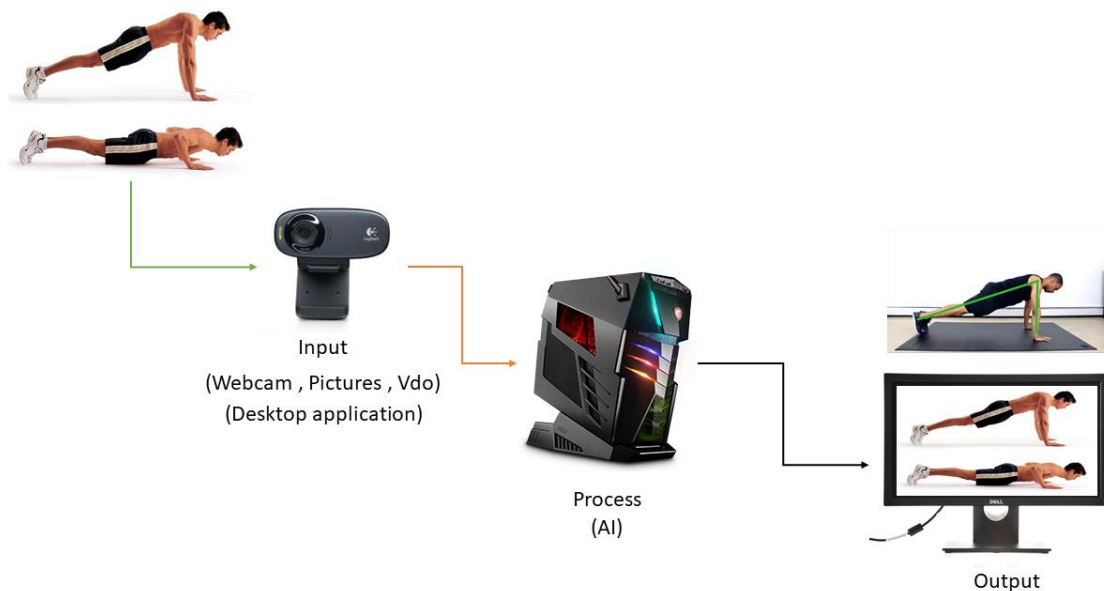


บทที่ 3

ขั้นตอนการดำเนินงาน

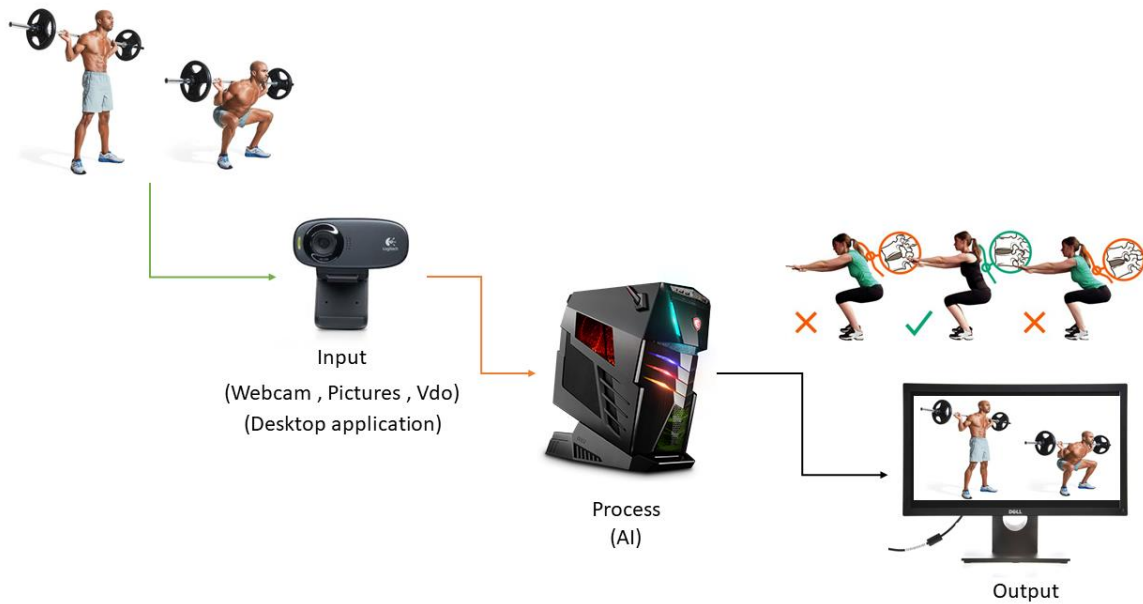
3.1 การออกแบบภาพรวมและหลักการทำงาน

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



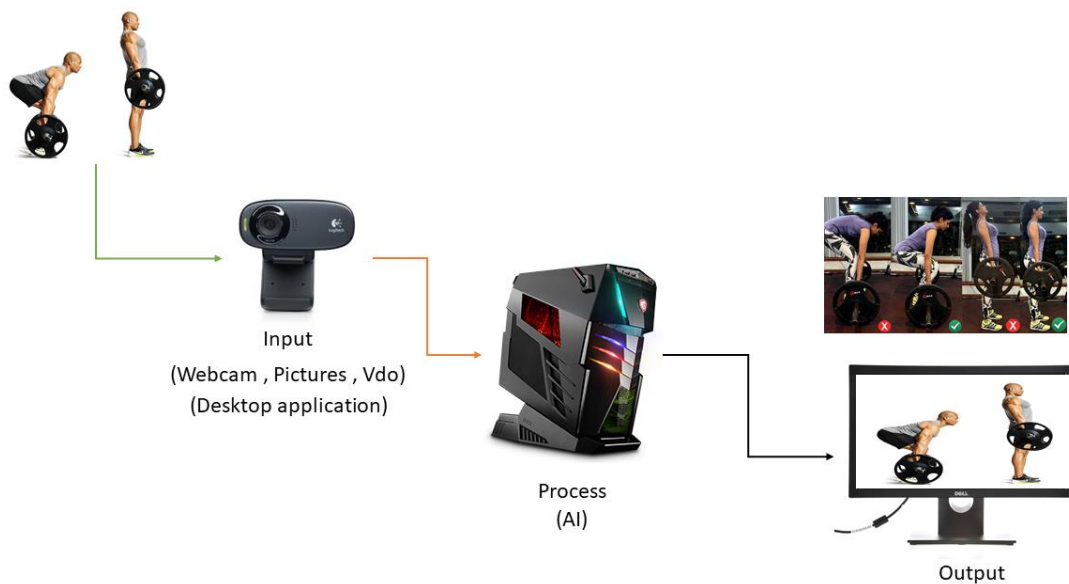
ภาพประกอบที่ 3-1 ภาพการทำงานการตรวจสอบท่า Push ups

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



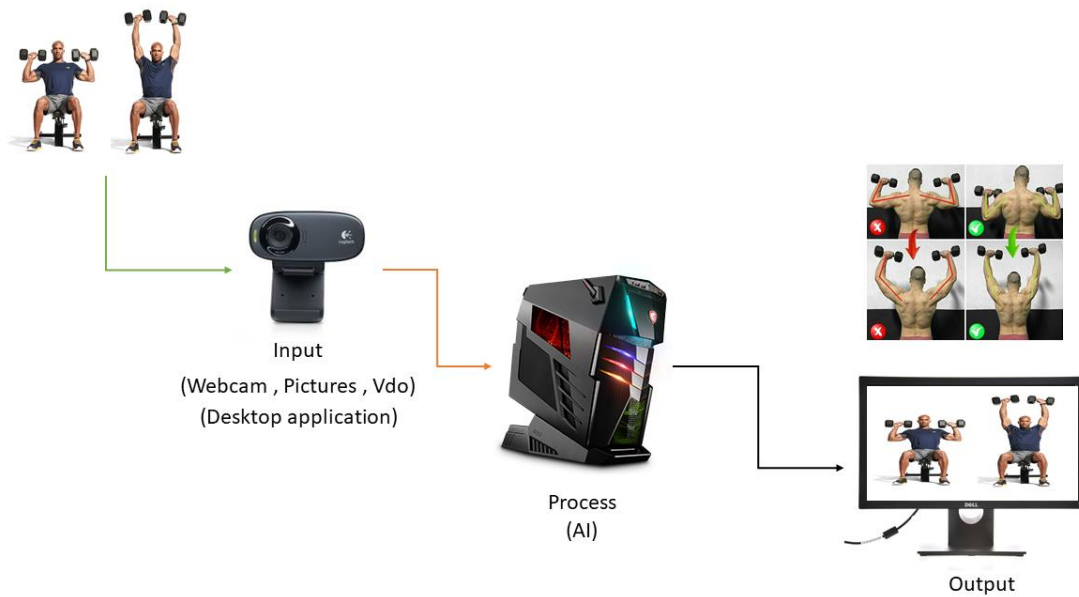
ภาพประกอบที่ 3-2 ภาพการทำงานการตรวจสอบท่า Squat

จากภาพประกอบที่ 3-2 การตรวจสอบท่า Squat จะตรวจสอบจากมุมด้านข้างโดยจะตรวจสอบจากลักษณะ ลำตัวส่วนบน คอ และ ขา



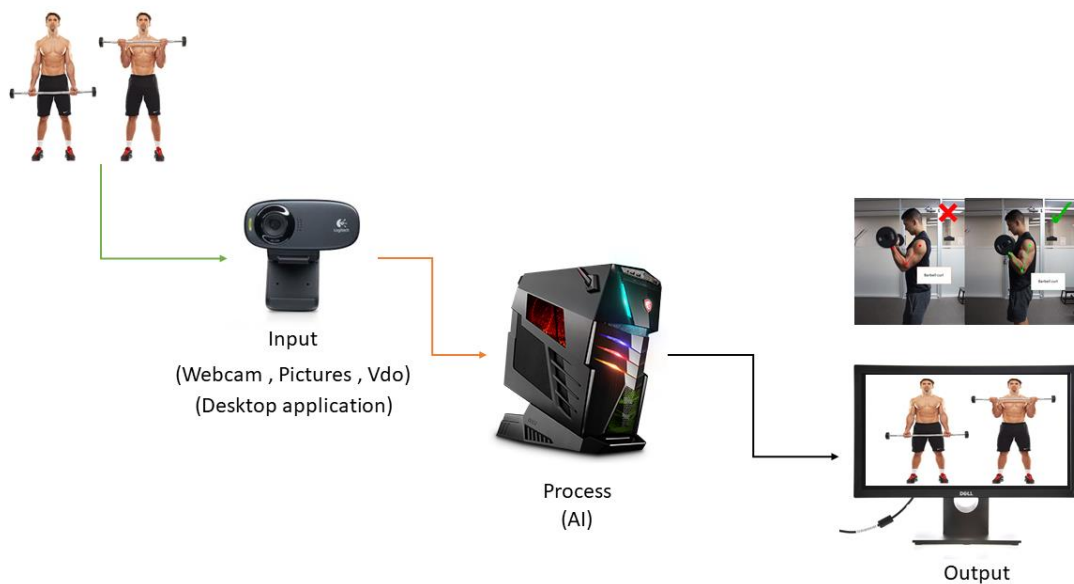
ภาพประกอบที่ 3-3 ภาพการทำงานการตรวจสอบท่า Deadlift

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



ภาพประกอบที่ 3-4 ภาพการทำงานการตรวจสอบท่า Dumbbell Shoulder Press

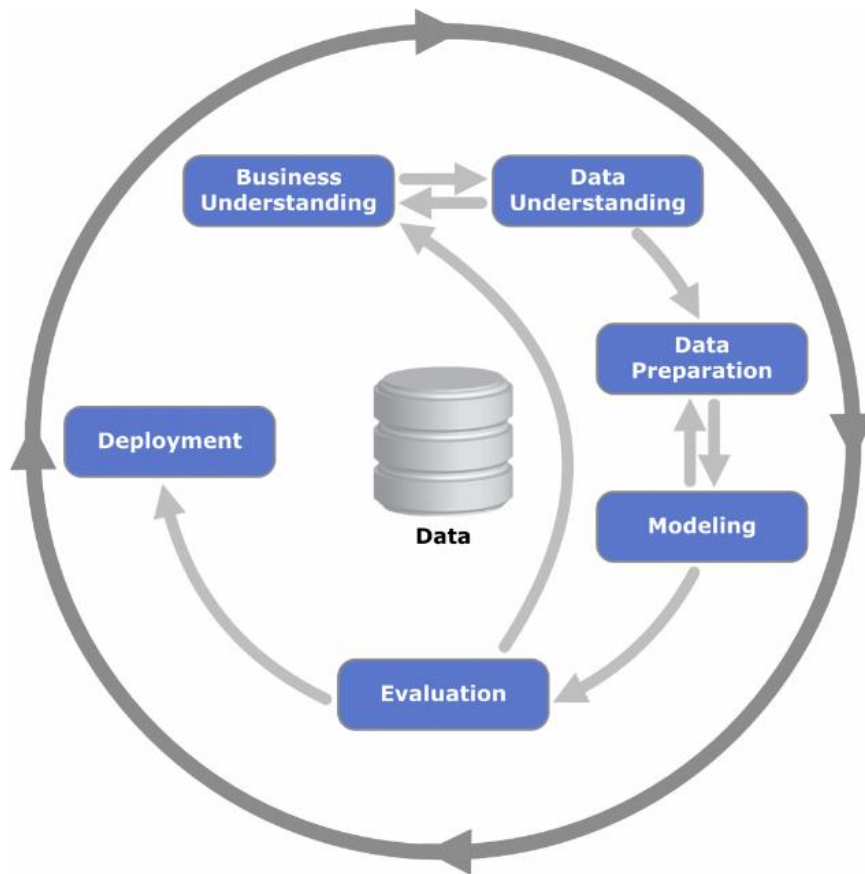
จากภาพประกอบที่ 3-4 การตรวจสอบท่า Dumbbell Shoulder Press จะตรวจสอบจากมุมด้านหน้า โดยจะตรวจสอบจาก ลักษณะ ของ ท่อนแขนบน และ ท่อนแขนล่าง



ภาพประกอบที่ 3-5 ภาพการทำงานการตรวจสอบท่า Barbell Curl

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

3.2 กระบวนการวิเคราะห์ข้อมูลด้วย CRISP-DM



ภาพประกอบที่ 3-6 กระบวนการวิเคราะห์ข้อมูลด้วย CRISP-DM

3.2.1 Business Understanding

วิเคราะห์ข้อมูลทางตลาด 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



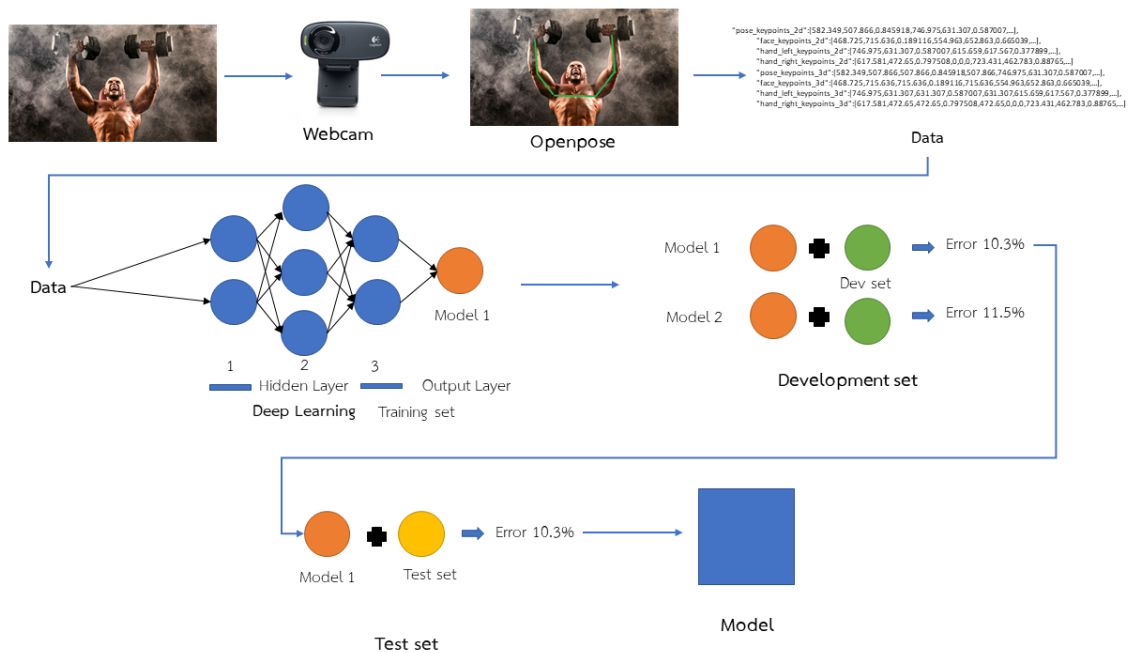
จากภาพประกอบที่ 3-7 แสดงการแบ่งชุดข้อมูล

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

3.3.2.1 Development set คือขั้นตอนในการเลือก Model ตัวใดที่มีค่าความผิดพลาดน้อยที่สุดที่ได้มาจาก Training set

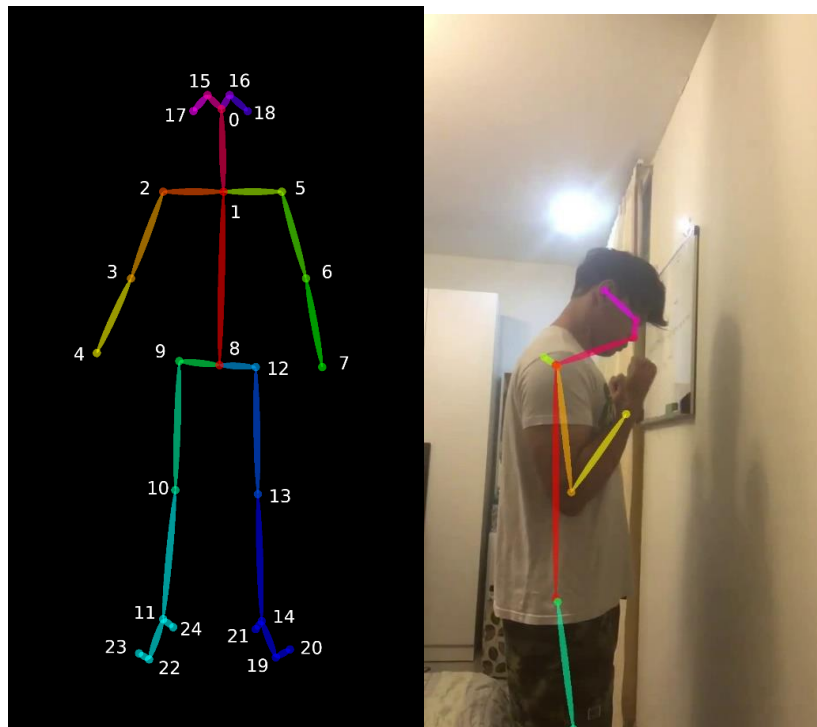
3.3.2.2 Test set คือเมื่อเราได้ Model มาแล้วจะนำมาวัด unbiased error ที่แท้จริงโดยจะใช้ข้อมูลในการทดสอบคนละชุดกับ Development set

3.2.6 Deployment

นำ Model ที่สร้างขึ้นมาไปใช้งานจริง โดยจะใช้วิธีบันทึกเก็บเป็นออบเจกต์ จะใช้มอดูลที่ชื่อ Pickle ที่เป็นมอดูลมาตรฐานของ 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 แสดงข้อมูลที่ได้จาก openpose ที่ได้ผ่านขั้นตอน Data Preparation โดยข้อมูลจะประกอบไปด้วยตัวแปรแกน x , y และ c ที่แสดงถึงความน่าจะเป็นในช่วง 0,1

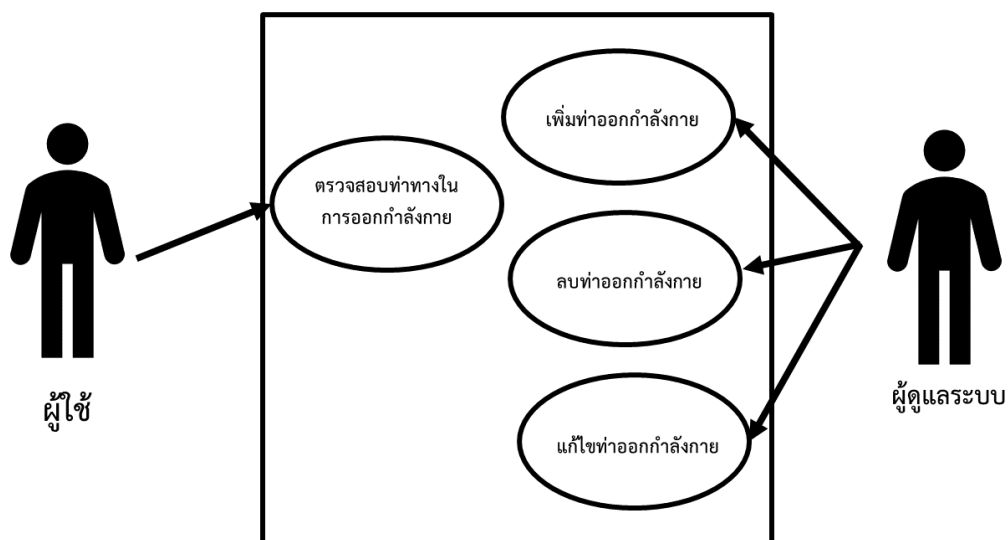
ตารางที่ 3.1 ตัวอย่างชุดข้อมูล

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

Pose Output	Keypoints 1	Keypoints 2	Keypoints 3	Keypoints 4	Keypoints 5	Keypoints 6
y7, "LWrist"	1047.6	1047.6	1058.1	1058.05	1058	1061.49
c7, "LWrist"	0.603187	0.678407	0.575413	0.60329	0.663122	0.639352
x8, "MidHip"	237.81	234.234	248.277	244.874	244.867	241.355
y8, "MidHip"	1194.16	1173.23	1173.25	1159.26	1141.83	1162.77
c8, "MidHip"	0.245804	0.271155	0.195336	0.254594	0.249424	0.277674
x9, "RHip"	185.453	182.101	199.431	202.863	202.935	209.879
y9, "RHip"	1201.14	1176.73	1187.2	1180.19	1159.29	1173.24
c9, "RHip"	0.284233	0.318744	0.230284	0.285912	0.282045	0.376008
x10, "RKnee"	0	0	0	0	0	0
y10, "RKnee"	0	0	0	0	0	0
c10, "RKnee"	0	0	0	0	0	0
x11, "RAnkle"	0	0	0	0	0	0
y11, "RAnkle"	0	0	0	0	0	0
c11, "RAnkle"	0	0	0	0	0	0
x12, "LHip"	300.649	283.226	311.134	311.122	307.644	290.161
y12, "LHip"	1183.73	1162.76	1166.26	1134.86	1124.36	1152.29
c12, "LHip"	0.192725	0.198357	0.160593	0.203428	0.197593	0.203159
x13, "LKnee"	0	0	0	0	0	0
y13, "LKnee"	0	0	0	0	0	0
c13, "LKnee"	0	0	0	0	0	0
x14, "LAnkle"	0	0	0	0	0	0
y14, "LAnkle"	0	0	0	0	0	0
c14, "LAnkle"	0	0	0	0	0	0
x15, "REye"	485.76	489.092	485.689	482.027	461.32	457.714
y15, "REye"	398.449	422.748	436.873	464.828	489.308	509.964
c15, "REye"	0.812815	0.811481	0.822228	0.789121	0.824947	0.801429
x16, "LEye"	513.575	0	0	0	0	0

Pose Output	Keypoints 1	Keypoints 2	Keypoints 3	Keypoints 4	Keypoints 5	Keypoints 6
y16, "LEye"	402.014	0	0	0	0	0
c16, "LEye"	0.0782677	0	0	0	0	0
x17, "REar"	377.472	394.905	398.365	401.996	401.891	398.425
y17, "REar"	380.938	398.4	405.454	422.866	440.341	457.746
c17, "REar"	0.662203	0.739947	0.829464	0.86402	0.870627	0.892092
x18, "LEar"	0	0	0	0	0	0
y18, "LEar"	0	0	0	0	0	0
c18, "LEar"	0	0	0	0	0	0
x19, "LBigToe"	0	0	0	0	0	0
y19, "LBigToe"	0	0	0	0	0	0
c19, "LBigToe"	0	0	0	0	0	0
x20, "LSmallToe"	0	0	0	0	0	0
y20, "LSmallToe"	0	0	0	0	0	0
c20, "LSmallToe"	0	0	0	0	0	0
x21, "LHeel"	0	0	0	0	0	0
y21, "LHeel"	0	0	0	0	0	0
c21, "LHeel"	0	0	0	0	0	0
x22, "RBigToe"	0	0	0	0	0	0
y22, "RBigToe"	0	0	0	0	0	0
c22, "RBigToe"	0	0	0	0	0	0
x23, "RSmallToe"	0	0	0	0	0	0
y23, "RSmallToe"	0	0	0	0	0	0
c23, "RSmallToe"	0	0	0	0	0	0
x24, "RHeel"	0	0	0	0	0	0
y24, "RHeel"	0	0	0	0	0	0
c24, "RHeel"	0	0	0	0	0	0

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: อุปกรณ์	
Main Flow: User จะทำการตรวจสอบท่าทางการออกกำลังกายกับอุปกรณ์	
Exception Flow: ในกรณีที่ไม่สามารถทำงานได้แสดงว่าไม่มีท่าออกกำลังกายในระบบ	

ตารางที่ 3.3 Use Case Diagram เพิ่มท่าออกกำลังกาย

Use Case Title: เพิ่มท่าออกกำลังกาย	Use Case ID: 2
Primary Actor: อุปกรณ์	
Main Flow: Admin จะเป็นคนเพิ่มท่าต่างๆ	
Exception Flow: ในกรณีที่ไม่สามารถเพิ่มทำได้แปลว่าข้อมูลไม่เพียงพอ	

ตารางที่ 3.4 Use Case Diagram ลบทำออกกำลังกาย

Use Case Title: เพิ่มทำออกกำลังกาย	Use Case ID: 3
Primary Actor: อุปกรณ์	
Main Flow: Admin จะเป็นคนลบทำต่างๆ	
Exception Flow: : ในกรณีที่ไม่สามารถลบได้ระบบจะไม่ทำงานใดๆหรือแสดงค่าใดๆ	

ตารางที่ 3.5 Use Case Diagram แก้ไขทำออกกำลังกาย

Use Case Title: แก้ไขทำออกกำลังกาย	Use Case ID: 4
Primary Actor: อุปกรณ์	
Main Flow: Admin จะเป็นคนแก้ไขทำต่างๆ	
Exception Flow: : ในกรณีที่ไม่สามารถแก้ไขได้แปลว่าข้อมูลทำทางไม่เพียงพอ	

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 วงษ์วีไล 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.