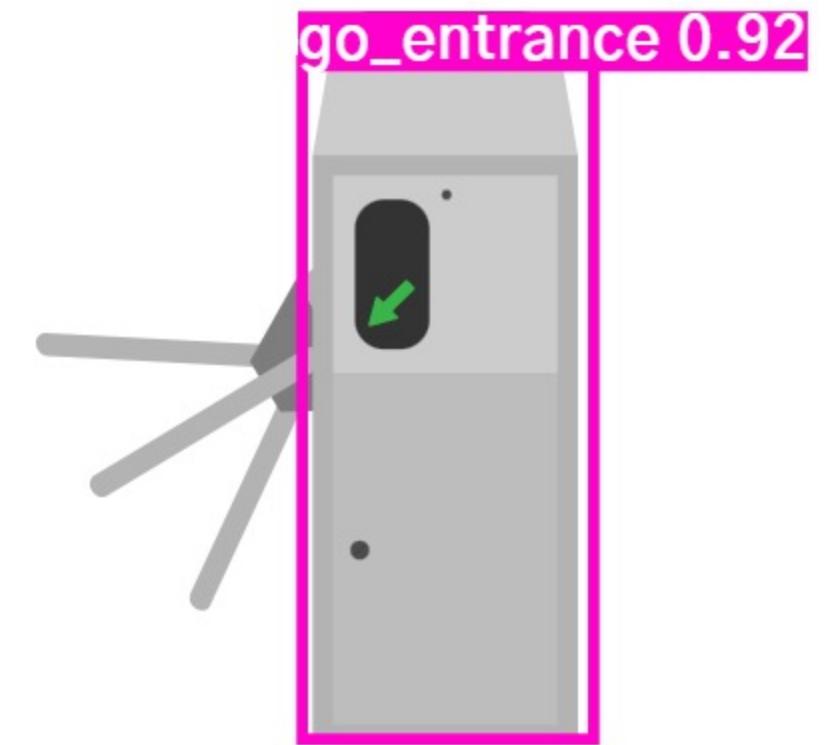


# 시각장애인을 위한 개찰구 인식 모델

Turnstile detection mobile application (TDMA)  
for the Visually Impaired



지하철 타기 쉽(10)조

나영은 박세희 이하현 정나영 최다정

# CONTENTS

## Part 01

### Motivation

1. Motivation
2. Problems
3. Related works & Limitations
4. Major Challenges

## Part 02

### Datasets & Model design

1. Architectural Design
2. Collecting Datasets & Data Labelling
3. Why YOLOv8m?

## Part 03

### Results

1. TDMA v1
2. Improvements
3. TDMA v2
4. Result Video

# PART 1. Motivation

Motivation &  
Problems

Related works &  
Limitations

Major Challenges



# Motivation

시각장애인 대중교통 이용 '불편'... "이용편의 개선 선행돼야"

▲ 박성용 기자 | ◎ 입력 2022.07.27 14:48 | □ 댓글 0



장애인 가장 많이 겪는 차별은 이동·대중교통…첫 실태조사

송고시간 | 2023-02-24 12:44

中文

HOME > 서울 & > 서울시의회 > 도봉

목숨 걸고 지하철 타는 시각장애인

▲ 서울로컬뉴스 | ◎ 승인 2018.11.10 13:14 | □ 댓글 0

[기획] '장애인 이동권', 지하철 '투쟁' 의미는?

▲ 김범규 기자 | ◎ 입력 2023.01.13



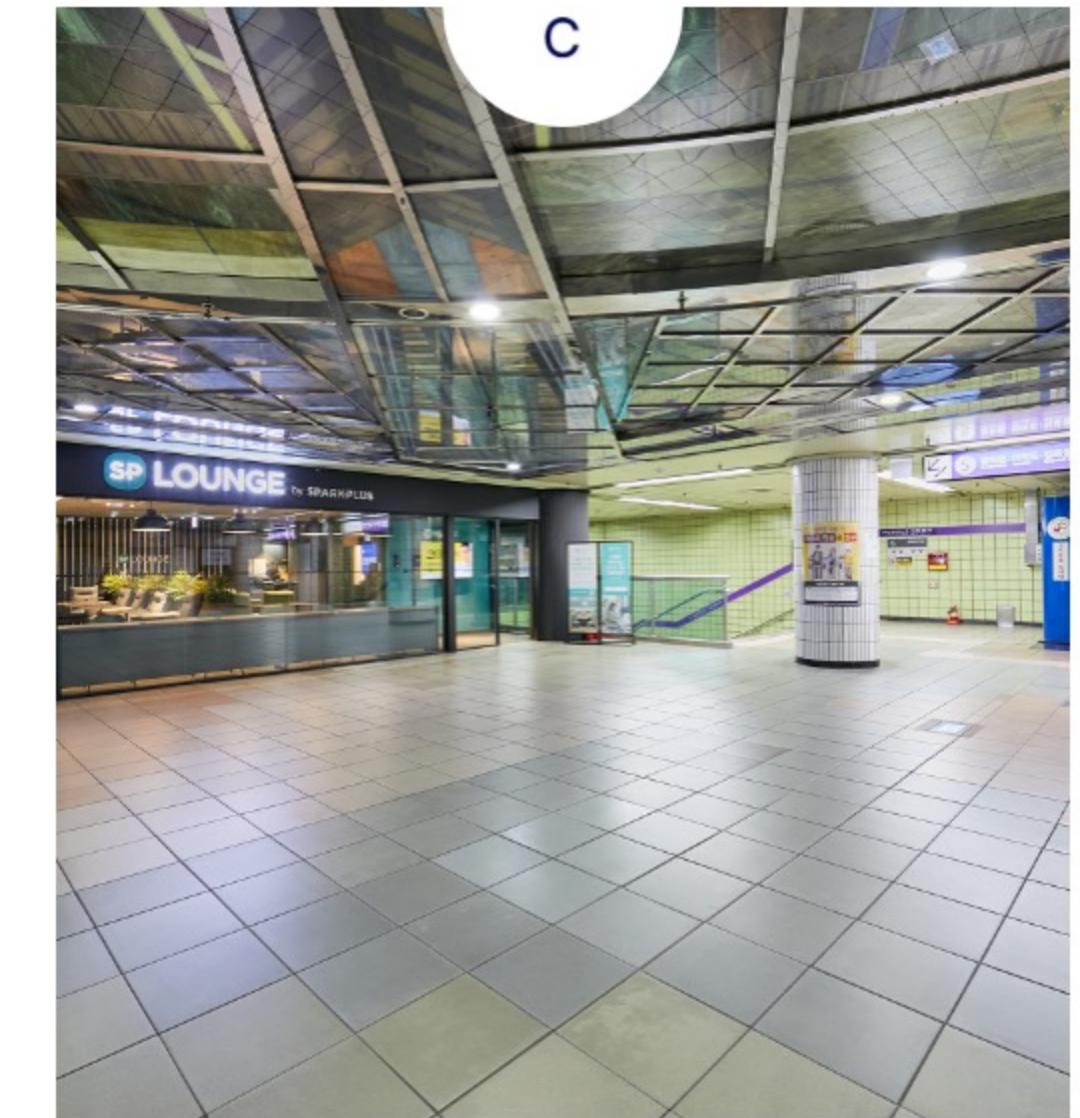
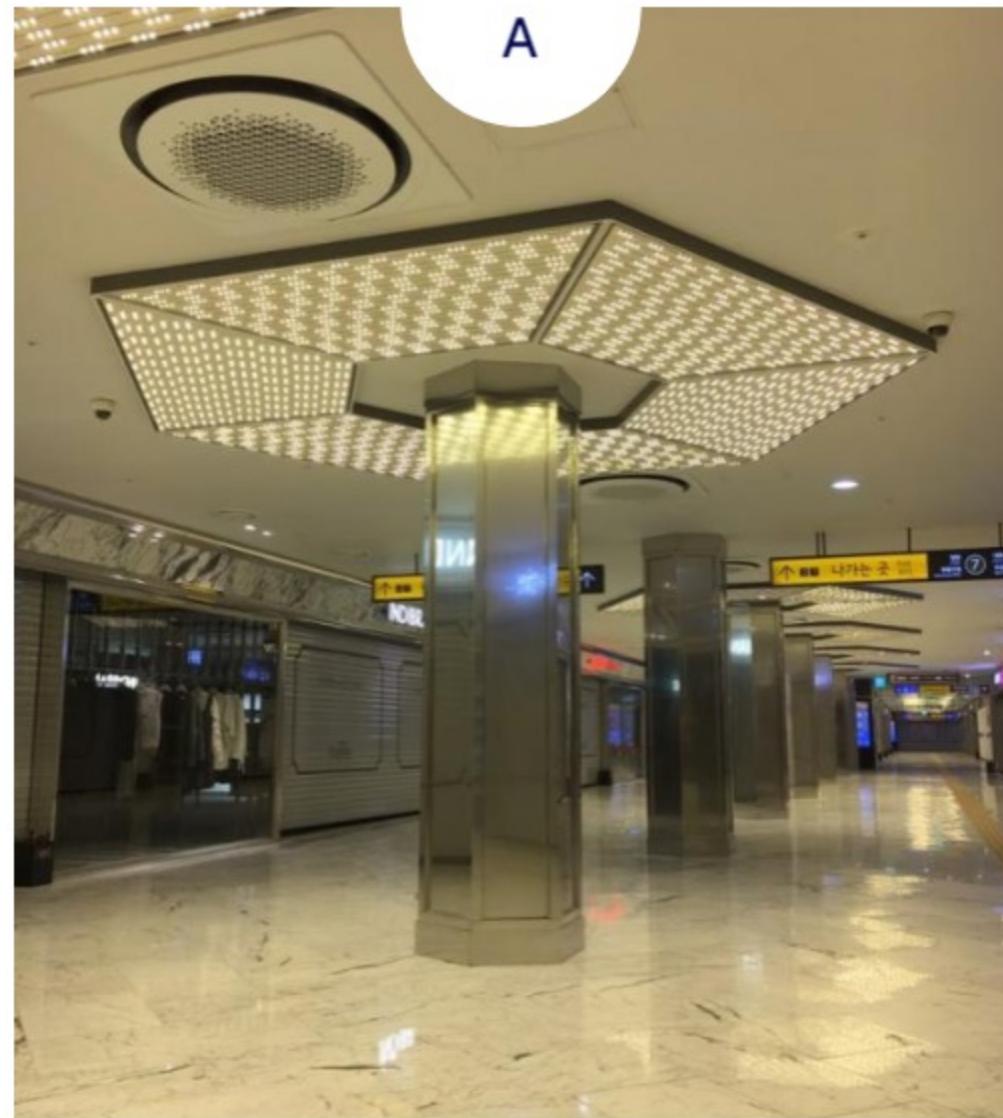
# Problem 1. Braille Blocks 점자 블럭

현재 지하철 내 대다수의 점자블럭은 **지속적인 관리의 부재**로 제대로 된 기능을 하지 못하고 있음



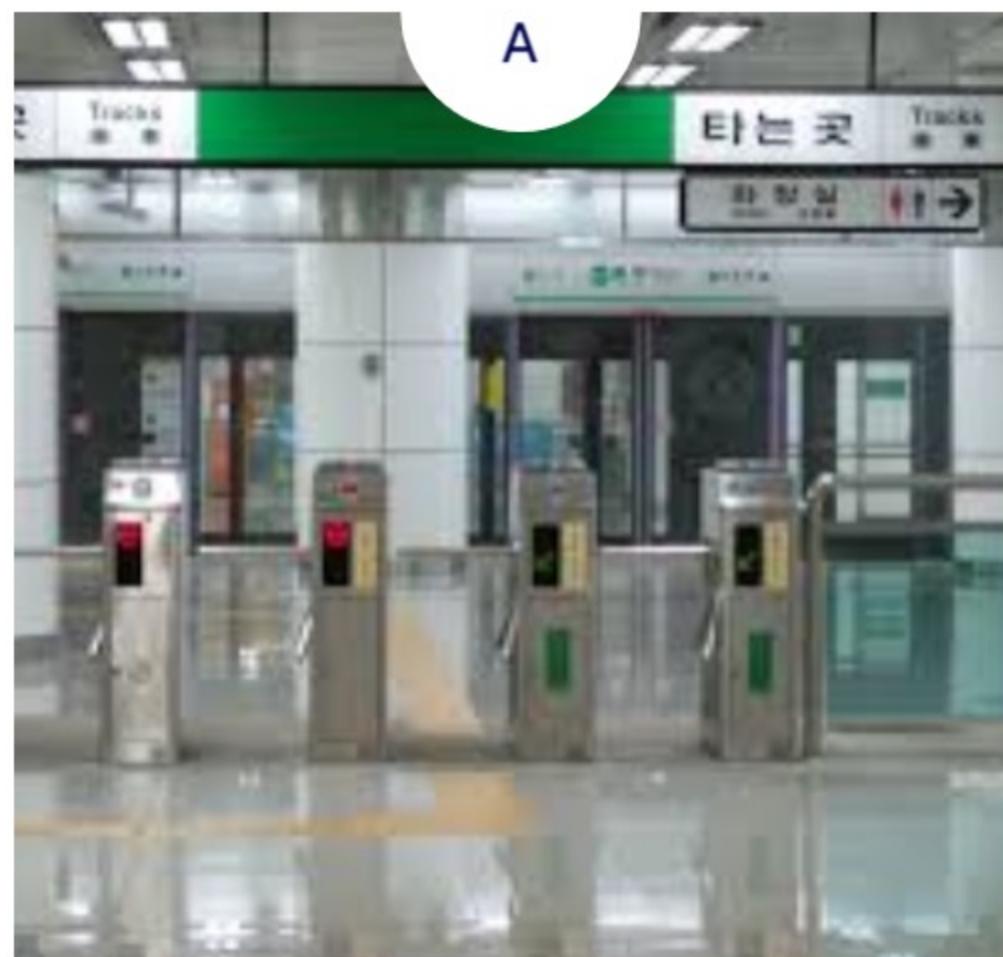
## Problem 2. Turnstile 개찰구

복잡하고 넓은 역 내에서 개찰구의 위치를 찾는 것은 시각장애인에게 매우 어려운 일



## Problem 2. Turnstile 개찰구

역마다 다른 종류의 개찰구 가운데 진입 가능한 개찰구를 구분하는 것은 안내 지팡이와 점자블럭만으로는 역부족



좌우로 출/입구가 나뉘어진 경우



반응형 : 입구만 표시된 경우



출/입구가 교대로 섞여있는 경우

# Related Works & Limitations

## Related Works

Unions of RFID and Braille Block

Detection camera detached on  
hat, glasses, etc.

Mobile app for detect bus,  
traffic lights, etc. (ex: 버스스로)

## Limitations

Depends on performance of  
the central server and braille block

Not as usable in real life as  
mobile services

Not optimized for the feature we need  
(subway turnstile detection)

We need Turnstile detection mobile application (TDMA) for Visually Impaired !

# Major Challenges

## High Accuracy

Need to accurately detect different types of turnstiles



## Real - time detection & Feedback

Need real-time route voice guidance for the visually impaired



## Optimized for Mobile Application

Must be light enough for mobile environments



## PART 2. Datasets & Model design

Architectural Design

Collecting Datasets  
& Labelling

Why YOLOv8m?



# Architectural Design

역내 진입 후 앱 실행

멀리 있는 개찰구 감지

“개찰구를 찾았습니다”

1단계 음성 안내

2단계 진입

진입 가능한 개찰구 감지

“직진하세요”, “좌/우로 이동하세요”

2단계 음성 안내

“Beep-”

실시간 피드백

개찰구 통과!

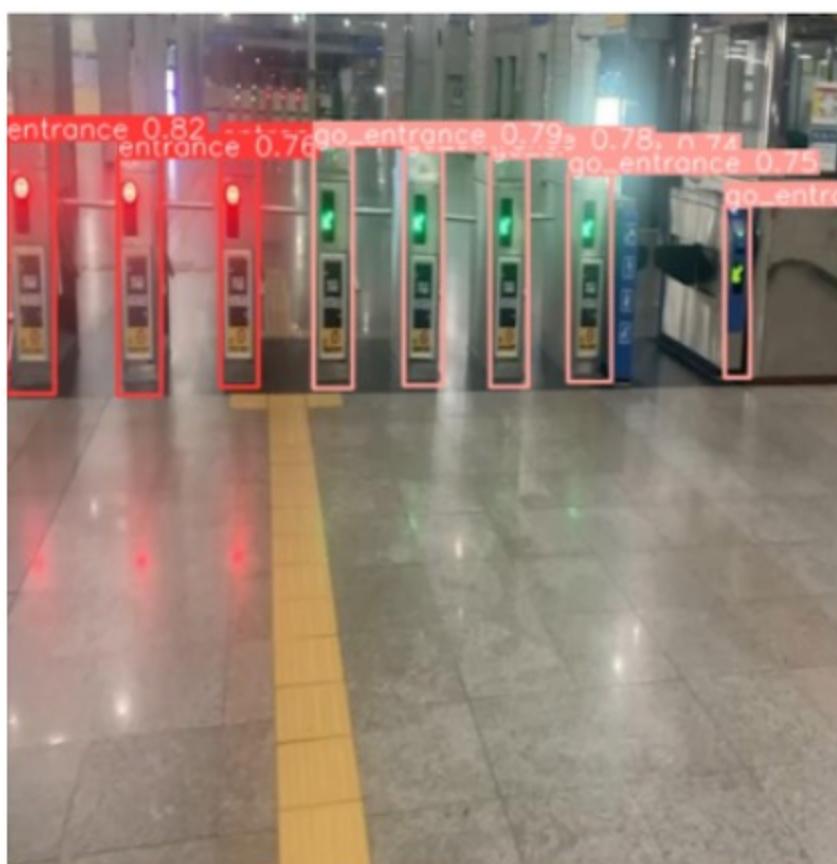
# Architectural Design

“개찰구를 찾았습니다”



1단계 : 개찰구 위치 감지

“직진하세요”, “좌/우로 이동하세요”



2단계 : 진입가능한 개찰구 감지

“Beep-”



통과하기까지 피드백 제공

# Collecting Datasets

2024.02.03~02.17

2 weeks



Line 1,2,3,7,9, 수인분당, 신분당

16 stations



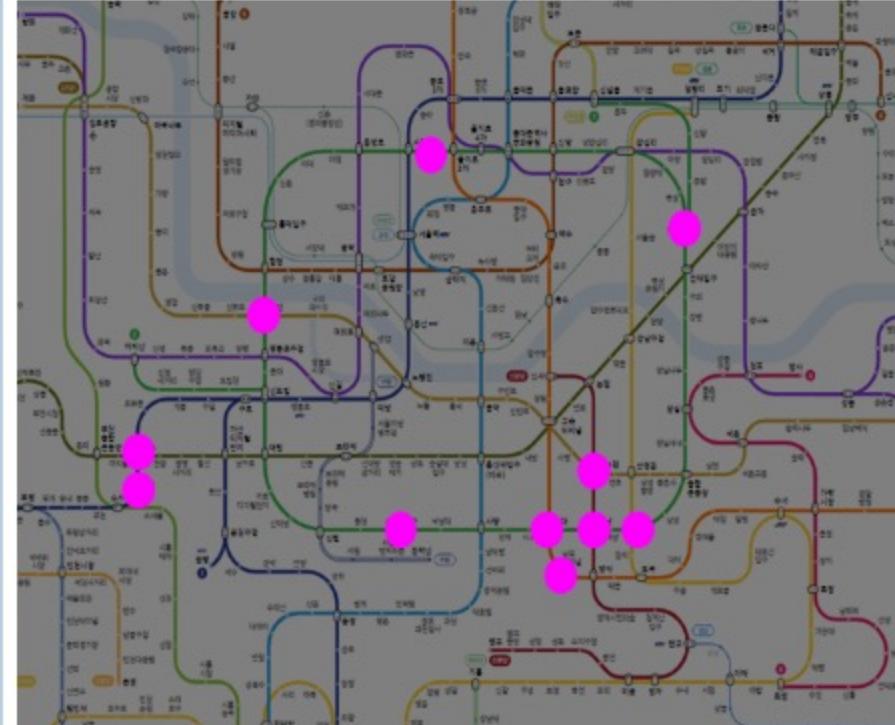
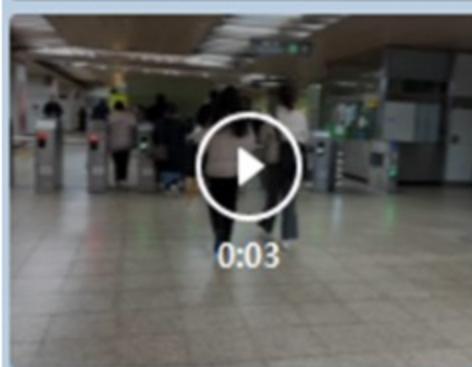
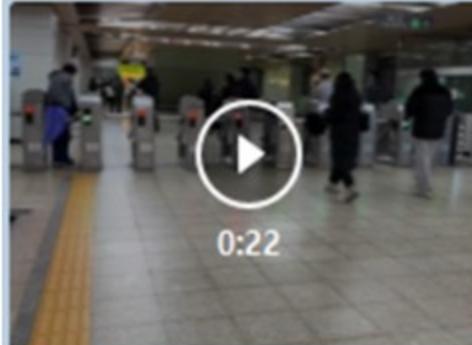
Classify special cases

9 types



35 videos, 4-5frame/sec, augmentation

3987 images



Simulations

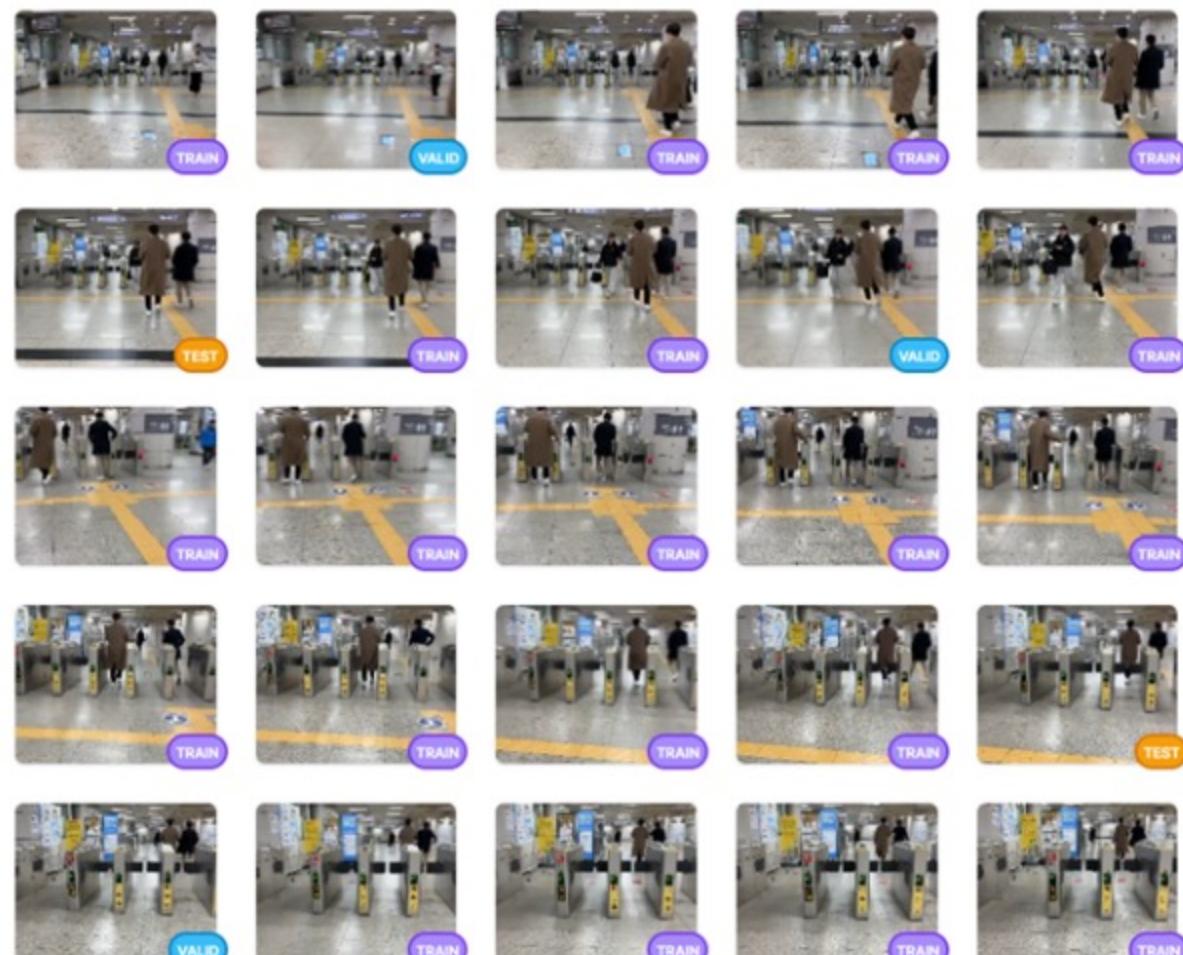
Shooting Videos

Choose Stations

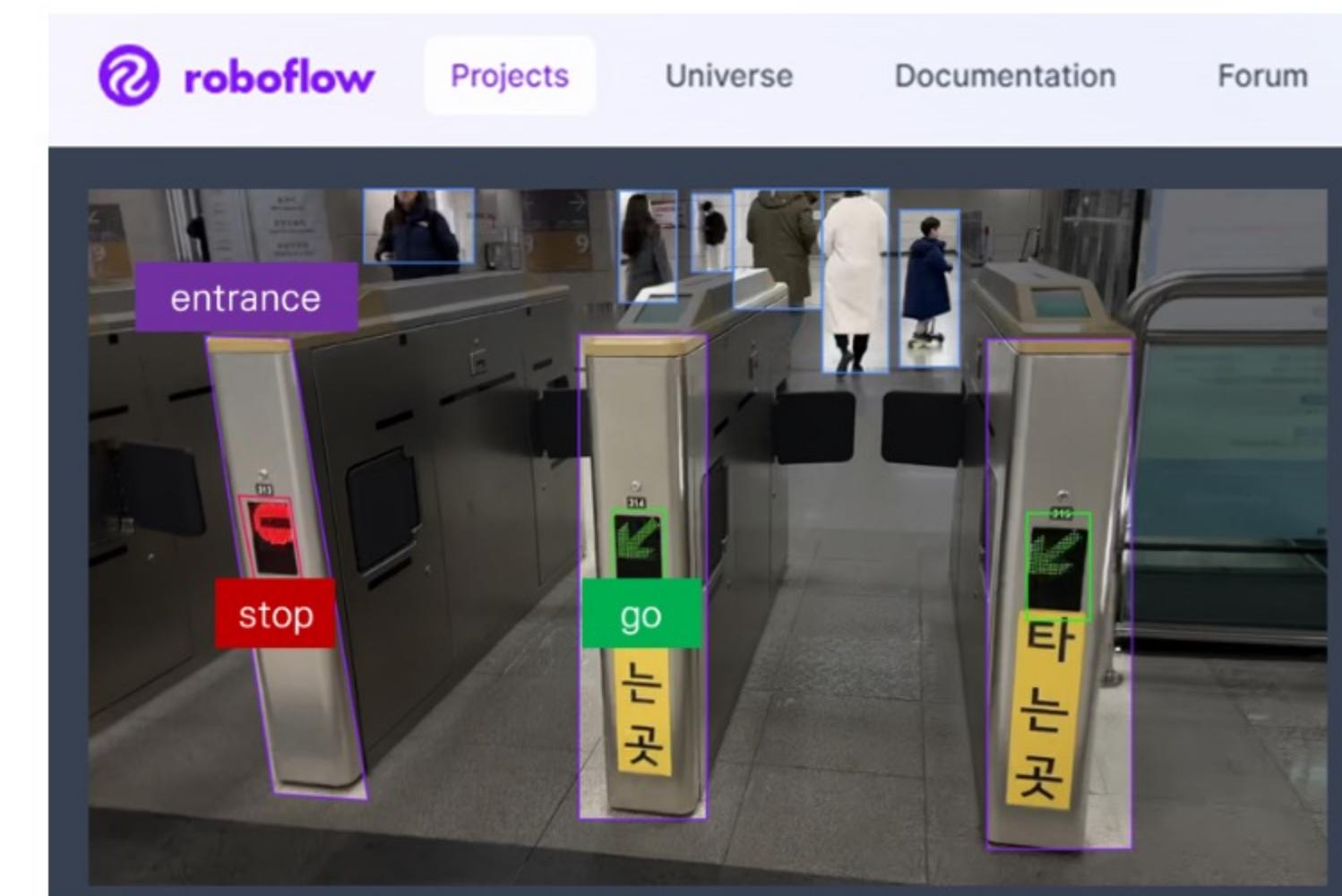
Type	Stop	혼잡도	점자 블록	반응형 개찰구	개찰구 크기	출입구 교차형	화면 방향	측면 촬영
A	X	한적	X	X	M	X	가로	O
B	O	한적	X	O	M	X	세로	X
C	O	보통	O	X	S	X	가로	X
D	O	보통	O	X	L	O	가로	X
E	O	보통	O	X	L	O	가로	O
F	O	혼잡	O	X	S	X	세로	O
G	O	보통	O	X	M	X	세로	O
H	X	혼잡	O	O	S	X	세로	O
I	O	혼잡	O	X	S	X	가로	X

Classify Types

# Data Labelling (using Roboflow)

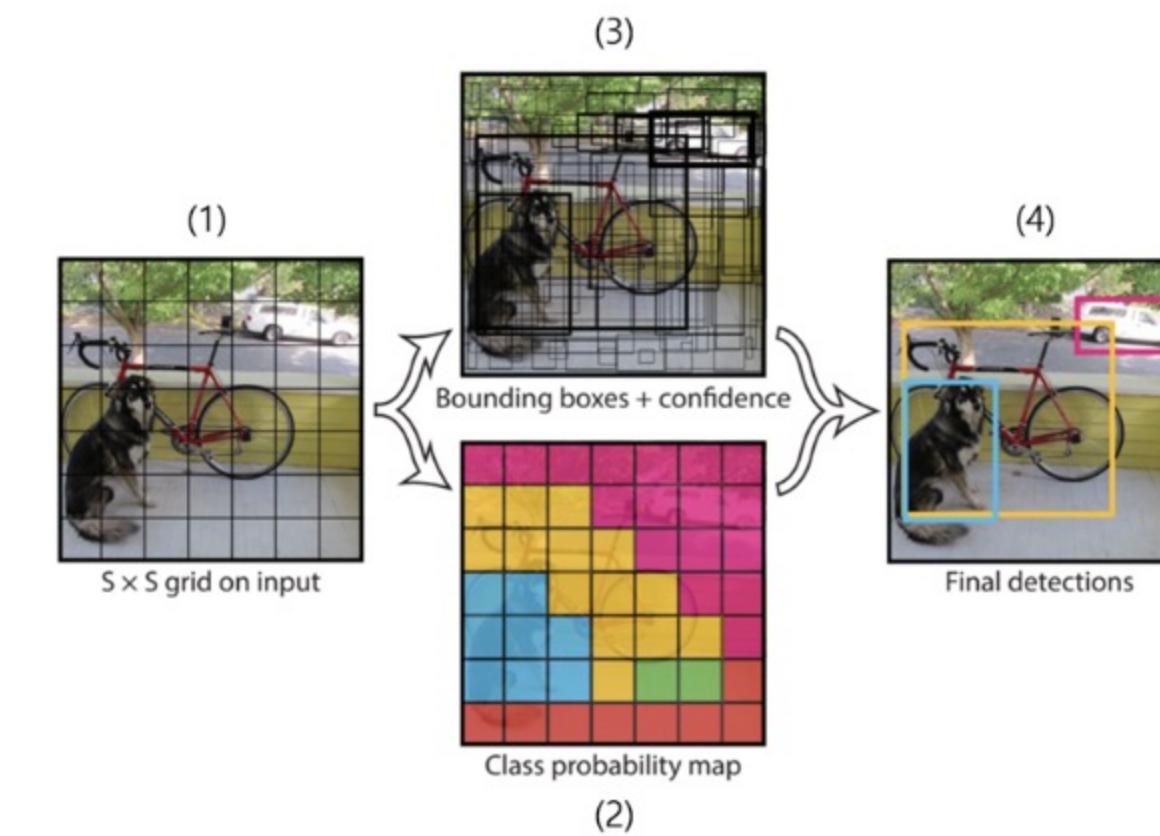
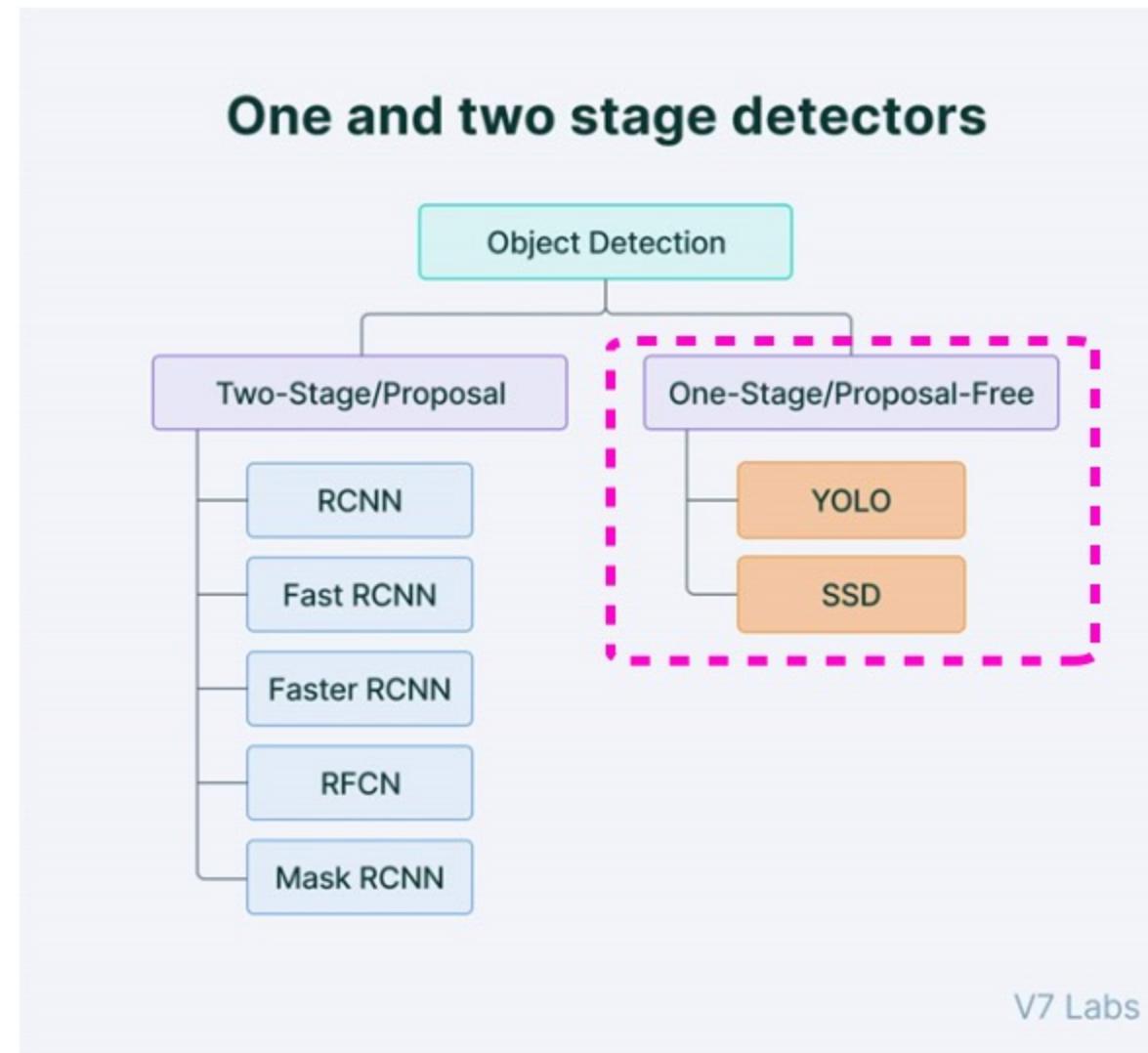


Dataset Split  
(Train 89%, Valid 7%, Test 4%)



Annotate Data

# Why YOLOv8m?

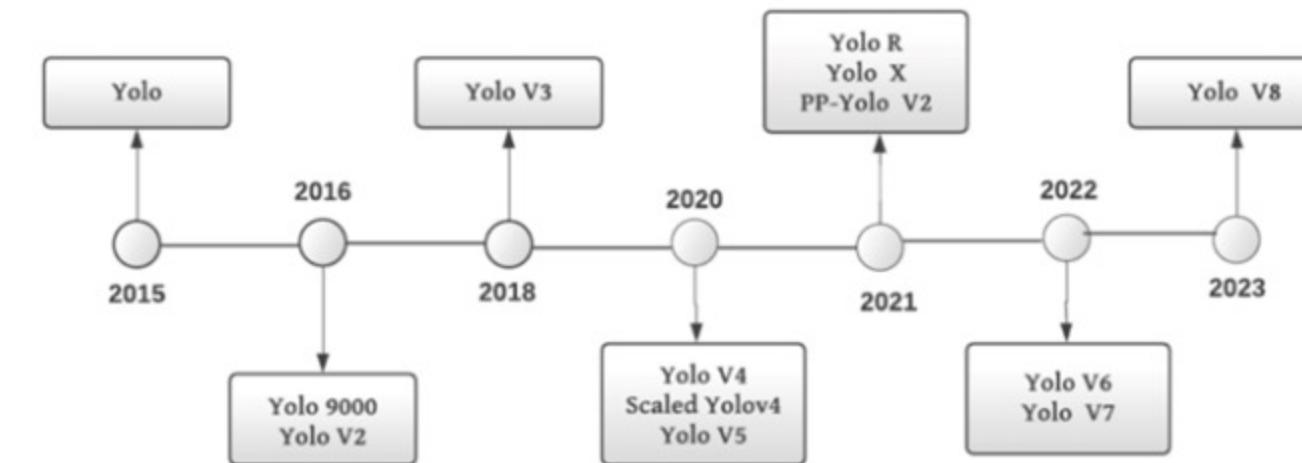


- faster than two-stage detector → real-time detection
- grid object detect → SSD 보다 accuracy 는 떨어지지만 속도가 빠르고 가벼움
- mobile 환경에 적합

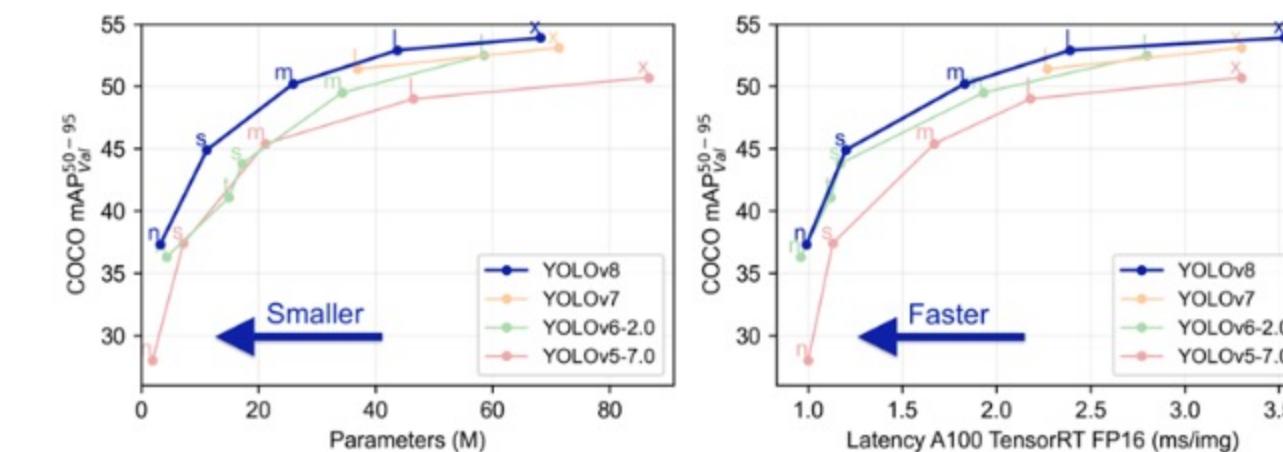
# Why YOLOv8m?

Model	size (pixels)	mAP <sup>val</sup> 50-95	Speed CPU ONNX (ms)	Speed A100 TensorRT (ms)	params (M)	FLOPs (B)
YOLOv8n	640	37.3	80.4	0.99	3.2	8.7
YOLOv8s	640	44.9	128.4	1.20	11.2	28.6
YOLOv8m	640	50.2	234.7	1.83	25.9	78.9
YOLOv8l	640	52.9	375.2	2.39	43.7	165.2
YOLOv8x	640	53.9	479.1	3.53	68.2	257.8

- mAP<sup>val</sup> values are for single-model single-scale on COCO val2017 dataset.  
Reproduce by `yolo val detect data=coco.yaml device=0`
- Speed averaged over COCO val images using an Amazon EC2 P4d instance.  
Reproduce by `yolo val detect data=coco128.yaml batch=1 device=0/cpu`



In comparison of the latest YOLO versions – YOLOv8 vs. YOLOv7 and YOLOv6 – the latest release (YOLOv8) shows the best performance in real-time benchmarks published by the creator.



Ambient AI  
2024 Winter

## PART 3. Results

TDMA V1

Improvements

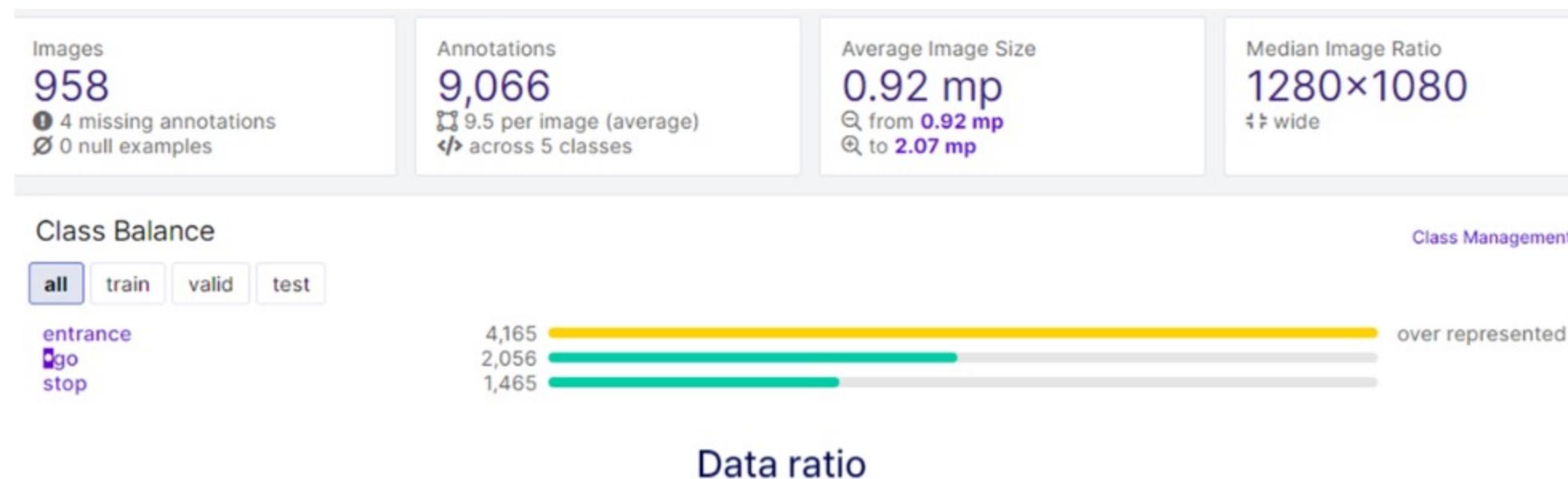
TDMA V2

Future Works



# TDMA v1

Data Quantity	Labelling Class	Data ratio	User scenario	Etc.
958 images	1. Entrance 2. Go 3. Stop	Entrance : go : stop = 4:2:1 <b>(lack of go&amp; stop data)</b>	Simple	- Lack of special case

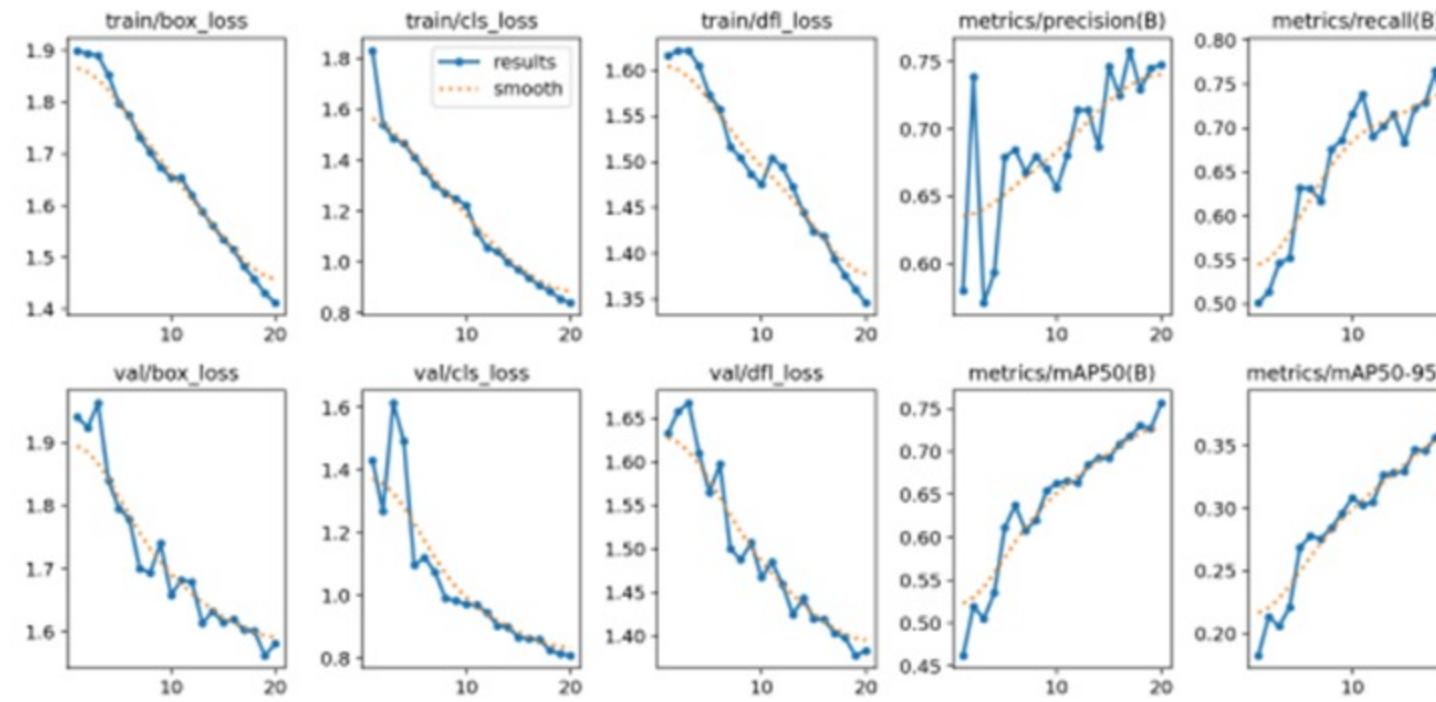


Data ratio

3 classes labelling

# TDMA v1 : Results

Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
<b>Entrance</b>	174	705	0.891	0.888	0.885	0.528
<b>Go</b>	174	342	0.865	0.851	0.867	0.362
<b>Stop</b>	174	256	0.778	0.738	0.769	0.367



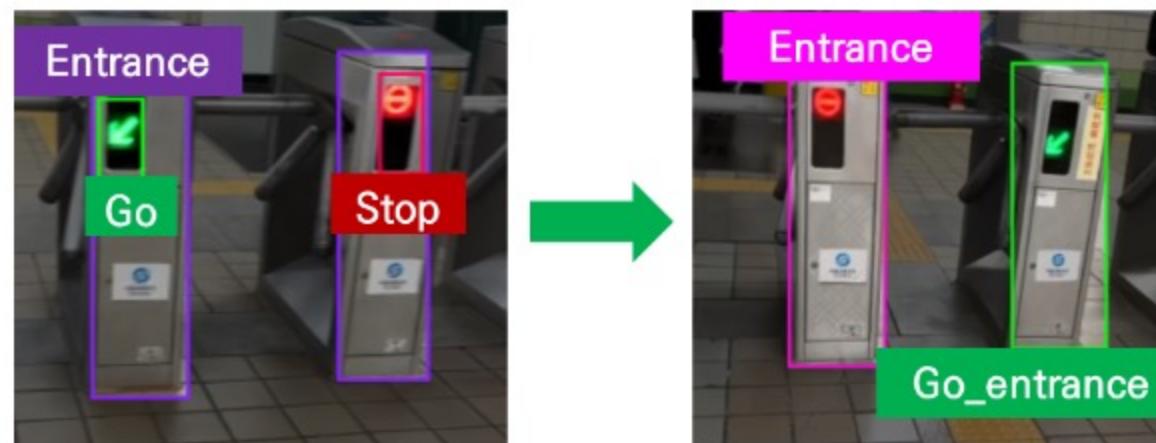
Failed to detect stop



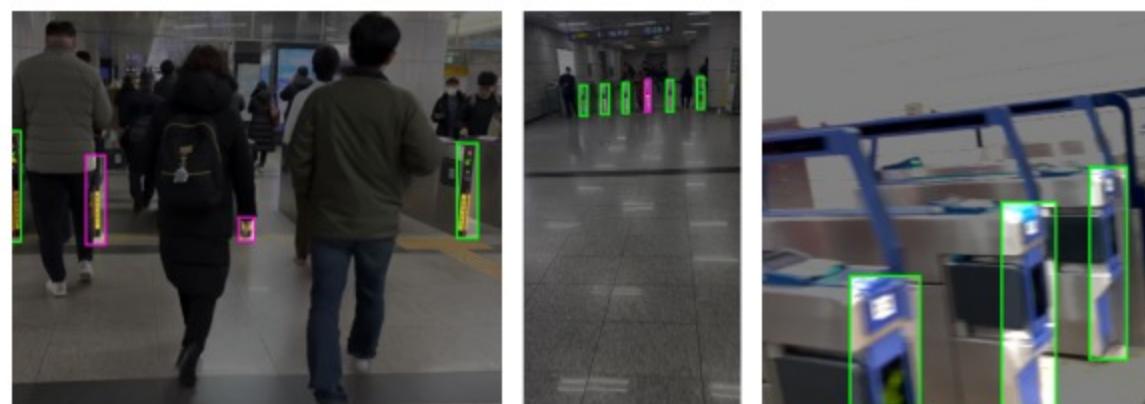
Failed to detect Side of entrance

# Improvements

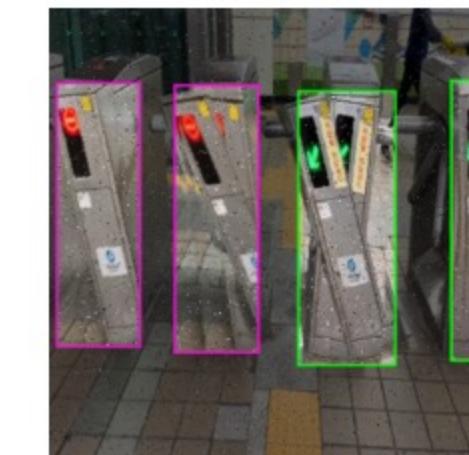
## 1. Change Labelling Class (clearer features)



## 2. Add datasets of special case



## 3. Data Augmentation



Outputs per training example: 3

Noise: Up to 10% of pixels

Bounding Box: Rotation: Between  $-20^\circ$  and  $+20^\circ$

Bounding Box: Shear:  $\pm 19^\circ$  Horizontal,  $\pm 18^\circ$  Vertical

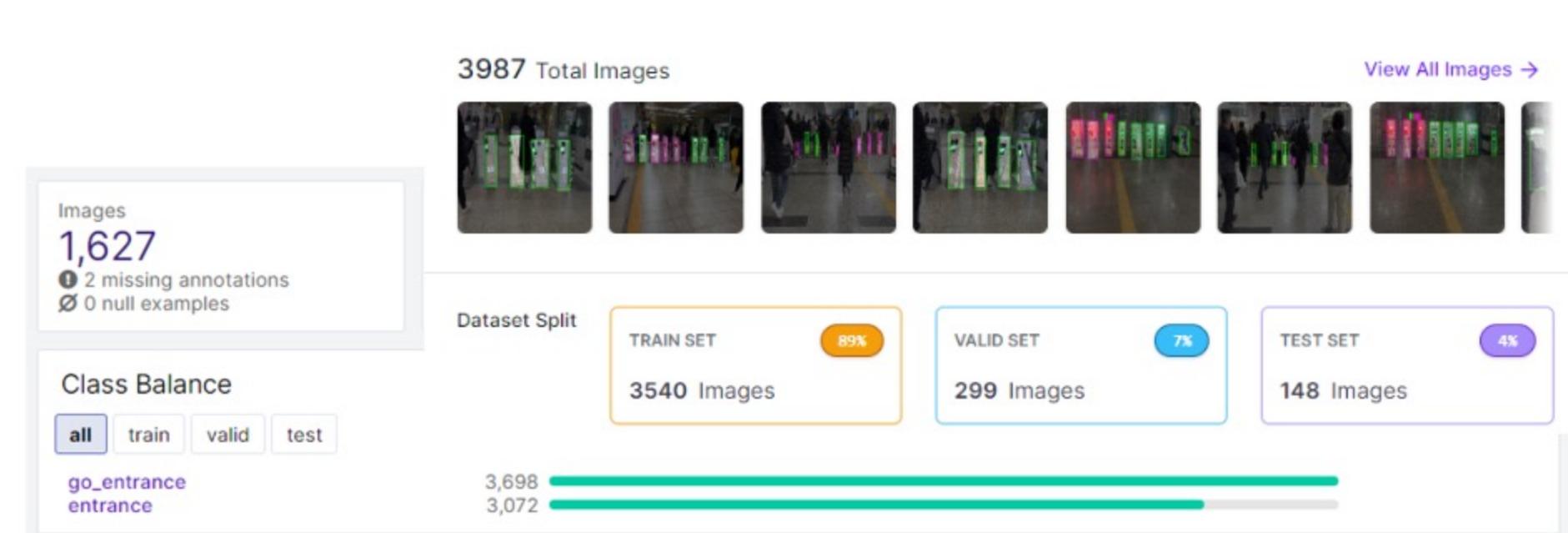
Bounding Box: Blur: Up to 10px

## 4. Develop User Scenario

## 5. Change Parameters (epoch, frame, ...)

# TDMA v2

Data Quantity	Labelling Class	Data ratio	User Scenario
1627 images & Augmentation -> 3987 images	1. Entrance 2. Go entrance	Entrance : Go entrance = 1:1	More detailed and complex



Data ratio



2 classes labelling

## TDMA v2 : User Scenario

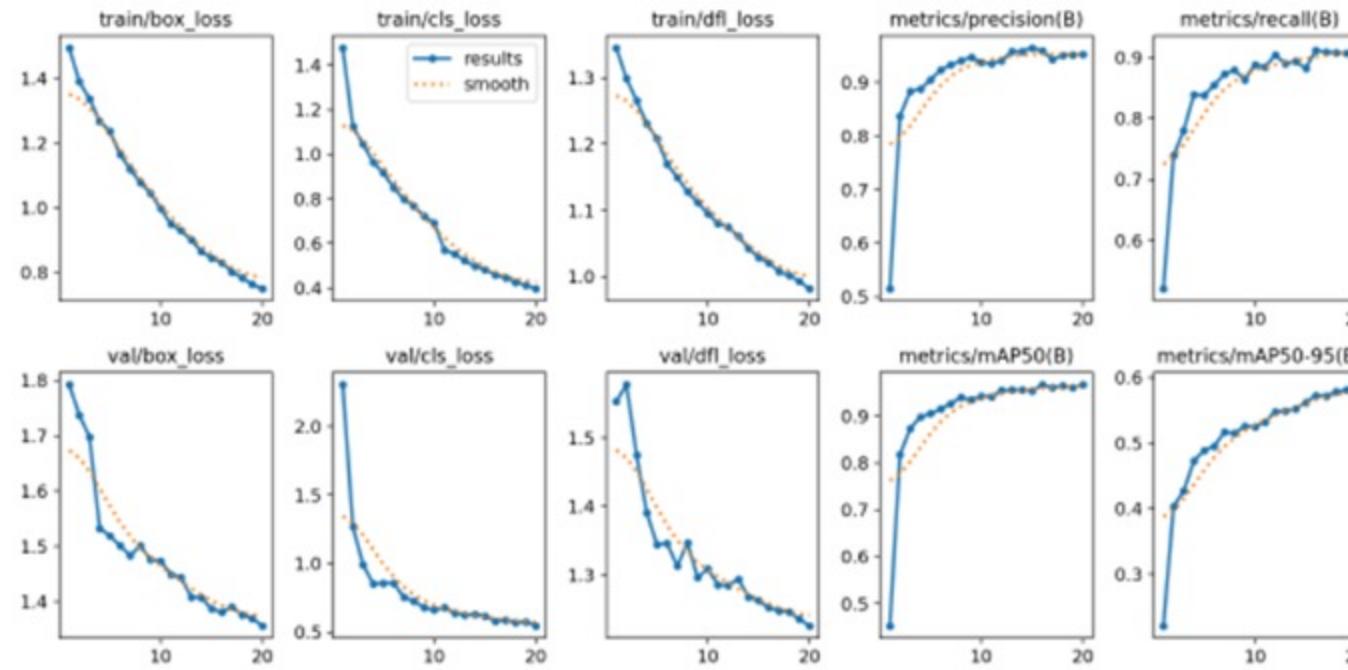
단계	1단계		
목표	개찰구 감지		
상황	앱 실행	개찰구 감지 성공	개찰구 감지 실패 (4초간)
안내 멘트	"개찰구를 찾고 있습니다. 삐 소리가 날 때까지 카메라를 천천히 움직여주세요"	"삐" "앞으로 이동하세요"	"개찰구를 찾고 있습니다. 삐 소리가 날 때까지 카메라를 천천히 움직여주세요"

단계	2-1단계			
목표	Go_entrance 감지			
상황	시작 멘트	Go & Go	Go & Stop	Stop & Stop
안내 멘트	"개찰구가 가까이 있습니다. 잠시 멈춰주세요"	"앞으로 이동하세요"	"카메라를 천천히 오른쪽/ 왼쪽으로 움직여주세요"	"카메라를 천천히 오른쪽/ 왼쪽으로 움직여주세요"

단계	2-2단계		
목표	개찰구 통과		
상황	시작 멘트	Go	Stop
안내 멘트	"삐" "앞으로 이동하세요"	"삐" (0.5초에 1회 반복)	"잠시 멈춰서 카메라를 천천히 움직여주세요 "

# TDMA v2 : Results

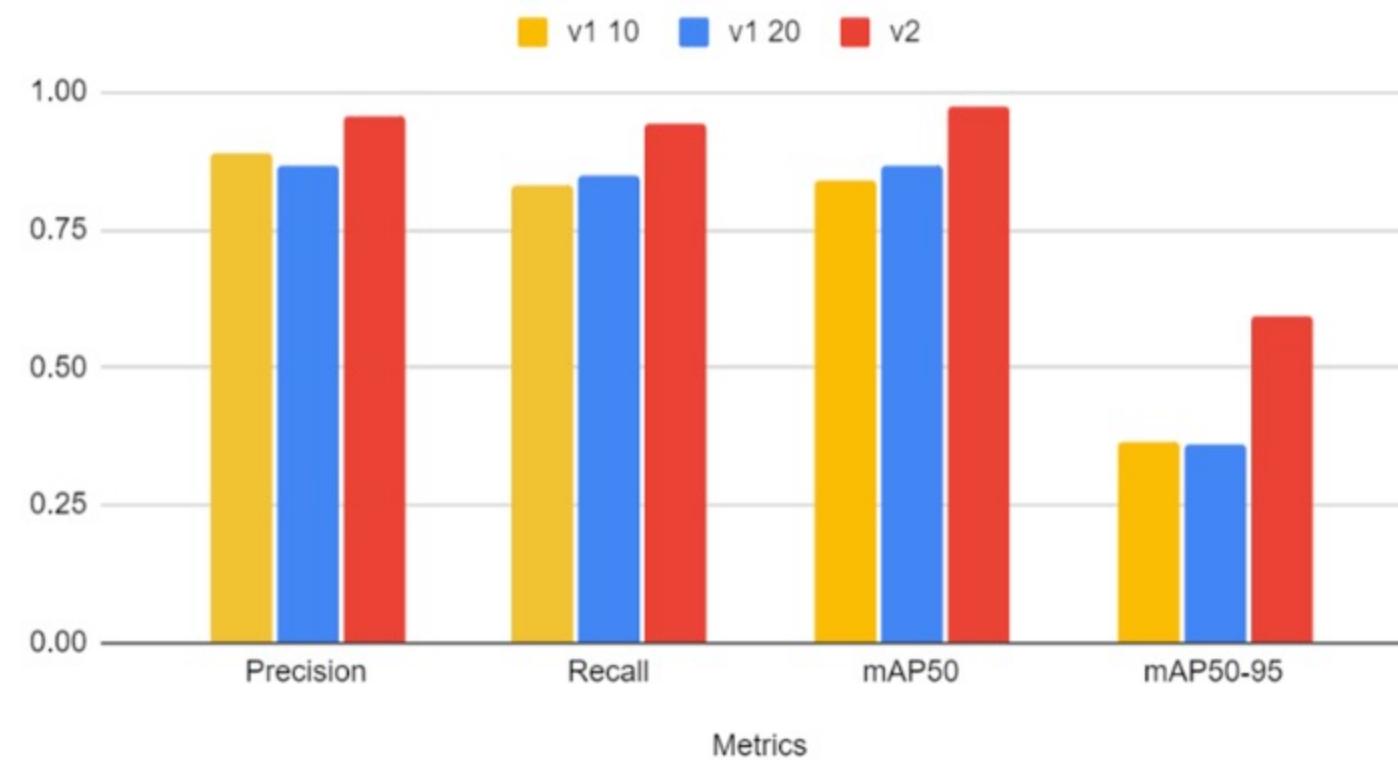
Class	Images	Instances	Precision	Recall	mAP50	mAP50-95
All	291	1262	0.95	0.917	0.968	0.591
Entrance	291	585	0.944	0.894	0.959	0.585
Go_entrance	291	677	0.956	0.941	0.941	0.597



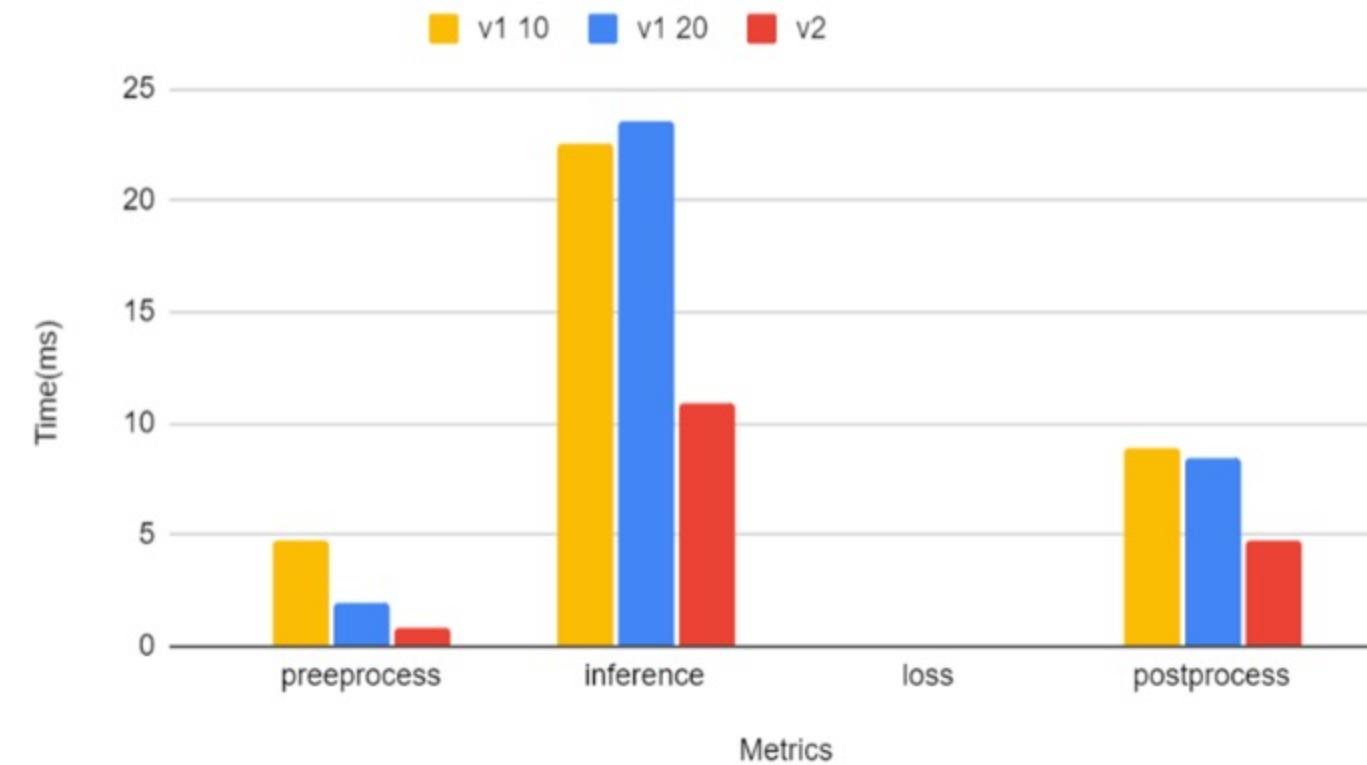
High accuracy even in special cases

# Results : TDMA v1 & v2

Validation



Speed



## **Results Video**

