The Korean Journal of Applied Statistics
2024, Vol. 37, No. 5, 643–661

DOI: https://doi.org/10.5351/KJAS.2024.37.5.643
Print ISSN1225-066X / Online ISSN 2383-5818

Classification of human actions using 3D skeleton data: A performance comparison between classical machine learning and deep learning models

Juhwan Kim[a], Jongchan Kim[a], Sungim Lee[1,b]

[a]Department of Applied Statistics, Dankook University;
[b]Department of Statistics and Data Science, Dankook University

Abstract

This study investigates the effectiveness of 3D skeleton data for human action recognition by comparing the classification performance of machine learning and deep learning models. We use the subset of the NTU RGB+D dataset, containing only frontal-view recordings of 40 individuals performing 60 different actions. Our study uses linear discriminant analysis (LDA), support vector machine (SVM), and random forest (RF) as machine learning models, while the deep learning models are hierarchical bidirectional RNN (HBRNN) and semantics-guided neural network (SGN). To evaluate model performance, cross-subject cross-validation is conducted. Our analysis demonstrates that action type significantly impacts model performance. Cluster analysis by action category shows no significant difference in classification performance between machine learning and deep learning models for easily recognizable actions. However, for actions requiring precise differentiation based on frontal-view joint coordinates such as 'clapping' or 'rubbing hands', deep learning models show a higher performance in capturing subtle joint movements compared to machine learning models.

Keywords: skeleton data, machine learning models, deep learning models, cross-subject cross-validation

1. 서론

스켈레톤 데이터는 객체의 움직임을 추적하기 위해 사용되며, 연속적인 3차원 프레임으로 관측된 관절 위치 정보를 나타낸다. 이는 마이크로소프트의 키넥트(Kinect), 인텔의 리얼센스(웨어러블 디바이스)와 같은 센서를 통해 수집할 수 있으며, 단순한 위치 정보뿐만 아니라 시간에 따른 연속적인 변화를 나타내어 동작 인식 연구에 매우 유용하다. 최근 인공지능 기술의 발전과 함께 스켈레톤 데이터를 활용한 동작 인식 연구가 활발히 진행되고 있으며, 이 연구는 병원, 요양원, 대형 쇼핑몰 등 공공장소에서 낙상 감지나 이상 행동 감지 시스템 구축 (Taha 등, 2015; Kang 등, 2021; Shin 등, 2021) 또는 보행 연구 (Tao 등, 2012; Chaaraoui 등, 2015), 헬스케어 서비스 (Jin 등, 2015; Lin 등, 2018), 그리고 인간-로봇 상호작용 (Du 등, 2012; Yang 등, 2015), 스마트 홈 시스템에서 노인과 어린이의 일상생활을 모니터링하는 연구도 진행되고 있고 (Jalal 등, 2012), 애니메이션 제작 (Jeong과 Park, 2018)에도 활용되는 등 스켈레톤 데이터 분석에 대한 활용이 다양함을 알 수 있다.

스켈레톤 데이터는 관절 J개의 각 위치에 대하여 T개의 3차원 프레임으로 수집되므로 분류모델에서 예측변수의 개수는 $J \times 3 \times T$이다. 이와 같은 스켈레톤 데이터를 활용한 동작 인식 연구에서 머신러닝 모델과

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea govern- ment (MSIT) (No. 2019R1A2C1003257).

[1]Corresponding author: Department of Statistics, 152 Jukjeon-ro, Suji-gu, Yongin-si, Gyeonggi-do 16890, Korea. E-mail: silee@dankook.ac.kr

Published 31 October 2024 / journal homepage: http://kjas.or.kr

딥러닝 모델은 각기 다른 접근 방식을 채택하고 있다. 본 연구에서는 이들 모델의 성능 차이를 분석하고자 한다. 머신러닝 모델은 일반적으로 3D 스켈레톤 관절 위치 좌표를 직접 사용하는 대신, 각 관절별로 T개의 3차원 프레임으로부터 구한 요약 통계량을 활용하여 데이터의 차원을 줄이는 방식으로 접근한다. 예를 들어, Reddy와 Chattopadhyay (2014)는 관절별로 모든 프레임에서 구한 (X, Y, Z) 좌표에 대해 평균값과 범위를 계산하여 예측변수로 사용했고, 서포트 벡터 머신(supportvector machine; SVM) 모델을 통해 '앉기', '걷기', '서 있기' 등의 동작을 97.3%의 정확도로 분류하였다. 또 다른 연구로, Ghazal 등 (2019)은 2D 스켈레톤 데이터를 통해 프레임별 관절 간 각도와 위치변화량의 평균을 예측변수로 사용했다. 이들은 엉덩이, 무릎, 발목, 어깨 관절의 오른쪽과 왼쪽을 활용하여 '앉기', '걷기', '서 있기' 등의 동작을 인식하였으며, K-최근접 이웃(K-nearest neighbors), SVM, 나이브 베이즈(Naive Bayes), 선형 판별 분석(linear discriminant analysis; LDA) 등의 머신러닝 모델들을 사용하였다. Kim 등 (2021)은 관절별로 모든 프레임에서의 위치 분포를 요약하기 위해 4차 적률까지를 예측 변수로 활용하였다. 이들은 LDA, Lasso, SVM, 랜덤 포레스트(random forest; RF), 그레디언트 부스팅 머신(gradinet boosting machine) 등의 다양한 머신러닝 모델을 적용하였다. 이와 같이 각 관절 위치에 대하여 요약 통계량을 계산하여 예측 변수로 사용하는 방식은 데이터의 차원을 줄여 모델의 복잡성을 낮추고, 빠른 학습과 높은 효율성을 제공한다. 그러나 이러한 접근 방식은 연속적인 동작의 정보 대신 각 관절 좌표의 분포만을 반영하는 문제가 있다.

딥러닝 모델은 관절 데이터를 그대로 사용하여 데이터의 복잡한 패턴을 학습한다. 초기에는 Lefebre 등 (2013), Gregor 등 (2015) 연구 등에서 사용된 순환 신경망(recurrent neural network; RNN) 모델과, Grushin 등 (2013), Veeriah 등 (2015), Zhu 등 (2016) 연구에서 활용한 LSTM (long short-term memory)의 모델이 주로 사용되었다. 이 모델들은 프레임별 관절 위치의 벡터 시퀀스를 연속적으로 처리하여 시간에 따른 관절의 움직임을 표현하였다. 그러나 기존 RNN 방식은 순차적인 데이터의 특성상, 각 프레임의 관절 좌표가 전체 시퀀스에서 차지하는 상대적인 중요도를 정확하게 평가하지 못하며, 관절 좌표 간의 복잡한 상호 작용을 모델링하는데 한계가 있다. 이 문제를 해결하기 위해, Shahroudy 등 (2016)과 Du 등 (2016)은 신체 부위별로 LSTM을 분리하여 각 관절 좌표의 중요도를 차별화하여 설계하였고, Liu 등 (2016)은 시공간적 LSTM을 통해 시간적 및 공간적 정보를 동시에 학습함으로써, 기존 RNN의 한계를 극복하고자 하였다.

한편, Cao 등 (2017)과 Ke 등 (2017)은 음성 및 언어 시퀀스 모델링 분야에서 우수성을 입증한 합성곱 신경망(convolutional neural network; CNN) 기반의 딥러닝 모델이 스켈레톤 기반 동작 인식 분야에서도 효과적임을 보였다. 특히, Du 등 (2015)과 Li 등 (2017)은 3D 스켈레톤 데이터를 이미지 픽셀의 RGB 채널로 변환하는 방식을 제안하여, 관절 좌표로부터 신체의 공간적 정보를 추출하고 복잡한 동작 패턴을 학습하였다. 하지만 CNN의 커널(Kernel)은 국소적인 영역의 정보만을 추출하도록 설계되어 있어 인접한 관절 간 관계는 잘 학습하지만, 멀리 떨어진 관절과의 상호작용이나 전체적인 동작 패턴을 파악하기는 어렵다. 이러한 문제를 해결하기 위해 Chao 등 (2018)은 계층적 방법론을 제안하여 전체 동작 패턴에서 관절 간 움직임의 연관성을 파악하기에는 한계가 있다. Kipf와 Welling (2016)은 그래프 합성곱 신경망(graph convolutional network; GCN) 모델을 활용하여 스켈레톤 데이터를 이용한 분류 문제에서 인체 고유의 관절 연결 관계를 효과적으로 설명할 수 있음을 연구하였다. Yan 등 (2018)은 각 관절을 그래프의 노드로 취급하고, 관절 간의 관계를 나타내는 엣지를 도메인 지식에 기반하여 정의하여 GCN 모델을 통해 동작분류를 수행하였다. Tang 등 (2019)은 물리적으로 연결되지 않은 관절 쌍과 연결된 관절 쌍 전체를 엣지로 정의하여 관절 위치를 나타낸 그래프를 더욱 개선하였다. 이러한 최신 GCN 기반 딥러닝 모델들은 동작 분류에 우수한 성능을 나타내었는데, 스켈레톤 데이터에 기반한 동작 인식 데이터셋으로 잘 알려진 NTU RGB+D에 대해 Zhang 등 (2020)의 SGN, Xu 등 (2023)의 LA-GCN, Lee 등 (2023)의 HD-GCN 등은 90% 이상의 높은 분류성능을 기록하고 있으며, 다수의 GCN 기반 모델들이 상위권에 위치하고 있다. 관련자료는 웹사이트 `https://paperswithcode.com/sota/skeleton-based-action-recognition-on-ntu-rgbd`에서 확인 가능하다. 이처럼 딥러닝 모델은 관절 위치 좌표를 입력 변수로 사용하여 의미 있는 특징들을 추출해내며 우수한 분류 성능을 보인다. 그러나 스켈레톤

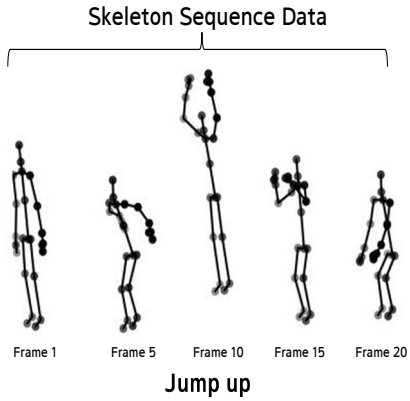


Figure 1: *Skeleton joint sequence for the "Jump Up" action from the NTU RGB+D dataset (Shahroudy et al., 2018).*

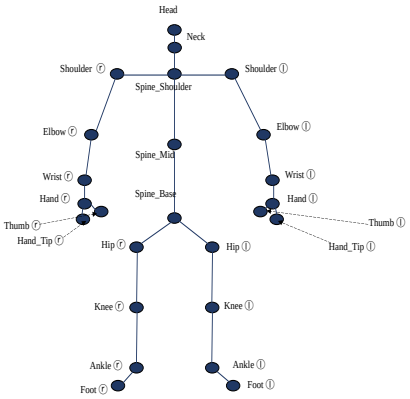


Figure 2: *25 joint locations in the skeleton data.*

데이터와 같이 차원이 큰 경우에는 계산적 비용이 상당히 요구되는 단점이 있다.

즉, 머신러닝 모델은 상대적으로 적은 수의 예측 변수를 사용하여 학습속도가 빠른 반면, 딥러닝 모델은 관절 좌표를 그대로 사용하여 복잡한 패턴을 학습하는 데 계산적 비용이 매우 크다는 차이점이 있다. 이에 본 연구에서는 기존의 머신러닝 모델과 딥러닝 기반 모델을 비교하여 그 성능 차이를 분석하고자 한다. Jeong과 Lim (2019)은 다양한 수요 예측 모델을 소개하며 머신러닝과 딥러닝 모델의 활용 가능성을 제시하였다. 본 연구는 이러한 선행 연구를 바탕으로, 스켈레톤 데이터를 이용한 다양한 동작 분류 문제에 있어 이들 두 종류의 모델의 분류 성능을 직접 비교하여 어떤 모델이 더 효율적인지 알아보고자 한다. 이를 위해 2절에서는 먼저 스켈레톤 데이터와 전처리 방법을 소개하고, 기존의 다양한 분류 모델을 머신러닝 모델과 딥러닝 모델로 나누어 소개하기로 한다. 3절에서는 실제 데이터를 활용하여 두 종류의 모델 성능을 비교 분석한다. 마지막으로 4절에서는 연구결과를 종합하고, 향후 연구 방향을 제시하기로 한다.

2. 스켈레톤 데이터 기반 분류분석

2.1. 전처리

이 논문에서는 3D 골격 위치를 관측한 스켈레톤 데이터는 i번째 사람의 j번째 관절이 t번째 프레임에서 $J_{it}^{j} = (x_{it}^{j}, y_{it}^{j}, z_{it}^{j}), i = 1, \ldots, n, j = 1, \ldots, J, t = 1, \ldots, T$ 로 표현하기로 한다. 예를 들어, Figure 1은 Jump up 동작을 $J = 25$개의 관절에 대해 첫 번째와 마지막 프레임을 포함하여 5개의 프레임을 선택하여 순서대로 그린 것이다. 센서 장치에 따라 관측 가능한 관절 위치의 개수가 다를 수 있지만, 일반적으로 Figure 2와 같이 보통 25개의 관절 위치가 포함된다. Figure 1와 같은 스켈레톤 데이터를 실험을 통해 수집할 때, 각 실험 참가자의 신체와 움직임 차이를 고려하여 데이터를 전처리하는 것은 머신러닝 모델이나 딥러닝 모두에서 매우 중요하다 (Cho와 Chen, 2014; Taha 등, 2015; Lee 등, 2017; Sandra, 2020). 일반적으로 3D 스켈레톤 좌표의 원점을 이동시켜 초기 프레임의 특정 신체 부위가 원점 $(0, 0, 0)$이 되도록 정규화한다. 정규화된 스켈레톤 관절은 다음과 같은 수식으로 표현된다:

$$J_{t(norm)}^{j} = \left(J_{t}^{j} - J_{t}^{Center}\right),$$

여기서 J_{t}^{Center}는 정규화의 기준점이 된다. Lee 등 (2017)은 Figure 2의 'Hip ⓡ', 'Hip ⓛ'을 기준점으로 사용하고, Sandra (2020)는 'Neck'을 기준점으로 삼았고, Kim 등 (2023)은 'Spine_Mid' 등을 사용하였다.

2.2. 머신러닝 접근법

3D 스켈레톤 데이터에 관한 머신러닝 기반 동작 분류 연구에서는 일반적으로 3차원 관절 위치 좌표 대신에 각 관절의 좌표가 T 프레임 동안 어떻게 분포하는 지를 나타내는 요약 통계량을 사용한다. 주로 평균, 범위 등이 사용되며, 더 많은 요약 통계량이 사용되기도 한다. 예를 들어, Reddy와 Chattopadhyay (2014) 연구에서는 '걷기'와 '일어서기'와 같은 동작을 구분하기 위해 T 프레임 동안의 좌표 평균값과 범위를 사용하였다. Ghazal 등 (2019)는 8개 관절에서 관절 간 2차원 관절좌표의 각도와 위치변화량을 기반으로 하여 16개의 입력변수를 사용하였으며, Kim 등 (2021)은 관절의 3차원 좌표의 프레임별 4차 적률까지를 예측변수로 사용하였다. 이와 같은 방법은 프레임별 관측된 모든 좌표를 예측 변수로 직접 사용하는 것보다 데이터의 차원을 줄일 수 있는 장점이 있다. 본 연구에서는 스켈레톤 데이터 분류를 위한 미선러닝 모델로 선형판별분석(linear discriminant analysis; LDA), 다중 클래스 서포트벡터머신(multi-class support vector machine; SVM), 랜덤포레스트(random forest; RF)을 적용하였다. SVM이나 RF 등은 Shan과 Akella (2014), Amor 등 (2016), Ghazal (2019) 등의 연구에서도 뛰어난 분류성능으로 널리 활용된 분류모델이다.

2.2.1. 선형판별분석

선형판별분석은 각 클래스의 데이터가 클래스별 특정 평균 벡터와 k개 모든 클래스에 공통인 공분산을 갖는 다변량 가우스분포를 따른다는 가정하에, 최적의 선형 결정경계를 찾는 방법이다. 각 관절 별(j) 3차원 위치 좌표에 대한 l번째 요약통계량을 나타낸 예측변수 $J_{l}^{j} = (x_{l}^{j}, y_{l}^{j}, z_{l}^{j})(j = 1, \ldots, 25; l = 1, \ldots, 5)$는 375×1 벡터로 k번째 클래스에서 평균벡터가 μ_k, 공분산 행렬이 Σ인 다변량 정규분포 를 따른다고 가정하자. 총 K개의 동작 클래스 $C_1, \ldots, C_K$를 가진다고 할 때, 관측벡터 J_{l}^{j}에 대한 판별 점수는 아래의 식으로 표현된다.

$$\delta_k\left(J_{l}^{j}\right) = (J_{i}^{l})^{\top}\,\Sigma^{-1}\mu_k - \frac{1}{2}\mu_k^{T}\Sigma^{-1}\mu_k + \log\pi_k, \quad k = 1, \ldots, K, \tag{2.1}$$

여기서 π_k는 랜덤하게 선택한 관측값이 k번째 클래스에서 속할 사전 확률을 나타낸다.

2.2.2. 다중 클래스 서포트벡터머신

SVM은 기본적으로 두 개의 그룹을 분리하는 방법으로, 두 그룹 사이의 최적 분리 초평면(hyperplane)을 선택하는 알고리즘이다 (Izenman, 2008). 여기서는 다중 클래스 문제를 해결하기 위해 각 클래스 쌍 (i, j)에 대해 이진 SVM을 학습한다. 총 K의 클래스가 있을 때, $K(K-1)/2$개의 이진 분류기를 학습한다. 각 이진 SVM은 다음과 같은 목적함수를 통해 적합된다.

$$\min_{\mathbf{w}_{ij}, b_{ij}} \frac{1}{2} \left\| \mathbf{w}_{ij} \right\|^2 + C \sum_{k=1}^{n_{ij}} \max\left(0, 1 - y_{ijk} \left(\mathbf{w}_{ij} \cdot \mathbf{x}_k + b_{ij} \right) \right),$$

이때, $\mathbf{w}_{ij}$는 클래스 i와 클래스 j를 구분하는 결정 경계의 가중치 벡터이고, b_{ij}는 절편 (bias)을 나타내고,

$$y_{ijk} = \begin{cases} 1, & \text{if } \mathbf{x}_k\text{가 클래스}i\text{에 속할 때} \\ -1, & \text{if } \mathbf{x}_k\text{가 클래스}j\text{에 속할 때} \end{cases}$$

가 된다. $\mathbf{x}_k$는 식 (2.1)에서 J_t^j 벡터에 해당한다. 이로부터 클래스 i와 클래스 j에 속하는 관측값들을 최대한 분리할 수 있도록 $\mathbf{w}_{ij}$와 b_{ij}를 최적화한다. 또한, C는 이 방법의 편의-분산 절충에 영향을 미치는 조율 모수이다. 예를 들어, C값이 크면 마진이 넓어지고 더 많은 마진 위반이 허용되어 편의는 커지지만 분산이 작아지는 분류기를 얻게 된다.

2.2.3. 랜덤포레스트

랜덤포레스트는 다범주 분류 문제에서 효과적인 앙상블 학습 방법이다. 이 방법은 다수의 결정 트리를 학습시키고, 각 트리가 개별적으로 예측한 결과를 종합하여 최종 예측을 하게 된다. 분류 문제에서는 각 트리의 예측 결과 중 다수결 투표를 통하여 최종 클래스를 결정한다. 랜덤 포레스트는 여러 결정 트리의 예측을 결합함으로써 단일 결정 트리가 가지는 과적합 문제를 줄이고, 예측의 정확성과 안정성을 높이게 된다. 각 결정트리는 붓스트랩 샘플링을 통해 생성된 데이터로 학습되며, 분할 시 랜덤하게 선택된 예측변수의 부분 집합을 사용하게 된다. 만약 예측변수 J_t^j에 대한 각 트리의 예측값을 $f_t(J_t^j)$라 하고, 총 T개의 트리가 있다고 할 때, 랜덤 포레스트의 최종 예측 클래스 $\hat{y}$는 다음과 같다.

$$\hat{y} = \text{mode}\left(f_t\left(J_t^j \right)_{t=1}^{T} \right). \tag{2.2}$$

2.3. 딥러닝 접근법

딥러닝 모델에서는 일반적으로 3차원 관절좌표를 직접 입력 변수로 사용한다. 이러한 접근법은 신경망이 자동으로 동작 패턴을 학습하며, 각 프레임의 관절 위치와 움직임을 직접 반영하여, 시간적 및 공간적 관계를 더 정밀하게 이해할 수 있도록 한다. 예를 들어, Veeriah 등 (2015)은 시간에 따라 연속적인 관절 좌표 벡터 시퀀스를 처리했으며, Zhang 등 (2020) 은 스켈레톤 시퀀스를 3차원 배열로 변환하였다. 그러나 이러한 딥러닝 모델은 예측변수의 수가 많은 스켈레톤 데이터의 경우 계산적 비용이 머신러닝 모델에 비해 매우 크다는 단점이 존재한다. 본 논문에서는 스켈레톤 데이터에 대한 딥러닝 모델로 순환 신경망(recurrent neural network; RNN) 기반의 모델과 그래프 합성곱 신경망(graph convolutional network; GCN) 기반의 모델을 적합하고자 한다.

2.3.1. HBRNN (hierechical bidirectional recurrent neural network)

Rumelhart 등 (1986)의 순환 신경망(RNN)은 순차적으로 관측되는 데이터를 분석하기 위해 설계된 기본적인 신경망 유형이다. RNN은 각 시점 단계에서 업데이트되는 숨겨진 상태(hidden state)를 유지함으로써 시간적

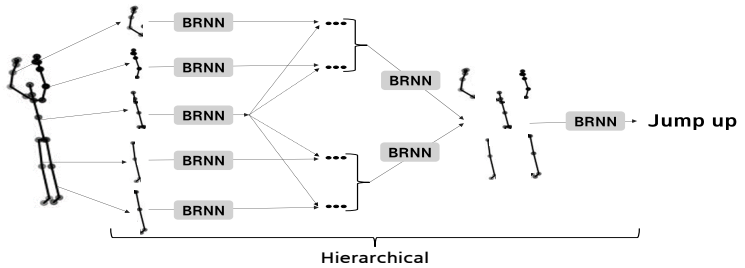


Figure 3: *Illustration of the hierarchical bidirectional RNN algorithm adapted from Du et al. (2015).*

종속성을 표현할 수 있고, 시점 t에서의 은닉층은 다음 공식을 사용하여 계산된다.

$$h_t = \tanh\left(W_h h_{t-1} + W_J J_t^j + b\right),$$

여기서 W_h와 W_J는 모든 시점에서 일관된 가중치 행렬을 의미하며, 하이퍼볼릭 탄젠트(hyperbolic tangent) 함수는 출력이 −1과 1 사이로 유지되도록 보장한다. 그러나 시퀀스의 길이가 길어질수록 RNN은 기울기 소실 문제(gradient vanishing)로 인해 장기적인 종속성을 학습하기 어려운 한계가 있다. 이를 해결하기 위해 Hochreiter 등 (1997)은 LSTM (long short-term memory)을 도입하여 장기 종속성을 보다 효과적으로 표현할 수 있는 게이팅 메커니즘을 사용한다. 텍스트 데이터 처리와 같이 과거와 미래의 문맥을 모두 이해해야 하는 분석에는 Schuster 등 (1997)의 양방향 RNN (bidirectional RNN)이 사용된다. 양방향 RNN은 시퀀스를 순방향과 역방향 모두에서 처리하여 데이터에 대한 보다 포괄적인 이해를 제공한다. 이러한 개념을 바탕으로 Du 등 (2015)이 제안한 HBRNN (hierarchical bidirection recurrent neural network)은 RNN의 기능을 확장하여 복잡한 인간 움직임을 더 잘 인식한다. HBRNN은 인간의 관절 좌표 데이터를 팔 2개, 다리 2개, 몸통 등 5개 부분으로 나누어 모델링한다. 동작에 따라 한 개의 관절만 작용하는 동작도 있지만 대다수의 동작에는 여러 부분에 걸쳐 조화로운 동작이 필요하다. HBRNN은 이러한 신체 부위의 움직임과 상호 작용을 동시에 모델링한다. Figure 3에 보이는 것처럼 특징 벡터는 5개 관절 좌표 각각에 대해 고유한 가중치를 가진 양방향 RNN을 사용하여 추출된다. 그런 다음 이러한 특징 벡터를 결합하여 인접한 부위 간의 상호 작용을 모델링한다. 이 계층적 프로세스는 양방향 RNN을 사용하여 상체와 하체의 결합된 움직임을 반영하는 기능을 생성하면서 반복된다. 궁극적으로 모든 관절 좌표의 정보를 통합하여 전체 시퀀스의 특징을 추출하고, 마지막으로 소프트맥스 함수가 적용된 완전 연결 레이어를 사용하여 동작을 분류한다. HBRNN이 계층적으로 쌓은 양방향 RNN 구조는 개별 신체 부위의 움직임과 그 조합을 모델링함으로써 RNN, LSTM, 양방향 처리의 장점을 결합하여 인간 행동을 인식할 수 있을 것으로 기대된다.

2.3.2. SGN (semantics-guided neural network)

Kipf와 Welling (2016)의 GCN (graph convolutional network)은 그래프의 구조적 특성을 활용하여 노드(node)와 엣지(edge)에 대한 정보를 효과적으로 학습하며, 소셜 네트워크, 화학 분자 구조 등 다양한 분야에서 활용된다. 스켈레톤 데이터의 경우, GCN을 사용하여 인접한 관절들 간의 연결관계를 그래프 형태로 표현할 수 있다. 그래프 데이터는 노드(정점, V)와 엣지(연결, E)로 구성된다. GCN에서는 그래프를 인접 행렬(A)과 특징 행렬(F)로 표현한다. 인접 행렬 A는 그래프의 노드 간 연결 상태를 나타내며, 노드 i와 노드 j간의 연결 여부는 0 또는 1로 표시한다. 특징 행렬 F는 각 노드의 특징 벡터를 행으로 가지며, f개의 특징을 포함한다. GCN의

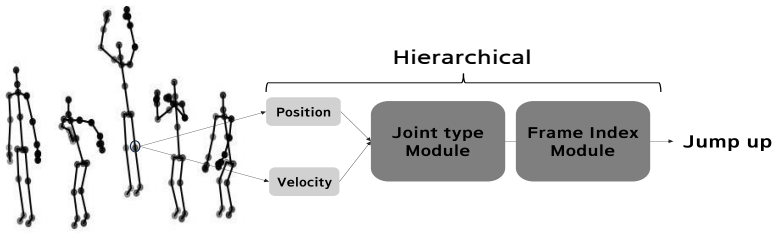


Figure 4: *Illustration of the semantics-guided neural network algorithm adapted from Zhang et al. (2020).*

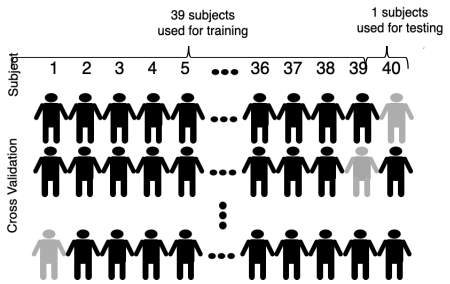


Figure 5: *Cross-Subject Cross-Validation.*

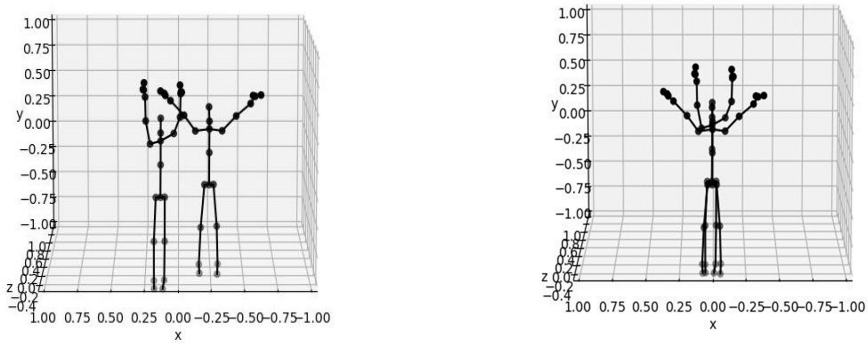
연산은 다음과 같은 수식으로 표현된다.

$$H^{l+1} = \sigma\left(AH^{l}W^{l}\right),$$

여기서 H^{l}은 l번째 은닉층에서의 특징 행렬, W^{l}는 l번째 가중치 행렬이며, $\sigma(\cdot)$는 비선형 활성함수이다. GCN은 인접 행렬을 통해 국소적인 정보를 활용하고, 특징 행렬의 모든 노드에 동일한 가중치를 적용함으로써 합성곱 연산(CNN)과 유사한 특성을 갖는다. 이러한 구조를 통해 GCN은 그래프의 복잡한 구조를 효과적으로 학습할 수 있다. 이러한 GCN의 원리를 바탕으로 Zhang 등 (2020)에서 제안한 semantics-guided neural network (SGN) 모델은 Figure 4에서와 같이 3D 스켈레톤 데이터를 3차원 배열로 변환하여, 공간적 및 시간적 신체 구조의 중요한 정보를 각각 '관절 수준 모듈(joint-level module)'과 '프레임 수준 모듈(frame-level module)'로 처리하며, 이러한 모듈을 계층적으로 탐색하여 동작을 인식한다. 각 관절좌표의 위치와 속도(현재 프레임과 이전 프레임간 위치 변화) 정보를 결합한 동적 표현(dynamic representation)을 관절 수준 모듈(joint-level module)에서 $J \times J$ 단위행렬과 결합하고 3개의 GCN 층을 통해 관절 유형 정보를 명확히 학습한다. 이후 프레임 수준 모듈(frame-level module)에선 $T \times T$ 단위행렬을 결합함으로써 프레임 인덱스를 명확히 학습하게 된다. 그래프 구조를 효과적으로 처리하는 GCN을 바탕으로 SGN은 두 모듈을 계층적으로 활용하여 신체 구조의 공간적 정보와 시간 정보를 효과적으로 학습하고, 복잡한 동작도 정확하게 인식한다.

Table 1: Confusion matrix for the k-th action: '1' indicates correct action, '0' indicates other actions

		$\widehat{\gamma_k}$		
		0	1	Sum
γ_k	0	a_k	b_k	$a_k + b_k$
	1	c_k	d_k	$c_k + d_k$
	Sum	$a_k + c_k$	$b_k + d_k$	n_k



(a) (b)

Figure 6: *Joint coordinates for two different subjects performing the 'Cheer up' action: (a) before normalization, (b) after normalization.*

2.4. 분류성능

동작분류 모델은 실제 응용분야에서 새로운 사람의 동작을 정확하게 인식해야 한다. 만약 모델이 학습 데이터에 포함된 사람들의 동작만 인식할 수 있다면 실제 환경에서의 유용성이 제한될 수 있다. 스켈레톤 데이터는 대부분 실험 데이터로 수집되는데, 주로 각 실험 참가자의 동작이 반복 측정되고, 반복 측정된 데이터는 독립적인 관측값으로 간주하여 분석된다. 만약 전체 데이터를 학습 데이터와 평가 데이터로 단순랜덤표집으로 분할하면, 동일한 실험 참가자의 데이터가 양쪽에 모두 포함될 수 있다. 이는 모델의 실제 성능을 과대 평가하게 된다. 따라서 본 연구에서는 Figure 5에서 보여주는 것처럼 실험 참가자 별 교차검증(cross-subject cross-validation; CSCV)을 수행하여 모델의 일반화 성능을 효과적으로 평가하고 과적합 위험을 줄이고자 하였다. 즉, CSCV는 각 실험 참가자를 번갈아 평가 데이터로 사용하고, 나머지 참가자들을 학습 데이터로 사용하는 방법이다. 이 결과 동작의 분류성능은 동작의 개수를 K라 할때, 정오분류표(confusion matrix) $K \times K$행렬로부터 정확도(accuracy)나 오분류율(misclassifcation rate) 등으로 평가할 수 있다. 동작별 성능을 좀 더 자세히 평가하기 위해 $k(= 1, \ldots, K)$번째 동작의 분류결과에 대하여 Table 1과 같은 정오 분류표를 생성한다. 여기서 γ_k는 실제 k번째 동작이 발생했는지 여부를 나타내는 이진형 변수이고, $\widehat{\gamma_k}$는 각 분류모형이 k번째 동작 발생 여부를 예측한 이진형 결과이다. 여기서 정확도(accuracy)는 $(a_k + d_k)/n_k$, 민감도(sensitivity)는 $d_k/(c_k + d_k)$, 정밀도(precision)는 $d_k/(b_k + d_k)$, F1 점수(F1 score)는 $2d_k/\,(b_k + c_k + 2d_k)$로 요약된다. 또한 최종 성능평가는 이러한 동작별 성능지표를 40명의 평가 데이터에 대한 평균 성능으로 제시하도록 한다.

Table 2: Comparison of the number of predictors in machine learning and deep learning models

	Machine learning	Deep learning
Predictors	Summary statistics (mean, variance, skewness, kurtosis, range)	Joint coordinates
No. of predictors	375 (25 joints x 3D x 5 statistics)	1,500 (25 joints x 3D x 20 frames)

3. 실제 데이터 분석

이 절에서는 실제 데이터를 소개하고, 머신러닝 모델과 딥러닝 모델을 적합한 후 그 성능을 비교 검토해 보고자 한다.

3.1. 데이터

이 논문에서는 Shahroudy 등 (2016)이 소개한 'NTU RGB+D' 데이터를 활용하여 60개 동작에 대한 분류 모델을 구축하였다. 이 데이터는 40명의 실험 참가자가 60가지 동작을 수행한 것으로, 일부 동작은 2~3회 반복되었으며, 3대의 카메라를 사용하여 높이와 거리를 달리한 17가지 서로 다른 조건에서 총 56,880개의 관측값이 수집되었다. 본 연구에서는 이 중 C002 카메라로 정면에서 촬영된 21,496개의 데이터만을 사용한다. 또한 동작별로 프레임의 길이가 다르지만 본 연구에서는 각 동작의 프레임 수를 20개로 제한하였다. 이를 위해 첫 번째 프레임과 마지막 프레임을 포함하여 등간격으로 프레임을 추출하였다. 이렇게 추출된 관절좌표는 i번째 사람의 j번째 관절이 t번째 프레임에서 $J_{it}^j = (x_{it}^j, y_{it}^j, z_{it}^j), i = 1, \ldots, 21,496, j = 1, 2, \ldots, 25, t = 1, 2, \ldots, 20$로 표현된다. 또한, 2.1절에서 소개한 대로 데이터를 정규화하기 위해 관절의 중심값을 Figure 2의 'Spine_Mid'로 선택하였다. Figure 6(a)는 서로 다른 참가자가 'cheer up' 동작을 수행하는 초기 프레임의 3D 관절좌표를 정규화 전 상태로 나타낸 것이다. Figure 6(b)는 정규화 이후의 모습을 나타내며, 두 프레임의 초기 스켈레톤 좌표가 훨씬 유사해졌음을 알 수 있다. 본 연구에서는 동작 분류를 위해 머신러닝 모델에 25개 관절에 대해 20개 프레임에서 관측된 3차원 각 좌표로부터 5가지 통계량(평균, 분산, 왜도, 첨도, 범위)만을 고려하여 총 $25 \times 3 \times 5 = 375$개의 예측변수를 사용했다. 반면, 딥러닝 모델 적합을 위해서는 20개 프레임에서 관측된 3차원 위치 좌표를 모두 사용하여 총 $25 \times 3 \times 20 = 1,500$개의 예측 변수를 사용하였다 (Table 2 참조). 이 데이터로부터 2.2절에서 소개한 머신러닝 모델을 사용하여 분류모델을 구축하였으며, 이를 위해 R 소프트웨어에서 MASS (LDA), e1071 (SVM), randomForest (RF) 패키지를 사용했다. 또한, 2.3절에서 소개한 딥러닝 모델로는 RNN 기반의 HBRNN 모델 (Du 등, 2016)과 GCN 기반의 SGN 모델 (Zhang 등, 2020)을 적합하였다. SGN 모델은 Zhang 등 (2020)에서 제공한 오픈소스 코드 `https://github.com/microsoft/SGN`를 참고하였고, HBRNN 모델의 오픈소스 코드는 비공개로 제공되지 않아 자체적으로 코드를 작성하여 네트워크를 구축하였다. 딥러닝 모델의 구현은 Python 소프트웨어의 tensorflow와 Keras를 활용하였다.

3.1.1. CSCV를 통한 모델 분류 성능 비교

실험 참가자별로 평가 데이터를 만들어 교차검증을 실시한 후, 정오분류표를 작성하여 60개 동작에 대한 정확도를 모델별로 계산한 결과는 Table 3과 같다. 이 표의 결과를 살펴보면 모델 간의 유의미한 차이보다는 특정 동작에서 모든 모델이 일관되게 낮거나 높은 정확도를 보이는 경향이 있음을 알 수 있다. 즉, 동작에 따라 비슷한 성능 패턴을 가지는 경향을 알 수 있다. 예를 들어, 동작 A9 (standing up)의 경우 모든 모델에서 80% 이상의 높은 정확도를 기록한 반면, 동작 A11 (reading), A12 (writing)의 경우 모든 모델에서 20%대의 낮은 정확도를 나타내었다. 또한 동작 A10 (clapping), A44 (touch head)에서는 딥러닝 모델의 정확도가 머신러닝 모델의 정확도보다 상대적으로 약간 높게 나타났고, 반대로 동작 A26 (hopping), A27 (jump up)에서는 머신

Table 3: Accuracy for 60 actions across various models

Cluster	Action	LDA	SVM	RF	HBRNN	SGN
1	A8. sitting down	0.86	0.88	0.86	0.77	0.86
	A9. standing up (from sitting position)	0.84	0.94	0.95	0.83	0.91
	A27. jump up	0.92	0.93	0.92	0.66	0.89
	A43. falling	0.78	0.86	0.86	0.68	0.86
2	A6. pick up	0.75	0.87	0.85	0.58	0.67
	A14. wear jacket	0.76	0.84	0.87	0.67	0.82
	A15. take off jacket	0.76	0.77	0.78	0.66	0.73
	A22. cheer up	0.71	0.80	0.82	0.78	0.73
	A26. hopping (one foot jumping)	0.81	0.84	0.82	0.67	0.73
	A35. nod head/bow	0.88	0.77	0.81	0.70	0.72
	A42. staggering	0.84	0.78	0.70	0.54	0.72
	A59. walking towards each other	0.61	0.73	0.73	0.67	0.65
3	A7. throw	0.56	0.60	0.64	0.63	0.63
	A20. put on a hat/cap	0.53	0.57	0.67	0.43	0.53
	A24. kicking something	0.58	0.65	0.62	0.57	0.57
	A36. shake head	0.59	0.62	0.73	0.51	0.43
	A38. salute	0.64	0.65	0.64	0.58	0.68
	A39. put the palms together	0.59	0.59	0.65	0.58	0.55
	A40. cross hands in front (say stop)	0.54	0.61	0.69	0.72	0.73
	A46. touch back (backache)	0.63	0.53	0.51	0.64	0.68
	A48. nausea or vomiting condition	0.64	0.61	0.63	0.62	0.64
	A55. hugging other person	0.46	0.62	0.61	0.57	0.69
4	A1. drink water	0.34	0.36	0.47	0.46	0.45
	A2. eat meal/snack	0.46	0.45	0.49	0.40	0.52
	A3. brushing teeth	0.52	0.52	0.55	0.49	0.58
	A4. brushing hair	0.33	0.39	0.49	0.49	0.42
	A5. drop	0.42	0.41	0.39	0.49	0.49
	A13. tear up paper	0.43	0.50	0.56	0.46	0.52
	A16. wear a shoe	0.36	0.43	0.41	0.40	0.43
	A21. take off a hat/cap	0.36	0.38	0.46	0.42	0.51
	A23. hand waving	0.38	0.42	0.46	0.49	0.46
	A25. reach into pocket	0.50	0.42	0.53	0.48	0.51
	A28. make a phone call/answe phone	0.28	0.31	0.32	0.45	0.45
	A29. plating with phone/tablet	0.41	0.47	0.45	0.39	0.41
	A30. typing on a keyboard	0.42	0.52	0.61	0.37	0.41
	A31. pointing to something with finger	0.41	0.34	0.39	0.43	0.50
	A32. taking a selfie	0.39	0.37	0.37	0.48	0.44
	A33. check time (from watch)	0.50	0.42	0.47	0.52	0.60
	A37. wipe face	0.47	0.44	0.52	0.44	0.41
	A41. sneeze/cough	0.51	0.50	0.50	0.48	0.55
	A45. touch chest (stomachache/heart pain)	0.43	0.38	0.38	0.52	0.62
	A47. touch neck (neckache)	0.33	0.36	0.38	0.42	0.51
	A49. use a fan (with hand or paper)/feeling warm	0.30	0.29	0.29	0.46	0.50
	A51. kicking other person	0.36	0.49	0.53	0.40	0.46
	A52. pushing other person	0.40	0.51	0.53	0.40	0.46
	A54. point finger at the otehr person	0.39	0.50	0.53	0.41	0.43
	A56. giving something to other person	0.41	0.39	0.42	0.46	0.47
	A58. handshaking	0.44	0.52	0.57	0.56	0.59
	A60. walking apart from each other	0.53	0.76	0.72	0.55	0.70
5	A10. clapping	0.20	0.18	0.20	0.39	0.34
	A11. reading	0.22	0.27	0.21	0.27	0.25
	A12. writing	0.20	0.24	0.26	0.23	0.26
	A17. take off a shoe	0.29	0.43	0.36	0.35	0.34
	A18. wear on glasses	0.30	0.33	0.38	0.40	0.36
	A19. take off glasses	0.26	0.24	0.26	0.42	0.43
	A34. rub two hands together	0.38	0.37	0.43	0.38	0.38
	A44. touch head (headache)	0.25	0.25	0.20	0.40	0.40
	A50. punching/slapping other person	0.24	0.31	0.34	0.38	0.44
	A53. pat on back of other person	0.16	0.24	0.31	0.33	0.39
	A57. touch other person's pocket	0.17	0.23	0.23	0.35	0.43

Actions are clustered into five groups based on F1 scores from each action's confusion matrix against all other actions, with Cluster 1 having the highest accuracy and Cluster 5 the lowest.

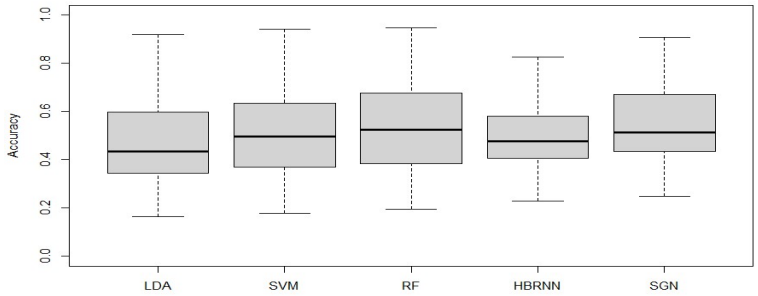


Figure 7: *Boxplot of accuracy for 60 actions across various models.*

Table 4: Comparison of model performance across various models based on each action's confusion matrix against all other actions, with the performance evaluated using cross-subject cross-validation

Metrics	Models				
	LDA	SVM	RF	HBRNN	SGN
Sensitivity	0.47 (0.31,0.63)	0.51 (0.35,0.67)	0.53 (0.37,0.69)	0.50 (0.34,0.66)	0.57 (0.41,0.73)
Precision	0.48 (0.32,0.64)	0.52 (0.36,0.68)	0.54 (0.38,0.70)	0.50 (0.34,0.66)	0.55 (0.39,0.71)
F1_score	0.47 (0.31,0.63)	0.51 (0.35,0.67)	0.53 (0.37,0.69)	0.50 (0.34,0.66)	0.56 (0.40,0.72)

러닝 모델의 정확도가 높게 나타났다. 모델별 정확도를 좀 더 쉽게 비교하기 위해 Table 3에 대해 상자그림을 작성한 결과는 Figure 7과 같다. 평균 정확도는 SGN, RF, SVM, HBRNN, LDA의 순으로 높게 나타났지만, 앞서 언급했듯이, 모델간의 유의미한 차이는 없다. 그러나 딥러닝 모델의 경우 정확도의 최솟값이 머신러닝 모델보다 높았으며, 반대로 정확도의 최댓값은 머신러닝이 높게 나타났다. 실제로, 동작 A9 (standing up)의 경우 RF 모델에서 정확도가 0.95로 가장 높았으며, 분류성능이 낮은 동작들에 관해서는 딥러닝 모델의 정확도가 머신러닝 모델의 정확도보다 약간 높게 나타나는 것을 확인할 수 있다. 이들 분류성능을 좀 더 자세히 정리하기위해 동작별 정오분류표(Table 1 참고)를 작성하여 모델별로 비교한 결과는 Table 4와 같다. Figure 7의 모델별 정확도에서 살펴본 바와 같이 SGN 모델이 다른 모델에 비하여 모든 동작에 대한 평균 성능이 약간 더 우수한 것으로 나타났다. SGN 모델의 경우 동작별 평균 민감도(sensitivity)가 0.57 (95% 신뢰구간: 0.41, 0.73), 평균 정밀도(precision)는 0.55 (95% 신뢰구간: 0.39, 0.71), 그리고 평균 F1 점수는 0.56 (95% 신뢰구간: 0.40, 0.72)로 가장 높은 분류 성능을 보였다. 그러나 신뢰구간을 비교했을 때 서로 다른 모델들과의 성능 차이는 뚜렷하지 않은 것을 알 수 있다.

3.2. F1 점수에 기반한 군집 분석

이 절에서는 동작별 정오분류표(Table 1 참고)에서 구한 F1점수를 모델별로 구하고, 이로부터 60개 동작을 k-평균 군집 분석으로 군집화한다. F1 점수는 민감도와 정밀도의 조화평균으로 두 지표를 모두 중요하게 고려한 지표이다. 이를 통해 동작들이 어떻게 군집화 되는지 파악하고, 군집별 분류성능을 평가하고자 한다. 군집의 크기를 결정하기 위해 군집의 개수를 x축, 군집 내 중심과 데이터 간 잔차 제곱합을 y축으로 한 Figure 8의 스크리 그림(scree plot)을 통해 동작을 5개의 군집으로 나누었다. 이 군집분석 결과를 시각화하기 위해, 모델별 F1 점수로 구성된 5차원 군집 특성에 대해 주성분 분석을 실시하였다. 그 결과, 첫 번째 주성분이 전체 데이터 변동성의 약 94.84%를 설명하고, 두 번째 주성분이 추가로 약 3.07%를 설명하여, 이 두 주성분만으로도 전체 데이터의 변동을 97.91%로 설명할 수 있어 Figure 9에 시각화하였으며, 오른쪽에 위치한 군집은 가장 높은 F1 점수를, 왼쪽으로 갈수록 낮은 F1 점수를 가진 그룹을 나타낸다. 즉, 군집 1> 군집 2>군집 3>

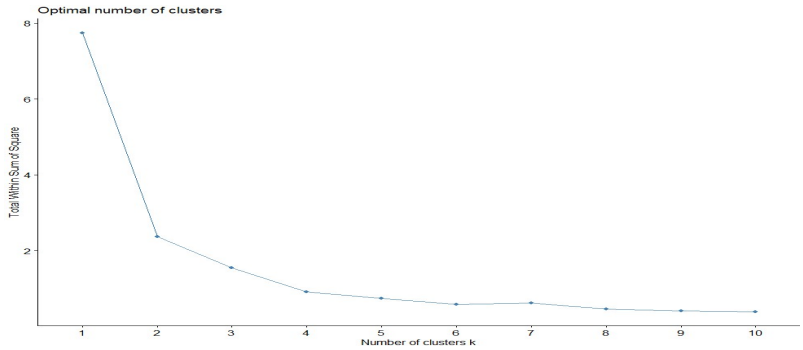


Figure 8: *Scree plot for determining optimal number of clusters.*

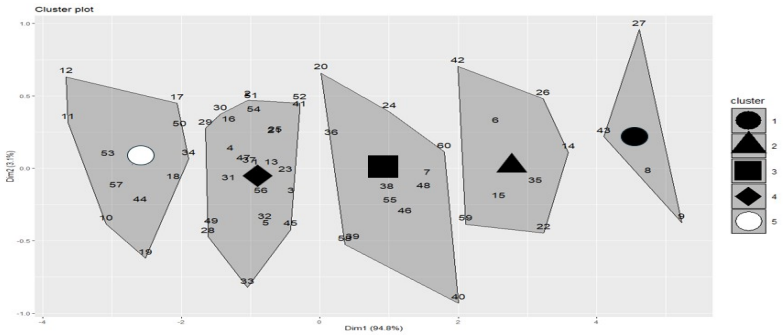
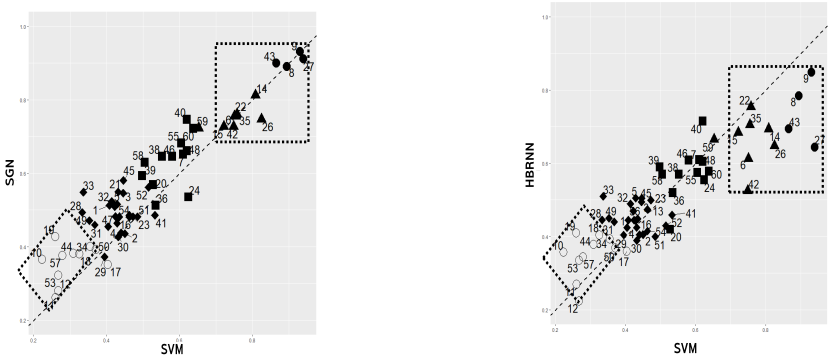


Figure 9: *Visualization of k-means clusters.*

군집 4> 군집 5의 순으로 F1 점수가 높음을 알 수 있다. Table 5를 통해 이러한 경향은 모든 모델에서 일치함을 확인할 수 있다. 군집 1에는 동작 A8 (sitting down), A9 (standing up from sitting down), A27 (jump up), A43 (falling) 등이 속해 있으며, 이 동작들은 3차원 프레임이 연속해서 관측될 때 동작의 변화가 상대적으로 크기 때문에 구별이 쉬운 동작들이라 할 수 있다. Table 5에서 확인하듯이 HBRNN 모델을 제외하고 80%이상의 높은 F1 점수를 나타낸다. 반면, 군집 4와 5는 F1 점수가 50% 이하로 낮아 분류성능이 저조하다. 특히, 군집 4의 동작 A3 (brushing teeth)나 A4 (brushing hair) 등처럼 대부분의 관절은 의미가 없고 팔 관절처럼 특정한 관절의 움직임만 있는 동작이나, 군집 5의 A11 (reading)이나 A12 (writing)과 같이 관절 변화가 적거나, 동작 A18 (wear on glasses), A19 (take off glasses)와 같이 비슷한 동작이 포함되어 있는 경우에는 분류성능이 낮은 편이다. 군집 2와 3에 속한 동작들의 경우에는 평균 F1 점수가 약 60점 이상으로 양호한 성능을 보였다. 이제 머신러닝 모델과 딥러닝 모델 간의 성능을 구체적으로 비교하기 위해, 머신러닝 모델의 대표적 모델로 다중 클래스 SVM을 선택하여 두 개의 딥러닝 모델과 비교하였다. Figure 10(a)에는 SVM 모델의 동작별 F1 점수와 SGN 모델의 F1 점수를, Figure 10(b)에는 SVM 모델의 동작별 F1 점수와 HBRNN 모델의 F1 점수를 나타냈다. 각 그림 내 점선은 $y = x$ 직선을 나타낸다. Figure 10(a)에서 비교적 구별하기 쉬운 단순한 동작들이 속한 군집 1과 군집 2에서는 성능값이 $y < x$인 영역에 존재하여 SVM 모델의 F1 점수가 높다는 것을 알 수 있다. 이는 요약 통계량을 예측 변수로 사용하는 머신러닝 모델이 큰 움직임이 필요한 동작을 효과적으로 인식할 수 있음을 보여 준다. 그러나 군집 3부터 군집 5에서와 같이 다소 복잡한 동작의 경우에는 딥러닝 모델이 높은 F1 점수를 보여 준다. Figure 10(b)에서도 HBRNN 모델은 분류가 어려운 동작에서 더 높은 성능을 보였고, 군집 1과 2에서는 SVM 모델의 성능이 더 우수한 것으로 나타났다. 실제로 Figure 10에서 박스로 표시된 부분

Table 5: Average F1 scores across cluster (cluster sizes)

	F1 score					
Cluster (cluster size)	LDA	SVM	RF	HBRNN	SGN	평균
1 (4)	0.83	0.91	0.87	0.74	0.91	0.85
2 (8)	0.70	0.75	0.72	0.66	0.75	0.71
3 (12)	0.54	0.58	0.60	0.58	0.63	0.59
4 (25)	0.40	0.43	0.46	0.45	0.49	0.45
5 (11)	0.26	0.30	0.33	0.35	0.36	0.32



(a) (b)

Figure 10: *F1 score comparison across different actions: (a) SGN vs. SVM and (b) HBRNN vs. SVM.*

(군집 5와 군집 1,2)에 대해 윌콕슨 부호순위검정을 실시한 결과, 유의수준 5%에서 군집 5의 경우는 SGN과 HBRNN 모델이, 군집 1~2에 대해서는 SVM 모델이 유의미하게 더 높은 F1 점수를 갖는 것으로 나타났다. 이 결과를 통해 HBRNN과 SGN 같은 딥러닝 모델이 복잡한 패턴 인식과 세부적인 움직임 학습에 머신러닝 학습보다 강점을 가지고 있음을 알 수 있다. 또한, 동작이 크고 분류가 어렵지 않은 경우에는 두 모델에 대한 성능 차이는 크지 않다고 생각해 볼 수 있다. 이러한 연구 결과는 동작의 구체적인 특성과 각 모델의 장점을 고려하여 최적의 모델을 선택하고 적용하는 데 중요한 기준을 제공한다.

마지막으로 분류성능이 낮은 동작에 대해 좀 더 살펴보고자 한다. 예를 들어, 동작 A10 (clapping)과 A33 (check time) 의 경우를 살펴보기로 한다. Table 6은 이들 두 동작을 SVM 모델과 SGN 모델로 분류했을 때, 가장 많이 오분류한 동작의 예시를 보여준다. 이 두 동작은 모두 공통적으로 동작 A34 (rub two hands together)와 A39 (put palms together)로 잘못 분류되는 빈도가 가장 높았다. 동작 A10의 경우, SVM 모델은 동작 A34로 오분류한 경우가 오분류 전체의 19%, 동작 A39로 오분류한 경우가 16%, 동작 A13으로 오분류한 경우가 5%에 해당한다. 반면 SGN 모델은 같은 동작을 각각 13%, 4%, 2%로 잘못 분류하여, 전반적으로 더 낮은 오분류율을 나타내었다. 그런데, 여기서 동작 A10 (clapping), A34 (rub two hands together), A33 (check time) 의 프레임을 Figure 11에서 비교해 보면 사용되는 관절이나 움직임 정도가 매우 비슷한 것을 확인할 수 있다. 따라서, 프레임 내 좌표의 요약 통계량을 사용하는 머신러닝 모델의 경우 높은 분류성능을 기대하기 어려웠을 것이다. 마찬가지로 동작 A33 (check time)을 가장 많이 오분류한 동작들을 살펴보면 SVM 모델이 동작 A34로 9%, 동작 A39로 7%, 동작 A53 (pat on back of other person)으로 3% 잘못 분류한 반면, SGN 모델은 각각 5%, 1%, 2%로 잘못 분류하여, 이 또한 SGN 모델의 오분류율이 더 낮음을 확인할 수 있었다. 특히, 동작 A10과 달리 동작 A33은 오분류되는 동작이 다양한 카테고리에 분포되어 있음을 확인할 수 있다. 이러한 오분류

Table 6: Examples of misclassification between actions A10 (clapping) and A13 (checking time)

Actual	Predictied	SVM		SGN	
		No. of cases	Proportion	No. of cases	Proportion
10. Clapping	34. Rubbing hands	56	19%	46	13%
	39. Put palms together	50	16%	46	4%
	13 Tear up paper	17	5%	14	2%
33. Check time	34. Rubbing hands	28	9%	7	5%
	39. Put palms together	22	7%	12	1%
	53. Pat on back of other person	11	3%	5	2%

패턴 분석 결과, 비슷한 동작들 간의 미세한 차이를 구분해야 하는 경우에 딥러닝 모델인 SGN의 분류성능이 상대적으로 우수함을 알 수 있었다.

4. 결론

본 연구는 3D 스켈레톤 데이터에서 동작을 인식하는 머신러닝 모델과 딥러닝 모델 간 분류 성능을 비교분석하였다. 머신러닝의 경우, 전체 프레임에서 각 관절 위치 좌표의 분포를 요약한 5개의 통계량(평균, 분산, 왜도, 첨도, 범위)을 LDA, SVM, RF 의 예측변수로 사용하였고. 딥러닝의 경우, 각 프레임의 모든 관절 위치 좌표를 HBRNN 모델, GCN 계열의 SGN 모델의 예측 변수로 사용하였다. NTU RGB+D 데이터를 활용하여 60개 동작을 분류하고, 40명의 실험 참가자별 교차점증(CSCV)을 통해 모델의 일반화 분류성능을 효과적으로 평가하였다. 분석 결과, 분류성능이 모델의 종류보다 동작 유형에 의해 큰 영향을 받는 것을 알 수 있었다. 동작에 따라 분류성능이 매우 다르게 나타났기 때문에 60개 동작 전체에 대한 정확도는 모든 모델에서 높지 않았다. 모델별 분류성능 지표(정확도, 민감도, 정밀도, F1점수)는 SGN 모델이 가장 높았지만, 신뢰구간을 비교했을 때 유의미한 성능 차이는 없었다. 심층적 성능평가를 위해 모델별 F1 점수에 따라 60개 동작에 대한 군집분석을 실시한 결과, 동작의 복잡성(관절의 수와 움직임의 정도)에 따라 5개의 군집으로 나뉘어졌고, 군집 간 분류성능의 차이가 명확했다. 각 군집별로 SVM 모델과 딥러닝 모델을 적용한 결과, 큰 움직임의 동작에는 머신러닝 모델, 작은 움직임의 동작에는 딥러닝 모델이 효과적인 것으로 나타났다.

이 연구에서는 NTU RGB+D 데이터의 일부만을 사용했는데, 추후 계산 환경을 개선하여 전체 데이터를 사용하여 결과를 비교하는 것도 의미가 있을 것으로 생각된다.

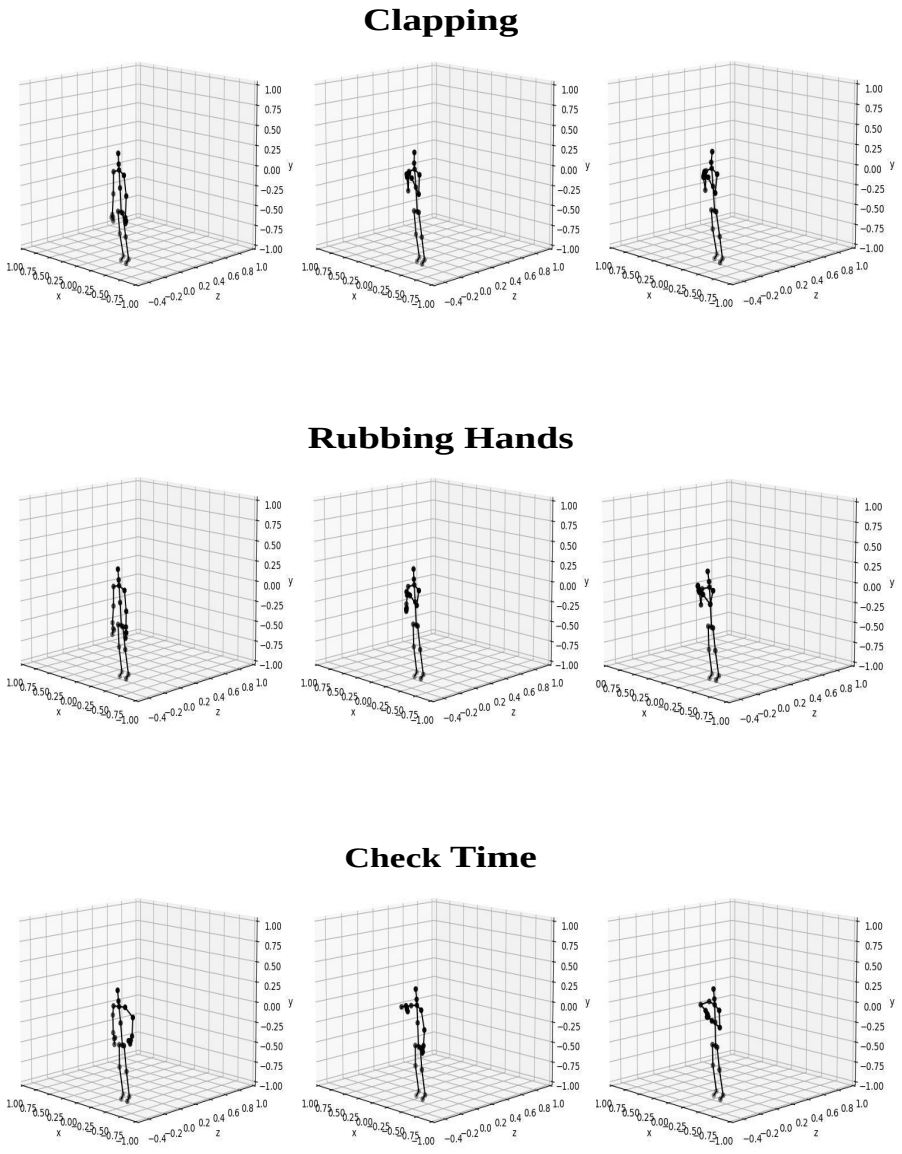


Figure 11: *Examples of actions with lower classification performance: Clapping, rubbing hands, and checking time.*

References

Amor BB, Su J, and Srivastava A (2015). Action recognition using rate-invariant analysis of skeletal shape trajectories, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **38**, 1–13.

Cao C, Lan C, Zhang Y, Zeng W, Lu H, and Zhang Y (2018). keleton-based action recognition with gated convolutional neural networks, *IEEE Transactions on Circuits and Systems for Video Technology*, **29**, 3247–3257.

Chaaraoui AA, Padilla-Lopez JR, and Florez-Revuelta F (2015). Abnormal gait detection with RGB-D devices using joint motion history features, *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition*, **7**, 1–6.

Cho K and Chen X (2014). Classifying and visualizing motion capture sequences using deep neural networks, *2014 International Conference on Computer Vision Theory and Applications*, **2**, 122–130.

Du G, Zhang P, Mai J, and Li Z (2012). Markerless kinect-based hand tracking for robot teleoperation, *International Journal of Advanced Robotic Systems*, **9**, 36.

Du Y, Fu Y, and Wang L (2015). Skeleton based action recognition with convolutional neural network, In *Proceedings of 2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, Kuala Lumpur, 579–583.

Du Y, Wang W, and Wang L (2015). Hierarchical recurrent neural network for skeleton based action recognition, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1110–1118.

Ghazal S, Khan US, Mubasher Saleem M, Rashid N, and Iqbal J (2019). Human activity recognition using 2D skeleton data and supervised machine learning, *IET Image Processing*, **13**, 2572–2578.

Gregor K, Danihelka I, Graves A, Rezende D, and Wierstra D (2015). Draw: A recurrent neural network for image generation, *International Conference on Machine Learning*, **37**, 1462–1471.

Grushin A, Monner DD, Reggia JA, and Mishra A (2013). Robust human action recognition via long short-term memory, In *Proceedings of The 2013 International Joint Conference on Neural Networks (IJCNN)*, Dallas, TX, 1–8.

Hochreiter S and Schmidhuber J (1997). Long short-term memory, *Neural Computation*, **9**, 1735–1780.

Izenman AJ (2008). *Modern Multivariate Statistical Techniques*, Springer, New York.

Jalal A, Uddin MZ, and Kim TS (2012). Depth video-based human activity recognition system using translation and scaling invariant features for life logging at smart home, *IEEE Transactions on Consumer Electronics*, **58**, 863–871.

Jeong H and Lim C (2019). A review of artificial intelligence based demand forecasting techiques, *The Korean Journal of Applied Statistics*, **32**, 795–835.

Jeong YS and Park Jh (2018). 3D skeleton animation learning using CNN, *Asia-pacific Journal of Multimedia Services Convergent with Art, Humanities, and Sociology*, **8**, 281–288.

Jin X, Yao Y, Jiang Q, Huang X, Zhang J, Zhang X, and Zhang K (2015). Virtual personal trainer via the kinect sensor, In *Proceedings of 2015 IEEE 16th International Conference on Communication Technology*, Hangzhou, 406–463.

Kang YK, Kang HY, and Weon DS (2021). Human skeleton keypoints based fall detection using GRU, *Journal of the Korea Academia-Industrial Cooperation Society*, **22**, 127–133.

Ke Q, Bennamoun M, An S, Sohel F, and Boussaid F (2017). A new representation of skeleton sequences for 3d action recognition, *In Proceedings of the IEEE conference on computer vision and pattern recognition*, 3288–3297.

Kim W, Kim D, Park KS, and Lee S (2023). Motion classification using distributional features of 3D skeleton data, *Communications for Statistical Applications and Methods*, **30**, 551–560.

Kipf TN and Welling M (2016). Semi-supervised classification with graph convolutional networks, Available from: *arXiv preprint arXiv:1609.02907*

Lee I, Kim D, Kang S, and Lee S (2017). Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks, In *Proceedings of the IEEE international conference on computer vision*, 1012–1020.

Lee J, Lee M, Lee D, and Lee S (2023). Hierarchically decomposed graph convolutional networks for skeleton-based action recognition, In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 10444–10453.

Lefebvre G, Berlemont S, Mamalet F, and Garcia C (2013). BLSTM-RNN based 3D gesture classification, *Artificial Neural Networks and Machine Learning–ICANN 2013: 23rd International Conference on Artificial Neural Networks Sofia, Bulgaria*, **23**, 381–388.

Li C, Zhong Q, Xie D, and Pu S (2017). Skeleton-based action recognition with convolutional neural networks. *In 2017 IEEE international conference on multimedia & expo workshops*, 597–600.

Li C, Zhong Q, Xie D, and Pu S (2018). Co-occurrence feature learning from skeleton data for action recognition and detection with hierarchical aggregation, Available from: *IarXiv preprint arXiv:1804.06055*

Lin BS, Wang LY, Hwang YT, Chiang PY, and Chou WJ (2018). Depth camera based system for estimating energy expenditure of physical activities in gyms, *IEEE Journal of Biomedical and Health Informatics*, **23**, 1086–1095.

Liu J, Shahroudy A, Xu D, and Wang G (2016). Spatio-temporal lstm with trust gates for 3d human action recognition, *Computer Vision–ECCV 2016: 14th European Conference*, **14**, 816–833.

Reddy VR and Chattopadhyay T (2014). Human activity recognition from kinect captured data using stick model, *International Conference on Human-Computer Interaction*, 305–315.

Rumelhart DE, Hinton GE, and Williams RJ (1986). Learning representations by back-propagating error, *Nature*, **323**, 533–536.

Sandra M (2020). *Clustering Gestures using Multiple Techniques*, Digital Sciences Tilburg University, Tilburg, The Netherlands.

Schuster M and Paliwal KK (1997). Bidirectional recurrent neural networks, *IEEE Transactions on Signal Processing*, **45**, 2673–2681.

Shahroudy A, Liu J, Ng TT, and Wang G (2016). Ntu rgb+ d: A large scale dataset for 3d human activity analysis, In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1010–1019.

Shan J and Akella S (2014). 3D human action segmentation and recognition using pose kinetic energy, In *Proceedings of 2014 IEEE International Workshop on Advanced Robotics and Its Social Impacts*, Evanston, IL, 69–75.

Shin BG, Kim UH, Lee SW, Yang JY, and Kim W (2021). Fall detection based on 2-stacked Bi-LSTM and human-skeleton keypoints of RGBD camera, *KIPS Transactions on Software and Data Engineering*, **10**, 491–500.

Taha A, Zayed HH, Khalifa ME, and El-Horbaty ESM (2015). Human activity recognition for surveillance applications, *In Proceedings of the 7th International Conference on Information Technology* , 577—586.

Tao W, Liu T, Zheng R, and Feng H (2012). Gait analysis using wearable sensors, *Sensors*, **12**, 2255–2283.

Xu H, Gao Y, Hui Z, Li J, and Gao X (2023). Language knowledge-assisted representation learning for skeleton-

based action recognition, Available from: *arXiv preprint arXiv:2305.12398*

Veeriah V, Zhuang N, and Qi GJ (2015). Differential recurrent neural networks for action recognition, In *Proceedings of the IEEE International Conference on Computer Vision*, 4041–4049.

Yang Y, Yan H, Dehghan M, and Ang MH (2015). Real-time human-robot interaction in complex environment using kinect v2 image recognition, *In 2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics*, 112–117.

Zhang P, Lan C, Zeng W, Xing J, Xue J, and Zheng N (2020). Semantics-guided neural networks for efficient skeleton-based human action recognition, In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1112–1121.

Zhu W, Lan C, Xing J, Zeng W, Li Y, Shen L, and Xie X (2016, March). Co-occurrence feature learning for skeleton based action recognition using regularized deep LSTM networks, *Proceedings of the AAAI Conference on Artificial Intelligence*, **30**.

Received July 31, 2024; Revised August 24, 2024; Accepted August 29, 2024

스켈레톤 데이터에 기반한 동작 분류: 고전적인 머신러닝과 딥러닝 모델 성능 비교

김주환[a], 김종찬[a], 이성임[1,b]

[a]단국대학교 응용통계학과; [b]단국대학교 통계데이터사이언스학과

요 약

본 연구는 3D 스켈레톤 데이터를 활용하여 머신러닝 및 딥러닝 모델을 통해 동작 인식을 수행하고, 모델 간 분류 성능 차이를 비교 분석하였다. 데이터는 NTU RGB+D 데이터의 정면 촬영 데이터로 40명의 참가자가 수행한 60가지 동작을 분류하였다. 머신러닝 모델로는 선형판별분석(LDA), 다중 클래스 서포트 벡터 머신(SVM), 그리고 랜덤 포레스트(RF)가 있으며, 딥러닝 모델로는 RNN 기반의 HBRNN (hierarchical bidirectional RNN) 모델과 GCN 기반의 SGN (semantics-guided neural network) 모델을 적용하였다. 각 모델의 분류 성능을 평가하기 위해 40명의 참가자별로 교차 검증을 실시하였다. 분석 결과, 모델 간 성능 차이는 동작 유형에 크게 영향을 받았으며, 군집 분석을 통해 각 동작에 대한 분류 성능을 살펴본 결과, 인식이 비교적 쉬운 큰 동작에서는 머신러닝 모델과 딥러닝 모델 간의 성능 차이가 유의미하지 않았고, 비슷한 성능을 나타냈다. 반면, 손뼉치기나 손을 비비는 동작처럼 정면 촬영된 관절 좌표만으로 구별하기 어려운 동작의 경우, 딥러닝 모델이 머신러닝 모델보다 관절의 미세한 움직임을 인식하는 데 더 우수한 성능을 보였다.

주요용어: 스켈레톤 데이터, 머신러닝 모델, 딥러닝 모델, 교차검증

이 성과는 정부 (과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2019R1A2C1003257).
[1]교신저자: (16890) 경기도 용인시 수지구 죽전로 152, 단국대학교 통계데이터사이언스학과. E-mail: silee@dankook.ac.kr