FIELDBOX

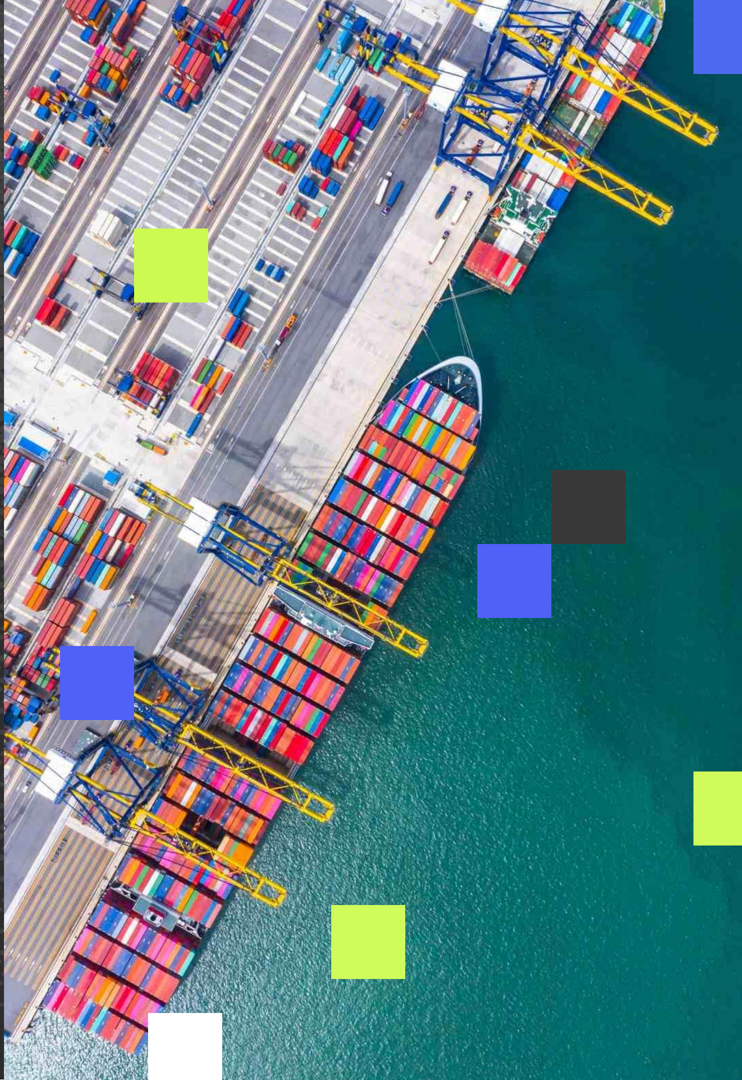
Container inspection automation: a proof of concept

APIA conference, Strasbourg, July, 6th, 2023

Speaker: Sonia Tabti, PhD

Co-authors: F. Vallée, D. Poon, M. Elion, Y. Alouini,

Team: C. Collignon, P. Oreistein, P. Chin, R. Yang, A. Cherifi, A. Coutant, J. Sauer



Container inspection: a major logistic challenge

The context in numbers at Singapore

11 container depot operators in Singapore

~ 20 container depot yards

3 to 5 surveyors at each yard

~ 500 containers inspected daily

~ 30% of the containers have some form of defects

Estimated time taken to inspect each container:

5 to 10 mins: no defects,

20 mins: minor defects,

40 mins: serious defects.

Inspection workflow

- Defects' pictures taken with a tablet
- Point of view: **really close** to the defect
- Multiple pictures taken per defect
- Repair recommendations proposed
- Defects are submitted to an ISO norm

Objectives: towards container inspection automation

Summarized client’s endgame : container inspection automation

- Automatic capture of images inside and outside of the container
- Detection and classification of defects
- Recommendations of repairs
- Identification of the container ID

POC scope (from Nov 2021 to June 2022)

- Project framing
- Cold data exploration
- Hardware install
- Data collection
- Build model(s) to detect prioritized defects
- Test this model(s) in realistic conditions

State of the art in container inspection

	X-ray inspection	Laser-based	Image-based
Commercial products	→ No AI	→ Found a 2014 doc, no recent one	→ Focus on OCR and exterior damages
Research	→ Most references found in inspections for contreband Example: [Abdolshah et al., 2017] Eg. of reference with neutron and gamma-ray systems: [Marques et al., 2023]		For most papers: ● Models work on pictures taken with close point of view (adapted for mobile/tablet inspection, not full automation) ● Focus on one or most recurrent defects ● Many use deep learning classification → But localization is key ! ● Focus on exterior inspection (interior is important too) Some references: [Baharmi et al., 2022], [Zixin Wang et al., 2021], [Klöver et al., 2020]

Our container inspection journey

01

Project framing

Historic data exploration
Scoping

Hardware selection

02

Data preparation

Image extraction

Labeling

Analysis and rescoping

03

Modeling

Object detection: Yolo

Data splitting and details

Results and analysis

04

Conclusion

REX

Perspectives

Project framing

01

- Cold data exploration
- Scope definition : defects shortlist
- Hardware selection

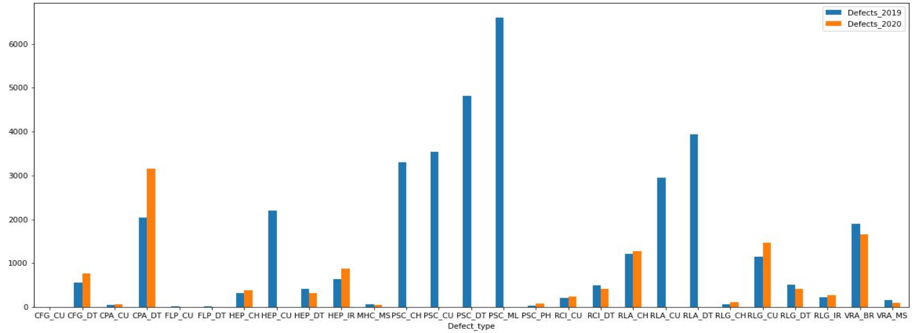
Cold data exploration: high imbalance and Covid impact

- **Goal:** define a shortlist of defect types to focus on for this POC
- Historic data from 2019 and 2020
  - Composed of: defects pictures, Jsons files
  - **116 defect types explored** in total
  - Total volume: 186 477 images / **96 GB** (compressed)
    → Big data
- **24 types of defects were selected**

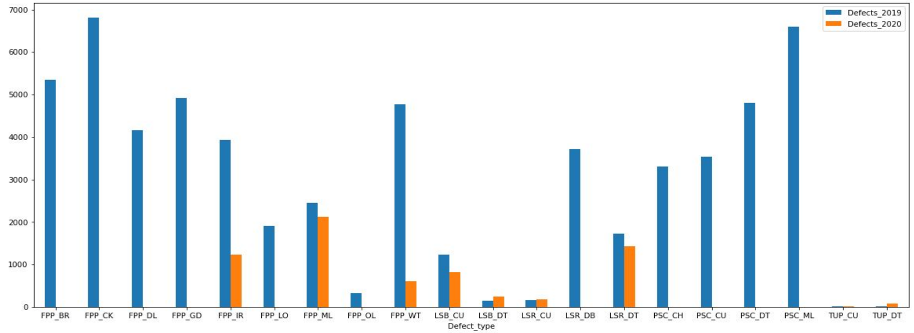
Location	Door	Exterior	Interior	Marking	Washing
Number of defect types	49	27	20	18	2
Number of images	52164	49045	66680	7176	11412

Computation of the defects distribution for each location

- Separate 2019 and 2020 to avoid the Covid bias
- Imbalanced defects distribution.
- Defects distribution can vary from one year to another



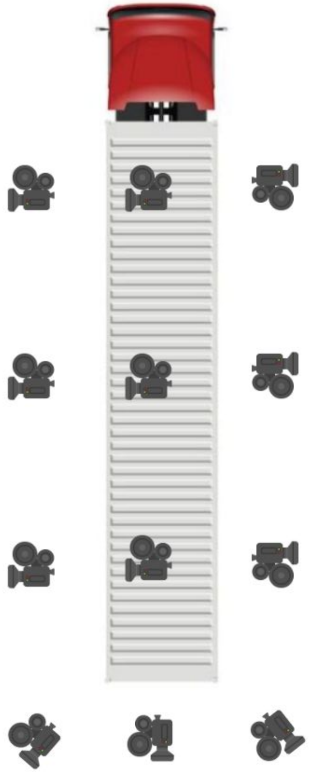
Exterior defects distribution (blue: 2019, orange: 2020)



Interior defects distribution (blue: 2019, orange: 2020)

Hardware: camera array for exterior inspection

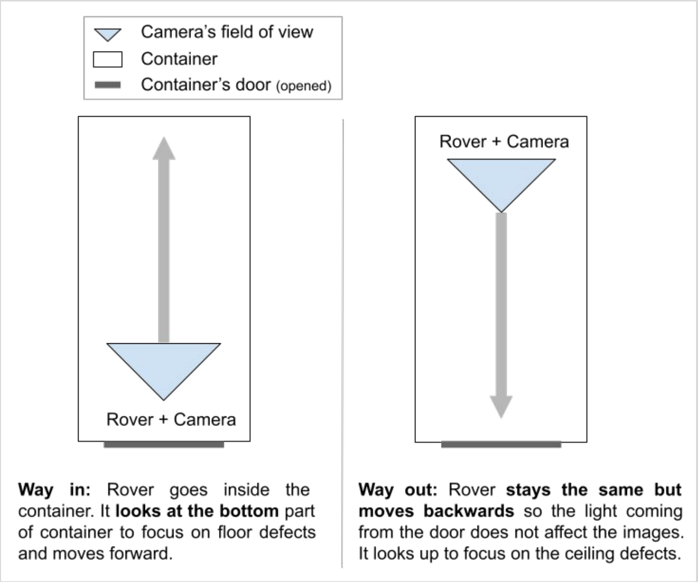
- Deciding the appropriate hardware to collect data was not straightforward
  - We settled for a set of 2K resolution CCTVs
- Deciding where and how to install the cameras in order to optimize defects capture was not straightforward either
- Data collection:
  - We acquired a NAS (Network Attached Storage) to record videos and access them remotely
  - Automated sync procedures with a cloud storage is a good way to go



Hardware: rover for interior inspection

- We settled for a rover for interior inspection
  - With a 1080p camera
- Current workflow:
  - An operator opens the container’s door
  - Puts the rover inside
  - Drives it according to a path we recommended

- Towards an automated workflow for a next project phase:
  - Sensors on the rover could have been a good option
  - **But:** they are not adapted to such a metallic structure
  - **TODO:** Hardcoding the rover’s path thanks to the rovers’s SDK

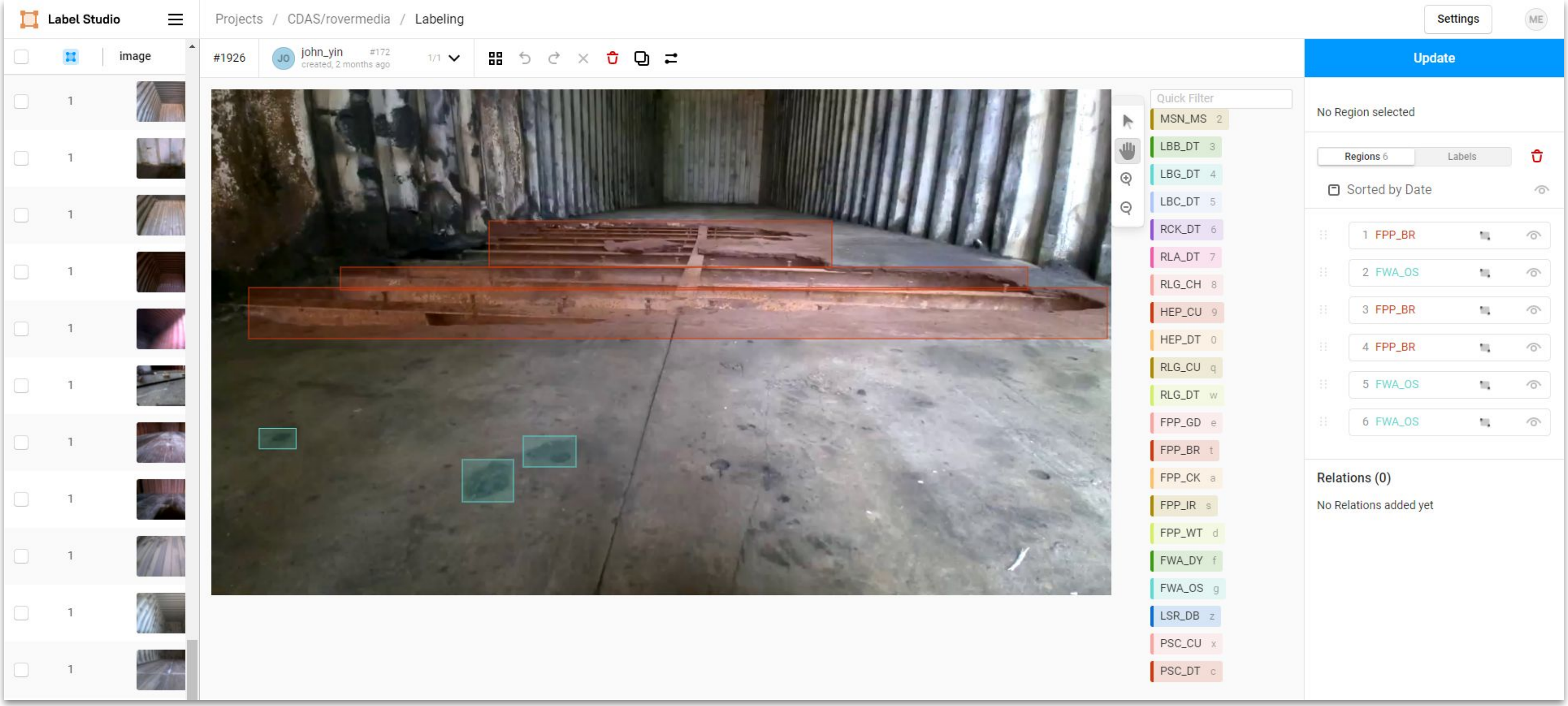


Data preparation

- Labeling
- Analysis of collected data and rescoping

02

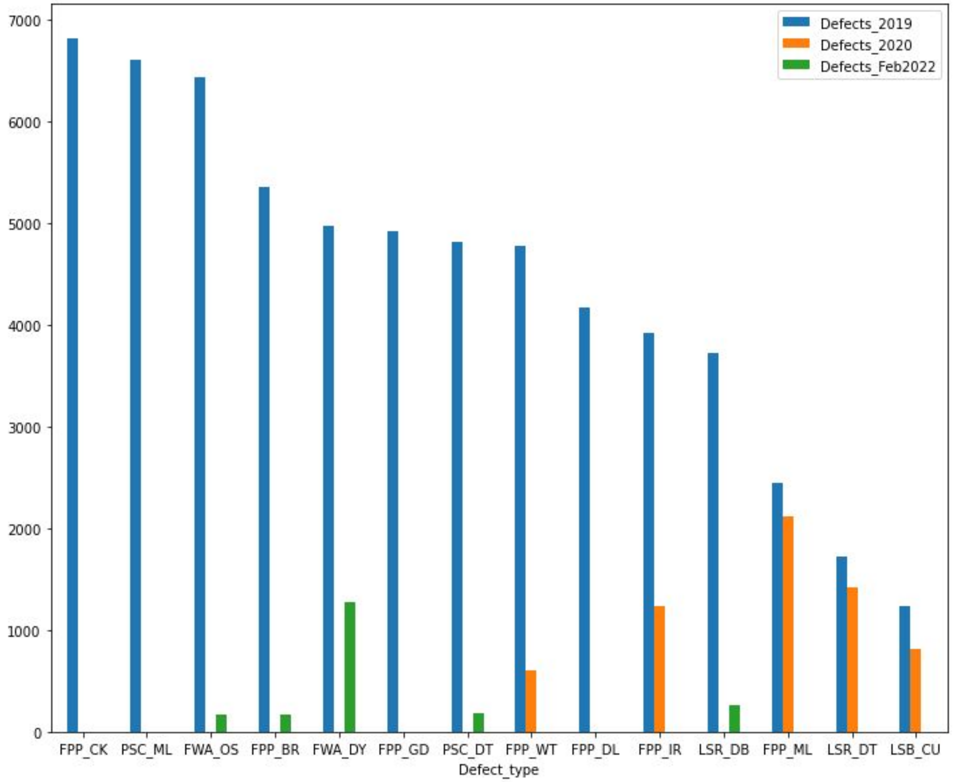
Data labeling with Label Studio



Data collection: analysis and rescoping to 5 interior defects

- One month of data collection
- Covid impact: not enough container traffic
- Not enough defects observed
- Highly imbalanced defects’ distribution
- More than 100 defects collected for 5 types only
- **All interior defects**

→ Rescoping to the 5 most occurring defects (interior and washing)



Interior and washing defects distribution in 2019, 2020, Feb 2022 (data collection)

Modeling

- Object detection with Yolov5
- Results and analysis

03

Modeling: object detection with Yolov5

- Modeling strategy: object detection
  - Locating the defect is important
- Model selected: Yolov5, size M
  - Good performance and fast inference
  - Good compromise on training time
- For a given image, Yolo inference works as follows:
  - splits the image into a grid
  - locates and classifies objects of interest in each cell

⇒ Regression problem: generation of a set of bounding boxes, object confidence and class probabilities for each cell

- For training, a **transfer-learning** approach was followed:
  - Using a model pretrained on the COCO dataset

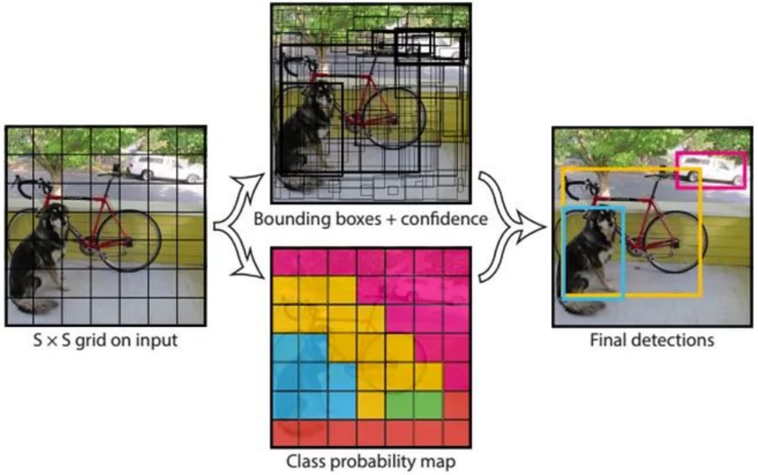


Figure 2: The Model. Our system models detection as a regression problem. It divides the image into an $S \times S$ grid and for each grid cell predicts B bounding boxes, confidence for those boxes, and C class probabilities.

Image courtesy from the first Yolo paper

Modeling: data splitting

- Dataset was cautiously split into subsets so as to **avoid introducing unwanted bias**
    - 80% train (~ 1100 imgs) - 20% test (~ 260 imgs)
        - the training (sub)set is used for model training
        - the test (sub)set is used for model evaluation
    - Images from the same video are not split across different subsets
    - The original distribution of defects is preserved in all subsets (used: <u>iterative stratified split</u> for multi-label cases)

- The percentage of background images (i.e the no_defects class) was lower than 10%

- Data augmentation operations have been applied
    - vertical and horizontal flip
    - alteration of contrast and colors
    - image translation

⇒ ~ 500 images generated with data augmentation

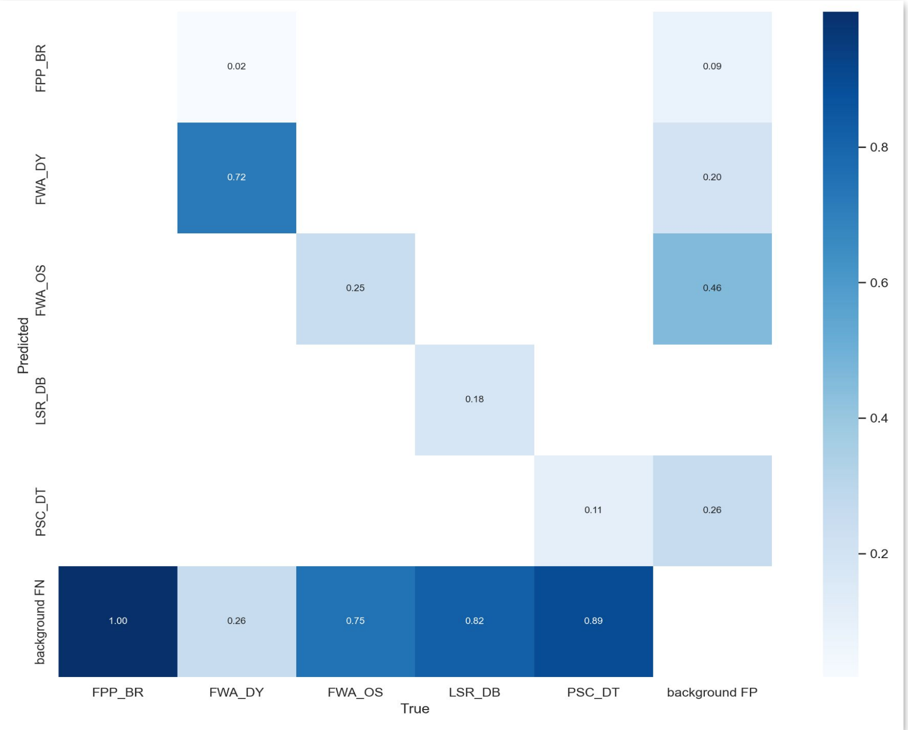
	FWA_DY	None	PSC_DT	FWA_OS	FPP_WT	PSC_CU	FPP_GD	FPP_BR	LSR_DB
percentage (original)	30.0	29.3	13.6	5.6	4.3	4.0	3.9	3.2	3.2
percentage (train)	29.7	30.2	12.8	6.3	4.2	3.1	4.1	3.4	3.1
percentage (test)	31.3	26.0	16.4	3.0	4.8	7.3	3.0	2.5	3.5

Defects distribution in each data subset

Results and analysis: the impact of data volume

- Model training has converged (using rover data)
- Preliminary results (obtained on test images) :
    - 70% of labeled FWA_DY defects have been detected
    - 25% of labeled Oil Stains have been detected
    - About 20% of labeled LSR_DB defects have been detected
- Preliminary qualitative results on test videos are promising:
    - 5/6 broken floor defects were detected !
- Even though the limitations are clear:
    - Many detected defects do not correspond to any labeled defect
    - The dataset is too small, hence performances improvable

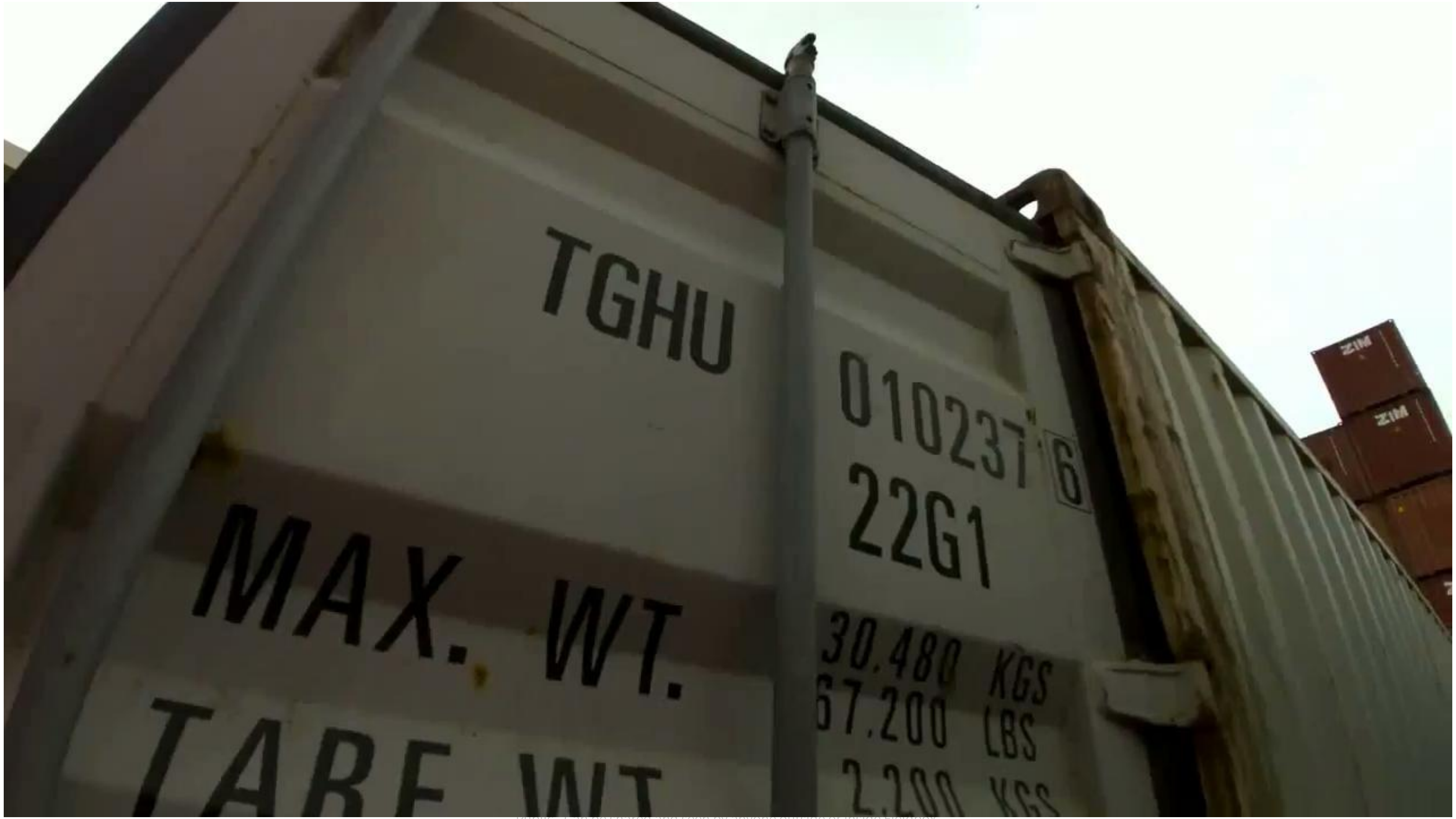
→ **Correlation between performance and number of image samples per class**



Confusion matrix computed using the test set

Demo: test video 1

Demo: test video 2



Demo: test video 3

Conclusions

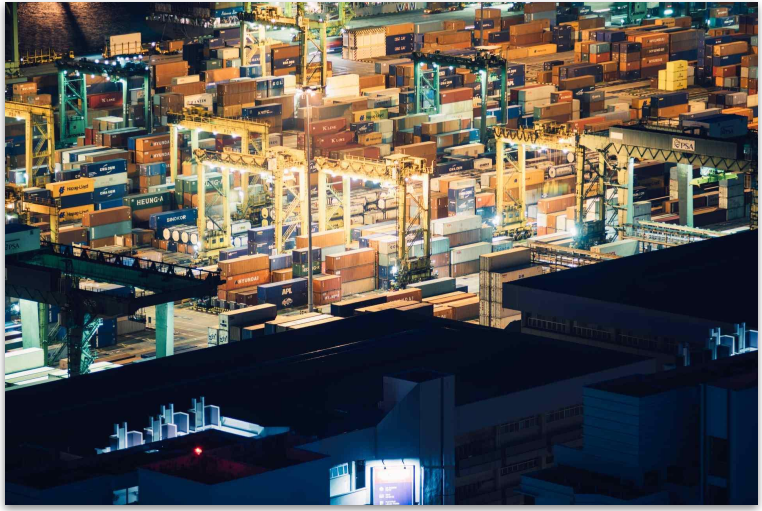
04

- REX
- Perspectives

Conclusion

REX

- Sizing a “project framing “ phase is a good strategy
- Choose your hardware wisely
- Plan data collection carefully
- Train operators for labeling with pedagogy
- Model efficiency: data volume and labels’ quality are key



Perspectives

- Collect more data
- Improve labeling process
- Extend defect detection to other classes
- Automate rover movement
- Deployment

Thank you for your attention !

Questions ?