KIPS Trans. Softw. and Data Eng.
Vol.10, No.11 pp.491~500
ISSN: 2287-5905 (Print), ISSN: 2734-0503 (Online)

https://doi.org/10.3745/KTSDE.2021.10.11.491

Fall Detection Based on 2-Stacked Bi-LSTM and Human-Skeleton Keypoints of RGBD Camera

Shin Byung Geun† · Kim Uung Ho†† · Lee Sang Woo††† · Yang Jae Young††† · Kim Wongyum††††

ABSTRACT

In this study, we propose a method for detecting fall behavior using MS Kinect v2 RGBD Camera-based Human-Skeleton Keypoints and a 2-Stacked Bi-LSTM model. In previous studies, skeletal information was extracted from RGB images using a deep learning model such as OpenPose, and then recognition was performed using a recurrent neural network model such as LSTM and GRU. The proposed method receives skeletal information directly from the camera, extracts 2 time-series features of acceleration and distance, and then recognizes the fall behavior using the 2-Stacked Bi-LSTM model. The central joint was obtained for the major skeletons such as the shoulder, spine, and pelvis, and the movement acceleration and distance from the floor were proposed as features of the central joint. The extracted features were compared with models such as Stacked LSTM and Bi-LSTM, and improved detection performance compared to existing studies such as GRU and LSTM was demonstrated through experiments.

Keywords : Fall Detection, Deep-learning, Skeleton Keypoints, Stacked Bi-LSTM

RGBD 카메라 기반의 Human-Skeleton Keypoints와 2-Stacked Bi-LSTM 모델을 이용한 낙상 탐지

신 병 근† · 김 응 호†† · 이 상 우††† · 양 재 영††† · 김 원 겸††††

요 약

본 연구에서는 MS Kinect v2 RGBD 카메라 기반의 Human-Skeleton Keypoints와 2-Stacked Bi-LSTM 모델을 이용하여 낙상 행위를 탐지하는 방법을 제안한다. 기존의 연구는 RGB 영상에서 OpenPose 등의 딥러닝 모델을 이용하여 골격 정보를 추출한 후 LSTM, GRU 등의 순환신경망 모델을 이용해 인식을 수행하였다. 제안한 방법은 카메라로부터 골격정보를 바로 전달 받아 가속도 및 거리의 2개의 시계열 특징을 추출한 후 2-Stacked Bi-LSTM 모델을 이용하여 낙상 행위를 인식하였다. 어깨, 척추, 골반 등 주요 골격을 대상으로 중심관절을 구하고 이 중심관절의 움직임 가속도와 바닥과의 거리를 특징으로 제안하였다. 추출된 특징은 Stacked LSTM, Bi-LSTM 등의 모델과 성능비교를 수행하였고 GRU, LSTM 등의 기존연구에 비해 향상된 검출 성능을 실험을 통해 증명하였다.

키워드 : 낙상 탐지, 딥러닝, 골격 정보, 다층 양방향 LSTM

1. 서 론

인구 고령화가 심화됨에 따라 1인 노인가구의 수도 증가하고 있다. 낙상은 근력이 약화된 노인들에게서 흔히 발생하는 사고로, 고관절 골절 등으로 이어질 경우 매우 심각한 결과를 초래할 수 있어 즉각 탐지와 대응이 필수적으로 요구된다.

최근 스마트홈 환경에서 휴대폰이나 카메라, 레이더, 가속도계, 자이로스코프 등의 다양한 센서를 통해 낙상을 탐지하고자 하는 연구가 진행되고 있다. 하지만 가속도계, 자이로스코프 등을 웨어러블 형태로 구성하여 신체에 부착하는 센서 기반의 탐지 방법은 부착에 따른 사용자의 불편을 야기하기 때문에 아직까지는 그 효과가 낮다고 할 수 있다. 카메라를 이용한 탐지 방법의 경우에는 촬영된 영상으로부터 OpenPose [9]나 PoseNet[10] 등의 딥러닝 모델을 이용하여 사람 객체나 골격 정보(skeleton keypoints)를 검출한 후 다시 트리 기반의 머신러닝 모델이나 RNN(Recurrent Neural Net),

※ 이 논문은 2021년도 정부(산업통상자원부)의 재원으로 한국산업기술평가관리원의 지원을 받아 수행된 연구임(No.20009899, 지능형 케어 서비스 개발).

† 정 회 원 : ㈜아이와즈 연구원
†† 정 회 원 : ㈜아이와즈 책임연구원
††† 정 회 원 : ㈜에이아이딥 선임연구원
†††† 종신회원 : ㈜에이아이딥 연구소장

Manuscript Received : September 30, 2021
First Revision : October 14, 2021
Accepted : October 17, 2021

* Corresponding Author : Kim Wongyum(wgkim@aideep.ai)

LSTM(Long short-Term Memory) 등의 딥러닝 모델을 이용하여 낙상 상황을 탐지하는 연구가 진행되고 있다[1,3,5,7,8].

본 논문에서는 카메라에서 추출된 골격 정보에서 추출된 가속도 및 거리 특징을 기반으로 Bi-LSTM(Bi-directional LSTM) 모델을 이용하여 낙상을 탐지하는 방법을 제안한다. 스켈레톤 정보를 습득하기 위해 영상으로부터 골격 정보만을 제공하는 MS사의 Kinect v2를 사용하였다. 또한 낙상 상황의 전후 정보를 깊이 있게 모두 반영하기 위해 Bi-LSTM 모델을 층(Stacked)으로 구성하여 사용함으로써 인식 정확도를 개선하였다.

2장에서는 낙상탐지에 대한 관련 연구를 서술하며 3장에서는 제안하는 낙상탐지 방법을 서술한다. 4장에서는 실험환경 및 데이터 수집 방법, 실험결과를 서술하고 마지막으로 5장에서는 연구의 성과를 고찰한다.

2. 관련 연구

카메라를 이용한 전통적인 낙상 탐지 연구는 영상에서 배경을 삭제한 후 사람의 모양이나 실루엣을 특성화하여 낙상 상황을 감지한다. 하지만 최근에는 영상 처리의 부담을 줄이기 위해 영상에서 스켈레톤 정보를 추출하여 낙상 상황을 탐지하는 연구가 활발히 진행되고 있다[1-3,5,7,8,17].

인체의 스켈레톤 정보는 2D 혹은 3D의 카메라 영상으로부터 OpenPose[9]나 PoseNet[10] 등과 같은 CNN 기반 모델을 사용해 추출된다. 추출된 스켈레톤 정보는 낙상이 일어나는 시간에 따라 변화하는 시공간(spatial-temporal) 데이터로 간주된다. 따라서 시계열 데이터의 분류가 가능한 순환신경망(RNN)을 사용한 연구가 시도되었다. 하지만 순환신경망 모델은 학습 시간이 길고 기울기 값이 사라지는 문제(vanishing gradient problem)가 존재하여 이를 개선한 LSTM과 GRU(Gated recurrent unit) 모델을 이용한 낙상 탐지 방법이 제안되고 있다[3-5,8,17].

GRU 모델은 3개의 게이트를 갖는 LSTM과 달리 Reset 게이트(r)와 Update 게이트(z)를 통한 간단한 구조로, LSTM 모델의 긴 학습시간을 개선한 모델이다. 최근에는 양방향 LSTM(Bi-LSTM) 모델과 웨어러블 레이더 센서를 이용한 낙상 탐지 방법이 제안되었다[2,6]. 양방향 LSTM 모델은 순방향과 역방향 두 개의 분리된 LSTM을 통해 학습하는 모델로, 기존의 순방향만 갖는 LSTM 모델에 역방향 레이어를 은닉층으로 추가하여 입력 순서에 따라 수렴하는 LSTM의 한계를 보완하였다.

인체의 스켈레톤 정보를 이용한 낙상 탐지 연구는 특징으로 사용하는 관절 종류에 따라 원시 골격 데이터(SD, Skeleton Data)[3,8], 머리와 어깨 관절(HSSC, Head and Shoulder Segment Keypoint Coordinator)[5], 머리와 어깨 관절의 움직임 가속도(VHSSC, Velocity of HSSC)[5], 그리고 골격의 가로세로비(RWHC, Ratio of body Width and Height Coordinator)[1,5] 등으로 구분된다.

연구 [1]에서는 엉덩이 관절의 가속도와 지면과의 각도, 신체의 가로-세로 비율(RWHC)을 낙상 탐지의 특징으로 사용하였다. 제안한 방법에서는 자체적으로 수집된 실험영상으로부터 추출된 스켈레톤에서 엉덩이 관절 중간포인트의 감속가속도와 스켈레톤 센터라인과 지면과의 각도, 골격의 가로세로비에 대한 통계적 임계값을 추출하였다. 하지만 이런 임계값 추출 방법은 스켈레톤 정보가 정확하게 추출되지 않으면 많은 오차가 발생할 수 있고 또한 실험 데이터에 최적화될 수 있는 제약점을 갖는다.

연구 [3]에서는 영상으로부터 OpenPose를 이용해 스켈레톤을 추출하고 이중 11개의 관절을 선택하여 그 좌표벡터를 정규화 한 후 LSTM 모델을 이용하여 인식하는 방법을 제안하였다. 하지만 실험에서는 영상의 전면부나 가로형태의 낙상만을 대상으로 제한적인 성능 평가를 진행하였다.

연구 [8]에서는 영상으로부터 OpenPose를 이용해 스켈레톤을 추출하고 이중 15개의 관절을 선택하여 RNN, LSTM GRU 모델을 사용하여 낙상을 탐지하였다. 또한 OpenPose의 검출에러를 보완하기 위해 탐지된 관절을 이용한 보간 방법을 함께 제안하였다. 하지만 탐지된 관절 좌표를 그대로 학습하는 방법은 OpenPose 등의 스켈레톤 탐지 모델의 성능에 대한 종속성이 증가되며, 학습에 있어 상대적으로 많은 연산량이 요구되는 단점이 존재한다.

본 논문에서는 MS Kinect v2 카메라에서 획득된 스켈레톤 정보를 이용한 낙상 탐지 방법을 제안한다. 획득된 골격 정보에서 HSSC, VHSSC, RWHC 특징을 포함하는 중심관절(Central joint)의 가속도를 기본특징으로 제안하였고, 지면과 중심관절과의 거리를 추가 특징으로 제안하여 스켈레톤 전체 좌표를 사용하는 방법보다 특징의 계산을 간소화하였다. 또한 스켈레톤 검출 성능과의 종속성을 줄이고 중심관절의 추출 정확성을 높이기 위해 손과 손목, 발, 발목 등 검출 정확성이 낮은 관절은 배제하고 목, 어깨, 척추, 엉덩이, 무릎 등 몸통관절 기반으로 중심관절을 계산하였다. 또한 낙상 데이터 수집 시 촬영하는 각도를 동서남북의 4가지를 사용하였고 낙상 동작도 전후좌우 4가지 형태로 다양화 하였다. 이후 추출된 2개의 특징을 기반으로 전후 관계를 포함하는 2-Stacked Bi-LSTM 모델을 학습 모델로 사용해 최종 낙상 여부를 탐지함으로써 그 성능을 개선하였다.

3. 제안 방법

3.1 Human Skeleton Keypoints

본 논문에서는 인체에서 스켈레톤 정보를 추출하기 위해 Microsoft사의 Kinect v2 카메라와 Kinect SDK를 사용하였다[11]. Kinect SDK는 영상 내 존재하는 신체에 대해 25개의 관절 포인트와 아래 Table 1과 같이 관절 이름 및 인덱스

Table 1. MS Kinect v2 Keypoints

No	Joint	No	Joint	No	Joint	No	Joint
0	SpineBase	7	Hand_L	14	Ankle_L	21	Handtip_L
1	SpineMid	8	Shoulder_R	15	Foot_L	22	Thumb_L
2	Neck	9	Elbow_R	16	Hip_R	23	Handtip_R
3	Head	10	Wrist_R	17	Knee_R	24	Thumb_R
4	Shoulder_L	11	Hand_R	18	Ankle_R		
5	Elbow_L	12	Hip_L	19	Foot_R		
6	Wrist_L	13	Knee_L	20	SpineShoulder		

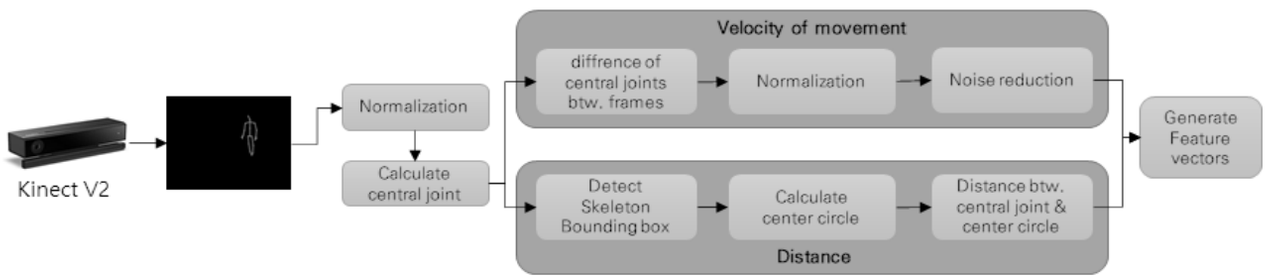


Fig. 1. Feature Extraction Process

(No)를 제공한다. Fig. 1은 Kinect v2에서 제공하는 관절의 위치를 나타낸다. Kinect v2에서 제공하는 각 관절 정보는 *(x, y, z)*의 3D 좌표계로 표현되는데 x, y는 2D 좌표계, *z*는 Kinect 센서와의 거리(depth)를 나타낸다.

3.2 특징 추출

본 논문에서는 낙상하는 인체의 골격은 빠르게 지면과 가까워지는 사실을 바탕으로 중심관절(Central joint) 포인트의 움직임 가속도(velocity)와 중심관절 포인트와 바닥과의 거리(Y-axis distance)를 특징으로 제안한다. 중심관절 포인트는 검출된 관절 중 몸통 등 주요 관절에 대한 중심점을 의미한다. 제안하는 특징 추출 과정은 Fig. 2와 같다.

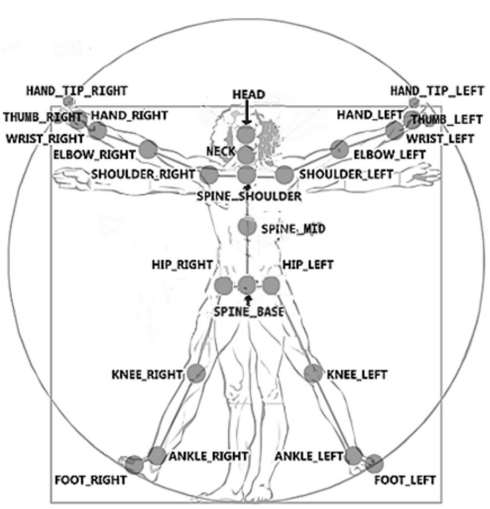


Fig. 2. MS Kinect v2 25-joint Skeleton
(출처: Stack Overflow)

첫 번째로 Kinect v2 카메라를 통해 추출된 스켈레톤 정보는 좌표정규화를 통해 단위벡터로 변환된다. 다음으로 안정적인 중심관절을 획득하기 위해 추출된 25개의 관절 중 Neck, Shoulder_L, Shoulder_R, SpineShoulder, SpineMid, Hip_R, Hip_L, Spinebase, Knee_R, Knee_L 관절을 선택하였다. 선택된 10개의 관절좌표를 가지고 중심관절의 좌표를 Equation (1)과 같이 계산한다.

$$(x_c, y_c, z_c) = (\frac{\sum_{i=0}^{N} x_i}{N}, \frac{\sum_{i=0}^{N} y_i}{N}, \frac{\sum_{i=0}^{N} z_i}{N}) \quad (1)$$

N은 선택된 관절의 개수를 나타낸다. 계산된 중심관절은 움직임 가속도 특징과 거리 특징을 계산하는데 입력으로 사용된다.

1) 중심관절의 움직임 가속도 특징

낙상 상황에서 골격의 움직임 속도를 표현하기 위한 움직임 가속도 특징은 프레임간 중심관절의 상대적인 차이값을 계산함으로써 구한다. 전 프레임에서의 중심관절 좌표와 현재 프레임에서의 중심관절 좌표간의 차이값에 L2 Norm을 사용하여 정량화하였다. 움직임 가속도 VN을 계산하는 수식은 Equation (2)와 같다.

$$V_N = \sqrt{(x_N - x_{N-1})^2 + (y_N - y_{N-1})^2 + (z_N - z_{N-1})^2} \quad (2)$$

N은 현재 프레임을 나타내며 *(xN, yN, zN)*는 현재 프레임에서의 중심관절 좌표를, *(xN-1, yN-1, zN-1)*는 이전 프레임에서의 중심관절 좌표 나타낸다. 정의된 움직임 가속도는 중심관절을 이루는 골격이 좌우나 상하로 빠르게 움직

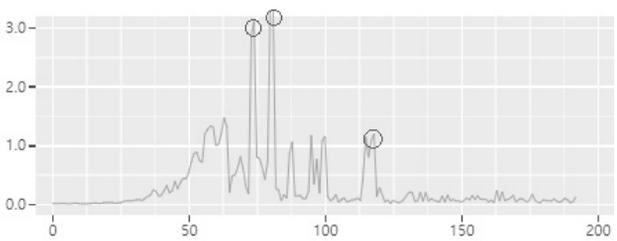


Fig. 3. Velocity of Movement Before Noise Reduction

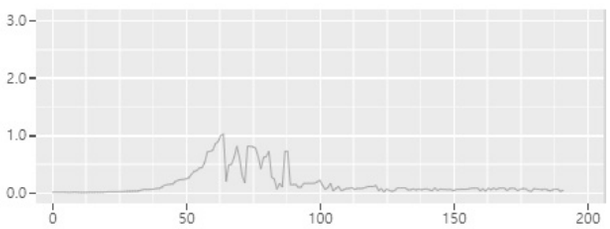


Fig. 4. Velocity of Movement After Noise Reduction

일 때 큰값으로 측정된다.

측정된 움직임 가속도는 낙상 구간 동안의 시계열 데이터로 나타난다. 하지만 짧은 구간 움직임처럼 보이는 피크형태의 노이즈가 존재한다. 현실적으로 큰 움직임을 보이기에는 아주 짧은 시간이기에 본 제안에서는 이런 피크 형태를 노이즈로 간주하고 이동평균 필터(Moving average filter)와 Robust scaler을 통해 제거하였다. 이동 평균 필터는 모든 측정 데이터가 아닌 지정된 개수의 최근 데이터만를 이용해 계산한 평균이다. Robust scaler는 중앙값(median)과 IQR (Interquartile range)를 사용하여 아웃라이어의 영향을 최소화하는 스케일러다[21]. 이동평균 필터의 윈도우 사이즈는 실험적으로 5 프레임으로 설정하였다.

Fig. 3은 낙상 상황의 샘플 동영상에 대한 움직임 가속도 특징을 나타낸다. 짧은 기간에 피크 형태의 노이즈가 생성됨을 보여 주고 있다. Fig. 4는 제안한 노이즈 제거 필터를 적용한 결과를 보여 주고 있다.

2) 거리 특징

낙상 상황에서는 대부분의 골격은 빠르게 아래로 향한다. 이는 제안한 중심관절의 좌표에서 *(y)*의 값이 빠르게 감소하는 것으로 나타난다. 하지만 인체가 직립 해 있는 일반적인 상황에서의 낙상을 가정한다면 중심관절의 좌표는 넘어지는 방향으로의 좌우변환도 함께 발생한다. 즉 중심관절의 *(x, z)* 좌표는 서 있을 때의 *(x, z)* 좌표를 기준으로 그 상대거리가 빠르게 증가한다. 본 논문에서는 *(y)*의 좌표값과 *(x, z)* 좌표의 프레임간 상대거리를 거리 특징으로 제안하였다.

거리 특징은 1) 바운딩 박스(Bounding Box) 탐지, 2) 중심원 계산, 3) 중심관절과 중심원과의 거리 및 *(y)*의 좌표값 측정의 3단계로 계산된다. 바운딩 박스는 탐지된 골격을 모두 포함하는 최소 크기의 박스를 말한다. 바운딩 박스 탐지 후 골격의 높이(Height)를 획득하고 이를 중심원 계산에 활용한다.

중심원(Center circle)은 중심관절을 원점으로 하는 *(x, z)* 평면에서의 원을 나타낸다. 중심원의 반지름(r)은 바운딩 박스로부터 구해지는 골격 높이의 1/2로 설정했다. 생성된 중심원은 낙상상황에서 인체의 골격이 중심관절을 중심으로 얼마나 좌우로 이동했는지를 측정하는데 활용된다. Fig. 5는 검출된 골격에 대해 중심관절 좌표(흰색점)와 바운딩 박스(노란색), 중심원(파란색)을 검출한 결과를 보여주고 있다.

(x, z) 평면에서의 거리 특징은 골격의 중심관절의 좌표와 직립 상황으로 가정한 최초의 시점에서 구한 중심원과의 상대거리를 매 프레임마다 연속적으로 측정한다. 즉 중심원은 최초 한번만 계산되고 이후 중심관절은 매 프레임마다 새롭게 계산된다. 낙상의 상황에서는 중심관절의 좌표가 중심원에 점점 가까워지므로 작은 값으로 수렴한다.

다음 단계는 거리 특징을 낙상의 상황만으로 한정하여 측정의 정확성을 높이기 위해 RWHC를 사용해 직립과 낙상을 구별한다. 탐지된 바운딩 박스의 너비와 높이 정보를 이용해 골격의 높이와 넓이의 비율인 R_{wh}를 Equation (3)과 같이 구한다[1].

$$R_{wh} = \frac{Width}{Height} \tag{3}$$

일반적으로 인체가 직립했을 경우 R_{wh}값은 1보다 작으나 낙상 상황에서는 점점 증가해 1보다 큰 수를 갖는다. 본 제안

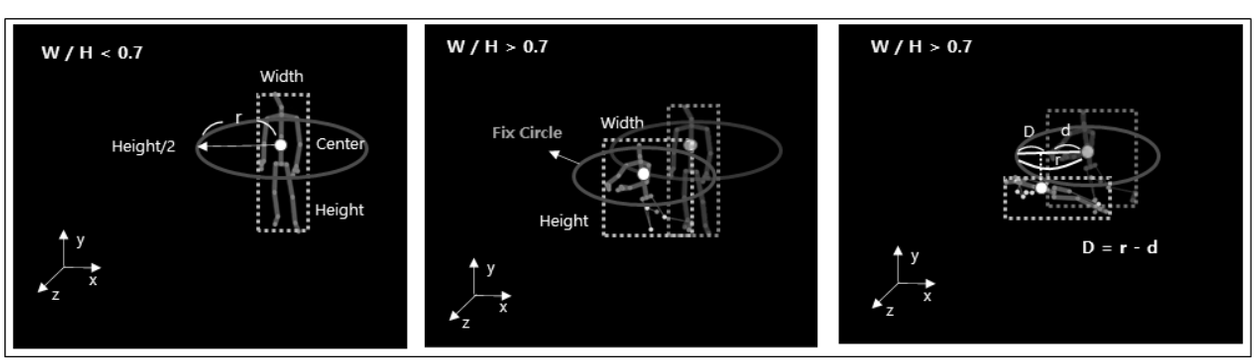


Fig. 5. Extraction of Bounding Box and Central Circle

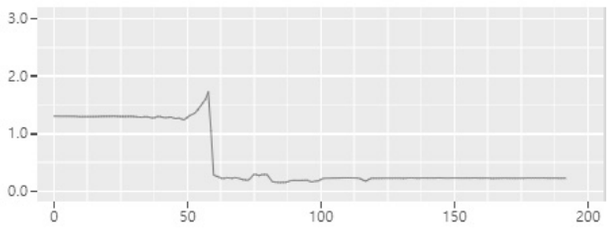


Fig. 6. Distance Feature

에서는 R_{wh}값이 특정 임계치*(T=0.7)*보다 클 경우 낙상의 시작이라고 보고 이 시점에서 중심원(center circle)을 고정하고 거리 측정의 기준으로 사용하였다. 즉 중심원의 고정은 중심원의 원점좌표를 고정한다는 것을 의미한다. 최종적으로 본 논문에서 제안하는 거리 특징*(D)*값은 Equation (4)와 같이 계산되어진다.

$$D=(r-\sqrt{(x_{center}-x_{circle})^2+(z_{center}-z_{circle})^2})+y_{center} \quad (4)$$

r은 중심원의 반지름, *(xcenter, zcenter)*은 중심관절의 좌표, *(xcenter, zcenter)*은 중심원의 원점 좌표, *(ycenter)*는 중심관절의 y 좌표값을 나타낸다. 움직임이 없는 스탠딩이나 걷기 상황의 경우 중심원이 매 프레임마다 다시 계산되기 때문에 거리 특징값의 변화가 미미하다. 하지만 낙상의 경우에는 Rwh값에 따라 중심원이 고정된 상태에서 중심원의 원점과 매 프레임 계산되는 중심관절과의 상대 거리와 중심관절의 바닥과의 거리(y)를 계산하므로 둘 다 작은 값으로 수렴하게 된다.

Fig. 6은 낙상 상황의 샘플 동영상에 대한 거리 특징을 보여 주고 있다. 처음 스탠딩 상황에서는 거의 일정한 값이 측정되다가 낙상의 순간에는 작은 값으로 급속히 감소하는 것을 볼 수 있다.

3.3 Stacked Bi-LSTM

시계열 데이터를 인식하고 예측하는 데 많이 활용되고 있는 순환신경망(RNN)과 장단기 기억 네트워크(LSTM) 모델은 은닉 계층에 과거의 정보를 기억하는 기능이 존재한다. 즉 기존 신경망은 입력층에서 은닉층, 출력층까지 한 번에 진행되는 반면 순환 신경망은 은닉층의 결과가 다음 은닉층의 입력으로 연결된다.

은닉층의 결과가 다시 은닉층의 입력으로 작용되기 때문에 시간적 흐름에 따른 데이터의 패턴을 분석하여 특정 상황이나 특정 시점의 결과 예측에 좋은 성능을 보이고 있다. 하지만 순환신경망에는 시계열의 데이터가 길수록 먼 과거의 정보를 현 예측 결과에 반영하지 못하는 구조적 단점이 존재한다. 이를 보완하기 위해 장단기 기억 네트워크나 GRU 모델이 제안되었고 낙상 상황에 적용된 연구가 진행되고 있다. 하지만 근본적으로 LSTM이나 GRU 모델도 입력 순서를 시간 순서대로 입력하기 때문에 결과물이 직전 패턴을 기반으로 수렴할 수밖에 없는 한계를 보인다[12]. 이런 단점을 해결하는 것을 목적으로 양방향 RNN 및 LSTM(Bi-LSTM) 방법이 제안되었다.

양방향 LSTM 모델은 순방향(forward)과 역방향(backward)의 2개의 분리된 LSTM을 통해 학습을 진행함으로써 시계열 입력을 양쪽 관점으로 분석이 가능해져 인식 및 예측 성능이 대폭 향상된 모델이다[6,13,14,20].

최근에는 웨어러블 센서 기반 낙상 탐지나 자연어 처리, 사운드 처리 분야 등에서 Bi-LSTM을 여러 개 겹친 형태의 Stacked Bi-LSTM 모델을 적용해 좋은 결과를 보여 주고 있다[6,15,16]. 본 연구에서는 Stacked Bi-LSTM 모델을 RGBD 카메라의 스켈레톤 기반 낙상 탐지 문제에 적용하여 향상된 인식 성능 결과를 도출하였다.

본 논문에서 사용한 Stacked Bi-LSTM 모델의 구조는 Ahmed 외 2명[20]이 사용한 모델에 Fully-connected 레이어를 쌓아 확장한 모델로 Fig. 7과 같다. 2개의 Bi-LSTM 모델을 사용하였고 아래층에는 128개 유닛, 위층에는 64 유닛으로 구성하였다. 2개의 Bi-LSTM 레이어 출력에는 접합[1] 노드가 존재하여 첫 번째 128개 노드의 출력을 두 번째 64개 노드의 입력으로 재구성한다. 두 번째 Bi-LSTM 레이어의 출력부분에도 접합 노드가 존재한다.

각각의 Bi-LSTM 모델에는 순방향 및 역방향 레이어가 포함되어 있다. Stacked Bi-LSTM의 최종 출력은 드랍아웃(Dropout) 레이어와 2개의 완전연결(Dense) 레이어를 거쳐 최종 1 비트의 결과(낙상 or 정상)를 출력하도록 구성하였다. 모델의 입력으로는 위에서 설명한 움직임 가속도와 거리, 2개의 특징 벡터가 128*2 형태로 Input shape을 구성하였다.

4. 실험 결과

4.1 실험데이터

본 논문에서 사용된 낙상 실험데이터는 MS Kinect v2 RGBD 카메라를 통해 직접 수집하였다. 사용된 카메라는 초당 30 프레임의 RGBD 영상으로부터 25개의 관절정보에 대한 좌표*(x, y)*와 깊이*(z)* 정보를 실시간으로 제공한다.

제한된 실내 공간에 매트릭스와 4대의 Kinect v2 카메라를 삼각대를 사용해 설치하였으며 일반 카메라를 추가로 설치하여 RGB 영상을 동시에 수집하였다. 데이터 수집을 위해 직접 구축한 수집 환경은 Fig. 8과 같다.

총 지원자 25명을 대상으로 넘어짐, 주저앉기의 2가지 낙상동작과 눕기, 앉기의 2가지 정상동작을 시연하게 하였고 동서남북의 4가지 카메라 방향과 전후좌우의 4가지 동작 방향으로 각각 나누어 수집하였다. 그리고 걷기 등의 일상 동작 2종에 대해 추가로 수집하였다. 수집된 데이터 수는 낙상동작 779건, 정상동작 511건으로 총 1,290건이다.

Fig. 9, 10은 수집된 낙상 및 정상 데이터에 대한 스켈레톤이 추출된 결과를 보여 주고 있다. 스켈레톤 이미지 아래

1) Concatenation

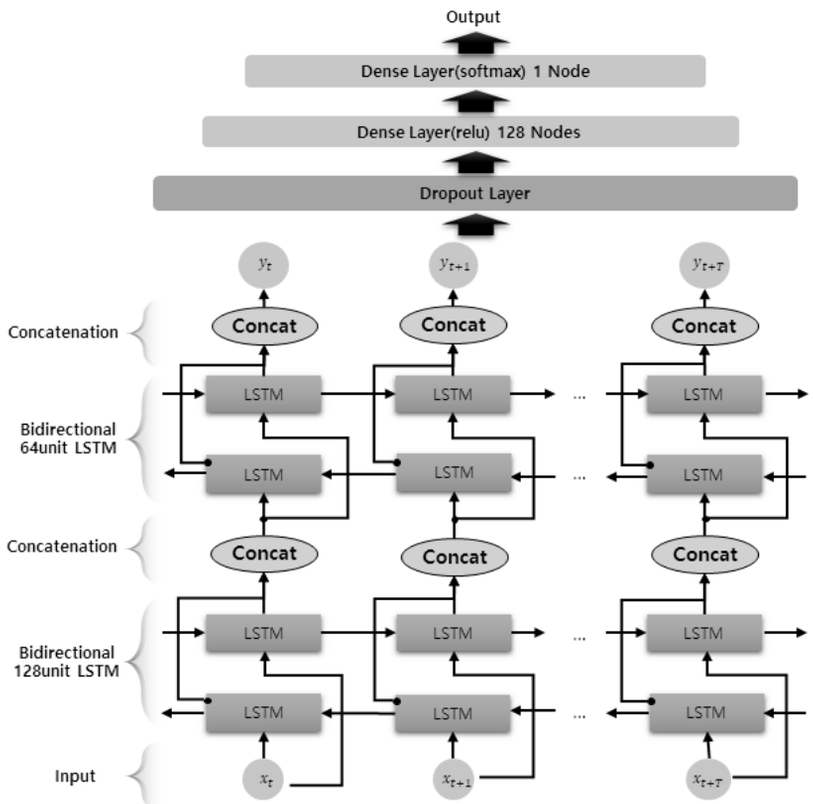


Fig. 7. 2-Stacked Bi-LSTM Structure

Fig. 8. Data Collection Environment

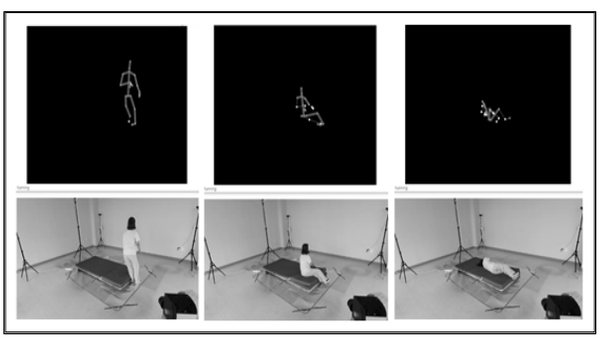


Fig. 9. Fall Data(Left-fall, North-right Side)

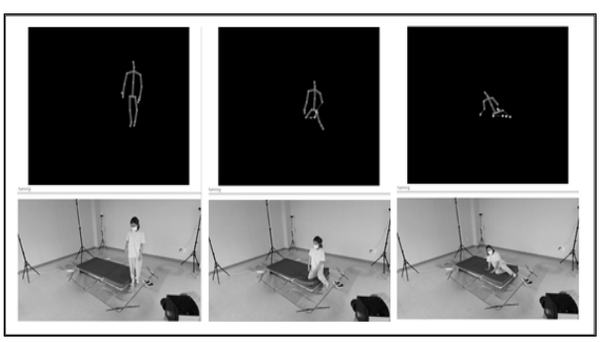


Fig. 10. Normal Data(Right Side Sitting, South-left Side)

나타나는 RGB 영상은 단지 시각화를 위해 다른 카메라로 따로 수집된 영상이며 본 연구의 입력에는 전혀 사용되지 않았다. MS Kinect v2 RGBD 카레라는 매 프레임마다 추출된 25개의 관절을 제공한다. 다만 정확히 검출되지 않은 손과 발 등의 관절 포인트는 예측 값으로 제공하며 그림에서 노란색으로 표시된다.

4.2 입력구성 및 성능비교

제안한 학습모델의 입력으로는 추출된 2개의 특징을 각각 115개 크기로 구성하였다. 사용된 카메라가 초당 30 프레임을 제공하므로 입력 구간은 약 3.83(110/30)초 정도로 설정하였다. 이는 낙상 행위를 충분히 포함하도록 실험적으로 설

정된 구간이다. 입력값은 낙상 동영상의 총 구간에서 230(115*2)개를 Fig. 11과 같이 슬라이딩 윈도우 방식으로 추출하였으며 추출 간격은 10 프레임(0.33초)으로 설정하였다. 추출된 특징은 115*2의 2D 배열로 정렬하였고 3가지 LSTM 모델 입력 유닛에 timestamp 115, feature 2로 설정하여 입력하였다. 슬라이딩 윈도우는 낙상 행위가 탐지되거나 동영상이 끝나면 중지된다. 수집된 데이터 1,290건 중 학습과 테스트를 위해 사용된 데이터 수는 아래 Table 2과 같다. 전체 데이터 중 90%를 학습에, 10%를 테스트에 사용하였다.

다음은 스켈레톤 동영상으로부터 제안한 방법으로 추출한 2가지의 특징을 보여 주고 있다. Fig. 12는 낙상 영상에 대한 특징이며 Fig. 13은 정상 영상에 대한 특징이다. 낙상 영상의 경우 거리와 가속도의 변화가 같은 시간대에 발생하는 반면 정상 영상의 경우에는 거리 특징의 감소가 먼저 나타난다. 가속도 특징도 낙상 데이터에 비해 뚜렷한 피크 형태를 보여주지 않고 있다.

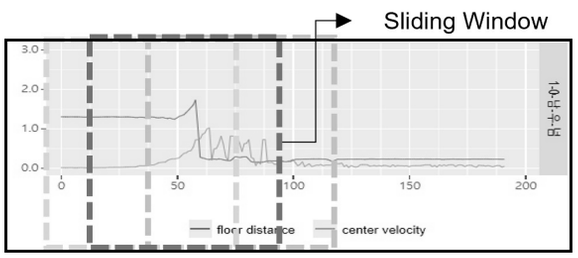


Fig. 11. Sliding Window

Table. 2. No. of Train and Test Data

	Train	Test	Total
Fall	718	61	779
Normal	443	68	511
Total	1,161	129	1,290

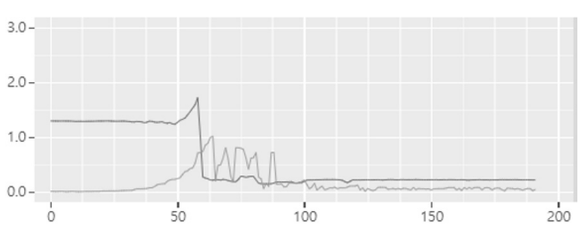


Fig. 12. Feature Extraction Result for (Fig. 9)

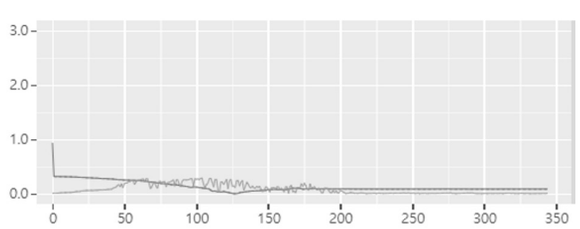


Fig. 13. Feature Extraction Result for (Fig. 10)

제안한 방법에 대한 성능비교는 스켈레톤을 이용한 관절 조합 알고리즘 4가지[1,3,5,8]와 3가지 딥러닝 모델과의 비교를 진행하였다. 비교를 위한 성능지표는 오차행렬(Confusion matrix)을 기반으로 정확도(Accuracy), 정밀도(Precision), 재현도(Recall), F1 Score를 Equation (5)~(8)과 같이 계산하였다.

$$Accuracy = \frac{TP+TN}{TP+FN+FP+TN} \tag{5}$$

$$Precision = \frac{TP}{TP+FP} \tag{6}$$

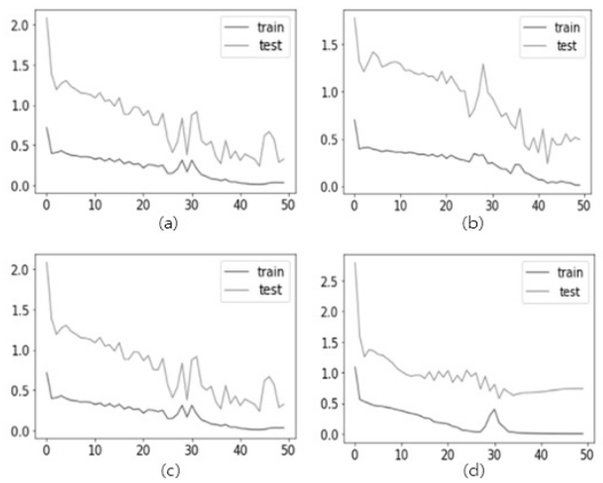
$$Recall = \frac{TP}{TP+FN} \tag{7}$$

$$F1\ Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{8}$$

기존 제안된 스켈레톤 기반의 알고리즘은 추출된 스켈레톤 전체를 이용하는 SD 방법[3,8]과 머리와 어깨 관절을 이용하는 방법(SD+HSSC)[5], SD와 머리와 어깨 관절의 가속도를 혼합하여 이용하는 방법(SD+VHSSC)[5], 마지막으로 머리와 어깨관절 포인트와 가속도, 스켈레톤 골격의 가로세로비를 혼합하여 이용하는 방법(HSSC+VHSSC+RWHC)이 있다[1,5].

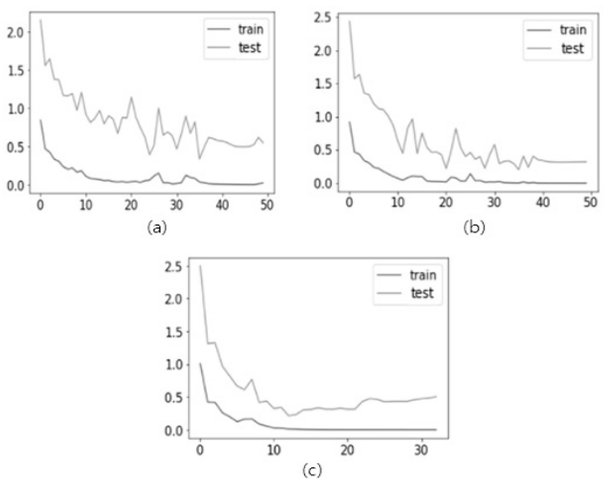
기존 알고리즘의 성능비교를 위한 학습모델은 연구[5,8]에서와 같이 256 유닛과 128 유닛의 2-Layer 구조를 갖는 GRU 모델을 사용하였다. GRU의 출력은 1개의 Dense layer와 1개의 Softmax layer를 통해 낙상 여부를 출력하도록 구성하였고 Optimizer는 Adam, learning rate은 0.0001, Dropout은 0.5, 손실함수는 categorical_crossentropy를 사용하였다. Epcho은 50, 배치 크기는 64를 설정하였다.

또한 LSTM의 단방향과 양방향 구조의 성능차이와 스택수에 따른 성능차이를 비교하기 위해 2-Stacked LSTM, Bi-LSTM, 2-Stacked Bi-LSTM의 3가지 딥러닝 모델에 대해 실험을 진행하였다. 2-Stacked LSTM의 구조는 128 유닛의 첫 번째 레이어와 64 유닛의 두 번째 레이어로 구성되었으며 출력은 1개의 Dense layer와 1개의 Softmax layer를 통해 낙상 여부를 출력하도록 구성하였다. Bi-LSTM 모델의 구조는 128 유닛을 갖는 1개의 레이어로 구성하였고 출력 구조는 2-Stacked LSTM 구조와 같게 하였다. 마지막으로 2-Stacked Bi-LSTM의 구조는 Fig. 7과 같으며 모델의 출력은 1개의 Dense layer와 1개의 Softmax layer를 통해 낙상 여부를 출력하도록 구성하였다. 3개의 모델에 대한 Optimizer는 Adam, learning rate은 0.001, Dropout은 0.5, 손실함수는 categorical_crossentropy를 사용하였다. Epoch는 50, Batch size는 64를 설정하였고 2-Stacked 모델의 각 레이어 출력에 대한 merge_mode는 'concat'을 사용하였다.



(a) SD (b) SD+HSSC (c) SD+VHSSC (d) HSSC+VHSSC+RWHC

Fig. 14. Lost Function Curve



(a) 2-Stacked LSTM (b) Bi-LSTM (c) 2-Stacked Bi-LSTM

Fig. 15. Lost Function Curve

Table 3. Comparison of Test Results

Model	Feature(Skeleton)	No. of features (frame)	Accuracy	Precision	Recall	F1
2-Layer GRU	SD	51[1)]	0.814	0.740	0.934	0.826
	SD+HSSC	52(51+1)	0.845	0.773	0.951	0.853
	SD+VHSSC	52(51+1)	0.876	0.817	0.951	0.879
	HSSC+VHSSC+RWHC	3	0.915	0.868	0.967	0.915
2-Stacked LSTM	Velocity & Distance of central joint (Proposed)	2	0.868	0.814	0.934	0.870
Bi-LSTM			0.915	0.879	0.951	0.913
2-Stacked Bi-LSTM			0.946	0.909	0.984	0.945

[1)]17개 관절 포인트에 대해 (x, y, z) 좌표 사용.

실험을 위한 모델은 Python(v3.7.7)으로 구현되었고 Tensorflow (v2.2.0), Keras(v 2.3.0), Numpy(v1.19.1), Scikit-learn (v0.23.1) 라이브러리를 사용하였다. 모델의 테스트는 MS Windows 10 OS에 32GB RAM, Intel i7-9700K CPU, GTX 1060 GPU(1EA) 사양의 서버에서 진행되었다.

Fig. 14, 15는 진행한 7개의 실험에 대해 학습과 테스트에 대한 손실함수곡선을 나타낸다. 기존 알고리즘의 경우 HSSC+VHSSC+RWHC 조합(d)의 경우가 가장 안정되게 수렴하는 것을 볼 수 있다. 3개의 딥러닝 모델에 대한 손실함수곡선의 경우 2-Stacked Bi-LSTM 모델(c)이 가장 안정되면서 빠른 수렴을 보여주고 있다.

자체적으로 수집된 실험데이터를 이용한 성능비교 결과는 Table 3과 같이 2-Stacked Bi-LSTM 모델에서 가장 좋은 정확도 결과를 보여주고 있다. 2-Stacked Bi-LSTM 모델의 탐지 정확도는 94.6%로 기존의 방법보다 Precision, Recall, F-1 Score 모든 수치가 향상되었다. 또한 본 실험을 통해 기존 제안 방법인 2-Layer GRU 모델 기반의 RWHC 방법과 Bi-LSTM 모델 방법의 성능이 비슷함을 알 수 있었다. 또한 추출된 특징의 크기면에서 Table 3의 특징수(No. of features/frame) 컬럼에서 알 수 있듯이 제안한 방법이 2개로 가장 작은 특징수를 나타내고 있다. 특징수는 프레임 당 추출되는 특징의 개수를 나타낸다. 본 표를 통해 제안한 방법이 기존의 다른 알고리즘보다 적은 수의 특징으로 더 높은 인식 성능을 보여주고 있음을 알 수 있다.

5. 결 론

본 논문에서는 최근 1인 가구 중심의 스마트홈 환경에서 많이 발생하는 낙상 검출에 대해 MS Kinect v2 스켈레톤 카메라와 Stacked Bi-LSTM 모델을 이용한 탐지 모델을 제안하였다. 특히 몸통 관절을 기반으로 하는 중심관절에 대한 가속도와 바닥과의 거리를 특징으로 제안하였고 이를 2차원 입력으로 구성하여 Stacked Bi-LSTM 모델을 학습시켰다.

본 논문에서 사용한 Stacked Bi-LSTM 모델은 2개의 Bi-LSTM 모델을 사용하였고 아래층에는 128개 유닛, 위층에는 64 유닛으로 구성하였다. Stacked Bi-LSTM의 출력은 Dropout 레이어와 2개의 Dense 레이어를 거쳐 최종 1 비트의 결과(낙상 or 정상)를 출력하도록 구성하였다. 모델의 입력으로는 위에서 설명한 움직임 가속도와 거리, 2개의 특징 벡터가 함께 사용되었다.

기존 스켈레톤 기반의 연구에서는 관절 포인트 전체, 머리와 어깨, 머리와 어깨의 가속도, 골격의 가로세로비 등을 LSTM과 GRU 모델을 이용하여 학습하고 분류하였다. 본 연구에서는 이러한 스켈레톤 기반의 기존 연구와 제안한 연구의 성능을 비교 측정하였다. 이를 통해 2-Stacked Bi-LSTM 모델을 이용한 제안 방법이 보다 우수한 성능을 보이는 것을 증명하였다.

References

[1] W. Chen, Z. Jiang, H. Guo, and X. Ni, "Fall detection based on key points of human-skeleton using openpose," *Symmetry*, Vol.12, No.5, pp.744, May 2020.

[2] H. Li, A. Shrestha, H. Heidari, J. Le Kernec, and F. Fioranelli, "Bi-LSTM network for multimodal continuous human activity recognition and fall detection," *IEEE Sensors Journal*, Vol.20, No.3, pp.1191-1201, 2020. doi: 10.1109/JSEN.2019.2946095

[3] J. W. Si, et al., "Fall detection using skeletal coordinate vector and LSTM model," *Journal of Korean Institute of Information Technology*, Vol.18, No.12, pp.19-29, Dec. 2020. http://dx.doi.org/10.14801/jkiit.2020.18.12.19

[4] S. Mekruksavanich and A. Jitpattanakul, "LSTM networks using smartphone data for sensor-based human activity recognition in smart homes," *Sensors*, Vol.21, Iss.5, pp.1636, 2021. https://doi.org/10.3390/s21051636

[5] Y. K. Kang, H. Y. Kang, and D. S. Weon, "Human skeleton keypoints based fall detection using GRU," *Journal of the Korea Academia-Industrial Cooperation Society*, Vol.22, No.2, pp.127-133, 2021.

[6] M. Waheed, H. Afzal, and K. Mehmood, "NT-FDS—A noise tolerant fall detection system using deep learning on weara ble devices," *Sensors*, Vol.21, Iss.6, Articles No.2006, 2021. https://doi.org/10.3390/s21062006

[7] Q. Xu, G. Huang, M. Yu, and Y. Guo, "Fall prediction based on key points of human bones," *Physica A: Statistical Mechanics and its Applications*, Vol.540, Feb. 2020.

[8] C. B. Lin, Z. Dong, W. K. Kuan, and Y. F. Huang, "A framework for fall detection based on openpose skeleton and LSTM/GRU models," *Applied Sciences*, Vol.11, No.1, pp.329, 2021. https://doi.org/10.3390/app11010329

[9] Z. Cao, T. Simon, S. E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition(CVPR)*, 2016.

[10] PoseNet [Internet], https://github.com/tensorflow/tfjs-models/tree/-master/posenet

[11] MS Kinect v2 [Internet], https://azure.microsoft.com/en-us/services/-kinect-dk/

[12] O. Yildirim, "A novel wavelet sequence based on deep bidirectional LSTM network model for ECG signal classification," *Computers in Biology and Medicine*, Vol.96, pp.189-202, 2018.

[13] A. Graves, N. Jaitly, and A.-R. Mohamed, "Hybrid speech recognition with deep bidirectional LSTM," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp.273-278, 2013.

[14] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, Vol.521, No.7553, pp.436-444, 2015.

[15] A. Onan and M. A. Toçoğlu, "A term weighted neural language model and stacked bidirectional LSTM based framework for sarcasm identification," *IEEE Access*, Vol.9, pp.7701-7722, 2021.

[16] D. Utebayeva, A. Almagambetov, M. Alduraibi, Y. Temirgaliyev, L. Ilipbayeva, and S. Marxuly, "Multi-label UAV sound classification using Stacked Bidirectional LSTM," *Fourth IEEE International Conference on Robotic Computing (IRC)*, 2020.

[17] K. Jun, D. Lee, K. Lee, S. Lee, and M. S. Kim, "Feature extraction using an rnn auto-encoder for skeleton-based abnormal gait recognition," *IEEE Access*, Vol.8, pp.19196-19207, 2020.

[18] K. Adhikari, H. Bouchachia, and H. Nait-Charif, "Activity recognition for indoor fall detection using convolutional neural network," In *Proceedings of the 2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, Nagoya, Japan, pp.81-84, May 2017.

[19] Z. Cao, T. Simon, S. Wei, and Y. Sheikh, "Realtime multi-person 2D Pose Estimation Using Part Affinity Fields," In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition(CVPR)*, Honolulu, HI, USA, pp.1302-1310, Jul. 2017.

[20] A. Fares, S. H. Zhong, and J. Jiang, "EEG-based image classification via a region-level stacked bi-directional deep learning framework," IEEE International Conference on Bioinformatics and Biomedicine Madrid, Spain, Dec. 2018.

[21] Robust Scaler [Internet], https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.RobustScaler.html

신 병 근

https://orcid.org/0000-0003-0381-6243
e-mail : sbg0712@iwaz.co.kr
2019년 ~ 현 재 부산대학교
정보컴퓨터공학부 학사과정
2020년 ~ 현 재 (주)아이와즈 연구원
관심분야 : 인공지능, 딥러닝, IoT

김 응 호

https://orcid.org/0000-0003-4730-2260
e-mail : ehkim@iwaz.co.kr
2011년 대전대학교 정보통신공학부(학사)
2010년 ~ 2011년 (주)하이엘리더스 투모로우 사원
2011년 ~ 현 재 (주)아이와즈 책임연구원
관심분야 : 인공지능, 딥러닝, IoT

이 상 우

https://orcid.org/0000-0002-2473-0775
e-mail : lswross012@aideep.ai
2017년 서울과학기술대학교 컴퓨터공학과 (학사)
2020년 한양대학교 산업공학과(박사수료)
2020년 ~ 현 재 (주)에이아이딥 선임연구원
관심분야 : 인공지능, 딥러닝, 스마트홈 IoT

양 재 영

https://orcid.org/0000-0001-5735-9936
e-mail : jyyang@aideep.ai
2016년 광운대학교 정보과학교육원 컴퓨터공학과(학사)
2018년 광운대학교 전자공학과(석사)
2018년 ~ 2019년 한국전자기술연구원 연구원
2021년 ~ 현 재 (주)에이아이딥 선임연구원
관심분야 : 인공지능, 딥러닝, 스마트홈 IoT

김 원 겸

https://orcid.org/0000-0003-3022-6230
e-mail : wgkim@aideep.ai
1992년 충남대학교 전산학과(학사)
1994년 충남대학교 전산학과(석사)
2001년 충남대학교 컴퓨터과학과(박사)
1995년 ~ 1997년 (주)SK하이닉스 대리
2002년 ~ 2006년 한국전자통신연구원(ETRI) 선임연구원
2007년 ~ 2008년 마크애니(주) 부장
2009년 ~ 2016년 (사)한국저작권단체연합회 저작권보호센터 팀장
2016년 ~ 현 재 (주)에이아이딥 연구소장
관심분야 : 인공지능, 딥러닝, 영상처리, 스마트홈 IoT