JDCS
디지털콘텐츠학회논문지
Journal of Digital Contents Society
Vol. 25, No. 2, pp. 529-534, Feb. 2024

사용자가 정의한 손 제스처 인식 기반의 윈도우 사용자 인터페이스 설계 및 구현

김 명 균[1] · 박 희 동[2*]
[1]나사렛대학교 IT인공지능학부 학사과정
[2*]나사렛대학교 IT인공지능학부 교수

Design and Implementation of a Windows User Interface Based on User-Defined Hand Gesture Recognition

Myung-Gyun Kim[1] · Hee-Dong Park[2*]

[1]Undergraduate, School of IT & Artificial Intelligence, Korea Nazarene University, Cheonan 31172, Korea
[2*]Professor, School of IT & Artificial Intelligence, Korea Nazarene University, Cheonan 31172, Korea

[요 약]

본 논문에서는 Google의 Mediapipie Hand Landmark 모델 및 순환신경망 (RNN)을 활용하여 사용자가 직접 정의한 손 제스처를 분류할 수 있는 제스처 분류 모델을 학습한 후, 인식된 제스처를 기반으로 마우스나 키보드와 같은 입력 장치 대신 데이터를 입력하고 컴퓨터를 제어할 수 있는 새로운 윈도우 사용자 인터페이스 시스템을 설계하고 구현하였다. 제안 시스템은 제스처 인식 모듈과 윈도우 제어 모듈로 구성되는데, 제스처 인식 모듈에 의해서 추론된 제스처 신호를 윈도우 제어 모듈에 전송하여 사전에 정의된 키보드나 마우스 입력을 수행한다. 제스처 분류 모델의 성능 평가 결과 학습한 제스처를 적절하게 분류하는 것을 확인할 수 있었으며, 제안 시스템을 윈도우 운영체제에 적용하여 사용자가 정의한 제스처를 사용하여 데이터를 입력하고 컴퓨터를 제어할 수 있음을 확인할 수 있었다.

[Abstract]

In this study, we propose a Windows user interface system enabling users to input data and control the computer through self-defined hand gestures. The system is based on a gesture classification model that is trained using Google's Mediapipe Hand Landmark model and a recurrent neural network. The proposed system consists of gesture recognition and Window control modules. The gesture signals inferred by the gesture recognition module are transmitted to the Window control module to perform predefined keyboard or mouse inputs. The gesture classification model is evaluated can correctly classify the learned gestures. The proposed system is applied to the Windows operating system, and it is confirmed that users can input data and control the computer using the self-defined gestures.

색인어 : Mediapipe Hand Landmark 모델, 순환신경망, 사용자 정의, 손 제스처 인식, 윈도우 사용자 인터페이스

Keyword : Mediapipe Hand Landmark Model, Recurrent Neural Network, User-Defined, Hand Gesture Recognition, Windows User Interface

http://dx.doi.org/10.9728/dcs.2024.25.2.529

This is an Open Access article distributed under the terms of the Creative Commons Attribution Non-CommercialLicense(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 26 January 2024; **Revised** 07 February 2024
Accepted 15 February 2024

***Corresponding Author; Hee-Dong Park**

Tel: +82-41-570-7954
E-mail: hdpark@kornu.ac.kr

Ⅰ. 서 론

최근 다양한 인간-컴퓨터 상호작용(HCI; human computer interaction)이 강조되면서, 키보드나 마우스와 같은 기존의 전통적인 입력 장치 외에 음성이나 제스처 인식 등을 이용한 데이터 입력 및 컴퓨터 제어에 대한 요구가 크게 늘어나고 있다[1]-[3]. 이와 같은 음성이나 제스처 인식 기반의 사용자 인터페이스 (UI; user interface)는 무선 통신을 사용할 수 없는 경우나 키보드나 마우스를 사용할 수 없는 장애인 사용자를 위해 유용한 인터페이스 기능을 제공할 수 있다[4]-[6].

이에 본 연구에서는 Google에서 제공하는 Mediapipe Hand Landmark 모델 및 순환 신경망(RNN; recurrent neural network)을 활용하여 사용자가 직접 정의한 손 제스처 분류 모델을 학습하고, 학습한 모델을 통해 인식한 제스처를 기반으로 마우스와 키보드와 같은 데이터 입력 장치를 대체할 수 있는 새로운 윈도우 사용자 인터페이스 시스템을 제안한다. 제안 시스템은 제스처 인식 모듈과 윈도우 제어 모듈로 구성되는데, 제스처 인식 모듈에 의해서 추론된 제스처 신호를 윈도우 제어 모듈에 전송하여 사전에 정의된 키보드나 마우스 입력을 수행한다. 특히 제안된 손 제스처 인식 기반의 사용자 인터페이스는 사용자가 직접 제스처 및 그에 대응하는 키브드 및 마우스 입력을 정의할 수 있어, 일반 사용자뿐만 아니라 키보드나 마우스를 잘 사용할 수 없는 장애인에게도 편리한 사용자 인터페이스를 제공할 수 있다. 제스처 분류 모델의 성능 평가 결과 학습한 제스처를 적절하게 분류하는 것을 확인할 수 있었으며, 제안 시스템을 윈도우 운영체제에 적용하여 사용자가 정의한 제스처를 사용하여 데이터를 입력하고 컴퓨터를 제어할 수 있음을 확인할 수 있었다.

Ⅱ. 관련 연구

2-1 Mediapipe Hand Landmark 모델

Google 사에서 제공하는 Mediapipe는 시계열 데이터를 처리하는 파이프라인을 통해 데이터를 실시간으로 처리할 수 있는 크로스 플랫폼 프레임워크다[7]. Android, iOS, Windows, Linux 등 운영체제에서 동작하고, C++, Python, Javascript 등의 언어를 지원하므로 활용도가 높다. Mediapipe에서 제공하는 Hand Landmark 모델은 손 영역을 검출하는 BlazePalm과 랜드마크를 검출하는 BlazeHand를 통해 손이 포함된 이미지에서 주요 랜드마크 위치를 추론하는 작업을 수행할 수 있다[8]-[11]. 각 랜드마크는 손의 관절 및 손가락 끝과 같은 특정 부분을 나타내며, 예측 값은 한 손마다 21개의 3차원 좌표 형태를 갖는다.

본 논문에서는 Mediapipe Hand Landmark 모델이 제공하는 랜드마크 데이터를 사용하여 손 제스처의 특징을 추출하고, 제스처 분류 모델을 학습하는데 활용한다. 그림 1은 Mediapipe Hand Landmark 모델이 추론하는 손의 랜드마크 노드를 나타낸다.

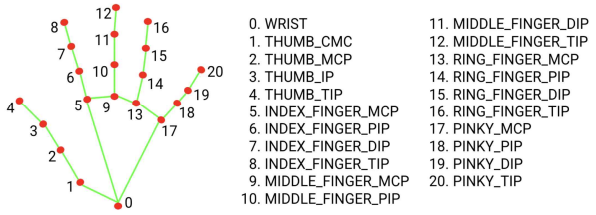

그림 1. 구글 Mediapipe Hand Landmark 노드
Fig. 1. Google Mediapipe Hand Landmark nodes

2-2 LSTM

LSTM (long short-term memory) 신경망은 RNN의 한 종류로, 주로 시계열 데이터 처리나 자연어 처리에 사용된다. RNN은 모든 입력이 서로 연관된 입력 시퀀스들을 처리하도록 설계되었으며, 시퀀스 길이와 관계없이 입력과 출력을 받아들일 수 있는 네트워크 구조이다. 그러나 RNN은 입력 데이터 길이가 늘어남에 따라 관련 정보와 그 정보를 사용하려는 출력 지점의 거리가 멀어질수록 역전파되는 오류가 점차 줄어들어 학습에 거의 영향을 미치지 못하는 그래디언트 소멸 문제로 인해 5~10개 이상의 타임 스텝을 연결할 수 없다[12]. LSTM은 이 문제를 해결하기 위해 고안된 신경망으로, 이전 타임 스텝의 기억을 보존한다[13],[14]. 그림 2는 입력 게이트, 출력 게이트, 망각 게이트와 메모리 셀로 구성된 LSTM 블록 구조를 보여준다. LSTM은 이러한 블록 구조를 통해 시간에 따라 변하는 제스처의 특징을 학습하여 동적 제스처 분류를 수행할 수 있다.

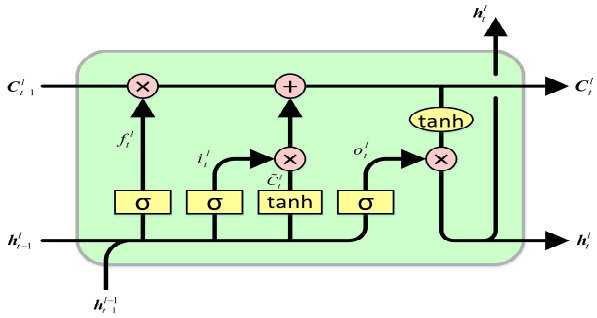

그림 2. LSTM 블록 구조
Fig. 2. LSTM block architecture

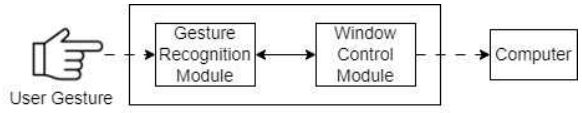

그림 3. 시스템 구성
Fig. 3. System architecture

Ⅲ. 제스처 인식 기반의 윈도우 사용자 인터페이스

3-1 시스템 구성

제안하는 시스템은 그림 3에 나타난 바와 같이 두 개의 모듈로 구성되어 있다. 제스처 인식 모듈(Gesture Recognition Module)은 Mediapipe Hand Landmark 모델 및 제스처 분류 모델(Gesture Classification Model)을 사용하여 사용자가 정의한 손 제스처를 인식한다. 윈도우 제어 모듈(Window Control Module)은 제스처 인식 모듈이 인식한 제스처 결과를 기반으로 컴퓨터에 제어 신호를 전송한다.

3-2 제스처 인식 모듈의 동작 흐름

그림 4는 제스처 인식 모듈의 동작 흐름을 나타낸다. 랜드마크 추론(Landmark Inference) 단계는 Mediapipe Hand Landmark 모델을 사용하여 입력된 이미지에서 손의 랜드마크를 추론하며, 특징 추출(Feature Extraction) 단계는 추론된 랜드마크 데이터에서 특징을 추출한다. 그리고 추론 데이터 생성(Create Inference Data) 단계는 추출한 특징을 연속적으로 배열에 저장하여 배열 형태의 시계열 데이터를 생성하며, 제스처 추론(Gesture Inference) 단계는 생성한 시계열 데이터를 제스처 분류 모델에 입력하여 제스처를 추론한다.

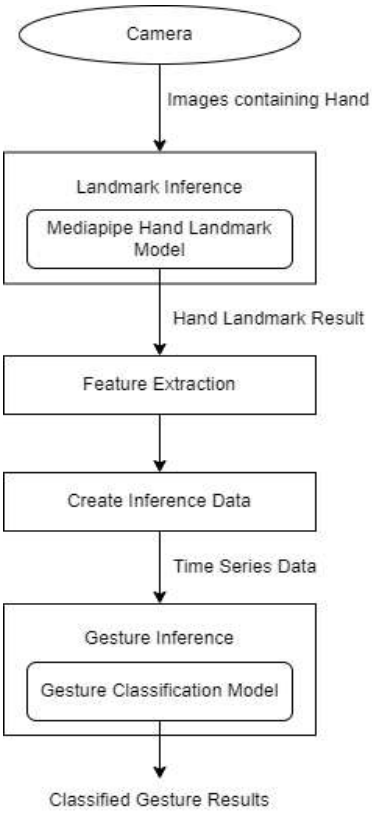

그림 4. 제스처 인식 모듈의 동작 과정
Fig. 4. The operation process of gesture recognition module

3-3 제스처 분류 모델 학습

제안 시스템에서 사용자는 자신의 손 제스처 및 그것에 해당하는 사용자 인터페이스 기능을 직접 정의하여 학습시킬 수 있다. 그림 5는 사용자가 정의한 제스처의 예시를 보여준다. 각 이미지는 사용자가 반복적으로 취할 제스처와 해당 학습 데이터의 윈도우 제어 동작을 나타내는데, 키보드 또는 마우스 입력을 대신하기 위해 복사(Copy), 마우스 좌 클릭(Mouse Left Click), 마우스 우 클릭(Mouse Right Click), 탭 전환(Switch Tabs), 전체 선택(Select All), 붙여 넣기(Paste) 등과 같은 기능을 정의할 수 있다.

그림 6은 제스처 분류 모델의 학습 데이터로 사용하기 위해 추출한 특징의 종류를 나타낸다. 학습 데이터로 사용할 데이터는 Mediapipe Hand Landmark 모델로 추론된 데이터에서 3가지 데이터 특징을 추출하여 그 특징의 조합을 활용한다. 첫 번째 특징은 Mediapipe Hand Landmark 모델 추론을 통해 얻은 랜드마크 데이터를 1차원 형태의 배열로 변환한 데이터 특징이고, 두 번째 특징은 첫 번째 특징 데이터에서 랜드마크 간의 각도를 구한 데이터 특징으로, 그림 상에서 초록색으로 연결된 랜드마크 간의 각도를 구했다. 세 번째 특징은 사용자 손바닥의 정중앙 좌표를 구한 데이터 특징으로, 그림 상에서 초록색 점의 좌표를 구했다. 특징들을 조합한 학습 데이터에 슬라이딩 윈도우 알고리즘을 적용하여 윈도우 사이즈가 각각 20, 25, 30, 35인 데이터의 집합을 생성하여 학습 데이터로 사용한다. 이 때 생성되는 학습 데이터의 크기는 윈도우 크기(W)와 데이터 길이(L)에 대해서 $W \times (L-W-1)$ 행렬의 데이터가 생성된다.

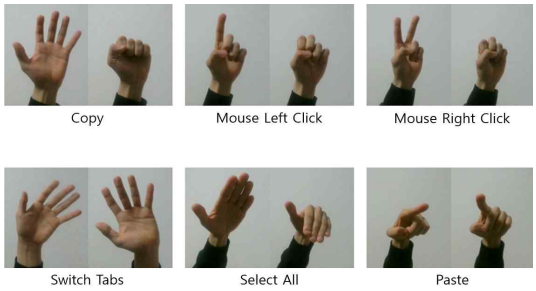

그림 5. 손 제스처 정의 예시
Fig. 5. Examples of hand gesture definitions

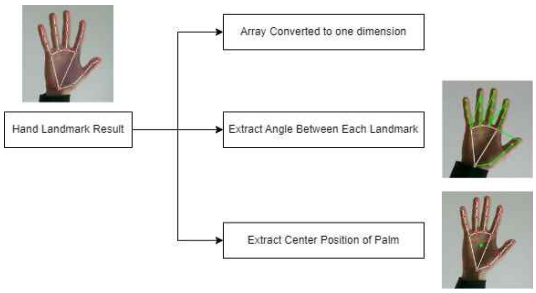

그림 6. 추출할 데이터 특징 종류
Fig. 6. Types of data features to extract

3-4 윈도우 제어 모듈의 동작 과정

그림 7은 윈도우 제어 모듈의 동작 과정을 나타낸다. 해당 모듈은 세 가지 기능을 수행한다. 첫 번째는 제스처와 컴퓨터 제어 동작 간에 매핑 테이블을 관리한다. 두 번째는 제스처 인식 모듈의 제스처 인식 결과를 바탕으로 미리 정의된 사용자 인터페이스 기능에 따라 마우스 클릭 및 키보드 입력을 수행한다. 세 번째는 Mediapipe Hand Landmark 모델의 추론 결과 중 특정 랜드마크 좌표를 기반으로 마우스 포인터 좌표를 결정하고 마우스 이동을 제어한다. 마우스 이동 제어를 위해 사용할 특정 랜드마크는 사용자가 직접 설정할 수 있다. 이 때 마우스 포인터의 좌표는 수식 (1)을 사용하여 결정된다.

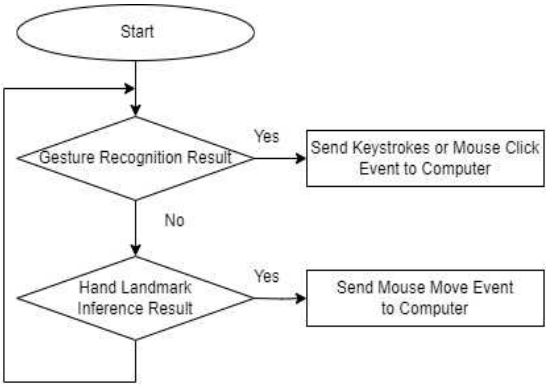

그림 7. 윈도우 제어 모듈의 동작 과정
Fig. 7. The operation process of window control module

$$X_{cursor} = W_{screen} \times X_{land}$$
$$Y_{cursor} = H_{screen} \times Y_{land} \quad (1)$$

웹캠이 출력하는 이미지 크기에 대해 0에서 1 사이의 값으로 정규화된 손 랜드마크 좌표 (X_{land}, Y_{land}) 및 사용자 모니터 크기 (W_{screen}, H_{screen})를 사용하여 모니터 상의 마우스 커서 좌표 (X_{cursor}, Y_{cursor})를 계산한다.

Ⅳ. 성능 평가 및 구현

4-1 제스처 분류 모델 성능 평가

본 논문에서 제안된 제스처 분류 모델의 성능을 평가하기 위하여 모델에 입력되는 데이터의 길이 및 데이터 특징의 조합에 따른 평균 인식률을 측정하였다. 모델 학습 및 평가에 사용할 데이터는 그림 5에서와 같이 정의된 제스처에 대해 촬영한 10,791프레임에서 3가지 데이터 특징을 추출하고 조합하여 데이터를 생성하였다. 그리고 생성한 데이터에 슬라이딩 윈도우 알고리즘을 적용하여 각각 20, 25, 30, 35의 윈도우 크기를 설정하고 전체 데이터에서 윈도우 크기를 갖는 데이터의 연속적인 집합을 생성하여 80%는 학습 데이터로, 20%는 평가 데이터로 사용했다.

표 1은 제스처 분류 모델에 입력되는 시계열 데이터의 길이에 따른 평균 인식률 결과를 보여준다. 입력 데이터의 시퀀스 길이가 20, 25, 30, 35인 모델의 평균 인식률은 각각 0.9719, 0.9753, 0.9614, 0.9701이었다. 즉 입력 데이터의 길이는 평균 인식률에 큰 영향을 미치지 않는 것으로 보인다. 그리고 표 2는 입력 시퀀스의 길이가 25인 모델에서 입력 데이터를 구성하는 특징을 다르게 조합하여 학습한 모델의 평균 인식률 측정 결과를 나타낸다. 특징 1, 2, 3을 조합했을 경우의 평균 인식률은 0.9586이었으며, 특징 1 및 2를 조합했을 경우의 평균 인식률은 0.9357이었다. 그런데 특징 2 및 3을 조합한 입력 데이터를 학습한 모델은 다른 모델과 비교했을 때 매우 적은 데이터로도 0.9783이라는 높은 인식률을 나타내었다.

특징들의 조합에 따른 인식률 결과 중 가장 높은 인식률을 보인 모델과 다른 모델의 차이점은 입력 데이터에 특징 1 포함 여부가 있다. 특징 1이 포함된 데이터를 학습한 모델의 경우, 입력 데이터에 포함된 랜드마크 좌표 데이터와 모델이 학습한 랜드마크 좌표 데이터의 차이가 제스처 인식률에 영향을 미치는 것으로 보인다. 이는 학습 데이터를 생성한 환경과 실제 인식 환경에서 웹캠이 인식한 손의 위치가 다를 경우 인식률이 변화할 수 있음을 의미한다. 이에 반해 특징 2를 입력 데이터로 사용한 모델의 경우 랜드마크 간의 상대적인 각도를 사용하므로 손의 위치가 제스처 인식률에 미치는 영향이 적어, 비교적 높은 인식률을 나타내는 것으로 보인다.

표 1. 입력 데이터 길이에 따른 평균 인식률
Table 1. Average recognition rate according to the length of input data sequence

Length of sequence	Average recognition rate
20	0.9719
25	0.9753
30	0.9614
35	0.9701

표 2. 데이터 특징 조합에 따른 평균 인식률 (입력 시퀀스 길이가 25일 경우)
Table 2. Average recognition rate according to the combination of data features (when the length of input sequence is 25)

Combination	Data size per frame	Averate recognition rate
1, 2, 3	80	0.9586
1, 2	77	0.9357
2, 3	14	0.9783

4-2 시스템 구현

제안 시스템에서는 사용자가 직접 컴퓨터 제어를 위해 사용할 손 제스처를 정의한 후 카메라를 통해 학습시킬 수 있다.

그림 8은 제스처 분류 모델의 학습 데이터 수집 및 시스템 구현에 사용한 웹캠 장비로서, FHD 60 FPS를 지원하는 카메라이다. 그림 9는 구현한 시스템에서 사용자가 컴퓨터를 제어하는 상황을 나타낸 그림이다. 시스템은 웹캠을 통해 사용자가 취하는 제스처를 인식하고, 모니터 상에 표시되는 웹캠 화면에 인식 결과를 출력하고 제어를 수행한다.

그림 8. FHD 60FPS 지원 웹캠 장비
Fig. 8. FHD 60FPS support webcam equipment

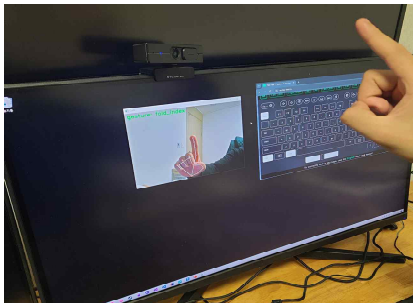

그림 9. 손 제스처를 사용한 컴퓨터 제어
Fig. 9. Computer control using hand gestures

V. 결 론

본 연구에서는 Google의 Mediapipie Hand Landmark 모델 및 RNN을 활용하여 사용자가 직접 정의한 손 제스처를 분류할 수 있는 제스처 분류 모델을 학습하고, 인식된 제스처를 기반으로 마우스나 키보드와 같은 입력 장치 대신 데이터를 입력하고 컴퓨터를 제어할 수 있는 새로운 윈도우 사용자 인터페이스 시스템을 설계하고 구현하였다. 제안 시스템은 제스처 인식 모듈과 윈도우 제어 모듈로 구성되는데, 제스처 인식 모듈에 의해서 추론된 제스처 신호를 윈도우 제어 모듈에 전송하여 사전에 정의된 키보드나 마우스 입력을 수행한다. 특히 제스처 인식 모듈은 기본적인 Mediapipe Hand Landmark 모델에 의한 제스처 인식뿐만 아니라 카메라가 챕처한 손 제스처 이미지들로부터 3가지 데이터 특징들을 추출하고 조합하여 학습 데이터를 생성한다.

제스처 분류 모델의 성능 평가 결과 학습한 제스처를 적절하게 분류하는 것을 확인할 수 있었다. 또한 제안 시스템을 윈도우 운영체제에 적용하여, 사용자가 정의한 제스처를 사용하여 데이터를 입력하고 컴퓨터를 제어할 수 있음을 확인할 수 있었다. 제안된 손 제스처 인식 기반의 사용자 인터페이스는 무선 통신을 사용할 수 없는 경우나 키보드나 마우스를 사용할 수 없는 장애인 사용자를 위해 유용한 인터페이스 기능을 제공할 수 있을 것으로 기대된다. 다만, 제안 시스템은 한 손 제스처를 이용한 사용자 인터페이스로, 복수의 제어 동작을 동시에 수행하기는 어렵다. 따라서 향후 연구에서는 양 손 및 몸 전체의 제스처를 인식하여 복수의 제어 동작을 동시에 수행할 수 있는 인터페이스를 연구할 계획이다.

감사의 글

본 연구성과물은 2023년도 정부(교육부)의 재원으로 한국연구재단의 지원을 받아 수행된 기초연구사업임(No. 2017R1D1A3B06035024). 본 연구는 2023년도 나사렛대학교 교내연구비 지원으로 이루어졌음.

참고문헌

[1] H. Kaur and J. Rani, “A Review: Study of Various Techniques of Hand Gesture Recognition,” in *Proceedings of 2016 IEEE 1st International Conference on Power Electronics, Intelligent Control and Energy Systems (ICPEICES)*, Delhi, India, pp. 1-5, July 2016. https://doi.org/10.1109/icpeices.2016.7853514

[2] T.-H. Tsai, C.-C. Huang, and K.-L. Zhang, “Design of Hand Gesture Recognition System for Human-Computer Interaction,” *Multimedia Tools and Applications,* Vol. 79, No. 9-10, pp. 5989-6007, March 2020. https://doi.org/10.1007/s11042-019-08274-w

[3] M. Kim, J. Won, and H. Park, “A Study on Computer Control Based on User Gesture Recognition,” in *Proceedings of 2023 Korea Digital Contents Society Summer Conference*, Jeju, pp. 204-205, July 2023.

[4] C. Kerdvibulvech, “A Review of Computer-Based Gesture Interaction Methods for Supporting Disabled People with Special Needs,” in *Proceedings of the 15th International Conference on Computers Helping People with Special Needs,* Linz, Austria, pp. 503-506, July 2016. https://doi.org/10.1007/978-3-319-41267-2_70

[5] O. M. Szczepaniak and D. J. Sawicki, “Gesture Controlled Human-Computer Interface for the Disabled,” *Medycyna Pracy: Worker’s Health and Safety*, Vol. 68, No. 1, pp. 11-21, 2017. https://doi.org/10.13075/mp.5893.00529

[6] G. W. Gao and X. Y. Duan, “An Overview of Human-Computer Interaction Based on the Camera for Disabled People,” *Advanced Materials Research*, Vol. 219-220, pp. 1317-1320, March 2011. https://doi.org/10.4028/www.scientific.net/amr.219-220.1317

[7] C. Lugaresi, J. Tang, H. Nash, C. McClanahan, E. Uboweja, M. Hays, ... and M. Grundmann, “MediaPipe: A Framework for Building Perception Pipelines,” arXiv:1906.08172, June 2019. https://doi.org/10.48550/arXiv.1906.08172

[8] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, “MediaPipe Hands: On-Device Real-Time Hand Tracking,” arXiv:2006.10214, June 2020. https://doi.org/10.48550/arXiv.2006.10214

[9] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “SSD: Single Shot MultiBox Detector,” in *Proceedings of the 14th European Conference on Computer Vision (ECCV 2016),* Amsterdam, Netherlands, pp. 21-37, October 2016. https://doi.org/10.1007/978-3-319-46448-0_2

[10] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, “Feature Pyramid Networks for Object Detection,” in *Proceedings of 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu: HI, pp. 936-944, July 2017. https://doi.org/10.1109/CVPR.2017.106

[11] Google for Developers. MediaPipe [Internet]. Available: https://developers.google.com/mediapipe/.

[12] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM - A Tutorial into Long Short-Term Memory Recurrent Neural Networks,” arXiv:1909.09586, September 2019. https://doi.org/10.48550/arXiv.1909.09586

[13] Y. Yu, X. Si, C. Hu, and J. Zhang, “A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures,” *Neural Computation*, Vol. 31, No. 7, pp. 1235-1270, July 2019. https://doi.org/10.1162/neco_a_01199

[14] Colah’s Blog. Understanding LSTM Networks [Internet]. Available: https://colah.github.io/posts/2015-08-Understanding-LSTMs/.

김명균(Myung-Gyun Kim)

2019년～현 재: 나사렛대학교 IT인공지능학부 (학사 과정)

※관심분야 : 컴퓨터 비전(Computer Vision), 인간 컴퓨터 상호작용(HCI), 딥러닝(Deep Learning) 등

박희동(Hee-Dong Park)

1993년 : 경북대학교 전자공학과 (공학사)

1998년 : 경북대학교 일반대학원 전자공학과 (공학석사)

2005년 : 경북대학교 일반대학원 전자공학과 (공학박사)

1998년～2007년: 포항대학교 컴퓨터응용계열 교수

2007년～현 재: 나사렛대학교 IT인공지능학부 교수

※관심분야 : 모바일 컴퓨팅, 사물인터넷 (IoT), 인공지능(Artificial Intelligence) 등