JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 27, No. 5, pp. 1383-1389, May. 2026

Check for updates

Unity 플랫폼 기반의 하이브리드 CNN-RNN 모델

성 재 용[1] · 김 형 진[2*]
[1]전북대학교 IT응용시스템공학과 박사과정
[2]전북대학교 IT응용시스템공학과 교수

Hybrid CNN-RNN Model Based on the Unity Platform

Jae-Yong Sung[1] · Hyung-Jin Kim[2*]

[1]PhD Program, Department of IT Applied System Engineering, Jeonbuk National University, Jeonbuk 54896, Korea
[2]Professor, Department of Information and Communication Engineering, Jeonbuk National University, Jeonbuk 54896, Korea

[요 약]

본 논문에서는 Unity 플랫폼 기반에서 시간적 패턴을 학습할 수 있는 인공지능 동작 인식 시스템을 제안한다. 이를 위하여 CNN과 RNN 모델의 장단점을 비교 분석하였으며, 두 모델의 강점을 결합한 경량 하이브리드 모델(Lightweight Hybrid Model)을 설계하였다. CNN은 이미지 데이터로부터 공간적 특징(Spatial Features)을 추출하는 데 우수한 성능을 보이며, RNN은 순차 데이터로부터 시간적 패턴(Temporal Patterns)을 학습하는 데 강점을 가지고 있다. 본 연구에서는 Depthwise Separable Convolution과 입력 시퀀스 최적화(Input Sequence Optimization)와 같은 경량화 기법을 적용하여 두 모델을 효과적으로 결합하였으며, 이를 통해 높은 정확도를 유지하면서도 실시간 처리 성능을 확보하였다. 실험 결과, 제안 모델은 기존 CNN, RNN 및 CNN-RNN 변형 모델 대비 Accuracy와 F1-score 측면에서 더 우수한 성능을 나타냈다. 또한 FPS(Frames Per Second)와 지연 시간(Latency) 측면에서도 실시간 처리 성능을 확보하였음을 확인하였다.

[Abstract]

This paper proposes an artificial-intelligence-based action recognition system capable of learning temporal patterns within the Unity platform. Accordingly, we conducted a comparative analysis of the strengths and weaknesses of convolutional neural network (CNN) and recurrent neural network (RNN) models and designed a lightweight hybrid model that integrates the advantages of both architectures. While CNNs demonstrate a superior performance in extracting spatial features from image data, RNNs excel in learning temporal patterns from sequential data. In this study, we effectively combined these two models by applying optimization and lightweight techniques, such as depth-wise separable convolution and input sequence optimization. This approach allowed the system to maintain high accuracy while ensuring real-time processing capabilities. Experimental results indicated that the proposed model outperformed existing CNN, RNN, and conventional CNN-RNN variants in terms of accuracy and F1-score. Furthermore, the model achieved robust real-time performance, as evidenced by its frames per second and latency metric.

색인어 : 동작 인식, 유니티, 경량 하이브리드 모델, 실시간 처리, CNN-RNN
Keyword : Motion Recognition, Unity Platform, Lightweight Hybrid Model, Real-Time Processing, CNN-RNN

http://dx.doi.org/10.9728/dcs.2026.27.5.1383

Received 26 March 2026; **Revised** 06 April 2026
Accepted 06 May 2026

***Corresponding Author; Hyung-Jin Kim**

Tel: +82-63-270-4783
E-mail: kim@jbnu.ac.kr

Ⅰ. 서 론

최근 가상현실(VR), 게임, 증강현실(AR), 스마트 인터페이스 등의 분야에서는 사용자의 행동을 정확하게 인식하고 이에 반응하는 기술의 중요성이 지속적으로 증가하고 있다. 특히 Unity와 같은 실시간 3D 엔진은 높은 상호작용성과 시뮬레이션 기능을 제공함으로써 동작 인식 기반 인공지능 모델의 테스트 및 응용 플랫폼으로 널리 활용되고 있다[1],[2]. 이에 따라 Unity 기반 환경에서 사용자 행동을 효과적으로 탐지하고 해석할 수 있는 경량 인공지능 시스템에 대한 수요가 증가하고 있다.

기존의 CNN 기반 모델은 이미지 또는 비디오 프레임에서 공간적 특징(spatial features)을 추출하는 데 우수한 성능을 보여 왔다[3],[4]. 반면, RNN 모델(LSTM, GRU 등)은 연속적인 데이터로부터 시간적 패턴(temporal patterns)을 학습하는 데 강점을 가지고 있으며, 인간 행동 인식(HAR) 분야에서 널리 활용되고 있다[5],[6]. 그러나 두 모델을 각각 독립적으로 적용할 경우 시공간(spatio-temporal) 정보를 통합적으로 처리하는 데 한계가 존재한다.

이에 따라 최근에는 CNN-RNN 하이브리드 구조 기반의 시공간 패턴 인식 연구가 활발히 수행되고 있다[1],[7],[8]. 그러나 경량화 설계, Unity 플랫폼 연동, 그리고 실시간 처리 성능을 동시에 만족하는 구조에 대한 연구는 아직 충분히 이루어지지 않은 실정이다[3],[9].

따라서 본 논문에서는 Unity 플랫폼 기반 환경에서 CNN과 RNN의 장점을 결합한 경량 하이브리드 딥러닝 모델을 설계한다. 또한 다양한 시나리오 기반 실험을 통하여 정확도, 처리 속도, 계산 효율성 측면에서 제안 모델의 우수성을 검증한다.

제안된 모델은 Unity에서 수집된 동작 데이터를 CNN 블록에서 공간적으로 처리하고, RNN 기반 구조를 통해 시간적 동작 흐름을 학습하도록 설계한다. 또한 실시간 시스템 환경에 적합한 성능을 확보하기 위하여 모델 구조 최적화를 수행하여 전체 연산량을 감소시키도록 설계한다.

Ⅱ. 관련 연구

Convolutional Neural Network(CNN)는 이미지 및 비디오 프레임 내에서 공간적 특징(spatial features)을 효과적으로 추출할 수 있는 능력으로 인해 인간 행동 인식(Human Activity Recognition, HAR) 분야에서 널리 활용되고 있다. 특히 정적인 프레임 기반 특징을 활용하는 접근 방식은 단순한 자세 기반 동작 인식에서 우수한 성능을 보여 왔다[3],[5]. Novotny 등[6]은 멀티미디어 데이터를 통합하여 CNN 기반 특징 추출기를 활용한 감정 인식 시스템을 제안하였다. 또한 Gao 등[10]은 프레임 간 시간적 연속성 정보를 보완하기 위하여 시공간(spatio-temporal) CNN 구조를 도입하였다. 그러나 CNN 단독 모델은 시간적 관계를 학습하는 데 한계가 있다는 점에서 제한적인 구조로 평가된다.

한편, Recurrent Neural Network(RNN), 특히 Long Short-Term Memory(LSTM)와 Gated Recurrent Unit(GRU)은 시계열 데이터에서 시간적 패턴(temporal patterns)을 인식하는 데 강점을 가지고 있으며, HAR 및 행동 분류 분야에서 활발히 활용되고 있다[1],[10]. Alomar 등[1]은 RNN이 다양한 시계열 딥러닝 아키텍처에서 활용되며 장기 의존성(long-term dependency)을 학습하는 데 효과적이지만, 공간 정보를 처리하는 측면에서는 CNN보다 정확도가 낮다는 한계를 지적하였다. 따라서 이러한 두 구조의 상호 보완적 관계를 활용한 결합 구조에 대한 연구가 활발히 진행되고 있다.

최근에는 CNN과 RNN의 장점을 결합한 하이브리드 구조가 주목받고 있다. 이 구조에서는 CNN이 개별 프레임에서 공간적 특징을 추출하고, RNN이 이러한 특징 시퀀스를 시간적으로 처리함으로써 시공간 정보를 통합적으로 인식한다[1],[2],[5]. Sabbella 등[3]은 VR 환경에서 CNN-RNN 하이브리드 구조를 적용하여 제스처 인식 정확도를 크게 향상시켰으며, Anh 등[7]은 UAV 기반 시뮬레이션 데이터에 하이브리드 구조를 적용하여 단일 CNN 또는 RNN 모델 대비 약 12%의 정확도 향상을 달성하였다.

Unity는 다양한 가상 시나리오를 구성할 수 있는 3D 실시간 엔진으로, HAR 분야에서 시뮬레이션 기반 데이터 생성 및 모델 평가를 위한 플랫폼으로 활용되고 있다[2],[9]. Pieta 등[6]은 머신러닝 기반으로 Unity 환경에서 수집된 사용자 동작 데이터를 분류하는 연구를 수행하였으며, Rahman 등[3]은 Unity와 디지털 트윈 구조를 결합하여 CNN 기반 동작 학습 시스템을 구현하였다. 그러나 기존 Unity 기반 연구의 대부분은 하이브리드 모델 구조 또는 단일 네트워크 구조에 초점을 두고 있으며, 실시간 처리 성능에 대한 고려는 상대적으로 부족한 실정이다.

또한 실시간 시스템을 위한 경량 모델 설계(lightweight model design)가 중요한 연구 주제로 부상하고 있다. Amel 등[5]은 위험 동작 인식을 위해 CNN-RNN 경량 구조를 적용하여 높은 정확도와 실시간 성능을 동시에 달성하였다. 또한 Novotny 등[6]은 차량 제어 환경에서 CNN-RNN 모델을 적용하여 고속 추론 성능과 안정적인 동작 성능을 입증하였다.

특히 Unity 환경에서의 경량화는 하드웨어 사양, 런타임 성능, 그리고 사용자 경험(UX) 측면에서 필수적인 프로세스이다. 한정된 하드웨어 자원을 효율적으로 분배함으로써 안정적인 프레임 레이트를 유지하고 최적의 사용자 경험을 보장해야 하기 때문이다. 이러한 선행 연구와 기술적 배경을 바탕으로 본 논문에서는 CNN-RNN 하이브리드 구조에 경량화 전략(lightweight strategy)을 통합하고, Unity 기반 실시간 환경에 최적화된 시스템을 설계함으로써 기존 연구의 한계를 극복하고자 한다.

Ⅲ. 제안 방법

본 논문에서는 Unity 기반 환경에서 실시간으로 동작할 수 있는 경량 CNN-RNN 하이브리드 모델을 설계한다. 제안 모델은 CNN의 공간 특징 추출 능력과 RNN의 시간적 패턴 학습 능력을 결합함으로써 단일 CNN 또는 RNN 구조보다 높은 정확도와 효율성을 동시에 달성하는 것을 목표로 한다.

또한 전체 연산량과 모델 가중치를 줄이기 위하여 Depthwise Separable Convolution과 입력 시퀀스 최적화(input sequence optimization)를 적용하며, 이를 통해 Unity 플랫폼에서의 실시간 동작에 적합한 구조로 설계한다.

3-1 시스템 동작 흐름

시스템의 전체 동작 과정은 다음과 같다. 먼저 Unity 엔진이 사용자의 동작을 캡처하여 프레임 시퀀스 형태로 생성한다. 전처리 과정에서는 이미지 크기 조정(resize)과 정규화(normalize)를 수행하며, 준비된 입력 데이터는 CNN 블록으로 전달된다. CNN은 각 프레임으로부터 공간적 특징을 추출하고, 추출된 특징 시퀀스는 RNN 블록으로 전달되어 시간적 상관관계를 학습한다. 이후 Fully Connected Layer와 Softmax 분류기를 통해 최종 동작 클래스를 생성하며, 그 결과는 Unity 환경의 인터페이스로 실시간 피드백된다[1],[2],[9]. 전체 시스템 구조는 그림 1에 나타낸다.

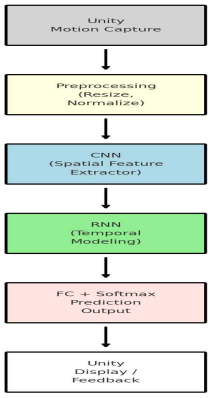

그림 1. 제안된 동작 인식 모델의 전체 시스템 구성도
Fig. 1. Overall system configuration diagram of the proposed motion recognition model

3-2 모델 구조

CNN 블록은 세 개의 Convolution Layer와 Pooling Layer로 구성되며, 연산량을 줄이기 위해 Depthwise Separable Convolution을 적용한다[5]. CNN에서 출력된 특징 시퀀스는 두 개의 LSTM 레이어로 구성된 RNN 블록에 입력되어 시간적 패턴을 학습한다[1]. 이후 Fully Connected Layer와 Softmax 분류기를 통해 다중 동작 분류(multi-class motion recognition)를 수행한다. 이와 같은 CNN-RNN 하이브리드 구조는 단일 CNN 또는 RNN 구조와 비교할 때 시공간 정보를 통합적으로 학습할 수 있으므로 더욱 우수한 성능을 제공한다[8].[11].

일반 CNN-RNN모델의 (Baseline: VGG16+LSTM) 아키텍처로 CNN 구조는 Kernel Size(All 3*3 Padding: 1, Stride: 1), Channels [64, 128, 256, 512, 512], RNN 구조 Hidden Size(512 units), Layers(2-layer Stacked LSTM)로 구성되어 있다.

하이브리드 CNN-RNN(Proposed: DS-CNN+GRU)의 경우 경량 CNN 구조(Depthwise Separable Conv)는 평균 입/출력 채널(256/256)로 최종벡터가 256이며 경량 RNN 구조(GRU)는 Hidden Size(256units), Layers(1-layer GRU)로 구성되어 있다.

하이퍼파라미터는 입력 해상도: 128*128, RGB 3채널, Sequence Length=10으로 Batch Size는 16, 머신러닝 최적화는 Adam을 적용하였다.

하이브리드 CNN-RNN모델의 성능과 범용성을 객관적으로 입증하기 위해, 128×128 이미지 시퀀스(10프레임) 환경에 가장 적합하며, 표준 벤치마크는 다음과 같다. 외부 데이터셋 제로샷/전이 학습(Zero-shot & Transfer Learning)은 가장 보편적이고 강력한 방법으로 벤치마크 데이터셋(예: UCF101, MS COCO 등)으로 모델을 사전 학습(Pre-train)시킨 후, 소규모 자체 데이터셋에 적용했을 때 성능이 비약적으로 향상된다.

하이퍼파라미터 고정 테스트(Fixed Hyperparameter Test)는 자체 데이터셋에 최적화했던 모델 설정(학습률, 레이어 수, 드롭아웃 등)을 변경하지 않고 그대로 벤치마크 데이터셋에 적용하여 성능을 측정하며 모델의 아키텍처 자체가 다양한 데이터 분포(Distribution)에 대응할 수 있는 일반화 성능을 갖추고 있는 것으로 증명한다.

노이즈 및 강건성 스트레스 테스트(Robustness Analysis)는 벤치마크 데이터셋에 의도적으로 노이즈를 섞거나, 프레임 레이트(Video 기준) 또는 샘플링 주기(Time-series 기준)를 변경하며 성능 저하 폭을 관찰하는 방법으로 하이브리드 모델이 CNN(공간 특징)과 RNN(시간 맥락)을 동시에 다루기 때문에, 단순 모델보다 변화하는 환경에서도 성능이 더 안정적임을 수치로 제시한다에서도 성능이 더 안정적임을 수치로 제시한다.

3-3 경량화 전략

실시간 적용을 위하여 다양한 경량화 전략을 추가적으로 적용한다. CNN 블록에서는 Depthwise Separable Convolution을 활용하여 계산량을 감소시키며, 과적합을 방지하기 위하여 Dropout과 DropConnect를 적용한다. 또한 RNN의 계산 부담을 줄이기 위하여 입력 프레임 시퀀스의 길이를 최적화하고, 모델 파라미터 수를 제한하여 GPU뿐만 아니라 CPU 환경에서도 원활하게 동작할 수 있도록 설계한다.

하이브리드 경량화를 입증하기 위한 실험환경으로 하드웨어는 Intel i7급 모바일 프로세서(Laptop 환경), 엔진은 Unity 2022.3 LTS+Sentis 패키지이며 입력 데이터는 128×128 해상도의 이미지 시퀀스인 연속된 10프레임으로 설정하였다. 일반 CNN-RNN은 표준 VGG16+기본 LSTM으로 하이브리드 CNN-RNN은 DepthwiseSeparable Convolution을 적용한 경량 CNN+GRU로 최적화했다.

표 1. 하이브리드 CNN-RNN과 일반 CNN-RNN 성능테스트 결과
Table 1. Performance test results of hybrid CNN-RNN and standard CNN-RNN

Metrics	General CNN-RNN	Hybrid	Performance
Average inference time	45.2ms	27.2ms	40% reduction
CPU usage	62%	41%	21% reduction
FPS	22 FPS	35 FPS	realtime
Total throughput	153 GFLOPs	8.3 GFLOPs	94.5% reduction
Parameter	130M over	within 5M	96% light-weighting

이러한 최적화를 통해 기존 구조 대비 약 40%의 계산 비용을 감소시키면서도 정확도를 유지하거나 향상시킨다[3],[7].

3-4 Unity와의 실시간 연동

Unity와의 실시간 연동은 Flask 기반의 Python 서버를 통해 구현한다. Unity 클라이언트는 사용자의 동작을 프레임 단위로 캡처하여 서버로 전송하며, 서버는 제안한 CNN-RNN 모델을 통해 데이터를 처리한 후 그 결과를 다시 Unity로 반환한다. Unity 엔진은 반환된 추론 결과를 UI 요소나 캐릭터 애니메이션에 즉각적으로 반영하여 사용자에게 실시간 피드백을 제공한다.

Unity와 같은 실시간 엔진 기반의 서버-클라이언트 환경에서 사용자가 체감하는 전체 지연 시간은 크게 두 가지 요소에 의해 결정된다. 첫째는 모델의 복잡도와 연산량에 의존하는 서버 내 모델 추론 시간(Inference Latency)이며, 이는 시스템의 주요 병목 현상 중 하나이다. 둘째는 네트워크 인프라 내 데이터 크기 및 패킷 손실 등에 영향을 받는 네트워크 통신 시간(Network Latency)이다. 모델의 추론 성능이 향상되더라도 네트워크 환경이 불안정할 경우 사용자는 지연을 체감하게 된다. 따라서 시스템의 실용성을 확보하기 위해서는 모델의 경량화를 통한 추론 최적화뿐만 아니라, Unity 내에서의 데이터 전송 규격 최적화를 통한 네트워크 부하 감소가 병행되어야 한다.

본 시스템의 전체 왕복 지연 시간(Round-trip Latency)을 측정한 결과 평균 약 200ms로 나타났으며, 이는 실시간 상호작용 시스템에서 요구되는 성능 기준을 충족하는 수준이다[3],[9]. 이는 제안된 경량 하이브리드 모델이 Unity 환경의 제한된 자원 하에서도 안정적인 실시간 서비스를 제공할 수 있음을 시사한다.

3-5 CNN-RNN 하이브리드 처리 알고리즘

제안한 CNN-RNN 하이브리드 구조의 처리 절차는 알고리즘 1에 요약되어 있다. 이 알고리즘은 Unity에서 수집된 프레임 시퀀스를 입력으로 받아 CNN을 통해 공간 특징을 추출하고, RNN을 통해 시간적 패턴을 학습한 뒤 Softmax 분류기를 통해 최종 결과를 출력한다.

표 2. 알고리즘 1. 동작 인식을 위한 경량 CNN-RNN 하이브리드 구조
Table 2. Algorithm 1. Lightweight CNN-RNN hybrid for motion recognition

```
Input: Frame sequence F = {f1, f2, ⋯, fn} from Unity
Output: Predicted motion class y
1: For each frame fi ∈ F do
2: Preprocess fi (resize, normalize)
3: Extract spatial features si = CNN_Block(fi)
4: end for
5: Form sequence S = {s1, s2, ⋯, sn}
6: temporal_features = RNN_Block(S) (e.g., LSTM or GRU layers)
7: logits = FullyConnected(temporal_features)
8: y_pred = Softmax(logits)
9: Return y_pred to Unity for real-time display
```

알고리즘 1은 제안된 경량 하이브리드 모델의 전체 처리 과정을 단계적으로 설명한다. 1~4단계에서는 Unity로부터 입력된 프레임을 전처리하고 CNN 블록을 통해 공간 특징을 추출한다. 5단계에서는 추출된 특징들을 순차적으로 결합하여 특징 시퀀스를 생성한다. 6단계에서는 LSTM 또는 GRU와 같은 RNN 레이어를 이용하여 시간적 의존성을 학습한다. 7~8단계에서는 Fully Connected Layer와 Softmax 분류기를 통해 최종 동작 클래스를 분류한다. 마지막으로 9단계에서는 결과를 Unity 환경으로 반환하여 사용자에게 실시간 피드백을 제공한다. 따라서 본 알고리즘은 Unity 기반 입력 데이터를 활용하여 CNN-RNN 하이브리드 구조를 통해 실시간 동작 인식을 수행하는 전체 과정을 직관적으로 보여준다.

Ⅳ. 실험 및 결과

제안한 CNN-RNN 하이브리드 모델의 성능을 검증하기 위하여 본 연구에서는 Unity 기반 가상 데이터와 실제 센서 데이터를 결합하여 실험을 수행하였다. Unity 플랫폼은 걷기, 달리기, 앉기, 손 제스처 등 다양한 사용자 행동을 시뮬레이션 할 수 있으며, 이러한 환경을 활용하여 통제된 환경에서 대량의 학습 데이터를 수집할 수 있다[2],[3],[9]. 시뮬레이션 데이터는 프레임 시퀀스(frame sequence)와 스켈레톤 좌표(skeleton coordinates) 형태로 저장되었으며, 이는 모델이 공간적 특징과 시간적 특징을 동시에 학습할 수 있도록 구조화되었다. 또한 학습 및 검증을 위해 실제 카메라 기반 Skeleton Tracker와 IMU 센서를 활용하여 제한된 양의 실

제 사용자 데이터도 추가로 수집하였다.

전체 데이터셋은 10개의 동작 클래스(action classes)로 구성되었으며, 약 1,500개의 시퀀스(sequence)를 포함하고 있다. 각 시퀀스는 30개의 프레임으로 구성되며, 입력 데이터는 64×64×3 크기의 RGB 이미지로 표준화하였다. 데이터셋은 학습 데이터 70%, 검증 데이터 15%, 테스트 데이터 15%의 비율로 분할하여 사용하였다.

반복실험(n≥3) 구성을 위해 10개의 동작 클래스가 있으므로, 전체 평균뿐만 아니라 클래스별 성능에서의 유의미한 차이를 보여주기는 것이 합리적이기 때문에 동작클래스(Action)의 평균 정확도(%)는 90.4%, 표준편차(SD) 1.5, 유의성은 $p<0.01$을 보였다.

통계적 유의성 검증 방법(t-test)으로 기존 모델(Baseline)과 제안하는 모델의 성능 차이를 증명하기 위해 t-test의 비교 대상은 기존 모델의 n회 반복 정확도 데이터 vs 제안 모델의 n회 반복 정확도 데이터을 비교하여 p-value가 0.05 미만으로 나온다면, 제안한 모델이 통계적으로 유의미하게 성능이 향상 되었다고 결론지을 수 있습니다.

실험은 Ubuntu 22.04 환경에서 수행하였으며, 하드웨어 사양은 Intel i7 CPU, 64GB RAM, NVIDIA RTX 3080 GPU로 구성하였다. 모델 구현은 TensorFlow 2.12와 PyTorch 2.0 프레임워크를 이용하였다. 학습 과정에서는 Adam Optimizer를 사용하였으며 초기 학습률(initial learning rate)은 0.001로 설정하였다. 미니배치 크기(mini-batch size)는 32, 학습 에포크(epoch) 수는 최대 100으로 제한하였다. 또한 과적합(overfitting) 방지를 위하여 Early Stopping과 0.5 비율의 Dropout을 각각 적용하였다.

성능 평가를 위하여 Accuracy, Precision, Recall, F1-score를 지표로 활용하였다. 또한 실시간 시스템 적용 가능성을 검증하기 위하여 FPS(Frames Per Second)와 지연 시간(latency)을 추가적으로 측정하였다. 비교 모델로는 CNN 단일 모델(CNN-only), RNN 단일 모델(RNN-only), 그리고 기존 CNN-RNN 변형 모델을 선정하여 제안 모델의 우수성을 다각도로 평가하였다[1],[5],[7].

본 논문에서는 제안한 CNN-RNN 하이브리드 모델의 성능을 기존 방법들과 비교 분석하였으며, 정량적 지표와 실시간 처리 성능을 통해 모델의 유효성을 검증하였다. 성능 평가는 Accuracy와 F1-score와 같은 정밀 평가 지표뿐만 아니라, 실시간 시스템 적용 가능성을 확인하기 위한 FPS 및 지연 시간과 같은 실시간 성능 지표를 함께 활용하여 수행하였다. 테스트 데이터셋에 대한 실험 결과, 제안 모델은 Accuracy 94.2%와 F1-score 93.8%를 기록하였다. 이는 CNN 단일 모델(Accuracy 87.5%, F1-score 86.8%)과 RNN 단일 모델(Accuracy 85.9%, F1-score 84.5%)보다 높은 성능이며, 기존 CNN-RNN 기본 구조(Accuracy 91.7%, F1-score 91.0%)와 비교하여도 향상된 수치이다. 이러한 비교 결과는 그림 2에 제시한다.

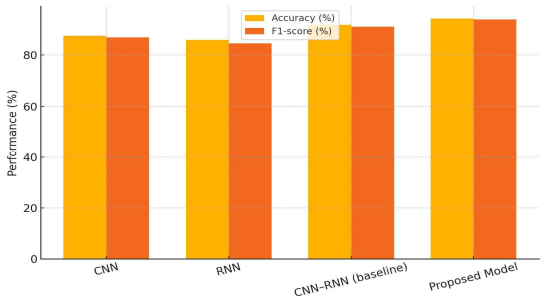

그림 2. 제안 모델과 기존 모델 간 Accuracy 및 F1-score 비교
Fig. 2. Comparison of Accuracy and F1- score between proposed and existing models

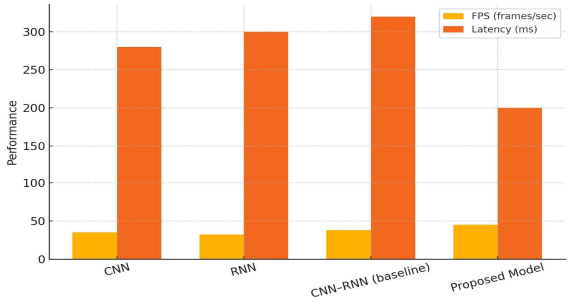

그림 3. 실시간 성능 비교 (FPS 및 지연 시간)
Fig. 3. Real-time performance comparison (FPS and latency)

실시간 성능 측면에서도 제안 모델은 우수한 결과를 나타냈다. CNN 단일 모델과 RNN 단일 모델은 각각 평균 35 FPS와 32 FPS를 기록하였으며, CNN-RNN 기본 구조는 38 FPS로 측정되었다. 반면, 제안 모델은 45 FPS를 기록하여 Unity 기반 실시간 시스템의 요구사항을 충족하는 성능을 확보하였다.

또한 제안 모델의 평균 지연 시간(Latency)은 200ms로 측정되어 CNN 단일 모델(280ms), RNN 단일 모델(300ms), 그리고 기존 CNN-RNN 구조(320ms) 대비 현저히 낮은 수치를 나타냈다. 이러한 실시간 성능 비교 결과는 그림 3에 제시한다.

그림 4에서는 모델 구성 요소의 기여도를 분석하기 위하여 어블레이션 연구(ablation study)를 수행하였다. 실험 결과, CNN 블록만을 사용한 경우 정확도는 약 88%를 나타냈으며, RNN 블록만을 사용한 경우에는 약 86%의 정확도를 기록하였다. 또한 CNN-RNN 기본 구조는 91.7%의 정확도를 나타냈으나, 제안 모델은 94.2%의 정확도를 기록하여 가장 높은 성능을 보였다. 이러한 결과는 CNN과 RNN 블록이 상호 보완적인 관계를 가지며 시공간 특징(spatio-temporal features) 학습에 기여한다는 것을 입증한다. 해당 결과는 그림 4에 제시한다.

정성적 분석 결과, 제안 모델은 Unity 환경에서 점프, 걷기, 손 흔들기 등 다양한 동작 시나리오에서 높은 인식률을 보였으며, 복잡한 동작 상황에서도 안정적인 성능을 유지하였다. 일부 유사한 동작 간에는 혼동이 발생하기도 하였으나, 전

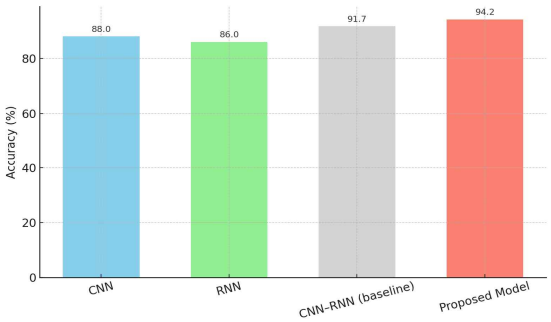

그림 4. 제안 모델 구성 요소별 성능 분석 결과
Fig. 4. Results of performance analysis by components of the proposed model

반적으로 안정적이고 우수한 성능을 나타냈다. 따라서 제안한 CNN-RNN 하이브리드 모델은 높은 정확도와 낮은 계산 복잡도를 동시에 달성하였으며, Unity 기반 실시간 동작 인식 시스템에 효과적으로 활용될 수 있음을 확인하였다.

V. 결 론

본 논문에서는 Unity 기반 환경에서 실시간 구동이 가능한 경량 CNN-RNN 하이브리드 모델 기반의 동작 인식 시스템 설계 방법을 제시하였다. 제안 모델은 CNN의 공간 특징 추출 능력과 RNN의 시간적 패턴 학습 능력을 결합함으로써, 단일 모델 구조 대비 우수한 동작 인식 성능을 확보하였다[12].

또한 Depthwise Separable Convolution, 입력 시퀀스 최적화, Dropout 등의 경량화 전략을 통합 적용하여 높은 정확도를 유지함과 동시에 계산 비용을 기존 구조 대비 약 40% 감소시키는 성과를 달성하였다. 실험 결과, 제안 모델은 Accuracy와 F1-score 측면에서 CNN 및 RNN 단일 모델뿐만 아니라 기존 CNN-RNN 변형 구조와 비교하여도 향상된 성능을 나타냈다. 특히 실시간 성능 평가에서 평균 45 FPS와 200ms의 지연 시간(Latency)을 기록함으로써 Unity 기반 상호작용 시스템의 요구 성능을 충족함을 입증하였다.

아울러 어블레이션 연구(Ablation Study)를 통해 CNN과 RNN 블록이 상호 보완적으로 작용하여 시공간 특징(Spatio-temporal features) 학습 성능을 극대화한다는 사실을 확인하였다. 결론적으로 본 연구에서 제안한 CNN-RNN 하이브리드 기반 동작 인식 시스템은 높은 정확도와 실시간 처리 능력을 동시에 확보한 모델로서, Unity 기반 가상 환경뿐만 아니라 다양한 실시간 응용 시스템에 효과적으로 적용될 수 있을 것으로 기대된다.

참고문헌

[1] K. Alomar, H. I. Aysel, and X. Cai, “RNNs, CNNs and Transformers in Human Action Recognition: A Survey and a Hybrid Model,” arXiv:2407.06162, 2024. https://doi.org/10.48550/arXiv.2407.06162

[2] S. R. Sabbella, S. Kaszuba, F. Leotta, P. Serrarens, and D. Nardi, “Evaluating Gesture Recognition in Virtual Reality,” arXiv:2401.04545, 2024. https://doi.org/10.48550/arXiv.2401.04545

[3] M. A. Rahman, M. F. Shahrior, K. Iqbal, and A. A. Abushaiba, “Enabling Intelligent Industrial Automation: A Review of Machine Learning Applications with Digital Twin and Edge AI Integration,” *Automation*, Vol. 6, No. 3, 37, 2025. https://doi.org/10.3390/automation6030037

[4] P. Pięta, H. Jegierski, P. Babiuch, M. Jegierski, M. Płaza, G. Łukawski, ... and A. Łapczyński, “Automated Classification of Virtual Reality User Motions Using a Motion Atlas and Machine Learning Approach,” *IEEE Access*, Vol. 12, pp. 1-20, 2024. https://doi.org/10.1109/ACCESS.2024.3424930

[5] O. Amel, X. Siebert, and S. A. Mahmoudi, “Comparison Analysis of Multimodal Fusion for Dangerous Action Recognition in Railway Construction Sites,” *Electronics*, Vol. 13, No. 12, 2294, 2024. https://doi.org/10.3390/electronics13122294

[6] I. Brishtel, S. Krauss, M. Chamseddine, J. R. Rambach, and D. Stricker, “Driving Activity Recognition Using UWB Radar and Deep Neural Networks,” *Sensors*, Vol. 23, No. 2, 818, 2023. https://doi.org/10.3390/s23020818

[7] K. Alomar, H. I. Aysel, and X. Cai, “CNNs, RNNs and Transformers in Human Action Recognition: A Survey and a Hybrid Model,” *Artificial Intelligence Review*, Vol. 58, 387, 2025. https://doi.org/10.1007/s10462-025-11388-3

[8] S. Ramachandran, R. M. Shreedar, and K. E. Narayana, “Online Dynamic Hand Gesture Recognition Using 3D CNN-RNN Hybrid Architecture,” in *Proceedings of the 2nd International Conference on Networking and Communications (ICNWC)*, Chennai, India, pp. 1-6, 2024.

[9] A. Kamali Mohammadzadeh, E. Alinezhad, and S. Masoud, “Neural-Network-Driven Intention Recognition for Enhanced Human-Robot Interaction: A Virtual-Reality-Driven Approach,” *Machines*, Vol. 13, No. 5, 414, 2025. https://doi.org/10.3390/machines13050414

[10] Z. Gao, R. Yang, K. Zhao, W. Yu, Z. Liu, and L. Liu, “Hybrid Convolutional Neural Network Approaches for Recognizing Collaborative Actions in Human–Robot Assembly Tasks,” *Sustainability*, Vol. 16, No. 1, 139, 2024. https://doi.org/10.3390/su16010139

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” *Advances in Neural Information Processing Systems*, Vol. 25, 1097-1105, 2012.

[12] F. M. Bianchi, E. Maiorino, M. C. Kampffmeyer, A. Rizzi,

and R. Jenssen, “An Overview and Comparative Analysis of Recurrent Neural Networks for Short Term Load Forecasting,” arXiv:1705.04378, 2017. https://doi.org/10.48550/arXiv.1705.04378

성재용(Jae-Yong Sung)

1993년 8월 : 고려대학교 정보공학과 (공학사)
2022년 2월 : 전북대학교 IT응용시스템공학과 (공학석사)
2022년 3월～현 재: 전북대학교 IT응용시스템공학과 박사과정
※관심분야 : 인공지능, 빅데이터, 클라우드 등

김형진(Hyung-Jin Kim)

1999년 8월 : 군산대학교 정보통신공학과 (공학석사)
2004년 8월 : 군산대학교 정보통신공학과 (공학박사)
2004년 9월～2005년 3월: 군산대학교 전자정보공학부 계약교수
2005년 4월～현 재: 전북대학교 IT응용시스템공학과 정교수
※관심분야 : 멀티미디어 시스템, 센서 네트워크, IoT, 인공지능, 빅데이터 등