

# 인공지능 기반 서비스 로봇을 위한 영상처리 프로세서 설계

문지윤\* · 김수민\*\*

## Image Processing Processor Design for Artificial Intelligence Based Service Robot

Ji-Youn Moon\* · Soo-Min Kim\*\*

### 요 약

다양한 분야에 서비스 로봇이 적용됨에 따라 각 임무에 적합한 영상처리 알고리즘을 빠르고 정확하게 수행할 수 있는 영상처리 프로세서에 관한 관심이 높아지고 있다. 본 논문에서는 로봇에 적용 가능한 영상처리 프로세서 설계방법을 소개한다. 제안한 프로세서는 CPU, GPU, FPGA가 융합된 형태로 AGX 보드, FPGA 보드, LiDAR-Vision 보드, Backplane 보드로 구성된다. 제안한 방법은 시뮬레이션 실험을 통해 검증한다.

### ABSTRACT

As service robots are applied to various fields, interest in an image processing processor that can perform an image processing algorithm quickly and accurately suitable for each task is increasing. This paper introduces an image processing processor design method applicable to robots. The proposed processor consists of an AGX board, FPGA board, LiDAR-Vision board, and Backplane board. It enables the operation of CPU, GPU, and FPGA. The proposed method is verified through simulation experiments.

### 키워드

Artificial Intelligence, Service Robot, Image Processing, CPU, GPU, FPGA, Parallel Processing  
인공지능, 서비스 로봇, 영상처리, CPU, GPU, FPGA, 병렬 처리

## 1. 서 론

로봇이 임무를 수행하는 데 있어 vision 센서를 이용한 환경 정보 이해는 중요하다[1-3]. 따라서 서비스 로봇에게 영상처리는 중요한 요소 중 하나라고 할 수

있다. Browne et al.은 autonomous sewer inspection 로봇의 vision 센서를 이용해 crack을 검출하고 분류했다[4]. Cao et al.은 YOLO 알고리즘을 이용하여 배드민턴 로봇의 셔틀콕을 감지했다[5]. Bong et al.은 가정에 적용할 서비스 모바일 로봇을 위한 카메라가

\*\* 레드원테크놀로지 (ksm@urc.kr)  
\* 교신저자 : 조선대학교 전자공학부  
· 접수일 : 2022. 06. 30  
· 수정완료일 : 2022. 07. 21  
· 게재확정일 : 2022. 08. 17

· Received : Jun. 30, 2022, Revised : Jul. 21, 2022, Accepted : Aug. 17, 2022  
· Corresponding Author : Ji-Youn Moon  
Dept. of Electronics Engineering, Chosun University  
Email : jymoon@chosun.ac.kr

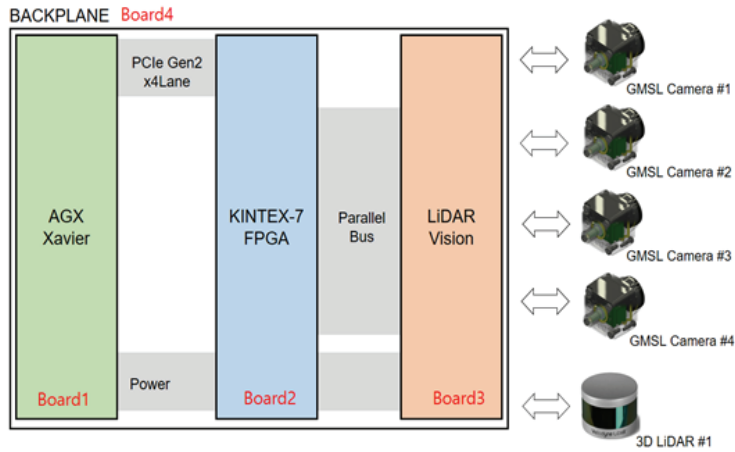


그림 1. rDNA 장치 하드웨어 구성도  
 Fig. 1 rDNA hardware architecture

장착된 소형 매니플레이터를 소개하였다[6]. 이와 같이 vision 센서를 이용한 다양한 연구가 이루어지고 있다[7-9]. 이러한 알고리즘들을 로봇의 임무 수행에 활용하기 위해서는 적합한 영상처리 프로세서 개발이 중요하다. 로봇을 구동하고 vision 알고리즘을 효율적으로 처리하기 위해서는 서로 다른 특징을 갖는 CPU, GPU, FPGA[10]를 모두 활용할 수 있어야 한다. 하지만 많은 연구자들은 각 프로세서를 개별적으로 활용하여 영상처리 알고리즘을 연구하고 있다 [11]. 이에 본 논문에서는 이를 모두 적용한 프로세서 설계법에 대해 제안한다.

## II. rDNA 모듈 개발

rDNA 모듈의 기본적인 구성은 CPU, GPU, FPGA가 융합된 통합 프로세서 형태의 구조를 가지도록 설계한다. 하드웨어 구성도는 그림 1과 같다. 본 모듈은 AI 기반 로봇 자율주행 어플리케이션을 위한 LiDAR-Vision 인터페이스를 지원하도록 설계한다. 최대 3~4채널의 GMSL(: Gigabit Multimedia Serial Link) 카메라 인터페이스 및 처리 알고리즘 및 1채널 3D LiDAR 인터페이스 및 처리 알고리즘 지원하도록 설계한다. 기능 확장 및 유지 보수의 간소화를 위하여

모듈화된 구조와 백플레인 기반의 신호 인터페이스 구조를 가지며 이기종 프로세서 간의 고속 데이터 교환을 위한 PCIe(PCI Express) 기반 인터커넥트 기술을 적용한다. rDNA 모듈은 인공지능 기반 서비스 로봇의 메인 프로세서 및 영상처리 프로세서, LiDAR-Vision 기반의 로봇 자율 항법 프로세서 및 무인 경비로봇의 다채널 카메라 처리용 프로세서 등에 적용 가능하다.

### 2.1 LiDAR-Vision 보드 설계

최대 3~4채널의 GMSL 카메라 및 3D LiDAR 카메라 신호 인터페이스를 제공하며, 해당 신호를 FPGA 보드 상에서 처리 가능한 신호로 변환하는 보드이다. GMSL 카메라의 통신을 위한 하드웨어의 기본적인 구조는 다음과 같다. GMSL 통신을 위해서는 GMSL Serializer 와 GMSL Deserializer 가 송/수신 단에 부착되어 있어야 한다. 카메라 모듈에는 GMSL Serializer가 부착되며, 수신 단에는 GMSL Deserializer 가 부착되는 구조이다. GMSL 카메라에 사용되는 커넥터는 크게 FAKRA RF 커넥터와 HSD 커넥터로 구분된다. 본 개발에서는 FAKRA RF 커넥터는 사용하는 카메라를 선정하였으므로 개발 시 FAKRA RF 커넥터를 적용하였다. GMSL 카메라 신호를 프로세서에서 사용하기 위해서는 신호 변환 과

정이 필요하다. 실제 본 개발에서는 FPGA 보드가 GMSL 카메라 신호를 전처리하므로 FPGA에서 처리 가능한 신호 형태로 변환되어야 한다.

### 2.2 FPGA 보드 설계

FPGA 보드는 LiDAR-Vision 부에서 측정되는 거리 및 영상 정보를 디코딩/전처리하고 각 센서의 동작 및 설정을 제어하는 기능을 수행하는 보드로써 FPGA 보드와 센서 간의 기능 연계는 LiDAR-Vision 보드를 통하여 중계된다. AXI DMA(: Direct Memory Access) 기반의 PCI Express 통신 구조에서 Write DMA(: WDMA) 부분에 전처리 된 LiDAR 및 Vision 데이터를 쓰기(Writing) 하여 PCI Express 통신을 상위 Host 로 전송하는 구조가 적용된다. AXI Lite 기반의 레지스터 인터페이스를 이용하여 LiDAR 및 카메라의 동작 및 설정 등을 수행한다. FPGA 보드의 메인 프로세서는 XILINX 사의 고급형 FPGA인 KINTEX-7 시리즈 칩 중 하나인 XC70K160T 프로세서를 적용하여, 고사양의 어플리케이션 처리가 가능하도록 한다.

### 2.3 AGX 보드 설계

GMSL 카메라의 메인 연산 및 응용 어플리케이션 처리가 수행되는 보드로서 NVIDIA 사의 JETSON AGX Xavier 프로세서가 적용된다. 개발 보드에 적용되는 Jetson AGX Xavie ( 32Gb 모델 ) SOM 모듈은 NVIDIA 사의 임베디드 AI 컴퓨팅 플랫폼 디바이스 중 높은 성능을 보유하고 있다. Jetson AGX Xavier SOM 모듈의 경우 임베디드 시스템을 위한 기본적인 구성장치인 프로세서, 메모리, 플래쉬, 주변장치가 이미 내장되어 있으므로, 이와 관련된 인터페이스 라인 및 커넥터들만 추가 설계한다. FPGA 보드와의 통신을 위한 백플레인 인터커넥트 라인으로 PCI Express Gen2, x4Lane 규격을 적용한다. 해당 인터 커넥트를 통해 FPGA 보드 상에서 처리된 영상 정보를 고속으로 취득할 수 있다.

### 2.4 Backplane 보드 설계

AGX Xavier 보드, FPGA 보드, LiDAR-Vision 보드 등에서 제공하는 연동 신호들이 상호 연결될 수 있도록 중계하는 보드이다. 백플레인 모듈을 통하여 중계되는 신호는 전원 신호 24V, IO 기준 전압인 3.3V, 고속의 PCI Express 신호, Parallel IO 신호 등으로 구성된다.

## III. 실험 및 검토

본 논문에서는 개발된 rDNA 보드는 그림 2에서와 같다. 보드의 성능을 테스트하기 위하여 PCIe 성능 테스트 시뮬레이션을 진행하였다. 실험을 위한 세부적인 내용은 다음과 같다.

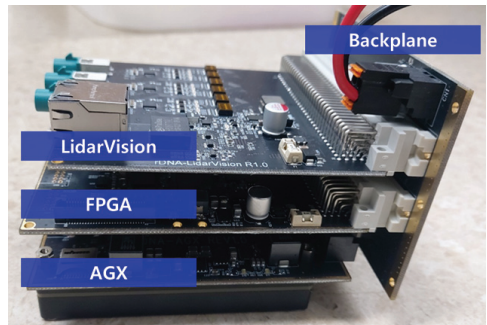


그림 2. rDNA 모듈  
Fig. 2 rDNA module

### 3.1 실험 정보 및 내용

FPGA는 크게 PCIe subsystem과 Algorithm Top 블록으로 구성되어 있다. PCIe subsystem은 PCIe 장치 구현을 위하여 PCIe endpoint를 포함하고 있으며 내부적으로 Memory Mapped 방식의 AXI(: ARM eXtensible Interface) Master와 Slave로 구성되어 있다. BAR #0 영역은 그림 3에서 회색으로 표시된 M\_AXI(AXI master) 인터페이스로 변환되어서 FPGA 내부 로직과 인터페이스 되고, DMA Coherent 영역은 그림 3에서의 파란색으로 표시된 S\_AXI (AXI slave) 인터페이스로 변환되어서 FPGA 내부 로직과 인터페이스 된다. 따라서 데이터의 전송 주체가 Host 시스템

인 BAR #0 영역은 PCIe endpoint가 그 역할을 맡아서 FPGA 내부 로직의 데이터 전송의 주체가 된다. 반대로 DAM Coherent 영역은 FPGA가 데이터 전송의 주체가 되어서 slave인 PCI endpoint를 통해서 Host 시스템으로 데이터를 전송하거나 가져온다. BAR #0 영역은 Host 시스템이 PCIe 장치를 제어 하기 위한 영역이다. 이 영역을 통해서 FPGA 내부의 제어 레지스터에 접근할 수 있으며, 저속의 단일 Read/Write operation으로 FPGA와 데이터를 교환한다.

본 논문에서는 BAR #0 영역에 32KB를 할당하였으며, FPGA 내부에서는 다시 하위 16KB를 4KB 영역 4개로 나눠서 사용하고 있고 (0x8000\_0000 ~ 0x8000\_3FFF), 나머지 상위 16KB는 향후 용도를 위해서 예약된 공간이다. 즉, Host 시스템에서 (BAR #0의 주소 + 0x0000 ~ 0x0FFF)에 데이터를 쓰거나 읽으면, FPGA의 (0x8000\_0000 ~ 0x8000\_0FFF) 영역에 존재하는 Test Memory에 데이터를 쓰거나 읽을 수 있다. 이 영역은 Random 데이터를 쓰고 읽어서 검증함으로써 PCIe 장치와의 통신에 문제가 없는지 검증하는데 활용하기 위해서 구현된 메모리 영역이다.

Host 시스템에서 (BAR #0의 주소 + 0x1000 ~ 0x1FFF)에 데이터를 쓰거나 읽으면, FPGA의 (0x8000\_1000 ~ 0x8000\_1FFF) 영역에 접근할 수 있

으며 이 영역은 FPGA의 PCIe Controller를 제어하는 Register가 놓여 있다.

Host 시스템에서 (BAR #0의 주소 + 0x2000 ~ 0x3FFF)에 데이터를 쓰거나 읽으면, FPGA의 (0x2000\_3000 ~ 0x8000\_3FFF) 영역에 존재하는 APB Bridge에 접근되며 이를 통해서 내부 System Configuration Register와 FPGA 내부의 Algorithm을 제어하는 register에 접근할 수 있다.

DMA Coherent 영역은 PCIe 장치와 대량의 데이터를 고속으로 교환하기 위한 영역이다. PCIe 장치가 DMA(: Direct Memroy Access) 방식으로, 즉 CPU의 간섭 없이 데이터를 가져가거나 가져다 준다. 일반적으로 CPU는, 시스템 메모리 접근으로 인한 성능 저하를 줄이기 위해서, Cache 컨트롤러를 내장하고 있으며 이 컨트롤러는 Cache 메모리라고 하는 고속의 로컬 메모리 영역에 CPU 연산에 필요한 데이터의 일부분을 Prefetch 해오거나, Cache 메모리에 있는 데이터를 시스템 메모리로 Write back하는 역할을 한다. 하지만, PCIe 장치와 CPU의 Cache 컨트롤러는 각자 단독으로 동작하기 때문에 동일한 영역의 시스템 메모리를 개별적으로 Update하였을 경우 상대방은 이를 알아챌 수가 없기 때문에 데이터의 불일치 문제가 발생할 수 있다.

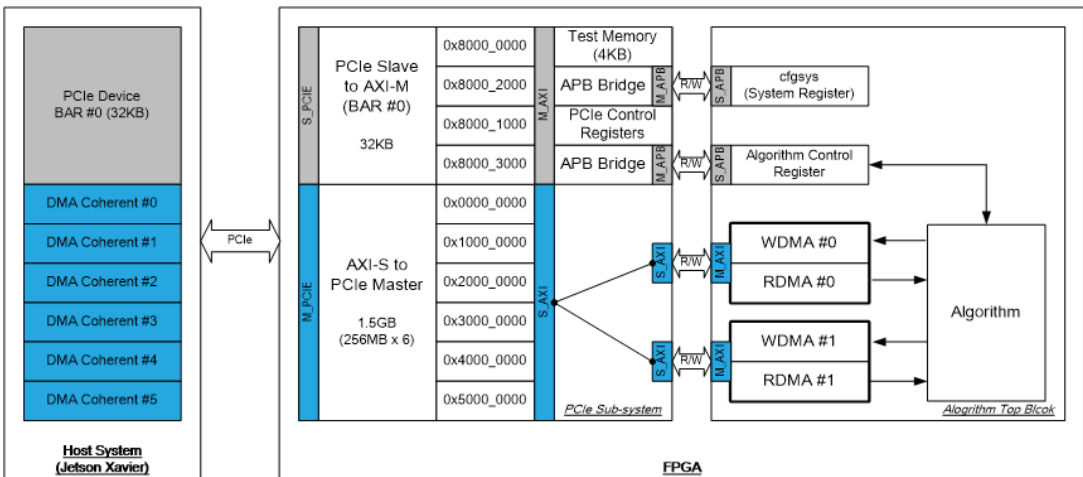


그림 3. Host 시스템 및 FPGA 구성도  
Fig. 3 Architecture of host system and FPGA

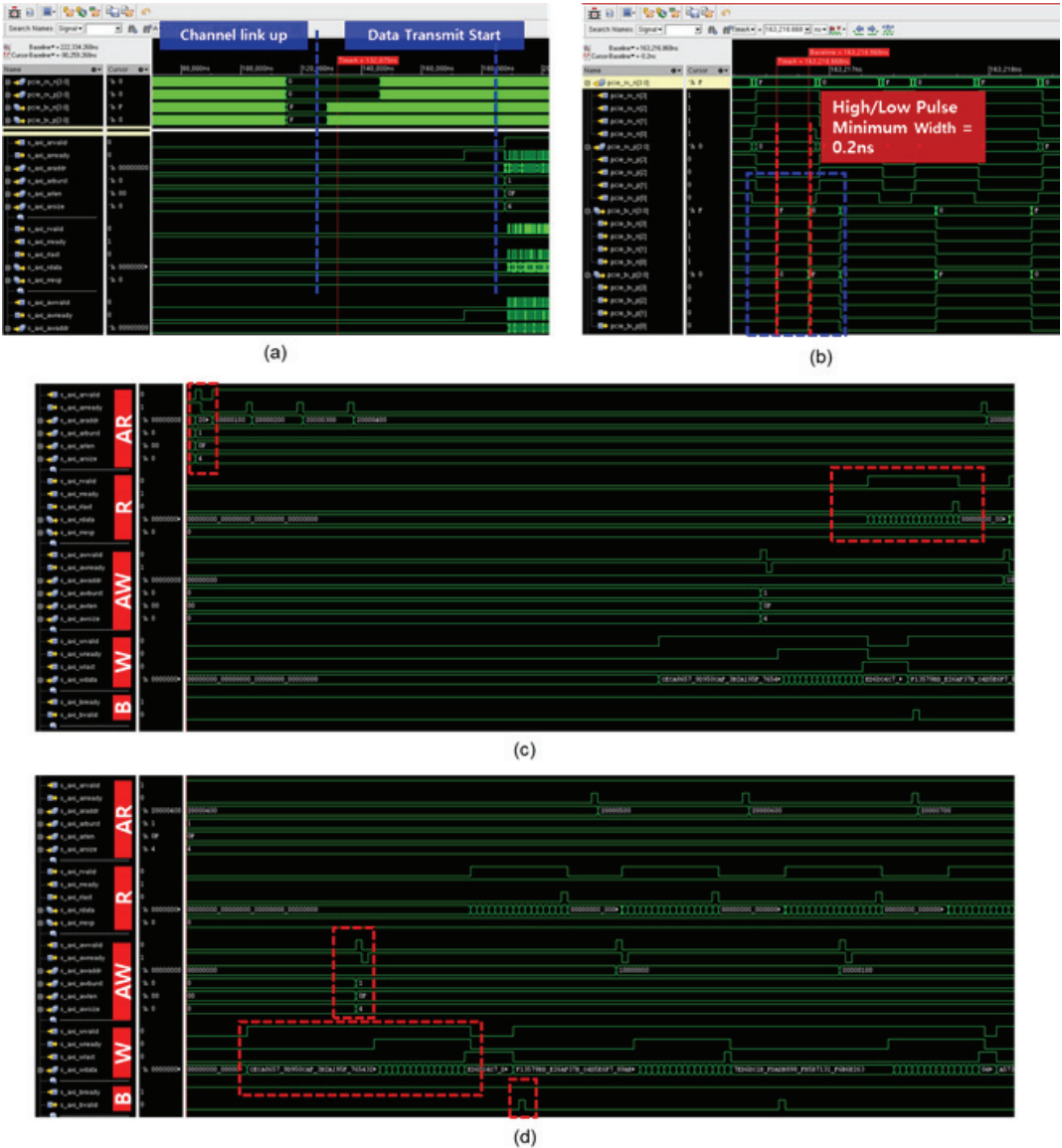


그림 4. 시뮬레이션 결과 파형  
Fig. 4 Pulse result of simulation

이러한 문제를 방지하기 위해서 Jetson 내부에는 PCIe가 접근할 수 있는 영역의 일부를 Cache Coherent 영역에 할당할 수 있다. Cache Coherent 영역은 PCIe 장치가 데이터를 Write할 경우 이 정보가 Cache controller에 전달되어 CPU가 이 영역에 접근하고 자 할 경우 데이터를 새롭게 Prefetch 한다.

본 논문에서는 FPGA가 Host 시스템 메모리의 6개 영역에 접근할 수 있도록 구현되었으며, 그림 3에서 FPGA에 표기되어있는 주소 영역에 데이터를 read/write하면 Host 시스템의 Cache coherent 영역으로 데이터가 read/write 된다.

## 4.2. 실험 결과

그림 4를 통해 데이터 전송을 통한 전체 시뮬레이션 파형 결과를 확인할 수 있다. 그림 4의 (a)에서와 같이 FPGA의 PCIe Tx Data (+/-), x4 레인의 파형을 보이고 있으며 Rx Data (+/-), x4 레인의 파형을 보인다. FPGA에 내장된 PCIe subsystem 내의 M\_AXI (AXI master) 인터페이스를 통해서 데이터가 전송된다. 그림 4의 (a)에서와 같이 호스트와 FPGA가 최초로 연결되고 서로 간의 채널의 link가 이뤄지기 전까지는 데이터 전송이 불가능하지만 채널 link 이후에 FPGA에 제어 명령을 통해서 데이터 전송을 활성화하고 난 이후에 데이터 전송이 수행됨을 확인할 수 있다. 데이터 레인의 High / Low 펄스의 최소 폭은 PCIe 속도에 대응된다. 그림 4의 (b)에서와 같이 측정된 High / Low 펄스의 최소 폭은 0.2ns이며 주파수로 환산하면 5GHz이다. 따라서 PCIe Gen2의 규격인 Lane 당 전송 속도 5Gbps와 정확히 일치함을 확인할 수 있다.

그림 4의 (c)에서와 같이 Read 동작은 AR, R 채널로 구성되어 있으며, AR 채널은 전송을 위한 address와 전송 크기 및 전송 길이를 요청하는 데 사용된다. 실질적인 데이터는 전송은 R 채널을 통해서 이뤄지며, read 요청에 문제가 없었는지를 응답하는 response도 R채널을 통해서 이뤄진다. Write 동작은 AW, W, B 채널로 구성되어 있으며, AW 채널은 전송을 위한 address와 전송 크기 및 전송 길이를 요청하는 데 사용된다. 실질적인 데이터는 전송은 W 채널을 통해서 이뤄지며, write 요청에 문제가 없었는지를 응답하는 response는 B 채널을 통해서 이뤄진다.

Read 채널에서 ARVALID와 ARREADY가 모두 'H' 이면 AR 채널의 전송이 승인된 것이고, RVALID와 RREADY가 모두 'H' 이면 R 채널의 전송이 승인된 것이다. 데이터 전송의 마지막에서는 RLAST가 'H'이다. 본 논문에서는 그림 4의 (c)에서와 같이 transfer size는 128비트, transfer length는 16 버스트를 적용하여 데이터를 전송한다. 즉, 한 번에 Host 시스템으로부터 256 바이트를 읽어오고 있다.

Write 채널에서 AWVALID와 AWREADY가 모

두 'H' 이면 AW 채널의 전송이 승인된 것이고, WVALID와 WREADY가 모두 'H' 이면 W 채널의 전송이 승인된 것이다. 데이터 전송의 마지막에서는 WLAST가 'H'이다. B 채널은 BVALID와 BREADY가 모두 'H' 이면 전송이 승인된다. 본 논문에서는 transfer size를 128비트로 transfer length는 16 버스트를 적용하여 데이터를 전송한다. 그 결과 그림 4의 (d)에서와 같이 한 번에 256 byte를 호스트 시스템 메모리에 쓰고 있음을 확인할 수 있다.

최종적으로, 본 논문에서는 한 Read/Write 1 count 당 데이터는 16바이트씩 전송되며, 측정 구간 1 count는 8ns에 해당한다. 이를 환산하면 Read throughput은 746MB/s, Write throughput은 758MB/s로써, 합산하면 총 1,504MB/s의 전송 대역을 사용하고 있다. 그 결과, 전송 효율은 75.21%에 해당하며, 버스 대역을 최대한 활용하고 있다고 볼 수 있다.

## IV. 결론

영상처리 프로세서는 로봇이 다양한 서비스를 제공하기 위해 중요한 부분이다. 이에, 본 논문에서는 AGX, FPGA, LiDAR-Vision, Backplane 보드로 구성된 영상처리 프로세서 설계방법을 제안한다. 직접 설계하여 구성한 프로세서는 시뮬레이션을 통해 출력된 결과로 검증을 진행하였으며 75.21%의 데이터 전송 효율을 보임을 확인할 수 있었다.

### 감사의 글

이 논문은 2021학년도 조선대학교 학술연구비의 지원을 받아 연구되었음.

## References

- [1] J. Fan, Z. Pai, and L. Shufei, "Vision-based holistic scene understanding towards proactive human - robot collaboration," *Robotics and Computer-Integrated Manufacturing*, vol. 75, 2022, pp. 102304.



- [2] M. Bujanca, P. Gafton, S. Saeedi, A. Nisbet, B. Bodin, M. OBoyle, and S. Furber, "SLAMBench 3.0: Systematic automated reproducible evaluation of SLAM systems for robot vision challenges and scene understanding," In *2019 International Conference on Robotics and Automation*, Montreal, Canada, 2019, pp. 6351-6358.
- [3] M. Naseer, S. Khan, and F. Porikli, "Indoor scene understanding in 2.5/3d for autonomous agents: A survey," *IEEE access*, vol. 7, 2018, pp. 1859-1887.
- [4] M. Browne and S. Ghidary, "Convolutional neural networks for image processing: an application in robot vision," In *Australasian Joint Conference on Artificial Intelligence*, Sydney, Australia, 2003, pp. 641-652.
- [5] Z. Cao, T. Liao, W. Song, Z. Chen, and C. Li, "Detecting the shuttlecock for a badminton robot: A YOLO based approach," *Expert Systems with Applications*, vol. 164, 2021, pp. 113833.
- [6] J. Bong, J. Lee, L. Jo, H. Jo, and G. Jeong, "Mobile Manipulator, Vision-based Manipulation, Rotational Inertia, Image Processing, Pick and Place," *J. of the Korea Institute of Electronics Communications Sciences*, vol. 17, no. 3, 2022, pp. 507-514.
- [7] J. Moon, J. Moon, and S. Bea, "Control for Manipulator of an Underwater Robot Using Meta Reinforcement Learning," *J. of the Korea Institute of Electronics Communications Sciences*, vol. 16, no. 1, 2021, pp. 95-100.
- [8] D. Madusanka, R. Gopura, Y. Amarasinghe, and G. Mann, "Hybrid vision based reach-to-grasp task planning method for trans-humeral prostheses," *IEEE Access*, vol. 5, 2017, pp. 16149-16161.
- [9] D. Ebert and D. Henrich, "Safe human-robot-cooperation: Image-based collision detection for industrial robots," In *IEEE/RSJ international conference on intelligent robots and systems*, Lausanne, Switzerland, 2002, pp. 1826-1831.
- [10] M. Vestias and H. Neto, "Trends of CPU, GPU and FPGA for high-performance computing," In *24th International Conference on Field Programmable Logic and Applications*, Belfast, United Kingdom, 2014, pp. 1-6.
- [11] S. Asano, T. Maruyama, and Y. Yamaguchi, "Performance comparison of FPGA, GPU and CPU in image processing," In *international conference on field programmable logic and applications*, Czech Republic, 2009, pp. 126-13.

저자 소개

### 문지윤 (Ji-Youn Moon)



2014년 광운대학교 로봇학부 졸업(공학사)  
2020년 서울대학교 대학원 전기정보공학부 졸업(공학박사)

2020년~현재 조선대학교 전자공학부 조교수

※ 관심분야 : 일반인공지능, 인지 로봇틱스, 뉴로-심볼릭, 강화 학습, 임베디드 AI 컴퓨팅

### 김수민(Soo-Min Kim)



2021년 조선대학교 전자공학과 졸업(공학사)

2021년~현재 레드윈테크놀로지 연구원

※ 관심분야 : 로봇 제어, 모터 제어, 임베디드 AI 컴퓨팅

