบบรายงานที่แสดงผลสามารถที่จะทำการระบุปี ไตรมาส เดือน และช่วงวันที่ที่ต้องการแสดงผลข้อมูล โดยกราฟเส้นที่ปรากฏจะเป็นข้อมูลของจำนวนนกที่นักได้ในแต่ละเดือน ทั้งยังสามารถแสดงจำนวนนกเฉลี่ยของข้อมูลล่าสุดได้อีกด้วย

รูปที่ 4.7 แสดงผลรายงานผ่านโปรแกรม Power BI

นอกจากนี้ รายงานที่แสดงผลยังได้ออกแบบให้สามารถเข้าดูข้อมูลหรือรายงานผ่านอุปกรณ์ต่างๆ เช่น คอมพิวเตอร์ โน๊ตบุ๊ค แท็บเล็ต หรือมือถือได้อีกด้วย โดยมีหน้าแสดงผลหน้าจอดังรูปที่ 4.8



รูปที่ 4.8 แสดงผลรายงานด้วยโปรแกรม Power Bi ผ่านอุปกรณ์มือถือ

บทที่ 5
บทสรุป อภิปราย และข้อเสนอแนะ

การศึกษาวิจัยเรื่องเทคนิคการตรวจนับนกนางแอ่น ด้วยกล้องตรวจจับความร้อน ซึ่งการวิจัยนี้มีวัตถุประสงค์ตามบทที่ 1 ได้นำเสนอข้อสรุปตามลำดับดังนี้

5.1 สรุปผลการวิจัย

5.2 อภิปรายผลวิจัย

5.3 ปัญหาและอุปสรรคในการทำวิจัย

5.4 ข้อเสนอแนะงานวิจัย

5.1 สรุปผลการวิจัย

5.1.1 สามารถออกแบบอุปกรณ์สำหรับถ่ายภาพจากกล้องตรวจจับความร้อนได้ โดยใช้บอร์ด Raspberry Pi 3 B+ ร่วมกับกล้องถ่ายภาพร้อนของ FILR รุ่น Lepton 3.5 ซึ่งสามารถถ่ายคลิปวีดีโอ หรือถ่ายภาพที่มีขนาด 160 x 120 พิกเซล โดยกล้องรุ่นนี้จำเป็นต้องต่อกับบอร์ดก่อนที่จะเชื่อมต่อกับ Raspberry Pi ผ่านทางช่องยูเอสบี สามารถเรียกใช้กล้องถ่ายภาพความร้อนได้จากฟังก์ชันในไลบรารี่ OPENCV (cv2.VideoCapture(vid)) ได้เลย

5.1.2 สามารถนำภาพจากกล้องถ่ายภาพความร้อนมาประมวลผลด้วยอัลกอริทึม YOLOv3 ด้วยโมเดลที่ได้เตรียมไว้ และทำการตรวจสอบความแม่นยำของโมเดลด้วยการทดลอง ที่เตรียมไว้โดยข้อมูลจากการทดลองสามารถสรุปได้ว่าระยะการถ่ายภาพนกที่ระยะ 0.5 เมตร 1 เมตร และ 1.5 เมตรจากตัว มีความแม่นยำในการตรวจพบและผลการนับจำนวนนกมีความถูกต้อง และแม่นยำ แต่พอเป็นระยะที่ 2 เมตร การตรวจพบยังสามารถตรวจพบนกได้ แต่ขาดความแม่นยำ ไม่สามารถตรวจนับจำนวนนกได้ถูกต้อง สุดท้ายที่ระยะ 2.5 เมตร ไม่สามารถทำการตรวจพบหรือนับจำนวนนกได้เลย ดังนั้นระยะการตั้งกล้องที่เหมาะสมในการใช้งานอยู่ที่ระยะ 1 เมตร เป็นระยะการตรวจนับนกที่ถูกต้องและแม่นยำที่สุด

5.1.3 สามารถนำข้อมูลที่ได้จากการตรวจนับจำนวนนกมาแสดงผลในรูปแบบของ วิชวลไลเซชันผ่านโปรแกรม Power BI ซึ่งสามารถเผยแพร่รายงานที่ได้จัดทำขึ้นผ่านทางคลาวด์ของบริษัทไมโครซอฟ และสามารถเรียกดูรายงานผ่านทางเบราว์เซอร์ จากอุปกรณ์โน๊ตบุ๊ค แท็บเล็ต หรือมือถือได้ โดยรายงานสามารถที่จะเลือกช่วงเวลา ตามปี ไตรมาส เดือน หรือวัน และรายงานยังสามารถแสดงผลเฉลี่ยจากข้อมูลที่เลือกได้

Distance (m)	0.5	1	1.5	2	2.5
Detection	✓	✓	✓	✓	✗
Accuracy	✓	✓	✓	✗	✗

✓ Completed
✗ Incompleted

รูปที่ 5.1 สรุปผลการทดลอง

5.2 อภิปรายผลวิจัย

5.2.1 จากผลการทดลองผลที่ได้จะเห็นว่าระยะการถ่ายภาพนกที่สามารถนำมาทำการตรวจพบ และตรวจนับได้อย่างแม่นยำจะไม่เกินระยะ 1 เมตร ซึ่งเป็นระยะที่ไม่มาก โดยส่วนหนึ่งเป็นเพราะวัตถุหรือนกที่ทำการถ่ายภาพ มีขนาดเล็ก อุณหภูมิที่สัมผัสได้มีปริมาณไม่มาก และเมื่อต้องถ่ายภาพผ่านกรงนก ซึ่งมีซี่กรงนกกันอยู่จึงอาจทำให้การตรวจพบมีความไม่ถูกในระยะที่ห่างออกมา อีกทั้งคุณสมบัติของกล้องเองที่สามารถถ่ายภาพได้ที่ความละเอียด 160 x 120 พิกเซล อาจจะไม่เพียงพอสำหรับการถ่ายภาพในระยะที่ไกลออกมา

5.2.2 จากข้อจำกัดของโปรแกรม Power BI การทำงานที่มีปริมาณข้อมูลจำนวนมากๆ เมื่อจะต้องทำรีเฟรชข้อมูลโอกาสที่โปรแกรมจะค้างหรือไม่ตอนสนองเป็นไปได้สูงเพราะว่าโปรแกรม Power BI เมื่อทำการรีเฟรชข้อมูลโปรแกรมจะทำการโหลดข้อมูลใหม่ทั้งหมดแม้ว่าจะมีการแก้ไขข้อมูลเพียงเล็กน้อยก็ตาม อีกทั้งจำนวนข้อมูลที่ส่งกลับมาให้ Power BI นั้นจะต้องไม่มากกว่า 1 ล้าน Row บางคำสั่งสำคัญๆ ของ DAX ก็จะไม่สามารถใช้ได้โดยเฉพาะคำสั่งของพวก Time Intelligence รวมไปถึง Feature บางอย่างก็ไม่สามารถใช้ได้เช่นกัน อย่างเช่น Q&A และ QuickInsight จะไม่สามารถใช้งานได้เมื่อเราใช้ Power BI ในโหมด Direct Query และที่สำคัญที่สุดคือโหมดนี้มันค่อนข้างช้าในการเรียกและประมวลผล [24]

5.3 ปัญหาและอุปสรรคในการทำงานวิจัย

5.3.1 จากการออกแบบการทดลองปัญหาเรื่องความละเอียดของกล้องตรวจจับความร้อนก็เป็นปัจจัยสำคัญที่พบ เพราะมูลค่าของกล้องตรวจจับความร้อนที่มีความละเอียดสูงก็มีราคาสูงตามไปด้วย

5.3.2 การทดลองจำเป็นต้องทำโมเดลที่จัดทำขึ้นมาใช้โดยเฉพาะด้วยกล้องตรวจจับความร้อน ซึ่งไม่สามารถใช้ดาต้าโมเดลที่มีอยู่ในอินเตอร์เน็ตได้ และด้วยต้องถ่ายภาพในแสงที่ค่อนข้างมืดและการถ่ายภาพในระยะต่างๆ ทำให้จำต้อง หานกมาเลี้ยงเพื่อทำการทดลอง

5.3.3 การเปิดคอมพิวเตอร์เป็นระยะเวลา ควรมีการตั้งค่าของคอมพิวเตอร์ให้ทำงานตลอดเวลา เพื่อไม่ให้เข้าสู่โหมดประหยัดพลังงาน เพราะถ้าคอมพิวเตอร์ดับ ระหว่างการเทรนโมเดลซึ่งใช้ระยะเวลาค่อนข้างนานมาก จะต้องเริ่มการเทรนใหม่ตั้งแต่ต้น ทำให้เสียเวลาค่อนข้างมาก

5.4 ข้อเสนอแนะงานวิจัย

5.4.1 การทดลองนี้ได้จำลองโมเดลด้วยการใช้นกหงส์หยก และถ่ายภาพในสภาพแวดล้อมที่จัดทำขึ้น ซึ่งถ้าจะปรับนำงานวิจัยนี้ไปใช้จริง จำต้องทำการเทรนโมเดลจากนกนางแอ่นกินรังในสถานที่จริงให้ถูกต้อง นอกจากนี้ยังสามารถนำเทคนิคการตรวจจับ จากกล้องตรวจจับความร้อนไปปรับใช้ให้มีการแจ้งเตือน เมื่อมีการตรวจจับความร้อนจากศัตรูตามธรรมชาติของนกนางแอ่นกินรังที่บุกรุก หรือเข้ามาในฟาร์มนกได้

บรรณานุกรม

บรรณานุกรม

[1] Today, "*รังนก คุณประโยชน์มากมาย แพทย์ชี้ ช่วยรักษาสุขภาพ พร้อมงานวิจัยรองรับ*," workpointtoday.com [Online]. Available: https://workpointtoday.com/edible-birds-nests. [Accessed: January 6, 2021].

[2] DITP, "*จีนต้องการนำเข้ารังนกเพิ่มขึ้นรังนกสำเร็จรูปบรรจุขวดได้รับความนิยม(สคต.เซี่ยเหมิน)*," ditp.go.th [Online]. Available: https://www.ditp.go.th/ditp_web61/article_sub_view.php?filename=contents_attach/659553/659553.pdf&title=659553&cate=413&d=0. [Accessed: January 6, 2021].

[3] Thai PBS, "*ตัดตรวนบ้านนกแอ่น*," news.thaipbs.or.th [Online]. Available: https://news.thaipbs.or.th/content/302460. [Accessed: January 6, 2021].

[4] ยุทธพงค์ ดำศรีสุข, "ชีววิทยาการสืบพันธุ์ของนกแอ่นกินรัง Aerodramus germani OUSTALET ที่วัดสุทธิวาตวราราม อำเภอเมือง จังหวัดสมุทรสาคร," *วารสารสัตว์ป่าเมืองไทย*, ปีที่ 23, ฉบับที่ 1, หน้า 63 – 75, ธันวาคม 2559.

[5] Engineeringtoday, "*เมืองอัจฉริยะ (Smart City) คืออะไร และจะเกิดขึ้นได้อย่างไร*," engineeringtoday.net [Online]. Available: https://www.engineeringtoday.net/เมืองอัจฉริยะ-smart-city-คืออะไร/. [Accessed: January 6, 2021].

[6] Raspberry Pi, "*Raspberry Pi 3 Model B+*," raspberrypi.org [Online]. Available: https://www.raspberrypi.org/products/raspberry-pi-3-model-b-plus/. [Accessed: January 6, 2021].

[7] Food Wiki, "*Infrared Thermography / กล้องอินฟราเรด*," Foodnetworksolution.com [Online]. Available: http://www.foodnetworksolution.com/wiki/word/7260/ infrared-thermography-กล้องอินฟราเรด. [Accessed: January 6, 2021].

[8] Senna Labs, "*Objective Detection เทคโนโลยีตรวจจับวัตถุ*," sennalabs.com [Online]. Available: https://www.sennalabs.com/th/blogs/Objective-Detection. [Accessed: January 8, 2021].

[9] Towards Data Science, "*What's New in YOLO v3?*," towardsdatascience.com [Online]. Available: https://towardsdatascience.com/yolo-v3-object-detection-53fb7d3bfe6b. [Accessed: January 8, 2021].

[10] StackExchange, "*Any Use of Non-Rectangular-Shaped Kernels in Convolutional Neural Networks? Especially when Analyzing Game Boards,"* stats.stackexchange .com [Online]. Available: https://stats.stackexchange.com/ questions/235032/any-use-of-non-rectangular-shaped-kernels-in-convolutional-neural-networks-espe. [Accessed: January 10, 2021].

[11] S. Ekabut, "*ตอนที่ 01: Power BI คืออะไร*," thepexcel.com [Online]. Available: https://www.thepexcel.com/what-is-power-bi/. [Accessed: January 10, 2021].

[12] ร่มเกล้า ช่วยดู, "*สถาปัตยกรรมเพื่อการอยู่ร่วมกันระหว่างคนกับนก*," วิทยานิพนธ์ สถ.บ. (สถาปัตยกรรม), มหาวิทยาลัยศรีปทุม, กรุงเทพฯ, 2559.

[13] ปราโมทย์ ปัญญาโต, "การสร้างระบบตรวจนับบุคคลแบบเวลาจริงราคาประหยัด บน Raspberry Pi โดยประยุกต์อัลกอริทึม Tiny YOLO V3," *Engineering Transactions*, ฉบับที่ 22, หน้า 72 – 78, กรกฎาคม – ธันวาคม 2562.

[14] S. Lee, "Practical monitoring of undergrown pigs for IoT-Based large-scale smart farm," *IEEE Access*, vol. 7, no. 17, pp. 173,796 – 173,810, November 2019.

[15] R. Bayareh et al., "Development of a thermographic image instrument using the raspberry Pi embedded system for the study of the diabetic foot," in *2018 IEEE International Instrumentation and Measurement Technology Conference (I2MTC)*, Houston, TX, USA, May 14-17, 2018, pp. 1-6.

[16] J.-K. Tsai et al., "Deep learning-based real-time multiple-person action recognition system," *Sensors*, vol. 20, no. 17, pp. 4758, August 2020.

[17] J. McGee, S. J. Mathew and F. Gonzalez, "Unmanned aerial vehicle and artificial intelligence for thermal target detection in search and rescue applications," *in 2020 International Conference on Unmanned Aircraft Systems (ICUAS)*, Athens, Greece, September 1-4, 2020, pp. 883-891.

[18] นันทชัย พงศพัฒนานุรักษ, แสงสรรค ภูมิสถาน, ดวงดาว รักษากุล และ จิรัชญา ตอยติ่ง, *คู่มือแนวทางการดำเนินการตามมาตรฐานการปฏิบัติที่ดีสำหรับบานนกแอนกินรัง*, กรุงเทพฯ:คณะวนศาสตร มหาวิทยาลัยเกษตรศาสตร, 2561.

[19] GroupGets, "*Lepton® 3.5 by FLIR*," groupgets.com [Online]. Available: https:// groupgets.com/manufacturers/flir/products/lepton-3-5. [Accessed: February 10, 2021].

[20] Towards Data Science, "*The 80/20 Split Intuition and an Alternative Split Method*," towardsdatascience.com [Online]. Available: https://towardsdatascience.com/finally-why-we-use-an-80-20-split-for-training-and-test-data-plus-an-alternative-method-oh-yes-edc77e96295d. [Accessed: April 26, 2021].

[21] ImageAI, "*Official English Documentation for ImageAI!*," imageai.readthedocs.io [Online]. Available: https://imageai.readthedocs.io/en/latest/index.html#official-english-documentation-for-imageai. [Accessed: March 07, 2021].

[22] COCO Common Objects in Context, "*What is COCO*," cocodataset.org [Online]. Available: https://cocodataset.org/#home. [Accessed: April 28, 2021].

[23] Towards Data Science, "mAP (Mean Average Precision) Might Confuse You!," towardsdatascience.com [Online]. Available: https://towardsdatascience.com/map-mean-average-precision-might-confuse-you-5956f1bfa9e2. [Accessed: April 26, 2021].

[24] BizOne, "*3 โหมดการทำงานของ PowerBI Desktop*," bizone.co.th [Online]. Available: https://www.bizone.co.th/blogs/business-intelligence/powerbi-connection-modes. [Accessed: April 29, 2021].

ภาคผนวก

ภาคผนวก ก.
โค้ดโปรแกรม

TNI
สถาบันเทคโนโลยีไทย - ญี่ปุ่น
THAI - NICHI INSTITUTE OF TECHNOLOGY

โค้ดโปรแกรมภาษา Python ใน Google Colab

แบ่งคลิปวีดีโอเป็นเฟรมภาพ

```
# Code Convert VDO CLIP TO IMAGE
# Install libraries
pip install tensorflow==2.4.0
pip install keras==2.4.3 numpy==1.19.3 pillow==7.0.0 scipy==1.4.1 h5py==2.10.0 ma
tplotlib==3.3.2 opencv-python keras-resnet==0.2.0
pip install opencv-python
pip install imageai –upgrade

# Code Program
import cv2
import os

listing = os.listdir(r'/content/drive/MyDrive/Count_bird/Clip')
count = 1

for vid in listing:
    vid = r"/content/drive/MyDrive/Count_bird/Clip/"+vid
    vidcap = cv2.VideoCapture(vid)

    def getFrame(sec):
        vidcap.set(cv2.CAP_PROP_POS_MSEC,sec*500)
        hasFrames,image = vidcap.read()

        if hasFrames:
            cv2.imwrite("/content/drive/MyDrive/Count_bird/Frame2/"+str(count)+".jpg",
image) # Save frame as JPG file
        return hasFrames
    sec = 0
```

```
frameRate = 1 # Change this number to 1 for each 1 second

success = getFrame(sec)

while success:
    count = count + 1
    sec = sec + frameRate
    sec = round(sec, 2)
    success = getFrame(sec)
```

เทรนโมเดลด้วยอัลกอริทึม YOLO V3

```
# Code Training Model YOLO V3
# Install libraries
pip install tensorflow==2.4.0
pip install keras==2.4.3 numpy==1.19.3 pillow==7.0.0 scipy==1.4.1 h5py==2.10.0 ma
tplotlib==3.3.2 opencv-python keras-resnet==0.2.0
pip install opencv-python
pip install imageai –upgrade

#Code Program
from imageai.Detection.Custom import DetectionModelTrainer

trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="/content/drive/MyDrive/birds")
trainer.setTrainConfig(object_names_array=["birds"], batch_size=4, num_experiments
=200, train_from_pretrained_model="/content/drive/MyDrive/birds/pretrained-
yolov3.h5")
trainer.trainModel()
```

ประเมินความแม่นยำของโมเดล

```
#evaluate the mAP score
# Install libraries
pip install tensorflow==2.4.0
pip install keras==2.4.3 numpy==1.19.3 pillow==7.0.0 scipy==1.4.1 h5py==2.10.0 ma
tplotlib==3.3.2 opencv-python keras-resnet==0.2.0
pip install opencv-python
pip install imageai –upgrade

from imageai.Detection.Custom import DetectionModelTrainer

trainer = DetectionModelTrainer()
trainer.setModelTypeAsYOLOv3()
trainer.setDataDirectory(data_directory="/content/drive/MyDrive/birds")
metrics = trainer.evaluateModel(model_path="/content/drive/MyDrive/birds/models
", json_path="/content/drive/MyDrive/birds/json/detection_config.json", iou_threshol
d=0.5, object_threshold=0.3, nms_threshold=0.5)
print(metrics)
```

โค้ดโปรแกรมภาษา Python ใน Raspberry Pi

```
# import the opencv library
import cv2
import time
import os

# set path save images
#execution_path = os.getcwd()+"/ImgTake"
execution_path = "/home/pi/Desktop/Count/ImgTake"
```

```
# define a video capture object
vid = cv2.VideoCapture(0)
result = True
# set image name with current date time
img_name = time.strftime("%Y%m%d%H%M%S")

while(result):

    # Capture the video frame
    ret, frame = vid.read()

    # Write the resulting frame
    cv2.imwrite(os.path.join(execution_path , img_name +'.jpg'),frame)
    #cv2.imwrite(/home/pi/Desktop/Count/Images/img_name +'.jpg',frame)
    result = False

# After the loop release the cap object
vid.release()
# Destroy all the windows
cv2.destroyAllWindows()
print("Take Image Complete")

#--- Detection Image ------
from imageai.Detection.Custom import CustomObjectDetection

input_image= "/home/pi/Desktop/Count/ImgTake/"+img_name+".jpg"
output_image= "/home/pi/Desktop/Count/ImgOutput/"+img_name+".jpg"

detector = CustomObjectDetection()
detector.setModelTypeAsYOLOv3()
detector.setModelPath("/home/pi/Count/Models/detection_model-ex-017--loss-
```

```
0011.473.h5")
detector.setJsonPath("/home/pi/Count/json/detection_config.json")
detector.loadModel()
detections = detector.detectObjectsFromImage(input_image, output_image)

count = 0
for detection in detections:
    print(detection["name"], " : ", detection["percentage_probability"], " : ",
detection["box_points"])
    count = count+1

print(output_image)
print("Detection Image Complete",count)

#--- Send line ------
import requests
import urllib.parse
import sys

LINE_ACCESS_TOKEN="PgQrWJ0WXR655eTcgvEmGic9V4aUMmE5ls5TWAlr56Y"
url = "https://notify-api.line.me/api/notify"
#file = {'imageFile':open('/home/pi/Desktop/Count/ImgOutput/Img12.jpg','rb')}
file = {'imageFile':open(output_image,'rb')}
data = ({
        'message': count
    })
LINE_HEADERS = {"Authorization":"Bearer "+LINE_ACCESS_TOKEN}
session = requests.Session()
r=session.post(url, headers=LINE_HEADERS, files=file, data=data)
print(r.text)
print("Send line Complete")
```

```
#--- Insert Data to Google Sheet ------
import gspread
from datetime import datetime
import time
from oauth2client.service_account import ServiceAccountCredentials

# define the scope
scope = ["https://spreadsheets.google.com/feeds",'https://www.googleapis.com/auth/spreadsheets',"https://www.googleapis.com/auth/drive.file","https://www.googleapis.com/auth/drive"]

# add credentials to the account
creds = ServiceAccountCredentials.from_json_keyfile_name('/content/drive/MyDrive/Colab Notebooks/PythonAPI.json', scope)
# authorize the clientsheet
client = gspread.authorize(creds)
# get the instance of the Spreadsheet
sheet = client.open('Birds_Count').get_worksheet(0)
data = sheet.get_all_records()
data = sheet.row_count
os.environ['TZ'] = 'Asia/Bangkok'
time.tzset()
datestring = datetime.strftime(datetime.now(), '%Y-%m-%d')
t = time.localtime()
current_time = time.strftime("%H:%M:%S", t)
row = [datestring,current_time,count]
index = sheet.row_count
sheet.insert_row(row, index)
print("Save Data Complete")
```