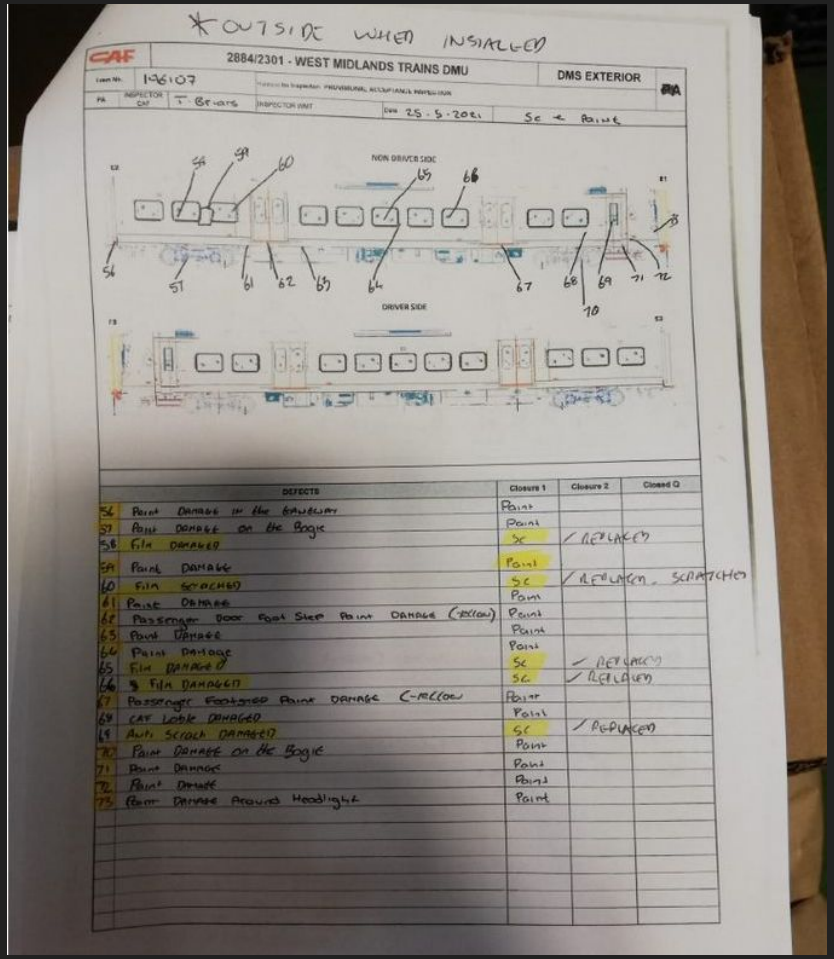
Detecting, Mapping and Verifying Signage with Computer Vision

ELEC6200: GDP: Group 52

Project Overview

Problem Context

- Client - Stewart Signs Ltd
- Signage on trains undergoes regular maintenance
- The inspection of these signs is done by eye
- Results are recorded manually



There is no reliable and efficient way to ensure the inspection, recording and remediation of non-conforming labels, exposing the manufacturers to the liquidated damages

Project Goals

- Develop the proof-of-concept system
- Evaluate performance of system
- Demonstrate viability of system as commercial product
- Test the system with potential users

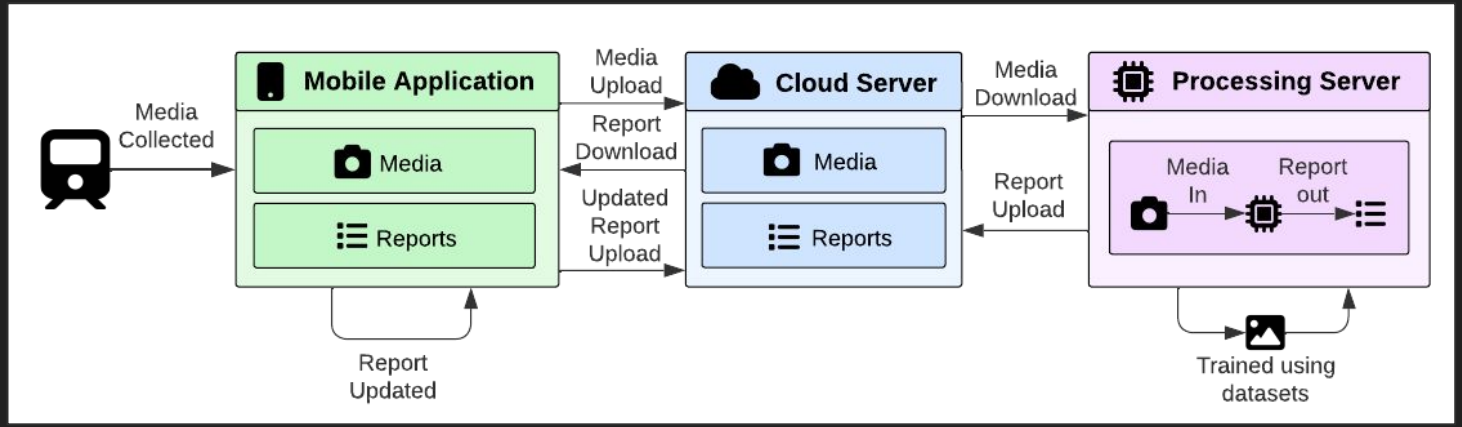
System Design

Core Project Components

- Application
- Processing Server
- Datasets
- Cloud Server

Work Allocation

- Dataset: handled as group
- Team 1:
  - Application + Cloud Server
- Team 2:
  - Processing server



Application

The Need

- Need a method of capturing footage from the train to be processed

How?

- Options:
  - Specialized device
  - Mobile application
- Final choice:
  - Mobile application - (Dart + Flutter)

Justification

- Reduced costs
- Skills audit
- Saves us work

https://dart.dev/assets/shared/dart-logo-for-shares.png?2
https://docs.flutter.dev/assets/images/flutter-logo-sharing.png

Defined app structure based on requirements

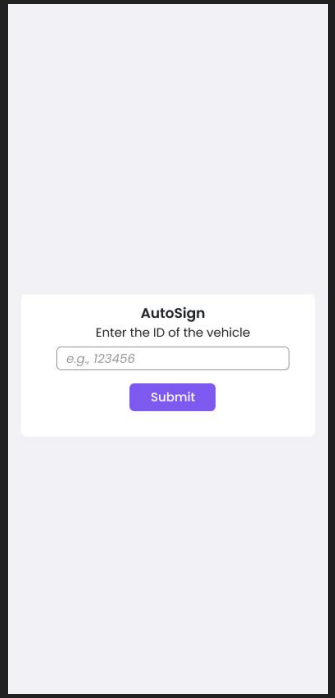
For each train carriage:

- Footage Capturing

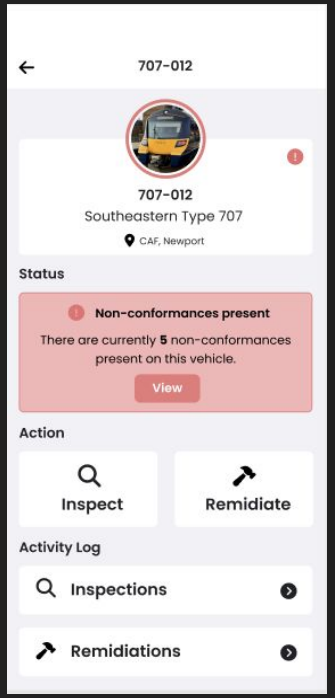
- Viewing Carriage Data (Results of Processing)

- Purchasing of Signs

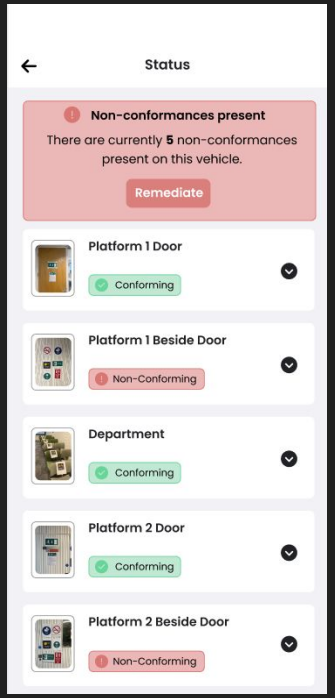
Mockups - Initial Pages



App Home Page

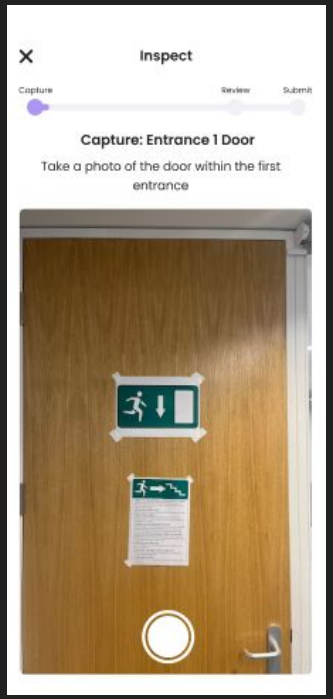


Train Profile Page

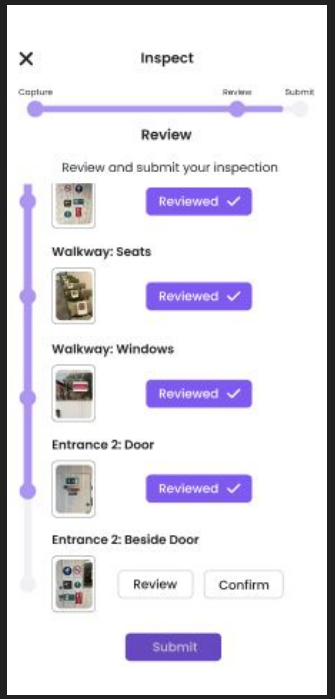


Train Status Page

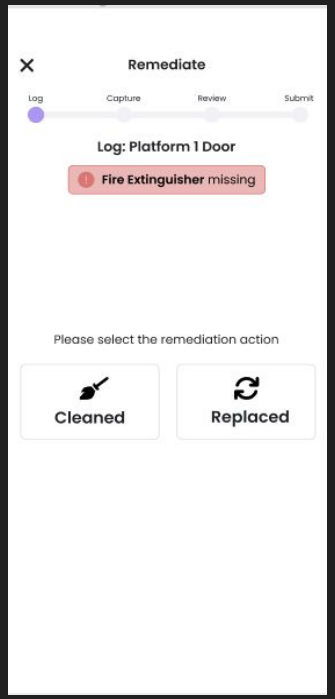
Mockups - Footage Capturing



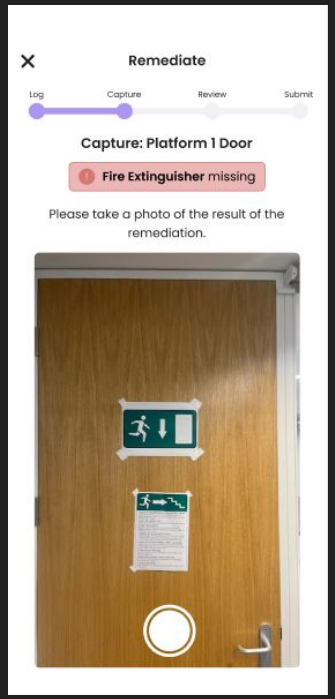
Inspection 1



Inspection 2

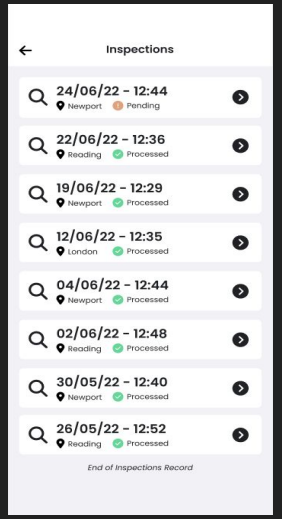


Remediate 1

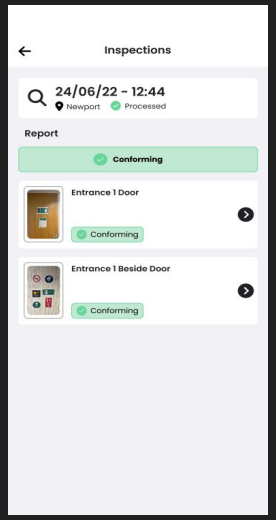


Remediate 2

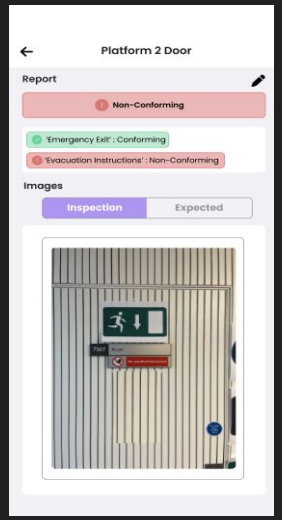
Mockups - Viewing Data



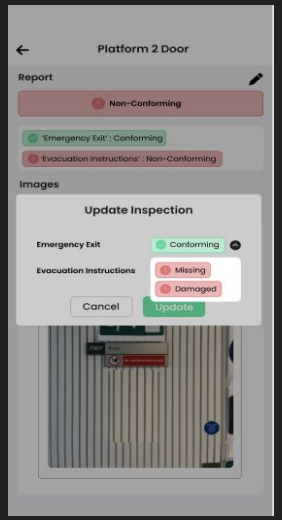
Inspections 1



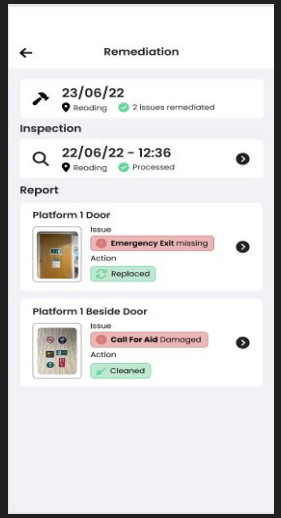
Inspections 2



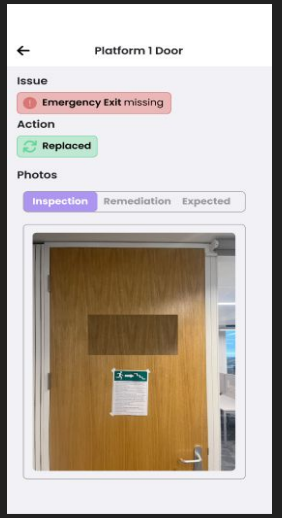
Inspections 3



Inspections 4

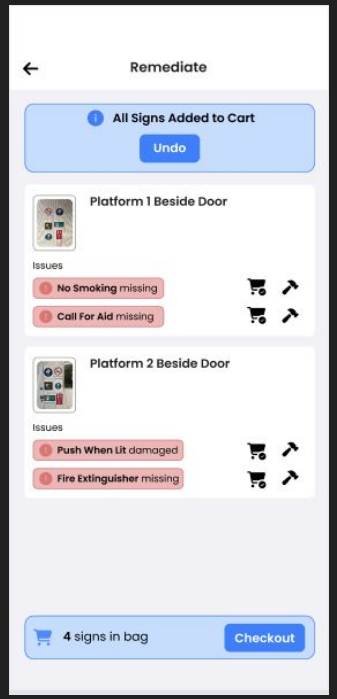


Remediation 1

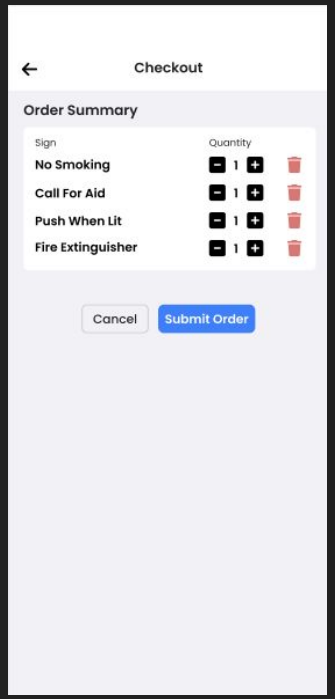


Remediation 2

Mockups - Purchasing Signs



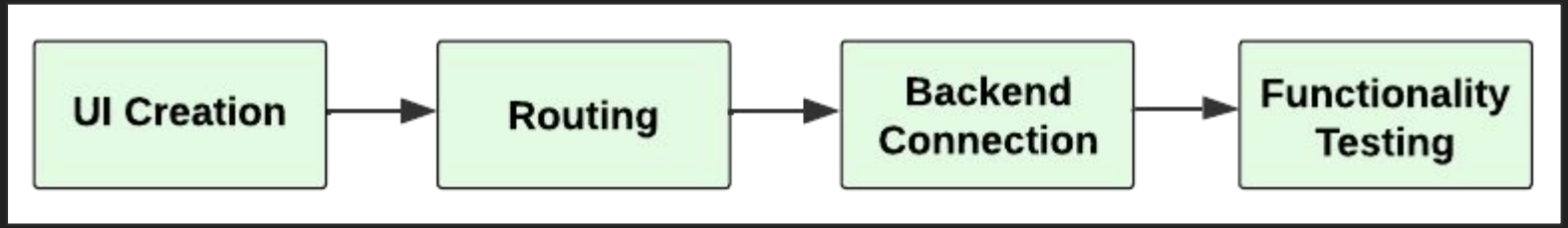
Checkout 1



Checkout 2

Development Process

1. Front-end
2. Routing and navigation
3. Backend development with data model
4. Connection to the cloud server
5. Testing (Scenarios & Storyboard)



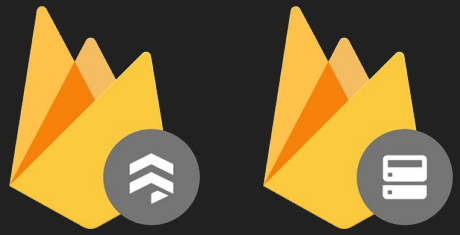
Cloud Server

The Need

- System data required by mobile application and processing server (database + storage)

How?

- Options:
  - Dedicated server?
  - Direct communication between application and processing server?
  - Backend As a Service (BaaS) platform?
- Final choice:
  - BaaS: Firebase (Firestore + Storage)

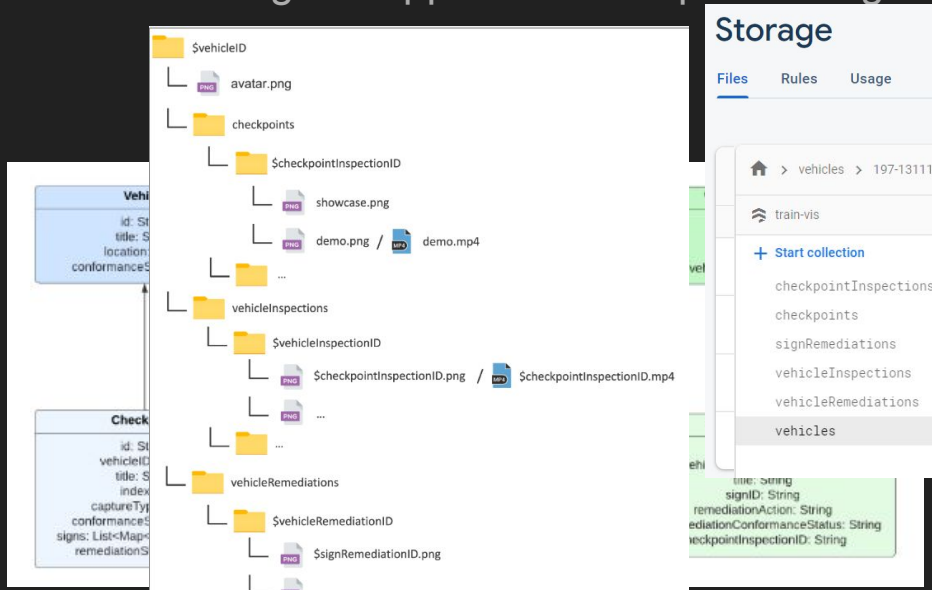


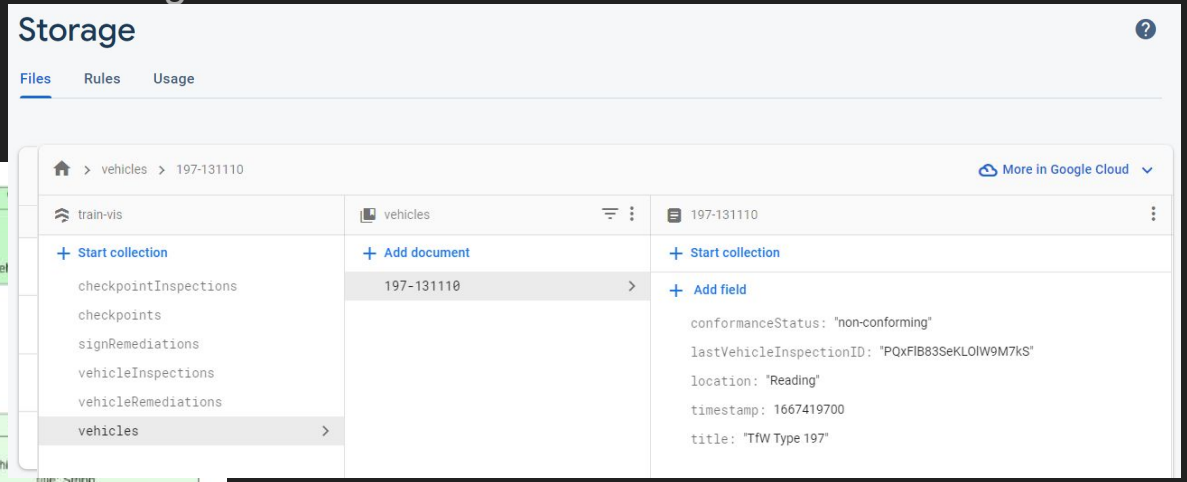
Justification

- No code
- Reliability
- Secure
- Well-Documented
- Easy Integration

What?

- Initialize required services
- Define application data model (database + storage)
- Integrate application and processing server with Firebase





Dataset

Dataset creation

Process:

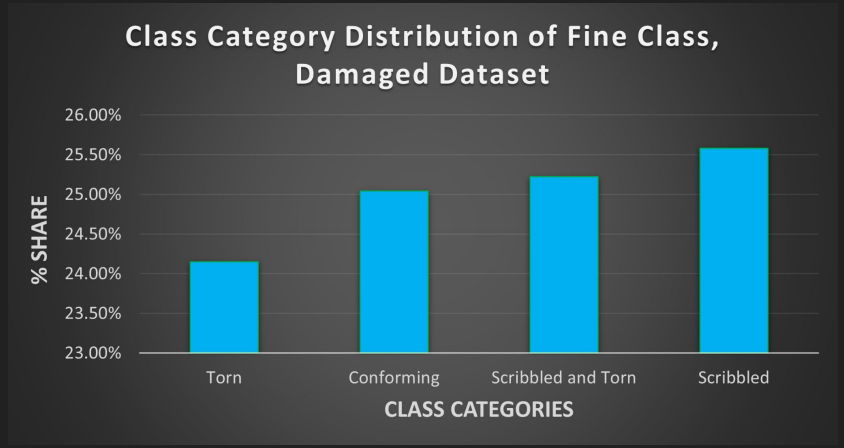
- Data collection
- Annotation

Created:

- Mock Dataset
- Real Dataset
- Damaged Dataset (From CAF)

Damaged Dataset

- Programmatically created
  - Drawn on signs
  - Torn signs



Processing Server

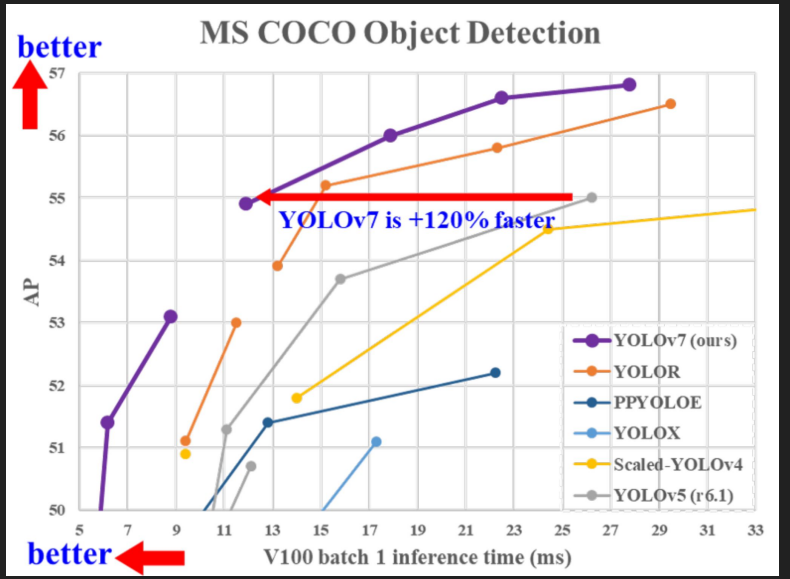
Objectives

- Identify signs in a scene
- Classify damaged signs
- Include functionality for videos

Output of Object Detector

Model Selection/Overall Approach

- First identified appropriate models to use
- Fine-tuned pre-trained YOLOv7 [1] object detector
- BEiT [2] classification model for classifying damaged signs
  - Fine: 4 classes
  - Coarse: binary conforming/damaged
- Custom object tracking algorithm for video-inspections

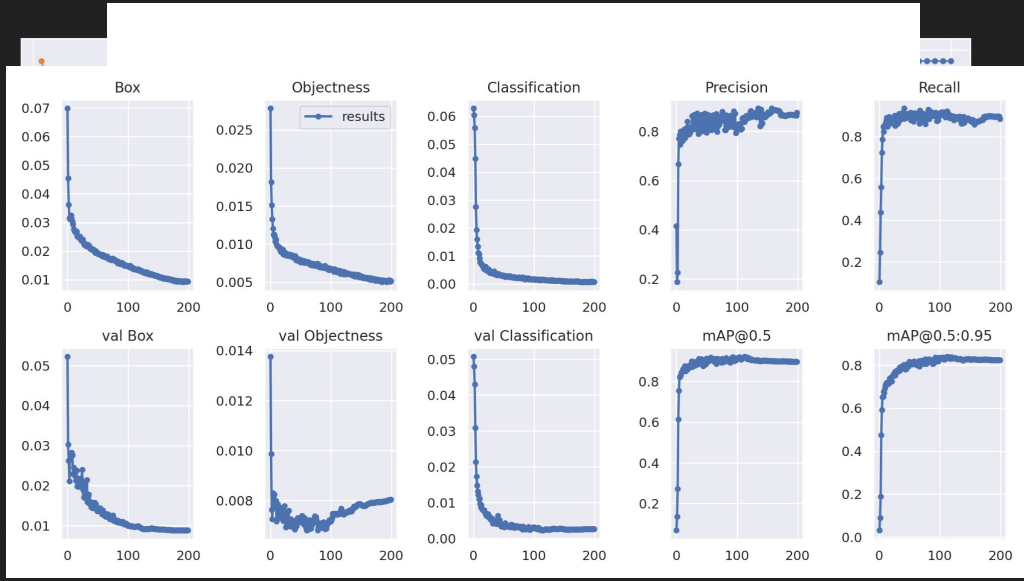


Source: [1]

[1]: Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. “YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors”. In: arXiv preprint arXiv:2207.02696 (July 2022)
[2]: Hangbo Bao, Li Dong, and Furu Wei. “Beit: Bert pre-training of image transformers”. In: arXiv preprint arXiv:2106.08254 (2021)

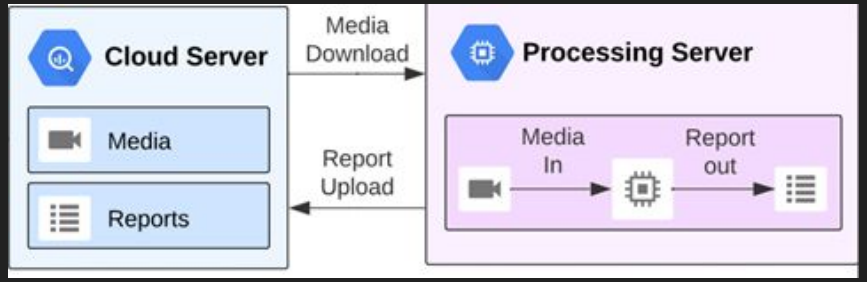
Performance

- Object detection dataset was unbalanced - oversampled
- YOLOv7 - high performance in trained environment
- BEiT
  - High performance for image media
  - Performance was impacted for videos



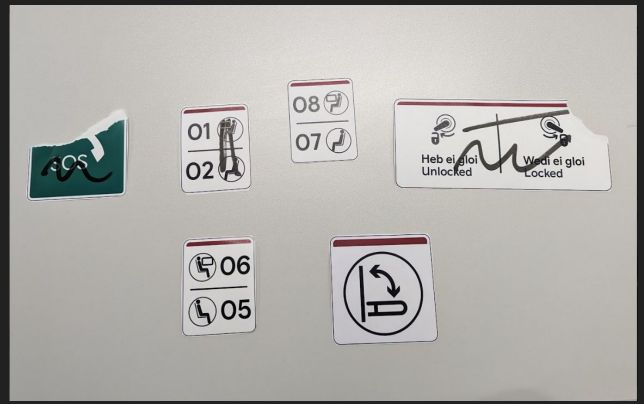
Server Workflow

1. Load the unprocessed inspections from the cloud
2. Localize and classify the objects within the inspections
3. Identify the non-conformances within inspections
4. Update the conformance status of processed inspections in cloud



Field Testing

- CAF difficulties - did not have enough time on site
- Recreated environment off-site to continue field testing



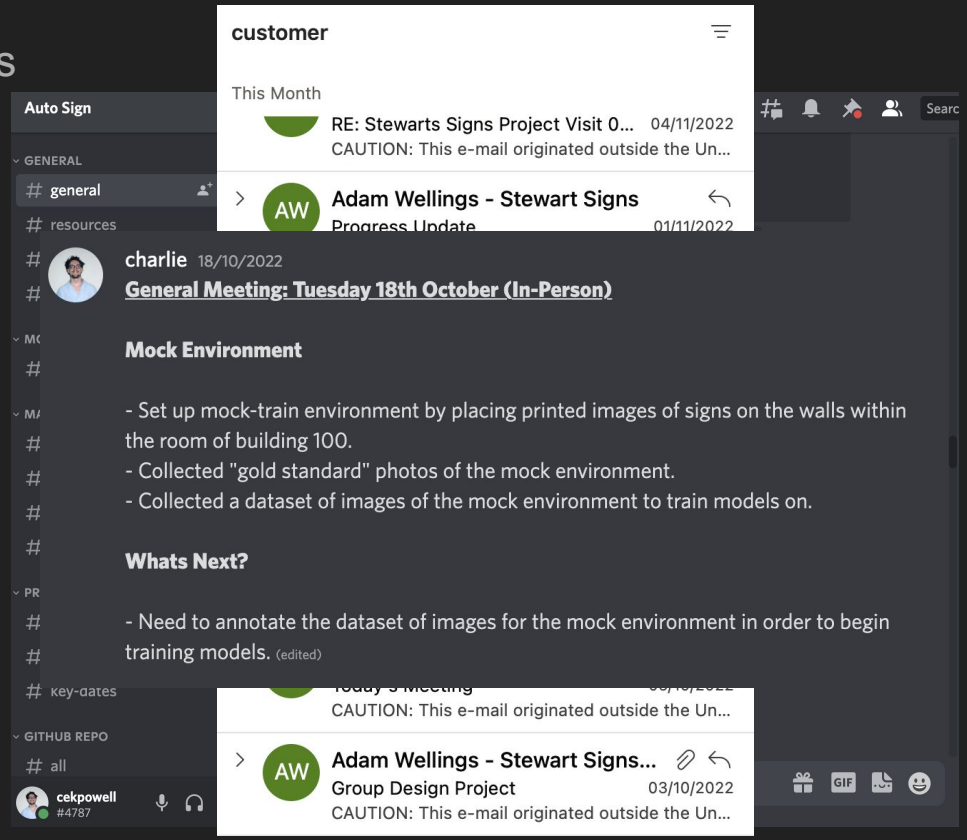
Project Management

Allocation of Work

- Project manager elected.
- Tasks assigned based on personal skills

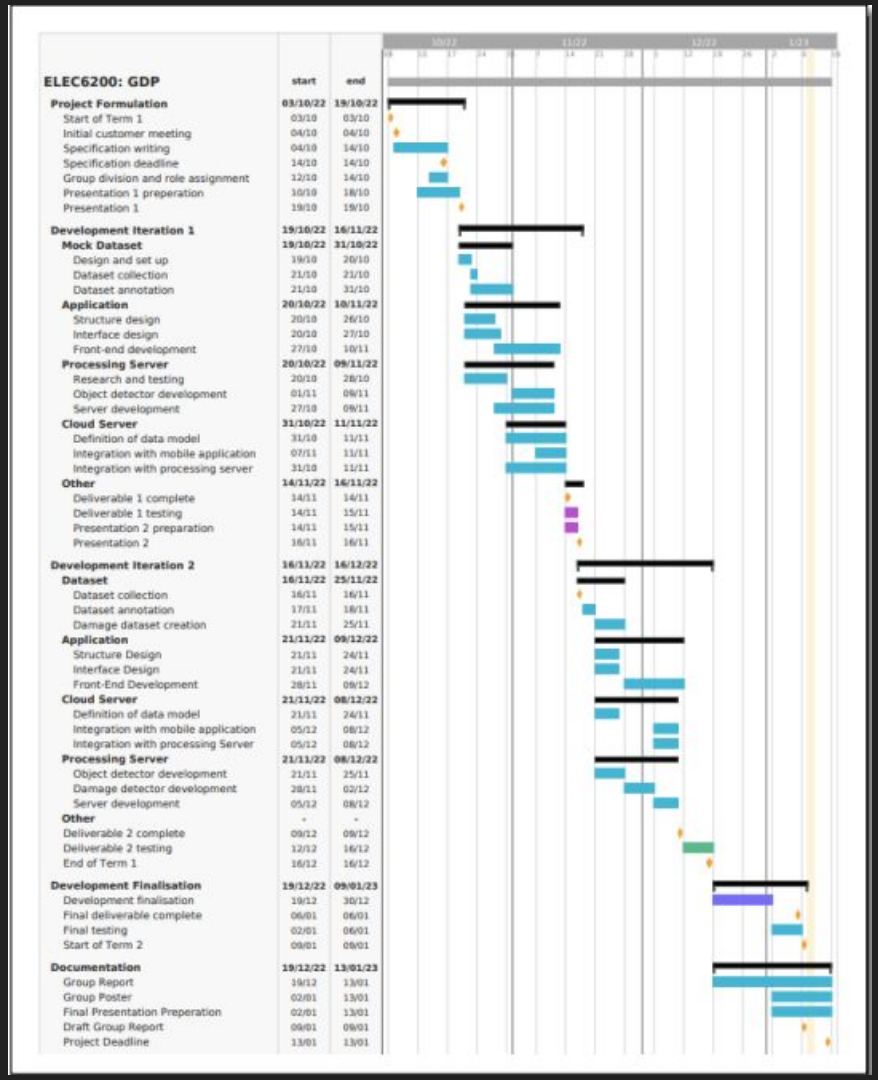
Communication

- Group communication:
  - Discord group
  - Bi-weekly meetings
- Supervisor communication:
  - Weekly meetings
- Customer communication:
  - Regular contact via email.
- Agenda and minutes for all meetings.



Time Management

- Weekly plans
- Weekly reports
- Regular progress meetings



ELEC6200: GDP	start	end
Project Formulation	**03/10/22**	**19/10/22**
Start of Term 1	03/10	03/10
Initial customer meeting	04/10	04/10
Specification writing	04/10	14/10
Specification deadline	14/10	14/10
Group division and role assignment	12/10	14/10
Presentation 1 preperation	10/10	18/10
Presentation 1	19/10	19/10
Development Iteration 1	**19/10/22**	**16/11/22**
Mock Dataset	**19/10/22**	**31/10/22**
Design and set up	19/10	20/10
Dataset collection	21/10	21/10
Dataset annotation	21/10	31/10
Application	**20/10/22**	**10/11/22**
Structure design	20/10	26/10
Interface design	20/10	27/10
Front-end development	27/10	10/11
Processing Server	**20/10/22**	**09/11/22**
Research and testing	20/10	28/10
Object detector development	01/11	09/11
Server development	27/10	09/11
Cloud Server	**31/10/22**	**11/11/22**
Definition of data model	31/10	11/11
Integration with mobile application	07/11	11/11
Integration with processing server	31/10	11/11
Other	**14/11/22**	**16/11/22**
Deliverable 1 complete	14/11	14/11
Deliverable 1 testing	14/11	15/11
Presentation 2 preparation	14/11	15/11
Presentation 2	16/11	16/11
Development Iteration 2	**16/11/22**	**16/12/22**
Dataset	**16/11/22**	**25/11/22**
Dataset collection	16/11	16/11
Dataset annotation	17/11	18/11
Damage dataset creation	21/11	25/11
Application	**21/11/22**	**09/12/22**
Structure Design	21/11	24/11
Interface Design	21/11	24/11
Front-End Development	28/11	09/12
Cloud Server	**21/11/22**	**08/12/22**
Definition of data model	21/11	24/11
Integration with mobile application	05/12	08/12
Integration with processing Server	05/12	08/12
Processing Server	**21/11/22**	**08/12/22**
Object detector development	21/11	25/11
Damage detector development	28/11	02/12
Server development	05/12	08/12
Other	-	-
Deliverable 2 complete	09/12	09/12
Deliverable 2 testing	12/12	16/12
End of Term 1	16/12	16/12
Development Finalisation	**19/12/22**	**09/01/23**
Development finalisation	19/12	30/12
Final deliverable complete	06/01	06/01
Final testing	02/01	06/01
Start of Term 2	09/01	09/01
Documentation	**19/12/22**	**13/01/23**
Group Report	19/12	13/01
Group Poster	02/01	13/01
Final Presentation Preperation	02/01	13/01
Draft Group Report	09/01	09/01
Project Deadline	13/01	13/01

Project Feedback

Project Feedback

Client

- Satisfied with project result
- Interested in future development
- “Very Satisfied” and “Opens an opportunity to a real world tool”

Potential Users

- More efficient inspections
- Users would prefer AutoSign over the current manual process
- “Could be used by everyone in the industry” as it “makes it easier for everyone”

Conclusion

Conclusion

- Developed proof-of-concept system
- Comprehensive testing and evaluation from users show project success of this system
- Demonstrated feasibility of approach
- Laid foundations for development into a commercial product

Customer Recommendations

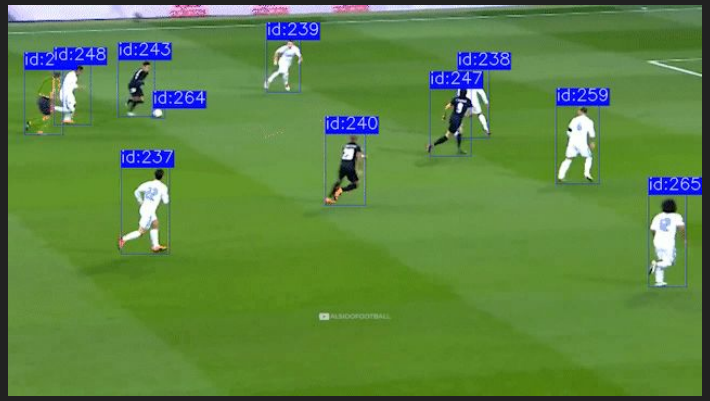
Customer Recommendations

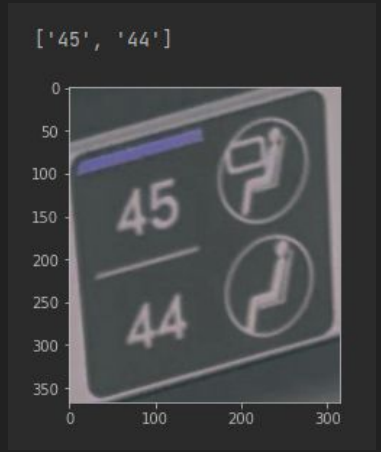
- Expand data collection:
  - Improve future ML features
  - Data Insights
- Desktop application:
  - Smarter work allocation
  - Customer audits

https://blog.video.ibm.com/ai-video-technology/using-big-data-to-enrich-your-streaming-experience/

Future Features

- Integration with customer's existing software
- Single video inspections
- Seat number recognition
- Position and orientation checking
  - App-integrated technical documents





https://learnopencv.com/wp-content/uploads/2022/06/04-football-demo.gif

Demo

Questions