

딥러닝 객체 탐지 기술을 사용한 스마트 쇼핑카트의 구현

Implementation of Smart Shopping Cart using Object Detection Method based on Deep Learning

IPL .

Received: date / Accepted: date

Abstract 최근 다양한 쇼핑 환경에서 결제에 소요되는 시간을 줄이기 위한 많은 시도들이 이루어지고 있다. 또한 4차 산업혁명시대에 들어서면서 인공지능 기술이 고도화되고 있으며, IoT 장비들은 더욱 소형화되고 저렴해져서 이 두 가지 기술을 융합시킴으로써 사용자의 시간을 절약할 인간을 대신하는 무인 환경을 구축하는 것에 대한 접근이 용이해졌다. 본 논문에서는 저가 IoT 장비들과 딥러닝 객체 탐지 기술을 기반으로 스마트 쇼핑카트 시스템을 제안한다. 제안된 스마트 카트 시스템은 실시간 상품 인식을 위한 카메라와 라즈베리파이, 트리거 역할을 하는 초음파 센서, 상품이 쇼핑카트에 들어온 것인지 나간 것인지를 판단하기 위한 무게 센서, 가상의 장바구니에 대한 UI를 제공하는 스마트폰 어플리케이션, 학습된 데이터가 저장되는 딥러닝 서버로 구성된다. 각 모듈간의 통신은 TCP /IP 네트워크 및 HTTP 네트워크로 이루어지며, 서버의 상품 인식을 위해서는 객체탐지 기술이 구현된 YOLO darknet 라이브러리를 사용한다. 사용자는 스마트폰의 앱을 통하여 스마트 카트에 넣은 물건들의 목록을 점검하고 자동으로 결제할 수 있다. 본 논문에서 제안된 스마트 카트 시스템은 가성비가 높은 무인 상점을 구현하는데 응용될 수 있다. Recently, many attempts have been made to reduce the time required for payment in various shopping environments. In addition, as the 4th Industrial Revolution era, artificial Intelligence technology is advancing and IoT devices are becoming more compact and cheaper. So, by integrating these two technologies, access to building an unmanned environment on behalf of human beings to save users' time became easier. In this paper, we propose a smart shopping cart system based on low-cost IoT equipment and deep learning object detection technology. The proposed smart cart system consists of a camera for real-time product detection, an ultrasonic sensor that acts as a trigger, a weight sensor to determine whether a product enters into or out of shopping cart, and an application of smartphone that provides a UI for a vir-

IPL

M612 Univ.SoonCheonHyang 22, Soonchunhyang-ro, Sinchang-myeon, Asan-si, Chungcheongnam-do, 31538, Rep. of KOREA
E-mail: ohjinjin0408@naver.com

tual shopping cart, and a deep learning server where learned product data are stored. Communication between each module is made of TCP/IP and HTTP network, and YOLO darknet library, an object detection system is used by the server to recognize the product. The user can check the list of items put in the smart cart through the app of the smartphone and automatically pay. The smart cart system proposed in this paper can be applied to implement unmanned stores with high cost-performance ratio.

Keywords Deep Learning · Real-time Object Detection · YOLO · Internet of Things · Smart Shopping Cart ·

1 서론

2 Introduction

기존 쇼핑 환경에서 사용되는 결제 시스템은 많은 인력과 소요 시간을 필요로 한다. 최근에는 딥러닝 기술의 발전에 따라 새로운 패러다임의 쇼핑환경을 제공하는 무인 상점이 등장하였다. 대표적인 무인 상점이 “아마존 고 (Amazon Go)”이다. 아마존 고에서는 계산을 위하여 줄을 설 필요가 없다. 무인 상점은 계산에 소요되는 시간을 크게 단축하는 사용자 경험을 지원한다는 점에서 의미가 크다. 따라서 국내외적으로 이에 대한 관심이 증가하고 있다 [Wankhede et al(2018)Wankhede, Wukkadada, and Nadar].

Payment System is used for shopping environments in general requires a lot of manpower and time. Recently, With the development of deep learning technology, unmanned stores that provide a new paradigm shopping environment have appeared. The representative unmanned shop is “Amazon Go”. In Amazon Go, there is no need to line up for calculation. Unmanned stores are significant in that they support a user experience that significantly reduces the time it takes to calculate. Therefore, there is increasing interest in this at home and abroad [Wankhede et al(2018)Wankhede, Wukkadada, and Nadar].

무인 상점의 대표 모델로 제시되는 “아마존 고”의 경우, 수백 대의 카메라와 마이크, 압력 센서들이 고객을 실시간으로 추적하여 고객과 상품 간의 접촉을 판단한다. 매장에 들어오는 고객이 자신의 스마트폰을 입구에 설치된 기기에 스캔하면, 이 때부터 매장 안에서 고객은 시스템의 3D 목표물로 표시된다. 하지만 아마존 고에는 결정적인 약점이 있는데 인공지능 판독 문제로 매장 내 수용 인원을 100여 명으로 제한하고 있다. 또 아마존 고에서는 모든 이용객의 이동경로를 추적하고 분석하기 때문에, 데이터 수집량이 매우 커지게 되고, 따라서 대형 유통망에 확대 적용하기에는 무리가 있다.

In the case of “Amazon Go”, which is presented as a representative model of unmanned stores, hundreds of cameras, microphones, and pressure sensors track customers in real time to determine contact between customers and products. When a customer entering a store scans his smartphone on a device installed at the entrance, the customer is displayed as a 3D target of the system from this point. However, Amazon Go has a decisive weakness that is limiting the number of people in the store to more than 100 due to AI reading

problems. In addition, Amazon Go tracks and analyzes the movement paths of all users, so the amount of data collected becomes very large. Therefore, it is difficult to apply it to large distribution networks.

본 논문에서는 앞서 출시된 무인 상점의 단점을 최소화하기 위하여 딥러닝 객체 탐지와 라즈베리파이를 사용한 스마트 쇼핑카트를 제안하고자 한다. 제안된 시스템은 카메라와 초음파 센서, 무게 센서, TCP/IP 기반 네트워킹 기능, 딥러닝 서버, 안드로이드 기반의 사용자 스마트폰 앱으로 구성되어있다. 무인 상점에서는 이 시스템을 통해 실시간으로 쇼핑카트에 투입된 상품의 번호와 수량, 상품 투입 정보를 사용자의 기기로 전송할 수 있어 가상의 장바구니 목록에 상품을 추가하거나 삭제할 수 있다.

In this paper, we propose a smart shopping cart using deep learning object detection and Raspberry Pi to minimize the disadvantages of the unmanned store. The proposed system consists of a camera, ultrasonic sensor, weight sensor, TCP / IP-based networking function, deep learning server, and Android-based user smartphone app. In an unmanned store, there can be added to or deleted from the virtual shopping cart list because through this system, the number, quantity, and product input information of the product input into the shopping cart can be transmitted to the user's device in real time.

스마트 쇼핑 카트는 현재 몇 개의 방법들이 제안되어 있다. 대부분의 스마트 쇼핑 카트는 사용자 인터페이스상의 얼굴 인식 기능을 통해, 고객을 인식하고 RFID (Radio-Frequency Identification) 태그를 사용하여 카트에 추가되는 다양한 제품을 자동으로 탐지하고 관련 정보를 사용자 인터페이스에 표시한다. 하지만 RFID를 모든 제품에 부착하려면 비용과 많은 노력이 필요하다. 본 논문에서는 RFID 대신에 카메라를 사용하여 상품들을 인식하므로 모든 상품에 RFID 태그가 부착될 필요가 없다 [Chiang et al(2016)Chiang, You, Lin, Shih, Liao, Lee, and Chen, Karjol et al(2017)Karjol, Holla, and Abhilash].

Several methods of smart shopping carts are currently proposed. Most smart shopping carts recognize customers through face recognition on the user interface and use RFID (Radio-Frequency Identification) tags to automatically detect various products added to the cart and display relevant information on the user interface. However, attaching RFID to all products requires cost and effort. In this paper, it is not necessary to have RFID tags attached to all products because products are recognized using cameras instead of RFID [Chiang et al(2016)Chiang, You, Lin, Shih, Liao, Lee, and Chen, Karjol et al(2017)Karjol, Holla, and Abhilash].

본 논문의 구성은 다음과 같다. 2장에서는 딥러닝 객체탐지와 라즈베리파이를 사용한 스마트 쇼핑카트의 개요와 시스템 설계를 설명한다. 3장에서 각 모듈별 구현 세부사항과 실험을 통한 본 시스템의 성능에 대하여 기술한다. 끝으로 4장에서 결론 및 향후 연구 방향에 대하여 기술한다.

The composition of this paper is as follows. Chapter 2 explains an overview of smart shopping cart and system design using deep learning object detection and Raspberry Pi. Chapter 3 describes the implementation details of each module and the performance of this system through experiments. Finally, Chapter 4 describes the conclusions and future research directions.

3 스마트 쇼핑 카트 시스템의 설계

4 Design of smart shopping cart system

4.1 개요

4.2 Abstract

Fig. 1과 Fig. 2에서는 본 논문에서 제안한 스마트 쇼핑카트의 프로토타입을 보여준다. 스마트 쇼핑카트는 상품이 카트로 투입되는 것을 탐지하는 초음파 센서, 상품을 인식하는 파이카메라, 전체 시스템을 제어하고 네트워크 통신을 맡은 라즈베리파이, 쇼핑카트에 놓인 상품들의 무게를 측정하는 무게센서로 구성된다.

Fig. 1 and Fig. 2 shows the prototype of the smart shopping cart proposed in this paper. The smart shopping cart consists of an ultrasonic sensor that detects that the product is being put into the cart, a pi camera that recognizes the product, a raspberry pi that controls the entire system and performs network communication, and a weight sensor that measures the weight of the products placed in the shopping cart.



Fig.

Front view of Smart Shopping Cart



Fig.

Back view of Smart Shopping Cart

Fig. 3에서는 제안된 스마트 쇼핑 카트 시스템을 구성하는 구성요소들과 인터페이스를 보여준다.

Fig. 3 shows the components and interfaces that make up the proposed smart shopping cart system.

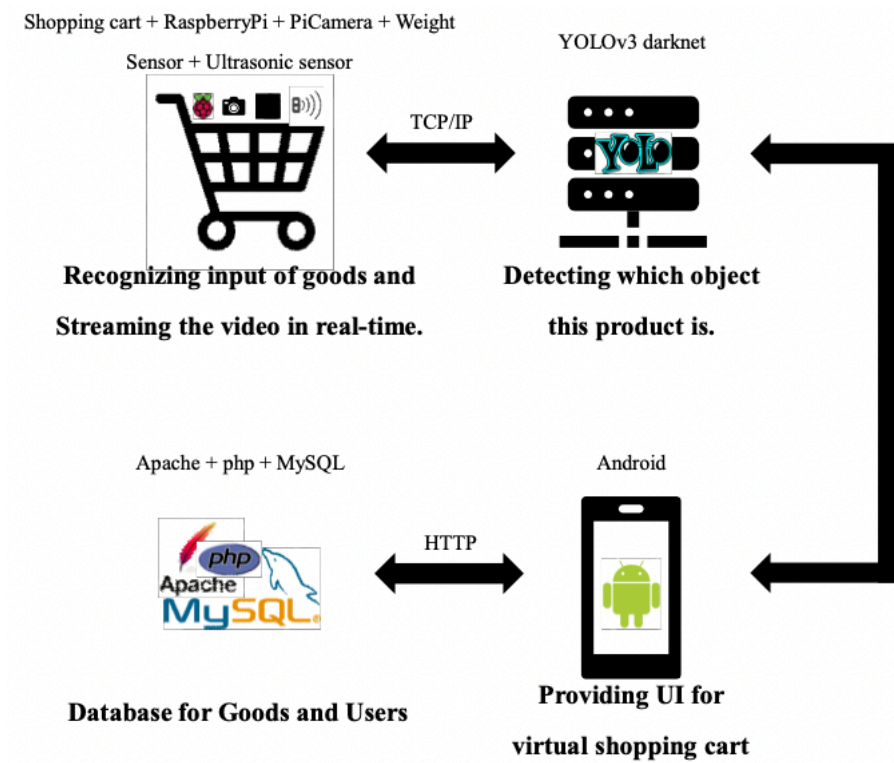


Fig. System schematic diagram of smart shopping cart 쇼핑카트와 딥러닝 서버는 TCP/IP 통신 기능을 이용하고 중앙 서버와 안드로이드 앱도 TCP/IP 통신을 사용한다. 데이터베이스 서버와 안드로이드 앱은 HTTP 프로토콜을 사용한다.

Shopping cart and deep learning server use TCP / IP communication function. And central server and Android app also use TCP / IP communication. The database server and Android app use the HTTP protocol.

스마트 쇼핑카트에 사용된 하드웨어 및 소프트웨어 모듈들은 Table 1과 같다.

The hardware and software modules used in the smart shopping cart is the same as Table 1.

Spec.	Component	M
HW	RaspberryPi	Raspb
	PiCamera	PiCa
	Ultrasonic distance sensor	HC
	weight sensor	Loadc
	Raspbian OS	Rasp
	Python	Pyt
	MJPG-streamer	MJPC
Server	YOLO	YOLO
Android	Android	API 14 4.0.3 (Ice
	JDK	Java SE nt E
DataBase	Apache	Apac (open
	MySQL	MyS
	php	php
	phpMyAdmin	phpMy

Table Overview of the modules used in Smart Shopping Cart

4.3 스마트 쇼핑카트의 설계 요구사항

4.4 Design requirements for smart shopping cart

스마트 쇼핑 카트를 설계할 때, 고려하였던 요구사항들은 다음과 같다. * 쇼핑의 편의성을 위하여 무선 배터리로 작동해야한다. * 최소 전력만을 소모해야 한다. * 카트에 동시에 여러 상품이 들어오거나 나가더라도 인식가능해야 한다. * 상품이 들어온 것인지 나간 것인지도 구분할 수 있어야 한다.

When designing smart shopping carts, the following requirements were considered. * For convenience of shopping, it should be operated with a wireless battery. * Only minimal power should be consumed. * It must be recognizable even if several products are simultaneously in or out of the cart. * It should be seperated whether a product came in or out.

4.5 스마트 쇼핑카트의 동작 순서

스마트 쇼핑카트의 동작 순서는 다음과 같다. 사용자가 쇼핑카트와 자신의 스마트폰을 연결시키며 쇼핑이 시작된다. 연결이 완료된 이후 사용자가 상품을 담거나 꺼내게 되면 쇼핑카트의 초음파 센서가 이를 인식하여 카트에 부착된 카메라로부터 입력된 영상을 7초 동안 라즈베리 파이에 저장한다. 이어서 TCP/IP 네트워크를 통해 스트리밍이 시작되었음을 알리는 메시지를 중앙 서버에게 송신한다. 중앙 서버는 라즈베리 파이에서 영상을 읽어온 후에 딥러닝을 이용하여 (상품의 종류, 수량, 출입 여부)를 판단하고 이 정보를 TCP/IP 네트워크를 이용하여 사용자의 스마트폰의 안드로이드 앱으로 전달한다. TCP/IP 메시지를 수신한 해당 앱에서는 가상 장바구니에 상품 데이터를 추가하거나 삭제한다. 쇼핑이 완료되면 사용자는 스마트폰에서 결제를 하고 쇼핑카트를 원래 위치로 반납한 후 마트에서 퇴장하며 쇼핑이 종료되고 쇼핑카트는 충전이 시작된다. 앞에서 설명한 과정을 Fig. 4에 나타내었다.

4.6 The order of operation of smart shopping cart

The order of operation of smart shopping cart is as follows.

Shopping begins when the user connects the shopping cart with his smartphone. After the connection is complete, the ultrasonic sensor in the shopping cart recognizes the product when the user puts or pulls out the product and stores the video entered from the camera attached to the cart in the Raspberry Pi for seven seconds. Then, it sends a message to the central server indicating that streaming has started over the TCP/IP network. The central server reads the video from the raspberry pie and then uses deep learning to determine (the type, quantity, and entry or exit of the product) and passes this information to the Android app on the user's smartphone using a TCP/IP network. Upon receiving the tcp/ip message, the app adds or deletes product data in the virtual shopping cart. When shopping is completed, the user pays with his or her smartphone, returns the shopping cart to its original location and exits the mart while closing their shopping. Then the shopping cart begins charging. The above-mentioned process is shown in Fig. 4

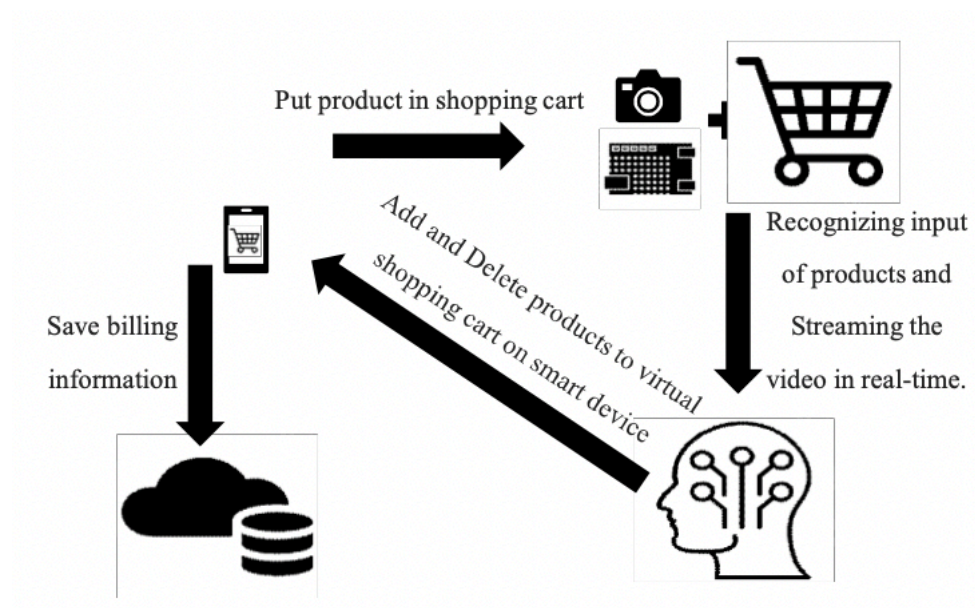


Fig. System operation diagram of smart shopping cart

5 스마트 쇼핑 카트 시스템의 세부 구현

6 Detailed implementation of smart shopping cart systems

6.1 스마트 쇼핑카트의 구현

스마트 쇼핑카트에는 라즈베리파이 보드를 부착하였다. 해당 보드에 내장된 Wi-Fi 네트워크 기능을 이용하여 인터넷에 접속해 중앙 서버와의 TCP/IP 서버-클라이언트 네트워크를 구현한다. 영상을 받아들이기 위하여 파이 카메라를 이용하였다. 파이 카메라와 MJPG Streamer 이용하여 jpg 형식의 촬영을 연속적으로 수행한 후에 라즈베리파이 호스트의 특정포트로 실시간으로 영상을 스트리밍한다. 중앙 서버에서는 해당 URL을 이용하여 영상 리소스에 접근할 수 있다.

6.2 Implementing smart shopping cart

The smart shopping cart has a Raspberry Pi board. It uses Wi-Fi network features built into the board to connect to the Internet and implement TCP/IP server-client networks with the central server. We used a Pi Camera to accept the video. Using a pie camera and an MJPG streamer, the video is streamed to a specific port on the raspberry pie host in real time after continuous jpg-type filming. The central server can access video resources using the URL.

6.3 전력 절약 방법

스마트 쇼핑카트는 배터리를 부탁하여서 동작하므로 전력을 절약하는 것이 매우 중요하다. 본 시스템에서는 Fig. 1에서 보는 바와 같이 초음파센서를 카트의 상단에 부착하여 이것을 영상 스트리밍의 트리거로 사용한다. 즉 영상은 쇼핑시간 내내 전송되는 것이 아니라 카트에 물건을 담거나 꺼낼 때에만 영상 전송을 시행하고, 이를 중앙 서버에게 알리도록 하는 비동기 방식의 전송을 구현하였다. 초음파센서는 탐지 범위가 쇼핑카트의 면적에 비하여 다소 좁은 편이므로 프로토타입에서는 세 개의 센서를 사용하여 이 점을 보완함으로써 정확도를 높였다.

6.4 Power Saving Method

It is very important to save power because smart shopping cart operates by asking for batteries. As seen in Figure 1, this system attaches an ultrasonic sensor to the top of the cart and uses it as a trigger for image streaming. In other words, the video was implemented as an asynchronous transmission that would require the central server to be notified only when the products were put in or taken out of the cart without continuing to be transmitted during shopping hours. Since ultrasonic sensors have a narrow detection range compared to the area of shopping carts, the prototype uses three sensors to compensate for this point to increase accuracy.

6.5 무게 센서의 이용

상품이 쇼핑카트 안으로 들어오는 것인지 밖으로 나가는 것인지를 판단하기 위한 방법으로 무게센서를 활용하였다. Fig. 1에서 보는 바와 같이 무게 센서를 쇼핑카트의 바닥에 설치하고, 이전 무게 대비 현재 무게를 비교하여 증감여부를 판단한다.

6.6 Using a Weight Sensor

Weight sensors were used as a way to determine whether a product enters or goes out of a shopping cart. As seen in Figure 1, the weight sensor is installed on the bottom of the shopping cart and the current weight compared to the previous weight to determine whether to increase or decrease.

6.7 안드로이드 앱

사용자는 앱 실행한 후에 자신이 고른 쇼핑 카트에 적혀진 단말기 고유 번호를 입력한다. 스마트폰의 고유 번호와 해당 카트의 고유 번호가 중앙 서버에서 매핑이 되면 쇼핑 환경이 갖춰진 것이므로 쇼핑을 시작하면 된다. 안드로이드 앱은 서버가 인식한 상품의 번호와 수량, 출입정보 데이터를 임시로 저장한다. 이후 마트의 상품 데이터들이 저장되어 있는 데이터베이스에 상품번호로 해당 상품을 조회하는

쿼리문을 보내고 응답으로 조회된 상품과 관련된 세부 정보들이 담긴 데이터를 받아 앱의 가상 장바구니에 추가하거나 삭제한다. 쇼핑이 완료되면 결제하기를 이용하여 총 구매내역과 산정된 금액을 보여주며 결제로 이어지도록 구현했다.

앱의 일부 화면을 Fig. 5에 보였다.

6.8 Android Application

After the user runs the app, the user enters the terminal unique number written on the shopping cart chosen by oneself. Once the unique number of the smartphone and the cart are mapped on the central server, the shopping environment is in place, so you can start shopping. Android application temporarily store product number, quantities and access information data recognized by the server. Afterwards, it sends a query to the database where the product data of the mart is stored and receives data containing details related to the product inquired by the product number and adds or deletes it in the app's virtual shopping cart. When shopping is completed, it is implemented to show the total purchase details and calculated amount using payment. Some screens of the application were shown in Fig. 5.

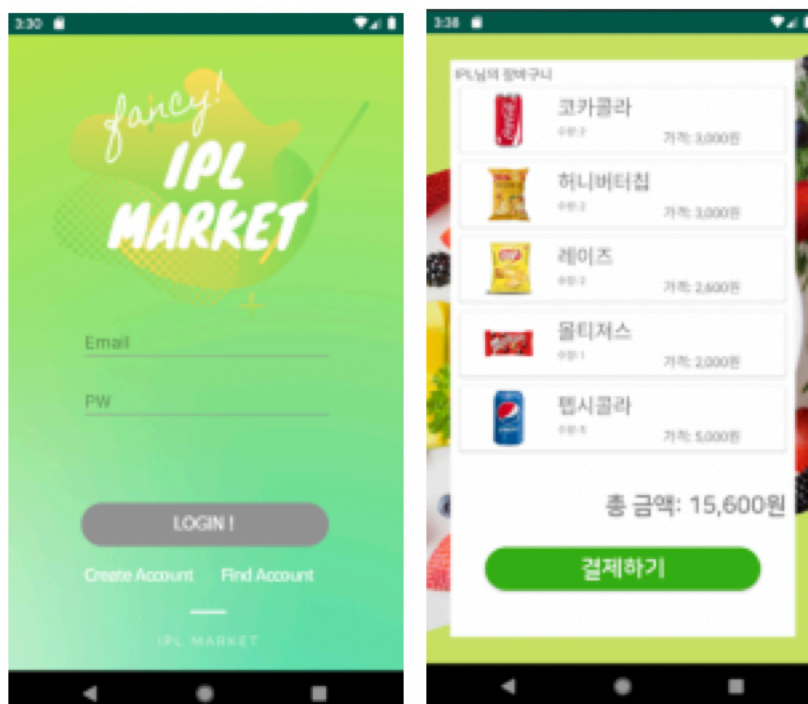


Fig.

The Screen of the Smartphone App

6.9 객체 탐지 모듈

스마트 쇼핑카트에서 카트에 담기는 상품을 정확하게 인식하는 것은 아주 중요한 작업이다. 쇼핑카트는 카메라를 통하여 이미지를 받은 후에 이미지 안의 객체를 탐지하고 객체의 위치를 정확하게 파악하여야 한다. 우리는 사용자가 카트로 동시에 여러 개의 상품을 담는 상황도 고려하여야 한다.

본 논문에서는 딥러닝 기술 중 하나인 그리드 기반의 객체 탐지를 제공하는 YOLO 라이브러리(You Only Look Once, 이하 YOLO)를 사용한다. YOLO는 분류기 역할 뿐 아니라 위치 알고리즘도 내장되어, 한 이미지 안에서 여러 객체를 탐지할 수 있도록 구현된 오픈소스 라이브러리이다 [Redmon et al(2016)Redmon, Divvala, Girshick, and Farhadi].

YOLO는 이전까지의 객체 탐지 신경망보다 우수한 성능을 제공하는 오픈소스 라이브러리이며, 파일 형식의 영상데이터 뿐 아니라 웹캠을 통한 영상에 대해서도 실시간 탐지까지 지원하는 강력한 실시간 객체 탐지 라이브러리이다. YOLO는 합성곱 신경망(Convolutionay Neural Network, 이하 CNN)에 기반을 두고 있어 합성곱 계층과 서브샘플링 계층을 거치는 등 입력 이미지의 크기를 점차 다운샘플링하는 컨볼루션 기반의 아키텍처를 제공한다. Fig. 6에서는 YOLO에 내장된 네트워크의 전 연결 신경망(Fully Connected Network, 이하 FCN)의 앞단, 즉 합성곱 계층을 보여준다.

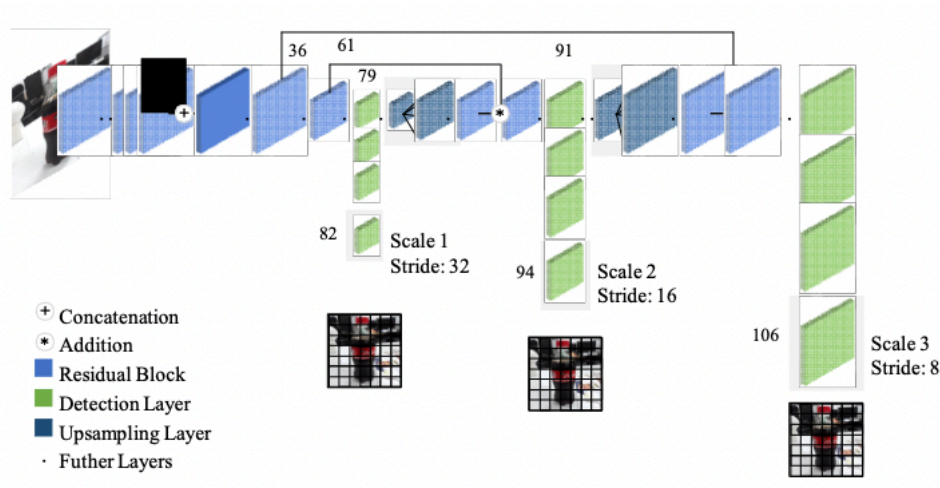


Fig.

Neural network architecture of YOLO object detection

YOLO는 하나의 CNN을 사용하여 이미지 전체에서 다수의 bounding box를 예측하고, 동시에 각 박스에서 class probability를 계산하는 통합된 모델을 사용한다. Fig. 7은 YOLO 학습이 완료된 이후에 상품을 인식하는 과정을 보여준다.

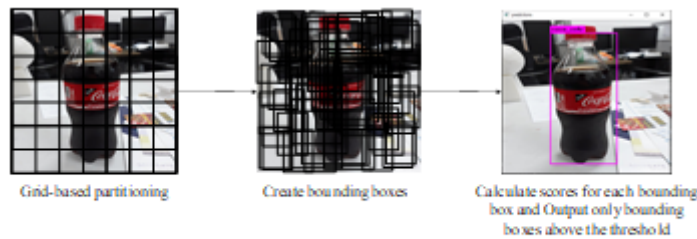


Fig. 7. Workflow of YOLO object detection

Fig.

Workflow of YOLO object detection YOLO는 기존의 R-CNN보다는 약 1000배, Fast R-CNN보다는 약 100배, 가장 최근에 나온 Faster R-CNN보다도 약 10배 이상 빠른 성능을 보인다[7].

6.10 Object detection

In a smart shopping cart, it is a very important task to accurately recognize the products added to the cart. After receiving the image through the camera, the shopping cart should detect the object in the image and accurately figure out the location of the object. We should also consider the situation in which the user puts several products into the cart at the same time.

The smart cart system proposed in this paper uses a YOLO library (You Only Look Once, hereinafter YOLO), which provides grid-based object detection, one of the deep learning technologies. YOLO is an open source library that implements not only a classifier but also location algorithms, enabling multiple objects to be detected within one image [Redmon et al(2016)Redmon, Divvala, Girshick, and Farhadi].

YOLO is an open-source library that provides better performance than previous object detection neural networks, and is a powerful real-time object detection system that supports real-time detection of videos via webcam as well as file format video data. Because YOLO is based on CNN, it provides a convolution-based architecture that gradually downsamples the size of the input image, such as going through the convolutional layer and subsampling layers. Fig. 6 shows the convolutional layer in front of the fully connected neural network (FCN) of the network embedded in YOLO.

YOLO uses a single CNN to predict a number of bounding boxes across the image, and uses an integrated model to calculate class probability in each box at the same time. Fig. 7. shows the progress of detecting products after training.

YOLO performs about 1,000 times faster than traditional R-CNN, about 100 times faster than Fast R-CNN, and about 10 times faster than the most recent Faster R-CNN [7].

6.11 YOLO 학습 과정

YOLO도 지도학습으로 문제를 해결하는 것이기 때문에 레이블링이 잘 된 양질의 데이터 확보가 필요하다. 객체 탐지 문제의 경우에는 정답 레이블은 각 객체의 레이블명과 경계박스의 쌍으로 구성되며 이를 Annotation이라고 부른다.

본 연구에서는 5개의 상품 클래스만을 가정하였고 각 클래스 당 촬영을 통하여 300-500장 정도 학습 데이터를 생성하였다. 이 학습 데이터는 Fig. 8에서 보듯이 Annotation tool인 YOLO-mark을 이용하여 총 2800장의 학습용 데이터의 리사이징 및 라벨링 등의 전처리 작업을 진행하였다.



Fig. 8. Annotation with YOLO-mark

Fig.

Annotation with YOLO-mark

YOLO 신경망에 대한 학습률은 0.001, 모멘텀은 0.9, weight decay는 0.0005로 설정하였으며, 16200 epoch 동안 학습을 진행하였고 최종 Loss 함수 값은 Fig. 9에서와 같이 0.0616으로 확인되었다. 프로토타입 모델에서는 threshold를 0.6으로 높여주어 잘못된 상품이 입력되는 비율을 줄였다.

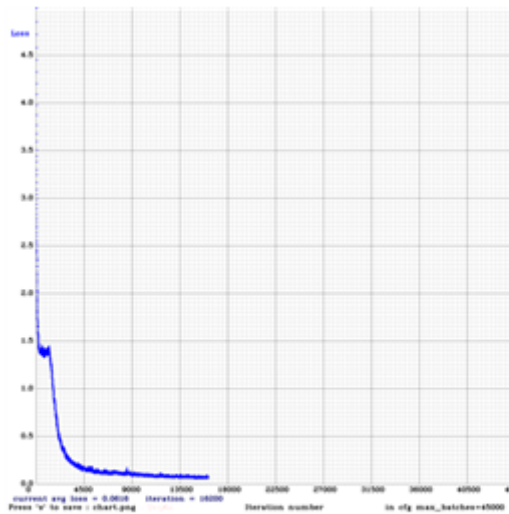


Fig. 9 Graph of loss function after training 16200 epoch

Fig.

Graph of loss function after training 16200 epochs

6.12 YOLO Learning Process

Since YOLO also solves the problem through supervised learning, it is necessary to secure high-quality, well-labeled data. In the case of the object detection problem, the correct answer label consists of a pair of each object's label name and a bounding box, and this is called an annotation.

In this study, only 5 product classes were assumed, and 300-500 learning data were generated through shooting using cameras for each class. As shown in Fig. 8, this learning data was pre-processed such as resizing and labeling of 2,800 learning data using the annotation tool YOLO-mark.

The learning rate for the YOLO neural network was set to 0.001, the momentum to 0.9, and the weight decay to 0.0005. As shown in 9, the training was conducted for 16200 epochs, and the final Loss function value was found to be 0.0616. In the prototype model, we increased the threshold to 0.6 to reduce the rate at which the wrong product was entered.

6.13 상품 인식을 실험

학습이 종료되면 학습된 가중치 파일을 가지고 실시간 탐지를 테스트해볼 수 있다. 상품 중에서 콜라, 버터링, 라면, 다우니, 콘프로스트의 5종의 상품을 가지고 테스트하였다. 1회의 트리거 당 스트리밍 데이터를 가져와서 실시간으로 39회의 탐지를 실행하는데 이 39회에 대한 평균 탐지 정밀도를 계산하여 한 회차의 결과로 기록하였다. 이 과정을 상품 클래스별로 각 10회씩 시행하였고, 각 클래스별 평균

예측 값들로부터 본 시스템 전체의 평균 예측 정밀도(mean Average Precision, mAP)를 산정하였다. 이 결과를 Table 2에 나타내었다. 본 시스템의 평균 정확도는 82.28%로 평가되었다.

	coca- cola	butte- r_rin- g	rame- n	dow- ny	com- frost	mAP
1	57.5 9	96.0 3	90.6 9	92.4 6	87.3 3	82.2 8
2	77.1 0	87.8 5	70.5 9	89.3 8	85.6 9	
3	77.0 5	96.7 9	91.7 9	89.2 3	83.3 3	
4	80.1 5	93.6 9	64.6 2	90.7 7	87.4 7	
5	54.1 5	91.5 6	88.1 8	82.5 1	84.3 8	
6	83.1 8	95.0	78.7 2	90.2 6	84.9 2	
7	36.1 5	85.5 9	67.2 3	90.1 3	79.3 3	
8	78.1 3	93.2 8	92.5 4	86.5 4	88.4 8	
9	54.4 1	50.9 2	79.5 9	90.9 7	89.5 3	
10	69.2 6	90.9 2	94.4 1	87.2 3	87.1	
average	66.7 2	88.1 5	81.8 4	88.9 5	85.7 6	

Table AP of each classes and mAP

Table 2에서 특정 테스트 회차에서의 인식률이 해당 클래스 평균 정확도에 비해 다소 낮게 측정된 경우가 있는 것을 알 수 있는데, 그 이유는 고의적으로 상품을 회전시키는 등, 인식이 어려울 수 있는 환경을 만들어 테스트를 했기 때문이다.

YOLO는 다른 탐지시스템과 비교하여 가장 빠른 속도로 실시간 객체 탐지를 제공한다는 강력한 장점을 지녔지만, 이전의 탐지 시스템에 비하여 작은 물체에 대해서는 검출률이 떨어진다는 단점이 있어 이를 개선하기 위한 YOLO의 장점과 이전의 객체 탐지 시스템의 장점을 모두 고려한 SSD(Single Shot Multibox Detector, 이하 SSD)라는 새로운 객체 탐지시스템도 대두가 되고 있다. SSD는 YOLO와 비교하여 속도는 떨어지나 성능은 우수하다 [Ren et al(2015)Ren, He, Girshick, and Sun, Liu et al(2016)Liu, Anguelov, Erhan, Szegedy, Reed, Fu, and Berg].

6.14 experiment for the ratio of product recognition

When learning is finished, real-time detection can be tested with a file containing the learned weights. We tested with five products: Coke, Buttermilk, Ramen, Downey and Confront. Streaming data is fetched and the number of detections per trigger in real time is 39. The average detection precision for these 39 detection result is calculated and recorded as the result of one round. This process was performed 10 times for each product class, and the mean average precision (mAP) of the entire system was calculated from the average prediction values for each class. Table 2 shows the results. The Map of the system was estimated to be 82.28%.

In Table 2, it can be seen that the recognition rate at a specific test round was measured slightly lower than the average accuracy of the class. The reason for this is that we tested by creating an environment that can be difficult to recognize, such as intentionally rotating a product.

YOLO has the strong advantage of providing real-time object detection at the fastest speed compared to other detection systems, but it has the disadvantage that the detection rate is low for small objects compared to previous detection systems. A new object detection system called SSD (Single Shot Multibox Detector, SSD) that considers all the advantages is also emerging. SSDs are slower than YOLO, but are superior in performance [Ren et al(2015)Ren, He, Girshick, and Sun, Liu et al(2016)Liu, Anguelov, Erhan, Szegedy, Reed, Fu, and Berg].

7 결론

무인 상점의 대표주자로 알려진 아마존 고의 정확도는 합격점이라는 평가가 잇따르며, 아마존 고의 아이디어와 기술력은 인정받았지만, 널리 확산시키기엔 상당한 비용이 든다. 또 입장할 수 있는 고객의 수가 한정적이다.

본 논문에서 제안된 스마트카트 시스템은 다른 무인 상점 솔루션에 비하여 높은 가성비를 제공한다. 제안된 시스템에서는 수용 인원이 증가하더라도 별 영향을 받지 않는다. 또 비교적 적은 컴퓨팅 자원을 사용한다. 또 기존의 방법처럼 모든 상품에 RFID를 부착할 필요가 없다.

상품 탐지 분야에 있어서 속도와 정확도는 트레이드-오프 관계인데, 본 시스템은 실시간 처리를 요구하기 때문에 YOLO를 응용하여 구현하였다. 차후 이 부분은 객체 탐지 모델의 수정 및 변경을 통하여 충분히 성능을 더 높일 수 있으리라고 추측한다.

8 Conclusion

The accuracy of Amazon Go, known as the representative of unmanned shops, is said to be successful, and Amazon Go's ideas and skills have been recognized by the public, but it is expensive to spread widely. Also, the number of customers who can enter at the same time is limited.

The smart cart system proposed in this paper provides a higher cost-performance ratio than other unmanned store solutions. In the proposed system, even if the capacity increases, it is not affected. It also uses relatively little computing resources. Also, it is not necessary to attach RFID to all products as in the conventional method.

In the field of product detection, speed and accuracy are trade-off, but since this system requires real-time processing, it was implemented by applying YOLO. In the future, it is assumed that this part can be improved sufficiently by modifying and changing the object detection model.

References

- Chiang et al(2016)Chiang, You, Lin, Shih, Liao, Lee, and Chen. Chiang HH, You WT, Lin SH, Shih WC, Liao YT, Lee JS, Chen YL (2016) Development of smart shopping carts with customer-oriented service. In: 2016 International Conference on System Science and Engineering (ICSSE), IEEE, pp 1–2
- Karjol et al(2017)Karjol, Holla, and Abhilash. Karjol S, Holla AK, Abhilash C (2017) An iot based smart shopping cart for smart shopping. In: International Conference on Cognitive Computing and Information Processing, Springer, pp 373–385
- Liu et al(2016)Liu, Anguelov, Erhan, Szegedy, Reed, Fu, and Berg. Liu W, Anguelov D, Erhan D, Szegedy C, Reed S, Fu CY, Berg AC (2016) Ssd: Single shot multibox detector. In: European conference on computer vision, Springer, pp 21–37
- Redmon et al(2016)Redmon, Divvala, Girshick, and Farhadi. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 779–788
- Ren et al(2015)Ren, He, Girshick, and Sun. Ren S, He K, Girshick R, Sun J (2015) Faster r-cnn: Towards real-time object detection with region proposal networks. In: Advances in neural information processing systems, pp 91–99
- Wankhede et al(2018)Wankhede, Wukkadada, and Nadar. Wankhede K, Wukkadada B, Nadar V (2018) Just walk-out technology and its challenges: a case of amazon go. In: 2018 International Conference on Inventive Research in Computing Applications (ICIRCA), IEEE, pp 254–257