

<https://doi.org/10.7236/IIBC.2019.19.1.161>

IIBC 2019-1-22

중소기업 스마트공장 구축을 위한 OpenCV 기반 재고관리 시스템의 설계 및 구현

Design and Implementation of OpenCV-based Inventory Management System to build Small and Medium Enterprise Smart Factory

장수환*, 정종필**

Su-Hwan Jang*, Jopil Jeong**

요 약 다품종 대량 생산 중소기업 공장에서는 제품의 종류가 다양하고 그 수량이 많기 때문에 재고의 관리를 위한 인력과 경비가 낭비되고 있다. 또한 재고의 현황을 실시간으로 확인 할 방법이 마련 되지 않아서 재고의 과적재, 과부족 현상으로 인한 경제적 피해를 받고 있다. 실시간 데이터 수집 환경을 구축하기 위한 많은 방안이 있지만 대부분 구축비용과 시간이 중소·중견기업이 감당하기 어려운 수준이다. 그렇기 때문에 중소·중견기업의 스마트 공장은 구현되기 어려운 현실을 마주하고 있으며, 적절한 대책을 찾기 힘든 실정이다. 따라서 본 논문에서는 현재 생산품 관리 기술로 많이 채택되는 바코드, QR코드와 함께 라벨에 표기되어 있는 글자추출을 통해 기존 재고관리 방법의 확장에 대한 내용을 구현하고 그 효과를 평가하였다. 기술적으로는 컴퓨터 이미지 처리를 통해서 기존의 생산품의 입고고 관리를 위한 방법인 재고라벨 및 바코드에 대한 자동 인식 및 분류를 하기 위한 OpenCV를 이용한 전처리, 구글 비전 API의 OCR(Optical Character Recognition)기능을 통해서 글자를 추출하고, Zbar를 통해서 바코드를 인식할 수 있게 설계하였고, 값비싼 장비를 사용하지 않고 라즈베리파이를 통해 실시간 영상을 통한 인식으로 재고를 관리할 수 있는 방법을 제안한다.

Abstract Multi-product mass production small and medium enterprise factories have a wide variety of products and a large number of products, wasting manpower and expenses for inventory management. In addition, there is no way to check the status of inventory in real time, and it is suffering economic damage due to excess inventory and shortage of stock. There are many ways to build a real-time data collection environment, but most of them are difficult to afford for small and medium-sized companies. Therefore, smart factories of small and medium enterprises are faced with difficult reality and it is hard to find appropriate countermeasures. In this paper, we implemented the contents of extension of existing inventory management method through character extraction on label with barcode and QR code, which are widely adopted as current product management technology, and evaluated the effect. Technically, through preprocessing using OpenCV for automatic recognition and classification of stock labels and barcodes, which is a method for managing input and output of existing products through computer image processing, and OCR (Optical Character Recognition) function of Google vision API. And it is designed to recognize the barcode through Zbar. We propose a method to manage inventory by real-time image recognition through Raspberry Pi without using expensive equipment.

Key Words : OpenCV, OCR, Raspberry Pi, Barcode, QR code, Inventory label

*정희원, 성균관대학교 스마트팩토리융합학과 석사과정

**정희원, 성균관대학교 스마트팩토리융합학과 교수

접수일자 2018년 11월 17일, 수정완료 2019년 12월 27일

게재확정일자 2019년 2월 8일

Received: 17 November, 2018 / Revised: 27 December, 2018 /

Accepted: 8 February, 2019

**Corresponding Author: jpjeong@skku.edu

Department of Smart Factory Convergence, Sungkyunkwan University, Korea

I. 서 론

다품종 대량 생산 중소기업 공장에서는 제품의 종류가 다양하고 그 수량이 많기 때문에 재고의 관리를 위한 인력과 경비가 낭비되고 있는 실태이다. 현재 대다수의 중소기업형 생산공장은 입고와 관리와 재고의 현황을 실시간으로 확인 할 효율적인 방법이 마련되어 있지 않아서 재고의 과적재, 과부족 현상이 나타나고 있다.

중소 기업형 공장에서의 재고는 사람의 직접적인 개입으로 관리가 진행되고 있기 때문에 철저한 절차가 없다면 예기치 못한 상황이 발생했을 때 적절한 대처가 어려워질 가능성이 높아지게 된다. 또한, 재고의 종류와 수량이 많아질수록 재고정보에 대한 기록도 복잡해지고 실수가 많이 나올 확률이 높아진다. 그렇기 때문에 많은 공장에서 생산품의 입고와 및 재고의 관리를 위해서 바코드와 라벨을 통해서 프로세스를 구축하려 한다. 이러한 방법이 어느 정도 효율적으로 재고를 관리할 수 있게 보조를 해주는 역할을 하고 있지만, 여전히 인간의 역할에 대한 비중이 높은 편이라서 생산품의 숫자와 종류가 많아질수록 프로세스는 복잡해지고 효율적인 운영이 불가능한 상황이 발생하고 있다. 그렇기 때문에 본 논문에서는 이 방법을 보조하기 위해서 영상 인식을 통한 라벨과 바코드의 정보를 자동으로 추출하고 관리하는 방법에 대한 프로세스와 웹캠과 소형보드를 통해 직접 구현 해보고, 실제 환경에서도 적용가능한지 평가를 하였다.

논문의 구성은 2절의 관련연구에서 본 논문에서 제안하고자 하는 바코드 및 라벨인식에 대한 연구에 대해서 설명하고 기존에 사용하는 Tessaract-ocr 방법으로 진행되는 글자인식에 대해서 설명하고 나아가 OpenCV를 사용해서 영역을 추출하고 글자 인식처리에 대한 내용을 설명한다. 3절의 재고 라벨/바코드 인식 프로세스 연구 및 설계에서는 본 논문에서 제안하는 시스템 설계에 대한 내용을 설명한다. 시스템에 기본 절차로는 입력이미지/영상에 ROI (Region Of Interest)를 추출하고, 구글 클라우드 비전 API를 통해서 텍스트인식을 통해 생산품에 정보를 실시간으로 서버에 전송하고 DB에 저장한다. 4절에서는 구현 및 평가 결과를 제시하고, 5절에서 본 논문의 결론을 설명한다.

II. 관련연구

1. 바코드/QR코드 재고관리

바코드는 상품 라벨, 재고관리, 보관/검색 시스템 및 검사와 같이 일상생활에서 광범위하게 쓰이는 기술이다. 특히 재고는 수량 및 실시간 입출현황 데이터를 수집 및 분석하는 것이 필요한데, 데이터 분석 기법에 대한 방법으로는 경제적 모델을 통한 분석, 딥러닝 알고리즘을 통한 분석등 많은 방법이 존재하지만 입고와 및 재고 데이터 수집 기법으로는 아직까지 많은 연구가 필요하다. 그렇기 때문에 RFID 방식으로 재고의 입고출고를 관리하는 방법의 연구들이 진행되고 있다[1-3]. RFID의 경우, 원거리에서 주파수를 통해 한번에 여러 물건을 인식할 수 있는 기술이 적용되어[4], 생산공장의 입고출고 관리에 적용하기 가장 타당한 기술이지만, 바코드/QR코드에 비해 높은 고정 단가와 리더기 한 대로 모든 태그의 정보를 읽을 수 없는 단점으로 인해 중소기업 입장에서 채택되기 어려운 부분이 존재한다. 따라서 대부분 기업들이 바코드/QR코드 방식을 선호하게 된다. 특히 컴퓨터 비전 기반 바코드/QR코드 판독은 인식이 자동으로 진행되기 때문에 사용하기가 간편하고[5], 복수개의 바코드 또한 자동으로 인식하고 어떤 방향이든지 처리해줄 수 있기 때문에 효율적으로 생산품 관리를 할 수 있다[6]. 현재는 모바일 어플리케이션에도 적용하여 인식이 가능하게 되었다[7]. 하지만 비전을 통한 바코드 인식은 아직까지 해상도 또는 색반전등의 이미지 특성에 대한 영향을 많이 받고 코드가 훼손되면, 인식이 현저히 저하되기 때문에 관련 연구가 진행되고 있지만 아직 개선될 내용이 많다[8]. 그래서 좀 더 보완된 방식인 QR코드를 사용이 각광 받고 있다. QR코드는 기존 바코드에 비해 많은 양의 데이터를 담을 수 있으면서 작은 정사각형 모양으로 만들기 때문에 어디에나 배치하기 쉬운 외형의 특징을 가지고, 바코드에 비해서 더 많은 양의 데이터를 보존할 수 있다. 또한 레벨에 따라서 코드 손상률 7%,15%,20%,30%이하에 대해서 복원도 지원하기 때문에 높은 효율성을 보여준다[9].

2. 바코드/QR코드 재고관리

OpenCV는 BSD 라이선스 하에 배포되는 학술적 및 상업적으로 무료인 오픈소스 라이브러리이다. C++, 파이썬 및 자바 인터페이스가 있으며 더 많은 언어를 지원하기 위한 개발이 진행되어 있고, 윈도우, 리눅스, 맥 OS, iOS 및 Android등 현재 나와 있는 거의 모든 운영체제에

서 지원가능하고 최적화 된 C/C++로 작성되었고 멀티 코어 프로세싱을 이용할 수 있다. 또한 OpenCL을 사용하여 하드웨어 가속을 활용하여 더욱 원활한 이미지 처리를 할 수 있도록 지원하는 다기능 라이브러리이다[10].

OpenCV를 통해서 원본이미지 데이터에 대한 여러 가지 처리를 해줄 수 있다[11]. 본 논문에서 이 라이브러리를 채택한 이유는 무료 라이선스이면서 이미지/영상처리에 최적화 되어있어서 낮은 프로세싱 성능으로도 빠르고 효율적으로 데이터를 처리할 수 있도록 구성되어있고, 많은 분야에서 이 라이브러리를 통해 의미있는 결과를 도출해냈고, 또한 이미지처리 분야에서 인지도가 높은 편이기 때문에 관련 커뮤니티의 활성화도 잘 되어있어서 개발의 편의성에 대한 장점도 가지고 있기 때문이다. 또한 OpenCV를 이용하면 이미지에서 관심영역을 설계된 알고리즘을 통해 관심 영역을 추출하고 처리 할 수 있다. 이런 방식으로 전처리를 해주는 이유는 관심영역만 따로 추출할 경우 인식이나 기계학습을 진행할 때 이미지 전체에 대해 프로세스를 할 필요가 없고 이미 처리된 영역 안에서만 계산하기 때문에 매우 높은 효율성을 보여주기 때문이다.

글자인식을 위한 이미지처리 관련된 연구에서 이러한 방식으로 개발을 진행하며 결과적으로 효율성이 높은 방법이라는 것이 입증되었다[12]. 그렇기 때문에 이미지 전처리는 중앙서버의 부하를 효과적으로 줄여줘서 실시간 영상인식 시스템 구현을 가능하게 한다.

3. Optical Character Recognition(OCR)

광학문자인식(OCR)은 텍스트가 포함 된 이미지를 편집 가능한 텍스트 형식으로 변환하는 기술이다. OCR을 사용하면 스캔한 책, 스크린 샷 및 사진을 텍스트로 처리하고 TXT, DOC 또는 PDF 파일과 같은 편집 가능한 문서를 얻을 수 있다. 이 기술은 많은 분야에서 널리 사용되고 있는데, 가장 진보된 OCR 시스템은 거의 모든 유형의 이미지를 처리 할 수 있다. 현재 OCR 전문 프로그램 들로는 ABBYY사의 FineReader, Adobe의 Acrobat Pro/DC, IRIS의 Readiris등이 있는데 모두 유료이고, 무료로는

오픈소스인 Tesseract가 대표적이다. Tesseract는 원래 1985~1994년동안 휴렛팩커드 연구소의 독점 소프트웨어로 개발이 되었고, 1996년에 윈도우로 포팅되고, 1998년에 일부가 C에서 C++로 마이그레이션 되었다. 대부

분의 코드는 C로 작성되었고, 일부분은 C++로 되어있었는데, 이후에 모든 코드가 C++컴파일러로 컴파일 되도록 변환되었다. 그리고 2005년에 Hewlett Packard 와 the University of Nevada, Las Vegas (UNLV)에서 오픈소스로 공개되었고, 2017년 6월 1일 안정화 버전인 3.05.01과 딥러닝 모델인 LSTM 기반 4.0 버전이 깃허브 마스터 브랜치에 배포되었다. 현재는 구글 스폰서 하에 개발이 진행되고 있다[13]. 이처럼 활발하게 연구 및 개발이 진행되고 있고 기존 OCR에서 지원하기 어려웠던 복잡한 배경에서 텍스트를 추출하고 인식해내는 과정이 가능하게 되었다.

III. OpenCV 기반 재고관리 시스템 설계

1. 바코드/QR코드 재고관리

본 논문에서 제안하는 중소기업형 스마트공장 구축을 위한 OpenCV 기반 바코드와 라벨 인식을 위한 재고관리 시스템이다. 그림 1에서 제안하는 시스템의 전체적인 구조를 보여준다. 먼저 라즈베리파이에 연결된 카메라를 통해서 재고를 스트리머를 통해 실시간으로 이미지를 전송받는다. 라즈베리파이에는 기본적으로 이미지 처리를 위한 OpenCV와 바코드/QR코드 인식을 위한 Zbar가 빌드되어 있다.

재고 라벨 및 바코드의 영역을 인식하기 위해서는 OpenCV의 이미지 처리를 통해 원본이미지에서 원하는 영역을 추출해 낼 수 있다. 바코드의 경우 이미지 패턴이 단순하여 쉽게 알 수 있지만, 라벨의 경우는 그 형태가 다양하기 때문에 글자 특징을 중점으로 인식을 진행한다[14]. 물론 딥러닝 모델로도 이런 라벨 영역과 바코드 영역에 대한 인식이 가능하게 학습시킬 수 있다. 하지만 딥러닝 모델에서 더욱 많은 데이터를 학습해야하고 복잡성 커질 수 밖에 없어서 소모되는 컴퓨팅 자원이 높아지기 때문에, 본 논문에서는 이미지 전처리를 통해서 시스템을 좀 더 효율적으로 운영할 수 있도록 설계하였다.

라즈베리파이에서 추출 된 바코드/QR 이미지는 서버에서 인공지능 알고리즘을 통해 바코드에 내제되어 있는 정보를 DB에 저장시키고 모니터링 어플리케이션으로 데이터를 시각화한다. 그리고 추출된 라벨영역은 서버에서 미리 학습된 OCR을 API를 통해서 글자를 인식해내서

DB에 저장하고 라즈베리파이를 통해 그 정보를 시각화한다. 라즈베리파이를 사용하여 웹 모니터링 뿐만 아니라 무선랜을 통해서 모바일 디바이스에 그 기능을 확장시킬 수도 있다. 또한 사용자의 요구에 맞게 기능을 추가하기 쉽게 모듈화가 간편한 파이썬, 자바스크립트 언어를 이용하여 확장성과 범용성을 모두 지원하도록 구성하였다.

데이터의 저장은 로컬 서버 또는 클라우드를 통해 별도로 DB를 두어서 관리하도록 하여 보안과 안정성을 높이도록 구성하였다. 기능별로 정리하자면 카메라는 이미지 원본을 라즈베리파이로 전송하고, 라즈베리파이는 OpenCV를 통해 이미지에서 개발된 알고리즘으로 관심 영역을 검출 및 추출하여 모니터링 할 수 있게 어플리케이션을 배포하고, 그 기능을 연결된 모바일 디바이스에 확장시킬 수 있고, 비전 API를 통해 추출된 영역에서 글자인식, Zbar를 통해 바코드를 인식할 수 있다. 그리고 서버는 이러한 데이터들을 정해진 플로우에 따라서 DB 영역에 저장하고, 마지막으로 저장된 데이터들을 웹 어플리케이션을 통해 사용자가 실시간 정보와 데이터의 누적 및 기본적인 통계자료를 확인할 수 있다.

그림 2는 카메라 이미지상의 바코드와 QR코드의 인식 및 결과를 처리하는 과정이다.

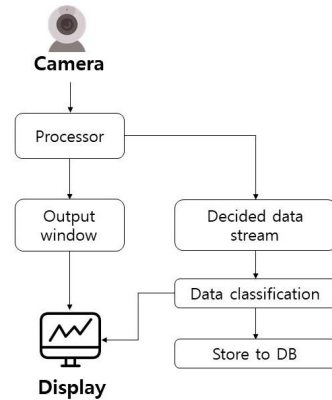


그림 2. 바코드 및 QR 인식 프로세스
Fig. 2. Barcode and QR recognition process

Zbar의 프로세서를 통해서 이미지 내의 바코드 및 QR 코드를 검출할 수 있으며 자동으로 분류한다. 그리고 인식결과를 디스플레이하고 동시에 그 데이터 스트림을 분류하여 DB에 저장한다. 본 논문에서는 제품의 데이터가 인식되며 제품명 기준으로 수량을 취합하였다.

완전히 독립적인 모듈식 접근법의 잠재적 단점 중 하나가 모든 모듈을 하나로 묶는 코딩이 필요하므로 간단한 응용 프로그램 조차 만들기 복잡해지기 때문에, Zbar

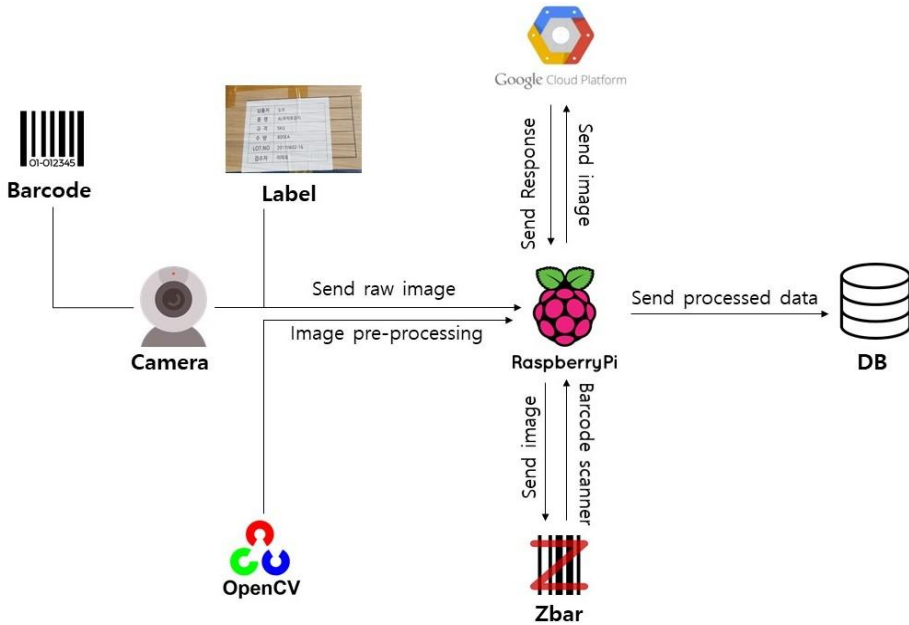


그림 1. 라벨 / 바코드 이미지 인식 과정
Fig. 1. Label / barcoard image recognition process

의 고수준 프로세스 모듈은 다른 모듈들을 연결하여 일반적인 용도를 유연하게 지원할 수 있게 한다.

인식된 영역을 시각화하기 위해서 OpenCV의 convexHull을 통해 바코드/QR영역의 최외각이 급격히 변하는 포인트를 잡아서 박스를 그릴 수 있다. 또한 한 이미지 프레임 안에서 여러개의 바코드 및 QR코드를 인식하여 표시하는 것도 가능하다.

2. OpenCV를 통한 라벨 영역 인식

이 절에서는 본 논문에서 이미지 전처리에 쓰일 핵심적인 OpenCV의 함수들의 이론을 설명하도록 하겠다. 그림 3은 라벨영역 인식 절차이다[11]

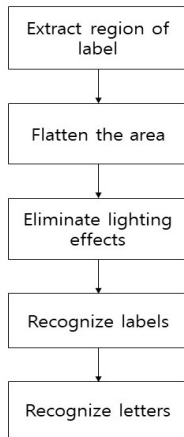


그림 3. 라벨 영역 인식 프로세스
 Fig. 3. Label area recognition process

a) 가우시안 필터를 통한 노이즈 제거

이미지의 불필요한 노이즈로 인한 잘못된 엣지 검출 결과를 사전에 제거하기 위해서 가우시안 필터를 통해 blurring을 해주는 과정이다. 가우시안 필터의 커널은 $(2k+1) \times (2k+1)$ 로 설정되어 있다. (k는 자연수)

b) 이미지의 intensity gradient 찾기

blurring된 이미지의 그라디언트는 다양한 방향이 있을 수 있기 때문에 수평방향 (G_x)과 수직방향의 (G_y)으로 나누어서 (1)식과 같이 정의하도록 한다.

$$G = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$\Theta = \tan^{-1}(G_y / G_x) \quad (2)$$

(2)식은 그라디언트 방향을 나타내고 항상 가장자리에 수직이다. 또한 수직, 수평 대각선 방향을 나타내는 4개의 각도 중 하나로 반올림된다.

c) non-maximum suppression 적용

그라디언트의 크기와 방향을 얻은 후 원치 않는 픽셀을 제거하기 위해서 이미지의 전체 스캔을 수행한다. 이를 위해, 모든 픽셀에서 그라디언트 방향에서 인접 픽셀의 로컬 최대값인지 확인한다. 결과로 얇은 바이너리 이미지 엣지를 얻는 과정이다.

d) Hysteresis Thresholding 적용

이 단계에서는 모든 edge가 실제 edge인지 아닌지를 결정하게 된다. 이를 위해서는 두 개의 임계값인 minVal과 maxVal이 필요하다. minVal보다 작은 부분은 버려지게 되고, maxVal보다 큰 부분은 edge로 정해지고, 만약에 판단할 값이 minVal과 maxVal의 사이값이 된다면 연결된 부위를 통해서 판단하게 되며, 연결된 부위중 maxVal보다 큰 값이 있다면 판단할 값 역시 edge의 일부로 간주된다. 그 반대면 삭제한다. 마지막으로 edge 검출을 통해 외곽선을 찾아 그려준다.

1) 영역 반듯이 전환

위의 방법으로 검출한 영역은 다양한 각도로 보여질 수 있기 때문에 정면이미지로 전환하는 것이 필요하다. 전체 이미지의 가장 왼쪽 위의 좌표가 (0, 0) 이라고 할 때 각 각 검출된 4개의 edge의 좌표를 구해서 cv2.getPerspectiveTransform() 함수에 보내고 cv2.warpPerspective()에 넣음으로써 최종적으로 반듯한 사각형으로 변환된 이미지를 받게 된다.

2) 조명 명암 제거

먼저 cv2.cvtColor()함수를 사용하여 이미지를 gray이미지로 바꿔준다. 이렇게 gray로 바꿔준 이미지를 threshold를 통해 처리 해주는데 global threshold로 처리할 경우 이미지의 명암 대조가 너무 크게 되어서 많은 부분이 손실된다. 그렇기 때문에 구역을 잘게 나누어 각각의 영역에 맞는 threshold값을 정해주는 adaptive threshold를 적용할 경우 좋은 결과를 얻을 수 있다. 함수는 cv2.adaptiveThreshold()를 쓰고 적용될 경우 grayscale이미지가 모두 흰색과 검은색으로 변환된다.

위와 같은 과정으로 글자 영역 추출에 대한 이미지 전처리가 완료되었고 다음 과정으로 그 영역에 텍스트 인식을 위한 처리를 진행해야 한다.

3) 라벨 외곽 추출

라벨 영역을 인식하기 위해서 먼저 원본 이미지로부터 모든 Edge를 검출하고 Edge중에서 라벨로 인식할 부분만을 검출하게 해야한다. Canny Edge Detection 알고리즘으로 이 부분을 진행 할 것이다. cv2.Canny()는 Canny Edge Detection 알고리즘의 함수이고 John F.Canny 가 개발하였다. 이 함수를 통해서 재고의 라벨에 해당하는 부분의 외곽 검출할 수 있다. 라벨의 인식 절차는 아래와 같다.

3. Vision API를 통한 라벨 텍스트 인식

본 논문에서는 클라우드 비전 API를 통해서 스틸 내의 텍스트를 인식하고 추출하였는데, OCR기능은 TEXT_DETECTION이라고 정의되어 있다[16]. 기본적으로 라즈베리파이 카메라 모듈을 통해 단일 스틸을 캡처한 후 텍스트인식을 진행한다. 그림에서 이미지처리와 영역추출은 파이썬 코드로 작성되었고, 비전 API와의 연결은 node.js 기반으로 작성하였다. 그림 4는 실시간으로 이미지 프레임을 비전 API를 통해서 글자를 인식하는 과정이다.

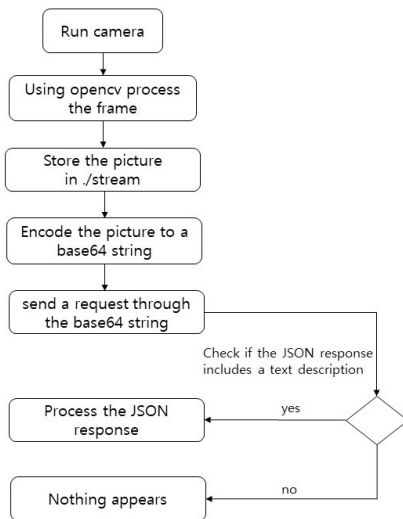


그림 4. 라벨 영역 인식 결과
Fig. 4. Label area recognition result

단일 스틸 사진들은 경로에 stream이라는 폴더에 cam.jpg형태로 저장하였다. 그리고 저장된 사진은 base64포맷으로 인코딩 해준 후 변환된 스트링 값을 통해서 구글 비전 API에 인식 요청을 의뢰할 수 있다. 요청 후 이미지 상에 글자가 발견되면 JSON 형태로 그 인식 결과를 반환해주고, 그렇지 않다면 아무것도 반환하지 않는다.

IV. 구현 및 평가

1. 구현환경 및 절차

본 논문의 한가지 목표인 저렴한 시스템 구성을 위해서 디바이스는 라즈베리파이 3B모델을 사용하였고, 사양은 CPU 1.2GHz ARM Cortex-A53 MP4, GPU Broadcom VideoCore IV MP2 400 MHz, RAM 1 GB LPDDR2이다. 이미지 촬영은 RPi Camera V2으로 3280 x 2464이미지 촬영이 가능하며 1080p30, 720p60, 640x480p90 비디오 영상 촬영이 가능하다. 그리고 라즈베리파이에 소프트웨어로는 python 2.7기반 OpenCV를 설치하여 카메라 프레임 단위로 이미지처리 작업하고, Nodejs를 통해서 실시간으로 OCR 결과를 추출하고 시스템에 맞게 가공하였다. 그림 5는 QR/바코드 및 라벨이

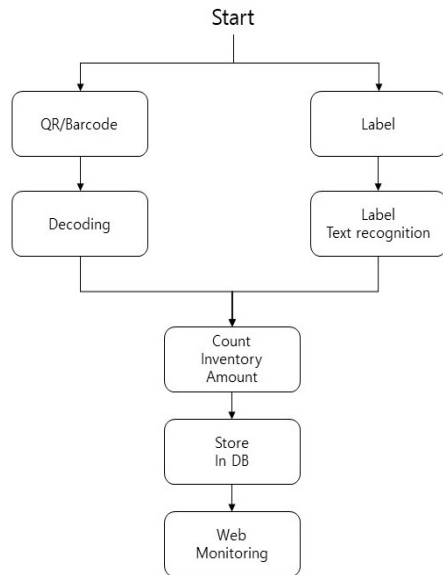


그림 5. 재고 관리 시스템 흐름도
Fig. 5. Inventory management system flow chart

인식되는 과정을 말한다. 만약 이미지 상에 QR/바코드가 있으면 디코딩을 하여 정보를 추출하고 만약 텍스트 정보만 있는 라벨이라면 텍스트를 인식하고 추출한다. 그리고 같은 품명이 있으면 카운트 해주고 DB에 저장된다. 또한 실시간으로 진행되기 때문에 DB에도 실시간 정보가 입력된다. 최종적으로 DB에 저장된 정보를 통해서 재고를 실시간으로 웹을 통해서 모니터링 할 수 있다.

2. 바코드 및 라벨 인식 결과

그림 6은 라벨의 텍스트 영역이 인식된 결과이다. 최종 구현에서는 따로 인식영역을 표시해주지 않고, 텍스트 추출 후 값을 모니터링 화면에 띄워주는 방식으로 진행하였다.

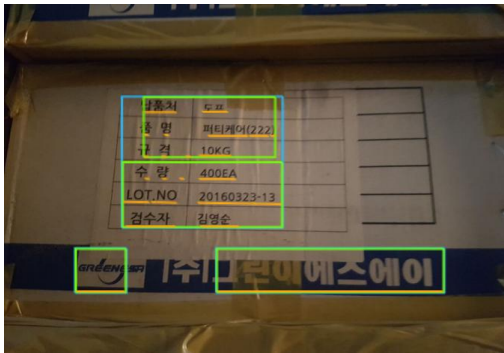


그림 6. 재고 관리 시스템 흐름도
 Fig. 6. Label area recognition result

영역에서 글자 인식을 진행하기 위해서 위에 설명과 같이 node.js를 통해서 base64 string 값을 기반으로 그림 7과 같이 구글 클라우드에 인식을 요청하였다. 여기서 letter.jpg가 인식을 요청할 이미지 파일이다. 그리고 요청이 처리되면 Json 형태의 스트링 값으로 반환 해준다.

```
var d = requitil.createRequests().addRequest(
    requitil.createRequest('./letter.jpg')
    .withFeature('TEXT_DETECTION', 3)
    .build());
```

그림 7. 텍스트 인식 요청
 Fig. 7. Text recognition request

3. 바코드 및 라벨 인식 결과

그림 8는 라즈베리파이 카메라를 통한 복수개의 바코드인식을 진행 하였을때의 결과이다. 실험을 통해서 실

제로 RPi Camera V2로 인식을 진행하였을 때 최대 2개의 바코드를 인식할 수 있었다. 포맷은 EAN 형식밖에 인식이 안되었으며, 숫자만 담을 수 있으며 포함시킬 수 있는 정보량도 낮은편이다.



그림 8. 바코드 다중 인식 결과
 Fig. 8. Barcode multiple recognition result

그림 9에서 복수개의 QR코드인식 결과이다. 라즈베리파이 카메라를 통해서 한 화면에 4개정도가 성공적으로 인식되었다. 바코드와 비교해봤을 때 담을 수 있는 정보량도 훨씬 많기 때문에 실제로 다른 환경에 응용하기에 QR코드의 활용도가 높아보인다.



그림 9. QRcode 다중 인식 결과
 Fig. 9. QRcode multiple recognition result

그림 10은 복수개의 라벨을 인식한 결과이다. 위에 바코드/QR과는 다르게 형식에 있어서 자유로운 편이고 여러 언어를 지원하기 때문에 범용성 또한 높다. 본 논문의 구현에서는 인식주기를 10초로 설정하여 콘솔 창에 인식된 재고의 이름과 그 수량을 나타내도록 하였다.

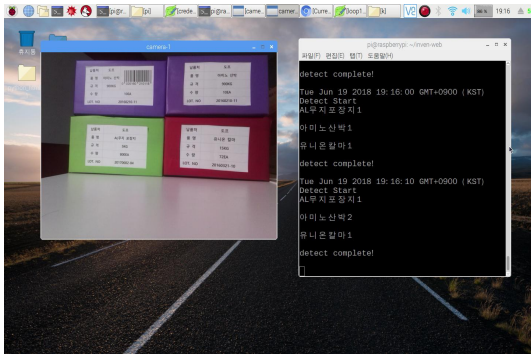


그림 10. QRcode 다중 인식 결과
Fig. 10. QRcode multiple recognition result

4. 시스템 평가

마지막으로 기존 재고관리 프로세스와 본 논문에서 제안하는 프로세스와 비교를 해왔다. 기존 프로세스의 재고관리는 재고의 종류와 수량이 많은 공장에서는 하루에 한번씩 진행 할 수 없고, 일정 주기를 가지고 관리한다. 그러기 때문에 실시간으로 수량을 파악할 수 없고 정확한 정보인지 알 수 있는 적절한 방법 또한 없다.

관리적인 측면에서 여러 방법이 거론되고 있지만, 중소기업형 공장 같이 인력의 변동이 많은 환경에서 적용하기에 다소 무리가 있을 수 밖에 없다. 이런 이유로 재고관리에 대한 전산화 환경이 꼭 필요하기 때문에 RFID, 바코드, QR코드 등의 방법을 도입하고 있지만 이 또한 거래기업간의 협약이 이루어지지 않는다면 구현하기 힘든 방법이다. 하지만 논문에서 제안하는 시스템은 카메라와 이미지를 처리할 수 있는 컴퓨터만 있으면 구축할 수 있고, 작업의 요구사항이 높지 않기 때문에 업무적인 효율이 높고, 다른 IoT 디바이스와의 연계로 확장성 또한 기대할 수 있다.

표 1. 기존 및 제안 된 프로세스의 비교
Table 1. Comparison of existing and proposed processes

	Existing process	Proposed process
Characteristic	Static	Dynamic
Method of storing data	Passive	Automatic
Method of inventory managemnet	Handwriting	Computer Vision
Business efficiency	Low	High

표 1을 통해서 기존 프로세스와 제안 프로세스의 특징을 비교하였다. 주요 특징점으로 재고 데이터 수집의 자동화, 이종 시스템과 연결의 위한 확장성, 원격 스케줄링 제어 및 모니터링 등이 있다. 제한점으로는 아직 한 카메라에서 많은 수의 재고를 인식하지 못하고, 라즈베리파이로 더 높은 요구의 이미지처리를 하기에는 성능적인 제약이 있다는 것으로 앞으로 보완해야 할 문제점이다.

표 2. 각 인식 유형 결과의 비교
Table 2. Comparison of existing and proposed processes

	Barcode	QRcode	Text
Data volume	Low	High	Flexible
Data format	Number	Text	Text
Recognition rate	Bad	Good	Good
Recognition Speed	High	High	Normal (> 1s)

표 2에서는 논문에서 사용한 웹캠을 통한 QR코드, 바코드 및 텍스트 인식결과를 정리하여 비교하였다.

실험은 같은 해상도에서 같은 라벨크기를 기준으로 결과를 비교할 수 있게 진행되었다. QR코드에 콘텐츠가 많아지면 QR이미지의 복잡성이 높아져서 라즈베리파이 카메라가 인식을 하기 어려워지기 때문에, 제품명과 용량 등의 제한적인 정보만 입력하였다. 그 결과 한 프레임에 바코드는 2개 인식, QR코드는 4개 인식, 텍스트 라벨도 4개까지 인식되었다.

제품 정보의 종류수가 많아지고 인식속도가 빠르지 않아도 되는 정적인 환경이라면 텍스트 라벨 인식의 관리가 적절하고, 동적인 환경에서의 관리를 원한다면 QR코드와 바코드 인식을 통한 관리방법이 알맞은 것으로 분석된다.

만약 통신 속도 개선과 고수준의 프로세싱을 할 수 있는 컴퓨터가 라즈베리를 대체한다면 텍스트 라벨이 바코드와 QR코드의 인식 방법을 모두 대체할 수 있을 것이다.

V. 결론

본 논문에서 제안하는 방법으로 바코드와 라벨에 대해 준수한 수준의 인식률을 보여주었고, 복수의 바코드

와 라벨에 대해서 인식이 가능하여, 실제로 다품종 대량 생산 공장에 환경에서도 적용 할 수 있을 것을 기대하고 있다. 또한 사람의 육안으로 검출해내는 것이 아니라 소프트웨어가 내장된 기계에서 정확한 검출 알고리즘을 통해 판별해 내는 것이기 때문에 실시간성, 정확성, 효율성 등의 장점을 가진다.

제안된 방법은 기존에 사람이 작업하는 부분을 대신 처리해주고, 작업자가 고차원적이고 생산적인 일을 할 수 있게 환경을 조성하는 스마트 공장의 핵심 취지와도 맞는 역할을 하게 된다.

기존에 재고 관리 프로세스는 재고관리를 하기 위한 작업자가 따로 필요하고, 만약 새로운 인력이 투입되면 다시 관리 프로세스에 적응하기 위한 시간이 필요하기 때문에 인력교체가 활발한 중소기업 공장 같은 경우 제대로 된 재고관리 프로세스를 갖추기가 어렵다. 그렇기 때문에 재고관리를 축소시켜서 운영하게 되는데, 그에 따른 경제적 손실이 생각보다 크다. 따라서 본 논문에서 제안한 프로세스와 같이 재고를 데이터화 시켜서 실시간 및 원격으로 모니터링 하여 위와 같은 문제점들을 해소할 수 있다. 또한 제안한 시스템을 구현해 봄으로써, 저 사양 소형보드로도 공장 환경에서 활용 가능한 기능을 구현할 수 있고, 낮은 단가로도 실시간 처리 및 모니터링을 충분히 할 수 있다는 것을 알 수 있었다.

후속 연구에서는 본 논문에서 제안한 시스템에서 실시간 영상처리를 더욱 빠르고 작업 부하를 줄이고 좀 더 정확한 인식을 위한 기술들[17-19]을 활용하는 방안에 대해서 연구할 것이다.

마지막으로 라즈베리파이 같은 저렴한 보드들은 점점 고성능으로 진화해가고 있기 때문에 이러한 개발보드들을 통한 중소기업 대상 스마트공장 구축 솔루션 개발이 활발해 질 것이라고 기대한다.

References

- [1] B. Hameed, I. Khan, F. Dürr, and K. Rothermel, "An RFID based consistency management framework for production monitoring in a smart real-time factory," *Internet of Things (IOT)*, pp.1-8, 2010.
DOI: <https://doi.org/10.1109/iot.2010.5678436>
- [2] Shouqin Zhou, Weiqing Ling, Zhongxiao Peng, "An RFID-based remote monitoring system for enterprise internal production management," *International Journal of Advanced Manufacturing Technology*, Vol.33, Issue 7/8, pp.837-844, 2007.
DOI: <https://doi.org/10.1007/s00170-006-0506-6>
- [3] K. Ding and P. Jiang, "RFID-based production data analysis in an IoT-enabled smart job-shop," in *IEEE/CAA Journal of Automatica Sinica*, Vol.5, No.1, pp.128-138, Jan 2018.
DOI: <https://doi.org/10.1109/jas.2017.7510418>
- [4] Y. Sato, J. Mitsugi, O. Nakamura and J. Murai, "Theory and Performance Evaluation of Group Coding of RFID Tags," in *IEEE Transactions on Automation Science and Engineering*, Vol.9, No.3, pp.458-466, July 2012.
DOI: <https://doi.org/10.1109/tase.2012.2193125>
- [5] Youssef, Sherin M., Salem, Rana M., "Automated barcode recognition for smart identification and inspection automation," In *Expert Systems With Applications*, Vol.33, No.4, pp.968-977, 2007.
DOI: <https://doi.org/10.1016/j.eswa.2006.07.013>
- [6] Lin, DT., Lin, MC. and Huang, KY., "Real-time automatic recognition of omnidirectional multiple barcodes and DSP implementation," *Machine Vision and Applications*, Vol.22
DOI: <https://doi.org/10.1007/s00138-010-0299-3>
- [7] Subpratatsavee, P.,Janthong, N.,Kuha, P.,Chintho, C., "HC2D barcode reader using embedded camera in Android phone," 2014 11th International Joint Conference on Computer Science and Software Engineering (JCSSE), pp.254-257, May 2014.
DOI: <https://doi.org/10.1109/jcsse.2014.6841876>
- [8] Zhang, C., Wang, J., Han, S., Yi, M., Zhang, Z., "Automatic realtime barcode localization in complex scenes," 2006 International Conference on Image Processing Image Processing, pp.497-500, October 2006.
DOI: <https://doi.org/10.1109/icip.2006.312435>
- [9] AKTAŞ, C., "The Evolution and Emergence of QR Codes. Newcastle upon Tyne," Cambridge

- Scholars Publishing, 2017.
- [10] OpenCV [Internet] <https://opencv.org/about.html>, 2018.
- [11] Robert Laganiere, “OpenCV 3 Computer Vision Application Programming Cookbook,” Packt Publishin-Third Edition, 2017.
- [12] Priyanka Deelip Wagh, D.R. Patil., “Text detection and removal from image using inpainting with smoothing,” 2015 International Conference on Pervasive Computing, pp.1-4.
DOI: <https://doi.org/10.1109/pervasive.2015.7087154>
- [13] Tesseract [Internet] <https://github.com/tesseract-ocr/tesseract>.
- [14] Narasimha Reddy Soora and Parag S. Deshpande., “Review of Feature Extraction Techniques for Character Recognition,” IETE Journal of Research, Vol.64, No.2, pp.280-295, 2018.
DOI: <https://doi.org/10.1080/03772063.2017.1351323>
- [15] Z bar code reader [Internet] <http://zbar.sourceforge.net/about.html>.
- [16] Google cloud vision API [Internert] <https://cloud.google.com/vision/docs/ocr>.
- [17] Jin-Kyoung Heo, “Distributed image preprocessing using object activation,” The Journal of The Institute of Internet, Broadcasting and Communication, Vol.11, No.1, pp. 87-92, 2011.
DOI: <https://doi.org/10.7236/IJWIT.2011.11.1.087>
- [18] Tae-Woo Kim, Hyeon-Joong Yoo, “Color Code Detection and Recognition Using Image Segmentation Based on k-Means Clustering Algorithm,” Journal of the Korea Academia-Industrial cooperation Society, Vol.7, No.6, pp.1100-1105, 2006.
- [19] Yeon-Hee Kim, Jae-Ho Kim, Yoon-Ho Kim, “Shadow Removal based on Chromaticity and Brightness Distortion for Effective Moving Object

Tracking,” The Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol.8, No.4, pp.249-256, 2015.
DOI: <https://doi.org/10.17661/jkiict.2015.8.4.249>

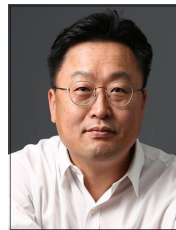
저자 소개

장 수 환(정회원)



- 2017년 9월 ~ : 성균관대학교 스마트팩토리융합학과 석사과정 재학중
- 주관심분야: 블록체인, 분산컴퓨팅 아키텍처, 스마트팩토리

정 종 필(정회원)



- 2008년 ~ 2009년 : 성균관대학교 컨버전스연구소 연구교수
- 2010년 ~ 현재 : 성균관대학교 정보통신대학 겸 산학협력단 교수
- 2015년 ~ 현재 : 전자부품연구원 IoT 융합연구센터 전문연구위원
- 2016년 ~ 현재 : 성균관대학교 스마트팩토리융합학과 사업총괄책임자
- 주관심분야 : 스마트팩토리, 모바일융합컴퓨팅, 센서 네트워크, 차량 모바일 네트워크, 네트워크 보안, IT융합, 인터렉션사이언스, 스마트 헬스케어, IoT/M2M, 웨어러블 컴퓨팅 등

※ 이 성과는 2018년도 경기도의 재원으로 경기도테크노파크의 지원을 받아 수행된 연구임(No.Y181802)
This work has supported by the Gyeonggi Techno Park grant funded by the Gyeonggi-D government (No.Y181802).