https://doi.org/10.7840/kics.2022.47.11.2007

토마토 수확 로봇을 위한 3D 위치 검출 임베디드 시스템 구현

이 기 범◆, 김 용 현*, 조 병 효*, 김 원 경*, 김 만 중*, 홍 영 기*, 김 경 철°

Implementation of 3D Location Detection Embedded System for Tomato Harvesting Robots

Ki-Beom Lee◆, Yong-Hyun Kim*, Byeong-Hyo Cho*, Won-Kyung Kim*,
Man-Jung Kim*, Youngki Hong*, Kyoung-Chul Kim°

요 약

최근 지속적인 농업 인구의 감소와 고령화에 따른 노동력 부족 문제를 해소하기 위하여 인공지능 기반의 농업 자동화 기술에 관한 연구가 많은 관심을 받고 있다. 본 논문에서는 대표적인 시설작물인 토마토의 자동화 수확 로봇 개발을 위하여 딥러닝 기반의 3차원 위치 검출 임베디드 시스템을 구현하고, 실제 환경에서 시스템의 적용 가능성 확인을 목적으로 한다. 토마토의 3차원 위치 검출 시스템은 저렴한 비용으로 구축이 가능한 저전력, 소형 임베디드 장비인 NVIDIA Jetson Xavier NX와 이미지의 3차원 정보를 얻기 위한 스테레오 타입의 ZED2 카메라로 구성된다. 제안한 시스템은 최신 YOLOv5 객체 검출 모델을 이용하여 토마토 이미지를 학습하고, 학습된 모델을 통해 검출된 토마토의 2차원 좌표를 3차원 좌표로 변환하여 토마토의 3차원 위치를 검출한다. 또한, 학습된 모델의 성능향상을 위하여 추론 속도를 수배에서 수십 배까지 향상시킬 수 있는 모델 최적화 엔진인 TensorRT를 적용하여 모델의 최적화를 수행한다. 구현한 시스템의 성능은 TensorRT를 적용한 최적화 모델의 평균 정밀도와 이미지 추론 시간에 대하여 비교하였으며, 성능개선을 확인하였다.

키워드 : 위치 검출, 객체 검출, 스테레오 비전, 수확 로봇, 임베디드 시스템
Key Words : Position Detection, Object Detection, Stereo Vision, Harvesting Robot, Embedded System

ABSTRACT

In recent years, the research on artificial intelligence-based agricultural automation technology has received a lot of attention to solve the problem of labor shortage due to the continuous decline of the agricultural population and aging. In this paper, we implement a deep learning-based three-dimensional location detection embedded system for the development of automated harvesting robots for tomatoes and aim to verify the applicability of the system in a real environment. The three-dimensional positioning system consists of NVIDIA Jetson Xavier NX, a low-power, small and low cost, and a stereo-type ZED2 camera to obtain three-dimensional information on images. The proposed system learns tomato images using the latest YOLOv5

※ 본 연구는 농림축산식품부 및 과학기술정보통신부, 농촌진흥청의 재원으로 농림식품기술기획평가원과 재단법인 스마트팜연구개발사업단의 스마트팜다부처패키지혁신기술개발사업의 지원을 받아 수행되었습니다(421031-04).
◆ First Author : National Institute of Agricultural Sciences, Jeonju, Korea, keywii@korea.kr, 정회원
° Corresponding Author : National Institute of Agricultural Sciences, Jeonju, Korea, kkcmole@korea.kr, 정회원
* National Institute of Agricultural Sciences, Jeonju, Korea
논문번호 : 202207-141-0-SE, Received June 7, 2022; Revised September 6, 2022; Accepted September 21, 2022

object detection model and converts the two-dimensional coordinates of the detected tomatoes into three-dimensional coordinates through the learned model to detect the three-dimensional position of the tomatoes. To improve the performance of the trained model, we also apply TensorRT, a model optimization engine that can improve inference speed from several to dozens of times. The performance of the implemented system was compared with the average precision and image inference time of the optimization model applied with TensorRT, and the performance improvement was confirmed.

Ⅰ. 서 론

지속적인 농업 인구의 감소와 고령화에 따른 노동력 및 숙련된 노동자 부족 문제를 해소하기 위하여 농업의 자동화 기술에 관한 연구의 필요성이 대두되고 있다. 특히, 인공지능, IoT (Internet of Things), 빅데이터 및 로봇 기술 등을 농업 분야에 적용한 지능화된 무인 농업 로봇에 대한 연구가 활발하게 진행되고 있다[1,2]. 사람을 대신하여 작물의 상태를 효율적으로 파악하고 정확한 의사결정이 필요한 농업 자동화와 농업 로봇 연구 분야에서 시각 지능 기술은 중요한 요소가 되고 있다. 작물 상태 모니터링 및 수확 등의 작업을 수행하는 농업 로봇은 영상 및 이미지로부터 작물을 인식하고 해당 작물에 접근하기 위하여 3차원 위치정보를 필요로 하기 때문에 작물 인식 및 3차원 위치 검출을 위한 정밀하고 신속한 시각 지능 기술에 대한 연구가 필수적이다.

이미지로부터 작물을 분류하고 위치정보를 얻기 위한 방법으로 컴퓨터 비전 기술을 이용한 분류 알고리즘과 방법론이 연구되어왔다[3,4]. 최근에는 ICT (Information and Communications Technology) 기술 발전과 인공지능 기술의 비약적인 발전으로 높은 정밀도와 빠른 객체 검출 능력을 갖춘 딥러닝 기반의 객체 검출 알고리즘이 전 산업 분야에서 활용되고 있으며, 농업 분야에서도 많은 관심을 받고 있다. 대표적인 객체 검출 모델로 SSD (Single Shot Multi-box Detector)[5] 및 Faster R-CNN (Faster Region-based Convolutional Neural Network)[6]과 같은 이단계 방식 모델과 단일 단계 방식의 YOLO (You Only Look Once) 모델을 기반으로 하는 연구가 진행되고 있다[7]. 하지만 이러한 딥러닝을 기반으로 하는 시스템은 고가의 GPU (Graphics Processing Unit) 로 구축된 시스템이 요구되기 때문에 실제 농가에 배치하는 것은 어려운 일이다[8,9]. 특히, 온실과 같이 작물이 밀집되어 있는 장소에서 운용하기 위해서는 가격뿐만 아니라 시스템의 소형화와 저전력도 중요한 요소가 된다.

최근 Nvidia의 소형 임베디드 시스템인 Jetson Xavier NX (Nvidia Inc., Jetson Xavier NX, Santa Clara, CA, USA) 모듈의 경우 상대적으로 높은 연산 속도와 처리능력을 보이기 때문에 교통, 의료, 농업 등 다양한 분야에서 임베디드 엣지 컴퓨팅 디바이스로 활용되고 있다[10-12]. 특히, NX 모듈은 범용 GPU와 비교하여 연산 능력은 낮지만, 딥러닝에 최적화된 설계로 수십만원대에 저렴한 비용으로 시스템을 구축할 수 있으며, 작은 크기와 함께 낮은 전력 소모로 인하여 배터리로 장시간 운용이 가능하다. 따라서 시스템의 소형화와 저전력이 중요한 고려사항이 되는 온실과 같은 장소에서 운용하기에 적합하다.

본 논문에서는 대표적 시설작물인 토마토를 대상으로 하는 자동화 수확 로봇을 위한 토마토 3차원 위치 검출 임베디드 시스템을 구현한다. 토마토 수확 로봇 임베디드 시스템을 구축은 소형 임베디드 장비와 스테레오 카메라를 결합한 형태로 임베디드 시스템을 구현하고 이를 통해서 실제 환경에서의 적용 가능성 확인을 목적으로 한다. 제안한 시스템에서는 토마토 검출을 위하여 최신 YOLOv5 모델을 이용한다. YOLOv5 모델은 이전의 다크넷 기반의 YOLOv4 모델과 비교하여 작은 용량에 빠른 추론 능력을 보인다. 또한, 크기에 따라 총 5가지 버전의 모델을 제공하여 정확도 또는 실시간 처리 등의 우선순위에 따라 모델 선택이 가능하다. 따라서 시스템이 구축된 상황에서 다양한 크기의 모델을 유연성 있게 적용이 가능한 장점이 있다. 토마토 검출 모델과 시스템의 성능은 평균 정밀도 및 이미지 처리 관점에서 평가된다. 또한, 학습된 모델의 최적화를 위하여, TensorRT를 적용하고 이에 따른 성능을 평가한다.

본 논문의 구성은 다음과 같다. 2장에서는 3차원 위치 정보를 계산하기 위한 스테레오 기하학과 딥러닝 기반 객체 검출 YOLO 모델에 대하여 기술하고, 3장에서는 제안한 임베디드 시스템의 구성과 각 요소들의 기술사양에 대하여 기술한다. 또한, 토마토의 3차원 위치 검출을 위한 과정에 대하여 설명하고, 모델 최적화 TensorRT에 대하여 언급한다. 4장에서는 딥러닝 학습 환경과 구현한 시스템의 성능을 평가한다.

마지막으로 5장에서 결론을 내린다.

Ⅱ. 관련 연구

2.1 스테레오 기하학을 이용한 3차원 정보 획득

영상 또는 이미지에서 객체의 3차원 정보는 스테레오 비전 기술을 이용하여 획득할 수 있다. 스테레오 비전은 두 개의 렌즈를 통해서 보는 한 점에 대하여 두 개의 이미지에서 나타나는 위치 차이인 시차(Disparity)를 이용하여 사물과의 거리를 추정할 수 있다. 두 개의 이미지를 이용한 스테레오 비전의 카메라 기하학을 이용하여 한 점에 대한 3차원 공간 좌표값 계산의 예는 그림 1[13]과 같다.

그림 1에서 한 점 p의 위치가 세계 좌표계로 표현될 때, $p=(x,y,z)$로 표현할 수 있다. 그림에서 p^l과 p^r은 점 p가 왼쪽과 오른쪽 영상 평면에 투영된 점이다. 이때 x좌표는 왼쪽 또는 오른쪽 영상 중 하나를 이용하여 구할 수 있다. 이 예에서는 세계 좌표계의 원점이 좌측 카메라 렌즈의 중심에 있다고 가정한다. 왼쪽 영상을 이용한다면 닮은꼴 삼각형의 pLo^l과 $p^lL^oo^l$를 이용하여 점 p의 x좌표는 수식 (1)로 구할 수 있다.

$$x=\frac{z \bullet x^l}{F} \tag{1}$$

여기서 F는 카메라 렌즈의 초점거리 (Focal Length)이며, x^l은 왼쪽 영상에 투영된 점 p^l의 x좌표값을 나타내고, z는 카메라의 렌즈에서 점 p의 거리이다.

점 p의 y 좌표는 다음 수식 (2)를 이용하여 구한다.

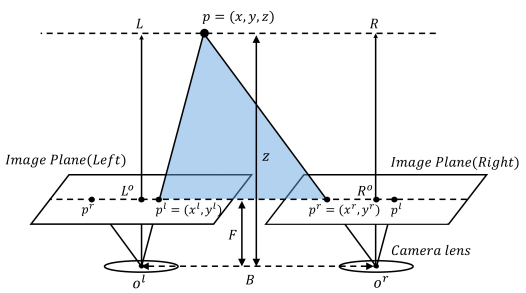


그림 1. 스테레오 시각의 카메라 기하학
Fig. 1. Camera geometry of stereo vision

$$y=\frac{z \bullet y^l}{F} \tag{2}$$

여기서 y^l은 왼쪽 영상에 투영된 점 p^l의 수직 좌표값을 나타낸다.

마지막으로 카메라의 렌즈에서 점 p의 거리인 z 좌표는 두 닮은꼴 삼각형 po^lo^r과 pp^lp^r의 닮음비를 기반으로 수식 (3)과 (4)를 이용하여 구할 수 있다.

$$d=x^l-x^r \tag{3}$$

$$z=\frac{F \bullet B}{d} \tag{4}$$

여기서 각 렌즈에 투영된 점 간의 거리 차이는 d (disparity)로 정의되며, 삼각형 pp^lp^r의 아래 변의 길이를 계산할 수 있다. 따라서 수식 (3)을 이용하여 d 값을 얻고, 렌즈의 초점거리 F와 기준선 (Baseline)의 길이인 B 값을 안다면, 한 점 p의 3차원 위치 좌표값(x,y,z)을 계산할 수 있다.

본 논문에서는 Stereolabs 사의 ZED2 (Stereolabs Inc., ZED2, San Francisco, CA, USA) 카메라를 이용하여 토마토의 영상과 3차원 정보를 수집한다. ZED2 카메라는 비전을 위한 이미지 센서로 듀얼 렌즈가 장착된 카메라로 양쪽 이미지를 이용에서 스테레오 비전 기술을 사용하여 고화질 영상과 깊이 맵(Depth map)을 얻을 수 있다[14]. 또한, ZED 카메라는 깊이 정보를 표현하는 방법으로 3차원 포인트 클라우드를 사용한다. 이 포인트 클라우드 방식은 3차원의

표 1. ZED2 카메라 기술 사양
Table 1. Technical Specifications of ZED 2 Camera

Output Resolution		4416×1242 @15fps 3840×1080 @30fps 2560×720 @60fps 1344×376 @100fps
Lenses	**Field of View**	Max. 110°(H) × 70°(V) × 120°(D)
	Baseline	120 mm
	Focal Length	2.12mm
	Aperture	f/1.8
Interface		USB 3.0/2.0
Depth Range		0.2 m to 20 m
Depth Accuracy		< 1% up to 3m < 5% up to 15m

깊이 맵으로 볼 수 있으며, 깊이 맵에는 각 픽셀의 거리 또는 Z 정보만 포함되어 있지만, 포인트 클라우드 방식은 색상 정보를 포함할 수 있는 3차원 포인트 (X,Y,Z)의 집합이다. ZED2 카메라의 기술 사양은 표 1과 같다. ZED2 카메라는 양쪽 렌즈에서 최대 2208×1242의 해상도로 초당 15 프레임의 영상을 처리할 수 있으며, 0.2 m에서 20 m까지 깊이 측정이 가능하다. 특히, 3 m 이내의 거리에 있는 물체에 대하여 1% 이하의 낮은 오차율을 보이며 15 m까지는 5% 이하의 오차율을 갖는다.

2.2 YOLO 기반 객체 검출

객체의 3차원 위치정보를 검출하기 위해서는 2차원 이미지에서의 위치와 분류가 선행되어야 한다. 이를 위하여 본 논문에서는 객체 검출 모델로 YOLO 모델을 사용한다. YOLO는 객체 탐지를 위하여 RPN (Region Proposal Network)과 분류 (Classifcation) 단계를 하나의 단일 네트워크로 결합한 모델로 이단 검출 모델에서 검출을 수행하는 이전의 RPN 기반 검출기와 비교하여 단일 피드 포워드 네트워크 (Feed Forward Network)를 통해 바운딩박스 (Bounding Box)와 해당 클래스를 예측한다. 이로 인하여 간결한 구조와 빠른 연산속도로 빠른 객체 탐지와 우수한 검출 능력을 보인 때문에 실시간 객체 탐지가 요구되는 시스템에 적용이 가능하다.

초기 YOLOv1은 Fast R-CNN과 비교하여 바운딩박스 위치를 제대로 예측하지 못하는 지역화 오류 (Localization Error)를 보였으며, YOLOv2 모델에서는 학습 이미지의 해상도를 높이고, 바운딩박스를 예측을 위하여 앵커박스 (Anchor box)를 활용한다. 또한, 완전 연결 계층 (fully connected layer) 대신 합성곱층 (Convolutional layer)을 이용한다[15]. YOLOv3에서는 Feature Pyramid Network와 유사한 3개의 스케일로 객체를 예측하고, 로지스틱 회귀를 이용해 각 바운딩박스의 점수를 예측하여 작은 물체 감지 기능이 대폭 개선되었다[16]. YOLOv4는 CSPDarknet53이라는 새로운 백본을 사용하고 SPP (Spatial Pyramid Pooling) 추가 블록, PANet (Path-Aggregation Network) 및 YOLOv3 헤드로 구성된다[17]. YOLOv4는 YOLOv3과 비교하여 평균 정밀도와 FPS가 각각 10%, 12% 개선된 결과를 보였다.

YOLOv5는 다크넷이 아닌 Pytorch 프레임워크를 기반으로 구현되었다. 또한, 백본의 크기에 따라서 제일 작고 가벼운 n (nano) 부터 s (small), m (medium), l (large), x (extra)까지 포함해서 총 5가지 버전이 있으며, 각 모델의 용량과 성능은 그림 2와 같다[18]. YOLOv5와 v4의 FPS 성능을 비교한 결과에서 YOLOv4는 약 50 FPS, YOLOv5는 140 FPS를 기록하였으며, 모델 용량은 다크넷 기반의 YOLOv4와 비교하여 1/10의 용량으로 비슷한 정밀도 보였다[19]. 따라서 YOLOv5 모델은 간결한 구조와 빠른 연산속도로 빠른 객체 탐지와 우수한 검출 능력으로 실시간 객체 검출 시스템에 적용하기에 적합하다.

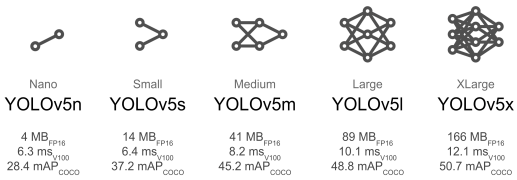


그림 2. YOLOv5 모델 비교
Fig. 2. Comparison of all YOLOv5 models

본 연구에서는 간결한 구조와 빠른 연산속도로 빠른 객체 탐지와 우수한 검출 능력을 보이는 YOLOv5 모델을 이용하여 토마토 3차원 위치 좌표 추정을 위한 객체 인식 시스템 구현하였다.

Ⅲ. 3차원 위치 인식 임베디드 시스템

본 장에서는 제안한 3차원 위치 인식 임베디드 시스템의 구성 및 기술 사양과 토마토 3차원 위치 좌표 추정을 위한 YOLOv5 모델 기반 객체 인식 시스템 구현에 대하여 설명한다.

3.1 시스템 구성

제안한 토마토 수확 로봇을 위한 임베디드 시스템의 주요 하드웨어 구성은 그림 3과 같다. 제안한 시스템은 영상 및 3차원 데이터 수집을 위하여 ZED2 카메라와 Nvidia Jetson Xavier NX 모듈 및

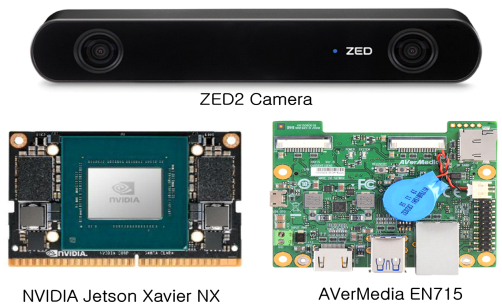


그림 3. 주요 하드웨어 구성
Fig. 3. Main hardware components

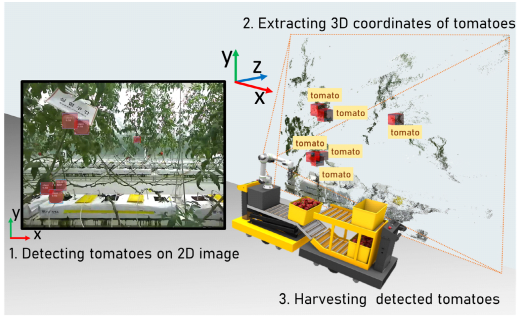


그림 4. 시스템 개요
Fig. 4. System overview

AVerMedia EN715 (AVerMedia Technologies, Inc., EN715, New Taipei City, Taiwan) 캐리어 보드로 구성된다. 여기서 ZED2는 토마토 영상 촬영과 3차원 깊이 정보 및 포인트 클라우드 정보를 NX 모듈에 제공한다. 그림 4와 같이 NX 모듈은 ZED2 카메라의 영상과 학습된 객체 탐지 모델을 이용하여 토마토 객체를 탐지하고 각 3차원 좌푯값을 계산하여 수확 로봇에 전달한다.

본 연구에서 사용된 NX 모듈은 Jetson Nano (Nvidia Inc., Jetson Xavier Nano, Santa Clara, CA, USA) 수준의 크기에 저전력 소형 폼 팩터 (Form factor)로 이전 TX2 모델보다 10배 이상의 성능을 제공한다. 표 2는 대표되는 Jetson 임베디드 모듈의 성능을 비교하여 나타낸 표이다. NX의 AI 퍼포먼스는 21 TOPS (Tera Operations Per Second)를 보이며, Nano는 0.472 TFLOPS (Tera Floating Point Operations Per Second), TX2는 1.33 TFLOPS의 성능을 보인다. 여기서 TOPS는 초당 정수 연산수를 의미하고, TFLOPS는 부동소수점 연산을 포함하는 연산수를 의미한다.

표 2. Jetson 임베디드 모듈 성능 비교
Table 2. Comparison of performance of Jetson embedded modules

Model	CPU	GPU	RAM	AI Performance
NX	6-core Carmel 6MB L2 + 4MB L3	384 CUDA Cores 48 Tensor Cores Nvidia Volta	8GB	21 TOPS
Nano	Quad-core A57	128 CUDA Cores Nvidia Maxwell	4GB	0.472 TFLOPS
TX2	Dual-core Denver / Quad-core A57	256 CUDA Cores Nvidia Pascal	8GB	1.33 TFLOPS

표 3. AVerMedia EN715 보드 기술 사양
Table 3. Specifications of AVerMedia EN715 board

Networking	1×GbE RJ-45
Display Output	3840×2160 at 60Hz
Temperature	Operating temperature 0°C ~ 70°C Storage temperature -40°C ~ 85°C Relative humidity 40°C @ 95%
MIPI Camera Inputs	2×2 lane MIPI CSI-2 1×4 Lane MIPI CSI-2
USB	1×USB 2.0 Micro-B for recovery 2×USB 3.0 Type-A
Storage	1×micro-SD card slot
GPIO Expansion	20 pins: 2×I2C, 1×UART, 9×GPIOs
Input Power	9V ~ 19V
Dimension/ Weight	W: 87 mm × L: 70.6 mm × H: 27.3 mm Weight: 70 g

NX 모듈과 주변 장치들과의 인터페이스를 제공하기 위하여 본 연구에서는 AVerMedia 사의 EN715 캐리어보드를 사용한다. EN715 캐리어보드는 나노 및 NX 모듈을 위한 표준 보드로 풍부한 입출력 인터페이스를 제공한다. 표 3은 EN715 보드의 기술 사양을 정리한 표이다. EN715 보드는 3개의 MIPI 카메라 입력 인터페이스와 USB 인터페이스를 제공하여 다수의 카메라로 시스템을 구성할 수 있으며, 16GB의 내부 저장 용량을 가진 NX 모듈의 부족한 저장 용량을 보완하기 위하여 micro-SD 카드 인터페이스도 제공한다. 본 연구에서는 부족한 내부 저장 용량으로 인하여 256GB micro-SD에 시스템을 구축하였다. 표 4는 제안한 시스템의 세부 환경 구축 정보를 나타낸 표이다. Nvidia Jetpack은 AI 및 컴퓨터 비전 애플리케이션을 개발하고 배포하기 위한 SDK로 Cuda, cuDNN, TensorRT 등을 패키지와 프레임워크를 함께 제공하여 상호 의존성 문제를 해결하여 연구 개발의 편의성을 제공한다. 본 논문에서는 Jetpack 버전 4.6을 사용하며 Pytorch 버전의 YOLOv5 모델을 사용하기 위하

표 4. 임베디드 시스템 환경 정보
Table 4. Embedded system environment

Jetpack	v4.6	**Cuda**	v10.2
OS	L4T_32.6.1	**Python**	v3.6
Pytorch	v1.7	**cuDNN**	v8.2.1
ZED SDK	v3.7.2	**Onnx**	v1.11
TensorRT	v8.0.1	**Torchvision**	v0.8.1

여 Pytorch 1.7 버전을 사용한다. 또한, NX 모듈에서 ZED2 카메라를 사용하기 위한 Jetson 보드 전용 SDK를 이용하여 ZED2에서 제공하는 API 및 기능을 사용한다.

3.2 3차원 토마토 위치 검출

토마토 수확 로봇이 작업을 수행하기 위해서는 토마토 객체를 검출하고 3차원 공간상의 위치정보를 얻어야 한다. 이를 위해서는 딥러닝 모델의 학습이 선행되어야 한다. 그림 5는 제안한 시스템의 전체 흐름도를 보인다. 먼저 토마토의 객체 검출을 위한 YOLO 모델이 학습되는 과정으로 학습 데이터셋을 이용하여 지도학습을 수행한다. 학습 과정은 학습에 사용되는 이미지 및 어노테이션 (Annotation) 데이터셋을 불러오고, 앵커박스와 파라미터를 설정한다. 앵커박스는 k-means 알고리즘을 이용하여 최적의 앵커박스를 조정하고, 전이 학습 (Transfer learning)을 수행한다. 전이 학습에 사용되는 모델은 COCO 데이터셋으로 사전에 학습된 모델로 해당 네트워크 가중치로 학습하고자 하는 모델을 초기화하여 학습을 진행한다. 이를 통해서 더 빠른 학습 속도와 성능을 얻을 수 있다[20]. 이후 설정된 Epoch만큼 학습이 수행되며 이후 학습된 모델이 토마토 객체 탐지에 사용된다. 이미지에서 토마토의 3차원 좌표값을 추출하기 위해서는 토마토 객체 탐지를 통해서 토마토가 있는 2차원 좌표를 찾아야 한다. 학습된 모델을 이용하여 입력된 이미지에서 토마토 객체를 탐지하고 3차원 공간 좌표값을 추정하는 과정은 우선 객체 탐지를 통해 토마토가 존재하는

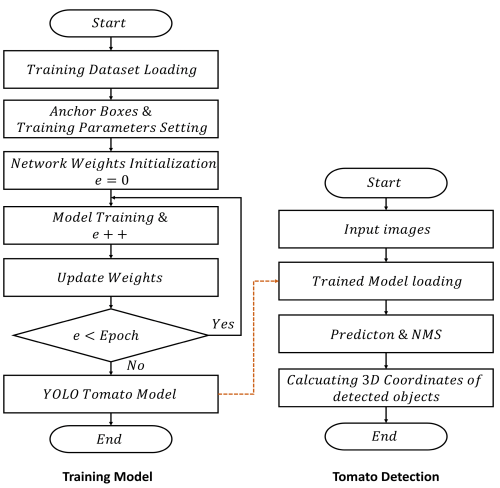


그림 5. 제안 시스템의 흐름도
Fig. 5. Flowchart of proposed system

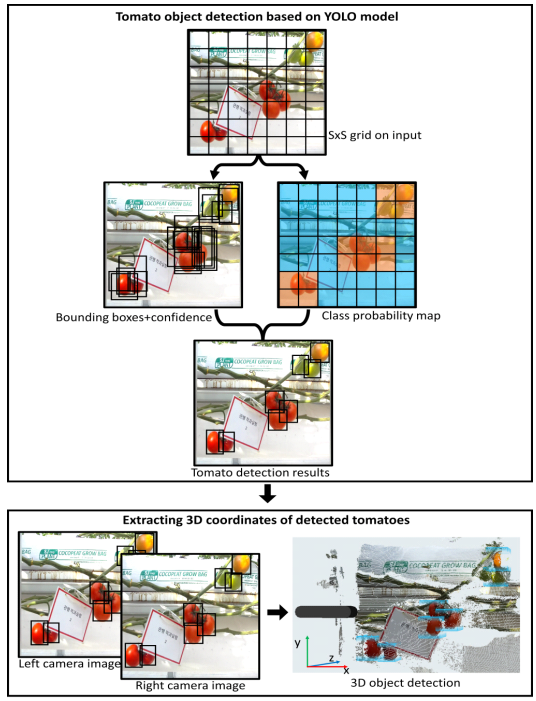


그림 6. 토마토 3차원 위치 검출 과정
Fig. 6. Tomato 3d location detection process

정확한 2차원 좌표를 찾는 과정이 필요하다.

먼저 학습된 YOLO 모델을 토마토 탐지 프로그램에 올리고 이미지 내의 객체를 탐지하여 객체 주위에 표시한 직사각형인 바운딩박스를 찾는다. 이때 다수의 중복되는 바운딩박스가 존재할 수 있는데 NMS (Non-Maximum Suppression)를 통해서 최적의 바운딩박스를 찾는 과정을 거친다. 토마토의 3차원 위치 검출 과정은 그림 6에 보인다. 다음으로 2차원 바운딩박스의 2차원 좌표값을 이용하여 3차원 공간의 세계 좌표로 변환하는 방법을 사용한다. 이를 위해서 ZED SDK에서 제공하는 ingest_custom_box_objects 함수를 이용하여 검출된 토마토의 3차원 공간 정보를 얻을 수 있다. 생성된 3차원 공간에 대한 정보는 3차원 바운딩박스인 정육면체 8개의 각 꼭짓점의 좌표값과 바운딩박스의 중심 위치 (position), 속도 (velocity), 크기 (dimensions) 값을 얻을 수 있다.

3.3 TensorRT를 이용한 최적화

TensorRT는 CUDA 및 엔비디아의 병렬 프로그래밍 모델을 기반으로 그림 7과 같이 학습된 딥러닝 모델을 최적화하고 실행을 지원하며 Nvidia GPU 상에서의 추론 속도를 수배에서 수십배까지 향상시킬 수 있는 모델 최적화 엔진이다[21]. TensorRT는 사용하면

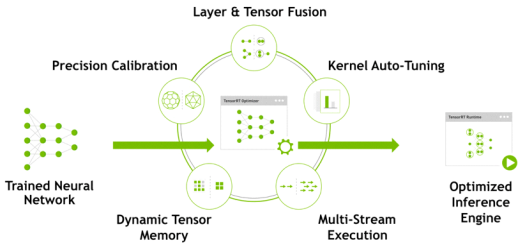


그림 7. TensorRT 기반 모델 최적화
Fig. 7. TensorRT based model optimization

Caffe, Pytorch, TensorFlow와 같은 모든 주요 프레임워크에서 학습된 신경망 모델을 최적화하는 것이 가능하다. 본 논문에서는 학습된 모델의 이미지 처리 속도를 높이기 위하여 TensorRT를 이용하여 각 모델을 FP16의 정밀도로 각 모델을 최적화한다. Pytorch로 학습된 모델의 최적화를 수행하기 위해서 우선 학습 모델을 Onnx[22] (Open Neural Network Exchange) 모델로 변환하는 과정이 필요하다. 따라서, YOLOv5의 각 모델을 Onnx 모델로 변환한 뒤에 각 Onnx 모델을 이용하여 TensorRT의 Engine 모델로 변환하여 기존 Pytorch 모델 대신 최적화된 Engine 모델을 사용한다. 기존 Pytorch 모델과 Engine 모델의 성능은 4장에서 평균 추론 시간의 관점에서 비교된다.

Ⅳ. 실험 및 성능평가

본 논문에서는 데스크톱 환경에서 Pytorch 기반의 YOLOv5 모델 학습을 수행하고 학습된 모델을 이용하여 임베디드 시스템의 성능을 평가한다. 4장에서는 제안한 시스템의 실험환경 구성 및 성능평가 결과를 보인다.

4.1 토마토 객체 탐지 모델 학습

객체 탐지 모델의 학습은 Ubuntu 18.04 운영체제에서 Intel(R) Core(TM) i9-7920X CPU와 32GB RAM, 두 대의 Nvidia GeForce RTX 2080 Ti GPU가 설치된 데스크톱에서 수행되었으며, 자세한 모델의 학습 환경은 표 4와 같다.

토마토 객체 탐지 모델을 학습시키기 위하여 사용된 데이터셋은 케글 (Kaggle)에서 제공하는 오픈 토마토 데이터셋을 이용하여 단일 토마토 클래스로 학습을 진행하였다[23]. 학습에 사용된 케글 데이터셋은 온실 토마토 이미지 895장 중에서 805장은 학습에 사용되고 90장은 학습된 모델의 평가에 사용되었다. 사

표 5. 학습 시스템 환경 및 하이퍼 파라메터
Table 5. System environment and hyperparameters for training

OS	Ubuntu 18.04
CPU	24 core
GPU	Nvidia GeForce RTX 2080 Ti
RAM	32GB
Pytorch	v1.7
Cuda	v10.2
Image size	640
Epoch	300

용된 데이터셋에는 각 이미지의 실제 토마토 좌표값을 나타내는 Ground Truth 값이 어노테이션 파일에 포함되며, 총 4,930개의 토마토 좌표정보가 포함되어 있다. 그림 8은 토마토 학습 데이터 이미지 중 일부를 나타낸다.

학습된 모델의 성능은 평균 정밀도(Average Precision: AP), 정밀도(Precision), 재현율(Recall), 평균 FPS (Frame Per Second) 항목으로 하여 평가된다. 정밀도는 모델이 True라고 검출한 것 중에서 실제 True인 것의 비율이며, 재현율은 실제 True인 것 중에서 모델이 True라고 예측한 것의 비율을 의미한다. 수식 (5)는 정밀도를 나타내며 수식 (6)는 재현율, 수식 (7)은 평균 정밀도를 나타낸다. 수식에서 TP (True Positive)는 실제 True인데, 분류모델에서 예측이 True라고 판단된 경우이다. TN (True Negative)는 실제 False인데, 분류모델에서 예측이 False라고 판단된 경우이다. FP (False Positive)는 실제 False인데, 분류모델에서 예측이 True라고 판단된 경우이다. FN (False Negative)는 실제 True인데, 모델에서 예측이 False라고 판단된 경우이다.

그림 8. 토마토 학습 및 검증을 위한 샘플 이미지
Fig. 8. Tomato sample images for training and test

$$Precision = \frac{TP}{TP+FP} \quad (5)$$

$$Recall = \frac{TP}{TP+FN} \quad (6)$$

정밀도와 재현율만으로 객체 검출 모델의 성능을 평가하는 것은 적절하지 않다. 정밀도가 높으면 재현율이 낮은 경향이 있고, 반대로 정밀도가 낮으면 재현율이 높은 경향이 있기 때문이다. 따라서 어느 하나의 기준으로 모델의 성능을 평가하는 것보다는 두 값을 종합하여 성능을 평가해야 한다. 일반적으로 객체 탐지 모델의 성능을 평가하기 위하여 AP를 이용하여 각 토마토 인식 모델의 성능을 평가하고 비교한다. AP는 객체 인식 모델의 정확도를 나타내기 위한 평가 지표로 사용되며 Precision-Recall 곡선을 아래쪽 면적으로 계산된다. AP는 PR 곡선을 단조적으로 감소하는 그래프로 표현하여 계산하며 수식 (7)과 (8)과 같다[24].

$$AP = \sum (r_{n+1} - r_n) p_{interp}(r_{n+1}) \quad (7)$$

$$p_{interp}(r_n) = \max_{\tilde{r} \geq r_n} p(\tilde{r}) \quad (8)$$

여기서 r_n은 n번째 재현율 지점을 의미하고, $p(r)$은 재현율이 r인 지점에서의 정밀도이다. $p_{interp}(r_n)$은 r_n의 오른쪽 지점에서 가장 큰 정밀도 값을 의미한다. 그림 9와 10은 학습이 진행되면서 각 YOLO 모델의 IoU (Intersection over Union)가 50 이상의 임곗 값을 가질 때와 50 이상 95 이하인 경우

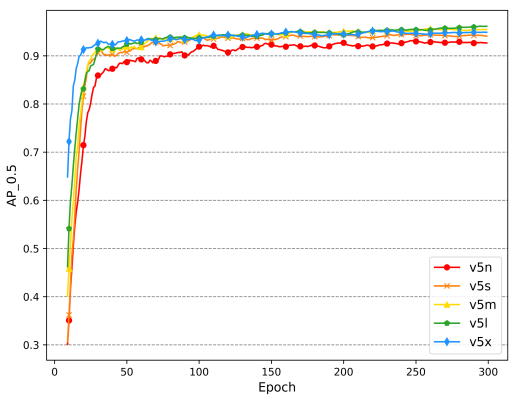


그림 9. 학습된 YOLO 모델에 따른 AP50 비교
Fig. 9. Comparison of AP50 of trained models

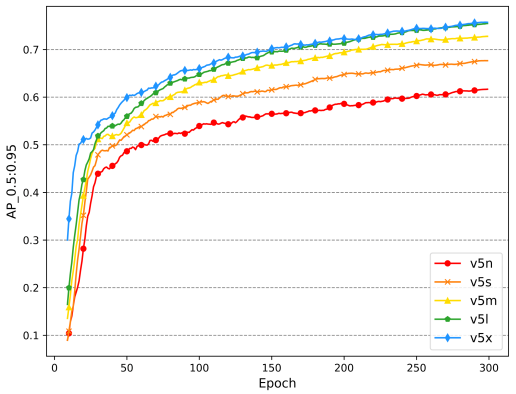


그림 10. 학습된 YOLO 모델에 따른 AP50:95 비교
Fig. 10. Comparison of AP50:95 of trained models

의 AP의 변화를 나타낸 그래프이다. 두 결과에서 모두 모델의 크기가 가장 작은 nano 모델이 가장 낮은 정밀도를 보임을 확인할 수 있다. AP50에서는 nano 모델을 제외하고 비슷한 정밀도를 보이만 large 모델이 가장 높은 약 96%의 정밀도를 보였다. 한편, AP50:95에서 large와 x large 모델에서 약 76%의 정밀도를 보여 단일 클래스 토마토 모델에서 large와 x large가 비슷한 성능을 보인다. 표 6은 각 YOLO 모델의 평균 정밀도와 재현율 등을 정리하여 나타낸 표이다.

표 6. YOLO 모델에 따른 학습결과 비교
Table 6. Comparison of training results of different YOLO models

Models	AP_{50}	$AP_{50:95}$	Precision	Recall
YOLOv5n	0.933	0.627	0.914	0.891
YOLOv5s	0.944	0.685	0.915	0.920
YOLOv5m	0.958	0.735	0.943	0.913
YOLOv5l	0.962	0.761	0.956	0.899
YOLOv5x	0.950	0.765	0.945	0.895

4.2 임베디드 시스템 성능평가

제안한 임베디드 시스템의 성능을 평가하고 실제 환경에서 적용 가능성을 확인하기 위하여 국립농업과학원 (전라북도 완주군)의 첨단온실에서 재배되고 있는 Dafnis 품종의 토마토를 촬영한 영상을 이용하였다. 검증을 위한 영상은 ZED2 카메라를 이용하여 3천개 이상의 프레임을 갖는 720p와 1080p의 해상도로 촬영된 영상을 사용하였다. 그림 11은 토마토의 3차원 위치 검출 결과이다. 이미지에서 토마토를 검출과 2차원 좌표를 얻고 2차원 좌표를 기반으로 3차원 위치 검출을 수행한 결과를 보인다. 검출된 토마토의 좌

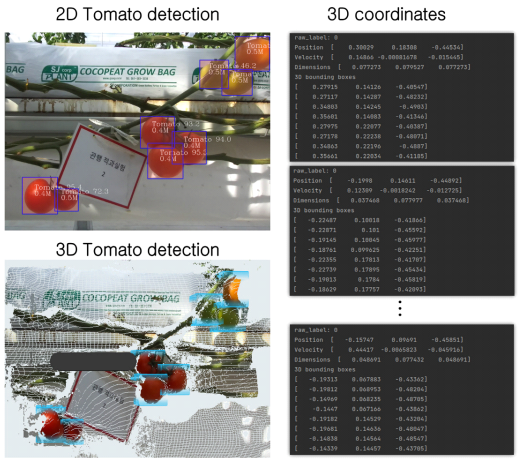


그림 11. 3차원 토마토 위치 검출 결과
Fig. 11. Results of 3D tomato location detection

표는 위치, 속도, 크기와 3차원 바운딩박스의 8개 꼭짓점의 좌표를 행렬 형태로 출력할 수 있다.

본 논문에서는 토마토 객체 검출 및 3차원 위치정보를 추출을 위해 ZED2 카메라로 촬영된 영상에서 이미지를 획득하고 최종적으로 토마토 객체를 탐지하여 3차원 위치를 추정하는 데까지 소요되는 처리시간을 추론 시간 (Inference time)으로 정의하고 추론 시간을 이용하여 FPS를 계산한다. 표 7은 딥러닝 모델 학습을 수행한 데스크톱과 제안한 임베디드 시스템의 추론 시간과 FPS를 정리한 표이며, 임베디드 시스템에 TensorRT 기반의 최적화 모델을 적용한 결과를 보인다. 그림 12는 평균 추론 시간을 나타낸 그래프로 각 모델과 적용 시스템에 따른 결과를 보여준다. 추론 시간을 비교한 결과 상대적으로 가장 컴퓨팅 성능이 좋은 데스크톱 환경에서 한 프레임의 이미지를 처리하는데 각 모델이 약 47에서 63 ms로 빠른 추론 시간을 보이며 각 모델 간의 차이가 크지 않은 것을 확인할 수 있었다. 제안한 임베디드 시스템은 평균 추론 시간이 약 147에서 280 ms로 데스크톱과 비교하여 각 모델 간의 차이가 존재하며, x large 모델의 경우 평균 3.6 프레임의 속도를 보여 빠른 추론 시간을 요구하는 응용에는 적용이 어려울 수 있다. TensorRT가 적용된 임베디드 시스템에서는 최적화가 적용되지 않은 시스템과 비교하여 평균 추론 시간이 1.2에서 1.5배까지 개선되어 평균 1.8 프레임 상승효과를 확인하였다. 이는 TensorRT가 적용된 객체 검출 모델의 이미지 추론 시간이 단축되어 실시간 검출 능력을 개선되었음을 의미한다. 또한, 그림 13과 14에서는 제안한 임베디드 시스템과 TensorRT가 적용된 시스템의 프레임에 따른 토마토 검출 수를 나타낸 그래프로 토마토 검출 능력에 차이가 없이 추론 시간 개선이 가능함을 확인하였다.

그림 15와 16에서는 프레임에 따른 추론 시간을 나타낸 그래프이다. 결과는 파이프라인을 이동하며 토마토 3차원 위치 검출 작업을 수행할 때 다수의 토마토가 있는 프레임에서는 추론 시간이 증가하지만 모든

표 7. YOLO 모델에 따른 평균 FPS 및 추론 시간 비교
Table 7. Comparison of average FPS and inference time of different YOLO models

Model	Desktop	NX	
		Default	TensorRT
YOLOv5n	21.36 (46.81ms)	6.81 (146.83ms)	8.94 (111.88ms)
YOLOv5s	21.51 (46.50ms)	6.75 (148.09ms)	8.56 (116.81ms)
YOLOv5m	19.62 (50.97ms)	6.20 (161.17ms)	7.62 (131.28ms)
YOLOv5l	18.39 (54.38ms)	5.31 (188.33ms)	6.97 (143.53ms)
YOLOv5x	15.89 (62.92ms)	3.57 (280.09ms)	5.55 (180.21ms)

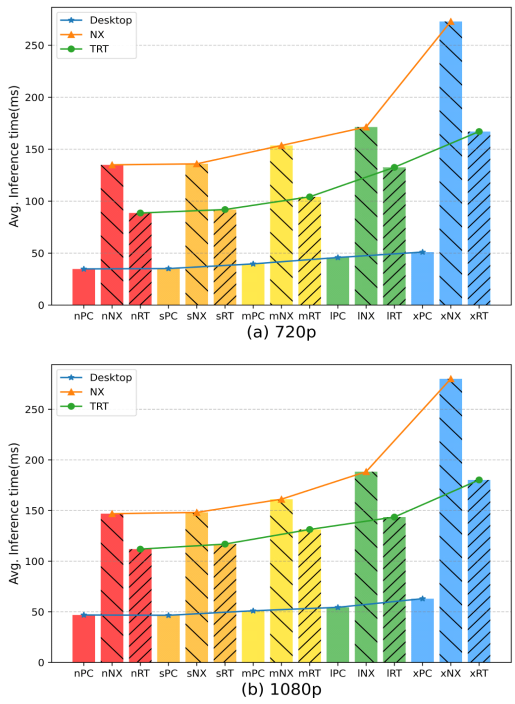


그림 12. 평균 추론 시간
Fig. 12. Average inference time

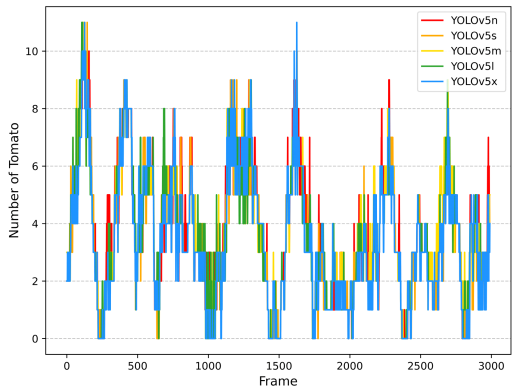


그림 13. 제안 시스템의 토마토 검출 수
Fig. 13. Number of tomato for proposed system

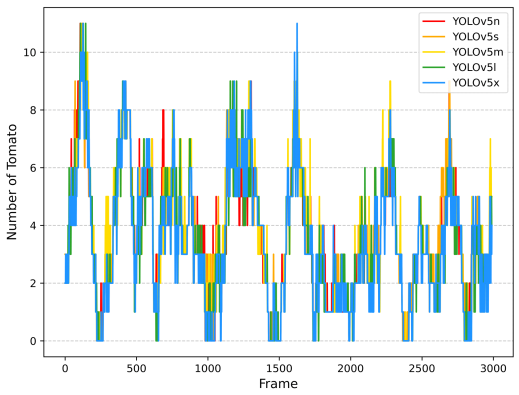


그림 14. TensorRT 적용 시스템의 토마토 검출 수
Fig. 14. Number of tomato for TensorRT applied system

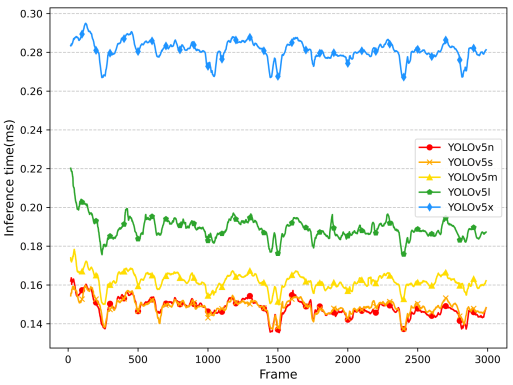


그림 15. 제안 시스템의 추론 시간
Fig. 15. Inference time of proposed system

모델에서 안정적인 토마토 3차원 위치 검출 작업이 가능함을 보였다.

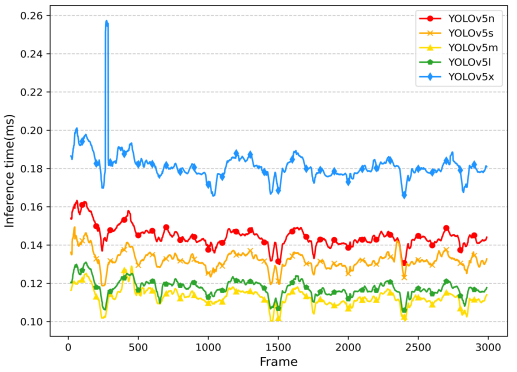


그림 16. TensorRT 적용 시스템의 추론 시간
Fig. 16. Inference time of TensorRT applied system

V. 결 론

본 논문에서는 토마토 수확 로봇 개발을 위한 3차원 위치 검출 임베디드 시스템을 구현하였다. 토마토 3차원 위치 검출 시스템을 구축하기 위하여 딥러닝 능력을 갖춘 산업용 컴퓨터와 비교하여 저렴한 비용으로 구축이 가능한 저전력, 소형 임베디드 보드인 Nvidia Jetson Xavier NX와 스테레오 타입의 ZED2 카메라로 시스템을 구현하였다. 제안한 시스템은 토마토 3차원 위치 검출을 위하여 최신 YOLOv5 객체 검출 모델을 이용하여 토마토 이미지를 학습시키고 모델을 통해 검출된 토마토의 2차원 좌표를 3차원 좌표로 변환하는 작업을 수행한다. 성능평가는 학습된 모델들의 평균 정밀도와 이미지 추론 시간 관점에서 수행되었다. IoU 50에서의 평균 정밀도는 모델 크기에 따라서 93%에서 96%의 높은 정밀도를 보이며, 이미지 추론 시간의 관점에서 제안한 시스템이 약 6.8에서 3.5의 FPS 값을 보였다. TensorRT를 이용하여 최적화가 적용된 모델에서는 평균 추론 시간이 1.2에서 1.5배까지 개선되어 평균 1.8 프레임 상승효과를 보여 모델 최적화를 통한 성능개선을 확인하였다. 다수의 토마토가 있는 일부 구간에서 추론 시간이 증가하지만, 안정적인 추론 능력을 보임을 확인하여 저비용, 저전력의 소형 임베디드 시스템을 이용하여 토마토 3차원 위치 검출 작업이 가능함을 확인하였다. 향후 연구로 검출된 3차원 위치정보를 이용하여 동일 개체 판정 및 추적, 모니터링 알고리즘에 관한 연구로 확장할 예정이다.

References

[1] G. J. Kim and J. D. Huh, "Trends and prospects of smart farm technology," *Electr. and Telecommun. Trends*, vol. 30, no. 5, pp. 1-10, Oct. 2015. (https://doi.org/10.22648/ETRI.2015.J.300501)

[2] U. Yeo, I. Lee, K. Kwon, T. Ha, S. Park, R. Kim, and S. Lee, "Analysis of research trend and core technologies based on ICT to materialize smart-farm," *J. Bio-Environ. Contr.*, vol. 25, no. 1, pp. 30-41, Mar. 2016. (https://doi.org/10.12791/KSBEC.2016.25.1.30)

[3] M. P. Arakeri, "Computer vision based fruit grading system for quality evaluation of tomato in agriculture industry," *Procedia Comput, Sci,*, vol. 79, pp. 426-433, 2016. (https://doi.org/10.1016/j.procs.2016.03.055)

[4] C. Costa, F. Antonucci, F. Pallottino, J. Aguzzi, D. W. Sun, and P. Menesatti, "Shape analysis of agricultural products: A review of recent research advances and potential application to computer vision," *Food and Bioprocess Technol.*, vol. 4, no. 5, pp. 673-692, Mar. 2011. (https://doi.org/10.1007/s11947-011-0556-0)

[5] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, and S. Reed, "SSD: Single shot multibox detector," *Eur. Conf. Comput. Vision*, pp. 21-37, 2016. (https://doi.org/10.1007/978-3-319-46448-0_2)

[6] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *Advances in NIPS*, pp. 91-99, 2015. (https://doi.org/10.48550/arXiv.1506.01497)

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. CVPR*, pp. 779-788, 2016. (https://doi.org/10.48550/arXiv.1506.02640)

[8] M. O. Lawal, "Tomato detection based on modified YOLOv3 framework," *Scientific Reports*, vol. 11, no. 1, pp. 1-11, 2021. (https://doi.org/10.1038/s41598-021-81216-5)

[9] K. Ko, H. J. Park, and I. H. Jang, "Real-time tomato instance tracking algorithm by using deep learning and probability model," *J. Korea Robotics Soc.*, vol. 16, no. 1, pp. 49-55, Feb. 2021. (https://doi.org/10.7746/jkros.2021.16.1.049)

[10] M. A. Rahman and M. S. Hossain, "An internet-of-medical-things-enabled edge computing framework for tackling COVID-19," *IEEE Internet of Things J.*, vol. 8, no. 21, pp. 15847-15854, 2021. (https://doi.org/10.1109/JIOT.2021.3051080)

[11] W. J. Chang, L. B. Chen, C. H. Hsu, C. P. Lin, and T. C. Yang, "A deep learning-based intelligent medicine recognition system for chronic patients," *IEEE Access*, vol. 7, pp. 44441-44458, 2019. (https://doi.org/10.1109/ACCESS.2019.2908843)

[12] P. Inthanon and S. Mungsing, "Detection of drowsiness from facial images in real-time video media using Nvidia Jetson Nano," *Int. Conf. ECTI-CON IEEE*, pp. 246-249, 2020. (https://doi.org/10.1109/ECTI-CON49241.2020.9158235)

[13] Y. C. Du, M. Muslikhin, T. H. Hsieh, and M. S. Wang, "Stereo vision-based object recognition and manipulation by regions with convolutional neural network," *Electronics*, vol. 9, no. 2, Jan. 2020. (https://doi.org/10.3390/electronics9020210)

[14] *ZED2*, Retrieved Feb. 22, 2022, from https://www.stereolabs.com/zed-2/

[15] J. Redmon and A. Farhadi, "YOLO9000: Better, faster, stronger," in *Proc. IEEE Conf. CVPR*, pp. 7263-7271, 2017. (https://doi.org/10.48550/arXiv.1612.08242)

[16] J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," arXiv preprint arXiv:1804.02767, 2018. (https://doi.org/10.48550/arXiv.1804.02767)

[17] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," arXiv preprint arXiv: 2004.10934, 2020. (https://doi.org/10.48550/arXiv.2004.10934)

[18] Ultralytics, yolov5, Retrieved Feb. 18, 2022, from https://github.com/ultralytics/yolov5.

[19] G. Yang, W. Feng, J. Jin, Q. Lei, X. Li, G. Gui, and W. Wang, "Face mask recognition system with YOLOV5 based on image recognition," in *ICCC*, pp. 1398-1404, 2020. (https://doi.org/10.1109/ICCC51575.2020.9345042)

[20] D. Lee, Y. G. Sun, S. H. Kim, I. Sim, K. S. Lee, M. N. Song, and J. Y. Kim, "Transfer learning-based object detection algorithm using YOLO network," *The J. Inst. Internet, Broadcasting and Commun.*, vol. 20, no. 1, pp. 219-223, 2020. (https://doi.org/10.7236/JIIBC.2020.20.1.219)

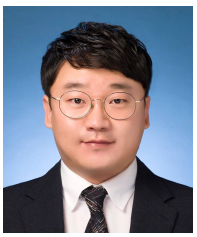
[21] *Nvidia tensorRT*, Retrieved Apr. 5, 2022, from https://developer.Nvidia.com/tensorrt.

[22] *Onnx*, Retrieved Apr. 5, 2022, from https://onnx.ai/

[23] Larxel, *Tomato Detection*, Kaggle, 2020, Available from: https://www.kaggle.com/andrewmvd/tomato-detection (accessed Apr. 2, 2022)

[24] G. Liu, J. C. Nouaze, P. L. T. Mbouembe, and J. H. Kim, "YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3," *Sensors*, vol. 20, no. 7, pp. 1-20, 2020. (https://doi.org/10.3390/s20072145)

이 기 범 (Ki-Beom Lee)



2014년 8월 : 전북대학교 컴퓨터공학과 석사
2021년 2월 : 전북대학교 컴퓨터공학과 박사
2022년 1월~현재 : 국립농업과학원 스마트팜개발과 박사후 연구원

<관심분야> 강화학습, 객체인식, 디지털트윈
[ORCID:0000-0002-7251-0953]

김 용 현 (Yong-Hyun Kim)

2021년 2월 : 성균관대학교 바이오메카트로닉스학과 석사
2021년 7월~현재 : 국립농업과학원 스마트팜개발과 전문연구원
2022년 3월~현재 : 충남대학교 바이오시스템기계공학과 박사과정

<관심분야> 인공지능, 컴퓨터비전, 스마트농업
[ORCID:0000-0003-4774-2354]

조 병 효 (Byeong-Hyo Cho)

2017년 8월 : 충북대학교 바이오시스템공학과 석사
2021년 3월 : 홋카이도대학 농학원 공생기반학전공 박사
2021년 7월~현재 : 국립농업과학원 스마트팜개발과 박사후 연구원

<관심분야> 영상처리, 농업 인공지능, 비파괴 분석
[ORCID:0000-0001-7276-5617]

김 원 경 (Won-Kyung Kim)

2014년 2월 : 서울대학교 바이오시스템·소재학부 석사
2022년 8월 : 부산대학교 바이오산업기계공학과 박사
2021년 7월~현재 : 국립농업과학원 스마트팜개발과 박사후 연구원

<관심분야> 스마트농업, 온실환경센서, IoT
[ORCID:0000-0003-4774-2354]

김 만 중 (Man-Jung Kim)

2019년 2월 : 전북대학교 기계시스템공학과 석사
2022년 2월 : 전북대학교 기계시스템공학과 박사
2022년 3월~현재 : 국립농업과학원 스마트팜개발과 박사후 연구원

<관심분야> 농업 로봇, 농업자동화, 자동화시스템
[ORCID:0000-0003-0600-489X]

김 경 철 (Kyoung-Chul Kim)

2015년 8월 : 전북대학교 정밀기계공학과 박사
2016년 7월~2019년 1월 : 농업기술실용화재단 연구원
2019년 2월~현재 : 국립농업과학원 스마트팜개발과 연구사

<관심분야> 농업로봇, 농업 인공지능
[ORCID:0000-0001-6699-881x]

홍 영 기 (Youngki Hong)

2004년 3월: 일본 동경농업대 생물환경조절학전공 박사
2007년 3월~2021년2월 : 국립농업과학원 농업연구사
2021년 3월~현재: 국립농업과학원 스마트팜개발과 연구관

<관심분야> 영상처리, 기계학습, 인공지능
[ORCID:0000-0002-9772-9820]