

# Self-Updating Map Robot P1

::: 실내 매장의 변화를 검출하여 지도를 업데이트하는 인공지능

# 팀원 소개



박규동



공은비



김규리



김송일



박병수

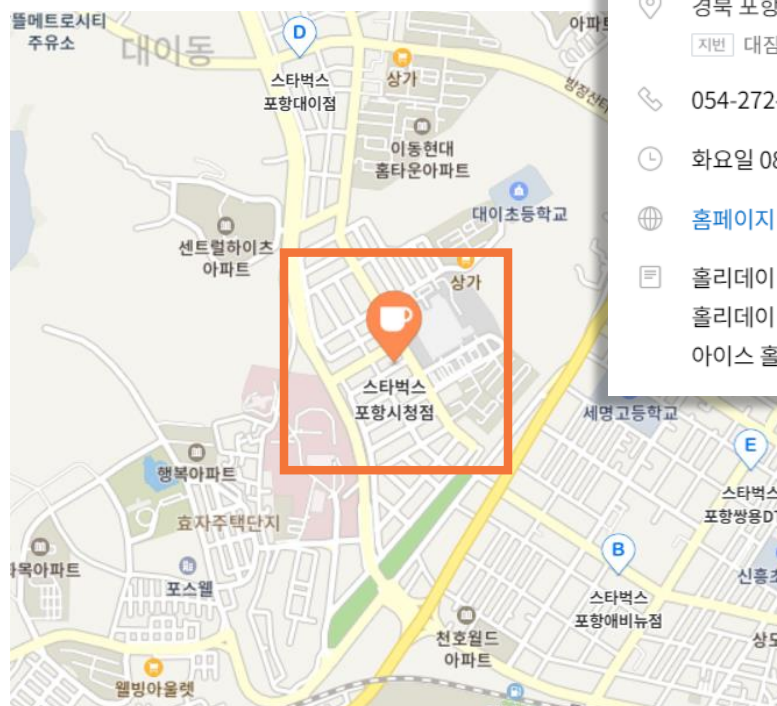


이수진

# CONTENTS

- 01 주제 선정 배경
- 02 Project Overview
- 03 P1 Spec
- 04 데이터 수집
- 05 POI Change Detection
- 06 Mapping
- 07 Conclusions

# 01 주제 선정 배경



## 스타벅스 포항시청점

카페 · 리뷰 46 >

☆ ↗ 주변

출발

도착

경북 포항시 남구 시청로 5

지번 대잠동 1005-1 (우) 37685

054-272-9737

화요일 08:00~22:00 12/24

홈페이지 · 블로그 · 페이스북 · 인스타그램

홀리데이 카라멜 돌체 프라푸치노 6,600원

홀리데이 카라멜 돌체 라떼 6,100원

아이스 홀리데이 카라멜 돌체 라떼 6,100원

## What is Point-of-Interest (POI)?

- 사용자에게 유용한 정보인 관심 지점

## 지도 최신성 유지의 중요성

- 국내 공간 정보는 해마다 30% 이상 변화
- POI 변화는 빠르고 비정기적으로 발생
- 지속적으로 업데이트되지 않은 지도는 사용자의 혼란 야기

# 01 주제 선정 배경



부평지하상가 쇼핑몰 내부

## 실내 지도의 필요성

- GPS, Wi-Fi 문제로 실내에서 위치 정보 알기 어려움
- 실외 지도처럼 실내 공간 정보를 얻을 수 있는 플랫폼 필요

## POI 변화가 잦은 쇼핑몰

- 실내 공간 중 특히 쇼핑몰의 변화 주기 빠름
- POI 변화가 잦은 쇼핑몰의 공간 정보를 업데이트하는 기술 필요

## 02 Project Overview

### 프로젝트 개요

- 대상 공간을 쇼핑몰로 지정하고 POI를 쇼핑몰 내 개별 매장으로 정의
- 시간 간격을 두고 촬영한 매장 이미지를 비교하여 매장의 변화 여부 판단
- 변화가 발생했을 경우 지도 업데이트

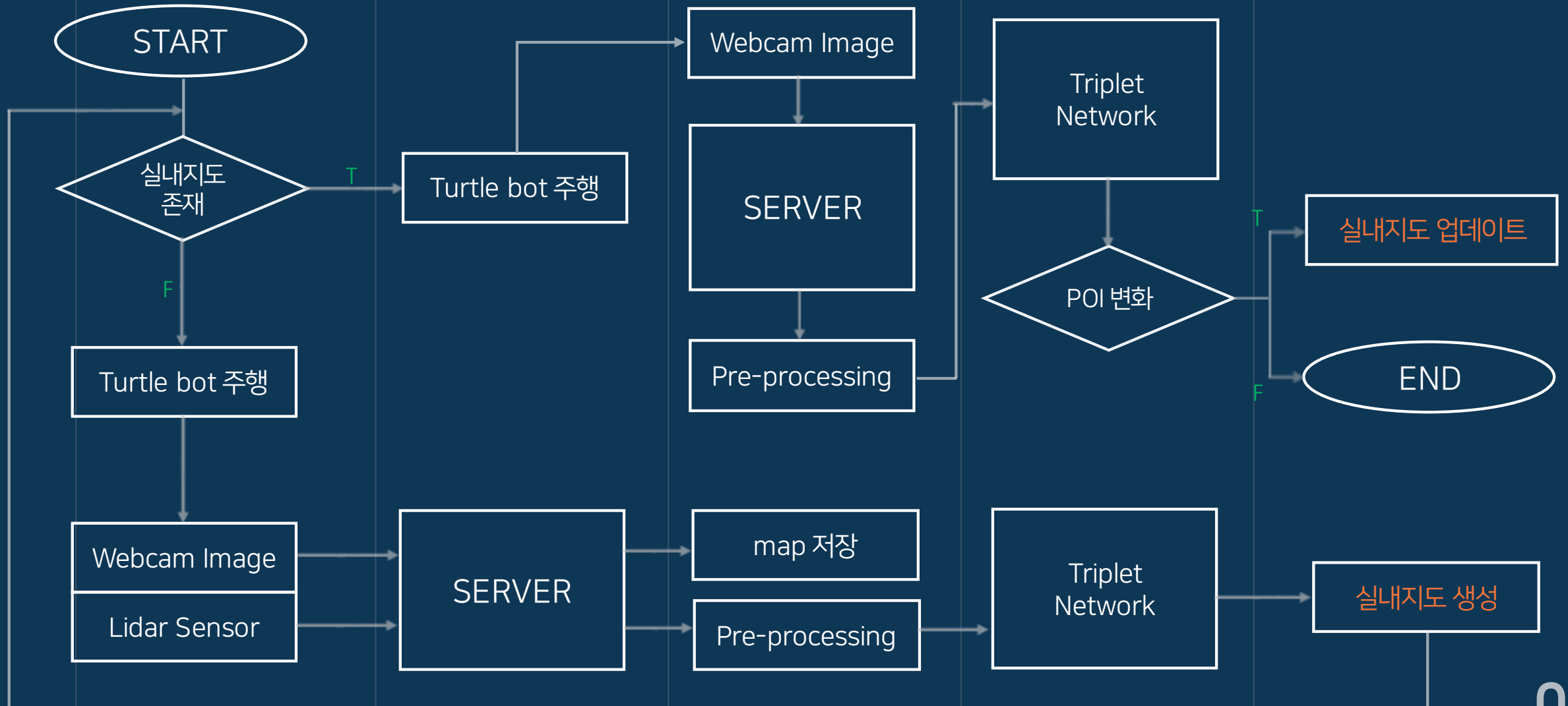
### Process

- 로봇이 쇼핑몰을 주행하며 매장 영상 촬영하고 동시에 라이다 센서를 통해 맵을 생성한다
- 매장 정보와 맵을 결합하여 실내 지도를 만든다
- 몇 개의 매장이 변화한 후 로봇이 동일한 공간을 재촬영한다
- 현재 매장과 과거 매장 이미지를 비교하여 변화가 있는지 판별한다
- 변화가 검출된 경우 변화된 매장 정보를 지도에 업데이트한다

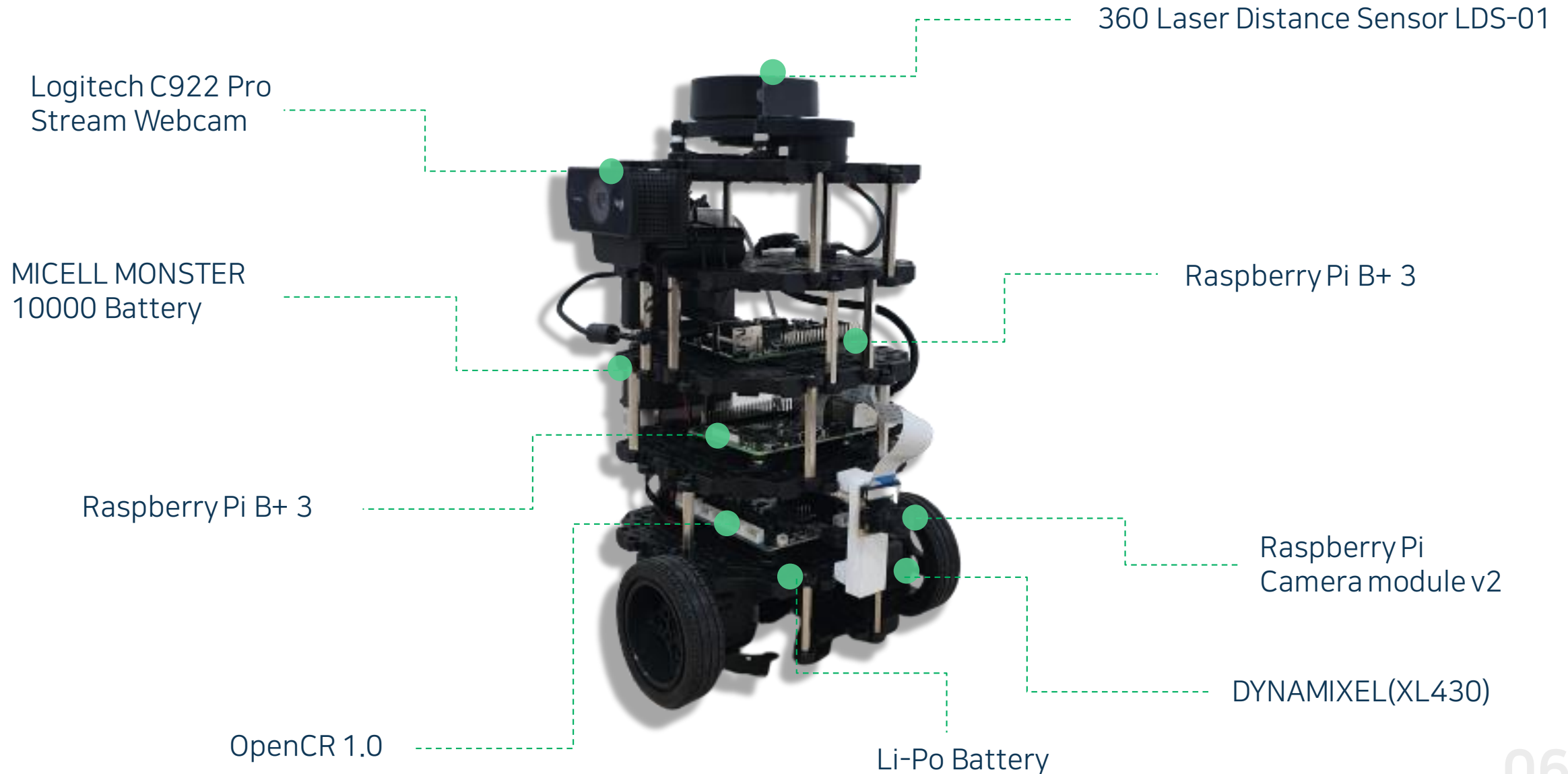


## 02 Project Overview

### Project Flow

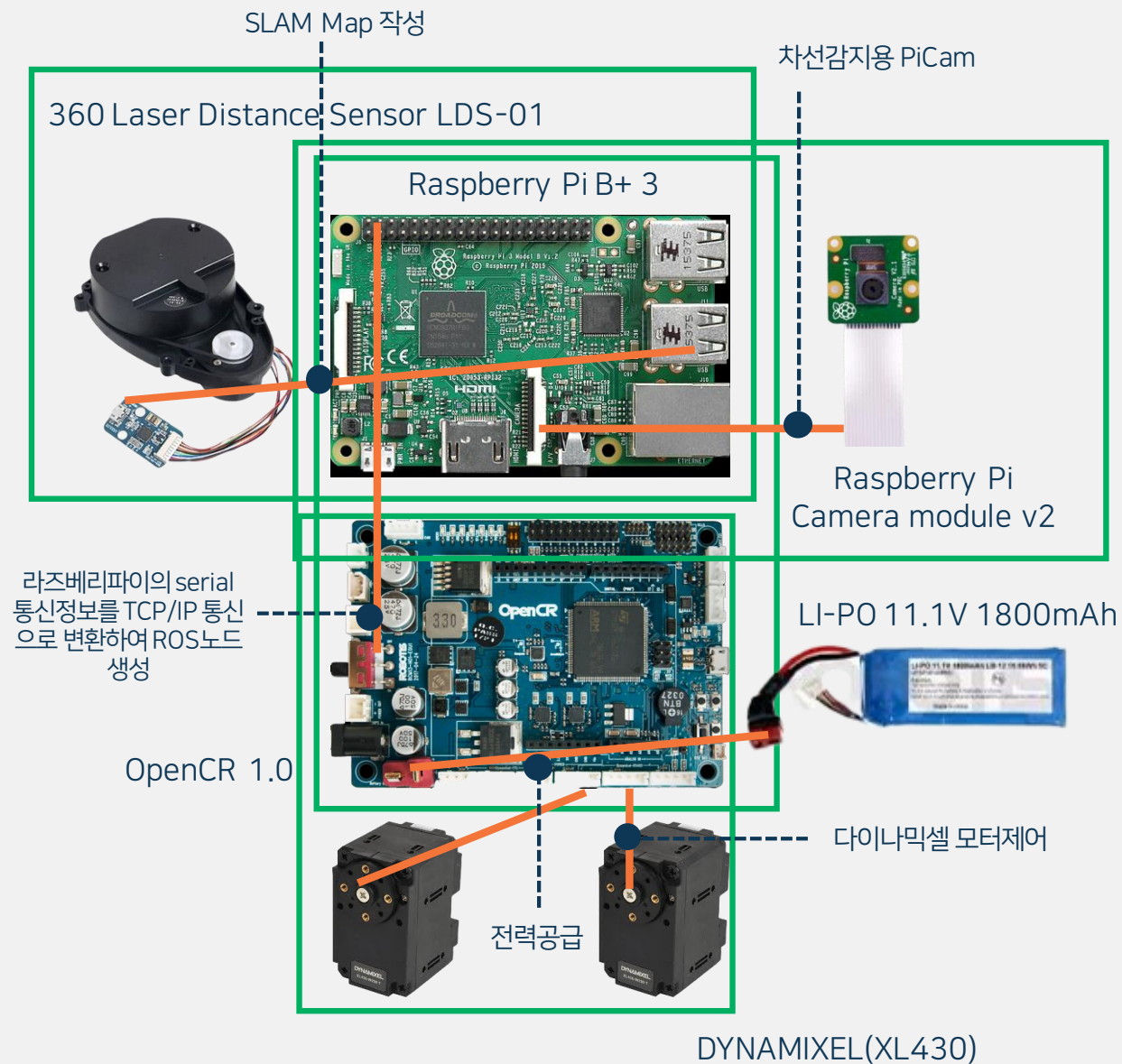


## 03 P1 Spec

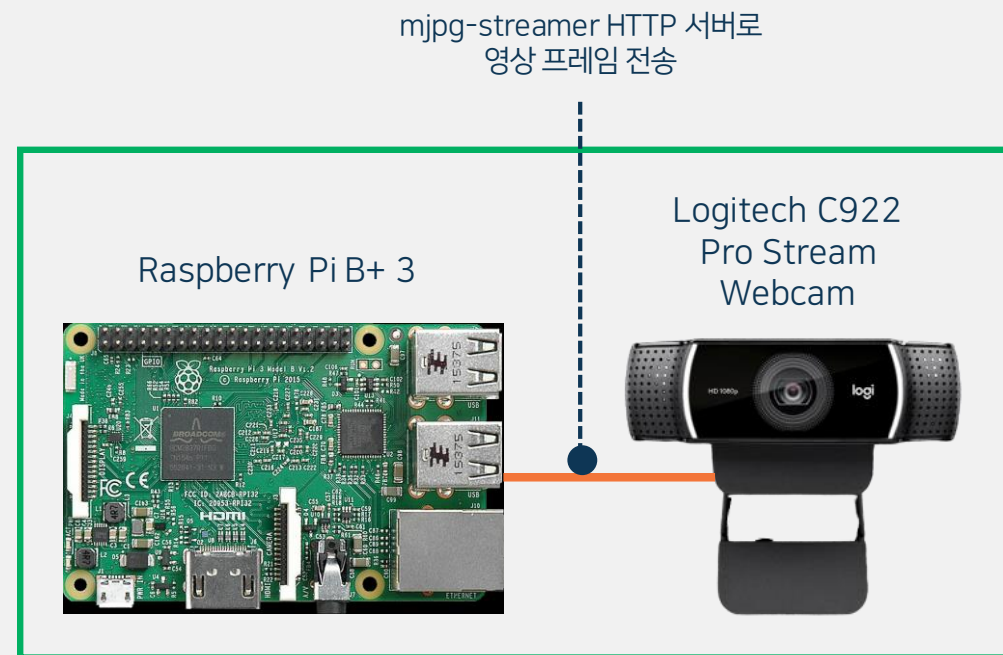




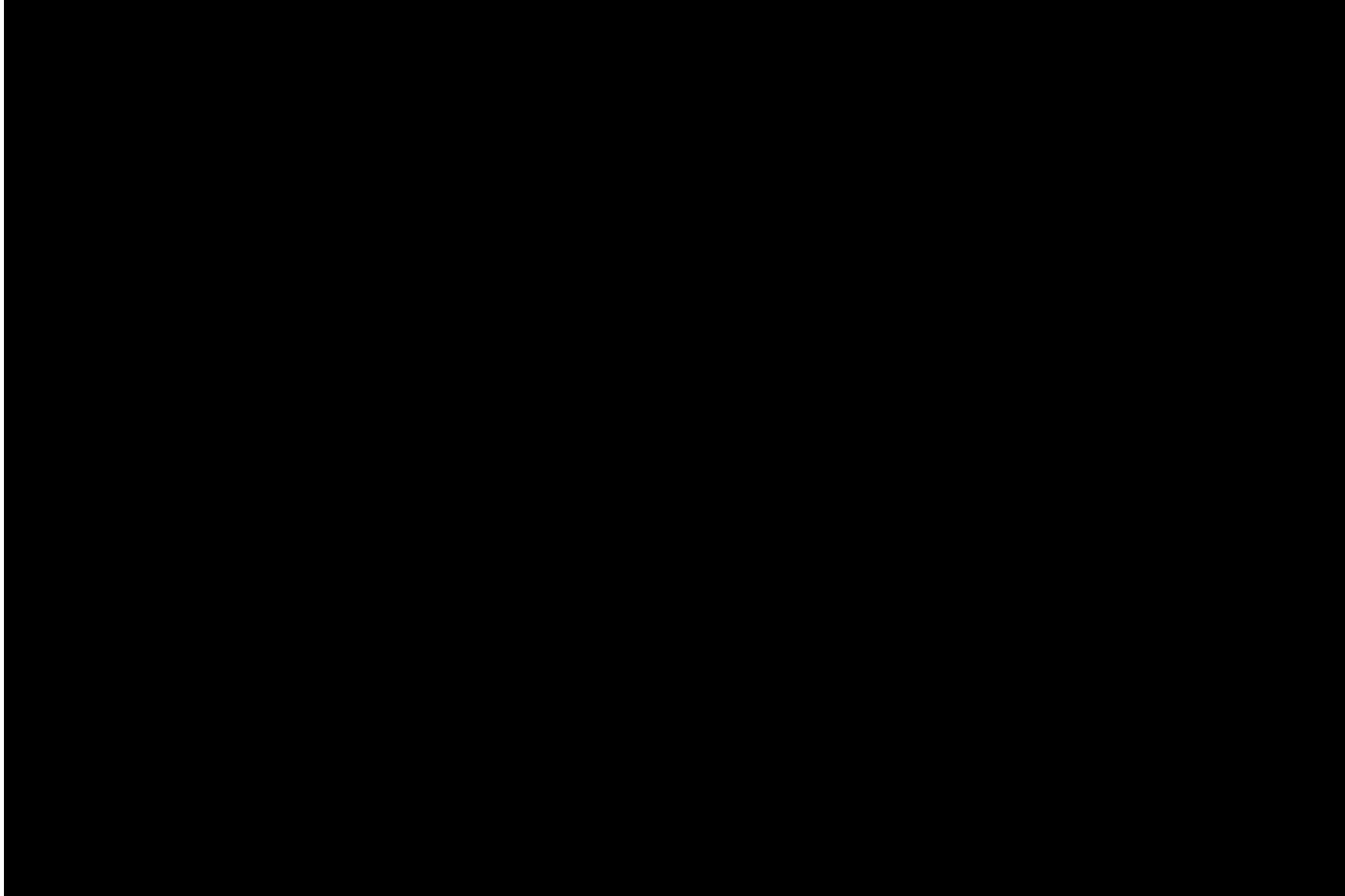
## P1 구동 및 센서구조



## 이미지 정보 수집

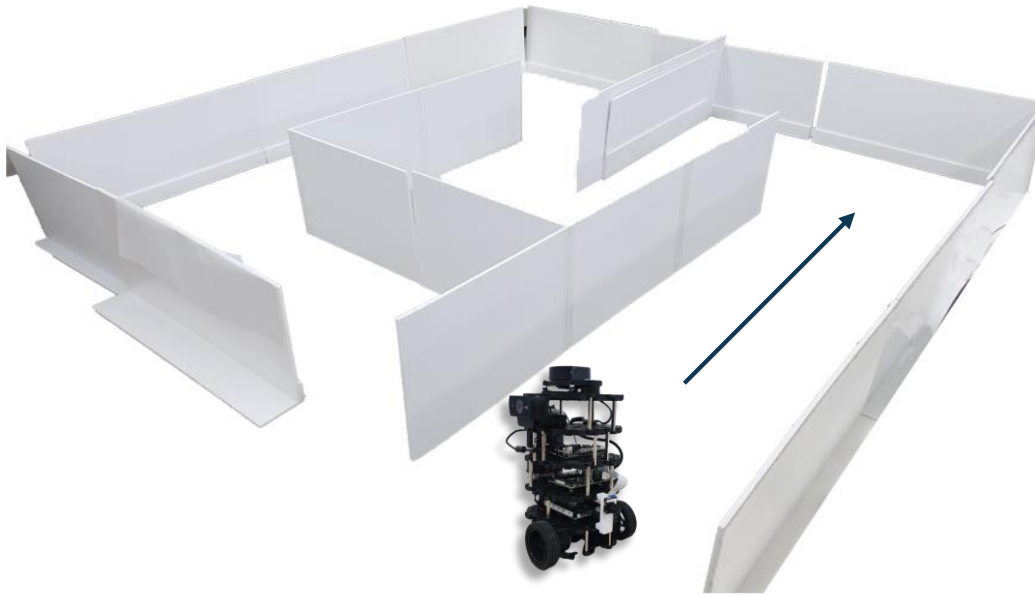


## 04 데이터 수집 영상



## 04 데이터 수집 수집 방법

시연 세트



Distance Total : 11.5 m  
Duration : 2분 38초  
Camera : Logitech C922 webcam  
Image size : 224 x 224 pixel  
Collected : 400~500 장



# 04 데이터 수집 수집 결과

- 학습에 적절하지 않은 이미지 삭제

(a)



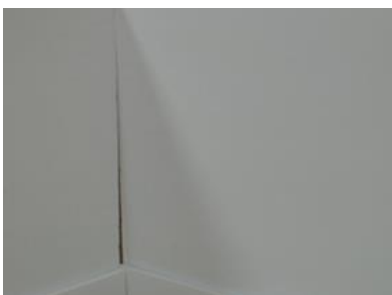
분류하기 애매한 경우

(b)



여러 POI를 포함하고 있는 경우

(c)



POI가 없는 경우

- Data augmentation

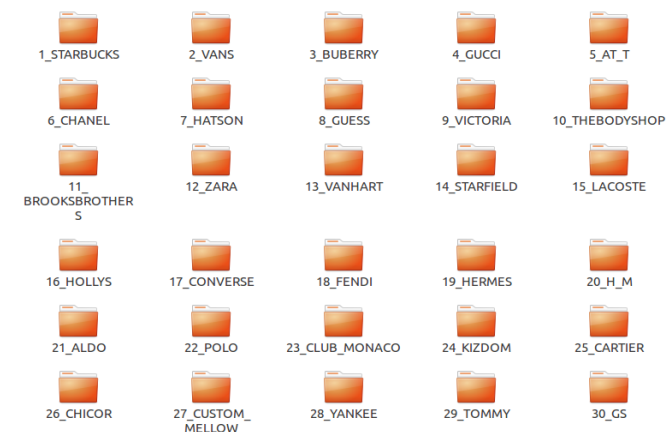


Original image



Augmented image

- 최종 데이터셋



- 총 매장 수 : 30개
- 바뀐 매장 수 : 2개
- 총 데이터셋 : 13,000장



## 05 POI Change Detection

- Naive Approach

Keypoint matching based



실내 공간에서 keypoint 정확도 보장 어려움

➡ POI 변화 검출 실패

Object detection based



간판 검출의 성능에 크게 의존, 간판 오분류 가능성

➡ POI 변화 검출 실패

# 05 POI Change Detection

- Related work

## Distance metric learning

### 1. Extract image signatures

= Points in high-dim space



$d_a \in \mathbb{R}^N$

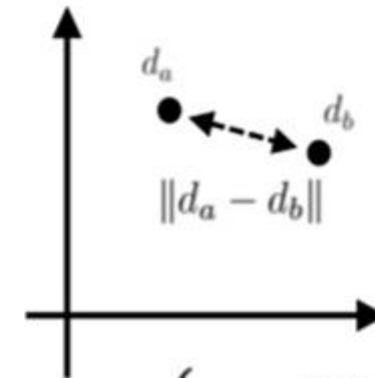


$d_b \in \mathbb{R}^N$



### 2. Decision

= Measuring the distance

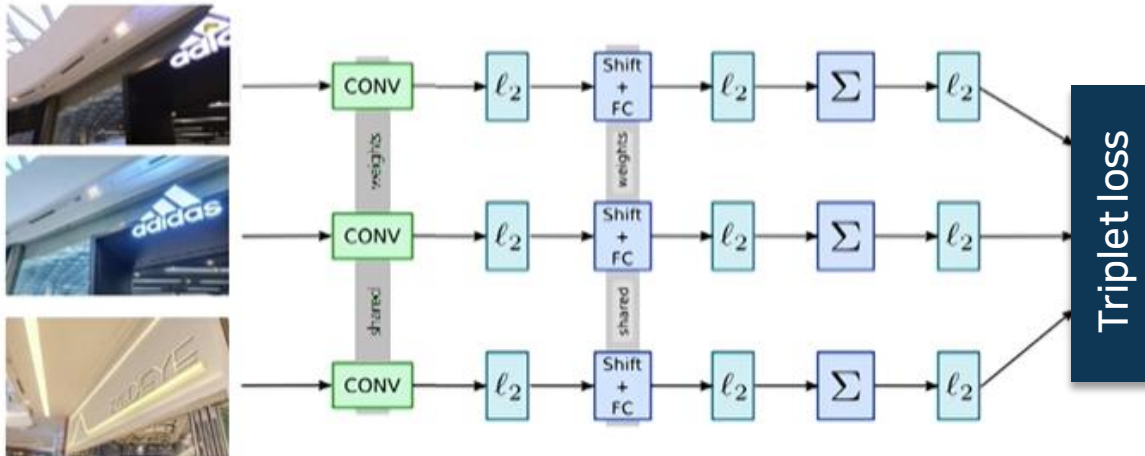


$$\text{is change } (I_a, I_b) = \begin{cases} \text{no} & \text{if } \|d_a - d_b\| \leq \tau, \\ \text{yes} & \text{otherwise.} \end{cases}$$

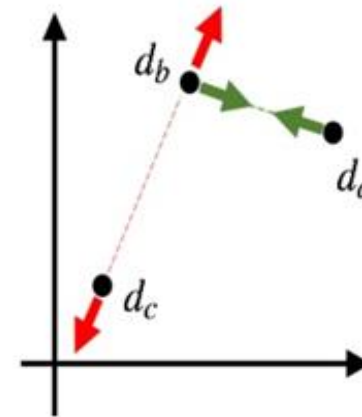
# 05 POI Change Detection

- Related work

## Triplet network



- The loss wants to
  - pull relevant images closer
  - push apart non-relevant images



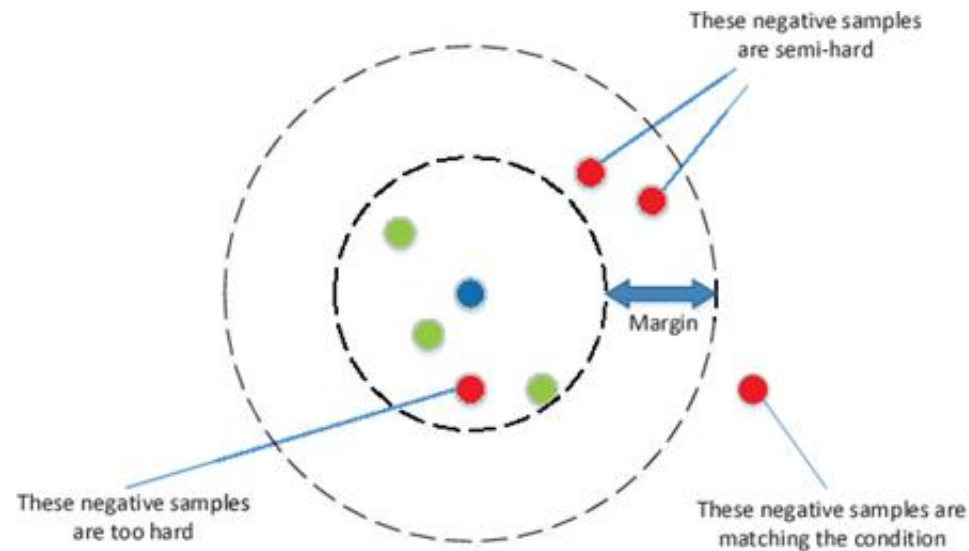
$$Loss = \max \left( 0, \underbrace{d(d_a, d_b) + m}_B - \underbrace{d(d_a, d_c)}_A \right)$$



# 05 POI Change Detection

- Related work

## Triplet sampling



$$D(a, p) < D(a, n) < D(a, p) + m.$$

- Hard Negative Sampling**

Anchor와 같은 클래스 내에서 가장 먼 Positive  
다른 클래스에서 가장 가까운 Negative 선택

- Semi-hard Negative Sampling**

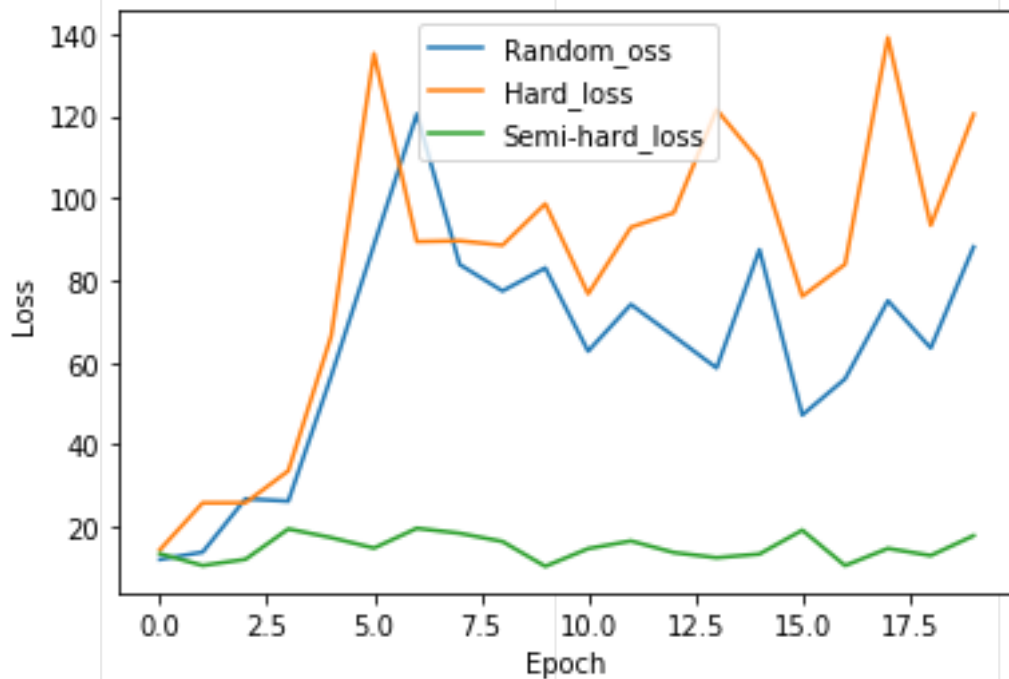
Random과 Hard Negative Sampling 중간

- Random Negative Sampling**

Anchor와 다른 클래스에서 Negative 랜덤하게 선택

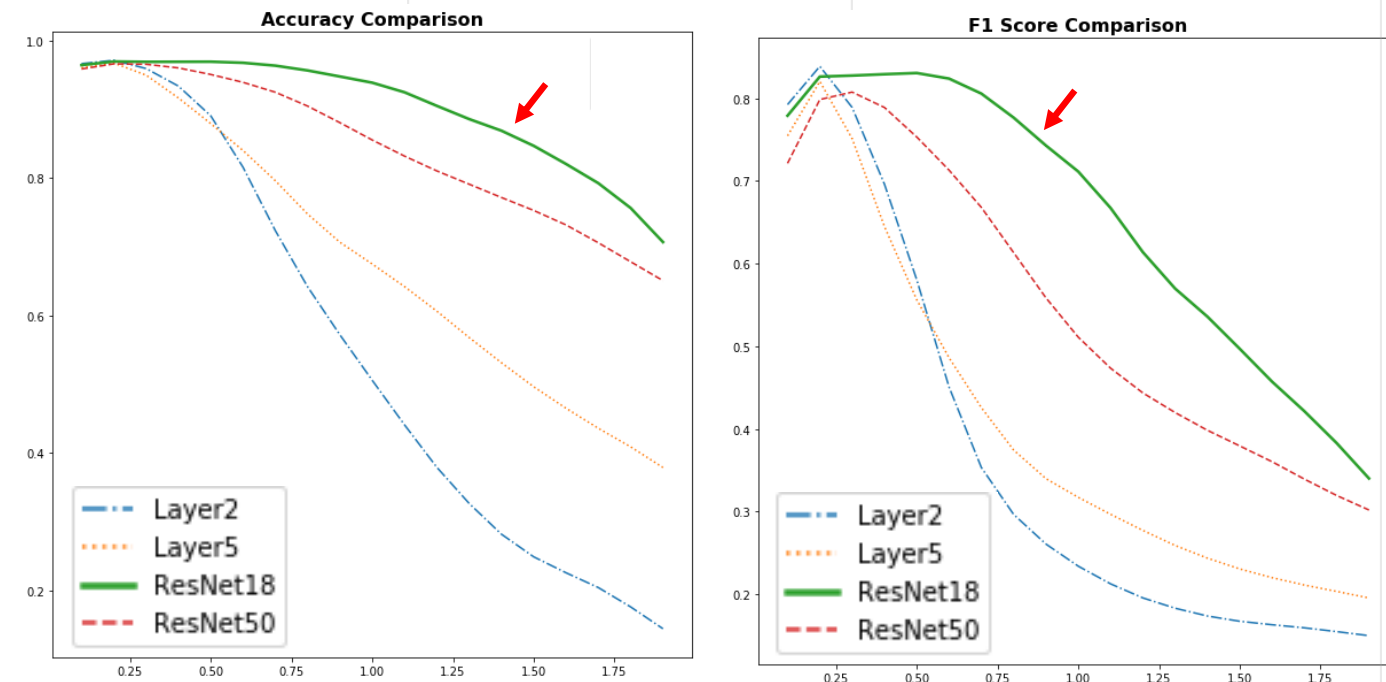
# 05 POI Change Detection

## • Sampling 방법 평가



Semi-hard Negative Sampling 선정

## • 모델 평가 : 최종 선정 모델 ResNet18



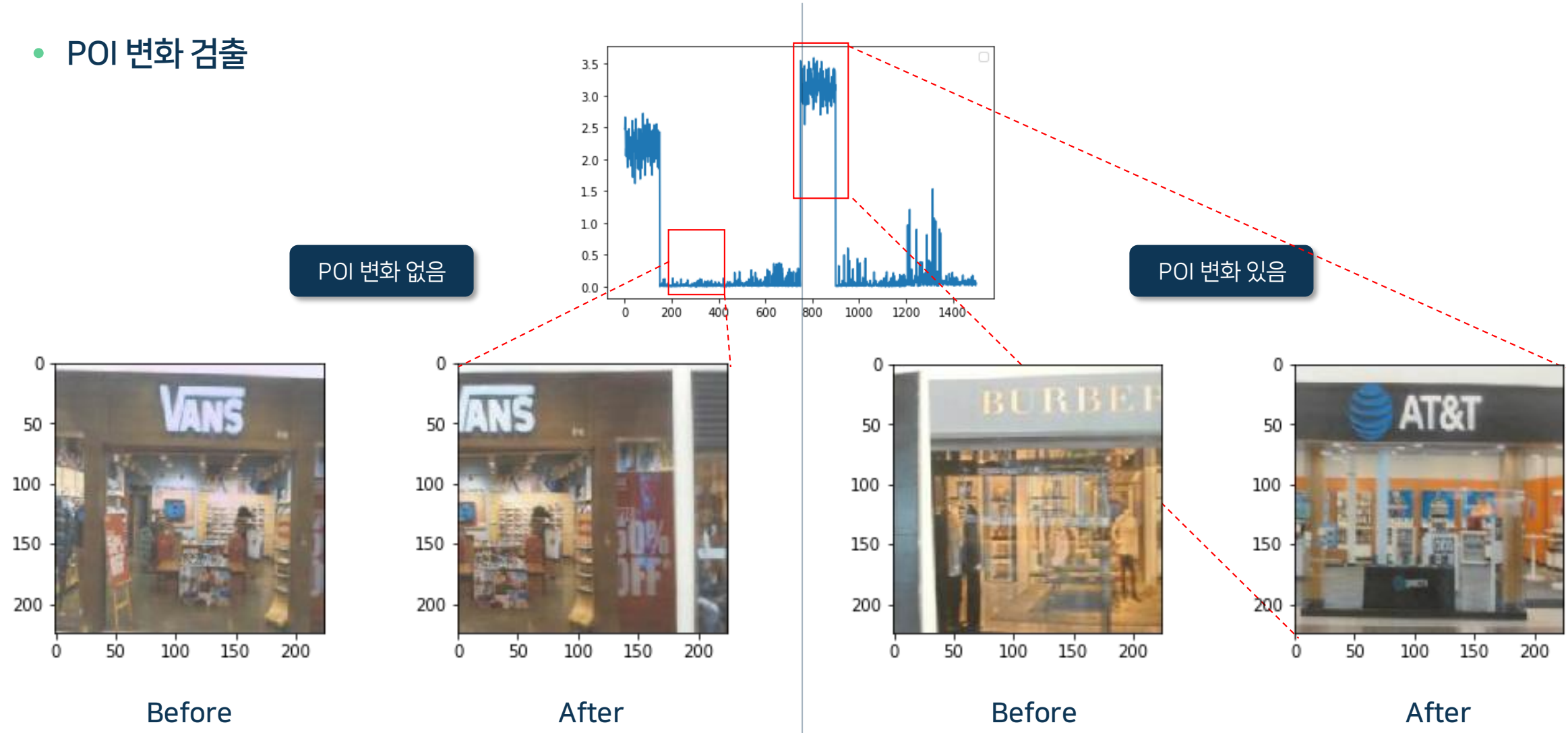
모델평가 기준 Parameters

- learning rate :  $2e-4$
- weight\_decay :  $1e-5$
- n\_epoch : 20

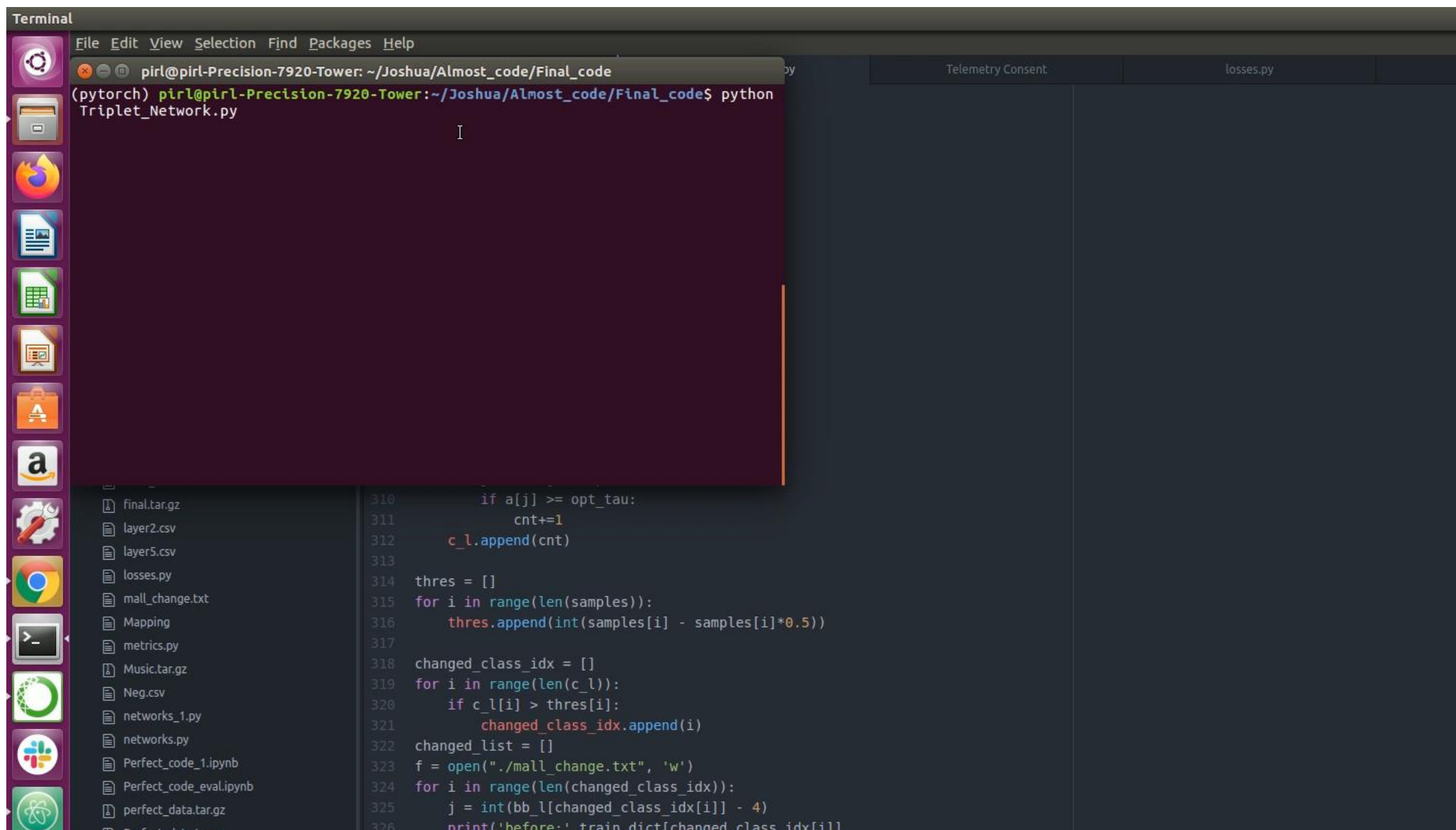
- n\_classes : 10
- n\_samples : 16
- sampling method : SemihardNegativeTripletSelector

# 05 POI Change Detection

- POI 변화 검출



# 05 POI Change Detection

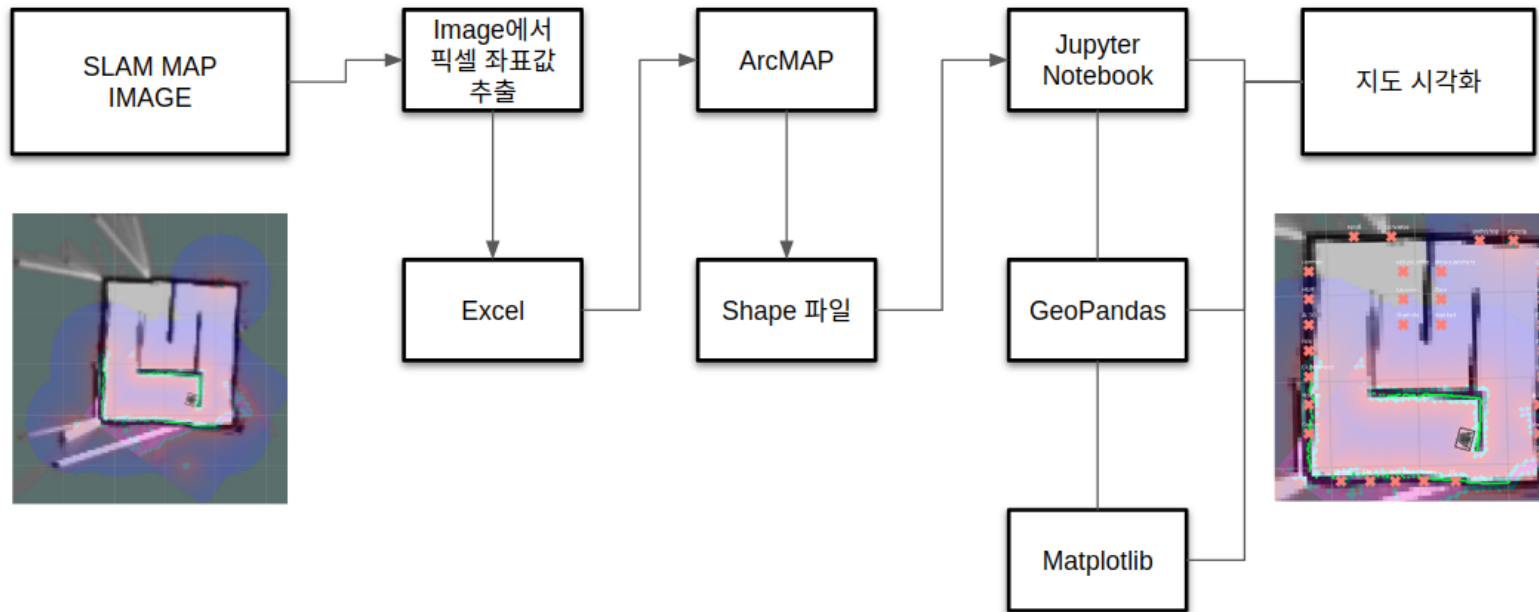


```
Terminal
File Edit View Selection Find Packages Help
pir1@pir1-Precision-7920-Tower: ~/.Joshua/Almost_code/Final_code
(pytorch) pir1@pir1-Precision-7920-Tower: ~/.Joshua/Almost_code/Final_code$ python
Triplet_Network.py

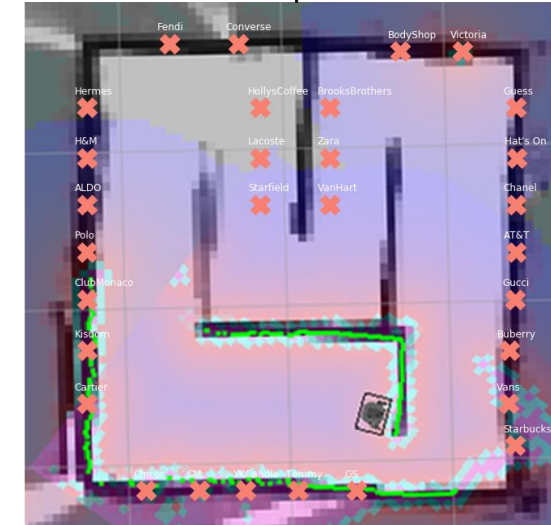
310         if a[j] >= opt_tau:
311             cnt+=1
312             c_l.append(cnt)
313
314     thres = []
315     for i in range(len(samples)):
316         thres.append(int(samples[i] - samples[i]*0.5))
317
318     changed_class_idx = []
319     for i in range(len(c_l)):
320         if c_l[i] > thres[i]:
321             changed_class_idx.append(i)
322     changed_list = []
323     f = open("./mall_change.txt", 'w')
324     for i in range(len(changed_class_idx)):
325         j = int(bb_l[changed_class_idx[i]] - 4)
326         print('before:', train_dict[changed_class_idx[i]]
```

# 06 Mapping

- 맵과 매장정보 결합 방법

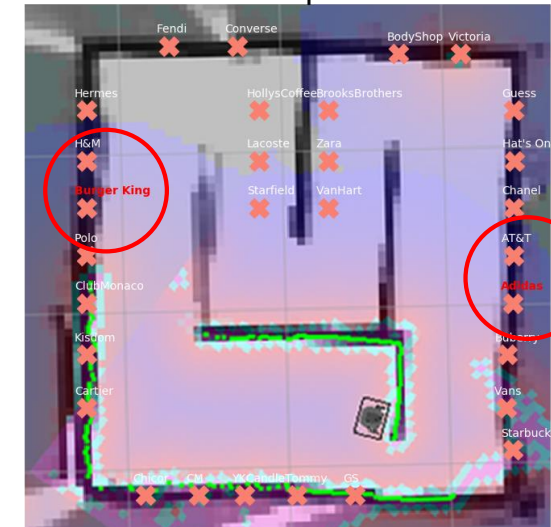


Store Map Before



변화 검출  
지도 업데이트

Store Map After



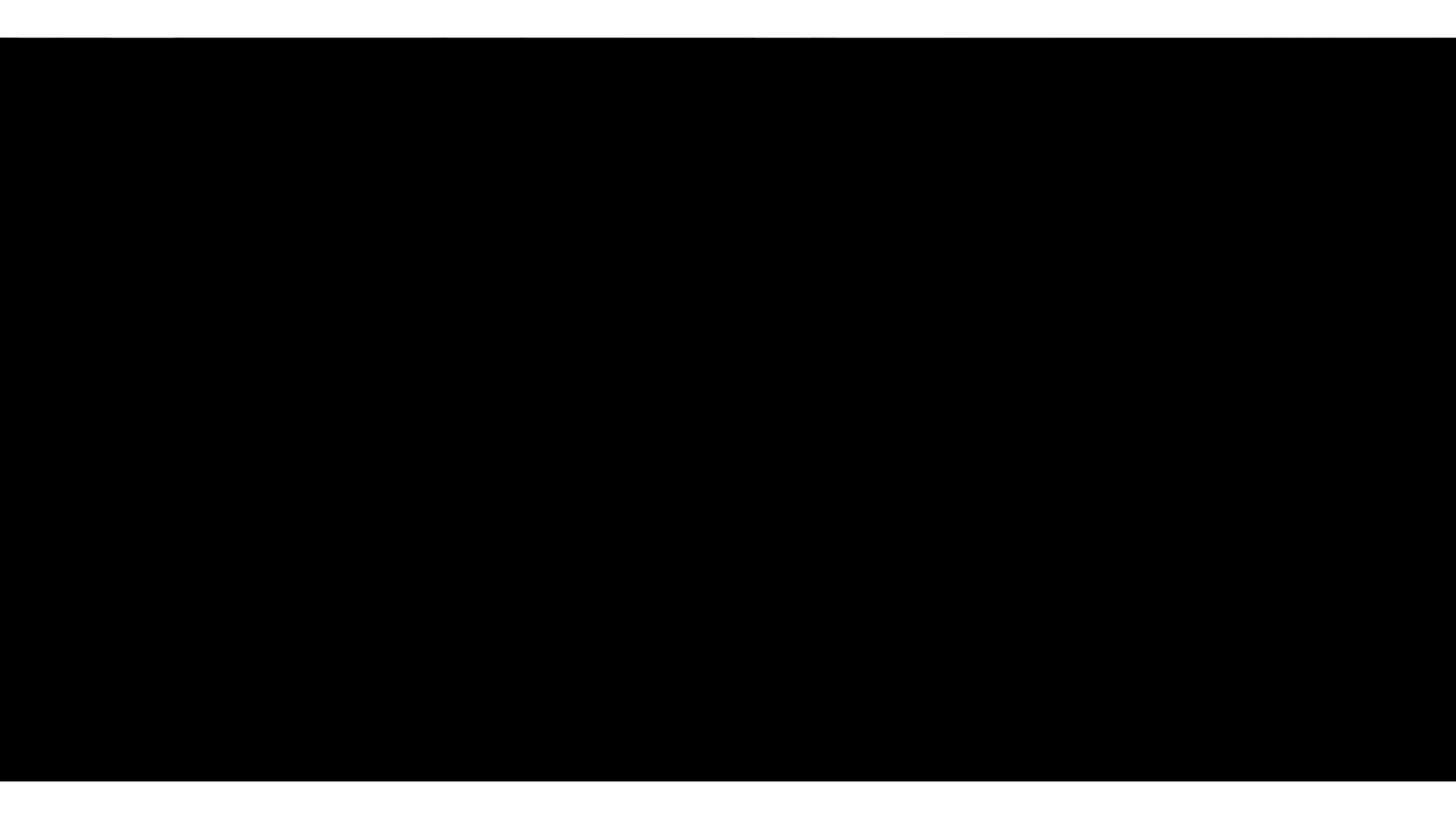
## 07 Conclusions

- 개선점
  - 자율 주행 시 학습 가능한 일관된 이미지 데이터를 수집하는 데 어려움이 있어 키보드 제어 방식 사용함
  - LiDAR와 함께 6 DoF camera를 사용하면 보다 정밀한 공간 정보 mapping 가능함
  - IoU를 활용하여 과거와 현재 두 영상의 pair를 맞추는 방식을 적용하면 환경이 달라져도 적절한 pair 구성 가능함

## 07 Conclusions

- 활용 분야
  - 매장 상호 외에도 세일 정보, 현재 매장 방문고객 수 등을 비롯한 다양한 정보를 실시간으로 반영하여 쇼핑몰 공간 내의 모든 유용한 정보를 제공하는 서비스를 생각해볼 수 있음
  - 실외의 스트리트 뷰 서비스와 같이 실내 공간에서도 AR 인터페이스와 네비게이션을 결합하여 AR을 활용한 길 안내 서비스를 생각해볼 수 있음
  - 변화가 자주 일어나지만 공간 정보를 알기 어려운 터널과 같은 실내 산업 현장에 적용하는 방안을 생각해볼 수 있음





# Q&A

The background is a detailed architectural floor plan of a building, rendered in a light green color. The plan shows various rooms, corridors, and structural elements. Several rooms are labeled with their areas in square meters (m²): 15m², 8m², 24m², 18m², 10m², and 24m². The plan also includes numerous dimension lines with numerical values, indicating the size and layout of the building. The overall aesthetic is technical and precise.

# Thank you