

<https://doi.org/10.7236/IIBC.2018.18.6.243>

IIBC 2018-6-33

실시간 객체 추적 기술을 활용한 스마트 케어 시스템에 대한 연구

A Study on the Smart Care System Using Real-time Object Tracking Technology

김혜정*, 강민구*, 이해규*, 고동범*, 김정준**, 박정민***

HyeJeong Kim*, MinGu Kang*, HyeGyu Lee*, Dongbeom Ko*, JeongJoon Kim**,
Jeongmin Park***

요 약 본 논문은 독거노인을 위한 스마트 케어 시스템을 설계하고 구현한다. 최근 의학, 생활수준, 환경의 급속한 개선으로 생활수준이 높아짐에 따라 고령 인구의 비율이 높아지고 있다. 또한 고령화 사회와 더불어 증가하는 독거노인 비율은 노인 케어 시스템, 응급 상황 알림과 같은 서비스의 제공이 중요한 이슈가 되고 있다는 것을 시사한다. 그러나 기존의 응급상황 알림을 위한 기술들은 고정된 CCTV 영상을 분석하기 때문에 CCTV 사각지대에서의 모니터링에 대한 어려움과 카메라가 설치되지 않은 장소로 이동하는 등 정해진 범위를 벗어나는 활동을 할 경우 실시간으로 케어 할 수 없다는 문제점이 존재한다. 따라서 본 논문에서는 이러한 단점을 보완하기 위해 실시간으로 이동이 가능한 로봇과 객체 추적 기술을 활용한 스마트 케어 시스템을 설계하고 개발한다. 이를 통해 장소에 구애받지 않고 실시간 모니터링이 가능하며, 응급상황시 빠르게 도움을 요청해 보호자 및 도우미들에게 편의를 제공할 수 있도록 한다.

Abstract This paper designs and implements a smart care system for the senior citizen who lives alone. Recently, as the level of living has increased due to the rapid improvement of medicine, living standard and environment, the proportion of the elderly population is increasing. In addition, the proportion of the elderly living alone, which is increasing with the aging society, suggests that the provision of services such as the elder care system and emergency notification is becoming an important issue. However, since the existing emergency notification technology analyzes fixed CCTV images, it is difficult to monitor in the blind spot of CCTV and to move to a place where the camera is not installed. There is a problem that it can not be performed. Therefore, in this paper, we design and develop a smart care system that utilizes robot and object tracking technology that can move in real time to overcome these shortcomings. This enables real-time monitoring regardless of the location, and prompts for assistance in case of an emergency, so that it can provide convenience to cares and assistants.

Key Words : Object detection, Object tracking, elder person, Smart care system, Emergency notification system

*준회원, 한국산업기술대학교

**정회원, 한국산업기술대학교 컴퓨터공학과 조교수

***정회원, 한국산업기술대학교 컴퓨터공학과 조교수(교신 저자) *Corresponding Author: jmpark@kpu.ac.kr

접수일자: 2018년 9월 5일, 수정완료: 2018년 11월 5일

게재확정일자: 2018년 12월 7일

Received: 5 September, 2018 / Revised: 5 November, 2018 /

Accepted: 7 December, 2018

Dept. of Computer Engineering, Korea Polytechnic University,
Korea

I. 서 론

최근 전 세계적으로 노인 인구 비율이 증가함에 따라 독거노인의 비율이 증가하고 있으며, 이에 따라 독거노인 케어에 관련된 안전 및 응급상황 알람을 위한 기술들이 개발되고 있다. 이러한 기술들과 관련 제품들은 단순한 부착형 센서^[1] 및 카메라^[2]에서부터 웨어러블 기기^[3-5]까지 다양한 형태로 연구 및 개발되고 있다.

또한 여러 촬영 장비 및 영상 처리 분야의 기술들이 나날이 발전하면서 이와 관련된 기술들의 중요성이 부각되고 있다. 특히 영상 기술들의 발전과 함께 영상 감시 분야의 중요성이 증가하고 이와 함께 다양한 지능형 감시 시스템들에 대한 관심과 연구가 지속되고 있다^[6]. 그러나 현재 개발되고 있는 영상을 통한 감시 시스템들은 대부분 CCTV의 형태로 제한이 되어있기 때문에 한자리에 부착된 장소 제한적인 문제점과 사각지대가 존재할 경우 이와 관련된 해결책을 제시하기 어렵다는 문제점을 갖고 있다^[2].

이러한 단점들은 카메라뿐만 아니라 센서 및 웨어러블 기기들을 이용한 모니터링 시스템에서도 마찬가지로 볼 수 있다. 센서를 이용한 모니터링 시스템들의 경우 또한 센서가 부착된 곳 즉, 중심이 되는 중요한 위치에 부착이 되지만 해당 위치를 벗어나거나 거리가 있는 곳 또는 센서의 범위를 벗어난 다른 장소에서는 상태 확인 및 모니터링을 할 수 없다는 단점이 있다^[3]. 또한 웨어러블 기기는 몸에 부착이 되어있지 않은 경우 제대로 사용할 수 없기 때문에 항상 소지하고 다녀야한다는 불편함이 있따^[4-5]. 따라서 이러한 문제점들을 해결하기 위한 모니터링 시스템을 개발할 필요가 있다.

본 논문에서는 기존 연구 및 실험들의 장소 및 위치 의존적이며 일상생활에 따른 제약이 생기는 단점을 보완하기 위해 카메라를 통하여 객체를 인식하여 인식한 객체를 따라 이동하는 실시간 객체 추적 시스템을 설계하고 개발한다.

II. 관련 연구

1. 기존 시스템

기존의 독거노인 케어를 위한 시스템은 센서를 이용하거나 웨어러블 기기를 이용한다.

센서 및 카메라를 이용하는 경우에는 실시간으로 모니터링하며 심박, 호흡, 맥박, 활동 상태 등을 측정하여 단순히 생사 여부뿐만 아니라, 생활 패턴을 분석하여 이상이 있을 경우 보호자에게 알람이 간다. 그러나 센서 및 카메라는 특정 위치에만 있다는 한계가 있어 해당 기기들의 시야 및 범위 밖으로 벗어나면 사용할 수 없다는 단점이 있다.

웨어러블 기기를 사용하는 경우에는 손목시계, 신발 내장형 등의 형태로 신체 또는 다른 사물에 부착을 하여 맥박, 활동성 등을 체크한다. 또한 위와 마찬가지로 생활 패턴을 수집, 분석하는데 이는 야외에서도 실시간으로 정보를 받아들일 수 있는 장점이 있으나 제품이 신체에 닿아있지 않으면 제대로 사용할 수 없다는 단점이 있다.

또한 위의 두 시스템은 위기 상황이 아닌 단순한 일상 생활에서 도움이 필요할 때 도움을 줄 수 없다는 단점이 있다. 이와 같이 위치 및 신체 의존적인 단점과 응급상황만 관리할 수 있는 단점들을 해결하기 위해서는 여러 장소를 이동할 수 있으며 별도의 디바이스로 사람의 신체에 부착하지 않아도 객체를 스스로 인식하여 모니터링할 수 있는 기술을 적용할 필요가 있다.

2. YOLO^[6]

YOLO는 You Only Look Once의 약자이다. YOLO가 등장할 당시 Object Detection은 주로 Faster R-CNN(Region with Convolution Neural Network) 계열이 좋은 성능을 내고 있었다. 그러나 R-CNN은 매우 느렸고(5초), Fast R-CNN도 0.5프레임으로 느렸으며, Faster R-CNN도 7프레임이 최대였다. 이때 YOLO가 등장하여 45프레임을 보여주었으며 빠른 버전의 경우 155프레임을 기록하며 사람들을 놀라게 했다. 게다가 성능도 Faster R-CNN에 비해 크게 떨어지지 않았다. R-CNN 계열의 검출 네트워크들은 영상에서 오브젝트가 있을 것 같은 후보(ROI - Region Of Interest)를 먼저 뽑고 후보로 뽑힌 ROI(영상의 작은 부분)들은 분류기(Classification network)에 의해 클래스 분류가 이뤄지고 경계박스(Bound Box)를 찾는다. 이와 달리 YOLO는 네트워크의 최종 출력 단에서 객체(물체)의 Bounding Box(경계 박스)와 Bounding Box안의 Object(객체)가 무엇인지 동시에 예측한다. 즉, 경계박스의 위치가 찾아지는 동시에 경계박스 내에서 클래스 분류가 동시에 이루어진다. 결국 단 하나의 네트워크가 한 번에 특정도 추출

하고, 경계박스도 만들고, 클래스도 분류기 때문에 간단하고 빠르다.

3. SSD [7-8]

SSD(Single Shot MultiBox Detector)는 오브젝트 검출기이며 Faster R-CNN보다는 성능이 조금 낮지만 YOLO보다는 성능과 속도에서 앞선다. SSD는 객체 검출 속도 및 정확도 사이의 균형이 있는 알고리즘으로 한 번만 입력 이미지에 대한 CNN을 실행하고 형상 맵(Feature map)을 계산한다. 경계 상자 및 객체 분류 확률을 예측하기 위해 이 형상 맵을 3 x 3 크기로 CNN을 수행한다. SSD는 CNN처리 후 경계 상자를 예측한다. 이 방법은 다양한 스케일의 물체를 검출할 수 있다.

4. Tensorflow

Tensorflow는 구글(Google)에서 기계학습과 딥러닝을 위해 만든 오픈소스 라이브러리이다. 데이터 플로우 그래프(Data Flow Graph)방식을 사용하여 수치 연산을 하며 수학 계산과 데이터의 흐름을 노드(Node)와 엣지(Edge)를 사용한 방향 그래프(Directed Graph)로 표현한다. 본 논문에서는 Tensorflow와 함께 Haar Classifier^[9-10]을 사용하여 객체를 인식한다.

5. Summary

앞서 설명한 기술들을 적용하여 본 논문에서는 Tensorflow와 Opencv를 이용하여 카메라 영상을 받아 SSD 알고리즘을 통해 카메라 영상에서 객체를 인식한다. 그리고 인식한 객체의 위치 좌표 값을 구하여 이 값을 이용해 인식한 객체를 트래킹하며 모니터링 및 케어 서비스를 제공할 것이다.

III. 시스템 프로세스 및 구조

본 장에서는 객체를 트래킹 및 독거노인을 케어하기 위한 시스템 프로세스와 시스템 구조를 설명한다.

1. 시스템 프로세스

본 실험에서는 영상을 처리하여 로봇의 움직임을 제어하기 위한 processing server와 사용자의 로봇과 보호자 및 봉사자들의 안드로이드 간 데이터 송수신을 위한

Communication Server 총 두 개의 서버가 사용된다.



그림 1. 모니터링 및 긴급 상황 알림을 위한 프로세스
 Fig. 1. Process of Monitoring and Emergency Notification

- Step1: Collect Image data

로봇의 카메라를 이용하여 실시간으로 영상 데이터를 수집한다. 영상은 base64로 인코딩된 이미지 배열이 문자열로 processing server에 전송된다.

- Step2: Image data process

로봇에서부터 processing server로 전송된 데이터에서 객체(사용자)를 추출하여 네모 박스를 씌운 후 해당 박스의 중간 좌표(x, y)값을 찾아 낸 후 로봇으로 전송한다.

- Step3: Object Tracking

processing server로부터 중앙 좌표 값 데이터를 받은 로봇은 카메라 화면의 중심을 좌표 값으로 맞추면서 객체를 향해 이동한다.

- Step4: Notify the Crisis with Android

로봇과 사용자는 일정 거리를 유지하며 사용자가 로봇의 긴급 버튼을 누를 경우 긴급 상황 데이터(사용자 이름, 위치, 시간)가 Communication Server로 전송되며 Communication Server는 응급 상황을 등록된 보호자 및 응급 상황이 보고된 위치를 기준으로 가장 가까운 봉사자들의 안드로이드로 전송하여 위험 상황을 알린다.

- Step5: Emergency resolution

알림을 받은 보호자 및 봉사자는 긴급 상황 데이터에 등록된 주소로 이동하여 사용자의 상태를 확인하고 Communication Server는 안드로이드를 통해 긴급 상황 해결 관련 정보를 받으면서 긴급 상황을 종료한다.

2. 시스템 구조

케어 로봇은 카메라 영상을 통해 객체를 인식하여 바퀴를 통해 사용자를 트래킹하며 초음파 센서를 이용하여 일정 거리를 조절한다. 트래킹의 과정은 Step1~3과 같으며 로봇은 사용자의 근처에 있기 때문에 사용자는 위급 상황 시 근처에 있는 로봇의 버튼을 빠르게 누름으로써 상황을 알릴 수 있다. 또한 내장된 마이크를 통해 보호자 및 도우미들과 간단한 대화 및 일상생활에서의 어려움 등을 상담하고 도움 받을 수 있다. 보호자 및 도우미는 모바일 어플리케이션을 이용하여 응급 상황을 전달받아 도움을 주고 응급상황뿐만이 아닌 일상생활에서의 곤란한 상황에 대한 답변 또한 제공할 수 있다.

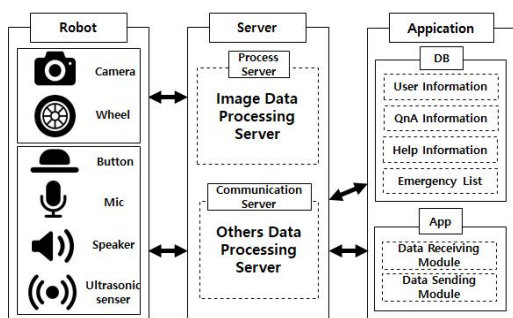


그림 2. 전체 시스템 구조

Fig. 2. Whole System Architecture

- Processing Server: Image Processing and Object Tracking

로봇은 카메라를 통해 영상을 Processing Server로 전송한다. Processing Server는 로봇에게 받은 영상을 base64로 인코딩된 이미지 배열의 문자열로 받아 이를 이용해 객체(사용자) 추출 및 추출된 객체의 중심 값을 찾아 이를 로봇에 반환한다. 중심 값을 반환받은 로봇은 해당 데이터를 이용하여 카메라의 중심을 객체에 맞추면서 이동할 수 있으며, 초음파 센서를 통해 일정 거리를 유지한 상태로 객체를 트래킹 한다.

- Communication Server(1): Communication between Robot and Android

로봇의 응급 버튼이 눌리면 버튼이 눌린 로봇 사용자의 정보 및 버튼이 눌린 시간이 DB에 저장되며 동시에 등록된 보호자 및 사용자의 거주지 근처의 도우미들의 모바일 어플리케이션으로 알림이 전송된다. 보호자 및 도우미들은 알림을 통해 응급상황 확인 및 대처 방법(신고, 직접 방문 등)을 선택하여 도움을 줄 수 있다. 응급상황이 아닌 단순 음성 입력의 경우 음성 내용이 STT를 통해 DB에 저장되며 보호자 및 도우미들은 모바일 어플리케이션의 게시판을 통해 이에 대한 답변을 줄 수 있으며 로봇은 답변이 등록될 경우 이를 사용자에게 TTS를 통해 읽어 도움을 줄 수 있다.

- Communication Server(2): DB 데이터 관리

Communication Server의 또 다른 역할 중 하나로 Database 관리가 있는데, 사용자의 정보를 관리할 뿐만 아니라, 응급 버튼이 눌렸을 때 해당 시간과 사용자 정보를 저장한 후 일정 시간이 경과 후에도 도움을 줄 수 있는 보호자 및 봉사자들의 응답이 없을 경우 바로 관련 응급의료기관에 신고가 될 수 있도록 한다. Database에서는 시간마다 응급 요청 상황을 체크하여 누락되는 사용자가 없도록 한다.

IV. 실험 및 평가

본 장에서는 카메라를 통해 들어오는 영상을 이용해 객체(사람)를 인식하여 인식된 객체의 중심 좌표를 반환받아 좌표 값을 이용해 트래킹 하는 시스템 데모를 소개하고 1) 입력 영상에서 객체 인식(Object Detection), 2) 인식된 객체의 중심 좌표 출력, 3) 반환받은 좌표를 이용한 로봇의 이동의 3단계 프로세스로 데모 및 실험을 진행한다. 실험을 위해 라즈베리파이에 필요 장비들(카메라, 바퀴, 초음파 센서 등)을 장착하였다. 이미지 처리 및 움직임 제어와 관련된 Processing Server는 라즈베리파이 자체에서, 그 외 모바일 어플리케이션과의 통신 관련된 Communication Server는 외부 서버에서 처리된다.

- 프로그램 구조 및 UI

실시간 객체 인식 및 트래킹 시스템은 로봇을 통해 받은 영상 데이터에서 실시간으로 객체를 인식하고, 해

당 객체의 중간 좌표를 반환한다. 그리고 반환된 중간좌표를 이용하여 인식한 객체를 일정 반경 안에서 거리를 유지하면서 트래킹 한다. 로봇 사용자 응급 상황의 경우 자신을 따라다니는 로봇의 상단의 응급버튼을 누름으로써 빠르게 외부에 응급상황을 알리고 해결할 수 있다.

• 1 단계: 입력된 영상의 객체 인식

먼저 로봇에 부착된 카메라를 통하여 로봇에 영상이 실시간으로 들어온다. 이를 Processing server에 base64로 인코딩된 이미지 배열이 문자열로 들어가게 된다.



그림 3. 객체 인식 및 중간 좌표 출력 전(위)과 후(아래)
 Fig. 3. Object recognition and intermediate coordinate output Before (above) and after (below)

• 2단계: 인식된 객체의 중심 좌표 반환

Processing server는 입력받은 이미지에서 객체를 추출한 후 객체에게 사각형 박스를 씌워 해당 박스의 중간좌표(x, y)를 출력하여 이를 다시 로봇으로 반환한다.

표 1. 좌표 정보 테이블

Table 1. Simulation Parameters

Data		Value
X	X Coordinate	272.5
Y	Y Coordinate	326.0
FPS	Frame rate	4.0246830495991
CM	Ultrasonic sensor Value	25.1

• 3단계: 중심좌표를 이용한 로봇 이동

Processing server에서 받아온 중간 좌표 값을 이용하여 해당 중심좌표(중심점)의 위치가 카메라의 중앙으로 오도록 로봇을 이동한다. 이 때, 사용자와의 거리와 센서를 이용하여 조절한다. 이를 통해 로봇은 일정 거리 안에서 사용자를 트래킹 할 수 있게 되며, 사용자는 자신의 손이 닿는 범위 내에 있는 로봇을 이용해 응급 상황 시 로봇 상단의 버튼을 누름으로써 빠르게 응급상황을 외부에 전달할 수 있다.

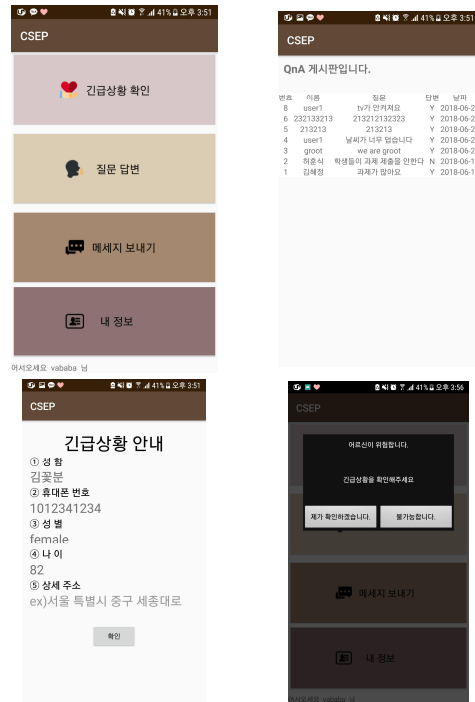


그림 4. 어플리케이션 UI 및 긴급 상황 시 알림 화면
 Fig. 4. Application UI and emergency notification screen

• 성능 평가

1단계에서 Processing server에 입력된 영상을 처리할 때 시간마다 얼마만큼의 속도로 객체가 인식되고 중심좌표가 출력되었는지를 확인한다. 이는 Tensorflow를 사용하기 때문에 CPU를 비교하고 이와 함께 컴퓨터별 사양 및 성능도 함께 기재하며 fps를 이용하여 비교한다. 표 2는 성능 평가를 위해 사용한 노트북 컴퓨터들의 CPU와 RAM 그리고 OS를 정리하였으며, 그림 6은 해당 성능의 노트북 컴퓨터들이 각각 얼마만큼의 fps가 나왔는지 비교하여 그래프를 그린 것이다. 이들은 본 논문의 실험을

위해 사용된 라즈베리파이와 함께 비교하여 성능에 따라 얼마만큼의 차이가 있는지를 확인한다.

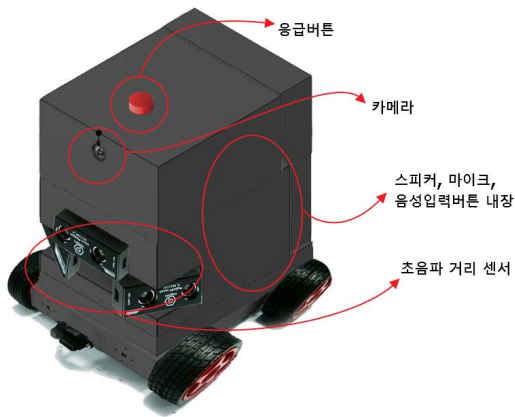


그림 5. 로봇 디자인(위) 및 실제 모습(아래)
Fig. 5. Robot design (above) and actual appearance (below)

표 2. 성능 비교를 위한 각 노트북 컴퓨터의 사양
Table 2. Specifications of each notebook computer for performance comparison

	CPU	RAM	OS
A	Intel Core(TM) i7-7500U	8GB	Windows 10
B	Intel Core(TM) i5-5257U	8GB	MacOS High Sierra
C	Intel Core(TM) i7-7700HQ	8GB	Windows 10
Raspberry Pi	1.2GHz ARM Cortex-A53 MP4	1GB	Raspbian

아래는 각 모델에 대한 fps를 나타낸 그래프이다.

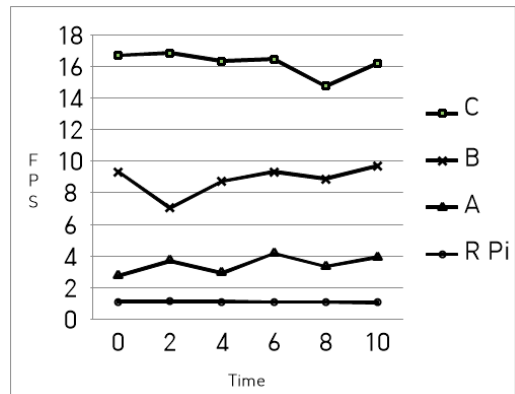


그림 6. 사양별 FPS 비교
Fig. 6. FPS comparison by specification

아래의 [표 3]은 이전에 소개한 YOLO 알고리즘과 위 실험을 위해 사용한 SSD 알고리즘의 정확도와 FPS를 비교^[8]한 표이다.

표 3. YOLO와 SSD의 정확도 비교 결과
Table 3. Comparison result of accuracy and FPS between YOLO and SSD

Method	Test accuracy	FPS
SSD algorithm	0.6503	53
YOLO algorithm	0.6215	42

실험을 통해 Object Detection을 장비에 따라 비교할 경우 라즈베리파이는 거의 0.7~1.5 정도로 매우 낮게 나오는 반면 성능이 좋은 CPU로 이동할수록 더욱 높은 fps 값에 처리되는 것을 확인할 수 있다. 그러나 본 논문은 라즈베리 파이와 같은 소형 PC에 적합하게 디자인 및 구상이 되었기 때문에 이를 보완하기 위한 더 좋은 성능의 라즈베리파이 또는 소형 PC가 필요하다. 따라서 데이터 처리 속도를 증가시킬 수 있는 알고리즘 및 컴퓨터 성능 개발에 대한 후속 연구를 진행할 필요가 있다.

V. 결론

본 논문에서는 영상을 이용한 객체 추출 및 트래킹을 실험하기 위해 1) Collect Image data, 2) Image data process, 3) Object Tracking, 4) Notify the Crisis with

Android, 5) Emergency resolution 의 단계를 제안하였다. 이는 라즈베리 파이에 카메라, 바퀴 및 여러 센서들을 부착한 형태로 구현되었으며, 응급 상황 및 일상생활에서의 도움을 주기 위한 방법으로 보호자 및 도우미에게는 전용 모바일 어플리케이션이 추가가 되었다. 이를 통해 로봇으로는 실시간으로 객체를 추적하면서 응급상황 시 빠르게 도움을 요청할 뿐만 아니라 일상생활에서의 어려움을 쉽게 외부에 물어볼 수 있으며 보호자 및 도우미들은 어플리케이션을 이용하여 간편하게 도움을 줄 수 있다. 그러나 실시간으로 객체를 인식한 후 인식한 객체를 정확하게 트래킹하기 위해서는 이미지 처리 부분에서 높은 성능의 컴퓨터를 필요로 하지만 라즈베리파이에서는 속도적인 측면에서 한계가 있다. 이는 앞으로 더 좋은 성능의 라즈베리파이 또는 소형 PC를 이용하여 해결될 수 있다.

References

- [1] Jooyoung Ko, Hyenki Kim, "A study on the Monitoring System for Emergency Recognition of Elderly People Living Alone", Journal of Korean Institute of Information Technology 12(3), 61-68 (8 pages), Mar 2014.
DOI: 10.14801/kiitr.2014.12.3.61
- [2] Tae-Woo Han, Yong-Ho Seo, "Emergencyn Situation Detection using Images from Surveillance Camera and Mobile Robot Tracking System", The Journal of the Institute of Internet, Broadcasting and Communication, VOL.9, No.5, Oct 10.
URL: <http://www.earticle.net/article.aspx?sn=113178>
- [3] Jong-Tae Kim, Jung-Shik Kong, Jin-Geol Kim, "Development of the Alarm System for Preventing of the Lonely Death of Elderly", j.inst.Korean.electr.electron.eng.Vol.20,No.1,115~118, Mar 2016.
DOI: <http://dx.doi.org/10.7471/ikeee.2016.20.1.115>
- [4] Kim Hyeong-Seok, Park Won-Woo, Lee Young-Bae, Seo Na-Hyun, Jeongchang Kim, "IoT-Based Location Tracking and Emergency Alert System for the Children and the Elderly", Korean Broadcasting Engineering Conference. Vol.2015 No.7, 2015.
URL: <http://www.dbpia.co.kr/Article/NOD>
- [5] Saac Lee, Jae-Soo Cho, "Tracking and Recognition of vehicle and pedestrian for intelligent multi-visual surveillance systems", Journal of the Korea Institute of Information and Communication Engineering. Vol. 19, No. 2 : 435~442, Feb. 2015.
DOI: 10.6109/jkiice.2015.19.2.435
- [6] Joseph Redmon, Santosh Divvala, Ross Girshick, Ali Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", IEEE Conference on Computer Vision and Pattern Recognition (CVPR) , 9 May 2016.
DOI: <https://doi.org/10.1109/cvpr.2016.91>
- [7] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, Alexander C. Berg, "SSD: Single Shot MultiBox Detector", Computer Science on pages 21 to 37, 2016
DOI: https://doi.org/10.1007/978-3-319-46448-0_2
- [8] Qiong WU, Sheng-bin LIAO, "Single Shot MultiBox Detector for Vehicles and Pedestrians Detection and Classification", DEStech Transactions on Economics, Business and Management issue apop, 16 Feb 2018
DOI: <https://doi.org/10.12783/dtem/apop2017/18546>
- [9] Rajashree Tripathy, R N Daschoudhury, "Real-time Face Detection and Tracking Using Haar Classifier on Soc", International Journal of Electronics and Computer Science Engineering, 2106
DOI: 10.1.1.447.1739
- [10] Vandana Singh, Dr.Vinod Shokeen, Bhupendra Singh, "FACE DETECTION BY HAAR CASCADE CLASSIFIER WITH SIMPLE AND COMPLEX BACKGROUNDS IMAGES USING OPENCV IMPLEMENTATION", International Journal of Advanced Technology in Engineering and Science Volume No.01, Issue No.12, December 2013.
URL: <https://www.semanticscholar.org/paper/Face-Det>

ction-by-Haar-Cascade-Classifer-with-and-Sin
gh-Shokeen/d2c6a82d317a07bcb642a074dfc3c2a2f1
daeb10

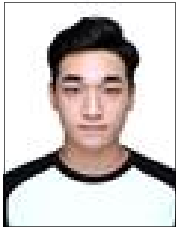
저자 소개

김혜정(준회원)



• 2015년 ~ 현재 : 한국산업기술대학교 컴퓨터공학부 학사과정

강민구(준회원)



• 2015년 ~ 현재 : 한국산업기술대학교 컴퓨터공학부 학사과정

이혜규(준회원)



• 2015년 ~ 현재 : 한국산업기술대학교 컴퓨터공학부 학사과정

고동범(준회원)



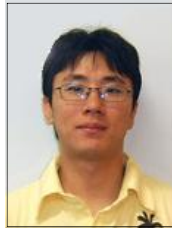
• 2016년 : 한국산업기술대학교 컴퓨터공학과 학사
• 2018년 : 한국산업기술대학교 컴퓨터공학과 석사
• 2018년 ~ 현재 : 한국산업기술대학교 스마트팩토리융합학과 박사과정
<주관심분야 : CPS, Autonomic Computing, Artificial Intelligence>

김정준(정회원)



• 2003년 : 건국대학교 컴퓨터공학과 학사
• 2005년 : 건국대학교 컴퓨터공학과 석사
• 2010년 : 건국대학교 컴퓨터공학과 박사
• 2016년 3월 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 조교수
<주관심분야 : Database Systems, Big Data, Semantic Web, Geographic Information Systems, Ubiquitous Sensor Network>

박정민(정회원)



• 2014년 ~ 현재 : 한국산업기술대학교 컴퓨터공학과 조교수
• 2012년 ~ 2014년 : 한국전자통신연구원 선임연구원
• 2011년 ~ 2012년 : 성균관대학교 연구교수
• 2008년 ~ 2011년 : 동양미래대학교 조교수
• 2009년 : 성균관대학교 컴퓨터공학과 박사
• 2005년 : 성균관대학교 컴퓨터공학과 석사
• 2003년 : 한국산업기술대학교 컴퓨터공학과 학사
<주관심분야 : CPS, Autonomic Computing, Software Engineering>

※ 이 성과는 2018년 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임 (No. 2017R1A2B4011243).