

การวิเคราะห์พฤติกรรมของวัวขณะมีอาการสืบพันธุ์ตามฤดูกาลโดยการประมาณ
ท่าทางด้วยปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึก

THE CATTLE RUT BEHAVIOR DETECTION BASE ON AI DEEP NEURAL
NETWORK



วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัจฉริยะเชิงคำนวณ
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
พ.ศ.2565
KMITL- 2022-EN-M-070-001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

THE CATTLE RUT BEHAVIOR DETECTION BASE ON AI DEEP NEURAL
NETWORK



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
MASTER OF ENGINEERING IN ROBOTICS AND INTELLIGENCE SYSTEMS
SCHOOL OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
2022
KMUTL- 2022-EN-M-070-001

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



COPYRIGHT 2022

SCHOOL OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

| | |
|-----------------------------|---|
| หัวข้อวิทยานิพนธ์ | การวิเคราะห์พฤติกรรมของวัวขณะมีอากาศสี่พันซ์ตามฤดูกาล โดยการประมาณท่าทางด้วยปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึก |
| นักศึกษา | นายพัศรุจย์ แพ้ไชยภูมิ |
| รหัสประจำตัว | 63601036 |
| ปริญญา | วิศวกรรมศาสตรมหาบัณฑิต |
| สาขาวิชา | วิศวกรรมหุ่นยนต์และระบบอัจฉริยะเชิงคำนวณ |
| พ.ศ. | 2565 |
| อาจารย์ที่ปรึกษาวิทยานิพนธ์ | ดร. ภูมิ คงห้วยรอบ |

บทคัดย่อ

วิทยานิพนธ์ฉบับนี้ นำเสนอตัวโมเดลงานกล้องตรวจจับพฤติกรรมของวัวผ่านท่าทาง โดยงานวิทยานิพนธ์ได้ศึกษา ทดลอง โดยใช้ความรู้พื้นฐานของปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึก ตลอดไปจนถึงการปรับปรุงประสิทธิภาพความแม่นยำผ่านวิธีการถ่ายโอนความรู้(Transfer learning) งานวิทยานิพนธ์ฉบับนี้ได้ประยุกต์การใช้ปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึกเปรียบเทียบจำนวนทั้งหมด 4 สถาปัตยกรรม ดังนี้ 1) SSD Mobile-Net V2 2) SSD ResNet101 V1 FPN 3) YOLOv5s and 4) Mask R-CNN. ซึ่งเครือข่ายสถาปัตยกรรมที่กล่าวมาข้างต้น ได้ถูกผ่านการเรียนรู้ หรือ Train มาก่อนแล้วโดยมีการเรียนรู้ผ่านชุดข้อมูลมาตรฐาน(COCO) สำหรับการเฝ้าติดตามพฤติกรรมของวัวได้มีการเก็บภาพในสภาวะการเลี้ยงแบบปล่อยตามธรรมชาติ โดยชุดข้อมูลภาพจะถูกแบ่งแยกสำหรับทำเป็น ชุดข้อมูลสำหรับเทรน และชุดข้อมูลสำหรับทดสอบ สำหรับงานวิจัยนี้มีการแบ่งพฤติกรรมของวัวออกเป็น 2 แบบ คือ (1) มีการแสดงพฤติกรรมแบบปกติ(Normal behavior) และ (2) มีการแสดงการติดสัด(Cattle rut behavior) ผลการทดลองแสดงให้เห็นว่า หลังจากการฝึกสอนหรือเทรนชุดข้อมูล โมเดลสถาปัตยกรรมแบบ YOLOv5s ให้ค่าประสิทธิภาพอัตราเฉลี่ยความแม่นยำ ได้สูงที่สุดคือ 95.5% และสามารถใช้เวลาในการเทรนน้อยที่สุดเมื่อเปรียบเทียบกับ 4 โมเดลสถาปัตยกรรมดังที่กล่าวมา ดังนั้นได้มีการนำเสนอการใช้โมเดลสถาปัตยกรรมแบบ YOLOv5s มาประยุกต์ใช้ตรวจจับพฤติกรรมการติดสัดของวัวที่ให้ค่าความแม่นยำที่สุด เพื่อนำมาส่งเสริมเพิ่มประสิทธิภาพการเลี้ยงดู เพิ่มผลผลิต สำหรับการบริโภคที่มากขึ้นในอนาคต

| | |
|-----------------------|--|
| Thesis | The cattle rut behavior detection base on AI deep neural network |
| Student | Mr.Phatsarut Phaechaiyaphum |
| Student ID. | 63601036 |
| Degree | Master of Engineering |
| Program | Robotics and Computational Intelligence Systems |
| Year | 2022 |
| Thesis Advisor | Dr.Poom Konghuayrob |

ABSTRACT

This thesis proposes an application model of camera object detection to observe the cattle rut through its pose. This research is based on deep neural network method along with transfer learning technique. This thesis employs four different deep convolutional neural network architectures 1) SSD Mobile-Net V2 2) SSD ResNet101 V1 FPN 3) YOLOv5s and 4) Mask R-CNN. All the network mentioned above are pre-trained on COCO dataset. To observe the cattle rut behavior, the images of cattle behavior were taken in natural open environment, the data obtained are split into training and testing datasets. In this research cattle behaviors are categorized mainly in two classes, (1) normal behavior and (2) the cattle rut behavior. The result obtained after training the models shows that the YOLOv5s model obtained the highest mean average precision which is 95.5% and least training time. Thus, this thesis proposes that YOLOv5s model can be applied to detect the cattle behavior for precise artificial breeding in order to increase the cattle population in farm to serve higher consumption demand in the near future.

กิตติกรรมประกาศ

วิทยานิพนธ์เล่มนี้สำเร็จได้ด้วยความกรุณาจากอาจารย์ที่ปรึกษา ดร. ภูมิ คงห้วยรอบ ที่ให้ความช่วยเหลือ ให้คำชี้แนะช่วยแก้ปัญหาตลอดจนให้ความรู้และประสบการณ์ที่ดีแก่ข้าพเจ้า

ขอขอบคุณ บ. เจริญโภคภัณฑ์อาหาร จำกัด(มหาชน) (Charoen Pokphand Foods) ที่ให้การสนับสนุนด้านทุนการศึกษางานวิจัยนี้ ขอขอบคุณเพื่อนร่วมงาน หน่วยงานเทคโนโลยีหุ่นยนต์ CPF สำหรับคำปรึกษาด้านเทคนิค

สุดท้ายนี้ต้องขอขอบคุณบิดา มารดา ทั้งสองท่าน ที่เป็นแรงผลักดันในการศึกษาตลอดมา สำหรับคุณงามความดีอันใดที่เกิดจากวิทยานิพนธ์ฉบับนี้ ข้าพเจ้าขอมอบให้กับบิดามารดา ซึ่งเป็นที่รักและเคารพยิ่ง ตลอดจนครูอาจารย์ที่เคารพทุกท่านที่ได้ประสิทธิ์ประสาทวิชาความรู้และถ่านทอดประสบการณ์ที่ดีให้แก่ข้าพเจ้า

พัศรุจย์ แพ้ไชภูมิ



สารบัญ

| | หน้า |
|--|------|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ..... | IV |
| สารบัญตาราง..... | VI |
| สารบัญรูป..... | VII |
| | |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ความเป็นมาของปัญหาและความสำคัญของปัญหา..... | 1 |
| 1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา..... | 6 |
| 1.3 สมมุติฐานของการศึกษา..... | 6 |
| 1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย..... | 6 |
| 1.5 ขอบเขตของงานวิจัย..... | 6 |
| 1.6 ขั้นตอนการศึกษา..... | 6 |
| 1.7 เครื่องมือและอุปกรณ์ที่ใช้ในวิจัย..... | 7 |
| 1.8 โครงสร้างของวิทยานิพนธ์..... | 7 |
| | |
| บทที่ 2 ทฤษฎีหลักการและงานวิจัยที่เกี่ยวข้อง..... | 8 |
| 2.1 วัยเจริญพันธุ์ของวัว..... | 8 |
| 2.2 พฤติกรรมอาการติดสัด..... | 9 |
| 2.3 ผสมพันธุ์เทียม..... | 9 |
| 2.4 โครงข่ายประสาทเทียม..... | 10 |
| 2.5 Convolutional Neural Network..... | 11 |
| 2.5.1 การทำงานของ Convolutional Neural Network | 11 |
| 2.5.2 Convolution..... | 11 |
| 2.5.3 Pooling layer..... | 12 |
| 2.5.4 Fully connected..... | 13 |
| 2.5.5 Activation function..... | 13 |
| 2.5.6 ความผิดพลาดในการเรียนรู้..... | 15 |
| 2.6 งานวิจัยที่เกี่ยวข้อง..... | 16 |

สารบัญ (ต่อ)

| | หน้า |
|--|------|
| บทที่ 3 การจำแนกพฤติกรรมการติดสัดของวัวด้วยปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึก..... | 18 |
| 3.1 โมเดลจำแนกพฤติกรรมการติดสัดของวัวโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน..... | 18 |
| 3.1.1 การออกแบบโมเดลจำแนกพฤติกรรมโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน..... | 18 |
| 3.1.2 Transfer learning..... | 19 |
| 3.2 ฐานข้อมูลที่ใช้ในการทดลอง..... | 20 |
| 3.2.1 COCO ชุดฐานข้อมูล..... | 21 |
| 3.3 Data preprocessing..... | 22 |
| 3.3.1 Roboflow labelling tool..... | 22 |
| 3.3.2 Data labelling..... | 23 |
| 3.3.3 Data augmentation..... | 25 |
| 3.4 Pre-trained model..... | 25 |
| 3.4.1 SSD MobileNet V1 FPN..... | 26 |
| 3.4.2 SSD ResNet101 V1 FPN..... | 26 |
| 3.4.3 YOLOv5s..... | 26 |
| 3.4.4 Faster-RCNN X101 FPN..... | 26 |
| 3.5 การประเมินโมเดล..... | 27 |
| 3.5.1 การวัดประสิทธิภาพความถูกต้อง..... | 27 |
| 3.6 ขั้นตอนการทดลอง..... | 28 |
| บทที่ 4 การทดลองและวิเคราะห์ผลการทดลอง..... | 29 |
| 4.1 การจัดการฐานข้อมูลในการทดลอง..... | 29 |
| 4.2 การฝึกสอนภาพฐานข้อมูลโดยโมเดลผ่านการเรียนรู้..... | 30 |
| 4.2.1 SSD MobileNet V1 FPN..... | 30 |
| 4.2.2 SSD ResNet101 V1 FPN..... | 31 |
| 4.2.3 YOLOv5s..... | 33 |
| 4.2.4 Faster-RCNN X101 FPB..... | 34 |
| 4.3 การเปรียบเทียบและวิเคราะห์ผลการทดลอง..... | 36 |
| 4.4 การประยุกต์โมเดลร่วมกับการประเมินผลภาพ..... | 37 |

สารบัญ (ต่อ)

| | หน้า |
|---|------|
| บทที่ 5 สรุปงานวิจัยและข้อเสนอแนะ..... | 39 |
| 5.1 สรุปเนื้อหาวิทยานิพนธ์..... | 39 |
| 5.2 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ..... | 40 |
| | |
| เอกสารอ้างอิง..... | 42 |
| ภาคผนวก ก..... | 45 |
| ภาคผนวก ข..... | 48 |
| ประวัติผู้เขียน..... | 67 |



สารบัญตาราง

| ตารางที่ | หน้า |
|--|------|
| 2.1 อัตราการแสดงผลกิจกรรมการติดสั้ด..... | 10 |
| 3.1 จำนวนข้อมูลสำหรับการทดลอง..... | 21 |
| 3.2 รูปแบบการสอนข้อมูลตามแต่ละโมเดล..... | 23 |
| 3.3 ประสิทธิภาพความแม่นยำของแต่ละ Pre-trained model..... | 26 |
| 4.1 การจัดการฐานข้อมูล..... | 29 |
| 4.2 เปรียบเทียบประสิทธิภาพความแม่นยำในการประเมินโมเดล..... | 36 |
| 4.3 เปรียบเทียบประสิทธิภาพความเร็วในการฝึกข้อมูล..... | 36 |



สารบัญรูป

| รูปที่ | หน้า |
|---|------|
| 1.1 เทรนการบริโภคนผลิตภัณฑ์จากสัตว์..... | 1 |
| 1.2 การทำประมาณท่าทาง กับวีว..... | 3 |
| 1.3 โมเดล Convolutional Pose machines..... | 3 |
| 1.4 โมเดล Stacked hourglass model..... | 4 |
| 1.5 โมเดล Convolutional heatmap regression..... | 4 |
| 1.6 ภาพ heatmap..... | 5 |
| 2.1 การทำงานของฮอร์โมน Gonadotropin..... | 8 |
| 2.2 พฤติกรรมการติดสัดของวัว..... | 9 |
| 2.3 จำลองโครงข่ายประสาทเทียม..... | 10 |
| 2.4 ตัวอย่างโครงสร้าง Convolutional Neural Network..... | 11 |
| 2.5 Max pooling layer ที่ ขนาด 2 x 2..... | 12 |
| 2.6 Average pooling layer ที่ ขนาด 2 x 2..... | 12 |
| 2.7 ฟังก์ชัน Sigmoid | 13 |
| 2.8 ฟังก์ชัน Tanh..... | 14 |
| 2.9 ฟังก์ชัน ReLU | 14 |
| 2.10 การตรวจจับและวิเคราะห์ของระบบ CowXNet..... | 16 |
| 2.11 การประสิทธิภาพของ YOLOv4 เทียบกับโมเดลอื่น..... | 17 |
| 3.1 การออกแบบโครงสร้าง CNN | 18 |
| 3.2 Transfer learning..... | 19 |
| 3.3 พฤติกรรมปกติทั่วไปของวัว..... | 20 |
| 3.4 พฤติกรรมการติดสัดของวัว..... | 21 |
| 3.5 ชุดข้อมูลมาตรฐาน COCO..... | 22 |
| 3.6 Roboflow labelling tool platform..... | 23 |
| 3.7 Data labelling แบบ Rectangular..... | 24 |
| 3.8 Data labelling แบบ Polygon..... | 24 |
| 3.9 ข้อมูล Over-fitting..... | 25 |
| 3.10 แสดงข้อมูล fitting..... | 25 |
| 3.11 วิธีการคำนวณ Intersection Over Union (IoU)..... | 27 |
| 3.12 ผลเทียบการตรวจจับกับข้อมูลที่สอน..... | 27 |
| 4.1 การสูญเสียของ Classification loss โมเดล SSD MobileNet V1 FPN..... | 30 |
| 4.2 การสูญเสียของ Localization loss โมเดล SSD MobileNet V2 FPN..... | 31 |
| 4.3 การสูญเสียของ Classification loss โมเดล SSD ResNet101 V1 FPN..... | 32 |

สารบัญรูป

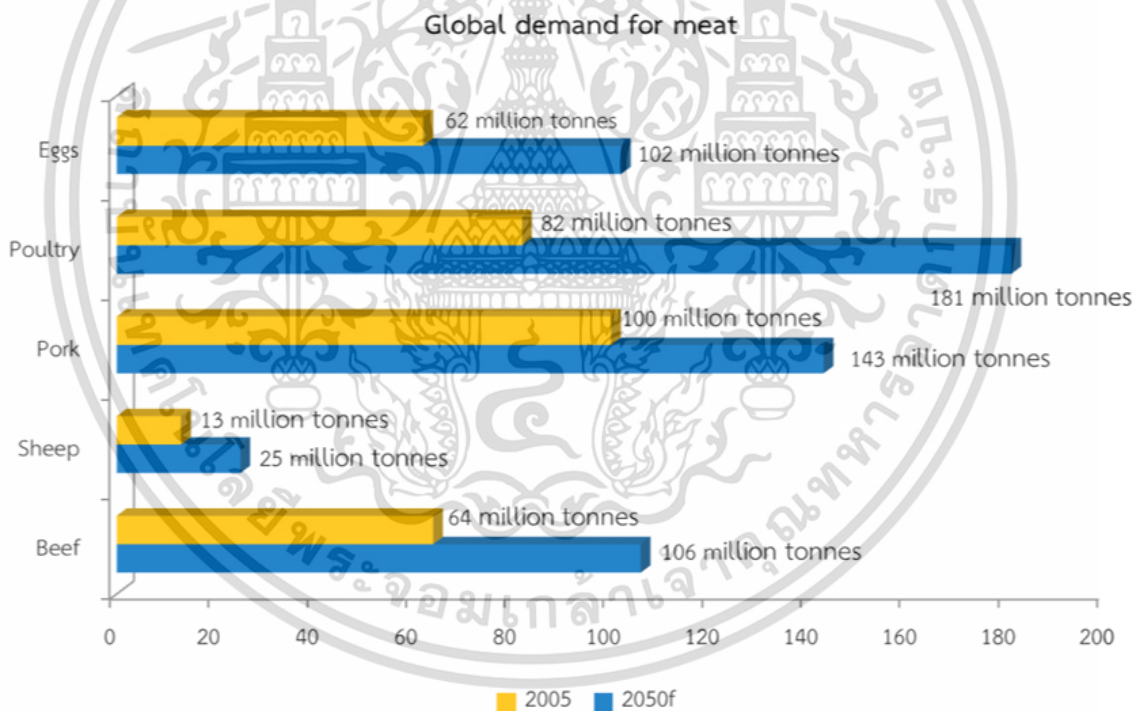
| รูปที่ | หน้า |
|---|------|
| 4.4 การลู่เข้าของ Localization loss โมเดล SSD ResNet101 V1 FPN..... | 33 |
| 4.5 การลู่เข้าของ Classification loss โมเดล YOLOv5s..... | 34 |
| 4.6 การลู่เข้าของ Localization loss โมเดล YOLOv5s..... | 34 |
| 4.7 การลู่เข้าของ Classification loss โมเดล Faster-RCNN X101 FPB..... | 35 |
| 4.8 การลู่เข้าของ Localization loss โมเดล Faster-RCNN X101 FPB..... | 35 |
| 4.9 การประยุกต์ร่วมกับการประมวลผลภาพ..... | 38 |
| 5.1 แนวทางพัฒนาต่อไปในอนาคต..... | 40 |
| ข.1 โครงสร้างโพลเตอร์Tensorflow..... | 52 |
| ข.2 โครงสร้างโพลเตอร์ย่อย Training demo..... | 53 |
| ข.3 pre-trained model ต่างๆใน model zoo..... | 56 |



บทที่ 1 บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ตามกรอบยุทธศาสตร์กรมปศุสัตว์ พุทธศักราช 2561 จนถึง 2565 การปศุสัตว์ของโลกได้มีการเปลี่ยนแปลงผ่านการพัฒนาและการเปลี่ยนแปลงไปจากอดีต เพราะต้องยกระดับให้สอดคล้องกับความต้องการของจำนวนประชากรโลกที่สูงขึ้น ทั้งนี้ จากการพยากรณ์ขององค์การอาหารและการเกษตรแห่งสหประชาชาติ (Food and Agriculture Organization of the United Nations หรือ FAO) ยืนยันว่า ปริมาณความต้องการเนื้อสัตว์ของประชากรโลก ในปี ค.ศ. 2050 โลกต้องการปริมาณผลิตภัณฑ์จากสัตว์ ไม่ว่าจะเป็น เนื้อสัตว์ นม และ ไข่ รวมหลายร้อยล้านตันต่อปี



รูปที่ 1.1 แนวโน้มการบริโภคผลิตภัณฑ์จากสัตว์

ฉะนั้นระบบปศุสัตว์ต้องได้รับการยกระดับให้สอดคล้องกับความต้องการที่สูงขึ้นของประชากรโลก การขยายตัวของเมืองที่อยู่อาศัย และการเจริญเติบโตทางเศรษฐกิจที่นับวันจะเพิ่มมากขึ้น การสร้างความมั่นคงทางอาหารจึงนับเป็นเรื่องสำคัญที่ทุกประเทศบนโลกต้องพบเจอ สำหรับประเทศไทย แม้ว่าภาพรวมผลิตภัณฑ์มวลรวมในประเทศ ไตรมาสที่3/2564 ลดลงร้อยละ 0.3 แต่ภาพรวมทั้งปียังขยายตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ร้อยละ 7.6 ผลกระทบจากการแพร่ระบาดอย่างต่อเนื่อง และกระจายเป็นวงกว้างของโรค Covid-19 ซึ่งกระทบต่อทั้งกิจกรรมทางเศรษฐกิจ และการดำเนินชีวิตของประชาชน เป็นตัวเร่งทำให้เกิดการปรับตัวใช้เทคโนโลยีมากขึ้น โดยเฉพาะในด้านการผลิต ตามกรอบนโยบายแผนพัฒนาการเกษตรในช่วงแผนพัฒนาเศรษฐกิจและสังคมแห่งชาติ ฉบับที่ 12 พ.ศ. 2560 - 2565 ที่กล่าวถึงยุทธศาสตร์กรมปศุสัตว์ พ.ศ. 2561 -2565 ข้อที่ 6 คือการยกระดับการบริหารจัดการเทคโนโลยีสารสนเทศและดิจิทัลด้านการปศุสัตว์ ให้มีความเหมาะสมและร่วมสมัย กล่าวถึงการพัฒนาองค์ความรู้ เทคโนโลยี นวัตกรรม และงานวิจัยต่างๆ เพื่อเป็นประโยชน์ต่อการปศุสัตว์ในทุกมิติ ทั้งทางด้านวิชาการ ด้านฐานข้อมูล ด้านการผลิต และด้านการตลาด เป็นที่ยอมรับในระดับสากล

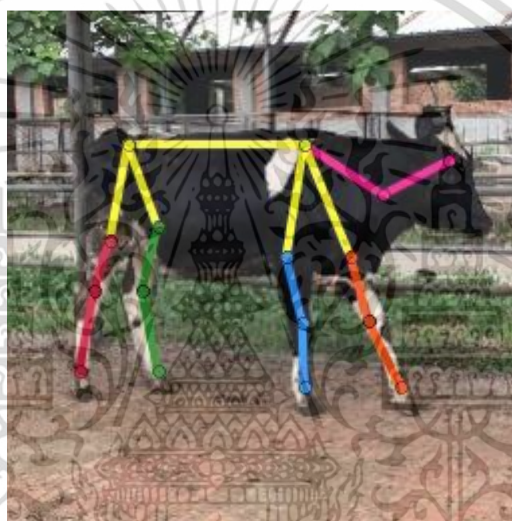
ปัจจุบันในการเลี้ยงโค ลูกโคนั้นถือเป็นผลผลิตที่สร้างรายได้และทำกำไรให้แก่เกษตรกร แม่โคที่มีประสิทธิภาพคือแม่โคที่สามารถให้กำเนิดลูกพันธ์ได้ทุกปี การจัดการด้านการผสมพันธ์ที่ถูกต้องจะช่วยเพื่อประสิทธิภาพ เป็นที่ต้องการของตลาดต่อไป โคสาวจะเริ่มแสดงอาการพร้อมที่จะผสมพันธ์หรือเป็นสัดเมื่ออายุได้ประมาณ 12-18 เดือน ปกติโคมีวงรอบแสดงอาการเป็นสัดเฉลี่ยอยู่ที่ประมาณ 21 วัน อาการแสดงที่ให้เห็นว่า โคเริ่มมีอาการการติดสัดคือมีอาการกระวนกระวาย ส่งเสียงร้องดัง ไล่ขี้ตัวอื่น กินอาหารน้อยลง อวัยวะเพศบวมแดงและมีเมือกใสไหลออกมา ทำยที่สุดโคตัวเมียจะยอมให้โคตัวผู้ขึ้นขึ้นนั่นเอง ซึ่งเหล่านี้เป็นระยะสำคัญที่ต้องสังเกตการ เนื่องจากเป็นระยะตกไข่ ดังนั้นจากสถิติในการผสมพันธ์เทียมโคนั้น ควรรีบทำการผสมหลังโคยืนนิ่ง 12 ถึง 18 ชั่วโมง หลังจากแสดงพฤติกรรมอาการติดสัดออกมา

การนำเทคโนโลยีปัญญาประดิษฐ์ หรือ Artificial Intelligent เข้ามาในงานด้านปศุสัตว์ จะช่วยทำให้เกิดโอกาสมากมายในการสร้างมูลค่าเพิ่ม ลดต้นทุน ลดความสูญเสีย ที่จะเกิดขึ้นในสัตว์ เช่น การประมาณรูปร่างท่าทาง(Pose estimate) ผ่านการเรียนรู้โครงข่ายประสาทเทียมเชิงลึกหรือ Dep Neural Network(DNN) จะช่วยให้เข้าใจลักษณะกลศาสตร์การเคลื่อนไหวตามร่างกาย เข้าใจพฤติกรรม รวมถึงเก็บเป็นฐานข้อมูลในการเลี้ยงต่อไป

Chen และคณะ (2558 : 179-185) ได้พูดถึงการประมาณท่าทาง หรือ ประมาณท่าทาง จากการกำหนดจุดบนตัวโคจากภาพ ผลการทดลองจากการทำประมาณท่าทางนั้นช่วยให้เราเข้าใจถึงพฤติกรรมของโค และมีโอกาสในการวิเคราะห์สุขภาพ Norring และคณะ (2556 : 1-5) อย่างไรก็ตามยังช่วยให้ทำให้แน่ใจในความผิดปกติที่สามารถพบเจอได้ตลอดเวลา วิธีดั้งเดิมของการทำประมาณท่าทางคือการทำโมเดลโครงสร้างแบบกราฟคือการทำจำลองโครงสร้างสิ่งมีชีวิต Felzenszwalb และคณะ (2544 : 55-79) และโมเดลแบบloopy structures Ren และคณะ (2544 : 824-831) ความสัมพันธ์เชิงพื้นที่ระหว่างส่วนต่างๆของร่างกายจะถูกแสดงในรูปแบบโครงสร้างโมเดลกลศาสตร์ภาพทางต้นไม้ โดยได้มีตัวอย่างของ Andriluka และคณะ (2554 : 3686-3693) ที่ได้ถูกพัฒนาวิธีการใช้โมเดลPictorial structures ซึ่งจำลองผลที่ได้จากการตรวจจับและประมาณท่าทางโดยรวมความสัมพันธ์เชิงพื้นที่

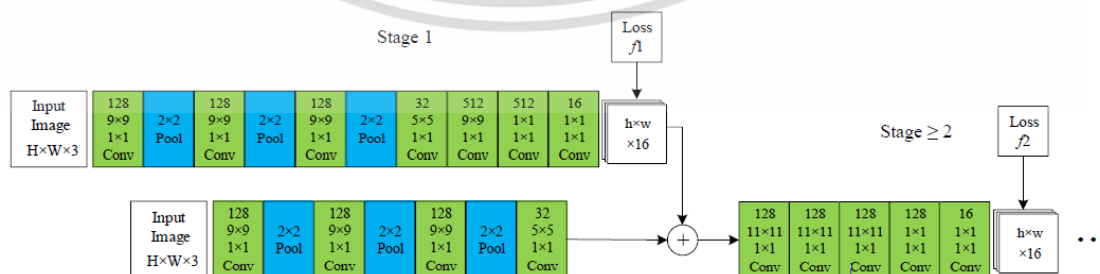
Toshev และคณะ (2556 : 1653-1660) ด้วยปัจจุบันการพัฒนาอย่างรวดเร็วของการทำ Deep Learning ที่การนำเสนออย่างมากมายของการทำเครือข่ายคอนโวลูชันเพื่อการให้เรียนรู้ซึ่งได้ผลการทำประมาณท่าทางได้ดีกว่า ซึ่งเปรียบเทียบแบบเก่าแล้วได้ผลลัพธ์ค่าความแม่นยำที่ดีกว่า ตัวอย่างเช่น DeepPose

Tompson และคณะ (2556 : 1799-1807) กล่าวถึงการใช้หลายคอนโวลูชันหลาย network เข้าด้วยกันเพื่อรวม Markov Random Field สำหรับทำ ประมาณท่าทาง Cai และคณะ (2559) ได้กล่าวถึงการประมาณท่าทางของวัวผ่านการคอนโวลูชันซึ่งในงานวิจัยกล่าวถึงการทำ ประมาณท่าทางกับวัว ดังแสดงในรูปที่ 1.2 ต่อไปนี้



รูปที่ 1.2 การทำประมาณท่าทางของวัว

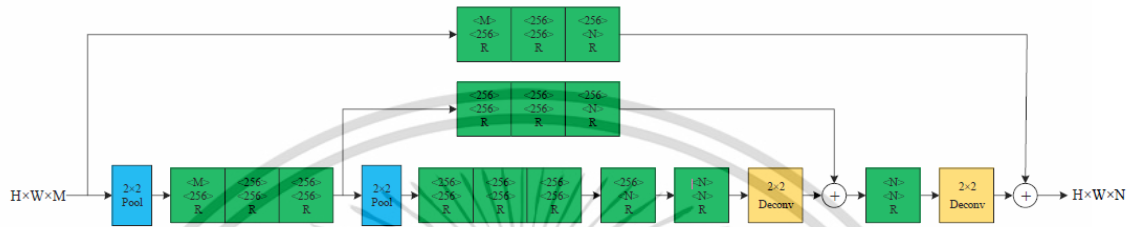
Wei และคณะ (2556 : 4724-4732) ได้นำเสนอโมเดลคอนโวลูชัน pose machine model หรือ CPMs ซึ่งเป็นสถาปัตยกรรมที่มีหน้าที่ทำประมาณท่าทาง เป็นโมเดลที่ประกอบไปด้วยหลายคอนโวลูชันเน็ตเวิร์ค ดังแสดงในรูปที่ 1.3



รูปที่ 1.3 โมเดล Convolutional Pose machines

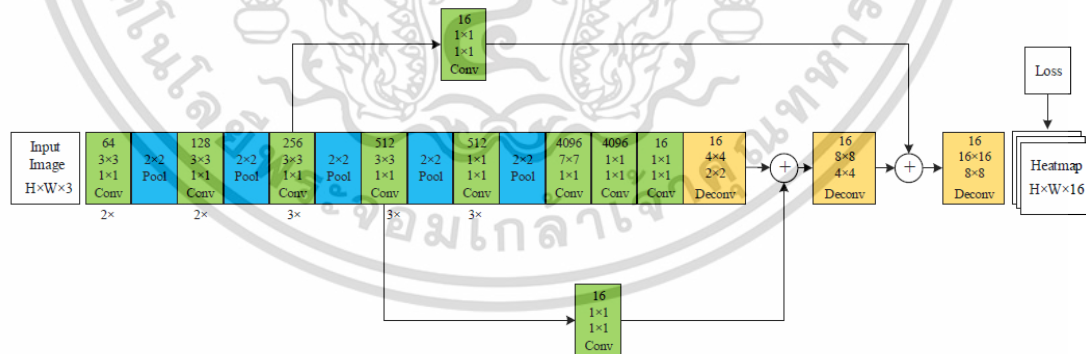
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Newell และคณะ (2556 : 483-499) ได้นำเสนอ Stacked hourglass model สำหรับงานตรวจจับท่าทาง ประมาณท่าทาง ของคนที่มีความสามารถประมาณผลร่างกายแต่ละส่วนได้เป็นอย่างดี ซึ่งแสดงในรูปที่ 1.4 ดังนี้



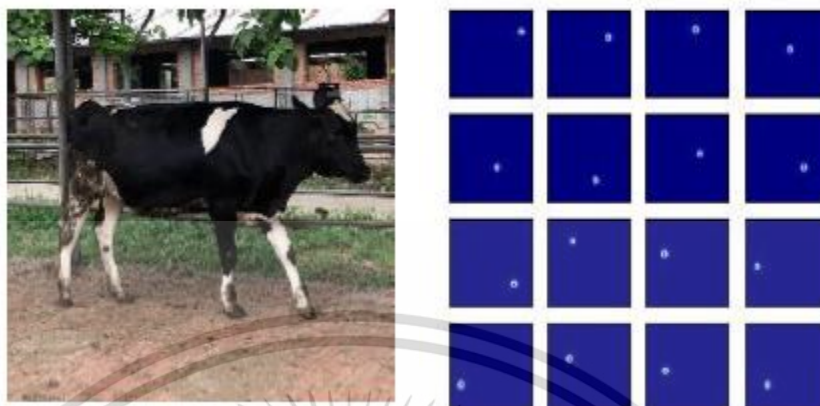
รูปที่ 1.4 โมเดล Stacked hourglass model

โมเดล Stacked hourglass ได้ถูกพัฒนาเพื่อให้สามารถทำการวิเคราะห์ ประมาณท่าทาง ของวัว Bulat และคณะ (2556) ได้นำเสนอโมเดลการทำ ประมาณท่าทาง กับคนโดยผ่านการทำ Heatmap ซึ่งในงานวิจัยของ Cai และคณะ (2559) ได้นำเทคนิคการทำ Heatmap บนส่วนต่างๆของวัวเพื่อนำมาเข้าโมเดล Convolutional heatmap regression ที่มีโครงสร้างสถาปัตยกรรมดังรูปที่ 1.5 และภาพการทำ heatmap แสดงในรูปที่ 1.6 ดังนี้



รูปที่ 1.5 โมเดล Convolutional heatmap regression

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.6 ภาพ heatmap

โดยเทคโนโลยีปัญญาประดิษฐ์ ช่วยให้เราได้เข้าใจว่า คอมพิวเตอร์สามารถเรียนรู้ เข้าใจเหตุผล แก้ไขปัญหาต่างๆ โดยที่ผู้ใช้มีหน้าที่สอนให้ระบบปัญญาประดิษฐ์เรียนรู้เอง หรือเรียกว่า Machine Learning นั่นเอง ในปัจจุบันหนึ่งในที่นิยมในงาน Machine Learning คือการนำมาให้เรียนรู้จดจำภาพ หรือเรียกว่า Convolutional Neural Network(CNN) โดยเป็นการนำฐานข้อมูลซึ่งเป็นภาพของสิ่งที่ผู้ใช้จะนำมาให้ คอมพิวเตอร์เรียนรู้ รวมถึงการทำแบ่งกลุ่มข้อมูลแบบเคมีน หรือ k-means clustering การทำนายข้อมูลแบบลดถอยเชิงเส้น หรือ Linear regression การเรียนรู้ข้อมูลแบบมีลำดับ หรือ Recurrent Neural Network

สำหรับ ชุดฐานข้อมูลจะรวมไปถึงท่าทางที่แตกต่างกันไปของทั้ง โคนม และโคนเนื้อโดยภาพโครงสร้างคร่าวๆ เริ่มจากนำภาพมา label การเทรนโมเดลคอนโวลูชันเพื่อเปรียบเทียบ RMSE(Root Mean Square Error) ของแต่ละท่าทาง โดยสามารถจำแนกสรุปได้ออกมาเป็นดังนี้

- (1) ชุดฐานข้อมูล ประกอบด้วย 50 ภาพ ของทั้งวัวตัวผู้ตัวเมีย ในกริยาต่างๆ และมีการกำหนด annotations ภาพต่างๆด้วยมือทั้งภาพใช้สำหรับเทรนฝึกและใช้ทดสอบ
- (2) โมเดลสถาปัตยกรรมสำหรับการทำ Convolutional Neural Network เพื่อการวิเคราะห์พฤติกรรมการติดสัด โดยใช้โมเดลถ่ายโอนการเรียนรู้ MobileNet-50, ResNet-101, Faster RCNN 50, YOLOv5
- (3) การจำแนกพฤติกรรมของโคโดยใช้ Weight ของโมเดลที่เรียนรู้ได้มาทำงานร่วมกับการประมวลผลภาพ

ในหัวข้อถัดไปจะกล่าวถึงข้อมูล ชุดฐานข้อมูล ที่ใช้สำหรับการเทรน และทดสอบหลักการของโมเดล การทดลอง ผลการทดลอง และการวิเคราะห์ผลการทดลอง สุดท้ายเป็นส่วนสรุปผลการทดลองและสิ่งที่จะทำต่อไปในอนาคต

1.2 ความมุ่งหมายและวัตถุประสงค์ของการศึกษา

1. เพื่อศึกษาการทำงานของระบบการเรียนรู้เชิงลึกผ่านโครงข่ายประสาทเทียม Deep Neural Network(DNN)
2. เพื่อศึกษาการจำลองท่าทางหรือ Pose estimate สัตว์ในฟาร์มในการทำปัญหาประดิษฐ์
3. เพื่อออกแบบระบบจำแนกพฤติกรรมการติดสัดในฟาร์มโดยใช้ Convolutional Neural Network ร่วมกับ TensorFlow Object Detection API
4. การติดตามพฤติกรรมสัตว์มีอาการติดสัดในฟาร์ม โดยทำงานร่วมกับการประมวลผลภาพ Image processing

1.3 สมมุติฐานของการศึกษา

การใช้ Weight ที่มีค่าความแม่นยำสูงจากการ เทรนชุดข้อมูล เพื่อนำมาตรวจสอบรูปที่มีความใกล้เคียงกัน

1.4 ทฤษฎีหรือแนวความคิดที่ใช้ในการวิจัย

ทฤษฎีสถาปัตยกรรม SSD หรือ Single Shot Detector ที่กล่าวถึง การดึงคุณลักษณะและการกรองของเมทริกซ์ภาพ เพื่อนำไปสู่การถ่ายโอนการเรียนรู้ผ่านโครงข่ายประสาทเทียม เพื่อให้ได้ค่าความแม่นยำที่สูงที่สุด

1.5 ขอบเขตการวิจัย

1. อุปกรณ์กล้อง สามารถโฟกัส บันทึกเป็นไฟล์วิดีโอ (.MP4)
2. ฟาร์มโคเปิด หรือ คอกวัวที่เลี้ยง
3. บันทึกพฤติกรรมการสัดของวัวจากการวัวตัวผู้ ขึ้นคร่อมวัวตัวเมีย
4. ทำการจำแนกภาพและตรวจจับวัตถุในภาพ ผ่าน TensorFlow Object Detection API
5. เทรน Weight จาก Pre-Trained Model จาก MobileNet-50, ResNet-101, Faster RCNN 50, YOLOv5
6. นำ Weight ที่ได้จากการเทรนมาแล้ว นำมาแสดงผลร่วมกับ OpenCV

1.6 ขั้นตอนของการศึกษา

- 1.6.1 ศึกษาทฤษฎีและความรู้พื้นฐานของอัลกอริทึมการเรียนรู้เชิงลึก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 1.6.2 ศึกษาทฤษฎี สถาปัตยกรรมโครงสร้างของอัลกอริทึมตรวจจับวัตถุแบบต่างๆ
- 1.6.3 ศึกษาทฤษฎีของพิสูจน์ประสิทธิภาพของตัวโมเดล
- 1.6.4 ศึกษาพฤติกรรมของการติดสัดในวีว
- 1.6.5 เก็บชุดฐานข้อมูล พฤติกรรมการติดสัดของวีว
- 1.6.6 ทดลองการถ่ายโอนความรู้ผ่านโมเดลอัลกอริทึมต่างๆ และพิสูจน์ประสิทธิภาพของโมเดล
- 1.6.7 สรุปผลการทดลองและวิเคราะห์ผลลัพธ์ที่ได้
- 1.6.8 จัดทำเอกสารประกอบวิทยานิพนธ์

1.7 เครื่องมือและอุปกรณ์ที่ใช้ในวิจัย

- 1.7.1 เครื่องคอมพิวเตอร์ส่วนบุคคล ยี่ห้อ/รุ่น Dell Precision 7750
 - หน่วยประมวลผลกลางยี่ห้อ Intel รุ่น Core i7
 - หน่วยความจำชั่วคราว RAM 16 GB
- 1.7.2 บริการ Cloud computing ของ Google Colab
- 1.7.3 บริการ Web labelling ของ Roboflow
- 1.7.4 โทรศัพท์สำหรับเก็บบันทึกชุดข้อมูล

1.8 โครงสร้างของวิทยานิพนธ์

วิทยานิพนธ์ฉบับนี้แบ่งเนื้อหาออกเป็น 5 ส่วนดังต่อไปนี้

บทที่ 1 กล่าวถึงที่มาของงานวิจัย ปัญหา ความมุ่งหมาย วัตถุประสงค์ สมมุติฐานทฤษฎีและแนวคิดที่ใช้รวมถึงขอบเขตการวิจัย และขั้นตอนการศึกษาวิจัย

บทที่ 2 กล่าวถึงงานวิจัยก่อนหน้าที่เกี่ยวข้องกับ การตรวจจับวัตถุ Object detection ทฤษฎีสถาปัตยกรรมของโมเดลการตรวจจับวัตถุแบบต่างๆ ซึ่งได้แก่ MobileNet-50, RestNet-101, Faster RCNN 50, YOLOv5

บทที่ 3 กล่าวถึงโครงข่ายประสาทเทียมแบบคอนโวลูชันกับการตรวจจับวัตถุ แบบการถ่ายโอนความรู้ ชุดข้อมูลที่ใช้สำหรับในงานวิจัยนี้

บทที่ 4 การออกแบบการทดลอง การวิเคราะห์ผลการทดลอง ซึ่งประกอบด้วยโครงข่ายประสาทเทียมแบบคอนโวลูชันผ่านโมเดลต่างๆ ผ่านการโอนถ่ายการเรียนรู้ การวิเคราะห์ผลการทดลองในแต่ละโมเดล การพิสูจน์ค่าความแม่นยำของโมเดล รวมถึงการประยุกต์โมเดลทำงานร่วมกับ ระบบประมวลผลภาพ Image processing เพื่อแสดงผลการทำงานของโมเดลออกมา

บทที่ 5 บทสรุปของวิทยานิพนธ์และข้อเสนอแนะ รวมทั้งแนวทางการนำไปพัฒนาต่อ

บทที่ 2

ทฤษฎีหลักการและงานวิจัยที่เกี่ยวข้อง

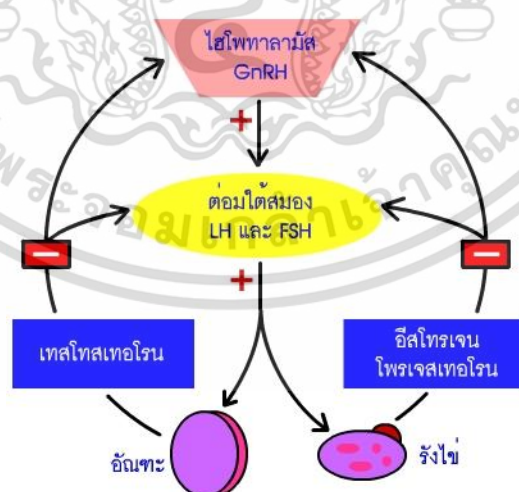
ในบทที่ 2 จะกล่าวถึงทฤษฎีและงานวิจัยที่เกี่ยวข้องกับการพัฒนาระบบจำแนกพฤติกรรมการติดสัดของวัว หลักการโครงข่ายประสาทเทียม การเรียนรู้เชิงลึก Convolutional Neural การตรวจจับวัตถุ การจำแนกภาพ การระบุตำแหน่งในภาพ การถ่ายโอนการเรียนรู้ รวมถึงการประเมินโมเดลจากการถ่ายโอนการเรียนรู้ ซึ่งจะอธิบายการพัฒนาตัวโมเดลจำแนกพฤติกรรมการติดสัดของวัว ในบทที่ 3 ต่อไป

2.1 วัยเจริญพันธุ์ของวัว

ในช่วงวงจรชีวิตของสัตว์นั้นเมื่อแบ่งตามการสืบพันธุ์แบ่งได้เป็น 3 ระยะ คือ

1. ช่วงเวลาเป็นทารกและแรกคลอด
2. ช่วงเข้าสู่เป็นวัยเจริญพันธุ์
3. ช่วงสมบูรณ์พร้อมที่จะให้การสืบพันธุ์

โดยในช่วงทารกในครรภ์นั้น จะสร้างสร้างฮอร์โมน Gonadotropin คือ FSH และ LH หลังจากคลอดระดับฮอร์โมนจะลดลงและคงอยู่ในสภาพคงที่ในช่วงระดับหนึ่ง ซึ่งฮอร์โมน Gonadotropin เป็นฮอร์โมนที่มีผลต่อการเจริญเติบโตของอวัยวะสืบพันธุ์ทั้งในด้านรูปร่างการทำงานที่ และการเจริญเข้าสู่ลักษณะของวัยรุ่น วัยหนุ่มสาว ทั้งทางด้านจิตใจและพฤติกรรม เมื่อเข้าสู่ช่วงเจริญพันธุ์ (Puberty) ต่อมาได้สมองส่วนหน้าจะสร้างสร้างฮอร์โมน Gonadotropin อีกครั้งมีผลให้ในเพศหญิงอวัยวะที่เรียกว่ารังไข่ และในเพศชายอวัยวะที่เรียกว่าอัณฑะทำหน้าที่เป็นต่อมไร้ท่อผลิตฮอร์โมนที่พื้นฐานสำหรับเหนี่ยวนำการเป็นสัดและตกไข่แบบระยะสั้นต่อไป (นลินทิพย์ พรหมขาว และคณะ, 2021 : 550)



รูปที่ 2.1 การทำงานของฮอร์โมน Gonadotropin

2.2 พฤติกรรมอาการติดสัด

อาการเป็นสัด แสดงให้เห็นถึงอากัปภิกิริยาสันชาตियานพร้อมสืบพันธุ์ของสัตว์ โดยอาการเป็นสัดที่ชัดเจนที่สุดของวัวคือ พฤติกรรมที่วัวตัวเมียยอมให้วัวตัวผู้ขึ้นขี่ได้ โดยมักจะมีพฤติกรรมเช่นนี้ประมาณ 6-12 ชั่วโมง อาการอย่างอื่นเช่น เมื่อกไหลออกจากอวัยวะเพศ หรือมีอาการร้องเสียงที่ผิดปกติ โดยจะมีอาการในช่วงเพศเมียมีอาการไซตค เฉลี่ยทุก21วัน



รูปที่ 2.2 พฤติกรรมการติดสัดของวัว

2.3 ผสมพันธุ์เทียม

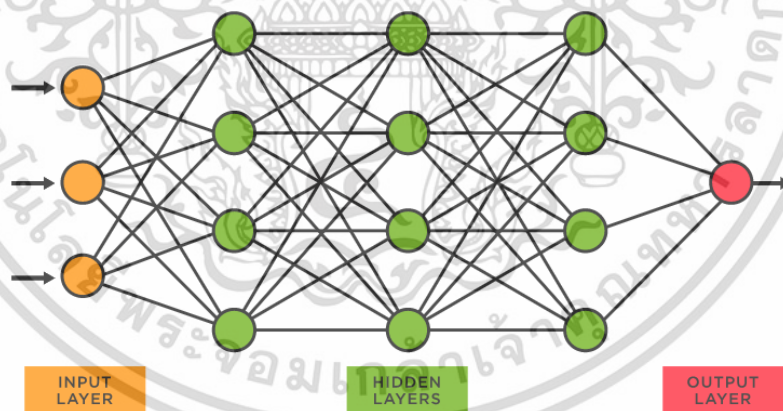
ในปัจจุบันในโลกมีเทคโนโลยีในการปรับปรุงพันธุ์สัตว์ไปถึงขั้นโคลนนิ่ง หรือเพาะเนื้อเยื่อในแลป ทดลองเพื่อให้ได้ผลผลิตตามที่ต้องการ แต่ในความเป็นจริงวิธีการผสมพันธุ์เทียมเป็นวิธีที่ได้ผลที่ให้ผลดีที่สุด รวดเร็ว และประหยัด เหมาะสมกับเกษตรกรผู้เลี้ยงสัตว์ แต่ถึงกระนั้นการผสมพันธุ์เทียมก็ยังเป็นสิ่งที่ทำทหาย เป็นปัญหาเสมอ นั่นคือการผสมพันธุ์เทียมไม่ติด ทำให้การลงทุนที่ผสมเทียมไปนั้นสูญเปล่าได้ โดยช่วงเวลาที่เหมาะสมในการผสมพันธุ์เทียมจะเกิดในช่วง ตัวเมียตกไข่ได้ 25-32 ชั่วโมง สเปิร์มมอยู่ในระบบสืบพันธุ์ตัวเมียนาน 48 ชั่วโมงหลังผสม แต่ความสามารถจะไข่แดงจะอยู่ที่ประมาณ 18-24 ชั่วโมง สเปิร์ม (Sperm) ต้องอยู่ในระบบสืบพันธุ์แม่วัวประมาณ 6 ชั่วโมง จึงจะปฏิสนธิกับไข่ได้ เรียกว่า “การเก็บประจุ” (Capacitation) ผลการทดลองของมหาวิทยาลัยเพนซิลวาเนีย สเตท สหรัฐอเมริกาพบว่า การผสมวัวหลัง “ยืนนิ่ง” ให้เห็นครั้งแรก 12 ชั่วโมง เพิ่มอัตราการผสมติดอ้อมท้องสูงขึ้น 30 เปอร์เซ็นต์ การสังเกตจับสัดวัวอย่างน้อยวันละ 2 ครั้ง นาน 30 นาที ตอนเช้าและตอนใกล้ค่ำ มีประสิทธิภาพผลมากที่สุด หรืออาจต้องติดตั้งกล้องวงจรปิด เพื่อสังเกตการ “ยืนนิ่ง” ครั้งแรก เพื่อกำหนดเวลาผสม (ชยุต ดงपालี ธรรม์. 2562)

ตารางที่ 2.1 แสดงอัตราการแสดงพฤติกรรมการติดสัด

| เวลา | ร้อยละแสดงพฤติกรรมติดสัด |
|---------------------|--------------------------|
| 6 A.M. – 12.00 A.M. | 22% |
| 12.00 A.M. – 6 P.M. | 10% |
| 6 P.M. – 12.00 P.M. | 25% |
| 12.00 P.M. – 6 A.M. | 43% |

2.4 โครงข่ายประสาทเทียม

โครงข่ายประสาทเทียม เป็นแขนงหนึ่งของวิทยาการปัญญาประดิษฐ์ Artificial Intelligence หรือ AI โดยเป็นการศึกษาการเลียนแบบการทำงานของระบบประสาทของสิ่งมีชีวิต ที่ประกอบไปด้วย เซลล์ประสาทนิวรอน(neuron) ต่างๆที่เชื่อมต่อประสาท (synapses) ต่อกันกว่า 60 ล้านๆจุดนั้น ซึ่งจะเรียกว่า โหนด (node) ในงาน AI โครงข่ายประสาทเทียมมีการนำไปประยุกต์ใช้งานมากมาย เช่น Pattern recognition, Clustering, Prediction, Forecasting, Optimization หรือ Control system เป็นต้น



รูปที่ 2.3 จำลองโครงข่ายประสาทเทียม

ซึ่งเมื่อเทียบกับโครงข่ายประสาทของสมองมนุษย์นั้น จะประกอบไปด้วย

1. นิวรอน(Neurons) เซลล์หน่วย
2. นิวเคลียส(Nucleus) ส่วนประกอบผลกลางของนิวรอน
3. เดรนไดน์ (Dendrite) ส่วนของ biological neuron ที่รับอินพุตเข้าสู่เซลล์
4. แอ็กซอน (Axon) จุดต่อด้านออก หรือ output

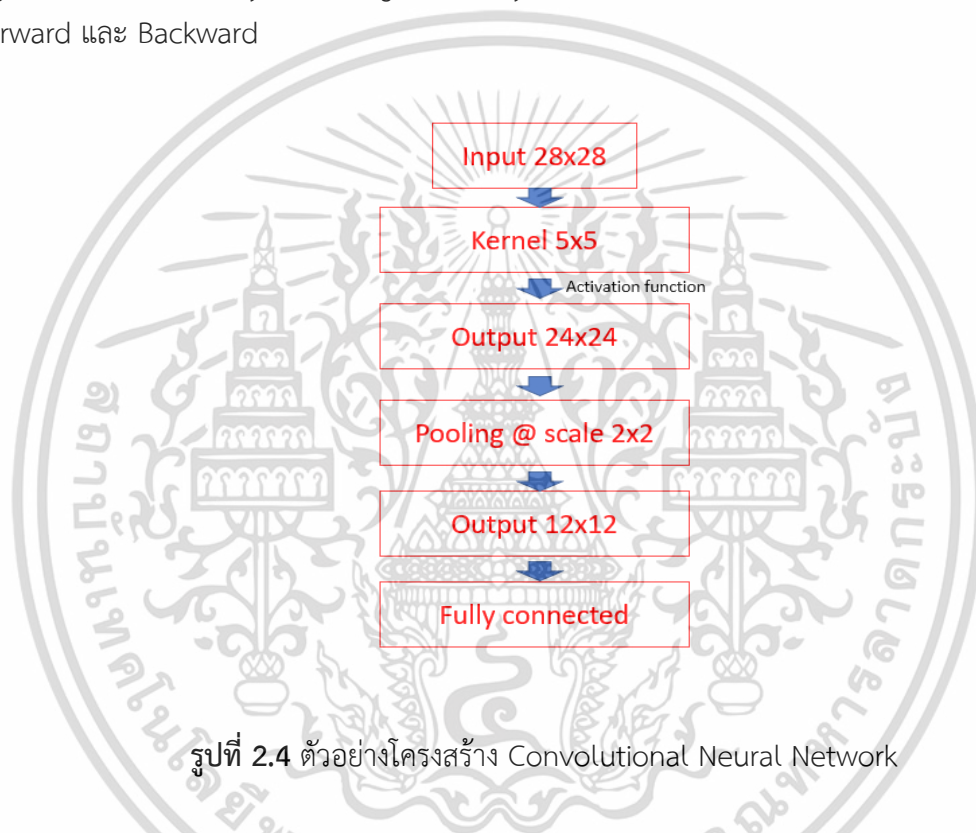
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ไฮแนพส์ (Synapse) จุดเชื่อมต่อ ที่มีการให้น้ำหนัก (Weights)

2.5 Convolutional Neural Network

2.5.1 การทำงานของ Convolutional Neural Network

จากรูป 2.4 แสดงให้เห็นว่าสถาปัตยกรรมของ Convolutional Neural Network ประกอบด้วย Layer Convolution, Layer Pooling และ Fully connected ซึ่งการทำงานแบ่งได้เป็น 2 ส่วน คือ Forward และ Backward



รูปที่ 2.4 ตัวอย่างโครงสร้าง Convolutional Neural Network

2.5.2 Convolution

การคอนโวลูชันเป็นการกรองข้อมูลภาพโดยอาศัยหลักการหาค่าเฉลี่ยเป็นจุด ซึ่งเป็นการลดสิ่งรบกวนรอบๆ พิกเซลของภาพนั้น และเพื่อเป็นการเน้นคุณสมบัติในภาพที่มีความชัดเจนยิ่งขึ้น กระบวนการลดคุณสมบัติรูปที่ไม่ต้องการคอนโวลูชันคือการประมวลผลภาพ (Image) กับ เทมเพลต (Template) โดยกำหนดให้ template มีขนาด $n \times m$ แล้วมาซ้อนกับภาพ เพื่อหาผลลัพธ์ทางเมทริกซ์

ต่อไปเป็นการนำเสนอสมการฟังก์ชันโดยให้ $K(x,y)$ เป็นเทมเพลตขนาด $n \times m$ และภาพ $I(X,Y)$ มีขนาด $N \times M$ ได้ดังนี้

$$I'(X,Y) = K(I) \sum_{i=0}^{n-1} \sum_{j=0}^{m-1} K(i,j) * I(X-i, Y-j) \quad (2.1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$O(X, Y) = f(I'(X, Y)) \quad (2.2)$$

$$f(X) = \max(O, x) \quad (2.3)$$

โดยที่

$I'(X, Y)$ คือ ผลลัพธ์การคอนโวลูชัน

$f(x)$ คือ ฟังก์ชัน *Relu*

$O(X, Y)$ คือ เอาต์พุตของคอนโวลูชัน

m, n คือ ขนาดเทมเพลต

2.5.3 Pooling layer

ในหัวข้อนี้จะกล่าวถึง Max pooling layer ซึ่งเป็น Matrix ที่ปรับค่าได้จากจำนวน Convolution Matrix ที่มีค่ามากที่สุดเทียบกับขนาด pooling ขณะที่ ขนาดของ Max pooling ที่มากเกินไปเวลาคำนวณ Convolution ค่าจะไม่เสถียร และถ้าน้อยไปก็จะใช้ระยะเวลาในการคำนวณนานขึ้น

| | | | | | |
|---|---|---|---|---|---|
| 1 | 0 | 5 | 6 | 5 | 7 |
| 5 | 4 | 1 | 7 | 8 | 4 |
| 8 | 3 | 2 | 4 | | |
| 0 | 7 | 1 | 0 | | |

รูปที่ 2.5 Max pooling layer ที่ ขนาด 2 x 2

Average pooling layer เป็นการหาค่าเฉลี่ยใน Matrix

| | | | | | |
|---|---|---|---|---|-----|
| 1 | 0 | 5 | 6 | 5 | 9.5 |
| 5 | 4 | 1 | 7 | 9 | 3.5 |
| 8 | 3 | 2 | 4 | | |
| 0 | 7 | 1 | 0 | | |

รูปที่ 2.6 Average pooling layer ที่ ขนาด 2 x 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 Fully connected

จากรูปที่ 2.3 ในโครงสร้าง Neural Network จะประกอบไปด้วย โหนดรับข้อมูล(Input) และ โหนด(Output) ในแต่ละความสัมพันธ์ระหว่างโหนดจะมีสิ่งที่เรียกว่า Weight bias ปรับค่าข้อมูลระหว่างโหนด นอกจากนี้ยังมี Hidden layer ในโครงสร้าง Feed forward neural network อีกด้วย สมการความสัมพันธ์ในกรณีที่มีรับข้อมูล 1 โหนดผลลัพธ์ 1 โหนด

$$y = wx + b \quad (2.4)$$

โดยที่

y คือ ค่าเอาต์พุตของโหนดผลลัพธ์
w คือ ค่า Weight
x คือ ค่าอินพุตของโหนดรับข้อมูล
b คือ bias

2.5.5 Activation function

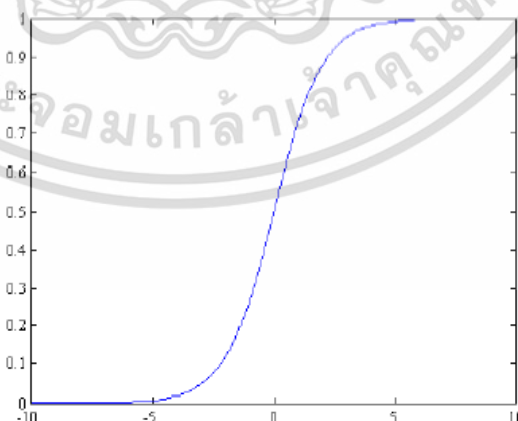
ฟังก์ชันกระตุ้นที่นิยมในงาน Deep Neural Network มีด้วยกันอยู่ 3 ชนิดด้วยกันได้แก่

1. Sigmoid function

สมการใน Sigmoid function แสดงได้ดังสมการที่ 2.5 และการทำงานของฟังก์ชันแสดงในรูปที่

2.7

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.5)$$



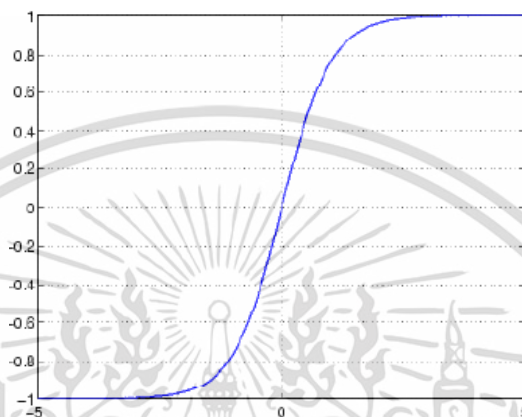
รูปที่ 2.7 ฟังก์ชัน Sigmoid

2. Tan hyperbolic function (Tanh function)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สมการใน Tanh function แสดงได้ดังสมการที่ 2.6 และการทำงานของฟังก์ชันแสดงในรูปที่ 2.8

$$f(x) = \tanh x = \frac{\sinh x}{\cosh x} = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.6)$$

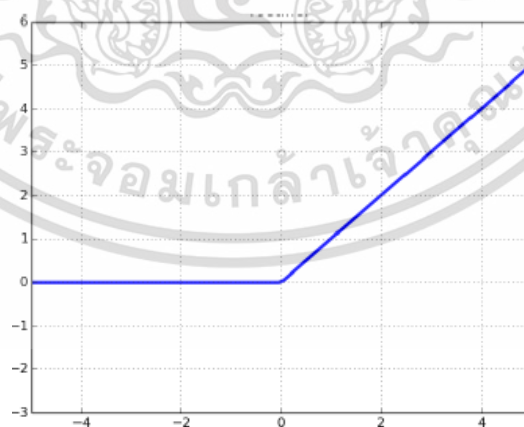


รูปที่ 2.8 ฟังก์ชัน Tanh

3. Rectified Linear function (ReLU function)

สมการใน Tanh function แสดงได้ดังสมการที่ 2.7 และการทำงานของฟังก์ชันแสดงในรูปที่ 2.9

$$f(x) = \max(0, x) \quad (2.7)$$



รูปที่ 2.9 ฟังก์ชัน ReLu

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากสมการที่ 2.4 แสดงให้เห็นความสัมพันธ์ระหว่างโหนดอินพุตและเอาต์พุต ในกรณีที่ในเครือข่ายมีตัว Activation function ขึ้นมาจะสามารถแสดงเป็นสมการที่ 2.8 ได้ดังนี้

$$y = A(wx + b) \quad (2.8)$$

โดยที่

A คือ Activation function

2.5.6 ความผิดพลาดในการเรียนรู้

ความผิดพลาดในการเรียนรู้ หรือ Learning rate error เป็นการวัดประสิทธิภาพของการเทรนนิวรอน โดยเป็นการวัดความผิดพลาดระหว่างเอาต์พุตที่ได้เทียบกับเป้าหมาย (Target) ซึ่งค่าลู่เข้าศูนย์มากเท่าไร ยิ่งแสดงถึงประสิทธิภาพการเรียนรู้เท่านั้น ดังสมการที่ 2.9 ต่อไปนี้

$$E = t - y \quad (2.9)$$

โดยที่

E คือ Learning error

t คือ เป้าหมาย (เป้าหมาย)

y คือ ค่าเอาต์พุตของโหนดเอาต์พุต

Mean Square Error (MSE) คือ ความผิดพลาดที่เกิดจากเอาต์พุตหลายโหนด รวมเข้าด้วยกันดังสมการที่ 2.10 ต่อไปนี้

$$MSE, E = \frac{1}{n} \sum_{n=1}^n (t - y)^2 \quad (2.10)$$

โดยที่

n คือ จำนวนโหนดเอาต์พุต

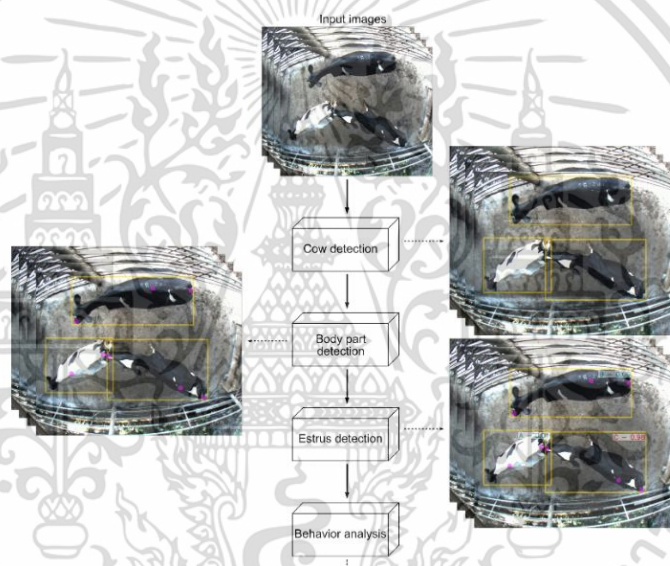
Root Mean Square Error (RMSE)) คือ ความผิดพลาดกำลังสองของข้อมูลจำนวน n เอาต์พุต ได้สมการที่ 2.11 ต่อไปนี้

$$RMSE, E = \sqrt{\frac{1}{n} \sum_{n=1}^n (t - y)^2} \quad (2.11)$$

2.6 งานวิจัยที่เกี่ยวข้อง

Lyu และคณะ (2558 : 1-6) การตรวจจับวัตถุในไฟล์วิดีโอมีการพัฒนาไปอย่างมากในปัจจุบัน โดยเฉพาะในงานจราจรสำหรับตรวจจับรถบนท้องถนน Raghunandan และคณะ กล่าวถึงการใช้งานตรวจจับวัตถุสำหรับงานเฝ้าระวัง Norouzzadeh และคณะ (2558 : E5716-E5725) รวมถึงงานเฝ้าระวังสัตว์

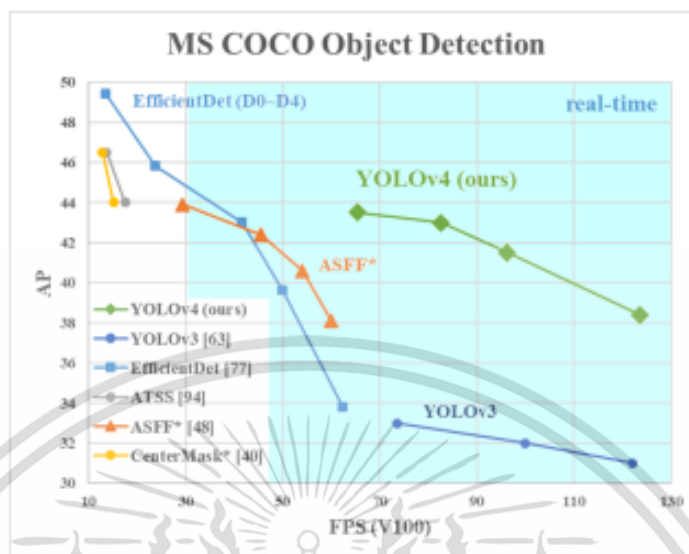
Thanawat และคณะ (2562) ได้เสนอระบบตรวจจับการเป็นสัตว์ผ่านระบบปัญญาประดิษฐ์ ซึ่งได้พัฒนาโมเดลคอนโวลูชันเน็ตเวิร์คขึ้นมาชื่อว่า CowXNet โดยแบ่งออกเป็น 4 ส่วนคือ 1) การตรวจจับวัวทั้งตัว 2) การตรวจจับส่วนต่างๆบนตัววัว 3) การตรวจจับการติดสัด 4) การวิเคราะห์พฤติกรรมของวัว สามารถแสดงได้ดังรูป ที่ 2.8 ดังนี้



รูปที่ 2.10 การตรวจจับและวิเคราะห์ของระบบ CowXNet

งานตรวจจับวัวในไฟล์วิดีโอ มีหลายโมเดลที่ได้รับความนิยมเช่น Convolutional Neural Network (CNN) อย่างเช่น Bochkovskiy และคณะ (2560 : 10934) ตัวโมเดล YOLOV4 หรือ Lin และคณะ [16] RetinaNet

โดยเฉพาะตัวโมเดล YOLOV4 มีจุดเด่นในเรื่องของความแม่นยำในการตรวจจับข้อมูลขนาดใหญ่ โดยเมื่อเทียบกับชุดฐานข้อมูลมาตรฐาน COCO ชุดฐานข้อมูลกับโมเดลอื่นแล้ว ดังรูปที่ 2.8



รูปที่ 2.11 การประสิทธิผลของ YOLOv4 เทียบกับโมเดลอื่น

จากรูปที่ 2.8 แสดงถึงการเปรียบเทียบโมเดลตรวจจับวัตถุ YOLOv4 กับโมเดลตรวจจับวัตถุอื่นๆ โดยแสดงให้เห็นว่า YOLOv4 สามารถประมวลผลได้เร็วกว่าโมเดล EfficientDet ถึงสองเท่า และ YOLOv4 เป็นรุ่นพัฒนาจาก YOLOv3 ที่ค่าความเฉลี่ยความแม่นยำสูงขึ้น 10% และ ประมวลผลเร็วขึ้น 12 %

บทที่ 3

การจำแนกพฤติกรรมการติดสัดของวัวด้วยปัญญาประดิษฐ์แบบ การเรียนรู้เชิงลึก

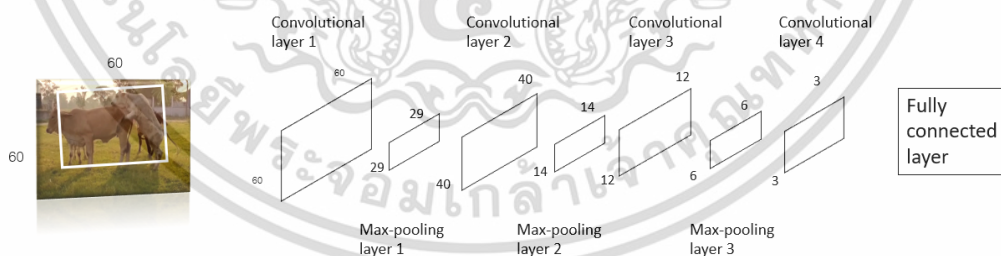
ในบทนี้จะกล่าวถึงการออกแบบโมเดลจำแนกพฤติกรรมการติดสัดของวัวด้วยปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึก การถ่ายโอนการเรียนรู้ การทำงานของโครงข่ายประสาทเทียมแบบคอนโวลูชัน ฐานข้อมูลที่ใช้ในการทดลอง อุปกรณ์ที่ใช้ ขั้นตอนการทดลอง และการประเมินความแม่นยำของการการจำแนกที่ใช้ในการเรียนรู้ และทดสอบ

3.1 โมเดลจำแนกพฤติกรรมการติดสัดของวัวโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน

โมเดลจำแนกพฤติกรรมการติดสัดของวัวโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชันในวิทยานิพนธ์นี้จะนำเสนอวิธีในการจำแนกพฤติกรรมตาม Feature extraction หรือ การดึงลักษณะเด่นออกมา

3.1.1 การออกแบบโมเดลจำแนกพฤติกรรมโดยใช้โครงข่ายประสาทเทียมแบบคอนโวลูชัน

อ้างอิงตามงานวิจัยของ สุทิพา และคณะ (2555 : 306-309) ได้ออกแบบโครงข่ายประสาทเทียมที่ดึงลักษณะเด่นออกมา โดยให้ภาพอินพุต มีขนาด 60 x 60 พิกเซล ดังนั้นรูปแบบโครงข่ายประสาทเทียมสำหรับการจำแนกพฤติกรรมการติดสัดของวัวก็จะเป็น 60 x 60 พิกเซล ได้ดังรูปที่ 3.1 ต่อไปนี้



รูปที่ 3.1 แสดงการออกแบบโครงสร้าง CNN

ภาพอินพุตเป็นภาพสีขนาด 60x60 พิกเซล จึงทำให้มิติในชั้นอินพุตเป็น 60x60x3 โดยเลข 3 คือ จำนวนชั้นของภาพ ซึ่งในที่นี้ภาพอินพุตเป็น ภาพสี RGB จึงทำให้ภาพมี 3 ชั้น ในชั้นคอนโวลูชันชั้นที่ 1 (Convolutional layer 1) เป็นการนำภาพอินพุตไปคอนโวลูชัน ทำให้ได้ภาพเอาต์พุตเป็นแผนภาพคุณลักษณะเด่นที่ได้จากการทำคอนโวลูชันขนาด 57x57x20 โดยสาเหตุที่ขนาดของภาพลดลงจาก

60×60 พิกเซล เหลือ 57×57 พิกเซล เนื่องจากการสูญเสียขอบของภาพไปจากการหาคอนโวลูชัน ซึ่งสมการการหาคอนโวลูชันสามารถแสดงได้ดังสมการที่ 3.1

$$y(i,j) = A(wx(i,j) + b) \quad (3.1)$$

โดยที่

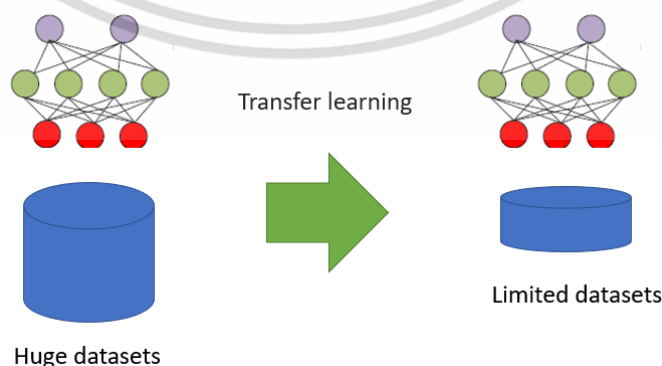
- $y(i,j)$ คือ ภาพเอาต์พุต
- A คือ ฟังก์ชันกระตุ้น เลือกใช้แบบ Relu
- w คือ ค่าน้ำหนักหรือ Weight ของคอนโวลูชัน
- b คือ ค่าไบแอสข้อมูล มีค่าเท่ากับ 0
- $x(i,j)$ คือ ภาพอินพุต

จากนั้นเข้าสู่ชั้น Pooling ด้วยค่าสูงสุดชั้นที่ 1 (Max-pooling layer 1) โดยใช้ขนาด 2×2 กับเอาต์พุตที่ได้จากชั้นคอนโวลูชันชั้นที่ 1 เพื่อทำให้ขนาดของภาพลดลงครึ่งหนึ่ง โดยได้แผนภาพคุณลักษณะเด่นที่ได้จากการทำ Pooling ขนาด 29×29×20 ซึ่งเหตุผลที่เลือกใช้การทำพูลลิงด้วยค่าสูงสุดนั้น เพื่อให้เอาต์พุตจากการทำ Pooling ยังคงมีลักษณะเด่นไว้

ซึ่งในโครงข่ายคอนโวลูชันชั้นสุดท้าย พบว่าค่าความถูกต้ององในการเรียนรู้มีค่าที่ต่ำ โดยจำนวนของชั้นคอนโวลูชันเลเยอร์เอง หรือด้วยประสิทธิภาพของอุปกรณ์ที่ใช้ในการทำงานของคอนโวลูชันก็ตาม ทำให้พบว่า การถ่ายโอนการเรียนรู้ หรือ Transfer learning ให้ค่าความแม่นยำที่สูงกว่า

3.1.2 Transfer learning

เนื่องจากวิทยานิพนธ์ฉบับนี้ มีฐานข้อมูลที่ใช้ในการเรียนรู้ค่อนข้างจำกัด ไม่มากพอที่จะให้โครงข่ายประสาทเทียมคอนโวลูชัน สามารถเกิดการเรียนรู้ได้ จึงมีการทำวิธีการทำ ถ่ายโอนการเรียนรู้ หรือ Transfer learning เพื่อเพิ่มอัตราความสามารถในการเรียนรู้ ซึ่งแสดงการทำงานการถ่ายโอนการเรียนรู้ได้ดังรูปที่ 3.2 ต่อไปนี้



รูปที่ 3.2 Transfer learning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.2 การเรียนเรียนรู้ผ่านคอนโวลูชันแบบถ่ายโอนการเรียนรู้ นั้น มีขั้นตอนการทำงานจากฐานข้อมูลขนาดใหญ่ที่ผ่านการเรียนรู้มาแล้ว เพื่อให้ระบบโมเดลของฐานข้อมูลของวิทยานิพนธ์สามารถเรียนรู้ลักษณะของฐานข้อมูลขนาดใหญ่ได้ ซึ่งโครงสร้างที่ใช้สอนกับฐานข้อมูลขนาดใหญ่และฐานข้อมูลขนาดเล็กจะต้องใช้โครงสร้างแบบเดียวกันทำให้ผลสุดท้ายในการทำงานของระบบนี้ดีเทียบเท่ากับการใช้ฐานข้อมูลขนาดใหญ่ แม้ว่าในความจริงนั้นจะใช้เพียงฐานข้อมูลขนาดเล็กเท่านั้น

ในงานวิทยานิพนธ์นี้ จะใช้โมเดลที่ผ่านการเรียนรู้มาแล้วคือ SSD Mobile Net V2 FPN, SSD ResNet101 V1 FPN, YOLOv5s และ Mask-RCNN

3.2 ฐานข้อมูลที่ใช้ในการทดลอง

ฐานข้อมูลที่ใช้ในการทดลองประกอบไปด้วย วิดีโอแสดงพฤติกรรมของวัว 2 แบบ คือ พฤติกรรมปกติทั่วไปของวัว และ พฤติกรรมแสดงอาการติดสัดของวัว



รูปที่ 3.3 พฤติกรรมปกติทั่วไปของวัว

รูปที่ 3.3 พฤติกรรมทั่วไปของวัว ในสถานะธรรมชาติแบบเปิด โดยพฤติกรรมปกติของวัวจะอยู่รวมกันเป็นฝูง



รูปที่ 3.4 พฤติกรรมการติดสัดของวัว

รูปที่ 3.4 แสดงพฤติกรรมการติดสัดของวัว ในสภาวะธรรมชาติแบบเปิด โดยพฤติกรรมปกติของวัว จะขึ้นคร่อมวัวอีกตัว แสดงถึงอาการกิริยาสันชาตियานพร้อมสืบพันธุ์ของวัวได้ โดยในการทดลองจะเก็บชุดข้อมูลมาจาก สมาร์ทโฟนซึ่งถ่ายคลิปวิดีโอสกุล (.MP4) มาจำนวน 10 วิดีโอไฟล์ ที่แตกต่างกัน ทั้งวัว สถานที่ เวลา ด้วยเฟรมเรทที่ 30 เฟรมต่อวินาที (fps) หลังจากนั้นจะถูกดึงเป็นภาพเพื่อแบ่งเป็นข้อมูล สำหรับเรียนรู้ (Test) และ ทดสอบ (Test) ตามตารางที่ 3.1

ตารางที่ 3.1 แสดงจำนวนข้อมูลสำหรับการทดลอง

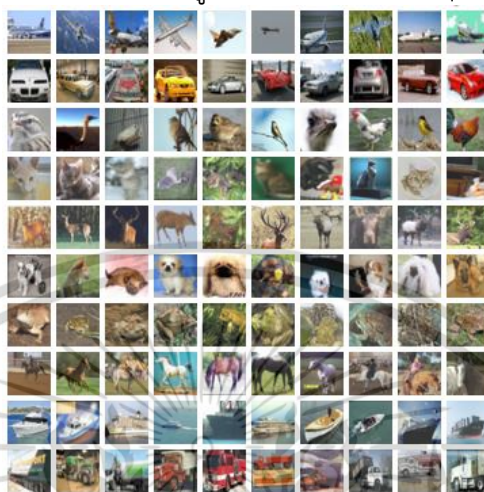
| ชนิดชุดข้อมูล | จำนวนข้อมูล (ภาพ) |
|----------------|-------------------|
| ข้อมูลเรียนรู้ | 425 |
| ข้อมูลทดสอบ | 75 |

3.2.1 COCO ชุดฐานข้อมูล

COCO ชุดฐานข้อมูล เป็นข้อมูลมาตรฐานขนาดใหญ่ ที่ถูกใช้สำหรับงานสาย object detection, segmentation, and captioning. ชุดข้อมูล COCO เป็นที่นิยมในงาน Computer vision ที่ทำการ labelling ทั้งภาพ วัตถุ สัตว์ ผู้คน ไปแล้วกว่า 328,000 กวาร์รูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่ง COCO มักถูกใช้เป็นชุดข้อมูลเปรียบเทียบ Benchmark กับ NN algorithms ต่างๆสำหรับในงาน real time object detection และ มักถูกใช้ใน libraries ต่างๆเช่นกัน



รูปที่ 3.5 ชุดข้อมูลมาตรฐาน COCO ชุดฐานข้อมูล

3.3 Data preprocessing

หลังจากที่จัดการแบ่งชุดข้อมูลเสร็จแล้วนั้น จะเข้าสู่กระบวนการ Preprocessing ซึ่งเป็นการทำความสะอาดข้อมูล (Data cleaning) การสอนข้อมูลให้ NN (Labelling) การเพิ่มข้อมูลสำหรับการป้องกันข้อมูล Overfitting หรือ การทำ augmentation ให้กับข้อมูล

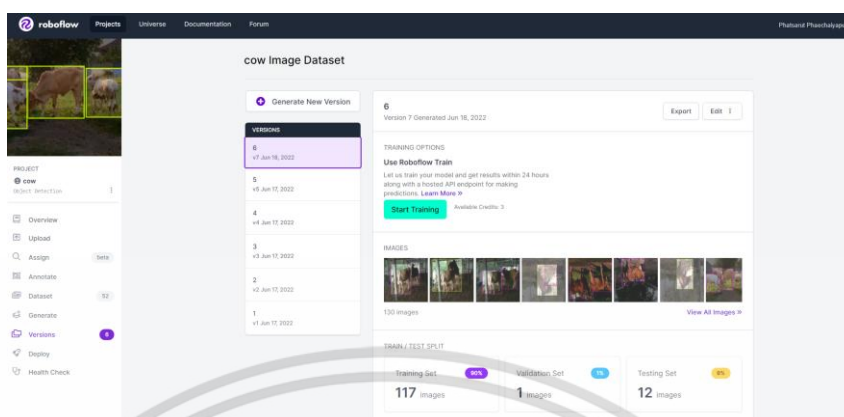
3.3.1 Roboflow labelling tool

Roboflow เป็นแพลตฟอร์มออนไลน์สำหรับการทำ labelling ข้อมูลที่เป็นภาพ ซึ่งมีเครื่องมือที่จำเป็นสำหรับงาน Computer vision ไม่ว่าจะเป็น งานรวบรวมภาพเป็นหมวดหมู่ หรือ data collection หรือ การทำ annotations ภาพ ซึ่งสามารถทำการ annotate ภาพได้หลายแบบ เช่น การตีกรอบ Rectangular box หรือ การทำ Mask ตีกรอบรอบวัตถุ

Roboflow สามารถทำ augmentation ภาพข้อมูลเพื่อเพิ่มจำนวนข้อมูล ไว้สำหรับปรับปรุงประสิทธิภาพการทำงานของคอนโวลูชัน ที่ง่ายและรวดเร็ว

สกุลไฟล์ annotate สามารถทำได้ทั้ง ไฟล์ JSON, XML, CSV และ TXT สำหรับรองรับทุก algorithms ต่างๆได้

Roboflow รองรับ api สำหรับการดึงฐานข้อมูลที่ได้จากการทำ preprocessing ไปแพลตฟอร์มการทำคอนโวลูชัน โดยไม่ต้องดาวน์โหลดลงเครื่องคอมพิวเตอร์



รูปที่ 3.6 Roboflow labelling tool platform

3.3.2 Data labelling

การทำให้โครงข่ายประสาทเทียมคอนโวลูชัน หรือ CNN เกิดการเรียนรู้ จำเป็นต้องสอนให้ NN ถึงข้อมูลที่จะเป็นอินพุตเข้าไป ในอัลกอริทึมต่างๆ ใช้สกุลไฟล์ และรูปแบบของข้อมูลที่ต่างกัน ในหัวข้อย่อยที่ 3.3.2 จะกล่าวถึงการสอนให้อัลกอริทึมรับรู้ถึงข้อมูลที่จะใช้ทำการเรียนรู้ ตามตารางที่ 2.2

ตารางที่ 3.2 แสดงรูปแบบการสอนข้อมูลตามแต่ละโมเดล

| Pre-trained model | Labelling |
|----------------------|-------------|
| SSD MobileNet V1 FPN | Rectangular |
| SSD ResNet101 V1 FPN | Rectangular |
| YOLOv5s | Rectangular |
| Faster-RCNN X101 FPN | Polygon |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.7 Data labelling แบบ Rectangular

รูปที่ 3.7 แสดงถึงการ label หรือสอนข้อมูลแบบ การตีกรอบสี่เหลี่ยมซึ่งใช้ในอัลกอริทึมหลายๆตัว ในวิทยานิพนธ์นี้ ใช้ อัลกอริทึม Pre-trained model ได้แก่ SSD MobileNet V1 FPN, SSD RestNet101 V1 FPN, YOLOv5s



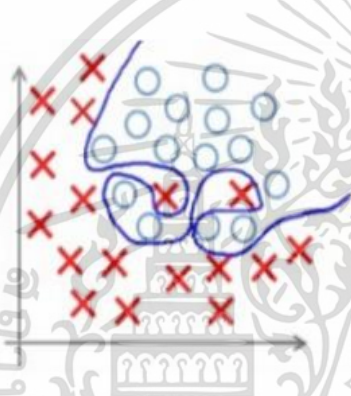
รูปที่ 3.8 Data labelling แบบ Polygon

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

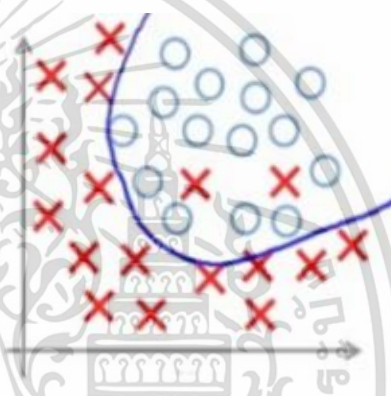
รูปที่ 3.8 แสดงถึงการ label หรือสอนข้อมูลแบบ การตีกรอบรอบตัววัตถุซึ่งใช้ในอัลกอริทึมแบบMask ในวิทยานิพนธ์นี้ ใช้ อัลกอริทึม Pre-trained model ได้แก่ Faster-RCNN X101 FPN ในการทดลอง

3.3.3 Data augmentation

ปัญหาหลักในการทำเรียนรู้เชิงลึก หรือ Deep Neural Network (DNN) คือปัญหาจำนวน ชุดข้อมูลไม่เพียงพอต่อการฝึกการเรียนรู้ การทำ Data augmentation คือการนำภาพชุดข้อมูลเดิมมาผ่านกระบวนการทางภาพ หรือ Image processing เช่นทำการ ย่อ/ขยาย หมุน Flip Crop เป็นต้น เพื่อเป็นการเพิ่มจำนวนให้มากพอที่จะทำให้ประสิทธิภาพการเรียนรู้ดีขึ้น ทำให้ข้อมูลมีการ fitting พอดี ลดการเกิดข้อมูลฝึกสอน Overfitting ขึ้นมา ตามตัวอย่างรูปที่ 3.7 ที่ผ่านการทำ Augmentation มาแล้ว



รูปที่ 3.9 แสดงข้อมูล Over-fitting



รูปที่ 3.10 แสดงข้อมูล fitting

รูปที่ 3.9 แสดงให้เห็นถึงข้อมูลมีการ Over-fitting ระหว่าง Test และ Train ชุดฐานข้อมูล ทำให้ไม่มีความแม่นยำในการทำนาย เทียบกับ รูปที่ 3.10 ที่ชุดข้อมูล มีการ fitting พอดี ทำให้มีความแม่นยำในการทำนาย

3.4 pre-trained model

Pre-trained model เป็นอัลกอริทึมโมเดล ที่ผ่านการเรียนรู้มาแล้ว กับชุดฐานข้อมูลขนาดใหญ่ COCO ชุดฐานข้อมูล ตัวโมเดลที่ผ่านการเรียนรู้มาแล้วสามารถลดเวลาในการฝึกสอน เพิ่มประสิทธิภาพในการเรียนรู้ให้ดียิ่งขึ้น

สำหรับการประเมินความสามารถในการเรียนรู้ในแต่ละชุดโมเดลที่ผ่านการเรียนรู้มาแล้วนั้นจะใช้ ค่าเฉลี่ยในความแม่นยำ หรือ Average precision model ซึ่งเปรียบเทียบ (Benchmark) กับชุดฐานข้อมูล COCO ชุดฐานข้อมูล

ในวิทยานิพนธ์นี้จะใช้ตัว pre-trained model ดังต่อไปนี้ คือ SSD MobileNet V2 FPN, SSD ResNet101 V1 FPN, YOLOv5s และ Mask-RCNN

3.4.1 SSD MobileNet V1 FPN

สถาปัตยกรรม MobileNet มีชั้นเลเยอร์สำหรับทำคอนโวลูชัน และ Pooling ทั้งหมด 28 ชั้น ซึ่งทำการฝึกสอนกับชุดฐานข้อมูลขนาดใหญ่ COCO ชุดฐานข้อมูล (2017) ให้ค่าความแม่นยำสูงสุดถึง 29.1 %

3.4.2 SSD ResNet101 V1 FPN

สถาปัตยกรรม MobileNet มีชั้นเลเยอร์สำหรับทำคอนโวลูชัน และ Pooling ทั้งหมด 101 ชั้น ซึ่งทำการฝึกสอนกับชุดฐานข้อมูลขนาดใหญ่ COCO ชุดฐานข้อมูล (2017) ให้ค่าความแม่นยำสูงสุดถึง 35.6 %

3.4.3 YOLOv5s

สถาปัตยกรรม YOLO เป็นอัลกอริทึมที่ได้รับความนิยมเป็นอย่างสูงในงานตรวจจับวัตถุ Object detection มีความโดดเด่นในเรื่องของการคำนวณที่รวดเร็ว และความแม่นยำในการฝึกสอน ให้ค่าความแม่นยำสูงสุดถึง 56.8 %

3.4.4 Faster-RCNN X101 FPN

สถาปัตยกรรม RCNN เป็นอัลกอริทึมสำหรับงานจำแนกวัตถุหรือ Object segmentation ซึ่งทำการฝึกสอนกับชุดฐานข้อมูลขนาดใหญ่ COCO ชุดฐานข้อมูล ให้ค่าความแม่นยำสูงสุดถึง 43.0 %

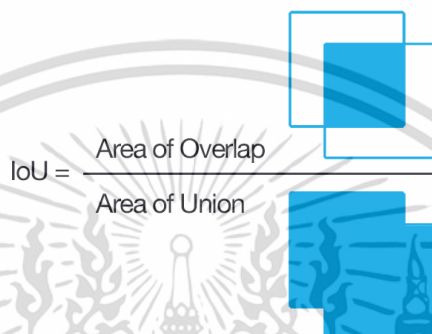
ตารางที่ 3.3 แสดงประสิทธิภาพความแม่นยำของแต่ละ Pre-trained model

| Pre-trained model | Average Precision (AP) |
|----------------------|------------------------|
| SSD MobileNet V1 FPN | 29.1 % |
| SSD ResNet101 V1 FPN | 35.6 % |
| YOLOv5s | 37.4 % |
| Faster-RCNN X101 FPN | 43.0 % |

3.5 การประเมินโมเดล

การวัดประสิทธิภาพของงาน Object detection วิธีที่เป็นที่นิยมในปัจจุบันคือใช้วิธี Intersection Over Union (IoU) เพื่อประเมินความถูกต้องของกรอบสี่เหลี่ยมที่ทำนาย เมื่อเปรียบเทียบกับกรอบสี่เหลี่ยม Ground truth

ดัชนี IoU สามารถคำนวณได้ดังรูปที่ 3.11 ดังนี้



รูปที่ 3.11 วิธีการคำนวณ Intersection Over Union (IoU)

นอกจาก IoU จะคำนวณความถูกต้องของกรอบแล้วนั้น IoU ยังวิเคราะห์การตรวจจับนั้นถูกต้อง (True Positive) หรือ ไม่ถูกต้อง (False Positive) ตามรูปที่ 3.12



รูปที่ 3.12 ผลเทียบการตรวจจับกับข้อมูลที่สอน

3.5.1 การวัดประสิทธิภาพความถูกต้อง

Average precision (AP) ถูกนำมาใช้ในการประเมินประสิทธิภาพของความแม่นยำการตรวจจับวัตถุ เมื่อเปรียบเทียบกับ Ground truth ข้อมูลที่สอนเข้าไปใน NN โดยคำนวณได้จากสมการที่ 3.2 ดังนี้

$$AP = \sum_{i=1}^{n-1} (r_{i+1} + r_i) p_{interp}(r_{i+1}) \quad (3.2)$$

โดยที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

r คือ Recall ความแม่นยำของข้อมูล
Pinterp คือ ค่าความแม่นยำของภาพใหม่ที่สอดแทรกเข้ามา

ค่าความแม่นยำ สามารถคำนวณได้จากสมการที่ 3.3 ดังนี้

$$P = \frac{TP}{TP+FP} \quad (3.3)$$

โดยที่

TP คือ True Positive

FP คือ False Positive

Mean average precision (mAP) คำนวณค่าเฉลี่ยกลางของความแม่นยำ ได้จากสมการที่ 3.4 ดังนี้

$$mAP = \frac{AP_i}{K} \quad (3.4)$$

โดยที่

K คือ จำนวนทั้งคลาสข้อมูลที่สอน

AP_i คือ ค่าเฉลี่ยของความแม่นยำที่ i

3.6 ขั้นตอนการทดลอง

ในวิทยานิพนธ์นี้ศึกษาการตรวจสอบพฤติกรรมการติดสัดของวัวด้วยโครงข่ายประสาทเทียมคอนโวลชัน โดยในขั้นตอนการทดลองนั้นมีลำดับดังต่อไปนี้

- 1) ชุดข้อมูลไฟล์วิดีโอ (.MP4) จะถูกนำเข้าแพลตฟอร์ม Roboflow เพื่อทำการลดรูป ปรับขนาดภาพ กำหนดคลาสข้อมูล ซึ่งประกอบไปด้วย 1.Cow_normal 2. Cow_jump หลังจากนั้นนำภาพมาเข้าเทคนิคAugmentation เพื่อเพิ่มฐานข้อมูล เป็นอันเสร็จสิ้นขั้นตอนการทำ Data preprocessing
- 2) ในส่วนของ โมเดล SSD MobileNet V1 FPN และ SSD RestNet101 V1 FPN จะใช้แพลตฟอร์มของ TensorFlow object detection ส่วน YOLOv5s ใช้แพลตฟอร์มของ Yolo เอง และสุดท้าย Faster-RCNN X101 FPN ใช้แพลตฟอร์มของ detectron ซึ่งสะดวกต่อการใช้งาน และรวดเร็วในการทดลอง
- 3) การทดลองจะใช้ CNN ของแต่ละโมเดล เรียนรู้และตรวจจับวัตถุตามที่สอนข้อมูลไป
- 4) การประเมินประสิทธิภาพความแม่นยำ ในวิทยานิพนธ์ใช้ ค่าเฉลี่ยกลางความแม่นยำ หรือ Mean average precision(mAP) ในเปรียบเทียบในแต่ละโมเดลที่ได้ทำการฝึกสอนไปก่อนหน้านี้แล้วนั้น

บทที่ 4

การทดลองและวิเคราะห์ผลการทดลอง

ในบทนี้จะกล่าวถึงการทดลองตรวจสอบพฤติกรรมการติดสัดของวัวผ่านปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึก หรือ Deep Neural Network(DNN) ตามขั้นตอนที่ได้กล่าวไปแล้วในบทที่ 3 ในส่วนของวิธีการทดลอง โดยในส่วนแรกนั้นจะกล่าวถึงการทำ Data Processing จากนั้นเตรียมเข้ามาเข้าโมเดลที่ทำการฝึกสอนไว้แล้ว หรือ Pre-trained model ซึ่งในการทดลองจะเรียงตามโมเดลต่อไปนี้ SSD MobileNet V2 FPN, SSD ResNet101 V1 FPN, YOLOv5s และ Mask-RCNN ตามลำดับ หลังจากที่ได้ตัวโมเดลที่ฝึกสอนเสร็จแล้ว จะทำการประเมินเปรียบเทียบประสิทธิภาพความแม่นยำโดยใช้ค่าเฉลี่ยกลางความแม่นยำ หรือ Mean Average Precision(mAP) ที่ Intersect of Union(IoU) ต่างๆ และในส่วนสุดท้ายเป็นการแสดงผลประสิทธิภาพของโมเดลที่ได้ฝึกมารวมกับไฟล์วิดีโอฐานข้อมูลที่ได้เตรียมมา หรือ Image Processing

4.1 การจัดการฐานข้อมูลในการทดลอง

ในขั้นตอนจัดการฐานข้อมูล Data preprocessing ซึ่งฐานข้อมูลทั้งหมดเป็นไฟล์วิดีโอ ที่ได้จากเกษตรกรในพื้นที่ จำนวน 10 ไฟล์ หลังจากนั้นไฟล์วิดีโอทั้งหมด จะถูกอัปโหลดเข้าสู่แพลตฟอร์ม Roboflow สำหรับทำการจัดการฐานข้อมูล หลังจากทำการลดขนาดรูปภาพจากวิดีโอออกมา เพื่อมาปรับแต่ง ขนาดภาพ และทำ Augmentation จำนวนฐานข้อมูลภาพใช้ในการสอนและทดสอบนั้น เลือกโดย 80% ของจำนวนภาพทั้งหมดของทั้ง 2 คลาสข้อมูลจะเป็นรูปที่ใช้ในการสอน และอีก 20% ของจำนวนภาพจะใช้ในการทดสอบ

ตารางที่ 4.1 การจัดการฐานข้อมูล

| ฐานข้อมูล | จำนวนข้อมูล (ภาพ) | ชุดสอน | ชุดทดสอบ |
|----------------------------|-------------------|--------|----------|
| ฐานข้อมูลพฤติกรรมวัวปกติ | 425 | 340 | 85 |
| ฐานข้อมูลพฤติกรรมวัวติดสัด | 75 | 60 | 15 |

จากตารางที่ 4.1 ฐานข้อมูลรูปที่ได้จากการลดขนาดรูปภาพจากวิดีโอไฟล์มมาทั้งหมด 500 รูป โดยแบ่งเป็น ภาพเหตุการณ์วัวปกติ 425 รูป ถูกใช้เป็นภาพชุดสอน 340 รูป ชุดทดสอบ 85 รูป และภาพเหตุการณ์วัวติดสัด 75 รูป ถูกใช้เป็นภาพชุดสอน 60 รูป ชุดทดสอบ 15 รูป

4.2 การฝึกสอนภาพฐานข้อมูลโดยโมเดลผ่านการเรียนรู้

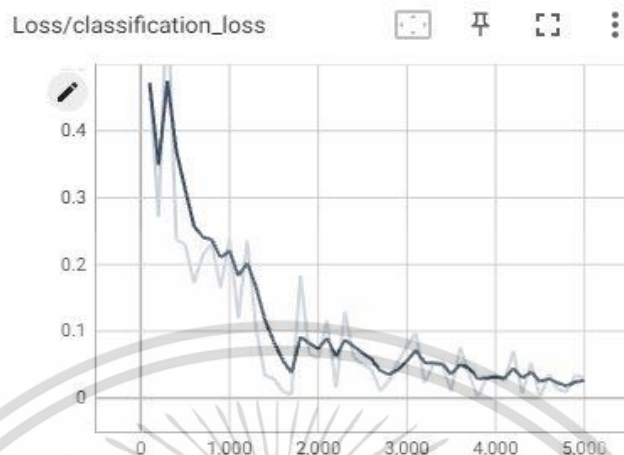
หลังจากที่จัดการฐานข้อมูล Data preprocessing ข้อมูลรูปภาพจะถูกเก็บไว้บนคลาวด์ ในหัวข้อนี้จะกล่าวถึงการทดลองนำข้อมูลไปฝึกการเรียนรู้กับโมเดลที่ผ่านการเรียนรู้มาแล้ว ในวิทยานิพนธ์นี้จะทำการฝึกการเรียนรู้ผ่านแพลตฟอร์มออนไลน์อย่าง Google Collaboratory ซึ่งมี GPU NVIDIA P100 สำหรับประมวลผลการคอนโวลูชันในนั้น

ในวิทยานิพนธ์นี้ใช้โมเดลดังต่อไปนี้ SSD MobileNet V2 FPN, SSD ResNet101 V1 FPN, YOLOv5s และ Mask-RCNN ตามลำดับ โดยในแต่ละโมเดลจะมีแพลตฟอร์มเป็นของตัวเอง เช่น โมเดล SSD MobileNet V2 FPN และ SSD ResNet101 V1 FPN จะใช้แพลตฟอร์มของ TensorFlow object detection และ Mask-RCNN จะใช้แพลตฟอร์มของ detectron เป็นต้น ซึ่งในการเรียนรู้ผ่านโมเดลที่ผ่านการเรียนรู้มาแล้ว จะได้ผลของ loss ข้อมูลออกมา เพื่อแสดงให้เห็นว่า การเรียนรู้ของโมเดล ลู่เข้าสู่ศูนย์ ไม่เกิด Overfitting ของข้อมูลขึ้นมา หลังจากนั้นจะกล่าวถึงการประเมินประสิทธิภาพความแม่นยำของโมเดล ในหัวข้อถัดไป

4.2.1 SSD MobileNet V1 FPN

โมเดล SSD MobileNet V1 FPN เป็นโมเดลที่มีสถาปัตยกรรมแบบ SSD หรือ Single Shot Detection มีคุณสมบัติเป็นโมเดลที่มีขนาดเล็ก ใช้เวลาในการเรียนรู้เร็ว มักถูกใช้ในอุปกรณ์สมาร์ตโฟน มีชั้นเลเยอร์อยู่ 28 ชั้น มีค่าความแม่นยำเมื่อเปรียบเทียบกับ ชุดฐานข้อมูลมาตรฐาน COCO ชุดฐานข้อมูลอยู่ที่ 29.1 % ในการทดลอง หลังจากขั้นตอน Data preprocessing ภาพขนาด 640 x 640 พิกเซล จะถูกอัปโหลดขึ้นบนคลาวด์ ของแพลตฟอร์ม TensorFlow object detection ทั้งชุดสอน และชุดทดสอบ โดยผลการเรียนรู้จะแสดงออกในมาแพลตฟอร์ม TensorFlow ในรูปแสดงที่ 4.1 และ 4.2 ตามลำดับ

ในการทดลอง พารามิเตอร์ที่สำหรับการทดลอง โดย input ชุดฐานข้อมูล จะรับรูปที่ความกว้าง 640 พิกเซล สูง 640 พิกเซล จำนวนรอบในการเรียนรู้อยู่ที่ 5000 steps และมีอัตราการเรียนรู้อยู่ที่ 0.05



รูปที่ 4.1 การลู่เข้าของ Classification loss โมเดล SSD MobileNet V1 FPN



รูปที่ 4.2 การลู่เข้าของ Localization loss โมเดล SSD MobileNet V2 FPN

จากรูปที่ 4.1 และ 4.2 แสดงถึงความสามารถในการเรียนรู้โมเดล SSD MobileNet V1 FPN ที่ข้อมูลการสูญหาย Loss ทั้ง classification แสดงถึงคุณลักษณะการจำแนกคลาสต่างๆ นั้นถูกต้องหรือไม่ และ localization แสดงถึงคุณลักษณะการทำนายตำแหน่งของคลาสที่ได้สอนไว้

จากการทดลองการเรียนรู้โมเดล SSD MobileNet V1 FPN พบว่า ค่า Total loss ถึงเป็นการคำนวณจาก classification loss และ localization loss มีค่าอยู่ที่ 90 % และเมื่อนำโมเดลมาวัดกับประสิทธิภาพความแม่นยำ Mean Average Precision (mAP) ที่ Intersect of Union (IoU) ที่ ช่วง 0.5 จนถึง 0.9 อยู่ที่ 0.65

4.2.2 SSD ResNet101 V1 FPN

โมเดล SSD ResNet101 V1 FPN เป็นโมเดลที่มีสถาปัตยกรรมแบบ SSD หรือ Single Shot Detection มีคุณสมบัติเป็นโมเดลที่มีชั้นเลเยอร์ทั้งหมด 101 ชั้นที่ทำคอนโวลูชัน และ Pooling ด้วย มีความแม่นยำมากกว่า SSD MobileNet V2 FPN เมื่อเปรียบเทียบกับกับตัว ชุดฐานข้อมูลมาตรฐาน COCO ชุดฐานข้อมูล อยู่ที่ 35.6 %

ในการทดลอง หลังจากขั้นตอน Data preprocessing ภาพขนาด 640 x 640 พิกเซล จะถูกอัปโหลดขึ้นบนคลาวด์ ของแพลตฟอร์ม TensorFlow object detection ทั้งชุดสอน และชุดทดสอบ โดยผลการเรียนรู้จะแสดงออกมาแพลตฟอร์ม TensorFlow ในรูปแสดงที่ 4.3 และ 4.4 ตามลำดับ

ในการทดลอง พารามิเตอร์ที่สำหรับการทดลอง โดย input ชุดฐานข้อมูล จะรับรูปที่ความกว้าง 640 พิกเซล สูง 640 พิกเซล จำนวนรอบในการเรียนรู้อยู่ที่ 5000 steps และมีอัตราการเรียนรู้อยู่ที่ 0.05 ซึ่งเป็นการตั้งค่าเดียวกับ SSD MobileNet V2 FPN นั้นเองเพราะทั้ง 2 โมเดลนี้มีสถาปัตยกรรม แบบ SSD เหมือนกัน เพื่อให้สามารถเปรียบเทียบความสามารถให้เท่าเทียมที่สุดนั่นเอง



รูปที่ 4.3 การดูเข้าของ Classification loss โมเดล SSD ResNet101 V1 FPN



รูปที่ 4.4 การลู่เข้าของ Localization loss โมเดล SSD ResNet101 V1 FPN

จากรูปที่ 4.3 และ 4.4 แสดงถึงความสามารถในการเรียนรู้โมเดล SSD ResNet101 V1 FPN ที่ข้อมูลการสูญหาย Loss ทั้ง classification แสดงถึงคุณลักษณะการจำแนกคลาสต่างๆ นั้นถูกต้องหรือไม่ และ localization แสดงถึงคุณลักษณะการทำนายตำแหน่งของคลาสที่ได้สอนไว้

จากการทดลองการเรียนรู้โมเดล SSD MobileNet V2 FPN พบว่า ค่า Total loss ถึงเป็นการคำนวณจาก classification loss และ localization loss มีค่าอยู่ที่ 42 % เมื่อนำโมเดลมาวัดกับประสิทธิภาพความแม่นยำ Mean Average Precision (mAP) ที่ Intersect of Union (IoU) ที่ ช่วง 0.5 จนถึง 0.9 อยู่ที่ 0.63

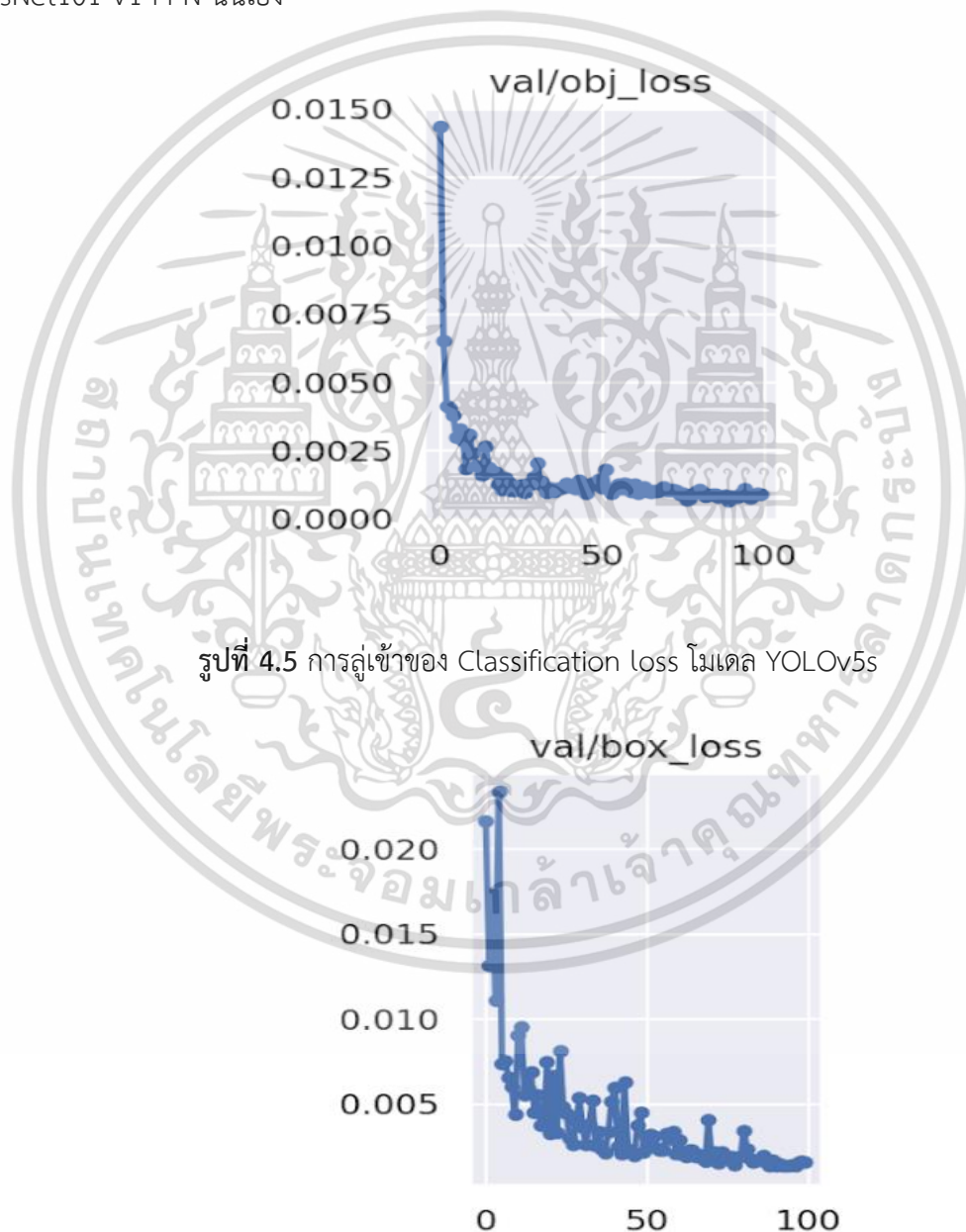
จากการทดลองสถาปัตยกรรม SSD ของทั้ง 2 โมเดล พบว่าโมเดล SSD MobileNet V1 FPN ให้ค่าความแม่นยำ mAP มากกว่าโมเดล SSD ResNet101 V1 FPN เล็กน้อยแม้ว่าเมื่อเปรียบเทียบกับฐานข้อมูล COCO ชุดฐานข้อมูล แล้วโมเดล SSD ResNet101 V1-FPN ซึ่งมีจำนวนชั้นคอนโวลูชันมากกว่าโมเดล SSD MobileNet V1 FPN เนื่องจากจำนวนรอบในการเทรนนั้น จากรูปภาพทั้ง 4.3 และ 4.4 นั้นแสดงให้เห็นว่าข้อมูลยังสามารถถูกเทรนให้ลู่เข้าได้อีก

4.2.3 YOLOv5s

โมเดล YOLOv5s เป็นโมเดลที่มีสถาปัตยกรรมเฉพาะตัวของ YOLO เองมีคุณสมบัติเป็นโมเดลที่มีความนิยมใช้งานอย่างมากในปัจจุบัน และยังมีการพัฒนาอย่างต่อเนื่อง ปัจจุบัน YOLO ได้ออกรุ่นสถาปัตยกรรมใหม่ล่าสุด ชื่อ YOLOv5 ซึ่งมีโมเดลย่อยอีกมากขึ้นอยู่กับขนาดและเวลาในการเทรนข้อมูล เมื่อเปรียบเทียบกับกับตัว ชุดฐานข้อมูลมาตรฐาน COCO ชุดฐานข้อมูล อยู่ที่ 37.4 %

ในการทดลอง หลังจากขั้นตอน Data preprocessing ภาพขนาด 640 x 640 พิกเซล จะถูกอัปโหลดขึ้นบนคลาวด์ ของแพลตฟอร์มโมเดล YOLO เองทั้งชุดสอน และชุดทดสอบ โดยผลการเรียนรู้จะแสดงออกมาในแพลตฟอร์ม TensorFlow ในรูปแสดงที่ 4.5 และ 4.6 ตามลำดับ

ในการทดลอง พารามิเตอร์ที่สำหรับการทดลอง โดย input ชุดฐานข้อมูล จะรับรูปที่ความกว้าง 640 พิกเซล สูง 640 พิกเซล จำนวนรอบในการเรียนรู้อยู่ที่ 100 epoch ซึ่งเพียงพอต่อการเรียนรู้ข้อมูลชุดฝึก และมีอัตราการเรียนรู้อยู่ที่ 0.05 ซึ่งเป็นการตั้งค่าเดียวกับทั้ง SSD MobileNet V2 FPN และ SSD ResNet101 V1 FPN นั่นเอง



รูปที่ 4.5 การดูเข้าของ Classification loss โมเดล YOLOv5s

รูปที่ 4.6 การดูเข้าของ Localization loss โมเดล YOLOv5s

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

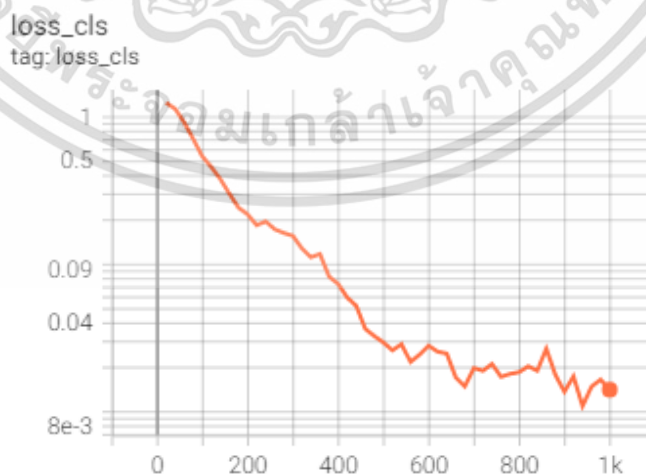
จากรูปที่ 4.5 และ 4.6 แสดงถึงความสามารถในการเรียนรู้โมเดล YOLOv5s ที่ข้อมูลการสูญหาย Loss ทั้ง classification แสดงถึงคุณลักษณะการจำแนกคลาสต่างๆ นั้นถูกต้องหรือไม่ และ localization แสดงถึงคุณลักษณะการทำนายตำแหน่งของคลาสที่ได้สอนไว้

จากการทดลองการเรียนรู้โมเดล YOLOv5s FPN พบว่า ค่า Total loss ถึงเป็นการคำนวณจาก classification loss และ localization loss มีค่าอยู่ที่ 2 % เมื่อนำโมเดลมาวัดประสิทธิภาพความแม่นยำ Mean Average Precision (mAP) ที่ Intersect of Union (IoU) ที่ ช่วง 0.5 จนถึง 0.9 อยู่ที่ 0.68

4.2.4 Faster-RCNN X101 FPN

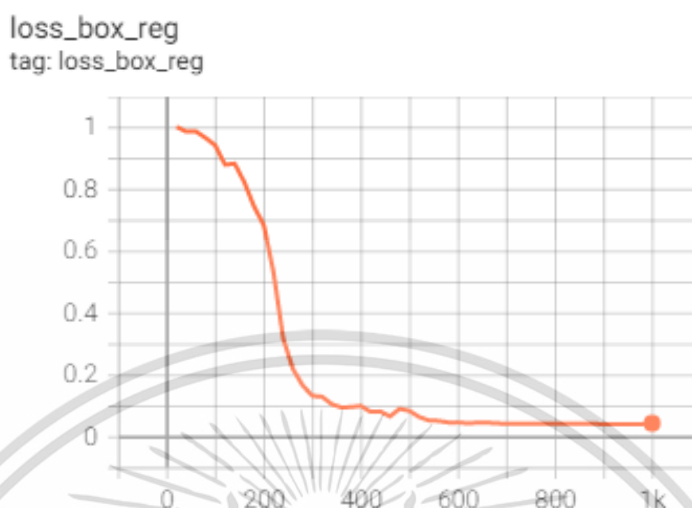
โมเดล Faster-RCNN X101 FPN เป็นโมเดลที่มีสถาปัตยกรรมเฉพาะตัวในการทำ Mask ข้อมูล ซึ่งในที่นี้จะเรียกว่า Mask-RCNN เองมีคุณสมบัติเป็นโมเดลที่มีความนิยมในงานตรวจจับที่ลักษณะพิเศษในการซ้อนทับของภาพในปัจจุบัน และยังมีพัฒนาอย่างต่อเนื่อง ปัจจุบัน โมเดลตรวจจับวัตถุแบบ Mask-RCNN ได้ถูกพัฒนาให้ใช้งานง่ายยิ่งขึ้นผ่านแพลตฟอร์มที่มีชื่อว่า Detectron ที่ถูกพัฒนาจากทีมนักวิจัยของ Facebook เอง ตัวโมเดลเมื่อเปรียบเทียบกับกับตัว ชุดฐานข้อมูลมาตรฐาน COCO ชุดฐานข้อมูลอยู่ที่ 43.0 %

ในการทดลอง หลังจากขั้นตอน Data preprocessing ภาพขนาด 640 x 640 พิกเซล จะถูกอัปโหลดขึ้นบนคลาวด์ ของแพลตฟอร์ม Detectron ของโมเดล Faster-RCNN X101 FPN เองทั้งชุดสอนและชุดทดสอบ โดยผลการเรียนรู้จะแสดงออกในมาแพลตฟอร์ม TensorFlow ในรูปแสดงที่ 4.7 และ 4.8 ตามลำดับ ในการทดลอง พารามิเตอร์ที่สำหรับการทดลอง โดย input ชุดฐานข้อมูล จะรับรูปที่ความกว้าง 640 พิกเซล สูง 640 พิกเซล จำนวนรอบในการเรียนรู้อยู่ที่ 1000 steps ซึ่งเพียงพอต่อการเรียนรู้ข้อมูลชุดฝึก และมีอัตราการเรียนรู้อยู่ที่ 0.05 ซึ่งเป็นการตั้งค่าเดียวกับทั้ง SSD MobileNet V1 FPN, SSD ResNet101 V1 FPN และ YOLOv5s นั้นเอง



รูปที่ 4.7 การลู่เข้าของ Classification loss โมเดล Faster-RCNN X101 FPN

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 การลู่เข้าของ Localization loss โมเดล Faster-RCNN X101 FPB

จากรูปที่ 4.7 และ 4.8 แสดงถึงความสามารถในการเรียนรู้โมเดล Faster-RCNN X101 FPB ที่ข้อมูลการสูญหาย Loss ทั้ง classification แสดงถึงคุณลักษณะการจำแนกคลาสต่างๆ นั้นถูกต้องหรือไม่ และ localization แสดงถึงคุณลักษณะการทำนายตำแหน่งของคลาสที่ได้สอนไว้

จากการทดลองการเรียนรู้โมเดล Faster-RCNN X101 FPB พบว่า ค่า Total loss ถึงเป็นการคำนวณจาก classification loss และ localization loss มีค่าอยู่ที่ 16 % เมื่อนำโมเดลมาวัดประสิทธิภาพความแม่นยำ Mean Average Precision (mAP) ที่ Intersect of Union (IoU) ที่ ช่วง 0.5 จนถึง 0.9 อยู่ที่ 0.50

4.3 การเปรียบเทียบและวิเคราะห์ผลการทดลอง

ในส่วนนี้จะกล่าวถึงการแสดงเปรียบเทียบผลการทดลองที่ได้จากโมเดลที่ผ่านการเรียนรู้ต่างๆ ที่ได้ทดลอง โดยในส่วนแรกจะแสดงผลการเปรียบเทียบประสิทธิภาพความแม่นยำของการประเมินโมเดล หลังจากฝึกการเรียนรู้ในตารางที่ 4.2 ในส่วนที่สองจะกล่าวถึงเวลาที่ใช้ในการทดลองของแต่ละโมเดล ซึ่งสามารถแสดงได้ดังตารางที่ 4.3

ตารางที่ 4.2 เปรียบเทียบประสิทธิภาพความแม่นยำในการประเมินโมเดล

| Pre-trained model | Total loss | mAP@IoU = 0.5 | mAP@IoU = 0.5:0.9 | AR@IoU =0.5:0.9 | Approx. Inference Compute. time/ image (ms) |
|-----------------------|-------------|---------------|-------------------|-----------------|---|
| SSD MobileNet V1 FPN | 0.90 | 0.86 | 0.65 | 0.72 | 4 |
| SSD RestNet101 V1 FPN | 0.42 | 0.80 | 0.63 | 0.75 | 4 |
| YOLOv5s | 0.02 | 0.94 | 0.68 | 0.88 | 1 |
| Faster -RCNN X101 FPB | 0.16 | 0.50 | 0.50 | 0.50 | 14 |

ตารางที่ 4.3 เปรียบเทียบประสิทธิภาพความเร็วในการฝึกข้อมูล

| Pre-trained model | Training time (hrs.) |
|-----------------------|----------------------|
| SSD MobileNet V1 FPN | 1.34 |
| SSD RestNet101 V1 FPN | 2.3 |
| YOLOv5s | 0.4 |
| Faster -RCNN X101 FPB | 1 |

จากตารางที่ 4.2 โมเดลที่ผ่านการเรียนรู้ YOLOv5s สามารถฝึกการเรียนรู้และประเมินประสิทธิภาพของโมเดลที่ผ่านการฝึก IoU ที่ 0.5 ได้ประสิทธิภาพความแม่นยำ mAP ที่สุดคือ 0.94 แต่ถ้าพิจารณา IoU ที่ช่วง 0.5 ถึง 0.9 จะพบว่า โมเดลที่ผ่านการเรียนรู้ YOLOv5s นั้นให้ค่าประสิทธิภาพความแม่นยำ mAP มากที่สุดคือ 0.68 และเมื่อพิจารณาค่าความถูกต้องหรือ Average Recall แล้วนั้น YOLOv5s ให้ค่าความเฉลี่ยถูกต้องได้มากที่สุดคือ 0.88 และยังใช้ในการประยุกต์โมเดลร่วมกับภาพเร็วที่สุดคือ 1 มิลลิวินาทีต่อภาพนั่นเอง

จากตารางที่ 4.3 แสดงความสามารถในการใช้เวลาการฝึกชุดข้อมูลทดสอบกับฐานข้อมูล 400 ภาพพบว่าโมเดลที่ผ่านการเรียนรู้ YOLOv5s ยังสามารถใช้เวลาการฝึกน้อยที่สุด คือ 0.4 ชั่วโมง หรือประมาณ 25 นาที รองลงมาอันดับที่สองเป็นคือโมเดล Faster -RCNN X101 FPB ที่ 1 ชั่วโมง อันดับสามโมเดล SSD MobileNet V1 FPN ที่ 1.34 ชั่วโมง หรือ 80 นาที อันดับสุดท้ายคือโมเดล SSD RestNet101 V1 FPN ที่ 2.3 ชั่วโมง หรือ 140 นาที ตามลำดับ

พิจารณาข้อมูลจากทั้ง 2 ตารางพบว่า การใช้โมเดลที่ผ่านการเรียนรู้มาแล้ว YOLOv5s ให้ค่าประสิทธิภาพความแม่นยำมากที่สุด ที่ IoU ระหว่าง 0.5 ถึง 0.9 แต่ถ้าเปรียบเทียบรูปแบบการสอนข้อมูล ซึ่งโมเดล SSD MobileNet V2 FPN, SSD ResNet101 V1 FPN, YOLOv5s เป็นการสอนข้อมูลแบบตีกรอบ หรือ Box ในขณะที่โมเดล Faster-RCNN X101 FPN เป็นการสอนข้อมูล แบบตีกรอบรอบวัตถุ หรือ ตีกรอบรอบวัตถุ(Polygon) ทำให้โมเดล Faster-RCNN X101 FPN ซึ่งเป็นการทำ Mask-RCNN ข้อมูลยังมีความโดดเด่นในเรื่องของการทำตรวจจับวัตถุ Object detection

4.4 การประยุกต์โมเดลร่วมกับการประมวลผลภาพ

ในหัวข้อนี้จะนำโมเดลที่ได้จากฝึกชุดข้อมูลนำมาแสดงร่วมกับวิทยาการประมวลผลภาพ(Image processing) เพื่อดูผลการทำนายของโมเดล ความน่าจะเป็นของวัตถุที่ถูกสอนเพื่อนำไปประยุกต์ใช้งานมากขึ้นในอนาคต รูปที่ประยุกต์เข้ากับการประมวลผลภาพแสดงดังในรูปที่ 4.9





รูปที่ 4.9 การประยุกต์ร่วมกับการประมวลผลภาพ

จากรูปที่ 4.9 แสดงการประยุกต์ร่วมกับการประมวลผลภาพ Image processing โดยในการฝึกชุดข้อมูล ผลลัพธ์ที่ได้จะเป็นไฟล์ weight ซึ่งจะนำไปเข้าสู่กระบวนการประมวลผลภาพ

ในการทดลอง จะใช้ library ของ OpenCV ซึ่งมีความโดดเด่นในเรื่องงานประมวลผลภาพต่างๆ ซึ่งในกาทำงานโปรแกรมคำสั่งจะประมวลผลเป็นภาพ เพราะฉะนั้นไฟล์วิดีโอที่นำมาจะถูกประมวลผลภาพทีละเฟรม ซึ่งในวิทยานิพนธ์นี้ ไฟล์วิดีโอที่นำมาเป็นฐานข้อมูลถูกเป็นบันทึกมาที่ 60 เฟรมต่อวินาที (FPS)

บทที่ 5

สรุปงานวิจัยและข้อเสนอแนะ

ในบทที่ 5 นี้จะกล่าวถึงการสรุปเนื้อหาของวิทยานิพนธ์ และสรุปผลการอภิปรายการทดลอง รวมทั้งข้อเสนอแนะและแนวทางการพัฒนาโมเดลการตรวจจับพฤติกรรมการติดสัดของวัวผ่านการเรียนรู้เชิงลึก

5.1 สรุปเนื้อหาวิทยานิพนธ์

วิทยานิพนธ์นี้นำเสนอการวิเคราะห์พฤติกรรมของวัวขณะมีอาการสืบพันธุ์ตามฤดูกาลโดยการประมวลผลทางด้วยปัญญาประดิษฐ์แบบการเรียนรู้เชิงลึก มีเป้าหมายที่จะพัฒนางานวิจัยด้านเทคโนโลยีปศุสัตว์ของประเทศไทย โดยการใช้ปัญญาประดิษฐ์ในการวิเคราะห์ตรวจจับพฤติกรรมของวัวในช่วงการติดสัด เนื่องจากในปัจจุบันในหลายพื้นที่ หลายฟาร์มที่เลี้ยงวัวเพื่อเลี้ยงชีพยังคงใช้การจ้างคนเค้าเข้าเฝ้าติดตามพฤติกรรม เพื่อเป็นการเพิ่มประสิทธิภาพการเลี้ยง เพิ่มผลผลิตทางปศุสัตว์ต่อไป

ในงานวิทยานิพนธ์นี้ได้รับความอนุเคราะห์ภาพวิดีโอพฤติกรรมการติดสัดของวัวมา 10 ไฟล์วิดีโอในสกุลไฟล์ (.mp4) เพื่อมาเข้าสู่กระบวนการ Data preprocessing โดยได้มีการดึงภาพจากในวิดีโอทั้งนำมาเป็นชุดฐานข้อมูล เพื่อแก้ปัญหาข้อมูล Overfitting เทคนิคการทำ data augmentation เป็นการเพิ่มข้อมูลไปในชุดฐานข้อมูล เพื่อเพิ่มประสิทธิภาพในการฝึกโมเดล ฐานข้อมูลสำหรับการทำโมเดลมีทั้งหมด 500 รูป แบ่งเป็น ชุดข้อมูลในการฝึก 425 รูป ชุดข้อมูลทดสอบ 75 รูปในอริยบทต่างๆ ด้วยมีการแบ่งคลาสข้อมูลออกเป็น 2 คลาส คือ 1.) ชุดภาพวัวในพฤติกรรมปกติ หรือ Cow_normal 2.) ชุดภาพวัวที่มีพฤติกรรมการติดสัดและมีการกระโดดคร่อมวัวตัวอื่น หรือ Cow_jump

สำหรับการฝึกการเรียนรู้ผ่านคอนโวลูชัน จะใช้เทคนิคการถ่ายโอนการเรียนรู้ Transfer Learning สำหรับโมเดลที่ผ่านการเรียนรู้มาแล้วในงานวิทยานิพนธ์นี้ใช้โมเดล SSD Mobile Net V2 FPN, SSD ResNet101 V1 FPN, YOLOv5s และ Faster-RCNN X101 FPN ตามลำดับ

ในการทดลองงานวิทยานิพนธ์นี้เสนอวิธีประเมินประสิทธิภาพความแม่นยำของโมเดลที่ผ่านการฝึกข้อมูลโดยใช้ ค่าเฉลี่ยความแม่นยำหรือ Mean average precision (mAP) ในการทดลองจะนำข้อมูลชุดฝึกนำเข้าโมเดลที่ผ่านที่เรียนที่ได้กล่าวไปแล้วในย่อหน้าก่อนหน้า เพื่อทดลองหาค่าเฉลี่ยความแม่นยำและเวลาในการฝึกชุดข้อมูล โดยในการทดลองจะใช้ ความแม่นยำในการทำนาย Intersect of union (IoU) ที่ 0.5 เป็นตัววัดประเมินประสิทธิภาพการใช้งาน

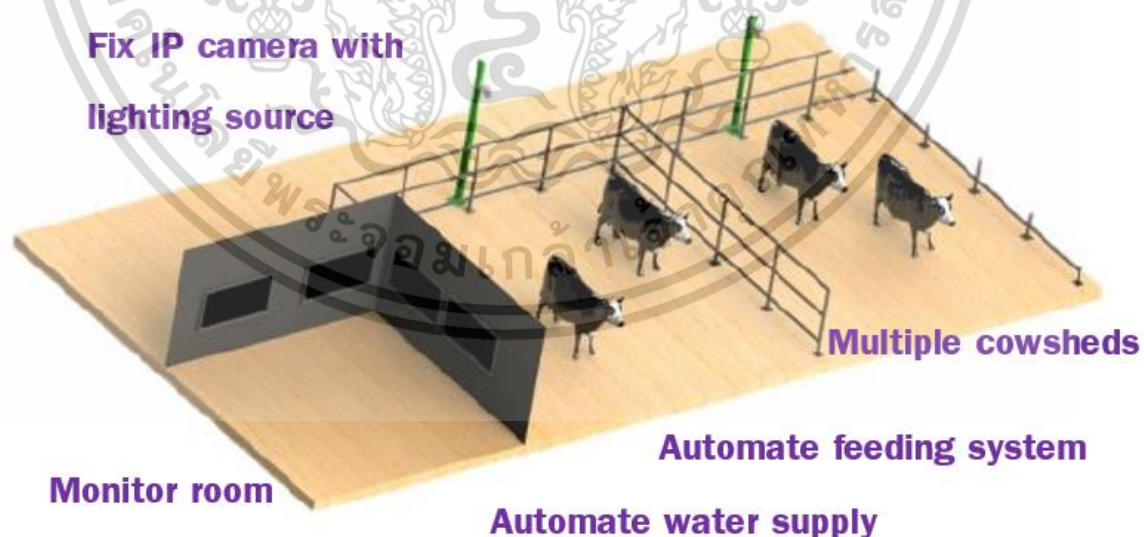
หลังจากการทดลองพบว่าโมเดลที่ผ่านการเรียนรู้ YOLOv5s ให้ค่าเฉลี่ยความแม่นยำได้สูงที่สุดคือ 94.7% และใช้เวลาในการฝึกชุดข้อมูลเร็วที่สุดคือ 0.4 ชั่วโมง อันดับสองคือโมเดล SSD ResNet101 V1 FPN ที่ 80.7% อันดับสามคือ Faster -RCNN X101 FPN ที่ 73.8 % และอันดับสุดท้ายคือ SSD MobileNet V1 FPN ที่ 65.0%

เมื่อได้โมเดลที่จากการฝึกจะได้ไฟล์ weight ซึ่งจะถูกนำไปใช้ในการประมวลผลภาพในไฟล์วิดีโอทดสอบ หลังจากนั้นมาแสดงผลแล้วพบว่าสามารถทำนายพฤติกรรมการติดตามตามฤดูการของวัวได้แม่นยำระดับหนึ่ง ทำให้โมเดลที่ฝึกจาก YOLOv5s สามารถนำไปประยุกต์การใช้งานต่อไปในอนาคตต่อไปได้

5.2 ข้อเสนอแนะและแนวทางในการพัฒนาต่อ

ในวิทยานิพนธ์นี้พบปัญหาที่พบเจอคือ จำนวนของข้อมูลค่อนข้างน้อยเนื่องจากการทดลองเป็นแบบธรรมชาติเปิด ไม่ได้มีการกำหนดการทดลองที่เป็นคอกที่ชัดเจนทำให้ข้อมูลที่ได้อาจไม่สม่ำเสมอหรือคุณภาพที่ดีพอที่จะนำไปทำเป็นสเกลงานที่ใหญ่ได้ เพราะฉะนั้นข้อมูลที่ได้อาจค่อนข้างสำคัญที่ต้องมีการจัดเก็บให้ดีกว่านี้

สำหรับแนวทางในการพัฒนาต่อ นั้น จะเป็นการจัดเตรียมการทดลองในสภาวะปิด ในฟาร์มเลี้ยงแบบปิดที่ได้มาตรฐานดังแสดงในรูปที่ 5.1 ต่อไปนี้



รูปที่ 5.1 แนวทางพัฒนาต่อไปในอนาคต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 5.1 แสดงถึงฟาร์มเลี้ยงแบบปิดที่ได้มาตรฐาน มีคอกเลี้ยงแบบกึ่งลักษณะโดยการทดลองจะ
เป็นการติดตั้งกล้อง IP camera เพื่อเฝ้าดูและเป็นการเก็บชุดข้อมูลไปในตัว โดยจะมีห้องมอนิเตอร์ที่ไว้
สำหรับจัดการข้อมูลต่างๆ รวมถึงการทำโมเดลฝึกโมเดลขนาดใหญ่ต่อไปในอนาคต เพื่อเป็นการเพิ่ม
ประสิทธิภาพการเลี้ยงดู ลดการสัมผัสระหว่างคนกับสัตว์ เพิ่มอัตราการให้กำเนิดของลูกวัวต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

- [1] Chen Y., He D., Song H., 2018. “Automatic monitoring method of cow Ruminant behavior based on spatio temporal context learning” **International Journal of Agriculture and Biological Engineering**. 179 -185.
- [2] Norring, M., Valros, A., 2016. “The effect of lying motivation on cow behavior.” **Applied Animal Behaviour Science**. pp.1-5.
- [3] Felzenszwalb, P. F., Huttenlocher D. P., 2005. “Pictorial Structures for Object Recognition.” **International Journal of Computer Vision**, vol. 61(1): 55-79.
- [4] Ren, X., Berg, A. C., Malik, J., 2005. “Recovering human body configurations using pairwise constraints between parts.” **International conference on computer Vision**, pp. 824-831.
- [5] Andriluka, M., Pishchulin, L., Gehler, P. V., Schiele, B., 2014. “2D Human Pose Estimation: New Benchmark and State of the Art Analysis.” **Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition**, pp. 3686-3693.
- [6] Toshev, A., Szegedy, C., 2014. “DeepPose: Human Pose Estimate via Deep Neural Networks.” **IEEE Conference on Computer Vision and Pattern Recognition**, pp. 1653-1660.
- [7] Tompson, J., Jain, A., Lecun, Y., Bregler, C., 2014. “Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimate.” In **Neural Information Processing Systems**, pp. 1799- 1807.
- [8] C.Cai. R.Zjang. L.Ju and J.He, 2019. "Deep cascaded convoultional models for cattle pose estimate," in **Computers and Electronics in Agriculture**, vol.164.
- [9] Wei, S., Ramakrishna, V., Kanade, T., Sheikh, Y., 2016. “Convolutional Pose Machines.” **IEEE Conference on Computer Vision and Pattern Recognition**, pp. 4724-4732.

- [10] Newell, A., Yang, K., Deng, J., 2016. “Stacked Hourglass Networks for Human pose estimate.” In **European Conference on Computer Vision**, pp. 483-499.
- [11] Bulat, A., Tzimiropoulos, G., 2016. “Human Pose Estimate via Convolutional Part Heatmap Regression”
- [12] Lyu, S., Chang, M.-C., Du, D., Li, W., Wei, Y., Del Coco, M., et al. (2018). “UA-DETRAC 2018: Report of AVSS2018 & IWT4s challenge on advanced traffic monitoring.” **IEEE international conference on advanced video and signal based surveillance**, pp. 1–6.
- [13] Norouzzadeh, M. S., Nguyen, A., Kosmala, M., Swanson, A., Palmer, M. S., Packer, C., et al. (2018). Automatically identifying, counting, and describing wild animals in camera-trap images with deep learning. **Proceedings of the National Academy of Sciences**, 115(25), E5716–E5725.
- [14] Thanawat, L., Kitsuchart, P., Chu, L., 2022. “CowXNet: An automated cow estrus detection system.” **Expert Systems with Applications**, Vol.211
- [15] Bochkovskiy, A., Wang, C.-Y., & Liao, H.-Y. M. (2020). YOLOv4: Optimal speed and accuracy of object detection. **arXiv:2004.10934**
- [16] Pauekpong, S., Chivapreecha, S., Methasate, I., 2015. “An Implementation of Face Recognition System Applications.” **2015 International Symposium on Multimedia and Communication Technology**. 306-309
- [17] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, “Focal loss for dense object detection,” In **Proceedings of the IEEE international conference on computer vision**, pp.2980-2988, 2017.
- [18] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, and H. Adam, “Mobilenets: Efficient convolutional neural networks for mobile vision applications,” **arXiv:1704.04861**, 2017.
- [19] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” **Proceedings of the IEEE conference on computer**
- [20] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” **Proceedings of the IEEE conference on computer vision and pattern recognition**, pp.770-778, 2016.

- [21] U. Christina and W. Anthony, “An automated sensor-based method of simple behavioural classification of sheep in extensive systems” **Computers and Electronics in Agriculture**, vol.64,pp.19–26,May 2008.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



2022 International Conference on Power, Energy and Innovations (ICPEI 2022)

October 19 - 21, 2022
Pattaya, Thailand

Abstract Book



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The cattle rut behavior detection base on AI deep neural network

Phatsarut Phaechaiyaphum
School of Engineering,
King Mongkut's Institute of
Technology Ladkrabang,
Bangkok, 10520, Thailand
63601036@kmitl.ac.th

Poom Konghuayrob
School of Engineering,
King Mongkut's Institute of
Technology Ladkrabang,
Bangkok, 10520, Thailand
poom.ko@kmitl.ac.th

Abstract

This paper proposes an application model of camera object detection to observe the cattle rut through its pose. This research is based on deep neural network method along with transfer learning technique. This paper employs four different deep convolutional neural network architectures 1) SSD Mobile-Net V2 2) SSD ResNet101 V1 FPN 3) YOLOv5s and 4) Mask R-CNN. All the network mentioned above are pre-trained on COCO dataset. To observe the cattle rut behavior, the images of cattle behavior were taken in natural open environment, the data obtained are split into training and testing datasets. In this research cattle behaviors are categorized mainly in two classes, (1) normal behavior and (2) the cattle rut behavior. The result obtained after training the models shows that the YOLOv5s model obtained the highest mean average precision which is 95.5% and least training time. Thus, this paper proposes that YOLOv5s model can be applied to detect the cattle behavior for precise artificial breeding in order to increase the cattle population in farm to serve higher consumption demand in the near future.

Keywords—Deep neural network, Object detection, Pose, Mask r-cnn, pre-trained model.

I. INTRODUCTION

To raise the population of cattle, artificial breeding is very common. The first process is to observe the behavior of cattle, where the rut behavior of the cattle with each other is detected. Research from Pennsylvania State University, USA., found that the female cow showed its rut behavior every 21 days. The duration of rut is noticed to be around 18-24 hours, after that the cattle starts to ovulate for around 14 hours. So, the suitable time for artificial breeding is before it ovulates. But sometimes agriculturalists use some tricks like, whenever some cattle show its rut for example, male cattle standing over female cattle, likes breeding after 12 hours. So, this period is suitable for artificial breeding. However, artificial breeding is a common method and widely used. To observe the cattle ruts daily is a labor-intensive work.

This work proposes the method or application to detect the cattle rut by using object detection utilizing pre-trained deep convolutional neural network models which are self-benchmark from COCO datasets. To

detect the specific object outside of COCO dataset classes, the process needs to prepare enormous datasets and spend a long time for training the network. This is the reason why transfer learning is applied to pre-trained models and save time and can use less data.

To obtain evaluation of transfer learning models, the mean average precision (mAP) is used as a training loss for all models. SSD Mobile-Net V2, SSD ResNet101 V1 FPN, YOLOv5s and Mask-RCNN. These models classify the data into two class: 1) normal cow and 2) the cow rut. Then, inference a trained model with video files to observe cattle rut.

II. RELATED WORK

A. Object detection model and feature extraction

Liu and team [1] introduced SSD architecture in 2016s for faster object detection with high average precision in figure 1. SSD or Single Shot Detector used for object detection. SSD consist of 2 parts which is (1) feature extraction and (2) convolution filters using for object detection. To compute difference between ground truth box and the predicted box, localization loss is uses for finalizing loss function by equation (1):

$$(1) \quad L(x, c, l, g) = \frac{1}{N} (L_{conf}(x, c) + \alpha L_{loc}(x, l, g))$$

Where N is the of positive matches amount and α is the weight for the localization loss.

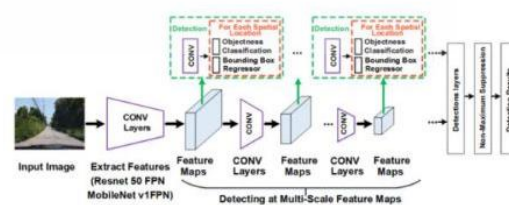


Fig. 1. SSD model architecture

B. MobileNet V1 FPN

Howard and developer team [2] from Google proposed MobileNet v1 which compute reduction and less convolutional layers. Moreover, FPN[3] use for reduced size and enlarge images size for feature extract. The pre-trained model on the COCO 2017 dataset has average precision up to 29.1 %.

C. RestNet101 V1 FPN

He and developer team [4] from Microsoft Research proposed RestNet101 which there are 101 layers for feature extraction. The pre-trained model on the COCO 2017 dataset has average precision up to 35.6 % [6].

D. YOLO object detection model

YOLO is an real time object detection algorithm which uses neural network. YOLO is widely use because speed and accuracy from training. YOLO algorithm use convolutional neural networks (CNN) to detect objects in real-time [5]. The pre-trained model from YOLOv5s average precision is up to 56.8%

E. Mask-RCNN

Mask R-CNN developed from Faster R-CNN to solve instance segmentation tasks. Mask bounding box use to recognize objects for good result training. The pre-trained model average precision (AP) for Mask R-CNN through Detectron2 framework is 43.0% which trained on COCO dataset

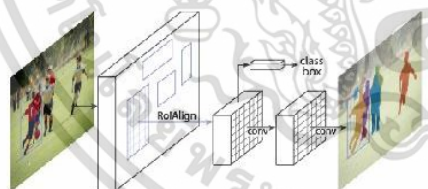


Fig.2. The Mask R-CNN framework for instance segmentation.

III. RESEARCH METHODOLOGY

A. Data preparation

Datasets of cow at northeast of Thailand. Supported and recorded freely by unspecified smartphone. The video files recorded are in mp4 format at 30 frames per second (fps). The videos consist of 2 main categories for observing the cattle rut behavior in total of 10 video files:

1) *Cow jump*: This class name signifies that the cow has the rut behaviour. This can be observed

when the bull and cow interact or the bull jump over the cow as shown in figure (3).



Fig.3. the video filed recorded the cattle rut behavior.

2) *Cow normal*: This class name is classified as normally behavior and consists of the cattle stand, walk, eat, lay down etc. behaviour as shown in (4).



Fig.4. the video filed recorded the normally cattle behavior.

B. Data preprocessing

The datasets in video format uploaded to Roboflow, a computer vision developer framework for data collection to processing. The images were cropped from videos and the rectangular bounding box is applied for SSD MobileNet V1 FPN, SSD RestNet101 V1 FPN, YOLOv5s and polygons for Mask-RCNN. The Datasets of cattle rut behavior consists of 500 images (including augmented images) with a size of 640 x 640 pixels and distributed over the class i.e., Cow jump and Cow normal. Then, the dataset is randomly spitted into training, valid and testing i.e., 80% training, 5% validation and 15% testing. The total number of images in the training set is 400 images, validation is 25 images and testing are 75 images.



Fig.5. Datasets for 1) SSD MobileNet V1 FPN 2) SSD RestNet101 V1 FPN 3) YOLOv5s



Fig.6. Datasets for 4) Mask-RCNN

C. Pre-trained model

A pre-trained model is a network that was trained with a large-scale standard dataset for use in image classification task. For this research the pre-trained models are selected based on their performance on the COCO datasets. The models are validated using the mean average precision (mAP) and the execution time, shown in the table 1.:

TABLE 1. Selected pre-trained model in this research.

| Pre-trained model | Labelling | COCO test mAP(%) |
|-----------------------|-------------|------------------|
| SSD MobileNet V1 FPN | Rectangular | 29.1 |
| SSD RestNet101 V1 FPN | Rectangular | 35.6 |
| YOLOv5s | Rectangular | 37.4 |
| Faster -RCNN X101 FPN | Polygon | 43.0 |

D. Experimentation:

The experiment was performed on the GPU Tesla P100 provided in the free version of Google Collaboratory (Colab). The hyperparameter used during the experimentation are stored in a config file by 1) batch size:16, 2) Number of epochs:100, 3) Initial learning rate:0.0005.

E. Model evaluation

In this research, to evaluate models with IoU, average precision (AP) or mean average precision (mAP)

a) *Intersection Over Union (IoU)*: Defines the overlap between the predicted and ground truth bounding box i.e. generally among 0 and 1. If IoU equals 0, it means no overlap. IoU equals 1 means that the boxes is completely overlap together.

IoU is use for measure how accuracy when tracking annotations.



Fig.7. Intersect over union (IoU)

2) *Average precision (AP)*: Collection of accurate object detections from a model when compared to ground-truth object annotations of a dataset by (2).

$$(2) \quad AP = \sum_{i=1}^{n-1} (r_{i+1} - r_i) p_{interp}(r_{i+1})$$

Where r is Recall of first interpolate n is overall images and pinterp is a new view from one or more RADIANCE images. Precision is how efficiency of a model in identifying only relevant object by (3).

$$(3) \quad p = \frac{TP}{TP + FP}$$

TP: True positive and FP and FP: False positive.

3) *Mean average precision (mAP)*: is used to determine average overall class AP by (5).

$$(5) \quad mAP = \frac{\sum_{i=1}^K AP_i}{K}$$

K: Number total of class

AP_i: Average precision at class number i

IV. RESULT

The results shown in table 2:

TABLE 2. Model precision.

| Pre-trained model | Training time (hrs.) | mAP@IoU=0.5 | mAP@IoU=0.5:0.9 | mAP@IoU=0.75 |
|-----------------------|----------------------|-------------|-----------------|--------------|
| SSD MobileNet V1 FPN | 1.34 | 65.0 | 87.0 | 72.9 |
| SSD RestNet101 V1 FPN | 2.3 | 80.7 | 63.0 | 73.0 |
| YOLOv5s | 0.4 | 94.7 | 68.1 | 81.4 |
| Faster -RCNN X101 FPB | 1 | 73.8 | 57.3 | 66.1 |

YOLOv5s is the fastest among all the models during training time which was observed to be 0.4 hrs. whereas Faster -RCNN X101 FPB, SSD MobileNet V1 FPN, SSD RestNet101 V1 FPN observed training time was 0.5 hrs., 1.34 hrs., 2.3 hrs. respectively. Mask-RCNN: Fast -RCNN X101 FPN has the higher mAP compared to YOLOv5s as shown in the table 1 but after the experimentation the results shows that YOLOv5s gives higher mAP. Although YOLOv5s uses rectangular box for labelling the dataset. But, Faster -RCNN X101 FPN uses polygon on the images to label the datasets.

In this research the experiment was in open environment. So, the cattle behavior was uncontrollable, the overlapping of the cow could happen. Despite the challenges YOLOv5s is selected

as a suitable model for this research and has also provided the better result as shown in fig 8.



YOLOv5s
Fig.8. Results of transfer learning from YOLOv5s

V. CONCLUSION

This research proposes applying a pre-trained deep convolutional neural network model trained on large scale COCO dataset. The mean average precision(mAP) is used as a validation criterion. In this experiment pre-trained model: SSD MobileNet V1 FPN, SSD ResNet101 V1 FPN, YOLOv5s and Faster-RCNN X101 FPN were selected since they have high mAP and speed of computation.

After the experimentation, YOLOv5s has shown the highest mAP i.e 94.7%, the second one is SSD ResNet101 V1 FPN which has obtained 80.7%. Inferring validation set, It is found that YOLOv5s solved the problem well. This research could apply to other object detection problems which use pre-trained deep convolutional neural and transfer learning to solve problems.

VI. FUTURE WORK

To implement trained model to real situation. Controllable environment must be considered. Especially, light source in daylight and night will be install. IP camera will be a device for monitoring and interfering trained model with video files signal. All cameras will connect to the plant

control room where the system will locate inside. The signal of cow rut will be computed and notify to person in charge to prepare artificial breeding as soon as possible.

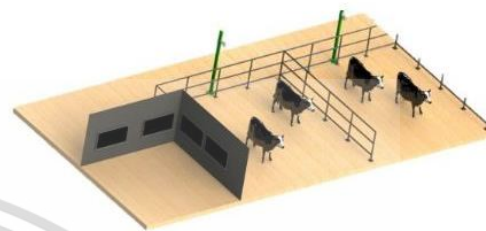


Fig.9. Designed field for real situation experiment

ACKNOWLEDGMENT

This research presented in this paper was supported and funded by Charoen Pokphand Food PCL. (CPF) and School of Engineering, King Mongkut's Institute of Technology Ladkrabang (KMUTL). Lastly, a sincere gratitude to all the farmers for advising and preparation the datasets for this research.

REFERENCES

- [1] T. Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," In Proceedings of the IEEE international conference on computer vision, pp.2980-2988, 2017.
- [2] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, and H. Adam, "Mobilenets: Efficient convolutional neural networks for mobile vision applications," arXiv:1704.04861, 2017.
- [3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer
- [4] Pkuzlc, TensorFlow detection model zoo, Available Online at https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/detection_model_zoo.md, accessed on 18 March 2019.
- [5] K.He, G.Gkioxari, P.Dollar, R.Girshick, "Mask R-CNN" in ICCV Computer Vision Foundation.,2017
- [6] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp.770-778, 2016.
- [7] C.Cai, R.Zjang, L.Ju and J.He, "Deep cascaded convolutional models for cattle pose estimation," in Computers and Electronics in Agriculture, vol.164, May 2019.
- [8] U. Christina and W. Anthony, "An automated sensor-based method of simple behavioural classification of sheep in extensive systems" Computers and Electronics in Agriculture, vol.64,pp.19-26,May 2008.



ภาคผนวก ข

โปรแกรมที่ใช้ในวิทยานิพนธ์ การติดตั้ง TensorFlow object
API, detection YOLOv5s, Detectron

TensorFlow object detection API

TensorFlow object detection API เป็นระบบตรวจจับวัตถุที่ถูกออกแบบและพัฒนาที่来解决ปัญหาการสร้าง custom object detector ที่ต้องใช้เวลาในการพัฒนา ในส่วนแรกจะกล่าว วิธีและขั้นตอนในการติดตั้ง รวมถึงการดึงนำ pre-trained ที่มีหลากหลายมาศึกษาทดลอง รวมถึงมาประมวลผลคอนโวลูชัน ในเนื้อหาจะเป็นการทดลองผ่าน Anaconda

การติดตั้ง TensorFlow object detection API

Create a new Anaconda virtual environment

```
>> conda create -n tensorflow pip python
```

Activate the Anaconda virtual environment

```
>> conda activate tensorflow
```

Tensorflow Installation

```
>> pip install --ignore-installed --upgrade tensorflow==2.2.0
```

Verify Installation

```
>> python -c "import tensorflow
as tf;print(tf.reduce_sum(tf.random.normal([1000, 1000])))"
```

Repository git of Tensorflow model garden

```
>> git clone https://github.com/tensorflow/models.git
```

Protobuf Installation/Compilation

```
# From within TensorFlow/models/research/
```

```
>> protoc object_detection/protos/*.proto --python_out=.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COCO API installation

```
>>git clone https://github.com/cocodataset/cocoapi.git
>>cd cocoapi/PythonAPI
>>make
>>cp -r pycocotools <PATH_TO_TF>/TensorFlow/models/research/
```

Install the Object Detection API

```
# From within TensorFlow/models/research/
>>cp object_detection/packages/tf2/setup.py .
>>python -m pip install .
```

Test Installation

```
# From within TensorFlow/models/research/
>>python object_detection/builders/model_builder_tf2_test.py
```

การเทรน Custom object detector

การจัดการเตรียม workspace ในโฟลเดอร์ Tensorflow ต้องประกอบไปด้วยโครงสร้างโฟลเดอร์ย่อยตามรูปที่ ข.1 ต่อไปนี้

```
TensorFlow/
├─ addons/ (Optional)
│   └─ labelImg/
└─ models/
    ├─ community/
    ├─ official/
    ├─ orbit/
    ├─ research/
    └─ ...
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ ข.1 แสดงโครงสร้างโฟลเดอร์ Tensorflow

สร้างโฟลเดอร์ย่อย workspace สำหรับจัดเก็บไฟล์ที่ได้จากเทรนเข้ามา หลังจากนั้นสร้างแฟ้มไฟล์ชื่อว่า training_demo ในโฟลเดอร์ย่อย workspace อีที หลังจากให้สร้างโฟลเดอร์ย่อยภายในตามรูปที่ ข.2 ต่อไปนี้

```
training_demo/
├── annotations/
├── images/
│   ├── test/
│   └── train/
├── models/
├── pre-trained-models/
└── README.md
```

รูปที่ ข.2 แสดงโครงสร้างโฟลเดอร์ย่อย Training_demo

หลังจากนั้นให้ดาวน์โหลดไฟล์ partition_ชุดฐานข้อมูล.py สำหรับแบ่งข้อมูลออกเป็นชุดฝึก และชุดสอน ตามลิงค์ต่อไปนี้ https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/2.2.0/_downloads/d0e545609c5f7f49f39abc7b6a38cec3/partition_ชุดฐานข้อมูล.py

จากนั้นให้ SSH ไปที่ TensorFlow/scripts/preprocessing แล้วพิมพ์คำสั่งต่อไปนี้

```
>> python partition_ชุดฐานข้อมูล.py -x -i [PATH_TO_IMAGES_FOLDER] -r 0.1
```

```
# For example
```

```
# python partition_ชุดฐานข้อมูล.py -x -i
```

```
C:/Users/sglvladi/Documents/Tensorflow/workspace/training_demo/images
```

```
-r 0.1
```

ขั้นตอนต่อไปเป็นการสร้าง Label map ข้อมูลทำให้ TensorFlow object detection API รู้ถึงชนิดของข้อมูลตามสคริปต์ดังต่อไปนี้

```

item {
  id: 1
  name: 'Cow_normal'
}

```

```

item {
  id: 2
  name: 'Cow_jump'
}

```

ต่อไปเป็นขั้นตอนการสร้าง tensorflow records โดยให้ดาวน์โหลดไฟล์สคริปต์ generate_tfrecord.py จากลิงค์ต่อไปนี้ https://tensorflow-object-detection-api-tutorial.readthedocs.io/en/2.2.0/_downloads/da4babe668a8afb093cc7776d7e630f3/generate_tfrecord.py แล้วเก็บไว้ใน TensorFlow/scripts/preprocessing จากนั้นให้ SSH ไปที่เก็บไฟล์ แล้วรันคำสั่งต่อไปนี้

Create train data:

```

python generate_tfrecord.py -x [PATH_TO_IMAGES_FOLDER]/train -l
[PATH_TO_ANNOTATIONS_FOLDER]/label_map.pbtxt -o
[PATH_TO_ANNOTATIONS_FOLDER]/train.record

```

Create test data:

```

python generate_tfrecord.py -x [PATH_TO_IMAGES_FOLDER]/test -l
[PATH_TO_ANNOTATIONS_FOLDER]/label_map.pbtxt -o
[PATH_TO_ANNOTATIONS_FOLDER]/test.record

```

For example

python generate_tfrecord.py -x

C:/Users/sglvladi/Documents/Tensorflow/workspace/training_demo/images/train -l

C:/Users/sglvladi/Documents/Tensorflow/workspace/training_demo/annotations/label_map.pbtxt -o

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
C:/Users/sglvladi/Documents/Tensorflow/workspace/training_demo/annotations/train.record
# python generate_tfrecord.py -x
C:/Users/sglvladi/Documents/Tensorflow/workspace/training_demo/images/test -l
C:/Users/sglvladi/Documents/Tensorflow2/workspace/training_demo/annotations/label_map.pbtxt -o
C:/Users/sglvladi/Documents/Tensorflow/workspace/training_demo/annotations/test.record
```

หลังจากนั้นให้ทำการดาวน์โหลดตัว pre-trained model จากลิงค์ต่อไปนี้

https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

ใน tensorflow 2 detection model zoo จะประกอบไปด้วย pre-trained model มากมาย ตามรูปที่ ข.3 ต่อไปนี้

TensorFlow 2 Detection Model Zoo

TensorFlow 2.2 Python 3.6

We provide a collection of detection models pre-trained on the [COCO 2017 dataset](#). These models can be useful for out-of-the-box inference if you are interested in categories already in those datasets. You can try it in our inference [colab](#).

They are also useful for initializing your models when training on novel datasets. You can try this out on our few-shot training [colab](#).

Please look at [this guide](#) for mobile inference.

Finally, if you would like to train these models from scratch, you can find the model configs in this [directory](#) (also in the linked [tar.gz](#)).

| Model name | Speed (ms) | COCO mAP | Outputs |
|---|------------|-----------|-----------------|
| CenterNet HourGlass104 512x512 | 70 | 41.9 | Boxes |
| CenterNet HourGlass104 Keypoints 512x512 | 76 | 40.0/61.4 | Boxes/Keypoints |
| CenterNet HourGlass104 1024x1024 | 197 | 44.5 | Boxes |
| CenterNet HourGlass104 Keypoints 1024x1024 | 211 | 42.8/64.5 | Boxes/Keypoints |
| CenterNet Resnet50 V1 FPN 512x512 | 27 | 31.2 | Boxes |
| CenterNet Resnet50 V1 FPN Keypoints 512x512 | 30 | 29.3/50.7 | Boxes/Keypoints |
| CenterNet Resnet101 V1 FPN 512x512 | 34 | 34.2 | Boxes |
| CenterNet Resnet50 V2 512x512 | 27 | 29.5 | Boxes |
| CenterNet Resnet50 V2 Keypoints 512x512 | 30 | 27.6/48.2 | Boxes/Keypoints |
| CenterNet MobileNetV2 FPN 512x512 | 6 | 23.4 | Boxes |
| CenterNet MobileNetV2 FPN Keypoints 512x512 | 6 | 41.7 | Keypoints |
| EfficientDet D0 512x512 | 39 | 33.6 | Boxes |
| EfficientDet D1 640x640 | 54 | 38.4 | Boxes |
| EfficientDet D2 768x768 | 67 | 41.8 | Boxes |

รูปที่ ข.3 pre-trained model ต่างๆใน model zoo

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต่อไปเป็น config ไฟล์ pipeline.config สำหรับตั้งค่า parameter ต่างๆในการเทรนข้อมูล โดยรายละเอียดสคริปต์เป็นไปตามต่อไปนี้

```

1 model {
2   ssd {
3     num_classes: 1 # Set this to the number of different label classes
4     image_resizer {
5       fixed_shape_resizer {
6         height: 640
7         width: 640
8       }
9     }
10    feature_extractor {
11      type: "ssd_resnet50_v1_fpn_keras"
12      depth_multiplier: 1.0
13      min_depth: 16
14      conv_hyperparams {
15        regularizer {
16          l2_regularizer {
17            weight: 0.000399999998989515007
18          }
19        }
20        initializer {
21          truncated_normal_initializer {
22            mean: 0.0
23            stddev: 0.0299999999329447746
24          }
25        }
26        activation: RELU_6
27        batch_norm {
28          decay: 0.996999979019165
29          scale: true

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

30     epsilon: 0.0010000000474974513
31   }
32 }
33 override_base_feature_extractor_hyperparams: true
34 fpn {
35   min_level: 3
36   max_level: 7
37 }
38 }
39 box_coder {
40   faster_rcnn_box_coder {
41     y_scale: 10.0
42     x_scale: 10.0
43     height_scale: 5.0
44     width_scale: 5.0
45   }
46 }
47 matcher {
48   argmax_matcher {
49     matched_threshold: 0.5
50     unmatched_threshold: 0.5
51     ignore_thresholds: false
52     negatives_lower_than_unmatched: true
53     force_match_for_each_row: true
54     use_matmul_gather: true
55   }
56 }
57 similarity_calculator {
58   iou_similarity {
59   }
60 }
61 box_predictor {
62   weight_shared_convolutional_box_predictor {
63     conv_hyperparams {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

64     regularizer {
65         l2_regularizer {
66             weight: 0.00039999998989515007
67         }
68     }
69     initializer {
70         random_normal_initializer {
71             mean: 0.0
72             stddev: 0.009999999776482582
73         }
74     }
75     activation: RELU_6
76     batch_norm {
77         decay: 0.996999979019165
78         scale: true
79         epsilon: 0.0010000000474974513
80     }
81 }
82     depth: 256
83     num_layers_before_predictor: 4
84     kernel_size: 3
85     class_prediction_bias_init: -4.599999904632568
86 }
87 }
88     anchor_generator {
89         multiscale_anchor_generator {
90             min_level: 3
91             max_level: 7
92             anchor_scale: 4.0
93             aspect_ratios: 1.0
94             aspect_ratios: 2.0
95             aspect_ratios: 0.5
96             scales_per_octave: 2
97         }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

98 }
99 post_processing {
100   batch_non_max_suppression {
101     score_threshold: 9.99999993922529e-09
102     iou_threshold: 0.6000000238418579
103     max_detections_per_class: 100
104     max_total_detections: 100
105     use_static_shapes: false
106   }
107   score_converter: SIGMOID
108 }
109 normalize_loss_by_num_matches: true
110 loss {
111   localization_loss {
112     weighted_smooth_l1 {
113     }
114   }
115   classification_loss {
116     weighted_sigmoid_focal {
117       gamma: 2.0
118       alpha: 0.25
119     }
120   }
121   classification_weight: 1.0
122   localization_weight: 4.0
123 }
124 encode_background_as_zeros: true
125 normalize_loc_loss_by_codesize: true
126 inplace_batchnorm_update: true
127 freeze_batchnorm: false
128 }
129}
130train_config {

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

131 batch_size: 8 # Increase/Decrease this value depending on the available memory
(Higher values require more memory and vice-versa)
132 data_augmentation_options {
133   random_horizontal_flip {
134   }
135 }
136 data_augmentation_options {
137   random_crop_image {
138     min_object_covered: 0.0
139     min_aspect_ratio: 0.75
140     max_aspect_ratio: 3.0
141     min_area: 0.75
142     max_area: 1.0
143     overlap_thresh: 0.0
144   }
145 }
146 sync_replicas: true
147 optimizer {
148   momentum_optimizer {
149     learning_rate {
150       cosine_decay_learning_rate {
151         learning_rate_base: 0.03999999910593033
152         total_steps: 25000
153         warmup_learning_rate: 0.013333000242710114
154         warmup_steps: 2000
155       }
156     }
157     momentum_optimizer_value: 0.8999999761581421
158   }
159   use_moving_average: false
160 }
161 fine_tune_checkpoint: "pre-trained-models/ssd_resnet50_v1_fpn_640x640_coco17_
tpu-8/checkpoint/ckpt-0" # Path to checkpoint of pre-trained model
162 num_steps: 25000

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

163 startup_delay_steps: 0.0
164 replicas_to_aggregate: 8
165 max_number_of_boxes: 100
166 unpad_groundtruth_tensors: false
167 fine_tune_checkpoint_type: "detection" # Set this to "detection" since we want to be
training the full detection model
168 use_bfloat16: false # Set this to false if you are not training on a TPU
169 fine_tune_checkpoint_version: V2
170}
171 train_input_reader {
172 label_map_path: "annotations/label_map.pbtxt" # Path to label map file
173 tf_record_input_reader {
174   input_path: "annotations/train.record" # Path to training TFRecord file
175 }
176}
177eval_config {
178 metrics_set: "coco_detection_metrics"
179 use_moving_averages: false
180}
181eval_input_reader {
182 label_map_path: "annotations/label_map.pbtxt" # Path to label map file
183 shuffle: false
184 num_epochs: 1
185 tf_record_input_reader {
186   input_path: "annotations/test.record" # Path to testing TFRecord
187 }

```

ต่อไปเป็นการรันคำสั่งในการ training model โดยให้ ssh ไปที่ training_demo ตามด้วยรันสคริปต์ต่อไป

```
>> python model_main_tf2.py --model_dir=models/my_ssd_resnet50_v1_fpn -
-pipeline_config_path=models/my_ssd_resnet50_v1_fpn/pipeline.config
```

วิธีประเมินผลประสิทธิภาพของโมเดล ให้รันคำสั่งต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
>> python model_main_tf2.py --model_dir=models/my_ssd_resnet50_v1_fpn -
-pipeline_config_path=models/my_ssd_resnet50_v1_fpn/pipeline.config --
checkpoint_dir=models/my_ssd_resnet50_v1_fpn
```

วิธีมอนิเตอร์ผลการเทรนใน Tensorboard ให้รันคำสั่งต่อไปนี้

```
>> tensorboard --logdir=models/my_ssd_resnet50_v1_fpn
```

YOLOv5s

YOLO เป็น อัลกอริทึมสำหรับงานตรวจจับวัตถุที่ได้รับความนิยมสูง และมีความโดดเด่นในเรื่องของความเร็วและประสิทธิภาพความแม่นยำ

ติดตั้ง yolov5 จากคำสั่ง git ต่อไปนี้

```
>> git clone https://github.com/ultralytics/yolov5 # clone
>> cd yolov5
>> pip install -r requirements.txt # install
```

หลังจากทำการ labelling ชุดฐานข้อมูลแล้วนำไปใส่ใน yolo แล้วให้รันคำสั่งต่อไปนี้

```
# Train YOLOv5s on COCO128 for 3 epochs
>>!python train.py --img 640 --batch 16 --epochs 3
-- data coco128.yaml --weights yolov5s.pt --cache
```

วิธีแสดงผลการเทรน ให้รันคำสั่งต่อไปนี้

```
>> pip install comet_ml # 1. install
>> export COMET_API_KEY=<Your API Key> # 2. paste API key
>> python train.py --img 640 --epochs 3 --data coco128.yaml --
weights yolov5s.pt # 3. train
```

ต่อไปเป็นการdetect โมเดล

```
>> python detect.py - path/
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Detectron

Detectron2 เป็น library ที่ไว้สำหรับการตรวจจับวัตถุ จำแนกหรือ segmentation ที่ใช้หลักการของการทำ maskrcnn-benchmark
ขั้นตอนติดตั้ง detectron2

```
!python -m pip install pyyaml==5.1
import sys, os, distutils.core
!git clone 'https://github.com/facebookresearch/detectron2'
dist = distutils.core.run_setup("./detectron2/setup.py")
!python -m pip install {' '.join([f"{x}" for x in dist.install_requires])}
sys.path.insert(0, os.path.abspath('./detectron2'))
import torch, detectron2
!nvcc --version
TORCH_VERSION = ".".join(torch.__version__.split(".")[2:])
CUDA_VERSION = torch.__version__.split("+")[-1]
print("torch: ", TORCH_VERSION, "; cuda: ", CUDA_VERSION)
print("detectron2:", detectron2.__version__)
# Some basic setup:
# Setup detectron2 logger
import detectron2
from detectron2.utils.logger import setup_logger
setup_logger()

# import some common libraries
import numpy as np
import os, json, cv2, random
from google.colab.patches import cv2_imshow

# import some common detectron2 utilities
from detectron2 import model_zoo
from detectron2.engine import DefaultPredictor
from detectron2.config import get_cfg
from detectron2.utils.visualizer import Visualizer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
from detectron2.data import MetadataCatalog, ชุดฐานข้อมูลCatalog
```

รับ pre-trained detectron2 model

```
!wget http://images.cocชุดฐานข้อมูล.org/val2017/000000439715.jpg -q -O input.jpg
im = cv2.imread("./input.jpg")
cv2_imshow(im)
cfg = get_cfg()
# add project-
specific config (e.g., TensorMask) here if you're not running a model in detectron2's core l
ibrary
cfg.merge_from_file(model_zoo.get_config_file("COCO-
InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml"))
cfg.MODEL.ROI_HEADS.SCORE_THRESH_TEST = 0.5 # set threshold for this model
# Find a model from detectron2's model zoo. You can use the https://dl.fbaipublicfiles...
url as well
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-
InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml")
predictor = DefaultPredictor(cfg)
outputs = predictor(im)
# look at the outputs. See https://detectron2.readthedocs.io/tutorials/models.html#mo
del-output-format for specification
print(outputs["instances"].pred_classes)
print(outputs["instances"].pred_boxes)
# We can use `Visualizer` to draw the predictions on the image.
v = Visualizer(im[:, :, :-1], MetadataCatalog.get(cfg.ชุดฐานข้อมูลS.TRAIN[0]), scale=1.2)
out = v.draw_instance_predictions(outputs["instances"].to("cpu"))
cv2_imshow(out.get_image()[:, :, :-1])
```

ขั้นตอนการเทรนข้อมูล

```
from detectron2.engine import DefaultTrainer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

cfg = get_cfg()
cfg.merge_from_file(model_zoo.get_config_file("COCO-
InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml"))
cfg.ชุดฐานข้อมูลS.TRAIN = ("balloon_train",)
cfg.ชุดฐานข้อมูลS.TEST = ()
cfg.DATALOADER.NUM_WORKERS = 2
cfg.MODEL.WEIGHTS = model_zoo.get_checkpoint_url("COCO-
InstanceSegmentation/mask_rcnn_R_50_FPN_3x.yaml") # Let training initialize from mod
el zoo
cfg.SOLVER.IMS_PER_BATCH = 2 # This is the real "batch size" commonly known to deep
learning people
cfg.SOLVER.BASE_LR = 0.00025 # pick a good LR
cfg.SOLVER.MAX_ITER = 300 # 300 iterations seems good enough for this toy ชุด
ฐานข้อมูล; you will need to train longer for a practical ชุดฐานข้อมูล
cfg.SOLVER.STEPS = [] # do not decay learning rate
cfg.MODEL.ROI_HEADS.BATCH_SIZE_PER_IMAGE = 128 # The "RoIHead batch size". 128 is
faster, and good enough for this toy ชุดฐานข้อมูล (default: 512)
cfg.MODEL.ROI_HEADS.NUM_CLASSES = 1

os.makedirs(cfg.OUTPUT_DIR, exist_ok=True)
trainer = DefaultTrainer(cfg)
trainer.resume_or_load(resume=False)
trainer.train()

แสดงผลการเทรนผ่าน tensorboard

%load_ext tensorboard
%tensorboard --logdir output

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประวัติผู้เขียน

| | |
|---------------------------------|--|
| ชื่อ-นามสกุล | นายพัศรุจย์ แพ้ไชยภูมิ |
| วัน เดือน ปีเกิด | 20 พฤษภาคม 2536 ที่กรุงเทพมหานคร |
| ที่อยู่ | 11 ซอยบางนา-ตราด19 แยก10 ถ.บางนา-ตราด แขวงบางนา เขตบางนา กรุงเทพฯ 10260 โทร.0-86411-5278 |
| ประวัติการศึกษา | 2558 วิศวกรรมศาสตรบัณฑิต สาขาวิศวกรรมหุ่นยนต์และระบบควบคุมอัตโนมัติ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าพระนครเหนือ |
| ความชำนาญเฉพาะด้าน | 1.) เทคโนโลยีหุ่นยนต์และคอนโทรล 2.) เทคโนโลยีควบคุมอัตโนมัติ |
| ประสบการณ์การทำงานและผลงานวิจัย | |
| พ.ศ.2558-2559 | ตำแหน่งวิศวกรฝ่ายผลิตบริษัทซูมิโตโม อิเล็กทริกฮาร์ดเมทัล ไทยแลนด์ จำกัด |
| พ.ศ.2559-ปัจจุบัน | ตำแหน่งวิศวกรด้านเทคโนโลยีหุ่นยนต์ สังกัดหน่วยงานด้านนวัตกรรมและพัฒนา ประสิทธิภาพ วิศวกรรมกลาง บ. เจริญโภคภัณฑ์อาหาร จำกัด(มหาชน) |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้