공 학 석 사 학 위 논 문

# 그래프 합성곱 LSTM과 HMM을 이용한 사람 행동 영상 분석




2021년 2월

부 경 대 학 교 대 학 원

인공지능융합학과

양 우 희

공 학 석 사 학 위 논 문

# 그래프 합성곱 LSTM과 HMM을 이용한 사람 행동 영상 분석

지도교수 신 봉 기

(공동지도교수 최 선 한)

이 논문을 공학석사 학위논문으로 제출함.

2021년 2월

부 경 대 학 교 대 학 원

인공지능융합학과

양 우 희

양우희의 공학석사 학위논문을 인준함.

2021 년 2 월 19 일




위 원 장 공학박사 송 하 주 (인)

위 원 공학박사 신 봉 기 (인)

위 원 공학박사 최 선 한 (인)

# 차례

**그림 차례** ..... **iii**

**표 차례** ..... **v**

**Abstract** ..... **vi**

**I. 서론** ..... **1**

1. 연구 배경 ..... 1
2. 연구 내용 및 구성 ..... 3

**II. 관련 연구** ..... **4**

1. 골격 영상을 이용한 행동 인식 ..... 4
2. 그래프 합성곱 네트워크 ..... 6

**III. 배경** ..... **8**

1. 그래프 합성곱 (Graph Convolution) ..... 8
2. 골격 구조 (Skeleton Construction) ..... 11
3. 부위별 그래프 합성곱 네트워크 (Part-based Graph Convolutional Network, PBGCN) ..... 13
4. 장단기 메모리 (Long Short Term Memory, LSTM) ..... 15
5. 은닉 마르코프 모델 (Hidden Markov Model, HMM) ..... 20

**IV. 제안 모델** ..... **24**

1. 기본 구조 ..... 24
   - 가. 입력 데이터 전처리 ..... 25
   - 나. PBGCN-LSTM(Part Based GCN-LSTM) 인코더 ..... 27
   - 다. 은닉 마르코프 모델 ..... 30
2. 주요 자세 요약 모델 ..... 31

**V. 실험 결과** ..... **32**

1. 데이터셋 ..... 32

2. 성능평가 및 시각화 .....34
가. NTU-RGB+D60: GCN-LSTM 인코더 행동 인식 성능 비교...34
나. NTU-RGB+D60: HMM 자세 분석 클러스터 시각화.....37
다. NTU-RGB+D60: HMM 자세 변환 시각화.....38
**VI. 결론 .....40**
**참고 문헌 .....41**

# 그림 차례

[그림 1] Kinect v2 로 추출된 한 프레임 내 골격 모형 ..... 11
[그림 2] 시공간 골격 모형 : 연속된 T 개 프레임 내 골격 모형 시퀀스 ..... 11
[그림 3] 영상 내에서 한 프레임 내 골격 모형 ..... 13
[그림 4] 나눠진 머리와 몸통 골격 ..... 14
[그림 5] 장단기 메모리 내부 ..... 15
[그림 6] 입력 게이트 내부 ..... 16
[그림 7] 망각 게이트 내부 ..... 17
[그림 8] 셀 상태 업데이트 ..... 18
[그림 9] 출력 게이트 내부 ..... 19
[그림 10] 은닉 마르코프 모델 ..... 20
[그림 11] 제안 모델 구조 ..... 24
[그림 12] 골격 데이터 원본 ..... 25
[그림 13] 프레임 내 주요관절과 거리 ..... 25
[그림 14] 연속된 두 프레임 사이 관절 간 위치 차 ..... 25
[그림 15] 제안된 PBGCN-LSTM 인코더 ..... 27
[그림 16] PBGCN 구조 ..... 28
[그림 17] Graph Convolution Layer 구조 ..... 28
[그림 18] 제안 모델에서 사용되는 4 개 골격 부위 ..... 29
[그림 19] 주요 자세 요약을 위한 PBGCN-LSTM 인코더 ..... 31
[그림 20] PBGCN-LSTM 을 이용하여 분류된 행동 군집 ..... 36
[그림 21] Pickup 행동에 대한 자세 군집 ..... 37
[그림 22] 피험자 20 번의 Pickup 자세 변화 ..... 38
[그림 23] 피험자 20 번의 Pickup 궤적 ..... 38

[그림 24] 피험자 20 번의 Pickup 행동 요약된 영상 .....38

# 표 차례

[표 1] NTU-RGB+D60 데이터셋 분리 기준 ..... 33
[표 2] NTU-RGB+D60 행동 종류 ..... 33
[표 3] NTU-RGB+D60 을 이용한 행동 인식 ..... 34

Human Activity Video Analysis using Graph Convolutional LSTM and HMM

Woohee Yang

Department of Artificial Intelligence Convergence,
The Graduate School, Pukyong National University

**Abstract**

We propose a part-based graph convolutional network and long short term memory(PBGCN-LSTM) encoder and hidden Markov model(HMM) model for activity analysis. PBGCN-LSTM is for extracting spatiotemporal features in a human skeleton activity video. HMM analyzes the spatiotemporal features for classifying postures in an activity.

First, PBGCN extracts spatial features in each frame. PBGCN divides human skeleton parts and extracts graph convolution features in a skeleton. The human skeleton can be represented as a graph, in which we view the nodes as the joints and the edges as the bones. And we give prior knowledge to GCN to divide parts. Second, LSTM extracts temporal features using the spatial features of each frame. Third, we obtain the spatiotemporal features represented as a 128 dimensional vector which is represented an activity video from PBGCN-LSTM. Finally, we input these vectors to HMMs. We train HMMs in each activity class.

We experiment PBGCN-LSTM and HMM with NTU-RGB+D60 dataset which provides the human skeleton activity video in 60 activity classes. We measure this model for action recognition tasks and show several visualizations for analyzing the postures in a video.

# I. 서론

## 1. 연구 배경

사람은 행동으로 말한다는 격언이 있을 만큼, 사람들은 각 행동에 의미를 두고 행동에 수 많은 정보를 담고 서로의 행동을 해석한다. 이런 생각은 컴퓨터 비전 분야에도 고스란히 적용되어 행동 인식, 행동 분석, 행동 예측 등 다양한 응용이 등장하였다.

예를 들어, 최근 자율 주행 자동차 산업이 부상하여 많은 연구자들이 자동차에 카메라와 기타 신호 장비를 부착하여 보행자 행동을 인식하거나 보행자 방향 예측을 다루고 있다. 또한, 개인을 위한 맞춤형 콘텐츠 서비스가 발전하며 스포츠 경기나 영화 장면 하이라이트 요약[1] 또는 사람들을 보호하기 위해 위험 행동 탐지 시스템 연구들[2]도 수행되고 있다. 이렇듯, 사람들은 서로의 행동을 눈으로 관찰하고 필요한 정보를 얻기 원하므로 컴퓨터를 이용하여 행동과 관련된 분야들을 발전시키고 있다.

컴퓨터 비전에서 행동 인식과 행동 분석은 어려운 일에 속한다. 사람은 다르 사람의 행동이 어떤 행동이고, 언제 시작했고, 언제 끝나는지를 눈으로 관찰하여 행동의 일부가 되는 현재 자세를 분석하여 알 수 있다. 하지만, 컴퓨터는 사람과 같이 빠르게 현재 프레임에서 나타나는 자세가 어떤 행동에 속하는지 판단하는 것과 동시에, 현재 자세가 해당 행동의 시작과 끝을 판단하기 어렵다. 그 이유는 컴퓨터가 한 행동을 판단하기 위해 일반적으로 두 가지 과정을 거치기 때문이다.

먼저, 컴퓨터는 영상 속의 각 프레임 내의 특징 추출 단계를 수행한다. 이 단계는 각 프레임 내에서 나타나는 사람들의 자세 형상의 특징을 파악하는 단계이다. 이전 연구들[22]에서는 사용자가 정의한 알고리즘에 기반하여 특징을

추출했고, 최근 심층 신경망 연구들[3]에서는 신경망이 경사하강법을 기반으로 특징에 대한 사용자 정의 없이 학습하여 각 자세들의 특징을 파악한다.

다음 단계로, 컴퓨터는 얻어진 각 프레임 당 자세 특징을 분석하여 행동을 인식한다. 인식은 크게 두 가지 방식으로 연구된다. 첫 번째는 입력 영상을 시계열 데이터로 보고 연속된 프레임 내의 특징 간의 연결성에 초점을 두고 분석하여 행동을 분류한다. 쉽게 말해, 연속된 두 프레임 내의 각 특징점들이 얼마나 변화하였는지를 학습하여 자세 특징 변화 흐름을 파악하는 방법이다. 두 번째는 각 프레임의 공간적 특징에 초점을 두는 방법이다. 이는 변화 보다는 각 프레임 내에서 추출된 자세 특징 자체에 관심을 두는 방법으로, 전체 영상에서 수집된 자세 특징들을 압축하여 분류한다. 예를 들어, 심층 신경망에서는 걷는 행동 영상의 모든 프레임에서 추출된 자세 특징들을 모두 수집하여 한 비선형 함수로 표현한다.

최근 많은 심층 신경망들은 특징 추출과 특징 분석을 동시에 진행한다. 따라서 행동 영상 전체를 보고 행동을 분류하므로, 현재 프레임에서 어떤 행동을 진행 중인지는 어렵다. 현재 프레임에서 어떤 행동인지 판단하기 위해서는 현재 프레임에서 나타나는 자세 특징이 어떤 행동에서 나타날 수 있는가를 추론해야 한다. 예를 들어, 보행자가 자율 주행 자동차 근처를 지나고 있는 상황이라 하자. 이 때 보행자가 어떤 이유로 보행 경로를 변경하여 위험할 수 있거나, 보행 도중 넘어지려는 상황이 발생할 수 있다. 이를 실시간으로 판단하기 위해서는 현재 보행자가 어떤 행동을 취하는 지를 프레임 단위로 분석해야 하므로 전체 프레임을 보고 특징 추출과 특징 분석을 동시에 진행하는 기존 방법은 어렵다.

따라서 본 논문에서는 프레임 단위 행동 분석을 위해 특징 추출과 특징 분석을 분리한 모델을 제안한다. 특징 추출 단계에서는 시간 및 공간 특징을 모두 고려하기 위해 프레임 단위로 자세 형태에 대한 공간 특징을 추출하고, 그를

입력으로 이용하여 연속된 프레임 간 시계열 특징을 추출한다. 마지막으로 추출된 시공간 특징들을 별개의 분석 모델에 입력으로 사용하여 프레임 단위 행동 분석을 수행한다.

## 2. 연구 내용 및 구성

본 논문에서는 영상 내 각 프레임 내 사람 자세에 대한 공간 및 시간 특징을 추출 후, 행동을 분류할 수 있는 시계열 특징 분석 모델을 제안한다. 우선, 공간 및 시간 공간 특징 추출은 그래프 합성곱 네트워크(Graph Convolutional Network, GCN)와 장단기 메모리(Long Short Term Memory, LSTM)를 결합한 그래프 합성곱 LSTM 모델을 사용한다. 공간 특징은 각 프레임 내의 사람의 자세 모형에 대한 특징을 일컫는다. 본 논문에서는 사람 자세를 뼈와 관절로 이루어진 골격(Skeleton) 데이터를 이용하여 표현한다. 골격 데이터는 정형화된 그래프이므로, 그래프 합성곱 네트워크를 이용하여 골격 모양에 대한 특징을 프레임 별로 추출한다. 시간 특징은 앞서 추출된 프레임 별 공간 특징들 간의 연속성을 파악하기 위해 장단기 기억 메모리를 사용한다. 그 후, 프레임 별로 추출된 공간 및 시간 특징들이 각 프레임에서는 어떤 행동의 어떤 자세를 나타내는 지 은닉 마르코프 모델(Hidden Markov Model, HMM)을 통해 한 자세를 한 상태로 표현하여 추론한다.

본 논문의 구성은 다음과 같다. 2 장에서는 제안 모델 설계를 위한 관련 연구를 소개한다. 시간 특징 추출을 위한 순환 신경망(Recurrent Neural Network, RNN) 변형 모델들, 공간 특징 추출을 위한 그래프 합성곱 네트워크 변형 모델들과 행동 분석 모델들을 살펴본다. 3 장에서는 그래프 합성곱 네트워크, 골격 그래프, 부위별 그래프 정의, 장단기 기억 메모리와 은닉 마르코프 모델에 대한 개념과 수식을 정의한다. 4 장에서는 입력 데이터 전처리 방법, 제안하는

부위별 그래프 합성곱 LSTM 인코더와 은닉 마르코프 모델을 결합한 모델에 대해 설명한다. 5 장에서는 제안 모델을 NTU-RGB+D60 데이터셋에 적용한 성능 평가와 시각화 결과를 설명하고, 6 장에서 제안 모델에 대한 결론을 짓는다.

# II. 관련 연구

컴퓨터 비전에서 영상을 이용한 행동 인식 연구들은 RGB 영상과 더불어 별도의 자세 추정 알고리즘을 통해 프레임 별로 사람 또는 사물의 골격 형상을 추출하여 사용한다. 이를 골격 영상이라 부른다. 골격 영상의 경우 배경 간섭 또는 잡음, 촬영 장비에 의한 잡음이 적고, 자세 추정 알고리즘을 통해 앞선 잡음들을 제거한 데이터를 사용하므로 RGB 영상 전체를 사용하는 것 보다 연산량이 적다. 그리고 RGB 영상을 사용하면 해상도에 따라 필요 저장 공간량이 기하급수 적으로 커질 수 있지만, 골격 영상을 사용하면 전자에 비해 적게 필요하다. 따라서, 본 논문에서는 골격 영상을 이용한 행동 인식과 행동 분석을 연구한다.

## 1. 골격 영상을 이용한 행동 인식

이전 연구들에서 골격 영상을 이용한 행동 인식은 크게 두 가지로 나뉜다. 첫 번째로, 골격 특징들을 추출 알고리즘에 의해 획득하고 특징점들을 군집화하거나 시계열 분석을 하는 방법과 두 번째로 심층신경망을 이용하여 골격 특징을 모델이 학습하는 하는 방법이 있다.

첫 번째 방법으로는 [4, 5] 등이 있다. [5]은 골격 데이터를 Lie group 내에서 회전하거나 이동시켜 특징으로 이용하였다. [4]은 장비로부터 얻어진 골격

데이터의 시간 특징과 공간 특징을 은닉 세미 마르코프 모델(Hidden Semi-Markov Model, HSMM) 변형을 이용하여 행동 인식을 하였다.

두 번째 방법인 심층 신경망 모델들은 골격 영상 데이터의 시계열 특징에 초점을 맞춘 순환 신경망 변형[6, 7, 8, 9]과 3 차원 텐서 입력으로 공간 및 시간 특징을 모두 고려하는 합성곱 네트워크를 변형한 모델들[10, 11, 12, 13, 14] 등이 제안되었다.

[6]은 기존 순환신경망의 입력 값과 은닉 값의 곱 연산을 하다마드 연산으로 바꿔 각 입력을 독립적으로 연산하여 기울기 소실(gradient vanishing)문제 해결 방안을 제시하였다. [7]은 골격 모형을 표현하고자 몸 부위를 분할하여 각 계층 순환신경망에서 학습하고 결과들을 결합하는 방식을 제안하였다. [8]은 기울기 소실 문제와 행동별로 수행하는 시간이 다른 점을 고려하여 각기 다른 크기의 윈도우를 이용하여 입력 데이터를 나누고 각각을 LSTM 모델에서 학습하여 결과를 결합하는 방법을 제안하였다.

합성곱 네트워크 변형 모델들인 [10]는 3 차원 잔차(residual) 합성곱 네트워크를 이용하여 한 영상 단위로 입력 데이터를 구성하여 학습하였다. [11]는 골격 데이터 인코딩을 위한 10 가지 시간 및 공간 이미지를 제안하였다. [12]은 여러 크기의 잔차 네트워크를 사용하여 골격 데이터에서 정보를 더 추출하는 방식을 제안하였다.

하지만, 이러한 모델들의 입력들은 문장, 이미지, 영상 데이터와 같이 1 차원 벡터 또는 정형화된 벡터나 텐서의 형태로 정의되었다. 따라서 이 모델들의 입력으로 비정형화된 그래프로 표현할 수 있는 골격 데이터는 표현해내는데 한계가 있다.

## 2. 그래프 합성곱 네트워크

그래프 합성곱 네트워크는 비정형화된 그래프로 표현되는 입력 데이터를 학습하기 위한 네트워크이다. 골격 기반 행동 인식에서 사람 골격은 관절과 두 이웃 관절들을 잇는 뼈로 구성 된다. 이는 관절을 노드, 뼈를 엣지로 정의하면 그래프로 표현할 수 있다. 그리고 골격 영상에서 행동 종류에 따라, 또는 사람들간의 간섭으로 인해 관절들이 겹쳐서 각 프레임마다 그래프 형상이 비정형화된다.

이전 순환 신경망이나 합성곱 네트워크의 변형 모델들은 1 차원 이상의 정형화된 벡터 또는 텐서를 입력으로 받았기 때문에 골격 그래프를 입력 데이터로 그대로 표현할 수 없었다. 따라서 골격 기반 행동 인식에서 그래프 합성곱 네트워크가 도입되었다.

그래프 합성곱 연산은 기존 합성곱 연산과 크게 다르지 않다. 기존 합성곱 연산과 다른 점은 입력 그래프는 인접 행렬(adjacency matrix)로 표현하고, 소셜 네트워크나 화합물 구조와 같이 이웃 노드 간의 관계가 그래프를 대표하는 정보가 되므로 이를 정의하는 수식이나 규정이 있다.

[18, 19, 20] 모델들이 행동 인식을 위한 그래프 합성곱 네트워크 모델들이다. [18]는 최초로 행동 인식에 그래프 합성곱 네트워크를 도입한 모델이다. 이는 이웃 노드를 정의하는 방법을 3 가지를 제시하였으며, 첫 번째는 각 노드 단독 사용, 두 번째는 바로 옆 노드 1 개만 이웃으로 정의, 세 번째는 현재 노드와 몸의 중심점과의 거리보다 가까운 노드 중 가장 가까운 노드와 반대로 가장 먼 거리에 있는 노드 2 개를 이웃으로 삼는 방법을 제시했다. 3 가지 방법 모두 임의의 그래프에서 관계를 학습시키는 방법으로는 효과적이었지만, 정해진 규칙에 따른 고정형 이웃을 얻는 다는 것이 단점이었다. 이는 곧 모델이 걷는 자세에서

오른 팔이 움직일 때, 왼쪽 다리가 같이 움직인다는 것을 학습하지 못한다는 것을 의미한다.

이를 해결하고자 [20]은 전체 그래프에서 각 노드들이 자신과 가장 연관성이 높은 관절을 찾는 인코더-디코더(encoder-decoder) 모듈을 도입하여 Actional-links(A-links)를 정의하고 [STGCN]에서 제시한 방법과 함께 사용하였다. 하지만, 이 모델도 두 노드들 간의 관계를 파악할 뿐 사람 골격에서 팔, 다리, 몸통, 머리처럼 같이 움직이는 부위의 의미를 퇴색시켰다.

[20]은 사람 골격을 2, 4, 6 개 부분으로 나누어 이웃을 정의하여 사람 몸의 부위를 명확하게 나눠주었다. 부위는 상체와 하체와 같이 각 개수 별로 나누어 모든 부분 그래프를 각각 학습한 결과를 추후에 합치는 방법을 제시하였다. 다시 말해, [20]은 [18, 19] 모델 등과 같이 입력이 임의의 그래프라고 정의한 모델들과 다르게 이미 알고있는 노드들 간의 확실한 관계를 사용할 수 있는 모델이다.

하지만, 그래프 합성곱 네트워크들 또한 3 차원 합성곱 네트워크에서 입력 데이터의 정의와 형태가 바뀐 것이고 실제 특징 추출 연산은 3 차원 합성곱을 이용하므로 프레임 단위로 시간 정보를 알 수 없다. 즉, 이 네트워크의 입력은 영상 단위로 필요하고 출력은 그 영상 하나 전체에 대한 행동 분류이다.

따라서 본 논문에서는 골격 영상 데이터를 이용한 행동 인식을 프레임 단위로 인식하고 분석하고자 그래프 합성곱 LSTM 인코더와 은닉 마르코프 모델을 결합한 모델을 제안한다. 먼저 제안 모델은 그래프 합성곱 네트워크에서 프레임 단위로 골격 그래프 형태 정보, 즉 공간 정보를 학습한다. 그 후, 프레임 단위 공간 정보를 LSTM 의 각 셀에 입력으로 넣고 각 셀의 출력 값 시퀀스를 시계열 정보로 얻는다. 마지막으로 얻어진 시계열 정보를 은닉 마르코프 모델의 입력으로 사용하여 골격 영상 속 행동을 분석한다.

-

# III. 배경

## 1. 그래프 합성곱 (Graph Convolution)

기존 합성곱 네트워크에서 합성곱 연산 수행 시 가중치 필터가 N 차원 정방형 텐서(tensor) 형태로 고정되어 있기 때문에 입력 데이터를 이에 맞춰 텐서로 정형화 시켜야 하다는 단점이 있다. 따라서 그래프와 같은 데이터 구조가 중요한 비정형 데이터에는 합성곱 연산이 적용되기 어려웠다. 이를 해결하고자 합성곱 연산을 일반화한 그래프 합성곱(Graph Convolution, GC)이 등장하였다.

일반적으로 2 차원 이미지 적용하는 합성곱 연산은 두 가지 과정이 있다. 첫 번째로, 입력 이미지보다 크기가 작은 합성곱 가중치 필터가 입력 이미지를 순서대로 훑어 연산할 부분을 추출한다. 두 번째로, 추출된 부분과 가중치 필터가 합성곱 연산을 한다. 그래프 합성곱 또한 두 가지 과정을 비정형 데이터에 두 단계를 동일하게 적용한다.

먼저, 그래프, $G = (V, E)$ 를 노드(node) 집합, $V = \{v_1, v_2, \dots, v_N\}$ (N : 모든 노드 개수)와 각 노드들을 잇는 엣지(edge) 집합, $E = \{e_1, e_2, \dots, e_M\}, E \subseteq (V \times V)$(M : 모든 엣지 개수) 로 정의하자. 여기서 각 노드들이 C 차원 특징 벡터를 가진다고 하면, 모든 노드 특징 벡터가 모인 노드 특징 행렬 X는 $X \in \mathbb{R}^{N \times C}$ 크기를 가진다.

이러한 그래프 G를 행렬로 표현한 것을 인접 행렬(Adjacency matrix)라 한다. 인접 행렬 A 의 원소 $A(i, j)$는 두 노드 $v_i$와 $v_j$사이에 엣지가 있다면 1, 아니면 0 으로 나타낸다. 수식으로 표현하면 다음과 같다.

$$A(i,j) = \begin{cases} 1 & if\ (v_i, v_j)\ \in\ E \\ 0 & otherwise \end{cases} \quad 1)$$

여기서 인접 행렬은 각 노드들의 특징 벡터를 반영하지 않으므로, 인접 행렬과 노드 특징 행렬을 곱하여 그래프 정보를 적용한다.

이때, 인접 행렬은 반드시 정규화하여 적용하여야 한다. 정규화 되지 않은 인접 행렬을 노드 특징 행렬과 곱한다면, 특징 벡터들의 크기가 인접 행렬 내 원소들의 크기 즉, 각 노드들 마다 연결된 이웃 노드 개수에 따라 변하게 된다. 따라서, 정규화된 인접 행렬이 필요하다. 그리고 각 노드들이 자신의 특징을 포함할 수 있게 자기 회귀(self-loop)도 추가한다. 이를 수식으로 표현하면 아래와 같다.

$$\tilde{A} = A + I \quad 2)$$

$$A^{norm} = D^{-\frac{1}{2}}\ddot{A}D^{-\frac{1}{2}} \quad 3)$$

$$D(i,i) = \sum_j A(i,j) \quad 4)$$

수식 3)의 행렬 $A^{norm}$이 정규화된 인접 행렬이다. D는 행렬 A의 차수 행렬(degree matrix)로 수식 4)와 같이 정의된다.

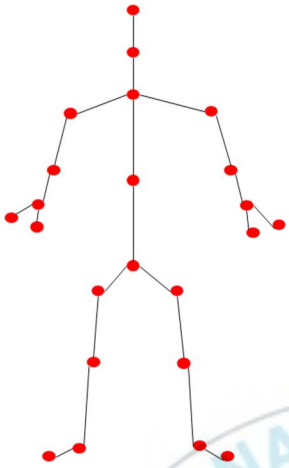
이제 그래프 합성곱의 첫 번째 과정으로 가중치 필터와 합성곱 연산을 할 그래프의 한 부분을 추출은 [17]의 그래프 라벨링 프로세스(Graph labeling process)를 사용한다. 이는 라벨 사상 함수(label mapping function) $\ell(v_i) : B(v_i) \rightarrow \{0, \dots k \dots, K-1\}$ 를 정의하여 한 노드 $v_i$와 연결된 이웃 노드 $v_j$, ($v_j \in V$)들의

집합을 $B(v_i)$라 할 때, 집합 $B(v_i)$ 내 $v_i$를 포함한 이웃 노드 $v_j$들을 고정된 K 개 부분 집합으로 나누고 라벨을 붙인다.

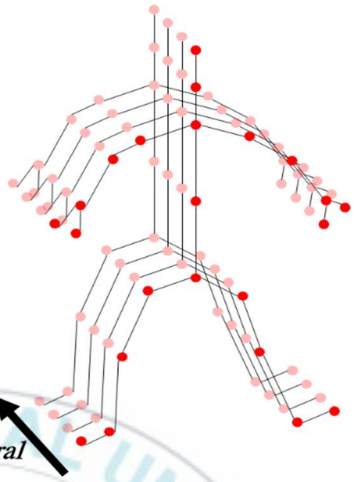
따라서, 라벨 사상 함수 $\ell$ 이 2 차원 합성곱 가중치 필터가 고정된 크기의 입력 이미지 부분을 추출하듯이, 그래프에서 각 노드들과 이웃들을 고정된 개수로 부분 추출을 가능하게 해준다. 결과적으로, 라벨 사상 함수 $\ell$ 에 의해 부분 추출된 부분 그래프들과 합성곱 가중치 필터들 간의 그래프 합성곱 연산은 다음과 같이 정의된다.

$$Y(v_i) = \sum_{v_j \in B(v_i)} A^{norm} W\left(\ell(v_j)\right) X(v_j) \qquad 5)$$

## 2. 골격 구조 (Skeleton Construction)



[그림 2] Kinect v2 로 추출된 한 프레임 내 골격 모형




[그림 1] 시공간 골격 모형 : 연속된 T 개 프레임 내 골격 모형 시퀀스

그림 1 과 같이 골격 데이터는 관절들과 관절들을 잇는 뼈로 구성되어있다. 한 프레임 내 골격 모형의 관절들을 노드, 뼈들을 엣지라 보면, 전체 골격은 그래프 구조이다. 또한, 영상은 프레임 시퀀스 입력이므로 시공간 골격 모형은 그림 2 와 같이 나타난다.

두 관절을 잇는 간선은 두 프레임 간 관절들을 잇는 시간상 간선 $E_T$과 한 프레임 내 관절들을 잇는 공간상 간선 $E_S$, 두 가지로 표현된다. 모든 관절 개수를 N, 전체 영상 길이를 T 이라 할 때 골격 그래프는 아래 수식으로 정의된다.

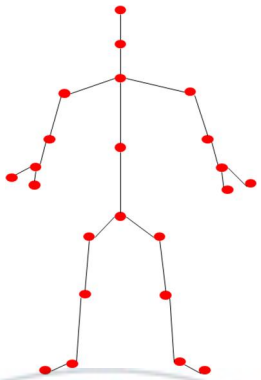
$$\boldsymbol{G} = (\boldsymbol{V}, \boldsymbol{E}) \qquad 6)$$

$$\boldsymbol{V} = \{\boldsymbol{v_1}, \boldsymbol{v_2}, \dots, \boldsymbol{v_N}\}$$

$$E_T = \{(v_{ti}, v_{qj}) | 0 \le t, q \le T, 0 \le i, j \le N, v_{ti}, v_{qj} \in V\}$$

$$E_S = \{(v_{ti}, v_{tj}) | 0 \le i, j \le N, v_{ti}, v_{tj} \in V\}$$

## 3. 부위별 그래프 합성곱 네트워크 (Part-based Graph Convolutional Network, PBGCN)



[그림 3] 영상 내에서 한 프레임 내 골격 모형

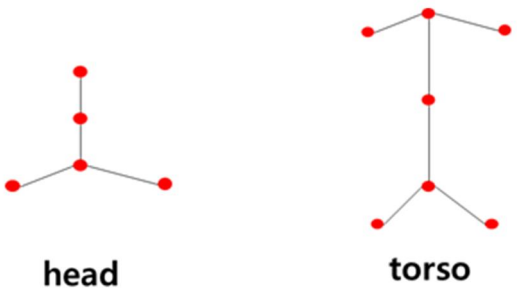
먼저 골격을 그래프 나타내고, 부위별 그래프 합성곱 출력과 최종적으로 네트워크 전체 출력을 수식적으로 정의한다. 그림 1 과 같이 골격 데이터는 관절들과 두 관절들을 잇는 뼈로 구성되어 있다. 한 프레임 내 골격 모형의 관절들 집합을 노드 집합 V, 뼈들의 집합을 엣지 집합 E 라 정의하면, 전체 골격은 비방향성 그래프 G라 할 수 있다. 또한, N 개의 관절은 C 차원으로 표현된 각 노드 위치 벡터를 가지고, 이 벡터들의 모음을 노드 특징 행렬 $X \in \mathbb{R}^{Nx \times C}$이라 하자.

이러한 전체 골격을 설정된 K 개 부위로 나누어 부분 그래프 집합으로 표현해주면 수식 6 과 같다.

$$G = \bigcup_{p \in \{1,\ldots,K\}} \mathcal{P}_p \mid \mathcal{P}_p = (V_p, E_p) \qquad 7)$$

$\mathcal{P}_p$는 그래프 G 의 부분 그래프를 뜻하고, $V_p$와 $E_p$는 각각 부위 p 에 속하는 노드와 엣지 집합이다. 이 때, 부분 그래프 $\mathcal{P}_p$들은 같은 노드 또는 엣지를 공유할

수 있다. 예를 들어 아래 그림과 같이 전체 골격에서 머리 부분과 몸통 부분을 나누면, 어깨에 해당되는 노드와 엣지를 두 부위가 공유하게 된다.




[그림 4] 나눠진 머리와 몸통 골격

부위별 그래프 합성곱 네트워크는 이렇게 나눠진 부분 그래프들을 이용하여 각 부위별 그래프 합성곱 특징점 $Y_p(v_i)$ 들을 얻고, 집계 함수 $\mathrm{F_{agg}}$를 통해 부위별 정보를 합하여 출력을 얻는다. 따라서 부위별 합성곱 네트워크는 다음과 같이 수식적으로 정의된다.

$$Y_p(v_i) = \sum_{v_j \in B_{kp}(v_i)} A_p^{norm} W_p\left(\ell_{\mathrm{p}}(v_j)\right) X_p(v_j) \qquad 8)$$

$$\mathrm{p} \in \{1, \ldots, \mathrm{K}\}$$

$$Y(v_i) = \mathrm{F_{agg}}\left(Y_{p1}(v_i), Y_{p2}(v_i) \mid (p_1, p_2) \in \right. \qquad 9)$$

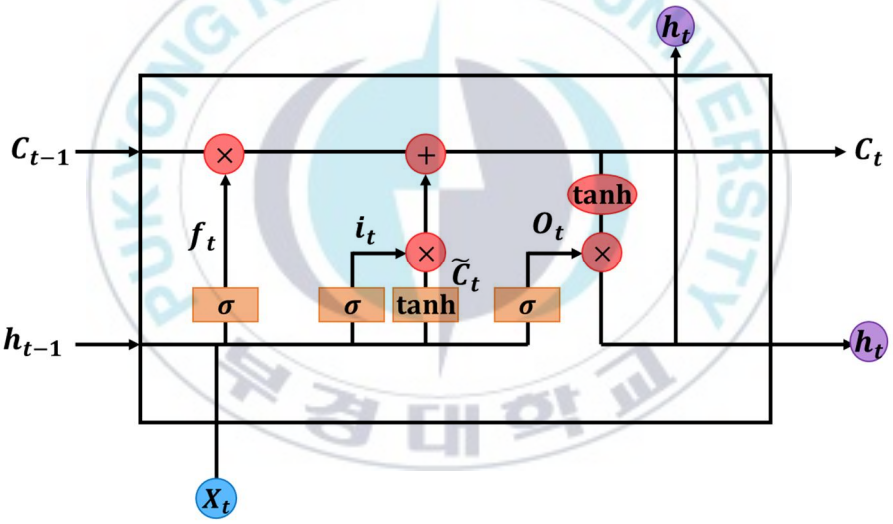
$$\left. \{1, \ldots, K\} x \{1, \ldots, K\}, p_1 \neq p_2\right)$$

수식 7은 각 부위별 그래프 합성곱 결과이고, 수식 8이 최종 부위별 그래프 합성곱 네트워크의 출력이다. 집계 함수 $\mathrm{F_{agg}}$는 임의로 정의할 수 있으며 이에

따라 전체 네트워크 성능이 달라질 수 있다. 본 논문에서는 집계 함수를 덧셈으로 정의하여 모든 부위별 그래프 합성곱 결과를 합하였다.

## 4. 장단기 메모리 (Long Short Term Memory, LSTM)

장단기 메모리[15]는 순환 신경망의 변형으로 기존 순환 신경망의 장기 의존성 문제(the problem of long-term dependency)를 해결하기 위해 제안되었다. 장기 의존성 문제란, 기존 순환 신경망에서 깊이가 깊어질 수 록 앞서 배웠던 정보를 충분히 뒤로 전달하지 못하는 문제를 말한다. 이를 해결하고자 장단기 메모리는 셀 상태(cell state)를 추가하여 장단기 정보 전달을 조절하였다.

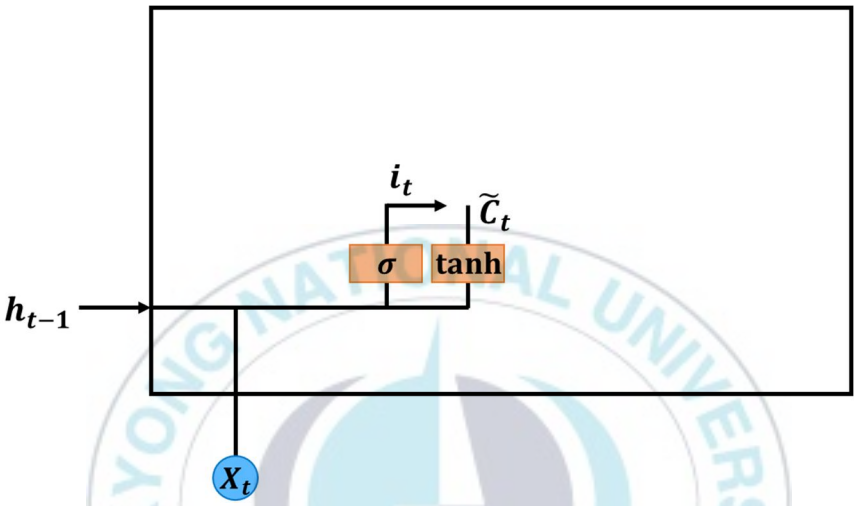



[그림 5] 장단기 메모리 내부

그림 3 은 장단기 메모리의 내부 모습이다. 장기 의존성 문제 해결을 위해 셀에 입력 게이트, 망각 게이트, 출력 게이트를 추가하여, 입력 게이트로는 다음 셀로 넘겨줄 정보를 추려내고, 망각 게이트로는 불필요한 정보 삭제를 수행한다. 그리고 두 게이트 정보를 셀 상태에서 합하여 장기 기억을 만들고, 단기

기억으로는 출력 상태와 셀 상태를 조합한 은닉 상태를 사용한다. 입력, 망각, 출력 3 가지 게이트 모두 시그모이드(sigmoid) 함수를 지나 0 과 1 사이의 값으로 각 정보 전달량을 조절한다.

(1) 입력 게이트 (Input gate)




[그림 6] 입력 게이트 내부

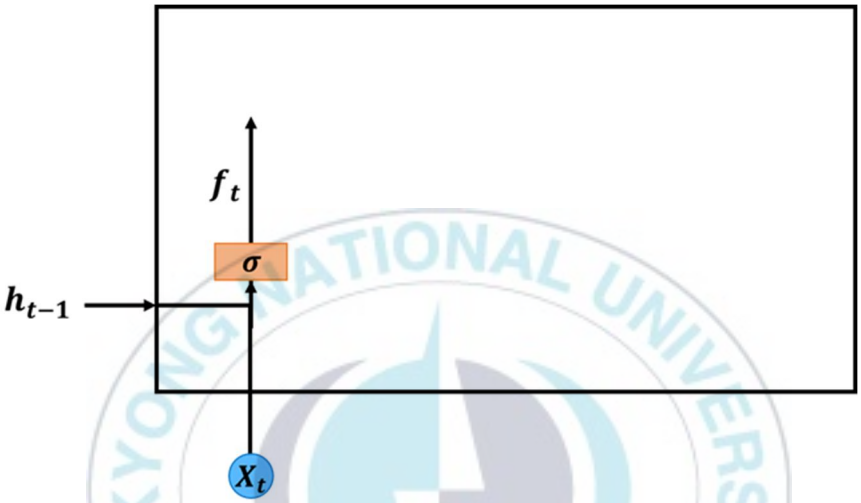
$$i_i = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad 10)$$

$$\tilde{C}_t = \tanh(W_{x\tilde{c}}x_t + W_{h\tilde{c}}h_{t-1} + b_{\tilde{c}})$$

입력 게이트는 전달 받은 정보 중 기억할 정보를 선택하는 게이트이다. 수식 9 에서 보듯, 현재 t 시각의 입력$\boldsymbol{x}$ 값과 입력 게이트에 이어지는 가중치 $\boldsymbol{W_{xi}}$ 를 곱한 값과 이전 t-1 시각 은닉 상태와 동일한 과정으로 이어진 가중치 $\boldsymbol{W_{h\tilde{c}}}$ 를 곱한 값을 더하고 시그모이드 함수를 지나면, 0 과 1 사이의 값을 가지는 $\boldsymbol{i_t}$ 가 된다. $\tilde{\boldsymbol{c}}_{\boldsymbol{t}}$ 도 비슷한 과정으로 얻고 대신 하이퍼볼릭 탄젠트(hyperbolic tangent)

함수를 지나므로 -1 과 1 사이 값을 가진다. 이 두 값으로 셀 상태에서 다음 상태로 넘겨줄 정보를 결정하게 된다.

(2) 망각 게이트 (Forget gate)

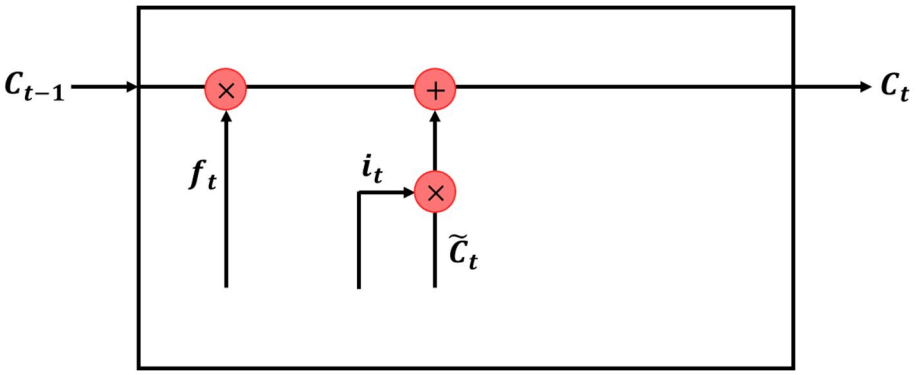



[그림 7] 망각 게이트 내부

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f) \quad 11)$$

망각 게이트는 입력 게이트와 반대로 현재 정보 중 버려야할 정보를 선택하는 게이트이다. 현재 t 시각 x 값과 이전 t-1 시각 은닉 상태 정보가 시그모이드 함수를 거쳐 0 과 1 사이 값으로 출력된다. 이 출력 값이 삭제된 정보량을 나타내며, 0 에 가까울수록 망각된 정보량이 많고 1 에 가까울수록 정보가 온전히 다음 셀로 전달된다.

(3) 셀 상태 (Cell state)

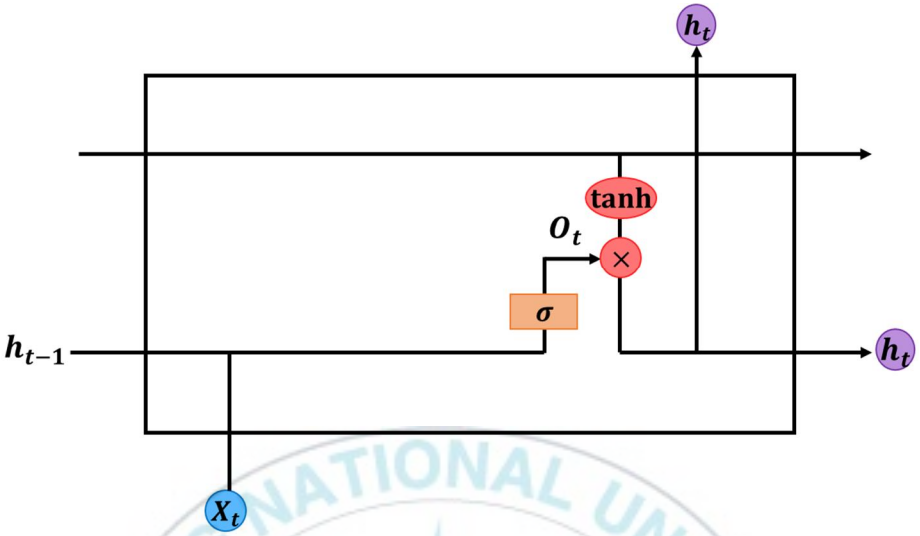



[그림 8] 셀 상태 업데이트

$$C_t = f_t \circ C_{t-1} + i_t \circ \tilde{c}_t \qquad 12)$$

셀 상태는 장기 기억을 보관하는 상태로 입력 게이트 출력 값과 망각 게이트 출력 값의 조합으로 계산된다. 먼저, 입력 게이트 출력인 $\boldsymbol{i_t}$ 와 $\tilde{c}_t$를 원소별 곱(element-wise product) 연산 한다. $\boldsymbol{i_t}$가 0 과 1 사이의 값을 가지므로 현재 t 시각 기억할 정보가 선택된다. 다음으로, 망각 게이트 출력 값도 마찬가지로 0 과 1 사이의 값을 가지고 이전 장기 기억과 원소별 곱을 하므로, 이전 장기 기억 정보를 잊어버릴 수 있도록 한다. 따라서, 선택된 이전 장기 기억과 현재 정보는 더한 형태로 새로운 장기 기억을 생성하여 셀 상태에 저장된다.

(4) 출력 게이트 (Output gate)



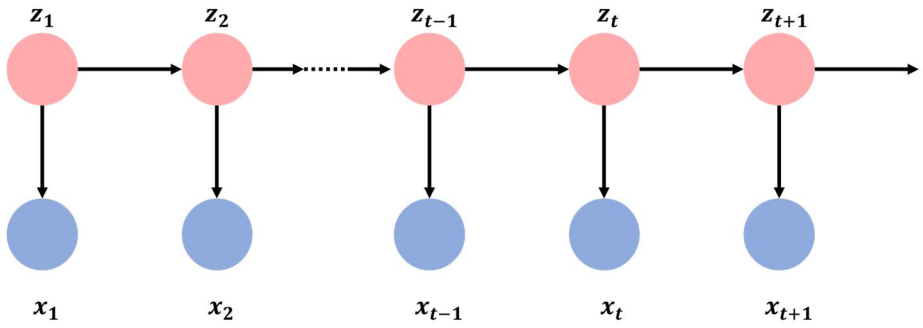

[그림 9] 출력 게이트 내부

$$o_i = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \qquad 13)$$

$$h_t = \mathrm{o_t} \circ tanh(C_t)$$

출력 게이트는 현재 시각 t 의 은닉 상태를 연산하는데 쓰인다. 은닉 상태는 장단기 메모리에서 단기 상태 라고도 한다. 은닉 상태는 -1 과 1 사이의 값으로 출력되는 현재 시각 t 의 장기 상태와 시그모이드를 거쳐 0 과 1 사이의 값인 출력 상태의 원소별 곱이다. 따라서, 이는 현재 기억하고 싶은 단기 기억을 출력층으로 보내 출력하거나 다음 셀에 은닉 상태로 입력으로 주어진다.

## 5. 은닉 마르코프 모델 (Hidden Markov Model, HMM)




[그림 10] 은닉 마르코프 모델

은닉 마르코프 모델은 1960 년대 후반 바움(Leonard E. Baum)이 은닉 마르코프 모델의 은닉 상태를 계산해내는 바움-웰치(Baum-Welch) 알고리즘을 제안하고, 앤드루 비터비(Andre Viterbi)가 동적프로그래밍을 이용하여 관측 값들을 가장 잘 설명하는 은닉 상태 순서를 계산해내는 비터비 알고리즘을 제안하게 되어 활발히 논의 되었다. 그리고 1970 년대부터 음성인식[16]에 응용되기 시작하여 문장 분석, DNA 염기 서열 분석, 행동 인식등 다양한 분야에서 응용되고 있다.

이 모델은 마르코프 연쇄(Markov chain)를 기반으로 관측 시퀀스 $X = \{x_1, x_2 \dots, x_t, \dots, x_T\}$ 의 각 관측 값이 영향을 받는 은닉 상태(state) $z_t$ 를 추론하는 모델이다. 마르코프 연쇄란 마르코프 성질을 지닌 유한한 N 개의 상태 집합$S = \{S_1, S_2, \dots, S_N\}$과 이들을 연결하는 상태 천이(state trasition) 집합으로 구성된 네트워크를 말한다. 이 네트워크는 마르코프 성질로 인해 시각 t+1 의 상태는 바로 직전 시각 t 또는 일정 n 시각 이전의 상태에서만 영향을 받는다. 이를 n 차 마르코프 가정이라 하고, 이 때, 현재 상태가 이전 상태에만 영향을 받는 1 차

마르코프 가정이고, 각 상태가 확률 변수로 표현될 때, 다음과 같이 확률 시퀀스로 정의한다.

$$\mathrm{P}(\mathrm{z_{t+1}} = \mathrm{S_i}\,|z_0 = S_q, z_1 = S_w, \dots, z_t = S_j) = P(z_{t+1} = \mathrm{S_i}|\mathrm{z_t} = S_j) \quad 14)$$

이렇듯 현재 상태가 이전 상태의 영향을 받아 상태가 결정되는 과정을 상태 천이라고 한다. 이는 확률 분포 A 로 정의되고 A 를 상태 천이 확률 분포(state transition probability distribution)라 부른다.

$$\mathbf{A} = \begin{pmatrix} \boldsymbol{a_{11}} & \cdots & \boldsymbol{a_{1N}} \\ \vdots & \ddots & \vdots \\ \boldsymbol{a_{N1}} & \cdots & \boldsymbol{a_{NN}} \end{pmatrix} = \{\boldsymbol{a_{ij}}\} \quad 15)$$

$$\boldsymbol{a_{ij}} = \boldsymbol{P(z_{t+1} = S_i\,|z_t = S_j)} \quad 16)$$

$$\sum_j a_{ij} = 1$$

$$(1 \le \mathrm{i,j} \le \mathrm{N}\,;\, 0 \le\ a_{ij} \le 1)$$

이 때, 은닉 상태 시퀀스에서 첫 번째 은닉 상태는 영향을 받을 이전 상태가 없으므로 초기 상태 확률 분포(initial state probability distribution), π 에 의해 생성된다.

$$\pi_i = P(z_1 = S_i) \quad 17)$$

$$\sum_i \pi_i = 1$$

$$(\mathbf{1 \le i \le N}, \mathbf{1 \le\ \pi_i \le 0})$$

상태 천이 행렬을 통해 마르코프 가정을 내포한 상태들은 각 시각의 관측 값에 영향을 준다. 예를 들어, t 시각에서 어떤 심볼 $\mathrm{V}_k$ 가 관측되었다고 하자. 그리고 시퀀스 내 심볼들 중 $\mathrm{V_k}$를 관측할 관측 심볼 확률 분포(observation symbol probability distribution) B 는 다음과 같다.

$$\mathrm{B} = \begin{pmatrix} b_{11} & \cdots & b_{1M} \\ \vdots & \ddots & \vdots \\ b_{M1} & \cdots & b_{MM} \end{pmatrix} = \{b(V_k)\} \quad 18)$$

이때, 확률 $\mathrm{b}(\boldsymbol{V_k})$가 은닉 상태 $a_{ij}$의 영향을 받았다면 다음과 같이 정의된다.

$$b_{ij}(V_k) = P(V_k | z_t = S_i, z_{t+1} = S_j) \quad 19)$$

$$\sum\nolimits_k b_{ij}(V_k) = 1$$

$$(1 \le \mathrm{k} \le \mathrm{K}\,; 0 \le b_{ij} \le 1)$$

따라서 은닉 마르코프 모델, $\Theta$ 는 다음과 같이 정의되고, 이 모델에 의해 전체 관측 시퀀스가 은닉 상태들의 영향을 받아 만들어질 결합 확률 분포(joint probability distribution)은 다음과 같다.

$$\theta = \{\pi, \mathrm{A}, \mathrm{B}\}$$

- N : 전체 상태 수

  상태 집합: $\mathrm{S} = \{\mathrm{S}_1, \mathrm{S}_2, \ldots, \mathrm{S}_\mathrm{N}\}$

  $\mathrm{z_t}$: 시각 t 에서 상태 확률 변수

- M : 전체 시퀀스에서 관측 가능한 심볼 수

  심볼 집합: $\mathrm{V} = \{\mathrm{V}_1, \mathrm{V}_2, \ldots, \mathrm{V}_\mathrm{M}\}$

$x_t$: 시각 t 에서 관측 값

- $\Pi = \pi_i$: 초기 상태 확률 분포(initial state probability distribution)

  $\pi_i = P(z_0 = S_i), \sum_i \pi_i = 1, (1 \leq i \leq N, 1 \leq \pi_i \leq 0)$

- $A = a_{ij}$: 상태 천이 확률 분포(state transition probability distribution)

  $a_{ij} = P(z_{t+1} = S_i \,|z_t = S_j), \sum_j a_{ij}\,, (1 \leq i, j \leq N, 0 \leq a_{ij} \leq 1)$

- $B = b_{ij}(V_k)$: 관측 확률 분포(observation symbol probability distribution)

  $b_{ij}(V_k) = P(x_t = V_k | z_t = S_i, z_{t+1} = S_j), \sum_k b_{ij}(V_k) = 1$

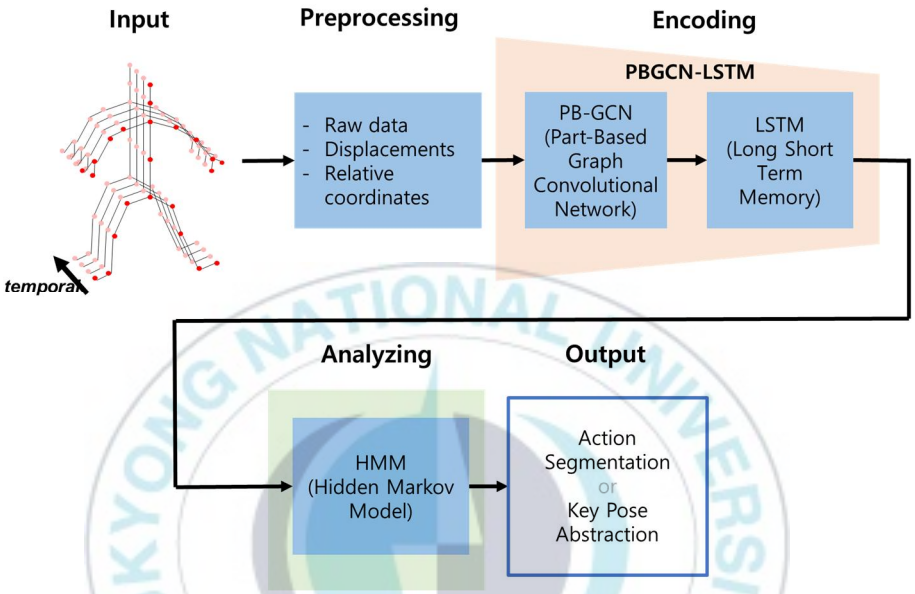
  $, (1 \leq k \leq K, 0 \leq b_{ij}(V_k) \leq 1)$

- 전체 시퀀스에서 결합 확률 분포(joint probability distribution)

$$P(X, Z|\Theta) = \prod_{i=1}^{N} \pi_i \left[ \prod_{t=2}^{T} \prod_{i=1}^{N} \prod_{j=1}^{N} a_{ij} \right] \prod_{t=1}^{T} \prod_{k=1}^{K} b_k(x_t) \qquad 20)$$

# IV. 제안 모델

## 1. 기본 구조

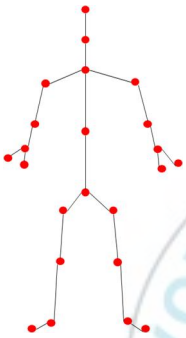



[그림 11] 제안 모델 구조

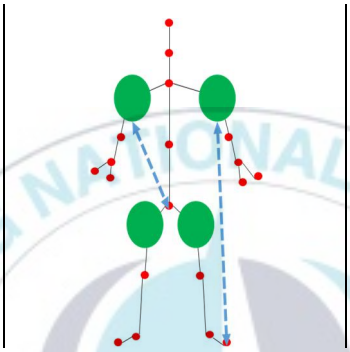
그림 1 은 본 논문의 제안 모델 구조도이다.. 먼저 카메라에서 얻어진 3 차원 골격 데이터가 입력으로 들어온다. 입력된 데이터는 전처리 단계에서 한 프레임 내 관절들 간 관계와 연속된 두 프레임 사이 두 관절의 차이를 계산한다. 그리하여 각 관절이 가지는 특징 벡터는 전처리 단계에서 계산된 두 특징들과 원본 3 차원 좌표를 결합 후 PBGCN-LSTM(Part-based GCN-LSTM) 인코더에 입력된다. PBGCN-LSTM 인코더는 프레임 단위로 입력된 결합 특징 벡터들에서 PBGCN 을 이용하여 공간 특징을 추출하고, LSTM 을 이용하여 프레임들 간 시간 특징을 순서대로 추출한다. 마지막으로, 인코더를 통해 추출된 특징 벡터들은 연속

마르코프 모델에 입력되어 비터비 알고리즘을 통해 각 프레임별로 현재 행동이나 자세를 추정한다. 따라서, 출력은 프레임별로 추정된 행동이나 자세, 즉 은닉 마르코프 모델에서 상태이고, 연속된 상태가 나타나는 구간을 한 행동이나 주요 자세라 정의한다. 단계별 자세한 설명은 다음 절부터 이어진다.

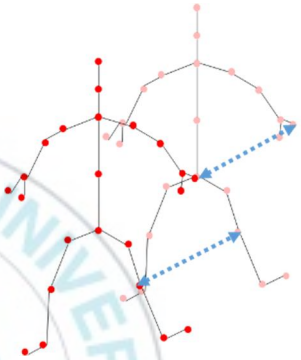
### 가. 입력 데이터 전처리



[그림 12] 골격 데이터 원본



[그림 13] 프레임 내 주요관절과 거리

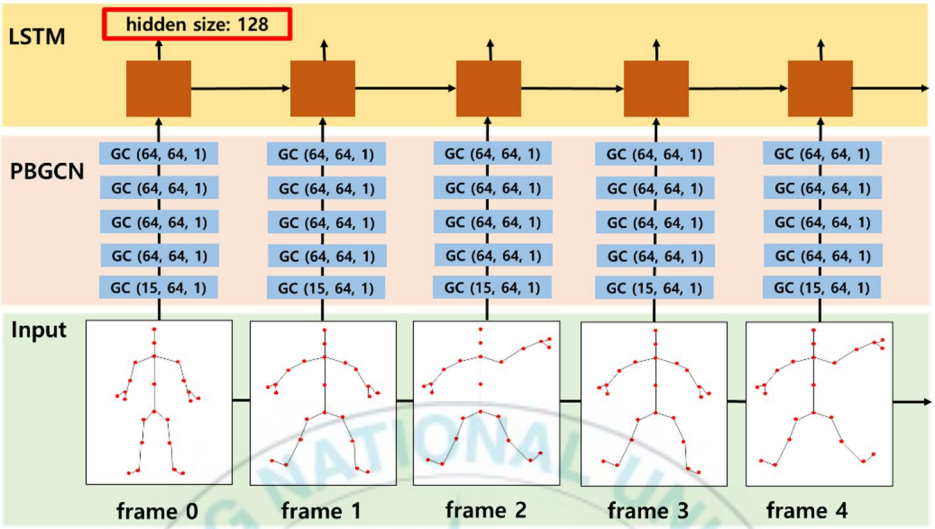


[그림 14] 연속된 두 프레임 사이 관절 간 위치 차

전처리 단계에서는 그림 2, 3, 4 와 같이 각 관절들의 특징점을 계산 후 결합하여 한 벡터로 표현한다. 먼저 그림 2 는 원본 데이터로, 각 관절들은 촬영에 사용된 카메라를 원점으로 삼아 얻어진 3 차원 좌표 x, y, z 를 가진다. 다음으로 그림 3 에서 볼 수 있 듯, 한 프레임 내에 각 부위를 대표하는 주요 관절 4 개와 각 관절들 간의 3 차원 거리 데이터이므로, 총 12 차원 벡터이다. 이는 같은 프레임 내에서 연산 되므로, 공간상 관절 정보라 부른다. 마지막으로 그림 4 와 같이 연속된 두 프레임 사이에서 각 관절들 간의 위치 차 데이터이다. 이 때 현재 프레임내 각 관절들은 다음 프레임 안에 자기 자신과 같은 자리 관절과 차이를

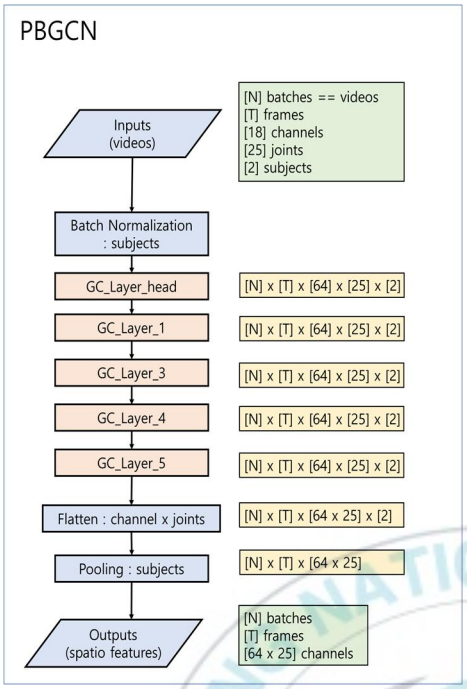
계산한다. 예를 들어, 시각 t 에서 1 번 관절(머리)은 시각 t+1 에서 1 번 관절(머리)에만 대응한다. 이 특징점은 서로 다른 시간 사이의 3 차원 위치 차 정보를 얻어내므로, 시간상 관절 정보라 부른다. 이렇게 전처리 단계에서 구해진 공간상 관절 정보, 12 차원 벡터와 시간상 관절 정보, 3 차원 벡터를 각 관절의 3 차원 위치 좌표와 결합하여 총 18 차원 벡터 형태로 골격 그래프의 각 노드 특징 벡터를 정의한다.

### 나. PBGCN-LSTM(Part Based GCN-LSTM) 인코더

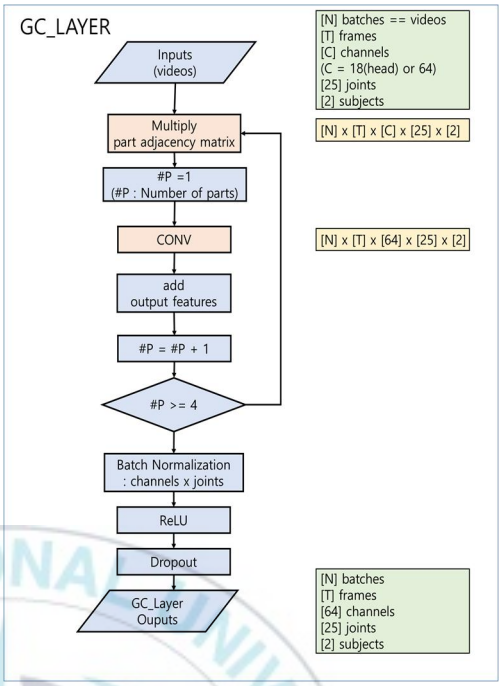



[그림 15] 제안된 PBGCN-LSTM 인코더 : Input 계층은 입력된 영상 내 각 프레임들을 나타낸다. PBGCN 과 LSTM 은 각각 PBGCN 와 LSTM 계층이다. PBGCN 계층의 경우 5 개의 그래프 합성곱 계층을 가진다.

PBGCN-LSTM 인코더는 그림 15 에서 볼 수 있듯, 골격 기반 영상에서 PBGCN 을 통해 각 프레임별로 공간상 특징점을 먼저 추출하고, 그 결과를 LSTM 의 입력으로 사용하여 시간상 특징점을 추출한다.




[그림 16] PBGCN 구조 : 왼쪽은 PBGCN 의 구조를 나타내고, 오른쪽은 각 계층의 출력 데이터 크기를 나타낸다.

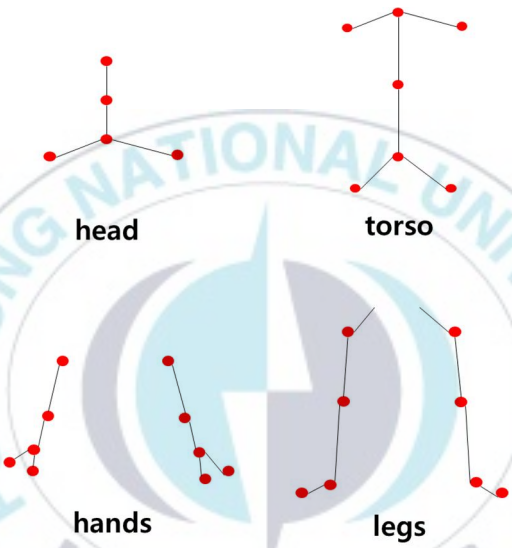



[그림 17] Graph Convolution Layer 구조 : 왼쪽은 그래프 합성곱 계층 구조를 나타내고, 오른쪽은 각 연산의 출력 데이터 크기를 나타낸다.

먼저, 제안된 부위별 그래프 합성곱 네트워크는 그림 4 와 같다. 이는 입력으로 배치 크기만큼 N 개 영상을 받고, 총 5 개의 그래프 합성곱 연산 계층과 1 개의 풀링 계층을 거쳐 출력으로 각 영상에서 공간 특징점을 추출한다.

입력되는 각 영상은 고정 길이의 T 개 프레임을 가지고 있다. 각 프레임에는 2 명 이하의 피험자가 등장하므로 골격 그래프는 최대 2 개를 포함한다. 각 골격 그래프는 25 개의 관절로 이루어져 있고, 각 관절은 원본 좌표, 공간 관절 정보, 시간 관절 정보로 이루어진 18 차원 노드 특징 벡터로 표현된다. 이때, 영상에 등장하는 피험자들이 항상 동일한 위치에서 행동하지 않기 때문에 배치 정규화를 통해 위치 차가 학습에 영향을 덜 미치도록 해준다.

다음으로, 그래프 합성곱 연산 계층은 그림 5 와 같다. 이 계층에서는 시간상 특징을 추출하지 않으므로, 노드 특징 벡터의 18 차원 데이터를 64 차원 데이터로 변환해주는 첫 번째, GC_Layer_head 를 제외한 각 그래프 합성곱 연산 계층에서 합성곱 필터 크기는 64x64x1 로 동일하다. 따라서 출력 데이터의 차원 크기는 항상 64 개로 고정되고 프레임 길이는 변하지 않는다.




[그림 18] 제안 모델에서 사용되는 4 개 골격 부위

또한, 그래프 합성곱 연산 시, 전체 골격을 나누어 부위별로 학습하기 위해 그림 6 과 같이 미리 정의된 4 개 부위별 인접 행렬을 노드 특징 행렬에 곱하여 합성곱 연산을 수행하고 그 결과들을 모두 합한다.

그 후, 골격 데이터에서 공간상 특징과 관련된 모든 관절들의 채널에 관해 배치 정규화를 거친다. 마지막으로 비선형성 부여와 과적합 방지를 위해 ReLU 와

드롭아웃 계층을 통과하여 출력으로 NxTx64x25x2 크기의 텐서를 얻는다. 이렇게 얻어진 출력 텐서들이 각 프레임별 공간상 특징이다.

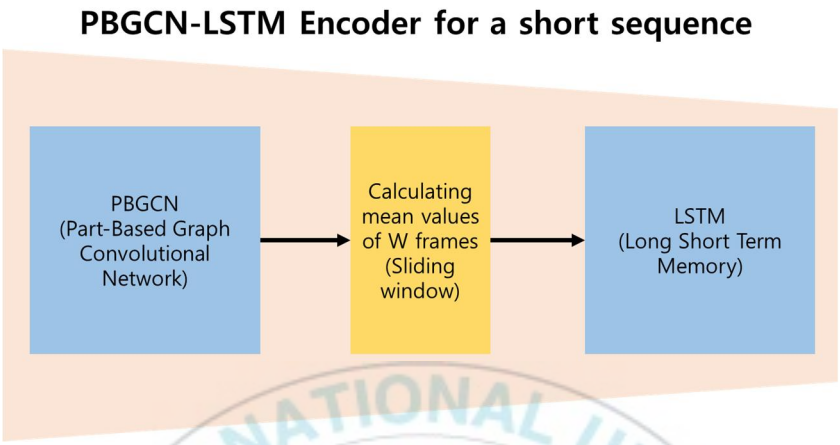
마지막으로, 부위별 그래프 합성곱 네트워크에서 추출된 각 프레임별 공간상 특징들을 LSTM 에 입력하기 위해 한 프레임 내에서 64x25 크기를 가지는 노드 특징 행렬을 벡터 형태로 펼쳐주고, 풀링 계층을 통해 두 피험자들 데이터를 평균 낸다.

그 후, LSTM 계층에서 입력으로 Nx(64x25)xT 크기 텐서를 받아 각 영상에서 프레임들 간 시간상 특징을 추출한다. 각 셀의 은닉 상태 크기는 128 차원으로 설정하여, 모든 셀의 출력 벡터를 모은 LSTM 의 최종 출력 데이터 크기는 Nx128xT 가 된다. 따라서 PBGCN 과 LSTM 을 거친 시공간 특징의 크기는 Nx128xT 가 되고, 이는 T 개 프레임을 가진 각 영상들의 각 프레임이 128 차원 벡터로 요약되었음을 나타낸다.

### 다. 은닉 마르코프 모델

은닉 마르코프 모델은 PBGCN-LSTM 인코더로 요약된 각 프레임 내 시공간 특징들에서 시계열성을 분석하기 위해서 사용된다. 한 영상 내 T 개 프레임들이 각각 128 차원 벡터로 요약되어 은닉 마르코프 모델의 입력으로 주어진다. 은닉 마르코프 모델은  학습된 PBGCN-LSTM 인코더로 추출된 특징들로 학습된다. 학습된 후, 인코딩된 다른 영상이 입력으로 들어오면 비터비(Viterbi) 알고리즘을 통해 프레임 별로 현재 특징이 나타난 상태를 추적한다. 이 상태를 행동 분석에서 자세라고 정의한다.

## 2. 주요 자세 요약 모델




[그림 19] 주요 자세 요약을 위한 PBGCN-LSTM 인코더

PBGCN 을 통해 추출된 각 프레임 별 공간상 특징들을 LSTM 에 입력으로 넣기 전, 프레임 시퀀스를 크기가 W 인 윈도우를 겹침 없이 슬라이딩하여 적당한 길이로 자른다. 그 후, 각 W 개 프레임을 가지는 시퀀스 토막 내의 공간상 특징을 평균 낸 값을 LSTM 입력으로 사용한다. 이는 실험적 결과로 얻어진 사실로, 한 행동만 나타나는 짧은 길이의 영상에서 주요 자세를 찾고자 할 때 모든 프레임의 정보는 불필요하다. 따라서, 주요 자세 요약 모델을 위해 공간상 특징들을 10, 50, 100 프레임 단위 윈도우로 시퀀스를 잘라 정보를 평균 낸 후, 원래 T 길이 시퀀스를 $t = \frac{T}{W}$ 길이로 정보량을 줄여 LSTM 에 입력한다.

# V. 실험 결과

## 1. 데이터셋

실험 데이터셋은 NTU-RGB+D60[21]을 사용한다. 사용한 데이터셋은 Microsoft Kinect v2 로 촬영하여 RGB 영상, Depth 영상, 3 차원 골격 데이터, Infrad 영상으로 구성되어있다. 본 논문에서는 카메라 기준 좌표로 추출된 3 차원 골격 데이터를 사용했다.

NTU-RGB+D60 은 행동 인식 문제를 위해 한 영상 내에 한 가지 행동만 녹화한 데이터셋이다. 총 40 명의 피험자가 60 개 행동들을 각각 수행하였고, 카메라는 3 대를 사용하여 정면과 양 쪽 측면에서 촬영하였다. 총 영상 개수는 56,800 개이고, 누락된 데이터를 제외한 56,578 개 영상을 사용했다.

본 실험에는 피험자 ID 를 기준으로 훈련, 검증, 테스트 데이터셋으로 나누어 훈련과 검증 데이터셋들은 PBGCN-LSTM 학습용으로 사용했다. 훈련 데이터셋 구성원은 20 명, 데이터 개수는 40,091 개, 검증 데이터셋 구성원은 12 명, 데이터 개수는 12,193 개, 테스트 데이터셋 구성원은 8 명, 데이터 개수는 4,294 개이다. 각 데이터셋 별 구성원은 다음 표와 같다.

[표 1] NTU-RGB+D60 데이터셋 분리 기준

| 종류 | 피험자 ID |
|---|---|
| 훈련 | 1, 2, 4, 5, 8, 9, 13, 14, 15, 16, 17, 18, 19, 25, 27, 28, 31, 34, 35, 38 |
| 검증 | 3, 6, 7, 10, 11, 12, 29, 30, 32, 33, 36, 37 |
| 테스트 | 20, 21, 22, 23, 24, 26, 39, 40 |

[표 2] NTU-RGB+D60 행동 종류

| | | |
|---|---|---|
| A1. drink water | A21. take off a hat/cap | A41. sneeze/cough |
| A2. eat meal/snack | A22. cheer up | A42. staggering |
| A3. brushing teeth | A23. hand waving | A43. falling |
| A4. brushing hair | A24. kicking something | A44. touch head (headache) |
| A5. drop | A25. reach into pocket | A45. touch chest (stomachache/heart pain) |
| A6. pickup | A26. hopping (one foot jumping) | A46. touch back (backache) |
| A7. throw | A27. jump up | A47. touch neck (neckache) |
| A8. sitting down | A28. make a phone call/answer phone | A48. nausea or vomiting condition |
| A9. standing up (from sitting position) | A29. playing with phone/tablet | A49. use a fan (with hand or paper)/feeling warm |
| A10. clapping | A30. typing on a keyboard | A50. punching/slapping other person |
| A11. reading | A31. pointing to something with finger | A51. kicking other person |
| A12. writing | A32. taking a selfie | A52. pushing other person |
| A13. tear up paper | A33. check time (from watch) | A53. pat on back of other person |
| A14. wear jacket | A34. rub two hands together | A54. point finger at the other person |
| A15. take off jacket | A35. nod head/bow | A55. hugging other person |
| A16. wear a shoe | A36. shake head | A56. giving something to other person |
| A17. take off a shoe | A37. wipe face | A57. touch other person's pocket |
| A18. wear on glasses | A38. salute | A58. handshaking |
| A19. take off glasses | A39. put the palms together | A59. walking towards each other |
| A20. put on a hat/cap | A40. cross hands in front (say stop) | A60. walking apart from each other |

## 2. 성능평가 및 시각화

### 가. NTU-RGB+D60: GCN-LSTM 인코더 행동 인식 성능 비교

[표 3] NTU-RGB+D60 을 이용한 행동 인식: Top1, Top5 Accuracy (%)

| 모델 종류 | Top1(%) | Top5(%) |
|---|---|---|
| PBGCN[PBGCN] | 80.48 | 97.65 |
| LSTM | 64.43 | 89.96 |
| HBRNN-L[HBRNN] | 59.07 | 63.98 |
| GCN-LSTM | 50.7 | 82.9 |
| PBGCN-LSTM | 59.49 | 89.08 |
| LSTM (10 frames) | 65.40 | 91.18 |
| GCN-LSTM (10 frames) | 77.86 | 95.25 |
| PBGCN-LSTM (10 frames) | 78.50 | 95.38 |
| PBGCN-LSTM (50 frames) | 74.94 | 94.25 |
| PBGCN-LSTM (100 frames) | 74.44 | 94.53 |

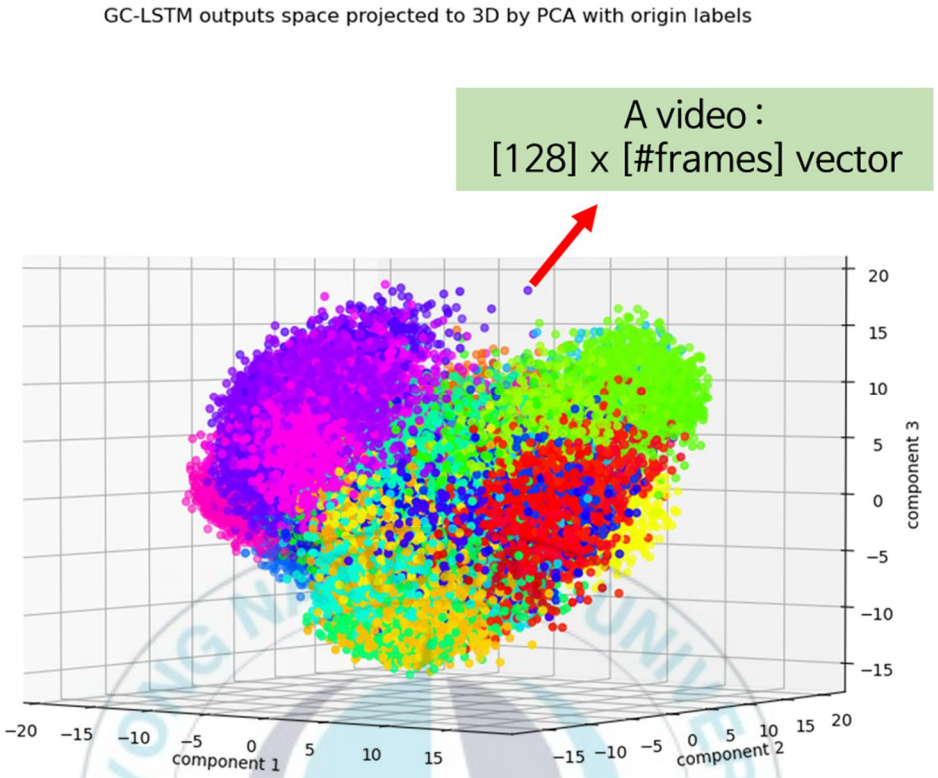
먼저 2 Layer LSTM 으로 골격 데이터를 1 차원으로 표현하여 학습하였을 때, top 이 64.43%, top5 이 89.96% 행동 인식 정확도를 보여준다. 이는 기존 HBRNN-L 보다는 장단기 특징이 학습되어 좋은 결과를 보여주는 것을 볼 수 있다. 한편, 시공간 특징 추출을 위한 결합 모델인 GCN-LSTM 은 LSTM 단독 사용보다 정확도가 떨어진다.

하지만, 10 개 프레임 단위 윈도우로 특징 값들을 평균 풀링(mean pooling)하여 학습한 결과는 오히려 GCN-LSTM 이 높은 성능을 보여주고 있다.

이는 프레임 별로 추출된 골격 그래프 특징이 많아서 제대로 학습할 수 없지만, 공간 특징을 일정 프레임 길이 단위로 압축하면 더 정확한 결과를 얻을 수 있다는 것을 알 수 있다.

이 사실을 기반으로 PBGCN-LSTM 에 윈도우를 적용하여 학습한 결과, GCN-LSTM 보다 고정된 그래프에서 부위별 학습을 할 수 있어 높은 정확도를 보여주었다. 이는 그래프에 따라 사전 지식에 의해 미리 정의되는 방식에 따라 결과가 달라질 수 있음을 시사한다. 그리고 윈도우 길이를 늘려본 결과, 한 행동 영상에서 주요 자세를 추출하기 위해서는 10 프레임 단위가 가장 높은 정확도를 보여주었다. 이는 한 행동 영상이 짧고 자세 변화가 짧은 순간에 일어나기 때문에 이러한 결과를 나타내는 것으로 보인다.

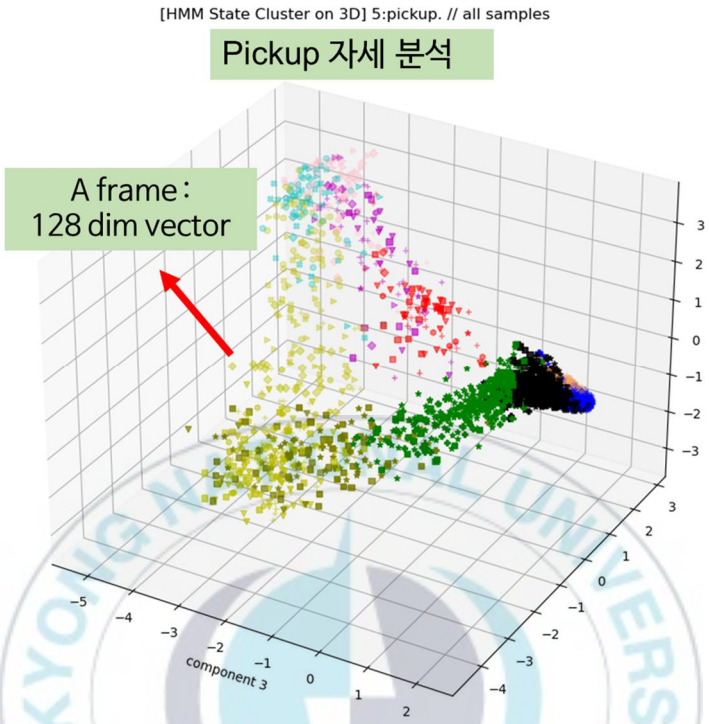
한편, 제안된 PBGCN-LSTM 의 경우 기존의 PBGCN 보다 정확도가 떨어지는 모습을 보여준다. 이는 PBGCN 은 3 차원 합성곱 필터에 시공간 정보를 담기 때문에 요약된 정보로 행동을 분류하지만, PBGCN-LSTM 은 시간상으로는 요약되지 않은 정보를 활용하여 행동을 분류하기 때문에 성능 저하가 발생하는 것으로 보인다. 하지만, 본 제안 모델은 기존 PBGCN 과는 다른 문제 상황에서 프레임 단위로 자세를 분석할 수 있는 모델이다.




[그림 20] PBGCN-LSTM 을 이용하여 분류된 행동 군집

위 그림은 제안된 PBGCN-LSTM 을 이용하여 각 입력 영상들의 행동을 분류하여 주성분 분석(Principle Component Analysis, PCA)로 투영한 결과이다. 그래프내 한 점은 한 영상의 시공간 특징을 의미하고, 128xT 차원 벡터로 표현된다. 그리고 점들의 색은 각 행동을 의미한다. 그래프 결과 행동들이 잘 분류되어 군집을 이루고 있음을 볼 수 있다.

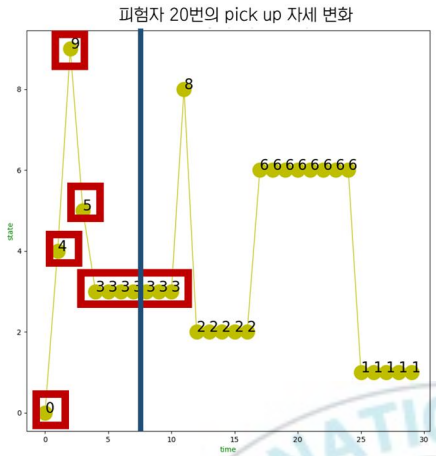
## 나. NTU-RGB+D60: HMM 자세 분석 클러스터 시각화



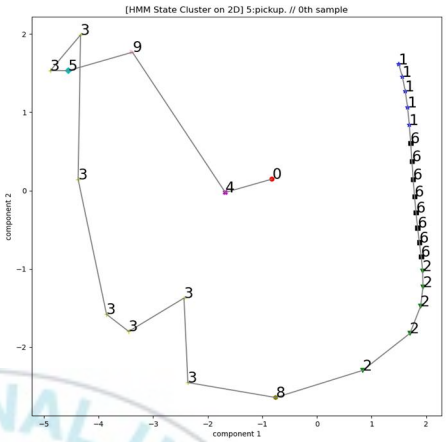

[그림 21] Pickup 행동에 대한 자세 군집

위 그림은 행동 종류별로 은닉 마르코프 모델에 각 골격 영상들의 추출된 10 프레임 길이 윈도우 단위 시공간 특징들을 학습하여 비터비 알고리즘을 통해 얻어진 상태 군집 결과이다. 이 때, 은닉 마르코프 모델의 상태 수는 모두 동일하게 10 이고, 관측 값은 가우시안 분포(Gaussian distribution)을 따른다. 그래프 위의 한 점은 한 시공간 특징을 의미하고 이는 128 차원 벡터를 주성분 분석을 이용해 3 차원으로 투영시킨 결과이다. 위 그림은 60 가지 행동 중 Pickup 행동에 해당되는 모든 영상에서 나오는 자세를 군집화한 결과이다. (색과 점 모양에 따라 자세가 분류된다.)

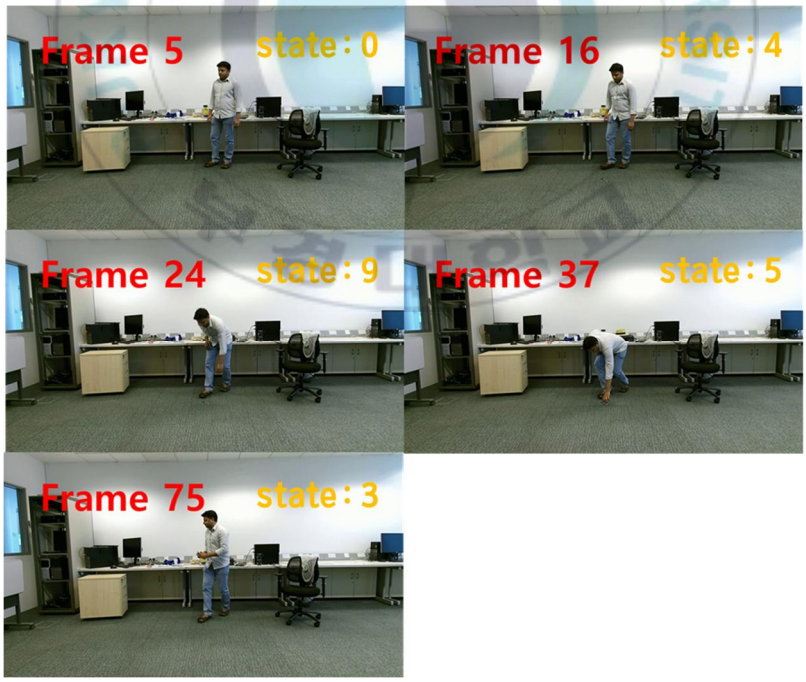
## 다. NTU-RGB+D60: HMM 자세 변환 시각화




[그림 22] 피험자 20 번의 Pickup 자세 변화




[그림 23] 피험자 20 번의 Pickup 궤적




[그림 24] 피험자 20 번의 Pickup 행동 요약된 영상

그림 22 과 23 는 피험자 20 번 수행한 Pickup 행동에 대한 자세 변화를 도식화한 결과이다. 이는 그림 21 에서 군집화된 자세들 중 0, 4, 9, 5, 3, 8, 2, 6, 1 상태들로 변화함을 볼 수 있다. 즉, 피험자 20 번의 Pickup 행동은 위의 9 가지 자세로 요약된다. 이 9 가지 자세들 중에서 영상이 75 프레임에서 끝나므로 0, 4, 9, 5, 3, 5 가지 자세로 추려진다. 그림 23 에서 5 가지 자세의 궤적을 보면 활발하게 변화함을 볼 수 있다. 그림 24 는 원본 영상에서 5 가지 자세에 해당하는 프레임들을 뽑아본 결과이다. 프레임 5 번-자세 0 번은 서 있는 자세, 프레임 16 번-자세 4 번은 줍기를 시작하는 자세, 프레임 24 번-자세 9 번은 허리를 굽히기 시작하는 자세, 프레임 37 번-자세 5 번은 줍는 자세, 프레임 75 번-자세 3 번은 줍기를 마친 자세이다. 이는 줍은 행동을 잘 요약하여 보이고 있음을 볼 수 있다.

# VI. 결론

본 논문에서는 사람 골격 영상 기반 행동 인식을 위해 부위별 그래프 합성곱 네트워크와 장단기 메모리(Parte-Based Graph Convolutional Network-Long Short Term Memory, PBGCN-LSTM)를 결합한 인코더와 인코더에서 추출된 특징을 분석하는 은닉 마르코프 모델을 제안한다. 제안된 모델은 사람 골격 영상에서 시공간 특징을 추출함과 동시에 각 프레임 별로 자세를 분석할 수 있다는 이점이 있다. 이는 자율 주행 자동차나 이상 행동 탐지와 같은 실시간으로 사람의 행동을 판단해야하는 문제에서 적용될 수 있다.

PBGC-LSTM 은 각 프레임별로 시공간 특징을 추출한다. 먼저 PBGCN 을 통해 각 프레임 내에 있는 골격 그래프의 공간 특징을 추출하고, LSTM 의 입력으로 각 프레임의 공간 특징을 사용한다. 그 후, LSTM 을 통해 각 프레임에서 128 차원 벡터로 표현되는 시공간 특징을 추출한다. 인코더의 학습이 끝난 후, 각 프레임이 128 차원으로 표현되는 영상을 행동 별로 HMM 에 학습한다.

실험결과, PBGCN-LSTM 은 기존 공간 특징만 추출하는 PBGCN 과 시간 특징만 추출한 LSTM 의 변형 모델들과 견줄만한 행동 인식을 보여주었다. 그리고 HMM 으로 분석하여 시각화한 결과 행동에서 의미 있는 자세들이 군집화 됨을 볼 수 있었다.

# 참고 문헌


[1] Z. Zhuang,Y. Xue (2019), “Sport-Related Human Activity Detection and Recognition Using a Smartwatch”, Sensors, vol. 19, no. 22, pp. 5001.

[2] V. Chandola, A. Banerjee, V. Kumar (2009), “Anomaly Detection: A Survey”, ACM Computing Surveys, vol. 41, no. 3, pp. 1-72.

[3] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, F. E. Alsaadi, “A Survey of Deep Neural Network Architectures and Their Applications”, Neurocomputing, vol. 234, no. 234, 2017, pp. 11–26.

[4] R. Zhao, W. Xu, H. Su and Q. Ji (2019), “Bayesian Hierarchical Dynamic Model for Human Action Recognition”, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 7725-7734, Long Beach, CA, USA.

[5] R. Vemulapalli, F. Arrate and R. Chellappa (2014), “Human Action Recognition by Representing 3D Skeletons as Points in a Lie Group”, 2014 IEEE Conference on Computer Vision and Pattern Recognition, pp. 588-595, Columbus, OH.

[6] S. Li, W. Li, C. Cook, C. Zhu and Y. Gao (2018), “Independently Recurrent Neural Network (IndRNN): Building A Longer and Deeper RNN”, 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 5457-5466, Salt Lake City, UT.

[7] Chaudhuri A. and Ghosh S.K. (2016) “Sentiment Analysis of Customer Reviews Using Robust Hierarchical Bidirectional Recurrent Neural Network”, Advances in Intelligent Systems and Computing, vol 464, Artificial Intelligence Perspectives in Intelligent Systems, pp. 249-261, Springer, Cham.

[8] I. Lee, D. Kim, S. Kang and S. Lee (2017), “Ensemble Deep Learning for Skeleton-Based Action Recognition Using Temporal Sliding LSTM Networks”, 2017 IEEE International Conference on Computer Vision (ICCV), pp. 1012-1020, Venice.

[9] J. Liu, G. Wang, P. Hu, L. Duan and A. C. Kot (2017), “Global Context-Aware Attention LSTM Networks for 3D Action Recognition”, 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3671-3680, Honolulu, HI.

[10] T. S. Kim and A. Reiter (2017), “Interpretable 3D Human Action Analysis with Temporal Convolutional Networks”, 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 1623-1631, Honolulu, HI.

[11] M. Liu, H. Liu, and C. Chen (2017), “Enhanced skeleton visualization for view invariant human action recognition”, Pattern Recognition, vol. 68, pp. 346–362.

[12] Y. Du, Y. Fu and L. Wang (2015), “Skeleton based action recognition with convolutional neural network”, 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 579-583, Kuala Lumpur.

[13] Bo Li, Yuchao Dai, Xuelian Cheng, Huahui Chen, Yi Lin and Mingyi He (2017), “Skeleton based action recognition using translation-scale invariant image mapping and multi-scale deep CNN”, 2017 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 601-604, Hong Kong.

[14] C. Cao, C. Lan, Y. Zhang, W. Zeng, H. Lu and Y. Zhang (2019), “Skeleton-Based Action Recognition With Gated Convolutional Neural Networks”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 29, no. 11, pp. 3247-3257.

[15] S. Hochreiter and J. Schmidhuber (1997), “Long Short-Term Memory”, Neural Computation, vol. 9, no. 8, pp. 1735–1780.

[16] Lawrence R. Rabiner (1990), “A tutorial on hidden Markov models and selected applications in speech recognition”, Readings in speech recognition, pp. 267–296, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[17] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov (2016), “Learning convolutional neural networks for graphs”, Proceedings of the 33rd International Conference on Machine Learning (ICML'16), vol. 48, pp. 2014–2023.

[18] S. Yan, Y. Xiong, and D. Lin (2018), "Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition", Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32, no. 1.

[19] K. C. Thakkar, and P. J. Narayanan (2018). “Part-Based Graph Convolutional Network for Action Recognition”, 2018 British Machine Vision Conference, pp. 270.

[20] Maosen Li, Siheng Chen, Xu Chen, Ya Zhang, Yanfeng Wang, Qi Tian(2019), “Actional-Structural Graph Convolutional Networks for Skeleton-Based Action Recognition”, Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 3595-3603.

[21] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, Gang Wang (2016), “NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis”, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1010-1019.

[22] Liliana Lo Presti, Marco La Cascia (2016), “3D skeleton-based human action classification: A survey”, Pattern Recognition, vol. 53, pp. 130-147.

# 감사의 글

여기까지 오는 것이 결코 순탄하지 않았습니다. 먼 길을 돌고 돌아 험난했던 여정 속에서 많은 분들을 뵙고, 이야기 나누고, 같이 성장했습니다. 먼저, 학부 2 학년때부터 지금까지 묵묵하고 든든하게 지켜봐주신 신봉기 교수님께 감사드립니다. 늘 부족하기만 했던 저에게 성실한 자세와 넘치는 탐구심을 가르쳐주시며 때로는 아버지처럼, 때로는 선배 연구자처럼 조언을 해주셨기에 지금의 제가 있습니다. 다시 한 번 감사드립니다.

그리고, 연구자로써의 마음가짐을 알려주신 최선한 교수님과 공학도로써 어떤 일을 해야하고, 어떤 꿈을 꿔야하는지를 알려주신 김채규 교수님, 물심양면 지원해주신 권기룡 교수님, 부족한 논문 심사를 맡아주시고 격려해주신 송하주 교수님 이외 저를 가르쳐주셨던 모든 교수님들께도 감사합니다.

이 여정이 끝이 아닌 새로운 시작이 되게끔 아낌없는 사랑과 지지를 해준 어머니 김양순 여사님과 동생 양건에게 감사합니다. 어머니께서 힘내서 버텨주시지 않았다면 이 글도 없었을 것입니다. 그리고 언제나 힘들때 제 얘기를 가장 속 깊게 많이 들어주고 제 스스로 답을 찾게 이끌어준 동생에게 감사합니다. 두 분 다 사랑합니다.

김혜빈, 김종민, 김채은 모두 애정합니다. 제 학부 시절부터 대학원 시절까지 같이 공부하고, 같이 웃고, 같이 울고, 같이 고생하고, 같이 성장한 정말 제 인생에서 몇 없을 언니, 친구, 동생입니다.

혜빈 언니와는 신입생 환영회때부터 지금까지도 희노애락을 나누고, 언니는 늘 정진하는 자세를 보여주며 배울 점이 많은 사람입니다. 그런 멋진 사람이 제 언니라 늘 감사했고, 지금도 감사합니다.

종민이는 가장 힘든 시기에 곁을 지켜준 든든한 사람입니다. 다사다난했던 지난 2020 년까지의 마무리를 지켜봐주고 끝까지 응원해주어 감사합니다. 앞으로도 서로 응원해주고 함께 성장하는 관계로 잘 발전해가도록 노력하겠습니다.

채은이는 처음 본 순간부터 제 사람임을 알았습니다. 그랬기에 함께 동아리도 하고, 연구실 생활도 하며 정말 많이 웃었고, 행복했습니다. 같이 어려운 시기에도 함께 있어주고 정말 감사했고, 지금까지도 다 이야기할 수 있는 사이로 남아주어 감사합니다.

그리고, 지난 6 년간 지능미디어 연구실부터 인공지능 연구실이 되기까지 저와 함께 연구실 생활을 해주신 김도현, 류연희, 임아현, 황세웅, 김정주, 박신혜, KOKOY, PAMELA, RENDRA, JEMMY 외에도 모든 연구실원 선후배님들께 감사합니다. 여러분들 덕분에 제가 길을 잃지 않고 여기까지 잘 오게 되었습니다.

끝으로, 포기하지 않고 이 여정의 종지부를 찍고 새로운 여행을 떠나는 저에게 감사합니다. 지금 여기에 글을 쓰는 이 마음을 잃지 말고, 감사할 줄 아는 사람이 되어 소중히 여기는 사람들과 함께 꾸준히 성장할 수 있는 사람이 되면 좋겠습니다.

모두 감사합니다.