

실시간 객체 검출을 통한 스마트 쇼핑카트 시스템 설계 및 구현

김명진, 김민우, 노정훈*

충북대학교

2018035094@cbnu.ac.kr, alsdn3401@cbnu.ac.kr, jh.noh@cbnu.ac.kr

Design and Implementation of a Smart Shopping Cart System Using Real-Time Object Detection

Myung Jin Kim, Min Woo Kim, Jung Hoon Noh*

Chungbuk National University

요약

본 논문은 실시간 객체 탐지와 딥러닝 기술을 활용하여 스마트 쇼핑 카트 시스템을 개발하는 데 중점을 두고 있다. 이를 위해 YOLO와 라즈베리파이를 사용하여 소비자에게 추가적인 노동을 전가하지 않는 효율적인 쇼핑 경험을 제공하는 시스템을 설계하였다. 시스템은 쇼핑 카트에 담긴 물품을 자동으로 인식하고 결제하는 기능을 제공하며, 사용자 편의성과 경제적 효율성을 고려한 직관적인 인터페이스를 포함한다. 실시간 데이터 처리와 웹 기반 결과 표시를 통합하여 사용자 인터페이스와의 상호작용을 간소화하였고, 연구 결과는 기존 셀프 계산대 시스템의 문제점을 해결하며 모든 연령대의 사용자가 쉽게 접근할 수 있는 해결책을 제시한다.

I. 서론

최근 유통 산업은 기술적 혁신을 통해 지속적으로 변화하고 있으며, 자동화된 쇼핑 솔루션은 효율성을 높이고 인건비를 절감할 수 있는 잠재력을 보여주고 있다. 그러나 셀프 계산대 시스템은 소비자에게 추가적인 노동을 전가하고, 기기 사용에 익숙하지 않은 고령층에게 불편을 초래하는 문제가 있다. 이러한 문제로 인해 셀프 계산대 시스템은 거부감을 일으키며, 이는 브랜드 이미지에 부정적인 영향을 미치고 재방문 의도에 유의미한 영향을 미친다 [1][2].

이러한 문제를 해결하기 위해 본 논문은 실시간 객체 탐지 기술인 YOLO와 라즈베리파이, 그리고 휴대폰 어플리케이션을 활용하여 스마트 쇼핑 카트 시스템을 설계하고 구현하는 것을 목표로 한다. 이 시스템은 소비자의 쇼핑 카트에 담긴 물품을 자동으로 인식하고, 인식 결과를 휴대폰 화면에 표시하여 사용자 편의성을 증진시킨다.

II. 본론

2.1 시스템 모델

라즈베리파이(Raspberry Pi)는 저비용, 신용카드 크기의 소형 컴퓨터로, 다양한 입출력 포트를 통해 여러 장치와 연결할 수 있다. 크기가 작고 경량이어서 다양한 장소에 쉽게 설치할 수 있으며, 카메라, 키보드, 마우스, 디스플레이, 센서 등 여러 장치를 손쉽게 연결할 수 있다. 또한, 강력한 오픈 소스 커뮤니티의 지원을 받아 다양한 운영체제와 소프트웨어를 실행할 수 있다. 이번 설계에서는 리눅스 환경에서 파이썬을 사용하여 라즈베리파이 카메라를 연결하고 YOLOv5s를 활용하였다. 사용된 최신 모델인 라즈베리파이 5는 고성능 칩셋을 탑재하여 실시간 객체 인식과 같은 고도 연산을 필요로 하는 어플리케이션에도 적합하다.

YOLO(You Only Look Once)는 실시간 객체 탐지 알고리즘 중 하나로, 이미지나 비디오에서 객체를 빠르고 정확하게 인식할 수 있는 기술이다.

YOLOv5s는 YOLOv5에서 가장 작은 모델로, 속도와 정확도 사이의 균형을 잘 맞추고 있다. 경량화된 모델로 메모리 사용량이 적고 연산 속도가 빨라 라즈베리파이와 같은 소형 컴퓨터에서도 원활하게 동작할 수 있다. 이러한 경량화에도 불구하고 높은 정확도를 유지하며, 실시간 객체 인식에서 중요한 역할을 한다.

본 논문에서 구현한 객체 검출 방식은 학습된 데이터셋을 바탕으로 사용자가 어플리케이션 인터페이스를 통해 원하는 상황에서 객체 탐지를 수행할 수 있도록 설계되었다. 구체적으로, 사용자가 Flask 서버와 연결된 어플리케이션의 WebView 인터페이스에서 특정 버튼을 클릭하면, 라즈베리파이 5와 연결된 Flask 서버로부터 라즈베리파이에 실시간 이미지 촬영 및 YOLOv5s를 사용한 객체 탐지를 수행하라는 명령이 전송된다.

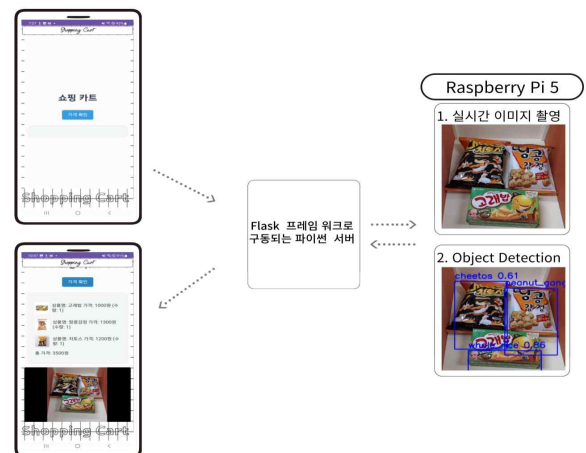


그림 1. 스마트 쇼핑카트 알고리즘 순서도

라즈베리파이 5는 Quad-Core Cortex-A76 CPU와 800MHz VideoCore VII GPU 칩셋을 탑재하고 있어, 직접 사진을 분석하고 YOLOv5s 모델을 사용하여 객체를 실시간으로 인식할 수 있다. 이는 별도의 데이터 서버로

사진을 전송하지 않고도 라즈베리파이 자체에서 YOLO 연산을 수행할 수 있게 하며, 클라우드 기반 처리를 필요로 하지 않아 실시간 데이터 처리를 용이하게 한다. 객체 탐지가 완료되면, 라즈베리파이는 탐지 결과를 그림3과 같이 Flask 서버에 즉시 전송한다. Flask 서버는 이를 받아 웹 페이지를 업데이트하며, 탐지된 객체의 항목, 개수, 가격 등을 표시한다. 또한, 실시간 이미지도 업로드되어 사용자가 어플리케이션에서 쉽게 확인할 수 있게 된다.

2.2 객체 검출 모델 훈련 및 실험 결과

데이터셋 생성은 roboflow[3]를 통해 진행하였다. 먼저 사진을 업로드한 후, 각 사진에 대해 주석 작업을 수행하여 객체를 인식하고 클래스에 맞게 분류하였다. 이 작업을 전체 이미지에 대해 수행한 후 데이터셋이 생성되었다. 다음으로 학습, 검증, 테스트 비율을 설정하였다. 총 51249장의 이미지 중 70%에 해당하는 35872장은 학습 세트, 20%에 해당하는 10252장은 검증 세트, 10%에 해당하는 5125장은 테스트 세트로 설정하였다.

비율 설정 후 전처리 과정을 거쳤다. 이미지 크기를 416×416으로 조정하고 Auto_Orient 및 Auto-Adjust Contrast 기능을 사용하여 훈련 시간을 줄이고 성능을 향상시켰다. 이후 데이터 증강 과정을 통해 좌우상하반전, 회전, 확대, 기울기, 색조 등의 변형을 적용하여 이미지 수를 증가시켰다. 본 연구에서는 최적의 epoch를 데이터 양과 overfitting을 고려하여 70으로 설정하여 YOLOv5s 모델을 학습하였다. epoch 수를 69로 설정한 결과, 모델의 성능이 가장 우수한 것으로 나타났다.

구체적으로, 그림 3과 같이 train.box_loss, train.obj_loss, train.cls_loss 값이 각각 0.017969, 0.000606, 0.000264로 매우 낮았으며, precision은 0.9918, recall은 0.9956, mAP@0.5는 0.9947로 높은 성능을 기록하였다. 비록 epoch 69에서의 성능이 가장 우수했지만, 다른 epoch과의 차이는 미미하였다. 이러한 미미한 차이는 모델이 안정적으로 학습되고 있음을 시사하며, 학습 과정이 잘 진행되고 있음을 보여준다.

그림 2의 그래프는 precision, recall, mAP@0.5의 변화를 나타내고 있다. 그래프에서 볼 수 있듯이, 학습이 진행됨에 따라 각 지표들이 점진적으로 향상되었으며, 최종적으로 안정화되는 모습을 확인할 수 있었다.

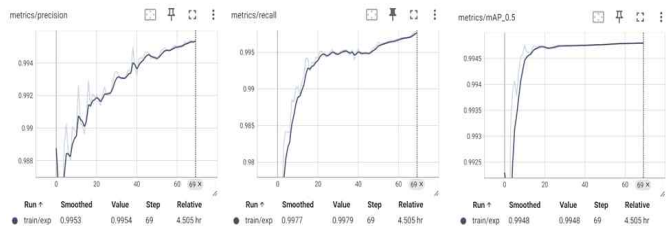


그림 2. Epoch 수치별 성능 비교 결과 그래프

epoch	train/box_loss	train/obj_loss	train/cls_loss	metrics/precision	metrics/recall	metrics/mAP_0.5
50	0.014885	0.0072417	0.00090021	0.99419	0.99643	0.99477
51	0.01503	0.007241	0.00099132	0.99448	0.9961	0.99477
52	0.01477	0.0071749	0.00084485	0.99475	0.99627	0.99478
53	0.014576	0.0071129	0.00089806	0.99486	0.99627	0.99478
54	0.01436	0.0070976	0.00075332	0.99492	0.99643	0.99478
55	0.014372	0.0070549	0.0007851	0.99481	0.9965	0.99478
56	0.01419	0.0069962	0.00068948	0.99473	0.99653	0.99479
57	0.01421	0.0070057	0.00073636	0.99487	0.99652	0.99479
58	0.013998	0.0069742	0.00077099	0.99498	0.99658	0.99479
59	0.013984	0.0069521	0.00065558	0.99507	0.99678	0.99479
60	0.013613	0.0068917	0.00060375	0.99498	0.99682	0.99479
61	0.013899	0.0068324	0.00060587	0.99503	0.99696	0.99479
62	0.013469	0.0067901	0.00063201	0.99506	0.99703	0.99479
63	0.01335	0.0067484	0.00062272	0.99515	0.99699	0.99479
64	0.013167	0.0067276	0.00055108	0.9953	0.99705	0.99479
65	0.013039	0.0067187	0.00049998	0.99521	0.99707	0.9948
66	0.012915	0.0066934	0.000477	0.99537	0.99722	0.9948
67	0.012757	0.0066177	0.0004704	0.99536	0.99755	0.9948
68	0.012573	0.0065825	0.00038777	0.99526	0.99778	0.9948
69	0.012495	0.0065626	0.00045644	0.99539	0.99793	0.9948

그림 3. Epoch 수치별 성능 비교 결과표

2.3 한계점 분석

본 연구에는 몇 가지 한계점도 존재한다. 본 연구에서 주요 문제는 라즈베리파이의 전력 소모 문제이다. 실시간 탐지 시스템의 지속적인 운영은 과도한 전력 소모를 요구한다. 쇼핑 카트에 부착된 카메라와 센서들이 지속적으로 활성화되면서 배터리의 빠른 방전을 초래하여 사용 지속 시간이 단축된다. 이는 이동형 장치로서의 실용성을 저하시킬 수 있다.

또 다른 한계점으로는 라즈베리파이의 연산 속도와 프레임 드롭 문제이다. 라즈베리파이5는 이전 모델인 라즈베리파이4에 비해 상당히 향상되었지만, 여전히 소형 컴퓨터라는 한계로 인해 실시간 이미지 처리 과정에서 프레임 드롭 현상이 발생할 수 있다. 이러한 현상은 중요한 데이터의 손실을 초래하며, 객체 인식의 정확도를 저하시킬 수 있다. 이러한 이유로 본 연구에서는 실시간 영상 처리보다는 실시간 이미지 처리를 채택하는 방향으로 진행되었다.

이러한 한계점에도 불구하고, 본 연구를 통해 제안된 스마트 쇼핑 카드 시스템은 실시간 객체 탐지와 딥러닝 기술을 활용하여 기존의 문제점을 효과적으로 해결할 수 있는 가능성을 보여주었다. 향후 더 다양한 환경과 상황에서의 테스트를 통해 시스템의 보완 및 개선을 진행하고, 전력 소모 문제를 해결하기 위한 에너지 효율적인 하드웨어 및 소프트웨어 최적화에 대한 연구가 필요하다.

III. 결론

본 논문에서는 실시간 객체 탐지 기술을 활용한 스마트 쇼핑 카드 시스템을 설계하고 구현하였다. 연구의 주요 성과와 결론은 다음과 같다

1. 효율적인 객체 탐지 및 인식: YOLOv5s 모델과 라즈베리파이 5를 결합하여 실시간으로 쇼핑 카트에 담긴 물품을 정확하고 빠르게 인식할 수 있었다. 이를 통해 별도의 데이터 서버나 클라우드 처리가 필요 없이 로컬에서 실시간으로 데이터를 처리할 수 있어 시스템의 효율성을 높였다.
2. 데이터 처리 및 증강: 데이터셋을 roboflow를 통해 생성하고, 다양한 전처리 및 증강 기법을 시도하여 연구 성과에 가장 적합한 수준의 효율적인 성능을 구현하였다. 이는 객체 탐지의 정확도와 신뢰성을 높이는 데 기여하였다.

ACKNOWLEDGMENT

이 논문은 2022년도 정부(방위사업청)의 재원으로 국방기술진흥연구소의 지원을 받아 수행된 연구임 (KRIT-CT-22-040, 이종 위성군 우주 감시정찰 기술 특화연구센터)

참고 문헌

- [1] 김효정 "대형마트 셀프 계산대에 대한 소비자의 이용 거부감에 관한 연구", 한국콘텐츠학회, pp. 340-352.
- [2] 나병진, 김준광. (2022). 셀프계산대형 무인매장에서 고객경험이 언택트 성향을 매개로 제방문의도에 미치는 영향. 대한경영학회지, 1, pp. 1-33, 2022.
- [3] Roboflow. "Roboflow Universe, ". Available: <https://universe.roboflow.com/>.