

AI 기반 실시간 온라인 랜덤 플레이 댄스 플랫폼 개발

김수빈¹ · 박지혜¹ · 이민영¹ · 문서연¹ · 이경미^{2*}

¹덕성여자대학교 컴퓨터공학과 학사과정

^{2*}덕성여자대학교 컴퓨터공학과 교수

AI-based Real-time Online Random-play Dance Platform

Soobin Kim¹ · Jihye Park¹ · Minyoung Lee¹ · Seoyeon Mun¹ · Kyoung-Mi Lee^{2*}

¹Bachelor's Course, Department of Computer Engineering, Duksung Women's University, Seoul 01369, Korea

^{2*}Professor, Department of Computer Engineering, Duksung Women's University, Seoul 01369, Korea

[요약]

최근 숏폼 동영상의 유행으로 자신의 댄스 동영상을 업로드하거나 다른 사용자의 댄스 동영상을 보며 즐길 수 있는 플랫폼의 인기가 높아지고 있지만, 여전히 직접 참여하는 데에는 한계가 있다. 본 논문에서는 이러한 제약을 해결하기 위해 누구나, 언제, 어디서든, 참여할 수 있는 온라인 랜덤 플레이 댄스 플랫폼을 제안한다. 사용자가 제안하는 댄스 플랫폼에 접속하고 모바일 카메라 앞에 서서 랜덤하게 선택된 음악에 맞춰 댄스를 추면, 플랫폼은 실시간으로 사용자의 관절을 검출하고 댄스 동작을 분석하여 해당 음악의 참조 댄스 동작과의 유사도를 계산한다. 실시간 플레이를 위해 프론트엔드는 추출한 관절을 서버로 전송하고 서버가 동작 벡터 생성 및 유사도 계산을 함으로써 프론트엔드의 부하를 분산시킨다. 또한 빠르게 변하는 데이터는 데이터 캐싱으로 빠른 처리가 가능하게 한다. 댄스 동작 유사도 계산을 위해 사전학습된 동작 유사도 계산 신경망을 미세조정 학습시켰고 몸통을 제외하고 팔과 다리만 사용하여 최대-최소 정규화를 적용한 경우에 동작 유사도 오차를 줄여 더 좋은 정확도를 획득했다.

[Abstract]

With the recent increase in short-form videos, platforms that allow a user to upload his/her dance videos or watch others' videos have become increasingly popular; however, direct participation is still limited. Hence, this study proposes an online random-play dance platform that allows participation from anywhere, anytime. A user accessing the proposed dance platform can stand in front of a mobile camera and dance to randomly selected music, whereby the platform detects the user's body joints in real time and analyzes the dance motions to calculate their similarity with the reference dance motions of the randomly selected music. In real-time play, the front end sends the extracted joints to the server, and the platform offloads the front end by allowing the server to generate motion vectors to calculate the similarity. In addition, data caching was performed to enable the fast processing of rapidly changing data. As a result of fine-tuning a predefined gesture-similarity neural network, applying min-max normalization, and considering movements of only arms and legs (excluding the torso), motion-similarity error was reduced, thereby achieving better accuracy in computing dance-motion similarity.

색인어 : 숏폼 동영상, 랜덤 플레이 댄스, 동작 유사도, 실시간, 모바일 앱

Keyword : Short Form Video, Random-play Dance, Motion Similarity, Real-time, Mobile Application

<http://dx.doi.org/10.9728/dcs.2024.25.3.685>



This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-Commercial License (<http://creativecommons.org/licenses/by-nc/3.0/>) which permits unrestricted non-commercial use, distribution, and reproduction in any medium, provided the original work is properly cited.

Received 20 January 2024; Revised 05 March 2024

Accepted 08 March 2024

*Corresponding Author; Kyoung-Mi Lee

Tel: +82-2-901-8348

E-mail: kmlee@duksung.ac.kr

1. 서론

최근 1분 내외의 짧은 동영상 숏폼(short form) 콘텐츠를 기반으로 하는 서비스의 인기가 높아지고 있다. 소비자 데이터 플랫폼 오픈서베이가 발표한 ‘소셜미디어·검색포털 리포트 2023’에 따르면 전국 15~59세 남녀 5,000명 중 81.9%가 숏폼 콘텐츠를 인지하고, 68.9%가 숏폼을 시청한 경험이 있는 것으로 조사되어, 작년 71% 인지와 56.5% 시청에 대비하여 각각 10.9%와 12.4%가 높아졌다[1].

많은 숏폼 콘텐츠 중에서도 한국 댄스와 노래 등이 전 세계적으로 많은 주목을 받으며[2], 댄스의 하이라이트 부분만 따라 추는 영상을 올리는 ‘댄스 챌린지’는 주요 콘텐츠 중 하나로 자리 잡았다. 특히 랜덤 플레이 댄스 챌린지는 사람들이 일정 장소를 마련하고 정해진 시간에 모여 무작위로 노래가 재생되면 해당 노래의 안무를 아는 사람들만 나와 1분 미만의 댄스를 추고 노래가 끝나면 다시 자리로 돌아가는 이벤트로, 이를 촬영하여 숏폼 채널에 업로드하여 노래와 댄스를 공유하며 즐길 수 있다. 그러나 숏폼 콘텐츠의 주요 채널인 유튜브의 숏츠(Shorts), 인스타그램의 릴스(Reels), 틱톡(Tik Tok) 등에서 다양한 댄스 챌린지 콘텐츠를 호스팅하고 있지만, 댄스 또는 랜덤 플레이 댄스만을 위한 특화된 서비스는 거의 제공되고 있지 않은 상황이다. 최근 들어 인공지능(AI) 기반 스타트업 사이드워크 엔터테인먼트는 글로벌 K팝 댄스 AI 플랫폼으로 스텝인(STEPIN)이라는 새로운 댄스 플랫폼을 출시했다[3]. 스텝인은 실시간 동작 추적으로 댄스 정확도를 분석해 점수화하여 다른 사람과 경쟁하고 자신의 댄스 비디오를 만들어 공유하는 댄스만을 위한 플랫폼이다. 하지만 랜덤 플레이 댄스에 필요한 다수의 사용자가 동시에 실시간으로 챌린지에 참여하는 기능은 아직까지 포함되지 않았다.

본 논문에서는 랜덤 플레이 댄스가 오프라인에서 정해진 시간에 이뤄지기 때문에 시간과 공간의 제약이 있고 참여하려 해도 이벤트를 알기 어려워 참여하기 까다롭다는 한계를 극복하기 위해, 누구나, 언제, 어디서든, 실시간으로 적극적인 참여가 가능한 온라인 랜덤 플레이 댄스 플랫폼을 제안한다. 사용자는 자신이 있는 위치에서 언제든지 랜덤 플레이 댄스 챌린지에 참여할 수 있다. 3명 이상의 사용자가 모였다면 랜덤 플레이 댄스 챌린지가 시작된다. 랜덤으로 음악이 나오면 사용자는 모바일 카메라 앞에서 음악에 맞춰 춤을 추고, 제안하는 플랫폼은 실시간으로 사용자의 댄스 동작을 분석하여 이미 저장해 둔 해당 음악의 참조(reference) 댄스 동작과 유사도를 계산하여 우승자를 결정한다.

제안하는 온라인 실시간 랜덤 플레이 댄스 플랫폼의 개발 공헌은 다음과 같다.

1. 순수하게 댄스 그 자체를 즐기는 익명의 사람들에게 일정 시간에 댄스 전문 학원이나 유희 장소를 가는 대신 시간·공간적으로 열린 온라인 댄스 플랫폼을 제공한다.
2. 기존 동영상 플랫폼이 주로 1:N의 구조로 미리 만든 동

영상을 시청하는 방식이라면, 제안하는 플랫폼은 다수의 참여자가 실시간으로 참여하여 소통하며 상호작용할 수 있는 N:M의 다대다 실시간 콘텐츠 플랫폼을 제공한다.

3. 제안하는 댄스 플랫폼은 인간의 다양한 행동에 대해 이미 학습된 동작 유사도 계산 신경망을 사용하여 댄스 동작 유사도를 계산하도록 미세조정(fine-tuning) 학습함으로써 적은 시간과 자원을 사용하는 방법을 제공한다.
4. 댄스 동작의 유사도를 비교할 때 인체를 모두 사용하기 보다는 댄스 동작에서 상대적으로 움직임의 차이가 작은 몸통을 제외하고 움직임의 차이가 더 큰 팔과 다리만을 사용하며, 최대-최소 정규화를 적용함으로써 유사도 오차를 줄이는 방법을 제공한다.

본 논문은 다음과 같이 구성되고 있다. 2장에서는 제안하는 플랫폼의 주요 기능에 관련된 연구를 살펴보고, 3장에서는 미세조정을 사용하여 댄스 동작 유사도를 계산하는 방법을 제안한다. 제안하는 온라인 실시간 랜덤 플레이 댄스 플랫폼의 설계 및 개발 결과를 4장에서 설명하고, 5장에서는 결론 및 향후 연구를 서술한다.

II. 동작 관련 연구

2-1 인체 검출

인간의 동작을 분석하기 위해서는 입력 이미지나 동영상으로부터 인체의 관절 부위(body joints)를 찾고 인체 골격(skeleton)을 구축하여야 한다. 최근 딥러닝의 도입으로 2D 이미지의 시계열 데이터(시퀀스)인 동영상을 처리하여 3D 자세를 검출하는 연구로 확장할 수 있게 되었다[4]. 하지만 한 차원이 손실된 2D 이미지로부터 3D 자세를 검출하는 것은 겹침이나 깊이 모호성 및 불충분한 훈련 데이터로 인해 서로 다른 3D 동작이 유사한 2D 포즈로 잘못 투영되는 문제가 발생할 수 있어 여전히 어려운 과제로 남아있다[5].

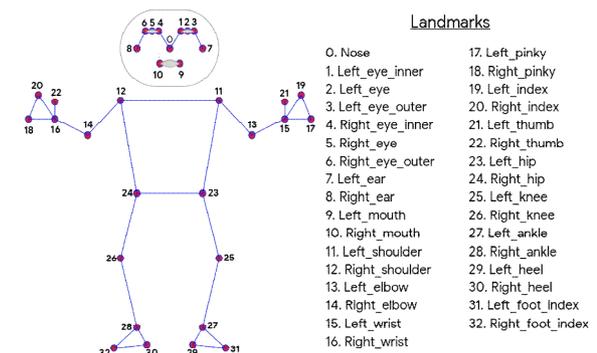


그림 1. 구글의 ML Kit Pose Detection API에서 추출하고 시각화한 33개의 랜드마크[6]

Fig. 1. 33 landmarks extracted and visualized by Google's ML Kit Pose Detection API[6]

본 논문에서는 구글에서 제공하는 ML Kit Pose Detection (MKPD) API를 사용하여 각 동영상 프레임에서 3D 인체 관절 좌표를 얻는다[6]. MKPD를 지원하는 미디어파이프 포즈 (mediapipe pose)는 BlazePose를 사용하는데, BlazePose는 첫 번째 프레임 또는 사람이 탐지될 때까지 실행되고, 이후 후속 프레임에서는 앞서 탐지된 사람을 추적하는 방식을 사용하기 때문에 실시간으로 동영상에서 인체를 검출하는데 유리하다[7]. 그림 1은 BlazePose에서 사용하는 33개의 3D 랜드마크로 구성된 인체 관절 모델을 보여주고 있다[6].

2-2 동작 유사도 계산

여러 개의 관절이 계층적으로 연결된 인체의 동작 유사도를 계산하는 것은 컴퓨터비전 연구자들로부터 많은 관심을 받으며 연구되어 온 분야이다. 인체 동작 유사도를 계산하기 위해 Y. Kim과 연구자는 프레임 내 인물의 관절 각도를 사용하여 두 개의 동영상 속 동작 사이의 유사도를 측정했다[8]. 그러나 이런 개별 프레임 간의 동작 유사도 계산은 동적인 동작의 유사도를 올바르게 계산하기 어렵다. 대신 A. Aristidou와 연구자들은 짧은 동작 시퀀스 단위로 동작을 표현하고 유사도를 계산했다[9]. 개별 프레임 대신 동작 시퀀스를 사용해서 동적 동작을 표현하는 것이 가능해졌지만, 여전히 실시간 응용 프로그램에는 적합하지 않은 문제점을 가지고 있다.

실시간 동적 동작 유사도 계산을 위해 J. Park과 연구자들은 동작 임베딩을 생성하여 동작 유사도를 계산하는 방법을 제안했다[10]. 그림 2는 논문 [10]에서 제안한 동작 유사도를 계산하는 방법의 흐름도를 보여주고 있다. 우선 비교할 두 개의 동영상 클립에서 [11]의 인체 모델을 확장하여 17개의 랜드마크로 구성된 인체 모델의 2D 관절을 각각 검출하고, 이들을 5개의 관절 그룹으로 분해했다. 각 관절 그룹은 비지도 학습 방법의 하나로 데이터의 특징을 추출하는 모델인 오토인코더(auto-encoder)를 사용하여 임베딩 벡터를 생성했다. 오토인코더의 구조는 크게 인코더와 디코더로 나뉘는데, 입력 데이터를 인코더에서 압축하여 동작 임베딩 벡터를 추출하고 그 벡터를 디코더에서 다시 데이터로 재구성했다. 두

개의 동영상 클립의 동작을 임베딩 벡터 공간에서 강제로 분리하기 위해 참조 동작, 동일한(positive) 동작, 다른(negative) 동작을 입력으로 하는 삼중 손실(triplet loss)를 사용했다. J. Park과 연구자들은 이 삼중 손실에 두 동작 데이터들이 얼마나 유사한지에 대한 정보를 포함시키기 위해 두 동작의 차이를 계산하고 이를 손실 계산에 포함시킨 사중 손실(quadruplet loss)을 제안했다.

동작 동영상을 관절값 시퀀스로 나누어 동작 임베딩 모델을 통해 임베딩 벡터를 생성한 후에, 두 동작의 임베딩 벡터에 대한 코사인 유사도를 계산했다. 코사인 유사도는 두 벡터 간의 코사인 각도를 이용하여 두 벡터의 유사도로 계산하는 방법으로 두 벡터의 방향이 완전히 같은 경우에는 1의 값을 가지며, 90°의 각을 이루면 0, 180°로 반대의 방향을 가지면 -1의 값을 갖게 된다. 즉, 코사인 유사도는 -1과 1 사이의 값을 가지며 값이 1에 가까울수록 유사도가 높다고 판단할 수 있다. 각 관절 그룹에 대한 코사인 유사도를 계산한 후, 그룹의 평균으로 전체 유사도를 계산했다.

$$sim_{body} = \frac{\sum_{i=0}^4 cos_sim_i}{5} \tag{1}$$

여기서 cos_sim_i 는 관절 그룹의 코사인 유사도를 의미한다.

III. 미세조정을 이용한 댄스 동작 유사도 계산

본 장에서는 온라인 실시간 랜덤 플레이 댄스 플랫폼에 포함될 댄스 동작의 유사도를 측정하기 위해 구글에서 제공하는 ML Kit Pose Detection(MKPD) API와 논문 [10]에서 제안한 동작 임베딩 벡터를 생성하여 유사도를 계산하는 딥러닝 모델을 사용하여[7] 미세조정하는 방법을 제안한다.

3-1 인체 관절 검출

댄스 동작의 유사도를 계산하기 위해 우선 동영상의 프레임으로부터 인체 관절을 검출해야 한다. 그림 3은 댄스 동작

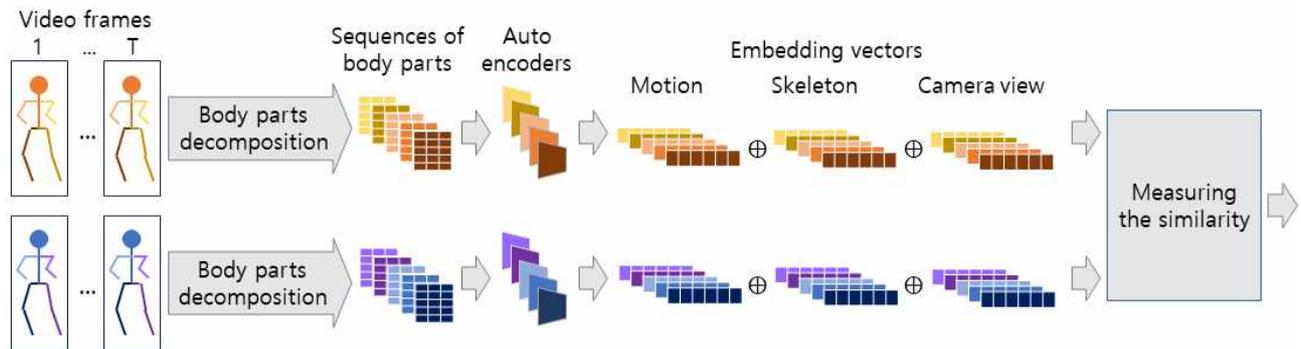
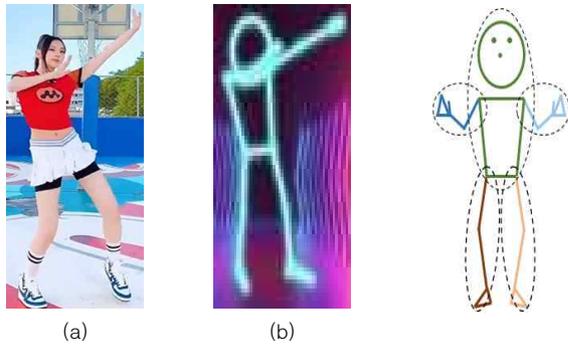


그림 2. 동작 유사도 계산을 위한 흐름도[10]
Fig. 2. Flowchart for measuring motion similarities[10]



Body parts in Fig. 2	Composition joints in Fig. 1
Right Arm	Right_shoulder, Right_elbow, Right_wrist, *Right_pinky, *Right_index, *Right_thumb
Left Arm	Left_shoulder, Left_elbow, Left_wrist, *Left_pinky, *Left_index, *Left_thumb
Rigt Leg	Right_hip, Right_knee, Right_ankle, *Right_heel, *Right_foot_index
Left Leg	Left_hip, Left_knee, Left_ankle, *Left_heel, *Left_foot_index
Torso	Nose, *Right_eye, *Left_eye, Right_shoulder, Left_shoulder, Right_hip, Left_hip

그림 3. 인체 관절 검출 및 분해 결과 : (a) 입력 댄스 이미지, (b) 검출된 인체 관절, (c) 5개 그룹으로 분해(*으로 표시된 관절은 [10]에서 사용된 모델에 추가된 관절을, +으로 표시된 관절은 대체된 관절을 나타냄)

Fig. 3. Results of human joints detection and decomposition : (a) input image, (b) detected human joints, (c) decomposition into 5 groups(Joints marked with * represent joints added to the model used in [7], and joints marked with + represent replaced joints)

영상과 검출된 결과를 보여주고 있다. 본 논문에서는 MKPD API를 사용하여 입력 영상(그림 3(a))에서 그림 1과 같은 33개의 인체 관절 랜드마크를 검출한다. 이들 중 댄스 동작 인식에 영향을 덜 주는 얼굴의 일부 랜드마크를 제외한 25개의 랜드마크를 사용하여 그림 3(b)와 같이 시각화한다.

동영상의 연속된 인체 관절 시퀀스에서 검출된 25개의 랜드마크는 5개의 그룹으로 분해된다: {오른팔, 왼팔, 오른다리, 왼다리, 몸통}. 그림 3(c)는 그림 1의 랜드마크 모델을 그림 2의 5개 그룹으로 분해한 모습을 보여주고 있다. BlazePose는 논문 [10]에서 사용된 모델 보다 손과 발에 더 많은 랜드마크를 검출하므로 이를 관절 그룹에 추가한다. 또한 BlazePose는 목에 대한 랜드마크를 제공하지 않으므로 눈의 2개의 랜드마크로 대체하여 사용한다. 이렇게 분해한 5개의 관절 그룹은 각각의 오토인코더(auto-encoder)가 포함된 신경망을 사용하여 관절 그룹의 차원을 낮추며 임베딩 벡터로 계산된다[12].

3-2 댄스 동작 임베딩 학습의 미세조정

제안하는 플랫폼은 댄스 동작을 대상으로 하므로 댄스 동작의 유사도를 계산하기 위한 학습 데이터와 학습 과정이 필요하다. 그러나 데이터를 확보하고 처음부터 학습하는 것은 시간과 비용이 많이 소요된다[13]. 일반적으로 새로운 신경망 모델로 사전 지식 없이 학습하기 보다는 사전학습(pre-trained)된 신경망을 바탕으로 비교적 소규모의 데이터 세트를 추가학습하여 미세조정(fine-tuning)하는 것이 더 좋은 성능을 이뤄냈다고 알려졌다[10]. 본 논문에서는 논문 [8]에서 동작 임베딩 벡터를 생성하기 위해 18명의 서로 다른 배우가 4가지 범주(전투, 모험, 스포츠, 댄스)의 동작으로 구성된 SARA(Synthetic Actors and Real Actions)라는 3D 동작 데이터세트로 학습시키고, 동작 유사도를 계산하기 위해 106명의 120가지 다양한 행동으로 구성된 NTU(Nanyang Technological University) RGB + D(Depth) 120 데이터 세트로 학습시킨 딥러닝 모델을

사전학습 모델로 사용한다. 사전학습 모델은 다양한 동작에 대한 임베딩 벡터 생성 및 유사도 계산에 비교적 우수한 성능을 보여주었지만, 비교적 팔과 다리에 동작이 많은 댄스 동작에 대한 유사도 계산을 위해서는 추가 학습이 필요하다.

댄스 동작 유사도를 계산하기 위해 기존 학습 과정의 1단 계인 오토인코더까지의 학습은 사전학습 모델을 통해 가져오고, 2단계인 비교 학습에서 댄스 데이터 셋을 통해 미세조정 학습을 진행하게 된다. 미세조정을 위한 실험 시나리오는 다음과 같다.

1. 몸통 제외 여부 : 몸통은 대부분의 동작에 대해 유사도가 지나치게 높기 때문에 댄스 동작 유사도를 계산하기 위해 5개의 관절 그룹 중 몸통을 포함 또는 제외하여 학습한다.
2. 최대-최소 정규화 유무 : 최종 유사도를 도출하기 직전의 유사도 창(window) 값 배열에서 최대값과 최소값의 차이의 역수를 곱한다. 동일한(positive) 동작의 유사도는 더 높게, 다른(negative) 동작의 유사도는 더 낮게 하여 유사도의 차이를 극대화할 수 있다.

$$\text{sim}_{\text{body}}^w = \frac{\text{sim}_{\text{body}}^w}{\max_{w=0}^{W-1}(\text{sim}_{\text{body}}^w) - \min_{w=0}^{W-1}(\text{sim}_{\text{body}}^w)} \tag{2}$$

여기서 W는 창(window)의 길이를 의미하며, 본 논문에서는 16 프레임을 사용한다.

3. 거리 함수 사용 : 두 동작의 차이를 계산하는 거리 함수로 평균절대오차(Mean Absolute Error, MAE)(식 (3))와 평균 제곱근 오차(Root Mean Square Error, RMSE)(식 (4))를 사용한 손실 함수로 학습한다.

$$\text{var}(m, m') = \frac{\|v_m - v_{m'}\|_1}{2 \times n_{v_m}} \tag{3}$$



그림 4. 댄스 동작 유사성 측정을 위해 학습에 사용된 이미지 쌍(초상권 침해를 막기 위해 논문 그림 속 얼굴을 모자이크 처리함)
 Fig. 4. Examples of image pairs used for training to measure dance gesture similarity(To prevent infringement on the portrait rights, we mosaiced the faces in this figure of this paper.)

$$var(m, m') = \sqrt{\frac{\sum_{i=1}^{2 \times n_{v_m}} (v_m - v_{m'})^2}{2 \times n_{v_m}}} \quad (4)$$

여기서 m 과 m' 은 댄스 동작이며, v_m 과 $v_{m'}$ 은 각각 m 과 m' 의 임베딩 벡터이다. n_{v_m} 은 벡터의 길이를 의미한다.

3-3 실험 결과

제안하는 댄스 동작 유사도를 계산하기 위해 사용한 데이터는 유튜브 등에서 수집한 102개의 댄스동영상(70,199개의 프레임)들로 이루어졌다. 이 중 71개를 학습 데이터로, 31개를 테스트 데이터로 사용해 미세조정 학습을 했다. 그림 4는 실험에 사용된 유사한 동작의 이미지 쌍의 예들이다. 대부분의 댄스 동영상은 카메라가 사용자의 정면을 비추어 사용자의 전신이 포함되도록 촬영된 것으로, 음악에 맞춰 여러 다양한 댄스 동작을 포함하고 있다. 댄스 음악마다 참조 댄스 동작이 있고 이를 보고 댄스를 따라 추는 것이므로, 같은 음악에 대한 댄스 동작을 동일한(positive) 동작으로 간주한다.

본 논문에서 댄스 동작 임베딩 벡터를 계산하여 유사도를 계산하기 위한 학습 환경은 윈도우 10의 운영체제, Intel i5의 CPU, NVIDIA GeForce RTX 2060의 GPU를 가진 컴퓨터에서 이루어졌다. 딥러닝 모델을 학습하는데 필요한 하이퍼 파라미터(hyper-parameter)는 학습률이 0.0001, 임베딩 모델에 입력하기 전에 입력 데이터를 조정하기 위한 삼중 손실 함수의 가중치(weight)는 0.3, 마진(margin)은 0.7, 삼중 손실 함수를 조정하기 위한 변형 제어 매개변수(variation control parameter)는 0.2를 사용한다. 또한 사중 손실의 가중치는 1.0, 마진은 0.5를 사용한다.

본 논문에서는 동작 유사도의 정확성을 측정하기 위해서 식 (5)와 같이 참조 동영상(=1)에 대한 유사도 오차의 합을 사용한다.

$$\frac{\sum_{w=0}^{W-1} (1 - \text{sim}_{\text{body}}^w)}{W} \quad (5)$$

식 (5)의 값이 작을수록 유사도 계산이 정확하게 수행된 것을 의미한다.

표 1은 제안하는 댄스 동작 유사도를 계산하기 위한 미세조정 실험 결과를 보여주고 있으며, 표의 수치는 식 (5)의 값을 의미한다.

1. 비교실험을 위해 테스트한 논문 [8]은 몸통을 포함하며 최대-최소 정규화를 포함하지 않은 모델로 제안하는 미세조정 학습을 수행한 경우가 더 정확하게 유사도를 계산한다.
2. 최대-최소 정규화를 적용한 경우가 그렇지 않은 경우보다 더 정확하게 유사도를 계산한다.
3. 5개의 관절 그룹 중 몸통을 제외한 경우가 포함한 경우보다 더 정확하게 유사도를 계산한다.
4. 댄스 유사도 계산을 위한 학습에서 몸통을 제외하고, 최대-최소 정규화를 사용한 경우가 가장 정확하게 계산한다.
5. 거리 함수로 평균절대 오차(식 (3))을 사용한 경우가 평균 제곱근 오차(식 (4))를 사용한 경우보다 조금 더 좋은 결과를 얻었다. 둘 중 어느 거리 함수를 사용하여도 사전학습 모델을 사용한 경우보다 미세조정 학습을 수행한 경우에 더 정확하게 유사도를 계산한다.

표 1. 미세조정 학습에 대한 비교 실험 결과
 Table 1. Comparative experiments' results on fine-tuning

Training	Torso exclusion	Min-max normalization	MAE (Eq. (3))	RMSE (Eq. (4))
[8]	X	X	0.59	0.61
proposed	X	X	0.46	0.49
	X	○	0.42	0.45
	○	○	0.38	0.40

IV. 제안하는 실시간 온라인 랜덤 플레이 댄스 플랫폼

본 장에서는 3장에서 제안한 댄스 동작 유사도 계산을 적용하기 위해 실시간 온라인 랜덤 플레이 댄스 플랫폼을 제안한다. 제안하는 플랫폼은 크게 플레이어가 온라인 실시간으로 랜덤 플레이 댄스를 진행하는 스테이지(Stage)와 사용자들이

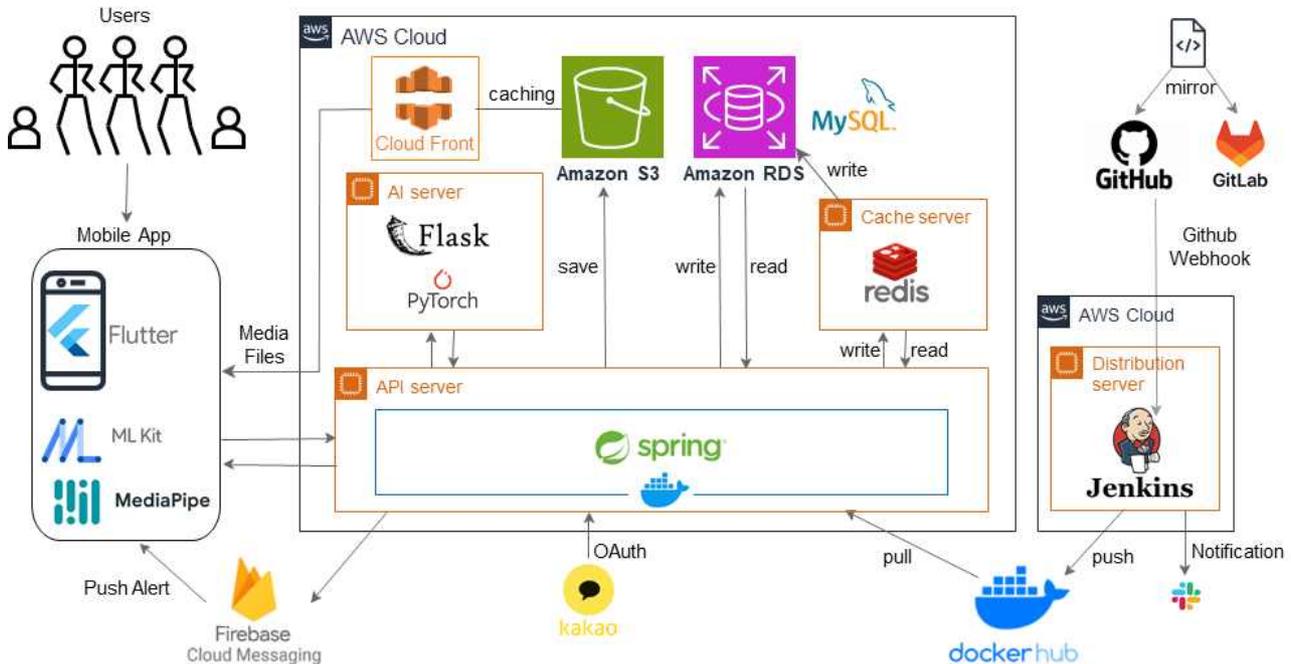


그림 5. 제안하는 온라인 랜덤 플레이 댄스 플랫폼의 시스템 구성도
 Fig. 5. System diagram of the proposed online random play dance platform

자신의 댄스를 촬영하여 업로드하고 다른 사용자들과 공유할 수 있는 커뮤니티(Community)로 나뉜다.

4-1 제안하는 온라인 랜덤 플레이 댄스 플랫폼 설계

그림 5는 제안하는 실시간 온라인 랜덤 플레이 댄스 플랫폼의 시스템 구성도를 보여준다. 제안하는 플랫폼의 프론트엔드는 플러터(Flutter)로 구현했으며, 소켓이 전해주는 스테이지 상태 및 데이터에 실시간으로 반응하기 위해 프로바이더(Provider) 패턴을 사용한다. 프론트엔드는 구글 MKPD API를 이용하여 플레이어의 관절을 추출하여 서버에 전달한다.

그림 5의 플랫폼 서버는 4개의 AWS EC2 서버를 사용하였다: 자동배포 서버, API 운영 서버, AI 서버, 캐시(cache) 서버. 자동배포 서버에서는 먼저 젠킨스(Jenkins)와 도커를 이용하여 CI(Continuous Integration)/CD(Continuous Deployment)를 구축한다. 깃허브 웹후크(github webhook)로 젠킨스가 메인(main) 브랜치의 변경을 감지하여 자동으로 스프링 부트(Spring boot) 이미지를 빌드하고 도커허브(dockerhub)에 푸시한다. 이 이미지를 API 운영 서버에서 풀(pull) 받아 컨테이너 형식으로 스프링 부트 기반의 API 운영 서버를 실행하게 된다.

그림 5의 중앙 하단에 위치한 API 운영 서버는 랜덤 플레이 댄스 스테이지에 관한 처리를 담당하며, 플라스크(Flask)를 통해 Rest API로 만들어진 AI 서버는 3장에서 제안한 댄스 동작 임베딩 벡터 생성 및 유사도 계산을 위한 학습 모델을 가지고 있다. 특히, 그림 6과 같이 프론트엔드에서 추출한 관절을 API 서버와 AI 서버로 전송하여 프론트엔드의 부하를 나눔으로써 빠른 통신이 가능하게 한다. API 서버는 프론트

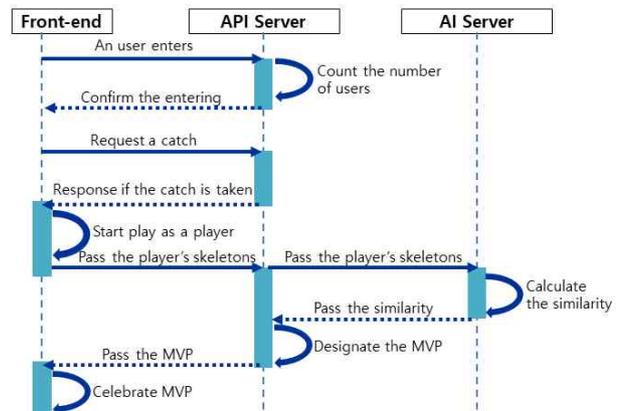


그림 6. 스테이지 진행 시의 기능 처리도
 Fig. 6. Function processing diagram during stage progression

엔드인 스테이지에 새로운 사용자가 입장할 때마다 현재 사용자 수를 확인하고 3명 이상이 되면 캐치 요청을 시작한다. 사용자의 캐치 요청을 받아, 캐치에 성공한 3명의 사용자에게만 캐치 성공을 응답한다. 캐치에 성공한 사용자인 플레이어가 랜덤 음악에 맞춰 댄스를 추면, 프론트엔드는 모바일 카메라를 통해 들어온 플레이어의 관절값을 검출하여 시각화하고 API 서버로 전송한다. API 서버는 플레이어의 관절값을 스테이지에 입장한 모든 사용자들에게 송출하여 댄스 플레이에 참여하지 않은 사용자도 다른 플레이어의 춤추는 모습을 관찰할 수 있도록 한다. 댄스 동작 유사도 계산을 위해 API 서버는 플레이어의 관절값과 해당 음악의 참조 관절값을 AI 서버로 전송하고, AI 서버는 댄스 동작 유사도를 계산한 후 API

서버로 그 값을 다시 전송한다. API 서버는 이를 이용해 우승자 MVP를 선정하고 그 결과를 프론트엔드에 전달한다. 마지막으로 선정된 우승자를 위한 축하 시간을 갖는다. 랜덤 음악 1곡에 대한 댄스 챌린지가 모두 끝나면 다시 대기 상태로 돌아가 챌린지를 반복한다.

랜덤 플레이 댄스 스테이지에서 사용되는 데이터는 음악에 맞춰 들어오는 3명의 플레이어의 관절 정보가 실시간으로 매우 빠르게 바뀌므로 데이터베이스에서 관리하는 것은 서버의 부하가 상당히 늘어날 수 있다. 이런 서버의 부하를 줄이기 위해 메모리 위에서 동작하며 NoSQL이므로 처리 속도가 빨라 데이터 캐싱 용도로 적합한 레디스(Redis)를 이용하여 캐시 서버를 구축한다. 그림 7은 제안하는 플랫폼에서 스테이지를 위해 사용된 NoSQL 데이터베이스를 보여주고 있다. 각 스테이지는 상태 정보(status), 현재 플레이되는 음악(music_info) 및 그 음악의 참조 댄스 동작(ref_skeleton), 스테이지에 입장한 사용자들(enterUser), 랜덤 댄스 캐치를 요청한 사용자들(catchUser), 캐치를 잡아 랜덤 플레이 댄스에 참여하는 사용자(player)에 대한 데이터를 키-값의 형태로 저장하고 json으로 전송한다.

반면 영구적으로 저장해야 하는 데이터를 보관하기 위해서는 관계형 데이터베이스(Relational Database)인 MySQL을 통한 AWS RDS(Relational Database Service)를 메인 데이터베이스로 사용하여 어떤 환경에서라도 동일한 데이터에 접근할 수 있도록 한다. 데이터베이스에서는 커뮤니티와 스테이지에서 사용되는 사용자 정보, 채팅룸, 사용자들이 업로드한 댄스 영상에 대한 좋아요, 주석 등의 데이터를 관리한다. 특히, 랜덤 플레이 댄스 스테이지에서 사용되는 영상과 음악

등의 미디어 파일은 AWS S3 스토리지에 저장하고, 빠른 서비스를 위해 AWS Cloudfront를 거쳐 사용자에게 제공된다.

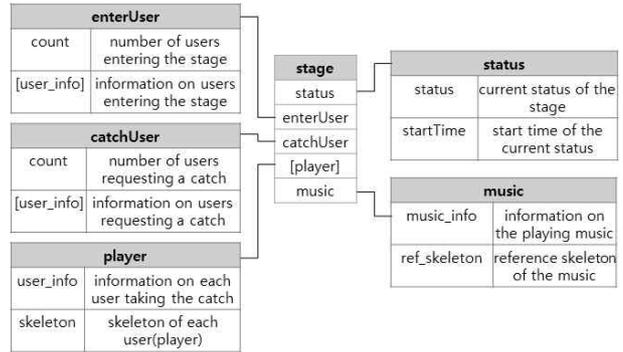


그림 7. 제안하는 플랫폼의 NoSQL 데이터베이스
Fig. 7. NoSQL database of the proposed platform

4-2 제안하는 댄스 플랫폼 개발 결과

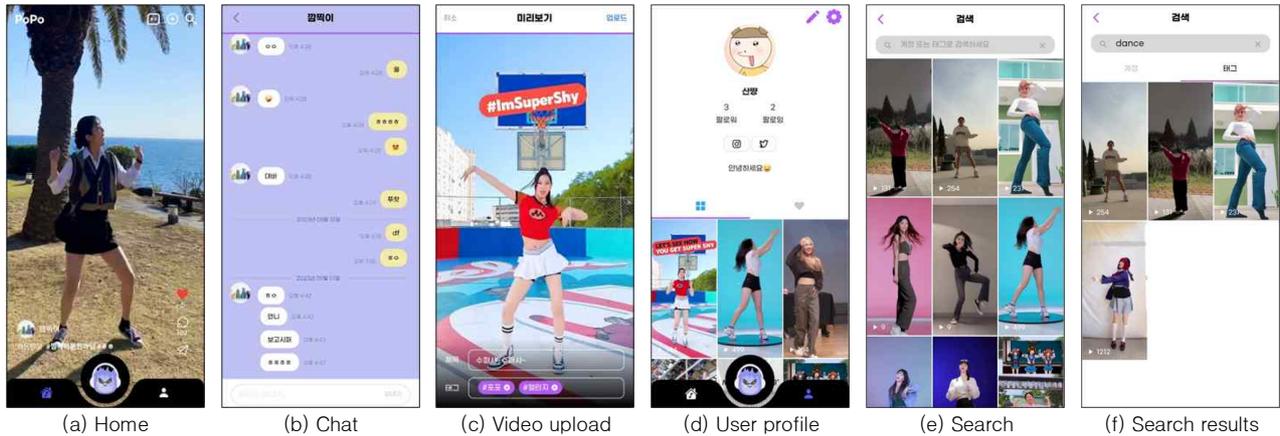
그림 8과 9는 본 논문에서 제안하는 온라인 랜덤 플레이 댄스 플랫폼의 개발 결과를 보여주고 있다. 그림 8(a)은 사용자가 스테이지에 입장하여 캐치와 댄스 플레이가 시작하기를 기다리는 화면, 그림 8(b)는 3명 이상의 사용자가 입장하여 캐치를 요청하는 화면이다. 그림 8(c)는 캐치 요청에 성공한 3명의 플레이어가 결정되면 랜덤 음악이 나오고 각 플레이어들은 카메라 앞에서 음악에 맞춰 춤을 추고 그 모습을 시각화하여 보여주는 화면이다. 플레이하는 동안 참조 댄스 동작과의 동작 유사도가 계산되어 중간 성적의 형식으로 'perfect',



* These figures are screen captures of the developed system, and this system is made for Korean users.

그림 8. 제안하는 댄스 플랫폼 화면 : 스테이지 화면

Fig. 8. Screen captures of the proposed random dance platform : Stage



* These figures are screen captures of the developed system, and this system is made for Korean users.

그림 9. 제안하는 랜덤 플레이 댄스 플랫폼 화면 : 커뮤니티 화면

Fig. 9. Screen captures of the proposed random dance platform : Community

‘great’, ‘bad’ 등이 화면에 표시되어 플레이어뿐만 아니라 입장하여 관람하고 있는 사용자들의 흥미를 높인다. 그림 8(d)는 음악이 끝나고 최종 성적으로 우승자인 MVP가 결정되고 축하해주는 화면이다.

그림 9는 사용자가 짧은 댄스 동영상을 공유하는 커뮤니티 개발 결과를 보여주고 있다. 그림 9(a)는 제안하는 플랫폼의 홈 화면으로 로그인한 사용자가 업로드한 댄스 동영상을 볼 수 있다. 그림 9(b)는 다른 사용자와 실시간으로 메시지를 주고받으며 소통할 수 있는 채팅룸 화면이고, 그림 9(c)는 동영상 업로드 화면으로 카메라로 바로 촬영하거나 갤러리에 있던 동영상을 커뮤니티에 업로드할 수 있으며, 태그도 함께 입력할 수 있다. 그림 9(d)는 사용자 프로필 화면으로 사용자가 업로드한 동영상과 ‘좋아요’를 누른 동영상 목록을 확인할 수 있다. 그림 9(e)와 그림 9(f)는 검색 화면으로 동영상 태그와 사용자 닉네임으로 검색하여 댄스 동영상을 검색한 결과를 볼 수 있다.

V. 결 론

본 논문에서는 랜덤 플레이 댄스라는 오프라인 이벤트를 언제 어디서나 참여할 수 있는 모바일 앱 플랫폼을 제안한다. 실시간 플레이를 위해 앱으로 들어온 사용자 동영상에서 바로 인체 관절을 추출하고 웹소켓으로 서버와 관절 데이터를 주고받으며 관절 시각화와 댄스 동작 유사도 계산을 한다. 댄스 동작 유사도 계산을 위한 학습은 기존 동작 유사도 학습 신경망[8]을 미세조정 학습시켰다. 인체를 모두 사용하기보다는 댄스 동작 사이에 상대적으로 움직임 차이가 작은 몸통을 제외하고 댄스 동작의 움직임에 더 영향을 미치는 팔과 다리만을 사용하며, 최대-최소 정규화를 적용한 경우에 유사도 오차를 30% 이상(거리 함수 MAE는 0.59에서 0.38로 35.59%, RMSE는 0.61에서 0.4로 34.43%) 줄여서 더 정확

한 댄스 동작 유사도를 획득했다.

향후 제안하는 댄스 플랫폼은 스테이지에서 네트워킹 최적화를 통해 과부하를 방지하고, 커뮤니티에서 많이 노출되는 동영상 역시 레디스로 캐싱하여 동영상 로딩속도를 향상시킬 계획이다. 또한 동적 동작의 유사도 계산의 정확도를 향상시키기 위해 학습 모델을 확장할 계획이다. 커뮤니티에서 사용자가 업로드한 동영상들에 대해 사용자의 선호를 반영한 추천 알고리즘을 적용할 계획이다.

참고문헌

- [1] Opensurvey. Social Media/Search Portal Trend Report 2023 [Internet]. Available: <https://blog.opensurvey.co.kr/trendreport/socialmedia-2023/>.
- [2] Y. Kim, “Development of Dance Learning System Using Human Depth Information,” *Journal of Digital Contents Society*, Vol. 18, No. 8, pp. 1627-1633, December 2017. <http://dx.doi.org/10.9728/dcs.2017.18.8.1627>
- [3] Sidewalk Entertainment. STEPIN [Internet]. Available: <https://stepin.ai/>.
- [4] S.-Y. Kim, S.-J. Urm, S.-Y. Yoo, S.-J. Kim, and K.-M. Lee, “Application of Sign Language Gesture Recognition Using Mediapipe and LSTM,” *Journal of Digital Contents Society*, Vol. 24, No. 1, pp. 111-119, January 2023. <http://dx.doi.org/10.9728/dcs.2023.24.1.111>
- [5] C. Zheng, W. Wu, C. Chen, T. Yang, S. Zhu, J. Shen, ... and M. Shah, “Deep Learning-Based Human Pose Estimation: A Survey,” *ACM Computing Surveys*, Vol. 56, No. 1, 11, January 2024. <https://doi.org/10.1145/3603618>
- [6] Google. ML Kit Pose Detection API [Internet]. Available: <https://developers.google.com/ml-kit/vision/pose-detection>.

[7] S. Mroz, N. Baddour, C. McGuirk, P. Juneau, A. Tu, K. Cheung, and E. Lemaire, "Comparing the Quality of Human Pose Estimation with BlazePose or OpenPose," in *Proceedings of the 4th International Conference on Bio-Engineering for Smart Technologies (BioSMART)*, Paris/Créteil, France, pp. 1-4, December 2021. <https://doi.org/10.1109/BioSMART54244.2021.9677850>

[8] Y. Kim and D. Kim, "Real-Time Dance Evaluation by Markerless Human Pose Estimation," *Multimedia Tools and Applications*, Vol. 77, No. 23, pp. 31199-31220, December 2018. <https://doi.org/10.1007/s11042-018-6068-4>

[9] A. Aristidou, D. Cohen-Or, J. K. Hodgins, Y. Chrysanthou, and A. Shamir, "Deep Motifs And Motion Signatures," *ACM Transactions on Graphics*, Vol. 37, No. 6, 187, December 2018. <https://doi.org/10.1145/3272127.3275038>

[10] J. Park, S. Cho, D. Kim, O. Bailo, H. Park, S. Hong, and J. Park, "A Body Part Embedding Model with Datasets for Measuring 2D Human Motion Similarity," *IEEE Access*, Vol. 9, pp. 36547-36558, March 2021. <https://doi.org/10.1109/ACCESS.2021.3063302>

[11] K. Aberman, R. Wu, D. Lischinski, B. Chen, and D. Cohen-Or, "Learning Character-Agnostic Motion for Motion Retargeting in 2D," *ACM Transactions on Graphics*, Vol. 38, No. 4, 75, August 2019. <https://doi.org/10.1145/3306346.3322999>

[12] K.-C. Sim and K. Kim, "Insider Anomaly Behavior Detection Method Using an Unsupervised Learning-Based Autoencoder," *Journal of Digital Contents Society*, Vol. 24, No. 8, pp. 1929-1936, August 2023. <http://dx.doi.org/10.9728/dcs.2023.24.8.1929>

[13] Y. Yu, K. Tang, and Y. Liu, "A Fine-Tuning Based Approach for Daily Activity Recognition between Smart Homes," *Applied Sciences*, Vol. 13, No. 9, 5706, May 2023. <https://doi.org/10.3390/app13095706>



김수빈(Soobin Kim)

2018년~현 재: 덕성여자대학교 컴퓨터공학과 학사과정
 ※ 관심분야: 서버, 데이터베이스, 데이터 사이언스



박지혜(Jihye Park)

2019년~현 재: 덕성여자대학교 컴퓨터공학과 학사과정
 ※ 관심분야: 서버, API, 클라우드



이민영(Minyoung Lee)

2019년~현 재: 덕성여자대학교 컴퓨터공학과 학사과정
 ※ 관심분야: 모바일 앱, UX/UI, Interaction Design



문서연(Seoyeon Mun)

2020년~현 재: 덕성여자대학교 컴퓨터공학과 학사과정
 ※ 관심분야: 모바일 앱, UX/UI, Interaction Design



이경미(Kyoung-Mi Lee)

1993년: 덕성여자대학교 전산학과 (이학사)

1996년: 연세대학교 대학원 (이학석사)

2001년: Univ. of Iowa, Dept. of Computer Science

(전산학박사-지능형멀티미디어)

2003년~현 재: 덕성여자대학교 컴퓨터공학과 교수

※ 관심분야: 영상처리, 컴퓨터비전, 패턴인식