



EGCO486 : Image Processing

Final Project

จัดทำ โดย

นายณัฏทพ	นรพันธ์พิมล	6613120
นายภวินท์	ทองเรือง	6613266
นายระพีพัฒน์	บุญสุข	6613269

เสนอ

ผศ.ดร. นริศ หนูหอม

คณะวิศวกรรมคอมพิวเตอร์ มหาวิทยาลัยมหิดล

ภาคเรียนที่ 1 ปีการศึกษา 2568

คำนำ

รายงานฉบับนี้จัดทำ ขึ้นเพื่อเป็นส่วนหนึ่งของรายวิชา Image Processing(EGCO486) โดยคณะผู้จัดทำ ได้จัดทำขึ้นเพื่อใช้ประกอบการทำ โครงการเรื่อง MEME 4 ME โดยโครงการนี้มุ่งพัฒนาระบบที่ใช้เทคนิค Human Pose Estimation ตรวจจับจุดสำคัญของร่างกาย (keypoints) เพื่อเปรียบเทียบและค้นหา มิมที่มีท่าทางคล้ายกันในฐานะข้อมูล ทำให้ผู้ใช้สามารถค้นหา มิมได้รวดเร็วและแม่นยำ รวมถึงนำเทคนิค ตรวจจับท่าทางมาประยุกต์ใช้กับฐานข้อมูลภาพอย่างมีประสิทธิภาพในการวิเคราะห์การแสดงออกทางร่างกายในบริบทของมิม.

ทางคณะผู้จัดทำ ขอขอบพระคุณ ผศ.ดร. นริศ หนูหอม ผู้ให้ความรู้ และ แนวทางการศึกษา สุดท้ายนี้ทางคณะผู้จัดทำ หวังว่ารายงานฉบับนี้จะสามารถเป็นประโยชน์ไม่มากนักน้อยแก่ผู้อ่านทุกท่าน หากมีข้อผิดพลาดประการใด ผู้จัดทำ ขออภัยไว้ และขออภัยมา ณ ที่นี้

ขอขอบพระคุณ

คณะผู้จัดทำ

สารบัญ

ที่มาและความสำคัญของโครงงาน (Background)	4
วัตถุประสงค์ของโครงงาน (Objective)	5
ขอบเขตของโครงงาน (Scope of Work)	5
ขอบเขตด้านข้อมูล (Data Scope)	5
ขอบเขตด้านการประมวลผล (Processing Scope)	5
ขอบเขตด้านระบบ (System Scope)	5
กลุ่มเป้าหมาย (Target Group)	6
วิธีการดำเนินงาน (Method)	7
การระบุตำแหน่ง Keypoints (Pose Estimation)	8
การเทรนโมเดล (Model Training)	9
การเตรียมข้อมูล (Data Preparation)	9
การแบ่งชุดข้อมูล (Data Splitting)	10
ขั้นตอนการสร้างไฟล์ data.yaml	10
การฝึกโมเดล (Model Training Process)	11
การคำนวณความคล้ายของรูปที่ต้องการกับฐานข้อมูลมีม	12
● Normalization ของ keypoints	13
● การจับคู่จุด keypoints ระหว่างภาพ	13
● คำนวณระยะห่างระหว่าง keypoints	13
● ดีความผลลัพธ์	13
ทดสอบและประเมินผล (Model Evaluation)	14
นำโมเดลไปใช้งานในการค้นหา Meme (Deployment)	14
ความแม่นยำในการทำนายของ Model	16
ข้อกำหนดของโปรแกรม	17
การจำกัดขอบเขตการตรวจสอบโครงกระดูก	17
ข้อกำหนดของชุดข้อมูลมีมภายในระบบ	17
ความแม่นยำในการเปรียบเทียบลดลงในกรณีที่มีหลายบุคคลในภาพ	17
บทสรุปของโครงงาน	18
การต่อยอดในอนาคต (Future Work)	19
เพิ่มความหลากหลายของมีมและท่าทาง	19
พัฒนาความแม่นยำของระบบ	19
การโต้ตอบและประสบการณ์ผู้ใช้	19

ที่มาและความสำคัญของโครงการ (Background)

ในยุคที่สื่อสังคมออนไลน์มีบทบาทสำคัญต่อการสื่อสารและความบันเทิงของผู้คนทั่วโลก “Meme” ได้กลายเป็นหนึ่งในรูปแบบการสื่อสารที่แพร่หลายและมีอิทธิพลสูง การสื่อสารผ่านมีมเป็นวิธีที่เรียบง่าย รวดเร็ว และช่วยสร้างความเข้าใจร่วมกันในสังคมดิจิทัล ไม่ว่าจะเป็นการล้อเลียน เหตุการณ์ กระแสวิพากษ์ หรือความรู้สึกต่าง ๆ ซึ่งการสร้างและเผยแพร่มีมกลายเป็นวัฒนธรรมออนไลน์ที่เติบโตอย่างรวดเร็ว

ภาพมีมจำนวนมากมักมีลักษณะท่าทาง (pose) ที่โดดเด่น เช่น การขี้นิ้ว การยกมือ การทำหน้าแปลก ๆ หรือท่าทางที่ถูกใช้ซ้ำในมีมต่าง ๆ ทำให้ “ท่าทางของมนุษย์” กลายเป็นหนึ่งในองค์ประกอบสำคัญของการสร้างและรู้จำมีม หากสามารถวิเคราะห์ท่าทางจากภาพของผู้ใช้ แล้วค้นหา meme ที่มีท่าทางคล้ายกันได้ ก็จะช่วยให้ผู้ใช้งานสามารถค้นหามีมที่ตรงตามความต้องการได้สะดวกยิ่งขึ้น

ด้วยเทคโนโลยีด้านปัญญาประดิษฐ์และการประมวลผลภาพ (Image Processing) ที่พัฒนาอย่างก้าวกระโดด โดยเฉพาะเทคนิค Human Pose Estimation ทำให้สามารถตรวจจับจุดสำคัญของร่างกายมนุษย์ (keypoints) ได้อย่างแม่นยำ จึงเกิดเป็นแนวคิดในการพัฒนาระบบที่นำจุด keypoints ของผู้เข้ามาเปรียบเทียบกับฐานข้อมูลมีม เพื่อค้นหามีมที่มีท่าทางคล้ายคลึงกันมากที่สุด

โครงการนี้มีความสำคัญเนื่องจากช่วยอำนวยความสะดวกให้ผู้ใช้งานสามารถค้นหามีมที่ต้องการได้อย่างรวดเร็วและแม่นยำจากท่าทางในภาพของตนเอง นอกจากนี้ยังเป็นการนำเทคนิคตรวจจับท่าทางมนุษย์มาประยุกต์ใช้กับฐานข้อมูลภาพอย่างมีประสิทธิภาพ ทำให้สามารถเข้าใจและวิเคราะห์การแสดงออกทางร่างกายของมนุษย์ในบริบทของมีมได้อย่างชัดเจน

วัตถุประสงค์ของโครงการ (Objective)

1. เพื่อพัฒนาระบบตรวจจับท่าทางมนุษย์จากภาพนิ่ง

โดยใช้เทคนิค Human Pose Estimation เพื่อดึงจุดสำคัญ (keypoints) ของร่างกาย

2. เพื่อออกแบบและพัฒนาวิธีการวัดความคล้ายของท่าทาง

ผ่านการเปรียบเทียบ keypoints ระหว่างภาพของผู้ใช้และฐานข้อมูลมีม

3. เพื่อสร้างระบบค้นหาและจัดอันดับ (ranking)

ที่สามารถแสดงผลมีมที่มีท่าทางใกล้เคียงที่สุดอย่างน้อย 1-3 อันดับ

ขอบเขตของโครงการ (Scope of Work)

ขอบเขตด้านข้อมูล (Data Scope)

- รับข้อมูลเป็นภาพนิ่งของบุคคล 1 คนต่อครั้ง ใช้ฐานข้อมูลมีมที่เตรียมไว้ล่วงหน้า 10 รูป

ขอบเขตด้านการประมวลผล (Processing Scope)

- ตรวจจับจุดสำคัญของร่างกาย (keypoints) ด้วยเทคนิค Human Pose Estimation
- เปรียบเทียบท่าทางของผู้ใช้กับฐานข้อมูลมีมเพื่อหาความคล้ายคลึง

ขอบเขตด้านผลลัพธ์ (Output Scope)

- แสดงผลมีมที่ใกล้เคียงที่สุดตามระดับความคล้าย (Ranking)
- ไม่ครอบคลุมการวิเคราะห์ข้อความบนมีม หรือการตรวจจับอารมณ์ของภาพ

ขอบเขตด้านระบบ (System Scope)

- ระบบทำงานบนภาพนิ่ง ไม่รองรับวิดีโอหรือภาพเคลื่อนไหว
- ใช้งานสำหรับบุคคลเดียวต่อการวิเคราะห์ครั้งหนึ่ง

กลุ่มเป้าหมาย (Target Group)

1. ผู้ใช้ทั่วไปบนสื่อสังคมออนไลน์

- ผู้ที่สนใจค้นหาและแชร์มีมที่ตรงกับทำทางหรืออารมณ์ของตนเอง

2. นักสร้างสรรค์คอนเทนต์ (Content Creators)

- ผู้ที่ต้องการแรงบันดาลใจจากทำทางในมีมเพื่อนำไปสร้างมีมใหม่ หรือใช้ในการทำคอนเทนต์สนุก ๆ

วิธีการดำเนินงาน (Method)

โครงการนี้เลือกใช้โมเดล YOLO11n สำหรับงานตรวจจับ Skeleton และ keypoints ของร่างกาย เนื่องจากมีจุดเด่นหลายประการที่เหมาะสมกับลักษณะงานและข้อจำกัดของระบบ ดังนี้:

1. ประสิทธิภาพสูงแม้ใช้ทรัพยากรน้อย (Lightweight & Efficient)

YOLO11n เป็นรุ่น Nano ที่ออกแบบให้มีจำนวนพารามิเตอร์น้อย ทำให้สามารถรันได้รวดเร็วบนเครื่องที่มีประสิทธิภาพต่ำ ซึ่งสอดคล้องกับการใช้งานของโครงการที่ต้องการความเร็วแต่ไม่ต้องใช้โมเดลขนาดใหญ่

2. ความเร็วในการประมวลผลสูง (High Inference Speed)

YOLO11n เหมาะกับงานที่ต้องการประมวลผลเร็ว เช่น การตรวจจับ keypoints จากภาพที่อัปโหลดแบบทันที เนื่องจากมีความเร็วสูงกว่า YOLO รุ่นใหญ่ เช่น YOLO11s หรือ YOLO11m ทำให้ระบบตอบสนองผู้ใช้ได้รวดเร็ว

3. เพียงพอต่อความต้องการของโครงการ (Fit for the Project Scope)

โครงการใช้เฉพาะการตรวจจับร่างกาย และตรวจจับบุคคลเพียง 1 คนต่อภาพ ดังนั้นไม่จำเป็นต้องใช้โมเดลขนาดใหญ่ที่ใช้เวลามาก YOLO11n ให้ความแม่นยำเพียงพอสำหรับงานระดับนี้

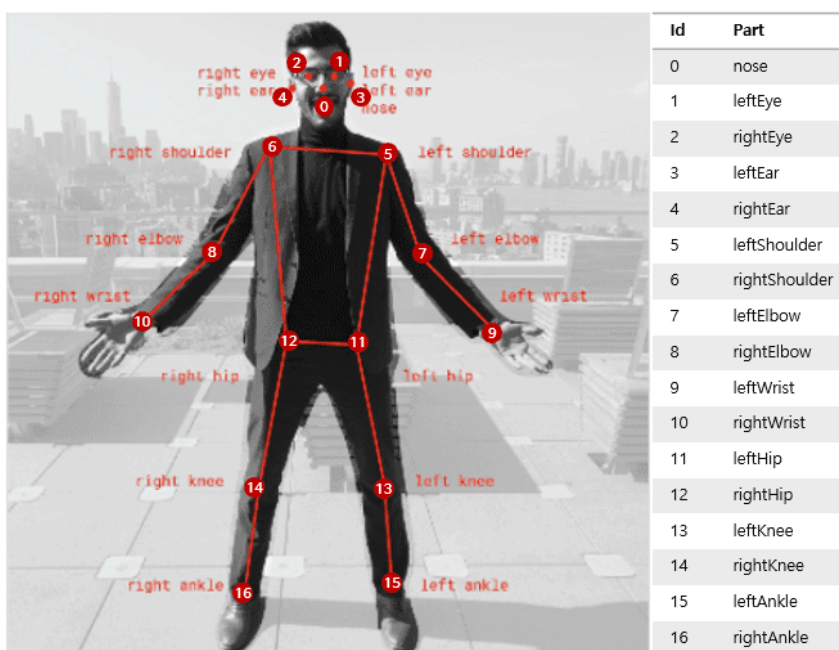
4. ประหยัดพลังงานและลดเวลาประมวลผล (Low Computational Cost)

YOLO11n ต้องการการประมวลผลน้อยกว่าโมเดลขนาดใหญ่ ทำให้สามารถรันได้ต่อเนื่องโดยไม่เกิดปัญหา เช่น ความร้อนสูงหรือความหน่วงของระบบ

การระบุตำแหน่ง Keypoints (Pose Estimation)

หลังจากตรวจจับบุคคลได้ YOLO Pose จะทำการประมวลผลเพื่อระบุตำแหน่งจุดสำคัญของร่างกาย (keypoints) เช่น

- ศีรษะ (head)
- คอ (neck)
- ไหล่ซ้าย-ขวา
- ศอกซ้าย-ขวา
- ข้อมือซ้าย-ขวา
- ท่าทางของลำตัว

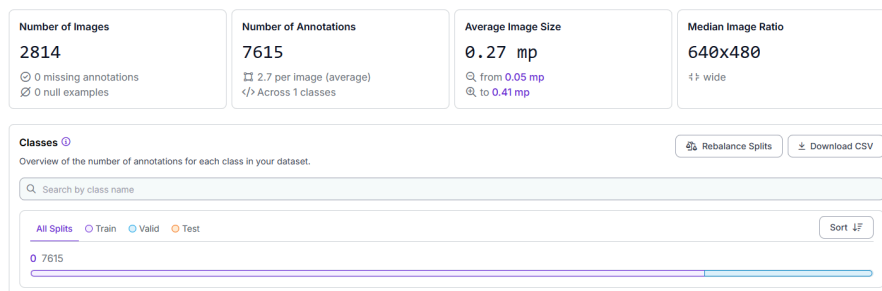


การเทรนโมเดล (Model Training)

การเตรียมข้อมูล (Data Preparation)

ดึงข้อมูลรูป coco_minitrain_25k จาก hugging face และ label ได้จาก json ใน GitHub giddyup/coco-minitrain ปรับข้อมูล json ให้อยู่ในรูปแบบ .txt

โดยข้อมูลมีใน hugging face ที่นำมาใช้ทั้งหมด 2814 รูป มี 1 class คือ person



รูปแบบข้อมูลต้องเป็นไปตามโครงสร้างของ YOLO Pose:

- โฟลเดอร์ images/ — เก็บรูปภาพ
- โฟลเดอร์ labels/ — เก็บไฟล์ .txt ที่มีข้อมูล keypoints
- ไฟล์ data.yaml — กำหนด path และจำนวน keypoints

ข้อมูลใน label ของ YOLO Pose กำหนดดังนี้:

<class> <x_center> <y_center> <w> <h> <k1_x> <k1_y> <k1_visible> ... <kN_x> <kN_y> <kN_visible>

- <class> คือ หมายเลขคลาสของอ็อบเจกต์
- <x_center> <y_center> คือ ตำแหน่งใจกลางของ bounding box (normalized)
- <w> <h> คือ ขนาดของ bounding box (normalized)
- ชุดของ keypoints รูปแบบของแต่ละ keypoint คือ:

<k_i_x> <k_i_y> <k_i_visible>

- <k_i_x> ตำแหน่งแกน X ของ keypoint ที่ถูก normalize
- <k_i_y> ตำแหน่งแกน Y ของ keypoint ที่ถูก normalize

- `<k_i_visible>` สถานะการมองเห็นของ keypoint

ค่า	ความหมาย
0	ไม่มี keypoint ($x = 0, y = 0$)
1	keypoint อยู่แต่ไม่ชัด / occluded
2	keypoint มองเห็นชัดเจน

การแบ่งชุดข้อมูล (Data Splitting)

- Training Set : 80% ใช้สำหรับฝึกโมเดล ทั้งหมด 2252 รูป
- Test Set : 20% ใช้สำหรับทดสอบประสิทธิภาพของ ทั้งหมด 562 รูป

ขั้นตอนการสร้างไฟล์ data.yaml

ใช้เพื่อ “บอกโครงสร้างข้อมูลที่ใช้ในการเทรนโมเดล” เช่น dataset อยู่ที่ไหน, มีกี่ class, ชื่อ class อะไรบ้าง รวมถึง config สำหรับ keypoints ด้วย

ไฟล์ data.yaml ใช้สำหรับบอกโครงสร้าง dataset ให้กับ YOLO ประกอบด้วย

- ที่อยู่ของโฟลเดอร์ train / val
- จำนวนคลาส และชื่อคลาส
- ประเภทงาน (ที่นี้คือ pose)
- จำนวน keypoints และ skeleton connections

ไฟล์ data.yaml สำหรับ YOLO Pose Estimation

```
train: images/train2017 #path ของ train image
val: images/val2017 #path ของ validate image
nc: 1 # มี 1 class
names: ['person'] #ชื่อ class
kpt_shape: [17, 3] #จำนวน keypoints
```

```
flip_idx: [0, 2, 1, 4, 3, 6, 5, 8, 7, 10, 9, 12, 11, 14, 13, 16, 15]
```

#บอกให้ YOLO รู้ว่า keypoints จัดเรียงแบบไหน

```
kpt_names:#ลำดับชื่อ keypoint
```

```
0:
```

- nose
- left_eye
- right_eye
- left_ear
- right_ear
- left_shoulder
- right_shoulder
- left_elbow
- right_elbow
- left_wrist
- right_wrist
- left_hip
- right_hip
- left_knee
- right_knee
- left_ankle
- right_ankle

การฝึกโมเดล (Model Training Process)

ระหว่างการฝึก โมเดลจะเรียนรู้สิ่งต่อไปนี้:

1. การจำแนกบุคคลในภาพ (Person Bounding Box)
2. การคาดตำแหน่งของ keypoints ให้แม่นยำขึ้น
3. การเพิ่มข้อมูลด้วย Data Augmentation

4. การเรียนรู้ท่าทางที่ปรากฏในชุดข้อมูล meme และท่าทางที่ระบบต้องใช้งานจริง

กำหนดค่าพารามิเตอร์สำคัญ

- epochs = 100: จำนวนรอบการฝึก 100 รอบ
- batch = 12: จำนวนภาพต่อครั้งที่ GPU จะนำไปประมวลผล
- imgsz= 640: ขนาดภาพ 640 x 640
- close_mosaic = 20: ปิด Mosaic augmentation เมื่อเหลือ 20 Epoch
- pose = 0.15: กำหนดค่า weight ในการวาง keypoint
- box = 0.85: กำหนดค่า weight ในการวาด Bounding box
- degrees = 0.10: สุ่มเอียงรูปตามตั้งแต่ -10° ถึง $+10^{\circ}$
- fliplr = 0.5: โอกาส 50% กลับรูปซ้ายหรือขวา
- hsv_h = 0.015: สุ่มเปลี่ยนค่า Hue ของรูปตั้งแต่ -1.5% ถึง $+1.5\%$
- hsv_s = 0.7: สุ่มเปลี่ยนค่า Saturation ของรูปตั้งแต่ -70% ถึง $+70\%$
- hsv_v = 0.4: สุ่มเปลี่ยนค่า Value ของรูปตั้งแต่ -40% ถึง $+40\%$
- device = 0: ใช้ GPU ในการเทรน

ผลลัพธ์ของการฝึกจะถูกบันทึกเป็นไฟล์:

- runs/pose/train/weights/best.pt — โมเดลที่แม่นยำที่สุด

การคำนวณความคล้ายของรูปที่ต้องการกับฐานข้อมูลมีม

วิธีการคำนวณความคล้ายกันของโครงกระดูก (Keypoints Similarity Calculation)

การเปรียบเทียบโครงกระดูกระหว่างภาพสองรูปเริ่มต้นจากการเตรียมข้อมูล keypoints ของแต่ละภาพให้อยู่ในรูปแบบที่สามารถเปรียบเทียบกันได้อย่างเหมาะสมแก่การทำ Feature Matching เนื่องจากแต่ละภาพอาจมีขนาดหรือสัดส่วนของร่างกายไม่เท่ากัน และบางภาพอาจมีการตรวจจับโครงกระดูกบาง

ส่วนไม่ครบ จึงต้องมีขั้นตอน Normalization เพื่อปรับตำแหน่งของจุด keypoints ให้มีมาตรฐานเดียวกัน ก่อนการเปรียบเทียบ

ขั้นตอนการคำนวณมีดังนี้:

- Normalization ของ keypoints

ปรับขนาดและตำแหน่งของ keypoints แต่ละจุดให้สัมพันธ์กับสเกลและตำแหน่งของร่างกายในภาพ ทำให้โครงกระดูกจากภาพที่มีขนาดต่างกัน หรือใช้บางส่วนของร่างกายต่างกัน สามารถนำมาวัดความคล้ายกันได้เหมาะสม

```
def normalize_keypoints(kpts):
    kpts = np.array(kpts)
    min_xy = kpts.min(axis=0)
    max_xy = kpts.max(axis=0)

    return (kpts - min_xy) / (max_xy - min_xy + 1e-6)
```

- การจับคู่จุด keypoints ระหว่างภาพ

สำหรับแต่ละจุด keypoint ของภาพต้นแบบและภาพเปรียบเทียบ จะนำมาจับคู่ตามตำแหน่งของจุดบนร่างกาย เช่น ศีรษะกับศีรษะ ไหล่ซ้ายกับไหล่ซ้าย เป็นต้น

- คำนวณระยะห่างระหว่าง keypoints

นำจุด keypoints ที่จับคู่แล้วมาหาค่า ระยะห่างเชิงเวกเตอร์ จากนั้นคำนวณค่าเฉลี่ยของระยะห่างทั้งหมด (Mean Euclidean Distance) เพื่อให้ได้ค่าตัวเลขเดียวแทนความแตกต่างรวมของท่าทาง

```
def pose_distance_norm(kpt1, kpt2):
    # ถ้าจำนวนจุดไม่เท่ากัน ไม่คำนวณ
    if len(kpt1) != len(kpt2):
        return float("inf")

    k1 = normalize_keypoints(kpt1)
    k2 = normalize_keypoints(kpt2)

    return np.mean(np.linalg.norm(k1 - k2, axis=1))
```

● ดีความผลลัพธ์

หากค่าระยะห่างเฉลี่ยมีค่าน้อย แสดงว่าท่าทางของทั้งสองภาพใกล้เคียงกันมาก ระบบจะตีความว่าภาพนั้นเป็น มิมที่ตรงกัน หรือมีท่าทางคล้ายกับภาพต้นแบบ

การใช้วิธีนี้ช่วยให้การเปรียบเทียบท่าทางมีความแม่นยำสูงขึ้น เนื่องจากคำนึงถึงการปรับสเกลและตำแหน่งของ keypoints ก่อนการวัดความคล้ายกัน อีกทั้งสามารถจัดอันดับความคล้ายของหลายภาพได้อย่างมีประสิทธิภาพ

```
for test_idx, test_kpt in enumerate(test_people):
    for meme_idx, meme_img in enumerate(meme_array):
        count=0
        for meme_person_idx, meme_kpt in enumerate(meme_img):
            count +=1
            score = pose_distance_norm(test_kpt, meme_kpt)
            if(count > 9):
                continue
            results.append({
                "score": score,
                "meme": meme_idx,
                "meme_person": meme_person_idx,
                "test_person": test_idx
            })
```

ทดสอบและประเมินผล (Model Evaluation)

หลังการฝึก จะทำการทดสอบด้วยข้อมูลที่โมเดลไม่เคยเห็น

- ตรวจจับความแม่นยำของ keypoints
- ดูค่า mAP ของ pose estimation
- ตรวจสอบว่าตรวจจับท่าทางของภาพ meme และภาพผู้ใช้ได้ถูกต้องหรือไม่

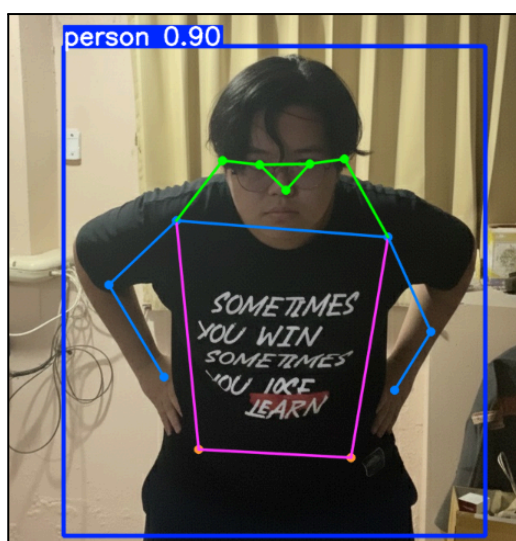
นำโมเดลไปใช้งานในการค้นหา Meme (Deployment)

โมเดลที่ผ่านการฝึกจะถูกนำไปใช้ในระบบเพื่อ

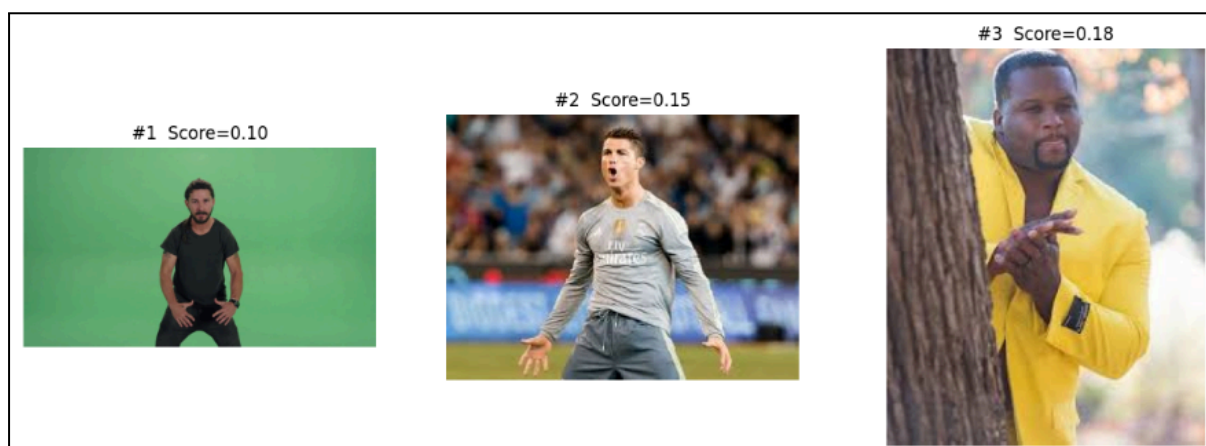
- ตรวจจับ keypoints จากภาพผู้ใช้

- เปรียบเทียบโครงสร้าง skeleton กับฐานข้อมูลมีม
- แสดงมีมที่มีท่าทางคล้ายที่สุดออกมา

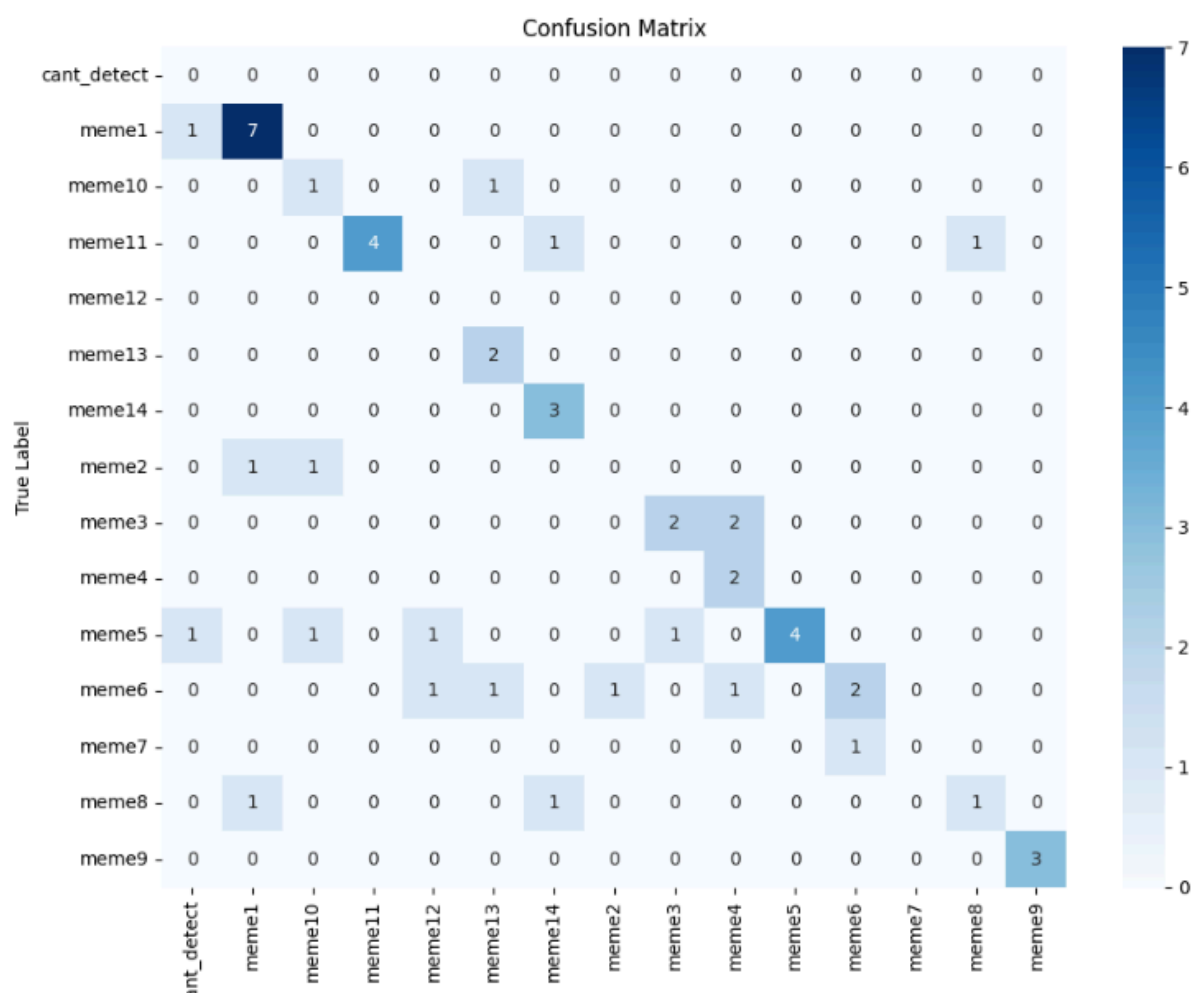
รูป Input



รูป Output



ความแม่นยำในการทำนายของModel



จากการทดสอบรูป ทั้ง 50 ครั้ง จะสังเกตได้ว่าโมเดลของเราอาจจะมีความไม่ถูกต้องในการ predict อยู่บ้าง แต่ก็สามารถใช้งานได้

Accuracy เท่ากับ 62.00 %

Precision เท่ากับ 49.63 %

Recall เท่ากับ 51.39 %

F1-score เท่ากับ 47.29 %

ข้อจำกัดของโปรแกรม

การจำกัดขอบเขตการตรวจสอบโครงกระดูก

เนื่องจากภาพมีม (Meme) ส่วนใหญ่แสดงเพียงช่วงลำตัวส่วนบน ทำให้การตรวจจับโครงกระดูกของร่างกายส่วนล่าง เช่น ช่วงเอวลงไป มีความคลาดเคลื่อนค่อนข้างสูง เพื่อลดความผิดพลาดในการเปรียบเทียบท่าทาง (Pose) ระบบจึงกำหนดให้ประมวลผลเฉพาะโครงกระดูกตั้งแต่ช่วงเอวขึ้นไปเท่านั้น วิธีนี้ช่วยเพิ่มความแม่นยำในการวิเคราะห์ท่าทางของบุคคลในภาพ และลดผลกระทบจากข้อมูลที่ไม่จำเป็นหรือมีความไม่สมบูรณ์

ข้อจำกัดของชุดข้อมูลมีมภายในระบบ

ระบบในปัจจุบันรองรับเฉพาะภาพมีมจำนวน 10 รูป ซึ่งเป็นข้อมูลที่เตรียมไว้ล่วงหน้า ทำให้ความหลากหลายของท่าทางและรูปแบบของมีมที่ระบบสามารถตรวจจับได้ยังมีข้อจำกัด ส่งผลให้การเปรียบเทียบท่าทางอาจครอบคลุมได้ไม่กว้างเท่าที่ควร หากต้องการรองรับความหลากหลายที่มากขึ้น จำเป็นต้องเพิ่มจำนวนชุดข้อมูลหรือขยายฐานข้อมูลมีมในอนาคต

ความแม่นยำในการเปรียบเทียบผลลงในกรณีที่มีหลายบุคคลในภาพ

ประสิทธิภาพของระบบจะลดลงอย่างมีนัยสำคัญเมื่อทำงานกับภาพที่มีบุคคลหลายคน เนื่องจากโมเดลอาจไม่สามารถระบุได้อย่างถูกต้องว่าควรนำโครงกระดูกของบุคคลใดไปใช้ในการเปรียบเทียบกับภาพมีม อีกทั้งระบบไม่สามารถตรวจจับวัตถุที่ไม่ใช่มนุษย์ เช่น หุ่น รูปปั้น หรือสัตว์ ส่งผลให้การตีความข้อมูลมีโอกาสผิดพลาดในกรณีที่ภาพมีองค์ประกอบที่คล้ายร่างกายมนุษย์แต่ไม่ใช่บุคคลจริง

บทสรุปของโครงการ

โครงการ MEME 4 ME เป็นการประยุกต์ใช้งานเทคนิค Image Processing และ Machine learning เพื่อพัฒนาระบบที่สามารถวิเคราะห์ท่าทางของมนุษย์จากภาพนิ่ง โดยใช้โมเดล Human Pose Estimation ในการตรวจจับจุดสำคัญของร่างกายและนำข้อมูลดังกล่าวมาเปรียบเทียบกับฐานข้อมูลรูปมีมเพื่อค้นหามีมที่มีท่าทางใกล้เคียงกันมากที่สุด ระบบสามารถช่วยให้ผู้ใช้ค้นหามีมที่ตรงกับความต้องการได้อย่างสะดวก รวดเร็ว และแม่นยำยิ่งขึ้น

ตลอดกระบวนการดำเนินงาน ทีมผู้พัฒนาได้ทำการเตรียมข้อมูลสำหรับการเทรนโมเดล ตั้งค่าระบบ และทดสอบการทำงานของโมเดล YOLO11n ซึ่งเหมาะสมกับลักษณะงานที่ต้องการความรวดเร็ว และมีข้อจำกัดด้านทรัพยากร ผลการทดลองพบว่าโมเดลสามารถตรวจจับ keypoints ได้อย่างมีประสิทธิภาพเพียงพอต่อการใช้งานในบริบทของโครงการนี้ นอกจากนี้ได้มีการออกแบบกระบวนการคำนวณหาความคล้ายของท่าทาง เพื่อจัดอันดับและแสดงผลมีมที่เหมาะสมต่อผู้ใช้

ในการทดลองเทรนโมเดลตรวจจับโครงกระดูกมนุษย์ด้วย YOLO11n-Pose และชุดข้อมูลที่จัดเตรียมขึ้น โมเดลได้ผ่านขั้นตอนการประเมินผล (Validation) บนชุดข้อมูลทดสอบ (Validation Set) โดยพิจารณาค่าความถูกต้องจากตัวชี้วัดมาตรฐาน

ค่าประเมินความแม่นยำของโมเดล

mAP@0.5 0.731

mAP@0.5:0.95 0.345

อย่างไรก็ตาม ระบบยังมีข้อจำกัดบางประการ เช่น ความแม่นยำที่ลดลงเมื่อภาพมีหลายบุคคล การรองรับมีมเพียง 10 รูปในฐานข้อมูล และประสิทธิภาพที่อาจลดลงในกรณีที่ท่าทางของร่างกายส่วนล่างไม่สามารถตรวจจับได้ชัดเจน ถึงแม้จะมีข้อจำกัดดังกล่าว โครงการนี้สามารถแสดงให้เห็นถึงความเป็นไปได้ของการนำ Human Pose Estimation มาใช้เพื่อค้นหาเนื้อหาภาพตามท่าทางได้อย่างมีประสิทธิภาพ

โดยสรุป โครงการ MEME 4 ME เป็นก้าวสำคัญในการผสมผสานความรู้ด้าน Image Processing, Machine Learning และการออกแบบระบบ เพื่อสร้างเครื่องมือค้นหามีมจากท่าทางของผู้ใช้ได้อย่างสร้างสรรค์และเป็นประโยชน์ ทั้งยังเป็นแนวทางสำหรับการพัฒนาต่อยอดในอนาคต เช่น การเพิ่มความ

หลากหลายของชุดข้อมูล การปรับปรุงโมเดลให้แม่นยำขึ้น และการนำเสนอฟีเจอร์แบบเรียลไทม์ เพื่อยกระดับประสบการณ์ของผู้ใช้ให้ดียิ่งขึ้น

การต่อยอดในอนาคต (Future Work)

เพิ่มความหลากหลายของมีมและท่าทาง

- ขยายฐานข้อมูลมีมให้ครอบคลุมท่าทางและอารมณ์ที่หลากหลายมากขึ้น
- รองรับมีมจากหลายวัฒนธรรมและหลายประเภทของสื่อ เช่น วิดีโอสั้น GIF หรือ AR Meme

พัฒนาความแม่นยำของระบบ

- ปรับปรุงโมเดล Human Pose Estimation ให้ตรวจจับ keypoints ในสภาพแวดล้อมที่ซับซ้อนหรือมุมกล้องที่หลากหลายได้ดีขึ้น
- ใช้เทคนิค Deep Learning เพิ่มความสามารถในการจับอารมณ์และบริบทจากท่าทาง

การโต้ตอบและประสบการณ์ผู้ใช้

- พัฒนาฟีเจอร์แนะนำมีมโดยอัตโนมัติจากภาพหรือวิดีโอของผู้ใช้
- สร้างระบบโต้ตอบแบบเรียลไทม์ เช่น แอปมือถือที่ผู้ใช้ถ่ายรูปแล้วระบบแนะนำมีมทันที