

## สรุปการทำงาน Week 7

### การพัฒนาโมเดล Action Recognition ด้วย CNN-LSTM

#### 1. การเตรียม Dataset

- 1.1 ทำการแยกคลิปวิดีโอออกเป็นภาพนิ่ง (frames)
- 1.2 ใช้ฟังก์ชัน `build_sequences()` เพื่อสร้างชุดข้อมูลภาพต่อเนื่อง (sequence) ความยาว 15 เฟรม โดยขยับทีละ 5 เฟรม
- 1.3 ใช้ฟังก์ชัน `explore_dataset_structure()` เพื่อตรวจสอบโครงสร้างและจำนวน sequence ในแต่ละ class

#### เหตุผล:

เพื่อจัดเตรียมข้อมูลให้อยู่ในรูปแบบที่เหมาะสมกับโมเดลแบบ sequence (LSTM) และสามารถควบคุมความยาวของลำดับ (sequence length) ให้สม่ำเสมอ ลดปัญหาข้อมูลขาดหาย

#### 2. Dataset Loader

ใช้คลาส `SequenceDataset` สำหรับโหลดข้อมูล sequence:

- Input: [T, C, H, W] (ลำดับของภาพ)
- Output: (video\_tensor, label)
- รองรับการ padding frame ถ้า sequence ไม่ครบ 15 เฟรม

ใช้ `torchvision.transforms` ในการ resize และแปลงภาพเป็น Tensor

#### เหตุผล:

PyTorch ต้องการข้อมูลลำดับที่มีขนาดเท่ากันในแต่ละ batch และต้องอยู่ใน Tensor format จึงต้องเตรียมและแปลงข้อมูลไว้ล่วงหน้า รวมถึงใช้ transform เพื่อให้ขนาดภาพเหมาะกับ CNN (ResNet ต้องการ 224x224)

### 3. สถาปัตยกรรมโมเดล

โมเดล CNNLSTMActionModel ประกอบด้วย:

- **CNN (ResNet18):** สำหรับดึง feature จากแต่ละภาพใน sequence
- **LSTM:** เรียนรู้ความสัมพันธ์เชิงเวลา (temporal patterns) จาก features ที่ดึงมาทั้ง sequence
- **Fully Connected Layer:** ทำนายคลาสของ action

Input: [B, T, C, H, W]

Output: [B, num\_classes]

เหตุผล:

- ResNet18 เป็น backbone ที่เบาแต่แม่นยำ ใช้งานง่ายสำหรับ dataset ขนาดกลาง
- LSTM เหมาะกับลำดับข้อมูลที่มีลักษณะต่อเนื่องเช่นการเคลื่อนไหวในวิดีโอ
- โครงสร้างนี้มีความยืดหยุ่นสูง ใช้กับ action recognition ได้อย่างมีประสิทธิภาพ

### 4. การเทรนโมเดล

- Optimizer: Adam (learning rate 1e-4)
- Loss: CrossEntropyLoss
- Batch size: 8
- Epochs: 10
- แบ่งข้อมูล train/validation เป็น 80/20
- ใช้ tqdm แสดง progress

ผลการเทรน:

- Train Accuracy: 98–100%
- Validation Accuracy: 100% ทุก epoch
- Loss มีขึ้นลง แต่ผลแม่นยำสูงมาก

เหตุผล:

Adam optimizer ให้การ converge ที่รวดเร็ว เหมาะกับงานที่มีจำนวนพารามิเตอร์ไม่มาก และ CrossEntropy เหมาะกับงาน classification หลายคลาส

## 5. ข้อสังเกต

- ความแม่นยำ validation สูงผิดปกติ อาจเกิดจาก:
  - ข้อมูล train/val คล้ายกันมากเกินไป
  - dataset ขนาดเล็ก หรือจำนวนคลาสน้อย
- Loss ยังไม่นิ่ง อาจเกิดจาก overfitting หรือมี noise บางส่วน