Journal of Korea Multimedia Society Vol. 21, No. 5, May 2018(pp. 626-637)
https://doi.org/10.9717/kmms.2018.21.5.626

시 공간 정규화를 통한 딥 러닝 기반의 3D 제스처 인식

채지훈[†], 강수명[††], 김해성[†††], 이준재[††††]

Deep Learning Based 3D Gesture Recognition Using Spatio-Temporal Normalization

Ji Hun Chae[†], Su Myung Gang[††], Hae Sung Kim[†††], Joon Jae Lee[††††]

ABSTRACT

Human exchanges information not only through words, but also through body gesture or hand gesture. And they can be used to build effective interfaces in mobile, virtual reality, and augmented reality. The past 2D gesture recognition research had information loss caused by projecting 3D information in 2D. Since the recognition of the gesture in 3D is higher than 2D space in terms of recognition range, the complexity of gesture recognition increases. In this paper, we proposed a real-time gesture recognition deep learning model and application in 3D space using deep learning technique. First, in order to recognize the gesture in the 3D space, the data collection is performed using the unity game engine to construct and acquire data. Second, input vector normalization for learning 3D gesture recognition model is processed based on deep learning. Thirdly, the SELU(Scaled Exponential Linear Unit) function is applied to the neural network's active function for faster learning and better recognition performance. The proposed system is expected to be applicable to various fields such as rehabilitation cares, game applications, and virtual reality.

Key words: Deep Neural Network, Gesture Recognition, Game Application

1. 서 론

의사소통을 위해서 인간은 언어를 활용하는 것 외에도 손동작이나 몸동작 같은 제스처를 활용함으로써 쉽게 정보를 전달할 수 있다. 특히 컴퓨터를 통해서 구축한 가상 환경에서는 제스처를 사용함으로써 사람과 사람, 컴퓨터와 사람 사이에서 직관적으로 정보를 전달하고 명령할 수 있다.

특히, 제스처 인식은 많은 플랫폼에서 활용되고 있으며 입력하는 방식 또한 적용환경에 따라 적절한 형태를 채택할 수 있도록 발전되었다. 특히 가상의 3차원 공간 안에서는 기존의 입력 형태인 마우스, 키보드의 불편함을 대체하고 효과적인 상호작용 형태가 요구됨에 따라 제스처 인식에 대한 필요성이 대두되었다. 기존 입력 방식은 2차원 환경의 특수성에 최적화되었을 뿐 아니라, 3차원 환경에서는 몰입도와

※ Corresponding Author : Joon Jae Lee, Address: 1095 Dalgubeol-daero, Dalseo-gu, Daegu 42601, Korea, TEL : +82-53-580-6682, FAX : +82-53-580-5165, E-mail : joonlee@kmu.ac.kr
Receipt date : Mar. 16, 2018, Revision date : Apr. 20, 2018
Approval date : Apr. 23, 2018
[†] Dept. of Computer Engineering, Graduate School, Keimyung University
(E-mail : cowlgns21@naver.com)
[††] Dept. of Computer Engineering, Graduate School, Keimyung University
(E-mail : smgang.kmu@gmail.com)
[†††] Faculty of Computer Engineering, Keimyung University
(E-mail : gotjd3697@gmail.com)
[††††] Faculty of Computer Engineering, Keimyung University
※ This work was supported by the Industry-Academic Co-Development R&D Program (C0512108) funded by the Ministry of SMEs and Startups(MSS, Korea)

직관성을 떨어뜨리기 때문에 새로운 입력 방법이 필요하게 되었다. 또한, 3차원 환경이 아니더라도 특수한 게임 인터페이스를 가지는 환경에서는 마우스의 조작 방식을 대체하기 위해서 립 모션 등의 기기를 활용해 효과적으로 게임에 몰입할 수 있다. 이러한 특징은 제스처의 직관적인 패턴과 게임의 상황을 자연스럽게 결합한다면 마우스 보다 효과적인 조작 방식이 될 수 있다[1-4].

제스처 인식은 결정 트리(decision tree), 랜덤 포레스트(random forest), SVM(Support Vector Machine), 신경망(neural network) 계열 알고리즘 등의 기계학습에서 파생된 다양한 알고리즘을 활용할 수 있다. 하지만 제스처 인식은 상대적으로 하드웨어 성능을 크게 요구하며 연산 속도가 느린 단점이 있다. 반면에 적은 자원을 필요로 하는 알고리즘은 인식률이 떨어져 효과적으로 제스처 인식을 수행할 수 없다. 경우에 따라서는 데이터의 대표적인 특징을 추출하는 것이 필요하며, 이러한 과정은 제스처를 인식하는데 복잡도를 높이며 많은 연산 시간을 요구할 뿐만 아니라 제한된 환경에서만 알고리즘이 구동될 가능성이 크다. 제스처 인식 알고리즘은 다양한 환경에서 강건해야 하며, 예외적인 상황에서도 효과적으로 제스처 인식을 수행할 수 있어야 한다[5 - 9].

제스처 인식 알고리즘의 성능을 높이기 위해서는 적절한 하드웨어 성능이 요구된다. 게임 어플리케이션과 같이 실시간으로 렌더링 작업을 수행하는 시스템에서는 높은 인식률을 위해 알고리즘의 요구 조건을 만족하기 쉽지 않다. 일반적으로 알고리즘의 속도를 개선하기 위해서는 Open CL, CUDA를 활용해 병렬처리하는 방식을 선택할 수 있다. 하지만 실시간으로 GPU를 사용하여 렌더링을 처리하는 프로그램에서는 이를 적용하기가 쉽지 않다. 또한, 제스처를 입력하는 형태, 환경 그리고 데이터의 가변성에 따라 알고리즘의 속도와 성능 평가가 달라질 수 있다[1, 10].

최근 신경망을 기반으로 한 딥 러닝(deep learning)은 빅 데이터의 지원, 하드웨어 성능이 발달함에 따라 크게 주목받고 있다. 딥 러닝 알고리즘은 GPU를 활용하여 데이터를 빠르게 연산을 할 수 있으며 효과적으로 데이터를 학습함으로써 다양한 변환에도 강건함을 유지한다. 딥 러닝의 장점은 한 분야에서만 국한되는 것이 아니라 음성 인식, 자연어 처리, 영상 및 이미지 분류에서도 넓게 사용되고 있다. 딥 러닝은 다양한 분야에서 돌파구로 인정되고 있으며 제스처 인식 분야에 적용함으로써 앞서 기술된 문제들을 해결할 수 있을 것이다. 하지만 딥 러닝 알고리즘은 높은 연산량, 소규모 데이터 학습에 대한 과적합 현상과 같은 문제를 갖고 있으며, 이를 해결한다면 효과적으로 제스처 인식을 수행할 수 있다[1, 11].

제스처 인식의 경우 3차원 공간에서 수행되나, 기존의 잘 알려진 CNN형태의 알고리즘에 기반한 딥러닝 모델들은 이동 경로에 대하여 프레임별 영상을 획득하여 이를 연산하기 때문에 고 사양의 컴퓨터에서만 동작되는 제약이 있다. 또한 3차원 정보를 2차원으로 투사할 때 발생하는 정보의 손실과 방향 정보에 대한 구별이 없는 문제점(기존연구 결과)이 있다.

본 연구에서는 실시간 처리가 필요한 계산 효율 및 성능을 얻기 위하여 제스처 이동 경로에 대한 위치 값을 벡터화 하는 방법을 제안한다. 이를 통해 기존 연구에서 발생했던 정보 손실을 줄이고 제스처를 그리는 3차원 공간의 정보를 그대로 활용하여 정확도를 높인다. 제안한 방법은 사용자마다 다른 제스처의 수행속도 및 형태크기에 불변한 데이터를 얻기 위하여 시간 정규화 과정과 공간 정규화 과정을 수행한다. 또한 해당 벡터를 1차원 레이어로 구성하여 기존 2차원 영상을 입력한 연구와는 달리 학습 및 수행의 연산 시간을 획기적으로 줄인다. 제스처 인식에 2차원 영상의 CNN 알고리즘을 사용하면, 컨볼루션 단계에서 계산비용이 증가하게 되나, 본 연구에서 제안하는 1차원 벡터 처리를 위한 다단의 MLP는 계산 비용이 훨씬 적어 실시간 처리가 필요한 게임 응용 단계에서 매우 효율적이다.

제안한 방법의 성능에 대한 신뢰성을 확보하기 위해, 21종의 제스처에 대해서 각 500개의 데이터를 확보한다. 같은 종류의 제스처라도 재활치료 적용을 위해 팔과 손이 불편한 환자와 유사한 데이터가 함께 수집되도록 하였다. 특히, 팔과 어깨 부분이 불편한 사람이 직접 참가하여 데이터의 신뢰도 높일 수 있도록 구성하였다. 제안한 방법으로 실험한 결과 빠른 학습속도와 실행속도 뿐만 아니라, 인식률에서도 매우 우수한 성능을 나타내어, 실제 병원에서 재활치료를 위한 매우 효율적인 시스템으로 사용될 수 있을 것이다.

2. 기존 연구

본 장에서는 기존 제스처 인식 관련 연구를 언급한다. 제스처 인식을 위해서 다양한 기계학습 알고리즘이 활용되어 왔다. 제스처의 데이터는 연속성을 가지고 입력되기 때문에, HMM(Hidden Markov Model), K-NN(K-Nearest Neighbors) 등의 알고리즘이 사용되어 왔으며, 기존 기계학습 알고리즘으로 SVM, 랜덤 포레스트 등의 알고리즘을 활용하여 제스처의 인식률을 높이는 방향으로 발전되어왔다[12-13].

Kim과 Lee[12]의 연구에서는 카메라를 통해서 2차원 평면에 제스처의 위치를 투영(projection)한 벡터를 16개의 방향으로 정규화(normalization)한다. 특징 추출을 통하여 연속적인 데이터에서 양자화된 입력 데이터는 HMM을 통해서 제스처 인식을 처리한다. 제안된 알고리즘은 비교적 빠른 제스처 인식 속도와 높은 인식률을 기대할 수 있다. 하지만 2D 평면에 투영하는 것을 통해서 제스처 인식을 수행하기 때문에, 움직임이 많은 직접적인 가상공간에 적용하기 어렵다.

Xu와 Lee[14]의 연구에서는 2차원 및 3차원 공간을 기반으로 HMM을 활용하여 제스처 인식을 수행하였다. 제스처 데이터의 특징 추출은 2차원 위치를 통해서 8방향의 각도를 추출하여 HMM을 통해 제스처 인식을 처리한다. 쉬운 패턴에서는 높은 성능 확인할 수 있으나 2차원 위치를 추출하기 때문에 실질적인 3차원 공간에서 복잡한 패턴을 인식하는 것은 해결 과제로 남아있다.

조선영[15]등의 연구는 키넥트 센서로부터 획득한 정보를 통하여 제스처를 인식한다. 획득한 데이터에 대하여 특징을 추출하기 위해서 다각도 결합 히스토그램이 활용된다. 추출된 특징은 랜덤 결정 포레스트를 분류기를 통해서 행동 인식과 손 제스처 인식을 수행했다. 기존 연구보다 상대적으로 인식률은 개선했지만, 손 제스처 인식에 대해서 특화된 모델이 아니기 때문에 3차원, 이하 3D(3 Dimension) 공간에서의 손 제스처 인식의 어려운 점이 있다. 따라서, 인식률에 대한 성능 개선이 필요하다.

오주희[16]등의 연구에서는 키넥트 카메라로 획득할 수 있는 관절 정보를 활용하여 제스처를 인식했으며, 3개의 이진 트리를 구성하여 입력되는 3차원 제스처 데이터를 구분한다. 그리고 제스처 인식의 성능 향상을 위해 다중 클래스 SVM을 동시에 활용한다. 그러나 시작점과 방향이 같은 입력 제스처에 대해서만 인식 성능을 할 수 있는 제한점이 있다.

장민선[17]등의 연구에서는 3축 가속도 센서를 활용하여 간단한 제스처 패턴을 인식한다. 제스처 인식을 위해서 신경망과 결정 트리 분류기를 사용했고 가속도 센서를 활용하여 사용자의 동작을 직접 측정할 수 있는 장점이 있다. 하지만 상대적으로 쉬운 제스처 패턴을 신경망과 결정 트리를 통해서 인식하였다. 단순한 좌우상하 방향을 인식하는 데 있어서 적절한 방법을 취했지만 복잡한 패턴을 인식하기에는 어려울 것이다.

Xu[18]의 연구에서는 가상현실에서 자주포를 훈련하기 위해서 제스처 인식을 활용하는 연구를 진행하였다. 이 방법은 신경망을 활용하여 실용적으로 제스처 인식을 적용했다. 하지만 손의 특정적인 자세를 인식하고 신경망의 MLP 모델을 활용함으로써 손의 움직임에 대한 대응이 어렵고 MLP 신경망 모델 자체의 한계점이 존재한다.

이찬수[19]등의 연구에서는 스마트폰, PDA의 모바일 기기에 장착되어 있는 카메라를 활용하여 손 제스처의 움직임을 인식한다. 이 방법은 광학 흐름(optical flow)을 활용하여 히스토그램을 통해 손의 움직임을 추적한다. 추출한 손의 위치 데이터는 SVM, K-NN, 베이지안(Bayesian) 분류기를 통해서 학습하고 인식하였다. 이 방법은 다양한 기계학습 모델을 활용하여 비교적 높은 인식률을 나타내었으나 인식 모델을 선정하기가 어려울 뿐만 아니라 쉬운 제스처 패턴을 대상으로 실험을 수행하였기 때문에, 복잡한 패턴의 제스처에 대응하기가 어렵다.

채지훈[1]등의 연구는 본 연구의 선행 연구로서 인지 재활 게임, 가상 현실 등 게임 어플리케이션 기반에서 실시간으로 컨볼루션 신경망을 활용하여 제스처를 인식하는 연구를 진행하였다. 이 방법은 가상현실 플랫폼으로 획득할 수 있는 별도의 컨트롤러, 마우스를 통하여 제스처를 입력할 수 있다. 추출한 제스처 데이터는 임의의 평면에 투영하는 전처리 방법을 거친다. 또한 전처리 방법은 기존의 연속적인 데이터에서 양자화된 이미지 데이터를 획득하여 컨볼루션 신경망 모델을 학습 가능하게 한다. 학습된 컨볼루션 신경망 모델은 제스처 데이터를 분류한다. 하지만 이 방법은 실시간으로 제스처 인식을 수행하

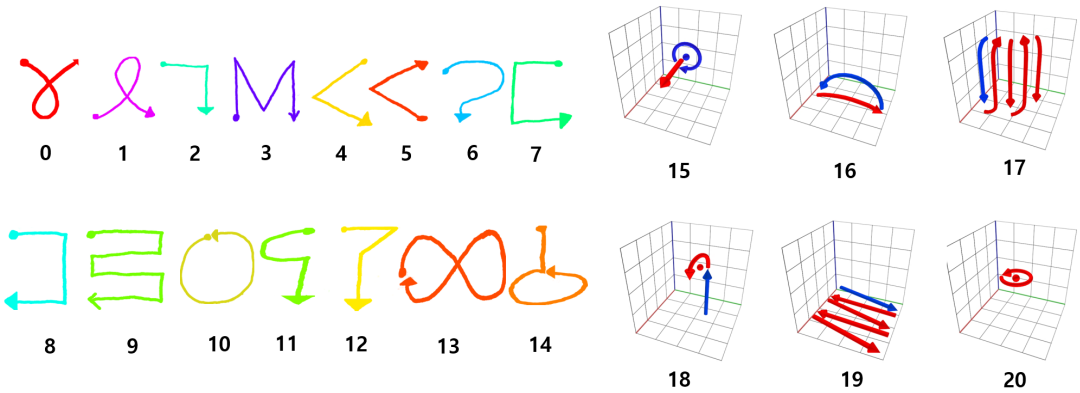


Fig. 1. Gesture type.

기 위해서 별도의 서버 및 분산 처리를 구축하였으나 임의의 평면에 투영하는 전처리를 수행하기 때문에 제한된 방향과 공간에서 제스처를 인식하는 한계점이 있다. 상기 언급한 제스처 인식 관련 기존 연구는 제한 사항과 성능 개선에 대한 필요성들을 확인할 수 있었다.

본 연구에서는 기존 연구에서 언급된 제한점을 통해서 효율적이고 보완된 제스처 인식 알고리즘을 제안하고 구축한다.

3. 딥 러닝 기반의 3D 제스처 인식 모델

3.1 데이터 구성

본 단원에서는 제스처의 학습 데이터에 대한 구성 및 수집 방법을 언급한다. 딥 러닝은 많은 데이터를 학습하고 모델의 파라미터를 조정하여 예측모델을 생성한다. 본 연구에서는 예측 모델을 생성하기 위해서 데이터를 구성하고 수집 방법을 제시한다. 또한 3차원 벡터 형태를 입력 데이터로 활용하기 때문에, 이의 적절한 학습 데이터를 구성하는 방법을 제시한다. 딥 러닝 기반의 제스처 인식 모델의 학습을 위해서 제스처 종류를 선정했으며 Fig. 1과 같다.

제스처 종류는 총 21개로 방향성을 고려하여 쉬운 패턴, 어려운 패턴, 비슷한 패턴까지 구성했으며, 곡선과 직선, 깊이 값을 혼합해서 제스처 인식의 복잡도를 높였다. 그리고 15 - 20번의 제스처는 각각 문 열기, 책장 넘기기, 머리 빗기, 물컵 들고 물 먹기, 닦기(책상, 바닥), 물병 따기 동작을 모사했다. 15-20번의 제스처는 파란색 선부터 시작하고, 중심에 점이 있는 경우 언급한 동작의 모사를 통한 고정 동작이다. 나머지 제스처는 방향을 따라서 입력을 수행하면 된다.

3.2 데이터 수집 방법

본 연구에서는 3차원 손 위치를 추적할 수 있는 립 모션(leap motion)을 활용하고 Unity3D 게임 엔진을 통해서 GUI를 구성하여 제스처 데이터를 수집했다. Fig. 2는 립 모션을 활용하여 제스처 데이터를 수집하고 있는 것을 보여준다.

제스처 데이터 수집 과정은 라벨링을 입력하고 스페이스 바를 활용하여 제스처의 시작과 끝을구성했다. 그리고 수집한 데이터는 시간별로 라벨링 번호가 입력된 폴더에 저장한다. 입력한 데이터는 3차원 손 위치를 저장한 데이터 파일과 제스처 인식 모델에 입력하기 위해서 정규화한 데이터를 저장한다.

3.3 시 공간 정규화 방법

본 연구에서는 딥 러닝 기반의 제스처 인식 모델에 적합한 입력 데이터 생성하고, 효율적으로 정규화

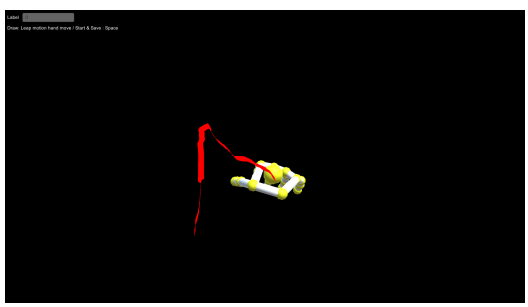


Fig. 2. Data collection using leap motion.

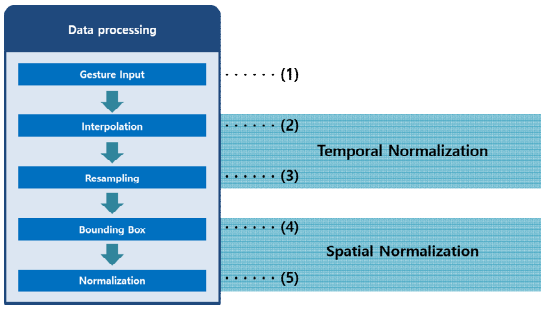


Fig. 3. Spatio-temporal normalization.

하는 방법을 제안한다. 이 방법은 사용자의 입력 제스처에 대해 시간적 및 공간적 정규화를 통해서 사용자의 다양한 형태의 제스처 행동에 대응하기 위한 것이다. 시간적 정규화는 사용자의 제스처 속도에 강건한 입력 데이터를 생성하고, 공간적 정규화는 제스처의 크기나 이동에 불변한 제스처 데이터를 얻기 위한 것이다. 정규화 과정은 Fig. 3과 같다.

전체적인 데이터 처리 과정은 제스처 데이터를 입력받고 보간을 통해서 데이터의 수를 조정하는 시간적 정규화 방법과 공간영역에서 정규화하는 방법을 적용하여 적절한 데이터 형식을 생성한다. 데이터 처리 과정은 중요하기 때문에, 단계적으로 언급한다.

알고리즘의 적용 환경은 게임 어플리케이션으로 설정한다. Fig. 4는 입력기기를 통해서 매 프레임마다 데이터를 입력받는 것을 도시했다.

제스처 입력 과정은 초당 프레임 수마다 벡터를 입력받는다. 그러나 제스처를 입력 받는 과정에서 입력 벡터는 소실될 수 있다. 입력 벡터의 소실을 막기 위해서 Fig. 5는 입력받은 벡터 간의 보간을 통해서 새로운 벡터를 생성한다. Fig. 5는 이의 보간 과정을 도시했다.

입력 데이터는 3차원 벡터이며, 데이터를 입력받을 때마다 두 벡터 사이의 선형 보간(linear interpolation)을 수행한다. 보간을 통해서 추가 된 벡터는 제스처 인식 모델의 적합한 차원의 입력 데이터를 생성하기 위해서 입력받은 벡터를 소거하거나 보충하는 재표본화(resampling) 과정을 수행한다. 그 과정은 Fig. 6과 같다.

딥 러닝 기반의 제스처 인식 모델은 신경망을 토대로 깊은 구조를 형성한 것이다. 인식 모델의 입력 차원을 고려하는 것은 실질적으로 입력 데이터를 학습하고 데이터를 분류하는 데 있어서 중요한 과정이다. 크게 두 가지 과정이 선택적으로 수행된다. 수행 과정은 아래와 같다.

(1) 벡터를 소거하는 과정은 현재까지 입력받은 벡터의 수와 데이터 처리 전의 제스처 인식 모델이 요구하는 벡터 수를 나누어서 간격을 구한다. 도출된 간격은 입력받은 벡터 중 보존해야 할 대상을 선정하고 나머지 벡터는 소거하는 과정을 수행한다.

(2) 사용자의 요구에 따라 프레임마다 벡터를 입력받는다면 제스처 인식 모델이 요구하는 입력 차원을 획득하지 못하고 입력 데이터가 누락될 수 있다. 본 연구에서 제안한 정규화 방법은 획득하지 못한 입력 벡터를 보충하는 과정을 수행한다.

Fig. 6에서 소거 및 생성하는 과정을 거친 입력 벡터는 제스처 인식 모델의 적합한 입력 형태를 생성

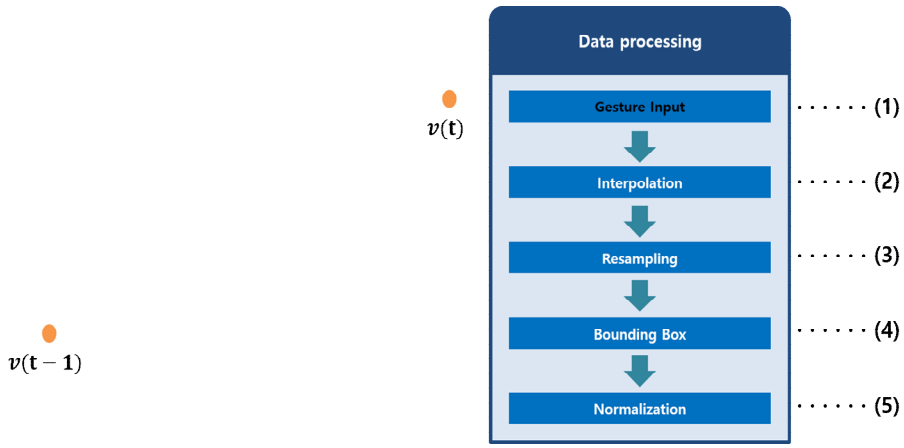


Fig. 4. Gesture input.

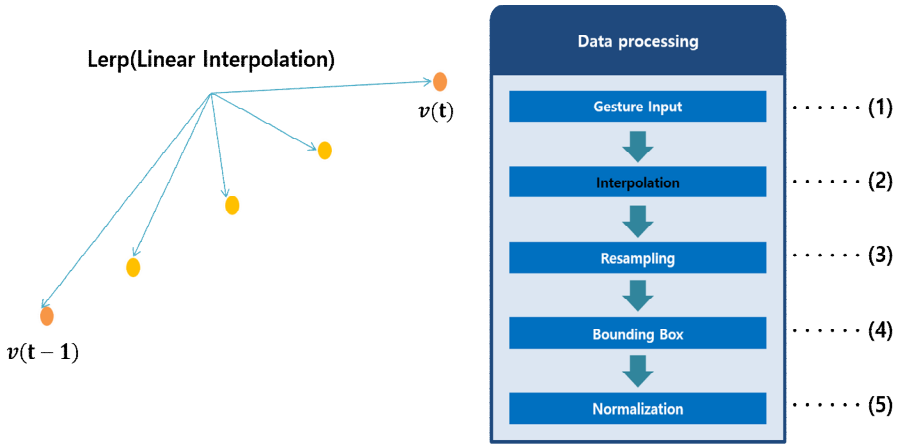


Fig. 5. Interpolation.

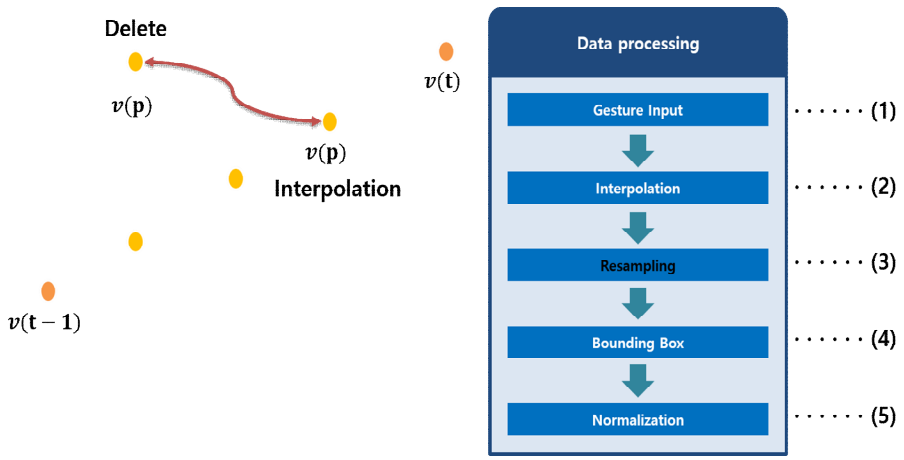


Fig. 6. Resampling.

하기 위해서 바운딩 박스(bounding box)를 생성하는 과정을 거친다. Fig. 7은 바운딩 박스 생성 과정을 도시했다.

바운딩 박스를 생성과정은 전체 입력 벡터로부터 가장 작은 원소를 추출한 벡터와 가장 큰 원소를 추출한 벡터를 생성한다. 생성된 벡터를 통해서 각 입력 벡터에 대해서 정규화 할 수 있다. 바운딩 박스 생성 후, 각 입력 벡터에 대한 정규화 과정은 Fig. 8과 같다.

정규화 과정에서는 모든 3차원 벡터가 0-1의 범위로 구성하기 위해서, 축의 값이 가장 작은 벡터와 큰 벡터를 도출하여 적절한 범위를 가지는 과정을 수행한다. 위의 과정을 정리하면 식 (1)과 같다.

$$
\begin{aligned}
v^i &= (v_x^i, v_y^i, v_z^i) \in R^3,\ i = 0, \cdots, n \\
v_{\min} &= (\min_i v_x^i,\ \min_i v_y^i,\ \min_i v_z^i) \\
v_{\max} &= (\max_i v_x^i,\ \max_i v_y^i,\ \max_i v_z^i) \\
v^k &= \frac{v_i - v_{\min}}{\|v_{\max} - v_{\min}\|_\infty}
\end{aligned}
\tag{1}
$$

입력 벡터를 v^i, 전체 입력 벡터로부터 가장 작은 원소를 추출한 벡터 $v_{\min}$, 전체 입력 벡터로부터 가장 큰 원소를 추출한 벡터 $v_{\max}$라 할 때, $v_{\min}$를 v^i의 차를 구한 후, 두 벡터 $v_{\max}$, $v_{\min}$의 차에 대한 최대 노름(max norm)을 통해서 원소 중 가장 큰 값이 도출된다. 도출된 값을 사용하여 $v_{\min}$를 v^i의 차에 나누는 과정을 수행하고 결과적으로 정규화된 벡터 v^k가 도출된다. 정규화 후, 데이터 생성은 3차원 위치 벡터

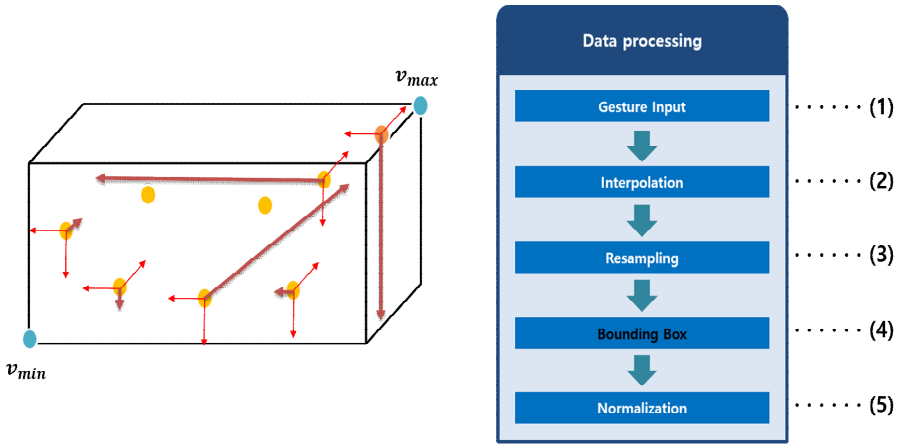


Fig. 7. Bounding box

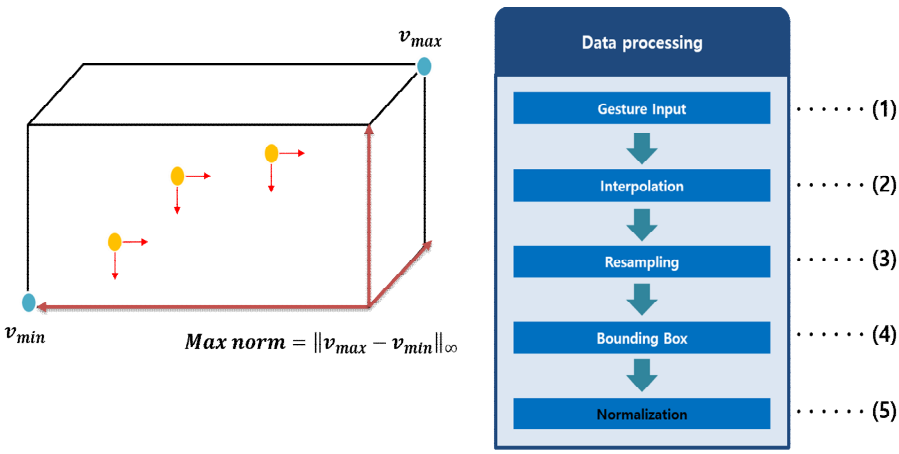


Fig. 8. Normalization.

x, y, z축의 값과 오른손, 왼손을 구분하는 파라미터 1개를 통해서 총 100차원의 벡터를 생성할 수 있다.

생성된 100차원의 벡터를 통해서 제안된 모델의 학습을 수행할 수 있다. 학습 과정에서도 본 연구에서 제안된 시공간 정규화 방법은 입력 벡터를 1차원 레이어로 구성하여 기존 2차원 영상을 입력한 연구와는 달리 계산비용을 줄이는데 그 의의를 가진다. 본 연구에서 제안하는 시공간 정규화 방법을 통해서 학습 과정의 계산 비용을 줄이면서 효율적으로 모델의 학습을 수행할 수 있다.

상기 과정을 통해서 시 공간적 정규화를 수행하고, 입력 벡터에 대한 범위를 규격화하여 딥 러닝 모델의 학습을 유연하게 유도할 수 있다. 유도한 과정은 앞서 언급한 총 5개의 과정을 수행한다. 수행한 과정을 통해서 딥 러닝 기반의 3D 제스처 인식 모델의 입력 벡터를 정규화할 수 있는 방법을 제안했다.

3.4 제안한 모델

본 연구에서는 신경망을 토대로 깊은 심층 구조를 형성하여 3차원 제스처 인식을 수행했다. 모델의 구조는 입력 계층, 은닉 계층, 출력 계층으로 간단한 구조를 가진다. 제안된 모델은 Fig 9와 같다. 첫 번째 입력 계층에서는 데이터의 차원을 고려하여 101차원을 구성했다. 구성된 차원과 입력 계층의 차원은 동일하므로 100차원의 노드를 구성하고 편향(Bias)를 포함하여 101차원을 구성했다. 두 번째 은닉 계층은 총 5개의 은닉 계층을 구성하여 심층 구조를 형성했다. 은닉 계층의 노드 개수는 Fig. 9에 표현된 것과

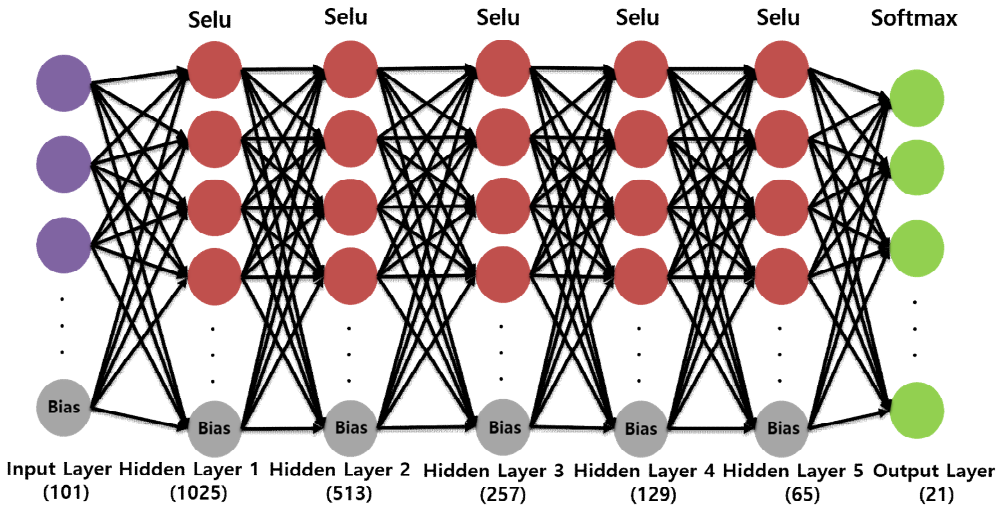


Fig. 9. Proposed model.

동일하다. 세 번째 출력 계층은 실제로 분류를 수행하는 개수와 동등해야 하므로 총 21개의 노드를 구성한다.

제안된 모델에서 은닉 계층의 활성화 함수는 SELU (Scaled Exponential Linear Unit)[20]을 활용했다. SELU의 식은 (2)과 같으며, 그래프로 나타냈을 때는 Fig. 10과 같다.

$$f(x)=\lambda\begin{cases}x & \text{if } x>0\\ \alpha e^{x}-\alpha & otherwise\end{cases} \tag{2}$$

SELU는 입력 값이 양수일 때는 λ을 곱해서 출력되고, 음수일 때는 λ을 곱하고 $\alpha e^{x}-\alpha$를 통 출력된다. 각각의 파라미터는 SELU를 제안한 논문의 설정된 값을 따른다. 과적합을 피하고 모델의 학습이 이루어지지 않는 현상을 방지하기 위해서 사용한다.

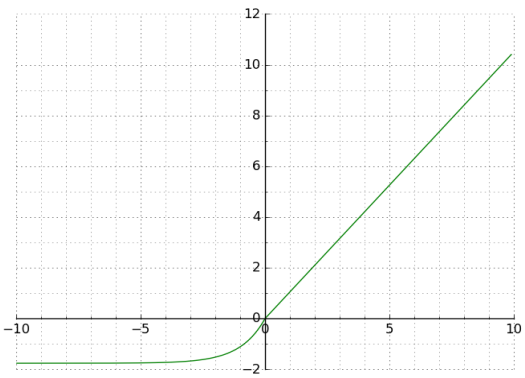


Fig. 10. SELU.

3.5 시스템 구성

본 연구의 딥러닝 모델 구현 및 실행을 위해 딥러닝 연산 구조와 실제 구현되는 모듈별 연동 관계를 Fig. 11에서 볼 수 있다. 11의 구조에서 (A-1) - (A-5)까지의 단계는 서버를 통해서 모델을 생성하는 단계이며, (B-1) - (B-5)까지의 단계는 서버를 통해서 모델을 활용하는 단계이다.

본 연구에서 구현된 모델은 실제 재활 환자를 위하여 활용될 게임과 연동되기 위해 구성된 환경으로서 재활치료를 위해 게임 등의 환경으로 연동하기 위해 무거운 딥러닝 연산구조를 분리하여 구성하였

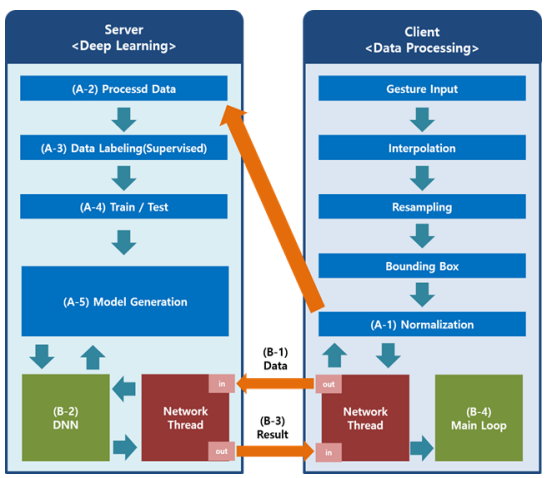


Fig. 11. Deep Learning Processing System for Gesture Recognition.

다. 딥 러닝에서 생성된 모델만을 활용할 시에는 구동에 무겁지 않겠으나, 본 연구의 결과가 적용되는 재활치료에서는 신규로 생성되는 환자의 데이터를 추가적으로 연산하여 지속적으로 모델 갱신이 필요하다.

최근의 딥러닝 연구는 연구 그 자체로서의 목적을 가지고 있기 때문에 게임 등에 대한 활용으로서의 응용은 고려되지 않은 바가 많다. 따라서 이러한 환경을 통한 실제 구현이 된 기존 연구가 미비하여 상세한 비교는 불가 하나, 향후 유사한 연구에서는 게임 등과 같이 무거운 시스템 구동에 딥 러닝을 함께 사용 할 때 반드시 필요한 과정이다.

실제 구현에 의해서 결과를 확인 할 수 있는 화면을 간단하게 아래 Fig. 12처럼 구성하였다. 실제 재활치료를 위한 게임에서는 사용자가 지루하지 않기 위한 그래픽적 요소를 구성하나, 해당 부분은 본 연구와 연관된 프로젝트에서 다른 연구기관의 연구범위임에 따라 결과를 출력하는 모듈 형태로 구성하였다.

4. 실험 결과

본 연구에서는 제안된 모델의 성능평가를 위해서 인식률과 학습 손실를 측정했다. 제안된 모델을 포함하여 총 5개의 경쟁 모델을 선정했다. 실험 환경은 Windows 10, RAM 16GB, NVIDIA GTX 1080에서 진행했다. 또한, 수집 인원 4명, 제스처 종류 21개, 총 10500개의 데이터를 통해서 학습 데이터(training set)는 6300개, 검증 데이터(validation set)는 2100개, 테스트 데이터(test set)는 2100개를 구축하였다.

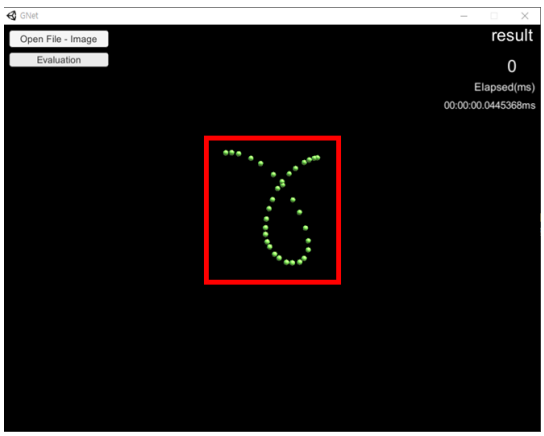


Fig. 12. Result of the System.

실험 결과는 Table 1과 같다.

실험 결과에서 제안된 모델은 99.3%로 가장 높은 인식률을 확인할 수 있다. 두 번째로 Logistic regression[21]은 인식률 86%로 가장 낮은 인식률을 보여주고 있다. 세 번째로 RNN[22]은 인식률 98.5%로 두 번째로 높은 인식률을 확인할 수 있다. 네 번째로 LSTM[23]은 인식률 91.7%로 비교적 높은 인식률을 보여주고 있다. 마지막으로 GRU[24]는 인식률 95.6%로 선정된 모델 중에서 평이한 인식률을 보여주고 있다.

제안된 모델과 경쟁 모델의 학습 성능을 측정하기 위해서 Fig. 13과 같이 각각의 경쟁 모델의 손실을 도시했다. 그리고 학습 세대(Epoch)는 70회로 한정했다.

제안된 모델의 학습 손실이 가장 빨리 떨어지는 것을 확인할 수 있다. RNN은 비교적 빨리 학습 손실이 감소를 보이나, 제안된 모델보다는 비우수한 결과를 보인다. 또한 Logistic regression은 점진적으로 학습 손실이 감소하는 것을 알 수 있다. 그리고 LSTM과 GRU는 Logistic regression보다 비교적 빠르게 감소하는 것을 확인할 수 있다. 따라서 상기 실험을 통해서 제안된 모델의 학습 성능과 인식률이 경쟁 모델에 비해서 우수한 것을 확인할 수 있다.

Table 1. Results of gesture classification

Model	Accuracy(%)
Proposed model	99.3
Logistic regression	86
RNN	98.5
LSTM	91.7
GRU	95.6

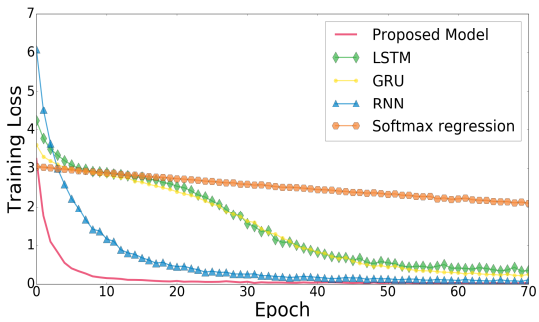


Fig. 13. Results of training loss.

5. 결 론

본 논문에서는 입력 벡터의 시 공간 정규화를 통해서 딥 러닝을 활용하여 3D 제스처를 인식했다. 총 21개의 제스처 종류를 구성하고 Unity3D와 립모션을 활용하여 제스처 데이터를 수집했다. 또한 3차원의 입력 벡터를 제안 모델에 학습하기 위해서 시 공간적 정규화를 적용하고, 제안된 모델은 신경망을 통해서 심층 구조를 갖는 DNN(Deep Neural Network)을 활용했다. 그리고 빠른 학습을 수행하고, 과적합을 방지하기 위해서 활성화 함수를 SELU를 적용했다. 제안된 모델의 인식률과 학습 성능을 측정하기 위해서 실험을 진행했으며, 제안된 모델은 99.3%의 인식률을 확인할 수 있다. 또한 기존의 알고리즘에 비해 높은 인식률을 보여주었으며, 학습 손실이 가장 빠르게 떨어지는 것을 확인 할 수 있다. 하지만 정규화 방법에 대한 추가적인 분석과 데이터 수집, 다양한 경쟁 모델에 대한 실험을 필요로 한다. 향후 연구에서는 이러한 사항을 개선할 수 있을 것으로 기대된다.

REFERENCE

[1] J.J. Chae, J.H. Lim, H.S. Kim, and J.J. Lee, "Study on Real-time Gesture Recognition Based on Convolutional Neural Network for Game Applications," *Journal of Korea Multimedia Society*, Vol. 20, No. 5, pp. 835-843, 2017.

[2] M.J. Kim, J. Heo, J.H. Kim, S.Y. Park, and J. Chang, "Development and Evaluation of Leapmotion-based Game Interface Considering Intuitive Hand Gestures," *Journal of The Korean Society for Computer Game*, Vol. 27, No. 4, pp. 69-75, 2014.

[3] J. Gu and D. Shin, "An Empirical Evaluation of the Representative Hand Gesture of Task from User Perspective," *Archives of Design Research*, Vol. 28, No. 1, pp. 133-145, 2015.

[4] K.S. Ham, D. Lee, H.J. Hong, S. Park, and J. Kim, "An Experimental Research on the Usability of Indirect Control Using Finger Gesture Interaction in Three Dimensional Space," *Journal of The Korea Contents Association*, Vol. 14, No. 11, pp. 519-532, 2014.

[5] L. Rokach and O. Maimon, "Top-down Induction of Decision Trees Classifiers-A Survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, Vol. 35, No. 4, pp. 476-487, 2005.

[6] L. Breiman, "Random Forests," *Machine Learning*, Vol. 45, pp. 5-32, 2001.

[7] C. Cortes and V. Vapnik, "Support-vector Networks," *Machine Learning*, Vol 20, No. 3, pp. 273-297, 1995.

[8] H.C. Yoon and J.S. Cho, "Hand Feature Extraction Algorithm Using Curvature Analysis For Recognition of Various Hand Gestures," *Journal of the Korea Society of Computer and Information*, Vol. 20, No. 5, pp. 13-20, 2015.

[9] D.H. Song, D.M. Kim, and C.W. Lee, "The Chinese Characters Learning Contents Based on Gesture Recognition Using HMM Algorithm," *Journal of Korea Multimedia Society*, Vol. 15, No. 8, pp. 1067-1074, 2012.

[10] A. Park, H. Jang, and K. Jung, "Fast and Efficient Implementation of Neural Networks using CUDA and OpenMP," *Journal of KISS : Software and Applications*, Vol. 36, No. 4, pp. 253-260, 2009.

[11] J. Schmidhuber, "Deep Learning in Neural Networks: An Overview," *Neural Networks*, Vol. 61, pp. 85-117, 2015.

[12] H.K. Lee and J.H. Kim, "An HMM-based Threshold Model Approach for Gesture Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 10, pp. 961-973, 1999.

[13] J. Suarez and R.R. Murphy, "Hand Gesture Recognition with Depth Images: A Review," *Proceeding of 2012 IEEE RO-MAN: The 21st IEEE International Symposium on Robot and Human Interactive Communication*, pp. 411-417, 2012.

[14] W. Xu and E.J. Lee, "Continuous Gesture

Recognition System Using Improved HMM Algorithm Based on 2D and 3D Space," *International Journal of Multimedia and Ubiquitous Engineering*, Vol. 7, No. 2, pp. 335-340, 2012.

[15] S. Cho, H. Byun, H.K. Lee, and J. Cha, "Hand Gesture Recognition from Kinect Sensor Data," *Journal of Broadcast Engineering*, Vol. 17, No. 3, pp. 447-458, 2012.

[16] J. Oh, T. Kim, and H. Hong, "Gesture Recognition Method Using Tree Classification and Multiclass SVM," *Journal of the Institute of Electronics and Information Engineers*, Vol. 50, No. 6, pp. 238-245, 2013.

[17] M. Jang, H. Choi, J. Kim, and S. Lee, "Neural Network-based Hand Gesture Recognition System," *Proceeding of Human Computer Interaction 2012*, pp. 676-681, 2012.

[18] D. Xu, "A Neural Network Approach for Hand Gesture Recognition in Virtual Reality Driving Training System of SPG," *Proceeding of The 18th International Conference on Pattern Recognition,* Vol. 3, pp. 519-522, 2006.

[19] C.S. Lee, S.Y. Chun, M.G. Sohn, and S.H. Lee, "Hand Gesture Interface Using Mobile Camera Devices," *Journal of the Korean Institute of Information Scientist and Engineers : Computing Practices and Letters*, Vol. 16, No. 5, pp. 621-625, 2010.

[20] Self-Normalizing Neural Networks, https://arxiv.org/abs/1706.02515 (accessed Jun., 10, 2017).

[21] D.R. Cox, "The Regression Analysis of Binary Sequences," *Journal of the Royal Statistical Society. Series B (Methodological)*, Vol. 20, No. 2, pp. 215-242, 1958.

[22] Recurrent Neural Network, https://en.wikipedia.org/wiki/Recurrent_neural_network (accessed Jun., 10, 2017).

[23] S. Hochreiter and J. Schmidhuber, "Long Short-term Memory," *Neural Computation*, Vol. 9, No. 8, pp. 1735-1780, 1997.

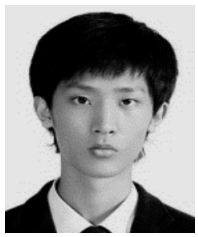
[24] K. Cho, B. Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk et al, "Learning Phrase Representations Using RNN Encoder-decoder for Statistical Machine Translation," *Proceeding of the 2014 Conference on Empirical Methods in Natural Language Processing*, pp. 1724-1734, 2014.

채 지 훈

2016년 계명대학교 컴퓨터공학부 게임모바일공학전공 졸업(학사)
2016년~2018년 계명대학교 컴퓨터공학부 졸업(석사)
관심분야 : 영상처리, 인공지능, 게임, 딥 러닝

강 수 명

2011년 2월 계명대학교 게임모바일콘텐츠학과 졸업(학사)
2013년 8월 계명대학교 미디어아트학과 게임모바일전공 졸업(석사)
2014년~2017년 ㈜지오씨엔아이 공간정보기술연구소 선임연구원
2016년 9월~현재 계명대학교 컴퓨터공학과 박사과정
관심분야 : 영상처리, 패턴인식, 컴퓨터비전, 딥 러닝



김 해 성

2018년 2월 계명대학교 컴퓨터공학부 게임모바일전공 졸업(학사)
2018년 3월~현재 계명대학교 컴퓨터공학과 석사과정
관심분야 : 그래픽스, 인공지능, 딥 러닝, 게임

이 준 재

1986년 경북대학교 전자공학과 졸업(학사)
1990년 경북대학교 전자공학과 졸업(석사)
1994년 경북대학교 전자공학과 졸업(박사)
1995년~2007년 동서대학교 컴퓨터 정보공학부 부교수
1998년~1999년 Georgia Institute of Technology 방문교수
2000년~2001년 (주)파미 연구소장
2007년~현재 계명대학교 컴퓨터공학부 교수
관심분야 : 영상처리, 3D 컴퓨터 비전, 딥 러닝, 게임