



เทรนเนอร์อัจฉริยะ

The Artificial Intelligence trainer

ปริญญานิพนธ์

ของ

อติเทพ คำภีระ

โครงการปริญญานิพนธ์เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

สาขาวิชาเทคโนโลยีสารสนเทศ

คณะวิทยาการสารสนเทศ มหาวิทยาลัยมหาสารคาม

พฤษภาคม 62

คำนำ

โครงการนี้เป็นระบบ ปัญญาประดิษฐ์ (Artificial Intelligence) ที่ช่วยให้การออกกำลังกายด้วยน้ำหนัก(Weight Training) ได้มีประสิทธิภาพมากขึ้น โดยเป็นตัวช่วยในการออกกำลังกายเพื่อลดความเสี่ยงในการเกิดการบาดเจ็บจากการออกกำลังกายที่ผิดวิธี โดยโปรแกรมจะตรวจสอบการเคลื่อนไหวของผู้ออกกำลังกายและสามารถบอกผู้ใช้ กำลังออกกำลังกายท่าใด และบอกได้ว่าถูกต้องหรือไม่ โดยหวังว่าระบบที่สร้างขึ้นจะสามารถช่วยทำให้คนที่ไม่ออกกำลังกายหันมาออกกำลังกายและ คนที่ออกกำลังกายอยู่แล้ว ออกกำลังกายได้ถูกวิธี และสามารถพัฒนาร่างกายได้ตามที่ต้องการ

อติเทพ คำภีระ

สารบัญ

เรื่อง	หน้า
คำนำ.....	ก
สารบัญ.....	ข
สารบัญรูปภาพ	ง
สารบัญตาราง.....	ฉ
กิตติกรรมประกาศ.....	ช
บทที่ 1	1
1.1.หลักการเหตุและผล	1
1.2 ขอบเขต.....	1
1.3 ลักษณะการทำงานของระบบ	2
1.4 เครื่องมือที่ใช้ในการดำเนินงาน	3
1.5 แผนการและระยะเวลาการดำเนินงาน.....	4
บทที่ 2	5
2.1 เทคโนโลยีที่ใช้ในการวิจัย.....	5
2.2 งานวิจัยที่เกี่ยวข้อง	15
บทที่ 3	17
3.1 การออกแบบภาพรวมและหลักการทำงาน.....	17
3.2 กระบวนการวิเคราะห์ข้อมูลด้วย CRISP-DM	20
3.3 ตารางตัวอย่างข้อมูล.....	23
3.4 การออกแบบ USE CASE DIAGRAM.....	26
บทที่ 4	28
4.1 GUI.....	28
4.2 การติดตั้งระบบ.....	31
บทที่ 5	32
5.1 สรุปผล	32
5.2 ปัญหาในการดำเนินงาน.....	32

สารบัญ (ต่อ)

เรื่อง	หน้า
การอ้างอิง	33
ประวัติผู้จัดทำโครงการปริญญานิพนธ์	34

สารบัญรูปภาพ

รูปภาพประกอบ	หน้า
ภาพประกอบที่ 2-1 อธิบายการทำงานของปัญญาประดิษฐ์	5
ภาพประกอบที่ 2-2 รูปแบบการทำงานของ Deepmind's AlphaGo	6
ภาพประกอบที่ 2-3 Machine Learning.....	7
ภาพประกอบที่ 2-4 Neural Network และ Deep Learning.....	8
ภาพประกอบที่ 2-5 แสดง Error rate จากการแข่งขัน Imagenet	9
ภาพประกอบที่ 2-6 การทำงานของ Openpose	10
ภาพประกอบที่ 2-7 ขั้นตอนการทำงานของ Openpose	11
ภาพประกอบที่ 2-8 Top-down Approach: Person Detection + Pose Estimation	12
ภาพประกอบที่ 2-9 Faster R-CNN.....	12
ภาพประกอบที่ 2-10 Bottom-up Approach: Parts Detection and Parts Association.....	13
ภาพประกอบที่ 2-11 Parts Detection.....	13
ภาพประกอบที่ 2-12 Part Affinity Fields for Part Association	14
ภาพประกอบที่ 2-13 โครงงานห้องลองเสื้อเสมือนโดยใช้คอมพิวเตอร์เทคเรียลลิตี้.....	15
ภาพประกอบที่ 2-14 การคาดเดาท่าทางมนุษย์เมื่อมีสิ่งกีดขวาง	16
ภาพประกอบที่ 3-1 ภาพการทำงานการตรวจสอบท่า Push ups.....	17
ภาพประกอบที่ 3-2 ภาพการทำงานการตรวจสอบท่า Squat.....	18
ภาพประกอบที่ 3-3 ภาพการทำงานการตรวจสอบท่า Deadlift	18
ภาพประกอบที่ 3-4 ภาพการทำงานการตรวจสอบท่า Dumbbell Shoulder Press	19
ภาพประกอบที่ 3-5 ภาพการทำงานการตรวจสอบท่า Barbell Curl	19
ภาพประกอบที่ 3-6 กระบวนการวิเคราะห์ข้อมูลด้วย CRISP-DM.....	20
ภาพประกอบที่ 3-7 แสดงการแบ่งชุดข้อมูล	21
ภาพประกอบที่ 3-8 การ Training Model	22
ภาพประกอบที่ 3-9 แสดงจุด human tracking keypoint.....	23
ภาพประกอบที่ 3-10 ตัวอย่าง raw data ที่ได้จาก Openpose	24
ภาพประกอบที่ 3-11 Use Case Diagram.....	26
ภาพประกอบที่ 4-1 หน้า การทำนายชื่อท่าออกกำลังกาย.....	28
ภาพประกอบที่ 4-2 นำไฟล์ video upload เข้าไปในโปรแกรมเพื่อนทำนาย	28
ภาพประกอบที่ 4-3 ผลลัพธ์ที่ได้จะการนำไฟล์เข้าไปทำนาย	29

สารบัญรูปภาพ (ต่อ)

รูปภาพประกอบ

หน้า

ภาพประกอบที่ 4-4 หน้า ตรวจสอบความถูกต้องในการเคลื่อนไหวในการออกกำลังกาย	29
ภาพประกอบที่ 4-5 ตรวจสอบการเคลื่อนไหวในท่า pushup โดยใช้ file video.....	30
ภาพประกอบที่ 4-6 ตรวจสอบการเคลื่อนไหวในท่า curl โดยใช้ webcam.....	30
ภาพประกอบที่ 4-7 หน้า การนำข้อมูลเข้าและ สร้าง model	31

สารบัญตาราง

ตาราง

หน้า

ตารางที่ 1-1 : แสดงระยะเวลาจัดทำโครงการ.....	15
ตารางที่ 3-1 : แสดงข้อมูลที่ได้จาก Openpose.....	25
ตารางที่ 3-2 : Use Case Diagram ตรวจสอบท่าทางในการออกกำลังกาย.....	26
ตารางที่ 3-3 : Use Case Diagram ตรวจสอบชื่อท่าออกกำลังกาย.....	26
ตารางที่ 3-4 : Use Case Diagram เพิ่มท่าออกกำลังกาย.....	27
ตารางที่ 3-5 : Use Case Diagram ลบท่าออกกำลังกาย.....	27
ตารางที่ 3-6 : Use Case Diagram แก้ไขท่าออกกำลังกาย.....	27

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี เนื่องจากได้รับความอนุเคราะห์ช่วยเหลือแนะนำให้คำปรึกษา รวมทั้งให้แนวทางในการแก้ไขปัญหาจากบุคคลและกลุ่มบุคคลต่าง ๆ ซึ่งผู้จัดทำปริญญานิพนธ์มีความซาบซึ้งเป็นอย่างยิ่ง จึงใคร่ขอขอบคุณทุกท่านมา ณ โอกาสนี้

ขอขอบพระคุณ ผู้ช่วยศาสตราจารย์ จิรัฏฐ ญบุญอบ อาจารย์ที่ปรึกษาโครงงานปริญญานิพนธ์ ที่กรุณาให้คำปรึกษา และตรวจสอบความถูกต้อง แนะนำแนวทางในการแก้ไขปัญหาจนสำเร็จลุล่วงได้ด้วยดี

ขอขอบพระคุณอาจารย์ในภาควิชาเทคโนโลยีสารสนเทศทุกท่าน ที่ประสิทธิ์ประสาทความรู้ที่มีคุณค่ายิ่งแก่ศิษย์ ไม่ว่าจะเป็นด้านคุณธรรมและจริยธรรมซึ่งล้วนแล้วแต่มีประโยชน์นานัปการ

ขอขอบคุณเพื่อนๆ ที่ให้ความช่วยเหลือและเป็นกำลังใจซึ่งกันและกัน

ขอกราบขอบพระคุณ คุณพ่อ คุณแม่ ญาติพี่น้อง ที่ให้กำลังใจ ให้โอกาส และสนับสนุนเรื่องค่าใช้จ่ายในการเล่าเรียน และการศึกษาในระดับปริญญาตรี

โครงงานนี้จะไม่บรรลุเป้าหมายได้ ถ้าหากไม่ได้ความกรุณาจากบุคคลที่ได้กล่าวถึงทั้งหมด คุณประโยชน์ใดใดอันเกิดจากปริญญานิพนธ์ฉบับนี้ ผู้ศึกษาขอมอบแก่ผู้มีพระคุณทุกท่าน

นายอดิเทพ คำภีระ

บทที่ 1

บทนำ

1.1. หลักการเหตุและผล

ในปัจจุบันมีผู้สนใจดูแลสุขภาพและร่างกายด้วยการออกกำลังกายเป็นจำนวนมาก แต่ผู้ที่ได้รับบาดเจ็บจากการออกกำลังกายก็มากขึ้นเช่นกัน ในนักกีฬาที่มีอาชีพก็จะมีเทรนเนอร์มืออาชีพคอยดูแล แต่บุคคลทั่วไปการจะมีเทรนเนอร์คอยดูแลก็จะมีค่าใช้จ่ายที่สูง การใช้ AI(Artificial Intelligence) มาทำหน้าที่แทนเทรนเนอร์จึงเป็นทางเลือกหนึ่งที่แก้ไขปัญหาดังกล่าวได้ AI จะช่วยในการตรวจสอบท่าทางการออกกำลังกายว่าผู้ออกกำลังกายทำท่าถูกต้องหรือไม่

ในปัจจุบัน AI เข้ามามีบทบาทในการทำงานและแก้ไขปัญหาต่าง ๆ และมนุษย์ ไม่ว่าจะเป็นไปในทางการแพทย์ ทางวิศวกรรม และงานอื่น ๆ การนำ AI มาเป็นผู้ช่วยในการออกกำลังกายจึงเป็นวิธีที่จะทำให้การออกกำลังกายของมนุษย์ในอนาคตมีการเปลี่ยนแปลงและประสิทธิภาพที่ดียิ่งขึ้น

ด้วยที่เล็งเห็นความสำคัญและความสามารถของเทคโนโลยี AI และความสำคัญของสุขภาพของผู้คนยุคปัจจุบันที่กล่าวมาข้างต้นจึงมีแนวคิดที่จะ สร้าง AI ที่เป็นตัวช่วยในการออกกำลังกายเพื่อลดความเสี่ยงในการเกิดการบาดเจ็บจากการออกกำลังกายที่ผิดวิธี การใช้แนวคิดนี้นอกจากจะทำให้ผู้ใช้ลดความเสี่ยงจะเกิดการบาดเจ็บแล้วยังทำให้ผู้ใช้เกิดการพัฒนาร่างกายได้มีประสิทธิภาพที่ดีขึ้นอีกด้วย

1.2 ขอบเขต

1.2.1 ผู้ดูแลระบบ

- สามารถเพิ่มข้อมูลในการสร้างโมเดลทำนายได้

1.2.2 ผู้ใช้งาน

- สามารถทำนายชื่อท่าออกกำลังกายด้วยไฟล์วิดีโอ
- สามารถทำนายชื่อท่าออกกำลังกายด้วย webcam
- สามารถตรวจสอบการเคลื่อนไหวในการออกกำลังกายด้วยไฟล์วิดีโอ
- สามารถตรวจสอบการเคลื่อนไหวในการออกกำลังกายด้วย webcam

1.3 ลักษณะการทำงานของระบบ

1.3.1 การตรวจจับท่าทางคัดแยกข้อมูลจุดท่าทางที่สำคัญสำหรับร่างกาย

- 1.3.1.1 จับจุดการเคลื่อนไหวของ ขา
- 1.3.1.2 จับจุดการเคลื่อนไหวของ แขน
- 1.3.1.3 จับจุดการเคลื่อนไหวของ ไหล่
- 1.3.1.4 จับจุดการเคลื่อนไหวของ ลำตัว
- 1.3.1.5 จับจุดการเคลื่อนไหวของ คอ

1.3.2 แจ้งเตือนและแสดงผล

- 1.3.2.1 มีการระบุตำแหน่งของข้อต่อในร่างกาย
- 1.3.2.2 มีการแจ้งว่าผู้ใช้กำลังออกกำลังกายประเภท Weight Training ด้วยท่าใด
- 1.3.2.3 มีการแจ้งว่าผู้ใช้ออกกำลังกายถูกวิธีหรือไม่
 - 1.3.2.3.1 องศาท่าทางถูกต้องเส้นมาร์คจุดข้อต่อจะเป็นสีเขียว
 - 1.3.2.3.2 องศาท่าทางไม่ถูกต้องเส้นมาร์คจุดจะเป็นสีแดง

1.3.3 ท่าออกกำลังกายมาตรฐานที่ใช้ทดสอบ

- 1.3.3.1 Push ups
- 1.3.3.2 Squat
- 1.3.3.3 deadlift
- 1.3.3.4 Dumbbell Shoulder Press
- 1.3.3.5 Barbell Curl

1.4 เครื่องมือที่ใช้ในการทำงาน

1.4.1 คอมพิวเตอร์

1.4.1.1.1 CPU: AMD Ryzen 7 2700X

1.4.1.1.2 GPU: Nvidia GTX 1070-Ti

1.4.1.1.3 SSD: Samsung 970 Evo NVMe PCIe M.2 250GB

1.4.1.1.4 RAM: G.SKILL F4 DDR4 2400 C15 2x8GB

1.4.1.1.5 MBD: Asus ROG STRIX X470-F GAMING

1.4.1.1.6 Web cam

1.4.2 ซอฟต์แวร์

1.4.2.1 ubuntu 18.04 lts

1.4.2.2 Open pose

1.4.2.3 CMake (cmake-gui)

1.4.2.4 Pycharm professional

1.4.3 ภาษาที่เขียน

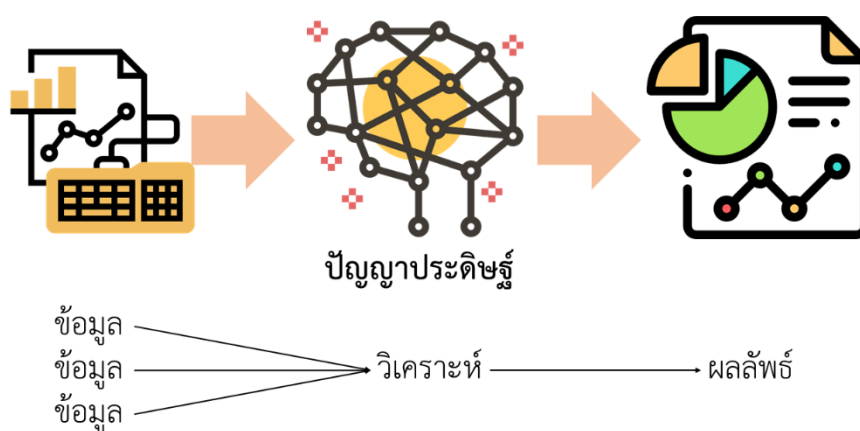
1.4.3.1 python3

บทที่ 2

ทฤษฎีและงานวิจัยที่เกี่ยวข้อง

ในการจัดการทำวิทยานิพนธ์ในด้านปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง มีความจำเป็นอย่างยิ่งที่ผู้จัดทำจะต้องทราบถึงหลักการทางเทคโนโลยีและระบบงานที่เกี่ยวข้อง เพื่อใช้เป็นแนวทางและหลักอ้างอิงในการศึกษาเพื่อให้การจัดทำวิจัยเกิดขึ้นได้ ทางผู้จัดทำจึงได้ทำการศึกษาหลักการทางทฤษฎีและระบบงานที่เกี่ยวข้องดังนี้

2.1 เทคโนโลยีที่ใช้ในการวิจัย



ภาพประกอบที่ 2-1 อธิบายการทำงานของปัญญาประดิษฐ์

2.1.1 AI (Artificial Intelligence)

ปัญญาประดิษฐ์คือสิ่งที่มนุษย์สร้างขึ้นมาเพื่อให้สามารถคิดและทำอะไรบางอย่างเองแทนมนุษย์ได้โดยไม่ต้องมีมนุษย์มาสั่ง ไม่จำเป็นต้องป้อนคำสั่งหรือโปรแกรมใด ๆ ให้กับปัญญาประดิษฐ์ การทำงานที่เกิดจากการเรียนรู้และวิเคราะห์ข้อมูลและได้ผลลัพธ์ออกมาเอง ปัญญาประดิษฐ์คือสิ่งที่พยายามทำให้ดูมีสติปัญญาเหมือนกับมนุษย์ทำสิ่งต่าง ๆ ผ่านการเรียนรู้ เหมือนกับมนุษย์ การเรียนรู้จึงเป็นสิ่งสำคัญสำหรับปัญญาประดิษฐ์ [1]

2.1.1.1 Acting Humanly การกระทำคล้ายมนุษย์

สื่อสารกับมนุษย์ได้ด้วยภาษาที่มนุษย์ใช้ เช่น ภาษาอังกฤษ เป็นการประมวลผลภาษาธรรมชาติ (natural language processing) เช่น การใช้เสียงสั่งให้คอมพิวเตอร์พิมพ์เอกสาร

- มีประสาทรับสัมผัสคล้ายมนุษย์ โดยสามารถมองเห็นและรับภาพได้โดยใช้อุปกรณ์รับสัญญาณภาพ (sensor)

- Machine learning เรียนรู้ด้วยตัวเองโดยสามารถตรวจจับรูปแบบการเกิดของเหตุการณ์ใด ๆ แล้วปรับตัวสู่สิ่งแวดล้อมที่เปลี่ยนไปได้

2.1.1.2 Thinking Humanly การคิดคล้ายมนุษย์

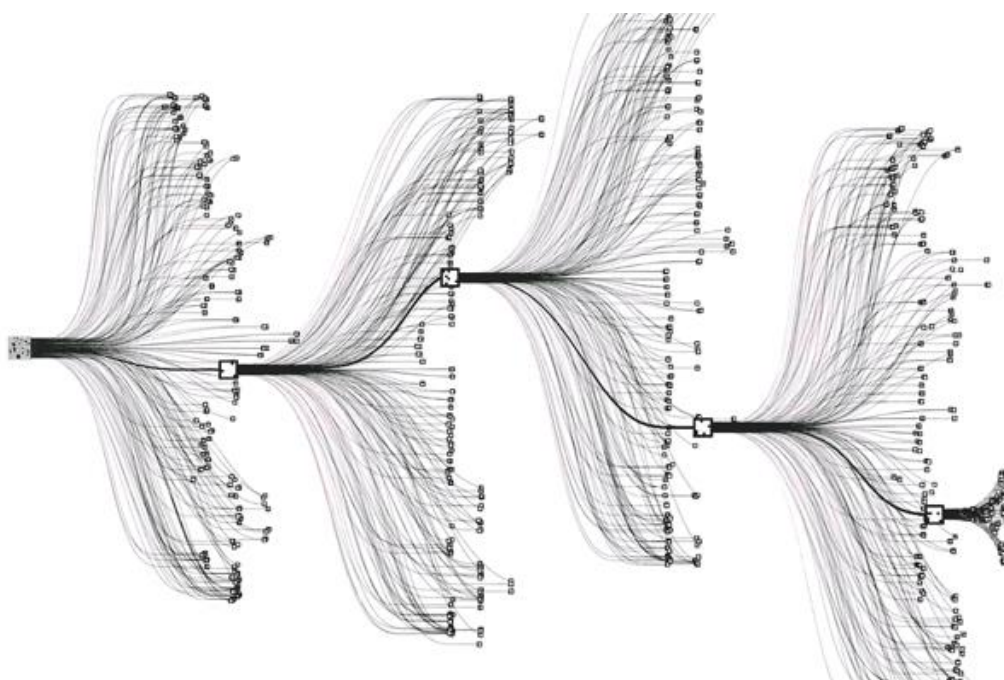
ก่อนที่จะทำให้เครื่องคิดอย่างมนุษย์ได้ ต้องรู้ก่อนว่ามนุษย์มีกระบวนการคิดอย่างไร ซึ่งการวิเคราะห์ลักษณะการคิดของมนุษย์เป็นศาสตร์ด้าน cognitive science เช่น ศึกษาโครงสร้างสามมิติของเซลล์สมอง การแลกเปลี่ยนประจุไฟฟ้าระหว่างเซลล์สมอง วิเคราะห์การเปลี่ยนแปลงทางเคมีไฟฟ้าในร่างกายระหว่างการคิด

2.1.1.3 Thinking rationally คิดอย่างมีเหตุผล

คิดอย่างมีเหตุผล หรือคิดถูกต้อง โดยใช้หลักตรรกศาสตร์ในการคิดหาคำตอบอย่างมีเหตุผล

2.1.1.4 Acting rationally กระทำอย่างมีเหตุผล

กระทำอย่างมีเหตุผล โดยจะตอบสนองต่อการกระทำแต่ละแบบโดยการนำเอาสภาพแวดล้อมเข้ามาวิเคราะห์ร่วมด้วย



ภาพประกอบที่ 2-2 รูปแบบการทำงานของ Deepmind's AlphaGo

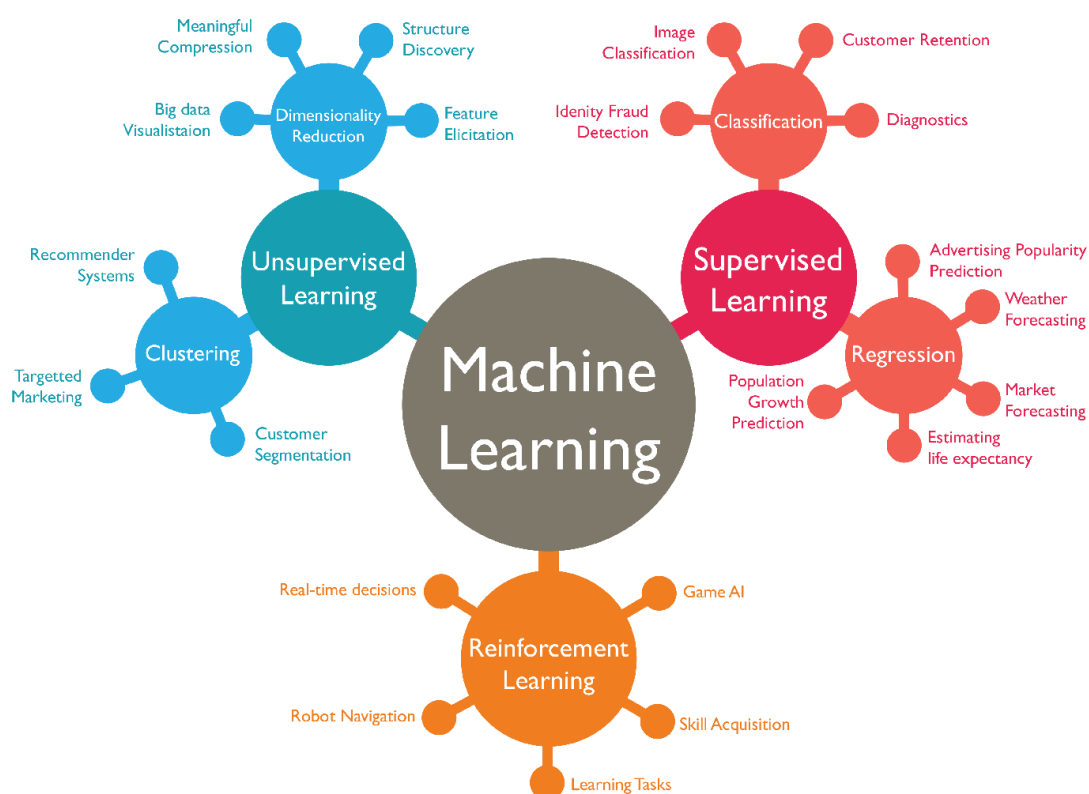
ตัวอย่างล่าสุดคือ AI ของ Google (Deepmind's AlphaGo) สามารถทำงานโดยการรับรู้และแยกแยะภาพและวิเคราะห์คำต่าง ๆ ว่าเป็นภาพหรือคำนั้นหมายถึงอะไรได้ นอกจาก

ความสามารถในการแยกแยะและการรับรู้แล้ว ยังมีความสามารถในการทำงานโดยการเรียนรู้ ซึ่งแผนกวิจัยของ Google ได้แสดงให้เห็นถึงการเรียนรู้ของหุ่นยนต์ในการเล่นเกมส์ที่ไม่เคยเล่นมาก่อน และ AI ก็ใช้เวลาไม่นานในการเรียนรู้การเล่นเกมนั้น จนในที่สุด AI ก็มีกลยุทธ์ในการเล่นเกมส์ต่าง ๆ จนสามารถเอาชนะได้อย่างรวดเร็ว [2]

2.1.2 Machine Learning

การทำให้ระบบคอมพิวเตอร์เรียนรู้ได้ด้วยตนเอง โดยใช้ ข้อมูล Machine Learning แบ่งได้ 3 แบบ [3]

- 1) Supervised Learning เรียนรู้โดยมี data มาสอน
- 2) Unsupervised Learning เรียนรู้โดยไม่มี data สอน
- 3) Reinforcement Learning เรียนรู้ตามสภาพแวดล้อม



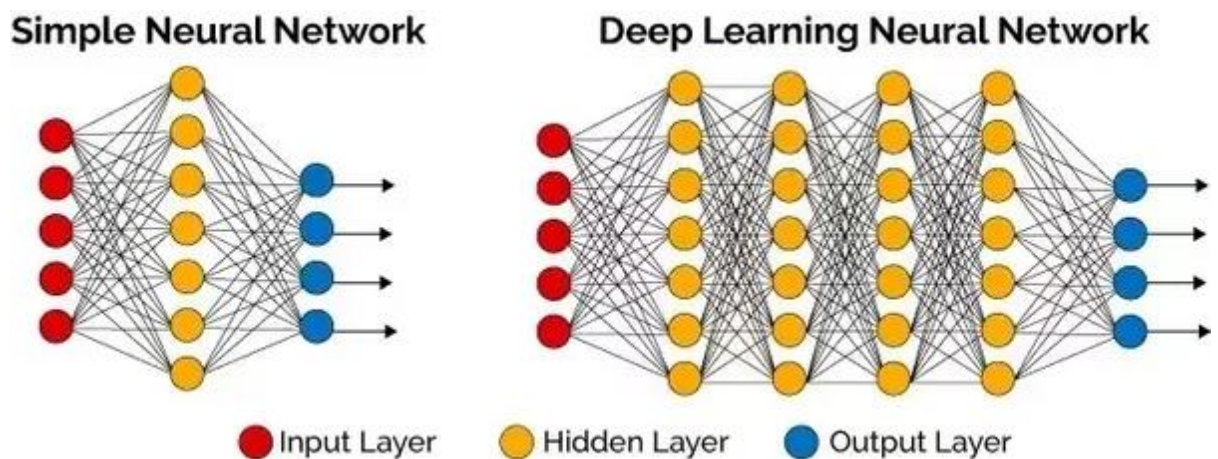
ภาพประกอบที่ 2-3 Machine Learning

Supervised Learning

คือการเรียนรู้ โดยมี data มาสอน ตัวอย่างเช่น เด็กชายต้องไปสอบแยกแยะประเภทหมาแมวเราจึงชี้ให้เด็กชาย รู้จักแมว ชี้ให้รู้จักหมา หลายๆครั้งจนเด็กจำได้ แล้วจึงอุ้มแมวมามาถามเด็กว่า นี้อะไร? เด็กก็จะแยกแยะได้ ว่านี่คือแมว คอมพิวเตอร์ก็ต้องเรียนรู้เช่นเดียวกัน

2.1.3 Deep Learning

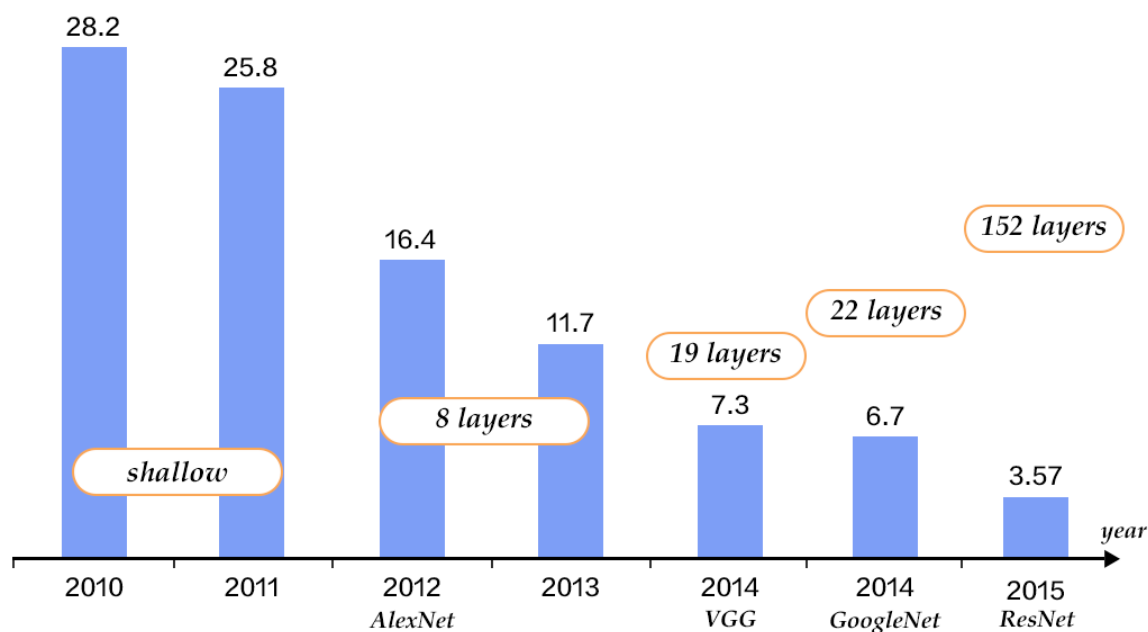
Deep Learning เป็น Machine Learning ชนิดหนึ่ง ที่ใช้การเรียนรู้ของข้อมูลโดยอาศัยความเข้าใจเป็นลำดับขั้น เพื่อให้ model เรียนรู้ความสัมพันธ์ที่มีความซับซ้อนได้ ซึ่งสามารถทำได้โดยการใช้ Deep neural network (Neural Network ที่มี hidden layer จำนวนมาก) ดังรูปที่ 2-4[4]



ภาพประกอบที่ 2-4 Neural Network และ Deep Learning

2.1.3.1 ความเป็นมาของ Deep Learning

Deep Learning เป็นที่รู้จักจากงานวิจัยที่ชื่อว่า ImageNet Classification with Deep Convolutional Neural Networks (AlexNet) ที่ได้ทำการ Classify รูปภาพ 1,200,000 รูปที่อยู่ใน Imagenet (Database ของรูปภาพที่จัดทำโดย Stanford) Imagenet จะให้นักวิจัยด้าน Machine Learning ส่งอัลกอริทึม เข้ามาแข่งขันว่าใครจะ Classify รูปภาพได้มีความผิดพลาดน้อยที่สุดและในปี 2012 Deep Learning ได้ทำ Error rate ได้น้อยที่สุด โดยได้ Error rate ที่ 15.3 % ห่างจากอันดับ 2 ถึง 10.9 % โดยอันดับ 2 ทำได้เพียง 26.2 % เลยทำให้นักวิจัยได้หันมาให้ความสนใจกับ Deep Learning

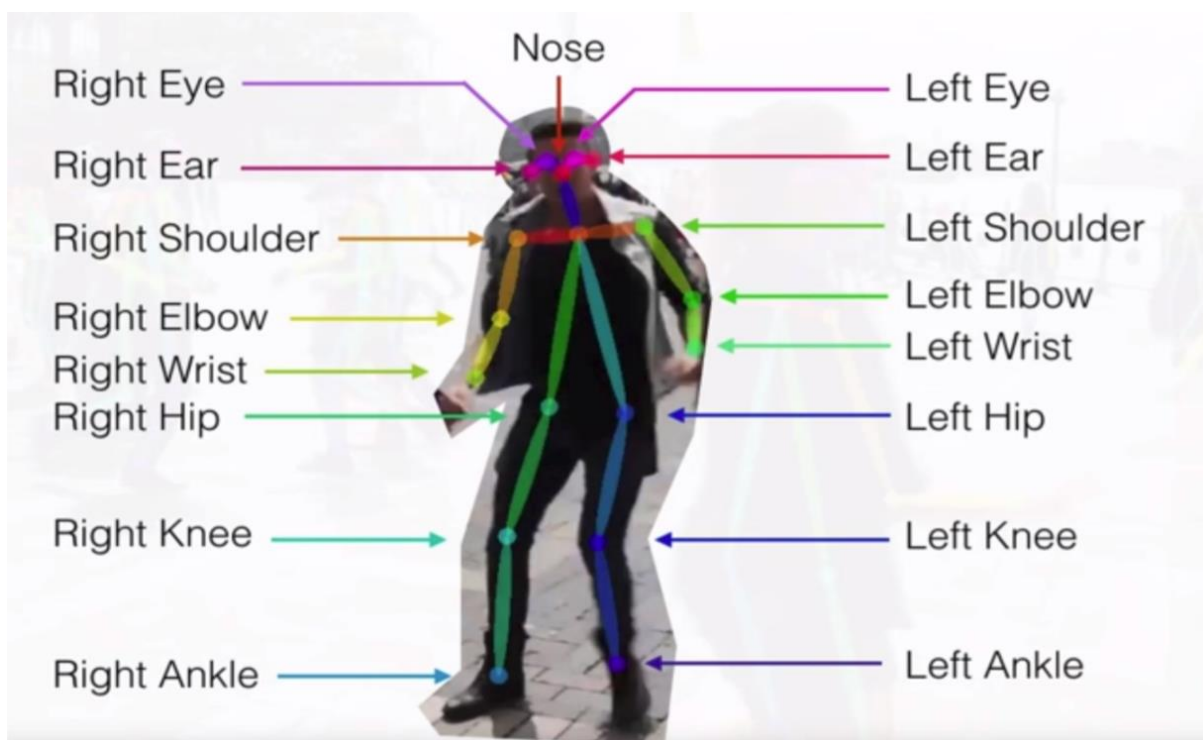


ภาพประกอบที่ 2-5 แสดง Error rate จากการแข่งขัน Imagenet

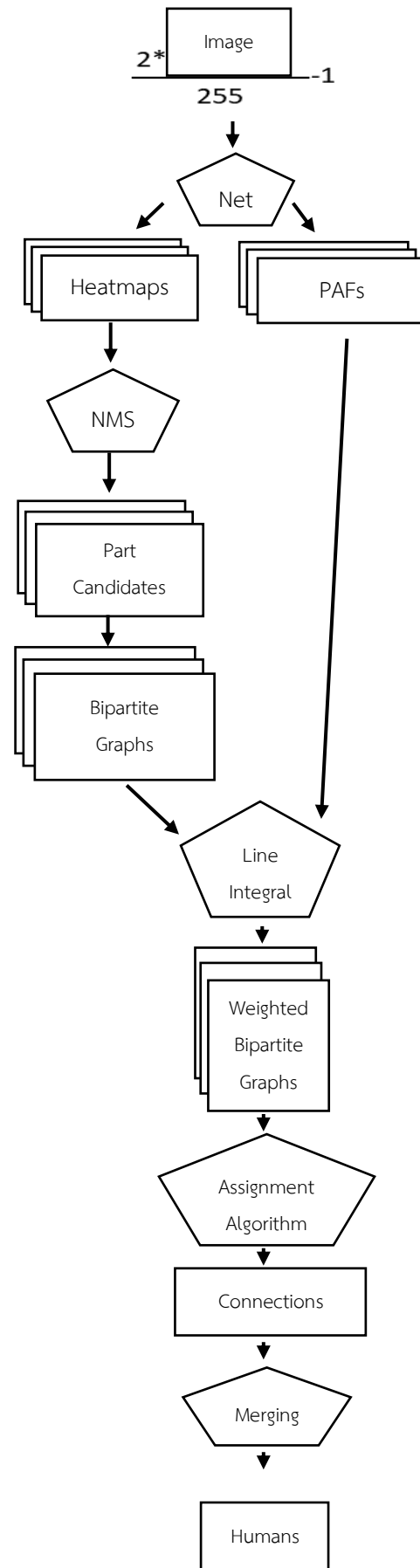
จากภาพประกอบที่ 2-5 ตั้งแต่ปี 2012 ที่มีการนำ Deep Learning มาใช้ Classify รูปภาพในการแข่งขันของ Imagenet ทำให้การแข่งขันปีหลังจากนั้นเป็นต้นมา Deep Learning จึงเป็นอัลกอริทึมหลักที่นักวิจัยเลือกใช้และเมื่อ Deep Learning ทำให้ Error rate ลดลงจนต่ำกว่า 10% จนได้ Error rate ที่นักวิจัยยอมรับว่าสามารถนำมาใช้งานจริงได้ บริษัทเทคโนโลยีต่าง ๆ จึงได้นำไปพัฒนาต่อยอดใช้กับงานด้านต่าง ๆ มากมาย

2.1.4 Openpose

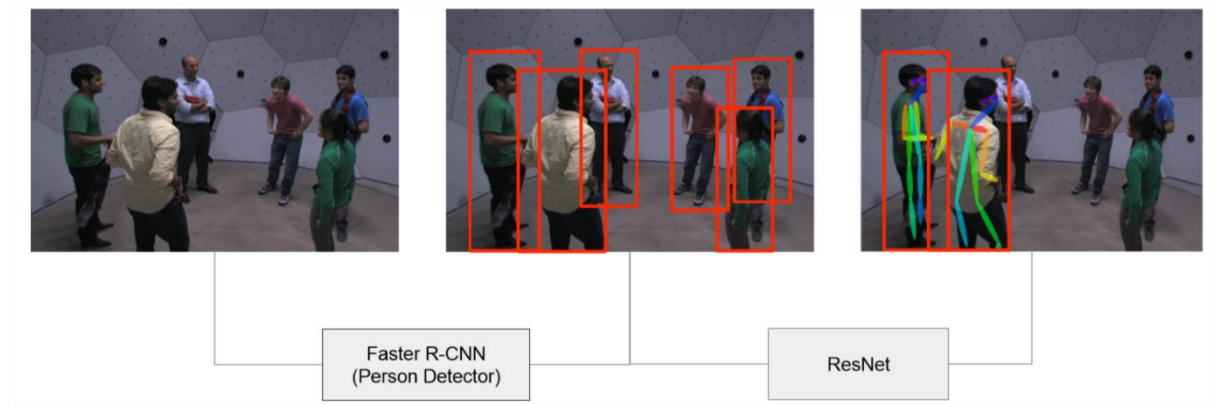
Openpose เป็น open-source library ที่ใช้ในการตรวจจับ key point หลายคนแบบเรียลไทม์ (real-time multi-person key point detection) พัฒนาโดย Gines Hidalgo, Zhe Cao, Tomas Simon, Shih-En Wei, Hanbyul Joo and Yaser Sheikh. ช่วยให้สามารถตรวจจับ 18 จุดสำคัญของร่างกายจากภาพถ่าย วิดีโอ และ Webcam



ภาพประกอบที่ 2-6 การทำงานของ Openpose

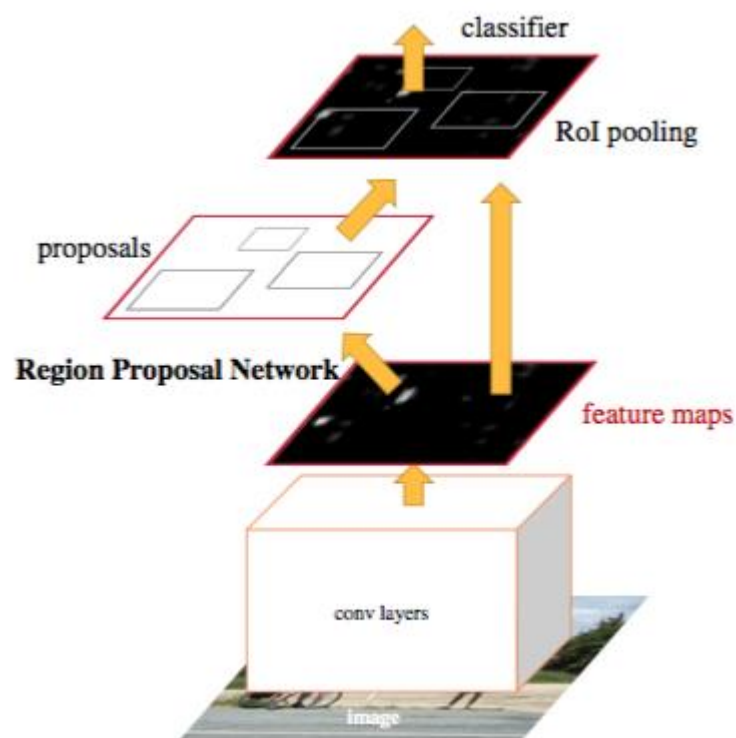


ภาพประกอบที่ 2-7 ขั้นตอนการทำงานของ Openpose



ภาพประกอบที่ 2-8 Top-down Approach: Person Detection + Pose Estimation

2.1.4.1 Faster R-CNN

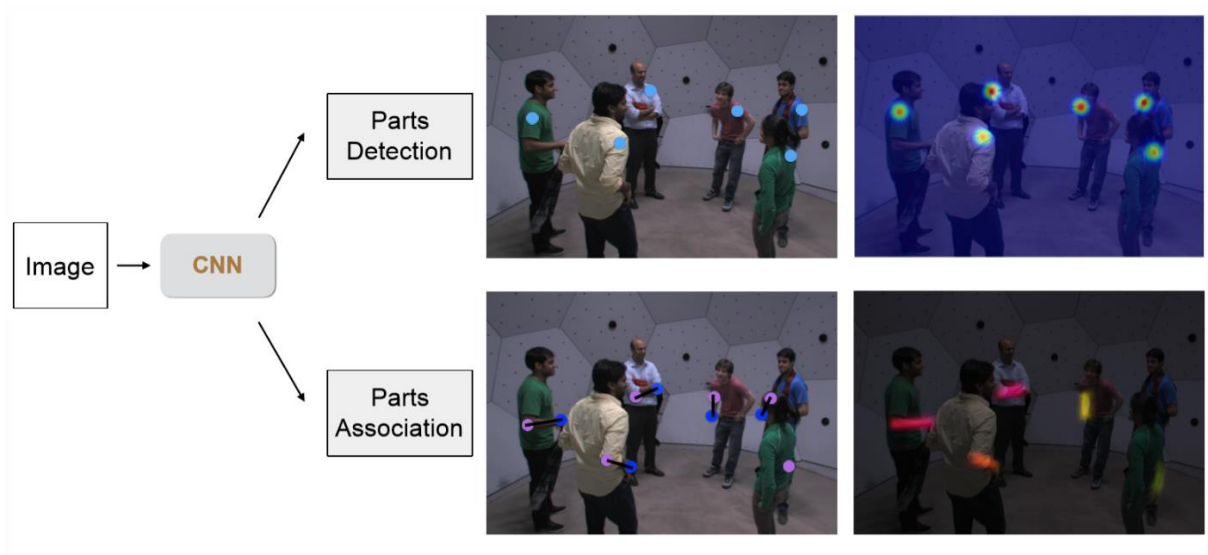


ภาพประกอบที่ 2-9 Faster R-CNN

Faster R-CNN คือ แนวความคิดที่ใช้จำแนกและตรวจจับวัตถุ ที่ได้รับการพัฒนาต่อยอดมาจาก R-CNN และ fast-R-CNN Selective Search ที่ช้าและ train ไม่ได้ Faster-RCNN เลยออกมาเพื่อพัฒนาจุดตรงนี้โดยรวม region proposal เข้าไป train เป็นส่วนหนึ่งของ CNN[5]

2.1.4.2 ResNet (Residual Network)

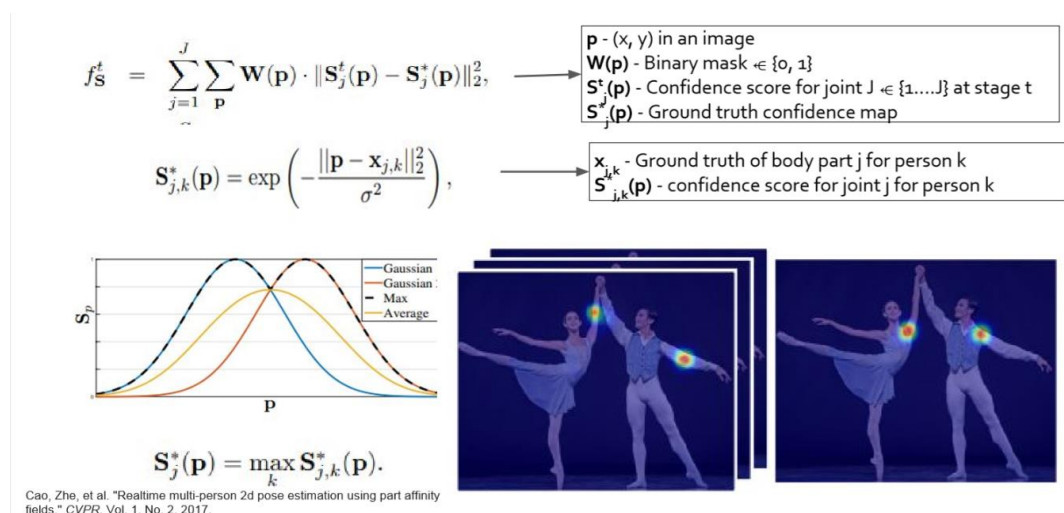
ResNet คือวิธีแก้ปัญหาเมื่อ CNN ที่มีหลายชั้น เมื่อชั้นไหนไม่จำเป็นจริง ๆ ค่าถ่วงน้ำหนักบนชั้นเหล่านั้นก็จะถูก train ให้เข้าใกล้ 0 ไปเอง ข้อมูลที่จำเป็นก็ยังสามารถ flow ผ่าน skip connection ไปชั้นต่อไปได้ นี่คือการมาของชื่อ Residual Network หรือ ResNet



ภาพประกอบที่ 2-10 Bottom-up Approach: Parts Detection and Parts Association

2.1.4.3 Parts Detection and Parts Association

ใช้อัลกอริทึม เสนอบริเวณที่น่าจะมีวัตถุอยู่ ถูกสร้างขึ้นมาโดยใช้ความรู้จากการรับรู้โดยการมองเห็นของมนุษย์ โดยใช้ feature ต่าง ๆ เช่น สี edge gradient การกระจายตัวของสี หรือ feature อื่น ๆ เพื่อทำการเสนอว่า ตรงไหนน่าจะมีวัตถุอยู่ [6] [7]



ภาพประกอบที่ 2-11 Parts Detection

$L = (L_1, L_2, \dots, L_C)$, L_i is a vector field - for each limb (c)

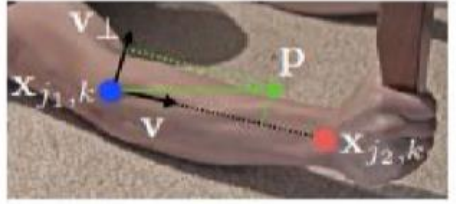
$$f_L^t = \sum_{c=1}^C \sum_p W(p) \cdot \|L_c^t(p) - L_c^*(p)\|_2^2,$$

$$L_{c,k}^*(p) = \begin{cases} v & \text{if } p \text{ on limb } c, k \\ 0 & \text{otherwise.} \end{cases}$$

$0 \leq v \cdot (p - x_{j_1,k}) \leq l_{c,k}$ and $|v_{\perp} \cdot (p - x_{j_1,k})| \leq \sigma_l$,

$$L_c^*(p) = \frac{1}{n_c(p)} \sum_k L_{c,k}^*(p),$$

v - normalized unit vector along $x_{j_1,k}, x_{j_2,k}$
 $l_{c,k}$ - distance between $x_{j_1,k}, x_{j_2,k}$
 σ_l - limb width



$n_c(p)$ - No of non-zero vectors at point p for all k people

Cao, Zhe, et al. "Realtime multi-person 2d pose estimation using part affinity fields." CVPR. Vol. 1. No. 2. 2017.

ภาพประกอบที่ 2-12 Part Affinity Fields for Part Association

2.1.5 ภาษา python

Python คือชื่อภาษาที่ใช้ในการเขียนโปรแกรมภาษาหนึ่ง ซึ่งถูกพัฒนาขึ้นมาโดยไม่ยึดติดกับแพลตฟอร์ม กล่าวคือสามารถรันภาษา Python ได้ทั้งบนระบบ Unix, Linux, Windows NT, Windows 2000, Windows XP หรือแม้แต่ระบบ FreeBSD อีกอย่างหนึ่งภาษาตัวนี้เป็น Open Source เหมือนอย่าง PHP ทำให้ทุกคนสามารถที่จะนำ Python มาพัฒนาโปรแกรมของเราได้ฟรีๆโดยไม่ต้องเสียค่าใช้จ่าย และความเป็น Open Source ทำให้มีคนเข้ามาช่วยกันพัฒนาให้ Python มีความสามารถสูงขึ้น และใช้งานได้ครอบคลุมกับทุกลักษณะงานโค้ดของ Python ถูกสร้างขึ้นมาจากภาษาซี การประมวลผลจะทำในแบบอินเทอร์พรีเตอร์ คือจะประมวลผลไปที่ละบรรทัดและปฏิบัติตามคำสั่งที่ได้รับ Python เวอร์ชันแรกคือ เวอร์ชัน 0.9.0 ออกมาเมื่อปี 2533 และเวอร์ชันปัจจุบันคือ 3.8 [8]

2.1.6 Weight training

weight training คือการออกกำลังกายโดยใช้น้ำหนัก ไม่ว่าจะเป็นใช้ อุปกรณ์ หรือใช้น้ำหนักจากตัวผู้ฝึกเองโดยการออกกำลังกายด้วยน้ำหนักจำเป็นต้องมีการฝึกท่าที่ถูกต้องเพื่อไม่ให้เกิดการบาดเจ็บ และช่วยให้พัฒนาได้ดีที่สุดโดยการออกกำลังกายด้วยน้ำหนักจุดประสงค์หลักจะเน้นไปที่การพัฒนากล้ามเนื้อเมื่อเรามีกล้ามเนื้อที่ดี เราจะสามารถต่อยอดไปกีฬาชนิดอื่นได้ง่าย [9]

2.2 งานวิจัยที่เกี่ยวข้อง

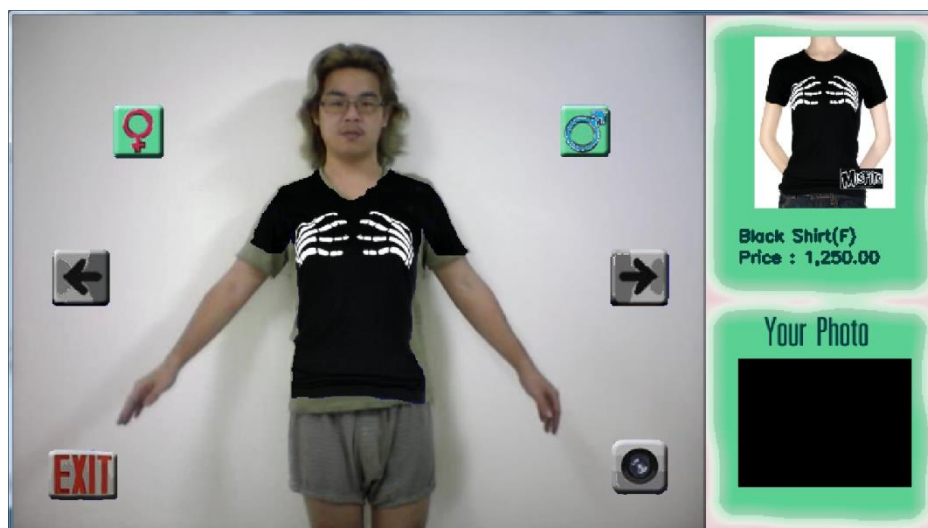
2.2.1 Virtual Fitting Room Using Augmented Reality

โครงการห้องลองเสื้อเสมือนโดยใช้คอมพิวเตอร์เทคโนโลยี เป็นโปรแกรมจำลองการลองเสื้อที่พัฒนาขึ้นเพื่อแก้ไขปัญหาของการลองเสื้อ ที่สามารถนำไปใช้งานได้จริง ซึ่งผลการทดสอบความผิดพลาดของการประมาณตำแหน่งศีรษะอยู่ที่ 24.76 pixelต่อภาพ การประมาณตำแหน่งลำตัวอยู่ที่ 30.89 pixel ต่อภาพ และการประมาณทิศทางของแขนท่อนบนอยู่ที่ 13.46 องศาต่อภาพ ซึ่งถือว่ายังไม่เป็นที่น่าพึงพอใจ ยังสามารถพัฒนาเพิ่มต่อไปได้ โดยยังพบปัญหาในกรณีที่มีความหลากหลายของสภาพแสงและท่าทางของผู้ใช้งานที่เกินข้อจำกัดของโปรแกรม แต่เมื่อทดสอบดูคุณภาพจากการใช้งานจริงแล้ว ความผิดพลาดที่เกิดขึ้นยังถือว่ายอมรับได้ คุณภาพของภาพและผลการทำงาน ถือว่าอยู่ในเกณฑ์ที่สามารถนำไปใช้งานได้ และสามารถให้ผู้สนใจนำไปใช้งานได้จริง [10]

นักประดิษฐ์ นาย ดนัยนันท์ เก่าเงิน, นาย ณัชนนท์ วงษ์วิไล

สถานที่วิจัย จุฬาลงกรณ์มหาวิทยาลัย

ตีพิมพ์เมื่อ 2553



ภาพประกอบที่ 2-13 โครงการห้องลองเสื้อเสมือนโดยใช้คอมพิวเตอร์เทคโนโลยี

2.2.2 Through-Wall Human Pose Estimation Using Radio Signals

เป็นงานวิจัยเกี่ยวกับการใช้ AI ในการวิเคราะห์ท่าทางของมนุษย์ และ คาดเดา ท่าทางเมื่อมีสิ่งกีดขวาง หรือก็คือการใช้ AI วิเคราะห์ท่าทางเมื่อ เรายู่หลังกำแพง [11]

นักประดิษฐ์ Mingmin Zhao, Tianhong Li, Mohammad Abu, Alsheikh Yonglong Tian, Hang Zhao, Dina Katabi, Antonio Torralba

สถานที่วิจัย MIT CSAIL

ตีพิมพ์เมื่อ 2018



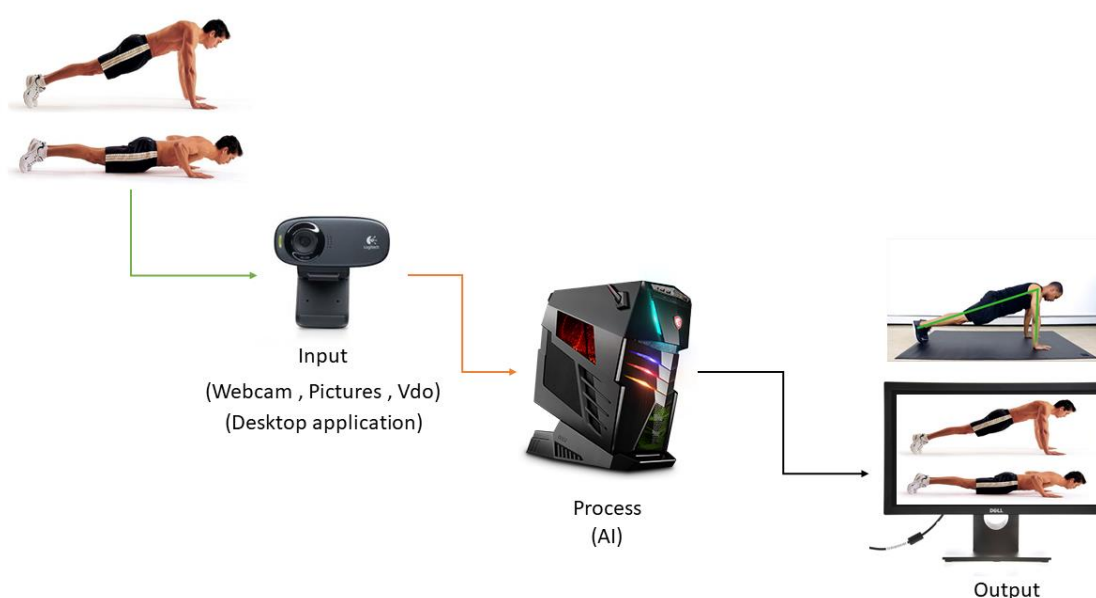
ภาพประกอบที่ 2-14 การคาดเดาท่าทางมนุษย์เมื่อมีสิ่งกีดขวาง

บทที่ 3

ขั้นตอนการดำเนินงาน

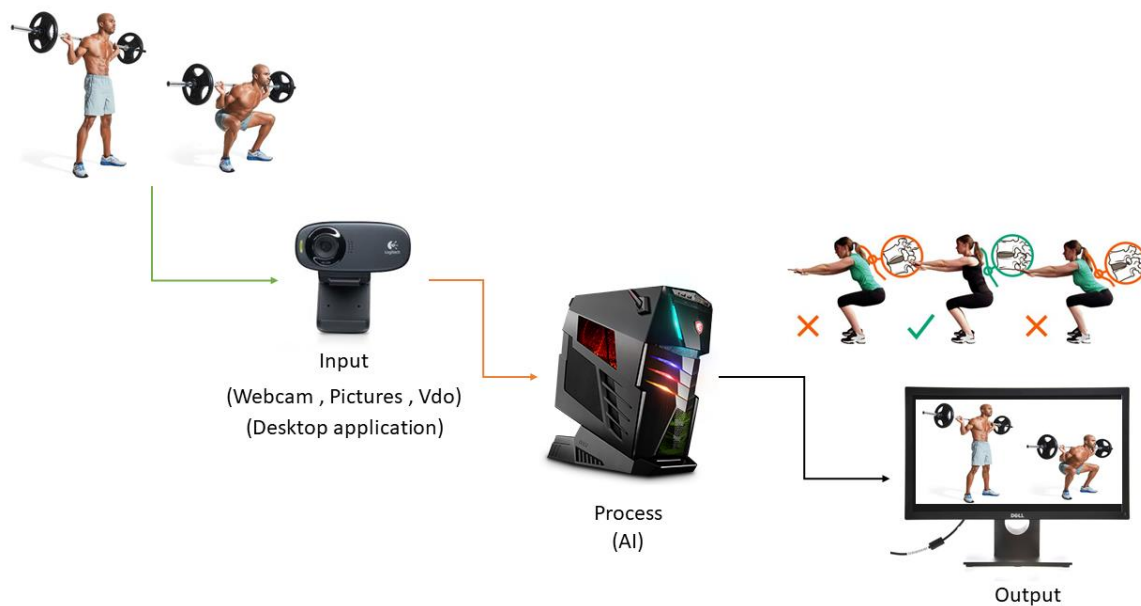
3.1 การออกแบบภาพรวมและหลักการทำงาน

การทำงานของระบบเทรนเนอร์อัจฉริยะจะเริ่มจากรับ input ผ่านทาง webcam และ นำข้อมูลไปประมวลผลสุดท้ายจะแสดงข้อความออกมาทาง Monitor บอกผู้ใช้งานว่าท่าทางออกกำลังกายถูกต้องหรือไม่ ซึ่งในระบบนี้ มีการใช้ 5 ท่าเป็นตัวอย่าง ดูได้จากภาพประกอบที่ 3-1 ถึงภาพประกอบที่ 3-5



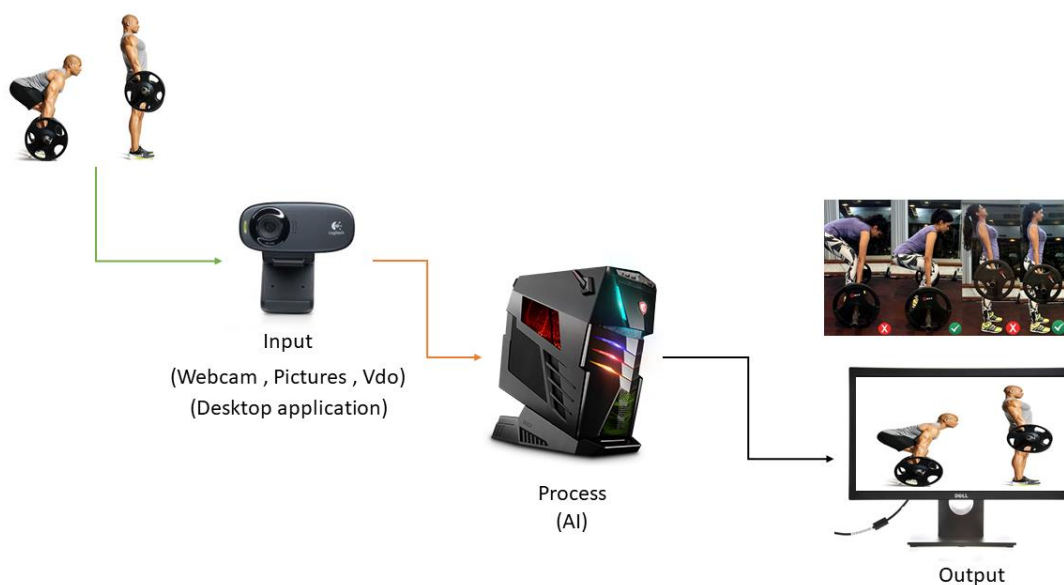
ภาพประกอบที่ 3-1 ภาพการทำงานการตรวจสอบท่า Push ups

จากภาพประกอบที่ 3-1 การตรวจสอบท่า Push ups จะตรวจสอบจากมุมด้านข้างโดยจะตรวจสอบจาก ลักษณะ ลำตัว และ แขน



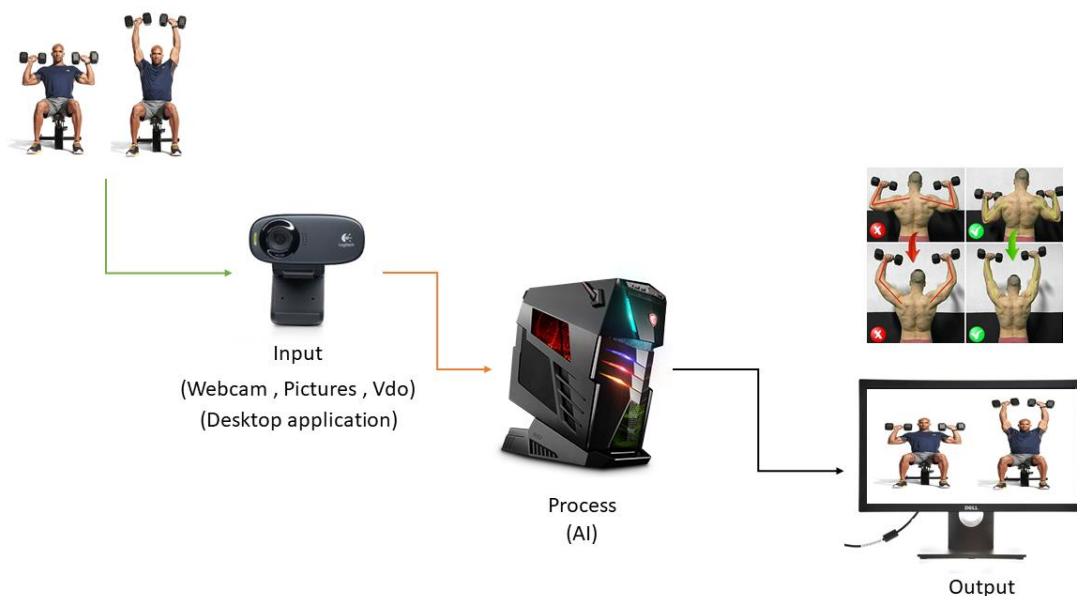
ภาพประกอบที่ 3-2 ภาพการทำงานการตรวจสอบท่า Squat

จากภาพประกอบที่ 3-2 การตรวจสอบท่า Squat จะตรวจสอบจากมุมด้านข้างโดยจะตรวจสอบจาก ลักษณะ ลำตัวส่วนบน คอ และ ขา



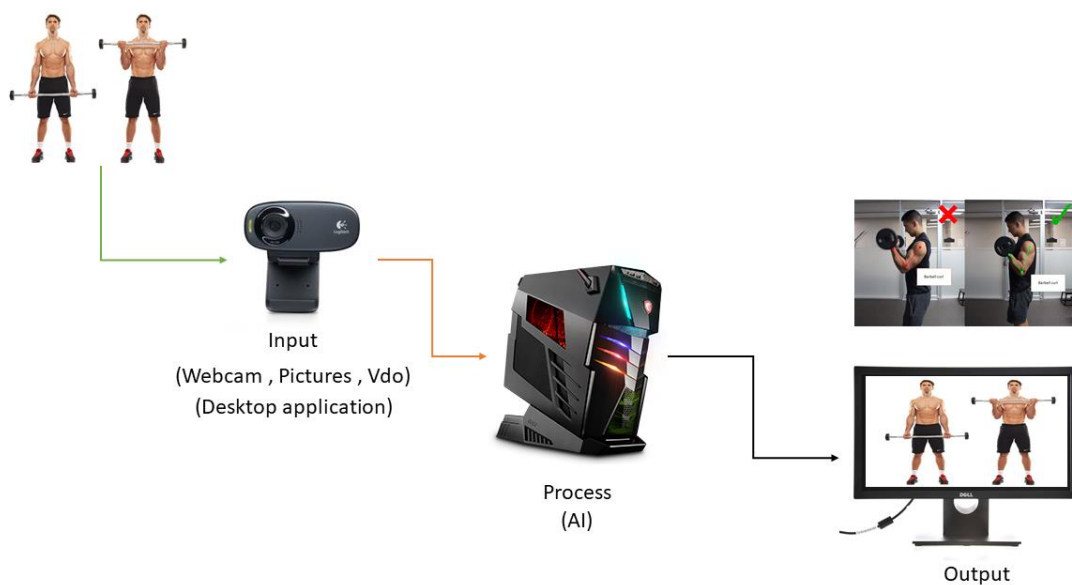
ภาพประกอบที่ 3-3 ภาพการทำงานการตรวจสอบท่า Deadlift

จากภาพประกอบที่ 3-3 การตรวจสอบท่า Deadlift จะตรวจสอบจากมุมด้านข้างโดยจะตรวจสอบจาก ลักษณะ ลำตัวส่วนบน คอ และ ขา



ภาพประกอบที่ 3-4 ภาพการทำงานการตรวจสอบท่า Dumbbell Shoulder Press

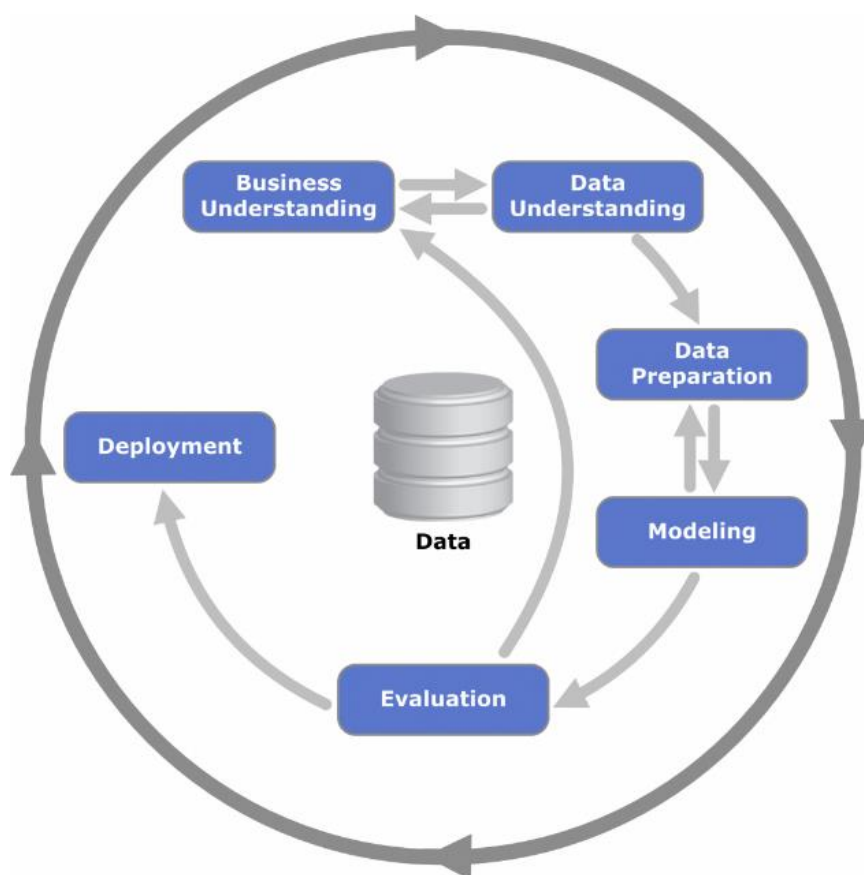
จากภาพประกอบที่ 3-4 การตรวจสอบท่า Dumbbell Shoulder Press จะตรวจสอบจาก มุมด้านหน้าโดยจะตรวจสอบจาก ลักษณะ ของ ท่อนแขนบน และ ท่อนแขนล่าง



ภาพประกอบที่ 3-5 ภาพการทำงานการตรวจสอบท่า Barbell Curl

จากภาพประกอบที่ 3-5 การตรวจสอบท่า Barbell Curl จะตรวจสอบจากมุมด้านข้างโดยจะตรวจสอบจาก ลักษณะ ของ ท่อนแขนบน และ ท่อนแขนล่าง

3.2 กระบวนการวิเคราะห์ข้อมูลด้วย CRISP-DM



ภาพประกอบที่ 3-6 กระบวนการวิเคราะห์ข้อมูลด้วย CRISP-DM

3.2.1 Business Understanding

วิเคราะห์ข้อมูลทางตลาดค้า ไม่นิ่งพร้อมทั้งวางแผนในการดำเนินการ

3.2.2 Data Understanding

เก็บรวบรวมข้อมูลและตรวจสอบข้อมูลที่ได้ทำการรวบรวมมาได้เพื่อดูความถูกต้องของข้อมูล และพิจารณาว่าจะใช้ข้อมูลทั้งหมดหรือจำเป็นต้องเลือกข้อมูลบางส่วนมาใช้ในการวิเคราะห์ จากภาพประกอบที่ 3-7 เราจะแบ่งข้อมูลออกเป็น 3 ส่วน

- 1) Training set แบ่งข้อมูล 80 % เพื่อนำไป สอน model
- 2) Development set แบ่งข้อมูล 10 % เพื่อนำไปเลือก model ที่มีค่าผิดพลาดน้อยที่สุด
- 3) Test set แบ่งข้อมูล 10 % เพื่อนำไปทดสอบ model ว่ามีประสิทธิภาพหรือไม่



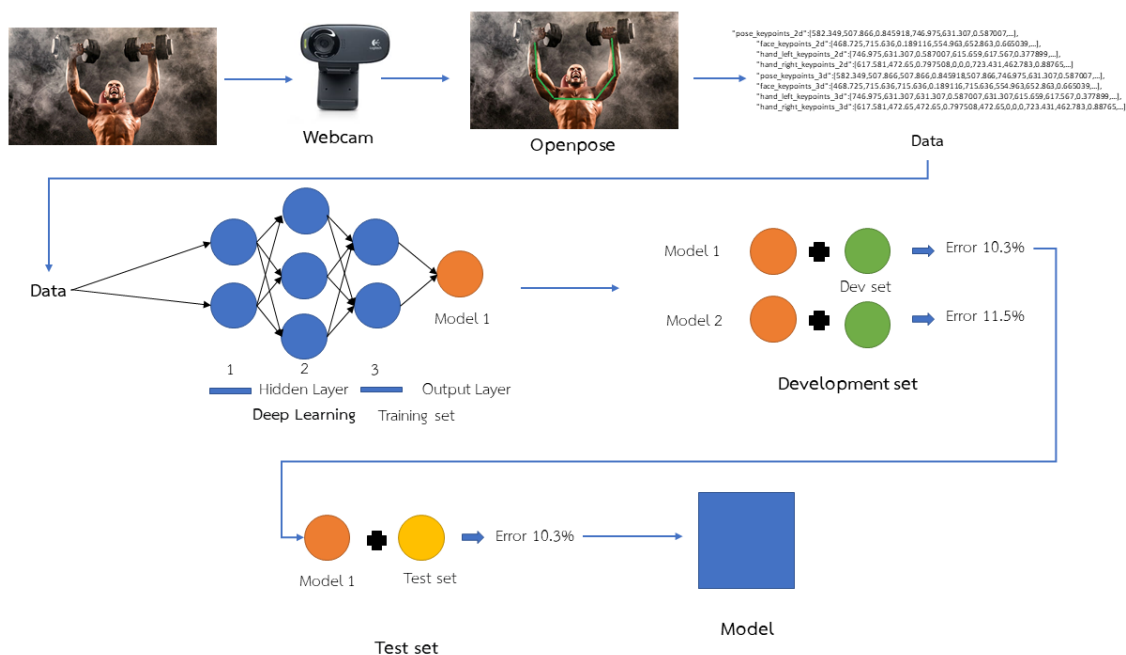
จากภาพประกอบที่ 3-7 แสดงการแบ่งชุดข้อมูล

3.2.3 Data Preparation

แปลงข้อมูลที่ได้ทำการเก็บรวบรวมมา (raw data) ให้กลายเป็นข้อมูลที่สามารถนำไปวิเคราะห์ในขั้นถัดไปได้ และทำ data cleaning เช่น การแปลงข้อมูลให้อยู่ในช่วง (scale) เดียวกัน หรือการเติมข้อมูลที่ขาดหายไปดังตารางที่ 3-1

3.2.4 Modeling

วิเคราะห์ข้อมูลด้วยเทคนิค Deep learning จากภาพประกอบที่ 3-8 แสดงหลักการเรียนรู้ของเครื่องโดยใช้การรับ input ด้วย webcam และใช้ Openpose API ในการแปลงจาก VDO มาเปลี่ยนเป็นชุดข้อมูลเพื่อนำข้อมูลไปผ่านกระบวนการ Training set และเมื่อได้ Model มาแล้วจะนำไปผ่านกระบวนการ Development set โดยวัด error จากชุดข้อมูลที่ model ไม่เคยเห็นมาก่อน เพื่อเลือกเอา Model ที่มี Error น้อยที่สุด และนำไปผ่านกระบวนการ Test Set เป็นลำดับสุดท้ายเพื่อวัด unbiased error โดยใช้ข้อมูลที่เป็นอิสระจากทุกข้อมูลที่เราเคยใช้มาทั้งหมด และสุดท้ายจะได้ Model ที่มีประสิทธิภาพที่สุด



ภาพประกอบที่ 3-8 การ Training Model

จากภาพประกอบที่ 3-8 ในการเรียนรู้จะใช้ Training set ในการสอน model ด้วยขั้นตอนของ Gradient Descent (การเคลื่อนลงตามความชัน) เป็นอัลกอริทึมที่ใช้หาค่าที่เหมาะสมที่สุดให้กับฟังก์ชันที่กำหนดขึ้นมา โดยอัลกอริทึมใช้การวนหาค่าที่ทำให้ค่าต่ำสุดจากการคำนวณจากความชันที่จุดที่เราอยู่แล้วพยายามเดินทางไปทางตรงข้ามกับความชันที่คำนวณขึ้นมา

3.2.5 Evaluation

จากภาพประกอบที่ 3-8 การทดสอบ Model จะมี 2 ขั้นตอนคือ Development set และ Test set

3.2.5.1 Development set คือขั้นตอนในการเลือก Model ตัวใดที่มีค่าความผิดพลาดน้อยที่สุดที่ได้มาจาก Training set

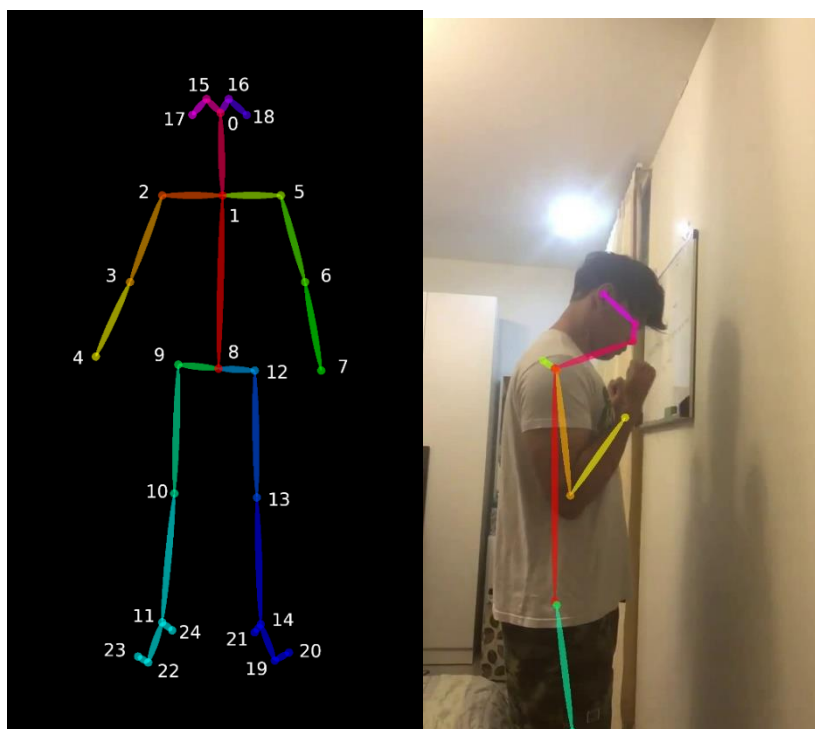
3.2.5.2 Test set คือเมื่อเราได้ Model มาแล้วจะนำมาวัด unbiased error ที่แท้จริงโดยจะใช้ข้อมูลในการทดสอบคนละชุดกับ Development set

3.2.6 Deployment

นำ Model ที่สร้างขึ้นมาไปใช้งานจริง โดยจะใช้วิธีบันทึกเก็บเป็นออบเจกต์ จะใช้โมดูลที่ชื่อ Pickle ที่เป็นโมดูลมาตรฐานของ Python ที่ใช้เก็บบันทึกออบเจกต์ไว้เพื่อใช้งานข้ามโปรแกรม

3.3 ตารางตัวอย่างข้อมูล

จากภาพประกอบที่ 3-9 แสดงข้อมูลจุด keypoints ที่ได้จาก Openpose ที่จะนำไปสร้าง model



ภาพประกอบที่ 3-9 แสดงจุด human tracking keypoint

```

"pose_keypoints_2d":[582.349,507.866,0.845918,746.975,631.307,0.587007,...],
"face_keypoints_2d":[468.725,715.636,0.189116,554.963,652.863,0.665039,...],
"hand_left_keypoints_2d":[746.975,631.307,0.587007,615.659,617.567,0.377899,...],
"hand_right_keypoints_2d":[617.581,472.65,0.797508,0,0,0,723.431,462.783,0.88765,...]
"pose_keypoints_3d":[582.349,507.866,507.866,0.845918,507.866,746.975,631.307,0.587007,...],
"face_keypoints_3d":[468.725,715.636,715.636,0.189116,715.636,554.963,652.863,0.665039,...],
"hand_left_keypoints_3d":[746.975,631.307,631.307,0.587007,631.307,615.659,617.567,0.377899,...],
"hand_right_keypoints_3d":[617.581,472.65,472.65,0.797508,472.65,0,0,0,723.431,462.783,0.88765,...]

```

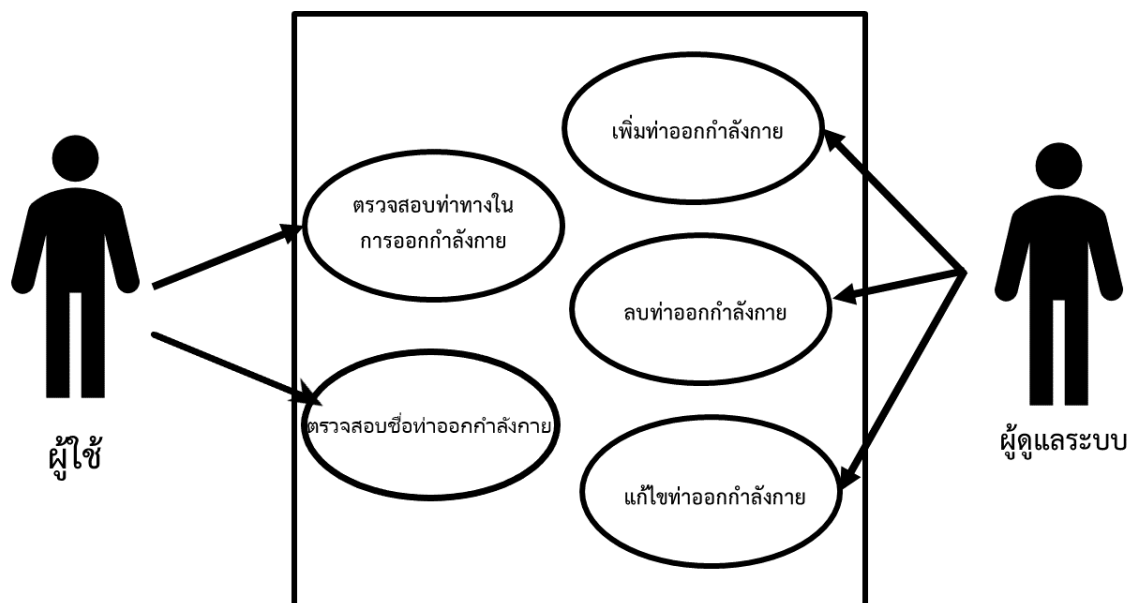
ภาพประกอบที่ 3-10 ตัวอย่าง raw data ที่ได้จาก Openpose

ตารางที่ 3.1 ตัวอย่างชุดข้อมูล

Pose Output	keypoints 1	keypoints 2	keypoints 3	keypoints 4	keypoints 5	keypoints 6
x0, "Nose"	509.948	496.216	489.066	475.202	457.875	454.211
y0, "Nose"	433.291	461.134	482.063	510.145	537.903	541.555
c0, "Nose"	0.814327	0.79594	0.782292	0.763333	0.796098	0.774719
x1, "Neck"	290.205	286.636	279.771	272.705	265.793	262.382
y1, "Neck"	607.826	614.831	621.726	628.72	628.859	632.344
c1, "Neck"	0.553807	0.531956	0.572829	0.577432	0.601575	0.576434
x2, "RShoulder"	213.355	216.861	213.451	216.909	220.44	227.355
y2, "RShoulder"	632.207	628.824	639.197	646.212	649.733	653.115
c2, "RShoulder"	0.503318	0.518037	0.53368	0.558326	0.584394	0.603601
x3, "RElbow"	185.54	199.386	206.46	209.818	206.408	202.915
y3, "RElbow"	1012.72	1016.16	1012.59	995.193	984.697	974.325
c3, "RElbow"	0.679362	0.673701	0.648009	0.582808	0.558327	0.602261
x4, "RWrist"	346.126	352.998	349.554	349.507	370.512	363.456
y4, "RWrist"	1274.48	1274.48	1250.08	1232.58	1239.5	1215.11
c4, "RWrist"	0.392098	0.351922	0.325699	0.279955	0.546362	0.634605
x5, "LShoulder"	360.001	346.04	342.564	332.057	314.64	307.605
y5, "LShoulder"	597.281	604.332	600.874	618.263	618.252	625.237
c5, "LShoulder"	0.376881	0.366441	0.400558	0.362316	0.39045	0.390814
x6, "LElbow"	311.154	314.61	314.585	293.693	293.718	286.753
y6, "LElbow"	939.391	925.407	949.866	942.88	932.395	946.387
c6, "LElbow"	0.167728	0.20738	0.217464	0.229447	0.206125	0.236712
x7, "LWrist"	374.059	373.943	374.066	373.883	370.404	366.963

ตารางที่ 3.1 แสดงข้อมูลที่ได้จาก openpose ที่ได้ผ่านขั้นตอน Data Preparation โดยข้อมูลจะประกอบไปด้วยตัวแปรแกน x , y และ c ที่ แสดงถึงความน่าจะเป็นในช่วง 0,1

3.4 การออกแบบ Use Case Diagram



ภาพประกอบที่ 3-11 Use Case Diagram

จาก Use Case Diagram ดังภาพประกอบที่ 3-11 ระบบจะมี 2 หน้าหลัก ได้แก่ ตรวจสอบท่าทางในการออกกำลังกาย โดยผู้ใช้ และ เพิ่ม,ลบ,แก้ไขท่าออกกำลังกายโดย ผู้ดูแลระบบ

ตารางที่ 3.2 Use Case Diagram ตรวจสอบท่าทางในการออกกำลังกาย

Use Case Title: ตรวจสอบท่าทางในการออกกำลังกาย	Use Case ID: 1
Primary Actor: ผู้ปกครอง	
Main Flow: User จะทำการตรวจสอบท่าทางการออกกำลังกายกับผู้ปกครอง	
Exception Flow: ในกรณีที่ไม่สามารถทำงานได้แสดงว่าไม่มีท่าออกกำลังกายในระบบ	

ตารางที่ 3.3 Use Case Diagram ตรวจสอบชื่อท่าออกกำลังกาย

Use Case Title: ตรวจสอบชื่อท่าออกกำลังกาย	Use Case ID: 2
Primary Actor: ผู้ปกครอง	
Main Flow: User จะทำการตรวจสอบชื่อท่าทางการออกกำลังกายกับผู้ปกครอง	
Exception Flow: ในกรณีที่ไม่สามารถทำงานได้แปลว่าข้อมูลท่าทางไม่เพียงพอ	

ตารางที่ 3.4 Use Case Diagram เพิ่มทำออกกำลังกาย

Use Case Title: เพิ่มทำออกกำลังกาย	Use Case ID: 3
Primary Actor: อุปกรณ์	
Main Flow: Admin จะเป็นคนเพิ่มทำต่างๆ	
Exception Flow: ในกรณีที่ไม่สามารถเพิ่มทำได้แปลว่าข้อมูลไม่เพียงพอ	

ตารางที่ 3.5 Use Case Diagram ลบทำออกกำลังกาย

Use Case Title: เพิ่มทำออกกำลังกาย	Use Case ID: 4
Primary Actor: อุปกรณ์	
Main Flow: Admin จะเป็นคนลบทำต่างๆ	
Exception Flow: : ในกรณีที่ไม่สามารถลบได้ระบบจะไม่ทำงานใดๆหรือแสดงค่าใดๆ	

ตารางที่ 3.6 Use Case Diagram แก้ไขทำออกกำลังกาย

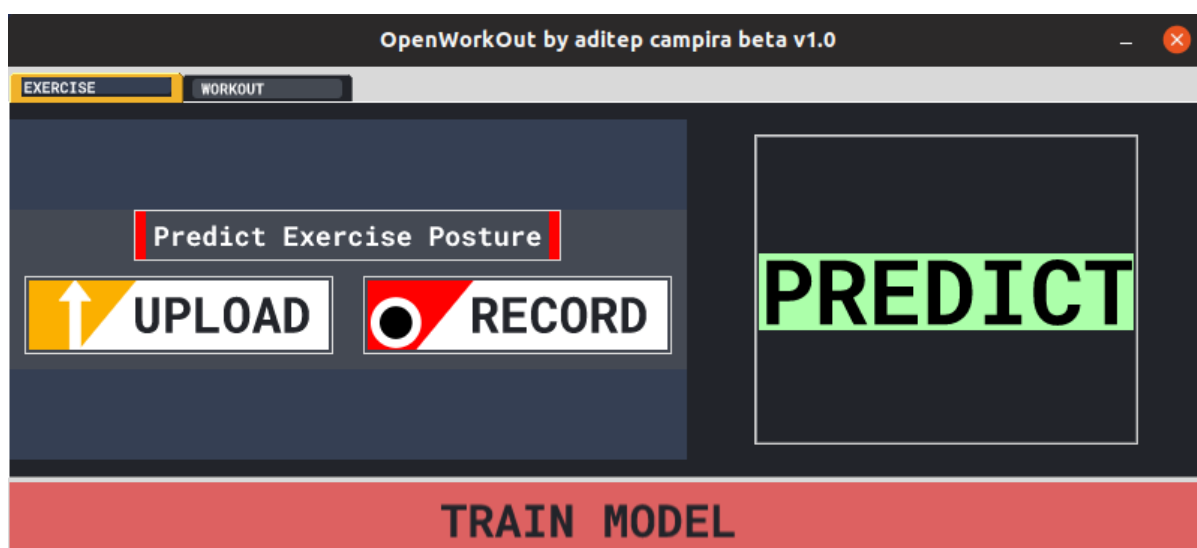
Use Case Title: แก้ไขทำออกกำลังกาย	Use Case ID: 5
Primary Actor: อุปกรณ์	
Main Flow: Admin จะเป็นคนแก้ไขทำต่างๆ	
Exception Flow: : ในกรณีที่ไม่สามารถแก้ไขได้แปลว่าข้อมูลทำทางไม่เพียงพอ	

บทที่ 4

จากการพัฒนาระบบด้านปัญญาประดิษฐ์และการเรียนรู้ของเครื่อง สำหรับช่วยในการออกกำลังกายที่สามารถบอกชื่อท่าออกกำลังกาย และ ตรวจสอบความถูกต้องในการเคลื่อนไหว โดยพัฒนาระบบขึ้นมาโดยแบ่งย่อย ๆ เป็น 3 ส่วนดังนี้

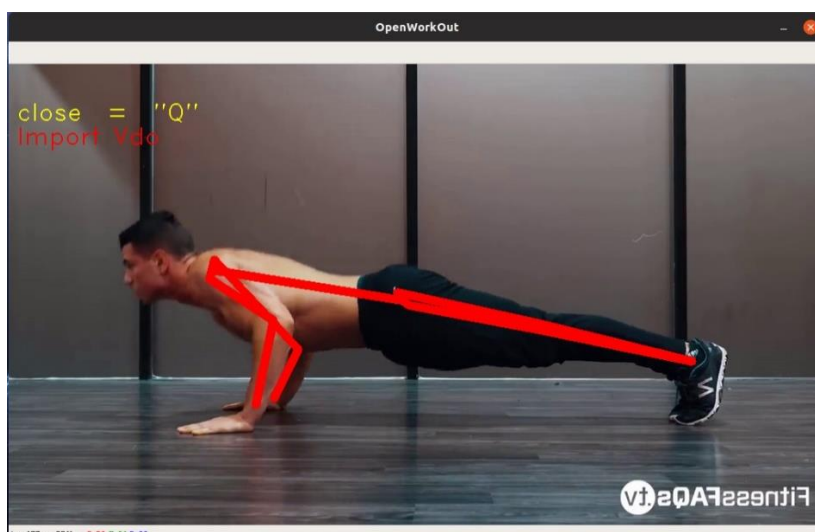
4.1 GUI

4.1.1 การทำนายชื่อท่าออกกำลังกายโดยใช้ไฟล์ Video หรือ Webcam



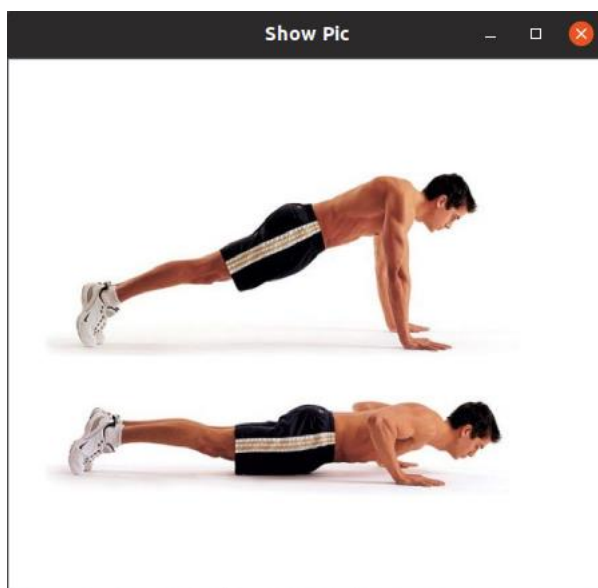
ภาพประกอบที่ 4-1 หน้า การทำนายชื่อท่าออกกำลังกาย

4.1.1.1 การทำนายชื่อท่าออกกำลังกายโดยใช้ไฟล์ Video



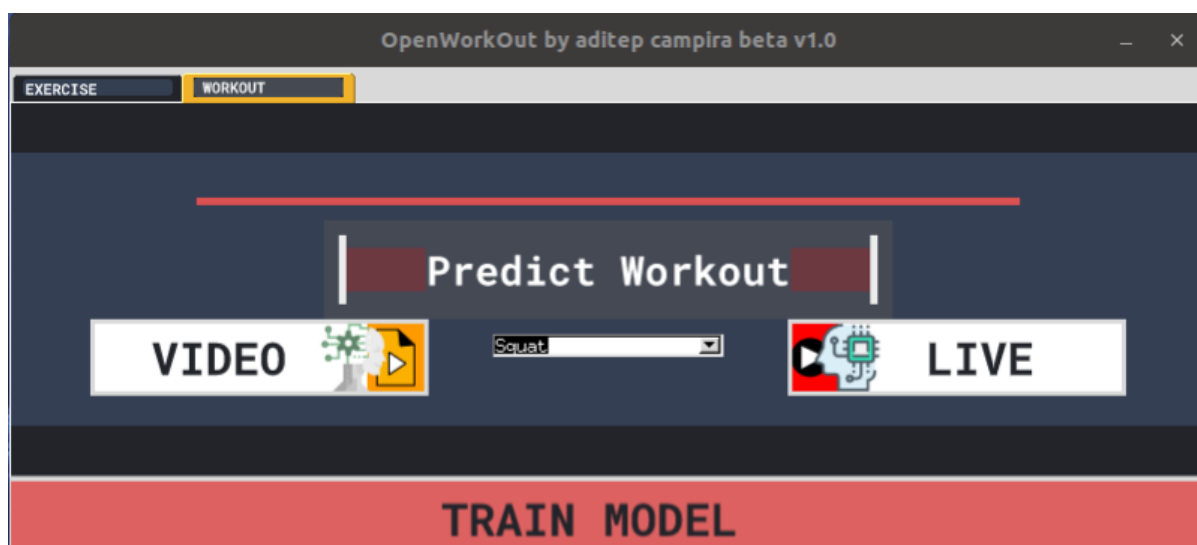
ภาพประกอบที่ 4-2 นำไฟล์ video upload เข้าไปในโปรแกรมเพื่อนทำนาย

4.1.1.2 ผลลัพธ์ที่ได้การนำ file video เข้าไปทำนายชื่อท่าออกกำลังกาย



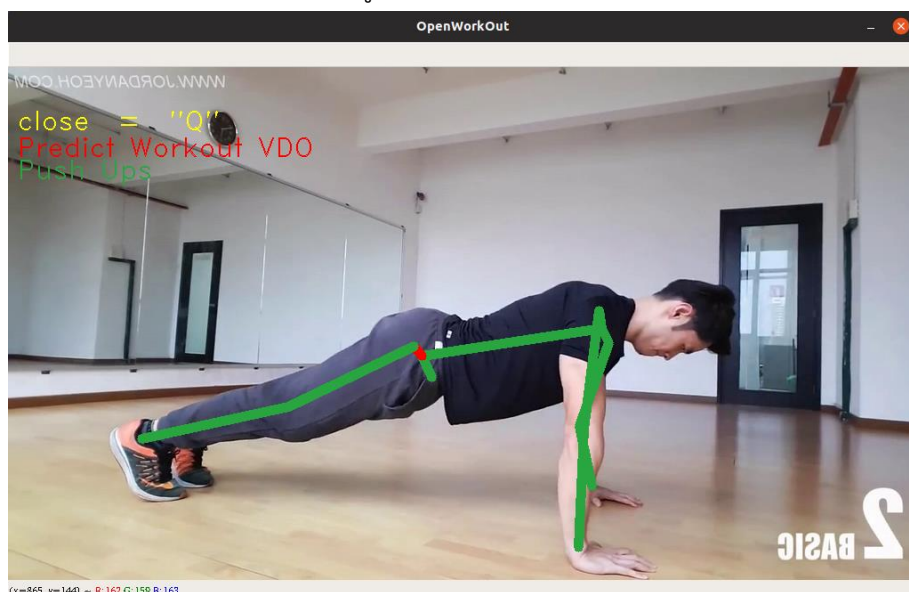
ภาพประกอบที่ 4-3 ผลลัพธ์ที่ได้จะการนำไฟล์เข้าไป ทำนายคือท่า Pushup

4.1.2 ตรวจสอบความถูกต้องในการเคลื่อนไหวในการออกกำลังกาย



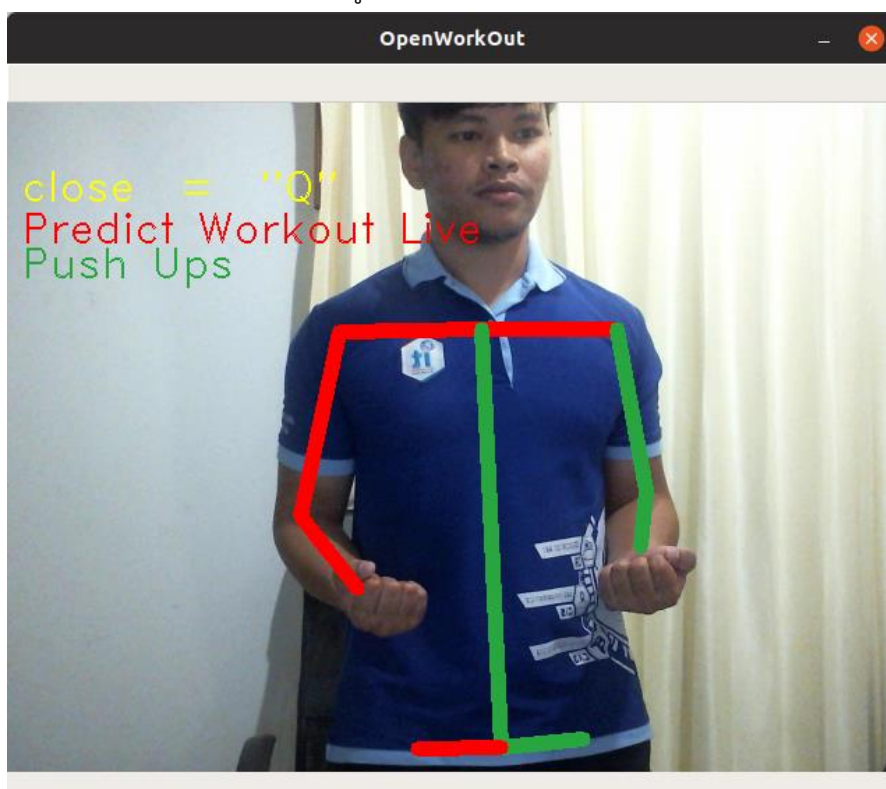
ภาพประกอบที่ 4-4 หน้า ตรวจสอบความถูกต้องในการเคลื่อนไหวในการออกกำลังกาย

4.1.2.1 ตรวจสอบความถูกต้องในการเคลื่อนไหวโดยใช้ไฟล์ Video



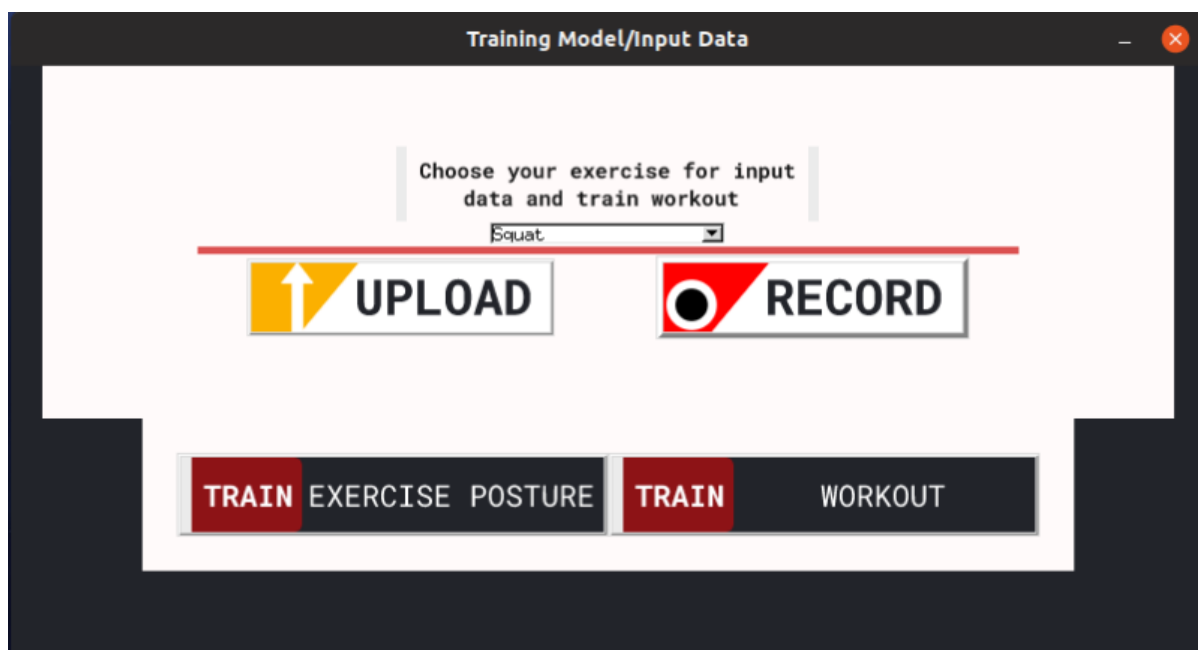
ภาพประกอบที่ 4-5 ตรวจสอบการเคลื่อนไหวในท่า pushup โดยใช้ file video

4.1.2.2 ตรวจสอบความถูกต้องในการเคลื่อนไหวโดยใช้ Webcam



ภาพประกอบที่ 4-6 ตรวจสอบการเคลื่อนไหวในท่า pushup โดยใช้ webcam

4.1.3 การนำเข้าข้อมูลและสร้าง Model ในการทำนายท่าออกกำลังกายและ Model ทำนายความถูกต้องในการเคลื่อนไหวด้วยไฟล์ Video หรือ Webcam



ภาพประกอบที่ 4-3 หน้า การนำเข้าข้อมูลเข้าและ สร้าง model

4.2 การติดตั้งระบบ

4.2.1 ติดตั้ง Ubuntu 18.04

4.2.2 เชื่อมต่ออุปกรณ์ webcam กับ computer

4.2.3 ติดตั้ง CUDA Toolkit 10.1 ลงใน Ubuntu 18.04

4.2.4 ติดตั้ง cuDNN: ver. 7.6.1 ลงใน Ubuntu 18.04

4.2.5 ติดตั้ง CMake GUI ลงใน Ubuntu 18.04

4.2.6 ติดตั้ง python3 ติดตั้ง

4.2.7 ติดตั้ง python3 module (numpy, codecs, json , os , opencv , matplotlib, pickle , scikit-learn , sys , tkinter, PIL)

4.2.8 ติดตั้ง Openpose ลงใน Ubuntu 18.04

4.2.9 ติดตั้ง Project OpenWorkout

4.2.10 เข้าไปใน folder Project เปิด terminal รุม คำสั่ง python3 gui.py

บทที่ 5

ผลการทำงานของระบบเทรนเนอร์อัจฉริยะ มีข้อสรุปและผลการทำงาน ดังนี้

5.1 สรุปผล

ผลการทำงานของระบบเทรนเนอร์อัจฉริยะ การทำนายซื้อและตรวจสอบการเคลื่อนไหวในการออกกำลังกาย เมื่อทำนายท่าที่ใกล้เคียงกันยังไม่แม่นยำ แต่สามารถเพิ่มความแม่นยำได้โดยการนำข้อมูลสอนเพิ่มเติม เพื่อเพิ่มความแม่นยำให้กับ model ได้

5.2 ปัญหาในการดำเนินงาน

ระบบที่พัฒนาเป็นเทคโนโลยีใหม่ ไม่มีตัวอย่างให้ศึกษามากนัก เมื่อเกิดปัญหาจึงต้องคิดและแก้ไขด้วยตนเองเลยใช้เวลาในการพัฒนามากขึ้น

การอ้างอิง

- [1] G-able. (2018). G-able. เรียกใช้เมื่อ 20 กันยายน 2561 จาก G-able: <https://www.g-able.com/thinking/artifact-intelligence/>
- [2] Phyblas.hinaboshi. (18 สิงหาคม 2561). Phyblas.hinaboshi. เรียกใช้เมื่อ 21 กันยายน 2561 จาก Phyblas.hinaboshi: จาก <https://phyblas.hinaboshi.com/panyapradit>
- [3] Vithan Minaphinant. (28 กุมภาพันธ์ 2561). blog.finnomena. เรียกใช้เมื่อ 23 กันยายน 2561 จาก blog.finnomena: <https://blog.finnomena.com/machine-learning-%E0%B8%84%E0%B8%B7%E0%B8%AD%E0%B8%AD%E0%B8%B0%E0%B9%84%E0%B8%A3-fa8bf6663c07>
- [4] Suphan Fayong. (2561). <http://codeonthehill.com>. เรียกใช้เมื่อ 25 พฤศจิกายน 2561 จาก <http://codeonthehill.com>: <http://codeonthehill.com/machine-learning-3-deep-learning/>
- [4] Sanparith Marukatat. (13 กรกฎาคม 2560). medium. เข้าถึงได้จาก medium: <https://medium.com/@sanparithmarukatat/%E0%B8%AA%E0%B8%99%E0%B8%B8%E0%B8%81%E0%B8%81%E0%B8%B1%E0%B8%9A-neural-network-2-11a7194ed236>
- [5] Tomas Simon, Shih-En Wei, Yaser Sheikh Zhe Cao. (2560). Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. Pittsburgh, เพนซิลเวเนีย: Carnegie Mellon University.
- [6] Sven Jens Jorgensen Miguel Arduengo. (2560). ROS Wrapper for Real-Time Multi-Person Pose Estimation with a Single Camera. csic.
- [7] Saixiii. (27 เมษายน 2560). Saixiii. เรียกใช้เมื่อ 2561 กันยายน 20 จาก Saixiii: <https://saixiii.com/python-programming/>
- [8] Daniel Bubnis. (1 มกราคม 2561). healthline. เรียกใช้เมื่อ 26 กันยายน 2561 จาก healthline: <https://www.healthline.com/health/fitness-exercise-weight-training>
- [9] ณัชนนท์ วงษ์วิไล ดนัยนันท์ เก่าเงิน. (2553). Virtual Fitting Room Using Augmented Reality. กรุงเทพฯ: จุฬาลงกรณ์มหาวิทยาลัย.
- [10] Tianhong Li, Mohammad Abu Alsheikh, Yonglong Tian, Hang Zhao, Antonio Torralba, Dina Katabi Mingmin Zhao. (2561). Through-Wall Human Pose Estimation Using Radio Signals. Cambridge: MIT CSAIL.

ประวัติผู้จัดทำโครงการปริญญานิพนธ์



ชื่อ-นามสกุล : นายอดิเทพ คำภีระ
รหัส : 60011270017
ชื่อปริญญานิพนธ์ : เทรนเนอร์อัจฉริยะ
 : The Artificial Intelligence trainer
สาขาวิชา : เทคโนโลยีสารสนเทศ
คณะ : วิทยาการสารสนเทศ

ประวัติส่วนตัว
เกิดวันที่ : 1 ธันวาคม 2538
ที่อยู่ : 93 หมู่ 3 เทศบาลตำบลบ้านเลื่อม อำเภอเมือง จังหวัดอุดรธานี
E-mail : aditep.cpr@gmail.co.th

ประวัติการศึกษา
ประถมศึกษา : โรงเรียนบ้านหมากแข้ง จังหวัดอุดรธานี
มัธยมศึกษาตอนต้น : โรงเรียนประจักษ์ศิลปาคม จังหวัดอุดรธานี
มัธยมศึกษาตอนปลาย : วิทยาลัยเทคนิคอุดรธานี จังหวัดอุดรธานี
ปริญญาตรี : สาขาวิชาเทคโนโลยีสารสนเทศ คณะวิทยาการสารสนเทศ
 มหาวิทยาลัยมหาสารคาม