

인공지능 기반 손 제스처 인식 정보를 활용한 지능형 인터페이스

¹ 조항준, ² 유준우, ³ 김은수, ^{4*} 이영재

Intelligent interface using hand gestures recognition based on artificial intelligence

¹Hangjun Cho, ²Junwoo Yoo, ³Eun Soo Kim and ^{4*}Young Jae Lee

요약

인공지능에 기반한 손 제스처 인식 정보를 활용한 지능형 인터페이스 알고리즘을 제안한다. 이 방법은 기능적으로 사용자 손 제스처의 추적 및 인식을 미디어파이프와 KNN, LSTM, CNN의 인공지능 기법을 사용해 다양한 동작을 빠르고 지능적으로 인식되는 인터페이스이다. 제안한 알고리즘 성능 평가를 위해 자체 제작한 2D 탑뷰 레이싱 게임과 로봇제어에 적용한다. 알고리즘 적용 결과 게임의 가상 객체의 다양한 움직임을 세밀하고 강건하게 제어할 수 있었으며, 실세계의 로봇 제어에 적용한 결과 이동과 정지, 좌회전, 우회전 등의 제어가 가능하였다. 또한 게임의 메인 캐릭터와 실세계 로봇을 동시에 제어하여 가상과 현실의 공존공간 상황 제어를 위한 지능형 인터페이스로 최적화된 동작도 구현하였다. 제안한 알고리즘은 신체를 활용한 자연스럽게 직관적 특성과 손가락의 미세한 움직임 인식에 따른 정교한 제어가 가능하며, 빠른 기간 내에 숙련되는 장점이 있어 지능형 사용자 인터페이스 개발을 위한 기본자료로 활용될 수 있다.

Abstract

We propose an intelligent interface algorithm using hand gesture recognition information based on artificial intelligence. This method is functionally an interface that recognizes various motions quickly and intelligently by using MediaPipe and artificial intelligence techniques such as KNN, LSTM, and CNN to track and recognize user hand gestures. To evaluate the performance of the proposed algorithm, it is applied to a self-made 2D top-view racing game and robot control. As a result of applying the algorithm, it was possible to control various movements of the virtual object in the game in detail and robustly. And the result of applying the algorithm to the robot control in the real world, it was possible to control movement, stop, left turn, and right turn. In addition, by controlling the main character of the game and the robot in the real world at the same time, the optimized motion was implemented as an intelligent interface for controlling the coexistence space of virtual and real world. The proposed algorithm enables sophisticated control according to natural and intuitive characteristics using the body and fine movement recognition of fingers, and has the advantage of being skilled in a short period of time, so it can be used as basic data for developing intelligent user interfaces.

Keywords: intelligent interface, artificial intelligence, hand gesture, MediaPipe, recognition

¹ 전주대학교 스마트미디어학과 졸업(joohz1089@naver.com)

² 전주대학교 스마트미디어학과 졸업(yjo428@naver.com)

³ 전주대학교 전기전자공학과 교수(eskim@ji.ac.kr)

^{4*}교신저자 전주대학교 스마트미디어학과 교수(leeyj@ji.ac.kr)

I. 서론

최근 인공지능, 빅데이터, 가상현실, 증강현실 등의 4 차 산업혁명 기술의 발전과 사람들의 문화·예술에 대한 수요가 높아짐에 미디어 콘텐츠와 인터페이스에 대한 관심이 고조되고 있다[1][2][3][4][5]. 특히 컴퓨터 기술의 급속한 발전에 따라 지능형 인터페이스 기술은 일상생활에서 점차 중요한 역할을 수행하고 있다. 지능형 인터페이스 기술은 사용자의 요구 사항들을 인공지능 기법을 사용하여 인지하고 다양한 미디어와 상호작용(interaction)을 구현하는 기술이다. 손동작은 인간의 동작 행위에 관해 중요한 많은 정보들을 포함하고 있기 때문에 자연스러운 상호작용을 위한 혁신적인 인터페이스로 자리매김하고 있다[3-5]. 특히 인터넷과 몇몇 센서를 사용한 단순 결합 서비스에서 다양한 사용자 중심의 첨단 인지 기법 및 친화적인 기능이 더해져 사용자 중심의 자연스럽고 편안한 인공지능 기반의 인터페이스에 대한 상용화 연구 또한 활발하게 진행되고 있다[6][7][8][9]. 그러나 사용자가 콘텐츠와 상호작용을 하는 과정에서 나타나는 특정한 행동 패턴을 시스템화하여 기존 콘텐츠 사용의 문제점들을 해결하는 연구는 부족한 부분이 많다. 이 같은 부분을 인공지능 기법으로 해결하고, 사용자 중심의 양질의 만족도 높은 콘텐츠를 제공을 위해 필수적인 인터페이스 개발이 필요하다. 인터랙티브(interactive) 미디어 콘텐츠 환경에서 몰입감을 높이고 자유로운 상호작용을 제공하기 위한 가장 자연스러운 방법은 인간의 신체를 이용한 제스처 인터페이스를 제공하는 것이다. 특히 손과 손가락의 경우 왼손과 오른손 및 각각의 손가락을 이용할 수 있으므로 다양하고 복잡한 정보까지도 세세하게 표현할 수 있다. 그러나 손, 손가락 인식, 연속적인 움직임 등의 손의 제스처 인식에 관한 기존의 연구들은 특화된 센서나 장비를 요구하거나 낮은 인식률 등의 문제점을 가지고 있다[8][9][10].

본 연구에서는 사용자 손 제스처의 추적 및 인식을 미디어파이프(MediaPipe)[11]와 인공지능을 사용한 지능형 인터페이스 알고리즘을 제안한다. 이 알고리즘은 증강현실, 가상현실, 게임 콘텐츠와 로봇 제어 등 직관적이고 세밀한 움직임이 요구되는 지능적인 분야에 적용 가능하다.

II. 핸드 제스처 인터페이스

손짓, 신체 자세 및 표정을 통한 의사 소통을 포함하는 비언어적 의사 소통은 인간과 인간 간의 모든 의사 소통의 약 2/3 를 차지한다. 손 제스처는 의사 소통 및 상호 작용에 사용되는 가장 일반적인 신체 언어 범주 중 하나로 일반적인 감정정보 뿐 아니라 특정 언어적 내용을 표현할 수 있다. 손 제스처를 분류하는 방법[12][13]에는 관찰 가능한 특징(feature)을 기반으로 하거나 해석(interpretation)을 기반으로 분류하는 등 여러 방법이 있다. 이중 특징 기반의 범주에서 제스처는 시간적 관계에 따라 그림 1 과 같이 두 가지 유형 즉, 정적 제스처(static gesture)와 및 동적 제스처(dynamic gesture)로 구분된다.

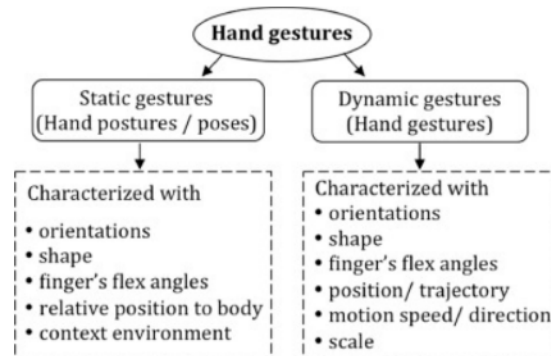


Figure 1. Hand gesture classification method example

그림 1. 손 제스처 분류 방법 예

정적 손 제스처(손 자세/손 포즈)는 제스처 기간 동안 손 위치가 변경되지 않는 제스처로 주로 손가락의 모양과 굴곡 각도 정보에 의존한다. 동적 손 제스처에서 손 위치는 시간과 관련하여 지속적으로 변경되며 일반적으로 준비, 스트로크 및 수축의 세 가지 동작 단계가 있으며 모양과 손가락의 각도 외에도 손 궤적과 방향 정보에 의존한다[14][15][16]. 제안한 알고리즘은 우리 일상에서의 많이 사용되는 정적 제스처와 동적 제스처를 사용하며 손과 손가락 특징점 추출은 미디어파이프(MediaPipe)를 사용한다. 미디어파이프는 라이브 및 스트리밍 미디어를 위한 사용자 지정 기계 학습 솔루션을 빌드하기 위한 플랫폼 간 파이프라인 프레임 워크이다. 이 프레임 워크는 구글에서 오픈 소스로 제공했으며, 비디오 또는 오디오와 같은 데이터에 대해 컴퓨터 비전 추론 수행을 위한 파이프라인 구축이 가능하다. 이 프레임 워크는 AI 모델을 사용하여 신속한 프로토타이핑에도 사용될 수 있으며, 컴퓨터 비전 응용프로그램을 다양한 하드웨어 플랫폼에서 데모와 프로그램 배포가 가능하다[17]. 미디어파이프를 사용한 손 추적(Hand Tracking)은 머신러닝을 사용하여 한 프레임에서 한 손마다 21 개의 3D 랜드마크(Landmark)를 추출하여 높은 확률로 손과 손가락 추적하는 기능을 제공하며[18] 얼굴 감지의 경우 6 개의 랜드마크와 초고속 얼굴 감지 솔루션으로 가볍고 성능이 우수한 얼굴 감지기인 BlazeFace 에 기반한다[19].

III. 시스템 구성 및 제안 알고리즘

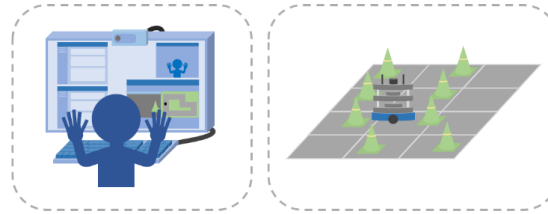


Figure 2(a)

Figure 2(b)

Figure 2. Proposed system configuration

그림 2. 제안한 시스템 구성

그림 2는 제안한 알고리즘을 구현하는 시스템 구성을 나타낸다. 사용자가 양손을 사용한 제스처를 통해서 특정한 동작을 하면 PC의 웹 카메라가 이를 입력 영상으로 사용한다. 입력영상의 제스처 인식을 위해 미디어파이프와 인공지능 알고리즘으로 동작을 추론하여 인식한다. 이때 사용하는 인공지능 알고리즘은 손 제스처에 따라 KNN(K-Nearest Neighbor), LSTM(Long Short Term Memory), CNN(Convolutional Neural Network)을 사용한다. 인식된 사용자 손 제스처 정보는 이벤트 처리기법을 사용하여 로봇의 리눅스 기반 노머신(nomachine) 상에 전달되어 로봇 동작을 제어하게 된다. 그림 2(a)의 모니터 화면은 웹캠(webcam) 영상, 로봇에 부착된 카메라 영상, 실험을 위해 자체 제작한 2D 탑뷰(top view) 레이싱(racing)게임, PC 주피터 랩(Jupyter lab), 로봇 주피터 랩으로 구성되어 있다. 이때 사용되는 주피터 랩은 코딩 및 실행을 위한 개발 도구로 인공지능 방법 추론과 인식 및 정보 전달을 위한 루틴으로 구성되어 있다. 그림 2(b)는 직진, 후진, 정지, 출발, 좌회전, 우회전 등 사용자 손 제스처 인식 결과를 사용해 로봇의 동적제어가 수행되고 있는 그림이다.

그림 3은 제안한 알고리즘 구성으로 손 제스처 정보 인식을 위해 웹캠 영상을 입력으로 사용한다. 또한 리눅스 기반의 로봇을 사용하기 위하여 초기화 루틴을 동작 시킨다. 특히 PC 윈도우즈 환경에서 리눅스 기반의 로봇에 손 제스처 인식정보를 전달하기 위한 방법으로 이벤트 처리 기법을 사용하므로 로봇의 초기화 동작 모드의 설정이 중요하며, 이를 위해 PC에서 로봇 주소로 접근해 리눅스 기반의 노머신(nomachine)을 통해 로봇의 동적제어를 구현한다. PC와 로봇 초기화 동작이 이상이 없는 경우에 입력영상을 미디어파이프를 사용해 핸드(손, 손가락)데이터 정보를 추출하고, 인공지능 기법을 사용한 왼손과 오른손을 구분과 각각의 손가락 정보를 추론하고 인식해 스테이틱(static) 핸드 제스처와 다이내믹

(dynamic) 핸드 제스처로 구분하여 인식한다. 만일 인식을 제대로 못한 경우엔 초기모드로 되돌아가 인식을 위한 기본 동작을 반복한다. 스테이틱(static) 핸드 제스처 인식을 위해 핸드(손, 손가락)데이터와 KNN 을 사용하여 인식하며 추가적인 제스처 정보 입력은 세부적인 자세 정보를 핸드데이터와 KNN 기법으로 추출해 게임의 메인 캐릭터와 로봇 이동 제어에 활용한다. 추가정보 인식이 안되는 경우 스테이틱 핸드 제스처 모드인지를 한번 더 확인해 스테이틱 핸드 제스처 모드이면 추가정보를 인식 루틴을 수행하고 아닌 경우 다이내믹 핸드 제스처 인식 루틴으로 이동한다. 이 핸드 제스처는 핸드 사인(hand sign) 모드와 핸드 다이내믹(hand dynamic) 모드로 구분해 사용한다. 핸드 사인 모드는 사용자가 특정 문자를 입력한 경우로 예를 들어 “S” 형태의 문자를 입력하면 이를 핸드데이터와 CNN 을 사용해 인식하고 추가적인 정보 분석을 통해 게임과 로봇에 적용해 적절한 “S” 행동을 구현한다. 이때 사용자는 카메라를 통하여 로봇의 움직임과 PC 모니터 화면에 나타난 게임의 가상객체의 움직임을 동시에 모니터링하여 세부 동작을 확인 및 제어를 할 수 있다. 핸드 다이내믹 모드는 손과 손가락을 반복적으로 같은 움직이는 경우로 예를 들어 일상에서 동작으로 이해가 되는 “COME”, “GO”, “ROTATE” 등의 움직임 명령을 내릴 수 있다. 이 명령은 핸드데이터와 LSTM 을 사용해 인식하고 추가적인 제스처 정보 입력은 세부 정보 인식을 통해 게임상의 객체와 로봇에게 전달해 동일한 움직임 제어를 수행할 수 있다. 또한 동시에 가상세계의 게임과 실세계의 로봇이 있는 공존 공간상에서의 두 객체의 움직임 제어도 제한한 인터페이스 알고리즘을 적용해 구현할 수 있다. 각각의 제스처 모드에서 원활한 동작이 안되는 경우 해당 모드를 반복 확인해 인식 여부에 따라 해당 모드를 수행하며, 최종적으로 전반적인 상황평가를 통한 보완적인 제어를 수행한다. 보완적인 처리가 필요 없는 경우 알고리즘 수행은 마무리된다.

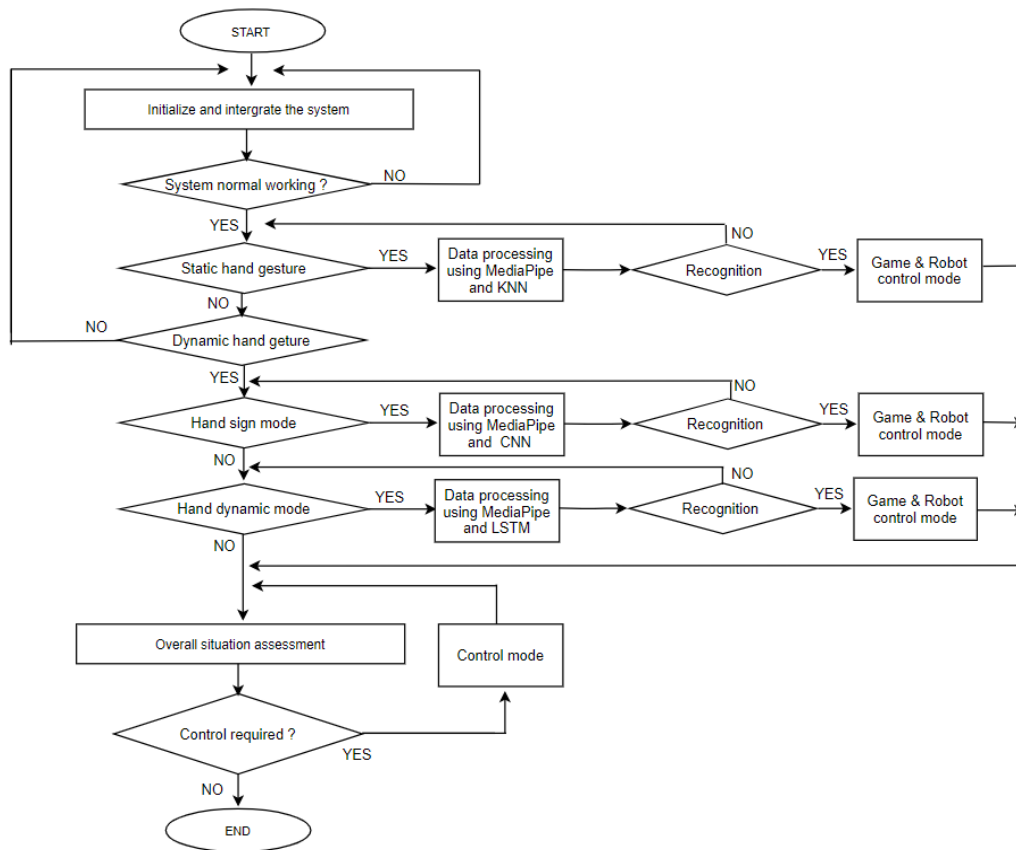


Figure 3. Proposed Algorithm Configuration Diagram

그림 3. 제안한 알고리즘 구성도

IV. 실험

4.1 스테이틱 핸드 제스처(static hand gesture)

스테이틱 핸드 제스처 인식은 정적인 사람의 몸짓을 인식하여 제스처를 분류하는 방법으로 구글의 미디어파이프의 파이프라인 처리를 통해 얻은 21 개 손가락 특징점을 위치와 특징점간 각도를 참조하고 데이터 셋을 만들고 KNN[20] 기법을 사용해 스테이틱 핸드 제스처 인식을 구현한다. KNN 은 n 개의 특성을 가진 데이터는 유사한 특성을 가진 데이터들 간의 거리를 구해 분류할 수 있다. 이때 분류를 알 수 없는 데이터 처리는 가장 가까운 이웃 k 개의 분류를 확인하여 결정할 수 있으며 분류기의 효과를 높이기 위해 파라미터를 조정할 수 있다. 본 연구에서는 작성한 데이터 정확도 값을 고려하여 $k=3$ 으로 설정해 구현한다.

4.1.1 실험 1

실험 1 은 사용자의 직진(FORWARD), 후진(BACKWARD)을 표시하는 손가락 제스처 정보를 KNN 과 미디어파이프를 사용해 스테이틱 핸드 제스처를 인식하고 세부적인 손가락 인식을 추론하고 인식한다. 또한 인식한 제스처 정보의 내용을 가시적으로 확인하기 위해 자체 제작한 2D 레이싱 게임의 메인 캐릭터인 로봇카(robot car)에 적용해 앞, 뒤 움직임 제어 결과를 확인한다.

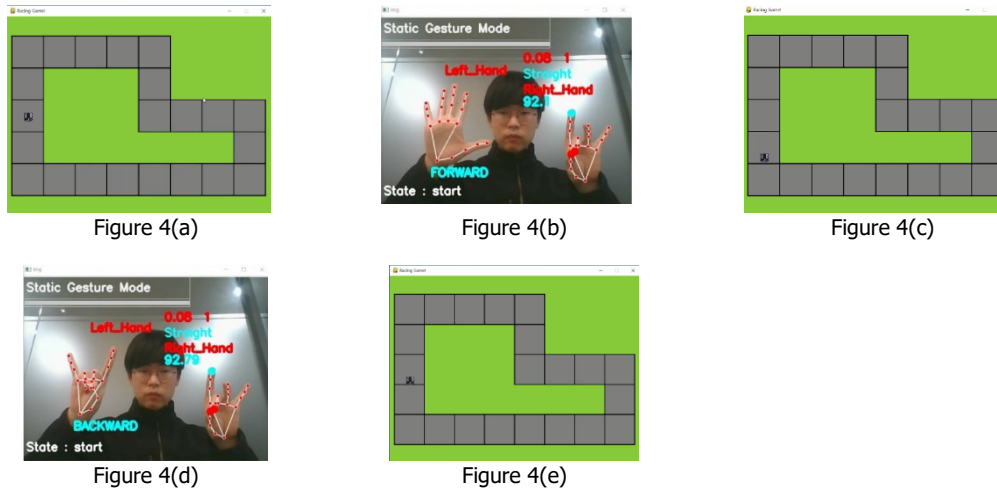


Figure 4. Image of experimental results of recognizing user forward and backward static hand gestures and applying games

그림 4. 사용자 전진과 후진의 스테이틱 핸드 제스처 인식 및 게임 적용 실험 결과 영상

그림 4(a)는 게임의 초기상태로 녹색배경에 회색 타일로 길을 구성하였다, 이 길을 따라 메인 캐릭터인 로봇카가 제어신호에 따라 움직인다. 초기 모드이므로 로봇카는 시작 위치에 정지해 있으며 제어 정보에 따라 주행을 시작한다. 그림 4(b)는 사용자가 왼손은 모두 펴고 동시에 오른손은 엄지와 중지를 거의 마주보고 있는 상태이며 나머지 손가락은 편상 상태이다. 이 상태를 스테이틱 핸드 제스처 앞 방향 직진모드(“FORWARD”, “Straight”)로 판단해 그 결과와 관련된 정보들을 표시한다. 왼손 아래 “FORWARD”로 방향을 표시하고, 오른손 위의 0.08 은 엄지와 중지의 거리를 나타내며 그 옆의 '1'은 로봇카의 이동 속도를 구분하여(최대 3) 표시한 것이며, 인식한 “Straight”, “Right Hand”, 오른손 회전각도 92.1 도 제어에 필요한 정보를 표시하고 있다. 이 정보를 종합하고 로봇카의 직진모드 이동 제어가 진행된다. 그림 4 (d)는 사용자의 오른손은 동일한 형태이며 왼손은 검지와 소지를 세운 모드로 이 영상을 스테이틱 핸드 제스처 후진 직진(“BACKWARD”, “Straight”) 모드로 인식한

결과와 관련된 정보를 출력한 그림이다. 그림 4(e)는 인식한 제스처 정보를 게임의 로봇카에 적용하여 뒤로 위치를 이동한 결과를 나타낸 그림이다. 게임 속 로봇카 위치를 확인해보면 그림 4(c)보다 뒤쪽 방향으로 직진 이동한 내용을 확인할 수 있다.

실험 1 을 통해 제안한 지능형 인터페이스 알고리즘을 사용해 사용자의 직진, 후진 등의 스테이틱 핸드 제스처를 제대로 인식하였고, 인식 결과를 게임에 적용해 게임의 로봇카가 사용자의 제스처정보에 동적으로 맞추어 실시간으로 이동한 결과를 가시적으로 확인할 수 있다.

4.1.2 실험 2

실험 2 에서는 좌회전, 우회전을 표현하는 스테이틱 핸드 제스처 정보를 인식하고 이를 게임에 적용해 인식정보에 따른 가시적인 이동 결과를 확인한다.

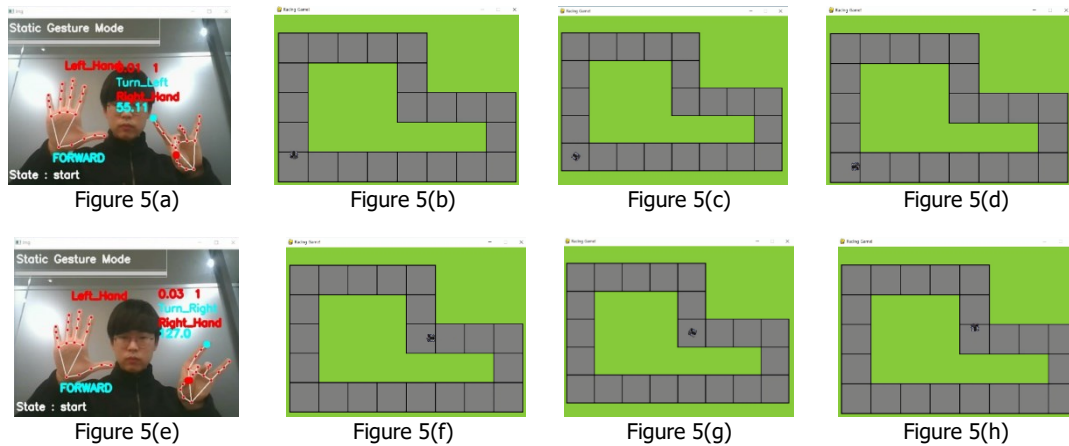


Figure 5. Image of experimental results of recognizing static hand gestures of user left and right turns and applying games

그림 5. 사용자 좌회전, 우회전의 스테이틱 핸드 제스처 인식 및 게임 적용 실험 결과 영상

그림 5(a)는 사용자가 왼손은 모든 손가락을 펴고 동시에 오른손은 엄지와 중지를 속도 제어를 위하여 거의 마주 보고 있는 상태이며 나머지 손가락은 편 상태이다. 이 영상을 스테이틱 핸드 제스처 앞 방향 좌회전(“FORWARD-Turn Left”)으로 인식한다. 왼손 아래 "FORWARD"로 방향정보를 나타내었으며, 오른 위의 0.03 은 엄지와 중지의 거리를 사용한 속도 '1'(최대 3)이며 그 아래 인식 결과 “Turn Left” (좌회전)를 출력한다. 이때 오른손의 회전각도는 55.11 도(오른손 위)는 앞 방향 좌회전으로 코너를 주행하기 위하여 사용자가 선택한 각도이다. 인식한 제스처 정보를 제작한 게임에 적용하면 로봇카가 그림 5(b)인 상태에서 좌측으로 회전하면서 그림 5(c)와 같이 이동한다. 그림(d)는 좌 회전이 마무리되는 시점을 출력한 그림이다. 그림 5(e)는 그림 5(a) 같이 왼손은 “FORWARD” 제스처를 하고 있으며 오른손은 회전각도가 127.0 도로 오른쪽 회전을 인식한 스테이틱 핸드 제스처이다. 이 인식 정보를 게임에 적용해 로봇카는 앞 방향 우회전(“FORWARD”, ”Turn right”)을 그림 5(f)(g)(h) 같이 수행한다. 실험 2 에서 사용자의 스테이틱 핸드 제스처에 따른 좌회전, 우회전 인식이 잘되었고, 게임상의 로봇카도 제대로 동작하는 것을 확인할 수 있다.

4.2 다이내믹 핸드 제스처(dynamic hand gesture)

다이내믹 핸드 제스처는 손과 손가락의 사용방법에 따라 핸드 다이내믹 모드 제스처와 핸드 사인 모드 제스처로 정의하고 각각의 모드로 구분하여 실험한다.

4.2.1 핸드 다이내믹 모드 제스처(hand dynamic mode gesture) 실험

시간의 흐름에 따른 지속적인 손(손가락)의 움직임이 있는 경우를 핸드 다이내믹 모드 제스처로 정의하고 LSTM[21]과 미디어파이프를 사용해 인식한다. 인식한 정보를 세분화해 제작한 레이싱 게임에 적용해 성능을 가시적으로 확인한다. 사용자의 핸드 다이내믹 모드 제스처는 3 개의 의미 있는 내용 [“GO”, “COME”, “ROATATE”] 등의 제어 명령어로 구성하여 실험을 진행하며 이때 사용한 데이터 셋은 사용자의 3 가지 제스처를 각 제스처마다 60 초씩 랜드마크 점의 위치와 각도를 계속해서 녹화하여 numpy 배열 형태로 저장한다. 저장한 데이터셋에서 학습데이터(train_data)와 테스트데이터(test_data)를 9:1 비율로 나눈다. 모델을 제작하기 위해 케라스가 제공하는 Sequential 클래스를 사용하며, 텐스레이어(Dense Layer)에서 3 개의 클래스를 구분하기 위해 출력(output)을 3 으로 설정하고, 활성화 함수는 소프트맥스(softmax)를, 손실(loss)은 categorical_crossentropy 로 설정해 모델을 학습시킨다.

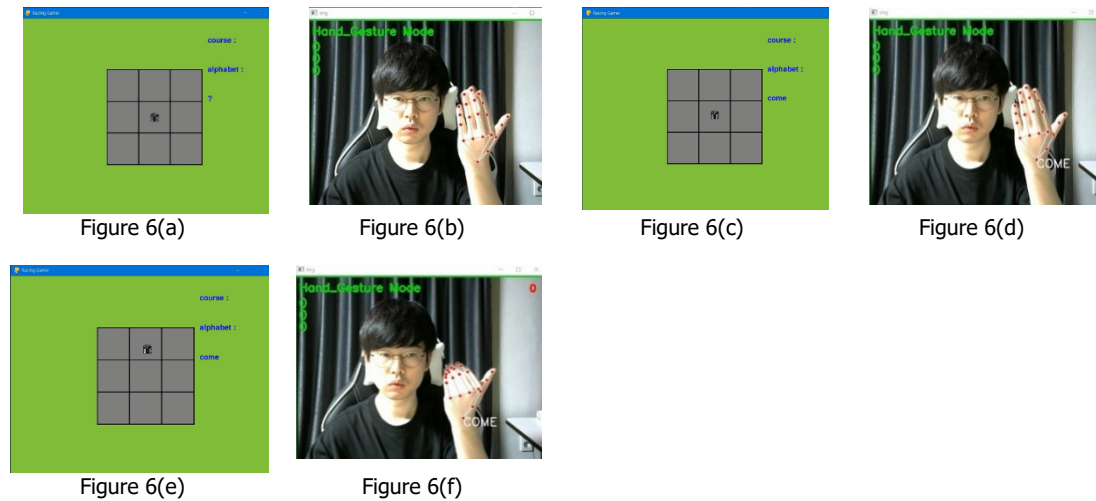


Figure 6. Image of the result of recognizing the hand dynamic mode gesture “COME” and applying it to the racing game

그림 6. 핸드 다이내믹 모드 제스처 “COME”을 인식하고 레이싱 게임에 적용한 결과 영상

그림 6(a)는 처음 시작할 때의 게임 영상으로 그림 6(a) 게임 초록색 배경 9 개 사각형 블록 맵(block map)으로 구성되어 있다. 블록[0,1]의 위치에 로봇카가 정지해 있다. 이때 오른손 손가락을 위로 모으고 앞뒤로 흔드는 그림 6(b)(d)(f) 같은 입력영상으로 움직임 정보를 미디어파이프를 사용해 손가락의 랜드마크의 위치를 인식해 표시하고(흰색선과 빨간색 포인트)이 좌표정보를 분석해 주기적이고 일정한 패턴을 유지하고 있는 조건을 만족하는지를 인식해 최종 판단한다. 이 경우 핸드 다이내믹 모드 제스처 중 “COME” 모드로 인식하며 사용자 오른손의 아래쪽에 “COME”으로 결과를 출력한다. 동시에 게임 영상에서도 블록 우측에 “COME”이라는 인식 결과를 그림 6(d)(f)같이 출력한다. 인식된 제스처 정보를 게임에 적용해 로봇카를 위쪽으로 그림 6(e)같이 이동한다. “COME” 모드가 인식되는 경우 맵의 끝까지 이동을 계속한다. 실험 결과 그림을 통해 핸드 다이내믹 모드 인식 및 제어가 동적으로 잘 수행함을 확인할 수 있다.

4.2.2 핸드 사인 모드 제스처(hand sign mode gesture)

핸드 사인 제스처는 손가락을 사용하여 특정한 정보를 생성해 특정 목적을 위하여 사용하는 제스처로 정의할 수 있다. 이번 실험에서는 일반적으로 이해될 수 있는 형태의 문자를 손가락으로 생성하고 생성된 문자정보를 미디어파이프와 CNN[22]을 사용하여 핸드 사인 모드 제스처로 추론하고 인식한다. 인식된 결과는 게임의 로봇카에 적용하여 제안한 방법의 결과와 성능을 확인한다. 이때 입력 형상 이미지를 ImageDataGenerator 함수를 사용해

100 개의 데이터파일로 증강시켜 데이터 셋을 구성한다. CNN 모델 학습 과정으로 케라스가 제공하는 Sequential 클래스 합성곱 신경망을 구성한다. Convolution Layer 의 Conv2D 에서 입력 이미지 크기 24x24 채널 3 개, 필터 크기 3x3 32 개, 패딩(padding)은 same 으로 하여 출력 이미지 크기와 입력 이미지 크기가 같게 설정하며 활성화함수는 렐루(relu)를 사용한다. 풀링레이어의 MaxPooling2D 에서 풀 크기는 2x2 로 설정하고 풀리 커넥트레이어의 텐스 층을 추가하여 활성화함수 소프트맥스를 사용하고 손실함수는 3 개 이상의 클래스를 분류하기 위해 'categorical_crossentropy'를 사용한다. 옵티마이저는 아담, 에포크는 100, validation_steps 은 5 로 설정해 모델로 사용한다.

• 4.2.2.1 실험 1

실험 1에서는 핸드 사인 모드 체크처로 입력된 정보를 “S” 형태로 인식하고 인식한 결과를 게임에 적용한다.

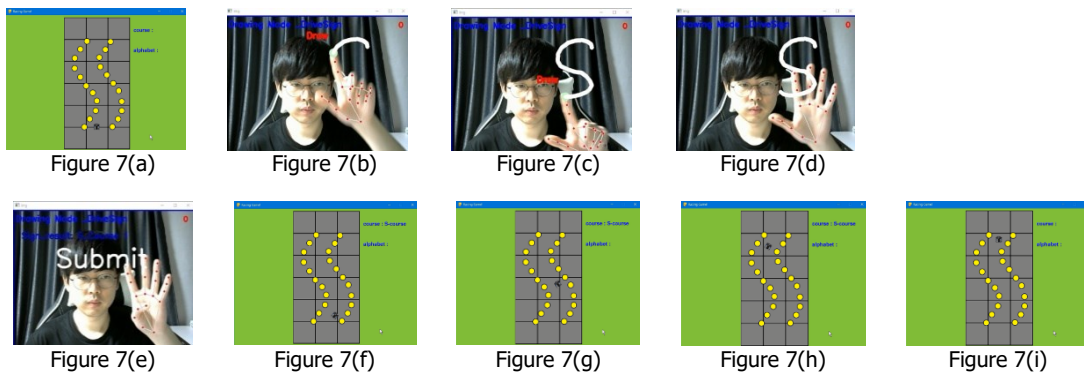


Figure 7. Result of recognizing the user's hand sign mode gesture input as an "S" shape using CNN and MediaPipe and applying it to the racing game

그림 7. 사용자의 핸드 사인 모드 체크처 입력을 CNN과 미디어파이프를 사용하여 “S” 형태로 인식하고 레이싱 게임에 적용한 결과

그림 7(a)는 제작한 게임의 초기 모드로 초록색 바탕화면에 16 개의 사각형 블록 맵으로 구성되어 있고 로봇카는 시작 위치에 정지해 있다. 그림 7(b)(c)(d)는 사용자가 오른손 검지를 사용해 “S” 형태의 정보를 그리고 그림 7(e)과 같은 손가락 체크처를 사용해 입력하면 CNN 과 미디어파이프를 사용해 핸드 사인 체크처 모드로 “S” 모양을 입력한 상태로 인식한다. 이 정보를 게임상의 로봇카에 적용하여 “S” 형태의 움직임을 수행한다. 이 경우 로봇카의 움직임 범주를 노란 고깔로 디자인해 이동영역을 벗어나지 않게 그림 7(f)(g)(h)(i) 같이 구현하였다.

4.3 스테이틱 핸드 체크처 인식과 로봇 제어실험

사용자의 스테이틱 핸드 체크처 인식 정보를 제안한 알고리즘 방법으로 인식하고 인식한 체크처 정보를 리눅스 기반의 그림 8과 같은 3개의 옴니 휠을 가진 로봇에 적용하여 이동방향과 속도로 로봇을 제어하는 실험이다. 이 실험을 통하여 실세계에 있는 로봇이 지능형 인터페이스를 사용하여 로봇의 동적 제어가 가능한지를 확인한다. 사용자 손 체크처 동작은 앞의 게임 실험에서 사용한 동작과 그림 9 와 같이 일부 다르게 구성해 다양한 손 체크처 입력에 대해서 인식하도록 구성했다.



Figure 8. Experimental robot with 3 omni wheels

그림 8. 3 개의 옴니 휠을 가진 실험용 로봇

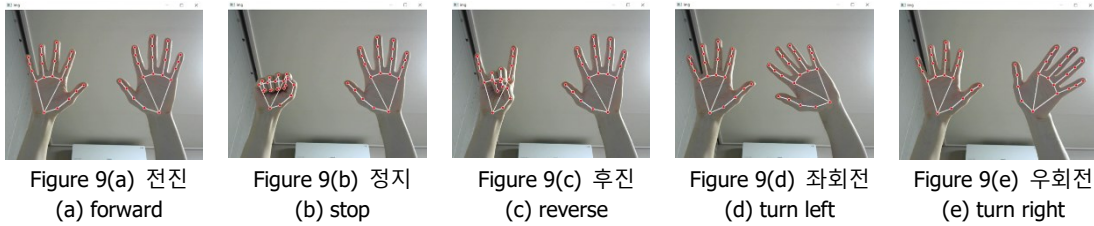


Figure 9. Classification of left and right hand gestures and types of finger gestures for recognizing user's static hand gestures

그림 9. 사용자의 스테이틱 핸드 제스처 인식을 위한 왼손과 오른손 구분과 손가락 제스처의 종류

그림 9는 사용자의 스테이틱 핸드 제스처 종류를 나타내며 왼손, 오른손 구분 및 손가락 모양에 따른 전진, 후진, 정지, 좌회전, 우회전 등의 핸드 제스처 내용을 구분하여 출력한 그림이다.

4.3.1 실험 1

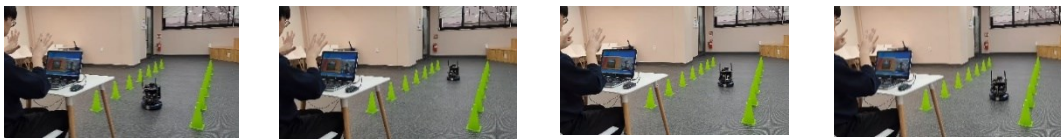


Figure 10(a)

Figure 10(b)

Figure 10(c)

Figure 10(d)

Figure 10. Forward and backward control result image of robot using static hand gesture recognition information

그림 10. 스테이틱 핸드 제스처 인식 정보를 활용해 로봇의 전진 및 후진 제어 결과 영상

실험 1 은 사용자 자세를 스테이틱 핸드 제스처 전진 및 후진으로 구분해 인식하고 이 정보를 리눅스 기반의 로봇에게 전달해 그림 10(a)(b) 같이 직진과 그림 10(c)(d) 같은 후진 동작을 제어한 결과 그림이다. 실험 1 을 통하여 인식한 제스처 정보를 실세계에 있는 로봇에게 보내어 전, 후진 이동 제어가 잘 동작되는 것을 확인할 수 있다.

4.3.2 실험 2

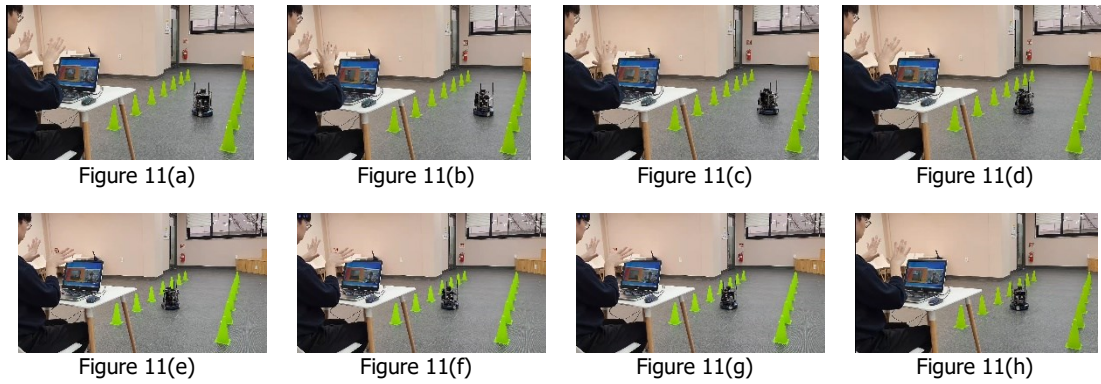


Figure 11. Left and right turn control result image of robot using static hand gesture recognition information

그림 11. 스테이틱 핸드 제스처 인식 정보를 활용해 로봇의 좌회전, 우회전 제어 결과 영상

실험 2 는 그림 11(a)(b)(c) 같은 사용자 동작을 스테이틱 핸드 제스처 좌회전 또는 그림 11(e)(f)(g) 같은 우회전으로 인식하고 인식된 제스처 정보를 사용해 로봇을 그림 11(a)(b)(c) 같은 좌회전, 그림 11(c)(f)(g) 같은 우회전 동작으로 제어했다. 결과 영상을 보면 좌회전 혹은 우회전 동작 후 그림 11 (d)(h) 같은 직진모드를 사용해 지속적인 이동제어를 하고 있는 것을 확인할 수 있다.

4.4 핸드 다이내믹 모드 제스처 인식과 로봇 제어 실험

실험 1 에서는 사용자의 핸드 다이내믹 모드 제스처 정보를 인식하고, 인식한 동적 제스처 정보를 3 개의 옴니 휠을 가진 로봇에 적용하여 로봇을 제어하는 실험이다. 이때 다이내믹 핸드 제스처는 “COME”, “GO”, “Rotate” 등 다양한 동작으로 구성 가능하다.

4.4.1 실험 1



Figure 12. Hand dynamic mode gesture “COME” recognition and robot control

그림 12. 핸드 다이내믹 모드 제스처 “COME” 인식과 로봇 제어

사용자가 그림 12(a)와 같이 연속적인 움직임을 하는 경우 다이내믹 핸드 제스처 중 그림 12(b)와 같이 “COME”으로 인식한다. 인식된 다이내믹 핸드 제스처 정보를 로봇에게 전달하여 로봇도 “COME” 동작을 수행해 사용자 앞으로 이동하는 있는 그림 12(c)(d)을 확인할 수 있다.

4.5 핸드 사인 모드 제스처 인식과 로봇 제어 실험

실험 1 에서는 사용자의 핸드 사인모드 제스처를 제안한 방법으로 인식하고, 인식한 제스처 정보를 로봇에 적용하여 제어하는 실험이다.

4.5.1 실험 1

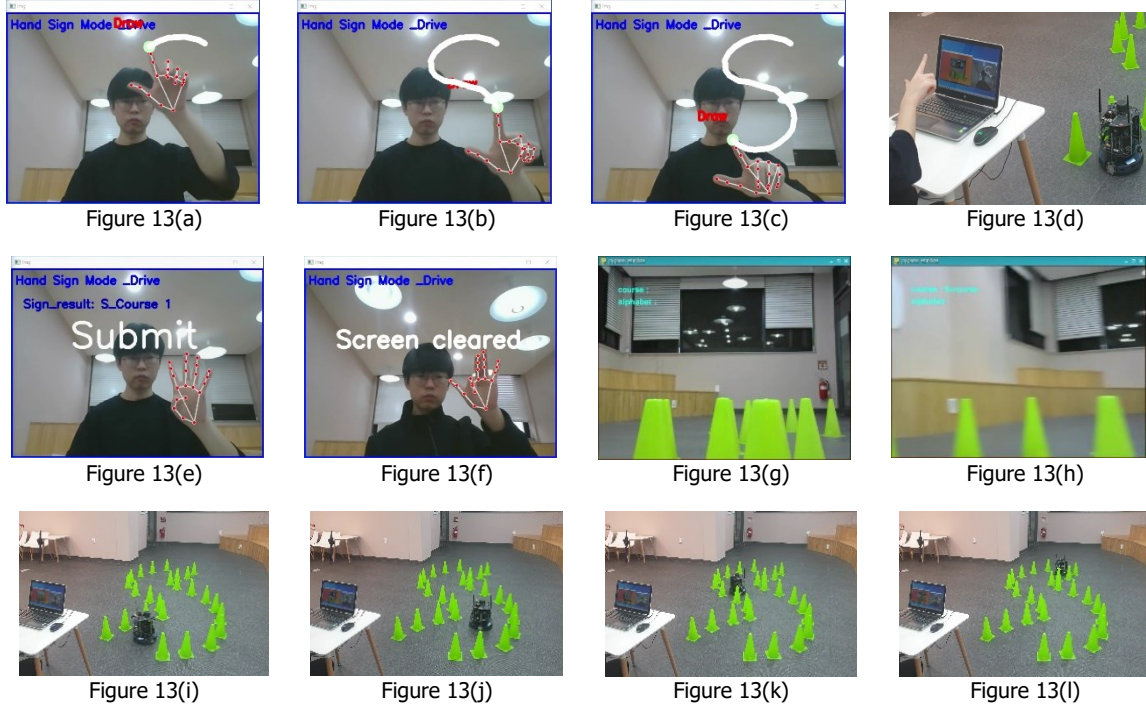


Figure 13. “S” information recognized using hand sign mode and the driving the robot result image

그림 13. 핸드 사인 모드를 사용해 인식한 “S” 정보와 로봇 주행 결과 영상

사용자가 오른손 중지, 약지, 소지 손가락은 접고 엄지, 검지 손가락을 편 채 화면에 핸드 사인을 그림 13(a), (b), (c)와 같이 검지손가락 끝으로 컬러색으로 특정문자 사인을 그린다. 그림 13(d)는 핸드 사인 모드 제스처를 사용자가 실행하고 있는 그림으로 사용자와 로봇이 함께 나타나 있다. 그림 13(e)와 같이 손가락을 편 상태에서 엄지손가락만을 손바닥 안쪽으로 접으면 화면에 “Submit” 텍스트 출력과 함께 그려진 캔버스에 CNN 모델을 사용하여 핸드 사인을 추론해 인식을 수행한다. 그림(g)(h)는 로봇이 가지고 있는 카메라를 사용해 이동 중인 상황을 화면으로 실시간으로 출력하고 있는 영상으로 원격제어에서 유용하게 사용할 수 있는 기능이다. 그림 13(i), (j), (k), (l)은 사용자의 핸드사인 모드 제스처로부터 인식된 명령을 전달받아 “S” 형태의 궤적을 그리며 동작을 수행하는 로봇의 주행 모습이다. 그림 13(f) 작업 수행이 끝난 후 새로운 입력을 위하여 사용자가 오른손의 검지를 접고 나머지 4 개 손가락을 편 상태로 화면을 클리어 명령하는 그림이다. 실험 1 은 핸드사인 모드 제스처로 정보를 입력하면 데이터셋과 CNN 을 사용해 인식을 하고, 이 인식 정보를 로봇에게 전달해 특정형태의 움직임을 따라 잘 동작하는 것을 확인할 수 있다. 이때 로봇의 궤적은 주어진 태스크 수행을 위해 환경에 맞게 미리 설정 놓을 수 있다.

4.6 핸드 다이내믹 모드 제스처 인식과 게임과 로봇 제어 실험

그림 14 의 그림(a), (b)처럼 사용자가 자세를 취하면 핸드 다이내믹 모드 제스처 “COME”으로 인식하고 이 인식 정보를 레이싱 게임의 로봇카에 전달해 그림 14(c)와 (d)와 같이 위에서 아래로 이동한다. 이 인식 정보를 이벤트 처리기법으로 로봇에게 전달하여 그림 14(e)(f)(g)와 같이 위에서 아래로(사용자 방향) 움직이게 된다. 즉 다이내믹 핸드 제스처 COME” 명령에 게임의 로봇카와 실세계의 로봇을 위에서 아래로 움직이는 제어를 수행한다. 이 실험을 통하여 사용자의 제스처 정보를 인식을 통하여 가상공간의 게임 캐릭터와 실세계 로봇을 제어하는 것이 가능하다는 것을 확인할 수 있다.

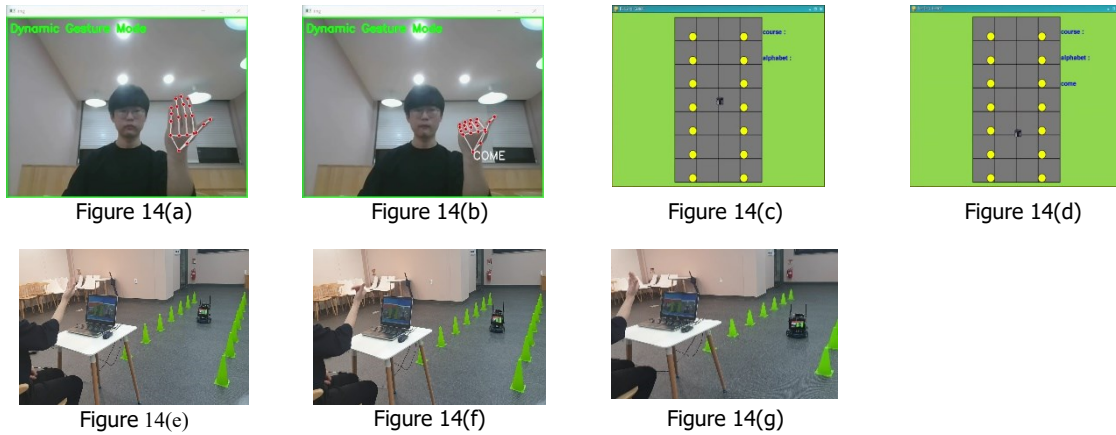


Figure 14. The image of implements simultaneous control of games and robots by "COME" recognition using hand dynamic mode gesture

그림 14. 핸드 다이내믹 모드 제스처 "COME" 인식과 게임과 로봇 동시제어를 구현한 영상

V. 결론

제안한 지능형 인터페이스 기술은 실생활에서 유용한 다양한 서비스를 제공한다. 별도의 제스처 인식을 위한 디바이스 없이 신체를 활용한 인터페이스로 직관적 특성과 손가락의 미세한 움직임 검출이 가능하므로 정교한 조작이 가능하므로 가상현실, 증강현실, 드론, 로봇 등의 제어를 위한 상호작용에 활용할 수 있다. 또한 빠른 시간내에 숙련되는 장점이 있어 좋은 학습성과 사용성을 가지고 있다. 특히 사용자의 핸드 제스처 정보의 세밀한 인식을 위하여 스테이틱 핸드 제스처와 다이내믹 핸드 제스처로 구분하고 다이내믹 핸드 제스처는 핸드 사인모드 제스처와 핸드 다이내믹 모드 제스처로 구분해 인터페이스의 유효성을 확장하였다. 이들 제스처들의 효과적으로 인식을 위해 움직임 특성을 고려한 KNN, LSTM, CNN 의 인공지능 기법을 구분해 적용하고 미디어파이프의 랜드마크 정보를 활용해 제스처 인식을 구현하였다. 인식된 제스처 정보 성능을 시각적으로 확인하기 위하여 2D 레이싱 게임을 제작하고 로봇카를 디자인해 인식 정보에 따른 직진, 후진, 좌회전, 우회전 등의 이동 결과를 확인하였으며 검지 손가락을 사용한 핸드 사인 정보를 구체적이고 다양한 정보로 표현하고 인식할 수 있었다. 뿐만 아니라 인식된 각 제스처 정보를 실세계의 로봇에 적용하여 로봇의 동적인 이동 제어를 구현해 제안한 방법의 성능을 확인할 수 있었다. 그러나 3 개의 옴니 휠을 가진 로봇 제어는 4 륜 로봇과 달리 360 를 회전 가능한 장점도 있으나, 바닥 등 주행 환경 상태에 민감하며 좌회전, 우회전, 직진 등에 오차가 누적되어 세밀한 추가 제어가 필요해 카운터를 사용한 제어방식을 통해 해결했다. 또한 가상 세계의 게임 캐릭터와 실세계의 로봇의 동적 제어 실험도 제안한 지능형 인터페이스로 가능하다는 것을 확인할 수 있었다.

VI. 참고문헌

- [1] Song Bok-Deuk, Lee Seung-Hwan, Choi Hong-Kyu and Kim Sung-Hun, "Design and Implementation of a Stereoscopic Image Control System based on User Hand Gesture Recognition", Korea Institute of information and Communication Engineering, v.26 no.3, pp. 396-402, 2022.
- [2] Heo Gyeong-Yong, Song Bok-Deuk, Kim Ji-Hong, "Hierarchical Hand Pose Model for Hand Expression Recognition", Korea Institute of information and Communication Engineering, v.25 no.10, pp. 1323-1329, 2021.
- [3] Lim Young-Jae, Jung Il-Hong, "Hand Gesture Recognition in the Virtual Space based on Deep

- Learning”, Korea Digital Contents Society, v.21 no.3, pp. 471 - 478, 2020.
- [4] Kim Kyu-Min, Baek Joong-Hwan, “Real-time hand gesture recognition based on deep learning”, Korea Multimedia Society, v.22 no.4, pp. 424 - 431, 2019.
- [5] Lee Byeong-Hee, Oh Dong-Han, Kim Tae-Young, “3D Virtual Reality Game with Deep Learning-based Hand Gesture Recognition”, Korea Computer Graphics Society, v.24 no.5, pp. 41 - 48, 2018.
- [6] P. K. Pisharady and M. Saerbeck, “Recent methods and databases in vision-based hand gesture recognition: a review,” *Computer Vision and Image Understanding*, vol. 141, no. 4356, pp. 152–165, 2015.
- [7] Kim Jin-young, Shim Hyun, "Development of Sign Language Learning Assistance System Using MediaPipe for Sign Language Education for the Hearing Impaired", *Korea Electronics and Telecommunications Society*, v.16 no.6 , pp.1355 - 1362 , 2021.
- [8] Y. Zhou, M. Habermann, W. Xu, I. Habibie, C. Theobalt, F.Xu ,“Monocular Real-Time Hand Shape and Motion Capture Using Multi-Modal Data,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 5345-5354.
- [9] Bread type developing countries website, <https://www.youtube.com/@bbanghyong>
- [10] B. D. Song, H. K. Choi, S. H. Kim, “Research Trends of User Hand Gesture Recognition Technologies for Utilizing User Interaction in Stereoscopic Images,” in *Proceeding of the Korea Institute of Communications and Information Sciences Summer Conference*, 2021, pp. 119-120.
- [11] MediaPipe, <https://mediapipe.dev/>
- [12] E. Stergiopoulou, N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique", *Engineering Applications of Artificial Intelligence*, vol.22, no8, pp 1141-1158, 2009
- [13] A. Kendon, “Current issues in the study of gesture, in: *The Biological Foundation of Gestures: Motor and Semiotic Aspects*”, Psychology Press, 1986, pp. 23-47.
- [14] Different approaches to Computer Visual-Based Hand Gesture Recognition, <https://medium.com/@bluebirb/different-approaches-to-computer-visual-based-hand-gesture-recognition-17164988e7f6>
- [15] Ha-Dang Ho, Hong-Quan Nguyen, Thuy-Binh Nguyen, Sinh-Thuong Vu, Thi-Lan Le, "Dynamic Hand Gesture Recognition from Egocentric Videos based on SlowFast Architecture", *2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*, 2022, pp.01-07.
- [16] Yongfeng Dong, Jielong Liu, Wenjie Yan, "Dynamic Hand Gesture Recognition Based on Signals From Specialized Data Glove and Deep Learning Algorithms", *IEEE Transactions on Instrumentation and Measurement*, vol.70, pp.1-14, 2021.
- [17] MediaPipe Framework for ML solutions, <https://viso.ai/computer-vision/mediapipe/>
- [18] MediaPipe Hands, <https://google.github.io/mediapipe/solutions/hands.html>
- [19] MediaPipe Face Detection, https://google.github.io/mediapipe/solutions/face_detection.html
- [20] Concept of K-Nearest Neighbor, <https://hleecaster.com/ml-knn-concept>
- [21] Introduction to LSTM: <https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/>
- [22] CNN concept : <https://www.ibm.com/topics/convolutional-neural-networks>

저자소개



조항준(Hangjun Cho)

2023 년 2 월 전주대학교 스마트미디어학과 졸업(학사)

관심분야 : 영상처리, AI, Game



유준우(Junwoo Yoo)

2023 년 2 월 전주대학교 스마트미디어학과 졸업(학사)
2023 년 3 월 ~ 현재 전주대학교 전기전자공학과 대학원

관심분야 : 영상처리, AI, App 개발



김은수(Eun Soo Kim)

1988 년 중앙대 전기공학과 졸업(석사)
2000 년 중앙대 전기공학과 졸업(박사)
2001 년~현재 전주대 전기전자공학과 교수

관심분야 : DSP 제어, 전력변환기술



이영재(Young Jae Lee)

1994 연세대 전자공학과 졸업(석사)
2000 경희대 전자공학과 졸업(박사)
2002 년~현재 전주대 스마트미디어학과 교수

관심분야 : 컴퓨터비전, App.개발, Game, AI
