

엣지 디바이스인 소셜 로봇에서의 영상 딥러닝을 위한 모듈 교체형 인공지능 서버 설계 및 개발

강아름*, 오현정*, 김도연*, 정구민**

Design and Development of Modular Replaceable AI Server for Image Deep Learning in Social Robots on Edge Devices

A-Reum Kang*, Hyun-Jeong Oh*, Do-Yun Kim*, Gu-Min Jeong**

요약 본 논문에서는 인공지능 블록을 구동할 수 있도록 Edge Device와 서버를 분리하는 영상 딥러닝용 모듈 교체형 인공지능 서버의 설계와 데이터 송수신 방법을 제시한다. 영상 딥러닝용 모듈 교체형 인공지능 서버를 통해 소셜 로봇과 로봇의 플랫폼이 구동될 Edge Device 간의 종속성을 줄여 구동 안정성을 향상할 수 있다. 사용자가 소셜 로봇과의 상호작용을 위해서 인공지능 서버에 기능을 요청하면 모듈화된 기능들을 이용해 결과만을 반환받을 수 있다. 인공지능 서버에서 모듈화되어있는 기능들은 서버 관리자에 의해 모듈별로 유지 보수 및 변경이 쉽게 가능하다. 기존 서버 시스템과 비교했을 때 모듈 교체형 인공지능 서버는 수행되는 프로그램의 규모 차이와 서버 유지 보수 면에서 더 효율적인 성능을 낸다. 이를 통해 사람-로봇 간의 상호작용이 가능한 로봇 시나리오에 더 다양한 영상 딥러닝을 포함 시킬 수 있으며, 로봇 플랫폼 외에 영상 딥러닝을 위한 인공지능 서버에 적용할 때 더 효율적인 성능을 낼 수 있다.

Abstract In this paper, we present the design of modular replaceable AI server for image deep learning that separates the server from the Edge Device so as to drive the AI block and the method of data transmission and reception. The modular replaceable AI server for image deep learning can reduce the dependency between social robots and edge devices where the robot's platform will be operated to improve drive stability. When a user requests a function from an AI server for interaction with a social robot, modular functions can be used to return only the results. Modular functions in AI servers can be easily maintained and changed by each module by the server manager. Compared to existing server systems, modular replaceable AI servers produce more efficient performance in terms of server maintenance and scale differences in the programs performed. Through this, more diverse image deep learning can be included in robot scenarios that allow human-robot interaction, and more efficient performance can be achieved when applied to AI servers for image deep learning in addition to robot platforms.

Key Words : AI server, Artificial Intelligence Server, Image Processing, Robot Interaction, Robot Platform

1. 서론

초심자를 위한 블록형 프로그래밍에 대한 필요성

이 증가했으며, 인공지능 기술을 다양한 서비스에 적용하기 위한 플랫폼으로서의 서비스가 개발되고

This paper is a research conducted by the government (Ministry of Trade, Industry and Energy) in 2020 with support of the core technology development project of the robotics industry (No. 10080615). and This research was supported by the National Research Foundation of Korea (NRF) Grant funded by the Korean Government (MSIP)(NRF-2018R1D1A1A09083894).

*Electronic Engineering, Kookmin University

**Corresponding Author : Electronic Engineering, Kookmin University (gm1004@kookmin.ac.kr)

Received October 10, 2020

Revised October 14, 2020

Accepted November 02, 2020

있다. 본 논문에서 사용하는 교육용 블록형 프로그래밍 플랫폼 MOCCA Studio[1]에서는 인공지능 기술에 더불어 로봇과 상호작용이 가능하다. 로봇과 사람이 상호작용을 할 수 있는 플랫폼과[2][3] AI 블록을 제공하는 블록 코딩 툴[4]이 개발되고 있는데, MOCCA Studio에서는 인공지능을 활용한 사람-로봇 간의 상호작용 기능을 둘 다 제공한다. 본 논문에서는 이를 소셜 로봇으로 사용한다. 사람과 로봇과의 상호작용은 카메라와 로봇, 인공지능 블록을 구동할 인공지능 서버, 인공지능 블록을 이용한 블록형 프로그래밍 등을 통해서 이루어진다.

인공지능 발달에 따라서 인공지능을 이용한 클라우드 연구가 활발히 진행되고 있는데, 이는 주로 스마트팩토리의 Edge Cloud 연산을 위한 연구[5][6]로 그 목적이 다르다. 본론에서 제시하는 모듈 교체형 인공지능 서버를 사용해 소셜 로봇과 로봇의 플랫폼이 구동될 Edge Device(모바일 장치를 비롯한 smart device) 간의 종속성을 줄이고, Edge Device에서 플랫폼을 구동할 성능이 부족한 상황을 방지해 구동 안정성을 향상할 수 있다.

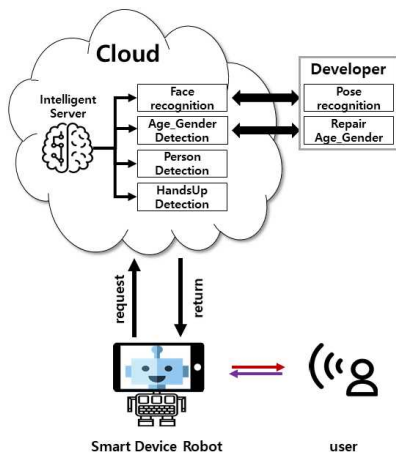


그림 1. 인공지능 서버의 운용 개요도
Fig. 1. Overview diagram of AI servers

본 논문에서는 지능 블록을 구동할 수 있도록 Edge Device와 서버를 분리하는 모듈 교체형 인공지능 서버의 설계와 데이터 송수신 방법을 제시한다. 그림 1에서 전체적인 서버 운용 방법을 확인할 수 있다. 사용자가 소셜 로봇과의 상호작용을 위해

서 인공지능 서버에 기능을 요청하면 모듈화된 기능들을 이용해 결과를 반환받을 수 있다. 이때 인공지능 서버에서 모듈화되어있는 기능들은 서버 관리자에 의해 모듈별로 유지 보수 및 변경이 쉽게 가능하다.

실험을 통해 서버에서 지능 블록 연산을 수행한 후 다시 로봇 플랫폼으로 돌아오는 시간을 측정한다.

2. 본론

본론에서는 로봇 플랫폼에서 사용할 수 있는 영상처리 지능 블록의 기능을 설명하고, 그 블록의 연산을 수행하는 인공지능 서버의 구현 방법과 데이터 송수신 방법에 대해 서술한다.

지능 블록은 후술할 여러 라이브러리를 활용하여 개발하였다. 지능 블록들은 인공지능 서버에서 구동할 수 있다. 일반적으로 인공지능을 이용한 동작은 HW의 성능에 따라 수행 시간이 달라지는 문제를 동반한다. 따라서 본 논문에서는 영상 딥러닝을 위한 모듈 교체형 인공지능 서버를 통해 지능 블록의 구동 시간을 감소시키는 방법을 제시한다. 이를 통해 Edge Device에서 프로그램 구동 성능을 향상할 수 있다.

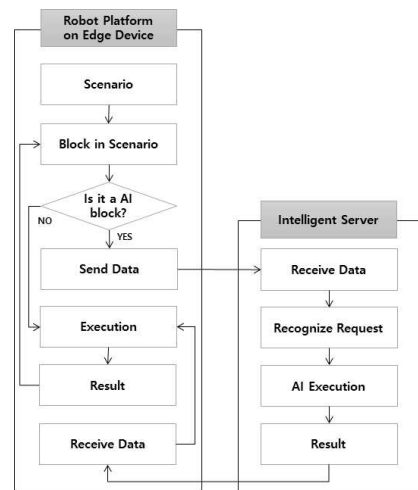


그림 2. 로봇 플랫폼이 탑재된 장치와 인공지능 서버 간의 흐름도
Fig. 2. Flowchart Between Robot Platform on Edge Device and Intelligent Server

그림 2에서 로봇 플랫폼이 구동될 Device와 인공지능 서버 간의 흐름도를 확인할 수 있다. 로봇 플랫폼의 상호작용 시나리오에 지능 블록이 포함되어 있다면, 인공지능 서버로 데이터를 송신해 모듈 연산을 수행하고 예측 결과를 로봇 플랫폼으로 반환해 시나리오를 진행한다.

다음에서 기존 서버 시스템과 영상 딥러닝 모듈 교체형 서버의 차이를 서술한다. 이어서 각 인공지능 블록에 대한 기능 설명을 진행하고, 인공지능 서버를 개발하고 데이터를 송수신하는 방법과 그 흐름에 관해 설명한다.

2.1 기존 서버 시스템과 모듈 교체형 인공지능 서버의 차이

기존 서버 시스템과 모듈 교체형 서버의 차이는 크게 수행되는 프로그램의 규모 차이와 서버 유지 보수 면에서 두드러진다.

먼저 수행 프로그램의 규모 관점에서 차이를 살펴본다. 기존 서버 시스템의 경우는 서버 기능 요청 시 요청받은 기능에 대해서 모듈화가 되어있지 않기 때문에 서버 프로그램 전체를 실행시켜 적절한 결과를 반환해야 한다. 반면 모듈 교체형 서버는 미리 구현된 기능들에 대해 모듈화가 되어있기 때문에 서버 기능 요청 시 요청받은 기능에 대한 모듈만 실행하게 되어 프로그램 구동 시간을 효율적으로 감소시키고 필요한 기능에 대한 메모리만 할당할 수 있다. 그림 3을 참고하면 서버에 기능 요청 시 구동되는 프로그램의 범위를 간략하게 살펴볼 수 있다.

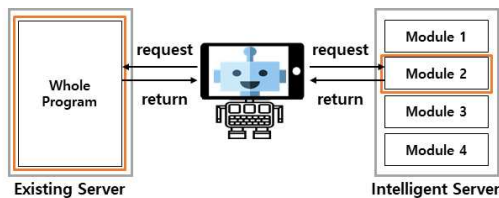


그림 3. 기존 서버 시스템과 모듈 교체형 인공지능 서버의 수행 프로그램 규모 관점에서의 차이
Fig. 3. Differences in the size of programs by existing server and modular replaceable AI servers

다음으로 서버 유지보수 관점에서의 차이를 살펴본다. 기존 서버 시스템의 경우는 전체 프로그램이 모듈화가 되어있지 않기 때문에, 기능과 변수들을 완벽하게 분리하기 힘들다. 따라서 서버를 유지 보수 할 때 기능별로 다른 방식으로 보수를 진행해야 하며, 다른 블록과의 종속성도 고려해야 하는 등 유지 보수 시 복잡성이 증가한다. 반면 모듈 교체형 서버는 모듈화가 되어있는 상태이기 때문에 어떤 모듈을 유지보수 하더라도 다른 모듈과는 독립적으로 모듈을 교체하거나 수리할 수 있다. 그림 4를 살펴보면 기존 서버 시스템의 경우 더 복잡한 형태의 서로 다른 모양의 프로그램 조각을 보수해야 한다는 것을 간략하게 살펴볼 수 있다.

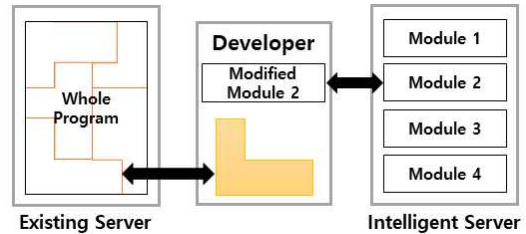


그림 4. 기존 서버 시스템과 모듈 교체형 인공지능 서버의 프로그램 유지보수 관점에서의 차이
Fig. 4. Differences in the program maintenance by existing server and modular replaceable ai servers

2.2 소셜 로봇 플랫폼에 사용되는 인공지능 블록

인공지능 블록은 yolov3[7], openpose[8], face_recognition 1.3.0[9] 같은 오픈 라이브러리와 직접 학습한 모델을 통해 개발되었다.

인공지능 블록의 종류는 'Face Recognition', 'Age, Gender Detection', 'Person_Detection', 'HandsUp Detection'이 있으며 본 논문에서는 각 블록이 인공지능 서버에서 어느 정도의 수행 시간을 갖는지 측정하는 데에 사용된다.

Face_Recognition 블록은 Python의 face_recognition 라이브러리를 통해 개발되었으며, 저장된 얼굴 이미지를 통해서 얼굴의 대상을 인식할 수 있다. Recognize_Age_Gender 블록은 오픈 데이터셋을 통해 연령대와 성별을 직접 학습

시킨 모델을 기반으로 사람의 연령대와 성별을 예측한다. Detect_People 블록은 yolov3를 기반으로 개발되었으며, 사람의 수와 위치를 추정하는 블록이다. Recognize_HandsUp 블록은 openpose를 기반으로 사용자가 손을 들었는지를 판별해 손 든 사람의 이미지를 반환하는 블록이다.

2.3 영상 딥러닝용 모듈 교체형 인공지능 서버 개발

모듈화된 영상 딥러닝 기능을 수행해 결과를 반환하고 모듈 교체가 가능한 인공지능 서버를 개발했다. 본 서버는 로봇 플랫폼의 backend에서 IP를 미리 할당된 상태에서 작동하게 된다. 2.1의 인공지능 블록들이 Python을 통해 구현되었기 때문에 동일 언어로 개발하였다. 모듈화된 인공지능 블록을 사용하기 위해서, 사용되는 로봇 플랫폼과 인공지능 서버와의 프로토콜을 맞춰 실행할 수 있다.

다음으로 인공지능 서버상에서의 구동 방식과 서버를 사용할 로봇 플랫폼상의 서버 구동 방식과 구현 방법에 대해 설명한다.

2.3.1 소셜 로봇 플랫폼 상의 서버 운용

로봇 플랫폼에서 블록을 통해서 backend processing을 진행할 수 있는데, 이때 블록이 인공지능 블록이라면 서버를 통한 처리가 이루어지고 반환

값을 로봇 플랫폼에 전달하게 된다. 그림 5를 통해서 로봇 플랫폼에서 플랫폼과 인공지능 서버와의 데이터 송수신 흐름도를 확인할 수 있다.

인공지능 블록일 경우의 처리는 IP에 더불어 서버에서 각 블록에 맞는 동작을 할 수 있는 keyword를 같이 연결하고 데이터를 전송하게 된다. 이를 통해 서버에서 알맞은 모듈을 실행해 연산할 수 있다. 로봇 플랫폼의 backend에서는 각 인공지능 블록에서 요구하는 파라미터들을 충족시키기 위한 처리를 진행한다.

그 파라미터들을 서버로 송신하기 위해서는 데이터에 따라서 적절한 처리가 필요하다. String, Integer와 같은 기본적인 데이터의 경우 json 형태로 변환하여 서버로 송신하게 된다. 그러나 이미지의 경우는 byte 형태인 png format으로 바꾸는 인코딩 과정이 필요하며 이 경우에도 json 형태로 인공지능 서버로 송신한다.

인공지능 서버에서 반환하는 데이터는 동일하게 json 형태이며 String 형태로 변환해서 사용할 수 있다. 이때 Image 데이터도 String 형태로 반환받게 되며, 이 데이터를 다시 서버에 송신해 추가적인 지능 연산을 진행할 수도 있다.

2.3.2 인공지능 서버 상의 서버 운용

로봇 플랫폼에서 연결된 keyword에 따라 인공지능 서버에서는 그에 맞는 함수를 실행하고 그에 대한 반환 값을 json 형태로 다시 로봇 플랫폼으로 송신하게 된다. 그림 6을 통해서 인공지능 서버에서 로봇 플랫폼과 인공지능 서버와의 데이터 송수신 흐름도를 확인할 수 있다.

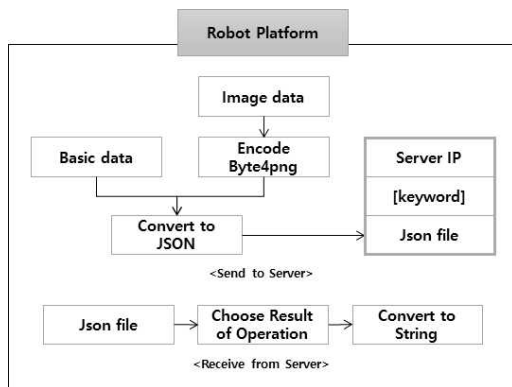


그림 5. 로봇 플랫폼에서 플랫폼과 인공지능 서버와의 데이터 송수신 흐름도

Fig. 5. Flowchart of Data Transmission and Reception Between Robot Platform and Intelligent Server in Robot Platform

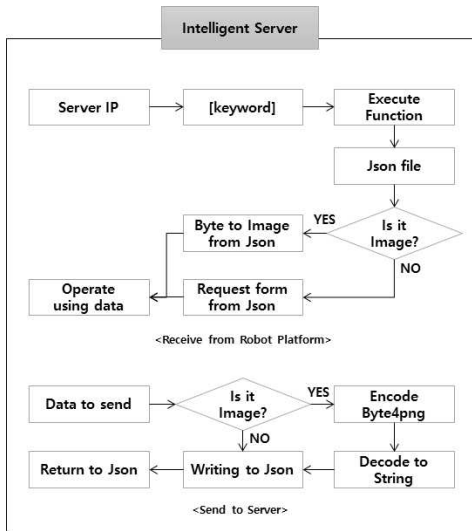


그림 6. 인공지능 서버에서 로봇 플랫폼과 인공지능 서버와의 데이터 송수신 흐름도

Fig. 6. Flowchart of Data Transmission and Reception Between Robot Platform and Intelligent Server in Intelligent Server

서버에서는 연결한 keyword에 따라서 flask method에 맞추어 작동하게 된다. json 형태로 입력 받은 argument를 get해서 사용하게 되는데, 이미지의 경우는 로봇 플랫폼에서처럼 추가적인 과정을 거치게 된다. byte 형태로 전송받은 데이터를 다시 image 형태로 변환하여 원본 이미지로 복원한다. 원본 이미지를 통해서 기존에 구성한 각 함수의 입력값으로 사용할 수 있다. 복원된 이미지를 통해서 호출된 명령에 맞는 연산을 진행 후 그 결과값을 다시 json 형태로 변환하여 로봇 플랫폼으로 전달한다.

3. 실험 및 결과

제시한 방법으로 영상 딥러닝용 모듈 교체형 인공지능 서버를 구현하고, Edge Device에서 인공지능 서버로 연산을 요청하고 결과를 반환받는 데에 걸리는 시간을 측정했다. 이때의 연산은 2.1에서 소개한 각 지능 블록을 구동한 것이다.

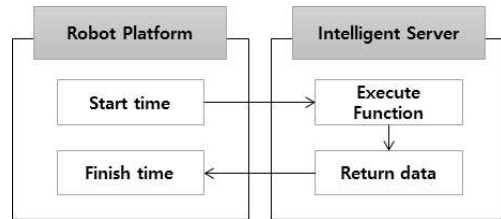


그림 7. 로봇 플랫폼에서 인공지능 서버 수행 시간 측정 흐름도
Fig. 7. Flowchart of Measuring Intelligent Server Operation Time in Robot Platform

그림 7에서 수행 시간 측정 흐름도를 확인할 수 있다. 로봇 플랫폼에서 인공지능 서버로 keyword와 Json File을 보내기 직전에 time을 시작해 인공지능 서버에서 keyword에 따라서 해당 모듈을 실행한 뒤 결과를 다시 로봇 플랫폼으로 반환하고 time을 종료한다. 인공지능 서버로부터 데이터를 수신 한 시간과의 차이를 통해서 수행 시간을 측정했다.

표 1. 인공지능 블록을 수행하고 결과를 반환하는 데에 소요된 시간

Table 1. Time spent executing AI blocks and returning results

Intelligence Block	Operation Time in Server
Face_Recognition	2.27 sec
Age_GenderDetection	2.29 sec
Person_Detection	2.26 sec
HandsUpDetection	2.3 sec

측정한 수행 시간은 표 1에서 확인할 수 있다. 여기서 수행 시간 전부 2초대인 것을 확인할 수 있는데, Webcam 기능을 사용할 때 발생하는 대기 시간을 포함한 것이다. 따라서 서버에서 지능 블록을 연산하고 로봇 플랫폼으로 값을 반환할 때 소요되는 시간은 0.26~0.3 sec이다. 기존 서버 시스템을 이용했을 때 보다 평균 0.5 sec 이상 빠른 성능을 보였다. 이는 인공지능 서버는 모듈화됨에 따라 기능에 해당하는 모듈 블록만을 실행했으며, 기존 서버 시스템은 전체 프로그램을 실행해야 하기 때문이다.

4. 결론

영상 딥러닝용 모듈 교체형 인공지능 서버를 로봇 플랫폼에 적용해 사용자 Edge Device에서 지능 블록을 이용한 시나리오를 제작하고 콘텐츠를 구동할 수 있다. 인공지능 서버를 활용했기 때문에, Edge Device에서 지능 연산을 수행할 때 발생할 수 있는 Performance 부족 문제를 방지하고, 빠른 연산 결과를 받아 콘텐츠의 실시간성을 향상할 수 있다. 다만 제안하는 모듈 교체형 인공지능 서버의 경우, 서버 기능을 동시에 다중으로 호출할 경우 호출 명령어가 복잡해져 로봇 플랫폼의 프로그램 복잡도가 상승할 수 있다.

이런 방식으로 서버를 구현하면, 다른 플랫폼에서도 Edge Device와 연산 서버를 분리해 프로그램의 성능을 향상하기 위한 인공지능 서버를 구현할 수 있다. 특히 많은 연산을 요구하는 영상처리 인공지능에 효과적일 것으로 생각한다.

실험을 통해서 로봇 플랫폼에서 인공지능 서버에서의 연산을 수행하고 돌아오는 시간을 각 인공지능 블록마다 측정했다. Webcam 대기 시간을 제외하면 소요 시간은 약 2.8 sec였다.

추후의 연구에서는 서버에서의 구동 시간뿐만 아니라 로봇 플랫폼상에서의 연산 속도를 개선하고, Webcam 대기 시간을 줄여 전체적인 구동 시간을 개선하겠다.

REFERENCES

[1] Dong-Hoon Lee, Seyun Jang, Hyekyung Cho, "MOCCA studio: a graphical tool for high-level programming of human-robot social interaction", HRI '19: Proceedings of the 14th ACM/IEEE International Conference on Human-Robot Interaction, pp. 620-621, March, 2019.

[2] Khandelwal, Piyush, Shiqi Zhang, Jivko Sinapov, Matteo Leonetti, Jesse Thomason, Fangkai Yang, Ilaria Gori, "BWIBots: A Platform for Bridging the Gap between AI and Human-Robot Interaction Research.", The International Journal of Robotics

Research 36, no. 5-7 : 635-59, June 2017

[3] M. A. Salichs, "Maggie: A Robotic Platform for Human-Robot Social Interaction," 2006 IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, pp. 1-7, 2006

[4] M. Hauck, R. Machhmer, L. Czenkusch, K. Gollmer and G. Dartmann, "Node and Block-Based Development Tools for Distributed Systems With AI Applications," in IEEE Access, vol. 7, pp. 143109-143119, 2019

[5] Sukhpal Singh Gill, Shreshth Tuli, Minxian Xu, Inderpreet Singh, Karan Vijay Singh, Dominic Lindsay, Shikhar Tuli, Daria Smirnova, Manmeet Singh, Udit Jain, Haris Pervaiz, Bhanu Sehgal, Sukhwinder Singh Kaila, Sanjay Misra, Mohammad Sadegh Aslanpour, Harshit Mehta, Vlado Stankovski, Peter Garraghan, "Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing", Evolution, vision, trends and open challenges. CoRR abs/1911.01941, 2019

[6] A. H. Sodhro, S. Pirbhulal and V. H. C. de Albuquerque, "Artificial Intelligence-Driven Mechanism for Edge Computing-Based Industrial Applications," in IEEE Transactions on Industrial Informatics, vol. 15, no. 7, pp. 4235-4243, July 2019

[7] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016

[8] CMU-Perceptual-Computing-Lab. OpenPose [Internet]. Available: <https://github.com/CMU-Perceptual-Computing-Lab/openpose>

[9] face-recognition[Internet]. Available: https://github.com/ageitgey/face_recognition

저자약력

강 아 름(A-Reum Kang)

[일반회원]



- 2020년 국민대학교 전자공학과 학사
- 2020년~현재 국민대학교 일반대학원 전자공학과 스마트임베디드시스템 연구실 석사과정

〈관심분야〉 차량용 소프트웨어, 인공지능, 영상처리, 패턴인식, 자율주행

오 현 정(Hyun-Jeong Oh)

[일반회원]



- 2020년 국민대학교 전자공학과 학사
- 2020년~현재 국민대학교 일반대학원 전자공학과 스마트임베디드시스템 연구실 석사과정

〈관심분야〉 차량용 소프트웨어, 인공지능, 영상처리, 패턴인식, 자율주행

김 도 연(Do-Yun Kim)

[일반회원]



- 2019년 국민대학교 전자공학과 학사
- 2019년~현재 국민대학교 일반대학원 전자공학과 스마트임베디드시스템 연구실 석사과정

〈관심분야〉 차량용 소프트웨어, 인공지능, 딥러닝, 영상처리, 패턴인식

정 구 민(Gu-Min Jeong)

[일반회원]



- 1995년 서울대학교 제어계측공학과 공학사
- 1997년 서울대학교 제어계측공학과 공학석사
- 2001년 서울대학교 전기컴퓨터공학부 공학박사
- 2001년~2004년 (주)네오엠텔 기반기술팀장
- 2004년~2005년 SK텔레콤 터미널 개발팀 과장
- 2005년~현재 국민대학교 전자공학부 교수
- 2011년~2013년 2월 UC Irvine 방문교수
- 2013년~현재 국민대학교 인피니언센터, 센터장
- 2015년~현재 국민대학교 현대오딘센터, 센터장
- 2015년~현재 국민대학교 사물인터넷연구센터, 센터장
- 2013~현재 유비벨록스 사외이사
- 2019~현재 휴맥스 사외이사

〈관심분야〉 차량용 마이컴, 차량용 소프트웨어, 커넥티드카, 자율주행