



EGCO486 : Image Processing

Final Project

จัดทำโดย

นายจารุภัทร โชติสีตานันท์	6513161
---------------------------	---------

นางสาวชลีชา บัวทอง	6513163
--------------------	---------

นายณพรุจ ฤทธิเนติกุล	6513168
----------------------	---------

เสนอ

ผศ.ดร. นริศ หนูหอม

คณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยมหิดล

ภาคเรียนที่ 1 ปีการศึกษา 2567

คำนำ

รายงานฉบับนี้จัดทำขึ้นเพื่อเป็นส่วนหนึ่งของรายวิชา Image Processing (EGCO486) โดยคณะผู้จัดทำ ได้จัดทำขึ้นเพื่อใช้ประกอบการทำโครงงานเรื่อง Exercise Pose Correction and Classification ซึ่งจัดทำขึ้นเพื่อศึกษาและพัฒนาโปรแกรมที่สามารถช่วยผู้ใช้งานใน การระบุข้อผิดพลาดของท่าออกกำลังกายในแต่ละประเภท พร้อมทั้งให้คำแนะนำเพื่อปรับปรุงท่าทาง และสามารถจำแนกประเภทท่าออกกำลังกายในแต่ละประเภทได้อย่างแม่นยำ โดยมีเป้าหมายเพื่อช่วยลดความเสี่ยงจากการบาดเจ็บ และเพิ่มประสิทธิภาพในการออกกำลังกายให้ดีขึ้น

ทางคณะผู้จัดทำขอขอบพระคุณ ผศ.ดร. นริศ หนูหอม ผู้ให้ความรู้ และแนวทางการศึกษา สุดท้ายนี้ทางคณะผู้จัดทำหวังว่ารายงานฉบับนี้จะสามารถเป็นประโยชน์ไม่มากนักน้อยแก่ผู้อ่านทุกท่าน หากมีข้อผิดพลาดประการใด ผู้จัดทำขอน้อมรับไว้ และขออภัยมา ณ ที่นี้

ขอขอบพระคุณ

คณะผู้จัดทำ

สารบัญ

หัวข้อ	หน้า
ที่มาและความสำคัญของโครงการ	1
วัตถุประสงค์ของโครงการ	2
ขอบเขตของโครงการ	3
วิธีการดำเนินงาน	5
ข้อจำกัดของโปรแกรม	17
ผลการทดลอง	18
บทสรุปของโครงการ	21
การต่อยอดในอนาคต	22

ที่มาและความสำคัญของโครงการ (Background)

ในปัจจุบัน การออกกำลังกายเป็นที่นิยมอย่างแพร่หลาย เนื่องจากผู้คนให้ความสำคัญกับสุขภาพ และ การพัฒนาสมรรถภาพร่างกายมากขึ้น อย่างไรก็ตาม การออกกำลังกายที่มีท่าทางที่ไม่ถูกต้องอาจก่อให้เกิดอาการบาดเจ็บหรือไม่ได้ผลลัพธ์ตามที่คาดหวัง โดยเฉพาะอย่างยิ่งการออกกำลังกายด้วยน้ำหนัก (Weight Training) ซึ่งต้องอาศัยการจัดท่าทางที่ถูกต้องเพื่อความปลอดภัยและประสิทธิภาพที่ดียิ่งขึ้น

โครงการนี้มีเป้าหมายในการพัฒนาโปรแกรมที่สามารถตรวจจับข้อผิดพลาดในท่าออกกำลังกาย เพื่อให้คำแนะนำที่ช่วยผู้ใช้งานปรับปรุงเทคนิคและการจัดท่าทางให้ถูกต้อง พร้อมทั้งสามารถวิเคราะห์และจำแนกประเภทท่าออกกำลังกายได้อย่างแม่นยำ โดยระบบนี้ถูกออกแบบให้รองรับการใช้งานผ่านวิดีโอ ทำให้สามารถช่วยผู้ใช้งานได้แม้ไม่มีผู้ฝึกสอนส่วนตัว

ระบบดังกล่าวไม่เพียงช่วยลดความเสี่ยงในการบาดเจ็บ แต่ยังช่วยเพิ่มประสิทธิภาพในการออกกำลังกาย และสนับสนุนการเรียนรู้ของผู้ใช้งานให้มีความเข้าใจในรูปแบบการออกกำลังกายที่ถูกต้องและปลอดภัย

วัตถุประสงค์ของโครงการ (Objective)

1. เพื่อตรวจจับข้อผิดพลาดในแต่ละท่าออกกำลังกาย พร้อมให้คำแนะนำในการปรับปรุงเทคนิคและการจัดท่าให้ถูกต้อง ซึ่งครอบคลุมท่าออกกำลังกายทั้งหมด 5 ท่า ได้แก่ Squat, Bicep Curl, Incline Dumbbell Press, Dumbbell Row, และ Lat Pulldown
2. เพื่อจำแนกประเภทท่าออกกำลังกายในแต่ละประเภทได้อย่างแม่นยำ
3. เพื่อสนับสนุนและส่งเสริมสุขภาพด้วยการออกกำลังกายด้วยท่าที่ปลอดภัย และถูกต้อง
4. เพื่อช่วยให้ผู้ใช้งานสามารถออกกำลังกายได้ด้วยตนเอง โดยไม่ต้องมีเทรนเนอร์มาคอยกำกับ

ขอบเขตของโครงการ (Scope of Work)

โครงการนี้มุ่งพัฒนาโปรแกรมที่สามารถให้คำแนะนำในการปรับปรุงเทคนิคและการจัดท่าให้ถูกต้อง และ จำแนกท่าทางการออกกำลังกาย โดยมีรายละเอียดดังนี้

- **การตรวจจับ Skeleton**

ใช้ **MediaPipe** เพื่อตรวจจับจุดสำคัญบนร่างกาย เช่น ข้อศอก หัวไหล่ เข่า และสะโพก ในวิดีโอที่ผู้ใช้งานอัปโหลดเข้าสู่ระบบ

- **การจำแนกประเภทของท่าออกกำลังกาย**

ใช้ **Deep Learning Model (LSTM : Long Short-Term Memory)** ในการวิเคราะห์ลำดับข้อมูลจาก Keypoints เพื่อจำแนกประเภทท่าออกกำลังกายที่รองรับทั้งหมด 5 ท่า ได้แก่

1. Squat
2. Incline Dumbbell Press
3. Lat Pulldown
4. Dumbbell Row
5. Bicep Curl

- **การให้คำแนะนำในการปรับปรุงท่าทาง**

ระบบจะวิเคราะห์จุดสำคัญและค่านวณมุมของข้อต่อ เพื่อตรวจหาข้อผิดพลาดในแต่ละท่า เช่น ข้อมืออยู่ห่างกันเกินไป, ข้อศอกขยับ พร้อมแนะนำวิธีแก้ไขท่าเพื่อป้องกันการบาดเจ็บและเพิ่มประสิทธิภาพของการออกกำลังกาย

- **การพัฒนา Web Application** : พัฒนาด้วย Python และ JavaScript
 - **Streamlit** : เพื่อสร้าง Web Application
 - **React** : สำหรับตกแต่ง User Interface ให้ดูทันสมัยและใช้งานง่าย

ซึ่งผู้ใช้สามารถอัปโหลดวิดีโอเพื่อตรวจสอบและรับคำแนะนำแต่ละท่า
ได้ผ่าน คอมพิวเตอร์ หรือ มือถือ โดยไม่ต้องติดตั้งซอฟต์แวร์เพิ่มเติม

- **กลุ่มเป้าหมาย**

โปรแกรมนี้ออกแบบมาสำหรับผู้ใช้งานทั่วไปที่สนใจในด้านการออกกำลังกาย และต้องการคำแนะนำในการพัฒนาท่าทางอย่างถูกต้องและปลอดภัย โดยไม่จำเป็นต้องมีเทรนเนอร์ส่วนตัว

วิธีการดำเนินงาน (Method)

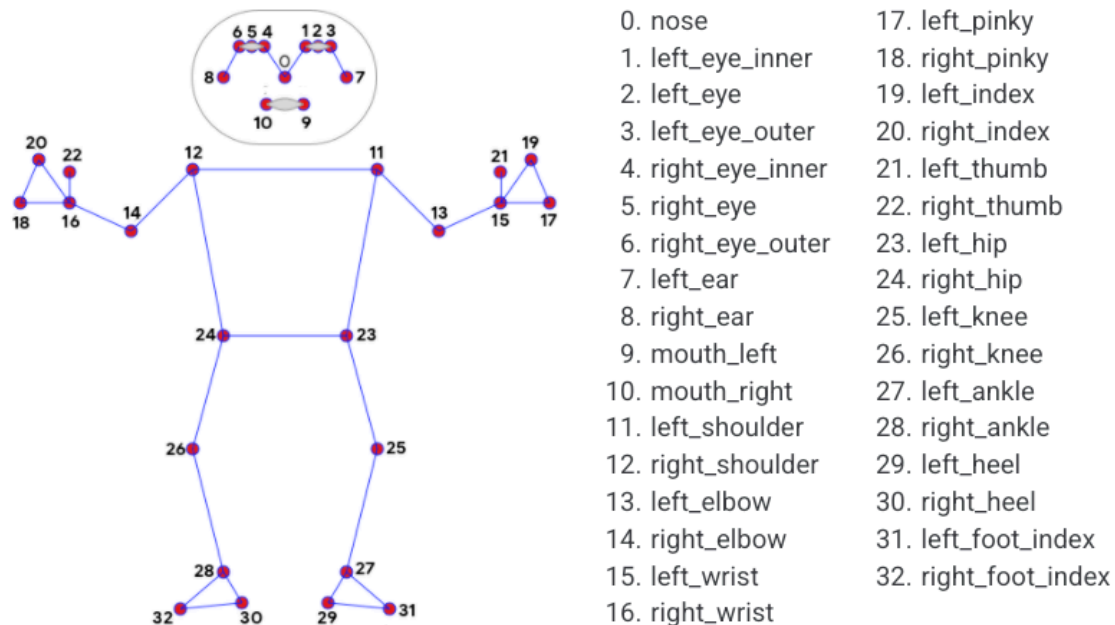
การประมวลผลเบื้องต้น (Pre-processing)

1. การเพิ่มข้อมูลด้วย Data Augmentation
 - เพิ่มความหลากหลายของข้อมูล โดยวิดีโอจะถูก Flip ซ้าย-ขวา เพื่อเพิ่มความสามารถในการเรียนรู้ของโมเดล
2. ใช้ **MediaPipe** ในการวาด Skeleton บนวิดีโอเพื่อเตรียมข้อมูลสำหรับการ Train โมเดล
 - นำวิดีโอที่ต้องการเข้าสู่ระบบ **MediaPipe** เพื่อสร้าง Skeleton
 - ตรวจสอบตำแหน่งของ Skeleton ว่าอยู่บนตัวบุคคลและมีความถูกต้องหรือไม่ (เช่น Skeleton ไม่ควรวาดบนวัตถุที่ไม่ใช่คน หรือมีตำแหน่งผิดเพี้ยนอย่างมาก)
 - หากพบว่าวิดีโอดังกล่าวมีข้อผิดพลาด จะลบวิดีโอนั้นทิ้งเพื่อป้องกันข้อมูลที่ไม่เหมาะสมเข้าสู่กระบวนการ train โมเดล

เหตุผลที่เลือกใช้ MediaPipe

MediaPipe เป็นเครื่องมือ **Open-Source Framework** สำหรับการประมวลผลข้อมูลภาพและวิดีโอแบบเรียลไทม์ โดยมีความสามารถในการสร้าง **Landmark** ได้ทั้งหมด 32 จุดในระบบ 3D (ดังแสดงในรูป 3.1) ซึ่งจุด Landmark สำคัญที่ MediaPipe มีและอาจไม่มีในโมเดล Skeleton อื่น ๆ ได้แก่ นิ้วเท้า (foot_index) และ นิ้วมือ (index) ซึ่งสำคัญในการตรวจสอบการวางตำแหน่งมือและเท้าที่ถูกต้องในท่าออกกำลังกายบางท่า โดยเฉพาะท่าที่เกี่ยวข้องกับการใช้ข้อมือในบริหารร่างกายส่วนบน หรือท่าที่ต้องใช้มือในการจับอุปกรณ์ จุดเหล่านี้จึงมีความสำคัญในการพิจารณาท่าทางที่ถูกต้อง

อย่างไรก็ตาม ในโครงงานนี้ยังไม่มี การคำนวณมุมของข้อมือ เนื่องจากข้อจำกัดด้านความแม่นยำในการระบุตำแหน่ง Landmark ของ MediaPipe โดยเฉพาะในกรณีที่ข้อมืออยู่ในมุมที่ยากต่อการระบุ เช่น การหมุนหรือการเอียงของข้อมือ แต่ MediaPipe ยังสามารถให้ข้อมูลที่เพียงพอสำหรับการตรวจสอบท่าทางอื่นๆ เช่น การวางขา, ข้อศอก, และหัวไหล่ที่สามารถใช้ในการประเมินท่าทางได้



รูป 3.1

MediaPipe ถูกใช้ในสองส่วนหลักของโครงงาน คือ

1. การสกัด Features เพื่อใช้ฝึกโมเดล

- คำนวณมุมในระบบ 2D จาก Landmark Points ที่ MediaPipe สร้างขึ้นทั้งหมด 8 มุม ได้แก่

ข้างขวา: right_shoulder_angle, right_elbow_angle, right_hip_angle, right_knee_angle

ข้างซ้าย: left_shoulder_angle, left_elbow_angle, left_hip_angle, left_knee_angle

- การคำนวณมุมจะดำเนินการในแต่ละเฟรมของวิดีโอ และบันทึกเป็นชุดข้อมูลในรูปแบบไฟล์ CSV โดยแต่ละ Row แทนแต่ละเฟรม และแต่ละ Column แทนมุมทั้ง 8

ฟังก์ชันที่ใช้คำนวณมุมในระบบ 2D :

- ใช้คำนวณมุม 2 มิติ ระหว่างจุดสามจุด โดยจุดที่รับมาจะเป็นพิกัด (x, y, z) เช่น ตำแหน่งของข้อต่อต่างๆ ในร่างกายที่รับมาจาก Skeleton (Landmark) ของ MediaPipe ตรวจจับได้

```
def calculate_angle_2d(a, b, c):
    a = np.array(a)
    b = np.array(b)
    c = np.array(c)

    radians = np.arctan2(c[1] - b[1], c[0] - b[0]) - np.arctan2(a[1] - b[1], a[0] - b[0])
    angle = np.abs(radians * 180.0 / np.pi)

    if angle > 180.0:
        angle = 360.0 - angle

    return angle
```

ตัวอย่างการคำนวณ :

```
right_shoulder_angle = calculate_angle_2d(right_hip_x, right_shoulder_x, right_elbow_x)
right_elbow_angle = calculate_angle_2d(right_shoulder_x, right_elbow_x, right_wrist_x)
right_hip_angle = calculate_angle_2d(right_knee_x, right_hip_x, right_shoulder_x)
right_knee_angle = calculate_angle_2d(right_hip_x, right_knee_x, right_ankle_x)
```

2. การคำนวณความถูกต้องของท่าออกกำลังกาย

- โปรแกรมจะคำนวณมุมทั้ง 8 มุม และใช้โมเดลทำนายว่าเป็นท่าออกกำลังกายประเภทใด
- การตรวจสอบความถูกต้องของท่าแต่ละประเภทจะมีการคำนวณที่แตกต่างกัน เช่น ระยะห่าง, มุมในระบบ 2D และ 3D หรือเปรียบเทียบตำแหน่งในแกน X และ Y

ตัวอย่างเงื่อนไขในแต่ละท่า

1. Dumbbell Row

- ใช้ตำแหน่งแกน Y เปรียบเทียบ : $\text{left_elbow.y} * 1.05 < \text{back_mid}[1]$ ถ้าเงื่อนไขนี้เป็นจริงแสดงว่าคนในวิดีโอขึ้นตัวมาสูงเกินไปและจะแสดงข้อความว่า Elbow too high และมีการใช้การเปรียบเทียบมุมในระบบ 2 มิติกับแขนข้างที่ไม่ได้ยกน้ำหนัก ถ้ามุมของข้อศอกน้อยกว่า 130 องศา ถือว่าแขนงอเกินไปจะแสดงข้อความว่า keep the support arm straight

```
if left_elbow.y*1.05 < back_mid[1]:
    set_error("elbow_hi",7,["elbow_hi","arm_st"])
    cv2.putText(frame, f'Elbow too high', (10, 260), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)

if rightangle2 < np.degrees(2.26892803) : # 130 degree
    set_error("arm_st",7,["elbow_hi","arm_st"])
    cv2.putText(frame, f'keep the support arm straight', (10, 290), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
```

2. Bicep Curl

- ใช้ตำแหน่งแกน X เปรียบเทียบและระยะห่างในระบบ 3 มิติ : $\text{elbow_x} > \text{left_elbow.x}$ โดย elbow_x คือค่า left_elbow.x เมื่อ 5 frame ที่แล้ว ถ้าค่า x เปลี่ยนไปเกิน $\text{sho_hip_dis} * 0.1$ (ระยะห่างของสะโพกกับไหล่) แสดงว่าเกินการขยับข้อศอกมากเกินไป จะแสดงข้อความว่า FIX Elbow

```
if frame_count > 5:
    sho_hip_dis = calculate_distance(
        [left_hip.x, left_hip.y, left_hip.z],
        [left_shoulder.x, left_shoulder.y, left_shoulder.z] )
    if elbow_x > left_elbow.x:
        if abs(elbow_x-left_elbow.x) > sho_hip_dis*0.1 :
            set_error("elbow_fi",15)
            cv2.putText(frame, f'FIX Elbow', (10, 260), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
        else :
            if abs(left_elbow.x-elbow_x) > sho_hip_dis*0.1 :
                set_error("elbow_fi",15)
                cv2.putText(frame, f'FIX Elbow', (10, 260), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
    if frame_count % 5 == 0:
        elbow_x = left_elbow.x
```

3. Squat

- ใช้การคำนวณมุม ระยะห่างในระบบ 3 มิติ และมุมในระบบ 2 มิติ : เปรียบเทียบ `squat_angle > np.degrees(2.7925268) #160 degree` เงื่อนไขนี้คือตอนที่กำลังยืน (คำนวณได้แม่นยำเฉพาะตอนยืน) เปรียบเทียบ `ankle_distance > (shoulder_distance * 1.43)` ถ้าเงื่อนไขเป็นจริงแสดงว่าท่าทางมีการวางเท้าที่กว้างเกินไป จะแสดงข้อความว่า `adjust your ankle closer` และในทางตรงกันข้ามแสดงว่าท่าทางมีการวางเท้าแคบเกินไป จะแสดงข้อความว่า `adjust your ankle wider`
- เปรียบเทียบ `right_knee.x < right_foot_index.x` และ `right_knee.x + (squat_stand) < right_foot_index.x` เป็นจริง แสดงว่าตำแหน่งของเข่านั้นเลยนิ้วเท้าไป จะแสดงข้อความว่า `false knee position`

```
squat_angle = calculate_angle(  
    a: [right_hip.x, right_hip.y, right_hip.z],  
    b: [right_knee.x, right_knee.y, right_knee.z],  
    c: [right_ankle.x, right_ankle.y, right_ankle.z]  
)  
squat_angle2 = calculate_angle_2d(  
    a: [right_hip.x, right_hip.y],  
    b: [right_knee.x, right_knee.y],  
    c: [right_ankle.x, right_ankle.y]  
)
```

```
if squat_angle > np.degrees(2.7925268 ) :#160 degree  
    if squat_angle2 > np.degrees(2.7925268 ):  
        if right_knee.x < right_foot_index.x :  
            squat_stand = abs(right_foot_index.x-right_knee.x)*1.3  
        else:  
            squat_stand = abs(right_knee.x-right_foot_index.x)*1.3  
  
        if ankle_distance > (shoulder_distance * 1.43) :  
            set_error("ankle_cl",7,["ankle_cl","knee_fa","ankle_wi"])  
            cv2.putText(frame, f'adjust your ankle closer', (10, 260), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)  
  
        if ankle_distance < (shoulder_distance * 0.8):  
            set_error("ankle_wi",7,["ankle_cl","knee_fa","ankle_wi"])  
            cv2.putText(frame, f'adjust your ankle wider', (10, 260), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)  
  
if squat_stand !=0:  
    if right_knee.x < right_foot_index.x :  
        if right_knee.x + (squat_stand) < right_foot_index.x:  
            set_error("knee_fa",7,["ankle_cl","knee_fa","ankle_wi"])  
            cv2.putText(frame, f'false knee position', (10, 290), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)  
        else:  
            if right_knee.x - (squat_stand) > right_foot_index.x:  
                set_error("knee_fa",7,["ankle_cl","knee_fa","ankle_wi"])  
                cv2.putText(frame, f'false knee position', (10, 290), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
```

4. Lat Pulldown

- ใช้การคำนวณระยะห่างในระบบ 3 มิติ : เปรียบเทียบ $wrist_distance > shoulder_distance * 2.5$ ถ้าเงื่อนไขเป็นจริงแสดงว่า การใช้มือจับอุปกรณ์นั้นห่างเกินไป จะแสดงข้อความว่า Keep Wrists Closer

```
shoulder_distance = calculate_distance(
    [right_shoulder.x, right_shoulder.y, right_shoulder.z],
    [left_shoulder.x, left_shoulder.y, left_shoulder.z]
)
wrist_distance = calculate_distance(
    [right_wrist.x, right_wrist.y, right_wrist.z],
    [left_wrist.x, left_wrist.y, left_wrist.z]
)
if wrist_distance > shoulder_distance * 2.5:
    set_error("wrist_c1", 7)
    cv2.putText(frame, "Keep Wrists Closer", (10, 260), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0), 2)
```

5. Incline Dumbbell Press

- ใช้การคำนวณมุมแบบ 3 มิติ แลการเทียบตำแหน่งในแกน y เปรียบเทียบ $left_elbow.y * 0.95 > left_shoulder.y$ คือท่าทางในการดันน้ำหนักขึ้น
- ถ้าเงื่อนไข $shoulder_angle > np.degrees(2.7925268)$ เป็นจริงแสดงว่า ข้อศอกนั้นกางออกเกินไป(เกิน 160 องศา)
- ถ้าเงื่อนไข $shoulder_angle < np.degrees(1.0471976)$ เป็นจริงแสดงว่า ข้อศอกนั้นหุบเข้าหาลำตัวเกินไป (น้อยกว่า 60 องศา)

```
if left_elbow.y * 0.95 > left_shoulder.y:
    shoulder_angle = calculate_angle([right_shoulder.x, right_shoulder.y, right_shoulder.z],
                                     [right_elbow.x, right_elbow.y, right_elbow.z],
                                     [right_hip.x, right_hip.y, right_hip.z])

    # Pose correction: Check if shoulder angle is too wide (indicating improper elbow position)
    if shoulder_angle > np.degrees(2.7925268): # Check for angle > 160° (example threshold)
        cv2.putText(frame, f'Too wide elbow!', (10, 160), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 255, 0),
                    2)

    # Pose correction: Check if elbow is too high (should be closer to the torso)
    elif left_elbow.y < left_shoulder.y:
        cv2.putText(frame, f'Elbow too high!', (10, 160), cv2.FONT_HERSHEY_SIMPLEX, 1, (0, 0, 255),
                    2)

    # Pose correction: Check for a proper range of motion (ROM)
    elif shoulder_angle < np.degrees(1.0471976): # Check for angle < 60° (example threshold)
        cv2.putText(frame, f'Elbow too close to torso', (10, 160), cv2.FONT_HERSHEY_SIMPLEX, 1,
                    (0, 0, 255), 2)
```

การเทรนโมเดล (Model Training)

ใช้โมเดล LSTM (Long Short-Term Memory) ซึ่งเป็นหนึ่งในเทคนิคของ **Deep Learning** โดยนำข้อมูลที่ผ่านการ Pre-processing มาใช้ในการเทรนโมเดลตามขั้นตอนดังนี้

1. การเตรียมข้อมูล (Data Preparation)

- อ่านข้อมูลจากไฟล์ CSV ซึ่งมีค่ามุมทั้ง 8 มุมและประเภทของท่าออกกำลังกายทั้งหมด 5 ท่าได้แก่ Squat, Bicep Curl, Dumbbell Row, Incline Dumbbell Press และ Lat Pulldown
- แปลงค่าประเภทของท่าออกกำลังกายให้เป็นตัวเลขด้วย LabelEncoder จาก sklearn เพื่อใช้ในกระบวนการเทรน
- การปรับขนาดข้อมูล (Standardization) : ใช้ StandardScaler ปรับข้อมูลให้อยู่ในช่วงมาตรฐาน (Mean = 0, Std = 1) เพื่อช่วยให้โมเดลเรียนรู้ได้ดีขึ้น

2. การสร้างลำดับข้อมูล (Sequence Preparation)

- แบ่งข้อมูลตามเฟรม โดยแต่ละเฟรมจะถือเป็น 1 ข้อมูล (1 frame = 1 data)
- กำหนดประเภทของท่าออกกำลังกาย (Labels) จะถูกกำหนดให้กับแต่ละเฟรมของวิดีโอ ซึ่งจะแสดงถึงประเภทของท่าที่ปรากฏในเฟรมที่เกี่ยวข้อง ตัวอย่างเช่น ถ้าในเฟรมที่ 1-40 แสดงท่า **Squat** ก็จะกำหนด **Label : Squat** ให้กับแต่ละเฟรมในลำดับนั้น

3. การแบ่งชุดข้อมูล (Data Splitting)

- Training Set : 80% ใช้สำหรับฝึกโมเดล
- Test Set : 20% ใช้สำหรับทดสอบประสิทธิภาพของโมเดล

4. การสร้างโมเดล LSTM (LSTM Model Architecture)

โครงสร้างของโมเดล มีดังนี้

- LSTM Layer ประกอบไปด้วย LSTM 2 ชั้นที่มีจำนวน 64 units ในแต่ละชั้น โดยชั้นแรกจะมีการตั้งค่า ReturnSequences = True เพื่อส่งข้อมูลไปยัง LSTM ชั้นถัดไป
- ใช้ Dropout ในการลดปัญหาการเกิด overfitting โดยตั้งค่า Dropout Rate ที่ 20%
- หลังจากการประมวลผลจาก LSTM จะมี Dense Layer ที่ประกอบด้วย 32 units พร้อมกับ activation function แบบ ReLU
- Output Layer ใช้ Softmax เพื่อให้การจำแนกประเภทท่าทางออกกำลังกาย (5 ประเภท) โดยมีผลลัพธ์เป็นค่าความน่าจะเป็นของท่าทางแต่ละประเภท

เหตุผลที่เลือกใช้ LSTM (Long Short-Term Memory)

ในโครงงานนี้ การใช้ **LSTM (Long Short-Term Memory)** เป็นทางเลือกที่เหมาะสมที่สุดสำหรับการตรวจจับและจำแนกท่าออกกำลังกายจากข้อมูลวิดีโอ เนื่องจาก **LSTM** เป็นส่วนหนึ่งของ **Recurrent Neural Networks (RNN)** ที่ได้รับการออกแบบมาเพื่อจัดการกับข้อมูลที่มีลำดับ (Sequence Data) ซึ่งสามารถประมวลผลข้อมูลที่มีการเปลี่ยนแปลงต่อเนื่อง เช่น ข้อมูลการเคลื่อนไหวของร่างกายในแต่ละเฟรมของวิดีโอ

- **จัดการกับข้อมูลลำดับ (Sequential Data):** ท่าออกกำลังกายที่ใช้ในโครงงานนี้เกี่ยวข้องกับการเคลื่อนไหวที่มีลำดับและต่อเนื่อง เช่น การยืดและงอของข้อต่อในท่า Squat หรือ Dumbbell Row ข้อมูลเหล่านี้มีการเปลี่ยนแปลงตามเวลาจากเฟรมหนึ่งไปยังเฟรมถัดไป ซึ่ง **LSTM** สามารถจัดการได้ดี เพราะมันถูกออกแบบมาเพื่อจัดการกับข้อมูลที่มีลำดับยาวและสามารถจดจำข้อมูลในระยะยาวได้

- **สามารถจดจำข้อมูลระยะยาว (Long-Term Dependencies) :** LSTM สามารถแก้ปัญหาการหายไปของเกรเดียนต์ (Vanishing Gradient Problem) ที่พบใน RNN ทั่วไปได้ จึงสามารถจดจำข้อมูลที่สำคัญระยะยาวได้ เช่น การเชื่อมโยงการเคลื่อนไหวในเฟรมต่างกันได้
- **ประสิทธิภาพในข้อมูลที่มีการเปลี่ยนแปลงตามเวลา:** การออกกำลังกายแต่ละท่ามีการเคลื่อนไหวที่ต้องเชื่อมโยงกับท่าทางในลำดับต่าง ๆ ซึ่ง LSTM สามารถ "จำ" ลำดับของการเคลื่อนไหวที่ซับซ้อนและแยกแยะความแตกต่างของท่าออกกำลังกายได้ดี
- **สามารถทำงานร่วมกับข้อมูลที่ได้จาก MediaPipe:** ในโครงงานนี้ ข้อมูลจาก MediaPipe ที่ใช้ในการสร้าง **skeleton** และคำนวณมุมของข้อต่อในแต่ละเฟรมของวิดีโอจะถูกส่งเข้าโมเดล LSTM ซึ่งจะช่วยให้โมเดลสามารถเรียนรู้จากลำดับของการเคลื่อนไหวจากแต่ละจุด (เช่น ข้อมือ, ข้อศอก, หัวเข่า) ได้อย่างแม่นยำ

การพัฒนา Web Application

หลังจากการสร้างโมเดลเสร็จสมบูรณ์ เราได้พัฒนา Web Application เพื่ออำนวยความสะดวกให้ผู้ใช้งานสามารถเข้าถึงโปรแกรมได้จากทุกที่ผ่านเว็บเบราว์เซอร์ โดยไม่จำเป็นต้องติดตั้งซอฟต์แวร์เพิ่มเติม ซึ่ง Web Application ถูกพัฒนาขึ้นโดยใช้เทคโนโลยีดังนี้

1. **Streamlit :** ใช้ในการพัฒนาโครงสร้างหลักของ Web Application ช่วยให้การสร้างหน้าเว็บเป็นไปอย่างรวดเร็ว
2. **React :** ใช้ในการพัฒนา User Interface (UI) เพื่อให้หน้าเว็บมีความทันสมัย ใช้งานง่าย และตอบสนองต่อผู้ใช้ได้อย่างราบรื่น
3. **Python :** ใช้ในการเรียกใช้งานโมเดลที่พัฒนาขึ้น โดยเชื่อมต่อกับส่วนของ Web Application เพื่อประมวลผลข้อมูล que ผู้ใช้งานส่งมา

Web Application มีหน้าเว็บทั้งหมด 4 หน้า ดังนี้

1. หน้ายินดีต้อนรับสู่โปรแกรม

- แสดงข้อความต้อนรับผู้ใช้งาน พร้อมอธิบายสั้น ๆ เกี่ยวกับโปรแกรม
- จัดทำให้ผู้ใช้งานสามารถเข้าใจวัตถุประสงค์ของโปรแกรมได้อย่างรวดเร็ว

X-POSE

How It Works

1. **Upload or Capture Video:** Record your movement or use live video input.
2. **Pose Analysis:** Our AI model identifies key points and evaluates your posture.
3. **Feedback Generation:** Receive actionable insights to improve your form.
4. **Progress Tracking:** Monitor improvements over time with detailed analytics.

2. หน้าแสดงรายชื่อสมาชิก

- ระบุชื่อและรหัสนักศึกษาของสมาชิกในโครงการ

Our Team



Jarupat Chodsitanan

6513161



Chalisa Buathong

6513163

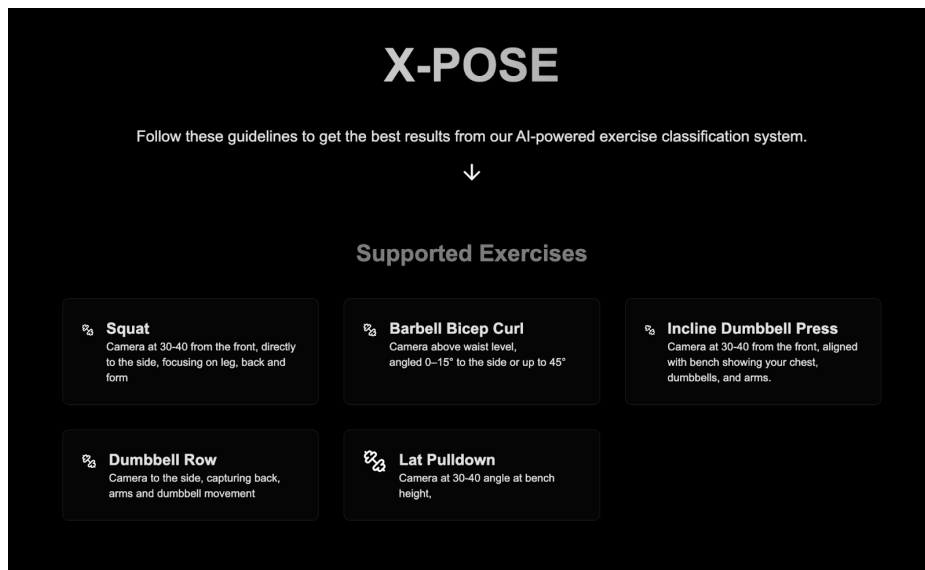


Nopparuj Ritnatikul

6513168

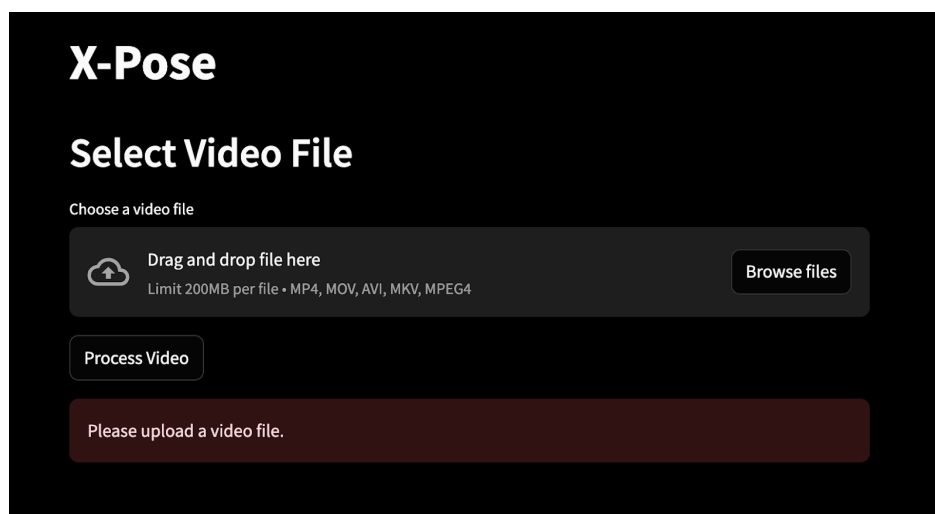
3. หน้าแสดงคำแนะนำก่อนเริ่มใช้โปรแกรม

- อธิบายวิธีการเตรียมวิดีโอและข้อควรระวังในการอัปโหลดไฟล์
- เพิ่มความเข้าใจแก่ผู้ใช้งานเกี่ยวกับการใช้งานโปรแกรมอย่างถูกต้องและมีประสิทธิภาพ



4. หน้าใช้งานโปรแกรม

- ผู้ใช้งานสามารถอัปโหลดวิดีโอการออกกำลังกายเข้าสู่ระบบได้
- โปรแกรมจะวิเคราะห์ข้อมูลจากวิดีโอ โดยใช้โมเดลที่พัฒนาขึ้น และแสดงผลลัพธ์ เช่น ประเภทของท่าออกกำลังกาย ข้อผิดพลาด และคำแนะนำในการปรับปรุงท่า



สรุปคือ Web Application นี้สามารถ

- **รองรับการใช้งานจากหลากหลายอุปกรณ์** : ผู้ใช้งานสามารถใช้ Web Application ผ่านคอมพิวเตอร์หรืออุปกรณ์พกพา เช่น โทรศัพท์มือถือหรือแท็บเล็ต
- **การประมวลผลแบบอัตโนมัติ** : ระบบสามารถประมวลผลวิดีโอและแสดงผลพร้อมได้ในระยะเวลาอันสั้น
- **ไม่ต้องติดตั้งซอฟต์แวร์เพิ่มเติม** : ช่วยลดความยุ่งยากในการใช้งาน และเพิ่มความสะดวกสำหรับผู้ใช้งาน

ข้อจำกัดของโปรแกรม

1. โปรแกรมรองรับท่าออกกำลังกายเพียง 5 ประเภท ได้แก่ Squat, Bicep Curl, Incline Dumbbell Press, Dumbbell Row และ Lat Pulldown ทำให้ไม่สามารถตรวจจับหรือให้คำแนะนำสำหรับท่าออกกำลังกายประเภทอื่นได้
2. MediaPipe อาจไม่สามารถตรวจจับ Landmark ได้อย่างแม่นยำในบางกรณี เช่น การจับภาพที่มีแสงไม่เพียงพอ หรือมุมกล้องที่ไม่เหมาะสม ส่งผลให้การคำนวณมุมและการให้คำแนะนำแก่ผู้ใช้ผิดพลาด
3. โปรแกรมมีข้อจำกัดในการคำนวณท่าทางที่ซับซ้อน โดยเฉพาะในระบบ 3 มิติที่อาจไม่แม่นยำ เนื่องจากข้อจำกัดในการจับตำแหน่งของ Landmark ที่ได้กล่าวไปข้างต้นในข้อที่ 2
4. เพื่อให้การคำนวณและการคาดเดาของโมเดลถูกต้อง มุมกล้องในวิดีโอต้องอยู่ในทิศทางที่กำหนดตามคำแนะนำ "How to use" ดังนั้นการถ่ายทำในมุมที่ไม่เหมาะสมอาจส่งผลต่อประสิทธิภาพของโมเดล
5. การประมวลผลวิดีโออาจใช้เวลานาน โดยเฉพาะวิดีโอที่มีความยาวหรือเฟรมเรตสูง
6. หากมีการเปลี่ยนท่าออกกำลังกายในคลิปเดียวกัน ช่วงที่เริ่มทำท่าต่อไปอาจเกิดดีเลย์เล็กน้อยในการคาดเดาท่าของโมเดล เนื่องจากขึ้นอยู่กับความยาวของลำดับข้อมูล (sequence length)
7. Web Application รองรับไฟล์วิดีโอที่มีขนาดไม่เกิน 200MB และต้องเป็นไฟล์วิดีโอเท่านั้น รวมไปถึงไม่สามารถรองรับ Real-Time ได้
8. เมื่อคลิปวิดีโอประมวลผลเสร็จแล้ว ไม่สามารถที่จะแสดงบน Web Application ได้ทันที จำเป็นที่จะต้องดาวน์โหลดเพื่อเปิดดูคลิปที่ผ่านการประมวลผลแล้วเท่านั้น
9. Web Application รองรับการประมวลผลได้ทีละวิดีโอ ไม่สามารถประมวลผลหลายวิดีโอ ได้ในเวลาเดียวกัน

ผลการทดลอง (Experiment results)

โครงการนี้ได้ทำการเทรนโมเดล ได้แก่ **LSTM**, **GRU** และ **MLP** โดยเทรนโมเดลทั้ง 3 ตัวด้วยข้อมูลในไฟล์ output.csv ซึ่งมีมุมทั้งหมด 8 มุม ได้แก่

ข้างขวา : right_shoulder_angle, right_elbow_angle, right_hip_angle,
right_knee_angle

ข้างซ้าย : left_shoulder_angle, left_elbow_angle, left_hip_angle,
left_knee_angle

ทำการตรวจสอบความแม่นยำ MLP โมเดล ด้วย sklearn train_test_split ในการแยกข้อมูลไว้ทดสอบ 20% และใช้ accuracy_score ในการวัดความแม่นยำ การตรวจสอบความแม่นยำ GRU และ LSTM โมเดลด้วย sklearn train_test_split ในการแยกข้อมูลไว้ทดสอบ 20% ใช้ tensorflow.keras.models Sequential และใช้ function evaluate ได้ค่าความแม่นยำเป็น

LSTM: 99.91%, GRU: 93.10% , MLP: 100%

ได้ทำการทดลองใช้โมเดล ที่ทำการ train กับวิดีโอชุดเดียวกันทั้งหมด โดยชุดวิดีโอเป็นวิดีโอที่ใช้ในการทำไฟล์ output.csv และได้หาความถูกต้องในการคาดเดาของแต่ละโมเดลบนชุดข้อมูลวิดีโอทดสอบได้ผลลัพธ์ดังนี้

LSTM: 99.9616% , GRU: 12.7177% , MLP: 15.6333%

โมเดล LSTM มีประสิทธิภาพสูงกว่าโมเดล GRU และ MLP ในการทำนาย frame ของการออกกำลังกายอย่างมีนัยสำคัญ ในทางกลับกัน โมเดล GRU และ MLP มีความถูกต้องต่ำกว่า ซึ่งบ่งชี้ว่าพวกมันมีประสิทธิภาพน้อยกว่าในงานนี้

สรุปผลการทดลองใช้โมเดล

โมเดลที่ใช้	ความแม่นยำที่ได้จากการประเมิน	ความแม่นยำที่ได้จากการทดลองใช้จริง
LSTM	99.91%	99.9616%
GRU	93.10%	12.7177%
MLP	100%	15.6333%

ความสามารถในการหา Long-Term Dependencies

- **LSTM** : Memory cells และ Gating mechanisms (มี 3 gate 1. input gate 2. Forget gate 3. Output gate) ใน LSTM ทำให้มีประสิทธิภาพในการหา **Long-Term Dependencies** : ในข้อมูลลำดับ ซึ่งเป็นสิ่งสำคัญสำหรับงานอย่างการทำนาย frame การออกกำลังกายที่โมเดลจำเป็นต้องจำและใช้ข้อมูลจาก frame ก่อนหน้า
- **GRU** : แม้ว่า GRU สามารถหา long-term dependencies ได้ แต่โครงสร้างที่ง่ายกว่า(มีแค่ 2 gate 1. update gate 2. reset gate)น่าจะไม่สามารถรักษาและใช้ข้อมูลนี้ได้อย่างมีประสิทธิภาพเท่ากับ LSTM ในข้อมูลลำดับที่ยาวขึ้น

ความเร็วในการเทรนโมเดล

- **LSTM** : มีพารามิเตอร์จำนวนมากขึ้น และ gate จำนวนมาก ใช้เวลาในการ train และใช้การคำนวณที่มากกว่า GRU
- **GRU** : สถาปัตยกรรมที่ง่ายกว่าของ GRU จำนวน gate น้อยกว่าหมายถึงการคูณเมทริกซ์น้อยกว่า มีประโยชน์ในการเทรนที่เร็วกว่าและลดภาระในการคำนวณ

ความเร็วในการคาดเดาคำตอบ

- **LSTM** : ความซับซ้อนของการดำเนินการของ LSTM สามารถทำให้เวลาในการประมวลผลต่อลำดับที่ยาวขึ้น
- **GRU** : การประมวลเร็วขึ้นเนื่องจากโครงสร้างที่ง่ายกว่า ทำให้มีประสิทธิภาพมากขึ้นสำหรับการใช้งาน real-time

บทสรุปของโครงการ

โครงการ **Exercise Pose Correction and Classification** มีเป้าหมายเพื่อพัฒนาโปรแกรมที่สามารถช่วยผู้ใช้งานตรวจจับข้อผิดพลาดในท่าทางการออกกำลังกาย พร้อมทั้งให้คำแนะนำการปรับปรุงเทคนิคให้อยู่ในลักษณะที่ถูกต้องและปลอดภัย นอกจากนี้ยังสามารถจำแนกประเภทท่าทางการออกกำลังกายได้อย่างแม่นยำ โดยมีการรองรับท่าพื้นฐาน 5 ท่า ได้แก่ Squat, Bicep Curl, Incline Dumbbell Press, Dumbbell Row และ Lat Pulldown

เครื่องมือและเทคโนโลยีที่ใช้

- **MediaPipe** สำหรับการตรวจจับ Skeleton และสร้าง Landmark ของร่างกาย
- **LSTM โมเดล** ใช้ในการวิเคราะห์และจำแนกท่าทางจากข้อมูลที่ได้ โดยได้ค่า Accuracy อยู่ที่ 99.9616%
- **Web Application** พัฒนาด้วย Python (Streamlit) และ React เพื่อให้ผู้ใช้งานสามารถอัปโหลดวิดีโอและรับคำแนะนำได้ง่าย

ขอบเขตการทำงาน

ระบบสามารถตรวจจับจุดสำคัญบนร่างกาย เช่น ข้อศอก หัวไหล่ สะโพก และเข่า เพื่อคำนวณมุมที่จำเป็นสำหรับการวิเคราะห์ข้อผิดพลาดและแนะนำการปรับปรุง โดยข้อมูลทั้งหมดถูกประมวลผลผ่านโมเดล LSTM ที่พัฒนามาจากชุดข้อมูลที่ผ่านการ Pre-process ด้วย MediaPipe

การต่อยอดในอนาคต (Future work)

1. หาวิธีคำนวณมุมกล้ามเนื้อและใช้ range มุมกล้ามเนื้อเพื่อเลือกโมเดลที่ใช้ข้อมูลจากมุมกล้ามเนื้อนั้นๆ ในการทำนายผลให้แม่นยำขึ้น หรือใช้ range มุมกล้ามเนื้อไปเป็น feature ของโมเดล
2. ศึกษาและปรับปรุง pipeline ของ MediaPipe เพื่อให้สามารถตรวจจับและสร้าง landmarks ได้แม่นยำยิ่งขึ้น ซึ่งจะช่วยในการคำนวณข้อผิดพลาดต่างๆ ได้ถูกต้องและหลากหลายมากขึ้น
3. เพิ่ม features ในการตรวจจับอุปกรณ์ (เช่น ดัมเบล) เพื่อเพิ่มความแม่นยำในการจำแนกท่าทาง เนื่องจากท่าที่มีมุมการเคลื่อนไหวคล้ายกันอาจมีความแตกต่างกันตามการใช้อุปกรณ์ เช่น bicep curl กับ tricep push down
4. เพิ่มความหลากหลายของ Dataset โดยเก็บข้อมูลในสภาพแวดล้อมที่แตกต่างกัน เช่น มุมกล้ามเนื้อหลายมุม ระดับแสงที่แตกต่างกัน เพื่อให้โมเดลสามารถทำงานได้แม่นยำในหลายสถานการณ์