



Goggle : People Video Analytics and Deep Learning Platform

Goggle แพลตฟอร์มการเรียนรู้เชิงลึกและระบบวิเคราะห์การกระทำของมนุษย์

นายปฐมพงศ์ สินธุจาม

นายศุภกร เบญจวิกรัย

นายอุกฤษฎ์ เลิศวรรณาการ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ

สถาบันวิทยาการหุ่นยนต์ภาคสนาม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ปีการศึกษา 2562



Goggle : People Video Analytics and Deep Learning Platform

Goggle แพลตฟอร์มการเรียนรู้เชิงลึกและระบบวิเคราะห์การกระทำของมนุษย์

นายปฐมพงศ์ สินธุจาม

นายศุภกร เบญจวิกรัย

นายอุกฤษฎ์ เลิศวรรณาการ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาชีวกรรมทุนยนต์และระบบอัตโนมัติ

สถาบันวิทยาการทุนยนต์ภาครสนา

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ปีการศึกษา 2562

Google แพลตฟอร์มการเรียนรู้เชิงลึกและระบบวิเคราะห์การกระทำของมนุษย์

นายปฐมพงศ์ สินธุ์งาม

นายศุภกร เบญจวิกราย

นายอุกฤษฎ์ เดิศวรณาการ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมหุ่นยนต์และระบบอัตโนมัติ

สถาบันวิทยาการหุ่นยนต์ภาคสนาม

มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

ปีการศึกษา 2562

คณะกรรมการสอบวิทยานิพนธ์

ประธานกรรมการสอบวิทยานิพนธ์

(ดร.วราสินี ฉายแสงมงคล)

อาจารย์ที่ปรึกษาวิทยานิพนธ์

(ดร.วราสินี ฉายแสงมงคล)

กรรมการสอบวิทยานิพนธ์

(อ.บวรศักดิ์ ศกุลเกื้อกุลสุข)

กรรมการสอบวิทยานิพนธ์

(ดร.บุญทริกา เกษมสันติธรรม)

ลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี

| | |
|------------------|---|
| ชื่อวิทยานิพนธ์ | Goggle แพลตฟอร์มการเรียนรู้เชิงลึกและระบบวิเคราะห์การกระทำของมนุษย์ |
| หน่วยกิต | 6 |
| ผู้เขียน | นายปัจมพงศ์ สินธุจัน นายศุภกร เบญจวิกรัย นายอุกฤษฎ์ เลิศวรรณการ |
| อาจารย์ที่ปรึกษา | ทีปรึกษาวิทยานิพนธ์หลัก ดร.วราสินี ฉายแสงมงคล |
| หลักสูตร | วิศวกรรมศาสตรบัณฑิต |
| สาขาวิชา | วิศวกรรมที่นุนต์และระบบอัตโนมัติ |
| คณะ | สถาบันวิทยาการหุ่นยนต์ภาคสนาม |
| ปีการศึกษา | 2562 |

บทคัดย่อ

งานวิทยานิพนธ์นี้เป็นงานที่เกี่ยวกับการออกแบบและสร้างเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ รวมถึงระบบวิเคราะห์การกระทำของมนุษย์ โดยใช้ชื่อว่า Goggle แพลตฟอร์มการเรียนรู้เชิงลึกและระบบวิเคราะห์การกระทำของมนุษย์ ซึ่งมีจุดประสงค์เพื่อให้ผู้พัฒนาสามารถใช้งานเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ในการสร้างชุดข้อมูลสำหรับสร้างปัญญาประดิษฐ์ได้ง่ายและสะดวกขึ้น ภาพรวมของวิทยานิพนธ์นี้จะแบ่งออกเป็นหัวข้อหลักสองส่วน โดยที่ส่วนแรกเป็นการออกแบบและสร้างแอปพลิเคชันที่ใช้ในการสร้างชุดข้อมูลจากวิดีโอ สำหรับพัฒนาโมเดลปัญญาประดิษฐ์ และส่วนที่สองเป็นใช้ชุดข้อมูลที่ได้จากแอปพลิเคชันในการพัฒนาโมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำการกระทำของมนุษย์ เพื่อทดสอบประสิทธิภาพของชุดข้อมูล

คำสำคัญ : ระบบวิเคราะห์การกระทำการกระทำของมนุษย์ / เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ / Goggle

กิตติกรรมประกาศ

ขอขอบพระคุณ ดร.วราสินี ฉายแสงมงคล อาจารย์ที่ปรึกษาวิทยานิพนธ์หลัก ที่ได้สละเวลามาให้คำปรึกษา ชี้แนะแนวทาง ให้ความรู้ในด้านต่างๆ ที่จำเป็นต่องานวิจัย รวมถึงการให้การสนับสนุนในเรื่องอุปกรณ์ในการทำวิจัย ช่วยตรวจสอบและแก้ไขวิทยานิพนธ์ให้เป็นไปอย่างสมบูรณ์ ตลอดจนกรุณาให้เกียรติเป็นประธานกรรมการสอบวิทยานิพนธ์

ขอขอบพระคุณอาจารย์อาจารย์ บวรศักดิ์ สกุลเกื้อกูลสุข ที่กรุณาให้เกียรติเป็นกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่เป็นประโยชน์ต่อการวิจัย และการแก้ไขปรับปรุงงานวิจัย ตลอดจนตรวจสอบแก้วิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณอาจารย์ ดร.บุญทริกา เกษมสันติธรรม ที่กรุณาให้เกียรติเป็นกรรมการสอบวิทยานิพนธ์ ให้คำแนะนำที่เป็นประโยชน์ต่อการวิจัย และการแก้ไขปรับปรุงงานวิจัย ตลอดจนตรวจสอบแก้วิทยานิพนธ์ให้ดำเนินไปอย่างสมบูรณ์

ขอขอบพระคุณคณาจารย์ และบุคลากรในสถาบันวิทยาการหุ่นยนต์ภาคนามทุกท่าน ที่ได้ให้คำปรึกษา และช่วยเหลือด้านสถานที่พร้อมทั้งส่งอำนวยความสะดวกต่างๆ ในระหว่างการทำวิทยานิพนธ์

ขอขอบคุณนักศึกษาปริญญาตรี สถาบันวิทยาการหุ่นยนต์ภาคนามทุกท่าน ที่ได้ให้คำแนะนำ ถามไถ่ และเป็นกำลังใจมาโดยตลอด

และสุดท้ายนี้ ขอน้อมรำลึกถึงพระคุณบิดา มารดา และครอบครัว ที่ส่งเสริมให้กำลังใจ และให้การสนับสนุนในเรื่องต่างๆ จนกระทั้งข้าพเจ้าประสบความสำเร็จในการศึกษา

นายปฐมพงศ์ สินธุรงาม
นายศุภกร เบญจวิกรัย
นายอุกฤษฎ์ เลิศวรรณาการ

สารบัญ

| เรื่อง | หน้า |
|--|----------|
| บทคัดย่อ | ค |
| กิตติกรรมประกาศ | ๔ |
| สารบัญ | จ |
| รายการรูปภาพ | ณ |
| รายการตาราง..... | ภ |
| ประมวลศัพท์และตัวย่อ..... | ธ |
| ประมวลศัพท์และตัวย่อ..... | ท |
| ประมวลศัพท์และตัวย่อ..... | ฒ |
| บทที่ 1 บทนำ..... | ๑ |
| 1.1 ที่มาและความสำคัญ..... | 1 |
| 1.2 วัตถุประสงค์..... | 1 |
| 1.3 ประโยชน์ที่คาดว่าจะได้รับ | 1 |
| 1.4 ขอบเขตการดำเนินงาน..... | 2 |
| 1.5 ขั้นตอนการดำเนินงาน | 2 |
| บทที่ 2 ทฤษฎี/การวิจัยที่เกี่ยวข้อง | ๔ |
| 2.1 ทฤษฎีพื้นฐาน | 5 |
| 2.1.1 Convolutional neural network..... | 5 |
| 2.1.2 Optical flow..... | 13 |
| 2.1.3 โมเดลปัญญาประดิษฐ์สำหรับตรวจสอบวัตถุที่เกี่ยวข้องกับงานวิจัย | 15 |
| 2.2 การประมวลผลวิดีโอ..... | 18 |
| 2.2.1 การตรวจจับวัตถุ..... | 18 |
| 2.2.2 การติดตามการเคลื่อนไหวของวัตถุ | 20 |
| 2.2.3 การระบุตัวตนของบุคคล..... | 25 |
| 2.2.4 การจำแนกการกระทำของมนุษย์..... | 27 |
| 2.3 ชุดข้อมูลสำหรับการประมวลผลวิดีโอ | 28 |
| 2.3.1 ชุดข้อมูล YouTube-8M..... | 28 |
| 2.3.2 ชุดข้อมูล Atomic visual action (AVA) | 35 |

สารบัญ (ต่อ)

| เรื่อง | หน้า |
|---|-----------|
| 2.3.3 ชุดข้อมูล Moments in Time | 38 |
| 2.4 โมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำมนุษย์..... | 41 |
| 2.4.1 ResNet..... | 41 |
| 2.4.2 Inflated 3D convolutional network | 43 |
| 2.5 เครื่องมือกำกับคุณลักษณะ | 45 |
| บทที่ 3 ระเบียบวิธีวิจัย | 47 |
| 3.1 ความต้องการของระบบ..... | 47 |
| 3.1.1 ความต้องการใช้งาน (functional requirements)..... | 47 |
| 3.1.2 ความต้องการเชิงวิศวกรรม (non-functional requirements)..... | 47 |
| 3.2 ภาพรวมระบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์..... | 48 |
| 3.3 หน้าที่ความรับผิดชอบ..... | 49 |
| 3.4 เครื่องมือที่ใช้ในงานวิจัย | 49 |
| 3.5 ภาษาที่ใช้ในการพัฒนาระบบ | 50 |
| 3.6 Program library ที่ใช้ในการพัฒนาระบบและแอปพลิเคชัน | 50 |
| 3.7 แผนการดำเนินงาน | 50 |
| 3.8 การออกแบบหน้าต่างแอปพลิเคชันของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์..... | 51 |
| 3.8.1 เครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์..... | 51 |
| 3.9 การออกแบบการทดสอบการตรวจจับวัตถุ..... | 61 |
| 3.9.1 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์สำหรับการทำการตรวจจับภาพบุคคล | 61 |
| 3.10 การออกแบบการทดสอบการติดตามการเคลื่อนไหวตำแหน่งต่อไปของมนุษย์ | 62 |
| 3.10.1 ทดสอบประสิทธิภาพการทำงานของระบบการติดตามการเคลื่อนไหวตำแหน่งต่อไปของวัตถุในวิดีโอ | 62 |
| 3.11 การออกแบบการทดสอบการระบุตัวตนของมนุษย์ | 63 |
| 3.11.1 ทดสอบประสิทธิภาพการทำงานของระบบระบุตัวตนของบุคคลภายในภาพ | 63 |
| 3.12 การออกแบบการทดสอบการจำแนกการกระทำของมนุษย์ | 64 |
| 3.12.1 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Faster RCNN ResNet101 ที่ถูกสร้างด้วยชุดข้อมูลของ AVA โดยใช้ชุดข้อมูลของ AVA ในการทดสอบและเทียบผลลัพธ์ กับแหล่งอ้างอิง | 64 |

สารบัญ (ต่อ)

| เรื่อง | หน้า |
|---|-----------|
| 3.12.2 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Faster RCNN ResNet101 ที่เคยสร้างด้วยชุดข้อมูลของ AVA และใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิง..... | 65 |
| 3.12.3 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ ResNet50 ที่ถูกสร้างด้วยชุดข้อมูลที่ผู้วิจัยสร้างขึ้น และใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิง..... | 66 |
| 3.12.4 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ I3D สร้างด้วยชุดข้อมูลที่ผู้วิจัยสร้างขึ้น โดยใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบ..... | 67 |
| บทที่ 4 ผลการดำเนินงาน | 68 |
| 4.1 เครื่องมือกำกับคุณลักษณะ | 68 |
| 4.1.1 หน้าต่างแสดงผลของแอปพลิเคชัน | 68 |
| 4.1.2 ผลลัพธ์การทำงานในแต่ละหน้าต่างของแอปพลิเคชัน | 73 |
| 4.2 ผลการทดลองการตรวจจับวัตถุ | 76 |
| 4.2.1 ข้อมูลรายละเอียดประกอบการทดสอบ..... | 76 |
| 4.2.2 ผลทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์สำหรับการทำตรวจนับภาพบุคคล | 76 |
| 4.3 ผลการทดสอบระบบติดตามตำแหน่งของมนุษย์ | 77 |
| 4.3.1 ข้อมูลรายละเอียดประกอบการทดสอบ..... | 77 |
| 4.3.2 ผลทดสอบประสิทธิภาพ และความเร็วในการประมวลผล..... | 77 |
| 4.4 ผลการทดสอบระบบระบุตัวตนของมนุษย์ | 78 |
| 4.4.1 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์สำหรับการระบุตัวตนของบุคคล | 78 |
| 4.5 ผลการทดสอบการจำแนกการกระทำของมนุษย์ | 80 |
| 4.5.1 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Resnet50 ที่ถูกสร้างด้วยชุดข้อมูลของ AVA โดยใช้ชุดข้อมูลของ AVA ในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิง ได้ผลการทดลองดังตารางต่อไปนี้..... | 80 |
| 4.5.2 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Resnet50 ที่เคยสร้างด้วยชุดข้อมูลของ AVA และใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิง | 80 |

สารบัญ (ต่อ)

| เรื่อง | หน้า |
|---|-----------|
| 4.5.3 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Resnet50 ที่ถูกสร้างด้วยชุดข้อมูลที่ผู้วิจัยสร้างขึ้น และใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบและเทียบผลลัพธ์การทดสอบก่อนหน้า..... | 81 |
| 4.5.4 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ 3D สร้างด้วยชุดข้อมูลที่ผู้วิจัยสร้างขึ้น โดยใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบ | 83 |
| บทที่ 5 การอภิปรายผล สรุปผลงานวิจัย และข้อเสนอแนะ | 86 |
| 5.1 การอภิปรายผล | 86 |
| 5.1.1 เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ | 86 |
| 5.1.2 ชุดข้อมูล | 86 |
| 5.1.3 โมเดลปัญญาประดิษฐ์ | 86 |
| 5.2 สรุปผลงานวิจัย | 87 |
| 5.2.1 เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ | 87 |
| 5.2.2 โมเดลปัญญาประดิษฐ์ | 87 |
| 5.3 ข้อเสนอแนะ | 87 |
| 5.3.1 เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ | 87 |
| 5.3.2 โมเดลปัญญาประดิษฐ์ | 88 |
| เอกสารอ้างอิง | 92 |
| ภาคผนวก ก ตัวอย่างชุดข้อมูลที่ผู้วิจัยสร้างขึ้น | 93 |

รายการรูปภาพ

| รูป | หน้า |
|---|------|
| รูปที่ 1.1 แผนการดำเนินงาน..... | 3 |
| รูปที่ 2.1 ตัวอย่างโครงสร้างของ CNN ที่ใช้ในการจำแนกหมวดหมู่ของวัตถุ..... | 5 |
| รูปที่ 2.2 ตัวอย่างเครื่องเรนล และภาพที่ใช้ในการประมวลผล | 5 |
| รูปที่ 2.3 ตัวอย่างการหาผังคุณลักษณะ | 6 |
| รูปที่ 2.4 ตัวอย่างการใช้ max pooling และ average pooling กับภาพ..... | 6 |
| รูปที่ 2.5 ภาพแสดงการหา IoU ด้วยการใช้คำตอบจริงของการทำงานและกรอบสี่เหลี่ยมที่ได้จากการทำงาน..... | 8 |
| รูปที่ 2.6 AUC - ROC Curve | 8 |
| รูปที่ 2.7 ตัวอย่างการทำงานของ NMS | 9 |
| รูปที่ 2.8 รูปสำหรับอธิบายการทำงานของ gradient descent..... | 10 |
| รูปที่ 2.9 เปรียบเทียบการทำงานของ gradient descent กับ Momentum | 11 |
| รูปที่ 2.10 ตัวอย่าง optical flow ของการเคลื่อนที่ของกลูกบอล | 13 |
| รูปที่ 2.11 กระบวนการทำงานของโครงสร้างโมเดลปัญญาประดิษฐ์ของ YOLO | 15 |
| รูปที่ 2.12 โครงสร้างทั่วไปของโมเดลปัญญาประดิษฐ์ของ YOLO | 15 |
| รูปที่ 2.13 โครงสร้างโมเดลปัญญาประดิษฐ์ของ YOLO | 16 |
| รูปที่ 2.14 โครงสร้างทั่วไปของโมเดลปัญญาประดิษฐ์ของ Faster RCNN | 17 |
| รูปที่ 2.15 โครงสร้างทั่วไปของโมเดลปัญญาประดิษฐ์ SSD | 18 |
| รูปที่ 2.16 ผังการทำงานของระบบติดตามการเคลื่อนไหวของวัตถุแบบ kalman filter | 20 |
| รูปที่ 2.17 ผังการทำงานของระบบติดตามการเคลื่อนไหวของวัตถุแบบ correlation filter | 23 |
| รูปที่ 2.18 การแบ่งภาพออกเป็น 8 ส่วนของระบบระบุตัวตนของบุคคล | 25 |
| รูปที่ 2.19 ตัวอย่างการประยุกต์ใช้งานระบบจำแนกการกระทำมนุษย์ ^[25] | 27 |
| รูปที่ 2.20 โครงสร้างโมเดลปัญญาประดิษฐ์ของ deep bag of frames | 30 |
| รูปที่ 2.21 โครงสร้าง LSTM ที่ใช้การอ้างอิงในบทความนี้..... | 30 |
| รูปที่ 2.22 หลักการของ Residual block ของ ResNet | 41 |
| รูปที่ 2.23 โครงสร้างของโมเดลปัญญาประดิษฐ์ I3D | 43 |
| รูปที่ 2.24 โครงสร้างของโมเดลปัญญาประดิษฐ์ I3D | 43 |
| รูปที่ 2.25 หน้าต่างการทำงานของโปรแกรม DarkLabel | 45 |

รายการรูปภาพ (ต่อ)

| รูป | หน้า |
|---|------|
| รูปที่ 2.26 หน้าต่างการทำงานของโปรแกรม OpenLabeling | 46 |
| รูปที่ 3.1 ภาพรวมระบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ | 48 |
| รูปที่ 3.2 กระบวนการหลักของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์ | 51 |
| รูปที่ 3.3 หน้าต่าง Select ของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์ | 52 |
| รูปที่ 3.4 ตำแหน่งของแต่ละวิดเจ็ตในหน้าต่าง Select | 53 |
| รูปที่ 3.5 หน้าต่าง Detect ของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์ | 54 |
| รูปที่ 3.6 ตำแหน่งของแต่ละวิดเจ็ตในหน้าต่าง Detect | 55 |
| รูปที่ 3.7 หน้าต่าง Track ของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์ | 56 |
| รูปที่ 3.8 ตำแหน่งของแต่ละวิดเจ็ตในหน้าต่าง Track | 57 |
| รูปที่ 3.9 หน้าต่าง Label ของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์ | 58 |
| รูปที่ 3.10 ตำแหน่งของแต่ละวิดเจ็ตในหน้าต่าง Label | 59 |
| รูปที่ 3.11 ตัวอย่างข้อมูลภายในไฟล์ xml | 60 |
| รูปที่ 4.1 รูปหน้าต่างแสดงผลของหน้าต่าง Select | 68 |
| รูปที่ 4.2 รูปหน้าต่างแสดงผลของหน้าต่าง Detect | 69 |
| รูปที่ 4.3 รูปหน้าต่างแสดงผลของหน้าต่าง Detect | 70 |
| รูปที่ 4.4 รูปหน้าต่างแสดงผลของหน้าต่าง Track | 71 |
| รูปที่ 4.5 รูปหน้าต่างแสดงผลของหน้าต่าง Label | 72 |
| รูปที่ 4.6 รูปผลลัพธ์การแยกเฟรมที่มีมนุษย์อยู่และไม่มีมนุษย์อยู่ภายใต้เฟรม | 73 |
| รูปที่ 4.7 รูปคีย์เฟรมที่ถูกตีกรอบสีเหลืองในส่วนที่มีมนุษย์อยู่ | 73 |
| รูปที่ 4.8 รูปผลลัพธ์การทำงานของหน้าต่าง Track | 74 |
| รูปที่ 4.9 รูปผลลัพธ์การทำงานของหน้าต่าง Label | 75 |
| รูปที่ 4.10 กราฟแสดงการเปรียบระหว่างการระบุว่าเป็นบุคคลเดียวกันกับไม่เป็นบุคคลเดียวกัน โดยพื้นที่ต่อกرافที่เป็นสีเขียวจะหมายถึงการระบุว่าเป็นบุคคลเดียวกัน ในขณะที่พื้นที่ต่อกرافสีแดงหมายถึงการระบุว่าไม่เป็นบุคคลเดียวกัน และแกน x คือค่า aligned distace ส่วนของแกน y จำนวนคู่ของภาพ | 78 |
| รูปที่ ก.1 รูปผลลัพธ์การทำงานของหน้าต่าง Track | 93 |

รายการตาราง

| ตาราง | หน้า |
|--|------|
| ตารางที่ 2.1 ตารางแสดงการเปรียบเทียบระหว่าง softmax function และ sigmoid function..... | 7 |
| ตารางที่ 2.2 ผลการทดสอบโมเดลต่างๆบนชุดข้อมูลสำหรับทดสอบของ YouTube-8M..... | 33 |
| ตารางที่ 2.3 ผลการทดลองของวิธีต่างๆบนคุณลักษณะระดับเฟรม | 37 |
| ตารางที่ 2.4 ประสิทธิภาพของโมเดล Resnet50 I3D ที่ใช้ชุดข้อมูล Kinetics และ Moments in Time .. | 39 |
| ตารางที่ 2.5 อัตราการถูกจำแนกของโมเดล ImageNet..... | 41 |
| ตารางที่ 2.6 ค่าความผิดพลาดที่ได้จากการทดลองจำแนกชั้นของโมเดลปัญญาประดิษฐ์ ResNet บนชุดของข้อมูล CIFAR-10 | 42 |
| ตารางที่ 2.7 ประสิทธิภาพของโมเดล I3D แบบ two-stream เมื่อใช้ข้อมูลจาก UCF-101, HMDB-51 และ Kinetics ในการสร้างและทดสอบด้วยเครื่องมือวัดผลแบบความแม่นยำจากการทำนายอันดับแรกสุด..... | 44 |
| ตารางที่ 3.1 ตารางแสดงจำนวนภาพที่ใช้ในการทดลองนี้ | 66 |
| ตารางที่ 3.2 ตารางแสดงจำนวนชุดของข้อมูลที่ใช้ในการทดลองนี้ | 67 |
| ตารางที่ 4.1 ข้อมูลผลการทำงานของโมเดลปัญญาประดิษฐ์สำหรับการทำการทำการตรวจจับภาพบุคคล | 76 |
| ตารางที่ 4.2 ผลการทดสอบประสิทธิภาพของการตรวจจับกรอบสี่เหลี่ยมภายในวิดีโอ..... | 77 |
| ตารางที่ 4.3 ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์..... | 78 |
| ตารางที่ 4.4 ผลการทดสอบค่า AUC ของโมเดลปัญญาประดิษฐ์ | 79 |
| ตารางที่ 4.5 ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่比べผลลัพธ์กับแหล่งอ้างอิง..... | 80 |
| ตารางที่ 4.6 ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ เมื่อใช้กับชุดข้อมูลที่ผู้วิจัยสร้างขึ้น.... | 80 |
| ตารางที่ 4.7 ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้น ใช้กับชุดข้อมูลที่ผู้วิจัยสร้างขึ้น | 81 |
| ตารางที่ 4.8 ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้นโดยใช้ weight จาก ImageNet ใช้กับชุดข้อมูลที่ผู้วิจัยสร้างขึ้น..... | 82 |
| ตารางที่ 4.9 ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้นโดยใช้ weight จาก ImageNet และการทำ scaling ใช้กับชุดข้อมูลที่ผู้วิจัยสร้างขึ้น..... | 82 |
| ตารางที่ 4.10 โมเดลปัญญาประดิษฐ์ ResNet50-ImageNet Centering Samplewise วัดค่า average precision (AP) ของแต่ละหมวดหมู่ | 83 |
| ตารางที่ 4.11 ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้นโดยใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นแบบภาพสีปกติ | 84 |

รายการตาราง (ต่อ)

| | |
|--|------|
| ตาราง | หน้า |
| ตารางที่ 4.12 ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้นโดยใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นแบบ optical flow | 84 |
| ตารางที่ 4.13 เปรียบเทียบค่า AP ของทุกการกระทำของแต่ละโมเดล | 84 |
| ตารางที่ 4.14 ผลการทดสอบความแม่นยำเมื่อนำโมเดลแบบภาพสีปกติ และ optical flow มาใช้ร่วมกัน..... | 85 |

ประมวลศัพท์และตัวย่อ

| | |
|------------------------------|---|
| Accuracy | ความถูกต้อง |
| Action classification | การจำแนกการกระทำ |
| AI assisted labeling tool | เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ |
| Algorithm | อัลกอริทึม |
| Amazon mechanical turk | ตลาดแรงงาน |
| Architecture | โครงสร้างของโมเดลปัญญาประดิษฐ์ |
| Artificial intelligence (AI) | ปัญญาประดิษฐ์ |
| Auditory | เสียง |
| Blacklist | บัญชีดำ |
| Binary feature vector | เวกเตอร์คุณลักษณะฐานสอง |
| Bounding box | กรอบสี่เหลี่ยม |
| Class | หมวดหมู่ |
| Dataset | ชุดข้อมูล |
| Directory path | เส้นทางในคอมพิวเตอร์ |
| Extract feature | การสกัดคุณลักษณะ |
| Feature | คุณลักษณะ |
| Feature map | ผังคุณลักษณะ |
| Feature vector | เวกเตอร์คุณลักษณะ |
| Filter | ตัวกรอง |
| Fourier transform | การแปลงฟูรีเยร์ |
| Framerate per second (FPS) | อัตราเฟรมต่อวินาที |
| Functional requirements | ความต้องการเชิงการใช้งาน |
| Gradient | เกรเดียน |
| Gray scale | ภาพขาวดำ |
| Ground-truth | คำตอบของชุดข้อมูล |
| Ground-truth tubes | กรอบสี่เหลี่ยมจริงในช่วงของเฟรมที่ต้องกัน |
| Height | ความสูง |
| Human working memory | ช่วงเวลาที่มนุษย์เข้าใจเหตุการณ์ที่เกิดขึ้น |
| Hungarian distance | อัลกอริทึมสำหรับการหาข้อเสนอที่ดีที่สุด |
| Image processing | การประมวลผลภาพ |
| Image understanding | การทำความเข้าใจภาพด้วยปัญญาประดิษฐ์ |

ประมวลศัพท์และตัวย่อ

| | |
|-------------------------------|--|
| Interaction with object | ปฏิสัมพันธ์กับวัตถุ |
| Interaction with people | ปฏิสัมพันธ์กับบุคคล |
| Intersection over union (IoU) | อัตราส่วนร่วมของกรอบสี่เหลี่ยม |
| Inverse fourier transform | การแปลงฟูรีเยร์ผกผน |
| Kernel | เคอร์เนล |
| Keyframe | คีย์เฟรม |
| Label | คำกำกับคุณลักษณะของภาพ |
| Labeling tool | เครื่องมือกำกับคุณลักษณะ |
| Layer | ชั้นการทำงานของโมเดลปัญญาประดิษฐ์ |
| Linked detection tubes | กรอบสี่เหลี่ยมที่ทำงานขึ้นมาในช่วงที่เฟรมต่อกัน |
| Machine learning model | โมเดลปัญญาประดิษฐ์ |
| Mode auto | การทำงานแบบอัตโนมัติ |
| Mode manual | การทำงานแบบทำด้วยตัวเอง |
| Neural network | โครงข่ายประสาทเทียม |
| Non-functional requirements | ความต้องการเชิงวิศวกรรม |
| Object detection | ระบบตรวจจับวัตถุ |
| Object tracker | ระบบติดตามการเคลื่อนไหวของวัตถุ |
| Open source | ชิ้นงานสาธารณะ |
| Person detection | ระบบตรวจจับมนุษย์ |
| Person embedding | ให้โครงข่ายประสาทเทียมในการหาคุณลักษณะ ของสิ่งและใช้เมทริกซ์ในการหาความสัมพันธ์ของแต่ละคน |
| Person re-identification | การระบุตัวตนของบุคคล |
| Person tracker | ระบบติดตามการเคลื่อนไหวของมนุษย์ |
| Pixel | พิกเซล |
| Pose | ท่าทาง |
| Predict | การทำนาย |
| Region of interest (ROI) | พื้นที่ที่สนใจ |
| Spatial | ภายนอก |
| Temporal | การเคลื่อนไหว |
| Test set | ชุดข้อมูลสำหรับการทดสอบ |
| Threshold | เกณฑ์สำหรับการตัดสินหรือแบ่ง |
| Train set | ชุดข้อมูลสำหรับสร้างโมเดลปัญญาประดิษฐ์ |

របៀបគ្រប់គ្រងព័ត៌មាន

| | |
|---------------------|---|
| User interface | អនាគារទំនើសនៃពិធីការណ៍ដូចជា |
| Valid set | ចុះឈ្មោះតាមលក្ខណៈរបៀបទាញយកព័ត៌មាន |
| Vector | វេភាពទេរី |
| Video analytic | ការប្រមិជ្ជការណ៍វិដីអូ |
| Video theme | ភាសាដំឡើងនៃវិដីអូ |
| Video understanding | ការធានាប្រព័ន្ធដោយប្រើប្រាស់បច្ចុប្បន្ន |
| widget | វិគិត |
| Width | ការពារការងារ |
| Whitelist | បញ្ជីខ្លាំង |

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญ

บริษัท เพอเช็ปตรา ดำเนินธุรกิจเกี่ยวกับการให้บริการและคำปรึกษาเกี่ยวกับปัญญาประดิษฐ์ (artificial intelligence) เนื่องจากปัจจุบันนั้นความสามารถและประสิทธิภาพของปัญญาประดิษฐ์มีความก้าวหน้าขึ้นจนสามารถก้าวข้ามความสามารถของมนุษย์ในงานหลายประเภท ทำให้ถูกค้นหานั้นมีความต้องการที่จะให้ทางบริษัทสร้างปัญญาประดิษฐ์เพื่อนำไปใช้งานหรือแก้ปัญหาที่ต่างกันออกไป เช่น ใช้ปัญญาประดิษฐ์มาช่วยประเมินผลภาพจากกล้องวงจรปิด เพื่อหาบุคคลที่มีท่าทางน่าสงสัย เป็นต้น ซึ่งการจะสร้างปัญญาประดิษฐ์ที่เหมาะสมกับการแก้ปัญหาเหล่านั้น จะเป็นต้องมีชุดข้อมูล (dataset) ที่เหมาะสม บางครั้งอาจต้องใช้มนุษย์ในการสร้างขึ้นมาโดยการเก็บข้อมูลวิดีโอ และลงมือสร้างชุดข้อมูลจากวิดีโอที่ได้ด้วยตัวเอง หนึ่งในปัจจัยสำคัญในการพัฒนาโมเดลปัญญาประดิษฐ์ (machine learning model) ให้มีประสิทธิภาพสูงคือจำนวนข้อมูล ซึ่งหากมีจำนวนวิดีโอเป็นจำนวนมาก การใช้มนุษย์ในการสร้างชุดข้อมูลนั้นอาจจะต้องใช้มนุษย์เป็นจำนวนมาก และใช้เวลานาน

ทางคณะผู้วิจัยจึงมีความต้องการที่จะออกแบบและสร้างต้นแบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ (AI-assisted video labeling tool) สำหรับสร้างชุดข้อมูลจากวิดีโอ เพื่อช่วยแบ่งเบ้าภาระของผู้พัฒนาในการสร้างชุดข้อมูลสำหรับการพัฒนาโมเดลปัญญาประดิษฐ์ในการแก้ปัญหาที่ลูกค้าต้องการ โดยโครงการสหกิจนี้จะศึกษาเกี่ยวกับการวิเคราะห์และจำแนกการกระทำของมนุษย์ (action classification) ภายในสำนักงานจากวิดีโอด้วยหลัก

1.2 วัตถุประสงค์

1. เพื่อสร้างต้นแบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ ที่มีนุชร์และปัญญาประดิษฐ์ทำงานร่วมกันเพื่อสร้างชุดข้อมูลในการนำไปพัฒนาปัญญาประดิษฐ์อื่นๆที่เหมาะสมกับปัญหาที่ต้องการ
 2. เพื่อออกแบบและสร้างต้นแบบของระบบประมวลผลวิดีโอ (video analytics) ที่สามารถตรวจจับมนุษย์ และจำแนกการกระทำพื้นฐานของมนุษย์ภายในสำนักงาน ประกอบด้วย ยืน นั่ง เดิน เล่นโทรศัพท์ กินข้าว นอน โดยใช้ปัญญาประดิษฐ์
 3. เพื่อสร้างเครื่องมือที่สามารถสร้างชุดข้อมูลสำหรับการจำแนกการกระทำการของมนุษย์ให้สามารถใช้งานได้ง่าย สะดวกสบายมากขึ้น และมีประสิทธิภาพที่สูงกว่าเครื่องมือที่เปิดให้ใช้งานสาธารณะ (open source) ตัวอื่นในปัจจุบัน

1.3 ประโยชน์ที่คาดว่าจะได้รับ

- เพิ่มความสอดคล้องในการสร้างชุดข้อมูลสำหรับพัฒนาโมเดลปัญญาประดิษฐ์จากวิดีโอ
 - ต้นแบบระบบประมวลผลวิดีโอที่สามารถจำแนกการกระทำของมนุษย์

1.4 ขอบเขตการดำเนินงาน

- สร้างต้นแบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ โดยระบบจะประกอบไปด้วยสี่ส่วนดังนี้
 - หน้าต่างของติดต่อ กับผู้ใช้ (user interface)
 - ระบบตรวจจับมนุษย์ในภาพ (person detection)
 - ระบบติดตามการเคลื่อนไหวของมนุษย์ในวิดีโอ (person tracker)
 - ระบบจำแนกการกระทำของมนุษย์ ซึ่งประกอบไปด้วย ยืน นั่ง เดิน เล่นโทรศัพท์ กินข้าว นอน
 - ทดสอบโมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำการของมนุษย์กับชุดข้อมูลที่ได้จากการเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ เพื่อทดสอบว่าชุดข้อมูลที่ได้สามารถใช้งานจริงได้หรือไม่
 - พัฒนาโมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำการของมนุษย์ภายในสำนักงานอย่างน้อย 2 โมเดล

1.5 ขั้นตอนการดำเนินงาน

การดำเนินงานวิจัยถูกแบ่งออกเป็นสามส่วน โดยส่วนแรกคือการศึกษาเทคโนโลยีในปัจจุบันเพื่อหาความเป็นไปได้และกำหนดขอบเขตของงาน ส่วนที่สองคือออกแบบและสร้างเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ เพื่อช่วยผู้พัฒนาในการสร้างชุดข้อมูล และส่วนที่สุดท้ายคือการนำชุดข้อมูลที่ได้จากการใช้เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ไปพัฒนาโมเดลปัญญาประดิษฐ์สำหรับการจำแนกการกระทำของมนุษย์ภายในสำนักงาน

ศึกษาค้นคว้าเอกสารและงานวิจัยที่เกี่ยวข้อง

- ศึกษาเกี่ยวกับการประมวลผลวิดีโอ
 - ศึกษาเกี่ยวกับชุดข้อมูลสำหรับการประมวลผลวิดีโอ
 - ศึกษาเกี่ยวกับโมเดลปัญญาประดิษฐ์ที่ใช้ในการประมวลผลวิดีโอ
 - ศึกษาเครื่องมือที่ใช้ในการช่วยสร้างชุดข้อมูลจากวิดีโอ (labeling tool)

เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

1. ออกแบบและสร้างหน้าต่างของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์
 2. ออกแบบและสร้างระบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์
 3. ทดสอบและปรับปรุงการทำงานของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

ไม่เดลปัญญาประดิษฐ์สำหรับการจำแนกการกระทำของนุชช์ภายใต้สำนักงาน

- สร้างชุดข้อมูลสำหรับสร้างโมเดลปัญญาประดิษฐ์จากเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์
 - สร้างโมเดลปัญญาประดิษฐ์สำหรับการจำแนกการกระทำของมนุษย์ภายในสำนักงาน
 - ทดสอบโมเดลปัญญาประดิษฐ์สำหรับการจำแนกการกระทำของมนุษย์ภายในสำนักงาน

แผนการดำเนินงาน

#2562

| ม.ย. | ก.ค. | ส.ค. | ก.ย. | ต.ค. | พ.ย. |
|--------------------|------|--------------------|------|------|------|
| #ช่วงการทำงานที่ 1 | | | | | |
| | | #ช่วงการทำงานที่ 2 | | | |
| | 1. | | | | |
| | 2. | | 3. | | |
| | | | 4. | | |
| | | #ช่วงการทำงานที่ 3 | | | |
| | 5. | | | | |
| | | | | 6. | |

รูปที่ 1.1: แผนการดำเนินงาน

รายละเอียดแผนดำเนินงาน

1. ช่วงการทำงานที่ 1 : ศึกษาและรวบรวมข้อมูล

(a) 3 มิถุนายน - 15 กรกฎาคม 2562

ศึกษาข้อมูลเกี่ยวกับการวิเคราะห์วิดีโอด้วยปัญญาประดิษฐ์และการสร้างชุดข้อมูลสำหรับจำแนกการกระทำของมนุษย์

2. ช่วงการทำงานที่ 2 : ออกแบบและพัฒนาเครื่องมือสำหรับกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

(a) 16 กรกฎาคม - 1 สิงหาคม 2562

ออกแบบและสร้างหน้าต่างการทำงานของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

(b) 16 กรกฎาคม - 15 สิงหาคม 2562

ออกแบบและสร้างระบบการทำงานของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

(c) 1 สิงหาคม - 15 กันยายน 2562

รวมระบบเข้ากับหน้าต่างการทำงานของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

(d) 16 สิงหาคม - 15 ตุลาคม 2562

ปรับปรุงระบบและหน้าต่างการทำงานของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

3. ช่วงการทำงานที่ 3 : สร้างและทดสอบโมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำการของมนุษย์

(a) 16 ตุลาคม - 11 พฤศจิกายน 2562

สร้างชุดข้อมูลสำหรับโมเดลปัญญาประดิษฐ์จากเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

(b) 21 ตุลาคม - 22 พฤศจิกายน 2562

สร้างและทดสอบโมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำการของมนุษย์กับชุดข้อมูลที่ได้จากการเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

บทที่ 2

ทฤษฎี/การวิจัยที่เกี่ยวข้อง

การประมวลผลวิดีโอในปัจจุบันนั้นมีวิธีการและเทคนิคมาหลาย ผู้วิจัยจึงต้องศึกษาองค์ความรู้และงานวิจัยที่เกี่ยวข้องกับวัตถุประสงค์ของงาน เพื่อศึกษาและใช้เป็นแนวทางในการประยุกต์สำหรับสร้างเครื่องมือกำกับข้อมูลด้วยปัญญาประดิษฐ์ และโมเดลปัญญาประดิษฐ์สำหรับการจำแนกการกระทำของมนุษย์ ซึ่งหัวข้อที่ผู้วิจัยได้ไปศึกษามา มีดังต่อไปนี้

2.1. ทฤษฎีพื้นฐาน

2.1.1. Convolutional neural network

2.1.2. Optical flow

2.1.3. โมเดลปัญญาประดิษฐ์สำหรับตรวจจับวัตถุที่เกี่ยวข้องกับงานวิจัย

2.2. การประมวลผลวิดีโอ

2.2.1. การตรวจจับวัตถุ (object detection)

2.2.2. การติดตามตำแหน่งของวัตถุ (object tracker)

2.2.3. การระบุตัวตนของบุคคล (person re-identification)

2.2.4. การจำแนกการกระทำการของมนุษย์ (action classification)

2.3. ชุดข้อมูลสำหรับการประมวลผลวิดีโอ

2.3.1. ชุดข้อมูล YouTube-8M

2.3.2. ชุดข้อมูล Atomic visual action (AVA)

2.3.3. ชุดข้อมูล Moments in Time

2.4. โมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำการ

2.4.1. ResNet

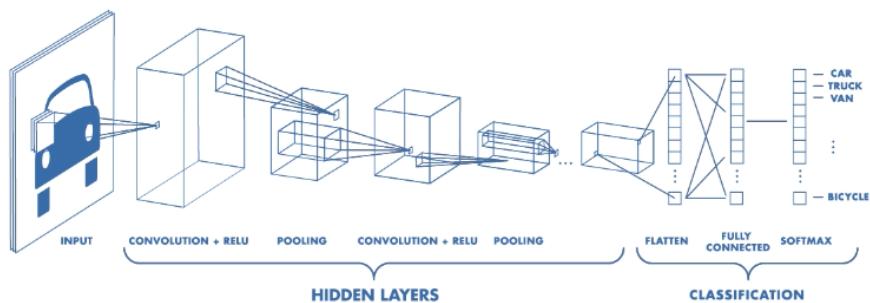
2.4.2. Inflated 3D convolutional network

2.5. เครื่องมือกำกับชุดข้อมูลภาพ

2.1 ທຖ່ງກົມື້ພື້ນຖານ

2.1.1 Convolutional neural network

Convolution neural network (CNN)^[1] คือโมเดลปัญญาประดิษฐ์ประเกทหนึ่งมักจะนำมาใช้กับงานที่เกี่ยวกับการจำแนกวัตถุในภาพ เช่น แมว หมา มนุษย์ รถ เป็นต้น การที่ CNN สามารถจำแนกภาพออกมานี้ได้ว่าเป็นหมวดหมู่อะไรนั้นต้องผ่านชั้นตัวกรอง (filter) หรือเคลอร์เนล (kernel), pooling layer, fully connected layer และใช้ softmax หรือ logistic function เพื่อสกัดคุณลักษณะ (extract feature) สำหรับใช้ในการจำแนกว่าเป็นวัตถุหมวดหมู่อะไร ดังรูปที่ 2.1



รูปที่ 2.1: ตัวอย่างโครงสร้างของ CNN ที่ใช้ในการจำแนกหมวดหมู่ของวัตถุ^[1]

2.1.1.1 ตัวกรอง/เครื่องเนล

ตัวกรองหรือเครื่องเนล คือชั้นที่ใช้ในการสกัดคุณลักษณะของรูปภาพออกมาด้วยสีเหลี่ยมเล็กๆขนาด NxN โดยที่ $N \in [1, 2, 3, \dots]$ ดังรูปที่ 2.2ก และสมมติให้ภาพที่ใช้ในการประมวลผลเป็นดังรูปที่ 2.2ข

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 1 |

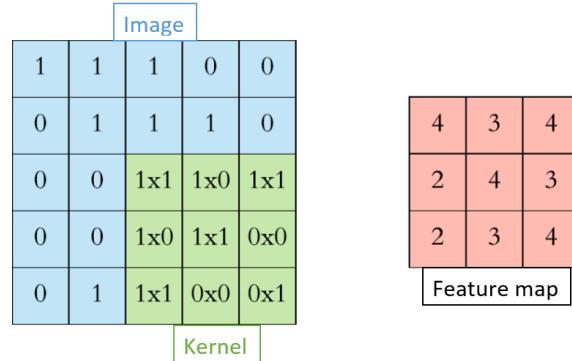
(ก) ตัวอย่างเคอร์เนลขนาด 3×3

| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 0 | 0 |
| 0 | 1 | 1 | 1 | 0 |
| 0 | 0 | 1 | 1 | 1 |
| 0 | 0 | 1 | 1 | 0 |
| 0 | 1 | 1 | 0 | 0 |

(ข) ตัวอย่างภาพที่ใช้ในการประเมินผล

รูปที่ 2.2: ตัวอย่างเครื่องเนล และภาพที่ใช้ในการประเมินผล

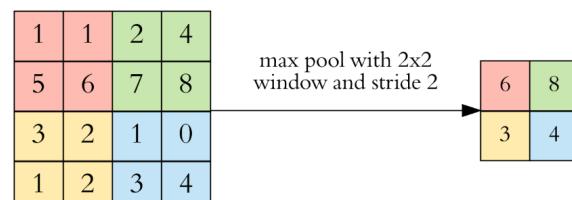
เมื่อนำเครื่องเนล (รูปที่ 2.2ก) ไปทาบกับภาพ (รูปที่ 2.2ข) และคุณค่าในเครื่องเนลกับพิกเซล (pixel) ที่ทาบจะได้คุณลักษณะของช่องนั้นจากนั้นเลื่อนต่อไปจนครบทั้งรูป ซึ่งระยะในการเลื่อนนั้นขึ้นอยู่กับผู้สร้างว่าต้องการจะให้เลื่อนครั้งละกี่ช่อง แต่ระยะการเลื่อนที่มากขึ้นจะทำให้ความสัมพันธ์ของคุณลักษณะที่ได้ออกมาแน่นอยู่ลง โดยการวางแผนที่ใช้ในภาพนั้นจะวางไม่ให้เกินกรอบรูป แต่ถ้าต้องการทابกับทุกพิกเซลในภาพสามารถทำได้ด้วยการให้พื้นที่ที่เกินขอบภาพไปเพ่ากับ 0 เทคนิคนี้เรียกว่า padding และคุณลักษณะที่ได้ออกมาทั้งหมดจะเรียกว่าผังคุณลักษณะ (features map) ตามรูปที่ 2.3



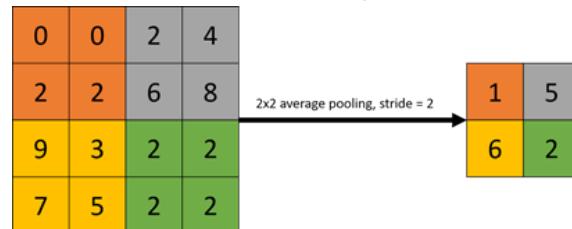
รูปที่ 2.3: ตัวอย่างการหาผังคุณลักษณะ^[1]

2.1.1.2 Pooling

Pooling คือขั้นที่สามารถลดขนาดของภาพลงเพื่อลดข้อมูลที่ไม่จำเป็นลง ซึ่งมีหลายประเภทแต่นิยมใช้มีสองประเภทได้แก่ max pooling และ average pooling โดยที่ max pooling จะใช้ในการหาค่าที่มากที่สุดในเครื่องเนลที่ทาบอยู่ดังรูปที่ 2.4ก ในขณะที่ average pooling จะหาค่าเฉลี่ยของภายในเครื่องเนลออกม�다ังรูปที่ 2.4ข



(ก) ตัวอย่างการทำ max pooling^[1]



(ข) ตัวอย่างการทำ average pooling^[2]

รูปที่ 2.4: ตัวอย่างการใช้ max pooling และ average pooling กับภาพ

2.1.1.3 Activation function^[3]

ในแต่ละชั้นของ CNN นั้นจะมี activation function เป็นสิ่งที่กำหนดว่าผลลัพธ์ของชั้นนั้นว่าจะอยู่ในช่วงไหน เช่น -1 ถึง 1, 0 ถึง 1 ขึ้นอยู่กับฟังก์ชันที่ใช้ ซึ่งจะแบ่งออกเป็นสองแบบคือ แบบเป็นเส้นตรงและแบบไม่เป็นเส้นตรง

แบบเป็นเส้นตรงจะมีสมการของฟังก์ชันดังนี้ $f(x) = x$ โดยที่ x คือข้อมูลป้อนเข้าของฟังก์ชัน ทำให้ผลลัพธ์ที่ได้จากฟังก์ชันนี้มีค่าอยู่ในช่วง $-\inf$ ถึง \inf ส่วนในแบบไม่เป็นเส้นตรงจะมีหลายฟังก์ชัน แต่ฟังก์ชันที่เป็นที่นิยมใช้คือ Rectified Linear Unit หรือ ReLU, Leaky ReLU และ Parametric ReLU โดยทั้งสามฟังก์ชันมีสมการดังนี้

ReLU

$$f(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{otherwise} \end{cases}$$

Leaky ReLU

$$f(x) = \begin{cases} 0.01x & \text{if } x < 0, \\ x & \text{otherwise} \end{cases}$$

Parametric ReLU โดยที่ $\alpha \in \mathbb{R}^+$

$$f(x) = \begin{cases} \alpha x & \text{if } x < 0, \\ x & \text{otherwise} \end{cases}$$

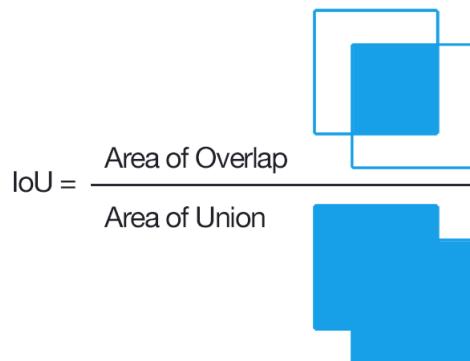
2.1.1.4 Fully connected layer

เป็นชั้นที่จะรวมคุณลักษณะทั้งหมดของชั้นก่อนหน้าให้เป็นเวกเตอร์ (vector) ก่อนที่จะนำไปคำนวณด้วย activation function เพื่อหาคำตอบสำหรับการแยกหมวดหมู่ของภาพ ซึ่งฟังก์ชันที่นิยมใช้จะมีสองฟังก์ชัน คือ softmax และ sigmoid (logistic) โดยทั้งสองมีความแตกต่างกันดังตารางที่ 2.1

| | Softmax | Sigmoid |
|---|---|--|
| 1 | สมการคือ $f(x_i) = \frac{\exp(x_i)}{\sum_{j=0}^k \exp(x_j)}$ โดยที่ k คือจำนวนข้อมูล และ $i \in 0, 1, 2, \dots, k$ | $f(x_i) = \frac{1}{1+\exp(-x_i)}$ โดยที่ $i \in 0, 1, 2, \dots, k$ |
| 2 | ผลรวมของความน่าจะเป็นจะเท่ากับ 1 เสมอ | ผลรวมของความน่าจะเป็นไม่จำเป็นต้องเท่ากับ 1 |
| 3 | มักใช้ในการจำแนกหมวดหมู่มากกว่าสองหมวดหมู่ขึ้นไป | มักใช้ในการจำแนกหมวดหมู่เพียงสองหมวดหมู่ |

ตารางที่ 2.1: ตารางแสดงการเปรียบเทียบระหว่าง softmax function และ sigmoid function

2.1.1.5 Intersection over union (IoU)



รูปที่ 2.5: ภาพแสดงการหา IoU ด้วยการใช้ค่าต่อของจักริ่งของการทำงานและกรอบสี่เหลี่ยมที่ได้จากการทำงาน^[4]

เป็นวิธีในการทดสอบประสิทธิภาพของการตรวจจับวัตถุ โดยค่า IoU นั้นสามารถหาได้จากการนำกรอบสี่เหลี่ยมจริงของวัตถุและกรอบสี่เหลี่ยมที่ได้จากการทำงาน มาหาอัตราส่วนระหว่างพื้นที่ที่กรอบสี่เหลี่ยมทั้งสองทับซ้อนกัน และหารด้วยพื้นที่ทั้งหมดของกรอบสี่เหลี่ยมทั้งสองรวมกัน ซึ่งสามารถเขียนในรูปสมการได้ดังนี้

$$IoU(P, G) = \frac{|P \cap G|}{|P \cup G|} \quad (2.1)$$

โดยที่

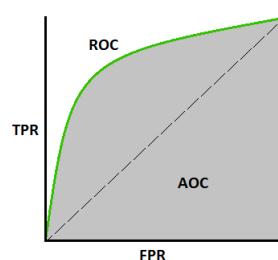
IoU = ค่าที่ใช้สำหรับวัดผลความใกล้เคียงระหว่างสองกรอบสี่เหลี่ยม

P = พื้นที่ของกรอบสี่เหลี่ยมที่ทำงานได้

G = พื้นที่ของกรอบสี่เหลี่ยมจริงของรูปภาพ

2.1.1.6 Area under the curve (AUC) - Receiver Operating Characteristics Curve (ROC)

AUC - ROC^[5] เป็นวิธีในการทดสอบประสิทธิภาพการแยกแยะของโมเดลปัญญาประดิษฐ์ โดยที่ ROC คือ เส้นโค้งของความน่าจะเป็น และ AUC คือค่าที่บ่งบอกถึงความสามารถในการแยกแยะ ค่า AUC จะอยู่ในช่วง 0 - 1 ถ้าค่า AUC ใกล้เคียง 1 มากเท่าไหรก็จะหมายถึงความสามารถในการแยกแยะของโมเดลปัญญาประดิษฐ์สูงตามไปด้วย



รูปที่ 2.6: AUC - ROC Curve^[5]

จากรูปที่ 2.6 เสน่โค้ง ROC สามารถหาได้ด้วยการนำค่า True positive rate (TPR) และ False positive rate (FPR) มาสร้างเป็นกราฟ โดยค่า TPR จะอยู่ในแกน y และค่า FPR จะอยู่ในแกน x ซึ่งสามารถหาค่าของ TPR และ FPR ได้ดังนี้

$$TPR = \frac{TP}{TP + FN} \quad (2.2)$$

โดยที่

TP = จำนวนของข้อมูลที่มีผลลัพธ์เป็นจริงและผลจากโมเดลปัญญาประดิษฐ์เป็นจริง
 FN = จำนวนของข้อมูลที่มีผลลัพธ์เป็นจริงและผลจากโมเดลปัญญาประดิษฐ์เป็นเท็จ

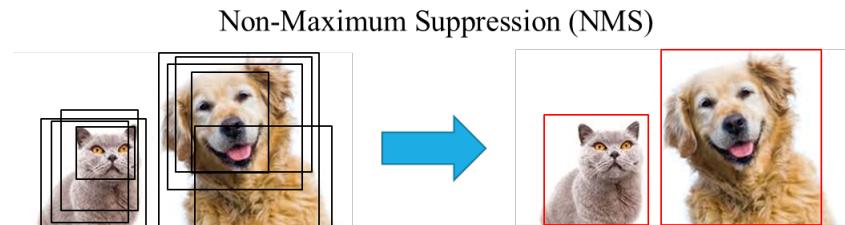
$$FPR = \frac{FP}{FP + FN} \quad (2.3)$$

โดยที่

FP = จำนวนของข้อมูลที่มีผลลัพธ์เป็นเท็จและผลจากโมเดลปัญญาประดิษฐ์เป็นจริง
 FN = จำนวนของข้อมูลที่มีผลลัพธ์เป็นจริงและผลจากโมเดลปัญญาประดิษฐ์เป็นเท็จ

2.1.1.7 Non Maximum Suppression (NMS)

Non Maximum Suppression^[6] คือ อัลกอริทึม (algorithm) ที่นิยมใช้เข้ามาช่วยจัดการปัญหากรอบสี่เหลี่ยมที่ซ้อนทับกันซึ่งเกิดจากการทำงานย้ำๆ เพื่อให้ได้กรอบสี่เหลี่ยมที่บ่งบอกถึงตำแหน่งของวัตถุนั้นเพียงกรอบเดียว



รูปที่ 2.7: ตัวอย่างการทำงานของ NMS^[7]

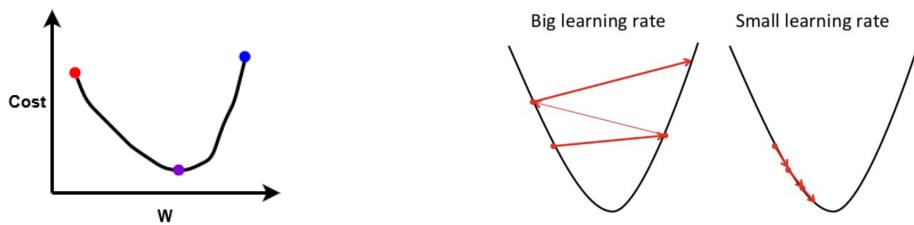
2.1.1.8 Optimization algorithm

ในขั้นตอนการสร้างโมเดลปัญญาประดิษฐ์นั้นสิ่งที่สำคัญคือ การลดความผิดพลาด (error) หรือค่าค่า kos (cost) นั้นทำให้ต้องมีเครื่องมือสำหรับปรับพารามิเตอร์ของโมเดลเพื่อให้มีความผิดพลาดน้อยที่สุด นั่นทำให้มีการคิดค้น optimization algorithm ต่างๆขึ้นมา ตัวอย่างเช่น gradient descent (GD), Momentum, RMS Prop และ Adam เป็นต้น ซึ่งเมทริกซ์ weight มักจะสูงค่าในตอนต้น แล้วใช้ optimization algorithm ในการลดค่า kos ที่ได้จาก cost function พร้อมทั้งหาเมทริกซ์ weight และค่าอคติ (bias) ที่ดีที่สุด โดยจากตัวอย่างที่ยกไป ก่อนหน้านี้แต่ละอัลกอริทึมนั้นมีรายละเอียดดังนี้

1. Gradient descent (GD) เป็นอัลกอริทึมที่จะทำการปรับเมทริกซ์ weight และค่าอคติของทุกข้อมูลที่ใช้ในการสร้างโมเดล โดยจะมีสมการดังนี้

$$W = W - \alpha \frac{\partial J(W)}{\partial W} \quad (2.4)$$

$$b = b - \alpha \frac{\partial J(b)}{\partial b} \quad (2.5)$$



(ก) กราฟสำหรับอธิบายการทำงานของ gradient descent (ข) ตัวอย่างผลลัพธ์ของการปรับค่าอัตราการเรียนรู้ที่ต่างกัน

รูปที่ 2.8: รูปสำหรับอธิบายการทำงานของ gradient descent

โดยที่

 W = เมทริกซ์ weight α = อัตราการเรียนรู้ (learning rate) J = cost function b = ค่าออดีต

สมการแรกนี้คือสมการการปรับเมทริกซ์ weight ในขณะที่สมการที่สองคือสมการปรับค่าออดีต ซึ่งตั้งสองสมการนี้จะมีอัตราการเรียนรู้เป็นตัวกำหนดอัตราการเปลี่ยนแปลง จากรูปที่ 2.8 ก จุดสีแดงหมายถึงกรณีที่ weight มีค่าน้อยกว่าค่าที่เหมาะสมในจุดสีม่วง จากรูปจะเห็นว่าความชัน ณ จุดสีแดงมีค่าเป็นลบทำให้เมื่อนำไปใส่ในสมการแล้วจะทำให้มีการปรับ weight ให้เพิ่มขึ้น ในทางตรงกันข้ามที่จุดสีน้ำเงินจะมีความชันเป็นบวกทำให้มีการปรับ weight ให้น้อยลง ซึ่งการปรับแต่ละครั้งจะมีระยะก้าวว่างขนาดใหญ่จะขึ้นอยู่กับอัตราการเรียนรู้ดังตัวอย่างในรูปที่ 2.8 ข

ข้อเสียของ gradient descent คือจะทำการปรับค่าพารามิเตอร์หลังจากการคำนวณครบทุกข้อมูลในชุดข้อมูลที่ใช้ในการสร้างแล้วเท่านั้น ทำให้เกิดปัญหาน่าวัยความจำไม่เพียงพอเมื่อชุดข้อมูลมีขนาดที่ใหญ่มาก จึงทำให้เกิดเทคนิคที่เรียกว่า stochastic gradient descent (SGD) กล่าวคือถ้าหาก gradient descent ทำการคำนวณครบทุกข้อมูลจึงจะมีการปรับพารามิเตอร์ stochastic gradient descent จะคำนวณเพียงข้อมูลหนึ่งตัวหรือเป็นชุดของข้อมูล (subset) ซึ่งหากใช้เป็นชุดของข้อมูลจะเรียกว่า mini-batch stochastic gradient descent การทำแบบนี้จะทำให้ลดการใช้หน่วยความจำลง และทำให้การสร้างโมเดลนั้นทำได้เร็วขึ้นกว่าการใช้ gradient descent

2. Momentum หรือ gradient descent with momentum มีการทำงานเหมือนกับ gradient descent แต่จะมีการเร่งความเร็วในการปรับลดค่าจาก cost function โดยใช้สิ่งที่เรียกว่าค่าเฉลี่ยเคลื่อนที่ (moving average) เพื่อลบข้อมูลรบกวน โดยสามารถหาได้จากสมการดังนี้

$$V_{dW} = \beta V_{dW} + (1 - \beta)dW \quad (2.6)$$

$$V_{db} = \beta V_{db} + (1 - \beta)db \quad (2.7)$$

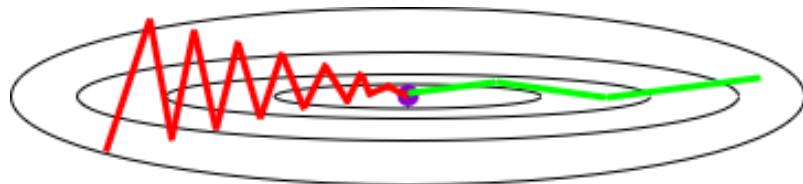
โดยที่

- V_{dW} = moving average ของอนุพันธ์ของเมทริกซ์ weight
 β = พารามิเตอร์ของ moving average (ค่าคงที่ momentum)
 dW = อนุพันธ์ของเมทริกซ์ weight
 V_{db} = moving average ของอนุพันธ์ของค่าอคติ
 db = อนุพันธ์ของค่าอคติ

ในการสร้างโมเดลปัญญาประดิษฐ์นั้นมีเป้าหมายถึงคือการหา cost function ที่เหมาะสมเพื่อให้ได้ค่าความผิดพลาดน้อยที่สุด หากดูจากรูปที่ 2.9 gradient descent จะเป็นไปตามเส้นสีแดง ในขณะที่ momentum จะเป็นเส้นสีเขียว และจุดสีม่วงคือจุดที่มีค่าความผิดพลาดน้อยที่สุด จะเห็นว่า gradient descent แกะง่ายในแกน y แต่มีการเคลื่อนที่ในแกน x เพียงเล็กน้อย วิธีแก้คือทำให้การแกะง่ายในแกน y น้อยลง นั่นทำให้มีการใช้ค่าเฉลี่ยเคลื่อนที่เข้ามาช่วย ทำให้สมการปรับ weight และค่าอคติเปลี่ยนเป็นดังสมการด้านล่าง

$$W = W - \alpha V_{dW} \quad (2.8)$$

$$b = b - \alpha V_{db} \quad (2.9)$$



รูปที่ 2.9: เปรียบเทียบการทำงานของ gradient descent กับ Momentum

3. RMS Prop หรือ root-mean-square propagation ซึ่งมีความคล้ายคลึงกับ Momentum หากดูจากรูปที่ 2.9 กำหนดให้แกน y เป็นค่าอคติ แกน x คือ weight และเส้นสีเขียวเป็นของ RMS Prop ในขณะที่ momentum ใช้ค่าเฉลี่ยเคลื่อนที่ในการลดการแกะง่ายในแนวแกน y RMS Prop จะใช้ค่าเฉลี่ยกำลังสองมหาศาลเพื่อลดการแกะง่ายในแกน y ซึ่งทำให้ต้องมีค่าคงที่ ϵ เพื่อไม่ให้เกิดเหตุการณ์หารด้วยศูนย์ ซึ่งสมการปรับ weight และค่าอคติจะเป็นดังนี้

$$W = W - \alpha \frac{dW}{\sqrt{S_{dW}} + \epsilon} \quad (2.10)$$

$$b = b - \alpha \frac{db}{\sqrt{S_{db}} + \epsilon} \quad (2.11)$$

โดยที่ S_{dW} และ S_{db} สามารถหาได้จากสมการที่ 2.12 และ 2.13 ตามลำดับ

$$S_{dW} = \beta S_{dW} + (1 - \beta) dW^2 \quad (2.12)$$

$$S_{db} = \beta S_{db} + (1 - \beta) db^2 \quad (2.13)$$

4. Adam หรือ Adaptive momentum เป็นอัลกอริทึมที่นำเทคนิคของ momentum และ RMS Prop มารวมกัน ทำให้อัลกอริทึมนี้มีประสิทธิภาพที่สูงและมีความเร็วในการปรับพารามิเตอร์ของโมเดลได้เร็ว ซึ่งได้

มีการใช้เทคนิค error correction เพื่อแก้ปัญหา cold start หมายถึงปัญหาที่ในช่วงแรกๆ นั้นพารามิเตอร์ของ momentum และ RMS Prop นั้นมีค่าห่างจากค่าจริงมากเกินไปทำให้การปรับพารามิเตอร์เกิดขึ้นช้าเกินไป โดยสมการ error correction ของทั้ง momentum และ RMS Prop คือ

$$V_{dW}^{correction} = \frac{V_{dW}}{1 - \beta_1} \quad (2.14)$$

$$V_{db}^{correction} = \frac{V_{db}}{1 - \beta_1} \quad (2.15)$$

$$S_{dW}^{correction} = \frac{S_{dW}}{1 - \beta_2} \quad (2.16)$$

$$S_{db}^{correction} = \frac{S_{db}}{1 - \beta_2} \quad (2.17)$$

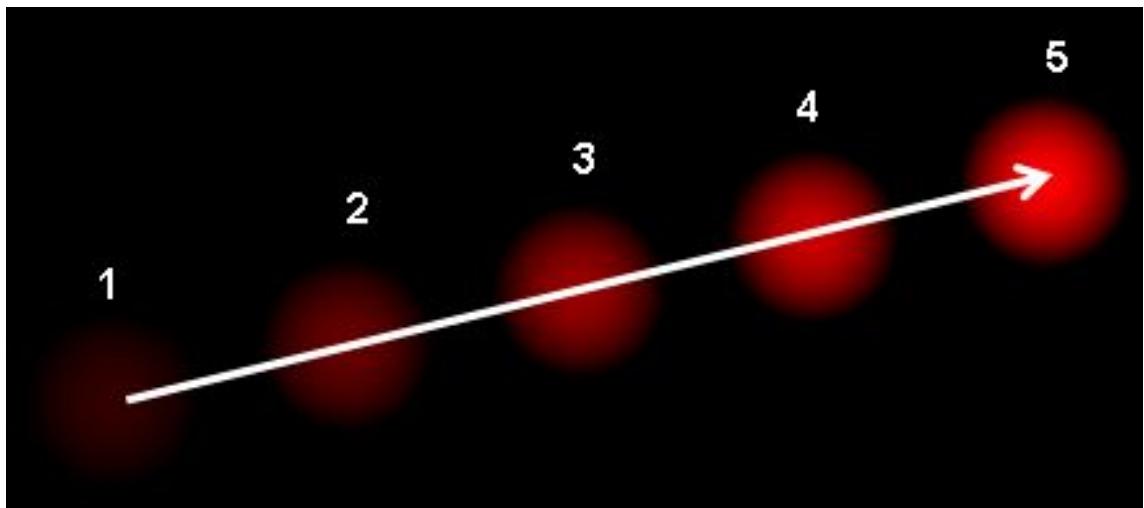
และสมการปรับ weight และค่าอคติของอัลกอริทึมนี้จะเป็นดังสมการด้านล่างนี้

$$W = W - \alpha \frac{V_{dW}^{correction}}{\sqrt{S_{dW}^{correction}}} + \epsilon \quad (2.18)$$

$$b = b - \alpha \frac{V_{db}^{correction}}{\sqrt{S_{db}^{correction}}} + \epsilon \quad (2.19)$$

2.1.2 Optical flow

Optical flow^[8] คือการแปลงการเคลื่อนที่ของวัตถุในระหว่างสองภาพซึ่งอาจเกิดจากการเคลื่อนที่ของวัตถุหรือตัวกล้องอุปกรณ์ในรูปแบบของเวกเตอร์ 2 มิติ โดยที่เวกเตอร์แต่ละตัวจะแสดงถึงทิศทางการเคลื่อนที่ของวัตถุหรือบุคคลระหว่างภาพดังรูปที่ 2.10



รูปที่ 2.10: ตัวอย่าง optical flow ของการเคลื่อนที่ของลูกบอล^[8]

จากรูปที่ 2.10 แสดงให้เห็นถึงการเคลื่อนที่ของลูกบอลในภาพที่ต่อเนื่องกัน 5 ภาพ โดยที่ลูกศรจะแสดงถึงทิศทางการเคลื่อนที่ของเวกเตอร์

การทำงานของ optical flow อุปบนสมมติฐาน 2 ประการได้แก่

1. ความเข้มพิกเซลของวัตถุจะไม่เปลี่ยนแปลงระหว่างภาพที่ต่อเนื่องกัน
2. พิกเซลที่อยู่ใกล้กันจะมีลักษณะการเคลื่อนไหวที่คล้ายกัน

เมื่อพิจารณาพิกเซล $I(x, y, t)$ จากภาพแรกจะเคลื่อนไหวเป็นระยะทาง (dx, dy) ไปยังภาพต่อไปหลังจากเวลาผ่านไปแล้ว dt ดังนั้นเนื่องจากพิกเซลเหล่านี้เหมือนกัน และความเข้มไม่มีการเปลี่ยนแปลง จึงทำให้พูดได้ว่า

$$I(x, y, t) = I(x + dx, y + dy, t + dt) \quad (2.20)$$

โดยที่

I = พิกเซลจากภายในภาพ

x = ตำแหน่งของพิกเซลในแกน x

dx = ระยะทางที่เคลื่อนที่ในแกน x

y = ตำแหน่งของพิกเซลในแกน y

dy = ระยะทางที่เคลื่อนที่ในแกน y

t = เวลา

dt = ระยะเวลาที่เปลี่ยนไประหว่างภาพ

จากนั้นใช้การประมาณค่าของ taylor series ทางฝั่งขวามือ และลบค่า common term แล้วหารด้วย dt เพื่อให้ได้สมการดังต่อไปนี้

$$f_x u + f_y v + f_t \quad (2.21)$$

$$f_x = \frac{\delta f}{\delta x}; f_y = \frac{\delta f}{\delta y} \quad (2.22)$$

$$u = \frac{\delta x}{\delta t}; v = \frac{\delta y}{\delta t} \quad (2.23)$$

โดยที่

f_x = เกรเดียน (gradient) ในแกน x

f_y = เกรเดียนในแกน y

f_t = เกรเดียนของเวลา

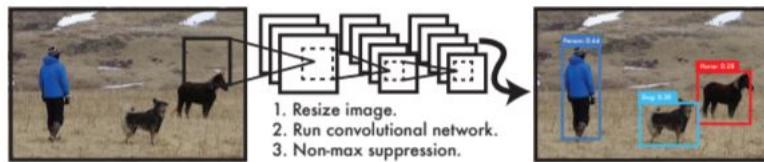
u = เวกเตอร์การเคลื่อนที่ของแกน x

v = เวกเตอร์การเคลื่อนที่ของแกน y

สมการข้างบนเรียกว่าสมการ optical flow จากสมการทำให้สามารถหา f_x และ f_y เป็นเกรเดียนของภาพในแกน x และแกน y ตามลำดับ และ f_t เป็นเกรเดียนของเวลา แต่ n กับ b เป็นตัวแปรที่ไม่ทราบ ทำให้สมการนี้ไม่สามารถแก้ไขด้วยมีตัวแปรที่ไม่ทราบถึง 2 ตัว จึงมีการนำวิธีการต่างๆเข้ามาใช้ในการแก้ปัญหานี้ โดยวิธีการที่นำเข้ามาใช้ในการแก้ปัญหา คือ อัลกอริทึมของ Gunner Farneback^[9] ซึ่งใช้หลักการ polynomial expansion เป็นการประมาณค่าของพื้นที่บางส่วนที่อยู่รอบ ๆ ของแต่ละพิกเซลด้วยสมการพหุนาม โดย optical flow เป็นหนึ่งในวิธีการสกัดข้อมูลการเคลื่อนไหวของวิดีโอออกแบบเพื่อใช้เป็นข้อมูลในการสร้างโมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำหรือเหตุการณ์ในวิดีโอ

2.1.3 โมเดลปัญญาประดิษฐ์สำหรับตรวจจับวัตถุที่เกี่ยวข้องกับงานวิจัย

YOLO



รูปที่ 2.11: กระบวนการทำงานของโครงสร้างโมเดลปัญญาประดิษฐ์ของ YOLO^[10]

โครงสร้างโมเดลปัญญาประดิษฐ์ของ YOLO^[10] เป็นโครงสร้างที่มีความเร็วในการประมวลผลถึง 45 เฟรมต่อวินาที ทำให้สามารถประมวลผลแบบเรียลไทม์ได้ นอกจากนั้นยังมีความแม่นยำ mAP มากกว่าโมเดลสำหรับตรวจจับวัตถุอื่นๆ ถึง 2 เท่า ซึ่งเหตุผลที่โครงสร้างโมเดลปัญญาประดิษฐ์ของ YOLO เร็วกว่าโมเดลปัญญาประดิษฐ์ตัวอื่นๆ เนื่องจากการตรวจจับวัตถุในวิธีการก่อนหน้าจะใช้วิธีทำนายกรอบสี่เหลี่ยมก่อน แล้วจึงค่อยนำกรอบสี่เหลี่ยมไปทำนายว่าเป็นหมวดหมู่อะไร ซึ่ง YOLO มีวิธีการที่ต่างออกไป คือ ติดตามการเคลื่อนไหวตำแหน่งของกรอบสี่เหลี่ยมและทำนายว่าเป็นหมวดหมู่อะไรร่วมกัน โดยใช้โครงข่ายประสาทแบบคอนโวลูชัน ด้วยแนวคิดนี้ จึงเป็นที่มาของชื่อ YOLO หรือ you only look once

โครงสร้างของโมเดลปัญญาประดิษฐ์ของ YOLO

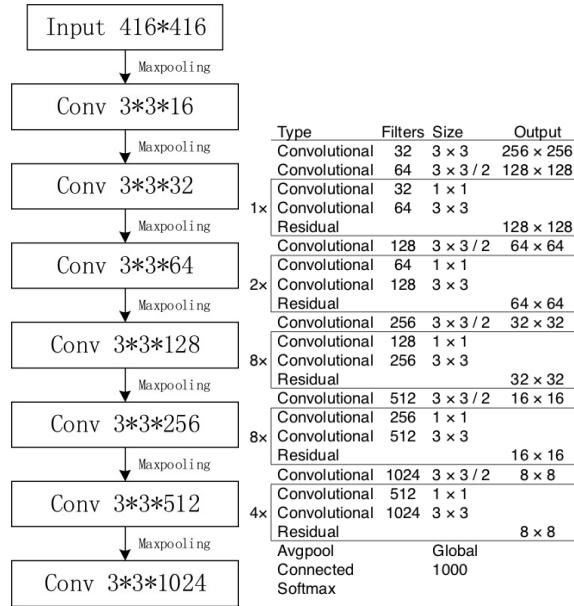


รูปที่ 2.12: โครงสร้างทั่วไปของโมเดลปัญญาประดิษฐ์ของ YOLO^[11]

จากรูปภาพที่ 2.12 จะเห็นได้ว่า YOLO ใช้โครงข่ายประสาทเทียมเพียงตัวเดียวซึ่งภายในโครงข่ายจะมีกระบวนการหลักๆอยู่สามอย่าง กระบวนการแรกคือการสกัดคุณลักษณะ กระบวนการนี้จะมีจำนวนขั้นของโมเดลปัญญาประดิษฐ์ที่แตกต่างกันไปตามความลึกของการสกัดแล้วแต่โมเดล ซึ่งตัวอย่างจะเป็นดังรูปที่ 2.13 และขั้นตอนถัดมาคือการทำนายผล หลังจากที่ได้คุณลักษณะมาแล้วจะนำไปทำนายผลผ่านชั้น fully connected ซึ่งจะได้ผลลัพธ์เป็นหมวดหมู่และตำแหน่งของกรอบสี่เหลี่ยม และขั้นตอนสุดท้ายคือการทำ NMS เพื่อให้ได้ผลลัพธ์ที่ดีที่สุดของมา

ซึ่งโครงสร้างโมเดลปัญญาประดิษฐ์ของ YOLO ที่ถูกใช้ในงานวิจัยนี้ประกอบไปด้วย YOLO-v3 tiny, YOLO-v3 และ YOLO-v3 spp ซึ่งทั้งสามโครงสร้างจะแตกต่างดังนี้

1. YOLO-v3 tiny ใช้ชั้น max pooling ในขั้นตอนของการลดจำนวนข้อมูลตัวอย่าง
2. YOLO-v3 ใช้ชั้นคอนโวลูชัน ในขั้นตอนของการลดจำนวนข้อมูลตัวอย่าง
3. YOLO-v3 spp ใช้ชั้นคอนโวลูชัน และคุณลักษณะที่ดีที่สุดของ max pooling ในขั้นตอนของการลดจำนวนข้อมูลตัวอย่าง

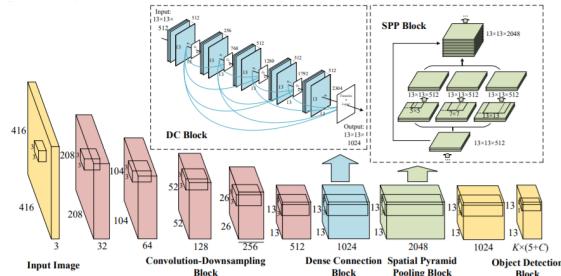


(ก) โครงสร้างโมเดล

ปัญญาประดิษฐ์ของ
YOLO-v3 tiny^[12]

Table 1. Darknet-53.

(ข) โครงสร้างโมเดลปัญญาประดิษฐ์
ของ YOLO-v3^[13]

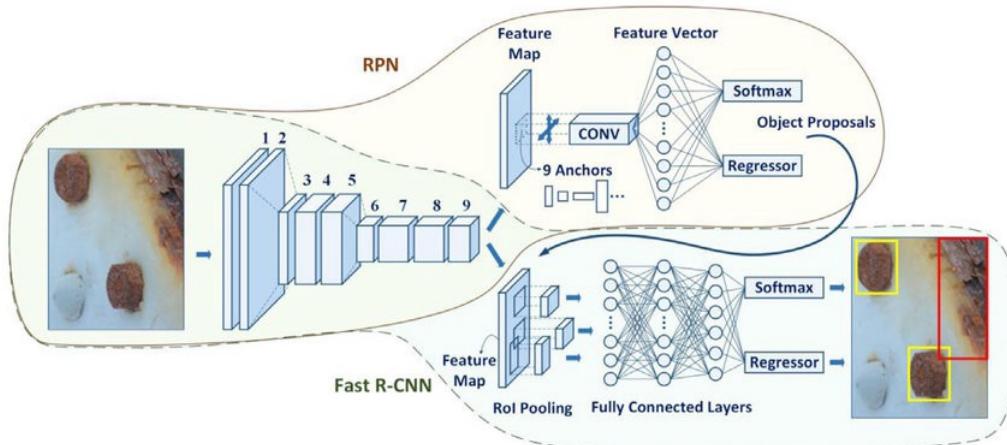


11

(ค) โครงสร้างโมเดลปัญญาประดิษฐ์ของ YOLO-v3 spp^[14]

รูปที่ 2.13: โครงสร้างโมเดลปัญญาประดิษฐ์ของ YOLO

Faster-RCNN



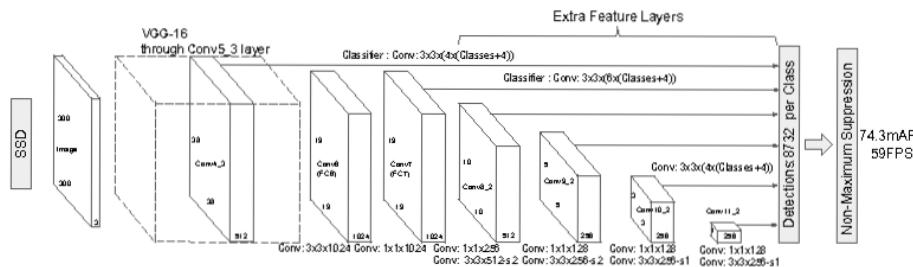
รูปที่ 2.14: โครงสร้างทั่วไปของโมเดลปัญญาประดิษฐ์ของ Faster RCNN^[15]

Faster-RCNN^[16] มีการพัฒนาในการหาพื้นที่ที่สนใจ (region of interest หรือ ROI) โดยเปลี่ยนจากใช้โครงข่ายหาพื้นที่ที่สนใจแยกเฉพาะ (selective search) นำมารวมในโครงข่ายเดียวกัน ดังนั้น Faster-RCNN จึงมีโครงข่ายประสาทเทียม (neural network) เดียวในการทำงาน ซึ่งภายในโครงข่ายจะประกอบไปด้วยการทำงานหลักสามอย่าง คือ

1. การสกัดคุณลักษณะ
นำภาพเข้าชั้นคอนโวลูชันเพื่อการสกัดคุณลักษณะของภาพ
2. การหาพื้นที่ที่คาดว่าจะมีวัตถุอยู่
หลังจากที่ภาพผ่านการสกัดคุณลักษณะแล้ว จะถูกนำเข้าไปใน region proposal network เพื่อหาพื้นที่ที่คาดว่าจะมีวัตถุอยู่
3. การทำนายผล
ทำการ pooling คุณลักษณะของภาพและพื้นที่ที่คาดว่าจะมีวัตถุอยู่ และนำเข้าไปในชั้น fully connected จะได้ผลลัพธ์เป็นหมวดหมู่ของวัตถุและตำแหน่งของกรอบสี่เหลี่ยม

Region proposal network (RPN)^[16] คือ โครงข่ายที่หาพื้นที่ที่คาดว่าจะมีวัตถุอยู่จะถูกใช้หลังภาพผ่านการสกัดคุณลักษณะ ซึ่ง RPN มีโครงสร้างที่มีองค์ประกอบ 2 อย่าง คือมีการบอกว่าบริเวณนั้นมีวัตถุอยู่หรือไม่ และสำหรับการระบุพิกัดของกรอบสี่เหลี่ยมที่คาดว่าจะมีวัตถุอยู่ ซึ่งผลลัพธ์จะได้เป็น ROI ของวัตถุในภาพ

SSD



รูปที่ 2.15: โครงสร้างทั่วไปของโมเดลปัญญาประดิษฐ์ SSD^[11]

SSD^[17] เป็นโมเดลปัญญาประดิษฐ์ที่ใช้โครงข่ายประสาทเทียมตัวเดียวสำหรับการตรวจจับวัตถุ ซึ่งภายในโครงข่ายจะประกอบไปด้วยการทำงานหลัก 3 อย่าง คือ

1. การสกัดคุณลักษณะ
นำภาพผ่าน VGG-16^[18] (โมเดล CNN ชนิดหนึ่ง) เพื่อการสกัดคุณลักษณะของภาพอุปกรณ์
2. การทำนายผล
หลังจากที่ได้คุณลักษณะมาแล้วจะนำไปทำนายผลในชั้น fully connected
3. การเลือกคัดกรองผลลัพธ์
หลังจากได้ผลลัพธ์เป็นประเภทของวัตถุ และตำแหน่งของกรอบสี่เหลี่ยมจะนำไปผ่านกระบวนการ NMS เพื่อให้ได้ผลลัพธ์ที่ดีที่สุด

2.2 การประมวลผลวิดีโอ

ในส่วนของงานวิจัยนี้สิ่งที่สนใจ คือ ข้อมูลการกระทำของมนุษย์แต่ละคนภายในวิดีโอ เพื่อที่จะได้ผลลัพธ์ที่มีประสิทธิภาพอุปกรณ์เป็นข้อมูลของสิ่งที่สนใจ เช่น จำนวนคนที่เดินผ่านกล้อง หรือทิศทางการเดินของคนในวิดีโอ จึงจำเป็นต้องใช้การประมวลผลวิดีโอเพื่อที่จะสกัดสิ่งที่สนใจออกจากวิดีโอ ซึ่งการประมวลผลวิดีโอมีหลากหลายกระบวนการ โดยในแต่ละกระบวนการจะมีจุดประสงค์ของการทำและผลลัพธ์หลังการประมวลผลที่แตกต่างกัน ในหัวข้อนี้จะมาอธิบายถึงกระบวนการในการประมวลผลของวิดีโอและผลลัพธ์ของกระบวนการนั้น

2.2.1 การตรวจจับวัตถุ

การตรวจจับวัตถุนั้นเป็นหนึ่งในกระบวนการประมวลผลวิดีโอ^[19] กล่าวคือกระบวนการที่ผู้วิจัยจะต้องทำการระบุสิ่งที่สนใจว่า คืออะไร อยู่ที่ตำแหน่งใด การตรวจจับวัตถุถูกค้นพบเมื่อเวลาแล้ว และในปัจจุบันนั้นสามารถทำได้หลากหลายวิธี โดยภายในบทความนี้จะสรุปให้ความสำคัญของวิธีการต่างในการตรวจจับวัตถุ เช่น Sliding window, Brute force search, RCNN, Fast-RCNN, Faster-RCNN, YOLO, SSD

1. Sliding window วิธีการที่เบริ่ยบเสมือนมีเครื่องเนลค่อนยาเลื่อนไปยังแต่ละพิกเซลบนภาพ ซึ่งก่อนการเลื่อนของเครื่องเนลแต่ละครั้งจะนำส่วนของภาพที่ถูกเครื่องเนลทับอยู่ไปทำนายว่าใช่วัตถุที่เราต้องการหรือไม่ จากนั้นจึงค่อยเลื่อนต่อไปจนครบทั้งภาพ
2. Brute force search ถูกสร้างขึ้นมาเพื่อแก้ปัญหาขนาดของเครื่องเนลไม่ตรงกับขนาดของวัตถุที่อยู่ในภาพทำให้มีโอกาสที่จะไม่พบวัตถุ โดยหลักการของวิธีการนี้คือ การย่อ-ขยายภาพและนำเข้าในหลายๆ อัตราส่วน ตั้งแต่ 0.1 เท่า จนถึง 2 เท่า แต่ข้อเสียของวิธีการนี้คือ มีการคำนวณพื้นที่ช้าๆ ทำให้ใช้เวลานาน

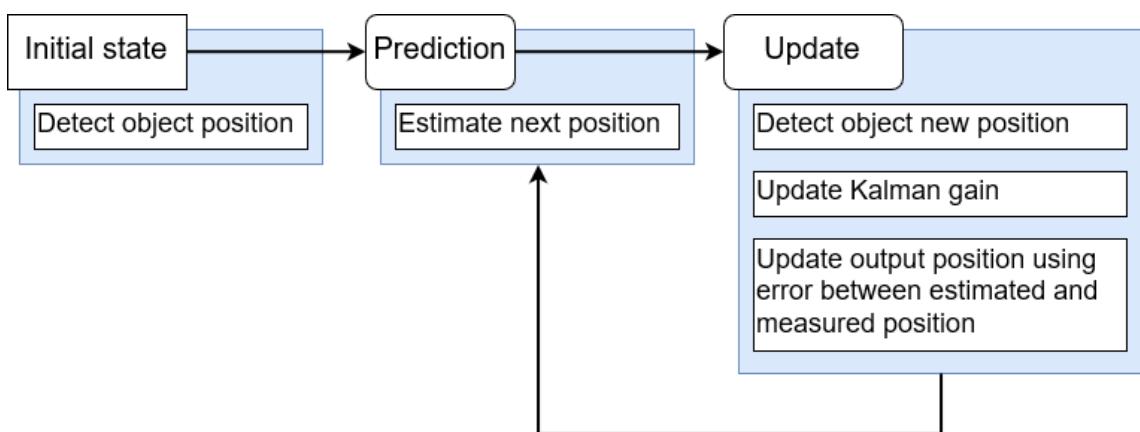
3. RCNN ใช้อัลกอริทึม selective search เข้ามาช่วยในการหาพื้นที่ที่น่าจะมีวัตถุอยู่ทัดแทนการค้นหาทุกๆ ตำแหน่ง จากนั้นก็นำภาพในส่วนพื้นที่นั้นไปทำนายว่าวัตถุนั้นคืออะไร กรณีที่มีพื้นที่ที่อยู่ใกล้ๆ วัตถุอยู่ เช่น象เข้ามาเป็นจำนวนมากด้วย เราจะใช้ NMS ใน การเลือกพื้นที่ที่ถูกทับซ้อนมากที่สุดในบริเวณนั้น
4. Fast-RCNN จากวิธีการ RCNN แต่ละพื้นที่จะถูกนำไปสกัดคุณลักษณะ และทำนายผลที่ละพื้นที่ทำให้เสียเวลา โดย Faster-RCNN จะมีส่วนที่คล้ายกับ RCNN ในส่วนการทำ selective search เพื่อหาพื้นที่ที่น่าจะมีวัตถุเหมือนเดิม แต่ Faster-RCNN จะนำภาพไปสกัดคุณลักษณะ หลังจากที่ได้คุณลักษณะแล้วจะนำพิกัดของพื้นที่ที่น่าจะมีวัตถุ บนภาพที่ถูกสกัดคุณลักษณะแล้วไปผ่านชั้น ROI Pooling (การลดขนาดข้อมูลให้มีขนาดคงที่เพื่อเป็นข้อมูลป้อนเข้าให้กับโมเดลในการทำนายผล)
5. Faster-RCNN พัฒนาจาก Fast-RCNN โดยวิธีของ Faster-RCNN จะรวมในส่วนของ selective search และการทำงานอื่นๆ ให้อยู่ในโครงข่ายเดียวกัน สรุปคือการทำงานของโครงข่ายของ Faster-RCNN จะมีการทำงานสามอย่างคือ 1) สกัดคุณลักษณะ 2) การหาส่วนที่น่าจะมีวัตถุอยู่ในภาพ 3) หลังจากได้ภาพจากการสกัดคุณลักษณะ นำพิกัดของพื้นที่ที่น่าจะมีวัตถุบนภาพที่ถูกสกัดคุณลักษณะแล้วไปผ่านชั้น ROI Pooling
6. YOLO เป็นวิธีการที่ใช้ CNN เพียงตัวเดียวทำงานภาพ โดยโครงข่ายจะแบ่งภาพออกเป็นพื้นที่ และใช้ fully connected การติดตามการเคลื่อนไหวตำแหน่งของกรอบสีเหลี่ยม พร้อมทั้งหมวดหมู่ของวัตถุไปพร้อมกัน
7. SSD ใช้โครงข่ายประสาทเทียมเหมือนกับ YOLO แต่การออกแบบโครงสร้างแตกต่างกัน โดยที่ SSD จะใช้ VGG-16 ในการสกัดคุณลักษณะ และใช้ Convolution layer ต่อ กันหลายชั้นเพื่อลดมิติและความละเอียดทำให้ตรวจจับวัตถุในหลายขนาด ซึ่งในแต่ละชั้นจะได้ผลลัพธ์เป็น Convolution filter จากนั้นจะนำ Convolution filter ไปทำงานผลต่อ

2.2.2 การติดตามการเคลื่อนไหวของวัตถุ

การติดตามการเคลื่อนไหวของวัตถุ^[20] คือระบบที่ใช้สำหรับการติดตามการเคลื่อนไหวของวัตถุที่สนใจที่อยู่ในรูปภาพ โดยใช้การคำนวณทางคณิตศาสตร์ และการประมวลผลภาพ (image processing) ทำให้การประมวลผลนั้นเร็กว่าการใช้โมเดลปัญญาประดิษฐ์ ซึ่งอัลกอริทึมติดตามการเคลื่อนไหวที่นิยมใช้มีสองอัลกอริทึม คือ correlation filter และ kalman filter ซึ่งหลักการของทั้งสองอัลกอริทึมนั้นมีรายละเอียดดังนี้

2.2.2.1 Kalman filter

Kalman filter มีขั้นตอนการทำงานอยู่สามช่วงหลัก คือ initialize, prediction และ update โดยที่ช่วง initialize นั้นจะทำเพียงครั้งเดียวตอนเริ่มทำงานในเฟรมแรก จากนั้นจะทำช่วง predition และ update สลับไปมาเรื่อยๆจนครบทุกเฟรมในวิดีโอ ซึ่งสามารถเขียนออกมานเป็นผังการทำงานได้ดังรูปที่ 2.16



รูปที่ 2.16: ผังการทำงานของระบบติดตามการเคลื่อนไหวของวัตถุแบบ kalman filter

ซึ่งในแต่ละช่วงการทำงานนั้นจะมีรายละเอียดดังนี้

1. Initialize ในขั้นตอนนี้จะเป็นการทำหน้าที่กำหนดค่าเริ่มต้นให้กับเมทริกซ์สถานะหรือ state (x) และเมทริกซ์ความแปรปรวนหรือ covariance matrix (P) โดยที่เมทริกซ์สถานะจะใช้ในการเก็บข้อมูลที่ต้องการติดตามการเคลื่อนไหว และเมทริกซ์ความแปรปรวนเป็นเมทริกซ์ที่ใช้บอกความแม่นยำของการคำนวณ หากสามารถกำหนดค่าในเมทริกซ์ได้อย่างเหมาะสมจะทำให้การปรับปรุงประสิทธิภาพของ kalman filter ในช่วง update เสถียร化ขึ้น หลังจากได้ดำเนินจุดกึ่งกลางของวัตถุในเฟรมแรกแล้วจะสามารถเขียนเมทริกซ์สถานะ และเมทริกซ์ความแปรปรวนได้ดังนี้

$$x_0 = [c_x \ c_y \ v_x \ v_y]^T$$

$$P_0 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

โดยที่

$$\begin{aligned}x_0 &= \text{เมทริกซ์สถานะ ณ เฟรมแรก} \\c &= \text{จุดกึ่งกลางของวัตถุในภาพ} \\v &= \text{ความเร็วในแกนที่กำหนด} \\P_0 &= \text{เมทริกซ์ความแปรปรวน ณ เฟรมแรก}\end{aligned}$$

จากตัวอย่างด้านบนจะเห็นว่าสิ่งที่สนใจนั้นเป็นตัวแหน่งของวัตถุ (c_x, c_y) แต่สำพั่งเพียงแค่ตำแหน่งไม่สามารถใช้ในการบอกตำแหน่งถัดไปได้จึงต้องใช้ความเร็วในแต่ละแกน (v_x, v_y) ในการหาตำแหน่งถัดไปของวัตถุ และได้กำหนดเมทริกซ์ความแปรปรวนให้เป็นเมทริกซ์เอกลักษณ์ (identity matrix)

2. Prediction ในขั้นตอนนี้จะเป็นการทำนายตำแหน่งหรือเมทริกซ์สถานะ (x) และปรับเมทริกซ์ความแปรปรวน (P) โดยใช้ข้อมูลจากเฟรมก่อนหน้าเป็นพื้นฐานในการคำนวณ ซึ่งมีสมการดังนี้

$$x' = Fx + u \quad (2.24)$$

$$P' = FPF^T + Q \quad (2.25)$$

โดยที่

$$\begin{aligned}x' &= \text{เมทริกซ์สถานะในเวลาถัดไปที่ได้จากการทำนาย} \\F &= \text{เมทริกซ์ปรับสถานะ (state transition matrix)} \\u &= \text{แรงกระทำจากภายนอกที่มีผลต่อการเคลื่อนที่ของวัตถุ (แรงเสียดทานหรือแรงลม)} \\Q &= \text{เมทริกซ์ความแปรปรวนที่สอดคล้องกับข้อมูลรบกวนของสถานะ}\end{aligned}$$

จากสมการที่ 2.24 แรงกระทำจากภายนอกที่มีผลต่อการเคลื่อนที่ของวัตถุ (u) นั้นไม่สามารถหาได้จากภาพทำให้ต้องตัดออกไป จะทำให้สมการเหลือเพียง $x' = Fx$ ซึ่งเมทริกซ์ปรับสถานะ (F) สามารถกำหนดให้สอดคล้องกับเมทริกซ์สถานะที่กำหนดไว้ สามารถเขียนตัวอย่างของเมทริกซ์ปรับสถานะได้ดังนี้

$$F = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

โดย dt หมายถึงความต่างของเวลาจากเฟรมก่อนหน้าถึงเฟรมปัจจุบัน จากตัวอย่างเมทริกซ์ปรับสถานะด้านบนสามารถกล่าวได้ว่าความเร็วในการเคลื่อนไหวของวัตถุนั้นคงที่ เนื่องจากการกำหนดเมทริกซ์ปรับสถานะให้เป็น 1 เมื่อนำเมทริกซ์ปรับสถานะไปคูณกับเมทริกซ์สถานะจะทำให้ได้เมทริกซ์สถานะที่ได้จากการทำนาย (x') ดังตัวอย่างด้านล่าง

$$x' = \begin{bmatrix} 1 & 0 & dt & 0 \\ 0 & 1 & 0 & dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} c_x \\ c_y \\ v_x \\ v_y \end{bmatrix} = \begin{bmatrix} c_x + v_x dt \\ c_y + v_y dt \\ v_x \\ v_y \end{bmatrix}$$

จากนั้นจะทำการคำนวณหาเมทริกซ์ความแปรปรวนในเวลาถัดไป (P') โดยใช้เมทริกซ์ความแปรปรวนที่สอดคล้องกับข้อมูลรบกวนของสถานะ (Q) ในการปรับประสิทธิภาพของการทำนายสถานะถัดไปของวัตถุซึ่งมักจะถูกกำหนดให้เป็นเมทริกซ์เอกลักษณ์ (identity matrix) ที่มีขนาดเท่ากันกับเมทริกซ์ P ทั้งนี้ขึ้นอยู่กับระบบของผู้ใช้

3. Update ในขั้นตอนนี้จะเกิดขึ้นหลังจากมีการวัดค่า (measurement) ครั้งใหม่เข้ามาเพื่อนำข้อมูลที่ได้จากการวัดค่ามาเทียบกับข้อมูลที่ได้จากขั้นตอน prediction เพื่อคำนวณหาความคลาดเคลื่อน และปรับตัวแปรต่างๆให้มีความเสถียรมากขึ้น โดยมีจะมีการคำนวณหาเมทริกซ์สถานะและเมทริกซ์ความแปรปรวนที่ผ่านการปรับแล้วด้วยสมการดังนี้

$$x = x' + Ky \quad (2.26)$$

$$P = (I - KH)P' \quad (2.27)$$

โดยที่

K = Kalman gain

y = ค่าความคลาดเคลื่อนระหว่างสถานะที่ได้จากการวัดค่ากับสถานะที่ได้จากการทำนาย

H = เมทริกซ์สำหรับปรับขนาด

I = เมทริกซ์เอกลักษณ์

จากสมการที่ 2.26 คือการนำสถานะที่ได้จากการทำนาย และค่าความคลาดเคลื่อนระหว่างสถานะที่ได้จากการวัดค่ากับสถานะที่ได้จากการทำนายที่ผ่านการคูณด้วย Kalman gain (K) เพื่อปรับเมทริกซ์สถานะของเวลาถัดไปให้ใกล้เคียงกับค่าที่วัดได้ให้มากขึ้น ซึ่งค่าความคลาดเคลื่อนสามารถหาได้จากสมการที่ 2.28 โดยที่ H คือเมทริกซ์สำหรับปรับขนาดเพื่อทำให้เมทริกซ์สถานะที่ได้จากการทำนายสามารถบวกกับเมทริกซ์สถานะที่ได้จากการวัดค่าได้ เนื่องจากเมทริกซ์สถานะที่ได้จากการวัดค่านั้นมักจะมีขนาดไม่เท่ากันกับเมทริกซ์สถานะที่ได้จากการทำนาย และ Kalman gain สามารถหาได้จากสมการที่ 2.29

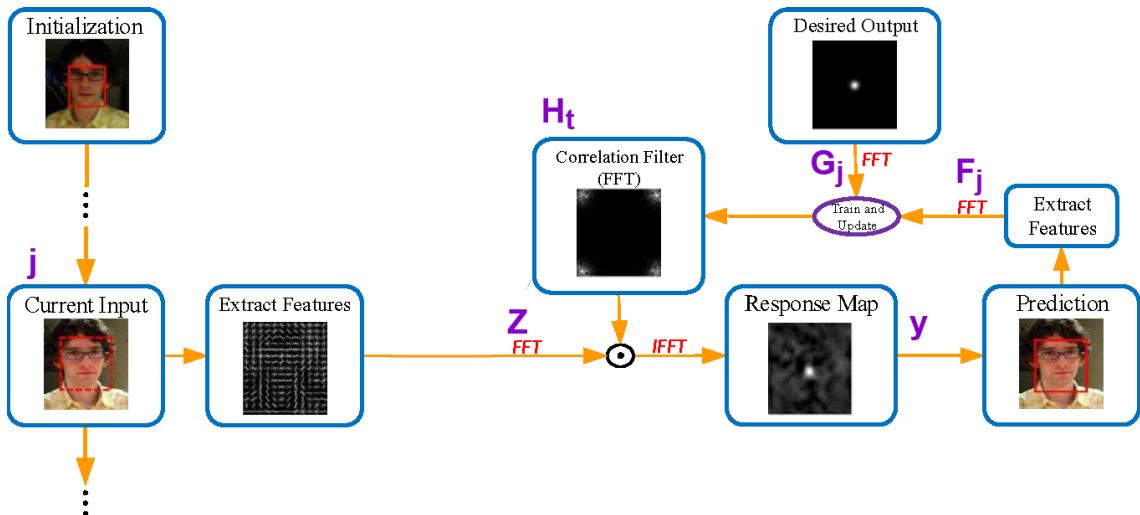
$$y = z - Hx' \quad (2.28)$$

$$K = P'H^TS^{-1} \quad (2.29)$$

โดยที่เมทริกซ์ S หาได้จากการนำเมทริกซ์ความแปรปรวนในเวลาถัดไปมารวมกับเมทริกซ์ของข้อมูลรบกวน (noise) โดยเมทริกซ์ของข้อมูลรบกวนนั้นผู้ใช้ต้องกำหนดขึ้นมาให้สอดคล้องกับระบบ หากยกตัวอย่างในการตรวจจับการเคลื่อนไหวของวัตถุในสิ่ติ์โอ ข้อมูลรบกวนจะเป็นตำแหน่งแกน x และแกน y ของวัตถุเป็นต้น โดยสมการสำหรับหาค่าของเมทริกซ์ S จะเป็นดังสมการที่ 2.30

$$S = HP'H^T + R \quad (2.30)$$

หลังจากปรับเมทริกซ์สถานะแล้วต่อไปคือการปรับเมทริกซ์ความแปรปรวนโดยใช้เมทริกซ์ความแปรปรวนในเวลาถัดไป และ Kalman gain ในการปรับให้เหมาะสมกับข้อมูลที่ได้จากการวัดค่ามา เพื่อเพิ่มประสิทธิภาพในการติดตามตำแหน่งของวัตถุให้ใกล้เคียงกับความเป็นจริงให้มากขึ้น โดยสมการจะเป็นดังในสมการที่ 2.27



รูปที่ 2.17: ผังการทำงานของระบบติดตามการเคลื่อนไหวของวัตถุแบบ correlation filter^[21]

จากรูปที่ 2.17 เป็นขั้นตอนการทำงานของระบบติดตามการเคลื่อนไหวของวัตถุแบบ correlation filter ซึ่งการทำงานจะมีสามขั้นตอนหลักๆ เมื่อมองกันกับ Kalman filter คือ initialize, prediction และ update โดยในแต่ละขั้นตอนจะมีรายละเอียดดังนี้

1. Initialize ในขั้นตอนนี้จะทำการใช้โมเดลปัญญาประดิษฐ์ในการตรวจจับวัตถุที่สนใจภายในเฟรมแรกของวิดีโอเพื่อหาตำแหน่งและกรอบสีเหลี่ยมของวัตถุนั้น จากนั้นนำภาพที่อยู่ในการอบสีเหลี่ยมไปผ่านกระบวนการแปลงฟูรีเยร์ (fourier transform) เพื่อใช้เป็นผลลัพธ์ที่ต้องการ (desired output) (G_j) สำหรับใช้ในการสร้าง correlation filter ในเฟรมตัดไป (H_t)
2. Prediction หลังจากได้รับภาพใหม่ (j) เข้ามาจะทำการสกัดผังคุณลักษณะ ซึ่งทั้งนี้ก็แล้วแต่ว่าผู้พัฒนาต้องการใช้อะไรในการสกัดผังคุณลักษณะของภาพอ кома หลังจากได้ผังคุณลักษณะของภาพที่เข้ามาใหม่แล้วก็นำไปผ่านกระบวนการแปลงฟูรีเยร์ (Z) ก่อนจะได้ทำการ element-wise ด้วย correlation filter และทำการแปลงฟูรีเยร์กลับ (inverse fourier transform) จนครบทั้งภาพเพื่อหาผังการตอบสนองของเฟรมปัจจุบัน (response map) (y) ที่มีค่ามากที่สุดอ กома โดยผังการตอบสนองที่มีค่ามากที่สุดจะเป็นตำแหน่งถัดไปของวัตถุที่สนใจ ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$y = \mathbb{F}^{-1}\{\bar{H}_t * Z\} \quad (2.31)$$

โดยที่

y = ผังการตอบสนอง

\mathbb{F}^{-1} = การแปลงฟูรีเยร์กลับ

\bar{H}_t = correlation filter ที่ผ่านการทำ complex conjugation มาแล้ว

$*$ = element-wise operator

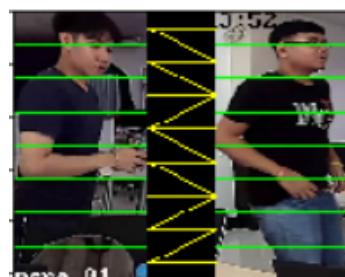
Z = ผังคุณลักษณะที่ผ่านกระบวนการแปลงฟูรีเยร์มาแล้ว

3. Update หลังจากทำนายตำแหน่งของวัตถุในภาพใหม่ได้แล้วจะทำการตัดภาพในกรอบสี่เหลี่ยมใหม่มาส กัดคุณลักษณะแล้วทำการบวนการแปลงพูรีเยอร์ (F_j) และผลลัพธ์ที่ต้องการจากเฟรมก่อนหน้า (G_j) ไป ผ่านกระบวนการตามสมการที่ 2.32 เพื่อปรับ correlation filter ให้เปลี่ยนไปตามเหตุการณ์ที่เกิดขึ้นใน เฟรมวิดีโอ แล้วทำการแทนที่ผลลัพธ์ที่ต้องการจากเฟรมก่อนหน้าด้วยผังการตอบสนองของเฟรมปัจจุบัน จะทำให้การทำนายตำแหน่งในเฟรมต่อๆไปมีประสิทธิภาพมากขึ้นเนื่องจากมีการปรับปรุงตัวกรองให้เป็น ปัจจุบันตลอดเวลา

$$H_t = \frac{\bar{G}_j F_j}{\bar{F}_j F_j} \quad (2.32)$$

2.2.3 การระบุตัวตนของบุคคล

ระบบระบุตัวตนของบุคคล^{[22][23]} คือการระบุตัวตนของบุคคลภายในวิดีโอหรือระหว่างสองภาพ สามารถนำมาประยุกต์ใช้ในด้านของการรักษาความปลอดภัย หรือการตามหาบุคคล ซึ่งการระบุตัวตนของบุคคลนั้น เป็นปัญหาที่ท้าทาย เนื่องจากคุณลักษณะทั่วไปของบุคคลในภาพไม่เพียงพอต่อการระบุตัวตนภายในภาพว่า เป็นบุคคลคนเดียวกันได้ ซึ่งวิธีการที่ใช้ในการระบุตัวตนของบุคคลเรียกว่า Dynamically Matching Local Information (DMLI) ที่สามารถจัดแนวรายละเอียดข้อมูลของภาพและเพิ่มประสิทธิภาพให้สูงขึ้น ถึงแม้ว่า DMLI นั้นจะไม่ใช่วิธีการที่มีประสิทธิภาพสูงสุดแต่มีประสิทธิภาพใกล้เคียงกับโมเดลอื่นๆ แต่ผู้วิจัยสามารถนำวิธีนี้มาประยุกต์เข้ากับงานวิจัยครั้งนี้ได้สะดวกที่สุด จึงนำวิธีการนี้มาใช้สำหรับงานวิจัยครั้งนี้



รูปที่ 2.18: การแบ่งภาพออกเป็น 8 ส่วนของระบบระบุตัวตนของบุคคล

การทำงานของระบบระบุตัวตนของบุคคลจะเริ่มจากการแบ่งภาพออกเป็น 8 ส่วนและนำคุณลักษณะทั่วไปของภาพมาผ่านกระบวนการ normalization เพื่อลดความซ้ำซ้อนของข้อมูล แล้วนำมาเปรียบเทียบความแตกต่างของคุณลักษณะทั่วไปของภาพโดยใช้วิธี euclidean distance หลังจากนั้นใช้วิธี DMLI หากความแตกต่างออกมาก โดยค่าที่ได้ออกมาจะเรียกว่า aligned distance ถ้าค่าที่ออกมากใกล้เคียงกับศูนย์ จะหมายถึงบุคคลในภาพทั้งสองเป็นบุคคลเดียวกัน โดยใช้การกำหนดเกณฑ์ของ aligned distance สำหรับระบุตัวตนของบุคคลในภาพว่าเป็นบุคคลเดียวกันหรือไม่

โดยทุกข้อมูลที่นำมาใช้สำหรับการทำโนเมเดลปัญญาประดิษฐ์ได้แก่

1. Market1501 เป็นชุดข้อมูลที่เก็บข้อมูลภาพของบุคคลโดยใช้กล้องจำนวนหกตัว ถ่ายภาพบุคคลที่ด้านหน้าของชูปเปลอร์มาร์เก็ตในมหาวิทยาลัย Tsinghua
2. DukeMTMCReID เป็นชุดข้อมูลที่เก็บข้อมูลภาพของบุคคลโดยใช้กล้องจำนวนแปดตัว ถ่ายภาพบุคคลที่วิทยาเขตของมหาวิทยาลัย Duke ซึ่งมีการเก็บภาพมากถึงสองล้านภาพของนักศึกษาสองพันคน
3. CUHK-03 เป็นชุดข้อมูลที่เก็บภาพของบุคคลที่มหาวิทยาลัยที่ฮ่องกง
4. MSMT17 เป็นชุดข้อมูลที่เก็บข้อมูลภาพของบุคคลโดยใช้กล้องจำนวนสิบห้าตัว โดยที่กล้องแต่ละตัวจะไม่ได้ตั้งอยู่สถานที่เดียวกัน และเก็บข้อมูลที่ในวันที่มีสภาพอากาศต่างกัน

โดยทุกชุดข้อมูลจะใช้โครงสร้าง (architecture) ResNet50 ในการสร้างโมเดลปัญญาประดิษฐ์ และทดสอบด้วยวิธี Global+DMLI คือการนำคุณลักษณะทั่วไปและคุณลักษณะเฉพาะของภาพที่ได้มาจากการโนเมเดลปัญญาประดิษฐ์ นำมาหาค่าระยะความแตกต่าง โดยที่ค่าระยะความแตกต่างของคุณลักษณะทั่วไปสามารถได้โดยใช้วิธี Euclidean distance และค่าระยะความแตกต่างของคุณลักษณะเฉพาะสามารถหาได้โดยใช้วิธี DMLI และนำมาเทียบกับชุดข้อมูลทดสอบเพื่อคำนวณหาค่า rank1 และ mAP โดยที่ค่า rank1 หมายถึงค่าอัตราเร้อยละของความมั่นใจสูงสุดของโมเดลปัญญาประดิษฐ์ที่ทำนายอุปกรณ์ต้อง และค่า mAP คือการหาค่าเฉลี่ยความแม่นยำใน

แต่ละหมวดหมู่ ซึ่งสามารถติดค่า rank1 และ mAP ของโมเดลปัญญาประดิษฐ์สำหรับการทำระบุตัวตนของบุคคลได้ในหัวข้อที่ 4.4.1

วิธีการคำนวณของ DMLI ในขั้นตอนการสร้างโมเดลปัญญาประดิษฐ์ของการระบุตัวตนบุคคล

$$d_{i,j} = \frac{e^{\|f_i - g_j\|^2} - 1}{e^{\|f_i - g_j\|^2} + 1} \quad i, j \in 1, 2, 3, \dots H \quad (2.33)$$

โดยที่

d = เมทริกซ์ของระยะความแตกต่างที่น้อยที่สุดของคุณลักษณะจำเพาะของทั้งสองภาพ

f = ค่าคุณลักษณะจำเพาะของรูปภาพที่ 1

g = ค่าคุณลักษณะจำเพาะของรูปภาพที่ 2

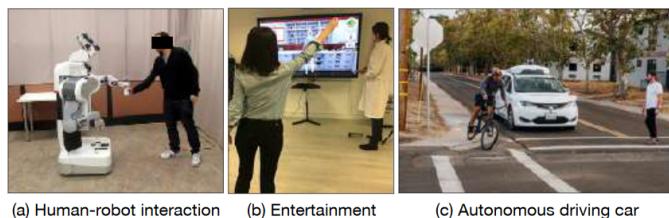
H = จำนวนภาพแนวตั้งที่แบ่งออกมา

$$S_{i,j} = \begin{cases} d_{i,j} & \text{if } i = 1, j = 1 \\ S_{i-1,j} + d_{i,j} & \text{if } i \neq 1, j = 1 \\ S_{i,j-1} + d_{i,j} & \text{if } i = 1, j \neq 1 \\ \min(S_{i-1,j}, S_{i,j-1}) & \text{if } i \neq 1, j \neq 1 \end{cases} \quad (2.34)$$

เมื่อทำการคำนวณ $S_{i,j}$ ซึ่งเป็นผลรวมของระยะความแตกต่างที่น้อยที่สุดแล้วตัวสุดท้ายของ $S_{i,j}$ จะเป็นระยะความแตกต่างที่น้อยที่สุดที่ของคุณลักษณะจำเพาะของทั้งสองภาพ แต่ในกรณีที่ทางผู้วิจัยนำมาใช้งานค่า $d_{i,j}$ นั้นจะเป็นค่าที่ได้มาจากการนำคุณลักษณะทั่วไปของภาพมาทำ euclidean distance แทน

2.2.4 การจำแนกการกระทำการทำของมนุษย์

ระบบจำแนกการกระทำการทำของมนุษย์^[24] เป็นหัวข้อที่มีการให้ความสนใจอย่างมากสำหรับการทำระบบประมวลผลวิดีโอบนปัจจุบัน เนื่องจากระบบประมวลผลวิดีโอนั้นสามารถใช้งานได้หลากหลายสถานการณ์ เช่น ใช้สอดส่องการจราจรบนท้องถนน วิเคราะห์พฤติกรรมการเคลื่อนที่ของลูกค้าภายในห้าง การตามหาคนหายหรือพัสดุหลงภายนอกอาคาร เป็นต้น ซึ่งการที่ระบบสามารถจำแนกการกระทำการทำของมนุษย์ภายใต้วิดีโอด้วยนั้นสามารถเพิ่มความสามารถของระบบประมวลผลวิดีโอด้วย เช่น สามารถแจ้งเตือนเมื่อพบบุคคลที่มีพฤติกรรมน่าสงสัยในวิดีโอด้วย เพื่อไม่ให้เกิดเหตุการณ์อันตรายขึ้น เป็นต้น



รูปที่ 2.19: ตัวอย่างการประยุกต์ใช้งานระบบจำแนกการกระทำการทำของมนุษย์^[25]

การสร้างโมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำการทำนั้น จำเป็นต้องมีชุดข้อมูลที่เหมาะสมกับการกระทำการที่สนใจ และโครงสร้างโมเดลปัญญาประดิษฐ์ที่เหมาะสม ซึ่งในปัจจุบันนั้นมีชุดข้อมูลสาธารณะที่สามารถนำมาใช้งานได้หลากหลายชุดข้อมูล เช่น YouTube-8M ชุดข้อมูลสำหรับการประมวลผลวิดีโอมีขนาดใหญ่ที่สุดและมีจำนวนคำจำกัดมากที่สุด, AVA ชุดข้อมูลของการกระทำที่มีการขยายเพียงเล็กน้อย, Sports-1M^[26] ชุดข้อมูลที่เกี่ยวกับกิจกรรมกีฬาต่างๆของมนุษย์ เป็นต้น ซึ่งชุดข้อมูลที่ผู้วิจัยเลือกนำมาศึกษาได้แก่ YouTube-8M, AVA และ Moment in Time โดยแต่ละชุดข้อมูลจะมีความแตกต่างกันในหลายๆด้านแต่จะมีสิ่งที่เหมือนกัน คือ เป็นชุดข้อมูลสำหรับการวิเคราะห์วิดีโอที่สนใจการกระทำการและกิจกรรมของมนุษย์ โดยจะกล่าวถึงความแตกต่างในด้านต่างๆ เช่น เป้าหมายของแต่ละชุดข้อมูล วิธีการเก็บข้อมูลสำหรับชุดข้อมูล วิธีการสร้างคำจำกัด และรายละเอียดอื่นๆของชุดข้อมูล จากนั้นจะสรุปข้อมูลของแต่ละชุดข้อมูลในหัวข้อที่ 2.3

2.3 ชุดข้อมูลสำหรับการประมวลผลวิดีโอ

ในปัจจุบันมีชุดข้อมูลมากมายถูกสร้างขึ้นมาสำหรับใช้สร้างโมเดลสำหรับแก้ปัญหาในด้านต่างๆ เช่น การตรวจจับวัตถุภายในภาพ การจำจำใบหน้าบุคคลภายในภาพ การจำแนกการกระทำของมนุษย์ เป็นต้น ซึ่งสิ่งที่ทำให้โมเดลปัญญาประดิษฐ์นั้นมีประสิทธิภาพสูงคือ จำนวนของข้อมูล โดยในปัจจุบันปัญหาด้านการทำความเข้าใจภาพด้วยปัญญาประดิษฐ์ (image understanding) สามารถพัฒนาให้มีประสิทธิภาพสูงนั้นเนื่องจากมีจำนวนข้อมูลที่มากมาก ในขณะที่ปัญหาด้านการทำความเข้าใจวิดีโอด้วยปัญญาประดิษฐ์ (video understanding) นั้นกำลังมีการให้ความสนใจเพิ่มขึ้นเรื่อยๆ ในช่วงระยะเวลาหลายปีที่ผ่านมา ในหัวข้อนี้จะพูดถึงการศึกษาชุดข้อมูลที่ใช้สำหรับการทำความเข้าใจวิดีโอ โดยจะมุ่งเน้นไปที่การจำแนกการกระทำการของมนุษย์เป็นหลัก โดยชุดข้อมูลที่หยิบมาศึกษาคือ YouTube-8M, atomic visual action และ moments in time

2.3.1 ชุดข้อมูล YouTube-8M

YouTube-8M^[27] คือชุดข้อมูลวิดีโอที่มีจำนวนวิดีโอเยอะที่สุดถึง 8 ล้านวิดีโอ (พ.ศ. 2559) โดยมีจุดมุ่งหมายหลักในการจำแนกสาระสำคัญของวิดีโอ (video theme) ด้วยคำสั้นๆ เช่น ถ้าวิดีโอนั้นมีมนุษย์กำลังปั่นจักรยานบนถนนในริมหน้าผาชุดข้อมูลนี้จะกำกับวิดีโอนี้ว่า mountain biking ซึ่งทำให้ YouTube-8M แตกต่างจากชุดข้อมูลวิดีโອื่นๆ ส่วนใหญ่ที่จะเน้นการกระทำหรือกิจกรรมของมนุษย์ ซึ่งข้อมูลโดยสรุปของชุดข้อมูลมีดังนี้

1. รายละเอียดของชุดข้อมูล

- (a) เป้าหมายของชุดข้อมูล : เพื่อจำแนกสาระสำคัญของวิดีโอ
- (b) จำนวนของวิดีโอ : 8,264,650 วิดีโอ
- (c) ความยาวเฉลี่ยของแต่ละวิดีโอ : 229.6 วินาที
- (d) จำนวนของหมวดหมู่ของคำกำกับ : 4,800 หมวดหมู่
- (e) กฎในการรวบรวมวิดีโอดังนี้

- i. ทุกๆคำกำกับต้องเป็นรูปธรรม
- ii. ในแต่ละคำกำกับต้องมีจำนวนวิดีโอมากกว่า 200 วิดีโอ
- iii. ความยาวของวิดีโอด้วยอยู่ระหว่าง 120 - 500 วินาที

หลังจากได้กฎในการรวบรวมวิดีโอลاء ขั้นตอนต่อไปคือการสร้างคำศัพท์ที่ใช้ในการค้นหาข้อมูลวิดีโອาจใน YouTube

(f) ขั้นตอนในการสร้างคำศัพท์มีดังนี้

- i. กำหนดบัญชีขาว (whitelist) ของคำกำกับที่เป็นรูปธรรมมา 25 ชนิด เช่น กีฬา เป็นต้น
- ii. กำหนดบัญชีดำ (blacklist) ของคำกำกับที่คิดว่าไม่เป็นรูปธรรมไว้ เช่น software เป็นต้น
- iii. รวบรวมคำกำกับที่มีอยู่ในบัญชีขาวอย่างน้อยหนึ่งคำ และต้องไม่มีอยู่ในบัญชีดำ ซึ่งจะทำให้ได้คำกำกับที่ต้องการมาประมาณ 50,000 คำ
- iv. จากนั้นใช้ผู้ประเมินจำนวนสามคน ในการคัดคำกำกับที่คิดว่าเป็นรูปธรรม และสามารถจำกัดหรือเข้าใจได้ยากโดยไม่ต้องเชี่ยวชาญในด้านนั้นๆ ซึ่งผู้ประเมิน ก็จะมีคำถามว่า “มันยกขนาดไหนถึงจะระบุได้ว่ามีคำกำกับดังกล่าวอยู่ในรูปหรือวิดีโอ โดยใช้เพียงแค่การมองเท่านั้น?” โดยแบ่งเป็นระดับดังนี้
 - A. บุคคลทั่วไปสามารถเข้าใจได้ (1)
 - B. บุคคลทั่วไปที่ผ่านการอ่านบทความที่เกี่ยวข้องมาแล้วสามารถเข้าใจได้ (2)
 - C. ต้องเชี่ยวชาญในด้านใดซักด้านจะเข้าใจได้ (3)
 - D. เป็นไปไม่ได้ ถ้าไม่มีความรู้ที่ไม่เป็นรูปธรรม (4)
 - E. ไม่เป็นรูปธรรม (5)

- v. หลังจากคำถ้ามีข้างบนและการให้คะแนน จะทำการเก็บไว้เฉพาะคำกำกับที่มีคะแนนเฉลี่ยมากที่สุดอยู่ที่ประมาณ 2.5 คะแนนหรือต่ำกว่าเท่านั้น
vi. ทำให้สุดท้ายเหลือเพียงประมาณ 10,000 คำที่สามารถใช้ได้
vii. หลังจากได้คำกำกับที่คิดว่าเป็นรูปธรรมแล้วก็นำไปค้นหาและรวบรวมด้วย YouTube annotation system โดยมีขั้นตอนดังนี้
- A. สุ่มเลือกวิดีโอมาก 10 ล้านวิดีโอพร้อมกับคำกำกับของวิดีโอด้วยใช้กฎที่กำหนดไว้
 - B. ทำให้เหลือจำนวนวิดีโอยู่ 8,264,650 วิดีโอ
 - C. แยกออกเป็นสามส่วนคือ ชุดข้อมูลสำหรับสร้างโมเดล (train set), ชุดข้อมูลสำหรับตรวจสอบ (validate set) และชุดข้อมูลสำหรับทดสอบ (test set) ในอัตราส่วน 70:20:10 ตามลำดับ

2. โมเดลปัญญาประดิษฐ์

(a) การเตรียมข้อมูล

- i. คุณลักษณะระดับเฟรม : ต้องทำการลดขนาดของข้อมูลลง เนื่องจากข้อมูลมีขนาดใหญ่มาก ทำให้ใช้เวลาในการประมวลผลนาน ซึ่งกระบวนการนี้จะมีการลดความเร็วเฟรมต่อวินาที ทางเวกเตอร์ของคุณลักษณะ และแปลงข้อมูลจาก 32 บิต ให้เป็น 8 บิต
- ii. คุณลักษณะระดับวิดีโอ : การสกัดคุณลักษณะระดับวิดีโอด้วยคุณลักษณะระดับเฟรมซึ่งการทำแบบนี้ทำให้ได้ประโยชน์สามข้อ คือโมเดลทั่วไปที่ไม่ใช้โครงข่ายประสาทเทียมสามารถนำไปใช้งานได้ขนาดข้อมูลเล็กลง และเหมาะสมกับการนำไปสร้างโมเดลในขอบเขตอื่นมากขึ้น

(b) โมเดลปัญญาประดิษฐ์ที่ใช้ในการทดสอบชุดข้อมูลแบบที่เป็นคุณลักษณะระดับเฟรม

i. one vs all logistic regression classifier + average pooling

สร้างโมเดลปัญญาประดิษฐ์ของทุกคำกำกับแยกกัน จะได้โมเดลปัญญาประดิษฐ์ 4800 โมเดล ซึ่งในการทำนายผลจะใช้การเฉลี่ยความน่าจะเป็นของแต่ละคำกำกับจากทุกๆเฟรมในวิดีโอด้วยคำกำกับที่มีความน่าจะเป็นมากที่สุดจะเป็นคำที่จริงของการทำนาย โดยมีสมการคำนวณความน่าจะเป็นเฉลี่ยของแต่ละคำกำกับดังนี้

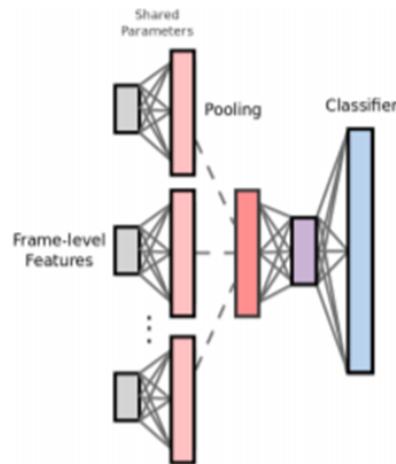
$$p_v(e|X_{1:F_v}^v) = \frac{1}{F_v} \sum_{j=1}^{F_v} p(e|X_j^v) \quad (2.35)$$

โดยที่

- | | |
|----------------------|--|
| v | = วิดีโอที่ใช้ในการทำนายผล |
| e | = คำกำกับ |
| F_v | = จำนวนเฟรมสูงสุดของวิดีโอ v |
| $p_v(e X_{1:F_v}^v)$ | = ความน่าจะเป็นของคำกำกับ e บนวิดีโอ v |

ii. Deep bag of frames

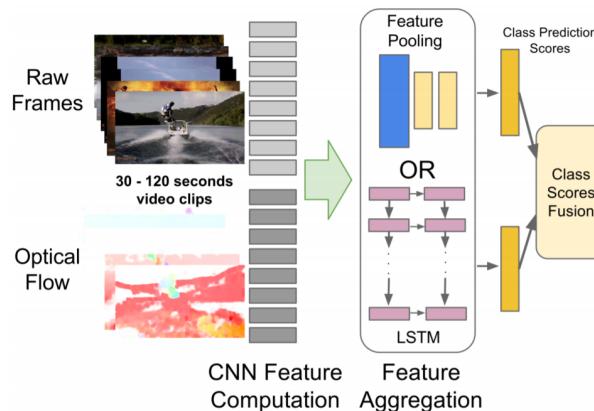
มีหลักการเหมือนกับ deep bag of words^[28] คือการแยกคุณลักษณะของเฟรมที่โมเดลคิดว่าสำคัญออกจากทำนายผล ซึ่งโครงสร้างโมเดลปัญญาประดิษฐ์เป็นดังนี้ดังรูปที่ 2.20 โดยที่จะสุ่มหยิบ 20 เฟรมของวิดีโอมาก่อน ReLU activation function จากนั้นทำการ batch normalization ก่อนจะใช้ max pooling ในการรวมคุณลักษณะที่ได้ให้เป็นคุณลักษณะระดับวิดีโอ สุดท้ายใช้ softmax ในการจำแนกว่าเป็นคำกำกับใด



รูปที่ 2.20: โครงสร้างโมเดลปัญญาประดิษฐ์ของ deep bag of frames

iii. Long short-term memory (LSTM)

โมเดล LSTM ที่ใช้ในบทความนิยมมีการอ้างอิงโครงสร้างมาจากบทความ "Beyond Short Snippets: Deep Networks for Video Classification"^[29] ซึ่งมีโครงสร้างดังรูปที่ 2.21



รูปที่ 2.21: โครงสร้าง LSTM ที่ใช้การอ้างอิงในบทความนี้

แต่เนื่องจากข้อมูลของ YouTube-8M นั้นไม่สามารถเข้าถึงเฟรมวิดีโอเดิม (raw frame) ได้ จึงทำให้สามารถใช้ได้เพียงชั้นของ LSTM และ softmax เท่านั้น ซึ่งจากการทดลองพบว่า การใช้ LSTM 2 ชั้นที่มี hidden unit 1024 หน่วย นั้นมีประสิทธิภาพมากที่สุด

(c) โมเดลปัญญาประดิษฐ์ที่ใช้ในการทดสอบชุดข้อมูลแบบที่เป็นคุณลักษณะระดับวิดีโอ

i. Logistic regression

สร้างโมเดลปัญญาประดิษฐ์ของทุกคำกำกับแยกกัน จะได้โมเดลปัญญาประดิษฐ์ 4800 โมเดล โดยที่พารามิเตอร์ Θ (parameter) ของแต่ละโมเดลหาได้จาก

$$\sum_{i=1}^N L(y_{i,e}, \sigma(w_e X_i)) \quad (2.36)$$

$$L(y_{i,e}, \sigma(w_e X_i)) = y_{i,e} \log(\sigma(w_e X_i)) + (1 - y_{i,e}) \log(1 - \sigma(w_e X_i)) \quad (2.37)$$

$$p(e|X) = \sigma(w_e^T X_i) \quad (2.38)$$

$$\sigma(z) = \frac{1}{1 + \exp(-z)} \quad (2.39)$$

โดยที่

N = จำนวนวิดีโอทั้งหมด

X = คุณลักษณะระดับวิดีโอ

$y_{i,e}$ = คำตอบของคำกำกับ e ในวิดีโอที่ i

w_e = weight ของคำกำกับ e

$p(e|X)$ = ความน่าจะเป็นของคำกำกับ e ของคุณลักษณะระดับวิดีโอ X

- ii. Support vector machine (SVM) สร้างโมเดลปัญญาประดิษฐ์ของทุกคำกำกับแยกกันทำให้จะได้โมเดล SVM 4800 โมเดล โดยได้ใช้ค่า -1 ถึง 1 ในการแสดงถึงคำกำกับด้านลบ (negative label) และคำกำกับค้านบวก (positive label) ตามลำดับ และใช้ hinge loss ในการคำนวณหา loss (L) ซึ่งสมการจะเป็นดังนี้

$$L(y, \hat{y}) = \max(0, b - (2y - 1)\hat{y}) \quad (2.40)$$

โดยที่

y = คำตอบจริงของการทำนาย โดยสามารถเป็น 0 หรือ 1 เท่านั้น

\hat{y} = คำผลการทำนายโดยจะมีค่าอยู่ในช่วง -1 ถึง 1

b = Hinge-loss พารามิเตอร์

- iii. Mixture of Expert (MoE) Mixture of experts ที่ถูกนำมาใช้ในการอ้างอิงในบทความนิยม “Hierarchical mixtures of experts and the EM algorithm”^[30] โดยการทำนายความน่าจะเป็นว่าคำกำกับ e ในวิดีโอ X ด้วยสมการ

$$p(e|X) = \sum_{h \in H_e} p(h|X) \sigma(u_h X) \quad (2.41)$$

ซึ่ง $p(h|X)$ สามารถเขียนได้ในรูปสมการดังนี้

$$p(h|X) = \frac{\exp(w_h X)}{1 + \sum_{h' \in H_e} \exp(w'_h X)} \quad (2.42)$$

โดยที่

H_e = hidden state หรือ expert ของโมเดล

X = วิดีโอที่ใช้ในการทำนายผล

u_h = logistic weight ของ expert h

w_h = softmax weight ของ expert h

$p(h|X) = |H_e| + 1$ ที่ผ่าน softmax function

ให้ขุดข้อมูลสำหรับใช้ในการสร้างโมเดล $(x_i, g_i)_{i=1 \dots N}$ โดยที่ x_i คือเวกเตอร์คุณลักษณะ g_i คือคำตอบจริงของการทำนายซึ่งสามารถเป็นได้เพียง 0 และ 1 เท่านั้น และ N คือจำนวน

วิธีอสูงสุด ซึ่งสมการ log-loss ระหว่างผลการทำนายกับคำตอบจริงของการทำนายคือ

$$L(p, g) = -g \log p - (1 - g) \log(1 - p) \quad (2.43)$$

ซึ่งสามารถเขียนในรูปอนุพันธ์ของ softmax weight และ logistic weight ได้ดังนี้

$$\frac{\partial L[p_{y|x,g}]}{\partial w_h} = x \frac{p_{h|x}(p_{y|h,x} - p_{y|x})(P_{y|x} - g)}{p_{y|x}(1 - p_{y|x})} \quad (2.44)$$

$$\frac{\partial L[p_{y|x,g}]}{\partial u_h} = x \frac{p_{h|x} p_{y|h,x} (1 - p_{y|x})(P_{y|x} - g)}{p_{y|x}(1 - p_{y|x})} \quad (2.45)$$

โดยที่

L = log-loss

p = ผลลัพธ์การทำนาย

g = คำตอบจริงของการทำนาย

u_h = logistic weight ของ expert h

w_h = softmax weight ของ expert h

X = วิธีอื่นๆในการทำนายผล

(d) เครื่องมือที่ใช้วัดผลสำหรับงานวิจัยนี้ คือ

- i. Mean Average Precision (mAP) ในแต่ละคำ做起กับได้ทำการปัดคะแนนให้อยู่ในช่วง 10^{-4} แล้วเรียงลำดับคะแนนทั้งหมดที่ไม่ใช่ 0 จากนั้นให้ค่า τ เป็นค่าแบ่งเกณฑ์ (threshold) โดยที่มี $P(\tau)$ คือ precision และ $R(\tau)$ คือ recall ซึ่งหาได้จาก

$$P(\tau) = \frac{\sum_{t \in T} \mathbb{I}(y_t \geq \tau) g_t}{\sum_{t \in T} \mathbb{I}(y_t \geq \tau)} \quad (2.46)$$

$$R(\tau) = \frac{\sum_{t \in T} \mathbb{I}(y_t \geq \tau) g_t}{\sum_{t \in T} g_t} \quad (2.47)$$

โดยที่

y_t = ค่าความน่าจะเป็นในการทำนาย ซึ่งมีค่าอยู่ในช่วง 0 ถึง 1

g_t = คำตอบจริงของการทำนาย โดยจะมีค่าเป็นได้แค่ 0 และ 1 เท่านั้น

$\mathbb{I}(y_t \geq \tau)$ = พังก์ชันซึ่งจะมีค่าเป็น 1 ถ้าหาก y_t มีค่ามากกว่า τ นอกเหนือจากนั้นจะมีค่าเป็น 0

สามารถหา average precision ได้จากการนี้

$$AP = \sum_{j=1}^{10000} P(\tau_j) [R(\tau_j) - R(\tau_j + 1)] \quad (2.48)$$

โดยที่ $\tau = \frac{j}{10000}$

ii. Hit@k

เหมือนกันกับ Top@k คือการจัดลำดับความน่าจะเป็นของแต่ละคำกำกับจำนวน k อันดับแรก ถ้าหากมีคำกำกับที่ถูกต้องอยู่ในลำดับเหล่านั้น จะถือว่าการทำนายถูกต้อง ซึ่งสามารถเขียนเป็นสมการได้ดังนี้

$$\frac{1}{|V|} \sum_{v \in V} V_e \mathbb{I}(rank_{v,e} \leq k) \quad (2.49)$$

โดยที่

$$\begin{aligned} V &= วิดีโอที่ใช้ในการทดสอบทั้งหมด \\ G_v &= คำตอบของวิดีโอ v \\ rank_{v,e} &= อันดับของคำตอบที่ถูกต้อง e ของวิดีโอ v ที่ได้จากการทำนาย \\ k &= อันดับที่ใช้เป็นเกณฑ์ \end{aligned}$$

iii. Precision at equal recall rate (PERR)

สำหรับแต่ละวิดีโอจะดูความแม่นยำของผลการทำนาย k อันดับแรก โดยที่ k คือจำนวนคำตอบทั้งหมดของวิดีโอนั้น จากนั้นเฉลี่ยค่าเหล่านั้นด้วยจำนวนวิดีโอทั้งหมด สามารถเขียนได้ในรูปสมการดังนี้ โดยใช้ตัวแปรเดียวกันกับของ Hit@k

$$\frac{1}{|V : |G_v| > 0|} \sum_{v \in V : |G_v| > 0} \left[\frac{1}{|G_v|} \sum_{e \in G_v} \mathbb{I}(rank_{v,e} \leq |G_v|) \right] \quad (2.50)$$

(e) ประสิทธิภาพของโมเดลปัญญาประดิษฐ์ที่ใช้ชุดข้อมูลของ YouTube-8M ในการสร้างเทียบกับชุดข้อมูลสำหรับทดสอบของ YouTube-8M

| Input features | Modeling approach | mAP | Hit@1 | PERR |
|----------------|----------------------|-------------|-------------|-------------|
| Frame-level | Logistic + average | 11.0 | 50.8 | 42.2 |
| | Deep bag of frames | 26.9 | 62.7 | 55.1 |
| | LSTM | 26.6 | 64.5 | 57.3 |
| Video-level | SVM | 17.0 | 56.3 | 47.9 |
| | Logistic regression | 28.1 | 60.5 | 53.0 |
| | Mixture-of-2-experts | 30.0 | 63.3 | 55.8 |

ตารางที่ 2.2: ผลการทดสอบโมเดลต่างๆบนชุดข้อมูลสำหรับทดสอบของ YouTube-8M

- (f) ไม่เดลปัญญาประดิษฐ์ที่ใช้ชุดข้อมูลของ YouTube-8M ในการสร้างแล้วปรับไม่เดลด้วยชุดข้อมูลของ Sports-1M เมื่อนำไปทดสอบกับชุดข้อมูลของ Sports-1M พบว่าประสิทธิภาพเมื่อทดสอบด้วย Hit@1 และ Hit@5 สูงสุดอยู่ที่ 65.7% และ 86.2% ตามลำดับ ซึ่งใกล้เคียงกับสถิติสูงสุดในตอนนั้นที่ทำไว้ 73.0% และ 91.0% (พ.ศ. 2559) โดยที่ใช้เพียงคุณลักษณะจากการเตรียมข้อมูลไว้แล้ว เมื่อเทียบกับสถิติเก่าที่มีการใช้ optical flow เข้ามาช่วย จึงเป็นการพิสูจน์ให้เห็นว่าจำนวนข้อมูลนั้นมีผลต่อการพัฒนาไม่เดลปัญญาประดิษฐ์
- (g) ไม่เดลปัญญาประดิษฐ์ที่ใช้ชุดข้อมูลของ YouTube-8M ในการสร้างแล้วปรับไม่เดลด้วยชุดข้อมูลของ ActivityNet^[31] เมื่อนำไปทดสอบกับชุดข้อมูลของ ActivityNet พบว่าประสิทธิภาพเมื่อทดสอบด้วย mAP สูงสุดอยู่ที่ 77.6% ในขณะที่สถิติเดิมทำไว้เพียง 53.8% เป็นการพิสูจน์ว่า ชุดข้อมูลนั้นมีความครอบคลุม (generalize) พอดีที่จะนำไปใช้กับงานประยุกต์ในหมวดอื่นๆ เนื่องจากว่าจำนวนข้อมูลที่มีขนาดใหญ่และมีความหลากหลาย
- (h) ปัญหาที่พบ
เนื่องจากว่า YouTube-8M นั้นมีจำนวนข้อมูลที่เยอะมาก ทำให้ไม่สามารถตรวจสอบความถูกต้องของชุดข้อมูลได้ทั้งหมดว่ามีความถูกต้องมากน้อยขนาดไหน ทำให้อาจเกิดข้อผิดพลาดได้ (ปัจจุบันปี 2019 YouTube-8M ได้มีการตรวจสอบข้อมูลอีกครั้ง เพื่อเพิ่มประสิทธิภาพของชุดข้อมูลซึ่งทำให้ปัจจุบันจำนวนข้อมูล และจำนวนคำกำกับลดน้อยลงจากข้อมูลที่ใช้อ้างอิงในบทความข้างต้นที่ได้กล่าวมา)

2.3.2 ชุดข้อมูล Atomic visual action (AVA)

AVA^[32] คือ ชุดข้อมูลที่รวมวิดีโอที่มีความยาว 15 นาที ถูกแบ่งด้วยความถี่ 1 hz (900 keyframes) จากในภาพนัตต์โดยยึดการกระทำของมนุษย์เป็นศูนย์กลาง เพื่อใช้สำหรับสร้างโมเดลที่เข้าใจกิจกรรมของมนุษย์ ในวิดีโอด้วยคำจำกัดความหมาย เช่น ข้อดีของ AVA คือ ชุดข้อมูลจะมีคำจำกัดเป็นแบบหลากหลายคำจำกัด (multiple label) และคำจำกัดของ AVA มีจำนวน 80 ประเภท สามารถแบ่งได้เป็นสามหมวดหมู่คือ ท่าทาง (Pose), ปฏิสัมพันธ์กับวัตถุ (Interaction with object) และปฏิสัมพันธ์กับบุคคล (Interaction with people) ซึ่งสามารถมีคำจำกัดได้มากถึง 7 คำจำกัด

1. รายละเอียดชุดข้อมูล

(a) ขั้นตอนการเก็บข้อมูลสำหรับการทำชุดข้อมูลมีขั้นตอนการทำ 5 ขั้นดังนี้

- i. การสร้างคำศัพท์การกระทำจะมีหลักการ 3 ข้อในการรวบรวมคำศัพท์ดังนี้
 - A. เก็บรวบรวมคำศัพท์ที่รู้ไปที่เกิดขึ้นในชีวิตประจำวัน
 - B. จะต้องมีเอกสารฉากรายชื่อของนักแสดงที่มีชื่อเสียง เช่น การถือของ
 - C. กำหนดรูปแบบของคำศัพท์ขึ้นมา และใช้ความรู้จากชุดข้อมูลอื่นในการทำให้ได้หมวดหมู่การกระทำการของมนุษย์ที่ครอบคลุม
- ii. ภาพนัตต์และส่วนที่เลือกมาใช้ทำชุดข้อมูล AVA ทั้งหมดจะถูกนำมาจาก YouTube โดยเริ่มจากการรวบรวมรายการชื่อของนักแสดงที่มีชื่อเสียง ซึ่งจะมีความหลากหลายของเชื้อชาติ รวมกันอยู่ วิดีโอด้วยคัดเลือกจะมีเกณฑ์ดังนี้
 - A. วิดีโอด้วยอยู่ในหมวด ภาพนัตต์ และละครโทรทัศน์
 - B. วิดีโอด้วยต้องมีความยาวมากกว่า 30 นาที
 - C. เผยแพร่มาแล้วเป็นระยะเวลาน้อยกว่า 1 ปี
 - D. มีจำนวนยอดคนดูมากกว่า 1,000 ครั้ง
 - E. ลงทะเบียนวิดีโอบางประเภท เป็นภาพขาว-ดำ มีความละเอียดต่ำ การ์ตูน หรือวิดีโอกลเมอร์
- iii. การสร้างกรอบสี่เหลี่ยมครอบมนุษย์ที่อยู่ภายในภาพประกอบด้วย 2 ขั้นตอน
 - A. สร้างกรอบสี่เหลี่ยมโดยใช้โมเดลปัญญาประดิษฐ์ faster RCNN สำหรับการตรวจจับมนุษย์
 - B. ใช้มนุษย์ในการตรวจสอบและแก้ไขกรอบสี่เหลี่ยมที่ผิดพลาด
- iv. การติดตามตำแหน่งของบุคคล

ทำการติดตามตำแหน่งของบุคคลที่อยู่ในช่วงเวลาเดียวกันด้วยใช้วิธีการแทร็คโดยยึดมนุษย์เป็นศูนย์กลาง โดยการคำนวณค่าความใกล้เคียงกันระหว่างบุคคล โดยใช้ person embedding (ใช้โครงข่ายประสาทเทียมในการหาคุณลักษณะขั้นสูงและใช้เมทริกซ์ในการหาความสัมพันธ์ของแต่ละคน) จากนั้นจะใช้อัลกอริทึม Hungarian distance (อัลกอริทึมสำหรับการหาข้อเสนอที่ดีที่สุด) ในการหาตัวเลือกคู่ของกรอบสี่เหลี่ยมที่ดีที่สุด
- v. การสร้างคำจำกัดคุณลักษณะ

การสร้างคำจำกัดของกิจกรรมจะถูกสร้างขึ้นโดยมนุษย์ ซึ่งผู้วิจัยจะใช้โปรแกรมสำหรับช่วยเหลือในการสร้างคำจำกัดคุณลักษณะ โดยสามารถกำหนดคำจำกัดของการกระทำได้สูงสุดถึง 7 คำต่อ 1 กรอบสี่เหลี่ยม นอกจากนั้นสามารถตั้งสถานะเนื้อหาที่ไม่เหมาะสม หรือ

กรอบสีเหลี่ยมที่ผิดพลาดได้อีกด้วย ซึ่งในทางปฏิบัติเพื่อลดโอกาสที่จะเกิดข้อผิดพลาด จึงแบ่งขั้นตอนในการสร้างคำกำกับออกเป็น 2 ขั้นตอนดังนี้

- A. สร้างข้อเสนอสำหรับคำกำกับของการกระทำ
- B. ข้อเสนอจะถูกตรวจสอบข้อเสนอที่ได้จากขั้นตอนแรก ซึ่งจะใช้มุขย์ในการตรวจสอบ 3 คน โดยคำกำกับจะต้องถูกตรวจสอบด้วยผู้ตรวจสอบอย่างน้อย 2 คน จึงจะถูกยืนยันว่าเป็นคำกำกับหลัก

2. โมเดลปัญญาประดิษฐ์

(a) โมเดลปัญญาประดิษฐ์ที่งานวิจัยนี้ใช้ คือ two stream variant ซึ่งจะทำการประมวลผลทั้ง RGB flow และ optical flow โดยเป็นโครงสร้างของ faster RCNN ที่นำ Inception network เข้ามาใช้

(b) เครื่องมือที่ใช้วัดผลสำหรับงานวิจัยนี้ คือค่า IoU และ 3D IoUs

- i. ค่า IoU คือค่าที่ใช้วัดความสอดคล้องระหว่างสองกรอบสีเหลี่ยม(กรอบสีเหลี่ยมจริงของเฟรม และ กรอบสีเหลี่ยมที่ทำนายขึ้นมา) ซึ่งใช้สำหรับการวัดผลระดับเฟรม
- ii. ค่า 3D IoU คือค่าที่ใช้วัดความสอดคล้องระหว่างกรอบสีเหลี่ยมภายในสองวิดีโอใช้สำหรับการวัดผลระดับวิดีโอ โดยเทียบกันระหว่างกรอบสีเหลี่ยมจริงในช่วงเฟรมที่ต่อกัน (ground-truth tubes) และกรอบสีเหลี่ยมที่ทำนายขึ้นมาในช่วงของเฟรมที่ต่อกัน (linked detection tubes)

(c) ประสิทธิภาพของโมเดลปัญญาประดิษฐ์ในปัจจุบัน

ข้อมูลโมเดลปัญญาประดิษฐ์ที่นำมาทดสอบ

- i. Actionness^[33] เป็นการหาความน่าจะเป็นของการกระทำ โดยใช้โครงสร้างของ hybrid fully convolutional network (HFCN) hybrid fully เป็นโครงสร้างที่ประกอบด้วยโครงข่ายประสาทเทียม 2 ชนิด คือ
 - A. Appearance-FCN (A-FCN) คือ โครงข่ายประสาทเทียมที่นำมาใช้แสดงลักษณะของวัตถุ (ตำแหน่งวัตถุ, ความตื้นลึกวัตถุ) ที่ปรากฏบนภาพ RGB1
 - B. MotionFCN (M-FCN) คือ โครงข่ายประสาทเทียมที่แยกการเคลื่อนไหวจากข้อมูลของ optical flow
- ii. Peng without MR, Peng with MR (Multi-region two-stream R-CNN)^[34] เป็นโมเดลปัญญาประดิษฐ์ที่ใช้สำหรับตรวจสอบวิดีโອิชีวิตจริง ซึ่งพื้นฐานของโมเดลนี้เป็น Faster R-CNN โดยโมเดลนี้มีกระบวนการ 3 กระบวนการคือ
 - A. สร้างข้อเสนอพื้นที่ที่มีการเคลื่อนไหว
 - B. สะสม Optical flow จากเฟรมหลายเฟรม เพื่อนำไปปรับปรุงการตรวจจับการกระทำ
 - C. นำพื้นที่หลายๆส่วนมาวิเคราะห์ผ่านโมเดล Faster R-CNN
- iii. ACT Action Tubelet Detector^[35] เป็นการระบุตำแหน่งของการกระทำที่มีระยะเวลาสั้นๆ ซึ่งใช้วิธีการตรวจจับระดับเฟรม และ ใช้การติดตามตำแหน่งในการเชื่อมระหว่างเฟรม ปัจจุบันไบยังเฟรมถัดไป. ACT ถูกสร้างต่อจาก SSD framework และ ใช้คอนโวลูชันในการสกัดคุณลักษณะในแต่ละเฟรมซึ่งการคิดคะแนนและความน่าจะเป็นของหมวดหมู่จะคิดจาก การนำคุณลักษณะเรียงต่อกัน และ หาข้อมูลจากลำดับข้อมูลนั้น

จากการทดสอบการเทียบโมเดลปัญญาประดิษฐ์ของงานวิจัยนี้และวิธีการอื่นๆ โดยนำไปทดสอบกับชุดข้อมูลวิดีโอ JHMDB และ UCF101-24 ได้ผลลัพธ์ออกมาดังนี้

| Frame-mAP | JHMDB (mAP) | UCF101-24 (mAP) |
|------------------------|-------------|-----------------|
| Actionness | 39.9 | - |
| Peng w/o MR | 56.9 | 64.8 |
| Peng w/ MR | 58.5 | 65.7 |
| ACT | 65.7 | 69.5 |
| 2 stream(Our approach) | 73.3 | 76.3 |

ตารางที่ 2.3: ผลการทดลองของวิธีต่างๆบนคุณลักษณะระดับเฟรม

(d) ปัญหาที่พบ

ในปัจจุบันยังไม่มีโมเดลปัญญาประดิษฐ์ที่ทดสอบด้วยชุดข้อมูล AVA และได้ผลการทำงานที่ดีเนื่องจากชุดข้อมูลนี้สนใจการกระทำของมนุษย์ที่มีรายละเอียดเล็กๆน้อยๆ ทำให้ยากต่อการทำนายสำหรับโมเดลปัญญาประดิษฐ์

2.3.3 ชุดข้อมูล Moments in Time

Moments in time^[36] คือชุดข้อมูลที่ใช้มนุษย์ในการกำกับข้อมูลทั้งหมดให้กับวิดีโอด้วย 1 ล้านวิดีโอ และมีจำนวนกิจกรรมหรือการกระทำต่างกัน 339 หมวดหมู่ โดยแต่ละวิดีโอมีความยาวอยู่ที่ 3 วินาที เนื่องจาก เป็นเวลาเฉลี่ยที่มนุษย์ใช้ในการเข้าใจกับเหตุการณ์ที่เกิดขึ้น (human working memory) รูปแบบของชุดข้อมูล จะมีอยู่ทั้งหมดอยู่ 3 รูปแบบ ได้แก่ ภาพนิ่ง (spatial) เสียง (auditory) และภาพเคลื่อนไหว (temporal) นอกจากนี้ชุดข้อมูลนี้ยังไม่วรับรวมเพียงแค่การกระทำของมนุษย์เท่านั้น ยังรวมไปถึง สัตว์ สิ่งของ และปรากฏการณ์ ธรรมชาติ ทำให้ชุดข้อมูลนี้เป็นการท้าทายรูปแบบใหม่ เพราะด้วยข้อมูลที่มีความซับซ้อนมากขึ้น

1. รายละเอียดชุดข้อมูล

- (a) เป้าหมายของชุดข้อมูล : สนใจเหตุการณ์ที่เกิดขึ้นในวิดีโอ เช่น การกระทำของคนหรือสัตว์เหตุการณ์ และประภากล่าวการณ์ธรรมชาติ

(b) จำนวนของวิดีโอ : 多กกว่า 1,000,000 วิดีโอ

(c) ความยาวเฉลี่ยของแต่ละวิดีโอ : 3 วินาที

(d) จำนวนของหมวดหมู่ : 339 หมวดหมู่

(e) วิธีการเก็บรวบรวมข้อมูล

 - เริ่มจากการรวบรวมคำที่ใช้อยู่ทั่วไปในชีวิตประจำวันมา 4,500 คำ จาก VerbNet^[37] เว็บไซต์ที่เก็บรวบรวมคำกริยาภาษาอังกฤษขนาดใหญ่ จากนั้นนำมาแบ่งกลุ่มคำที่มีความหมายใกล้เคียงกันโดยใช้คุณลักษณะจาก Propbank^[38] และ FrameNet^[39] โดยเก็บข้อมูลเป็นแบบเวกเตอร์คุณลักษณะฐานสอง (binary feature vector) ซึ่งถ้าคำใดมีความเกี่ยวข้องกันทางคุณลักษณะก็จะให้ค่าเป็น 1 ถ้าไม่เกี่ยวข้องกันจะให้ค่าเป็น 0 จากนั้นจึงใช้วิธี k-means clustering ในการแบ่งกลุ่ม เมื่อแบ่งกลุ่มแล้วจากนั้นจะเลือกคำจากในแต่ละกลุ่มนั้น โดยคำที่เลือกนั้นจะเป็นคำที่ใช้บ่อยที่สุดในกลุ่มนั้น และลบคำนั้นออกจากกลุ่มนั้นๆ ทั้งหมด (คำคำหนึ่งสามารถอยู่ได้หลายกลุ่ม) จากนั้นจะทำการวนการนี้ไปเรื่อยๆ แต่คำที่เลือกมาจะต้องไม่มีความหมายคลุมเครือ หรือเป็นสิ่งที่ไม่สามารถมองเห็นหรือได้ยินได้ และต้องไม่มีความหมายเหมือนกับคำที่เคยเลือกมาก่อน จนสุดท้ายแล้วได้ออกมาที่ 339 หมวดหมู่
 - ต่อมาทำการหาชุดข้อมูลวิดีโอด้วยจะตัดออกมาเพียง 3 วินาทีที่มีความเกี่ยวข้องกับคำใน 339 หมวดหมู่ที่เลือกมาจากวิดีโอแหล่งต่างกัน 10 แหล่ง การตัดวิดีโอนั้นจะไม่ใช้พวก Video2Gif (โมเดลที่ระบุตำแหน่งของสิ่งที่น่าสนใจในวิดีโอ) เพราะจะทำให้เกิดอคติขึ้นจะเกิดขึ้นตอนสร้างโมเดล ดังนั้นจึงใช้มนุษย์ในการตัดวิดีโอ จากนั้นจะทำการส่งข้อมูลของคำ และวิดีโอที่ตัดไปยัง Amazon Mechanical Turk (AMT หรือตลาดแรงงาน) เพื่อทำการสร้างคำจำกัดโดยพนักงานของ AMT ทำให้ได้ 64 วิดีโอที่เกี่ยวข้องกับคำหนึ่ง และอีก 10 วิดีโอที่มีคำจำกัดอยู่แล้ว โดยวิดีโอด้วยมีคำจำกัดบัญญัติแล้วนั้นถ้าพนักงานของ AMT ตอบเหมือนกันเกิน 90% ถึงจะนำเข้าไปรวมกับชุดข้อมูลส่วนอีก 64 วิดีโอ ถ้าเป็นชุดข้อมูลสำหรับสร้างโมเดลจะต้องผ่านพนักงานของ AMT อย่างน้อย 3 ครั้ง และต้องมีคำจำกัดเหมือนกัน 75% ขึ้นไปถึงจะถือว่าเป็นคำจำกัดที่ถูกต้อง ถ้าเป็นชุดข้อมูลสำหรับตรวจสอบ และชุดข้อมูลสำหรับทดสอบจะต้องผ่านพนักงานของ AMT อย่างน้อย 4 ครั้ง และต้องมีคำจำกัดเหมือนกัน 85% ขึ้นไป เหตุผลที่ไม่ตั้งเกณฑ์ไว้ที่ 100% เพราะจะทำให้วิดีโอนั้นยากเกินไปที่จะทำให้สามารถจำการกระทำได้

2. การเตรียมข้อมูล

- (a) ชุดข้อมูลสำหรับสร้างโมเดลจะมี 802,264 วิดีโอ และมีวิดีโອนในแต่ละหมวดหมู่อยู่ที่ 500 ถึง 5,000 วิดีโอ
- (b) ชุดข้อมูลสำหรับตรวจสอบคำตอบจะมี 33,900 วิดีโอ และมีวิดีโອนในแต่ละหมวดหมู่อยู่ที่ 100 วิดีโอ
- (c) แยกเฟรม RGB ออกมาจากวิดีโอ และทำการเปลี่ยนขนาดให้เป็น 340x256 พิกเซล
- (d) ใช้อัลกอริทึม TVL1 optical flow จาก OpenCV เพื่อคัดข้อมูลrgb กวนที่จะเกิดขึ้น
- (e) ทำการแปลงค่าที่อยู่ใน optical flow ให้เป็นเลขจำนวนเต็มเพื่อทำให้การคำนวนนั้นเร็วขึ้น
- (f) ปรับค่า displacement ใน optical flow ให้ค่าสูงสุดเป็น 15 ต่ำสุดเป็น 0 และทำการปรับขนาดให้เป็นช่วง 0 - 255
- (g) เก็บข้อมูลอกมาในรูปแบบของภาพขาวดำเพื่อลดพื้นที่ในการเก็บข้อมูล
- (h) แก้ปัญหาเรื่องการเคลื่อนไหวของกล้องด้วยการนำค่าเฉลี่ยของเวกเตอร์ไปลบกับ displacement
- (i) สุดท้ายจะเป็นสุ่มตัดภาพอกมาเพื่อเพิ่มจำนวนข้อมูล

3. โมเดลปัญญาประดิษฐ์

- (a) ในงานวิจัยนี้ มีการทดสอบโมเดลปัญญาประดิษฐ์หลายรูปแบบ โดยโมเดลปัญญาประดิษฐ์ที่มีประสิทธิภาพการทำงานที่ดีที่สุด 5 ลำดับแรกมีดังนี้
 - i. SVM มีรูปแบบข้อมูลที่ป้อนเข้า คือ ภาพนิ่ง + ภาพเคลื่อนไหว + เสียง
 - ii. I3D มีรูปแบบข้อมูลที่ป้อนเข้า คือ ภาพนิ่ง + ภาพเคลื่อนไหว
 - iii. TRN-Multiscale มีรูปแบบข้อมูลป้อนเข้า คือ ภาพนิ่ง + ภาพเคลื่อนไหว
 - iv. TSN-2stream มีรูปแบบข้อมูลป้อนเข้า คือ ภาพนิ่ง + ภาพเคลื่อนไหว
 - v. ResNet50-ImageNet มีรูปแบบข้อมูลป้อนเข้า คือ ภาพนิ่ง
- (b) เครื่องมือที่ใช้วัดผลงานวิจัยนี้
 - i. Classification accuracy Top-1, Top-5
- (c) ประสิทธิภาพของโมเดลปัญญาประดิษฐ์ในปัจจุบัน
 - i. ทำการทดสอบด้วยวิธี cross dataset transfer โดยการนำโมเดล ResNet50 I3D ที่สร้างด้วยชุดข้อมูล Kinetics และ Moments in Time และแล้วนำทั้ง 2 โมเดลไปทดสอบกับชุดข้อมูลอื่น โดยจะปรับอัตราเฟรมต่อวินาทีของวิดีโอให้เป็น 5 fps

| Pretrained | Fine-Tuned | | |
|------------|------------------------------|------------------------------|------------------------------|
| | UCF-101 | HMDB-51 | Something Something |
| Kinetics | Top-1 : 92.6 Top-5 : 99.2 | Top-1 : 62.0 Top-5 : 88.2 | Top-1 : 48.6 Top-5 : 77.9 |
| Moments | Top-1 : 91.9 Top-5 : 98.6 | Top-1 : 65.9 Top-5 : 89.3 | Top-1 : 50.0 Top-5 : 78.8 |

ตารางที่ 2.4: ประสิทธิภาพของโมเดล Resnet50 I3D ที่ใช้ชุดข้อมูล Kinetics และ Moments in Time

- ii. จะเห็นได้ว่า Kinetics ให้ผลลัพธ์ที่ดีกว่าใน UCF-101 เพราะว่ามีหมวดหมู่ที่ตรงกันอยู่หลายอย่าง ในขณะที่ HMDB-51 นั้นมีการรวบรวมข้อมูลจากหลายแหล่ง และมีจำนวนหมวดหมู่ที่หลากหลายจึงทำให้มีความใกล้เคียงกับตัวข้อมูลของ Moments in Time ดังนั้นจึงเทียบผลลัพธ์จาก Something Something ซึ่งจะทำให้เห็นว่า Moments in Time มีประสิทธิภาพที่ดีกว่าและวิดีโอที่มีความยาวมากกว่า 3 วินาทีจะไม่ส่งผลกระทบกับประสิทธิภาพของ Moments in Time

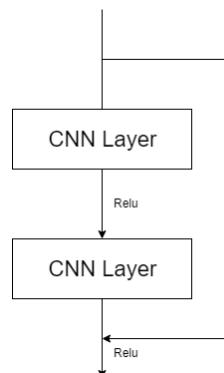
4. ปัญหาที่พบ

ผลลัพธ์จากการทำงานด้วยโมเดลถ้าผ่านภาพที่มีรายละเอียดเยอะจะทำให้การ ทำงานโอกาสผิดนั้นค่อนข้างสูง ซึ่งปัญหานี้สามารถทำให้เกิดน้อยลงด้วยการนำวิธี class activation mapping (CAM) จะเป็นการเน้นภาพในส่วนที่มีข้อมูลมากที่สุดและทำงานผลลัพธ์ แต่ก็ยังมีจุดที่เป็นปัญหาอยู่ เช่น การกระทำที่เกิดขึ้นเร็วมากจะทำให้การทำงานนั้นมีโอกาสผิดสูงขึ้น

2.4 โมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำมนุษย์

2.4.1 ResNet

ในการสร้างโมเดลปัญญาประดิษฐ์นั้น การใช้จำนวนขั้นเยือนนั้นจะทำให้ได้คุณลักษณะของข้อมูลที่อกรมา เยอะตามไปด้วย แต่การที่คุณลักษณะของข้อมูลเหล่านี้ไม่ได้หมายความว่าโมเดลปัญญาประดิษฐ์จะทำให้ประสิทธิภาพ ที่ดีเสมอไป ซึ่งสามารถแก้ปัญหานี้ได้โดยใช้ residual network (ResNet)^[40] ซึ่งเป็น CNN ประเภทหนึ่ง ที่ส่วนใหญ่จะนำมาใช้กับข้อมูลที่เป็นภาพ เช่น การจดจำวัตถุ เป็นต้น โดย ResNet นี้จะสามารถทำการข้ามชั้นที่ไม่ จำเป็นได้ การข้ามชั้นที่ไม่จำเป็นจะช่วยลดเวลาที่ใช้ในการสร้างโมเดลปัญญาประดิษฐ์ และทำให้ประสิทธิภาพ ของโมเดลปัญญาประดิษฐ์ดีขึ้น



รูปที่ 2.22: หลักการของ Residual block ของ ResNet

การทดลองโมเดลปัญญาประดิษฐ์ ResNet ด้วยการทำจำแนกภาพโดยใช้ชุดข้อมูลทดสอบ ImageNet ที่มี หมวดหมู่มากกว่า 1,000 หมวดหมู่ มาเทียบกับโมเดลปัญญาประดิษฐ์ทั่วไป (plain model) ที่จำนวนชั้น 18 ชั้น และ 34 ชั้น โดยโครงสร้างพื้นฐานของโมเดลปัญญาประดิษฐ์ ResNet และโมเดลปัญญาประดิษฐ์ทั่วไปเหมือนกัน ซึ่ง ผลลัพธ์อัตราเร็วอย่างของความผิดพลาดจะได้ออกมาตามตารางที่ 2.5

| จำนวนชั้นของโมเดลปัญญาประดิษฐ์ | Training error | |
|--------------------------------|----------------|--------|
| | plain | ResNet |
| 18 | 27.94 | 27.88 |
| 34 | 28.54 | 25.03 |

ตารางที่ 2.5: อัตราเร็วอย่างของความผิดพลาดของชุดข้อมูลทดสอบ ImageNet

จากตาราง 2.5 จะเห็นได้ว่าโมเดลปัญญาประดิษฐ์ทั่วไป 34 ชั้นมีค่าอัตราเร็วอย่างของความผิดพลาดสูงกว่า โมเดลปัญญาประดิษฐ์ ResNet ได้อย่างชัดเจน ในขณะที่โมเดลปัญญาประดิษฐ์ทั่วไปจะมีอัตราเร็วอย่างของความ ผิดพลาดสูงขึ้นเมื่อเทียบกันระหว่าง 18 ชั้นและ 34 ชั้น

ต่อมาจะนำโมเดลปัญญาประดิษฐ์ ResNet มาทดสอบกับชุดข้อมูล CIFAR-10 ซึ่งเป็นชุดข้อมูลที่มีรูปสำหรับใช้สร้างโมเดลปัญญาประดิษฐ์ 50,000 รูป รูปสำหรับทดสอบ 10,000 รูป และมีจำนวนหมวดหมู่ทั้งหมด 10 หมวดหมู่ โดยจะมีการออกแบบของจำนวนชั้นของโมเดลปัญญาประดิษฐ์ ResNet ตามจำนวนของชั้น convolution ที่มีผังคุณลักษณะเท่ากัน 6 ชั้นติดกันและการข้ามชั้นทีละ 2 ชั้น จึงทำให้ได้รูปแบบการคิดชั้นดังนี้ $6n + 2$ สำหรับการทดสอบจะให้ค่า $n = [3, 5, 7, 9, 200]$ ดังตารางต่อไปนี้

| โมเดลปัญญาประดิษฐ์ | จำนวนชั้น | Training error |
|--------------------|-----------|----------------|
| ResNet | 20 | 8.75 |
| ResNet | 32 | 7.51 |
| ResNet | 44 | 7.17 |
| ResNet | 56 | 6.97 |
| ResNet | 110 | 6.43 |
| ResNet | 1202 | 7.93 |

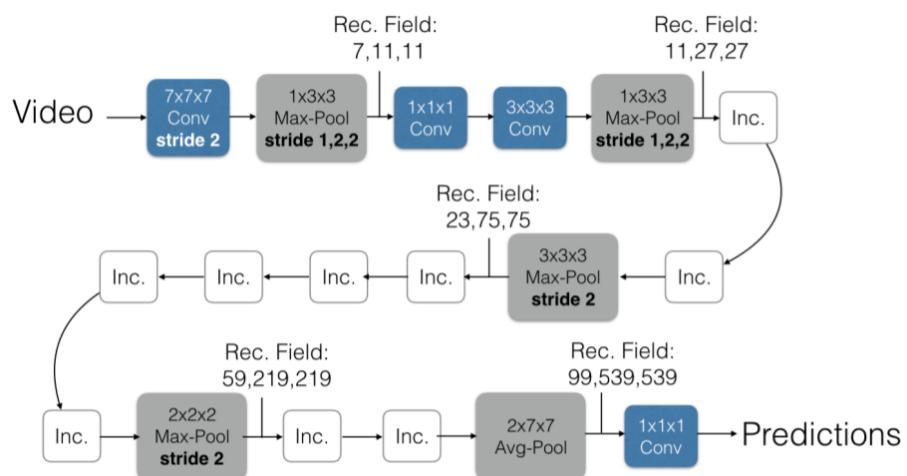
ตารางที่ 2.6: ค่าความผิดพลาดที่ได้จากการทดลองจำนวนชั้นของโมเดลปัญญาประดิษฐ์ ResNet บนชุดของข้อมูล CIFAR-10

จากตาราง 2.6 จะเห็นได้ว่าที่โมเดลปัญญาประดิษฐ์ ResNet ที่มีจำนวนชั้น 1,202 นั้นมีค่าความผิดพลาดเกิดขึ้นมากกว่าจำนวนชั้น 110 ซึ่งอาจจะเป็นไปได้ว่าขนาดของโมเดลปัญญาประดิษฐ์ ResNet ที่มีจำนวนชั้น 1,202 นั้นมากเกินไปสำหรับชุดข้อมูลขนาดเล็กนี้

2.4.2 Inflated 3D convolutional network

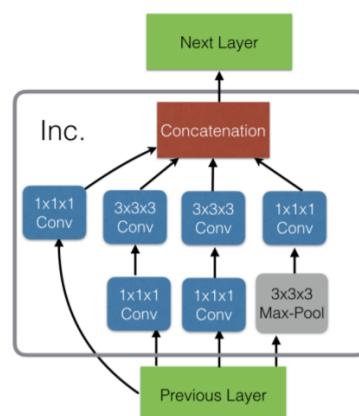
ในการพัฒนาโมเดลปัญญาประดิษฐ์สำหรับจำแนกการกระทำของมนุษย์นั้นมีพื้นฐานมาจากการจำแนกวัตถุ หมายถึงการใช้รูปภาพหนึ่งรูปในการประมวลผลและทำนายอุปกรณ์ว่าภายในรูปนั้นมีบริบทการกระทำอย่างไร โดยไม่ได้คำนึงถึงข้อมูลเชิงต่อเนื่อง (spatio-temporal information) จากบทความ "Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset"^[41] นั้นได้พัฒนาโครงสร้างของโมเดลปัญญาประดิษฐ์ที่มีประสิทธิภาพในการประมวลผลภาพเคลื่อนไหวได้ชื่อว่า I3D หรือ inflated 3D-convolution network โดยโครงสร้างพื้นฐานของ I3D นั้นมาจากการนำ Inception-v1^[42] ที่ถูกพัฒนาโดย Google ซึ่งเป็นโครงสร้างที่มีประสิทธิภาพสูงในการจำแนกวัตถุในรูปภาพ แล้ว I3D นั้นได้ทำการขยายมิติของเครื่องเรนาลจาก 2 มิติ เป็น 3 มิติ เพื่อให้โมเดลปัญญาประดิษฐ์สามารถเรียนรู้ข้อมูลเชิงต่อเนื่องได้

Inflated Inception-V1



รูปที่ 2.23: โครงสร้างของโมเดลปัญญาประดิษฐ์ I3D^[41]

Inception Module (Inc.)



รูปที่ 2.24: โครงสร้างของโมเดลปัญญาประดิษฐ์ I3D^[41]

จากรูปที่ 2.23 ในส่วนของชั้น Inception (Inc.) จะมีลักษณะโครงสร้างดังรูปที่ 2.24 เนื่องจากว่าการที่โมเดลนั้นซับชั้อนมากขึ้นก็ต้องใช้ทรัพยากรในการประมวลผลมากขึ้น Google จึงออกแบบโครงสร้างที่สามารถลดความซับซ้อนของโมเดลลงด้วยการใช้เครอร์เนลขนาด 1×1 ในเครอร์เนล 2 มิติ ($1 \times 1 \times 1$ ใน 3 มิติ) เพื่อลดจำนวน channel ของเครอร์เนลลง ตัวอย่างเช่น หากในชั้นก่อนหน้าได้ผลลัพธ์ที่มีขนาด $14 \times 14 \times 480$ (480 คือจำนวน channel ของเครอร์เนล) หากในชั้นถัดไปใช้เครอร์เนลที่มีขนาด $5 \times 5 \times 48$ จะทำให้มีจำนวนพารามิเตอร์ถึง $(14 \times 14 \times 480) \times (5 \times 5 \times 48) = 112.9$ ล้าน แต่ถ้าหากใช้เครอร์เนลขนาด $1 \times 1 \times 16$ มาคั่นระหว่างสองชั้นนี้จะทำให้จำนวนพารามิเตอร์ถูกย่อเป็น $(14 \times 14 \times 480) \times (1 \times 1 \times 16) = 1.5$ ล้าน และผลลัพธ์ของชั้นนี้จะมีขนาด $14 \times 14 \times 16$ ก่อนจะนำไปคำนวณในชั้นถัดไป $(14 \times 14 \times 16) \times (5 \times 5 \times 48) = 3.8$ ล้าน เมื่อนำจำนวนพารามิเตอร์รวมกันจะได้พารามิเตอร์เพียง $3.8 + 1.5 = 5.3$ ล้านเท่านั้น ซึ่งน้อยกว่าการใช้เครอร์เนลขนาด $5 \times 5 \times 48$ โดยตรง ซึ่งทำให้การพัฒนาโมเดลนั้นเป็นไปได้เร็วขึ้นมาก ทั้งยังสามารถลดปัญหาการเกิด overfit ได้ด้วย ประสิทธิภาพของโมเดล I3D แบบ two-stream เมื่อเทียบกับ long-short term memory (LSTM), 3D-convolution network, two-stream และ 3D-fused โดยใช้เครื่องมือในการวัดผลคือ Top@1 accuracy ตามตารางที่ 2.7

| Architecture | UCF-101 | | | HMDB-51 | | | Kinetics | | |
|----------------|---------|------|------------|---------|------|------------|----------|------|------------|
| | RGB | Flow | RGB + Flow | RGB | Flow | RGB + Flow | RGB | Flow | RGB + Flow |
| LSTM | 81.0 | – | – | 36.0 | – | – | 63.3 | – | – |
| 3D-ConvNet | 51.6 | – | – | 24.3 | – | – | 56.1 | – | – |
| Two-Stream | 83.6 | 85.6 | 91.2 | 43.2 | 56.3 | 58.3 | 62.2 | 52.4 | 65.6 |
| 3D-Fused | 83.2 | 85.8 | 89.3 | 49.2 | 55.5 | 56.8 | – | – | 67.2 |
| Two-Stream I3D | 84.5 | 90.6 | 93.4 | 49.8 | 61.9 | 66.4 | 71.1 | 63.4 | 74.2 |

ตารางที่ 2.7: ประสิทธิภาพของโมเดล I3D แบบ two-stream เมื่อใช้ข้อมูลจาก UCF-101, HMDB-51 และ Kinetics ในการสร้างและทดสอบด้วยเครื่องมือวัดผลแบบความแม่นยำจากการทำนายอันดับแรกสุด

2.5 เครื่องมือกำกับคุณลักษณะ

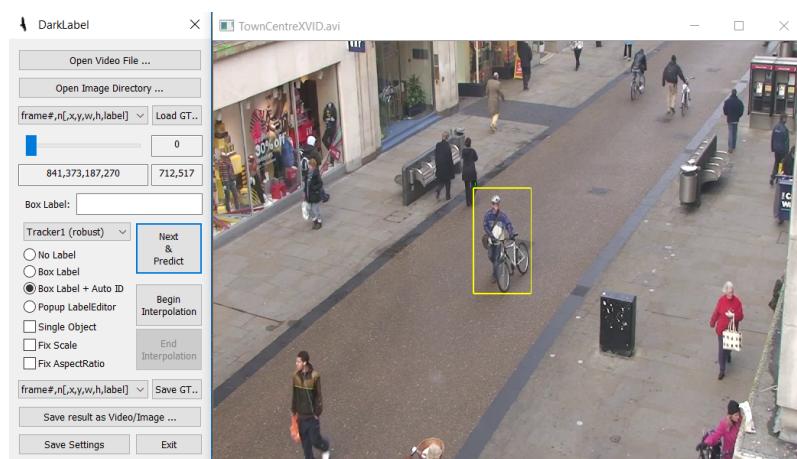
จากการค้นคว้าหาเครื่องมือในการสร้างคำกำกับข้อมูลเพื่อใช้เป็นแนวทางในการออกแบบเครื่องมือกำกับข้อมูลด้วยปัญญาประดิษฐ์ พบรเครื่องมือที่เปิดให้ใช้งานสาธารณะ 2 โปรแกรม คือ DarkLabel และ OpenLabeling โดยสรุปข้อมูลสำคัญได้ดังนี้

โปรแกรม DarkLabel^[43]

เป็นโปรแกรมที่ช่วยในการทำนายคำกำกับและบันทึกในรูปแบบต่างๆ รองรับข้อมูลป้อนเข้าในรูปแบบไฟล์วิดีโอ avi, mp4 หรือกลุ่มรูปภาพ มีขั้นตอนการสร้างคำกำกับดังนี้

1. สร้างกรอบสี่เหลี่ยมครอบบริเวณวัตถุที่สนใจโดยใช้มนุษย์เป็นคนสร้าง
2. กดปุ่ม Next และ Predict อย่างต่อเนื่อง เพื่อติดตามการเคลื่อนไหวของวัตถุในเฟรมถัดๆไป จนกระทั่งการเกิดข้อผิดพลาด
3. ลบกรอบสี่เหลี่ยมที่พลาด และเริ่มทำขั้นตอนที่ 1 ใหม่ อีกครั้งจนครบทุกเฟรมในวิดีโอ

หลังจากที่ผู้จัดได้ทดลองใช้โปรแกรม DarkLabel พบร่วมโปรแกรมมีการทำงานส่วนใหญ่ใช้มนุษย์ในการทำด้วยตัวเอง ซึ่งทำให้ใช้เวลาในการทำงาน



รูปที่ 2.25: หน้าต่างการทำงานของโปรแกรม DarkLabel

โปรแกรม OpenLabeling^[44]

เป็นโปรแกรมที่ช่วยในการสร้างคำจำกัด โดยโปรแกรมจะมีการทำงานอยู่ 2 รูปแบบการทำงาน คือแบบทำด้วยตัวเอง (Mode Manual) และแบบอัตโนมัติ (Mode Auto) ซึ่งมีการทำงานแยกกันอย่างชัดเจน

1. การทำงานแบบอัตโนมัติ

หลังจากป้อนวิดีโอเข้าไปในโปรแกรมแล้วมีขั้นตอนการสร้างคำจำกัดดังนี้

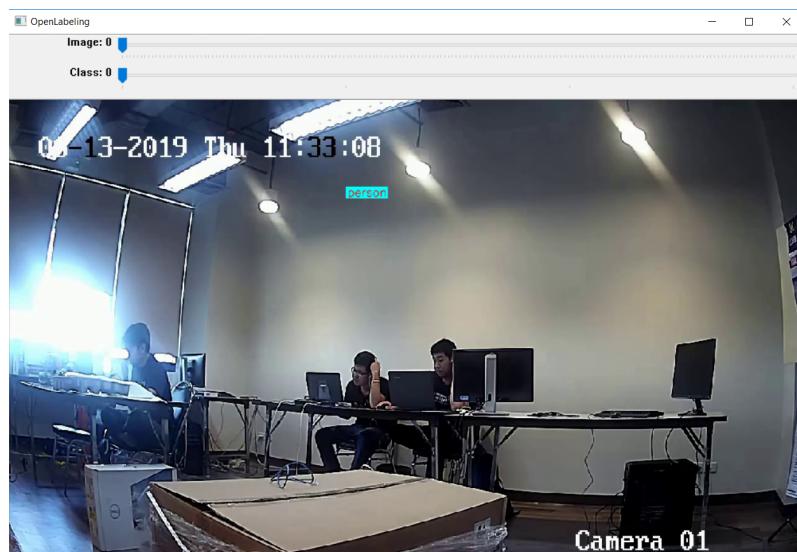
- (a) โปรแกรมจะทำงานอัตโนมัติโดยใช้โมเดลปัญญาประดิษฐ์ในการทำนายคีย์เฟรม (keyframe) และติดตามการเคลื่อนไหวตำแหน่งต่อไปของกรอบสีเหลี่ยมในเฟรมถัดไปด้วยอัลกอริทึมที่ใช้การคำนวนคอมพิวเตอร์และการประมวลผลภาพในภาพที่เหลือ ผลลัพธ์ที่ได้คือรูปภาพและไฟล์คำจำกัดภาพ

2. การทำงานแบบทำด้วยตัวเอง

หลังจากป้อนวิดีโอเข้าไปในโปรแกรมแล้วมีขั้นตอนการสร้างคำจำกัดดังนี้

- (a) สร้างกรอบสีเหลี่ยมขึ้นมาโดยใช้มนุษย์เป็นคนสร้าง
- (b) กดปุ่มเพื่อติดตามการเคลื่อนไหวตำแหน่งต่อไปของกรอบสีเหลี่ยมในเฟรมถัดไป จนกระทั่งเกิดข้อผิดพลาด
- (c) ลบกรอบสีเหลี่ยมที่พลาด และเริ่มทำขั้นตอนที่ 1 อีกครั้งจนครบทุกเฟรมในวิดีโอ

หลังจากที่ได้ทดลองใช้โปรแกรม OpenLabeling ทั้ง 2 รูปแบบการทำงานแล้วพบว่า การทำงานแบบอัตโนมัติไม่สามารถปรับแก้ไขสิ่งใดในระหว่างกระบวนการนั้น ทำให้หากเกิดกรณีที่ไม่สามารถทำนายกรอบสีเหลี่ยมพลาดจะไม่สามารถแก้ไขได้ และการทำงานแบบทำด้วยตัวเองผู้ใช้งานจะต้องสร้างกรอบสีเหลี่ยมขึ้นมาเอง



รูปที่ 2.26: หน้าต่างการทำงานของโปรแกรม OpenLabeling

บทที่ 3

ระเบียบวิธีวิจัย

ในการทำโครงการวิจัยเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ จะมีการทำงานหลากหลายส่วนมาทำงานร่วมกัน ซึ่งต้องมีระเบียบวิธีวิจัยอธิบายถึงขั้นตอนการดำเนินงานตั้งแต่เริ่มศึกษาข้อมูลจนไปถึงสิ้นสุดกระบวนการวิจัย

3.1 ความต้องการของระบบ

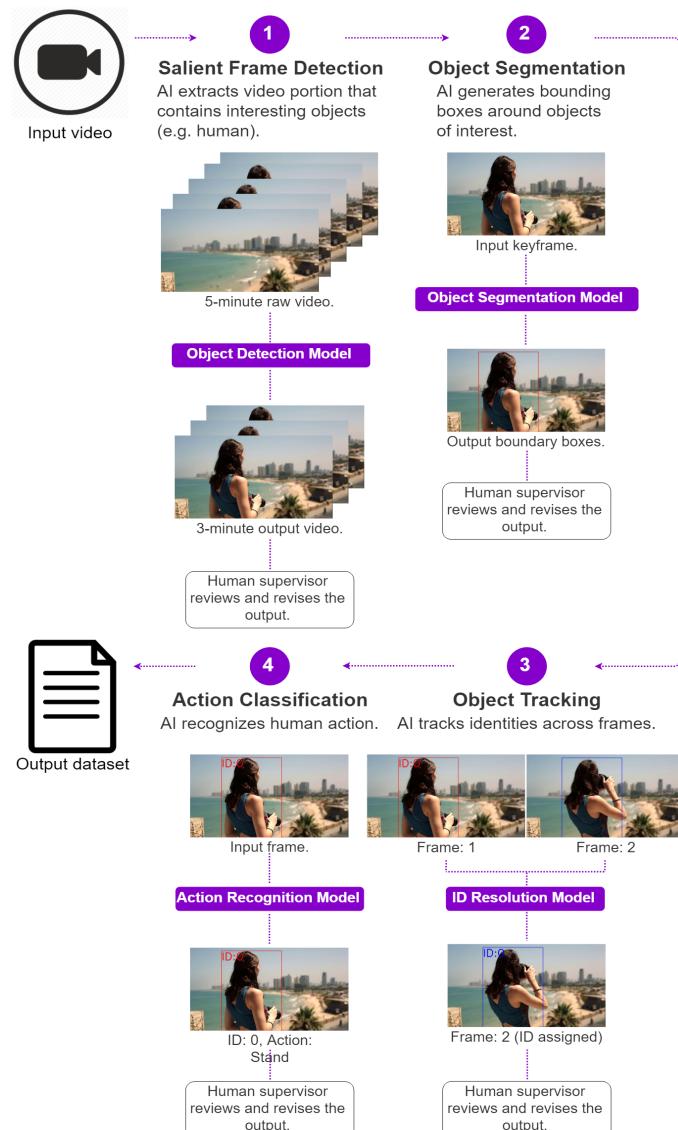
3.1.1 ความต้องการเชิงการใช้งาน (functional requirements)

1. เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ต้องสามารถตัดวิดีโอช่วงเวลาที่ไม่มีมนุษย์อยู่ออกได้อัตโนมัติ โดยใช้ปัญญาประดิษฐ์
2. เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์สามารถระบุตำแหน่งมนุษย์แต่ละคนในวิดีโอด้วย การกระทำของมนุษย์ในวิดีโอด้วย การกระทำที่กำหนดจะประกอบไปด้วย ยืน นั่ง นอน เล่นโทรศัพท์ เดิน กินข้าว
3. ชุดข้อมูลที่ได้จากการใช้งานมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ต้องสามารถนำไปใช้ในการพัฒนาโมเดล ปัญญาประดิษฐ์ต่อได้
4. สร้างระบบต้นแบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ที่มีมนุษย์สามารถทำงานร่วมกับ ปัญญาประดิษฐ์ได้
5. เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์สามารถนำวิดีโอมาวิเคราะห์ข้อมูลการกระทำและตำแหน่ง ของมนุษย์แต่ละคน และนำข้อมูลเหล่านั้นไปสร้างรายงานที่มีคำจำกัดอุปกรณ์ได้ โดยรายละเอียดรายงาน จะมีดังนี้
 - (a) ภาพ (Frame)
 - (b) ตำแหน่ง โดยจะบอกในลักษณะของกรอบสี่เหลี่ยมครอบพื้นที่ที่มีมนุษย์คนนั้นๆอยู่
 - (c) การกระทำ ซึ่งประกอบไปด้วย ยืน นั่ง เดิน เล่นโทรศัพท์ กินข้าว นอน

3.1.2 ความต้องการเชิงวิศวกรรม (non-functional requirements)

1. สร้างเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์โดยใช้ภาษาเพรอัน
2. ความละเอียดอย่างต่ำของวิดีโอต้องมากกว่า 640×480 (กว้าง x สูง)
3. วิดีโожะต้องมีอัตราเฟรมต่อวินาทีอย่างต่ำ 10 เฟรมต่อวินาที

3.2 ภาพรวมระบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์



รูปที่ 3.1: ภาพรวมระบบของเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

3.3 หน้าที่ความรับผิดชอบ

ปฐมพงศ์ สินธุ์งาม สร้างและทดสอบโมเดลปัญญาประดิษฐ์สำหรับจัดทำการกระทำมนุษย์ 3D รวมถึงออกแบบและสร้างระบบ Tracker

ศุภกร เบญจวิกรัย รวมฟังก์ชันและระบบต่างๆของเครื่องมือ รวมถึงออกแบบและสร้างระบบ Select และ Detect

อุกฤษฎ์ เลิศวรรณาการ สร้างและทดสอบโมเดลปัญญาประดิษฐ์สำหรับจัดทำการกระทำมนุษย์ Resnet-50 รวมถึงออกแบบและสร้างระบบ Person ReID

3.4 เครื่องมือที่ใช้ในงานวิจัย

ในหัวข้อนี้จะกล่าวถึงซอฟต์แวร์ ภาษา และ program library ที่ใช้ในการพัฒนาระบบ รวมถึงข้อมูลจำเพาะของคอมพิวเตอร์ที่ใช้ในการพัฒนาระบบ

Pycharm community 2017.1.2

เป็นโปรแกรมໄ่วยใช้สำหรับเขียนและแก้ไขโค้ดซึ่งข้อดีของโปรแกรมนี้ คือ มีคุณสมบัติต่างๆที่สามารถอำนวยความสะดวกในการเขียนโปรแกรมได้ เช่น syntax highlighting, auto-completion ฯลฯ และสามารถประมวลผลโปรแกรมทดสอบแอปพลิเคชันได้

Jupyter 2017.1.2

เป็นโปรแกรมสำหรับเขียนโปรแกรมที่เหมาะสมสำหรับใช้ในการทดสอบโปรแกรมแต่ละส่วนได้ ซึ่งมีข้อดีคือ หากมีการแก้ไขโปรแกรมเพียงแค่บางส่วน ก็สามารถประเมินผลเฉพาะส่วนที่ต้องการได้มักจะใช้ในการสร้างโมเดลปัญญาประดิษฐ์

Qt Creator 4.9.2 (community)

เป็นเครื่องมือสำหรับออกแบบหน้าต่างแอปพลิเคชันของ library PyQt ซึ่งมีข้อดีคือ เรียกใช้ง่ายมีวิดเจ็ต (widget) ที่สามารถใช้ได้หลากหลายเหมาะสมสำหรับการออกแบบ

3.5 ภาษาที่ใช้ในการพัฒนาระบบ

ใช้ภาษาไพธอนในการพัฒนาเป็นหลัก เพราะเป็นภาษาที่ปัจจุบันมีการใช้กันอย่างแพร่ มีเครื่องมือและ library ที่อำนวยความสะดวกในการพัฒนาอย่างมาก ทั้งยังเป็นภาษาที่สามารถเข้าใจได้ง่าย โดยในการทำวิจัยครั้งนี้ได้เลือก python 3.6.8 มาใช้ในการพัฒนา เนื่องจากเป็นรุ่นที่รองรับการทำงานของ library Tensorflow 1.12 และ CUDA 9

3.6 Program library ที่ใช้ในการพัฒนาระบบและแอปพลิเคชัน

| Library | Version | Description |
|--------------|----------|--|
| numpy | 1.16.4 | library ใช้สำหรับการคำนวนและ array |
| pandas | 0.24.2 | library ใช้สำหรับการจัดการข้อมูลที่อยู่ในรูปแบบของ excel |
| opencv | 4.1.0.25 | library ใช้สำหรับการจัดการข้อมูลที่เป็นรูปภาพและวิดีโอ |
| pillow | 6.0.0 | library ใช้สำหรับการจัดการข้อมูลที่เป็นรูปภาพ |
| torchsummary | 1.5.1 | library ใช้สำหรับการวิเคราะห์โครงสร้างของโมเดล |
| pytorch | 1.10.0 | library ใช้สำหรับการสร้างปัญญาประดิษฐ์ |
| torchvision | 0.3.0 | library ใช้สำหรับการสร้างปัญญาประดิษฐ์ |
| tensorflow | 1.12.0 | library ใช้สำหรับการสร้างปัญญาประดิษฐ์ |
| keras | 2.2.4 | library ใช้สำหรับการสร้างปัญญาประดิษฐ์ |
| scikit-learn | 0.21.2 | library ใช้สำหรับการสร้างปัญญาประดิษฐ์ |
| scipy | 1.3.0 | library ใช้สำหรับการสร้างปัญญาประดิษฐ์ |
| sklearn | 0.0 | library ใช้สำหรับการสร้างปัญญาประดิษฐ์ |
| pickleshare | 0.7.5 | library ใช้สำหรับการทำรหัส (encoding) โมเดลปัญญาประดิษฐ์ |
| tqdm | 4.32.1 | library ใช้สำหรับจัดการการทำงานซ้ำ (loop) |
| pyqt5 | 5.9.2 | library ใช้สำหรับการทำแอปพลิเคชัน |

3.7 แผนการดำเนินงาน

โดยจากที่กล่าวไปตอนต้นในบทนำการดำเนินงานและการออกแบบการสร้างเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ และระบบวิเคราะห์การกระทำการของมนุษย์ในวิดีโอ มีแผนการทำงานซึ่งถูกแบ่งออกเป็นสามขั้นตอนดังนี้

- ศึกษาหาความเป็นไปได้ รวมถึงเทคโนโลยีในปัจจุบันที่เกี่ยวกับการสร้างแอปพลิเคชัน และการจำแนกการกระทำการของมนุษย์ด้วยปัญญาประดิษฐ์ เพื่อศึกษาและทำความเข้าใจ ข้อดี-ข้อเสีย ของเทคนิคหรือกระบวนการต่างๆ เพื่อนำมาประยุกต์ใช้กับงานวิจัยนี้
- ออกแบบและสร้างแอปพลิเคชันที่ใช้ในการสร้างชุดข้อมูลสำหรับการสร้างโมเดลปัญญาประดิษฐ์จากวิดีโอ
- ออกแบบและสร้างระบบวิเคราะห์การกระทำการของมนุษย์ได้โดยมีข้อกำหนดตามที่กล่าวไว้ในบทนำ

ในการศึกษาเกี่ยวกับการออกแบบและการสร้างแอปพลิเคชันที่ใช้ในการสร้างชุดข้อมูลสำหรับการสร้างโมเดลจากวิดีโอ สิ่งที่ต้องให้ความสนใจคือฟังก์ชันการทำงาน การออกแบบและการจัดวางองค์ประกอบต่างๆ ในหน้าต่างแอปพลิเคชัน และความสะดวกในการใช้งาน จากนั้นจึงเริ่มศึกษาเกี่ยวกับ library ที่ใช้ในการสร้างแอปพลิเคชัน ส่วนการศึกษาเกี่ยวกับการสร้างระบบบุคลิกภาพที่การกระทำมนุษย์ จะมุ่งความสนใจไปที่ชุดข้อมูลสำหรับการบุคลิกภาพวิดีโอ โมเดลสำหรับการบุคลิกภาพวิดีโอด้วยเทคโนโลยีในการทำงานของมนุษย์ในวิดีโอด้วยมีประสิทธิภาพ ในบทนี้จะกล่าวถึงกระบวนการออกแบบและการดำเนินการตามแผนที่วางแผนไว้

3.8 การออกแบบหน้าต่างแอปพลิเคชันของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์

การออกแบบเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์ ผู้จัดได้เลือกใช้ library PyQt และภาษา Python ในการพัฒนา เนื่องจาก PyQt นั้นเป็น library ที่มีผู้พัฒนาใช้กันอย่างแพร่หลาย จึงสะดวกในการศึกษา หาข้อมูลในการสร้างหรือแก้ไข อีกทั้งยังเป็น library ที่สามารถพัฒนาด้วยภาษา Python ได้ และใช้งานง่าย สามารถปรับปรุงแก้ไขได้สะดวก

3.8.1 เครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์

แอปพลิเคชันแบ่งการทำงานออกเป็นสี่ส่วนประกอบด้วยกระบวนการ Select, Detect, Track และ Label เพื่อช่วยแบ่งเบาภาระของผู้พัฒนาในการสร้างชุดข้อมูลสำหรับสร้างโมเดลจากข้อมูลประเภทวิดีโอ โดยกระบวนการ Select จะต้องสามารถตัดวิดีโอส่วนที่ไม่มีมนุษย์อยู่ออกจากวิดีโอด้วย กระบวนการ Detect จะต้องหาตำแหน่งของมนุษย์ภายในวิดีโอด้วย แล้วใช้กระบวนการ Track ติดตามการเคลื่อนไหวตำแหน่งต่อไปของมนุษย์ในเฟรมถัดๆ ไป และกระบวนการสุดท้าย คือ Label นั้นต้องสามารถทำงานยกร่างกายที่พื้นฐานของมนุษย์ได้ เช่น ยืน เดิน นั่ง กินข้าว หรือ นอน เป็นต้น โดยทุกส่วนการทำงานมนุษย์ต้องสามารถทำงานร่วมกับปัญญาประดิษฐ์ได้ ดังรูปที่ 3.2

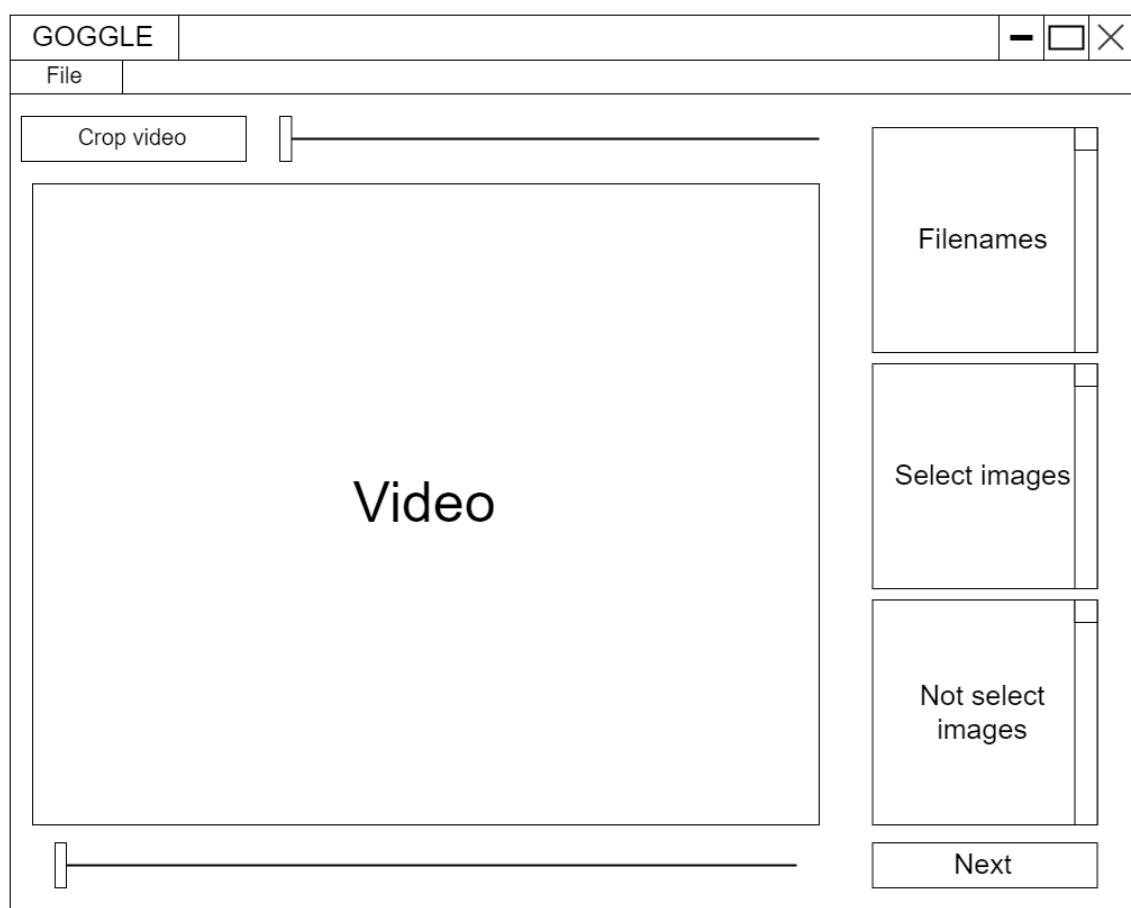


รูปที่ 3.2: กระบวนการหลักของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์

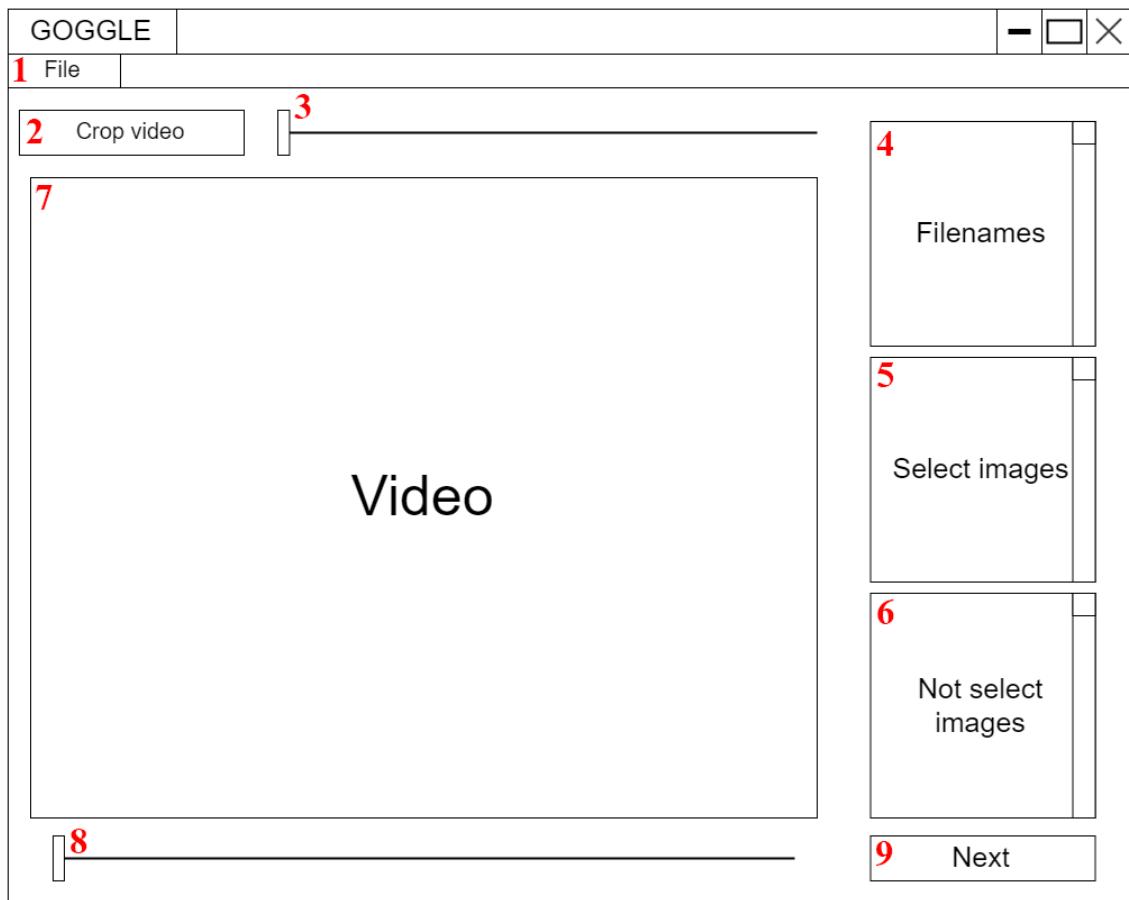
โดยแต่ละกระบวนการจะมีรายละเอียดดังนี้

หน้าต่าง Select

กระบวนการ Select จะต้องสามารถรับวิดีโอเข้ามา แล้วตัดวิดีโອนในช่วงที่ไม่มีมนุษย์อยู่ในเฟรมออกได้ อัตโนมัติด้วยปัญญาประดิษฐ์ แต่เนื่องจากการประมวลผลทุกเฟรมในวิดีโอนั้นจะทำให้เสียเวลามากเกินไป จึงใช้ วิธีการเลือกตัวอย่างเฟรมด้วยอัตราคงที่ (สามารถกำหนดได้) ซึ่งเรียกว่าเฟรมเหล่านี้ว่า คีย์เฟรม จากนั้นใช้ปัญญา ประดิษฐ์ประมวลผลคีย์เฟรมที่เหล่านั้น เพื่อลดระยะเวลาในการประมวลผลลง และมนุษย์จะต้องสามารถแก้ไข ข้อผิดพลาดของปัญญาประดิษฐ์ได้ เพื่อเพิ่มคุณภาพของข้อมูล จึงออกแบบหน้าต่างได้ดังรูปที่ 3.3



รูปที่ 3.3: หน้าต่าง Select ของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์



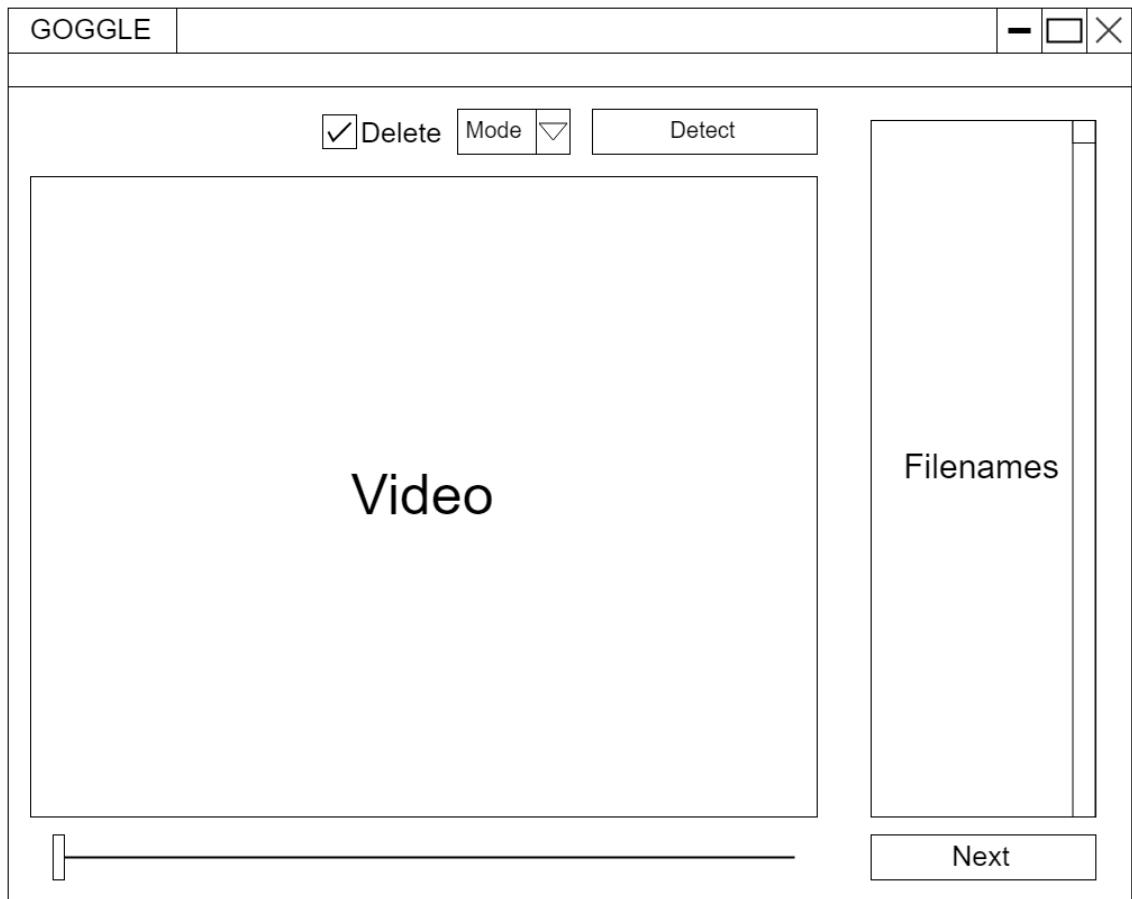
รูปที่ 3.4: ตำแหน่งของแต่ละวิดเจ็ตในหน้าต่าง Select

โดยที่แต่ละวิดเจ็ตตามหมายเลขที่กำหนดตามรูปที่ 3.4 มีรายละเอียดดังนี้

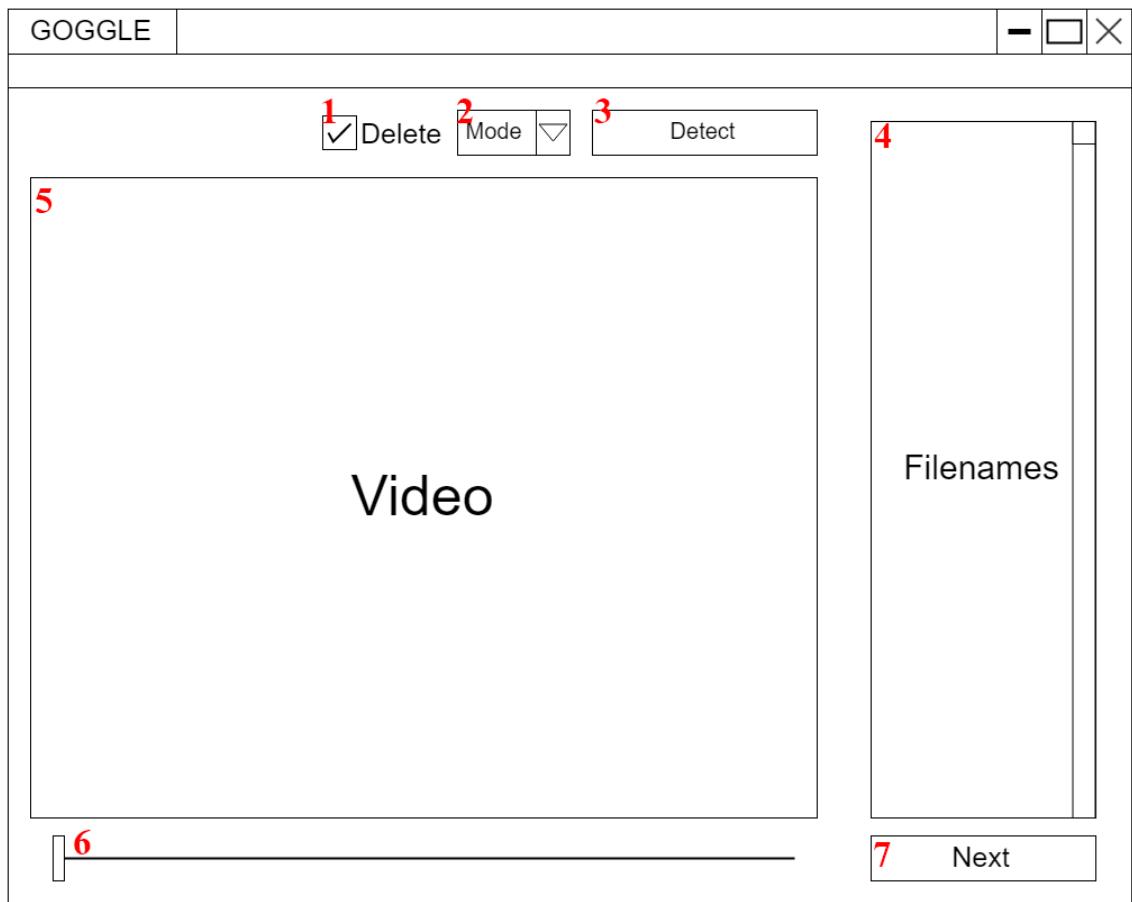
1. หมายเลข 1 คือปุ่มสำหรับเลือกไฟล์วิดีโอที่ต้องการจากในคอมพิวเตอร์เข้ามาในโปรแกรม
2. หมายเลข 2 คือปุ่มสำหรับสั่งให้ระบบทำการสร้างคีย์เฟรมขึ้นมา แล้วใช้ปัญญาประดิษฐ์ประมวลผลเพื่อแยกคีย์เฟรมใหม่มีคนอยู่ และคีย์เฟรมใหม่ไม่มีคนอยู่แบบอัตโนมัติ
3. หมายเลข 3 คือแถบเลื่อนเพื่อกำหนดความถี่ในการหยิบคีย์เฟรม โดยจะมีช่วงอยู่ที่ 1 เฟรมต่อวินาที จนถึงอัตราเฟรมต่อวินาทีสูงสุดของวิดีโอิที่รับเข้ามา
4. หมายเลข 4 คือกล่องสำหรับแสดงชื่อวิดีโอที่รับเข้ามาในโปรแกรมเพื่อเลือกเข้ามาใช้ในการประมวลผล
5. หมายเลข 5 คือกล่องสำหรับแสดงว่าคีย์เฟรมได้มีมนุษย์อยู่ในเฟรม โดยที่ผู้ใช้งานสามารถตรวจสอบความถูกต้องและแก้ไขข้อผิดพลาดของปัญญาประดิษฐ์ได้
6. หมายเลข 6 คือกล่องสำหรับแสดงว่าคีย์เฟรมได้มีมนุษย์อยู่ในเฟรม โดยที่ผู้ใช้งานสามารถตรวจสอบความถูกต้องและแก้ไขข้อผิดพลาดของปัญญาประดิษฐ์ได้
7. หมายเลข 7 คือหน้าต่างสำหรับแสดงเฟรมที่เลือกจากหมายเลข 5 หมายเลข 6 หรือหมายเลข 8
8. หมายเลข 8 คือแถบเลื่อนสำหรับเลือนดูคีย์เฟรมทั้งหมดที่ระบบสร้างขึ้น
9. หมายเลข 9 คือปุ่มสำหรับไปกระบวนการต่อไปหลังจากระบบประมวลผลเสร็จแล้ว

หน้าต่าง Detect

กระบวนการ Detect จะต้องสามารถรับคีย์เพرمจากกระบวนการ Select มาประมวลผลด้วยปัญญาประดิษฐ์เพื่อหาตำแหน่งของมนุษย์ที่อยู่ในคีย์เพرم และสร้างกรอบสีเหลี่ยมครอบบริเวณดังกล่าวได้ในแบบอัตโนมัติ เพื่อแบ่งเบาภาระผู้ใช้ในการที่ต้องสร้างกรอบสีเหลี่ยมครอบตำแหน่งของมนุษย์ด้วยตัวเอง และผู้ใช้ต้องสามารถสร้างหรือลบกรอบสีเหลี่ยมได้ด้วยตัวเองสำหรับแก้ไขความผิดพลาดของปัญญาประดิษฐ์ เพื่อเพิ่มคุณภาพของชุดข้อมูล จึงออกแบบหน้าต่างได้ดังรูปที่ 3.5



รูปที่ 3.5: หน้าต่าง Detect ของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์



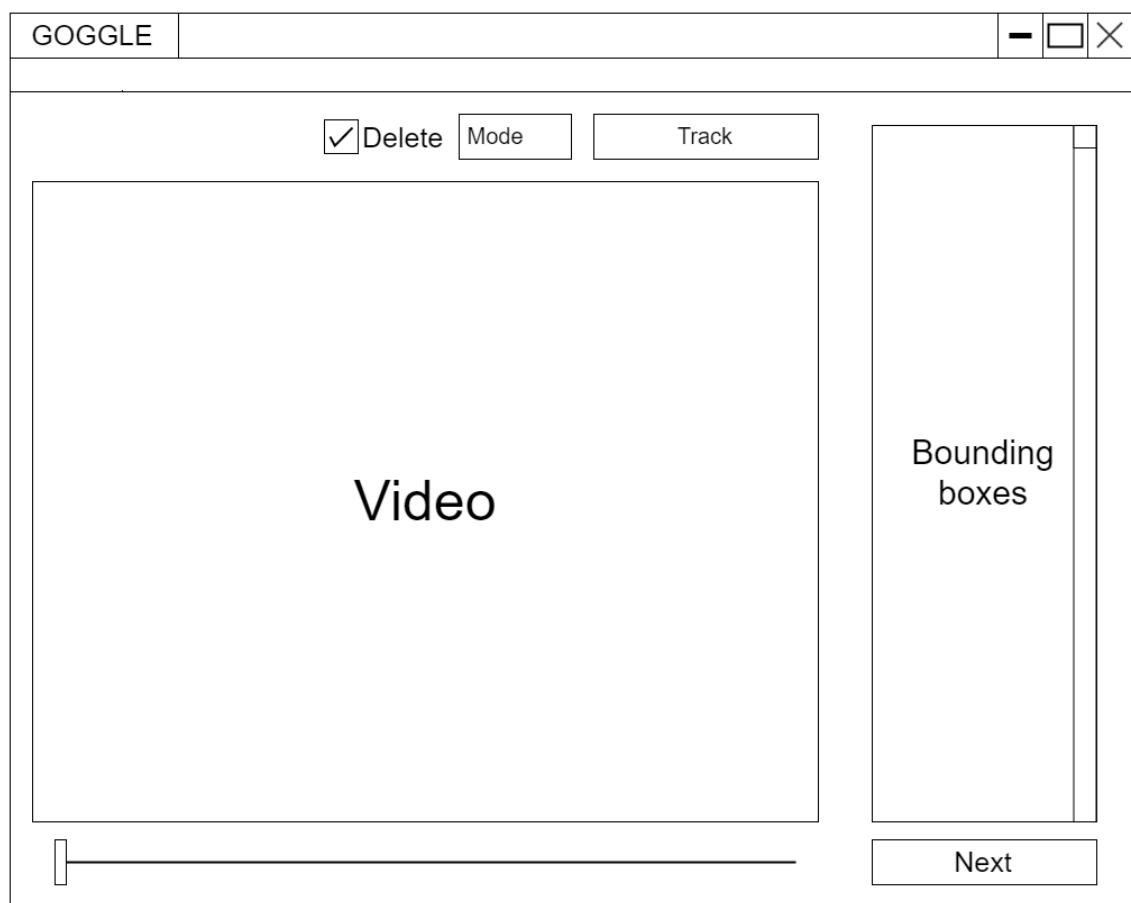
รูปที่ 3.6: ตำแหน่งของแต่ละวิดเจ็ตในหน้าต่าง Detect

โดยที่แต่ละวิดเจ็ตตามหมายเลขที่กำหนดตามรูปที่ 3.6 มีรายละเอียดดังนี้

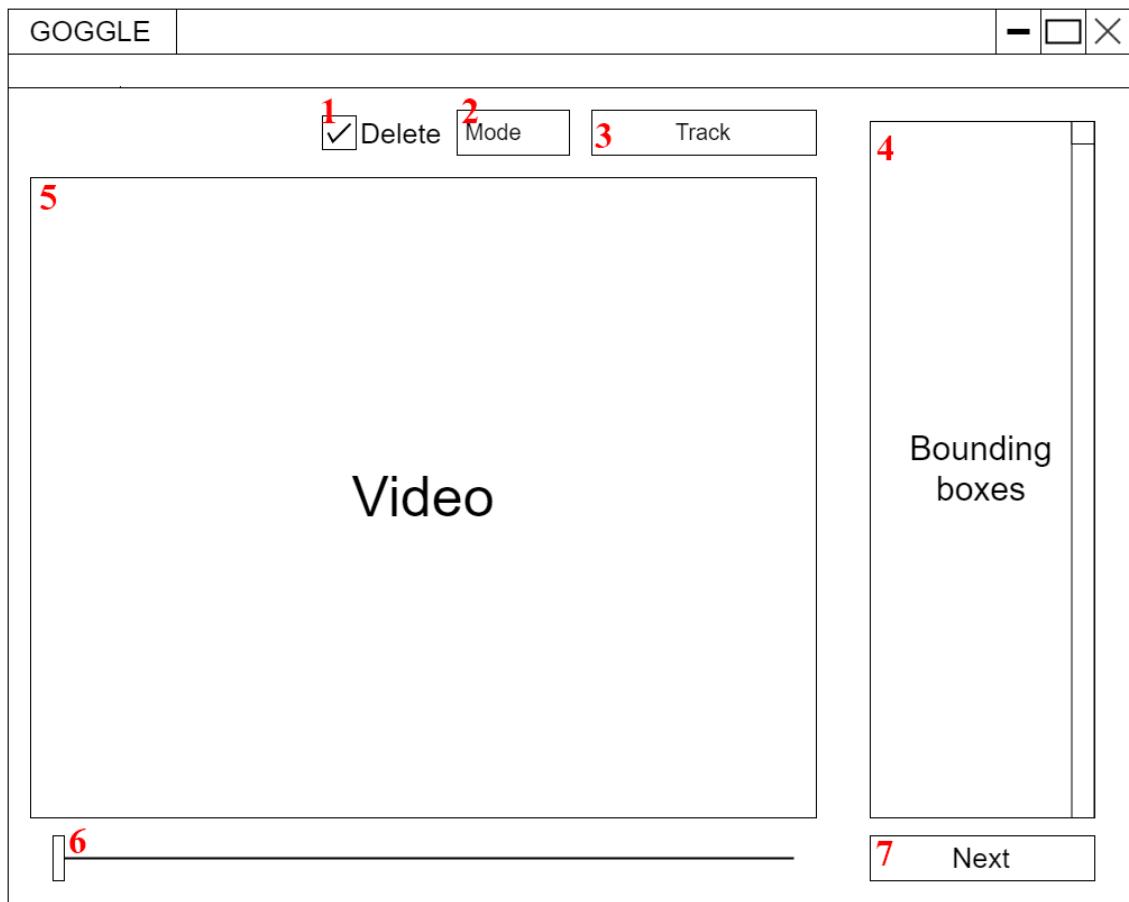
1. หมายเลข 1 คือช่องสำหรับกดเพื่อเปลี่ยนระบบจากสร้างกรอบสี่เหลี่ยมในแบบแก้ไขด้วยตนเองเป็นลบกรอบสี่เหลี่ยมแทน
2. หมายเลข 2 คือช่องสำหรับเลือกว่าจะใช้ระบบแบบใด ระหว่างแบบอัตโนมัติและแบบแก้ไขด้วยตนเอง
3. หมายเลข 3 คือปุ่มสำหรับสั่งให้ระบบทำการตรวจหาตำแหน่งของมนุษย์ในคิร์เฟรมทั้งหมดแล้วสร้างกรอบสี่เหลี่ยมขึ้นมาครอบบริเวณที่กำหนด
4. หมายเลข 4 คือกล่องสำหรับแสดงคิร์เฟรมทั้งหมด
5. หมายเลข 5 คือหน้าต่างสำหรับแสดงเฟรมที่เลือกจากหมายเลข 4 หรือหมายเลข 6
6. หมายเลข 6 คือแบบเลื่อนสำหรับเลื่อนดูคิร์เฟรมทั้งหมดที่มี เพื่อตรวจสอบความถูกต้องของปัญญาประดิษฐ์
7. หมายเลข 7 คือปุ่มสำหรับไปกระบวนการต่อไปหลังจากระบบประมวลผลเสร็จแล้ว

หน้าต่าง Track

เนื่องจากกระบวนการ Detect นั้นจะทำเฉพาะในคีย์เฟรมทำให้ในเฟรมอื่นๆ นอกเหนือจากนั้นจะไม่มีกรอบสี่เหลี่ยมอยู่ ดังนั้นกระบวนการ Track จึงต้องสามารถติดตามการเคลื่อนไหวตำแหน่งต่อไปของมนุษย์แล้วสร้างกรอบสี่เหลี่ยมขึ้นมาบนเฟรมระหว่างคีย์เฟรมทั้งหมดได้โดยอัตโนมัติ เพื่อสร้างข้อมูลตำแหน่งของมนุษย์ในเฟรมเหล่านั้น และผู้ใช้ต้องสามารถสร้างหรือลบกรอบสี่เหลี่ยมได้ด้วยตัวเองสำหรับแก้ไขความผิดพลาดของอัลกอริทึม จึงออกแบบหน้าต่างได้ดังรูปที่ 3.7



รูปที่ 3.7: หน้าต่าง Track ของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์



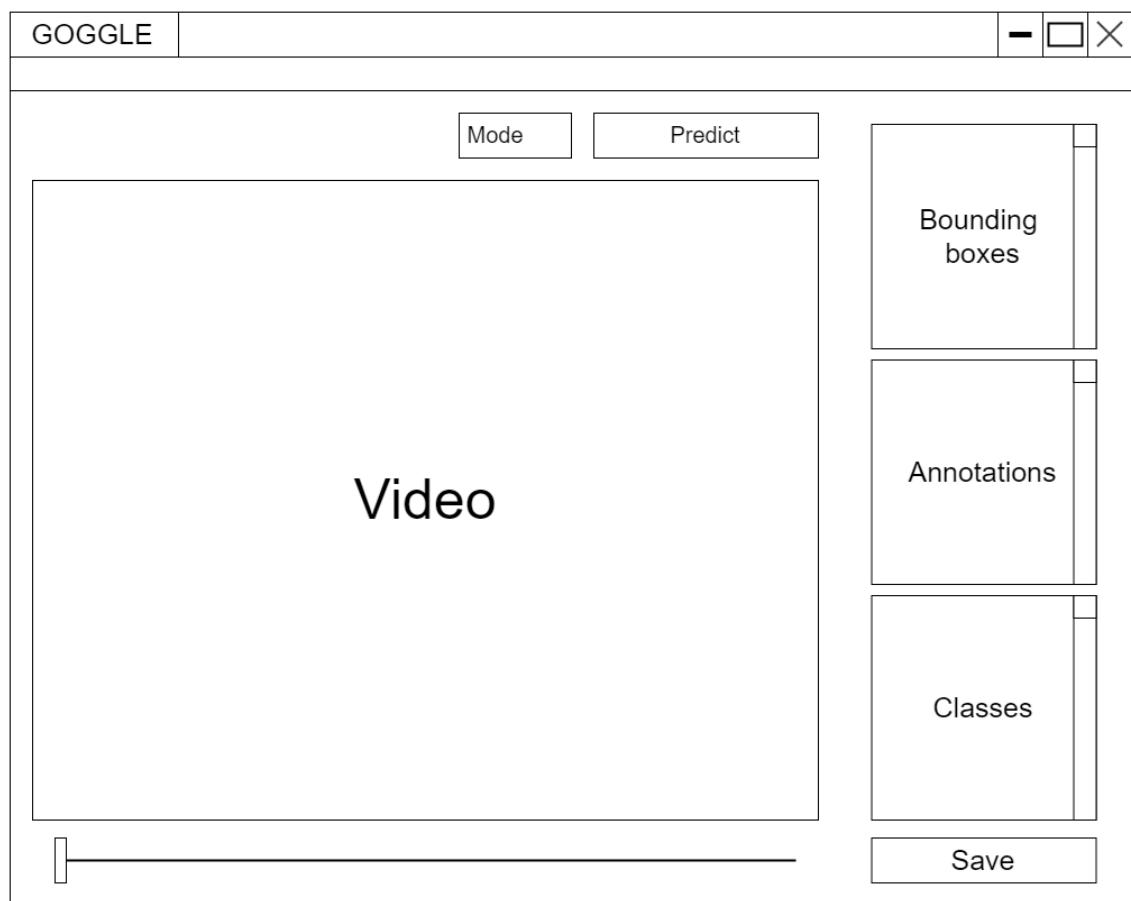
รูปที่ 3.8: ตำแหน่งของแต่ละวิดเจ็ตในหน้าต่าง Track

โดยที่แต่ละวิดเจ็ตตามหมายเลขที่กำหนดตามรูปที่ 3.8 มีรายละเอียดดังนี้

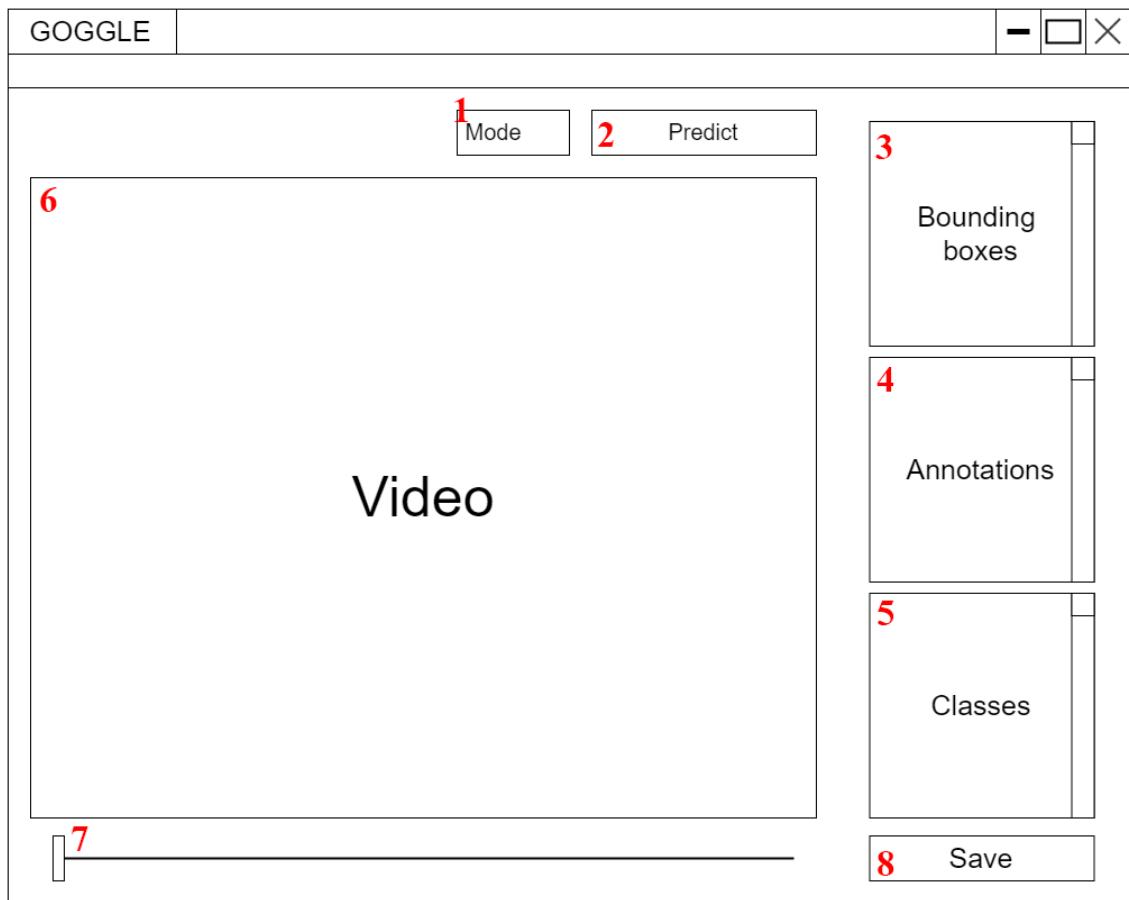
1. หมายเลข 1 คือช่องสำหรับกดเพื่อเปลี่ยนระบบจากสร้างกรอบสี่เหลี่ยมในแบบแก้ไขด้วยตนเองเป็นลบกรอบสี่เหลี่ยมแทน
2. หมายเลข 2 คือช่องสำหรับเลือกว่าจะใช้ระบบแบบใด ระหว่างแบบอัตโนมัติและแบบแก้ไขด้วยตนเอง
3. หมายเลข 3 คือปุ่มสำหรับสั่งให้ระบบทำการตรวจสอบตำแหน่งของมนุษย์ในเฟรมระหว่างคิ้ยวเฟรมทั้งหมดแล้วสร้างกรอบสี่เหลี่ยมขึ้นมาครอบบริเวณที่กำหนด
4. หมายเลข 4 คือกล่องสำหรับแสดงกรอบสี่เหลี่ยมทั้งหมดที่อยู่ในเฟรม
5. หมายเลข 5 คือหน้าต่างสำหรับแสดงเฟรมที่เลือกจากหมายเลข 6
6. หมายเลข 6 คือแถบเลื่อนสำหรับเลื่อนดูเฟรมทั้งหมดที่มี เพื่อตรวจสอบความถูกต้องของอัลกอริทึม
7. หมายเลข 7 คือปุ่มสำหรับไปกระบวนการต่อไปหลังจากระบบประมวลผลเสร็จแล้ว

หน้าต่าง Label

กระบวนการ Label นั้นต้องสามารถทำนายว่าการกระทำของมนุษย์ที่อยู่ในแต่ละเฟรมว่าคืออะไรได้โดยอัตโนมัติด้วยปัญญาประดิษฐ์ และผู้ใช้จะต้องสามารถแก้ไขข้อผิดพลาดของปัญญาประดิษฐ์ได้หากมีการทำนายที่ผิดพลาดเกิดขึ้น หรือถ้าหากผู้ใช้ต้องการเพิ่มการกระทำที่ไม่ได้มีอยู่ในชุดการกระทำพื้นฐานที่มีอยู่แล้วของปัญญาประดิษฐ์ ผู้ใช้ก็สามารถเพิ่มการกระทำนั้นเข้ามาได้ จึงออกแบบหน้าต่างได้ดังรูปที่ 3.9



รูปที่ 3.9: หน้าต่าง Label ของเครื่องมือสำหรับกำกับข้อมูลด้วยปัญญาประดิษฐ์



รูปที่ 3.10: ตำแหน่งของแต่ละวิดเจ็ตในหน้าต่าง Label

โดยที่แต่ละวิดเจ็ตตามหมายเลขที่กำหนดตามรูปที่ 3.8 มีรายละเอียดดังนี้

1. หมายเลข 1 คือช่องสำหรับเลือกว่าจะใช้ระบบแบบใด ระหว่างแบบอัตโนมัติและแบบแก้ไขด้วยตนเอง
2. หมายเลข 2 คือปุ่มสำหรับสั่งให้ระบบคำนวณรูปแบบของมนุษย์ในทุกๆเฟรม
3. หมายเลข 3 คือกล่องสำหรับแสดงกรอบสีเหลืองทั้งหมดที่อยู่ในเฟรมที่เลือก
4. หมายเลข 4 คือกล่องสำหรับแสดงการกระทำของมนุษย์แต่ละคนที่อยู่ในเฟรมที่เลือก โดยจะเรียงลำดับคู่กับกรอบสีเหลืองที่อยู่ในข้อหมายเหตุ 3
5. หมายเลข 5 คือกล่องสำหรับแสดงชุดการกระทำที่ปัญญาประดิษฐ์มีอยู่แล้ว ซึ่งในการทำงานแบบแก้ไขด้วยตนเองนั้น จะสามารถค้นหาการกระทำที่มีอยู่แล้วได้ และหากคำที่ใส่เขามานั้นมีอยู่ในชุดการกระทำก็จะเป็นการเพิ่มการกระทำนั้นเข้ามาแทน
6. หมายเลข 6 คือหน้าต่างสำหรับแสดงเฟรมที่เลือกจากหมายเลข 7
7. หมายเลข 7 คือແນບเลื่อนสำหรับเลื่อนดูเฟรมทั้งหมดที่มี เพื่อตรวจสอบความถูกต้องของปัญญาประดิษฐ์
8. หมายเลข 8 คือปุ่มสำหรับสร้างไฟล์ xml ของทุกๆเฟรมสำหรับใช้ในการสร้างโมเดลโดยรายละเอียดข้อมูลภายในไฟล์ xml จะอยู่ในหัวข้อ 3.8.1

รายละเอียดข้อมูลภายในไฟล์ xml

ไฟล์ xml นั้นเป็นรูปแบบที่นิยมใช้ในการเก็บข้อมูลสำหรับการสร้างโมเดลประเพณีตรวจสอบจับตัวตุ๊ก โดยจะเก็บข้อมูลในรูปแบบของ PASCAL VOC ที่นิยมใช้ในการสร้างโมเดลด้วย Tensorflow โดยภายในไฟล์จะมีข้อมูลดังรูปที่ 3.11 โดยข้อมูลส่วนสำคัญของรูปแบบนี้นั้นจะถูกใส่หมายเลขกำกับไว้ซึ่งแต่ละหมายเลขนั้นหมายถึง

```

<annotation>
    <folder>GeneratedData_Train</folder>1
    <filename>000001.png</filename>2
    <path>/my/path/GeneratedData_Train/000001.png</path>3
    <source>
        <database>Unknown</database>
    </source>
    <size> 4
        <width>224</width>
        <height>224</height>
        <depth>3</depth>
    </size>
    <segmented>0</segmented>
    <object>
        <name>21</name> 5
        <pose>Frontal</pose>
        <truncated>0</truncated>
        <difficult>0</difficult>
        <occluded>0</occluded>
        <bndbox> 6
            <xmin>82</xmin>
            <xmax>172</xmax>
            <ymin>88</ymin>
            <ymax>146</ymax>
        </bndbox>
    </object>
</annotation>
```

รูปที่ 3.11: ตัวอย่างข้อมูลภายในไฟล์ xml

- หมายเลข 1 คือชื่อโฟลเดอร์ที่เก็บไฟล์รูปภาพที่เกี่ยวข้องกับไฟล์ xml นี้อยู่
- หมายเลข 2 คือชื่อไฟล์ที่เกี่ยวข้องกับไฟล์ xml นี้
- หมายเลข 3 คือเส้นทางในคอมพิวเตอร์ (directory path) ของไฟล์รูปภาพที่เกี่ยวข้องกับไฟล์ xml นี้
- หมายเลข 4 คือขนาดและมิติของรูปภาพ ซึ่งจะประกอบด้วยความกว้าง (width) ความยาว (height) และจำนวนช่องสี (depth) โดยที่จำนวนช่องสีที่มีความลึก 3 มักจะหมายถึงภาพสี RGB และจำนวนช่องสีที่มีความลึก 2 จะหมายถึงภาพขาวดำ (gray scale)
- หมายเลข 5 คือ label ของวัตถุหรืออย่างอื่น ที่อยู่ในกรอบสีเหลี่ยมที่ถูกกำหนดไว้ในส่วนของหมายเลข 6
- หมายเลข 6 คือ กรอบสีเหลี่ยมที่ครอบวัตถุที่สนใจ เช่นมนุษย์ เป็นต้น

3.9 การออกแบบการทดสอบการตรวจจับวัตถุ

3.9.1 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์สำหรับการทำการตรวจจับภาพบุคคลสิ่งที่ใช้ในการวัดผล

1. ความเร็วในการทำนายต่อรูปภาพ (มิลลิวินาที)
2. ความแม่นยำ โดยคำนึงถึง IoU

จุดประสงค์

1. ผู้วิจัยได้ตั้งจุดประสงค์การทดลอง การใช้โมเดลปัญญาประดิษฐ์สำหรับตรวจจับวัตถุ เพื่อวัดผลโมเดลปัญญาประดิษฐ์ที่ใช้ในปัจจุบัน และหาโมเดลปัญญาประดิษฐ์สำหรับการทำการตรวจจับวัตถุที่มีความเร็วมากที่สุดและแม่นยำสูงที่สุดเมื่อทดสอบกับชุดข้อมูลของผู้วิจัย

ตัวแปร

1. โมเดลปัญญาประดิษฐ์ ได้แก่
 - (a) SSD MobileNet v1 ppn
 - (b) YOLO-v3 tiny
 - (c) YOLO-v3 spp
 - (d) YOLO-v3 320
 - (e) Faster RCNN inception v2

ตัวแปรควบคุม

1. ชุดข้อมูล : ชุดข้อมูลสำหรับทดสอบวัดผลที่ผู้วิจัยสร้างขึ้น (สูม 20 เฟรมจากวิดีโอที่ผู้วิจัยใช้สำหรับสร้างชุดข้อมูล)

วิธีการทดลอง

1. แบ่งชุดข้อมูลออกเป็นชุดข้อมูลสำหรับทดสอบ และชุดข้อมูลที่มีคำกำกับเพื่อเป็นคำตอบ
 - (a) ชุดข้อมูลสำหรับทดสอบ ประกอบด้วย : ชื่อของวิดีโอ และเฟรม
 - (b) ชุดข้อมูลที่มีคำกำกับเพื่อเป็นคำตอบ ประกอบด้วย : ชื่อของวิดีโอ เฟรม และตำแหน่งของกรอบสี่เหลี่ยม
2. เรียกชื่อและเฟรมของวิดีโอด้วยชุดข้อมูลทดสอบ และนำโมเดลปัญญาประดิษฐ์ทำนายผลลัพธ์ จากนั้นเก็บผลลัพธ์เป็นชุดข้อมูลผลลัพธ์จากการทำนาย
 - (a) ชุดข้อมูลผลลัพธ์จากการทำนาย ประกอบด้วย : ชื่อของวิดีโอ เฟรม และตำแหน่งของกรอบสี่เหลี่ยม
3. ประเมินผลค่าความแม่นยำในการทำงานโดยเทียบระหว่างชุดผลลัพธ์จากการทำนาย และชุดข้อมูลที่มีคำกำกับเพื่อเป็นคำตอบ โดยกำหนดให้ค่า IoU มากกว่าหรือเท่ากับ 0.5 จึงจะนับว่าทำนายได้ถูก
4. เปรียบเทียบผลลัพธ์จากแหล่งที่มา

3.10 การออกแบบการทดสอบการติดตามการเคลื่อนไหวตำแหน่งต่อไปของมนุษย์

3.10.1 ทดสอบประสิทธิภาพการทำงานของระบบการติดตามการเคลื่อนไหวตำแหน่งต่อไปของวัตถุในวิดีโอ สิ่งที่ใช้ในการวัดผล

1. ความเร็วในการคำนวณต่อวิดีโอ (วินาที)
2. ความแม่นยำ โดยคำนึงถึง IoU

วัตถุประสงค์

ผู้วิจัยมีความต้องการทดสอบความแม่นยำและความเร็วของการใช้โมเดลปัญญาประดิษฐ์สำหรับตรวจจับวัตถุและสร้างกรอบสี่เหลี่ยมทุกๆ N เพรอม และใช้ระบบติดตามการเคลื่อนไหวตำแหน่งต่อไปของวัตถุในการสร้างกรอบสี่เหลี่ยมในเฟรมระหว่างนั้น ว่าด้วยความเร็วที่เพิ่มขึ้นมากนั้นทำให้ความแม่นยำลดลงไปเท่าไหร่

ตัวแปรควบคุม

1. วิดีโอสารสนเทศที่ไม่ติดลิขสิทธิ์ ความยาวประมาณ 10 - 30 วินาที หนึ่งวิดีโอ
2. ใช้โมเดลปัญญาประดิษฐ์สำหรับตรวจจับตำแหน่งวัตถุ ResNet50 ในการสร้างชุดข้อมูลที่มีการกำกับตำแหน่งวัตถุไว้ และใช้มนุษย์ในการตรวจสอบความถูกต้อง เพื่อใช้เป็นค่าตอบของการทำนาย
3. โมเดลปัญญาประดิษฐ์สำหรับตรวจจับตำแหน่งที่ใช้ในการเบรียบเทียบ: YOLO-v3 spp
4. อัลกอริทึมสำหรับระบบการติดตามการเคลื่อนไหวตำแหน่งต่อไปของวัตถุ: DLib
5. IoU: มีส่วนที่ทับกันมากกว่า 80% ขึ้นไปจึงจะนับว่าผลการทำนายถูกต้อง

วิธีการทดลอง

1. ใช้โมเดลปัญญาประดิษฐ์ YOLO-v3 spp ประมวลผลทุกเฟรมในวิดีโอ และเบรียบเทียบผลลัพธ์กับชุดข้อมูลที่ถูกกำกับตำแหน่งวัตถุไว้แล้ว เพื่อคำนวณหาความแม่นยำ
2. ใช้โมเดลปัญญาประดิษฐ์ YOLO-v3 spp ประมวลผลทุกๆ N เพรอมในวิดีโอ และใช้ระบบติดตามการเคลื่อนไหวตำแหน่งต่อไปของวัตถุในการสร้างกรอบสี่เหลี่ยมในเฟรมระหว่างนั้น และเบรียบเทียบผลลัพธ์กับชุดข้อมูลที่ถูกกำกับตำแหน่งวัตถุไว้แล้ว เพื่อคำนวณหาความแม่นยำ โดยที่ค่า N จะเท่ากับ 10 20 และ 25
3. เบรียบเทียบความเร็วในการประมวลผล และความแม่นยำ

3.11 การออกแบบการทดสอบการระบุตัวตนของบุคคลภายในภาพ

3.11.1 ทดสอบประสิทธิภาพการทำงานของระบบระบุตัวตนของบุคคลภายในภาพ

สิ่งที่ใช้ในการวัดผล

- ค่า AUC ที่ใช้สำหรับการระบุตัวตนของบุคคลภายในภาพ

สมมติฐาน

ผู้จัดได้ตั้งสมมติฐานว่า ผลลัพธ์ของการทดลองการใช้งานจริงของโมเดลปัญญาประดิษฐ์ ResNet50 ที่สร้างด้วยชุดข้อมูล Market1501 นั้นควรจะมีความแม่นยำในการระบุตัวตนของบุคคลภายในภาพมากที่สุดเมื่อเทียบกับโมเดลปัญญาประดิษฐ์ที่สร้างด้วยชุดข้อมูลอื่นๆ เพราะเมื่อเทียบกับโมเดลปัญญาประดิษฐ์ที่ถูกสร้างด้วยชุดข้อมูลอื่นที่มาจากการแล่งข้อมูลเดียวกัน โมเดลปัญญาประดิษฐ์ ResNet50 ที่สร้างด้วยชุดข้อมูล Market1501 นั้นจะมีความแม่นยำสูงสุด

ตัวแปร

- โมเดลปัญญาประดิษฐ์ ซึ่งได้แก่
 - ResNet50 ที่ถูกสร้างด้วยชุดข้อมูล Market1501
 - ResNet50 ที่ถูกสร้างด้วยชุดข้อมูล DukeMTMCReID
 - ResNet50 ที่ถูกสร้างด้วยชุดข้อมูล CUHK03
 - ResNet50 ที่ถูกสร้างด้วยชุดข้อมูล MSMT17

ตัวแปรควบคุม

- ชุดข้อมูล : ชุดข้อมูลที่ทางผู้จัดสร้างขึ้นสำหรับการทดสอบ
- โมเดลปัญญาประดิษฐ์ : YOLO-V3 spp สำหรับการทำแท่งของบุคคล

วิธีการทดลอง

- นำชุดข้อมูลที่ผู้จัดสร้างขึ้นมาผ่านโมเดลปัญญาประดิษฐ์ YOLO-V3 spp เพื่อหาทำแท่งของบุคคล
- นำโมเดลปัญญาประดิษฐ์แต่ละอันมาทดสอบความแม่นยำสำหรับการระบุตัวตนของบุคคลภายในภาพ ด้วยทำแท่งของบุคคลที่ได้มาจากขั้นตอนก่อนหน้านี้
- ประเมินผลการทำงานโดยเทียบค่า AUC สำหรับการระบุตัวตนของบุคคลภายในภาพของแต่ละโมเดล ปัญญาประดิษฐ์ เพื่อหาโมเดลปัญญาประดิษฐ์ที่เหมาะสมกับชุดข้อมูลของผู้จัดมากที่สุด

3.12 การออกแบบการทดสอบการจำแนกการกระทำของนุษย์

3.12.1 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Faster RCNN ResNet101 ที่ถูกสร้างด้วยชุดข้อมูลของ AVA โดยใช้ชุดข้อมูลของ AVA ในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิงสิ่งที่ใช้ในการวัดผล

1. ความเร็วในการทำงานต่อรูปภาพ (มิลลิวินาที)
2. ความแม่นยำ (PASCAL mAP)

สมมติฐาน

ผู้วิจัยได้ตั้งสมมติฐานว่า ผลลัพธ์ของการทดลองจะมีความแม่นยำเทียบเท่ากับผลลัพธ์จากแหล่งที่มาและความเร็วต่อรูปภาพจะมีความเร็วมากกว่าผลลัพธ์จากแหล่งที่มา เนื่องจากแหล่งที่มาของข้อมูลได้ทำการทดสอบโดยใช้กราฟิกการ์ดรุ่น Nvidia GeForce GTX TITAN X ซึ่งเป็นกราฟิกการ์ดที่มีประสิทธิภาพการทำงานดีน้อยกว่ากราฟิกการ์ดของผู้วิจัย

ตัวแปรควบคุม

1. ชุดข้อมูล : The validation split of AVA v2.1
2. โมเดลปัญญาประดิษฐ์ : Faster RCNN ResNet101 AVA v2.1

วิธีการทดลอง

1. ดาวน์โหลดชุดข้อมูล The validation split of AVA v2.1
2. แบ่งชุดข้อมูลออกเป็น ชุดข้อมูลสำหรับทดสอบ และชุดข้อมูลที่มีคำกำกับเพื่อเป็นคำตอบ
 - (a) ชุดข้อมูลสำหรับทดสอบ ประกอบด้วย : ชื่อของวิดีโอ
 - (b) ชุดข้อมูลที่มีคำกำกับเพื่อเป็นคำตอบ ประกอบด้วย : ชื่อของวิดีโอ เพرم ตำแหน่งของกรอบสี่เหลี่ยม และรหัสของการกระทำ
3. เรียกชื่อของวิดีโอด้วยชุดข้อมูลทดสอบ และนำโมเดลปัญญาประดิษฐ์ที่มีคำกำกับเพื่อเป็นคำตอบ นำชุดข้อมูลผลลัพธ์จากการทำงาน
 - (a) ชุดข้อมูลผลลัพธ์จากการทำงาน ประกอบด้วย : ชื่อของวิดีโอ เพرم ตำแหน่งของกรอบสี่เหลี่ยม รหัสของการกระทำ และความมั่นใจ
4. ประเมินผลการทำงานโดยเทียบระหว่างชุดผลลัพธ์จากการทำงาน และชุดข้อมูลที่มีคำกำกับเพื่อเป็นคำตอบ
5. เปรียบเทียบผลลัพธ์จากแหล่งที่มา

3.12.2 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Faster RCNN ResNet101 ที่เคยถูกสร้างด้วยชุดข้อมูลของ AVA และใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิงสิ่งที่ใช้ในการวัดผล

1. ความในการทำงานเร็วต่อรูปภาพ (มิลลิวินาที)
2. ความแม่นยำ (PASCAL mAP)

สมมติฐาน

ผู้วิจัยได้ตั้งสมมติฐานว่าผลลัพธ์ของการทดลองจะมีความแม่นยำต่อลงเมื่อเทียบกับความแม่นยำของการทดลองที่ผ่านมา เนื่องจากชุดข้อมูลที่ผู้วิจัยสร้างขึ้น ได้มีการตัดหมวดหมู่บางอย่างออกไป ทำให้โมเดลปัญญาประดิษฐ์ที่ถูกสร้างด้วย AVA มีหมวดหมู่ของการกระทำไม่ตรงกับชุดข้อมูลที่ผู้วิจัยสร้างขึ้น ซึ่งมีผลทำให้ความแม่นยำลดลง ในส่วนของความเร็วต่อรูปภาพจะมีความเร็วมากกว่าผลลัพธ์จากแหล่งที่มา เนื่องจาก แหล่งที่มาของข้อมูลได้ทำการทดสอบโดยใช้กราฟิกการ์ดรุ่น Nvidia GeForce GTX TITAN X card ซึ่งเป็นกราฟิกการ์ดที่มีประสิทธิภาพการทำงานดีน้อยกว่า กราฟิกการ์ดของผู้วิจัย

ตัวแปรควบคุม

1. ชุดข้อมูล : ชุดข้อมูลที่ผู้วิจัยสร้างด้วยเครื่องมือกำกับคุณลักษณะ
2. โมเดลปัญญาประดิษฐ์ : Faster RCNN ResNet101 AVA v2.1

วิธีการทดลอง

1. แบ่งชุดข้อมูลออกเป็น ชุดข้อมูลสำหรับทดสอบ และชุดข้อมูลที่มีคำจำกัดเพื่อเป็นคำตอบ
 - (a) ชุดข้อมูลสำหรับทดสอบ ประกอบด้วย : ชื่อของวิดีโอ
 - (b) ชุดข้อมูลที่มีคำจำกัดเพื่อเป็นคำตอบ ประกอบด้วย : ชื่อของวิดีโอ เพرم ตำแหน่งของกรอบสีเหลือง และรหัสของการกระทำ
2. เรียกชื่อของวิดีโอด้วยชุดข้อมูลทดสอบ และนำโมเดลปัญญาประดิษฐ์ที่นายผลลัพธ์ จากนั้นเก็บผลลัพธ์ เป็นชุดข้อมูลผลลัพธ์จากการทำงาน
 - (a) ชุดข้อมูลผลลัพธ์จากการทำงาน ประกอบด้วย : ชื่อของวิดีโอ เพرم ตำแหน่งของกรอบสีเหลือง รหัสของการกระทำ และความมั่นใจ
3. ประเมินผลการทำงานโดยเทียบระหว่างชุดผลลัพธ์จากการทำงาน และชุดข้อมูลที่มีคำจำกัดเพื่อเป็นคำตอบ
4. เปรียบเทียบผลลัพธ์กับผลการทดลองที่ผ่านมา

3.12.3 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ ResNet50 ที่ถูกสร้างด้วยชุดข้อมูลที่ผู้วิจัยสร้างขึ้น และใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิง สิ่งที่ใช้ในการวัดผล

1. ความเร็วในการทำนายต่อรูปภาพ (มิลลิวินาที)
2. ความแม่นยำ (PASCAL mAP)

สมมติฐาน

ผู้วิจัยได้ตั้งสมมติฐานว่าผลลัพธ์ของการทดลองจะมีความแม่นยำสูงขึ้นเมื่อเทียบกับความแม่นยำของการทดลองที่ผ่านมา เนื่องจากโมเดลปัญญาประดิษฐ์ในการทดลองนี้ เป็นโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยได้สร้างขึ้น ซึ่งจะมีหมวดหมู่ของการกระทำของโมเดลปัญญาประดิษฐ์และชุดข้อมูลทดสอบตรงกัน ในส่วนของความเร็วต่อรูปภาพจะมีความเร็วมากกว่าผลลัพธ์จากแหล่งที่มา เนื่องจากแหล่งที่มาของข้อมูลได้ทำการทดสอบโดยใช้กราฟิกการ์ดรุ่น Nvidia GeForce GTX TITAN X ซึ่งเป็นกราฟิกการ์ดที่มีประสิทธิภาพการทำงานดีน้อยกว่ากราฟิกการ์ดของผู้วิจัย

ตัวแปรควบคุม

1. ชุดข้อมูล : ชุดข้อมูลที่ผู้วิจัยสร้างด้วยเครื่องมือกำกับคุณลักษณะ
2. โมเดลปัญญาประดิษฐ์ : โมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้น

วิธีการทดลอง

1. แบ่งชุดข้อมูลออกเป็น ชุดข้อมูลที่ใช้สำหรับการสร้างโมเดลปัญญาประดิษฐ์ และชุดข้อมูลสำหรับทดสอบ ที่มีคำจำกัดเพื่อเป็นคำตอบ โดยคำจำกัดจะประกอบไปด้วย : ชื่อของวิดีโอ เพรม ตำแหน่งของกรอบสี่เหลี่ยม และรหัสของการกระทำ
2. ทำการสร้างโมเดลปัญญาประดิษฐ์ด้วยชุดข้อมูลที่เตรียมไว้ และนำโมเดลปัญญาประดิษฐ์ที่มีประสิทธิภาพ ไปทดสอบ จำนวนนี้ เก็บผลลัพธ์จากการทำนาย ประกอบด้วย : ชื่อของวิดีโอ เพรม ตำแหน่งของกรอบสี่เหลี่ยม รหัสของการกระทำ และค่าความมั่นใจ โดยรายละเอียดของชุดข้อมูลทดสอบ 18,501 ภาพ มีดังนี้

| Label | Test set |
|------------|----------|
| Play phone | 3,888 |
| Eat | 3,000 |
| Sit | 5,591 |
| Sleep | 2,458 |
| Stand | 1,402 |
| Walk | 2,162 |

ตารางที่ 3.1: ตารางแสดงจำนวนภาพที่ใช้ในการทดลองนี้

3. มีการปรับเปลี่ยนการเตรียมข้อมูล เช่น การใช้ pretrain โมเดลหรือการปรับขนาดสเกลของชุดข้อมูลที่ใช้ในการสร้างโมเดลปัญญาประดิษฐ์
4. เปรียบเทียบผลลัพธ์ของโมเดลปัญญาประดิษฐ์ เพื่อเปรียบเทียบประสิทธิภาพ

3.12.4 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ I3D สร้างด้วยชุดข้อมูลที่ผู้วิจัยสร้างขึ้น โดยใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบ สิ่งที่ใช้ในการวัดผล

1. PASCAL mAP
2. Top@1 accuracy
3. Top@3 accuracy

สมมติฐาน

ผู้วิจัยได้ตั้งสมมติฐานว่า โมเดลปัญญาประดิษฐ์ I3D ที่ถูกสร้างด้วยชุดข้อมูลแบบ optical flow จะมีความแม่นยำสูงกว่าโมเดลปัญญาประดิษฐ์ที่ถูกสร้างด้วยชุดข้อมูลแบบปกติ

ตัวแปรควบคุม

1. ชุดข้อมูล : ชุดข้อมูลที่ผู้วิจัยสร้างด้วยเครื่องมือกำกับคุณลักษณะ
2. โมเดลปัญญาประดิษฐ์ : I3D

วิธีการทดลอง

1. แบ่งชุดข้อมูลทั้งหมดในแต่ละการกระทำให้เป็นชุด ชุดละ 20 - 40 เฟรม
2. แบ่งชุดข้อมูลออกเป็น ชุดข้อมูลสำหรับสร้างโมเดลปัญญาประดิษฐ์ ชุดข้อมูลสำหรับทดสอบ และชุดข้อมูลที่มีคำจำกัดเพื่อเป็นคำตอบ โดยจะมีรายละเอียดดังนี้

| Label | Train set | Validation set | Test set |
|------------|-----------|----------------|----------|
| Play phone | 705 | 191 | 96 |
| Eat | 825 | 165 | 83 |
| Sit | 775 | 206 | 104 |
| Sleep | 299 | 96 | 48 |
| Stand | 426 | 108 | 54 |
| Walk | 261 | 75 | 38 |

ตารางที่ 3.2: ตารางแสดงจำนวนชุดของข้อมูลที่ใช้ในการทดลองนี้

3. สร้างโมเดลด้วยโครงสร้าง I3D และชุดข้อมูลที่ผู้วิจัยสร้างด้วยเครื่องมือกำกับคุณลักษณะทั้งแบบปกติและแบบ optical flow จากนั้นทดสอบและบันทึกผล
4. ปรับเปลี่ยนพารามิเตอร์บางตัวของโมเดลปัญญาประดิษฐ์เพื่อเปรียบเทียบประสิทธิภาพ
5. เปรียบเทียบผลลัพธ์ของโมเดลปัญญาประดิษฐ์

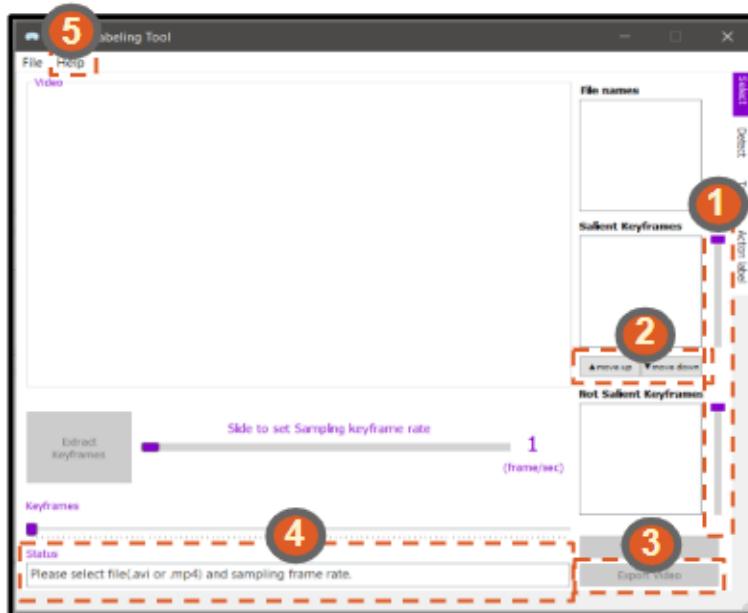
บทที่ 4

ผลการดำเนินงาน

4.1 เครื่องมือกำกับคุณลักษณะ

4.1.1 หน้าต่างแสดงผลของแอปพลิเคชัน

หน้าต่าง Select

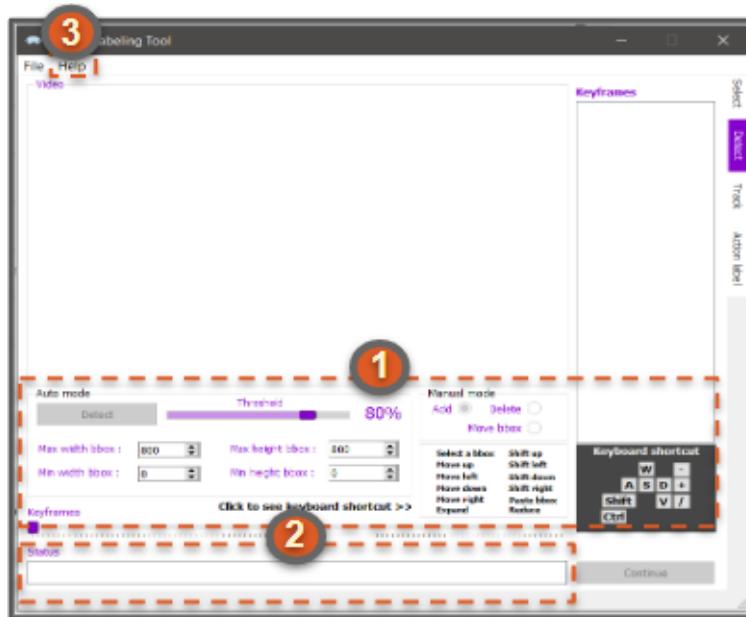


รูปที่ 4.1: รูปหน้าต่างแสดงผลของหน้าต่าง Select

จากรูปที่ 4.1 แสดงหน้าต่าง Select ของแอปพลิเคชัน ซึ่งเมื่อเทียบกับฉบับร่างตามรูปที่ 3.3 จะมีส่วนที่เพิ่มเติมขึ้นมาดังนี้

1. แถบเลื่อนสำหรับเลื่อนดูเฟรมที่มีมนุษย์หรือไม่มีมนุษย์ เพื่อเพิ่มความสะดวกในการเลือกดูเฟรม
2. ปุ่มสำหรับแก้ไขเฟรมที่มีมนุษย์หรือไม่มีมนุษย์
3. แถบแสดงสถานะกระบวนการทำงาน
4. ปุ่มสำหรับนำผลลัพธ์ออกเป็นไฟล์วิดีโอเฉพาะในช่วงที่มีมนุษย์อยู่
5. แถบสำหรับคำแนะนำช่วยเหลือ

หน้าต่าง Detect

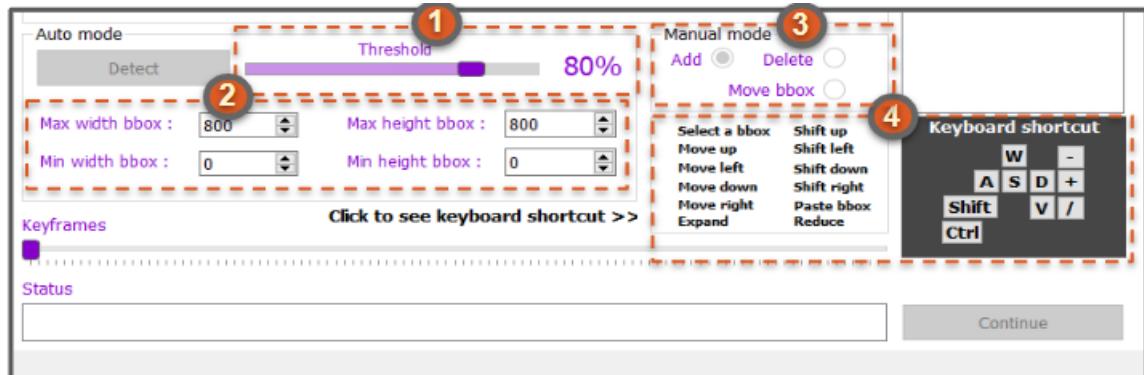


รูปที่ 4.2: รูปหน้าต่างแสดงผลของหน้าต่าง Detect

จากรูปที่ 4.2 แสดงหน้าต่าง Detect ของแอปพลิเคชัน ซึ่งเมื่อเทียบกับฉบับร่างตามรูปที่ 3.5 จะมีส่วนที่เพิ่มเติมขึ้นมาดังนี้

1. ปรับหน้าตารูปแบบการทำงานแบบอัตโนมัติ และกำหนดเองสามารถใช้งานได้หลากหลายขึ้น และเพิ่มความหลากหลายในการปรับแก้ในการทำงานอัตโนมัติ
2. ແນບแสดงสถานะกระบวนการทำงาน
3. ແນບสำหรับคำแนะนำช่วยเหลือ

รายละเอียดของการปรับแก้ไขของโหมดการทำงานอัตโนมัติและกำหนดเอง



รูปที่ 4.3: รูปหน้าต่างแสดงผลของหน้าต่าง Detect

จากรูปที่ 4.3 แสดงรายละเอียดของการปรับแก้ไขการตั้งค่าของโหมดการทำงานแบบอัตโนมัติและกำหนดเอง เพื่อผลลัพธ์ที่ตรงความต้องการของผู้ใช้งาน ซึ่งจะมีการตั้งค่าดังนี้

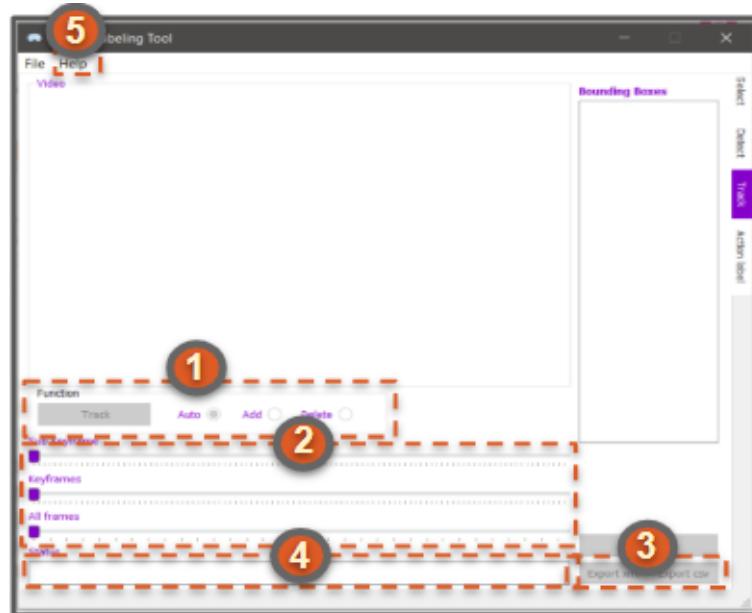
1. โหมดการทำงานแบบอัตโนมัติ

- การปรับค่า threshold ซึ่งคือค่าที่ใช้คัดกรองผลลัพธ์ โดยจะคัดกรองจากผลลัพธ์จะต้องมีค่าความมั่นใจในการทำนายมากกว่าค่า threshold
- การปรับค่า มากสุด-น้อยสุด ของ ความสูงและความกว้าง

2. ແບບแสดงสถานะกระบวนการทำงาน

- การเพิ่ม,ลด กรอบสี่เหลี่ยม
- การปรับโดยใช้คีย์ลัดจากปุ่มคีย์บอร์ด
 - การเลื่อนตำแหน่งกรอบสี่เหลี่ยม (บน , ล่าง , ซ้าย , ขวา)
 - การปรับขนาดของกรอบสี่เหลี่ยม (ขยายกรอบสี่เหลี่ยมทุกด้าน , ย่อกรอบสี่เหลี่ยมทุกด้าน , ขยายกรอบสี่เหลี่ยมด้านบน , ขยายกรอบสี่เหลี่ยมด้านล่าง , ขยายกรอบสี่เหลี่ยมด้านซ้าย , ขยายกรอบสี่เหลี่ยมด้านขวา)
 - การเลือกกรอบสี่เหลี่ยมที่ต้องการปรับ
 - การตัดลอกขนาดและตำแหน่งของกรอบสี่เหลี่ยม และ วางในเฟรมด้านไป

หน้าต่าง Track

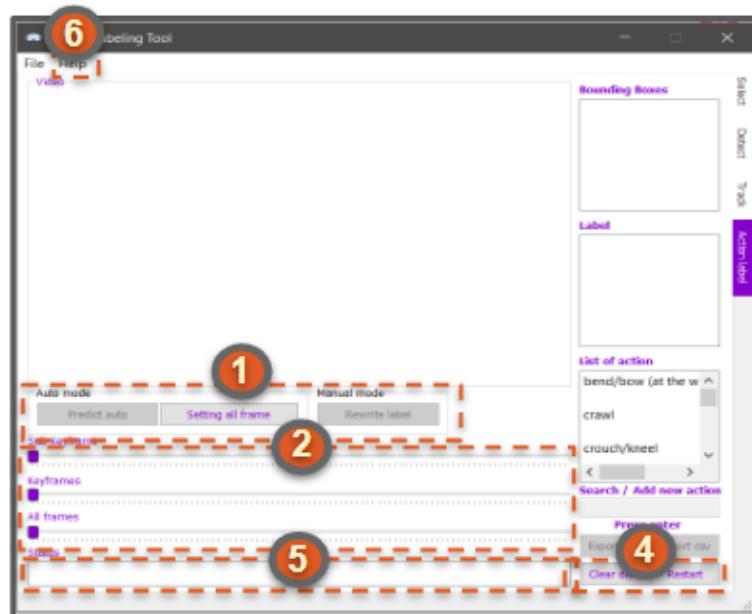


รูปที่ 4.4: รูปหน้าต่างแสดงผลของหน้าต่าง Track

จากรูปที่ 4.4 แสดงหน้าต่าง Track ของแอปพลิเคชัน ซึ่งเมื่อเทียบกับฉบับร่างตามรูปที่ 3.7 จะมีส่วนที่เพิ่มเติมขึ้นมาดังนี้

1. ปรับหน้าตารูปแบบการทำงานแบบอัตโนมัติและกำหนดเองจากฉบับร่างเพื่อให้สามารถใช้งานได้สะดวกขึ้น
2. เพิ่มแถบเลื่อนเป็น 3 แถบเลื่อนทำให้สามารถดูคีย์เฟรมและเฟรมที่อยู่ระหว่างช่วงคีย์เฟรมได้สะดวกขึ้น
3. เพิ่มปุ่มสำหรับนำผลลัพธ์ออกเป็นไฟล์ xml หรือ ออกเป็นไฟล์ csv
4. แถบแสดงสถานะกระบวนการทำงาน
5. แถบสำหรับคำแนะนำช่วยเหลือ

หน้าต่าง Label



รูปที่ 4.5: รูปหน้าต่างแสดงผลของหน้าต่าง Label

จากรูปที่ 4.5 แสดงหน้าต่าง Label ของแอปพลิเคชัน ซึ่งเมื่อเทียบกับกับฉบับร่างตามรูปที่ 3.9 จะมีส่วนที่เพิ่มเติมขึ้นมาดังนี้

1. ปรับหน้าตารูปแบบการทำงานแบบอัตโนมัติและกำหนดเองจากฉบับร่างเพื่อให้สามารถใช้งานได้สะดวกขึ้น
2. เพิ่มแถบเลื่อนเป็น 3 แถบเลื่อนทำให้สามารถคุยกับเฟรมและเฟรมที่อยู่ระหว่างช่วงคีย์เฟรมได้สะดวกขึ้น
3. เครื่องมือสำหรับค้นหาหรือเพิ่มหมวดหมู่ของการกระทำ
4. แถบแสดงสถานะกระบวนการทำงาน
5. ปุ่มสำหรับลบข้อมูล และ เริ่มต้นการทำงานใหม่
6. แถบสำหรับคำแนะนำช่วยเหลือ

4.1.2 ผลลัพธ์การทำงานในแต่ละหน้าต่างของแอปพลิเคชัน

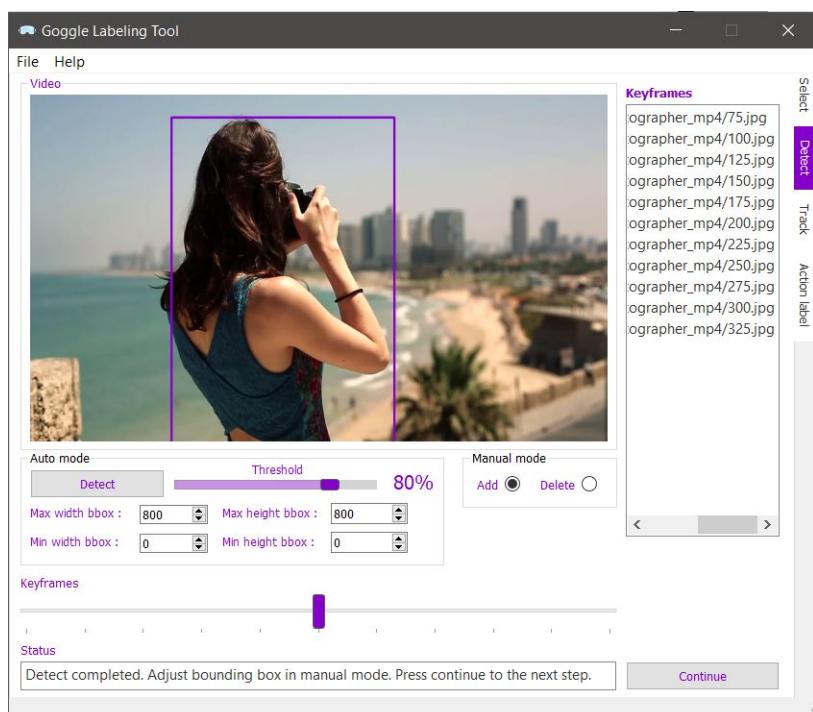
ผลลัพธ์การทำงานของหน้าต่าง Select



รูปที่ 4.6: รูปผลลัพธ์การแยกเฟรมที่มีมนุษย์อยู่ และไม่มีมนุษย์อยู่ภายในเฟรม

ขั้นตอนแรก แอปพลิเคชัน จะ สกัด แยก วิดีโอ ออก เป็น เฟรม ทั้งหมด และ ทำการ สุ่ม คีย์เฟรม ออกมา ตาม ความถี่ ที่ ผู้ใช้ ตั้ง ไว้ จากนั้น แอปพลิเคชัน จะ นำ โมเดล YOLO-v3 320 มา ตรวจสอบ ว่า แต่ละ คีย์เฟรม มี เฟรม ใด บ้าง ที่ มี มนุษย์ อยู่ ภายใน เฟรม จากนั้น จะ ทำการ แยก เฟรม ที่ มี มนุษย์ อยู่ และ ไม่มี มนุษย์ อยู่ ดัง รูปที่ 4.6

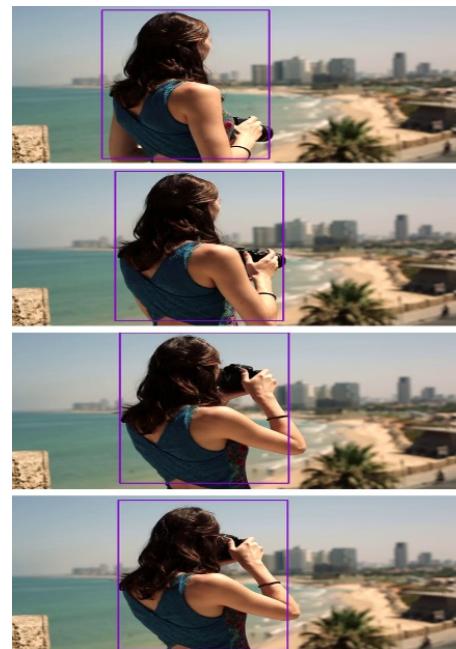
ผลลัพธ์การทำงานของหน้าต่าง Detect



รูปที่ 4.7: รูปคีย์เฟรมที่ถูกตีกรอบสีเหลืองในส่วนที่มีมนุษย์อยู่

แอปพลิเคชันจะนำคีย์เฟรมที่มีนุชย์ที่ได้จากหน้าต่าง Select นำมาตีกรอบสีเหลี่ยมในส่วนของเฟรมที่มีมนุชย์อยู่โดยสามารถใช้รูปแบบการทำงานแบบอัตโนมัติหรือแบบแก้ไขด้วยตัวเองก็ได้ ซึ่งผลลัพธ์ที่ได้จะได้คีย์เฟรมที่มีกรอบสีเหลี่ยม ดังรูปที่ 4.7 จากนั้นจะบันทึกข้อมูลในไฟล์ txt

ผลลัพธ์การทำงานของหน้าต่าง Track



(ก) ตัวอย่างเฟรมที่ถูกตีกรอบสีเหลี่ยม

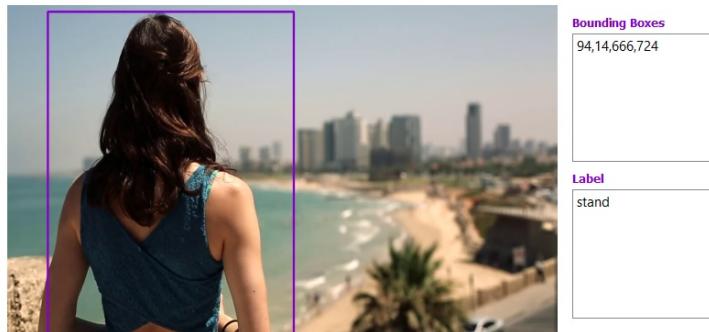
```
<?xml version="1.0"?>
- <annotation>
  <folder>D:/Goggle/Goggle_team/out/Photographer_mp4/img</folder>
  <filename>75.jpg.txt</filename>
  <path>D:/Goggle/Goggle_team/out/Photographer_mp4/img/75.jpg</path>
  - <source>
    <database>Unknown</database>
  </source>
  - <size>
    <width>1280</width>
    <height>720</height>
    <depth>3</depth>
  </size>
  <segmented>0</segmented>
  - <object>
    <name>person</name>
    <pose>Unspecified</pose>
    <truncated>0</truncated>
    <difficult>0</difficult>
    - <bndbox>
      <xmin>2</xmin>
      <ymin>35</ymin>
      <xmax>368</xmax>
      <ymax>714</ymax>
    </bndbox>
  </object>
</annotation>
```

(ข) ตัวอย่างไฟล์ xml

รูปที่ 4.8: รูปผลลัพธ์การทำงานของหน้าต่าง Track

แอปพลิเคชันจะนำคีย์เฟรมที่ถูกตีกรอบสีเหลี่ยมจากหน้าต่าง Detect มาทำนายกรอบสีเหลี่ยมในเฟรมที่เหลือระหว่างช่วงคีย์เฟรม ซึ่งผลลัพธ์ที่ได้จะได้เฟรมทุกเฟรมที่มีมนุชย์อยู่จะถูกตีกรอบสีเหลี่ยม ดังรูปที่ 4.8ก จากนั้นสามารถบันทึกข้อมูลออกเป็นไฟล์ xml ได้ดังรูปที่ 4.8ข

ผลลัพธ์การทำงานของหน้าต่าง Label



(ก) ตัวอย่างเฟรมที่ถูกตีกรอบสีเหลืองและคำทำนายการกระทำ

```
<?xml version="1.0"?>
- <annotation>
  <folder>D:/Goggle/Goggle_team/out/Photographer_mp4/Photographer_mp4/img</folder>
  <filename>75.jpg.txt</filename>
  <path>D:/Goggle/Goggle_team/out/Photographer_mp4/Photographer_mp4/img/75.jpg</path>
- <source>
  <database>Unknown</database>
</source>
- <size>
  <width>1280</width>
  <height>720</height>
  <depth>3</depth>
</size>
<segmented>0</segmented>
- <object>
  <name>carry/hold (an object)</name>
  <pose>Unspecified</pose>
  <truncated>0</truncated>
  <difficult>0</difficult>
- <bndbox>
  <xmin>2</xmin>
  <ymin>35</ymin>
  <xmax>368</xmax>
  <ymax>714</ymax>
</bndbox>
</object>
</annotation>
```

(ข) ตัวอย่างไฟล์ xml

รูปที่ 4.9: รูปผลลัพธ์การทำงานของหน้าต่าง Label

แอปพลิเคชันจะนำกรอบสีเหลืองของทุกเฟรมที่มีมนุษย์อยู่มาทำนายมนุษย์ในกรอบสีเหลืองนั้นกำลังมีการกระทำการอะไรอยู่ โดยสามารถทำงานได้ทั้งรูปแบบอัตโนมัติหรือรูปแบบแก้ไขด้วยตัวเอง และสามารถบันทึกข้อมูลออกเป็นไฟล์ xml ได้ดังรูปที่ 4.9ข

4.2 ผลการทดลองการตรวจจับวัตถุ

4.2.1 ข้อมูลรายละเอียดประกอบการทดสอบ

จำนวนเฟรมทั้งหมด: 20 เฟรม

จำนวนมนุษย์ที่อยู่ในเฟรม: 0-5 คน

ความละเอียดรูปภาพ: 1280x720 พิกเซล

ขอบเขตอัตราส่วนร่วมของกรอบที่เหลือที่จะนับว่าการทำนายถูกต้อง: 50% ขึ้นไป

4.2.2 ผลทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์สำหรับการทำการตรวจจับภาพบุคคล

ข้อมูลความแม่นยำของโมเดลปัญญาประดิษฐ์เมื่อทดสอบด้วยชุดข้อมูลของผู้วิจัย

| โมเดลปัญญาประดิษฐ์ | ความเร็วต่อรูปภาพ (มิลลิวินาที) | ความแม่นยำ (0.5 IOU) |
|--------------------------|---------------------------------|----------------------|
| SSD Mobilenet v1 ppn | 63.82 | 37.03 |
| YOLO-v3 320 | 65.00 | 64.91 |
| YOLO-v3 tiny | 17.21 | 44.44 |
| YOLO-v3 spp | 65.40 | 70.30 |
| Faster RCNN inception v2 | 981.21 | 42.59 |

ตารางที่ 4.1: ข้อมูลผลการทำงานของโมเดลปัญญาประดิษฐ์สำหรับการทำการตรวจจับภาพบุคคล

จากตารางที่ 4.2 ผู้วิจัยได้ทำการทดสอบความแม่นยำและความเร็วในการประมวลผลของโมเดลปัญญาประดิษฐ์สำหรับการทำการตรวจจับภาพบุคคล พบว่าโมเดลปัญญาประดิษฐ์ที่มีความแม่นยามากที่สุดคือ YOLO-v3 spp และโมเดลปัญญาประดิษฐ์ที่มีความเร็วในการทำนายต่อรูปภาพเร็วที่สุดคือ YOLO-v3 tiny จากผลลัพธ์การทดลองดังกล่าว ทุกโมเดลปัญญาประดิษฐ์ยกเว้น Faster RCNN มีความเร็วในการประมวลผลต่อรูปภาพที่ผู้วิจัยสามารถรับได้ (ไม่เกิน 1 วินาที) ดังนั้นผู้วิจัยจึงเลือกโมเดลปัญญาประดิษฐ์ที่จะใช้จากการความแม่นยามากที่สุด คือ YOLO-v3 spp

4.3 ผลการทดสอบระบบติดตามตำแหน่งของมนุษย์

4.3.1 ข้อมูลรายละเอียดประกอบการทดสอบ

ชื่อวิดีโอ: Photographer beach photography

ความยาววิดีโอ: 15 วินาที

จำนวนเฟรมทั้งหมด: 374 เฟรม

อัตราเฟรมต่อวินาที: 24.9 เฟรมต่อวินาที

ความละเอียดของวิดีโอ: 1920x1080 พิกเซล

ความละเอียดของวิดีโอที่ใช้ในการประมวลผลจริง: 1280x720 พิกเซล

ขอบเขตอัตราส่วนร่วมของกรอบที่เหลือที่จะนับว่าการทำนายถูกต้อง: 80% ขึ้นไป

4.3.2 ผลทดสอบประสิทธิภาพ และความเร็วในการประมวลผล

| วิธีการทดสอบ | ความแม่นยำ (%) | ความเร็วในการประมวลผล (วินาที) | | |
|--|----------------|--------------------------------|-----|------|
| ใช้โมเดลปัญญาประดิษฐ์ YOLO-v3 spp ประมวลผลทุกเฟรมในวิดีโอ | 95 | - | 452 | - |
| ใช้โมเดลปัญญาประดิษฐ์ YOLO-v3 spp ประมวลผลทุกๆ N เฟรมในวิดีโอ แล้วใช้ระบบติดตามการเคลื่อนไหวตำแหน่งต่อไปของวัตถุในเฟรมระหว่างนั้น | | | | |
| N = 10 | 85 | -10 | 69 | -383 |
| N = 20 | 80 | -15 | 41 | -411 |
| N = 25 | 75 | -20 | 35 | -417 |

ตารางที่ 4.2: ผลการทดสอบประสิทธิภาพของการตรวจจับกรอบสีเหลี่ยมภายในวิดีโอ

จากตารางที่ 4.2 ผู้วิจัยได้ทำการทดสอบความแม่นยำและความเร็วในการประมวลผลของการใช้โมเดลปัญญาประดิษฐ์ YOLO-v3 spp ประมวลผลทุกเฟรม แม้จะตั้งขอบเขตอัตราส่วนร่วมของกรอบที่เหลือที่จะนับว่าการทำนายถูกต้องสูงถึง 80% แต่ความแม่นยำสูงถึง 95% ใช้เวลาในการประมวลผล 452 วินาที เฉลี่ยเฟรมละ 1.2 วินาที ซึ่งถือเป็นความแม่นยำที่สูงมากเมื่อเทียบกับเวลาที่ใช้ในการประมวลผล

ต่อมาเป็นการทดสอบโดยใช้โมเดลปัญญาประดิษฐ์ประมวลผลเฉพาะบางเฟรมทุกๆช่วงหนึ่ง แล้วใช้ระบบติดตามการเคลื่อนไหวตำแหน่งต่อไปของวัตถุในการสร้างกรอบสีเหลี่ยมในเฟรมระหว่างนั้น เพื่อเพิ่มความเร็วในการประมวลผล โดยระยะที่ใช้ในการทดสอบคือ ทุกๆ 10 เฟรม 20 เฟรม และ 25 เฟรม ซึ่งจากการทดสอบนั้นพบว่าวิธีการนี้มีความแม่นยำลดลงมากกว่า 85% น้อยกว่า 80% น้อยกว่า 75% น้อยกว่า 70% น้อยกว่า 65% น้อยกว่า 60% น้อยกว่า 55% น้อยกว่า 50% น้อยกว่า 45% น้อยกว่า 40% น้อยกว่า 35% น้อยกว่า 30% น้อยกว่า 25% น้อยกว่า 20% น้อยกว่า 15% และเมื่อใช้ระยะประมวลผล 25 เฟรมจะเร็วกว่าประมวลผล 13 เท่า และความแม่นยำต่ำลงถึง 20%

4.4 ผลการทดสอบระบบระบุตัวตนของมนุษย์

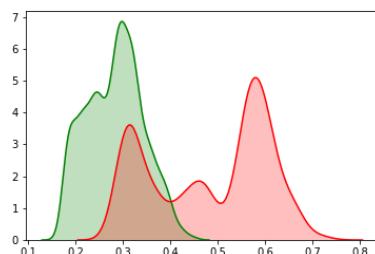
4.4.1 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์สำหรับการระบุตัวตนของบุคคล

ความแม่นยำของโมเดลปัญญาประดิษฐ์จากแหล่งที่มีมาค่าดังตารางด้านล่างดังนี้

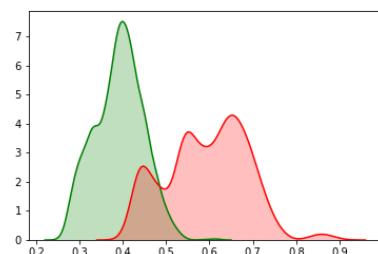
| โมเดลปัญญาประดิษฐ์ | rank1/mAP โดยใช้วิธีการทดสอบด้วย Global+DMLI |
|----------------------|--|
| ResNet50 Market1501 | 91.0/77.6 |
| ResNet50 DukeMTMCRID | 80.7/68.0 |
| ResNet50 CUHK03 | 60.9/59.7 |
| ResNet50 MSMT17 | 66.3/40.6 |

ตารางที่ 4.3: ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์

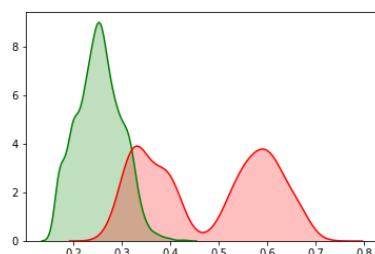
ต่อมานำโมเดลปัญญาประดิษฐ์แต่ละอันมาทดสอบกับตัวอย่างภาพชุดข้อมูลที่ทางคณะผู้จัดได้สร้างขึ้น โดยภาพชุดข้อมูลที่นำมาใช้จะผ่านการตรวจหาบุคคลภายในภาพด้วยโมเดลปัญญาประดิษฐ์ YOLO-v3 spp นำภาพชุดข้อมูลที่ผ่านการตรวจหาบุคคลภายในภาพเข้าระบบการระบุตัวตนของบุคคล โดยจะให้ผลลัพธ์ออกมาเป็นค่า AUC และกราฟที่แสดงการเปรียบเทียบระหว่างการระบุว่าเป็นบุคคลเดียวกันไม่เป็นบุคคลเดียวกันตามรูปที่ 4.10



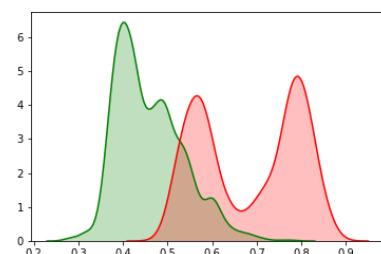
(ก) กราฟของโมเดลปัญญาประดิษฐ์ที่ผ่านการสร้างด้วยชุดข้อมูล Market1501



(ข) กราฟของโมเดลปัญญาประดิษฐ์ที่ผ่านการสร้างด้วยชุดข้อมูล DukeMTMCRID



(ค) กราฟของโมเดลปัญญาประดิษฐ์ที่ผ่านการสร้างด้วยชุดข้อมูล CUHK03



(ง) กราฟของโมเดลปัญญาประดิษฐ์ที่ผ่านการสร้างด้วยชุดข้อมูล MSMT17

รูปที่ 4.10: กราฟแสดงการเปรียบระหว่างการระบุว่าเป็นบุคคลเดียวกันกับไม่เป็นบุคคลเดียวกัน โดยพื้นที่ใต้กราฟที่เป็นสีเขียวจะหมายถึงการระบุว่าเป็นบุคคลเดียวกัน ในขณะที่พื้นที่ใต้กราฟสีแดงหมายถึงการระบุว่าไม่เป็นบุคคลเดียวกัน และแกน x คือค่า aligned distance สำหรับแต่ละคน y จำนวนครั้งของภาพ

| โมเดลปัญญาประดิษฐ์ | AUC |
|-----------------------|------|
| ResNet50 Market1501 | 0.91 |
| ResNet50 DukeMTMCRelD | 0.96 |
| ResNet50 CUHK03 | 0.97 |
| ResNet50 MSMT17 | 0.94 |

ตารางที่ 4.4: ผลการทดสอบค่า AUC ของโมเดลปัญญาประดิษฐ์

จากการทดลองสมมติฐานที่ตั้งไว้วันไม่เป็นจริง เพราะโมเดลปัญญาประดิษฐ์ที่ผ่านการสร้างด้วยชุดข้อมูล Market1501 นั้นไม่ได้มีค่า AUC สูงที่สุดเมื่อนำมาใช้กับชุดข้อมูลที่ทางผู้วิจัยสร้างขึ้น เมื่อเทียบ โมเดลปัญญาประดิษฐ์ที่ผ่านการสร้างด้วยชุดข้อมูล CUHK03 นั้นที่ให้ผลลัพธ์ค่า AUC สูงที่สุดเมื่อนำมาใช้กับชุดข้อมูลที่ทางผู้วิจัยสร้างขึ้น ทางผู้วิจัยจึงได้เลือกโมเดลปัญญาประดิษฐ์ที่ผ่านการสร้างด้วยชุดข้อมูล CUHK03 นี้มาใช้ในงานวิจัย

4.5 ผลการทดสอบการจำแนกการกระทำของมนุษย์

4.5.1 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Resnet50 ที่ถูกสร้างด้วยชุดข้อมูลของ AVA โดยใช้ชุดข้อมูลของ AVA ในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิง ได้ผลการทดลองดังตารางต่อไปนี้

| | ความเร็วต่อรูปภาพ(มิลลิวินาที) | ความแม่นยำ (PASCAL mAP) |
|-----------------------|--------------------------------|-------------------------|
| แหล่งอ้างอิง | 93.00 | 0.110 |
| ผลการทดสอบของผู้วิจัย | 85.35 | 0.068 |

ตารางที่ 4.5: ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์เทียบผลลัพธ์กับแหล่งอ้างอิง

ความเร็วของต่อรูปภาพทางผู้วิจัยได้ใช้กราฟิกการ์ด GTX 2080 Ti ในการทดสอบซึ่งจะให้ความเร็วอยู่ที่ 85.35 มิลลิวินาที/ภาพ ซึ่งทางแหล่งอ้างอิงนั้นใช้กราฟิกการ์ด Nvidia GeForce GTX TITAN X ในส่วนของค่าความแม่นยำที่ไม่เท่ากัน คาดว่าจะเป็นเพราะการประมวลผลของกราฟิกการ์ดของรุ่นที่ต่างกันและสเปคของเครื่องคอมพิวเตอร์จึงทำให้ค่า mAP ที่ออกมากไม่เท่ากัน

4.5.2 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Resnet50 ที่เคยถูกสร้างด้วยชุดข้อมูลของ AVA และใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบและเทียบผลลัพธ์กับแหล่งอ้างอิง

| | ความเร็วต่อรูปภาพ(มิลลิวินาที) | ความแม่นยำ (PASCAL mAP) |
|-----------------------|--------------------------------|-------------------------|
| แหล่งอ้างอิง | 93.00 | 0.110 |
| ผลการทดสอบของผู้วิจัย | 63.11 | 0.152 |

ตารางที่ 4.6: ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ เมื่อใช้กับชุดข้อมูลที่ผู้วิจัยสร้างขึ้น

ความเร็วของต่อรูปภาพทางผู้วิจัยได้ใช้กราฟิกการ์ด Tesla V100-SXM2 ในการทดสอบซึ่งจะให้ความเร็วอยู่ที่ 63.11 มิลลิวินาที/ภาพ ซึ่งทางแหล่งอ้างอิงนั้นใช้กราฟิกการ์ด Nvidia GeForce GTX TITAN X ซึ่งการนำโมเดลปัญญาประดิษฐ์ AVA มาใช้ในการทดสอบกับชุดข้อมูลทดสอบที่ทางผู้วิจัยสร้างขึ้น ซึ่งผลลัพธ์ที่ได้ออกมาดีกว่าชุดข้อมูลที่ทางผู้วิจัยได้สร้างขึ้นจะมีการตัดหมวดหมู่ที่ไม่มีในชุดข้อมูลของ AVA ออกไป ดังนั้นจึงทำให้ผู้วิจัยสรุปได้ว่าการนำโมเดลปัญญาประดิษฐ์ของ AVA มาใช้กับชุดข้อมูลที่มีหมวดหมู่ไม่ครบก็สามารถทำงานได้มีประสิทธิภาพสูงกว่าแหล่งอ้างอิง

4.5.3 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ Resnet50 ที่ถูกสร้างด้วยชุดข้อมูลที่ผู้วิจัยสร้างขึ้น และใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบและเทียบผลลัพธ์การทดสอบก่อนหน้า

ในส่วนนี้จะเป็นการทดสอบโดยใช้โครงสร้างโมเดลปัญญาประดิษฐ์เป็น ResNet50 โดยจะมีการใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นซึ่งเป็นชุดข้อมูลทดสอบเดียวกับที่ทางผู้วิจัยใช้ในการทดสอบโมเดลปัญญาประดิษฐ์ AVA มาใช้ในการทดสอบครั้งนี้ด้วย โดยชุดข้อมูลที่นำมาใช้สำหรับการสร้างโมเดลปัญญาประดิษฐ์จะมีอยู่ 2 ชุดข้อมูล ซึ่งได้แก่ goggle dataset v1 และ goggle dataset v2 ชุดข้อมูลทั้งสองอันนี้จะแตกต่างกันตรงที่ goggle dataset v2 นั้นเป็นชุดข้อมูลเกิดจากการที่ทางผู้วิจัยได้เข้าไปลบในส่วนที่มีการกำกับข้อมูลภาพผิดและมีการสูญเสียของข้อมูลอ กามาเพื่อลดโอกาสที่จะเกิด overfitting ของข้อมูล โดยจำนวนชุดข้อมูลของ goggle dataset v1 ที่ใช้สำหรับการสร้างโมเดลจะมี 213,061 ภาพ และในส่วนของ goggle dataset v2 จะมีจำนวนชุดข้อมูล 120,177 ภาพ และมีการตั้งค่าตัวแปรต่าง ๆ ที่ใช้สำหรับการสร้างโมเดลปัญญาประดิษฐ์ดังนี้

1. ขนาดของรูปภาพ : 224x224
2. Pooling : Average pooling
3. Activation function : Softmax
4. Epoch : 50
5. Batch size : 16
6. Optimize : Momentum
 - (a) Learning rate : 0.001
 - (b) Momentum : 0.9

4.5.3.1 การทดสอบสร้างโมเดลปัญญาประดิษฐ์ ResNet50 ด้วยชุดข้อมูล goggle dataset v1 และ goggle dataset v2

| ชุดข้อมูล | ความเร็วต่อรูปภาพ(มิลลิวินาที) | ความแม่นยำ (PASCAL mAP) |
|----------------------------|--------------------------------|-------------------------|
| goggle dataset v1 ResNet50 | 2.78 | 0.279 |
| goggle dataset v2 ResNet50 | 2.52 | 0.294 |

ตารางที่ 4.7: ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้น ใช้กับชุดข้อมูลที่ผู้วิจัยสร้างขึ้น

จากตาราง 4.7 การทดลองจะเป็นได้ว่าความเร็วของต่อรูปภาพทางผู้วิจัยได้ใช้กราฟิกการ์ด Tesla V100-SXM2 แต่ความเร็วนั้นเร็วกว่าตอนที่ทดสอบด้วยโมเดลปัญญาประดิษฐ์ AVA เป็นอย่างมาก เนื่องจากโมเดลปัญญาประดิษฐ์ของ AVA นั้นจะมีการทำในส่วนของการหกรอบสี่เหลี่ยมรอบตัวมนุษย์ด้วย ในขณะที่โมเดลปัญญาประดิษฐ์ของผู้วิจัยนั้นจะไม่มีหกรอบสี่เหลี่ยมรอบตัวมนุษย์ เพราจะมีการนำโมเดลปัญญาประดิษฐ์ของ YOLO-v3 spp มาหกรอบสี่เหลี่ยมรอบตัวมนุษย์ตั้งแต่แรกแล้ว ในส่วนของค่า mAP ที่ได้ออกมาใน goggle dataset v2 มีค่า mAP มากกว่า แต่มีความแตกต่างกันไม่มาก อาจจะเป็นไปได้ว่า เพราะจำนวนข้อมูลที่น้อยกว่า goggle dataset v1 ดังนั้นทางผู้วิจัยจึงได้เลือก goggle dataset v2 มาใช้กับการทดลองครั้งต่อไป

4.5.3.2 การทดสอบสร้างโมเดลปัญญาประดิษฐ์ ResNet50 ด้วยชุดข้อมูล Google dataset v2 โดยใช้ weight ของ ImageNet

| | ความเร็วต่อรูปภาพ(มิลลิวินาที) | ความแม่นยำ (PASCAL mAP) |
|-------------------|--------------------------------|-------------------------|
| ResNet50-ImageNet | 2.51 | 0.454 |

ตารางที่ 4.8: ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้จัดสร้างขึ้นโดยใช้ weight จาก ImageNet ใช้กับชุดข้อมูลที่ผู้จัดสร้างขึ้น

จากตาราง 4.8 เมื่อทางผู้จัดได้นำ weight ของ ImageNet มาช่วยในการสร้างโมเดลทำให้ประสิทธิภาพของโมเดลที่ได้ออกมานั้นสูงขึ้นมากเมื่อเทียบกับการทดสอบโมเดลปัญญาประดิษฐ์ของผู้จัดก่อนหน้านี้

4.5.3.3 การทดสอบสร้างโมเดลปัญญาประดิษฐ์ ResNet50-ImageNet ด้วยชุดข้อมูล Google dataset v2 ด้วยวิธีการทำ scaling

| | ความเร็วต่อรูปภาพ(มิลลิวินาที) | ความแม่นยำ (PASCAL mAP) |
|---|--------------------------------|-------------------------|
| ResNet50-ImageNet Normalization | 2.51 | 0.449 |
| ResNet50-ImageNet Centering Featurewise | 2.40 | 0.457 |
| ResNet50-ImageNet Centering Samplewise | 2.49 | 0.466 |
| ResNet50-ImageNet Standardize Featurewise | 2.48 | 0.407 |
| ResNet50-ImageNet Standardize Samplewise | 2.49 | 0.432 |

ตารางที่ 4.9: ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้จัดสร้างขึ้นโดยใช้ weight จาก ImageNet และการทำ scaling ใช้กับชุดข้อมูลที่ผู้จัดสร้างขึ้น

จากตาราง 4.9 จะเป็นการทดลอง scaling ด้วยรูปแบบต่าง อย่างการทำ normalization จะเป็นทำให้ค่าในพิกเซลอยู่ในช่วง 0 ถึง 1 การทำ centering คือการลบค่าในพิกเซลด้วยค่าเฉลี่ยของพิกเซล โดยจะแบ่งออกเป็น 2 แบบได้แก่ featurewise และ samplewise โดย featurewise จะเป็นการทำค่าเฉลี่ยพิกเซลจากทุกรูปในชุดข้อมูลแล้วนำมารอบออกในแต่ละพิกเซลของรูป ส่วนของ samplewise จะไม่มีการไปยุ่งเกี่ยวกับรูปอื่น คือจะหาค่าเฉลี่ยของพิกเซลของรูปนั้น ๆ และนำค่าพิกเซลในรูปนั้น ๆ ลบออกด้วยค่าเฉลี่ย ต่อมาการทำ standardize คือการทำหารค่าในพิกเซลด้วยค่า standard deviation ของพิกเซลในรูป ซึ่งจะแบ่งออกเป็น 2 แบบ ได้แก่ featurewise และ samplewise เหมือนกับ centering โดย featurewise จะหาค่า standard deviation ของทุกพิกเซลในชุดข้อมูลแล้วนำมาราห์โนดแต่ละพิกเซลของรูป ส่วนของ samplewise จะเป็นการทำค่า standard deviation ของรูปนั้น ๆ มาหารกับทุกพิกเซลในรูปนั้น ๆ จากการทำลองจะทำให้เห็นว่าโมเดลปัญญาประดิษฐ์ที่มีประสิทธิภาพสูงที่สุดคือโมเดลปัญญาประดิษฐ์ ResNet50-ImageNet Centering Samplewise

4.5.3.4 การทดสอบ average precision (AP) ของโมเดลปัญญาประดิษฐ์ ResNet50-ImageNet Centering Samplewise

| Label | ResNet50-ImageNet Centering Samplewise |
|------------|--|
| Play phone | 0.824 |
| Eat | 0.975 |
| Sit | 0.215 |
| Sleep | 0.473 |
| Stand | 0.043 |
| Walk | 0.266 |

ตารางที่ 4.10: โมเดลปัญญาประดิษฐ์ ResNet50-ImageNet Centering Samplewise วัดค่า average precision (AP) ของแต่ละหมวดหมู่

จากตาราง 4.10 จะเห็นได้ว่าหมวดหมู่ที่ได้ค่า average precision น้อยได้แก่หมวดหมู่ sit, stand และ walk จากหมวดหมู่ sit นั้นที่ได้ค่า average precision น้อยนั้นเป็นเพราะว่าการกระทำนั้นใกล้เคียงกับท่า eat และในส่วนของหมวดหมู่ stand และ walk จะเป็น เพราะว่ามีการทำนายในสองหมวดหมู่นี้สลับกัน ซึ่ง เพราะทั้งสองหมวดหมูนมีการกระทำที่คล้ายคลึงกัน

4.5.4 ทดสอบประสิทธิภาพการทำงานของโมเดลปัญญาประดิษฐ์ I3D สร้างด้วยชุดข้อมูลที่ผู้วิจัยสร้างขึ้นโดยใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นในการทดสอบ

คุณลักษณะที่ใช้ในการสร้างโมเดลปัญญาประดิษฐ์ I3D ที่ผู้วิจัยได้พัฒนาเป็นชุดของเฟรมที่เป็นภาพสีปกติ (RGB) และชุดของเฟรมที่เป็น optical flow (OF) โดยใช้ PASCAL mAP, Top@1 และ Top@3 ในการวัดผลความแม่นยำของแต่ละโมเดล ซึ่งมีรายละเอียดและพารามิเตอร์ดังนี้

1. โมเดลที่ 1

- (a) Learning rate: 0.001
- (b) Dropout: 0.36
- (c) Optimizer: Momentum
- (d) Optimizer parameter:
 - i. Momentum: 0.8

2. โมเดลที่ 2

- (a) Learning rate: 0.001
- (b) Dropout: 0.36
- (c) Optimizer: Adam
- (d) Optimizer parameters:
 - i. β_1 : 0.9
 - ii. β_2 : 0.999
 - iii. ϵ : 10^{-8}

4.5.4.1 การทดสอบประสิทธิภาพของโมเดลปัญญาประดิษฐ์ I3D ที่ใช้ชุดข้อมูลที่เป็นแบบภาพสีปกติ

| โมเดล | PASCAL mAP | Top@1 | Top@3 |
|----------------|------------|-------|-------|
| RGB โมเดลที่ 1 | 0.564 | 0.482 | 0.641 |
| RGB โมเดลที่ 2 | 0.356 | 0.265 | 0.487 |

ตารางที่ 4.11: ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้นโดยใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นแบบภาพสีปกติ

4.5.4.2 การทดสอบประสิทธิภาพของโมเดลปัญญาประดิษฐ์ I3D ที่ใช้ชุดข้อมูลที่เป็นแบบ optical flow

| โมเดล | PASCAL mAP | Top@1 | Top@3 |
|-------------------------|------------|-------|-------|
| Optical flow โมเดลที่ 1 | 0.748 | 0.737 | 0.908 |
| Optical flow โมเดลที่ 2 | 0.777 | 0.759 | 0.959 |

ตารางที่ 4.12: ผลการทดสอบความแม่นยำของโมเดลปัญญาประดิษฐ์ที่ผู้วิจัยสร้างขึ้นโดยใช้ชุดข้อมูลที่ผู้วิจัยสร้างขึ้นแบบ optical flow

4.5.4.3 ตารางแสดงการเปรียบเทียบค่า average precision (AP) ของทุกการกระทำของแต่ละโมเดล

| Label | RGB โมเดลที่ 1 | RGB โมเดลที่ 2 | OF โมเดลที่ 1 | OF โมเดลที่ 1 |
|------------|----------------|----------------|---------------|---------------|
| Play phone | 0.239 | 0.011 | 0.552 | 0.599 |
| Eat | 0.282 | 0.058 | 0.787 | 0.839 |
| Sit | 0.450 | 0.113 | 0.795 | 0.799 |
| Sleep | 0.800 | 0.655 | 0.704 | 0.628 |
| Stand | 0.865 | 0.822 | 0.731 | 0.797 |
| Walk | 0.748 | 0.476 | 0.921 | 1.000 |

ตารางที่ 4.13: เปรียบเทียบค่า AP ของทุกการกระทำของแต่ละโมเดล

จะเห็นได้ว่าโมเดลที่ถูกสร้างจากชุดข้อมูลแบบ optical flow นั้นมีความแม่นยำสูงกว่าโมเดลที่ใช้ข้อมูลภาพสีแบบปกติในการสร้าง แต่ถ้าหากพิจารณาจากค่า average precision (AP) ตามตารางที่ 4.13 โมเดลที่ผ่านการสร้างด้วยชุดข้อมูลแบบภาพสีปกตินั้นจะมีประสิทธิภาพสูงเมื่อเป็นการกระทำที่แทบจะไม่มีการเคลื่อนไหวคือนอนและยืน ในขณะที่โมเดลที่ผ่านการสร้างด้วยชุดข้อมูลแบบ optical flow นั้นมีความแม่นยำสูงกว่ามากในการกระทำที่มีการเคลื่อนไหว จึงสามารถกล่าวได้ว่าโมเดลแบบ optical flow นั้นเหมาะสมสำหรับใช้ในการจำแนกการกระทำที่มีการเคลื่อนไหว และโมเดลแบบภาพสีปกติเหมาะสมสำหรับใช้ในการจำแนกการกระทำที่แทบจะไม่มีการเคลื่อนไหว

4.5.4.4 การทดสอบประสิทธิภาพของโมเดลปัญญาประดิษฐ์ I3D ที่ใช้ชุดข้อมูลทั้งสองแบบ (RGB + optical flow)

จากการวิเคราะห์ในหัวข้อที่ 4.5.4.3 ทำให้ผู้วิจัยสนใจนำโมเดลทั้งสองแบบมาใช้ร่วมกันในการจำแนกการกระทำ โดยจะใช้การรวมผลลัพธ์ความน่าจะเป็นที่ถูกคูณด้วยอัตราส่วนความน่าเชื่อถือ หมายความว่าในการกระทำ l โมเดล M_1 นำรายผลลัพธ์ความน่าจะเป็นที่ได้จากการกระทำดังกล่าว $P_l^{M_1}$ และโมเดล M_2 นำรายผลลัพธ์ความน่าจะเป็นที่ได้จากการกระทำดังกล่าว $P_l^{M_2}$ สมมติว่าให้อัตราส่วนความน่าเชื่อถือของโมเดล M_1 เป็น W_1 และของโมเดล M_2 เป็น W_2 (โดยที่ $W_1 + W_2 = 1$) ทำให้

สามารถเขียนสมการความเป็นไปได้ว่า ชุดข้อมูลนี้จะเป็นการกระทำ l ได้ดังนี้

$$P_l = \frac{W_1 P_l^{M_1} + W_2 P_l^{M_2}}{2} \quad (4.1)$$

ซึ่งผลลัพธ์จากการทดลองเป็นดังตารางที่ 4.14 และผลลัพธ์ค่า average precision เป็นดังตารางที่ 4.13 จาก

| โมเดล (อัตราส่วนความน่าเชื่อถือ) | PASCAL mAP | Top@1 | Top@3 |
|--|------------|-------|-------|
| RGB โมเดลที่ 1 + OF โมเดลที่ 1 (50:50) | 0.765 | 0.740 | 0.903 |
| RGB โมเดลที่ 1 + OF โมเดลที่ 2 (50:50) | 0.823 | 0.806 | 0.945 |
| RGB โมเดลที่ 2 + OF โมเดลที่ 1 (50:50) | 0.706 | 0.679 | 0.865 |
| RGB โมเดลที่ 2 + OF โมเดลที่ 2 (50:50) | 0.804 | 0.780 | 0.931 |

ตารางที่ 4.14: ผลการทดสอบความแม่นยำเมื่อนำโมเดลแบบภาพสีปกติ และ optical flow มาใช้ร่วมกัน

ผลลัพธ์ในตารางที่ 4.14

บทที่ 5

การอภิปรายผล สรุปผลงานวิจัย และข้อเสนอแนะ

5.1 การอภิปรายผล

5.1.1 เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ สามารถทำงานได้ครอบคลุมความต้องการของระบบทั้งหมด คือ สามารถตัดวิดีโอช่วงที่ไม่มีมนุษย์อยู่ออกได้, สามารถระบุตำแหน่งของมนุษย์ได้, สามารถสร้างชุดข้อมูลสำหรับการนำไปพัฒนาโมเดลปัญญาประดิษฐ์ได้, สามารถนำโมเดลปัญญาประดิษฐ์มาทำงานร่วมกับมนุษย์ได้, สามารถวิเคราะห์การกระทำการของมนุษย์ต่อลูกน้ำได้ แต่สำหรับการนำไปใช้จริงในงานเกี่ยวกับการวิเคราะห์ผลวิดีโอด้วยมีข้อจำกัดในการใช้งาน ดังนี้

- เนื่องจากในความต้องการของระบบ ต้องการเครื่องมือสำหรับการกำกับข้อมูลที่สนใจมนุษย์เป็นศูนย์กลาง ดังนั้นเครื่องมือกำกับข้อมูลจึงมีข้อจำกัดที่สามารถกำกับได้เพียงแต่มนุษย์
- ใน 1 รอบการทำงาน จะสามารถนำวิดีโอเข้ามาสร้างเป็นชุดข้อมูลได้เพียงแค่วิดีโอดียว เนื่องจาก การออกแบบหน้าต่างการทำงานและโครงสร้างของโปรแกรมถูกเขียนมาเพื่ออำนวยต่อการสร้างชุดข้อมูลชุดเดียวต่อการทำงานหนึ่งครั้ง

5.1.2 ชุดข้อมูล

ชุดข้อมูลจะเป็นชุดข้อมูลที่ถูกสร้างมากจากเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์ที่ผู้วิจัยได้สร้างขึ้นมา โดยจุดประสงค์ของชุดข้อมูลเพื่อนำไปพัฒนาโมเดลปัญญาประดิษฐ์เพื่อวิเคราะห์การกระทำการของมนุษย์ ซึ่งชุดข้อมูลที่ผู้วิจัยได้สร้างขึ้นมา มีจำนวน 3 ชุดข้อมูล คือ

- ชุดข้อมูล Google datasetV1 คือ ชุดข้อมูลที่เก็บรวบรวมการกระทำการของมนุษย์ทั้ง 6 หมวดหมู่ คือ นั่ง เดิน กิน นอน เล่นโทรศัพท์ ยืน โดยชุดข้อมูลใช้สำหรับการนำไปสอนให้กับโมเดลปัญญาประดิษฐ์
- ชุดข้อมูล Google datasetV2 คือการนำชุดข้อมูล Google datasetV1 มาเก็บข้อมูลเพิ่มเติม และ คัดกรองข้อมูลที่ไม่จำเป็นออกเพื่อใช้สำหรับการนำไปสอนและพัฒนาให้กับโมเดลปัญญาประดิษฐ์
- ชุดข้อมูล All action คือ ชุดข้อมูลจากวิดีโอที่ภายในวิดีโอมีประกอบด้วยการกระทำหลากหลายกรรรม ทำอยู่ในวิดีโอ โดยชุดข้อมูลนี้จะใช้สำหรับทดสอบโมเดลปัญญาประดิษฐ์

5.1.3 โมเดลปัญญาประดิษฐ์

5.1.3.1 การสร้างโมเดลปัญญาประดิษฐ์ด้วยโครงสร้าง ResNet50

ในการสร้างโมเดลปัญญาประดิษฐ์ด้วยโครงสร้าง ResNet50 คุณภาพของโมเดลปัญญาประดิษฐ์ที่ได้ออกมานั้นจะมีของบางหมวดหมู่ที่ได้ค่าของการทำนายออกมาน้อย ซึ่งเป็นไปได้ว่าชุดข้อมูลที่เก็บมาใช้ในการสร้างโมเดลปัญญาประดิษฐ์มีความหลากหลายที่น้อย เพราะเนื่องจากการใช้ภาพจากวิดีโอดียว กัน และในด้านของจำนวนชุดข้อมูลที่นำมาใช้ในการสร้างโมเดลปัญญาประดิษฐ์มีจำนวนน้อยลง เพราะต้องมีการลบข้อมูลภาพใน

ส่วนที่ผิดพลาดออก เช่น การกำกับคำพิจารณา หรือกรอบสีเหลี่ยมที่ไม่เจอคน เป็นต้น การที่ในบางหมวดหมู่ที่ทำทางการกระทำที่คล้ายคลึงกัน หรือในส่วนของความสมดุลของหมวดหมู่ที่นำมาใช้ในการสร้างโมเดลปัญญาประดิษฐ์ที่มีจำนวนของข้อมูลในแต่ละหมวดหมู่ไม่เท่ากัน

5.2 สรุปผลงานวิจัย

5.2.1 เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

1. เครื่องมือสามารถปรับความละเอียดในการตรวจสอบมนุษย์ในเฟรมได้ (ความถี่ในการสูบตัวอย่างเฟรมมาตรวจสอบ)
2. เครื่องมือสามารถคัดกรองคีย์เฟรมที่ไม่มีมนุษย์อยู่ออกได้
3. เครื่องมือสามารถตั้งค่า threshold, min-max ของกรอบสีเหลี่ยม เพื่อให้ได้ผลลัพธ์ที่ตรงตามความต้องการของผู้ใช้
4. เครื่องมือสามารถตรวจหาตำแหน่งของมนุษย์บนคีย์เฟรม และ สร้างกรอบสีเหลี่ยมครอบมนุษย์ได้
5. เครื่องมือสามารถปรับแก้ไขตำแหน่งและขนาดของกรอบสีเหลี่ยม
6. เครื่องมือสามารถสร้างชุดข้อมูลสำหรับการนำไปพัฒนาโมเดลปัญญาประดิษฐ์ได้
7. เครื่องมือสามารถนำโมเดลปัญญาประดิษฐ์มาทำงานร่วมกับมนุษย์ได้
8. เครื่องมือสามารถบุหมวดหมู่ของการกระทำการกระทำของกรอบสีเหลี่ยมได้
9. เครื่องมือสามารถสร้างผลลัพธ์ออกเป็นไฟล์ xml , csv หรือ รายงานได้

5.2.2 โมเดลปัญญาประดิษฐ์

1. สามารถนำชุดข้อมูลที่ได้จากเครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์มาสร้างโมเดลปัญญาประดิษฐ์ได้
2. โมเดลปัญญาที่สร้างออกมานั้นสามารถนำมาทำนายการกระทำการกระทำของมนุษย์ได้

5.3 ข้อเสนอแนะ

5.3.1 เครื่องมือกำกับคุณลักษณะด้วยปัญญาประดิษฐ์

แนวคิดในการทำงานเพิ่มเติมเพื่อทำให้เครื่องมือสามารถนำไปใช้ได้หลากหลายมากขึ้นมีดังนี้

1. ในขณะนี้เครื่องมือใช้โมเดลปัญญาประดิษฐ์ YOLO เป็นตัวตรวจสอบวัตถุซึ่งมีการเขียนเงื่อนไขให้เลือกตรวจสอบโดยยึดคนเป็นหลัก ดังนั้นหากมีการปรับเปลี่ยนเงื่อนไขของเครื่องมือจะทำให้สามารถเลือกตรวจสอบวัตถุอื่นๆได้หลากหลายมากขึ้น
2. ในขณะนี้เครื่องมือสามารถมีรูปแบบไฟล์ที่สามารถนำเข้ามาวิเคราะห์ผลได้เพียงแค่ไฟล์วิดีโอ ซึ่งหากเครื่องมือสามารถรับรูปแบบไฟล์อื่นๆได้เพิ่มขึ้น เช่น ไฟล์รูปภาพ , ไฟล์เดอร์ของรูปภาพ จะเพิ่มความยืดหยุ่นของเครื่องมือมากขึ้น

5.3.2 ໂມເດລປັນຍາປະຕິບັດ

ແນວຄົດໃນການເພີ່ມປະສົງທີ່ຂອງໂມເດລປັນຍາປະຕິບັດໃຫ້ສູງເຖິງມີດັ່ງນີ້

1. ເພີ່ມຈຳນວນຊຸດຂໍ້ອມູລ
2. ເພີ່ມຄວາມຫລາກຫລາຍໃຫ້ກັບຊຸດຂໍ້ອມູລ
3. ກາຣທຳຊຸດຂໍ້ອມູລໃໝ່ມີຄວາມສມດຸລກັນໃນແຕ່ລະໜວດໜຸ້ງ
4. ກາຣປັບປຸງວິທີກາຮັດເຕັມຂໍ້ອມູລ ຮູ່ກາຣປັບປຸງກ່າວມີເຕອວ່າ

ເອກສາຮອ້າງອີງ

- [1] Natthawat Phongchit, 2018, **Convolutional Neural Network (CNN)** [Online], Available : <https://bit.ly/2QeCuPB>, [2019,September,23].
- [2] embARC Machine Learning Inference Library, **Average-Pooling** [Online], Available : <https://bit.ly/2mvmb6B>, [2019,September,23].
- [3] Nwankpa, Chigozie and Ijomah, Winifred and Gachagan, Anthony and Marshall, Stephen, 2018, **Activation functions: Comparison of trends in practice and research for deep learning**, arXiv preprint arXiv:1811.03378, [2018,November,8].
- [4] Adrian Rosebrock, 2016, **Intersection over Union for object detection** [Online], Available : <https://www.pyimagesearch.com/2016/11/07/intersection-over-union-iou-for-object-detection/>, [2019,September,18].
- [5] Sarang Narkhede, 2018, **Understanding AUC - ROC Curve** Available : <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>, [2019,September,25].
- [6] Rasmus Rothe, Matthieu Guillaumin, Luc Van Gool, **Non-Maximum Suppression for Object Detection** [Online], Available : http://www.vision.ee.ethz.ch/publications/papers/proceedings/eth_biwi_01126.pdf, [2019,September,18].
- [7] Tommy Huang, **Machine / Deep Learning: Object Detection Non-Maximum Suppression (NMS)** [Online] , Available : <http://wow.in.th/Y2x9> [2019,September,18].
- [8] OpenCV, **Optical Flow** [Online], Available : https://docs.opencv.org/3.4/d4/dee/tutorial_optical_flow.html, [2019,August,17].
- [9] Farnebäck, Gunnar, 2003, **Two-frame motion estimation based on polynomial expansion**, Scandinavian conference on Image analysis, [2003], pages 363-370.
- [10] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi , 2016 ,**You Only Look Once: Unified, Real-Time Object Detection** , vol.5.
- [11] Sik-Ho Tsang, 2018, **Review: SSD — Single Shot Detector (Object Detection)** [Online], Available : <https://bit.ly/2n1NSyX>, [2019,September,23].
- [12] Yi,Zhang,Yongliang,Shen, An improved tiny-yolov3 pedestrian detection algorithm [Online], Available : <https://www.sciencedirect.com/science/article/pii/S003040261930155X>, [2019,September,23].

- [13] Paweł Kapica, **Implementing YOLO v3 in Tensorflow (TF-Slim)**[Online], Available : <https://bit.ly/2mzitc8>, [2019,September,23].
- [14] Christopher Dossman, **AI Scholar: New DC-SPP-YOLO Model Achieves More Accurate Real Time Object Detection**[Online], Available : <https://bit.ly/2o9ENcz>, [2019,September,23].
- [15] Achraf KHAZR, **Faster RCNN Object detection**[Online], Available : <https://towardsdatascience.com/faster-rcnn-object-detection-f865e5ed7fc4>, [2019,September,24].
- [16] Shaoqing Ren, Kaiming He, Ross and Jian Sun, 2016, **Faster R-CNN: Towards Real-Time ObjectDetection with Region Proposal Networks**, arXiv, [2016].
- [17] Liu, Wei and Anguelov, Dragomir and Erhan, Dumitru and Szegedy, Christian and Reed, Scott and Fu, Cheng-Yang and Berg, Alexander C, 2016, **Ssd: Single shot multibox detector**, European conference on computer vision, [2016,December,29], pages 21-37.
- [18] Neurohive, 2018, **VGG16 – Convolutional Network for Classification and Detection**, Available : <https://neurohive.io/en/popular-networks/vgg16/>, [2019,September, 23].
- [19] Zhong-Qiu Zhao, Member, IEEE, Peng Zheng, Shou-tao Xu, and Xindong Wu , 2019 ,**Object Detection with Deep Learning: A Review** , vol.2
- [20] Danelljan, Martin and Häger, Gustav and Khan, Fahad and Felsberg, Michael, 2014, **Accurate scale estimation for robust visual tracking**, British Machine Vision Conference, Nottingham, [2014,September,1-5].
- [21] Chen, Zhe and Hong, Zhibin and Tao, Dacheng, 2015, **An experimental survey on correlation filter-based tracking**, arXiv preprint arXiv:1509.05520, [2015,September,18].
- [22] Luo, Hao and Jiang, Wei and Zhang, Xuan and Fan, Xing and Qian, Jingjing and Zhang, Chi, 2019, **AlignedReID++: Dynamically matching local information for person re-identification**, Pattern Recognition, [2019], vol.94, pages 53-61.
- [23] Zhang, Xuan and Luo, Hao and Fan, Xing and Xiang, Weilai and Sun, Yixiao and Xiao, Qiqi and Jiang, Wei and Zhang, Chi and Sun, Jian, 2017, **Alignedreid: Surpassing human-level performance in person re-identification**, arXiv preprint arXiv:1711.08184, [2018,January, 31]
- [24] Ma, Shugao and Bargal, Sarah Adel and Zhang, Jianming and Sigal, Leonid and Sclaroff, Stan, 2017, **Do less and achieve more: Training cnns for action recognition utilizing action images from the web**, Pattern Recognition, [2017], vol.68, pages 334-345.

- [25] Kong, Yu and Fu, Yun, 2018, **Human action recognition and prediction: A survey**, arXiv preprint arXiv:1806.11230, [2018].
- [26] Karpathy, Andrej and Toderici, George and Shetty, Sanketh and Leung, Thomas and Sukthankar, Rahul and Fei-Fei, Li, 2014, **Large-scale video classification with convolutional neural networks**, Proceedings of the IEEE conference on Computer Vision and Pattern Recognition, [2014], pages 1725-1732.
- [27] Abu-El-Haija, Sami and Kothari, Nisarg and Lee, Joonseok and Natsev, Paul and Toderici, George and Varadarajan, Balakrishnan and Vijayanarasimhan, Sudheendra, 2016, **Youtube-8m: A large-scale video classification benchmark**, arXiv preprint arXiv:1609.08675, [2016,September,27].
- [28] Liu, Xiaobin and Zhang, Shiliang and Huang, Tiejun and Tian, Qi, 2017, **E ^2 BoWs: An End-to-End Bag-of-Words Model via Deep Convolutional Neural Network**, arXiv preprint arXiv:1709.05903, [2017].
- [29] Yue-Hei Ng, Joe and Hausknecht, Matthew and Vijayanarasimhan, Sudheendra and Vinyals, Oriol and Monga, Rajat and Toderici, George, 2015, **Beyond short snippets: Deep networks for video classification**, Proceedings of the IEEE conference on computer vision and pattern recognition, [2015], pages 4694-4702.
- [30] Jordan, Michael I and Jacobs, Robert A, 1994, **Hierarchical mixtures of experts and the EM algorithm**, Neural computation, [1994], vol.6, pages 181-214.
- [31] Caba Heilbron, Fabian and Escorcia, Victor and Ghanem, Bernard and Carlos Niebles, Juan, 2015, **Activitynet: A large-scale video benchmark for human activity understanding**, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, [2015], pages 961-970.
- [32] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, Jitendra Malik , 2017 , **A video dataset of spatio-temporally localized atomic visual actions** , vol.4 , pp.1-15.
- [33] Limin Wang,Yu Qiao Xiaoou Tang,Luc Van Gool, 2016,**Actionness Estimation Using Hybrid Fully Convolutional Networks**, 25 Apr 2016 , vol.1
- [34] Xiaojiang Peng,Cordelia Schmid , 2016 ,**Multi-region two-stream R-CNN for action detection**, 5 Jan 2017, vol.3
- [35] Action Tubelet Detector for Spatio-Temporal Action Localization, 2017, **Action Tubelet Detector for Spatio-Temporal Action Localization**, 21 Aug 2017,vol.3

- [36] Mathew Monfort, Alex Andonian, Bolei Zhou, Kandan Ramakrishnan, Sarah Adel Bargal, Yan Yan, Lisa Brown, Quanfu Fan, Dan Gutfreund, Carl Vondrick, et al, 2019, **Moments in time dataset: one million videos for event understanding.**, IEEE transactions on pattern analysis and machine intelligence, [2019].
- [37] Schuler, Karin Kipper, 2005, **Verbnet: A Broad-coverage, Comprehensive Verb Lexicon**, [2005].
- [38] Zaghouani, Wajdi and Diab, Mona and Mansouri, Aous and Pradhan, Sameer and Palmer, Martha, 2010, **The Revised Arabic PropBank**, [2010], pages 222-226.
- [39] Baker, Collin F. and Fillmore, Charles J. and Lowe, John B., 1998 **The Berkeley FrameNet Project**, Proceedings of the 17th International Conference on Computational Linguistics, [1998], vol.1, pages 86-90.
- [40] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, Jian, 2016, **Deep residual learning for image recognition**, Proceedings of the IEEE conference on computer vision and pattern recognition, [2016], pages 770-778.
- [41] , 2018 Joao Carreira and Andrew Zisserman, 2018, **Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset**, [2018,February,12].
- [42] Szegedy, Christian and Liu, Wei and Jia, Yangqing and Sermanet, Pierre and Reed, Scott and Anguelov, Dragomir and Erhan, Dumitru and Vanhoucke, Vincent and Rabinovich, Andrew, 2015, **Going deeper with convolutions**, [2014,September,17], pages 1-9.
- [43] Darkpgmr.leek, 2013, **DarkLabel1.3-Image Labeling and Annotation Tool** [Online], Available : <https://darkpgmr.tistory.com/16>, [2019,June,25].
- [44] Cartucho, 2017, **OpenLabeling: open-source image and video labeler** [online] , Available : <https://github.com/Cartucho/OpenLabeling>, [2019,September,23].

ภาคผนวก

ภาคผนวก ก

ตัวอย่างชุดข้อมูลที่ผู้วิจัยสร้างขึ้น

ตัวอย่างชุดข้อมูลสำหรับการทดสอบโมเดลปัญญาประดิษฐ์ในการตรวจจับภาพบุคคล



รูปที่ ก.1: รูปผลลัพธ์การทำงานของหน้าต่าง Track