

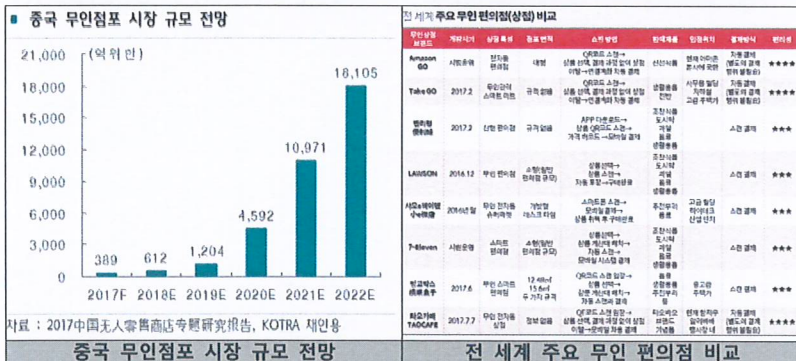
DU-도전학기 결과보고서

성명	[국문] [영문]	학번	
단과대학	정보통신대학	학과(전공)	전자제어공학과 (복수전공 시응용전공으로 신청)
도전학기 과제명	[국문] 라즈베리파이를 이용한 재고관리 시스템 [영문] Inventory management system using Raspberry Pi		
지도교수 의견	본인이 설정한 목표를 달성하지 못하여 아쉽겠지만, 이 실패를 통해 앞으로 정진하는 기회가 될 것 같습니다. [소속] 전자제어공학과 [서명] 날인 [성명] [서명] 날인		

1. 도전 과제의 목표



매장을 가득 채운 재고



무인점포 브랜드	제조사	상업 목표	중요 전략	서비스 범위	판매 채널	입점 지역	운영 방식	평가
Amazon GO	미국	점포 운영	내선 상품	인공지능 기반 재고 관리 및 고객 서비스	선진 시장	미국, 영국, 일본	주요 고객: 편의점, 대형 마트	★★★★
Take GO	2017.2	미국	규격화	인공지능 기반 재고 관리 및 고객 서비스	미국	미국	주요 고객: 편의점, 대형 마트	★★★★
편의점 운영	2017.2	대한민국	규격화	인공지능 기반 재고 관리 및 고객 서비스	미국	미국	주요 고객: 편의점, 대형 마트	★★★
LAUNCH	2016.12	미국	규격화	인공지능 기반 재고 관리 및 고객 서비스	미국	미국	주요 고객: 편의점, 대형 마트	★★★
시리아이점	2016.12	미국	규격화	인공지능 기반 재고 관리 및 고객 서비스	미국	미국	주요 고객: 편의점, 대형 마트	★★★
7-Eleven	미국	점포 운영	내선 상품	인공지능 기반 재고 관리 및 고객 서비스	미국	미국	주요 고객: 편의점, 대형 마트	★★★
편의점 운영	2017.6	미국	규격화	인공지능 기반 재고 관리 및 고객 서비스	미국	미국	주요 고객: 편의점, 대형 마트	★★★★
편의점 운영	2017.7	미국	규격화	인공지능 기반 재고 관리 및 고객 서비스	미국	미국	주요 고객: 편의점, 대형 마트	★★★★

4차 산업혁명이 진행됨에 있어 전 세계적으로 무인 매장 및 재고 관리 시스템이 관심을 보이고 있습니다. 우리의 일상 속에서 가장 가까운 매장에 관련된 IoT를 공부하고 어떠한 IoT가 실생활에 얼마나 어떻게 영향을 주는지 배우고 다양한 기술들을 접목시켜 프로젝트 설계를 진행해 나아가는 것이 목표입니다. 그 중에서 MySQL을 통해 DB를 관리하여 재고의 정보들을 체계적으로 정리하는 것이 목표입니다. 재고를 인식하는 것도 중요하지만 재고의 정보관리를 중점으로 잡아서 과제를 진행을 하도록 하겠습니다. 프로젝트를 완성하여 결과물이 나오면 주제와 맞는 교내외 공모전 및 행사에 참가하여 입상 및 경험을 쌓도록 할 것입니다.

2. 도전 과제 내용

2020-2학기에 진행할 프로젝트는 재고를 인식하여 재고 관리 체계 Web Server를 만들어 재고관리를 체계적으로 도와주는 프로젝트를 진행할 것입니다. Flask를 통해 Web Server를 제작할 예정이며, 재고관리를 위한 DB는 MySQL로 진행할 것입니다. Web Server와 DB가 중점으로 진행할 것이기 때문에 재고 인식은 원형 물품만을 기준으로 잡고 OpenCV를 사용하여 영상처리를 진행할 것입니다. Web Server로 제작하여 언제 어디서든 인터넷만 가능한 곳이라면 어떠한 전자기기로도 재고의 정보를 볼 수 있을 것입니다. 진행이 잘된다면 재고가 특정범위가 되면 서버에 보여주는 것이 아니라 휴대폰에 직접 알림이 가게끔 해보고 싶습니다. 이러한 방식으로 라즈베리파이를 사용하여 IoT 프로젝트를 진행할 것이며 진행 완료 후, 교내외 여러 공모전 및 행사에 참가하여 입상하도록 노력할 것입니다.

3. 도전 과제의 성과

현재 라즈베리파이와 라즈베리파이 카메라를 통해 Web Server를 열었고, 이를 통해 Hough circle을 이용한 원검출을 하여 재고의 개수를 갱신합니다. Web page에서 스트리밍까지 도전하려 했으나, 원검출과 스트리밍이 동시에 실행될시 카메라 충돌로 인해 오류가 발생하여 스트리밍은 도중에 제외하기로 했습니다. 라즈베리파이 1개로 모든 것을 구현을 하니 과부하가 와서 검출은 원검출 하나만 사용하기로 했습니다. 프로젝트를 완성시키고 나서 목표에 도달하지 못하여 본인에 자책이 심하지만, 이 프로젝트를 하면서 부족한 점이 많고 자신에 대해서 돌아보는 계기가 되었습니다.

4. 자기평가

현재 한국 대다수의 잡화점은 수량체크를 노동력을 통해 진행하고 있습니다. 이 부분에서 많은 기술들이 출시되고 있지만, 독점할만한 기술이 아직 등장하지 않았습니다. 그래서 저는 알바를 했던 경험을 통해 이러한 부분의 기술이 있었으면 좋겠다. 생각하여서 아이디어를 내고 아이디어를 실현하기 위해 자신 스스로 도전했다는 것에 뿌듯하지만, 결과적으로는 프로젝트를 계획한 주요점들은 실행이 되었지만, 세밀한 점들이 많이 부족하여 본인 스스로 만족하지는 못했습니다. 그리고 이 프로젝트를 통해 각종 대회 및 공모전에 참가하려고 했지만, 너무 늦게 완성하는 바람에 주제에 맞는 대회나 공모전이 없었고, 겨울방학에 기회가 된다면 참가할 것입니다.

5. 최종 결과물

5-1 요약


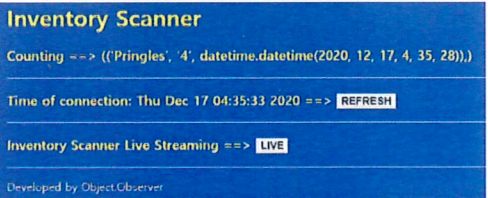

○ 작품 설명

- 카메라를 통해 재고를 인식하여 Web page를 통해 재고의 정보 제공
- 언제 어디서든 재고 정보 제공

- 실시간 재고 정보를 수집
- 작품이 작동하면 재고 정리 시 재고 추가만 요구

○ 핵심 기능

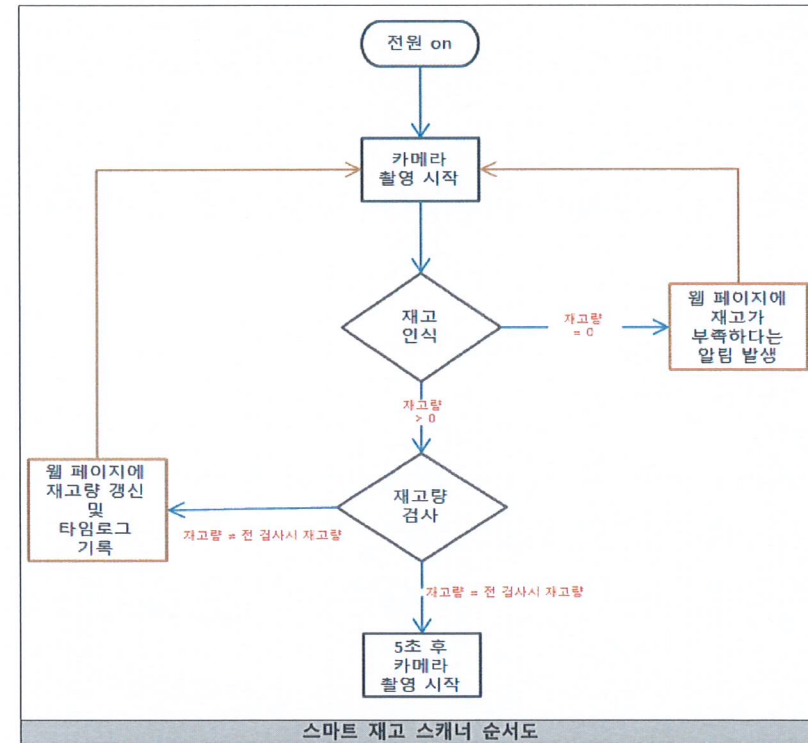
- 시공간 제약 없이 실시간 재고량 확인 가능
- 제품의 출고 타임로그가 기록
- 재고가 부족할 시 알람 기능
- 제품의 선호도 및 프라임 타임을 확인 가능

외부 프레임	
스마트 재고 스캐너 시작 Web page	
스마트 재고 스캐너 Streaming Web page	

2. 작품 설명

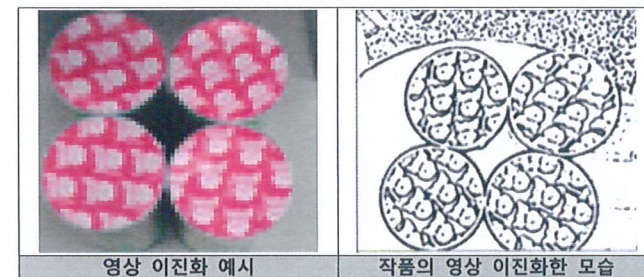
2-1 주요 동작 및 특징

○ 주요 동작 순서도



- 재고 인식에서 영상 이진화 변환, Hough Circles를 사용
- 작품을 처음 켜고 재고량은 고기 전의 재고량을 비교
- 작품이 꺼지면 Web page 접속불가

○ 영상 이진화 변환 (OpenCV 활용)



- 작품에서 영상 이진화 변환방식은 cv2.cvtColor(frame,cv2.COLOR_BGR2GRAY) 사용하여 frame을 이진화 처리
- 재고를 정밀하게 인식하기 위해 사용
- Hough Circles의 오차를 없애기 위해 사용

○ Hough Circles (OpenCV 활용)



- Hough Circles는 원 검출을 위해 사용
- 재고 윗면의 반지름을 범위로 설정
- 재고 윗면 중심들의 사이 간격을 Hough Circles에 입력
- Canny Edge를 사용하여 재고 윗면의 노이즈를 제거
- 재고에 원을 검출

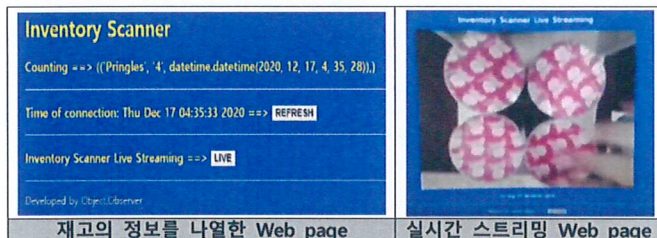
○ Web Server 구동

- 재고 정보를 보여주기 위해 Web page를 디스플레이로 선택
- Web page를 보여주기 위해 Web Server를 구동
- Flask인 마이크로 웹 프레임워크를 활용하여 Web Server를 구동

○ MySQL를 통해 DB 연동 및 정보 갱신

- 재고 정보를 Web page로 보여주기 위해 사용
- 재고 정보를 MySQL 테이블을 생성하여 정보를 입력
- 테이블 안에 타임스탬프를 넣어 재고량이 변할 시 타임로그가 남게 설정

○ HTML와 CSS로 만든 Web page



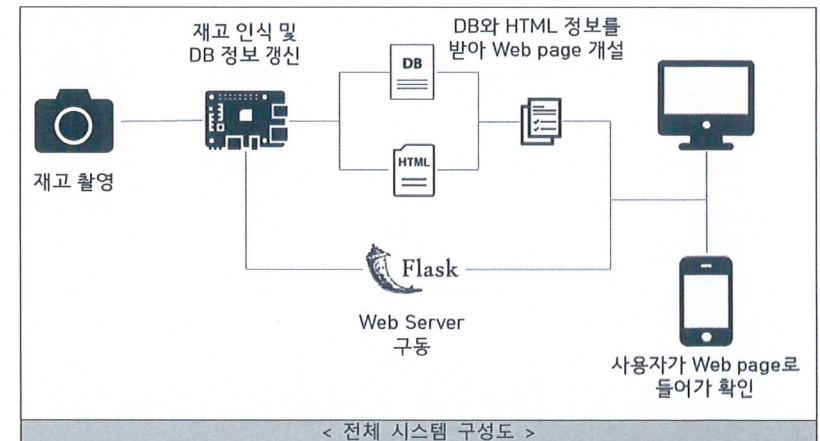
- Counting 항목은 재고명, 재고량, 최근에 재고 변화한 시간

- Last Sensors Reading 항목은 최근에 재고인식을 한 시간
- HTML과 CSS로 정보를 표현할 Web page 제작
- HTML로 Web page를 구도를 잡고 정보 출력
- CSS로 표현할 Web page 설계
- Web page는 재고의 수량을 표현하는 page, 스트리밍을 하는 page 2개가 존재
- Inventory Scanner Live Streaming 항목은 실시간 스트리밍 이동 링크 (스트리밍, 재고인식 하나만 가능, 동시에는 불가능)

2-2 전체 시스템 구성

○ 시스템 구성

구분	설명
하드웨어(HW)	라즈베리파이3 (Raspberry Pi 3 Model B)
	라즈베리파이 카메라모듈 V2, 8MP (RPI 8MP CAMERA BOARD)
	보조배터리 또는 전원 연결 (5pin, USB port)
소프트웨어 (SW)	Python
	MySQL
	Flask
	OpenCV
	HTML












○ 시스템 흐름

- 라즈베리파이 카메라모듈로 재고를 인식
- 재고의 정보를 DB에 전달
- Flask인 웹 프레임워크를 통해 Web Server 구동
- Web Server를 통해 DB, HTML 정보를 가지고 Web page를 표현
- Web page를 사용자가 확인

2-3 개발 환경(개발 언어, Tool, 사용 시스템 등)

구분	상세내용	
S/W 개발환경	OS	윈도우10, Raspbian buster
	개발환경(IDE)	Visual Studio Code
	개발도구	Visual Studio Code, Vim
	개발언어	Python, HTML, CSS
H/W 구성장비	디바이스	라즈베리파이 3 B
	센서	라즈베리파이 카메라모듈 V2
	언어	Python
프로젝트	의사소통관리	Kakao talk
관리환경	기타사항	Kakao talk, 오프라인 미팅

개발 언어	  
Tool	 
사용 시스템	   

3. 단계별 제작 과정

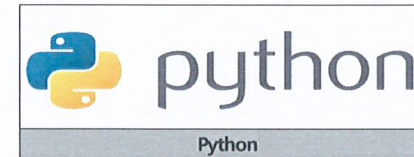
3-1 프레임 제작 과정



- 라즈베리파이와 카메라 모듈의 위치를 고려해 3D 모델 제작
- 초기 프레임은 열 배출이 미약한 구조
- Inventor를 이용하여 최적의 프레임을 설계하여 카메라 모양으로 제작

3-2 소프트웨어 설계 과정

○ Python



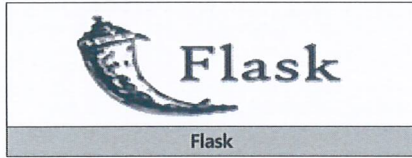
- Python을 기반으로 스마트 재고 스캐너를 제작
- Python을 사용하여 OpenCV를 영상처리에 활용
- Python을 기반으로 MySQL 및 Flask를 활용하여 서버를 구동
- 라즈베리파이 카메라 모듈 제어

○ OpenCV



- 영상처리를 위한 라이브러리
- 스마트 재고 스캐너에서 영상 이진화, Hough Circle 함수를 사용
- 재고 인식 및 실시간 스트리밍 담당

○ Flask



- Python으로 작성된 마이크로 Web 프레임워크
- 라우터를 설정하여 특정 Web page를 받는 Web Server 구동
- Web Server의 주소는 Wifi 주소를 활용

○ MySQL



- 데이터베이스 관리 시스템
- 재고의 정보를 실시간 갱신을 위해 사용
- 테이블을 이용하여 재고의 정보 기록

○ HTML, CSS



- HTML과 CSS로 정보를 표현할 Web page 제작
- 실시간 정보를 받아와 재고의 정보를 구조화
- Web page 2개 개설
- CSS 파일 한 개로 Web page 2개 모두 적용

4. 기타

4-1 소스코드

```
Python으로 MySQL을 사용하여 테이블 생성 코드
import pymysql

con = pymysql.connect(host="localhost", user="root", password="3695",
                      db='mysql', charset='utf8')

cur = con.cursor()

sql="create table sabjill(" # 테이블 생성 코드
    "Name varchar(100)," #
    "Count varchar(50)," #
    "Time timestamp," #
    "primary key (Name))"

cur.execute(sql)
con.commit()
con.close()
```

```
OpenCV를 통해 영상 이진화, HoughCircles를 사용하여 재고 인식 함수 코드
import numpy as np
import argparse
import cv2
import time

cap = cv2.VideoCapture(0)

global y
y=0
s=0
def detect(cap)
    ret, frame = cap.read()
    output = frame.copy()
    gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
    gray = cv2.GaussianBlur(gray,(5,5),0);
    gray = cv2.medianBlur(gray,5)
    gray =
    cv2.adaptiveThreshold(gray,255,cv2.ADAPTIVE_THRESH_GAUSSIAN_C,#
        cv2.THRESH_BINARY,11,3.5)

    kernel = np.ones((2,2),np.uint8)
    gray = cv2.erode(gray,kernel,iterations = 1)
```

```

gray = cv2.dilate(gray, kernel, iterations = 1)

circles = cv2.HoughCircles(gray, cv2.HOUGH_GRADIENT, 1, 20,
param1=30,
param2=65, minRadius=0, maxRadius=0) # Hough Circle 코드
c = 0;

if circles is not None:

    circles = np.round(circles[0,:]).astype("int")

    for (x, y, r) in circles:

        cv2.circle(output, (x, y), r, (0, 255, 0), 4)
        cv2.rectangle(output, (x - 5, y - 5), (x + 5, y + 5), (0, 128,
255), -1)
        time.sleep(3)

        c+=1

    y=c
    c=0
    return y

cap.release()
cv2.destroyAllWindows()

```

재고를 인식하는 코드의 함수를 따라 인식 후 MySQL에 데이터 저장 코드

```

import pymysql
from cir import detect
import cv2
#z=cir.detect(cap)
cap = cv2.VideoCapture(0)
z=detect(cap)
con = pymysql.connect(host="localhost", user="root", password="3695",
db='mysql', charset='utf8')

#now = time.asctime( time.localtime(time.time()) )
cur = con.cursor()

sql="UPDATE sabjill set Count = %s" # 데이터 갱신 코드

cur.execute(sql, (z))

```

```

con.commit()
con.close()

```

서버를 구축하는 전체적인 소스 코드

```

from flask import Flask, render_template, Response
app = Flask(__name__)
from camera_pi import Camera
import time
import pymysql

con = pymysql.connect(host="localhost", user="root", password="3695",
db='mysql', charset='utf8')

@app.route("/")
def index():
    timeNow = time.asctime( time.localtime(time.time()) )
    cursor = con.cursor()
    cur = con.cursor()
    sql= "select * from sabjill";
    cur.execute(sql)
    templateData = {
        'temp' : cur.fetchall()
    }
    con.commit()
    return render_template('index.html',**templateData)

@app.route('/camera')
def cam():
    """Video streaming home page."""
    timeNow = time.asctime( time.localtime(time.time()) )
    templateData = {
        'time': timeNow
    }
    return render_template('camera.html', **templateData)

def gen(camera):
    while True:
        frame = camera.get_frame()
        yield (b'--frame\r\n'
            b'Content-Type: image/jpeg\r\n\r\n' + frame + b'\r\n')

@app.route('/video_feed')
def video_feed():
    return Response(gen(Camera()),
                    mimetype='multipart/x-mixed-replace; boundary=frame')

if __name__ == '__main__':
    app.run(host='0.0.0.0', port = 8888, debug=True, threaded=True) # Web
Server 구동 코드

```

```

Web Server에 사용되는 html 소스 코드
<!doctype html>
<html>
<head>
  <title> Inventory Scanner </title>
  <link rel="stylesheet" href='../static/style.css?after'/>
</head>
<body>
  <h1> Inventory Scanner </h1>
  <h3> Counting ==> {{ temp }} </h3>
  <hr>
  <h3> Last Sensors Reading: {{ time }} ==> <a
href="/"class="button">REFRESH</a></h3>
  <hr>
  <h3> Inventory Scanner Live Streaming ==> <a href="/camera"
class="button">LIVE</a></h3> # Streaming page 링크 버튼 코드
  <hr>
  <p> Developed by ObjectObserver</p>
</body>
</html>

```

```

html를 꾸며주는 css 소스 코드
body{
  background: blue;
  color: yellow;
  padding:1%
}
.button {
  font: bold 15px Arial;
  text-decoration: none;
  background-color: #EEEEEE;
  color: #333333;
  padding: 2px 6px 2px 6px;
  border-top: 1px solid #CCCCCC;
  border-right: 1px solid #333333;
  border-bottom: 1px solid #333333;
  border-left: 1px solid #CCCCCC;
}

```

4-2 참고문헌

Hough Circle 참고문헌	https://en.wikipedia.org/wiki/Circle_Hough_Transform https://sonsondiary.tistory.com/entry/%EC%98%81%EC%83%81%EC%B2%98%EB%A6%AC-Hough-Circle-Detection
영상 이진화 참고문헌	https://darkpgmr.tistory.com/115
HTML 참고문헌	https://www.youtube.com/watch?v=tZooW6PritE&list=PLuHgQVnccGMDZP7FJ_ZsUrdCGH68ppvPb
CSS 참고문헌	https://code.tutsplus.com/tutorials/5-techniques-to-a-cquaint-you-with-css-3--net-2338
Web Server 참고문헌	https://velog.io/@decody/%ED%8C%8C%EC%9D%B4%EC%8D%AC-Flask%EB%A1%9C-%EA%B0%84%EB%8B%A8-%EC%9B%B9%EC%84%9C%EB%B2%84-%EA%B5%AC%EB%8F%99%ED%95%98%EA%B8%B0
MySQL 참고문헌	https://www.everdevel.com/MySQL/creating-table/

4-3 본인 작품 YouTube 영상

https://youtu.be/1u_RZtMAeNQ