

# **LIVE SECURITY**

## **:실시간 CCTV 감지 시스템**

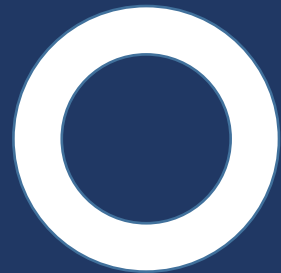
MID - Presentation

오픈소스(가)분반 7팀

남채린, 손준오, 송병현, 장준용

# Table of contents

---



## 01

### Aim & Background

목표 & 핵심내용, 개발동기, 극복방안, 차별점, 시스템 요약

## 02

### Contents

시스템 구조, 카메라 연결, APPENDIX, 구현테스트, 향후 계획

## 03

### Source & References

출처 및 참고자료

# 목표 & 핵심내용

## 목표 :

허점이 많은 기존의 도난방지 시스템과 손님 출입 벨을 보완하여 직접 매장주인과 근무자에게 알림을 제공하는 앱을 개발하는 것

## 핵심내용 :

카운터 동작인식 기능을 통하여 기존 무인 편의점 시스템들과는 다르게 도난을 즉각적으로 확인할 수 있고, 출입 감지를 통해 근무자의 더욱 수월한 업무 수행을 가능하게 한다.

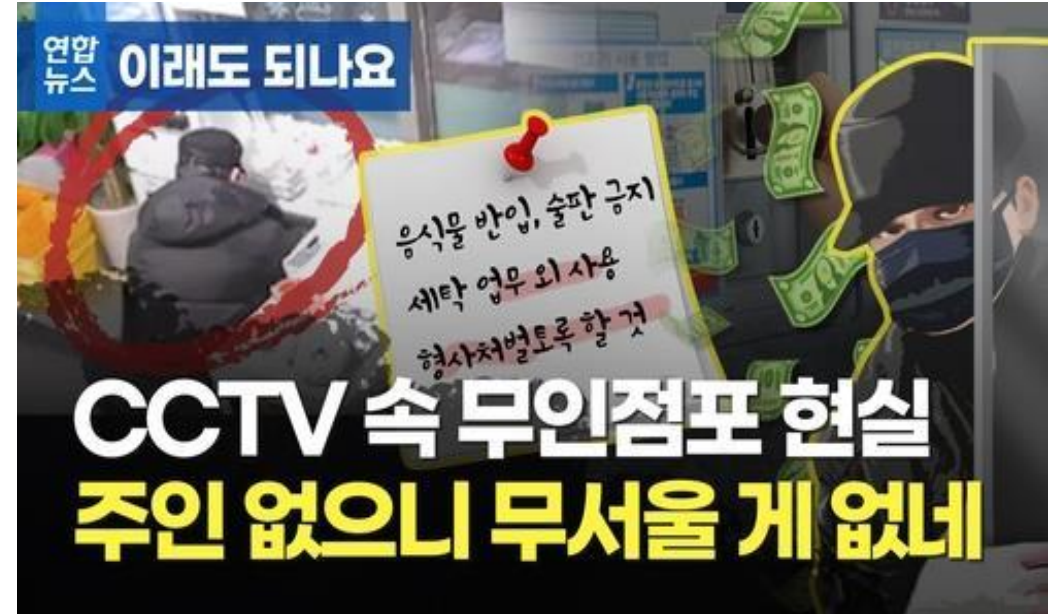
# 개발 동기

## 1. 무인 병행 점포에 대한 도난사건 증가

- 최근 들어 **무인운행을 병행**하는 점포의 도난 사건 발생
- 이를 신속하게 대응하기 위해 CCTV의 단점을 보완
- 실제로 팀원 중 한명이 근무하는 편의점에서 부실한 도난 방지 시스템으로 인해 도난 범죄를 겪음

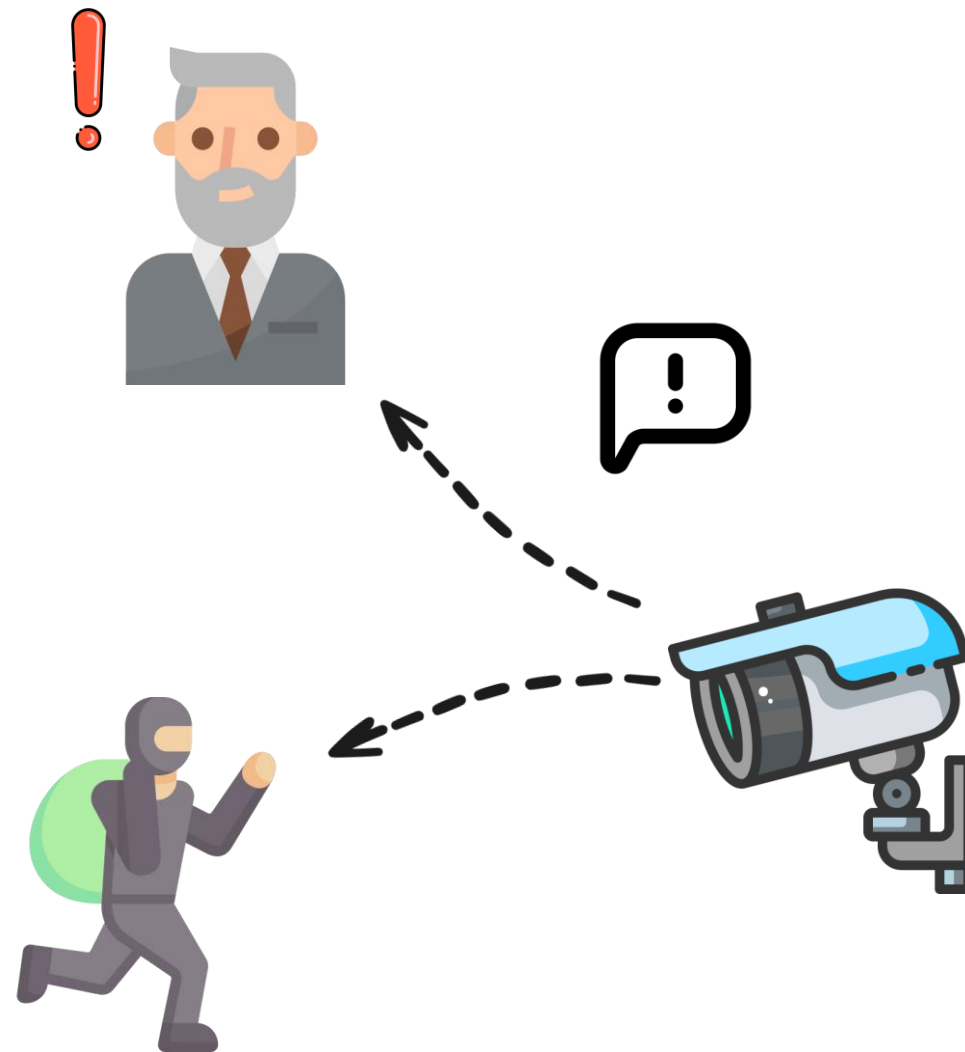
## 2. 1인 매장 출입 손님에 대한 부족한 시스템

- 혼자 매장을 운영할 시 손님 출입에 대해 알아차리지 못할 수 있다
- '손님 출입 벨'이 존재하나 성능이 확실하게 발휘되지 못할 때가 존재



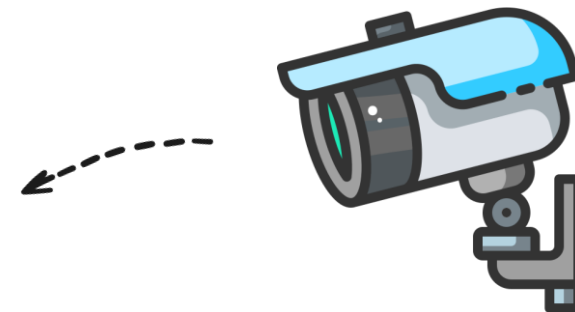
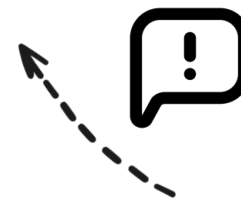
<편의점 내 카운터 도난 장면>

# 극복 방안



Part 3

# 극복 방안



# 차별점



<이마트24의 출입 인증기>



<동작감지 비상벨>

## 기존 무인매장의 시스템

**출입인증기:** 고객의 카드를 인식하여 출입하게 하는 시스템. 고객의 신원을 파악하고 고객의 출입정보를 저장한다

**단점:** 고객이 매장 내부에서 도난행위를 할 수 있는지 실시간으로 알 수 없으며, 고객들이 동시에 들어가면 카드를 인식하지 않은 고객의 신분을 파악할 수 없다.

**동작감지 비상벨:** 카운터 내부나 입구의 동작을 감지하여 울리는 벨. 카운터의 동작이 감지되면 벨이 울려 범죄자를 쫓아낸다. 또한 고객의 출입을 감지하여 근무자가 들을 수 있도록 벨을 울려준다.

**단점:** 도난행위자에게 겁을 먹일 뿐 실시간으로 도난 행위를 막을 순 없다. 또한 고객의 출입을 감지하여 울려도 근무자가 창고에 있어 듣지 못하는 등 알아차리지 못하는 경우가 많다.

# 차별점



<이마트24의 출입 인증기>



<동작감지 비상벨>

## 실시간 CCTV 감지 시스템

- 실시간으로 CCTV 동작감지를 통해 카운터의 도난행위를 즉각적으로 운영자에게 알려 **출입인증기**와 **동작감지 비상벨의 단점**을 보완할 수 있다.
- 실시간으로 CCTV 동작감지를 통해 고객의 출입을 근무자의 핸드폰에 즉각적으로 알림을 제공해 **동작감지 비상벨의 단점**을 보완할 수 있다.



# 시스템 요약

실시간 CCTV  
동작감지 시스템

## 고객 출입 감지 시스템

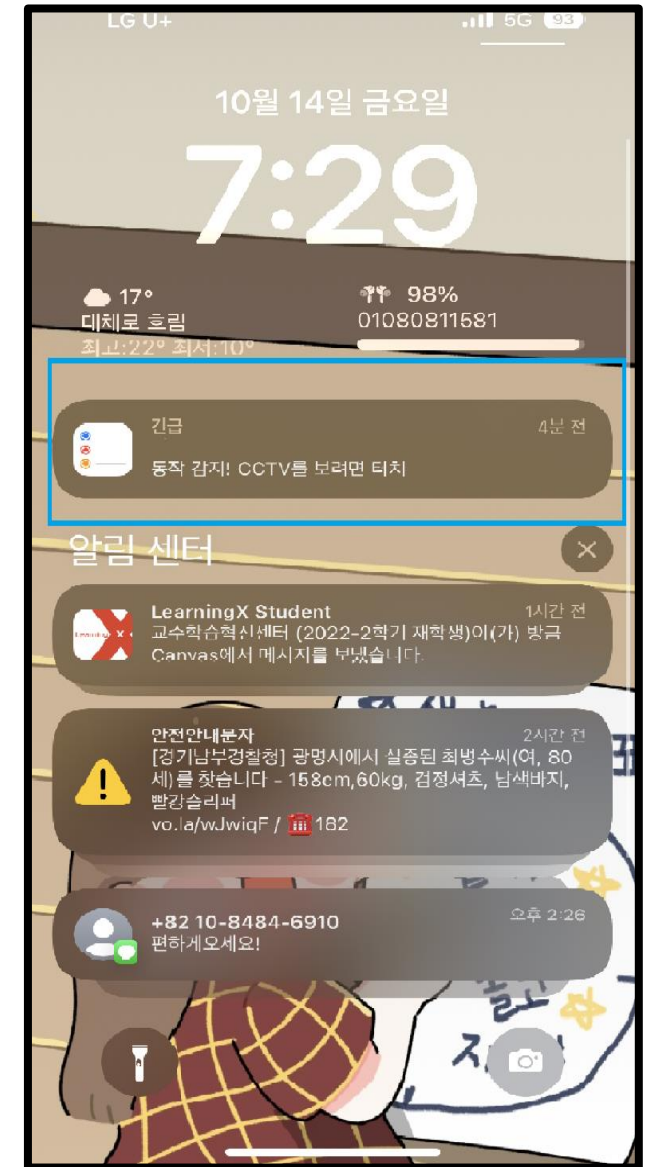
고객이 출입하면 운영자의 휴대폰  
으로 진동과 알림을 제공

## 도난 방지 시스템

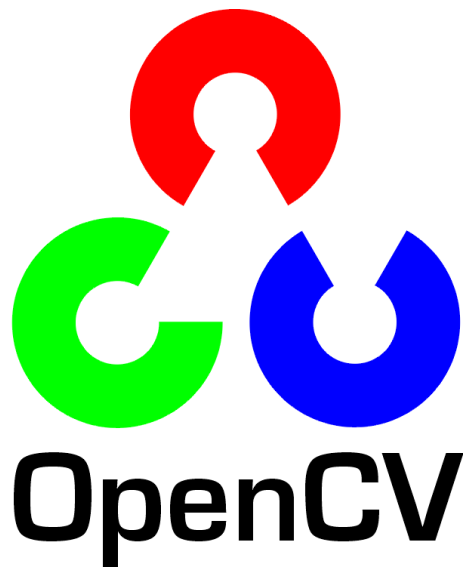
실시간으로 카운터를 녹화해  
도난을 실시간으로 감지

# Contents

시스템 구조, 카메라 연결, APPENDIX, 구현테스트, 향후 계획

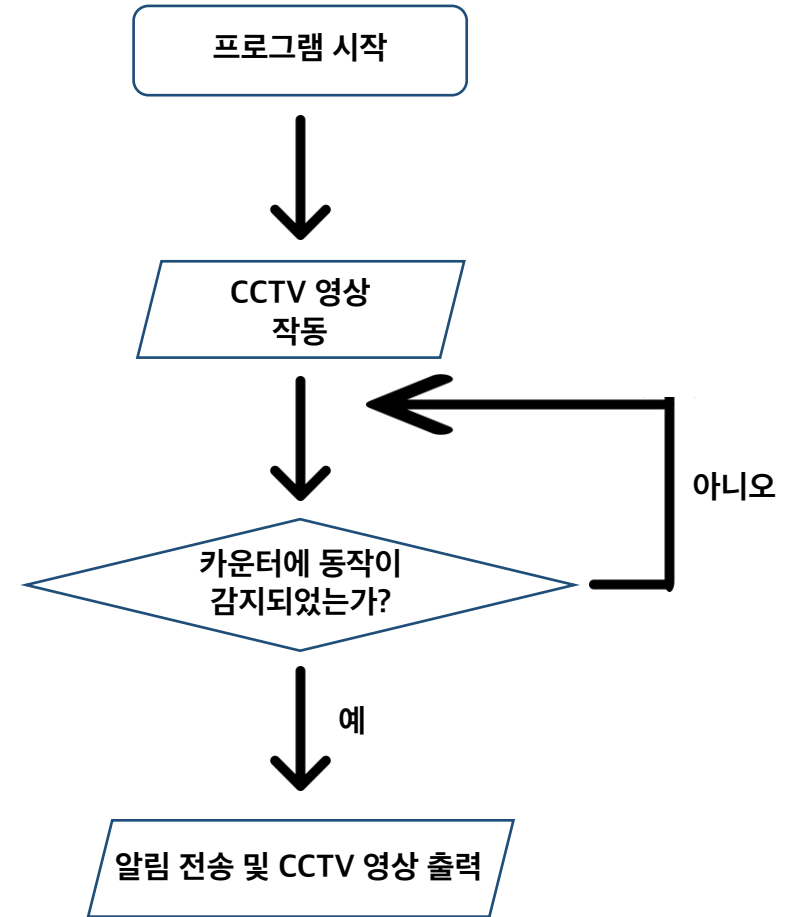
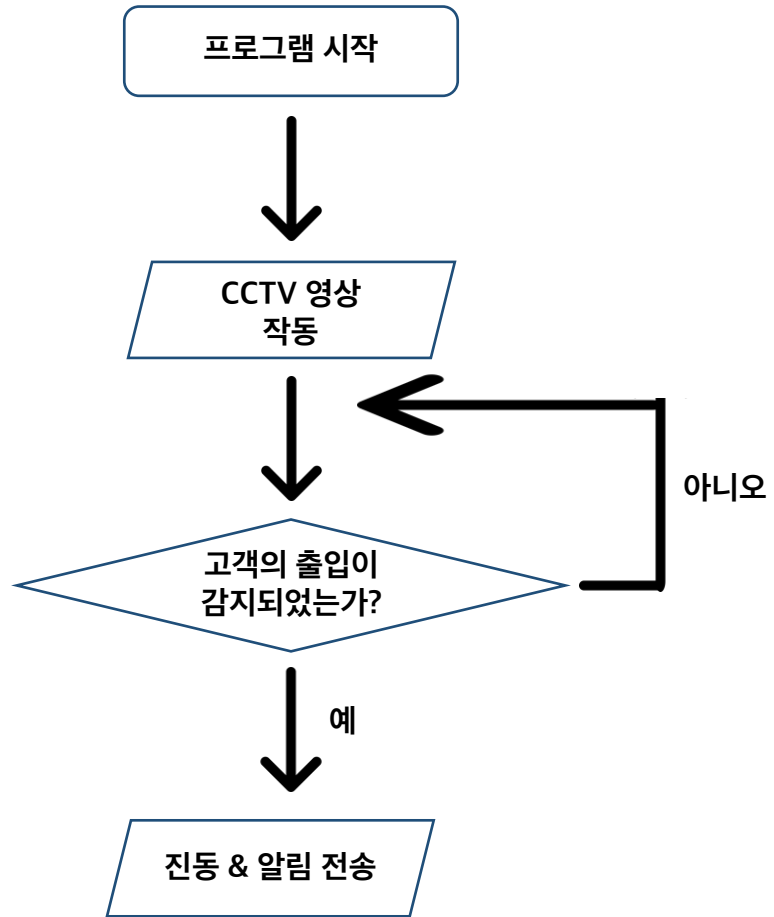


# 시스템 구현

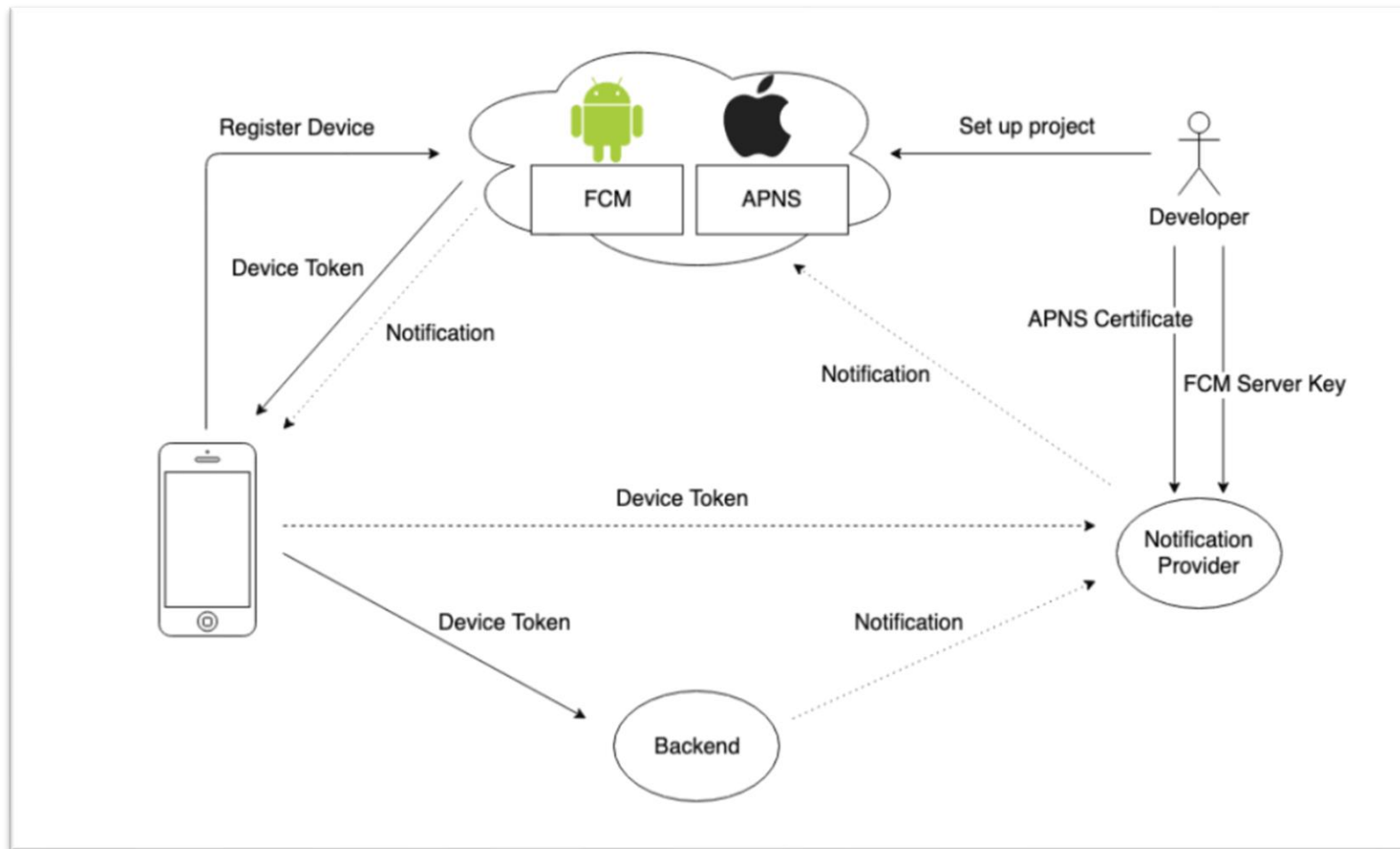


Android  
Studio

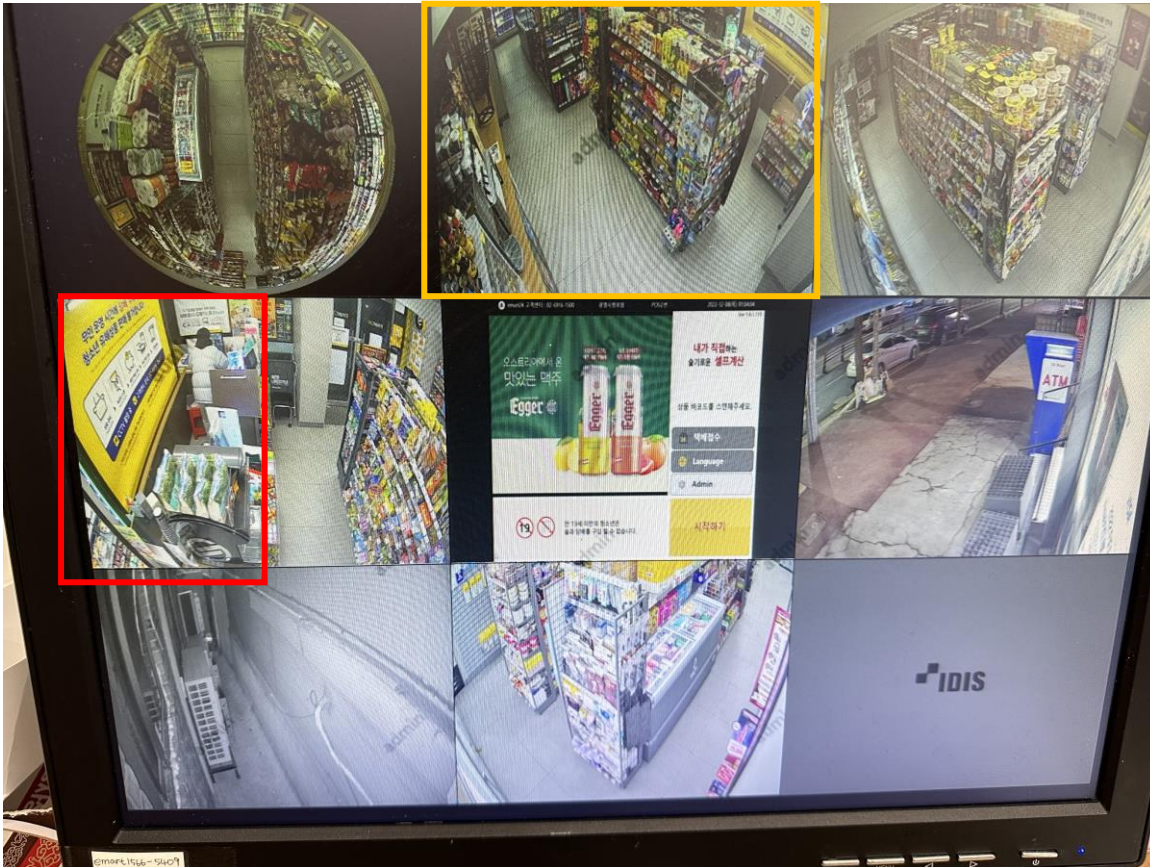
# 시스템 구조



# 시스템 구조



# 카메라 연결



<실제 편의점 CCTV>

**도난 방지 시스템:** 카운터를 촬영하는 CCTV(사진의 **적색 사각형**)를 연결하여 구현

**이유**

- 기술의 한계로 매장의 상품을 도난하는 것은 막을 수 없다

- 무인매장은 카운터가 없지만, 무인운영과 유인운영을 병행하는 매장은 카운터가 존재

- 도난 방지 시스템의 목표는 카운터에 있는 귀중품(돈, 담배 등)의 도난을 막는 것

**고객 출입 감지 시스템:** 매장 입구를 촬영하는 CCTV(사진의 **주황색 사각형**)를 연결하여 구현

**이유**

- 출입하는 손님들을 더욱 효율적으로 감지할 수 있다

## APPENDIX

```
7  import cv2
8  import datetime
9  from PIL import ImageFont, ImageDraw, Image
10 import numpy as np
11 import os
12 from tkinter import *
13
14 def get_diff_img(frame_a, frame_b, frame_c, threshold): # 쉽게 말해서 프레임이 바뀌면 차이가 있구나 그리고 얼마정도 차이가 있구나를
15     # 3 프레임의 영상을 모두 흑백으로 전환 -- 흑백은 처리시간이 3분의1이므로 흑백으로 바꾼것(RGB색) # 출력해주는 거임
16     frame_a_gray = cv2.cvtColor(frame_a, cv2.COLOR_BGR2GRAY)
17     frame_b_gray = cv2.cvtColor(frame_b, cv2.COLOR_BGR2GRAY)
18     frame_c_gray = cv2.cvtColor(frame_c, cv2.COLOR_BGR2GRAY)
19
20     # 1,2 프레임, 2,3 프레임 영상들의 차를 구함
21     diff_ab = cv2.absdiff(frame_a_gray, frame_b_gray)
22     diff_bc = cv2.absdiff(frame_b_gray, frame_c_gray)
23
24     # 영상들의 차가 threshold 이상이면 값을 255(백색)으로 만들어줌
25     ret, diff_ab_t = cv2.threshold(diff_ab, threshold, 255, cv2.THRESH_BINARY)
26     ret, diff_bc_t = cv2.threshold(diff_bc, threshold, 255, cv2.THRESH_BINARY)
27
28     # 두 영상 차의 공통된 부분을 1로 만들어줌
29     diff = cv2.bitwise_and(diff_ab_t, diff_bc_t)
30     # 영상에서 1이 된 부분을 적당히 확장해줌(morpholgy)
31     k = cv2.getStructuringElement(cv2.MORPH_CROSS, (3, 3))
32     diff = cv2.morphologyEx(diff, cv2.MORPH_OPEN, k)
33
34     # 영상에서 1인 부분의 갯수를 셈
35     diff_cnt = cv2.countNonZero(diff)
36     return diff, diff_cnt
37
38
```

# APPENDIX

```
39  # 카메라 영상을 받아올 객체 선언 및 설정(영상 소스, 해상도 설정)
40  capture = cv2.VideoCapture(0)
41  capture.set(cv2.CAP_PROP_FRAME_WIDTH, 640)
42  capture.set(cv2.CAP_PROP_FRAME_HEIGHT, 480)
43
44  fourcc = cv2.VideoWriter_fourcc(*'XVID')    # 영상을 기록할 코덱 설정
45  font = ImageFont.truetype('fonts/SCDream6.otf', 20) # 글꼴파일을 불러옴
46  is_record = False                          # 녹화상태는 처음엔 거짓으로 설정
47  on_record = False
48
49  threshold = 40    # 영상 차이에 사용할 threshold 설정
50  diff_max = 10     # 영상 차이 픽셀의 개수(이것 이상이면 움직임이 있다고 결정)
51  cnt_record = 0    # 영상 녹화 시간 관련 변수
52  max_cnt_record = 5 # 최소 촬영시간
53
54  # 초기 프레임으로 사용할 프레임들을 저장
55  ret, frame_a = capture.read()
56  ret, frame_b = capture.read()
```



## APPENDIX

```
58 while True:
59     # 현재시각을 불러와 문자열로저장
60     now = datetime.datetime.now()
61     nowDatetime = now.strftime('%Y-%m-%d %H:%M:%S')
62     nowDatetime_path = now.strftime('%Y-%m-%d %H_%M_%S') # 파일이름으로는 :를 못쓰기 때문에 따로 만들어줌
63
64     # 현재 영상을 입력받아 움직임 감지
65     ret, frame_c = capture.read()
66     diff, diff_cnt = get_diff_img(frame_a=frame_a, frame_b=frame_b, frame_c=frame_c, threshold=threshold)
67
68     # 움직임이 일정 이상일시
69     if diff_cnt > diff_max:
70
71
72     # 움직임이 일정 이상이면(화면 캡처)
73     | cv2.imwrite("capture/capture" + nowDatetime_path + ".png", frame) # 파일이름(한글안됨), 이미지
74
75     # 움직임이 일정 이상이면(영상 녹화)
76     | is_record = True # 녹화 준비
77     | if on_record == False:
78     |     | video = cv2.VideoWriter("움직임이 감지됨 " + nowDatetime_path + ".avi", fourcc, 1, (frame_c.shape[1], frame_c.shape[0]))
79     |     cnt_record = max_cnt_record
80     if is_record == True: # 녹화중이면
81         | print('녹화 중')
82         | video.write(frame_c) # 현재 프레임 저장
83         | cnt_record -= 1 # 녹화시간 1 감소
84         | on_record = True # 녹화중 여부를 참으로
85     if cnt_record == 0: # 녹화시간이 다 되면
86         | is_record = False # 녹화관련 변수들을 거짓으로
87         | on_record = False
88
89     # 영상 차이를 출력(실제 사용시는 꺼도 됨)
90     cv2.imshow("diff", diff)
91     frame = np.array(frame_c)
```

# APPENDIX

```
7  # 알림
8  from tkinter import *
9  import wo
10
11  root=Tk()
12  root.title("알림")
13  root.geometry("320x240")
14
15  def start():
16      wo.cm()
17      root.quit()
18
19  def done():
20      root.quit()
21
22  Label(root, text="움직임 감지됨\n카메라 연결", padx=20, pady=20).pack(side="top")
23
24
25  frame_run = Frame(root)
26  frame_run.pack(fill="x", padx=5, pady=5)
27
28  btn_close = Button(frame_run, padx=5, pady=5, text="아니오", width=12, command=done)
29  btn_close.pack(side="bottom", padx=5, pady=5)
30
31  btn_start = Button(frame_run, padx=5, pady=5, text="예", width=12, command=start)
32  btn_start.pack(side="bottom", padx=5, pady=5)
33
34  root.mainloop()
```

# APPENDIX

```
import firebase_admin
from firebase_admin import credentials
from firebase_admin import messaging

cred_path = "AIzaSyCXFGa4f_FQqDdbFcf8-Z55Vo26PEIiu0Y"
cred = credentials.Certificate(cred_path)
firebase_admin.initialize_app(cred)

registration_token = 'cBPtDKvuSBK1C7jsu9j5Tk:APA91bE52VQRECUAILH4XVmlcmMuISbV9QTHZgr3II_03cbuSha9a-MB15xgop9jlQ6-tSdBRumj5zu8pdwCgvK2ySQXt-RMTQveK34pQjS'
message = messaging.Message(
    notification = messaging.Notification(
        title='카운터 동작 감지',
        body='CCTV를 확인하시려면 클릭하십시오'
    ),
    token=registration_token,
)
response = messaging.send(message)
```

파이어베이스 서버에서 python코드를 통해 모바일로 알림을 보낸다

```
22 minifyEnabled false
23 proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
24 }
25 }
26 compileOptions {
27     sourceCompatibility JavaVersion.VERSION_1_8
28     targetCompatibility JavaVersion.VERSION_1_8
29 }
30 }
31 }
32 dependencies {
33     implementation 'com.google.firebase:firebase-messaging:21.1.0'
34     implementation platform('com.google.firebase:firebase-bom:31.1.0')
35     implementation 'com.google.firebase:firebase-analytics'
36     implementation 'com.android.support:appcompat-v7:28.0.0'
37     implementation 'com.android.support.constraint:constraint-layout:2.0.4'
38     testImplementation 'junit:junit:4.13.2'
39     androidTestImplementation 'com.android.support.test:runner:1.0.2'
40     androidTestImplementation 'com.android.support.test.espresso:espresso-core:3.0.2'
41 }
42 apply plugin: 'com.google.gms.google-services'
```

```
1 // Top-level build file where you can add configuration options common to all sub-projects/modules.
2
3 buildscript {
4     repositories {
5         // Make sure that you have the following two repositories
6         google() // Google's Maven repository
7         mavenCentral() // Maven Central repository
8     }
9     dependencies {
10         // Add the dependency for the Google services Gradle plugin
11         classpath 'com.google.gms:google-services:4.3.13'
12     }
13 }
14
15
16 plugins {
17     id 'com.android.application' version '7.3.1' apply false
18     id 'com.android.library' version '7.3.1' apply false
19 }
20
21
```

모바일 푸시 알림을 위해 안드로이드 스튜디오에 파이어베이스 모듈 추가

위 코드를 통해 앱이 백그라운드에서 실행될 때 메시지를 전송받을 수 있다.

# 구현 테스트

실제로쓸거 > 움직임 감지 CCTV.py > ...

37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60

문제

원래

현재시간 : 2022-12-02 00:43:49

움직임 감지 CCTV.py

움직임이 감지됨 2022-12-02 0...

움직임이 감지됨 2022-12-02 0...

현재 시간

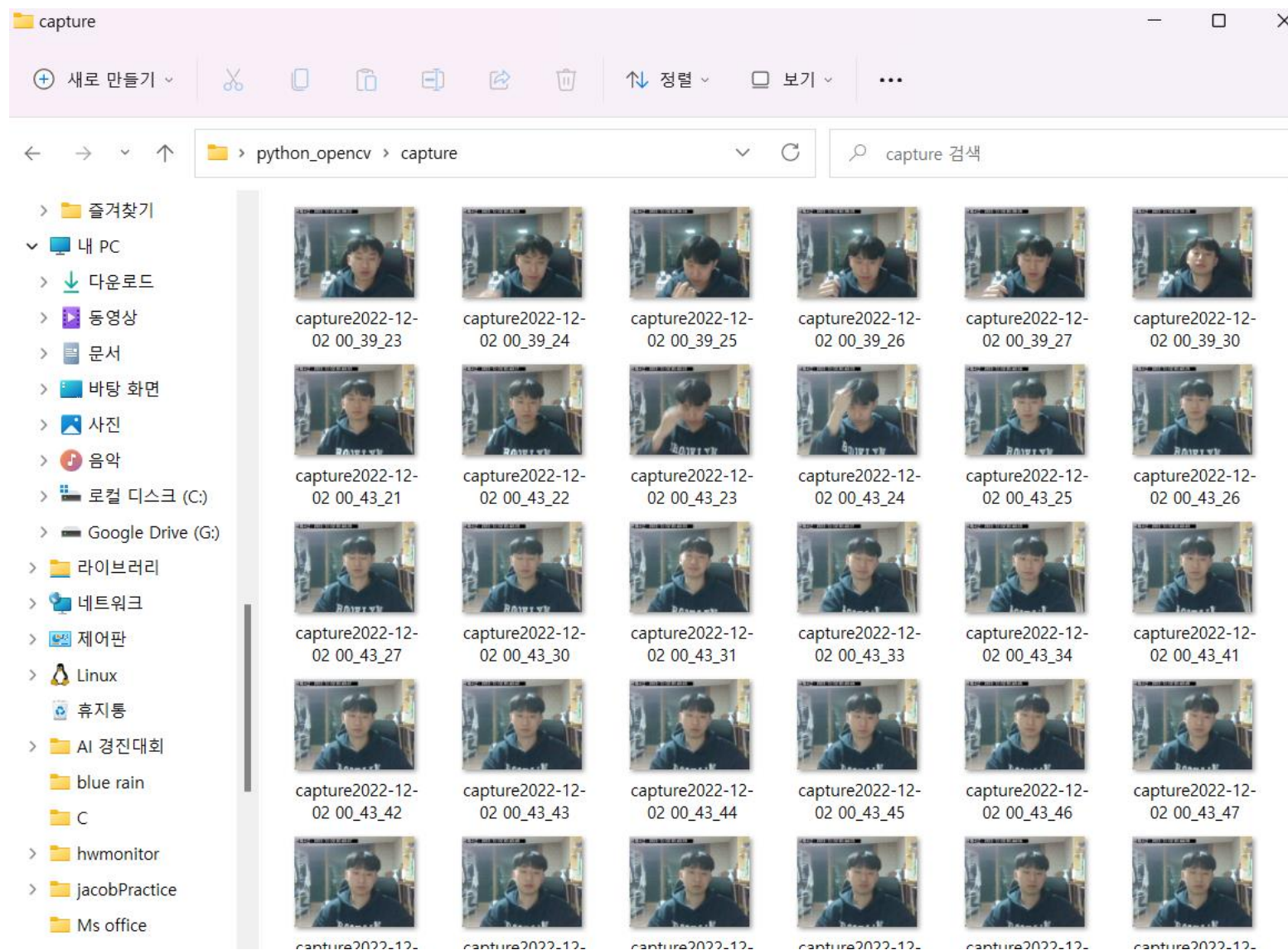
개요

타임라인

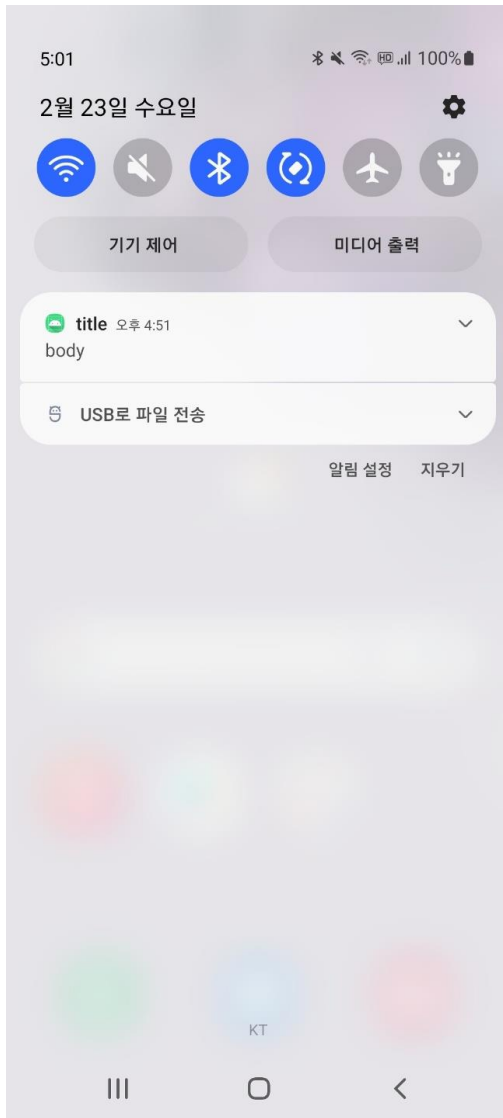
출 113,

## Part 4

# 구현 테스트



# 구현 테스트



**CCTV에서 동작 감지 시 알림이 오는 것을 확인할 수 있다.**

# Source & References

출처 및 참고자료



# 출처 & 참고자료

[https://www.youtube.com/watch?v=bKPIcoou9N8&list=PLMsa\\_0kAjjrd8hYYCwbAuDsXZmHpgHvIV&index=3](https://www.youtube.com/watch?v=bKPIcoou9N8&list=PLMsa_0kAjjrd8hYYCwbAuDsXZmHpgHvIV&index=3)

<https://pyimagesearch.com/2015/05/25/basic-motion-detection-and-tracking-with-python-and-opencv/>

<https://velog.io/@jun7332568/flutter%ED%94%8C%EB%9F%AC%ED%84%B0-%ED%91%B8%EC%8B%9C%EC%95%8C%EB%9E%8C%EC%95%88%EB%93%9C%EB%A1%9C%EC%9D%B4%EB%93%9C-%EA%B5%AC%ED%98%84%ED%95%98%EA%B8%B0-with-firebase>

<https://www.youtube.com/@GDSB>

<https://stackoverflow.com/questions/40514508/opencv-detect-movement-in-python>

<https://github.com/opencv/opencv/tree/master/data/haarcascades>

<https://sonagiya.tistory.com/entry/FCM-%ED%91%B8%EC%8B%9C-%EB%B3%B4%EB%82%B4%EA%B8%B0-with-Python>

Part 1

# 출처 & 참고자료

-파이어 베이스 링크

<https://console.firebase.google.com/project/opensource-team7/overview?hl=ko4>

-오픈소스 7팀 깃허브

<https://github.com/Opensource7team/Opensource7team>