

손동작을 이용한 실시간 로봇 팔 제어 임베디드 시스템

김충근*¹, 김은수*², 신재욱**³, 박범용***⁴

Real-time Robotic Arm Control Embedded System using Hand Gestures

Chung-Geun Kim*¹, Eun-Su Kim*², Jae-Wook Shin**³, and Bum-Yong Park***⁴

본 연구는 과학기술정보통신부 및 정보통신기획평가원의 지역지능화혁신인재양성(Grand ICT연구센터) 사업의 연구 결과로 수행되었음. (IITP-2021-2020-0-01612)

요 약

최근 로봇 분야에서 중요성이 커지고 있는 사람-로봇 상호작용(Human-Robot Interaction, HRI) 기술은 로봇 제어를 위한 소통 수단으로 손동작을 많이 이용하고 있으며, 이미지와 생체신호를 이용해 손동작을 분류하는 다양한 연구가 진행되고 있다. 본 논문에서는 기존 연구에서 더 나아가 임베디드 보드 환경에서 이미지를 이용해 손동작을 분류하여 실시간으로 로봇 팔을 제어하는 시스템을 제안한다. 제안하는 시스템은 웹캠으로 촬영되는 이미지 데이터를 전처리한 뒤, CNN(Convolutional Neural Network) 분류기 모델을 이용해 분류하는 과정과 ROS(Robot Operating System) 메시지 통신을 이용해 로봇 팔을 제어하는 과정으로 구성된다. 시스템의 성능은 학습 모델의 분류 정확도, 실시간 로봇 제어 정확도 및 지연 시간을 측정하는 실험을 기준으로 평가했으며, 이 실험을 통해 제안하는 시스템의 높은 정확도와 빠른 처리 속도를 확인하였다.

Abstract

Recently, HRI(Human-Robot Interaction) technology which has become increasingly important in the field of robotic has used hand gestures as communication means for robot control. Various studies are being researched to classify hand gestures using images and biological signals. In this paper, we propose a system that controls the robotic arm in real-time by classifying hand gestures using images in an embedded board environment. The proposed system consists of pre-processing the image data captured by the webcam, classifying it using CNN(Convolutional Neural Network) model, and controlling the robotic arm using ROS(Robot Operating System) in an embedded board environment. The performance of the system is evaluated based on experiments measuring the classification accuracy of the trained model, real-time robot control accuracy and the delay. Through these experiments, the high accuracy and fast processing speed of the proposed system are confirmed.

Keywords

embedded board, CNN, image processing, hand gesture, ROS, robotic arm

* 금오공과대학교 전자공학과 석사과정
- ORCID¹: <https://orcid.org/0000-0002-3765-2796>
- ORCID²: <https://orcid.org/0000-0002-4043-0713>
** 금오공과대학교 전자공학부 조교수
- ORCID: <https://orcid.org/0000-0002-3765-6009>
*** 금오공과대학교 전자공학부, IT융복합공학과 조교수 (교신저자)
- ORCID: <https://orcid.org/0000-0001-9490-0365>

• Received: Jul. 30, 2021, Revised: Sep. 17, 2021, Accepted: Sep. 20, 2021
• Corresponding Author: Bum-Yong Park
Dept. of Electronic Engineering, IT Convergence Engineering, Kumoh National Institute of Technology, 61, Daehak-ro, Gumi-si, Gyeongsangbuk-do, 39177, Korea
Tel.: +82-54-478-7438, Email: bumyoung.park@kumoh.ac.kr

1. 서 론

일반적으로 손동작은 말을 못 하거나 듣는 데 어려움을 겪는 사람들이 사용하는 비언어적인 의사소통 수단임과 동시에, 우리가 전달하고자 하는 의미와 감정을 표현할 수 있는 소통 수단이다. 이러한 손동작은 사람을 완전히 대체하지 못하는 로봇을 효과적으로 제어하기 위한 수단으로 사람-로봇 상호작용(HRI, Human-Robot Interaction) 기술에 적용되어 로봇을 제어하는 연구가 활발하게 이루어지고 있다[1]-[7]. 이 중에서 CNN(Convolutional Neural Network) 모델을 기반으로 근전도(EMG, Electromyogram) 신호를 이용해 실시간으로 로봇 팔을 제어하는 연구[5], 다양한 UAV(Unmanned Aerial Vehicle) 제어를 위해 CNN 모델을 기반으로 손동작을 분류하는 연구[6], CNN 모델을 기반으로 이미지, 거리 데이터를 통해 실시간으로 로봇 팔을 제어하는 연구[7] 등이 있다.

실시간으로 손동작을 인식함과 동시에 로봇과 통신하여 제어하는 과정은 고성능의 개발 환경이 요구되며, 앞서 소개한 기존의 연구도 고성능의 PC(Personal Computer) 환경에서 진행된다. 하지만 최근 들어 고성능의 임베디드 보드가 계속 출시되고 있고, 이를 이용해 실시간으로 손동작 분류를 진행했을 때 높은 분류율을 보여주는 선행 연구가 이루어지고 있다[8]-[10]. 따라서 본 논문은 PC가 아닌 임베디드 보드 환경에서 이미지 데이터로 학습시킨 CNN 분류기 모델로 손동작을 분류하여 실시간으로 로봇 팔을 제어하는 시스템을 제안한다.

제안하는 시스템에서 정의한 손동작은 10가지이며, 수집한 데이터로 CNN 모델을 학습시켜 분류기를 생성한다. 웹캠으로 촬영된 손동작 이미지 데이터는 전처리 과정을 거쳐 분류기에 입력되고, 분류된 데이터를 ROS(Robot Operating System)의 메시지 통신으로 전달하여 로봇을 제어한다.

본 논문의 구성은 다음과 같다. 2장은 제안하는 시스템의 개발환경인 임베디드 보드와 로봇 및 ROS에 관한 내용을 서술하고, 3장은 제안하는 시스템에서 정의한 손동작 종류와 손동작에 따른 로봇의 제어, 이미지 데이터를 전처리하는 과정 그리고 손동작 분류를 위해 사용되는 CNN 모델에 대해 서

술한다. 4장은 제안하는 시스템을 평가하기 위한 실험 내용과 결과를 서술하고, 마지막으로 5장에서 본 논문의 결론 및 향후 연구에 관하여 서술한다.

II. 제안 시스템의 개발환경

제안하는 시스템은 임베디드 보드와 로봇 및 ROS 개발환경을 기반으로 하며 그림 1과 같이 이루어져 있다. 구성은 그림 2와 같이 웹캠으로 촬영된 640×480×3, RGB 데이터를 YCrCb 색공간으로 변환한 뒤, 피부색에 해당하는 색 범위를 지정한다. 지정된 피부색의 경계선을 따라 컨투어가 그려지고, 컨투어 좌표를 추출하여 좌표에 따른 범위의 손동작 이미지 데이터를 잘라내고 64×64×3, RGB의 이미지로 크기를 조절하는 과정을 거쳐 생성된 분류기에 입력한다.

분류기는 CNN 모델 중에서 ResNet-50 모델을 사용하여 손동작을 학습시켜 분류기를 생성하였다. 분류기는 입력된 데이터에 따라 하나의 손동작 데이터를 출력하고, 출력된 데이터는 ROS의 메시지 통신을 통해 로봇으로 수신되어 UR3 로봇과 그리퍼를 제어하는 과정으로 이루어진다.



그림 1. 제안 시스템의 구성 결과
Fig. 1. Result of the proposed system

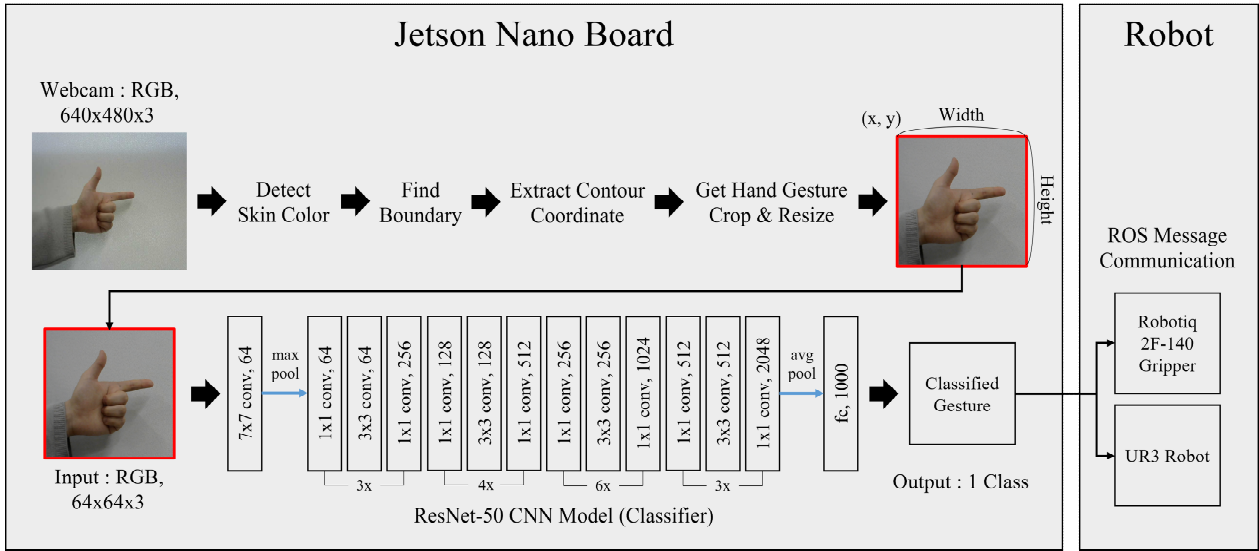


그림 2. 제안하는 시스템의 구성도
 Fig. 2. Configuration diagram of the proposed system

2.1 임베디드 보드

제안하는 시스템에서 사용된 임베디드 보드는 NVIDIA 사의 Jetson nano이다. 표 1과 같이 Jetson nano는 ARM A57의 CPU, 128코어 Maxell의 GPU, 4GB의 RAM의 성능을 갖춘 머신러닝 알고리즘 수행에 적합한 단일 보드 컴퓨터로 CUDA, cuDNN, OpenCV와 같은 머신러닝 시스템 구현에 필요한 모듈이 포함된 Ubuntu 18.04 LTS의 환경을 가진다.

2.2 UR3 로봇 및 그리퍼

제안하는 시스템에서는 Universal Robots 사의 UR3 CB-Series와 Robotiq 사의 2F-140 Gripper를 사용한다. 표 2와 같이 UR3 CB-Series는 6축의 동작 자유도, 500mm의 최대 가동 범위, 3kg의 최대 페이로드를 가졌다. 그림 3과 같은 관절로 이루어져 있고, 표 3과 같이 동작 범위가 무한인 Wrist3을 제외하고 모두 $\pm 360^\circ$ 의 동작 범위를 가진다.

제안하는 시스템에서는 Base, Shoulder, Elbow 3축의 동작 자유도까지 제어할 수 있게 구현하였으며, Wrist3에 위치한 (x, y, z) 점을 말단좌표로 설정하고 이동시켜 UR3 로봇을 제어한다. 말단좌표에 따른 UR3 로봇의 각 관절 각도는 역기구학 해를 이용해 얻는다[11].

표 1. Jetson nano의 사양

Table 1. Specification of Jetson nano

Jetson nano specifications	
CPU	4 Core ARM A57 @ 1.43 GHz
GPU	128 Core Maxwell 472 GFLOPs (FP16)
RAM	4 GB 64 bit LPDDR4 25.6 GB/s
Storage	16 GB eMMC

표 2. UR3 CB-Series의 사양

Table 2. Specification of UR3 CB-Series

UR3 CB-Series specification	
Repeatability	± 0.1 mm / ± 0.0039 in
Ambient temperature range	0 - 50°C
Payload	3 kg / 6.6 lbs
Reach	500 mm / 19.7 in
Degrees of freedom	6 rotating joints

표 3. UR3 로봇 관절의 작동 범위 및 속도

Table 3. Operation range and speed of UR3 robot joint

Robot arm joint	Working range	Maximum speed
Base	$\pm 360^\circ$	$\pm 180^\circ/\text{Sec.}$
Shoulder	$\pm 360^\circ$	$\pm 180^\circ/\text{Sec.}$
Elbow	$\pm 360^\circ$	$\pm 180^\circ/\text{Sec.}$
Wrist 1	$\pm 360^\circ$	$\pm 360^\circ/\text{Sec.}$
Wrist 2	$\pm 360^\circ$	$\pm 360^\circ/\text{Sec.}$
Wrist 3	Infinite	$\pm 360^\circ/\text{Sec.}$

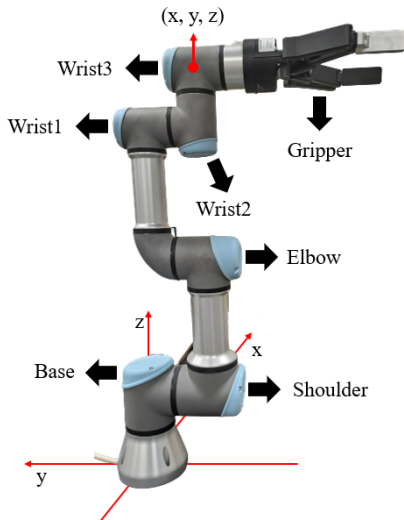


그림 3. UR3 로봇의 관절 구성 및 그리퍼
Fig. 3. UR3 robot joint configuration and gripper

2.3 ROS

ROS는 로봇용 오픈소스 메타 운영체제로 기존의 운영체제에서 수행되는 로봇 제어를 위한 하드웨어 추상화, 저수준 기기 제어, 프로세스 간 메시지 전달, 로봇 패키지 관리 등과 같은 기능들을 제공해주는 프레임워크를 의미한다. 또한, 서로 다른 운영체제 및 하드웨어 간 연동이 가능하고, Python, C++, C#, Java 등 지원하는 프로그래밍 언어에 있어 독립적이어서 다양한 환경에서의 로봇 개발에 적합하다. 그리고 특정 로봇 제어에 필요한 패키지를 직접 개발하거나 받는 것이 가능하다는 접근성과 편리성도 가지고 있다. 추가적으로 3차원 시뮬레이터 도구인

Gazebo와 시각화 도구인 Rviz, rqt로 로봇의 가상 제어와 통신 연결 정보를 확인할 수 있다[12].

ROS에서 로봇의 제어는 최소 단위의 프로그램인 노드가 기능별로 세분화되어 작동된다. 각 노드는 메시지를 주고받는 형식으로 토픽, 서비스, 액션의 방식으로 통신하며, 제안하는 시스템에서는 단방향과 연속성을 가진 토픽 방식을 사용한다. 그림 4는 ROS 메시지 통신 중 토픽 방식으로 통신이 이루어지는 과정이며, ① ~ ⑤의 과정을 통해 Publisher Node와 Subscriber Node의 지속적인 통신이 가능해진다. 제안하는 시스템은 ROS Melodic Morenia 버전을 사용하며, Universal Robots 사에서 제공하는 Universal_Robots_ROS_Driver 패키지, Robotiq 사에서 제공하는 robotiq-kinetic-devel 패키지를 이용해 UR3 로봇과 그리퍼를 제어한다.

III. 제안 시스템의 구현

3.1 손동작 종류 및 제어 설정

본 논문에서 정의한 손동작의 종류와 손동작에 따른 UR3 로봇 및 그리퍼의 제어는 그림 5와 같으며, 학습에 사용된 데이터는 손동작 종류별로 200개씩 수집하였다. 또한, 손동작에 따른 로봇의 제어는 다음과 같다. Clockwise, Counterclockwise는 UR3 로봇의 Wrist3 회전이고, OK, Left, Right, Up, Down은 (x, y, z) 좌표에 따라 상하좌우로 움직이는 UR3 로봇 제어이다.

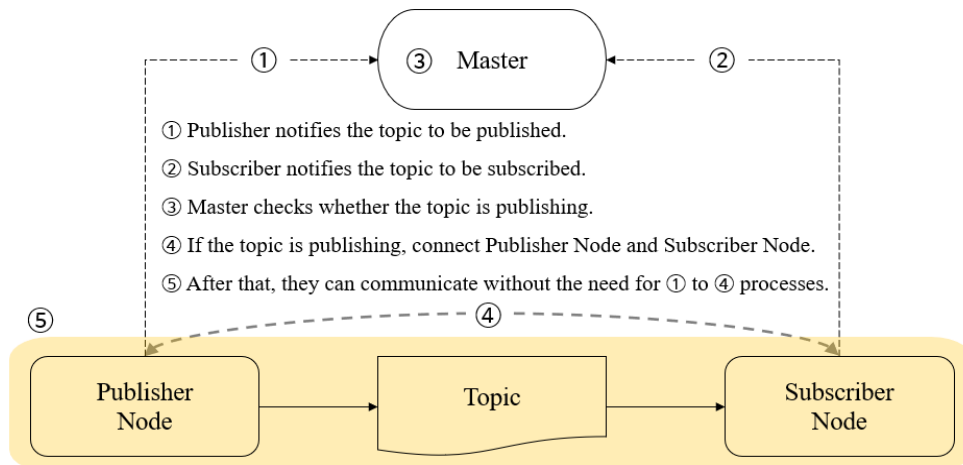


그림 4. ROS 메시지 통신 : 토픽 방식
Fig. 4. ROS message communication : Topic method

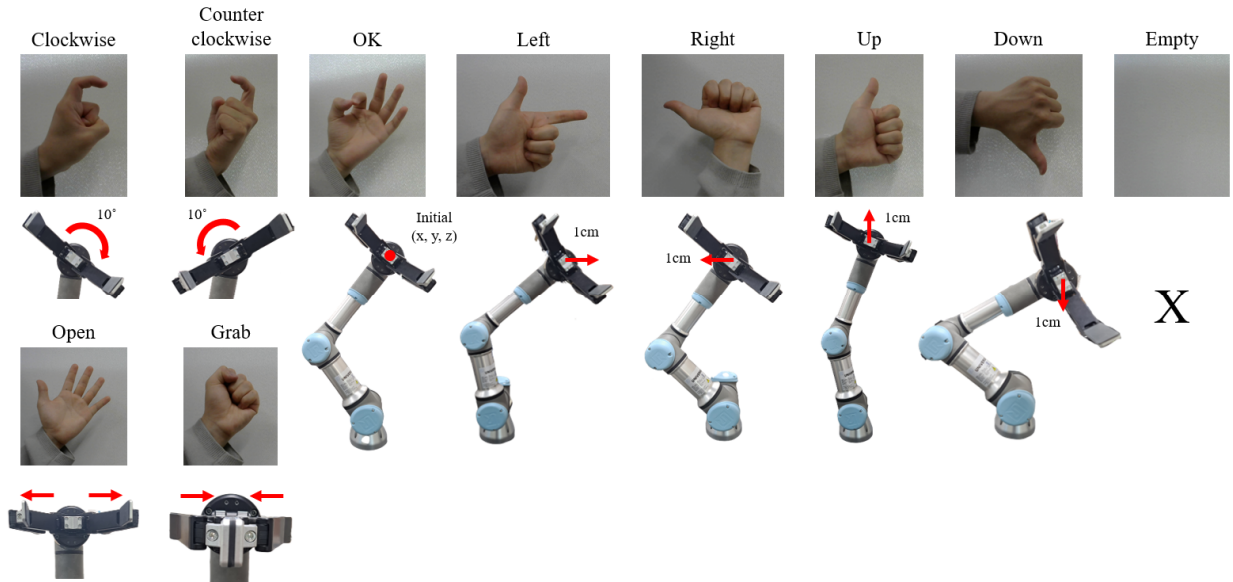


그림 5. 손동작 종류별 로봇 팔 및 그리퍼 제어
 Fig. 5. Robotic arm and gripper control by types of hand gesture

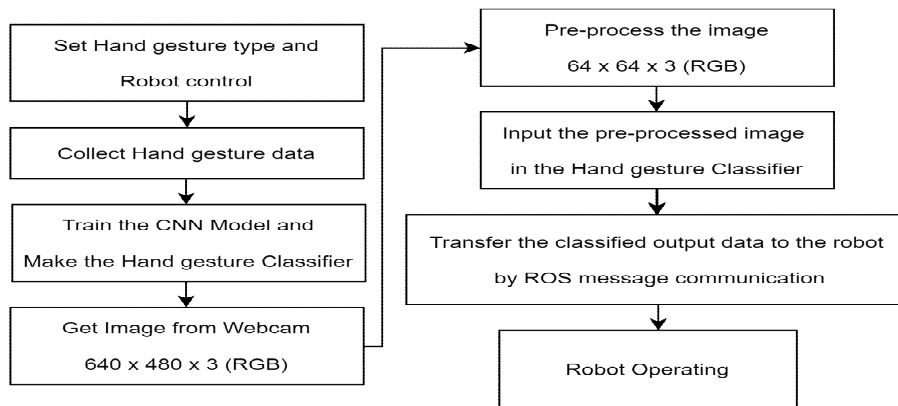


그림 6. 제안 시스템의 구현 과정 흐름도
 Fig. 6. Implementation process flowchart of the proposed system

Open, Grab은 그리퍼를 여닫는 제어이며, Empty를 통해 손동작 입력을 구분한다. 전체적인 시스템의 구현 과정은 그림 6에서 확인할 수 있다.

3.2 데이터 전처리

제안하는 시스템은 웹캠의 이미지 데이터에서 손 동작에 해당하는 범위의 데이터만 분류기에 입력하기 위해 전처리한다. 전처리 과정은 웹캠의 640×480×3, RGB 이미지 데이터를 YCrCb 색공간으로 변환해주는데, YCrCb는 RGB의 인코딩 방식 중 하나로 RGB와 비슷한 색공간을 가진다. YCrCb 색

상은 휘도 성분인 Y, 색차 성분인 Cr, Cb를 (Y, Cr, Cb)로 표현하는 것이 가능하며, 표 4와 같이 피부색 범위에 해당하는 Lower부터 Upper까지의 범위로 설정하여 검출하면, 손에 해당하는 부분에 경계선이 생긴다. 이 경계선을 따라 타원 모양의 매트릭스를 생성하고, 가우시안 블러링을 거쳐 노이즈를 제거한다.

표 4. 피부색 검출이 가능한 YCrCb 색상 범위
 Table 4. YCrCb color range with skin color detection

Color	Y	Cr	Cb
Lower	0	133	77
Upper	255	173	127

이렇게 만들어진 경계선 컨투어의 시작점, 폭, 높이 좌표에 손 이외에 좀 더 넓은 범위의 데이터를 얻기 위해 컨투어의 폭과 높이에 비례하는 값을 좌표에 더한 뒤, 데이터를 잘라내고 크기를 조절하여 손에 해당하는 범위의 64×64×3, RGB 데이터를 분류기에 입력하였다.

3.3 CNN 모델

제안하는 시스템에서는 합성곱 형태의 CNN 모델 중에서 이미지를 분류하는 성능에 있어 높은 평가를 받은 ResNet-50 모델을 사용하였으며 모델의 구조는 그림 2에 포함되어있다. 이 모델은 기존의 layer에 입력된 학습 정보를 보존해 다음 layer에도 학습시켜 각각의 layer가 학습할 양을 줄여주는 residual connection 방법과 입력값과 출력값의 같은 차원을 유지해주는 1×1 conv filter, 그리고 활성화 함수로 ReLU(Rectified Linear Unit)를 사용하였다. 이를 통해 기존의 많은 layer가 포함된 CNN 모델에서 학습을 진행할 때 기울기 값이 서서히 감소하여 학습이 어려워지는 vanishing gradients 문제를 해결하였다[13].

모델의 구현을 위해서 64×64×3 RGB 이미지 데이터가 학습에 사용되었고, batch size는 16으로 설정하였다. 입력된 데이터는 각 1개씩의 1×1, 3×3, 1×1의 conv filter로 이루어진 bottle neck 구조의 layer와 BN(Batch Normalization), ReLU의 연산을 거쳐 모델에 학습되며 총 100 epochs 실시하여 분류기를 생성하였다. 학습에 사용된 함수로 손실 함수는 Cross-Entropy Loss, 최적화 함수는 SGD(Stochastic Gradient Descent)를 사용하였다. 최적화 함수의 매개 변수는 learning rate를 1×10^{-3} , 손실 값의 변화에 관성을 주어 최솟값에 도달하기 위해 momentum을 9×10^{-1} , 특정 가중치 값이 커져 새로운 데이터를 예측 못 하는 오버피팅 현상을 줄이기 위해 weight decay를 1×10^{-4} 로 설정하였다.

IV. 실험

4.1 학습 모델(분류기) 분류 정확도

제안하는 시스템에서 사용된 분류기를 평가하기 위해 학습에 사용되지 않은 데이터를 손동작 종류별 50개씩 테스트 데이터로 사용하여 분류기의 정확도를 확인하였다. 결과는 그림 7과 같이 정확도가 99.2%로 측정되었으며, 그림 8과 같이 Down과 Right 동작에서 주먹 부분의 위치와 색이 Grab과 유사하여 학습 모델이 50개 중에서 1개씩 Grab으로 분류한 것이다.

4.2 실시간 로봇 제어 정확도 및 지연 시간

제안하는 시스템은 임베디드 보드 환경에서 분류된 손동작 데이터를 이용해 UR3 로봇과 그리퍼를 제어하는 것이다. 이에 손동작 종류별로 50회씩 실시간으로 로봇이 알맞은 제어를 수행하는지에 대한 실험과 로봇의 각 관절에 제어 값이 입력되는 시점과 로봇이 작동되는 시점 사이의 지연을 확인하는 실험을 진행하였다.

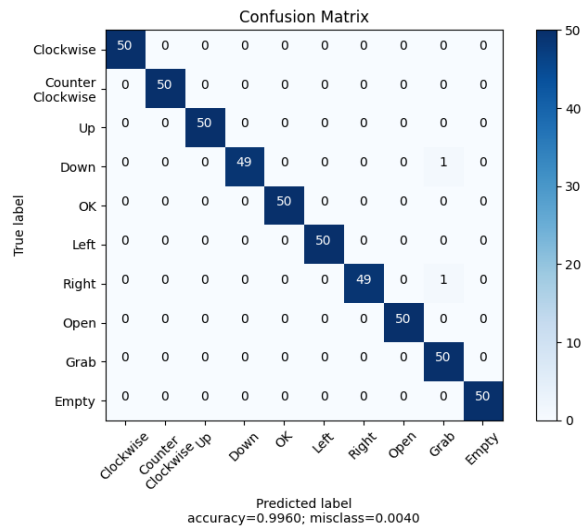
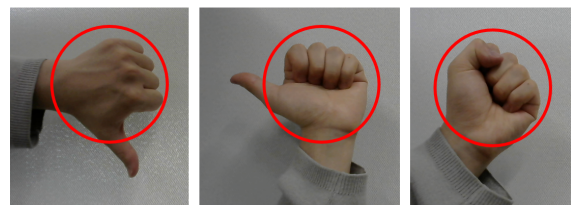


그림 7. 학습 모델의 confusion matrix
Fig. 7. Confusion matrix of the trained model



Down Right Grab

그림 8. 3가지 유사한 손동작
Fig. 8. Confusion matrix of the trained model

결과는 표 5와 같이 Down, Left, Right, Grab, Empty 손동작에서 가장 높은 100%, Counterclockwise 손동작에서 가장 낮은 90%의 정확도를 보였으며, 총 500회의 실험을 통해 97%의 정확도를 보였다. 손동작별로 정확도가 다른 이유는 손동작을 변경하는 도중에 촬영된 이미지 데이터가 분류기에 입력되어 정확한 분류가 되지 않았기 때문이다.

표 5. 제안하는 시스템의 실시간 제어 정확도
Table 5. Real-time control accuracy of the proposed system

Hand gesture	Success / Fail	Accuracy (%)
Clockwise	47 / 3	94
Counterclockwise	45 / 5	90
Up	49 / 1	98
Down	50 / 0	100
Ok	46 / 4	92
Left	50 / 0	100
Right	50 / 0	100
Open	48 / 2	96
Grab	50 / 0	100
Empty	50 / 0	100
Total	485 / 15	97

웹캠에서 얻은 이미지를 전처리하는 과정에서 약 24ms, 학습 모델의 분류에서 약 9ms, 분류된 출력

데이터를 소켓 통신으로 로봇을 제어하는 ROS 패키지 부분에 전달하는 데 약 40ms의 지연이 발생하며, ROS의 메시지 통신으로 로봇에 제어 값이 입력되는 시점과 로봇이 작동되는 시점 사이의 지연은 그림 9와 같이 1.4ms로 UR3 로봇 자체의 지연이다.

전체 시스템에서 발생하는 지연은 총 74.4ms로 임베디드 보드 환경에서도 매우 빠른 처리 속도를 가지며, 실제 로봇으로 제어 정확도를 측정하는 실험에서 영향을 끼치지 않는 것을 확인하였다.

V. 결 론

본 논문에서는 임베디드 보드 환경에서 총 10종류의 손동작이 학습된 분류기에 웹캠의 이미지 데이터를 입력하여 손동작을 분류하고 ROS의 메시지 통신을 이용해 UR3 로봇과 그리퍼를 실시간으로 제어하는 시스템을 구현하였다. 4장 실험을 통해 임베디드 보드 환경에서 분류기 모델의 손동작 분류 정확도는 99.2%, 실시간 제어 정확도는 97%의 성능을 보이며, 로봇에 제어 값이 입력되는 시점과 로봇이 작동하는 시점 사이에 1.4ms 정도의 지연을 보인다. 따라서 제안하는 시스템은 임베디드 보드 환경에서도 높은 정확도와 빠른 처리 속도를 갖는다.

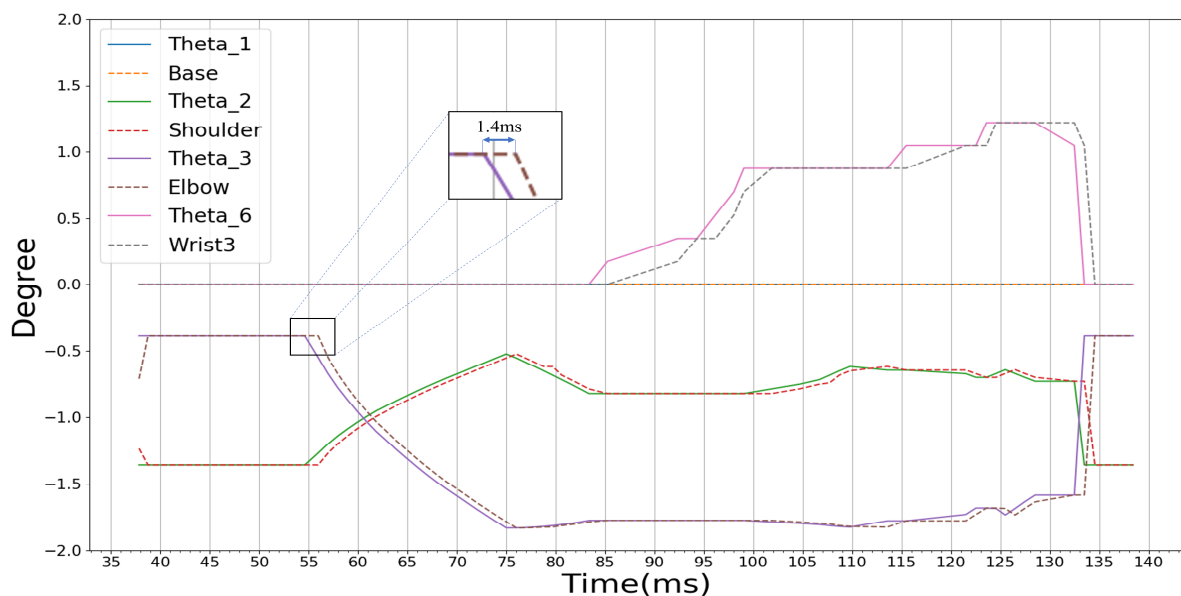


그림 9. 로봇에 제어 값이 입력되는 시점과 로봇이 작동되는 시점 사이의 지연
Fig. 9. Delay between when the control value is inputted into the robot and when the robot is operated

추후 시각적으로 유사한 손동작도 정확하게 분류하는 학습 모델을 생성하고 정적인 손동작 이미지가 아닌 동적인 손동작 영상을 사용해 손동작을 취하는데 제약이 없도록 개선하는 연구를 진행할 것이다.

References

- [1] Nuno Mendes, João Ferrer, João Vitorino, Mohammad Safeea and Pedro Neto, "Human Behavior and Hand Gesture Classification for Smart Human-robot Interaction", *Procedia Manufacturing*, Vol 11, pp. 91-98, 2017. <https://doi.org/10.1016/j.promfg.2017.07.156>.
- [2] Y. Yu, X. Wang, Z. Zhong, and Y. Zhang, "ROS-based UAV control using hand gesture recognition", 2017 29th Chinese Control And Decision Conference (CCDC), Chongqing, China, pp. 6795-6799, May 2017. <https://doi.org/10.1109/CCDC.2017.7978402>.
- [3] A. Sylari, B. R. Ferrer, and J. L. M. Lastra, "Hand Gesture-Based On-Line Programming of Industrial Robot Manipulators", 2019 IEEE 17th International Conference on Industrial Informatics (INDIN), Helsinki, Finland, pp. 827-834, Jul. 2021. <https://doi.org/10.1109/INDIN41052.2019.8972301>.
- [4] C. Yang, C. Zeng, P. Liang, Z. Li, R. Li, and C. Su, "Interface Design of a Physical Human-Robot Interaction System for Human Impedance Adaptive Skill Transfer", in *IEEE Transactions on Automation Science and Engineering*, Vol. 15, No. 1, pp. 329-340, Jan. 2018. <https://doi.org/10.1109/TASE.2017.2743000>.
- [5] Se-Yun Jeon, Eun-Su Kim, and Bum-Yong Park, "CNN-based Hand Gesture Recognition Method for Teleoperation Control of Industrial Robot", *IEMEK Journal of Embedded Systems and Applications*, Vol. 16, No. 2, pp. 65-72, Apr. 2021. <https://doi.org/10.14372/IEMEK.2021.16.2.65>.
- [6] Yuntao Ma, Yuxuan Liu, Ruiyang Jin, Xingyang Yuan, Raza Sekha, Samuel Wilson, and Ravi Vaidyanathan, "Hand gesture recognition with convolutional neural networks for the multimodal UAV control", 2017 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED-UAS), Sweden, pp. 198-203, Oct. 2017. <https://doi.org/10.1109/RED-UAS.2017.8101666>.
- [7] O. Mazhar, S. Ramdani, B. Navarro, R. Passama, and A. Cherubini, "Towards Real-Time Physical Human-Robot Interaction Using Skeleton Information and Hand Gestures", 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, pp. 1-6, Oct. 2018. <https://doi.org/10.1109/IROS.2018.8594385>.
- [8] Y. Jhang, Y. Chu, T. Tai, W. Hwang, P. Cheng, and C. Lee, "Sensor Based Dynamic Hand Gesture Recognition by PairNet", 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), Atlanta, GA, USA, pp. 994-1001, Jul. 2019. <https://doi.org/10.1109/iThings/GreenCom/CPSCom/SmartData.2019.00174>.
- [9] Y. Maret, D. Oberson, and M. Gavrilova, "Real-Time Embedded System for Gesture Recognition", 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, pp. 30-34, Oct. 2018. <https://doi.org/10.1109/SMC.2018.00014>.
- [10] Y. Lei, W. Hongpeng, T. Dianxiong, and W. Jue, "A real-time hand gesture recognition algorithm for an embedded system", 2014 IEEE International Conference on Mechatronics and Automation, Tianjin, China, pp. 901-905, Aug. 2014. <https://doi.org/10.1109/ICMA.2014.6885817>.
- [11] M. M. U. Atique and M. A. R. Ahad, "Inverse Kinematics solution for a 3DOF robotic structure using Denavit-Hartenberg Convention", 2014 International Conference on Informatics, Electronics

& Vision (ICIEV), Dhaka, Bangladesh, pp. 1-5, May 2014. <https://doi.org/10.1109/ICIEV.2014.6850854>.

[12] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Ng, "ROS: an open-source Robot Operating System", ICRA Workshop on Open Source Software, Vol. 3, Jan. 2009.

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition", IEEE Conf. Computer Vision and Pattern Recognition, pp. 770-778, 2016. <https://doi.org/10.1109/cvpr.2016.90>.

저자소개

김 충 근 (Chung-Geun Kim)



2021년 2월 : 금오공과대학교
전자공학부 정보전자전공(공학사)
2021년 3월 ~ 현재 :
금오공과대학교 전자공학과
석사과정
관심분야 : 지능형로봇 및 제어,
딥러닝, ROS, 인간-로봇

상호작용(HRI) 기술

김 은 수 (Eun-Su Kim)



2020년 2월 : 금오공과대학교
전자공학부 정보전자전공(공학사)
2020년 3월 ~ 현재 :
금오공과대학교 전자공학과
석사과정
관심분야 : 지능형로봇 및 제어,
딥러닝, ROS, 인간-로봇

상호작용(HRI) 기술

신 재 욱 (Jae-Wook Shin)



2008년 2월 : 경북대학교
전자전기컴퓨터공학부(공학사)
2010년 2월 : 포항공과대학교
전자전기공학과(공학석사)
2014년 8월 : 포항공과대학교
전자전기공학과(공학박사)
2014년 3월 ~ 2016년 12월 :

삼성전자 무선사업부 책임연구원

2017년 2월 ~ 2021년 2월 : 순천향대학교

의용메카트로닉스공학과 조교수

2021년 3월 ~ 현재 : 금오공과대학교 전자공학부 조교수

관심분야 : Adaptive filter, 음향신호처리, 생체신호처리,
임베디드 시스템

박 범 용 (Bum-Yong Park)



2009년 2월 : 경북대학교
전자전기컴퓨터공학부(공학사)
2011년 2월 : 포항공과대학교
전자전기공학과(공학석사)
2015년 8월 : 포항공과대학교
전자전기공학과(공학박사)
2017년 8월 : 삼성전자

생산기술연구소 선임연구원

2017년 9월 ~ 현재 : 금오공과대학교 전자공학부,

IT융복합공학과 조교수

관심분야 : Robust control, robot manipulator system,

신호 처리, 임베디드 제어 시스템