DB SCHENKER

Real-time applications of Computer Vision in Logistics

Dr. David Zibriczky | Budapest ML Forum | June 2023

DB Schenker is a global company providing third party logistics service in Air/Ocean/Land Freight and Warehousing

DB SCHENKER

Contract Logistics

Air Freight

Ocean Freight

Land Transport

Automobility

Electronics

Industrial

Consumer/Retail

Aerospace/Marine/ Defense

Healthcare

Semicon./Solar

1872
Foundation Year

€23.4B
Revenue[1,2]

140
Countries

76K
Employees[2]

Essen, DE
Head Office

Deutsche Bahn
Parent Company

(1) Adjusted for FX
(2) in 2021

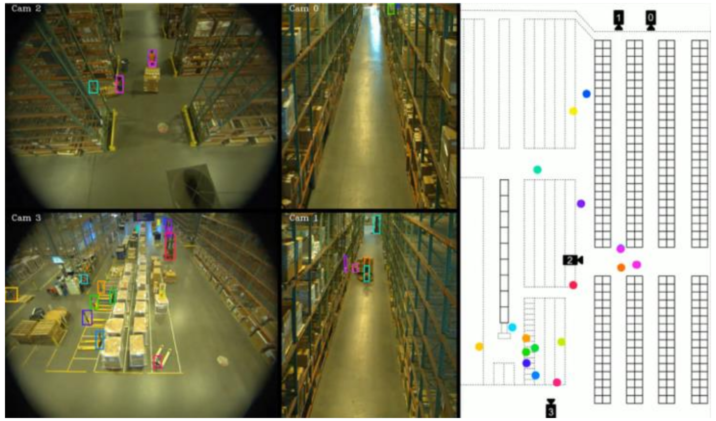
Agenda for this presentation

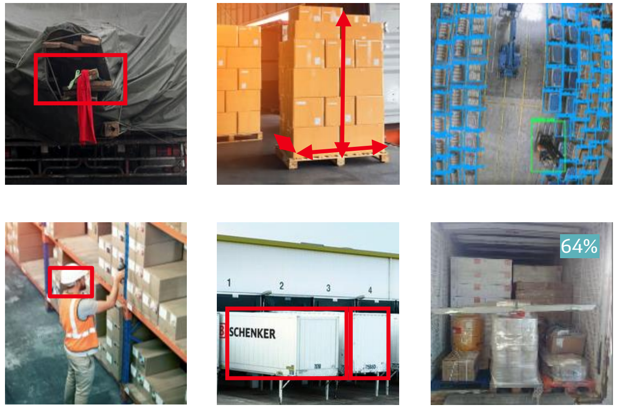
DB SCHENKER

License plate and ILU code recognition

Multi-camera object detection and tracking in warehouses

Overview of other use cases



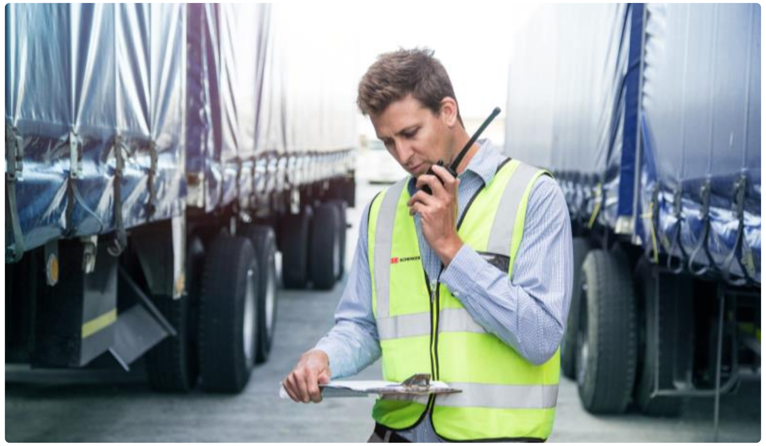


Photos: Schenker AG

License plate and ILU code recognition

Management of arriving/departing trucks at warehouses is usually a manual process

DB SCHENKER



- Warehouse is a hub for collecting and distributing goods in the supply chain
- Used by manufacturers, importers, exporters, wholesalers

- Vehicles enter and leave the yard of the warehouse via entry and exit gate, they are recognized by license plate and ILU code
- Vehicles are routed to the warehouse gate to unload and load cargo

- Registering and routing vehicles are time-consuming work
- License plates are often registered manually on paper, that may include to typing error as well

Use Case: Recognize and report License Plate and ILU codes of arriving and departing vehicles at the entrance gate of a selected DB Schenker terminal, so that the Yard Management can register, route, and track the traffic automatically.

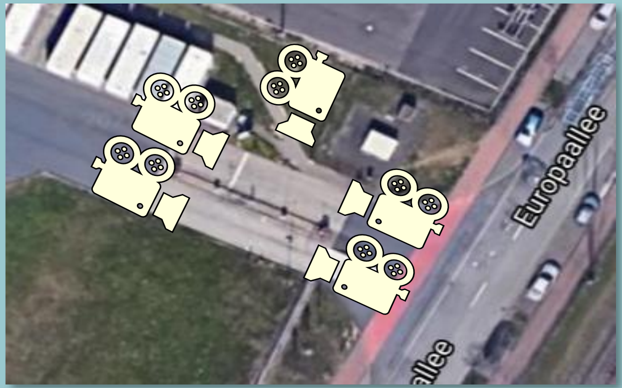
Photos: Schenker AG

We installed 5 cameras at a Pilot Branch in Germany that are collecting data from multiple angles at the entry gate

DB SCHENKER

Camera selection

2 x AXIS P1455-LE
2 x AXIS M2025-LE
1 x AXIS Q1700-LE



Camera installation

Entry and exit
Truck front and back
Side of truck

Source: AXIS P1455-LE Network Camera, https://www.axis.com/de-de/products/axis-p1455-le [Last Access on 01.06.2023]; pictures from Google Maps, https://www.google.de/maps [Last Access on 01.06.2023]; own photos.

We define an object detection problem of trucks, license plates, and ILU codes

DB SCHENKER

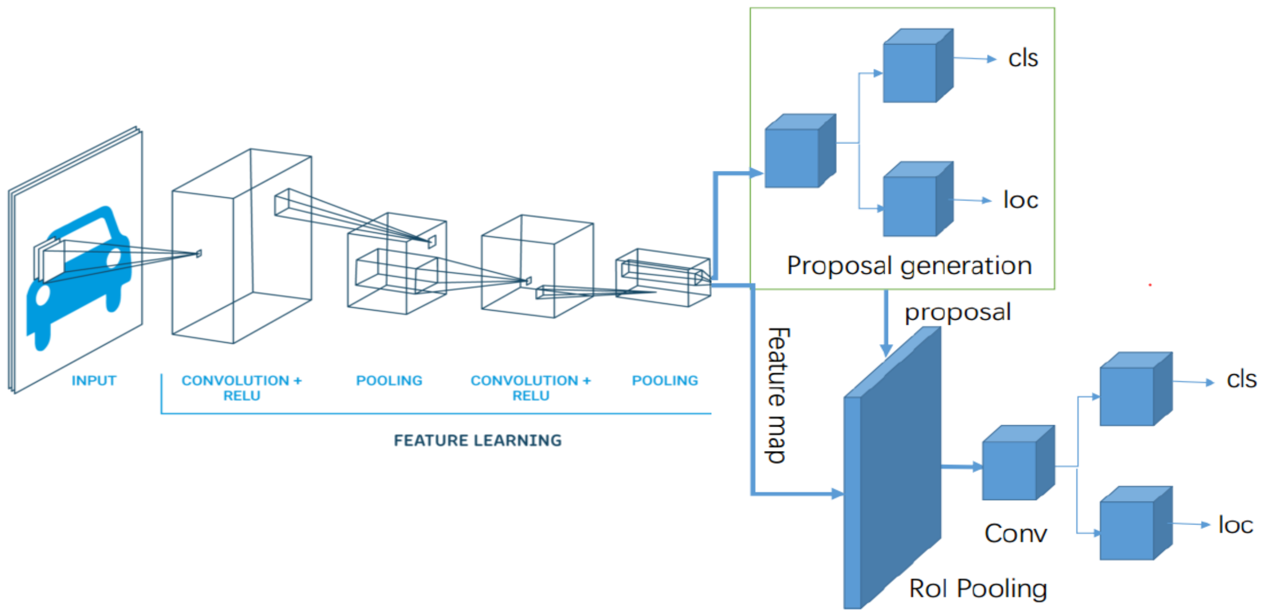
License Plate (LP)

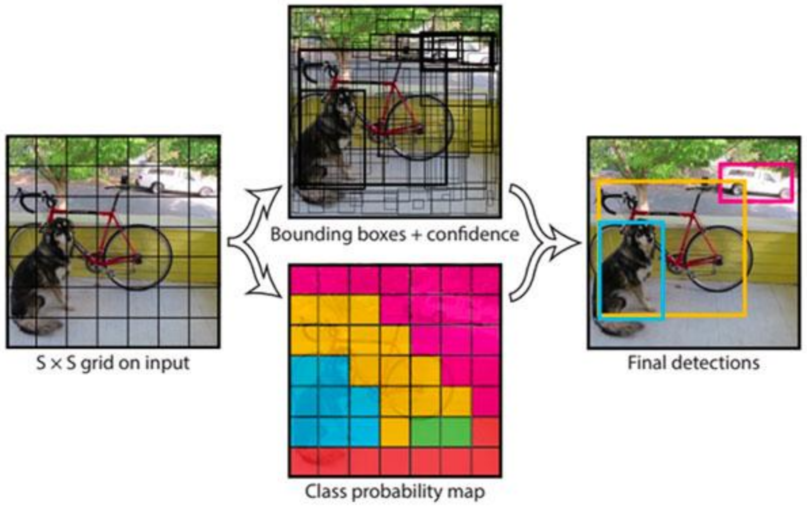
Vehicle

Intermodal Loading Unit (ILU) Code

Photos: Schenker AG

For vehicle/text detection, we consider extended CNN architectures with proposal generation and classification





Source: Understanding Deep Convolutional Neural Networks, https://guides.run.ai/deep-learning-for-computer-vision/deep-convolutional-neural-networks [Last Access on 05.06.2023]; YOLO object detection with OpenCV, https://pyimagesearch.com/2018/11/12/yolo-object-detection-with-opencv/ [Last Access on 05.06.2023].

Out of the investigated object detectors, the clear winner is YOLOX-m with fast runtime on RTX 3090 and Tesla V100

DB SCHENKER

Experimental procedure:

- Annotation with Computer Vision Annotation Tool (CVAT)
- Train and test set: **5023 and 2043 images**
- Open-source algorithms tested[1]: **HTC, YOLOv3, YOLOX**
- Evaluation metrics: **Precision, Recall** and **Runtime**
- Latency measured on various hardware
- Additional detectors to consider[1]: YOLOv7, YOLO-NAS

Key findings:

- **YOLOX-m** detector is our favorite due to high accuracy and acceptable runtime
- **Training** of YOLOX-m took **19 hours**
- **Embedded boards** are **not suitable** for real-time service
- Server with one/two GPUs can serve all cameras simultaneously

Recall of detectors per object type

Detector	Truck	LP	ILU	Ignored Vehicles	All
HTC	98.76%	97.65%	89.52%	92.31%	**97.60%**
YOLOv3	96.58%	87.23%	90.32%	84.62%	**95.32%**
YOLOX-s	92.96%	98.66%	98.39%	92.31%	**95.32%**
YOLOX-m	98.14%	98.82%	98.39%	92.31%	**98.31%**

Runtime of detectors per GPU

Detector	Jetson Nano	Jetson TX2	Jetson Xavier	GeForce 3060	GeForce 3090	Tesla T4	Tesla V100
HTC	N/A	4775ms	1286 ms	371 ms	145 ms	423 ms	208 ms
YOLOv3	N/A	282 ms	89 ms	44 ms	30 ms	34 ms	23 ms
YOLOX-s	614 ms	114 ms	46 ms	30 ms	15 ms	12 ms	13 ms
YOLOX-m	1368 ms	244 ms	75 ms	35 ms	18 ms	26 ms	17 ms

(1) Source of Detectors: HTC, https://arxiv.org/abs/1901.07518v2 [Last Access on 05.06.2023]; YOLOv3, https://arxiv.org/abs/1804.02767 [Last Access on 05.06.2023]; YOLOX, https://arxiv.org/abs/2107.08430 [Last Access on 05.06.2023]; YOLOv7, https://arxiv.org/abs/2207.02696 [Last Access on 05.06.2023]; YOLO-NAS, https://deci.ai/blog/yolo-nas-object-detection-foundation-model/ [Last Access on 05.06.2023].

We define a 3-step text recognition task: cropping text box, recognizing characters, and validating the result

DB SCHENKER

Input:
Image with bounding boxes

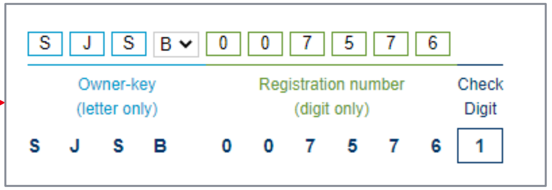
Step 1:
Crop Text Box

Step 2:
Recognize Characters

Step 3:
Validate Recognition

License Plate
S EC 70

ILU Code
SJSB 007576 1



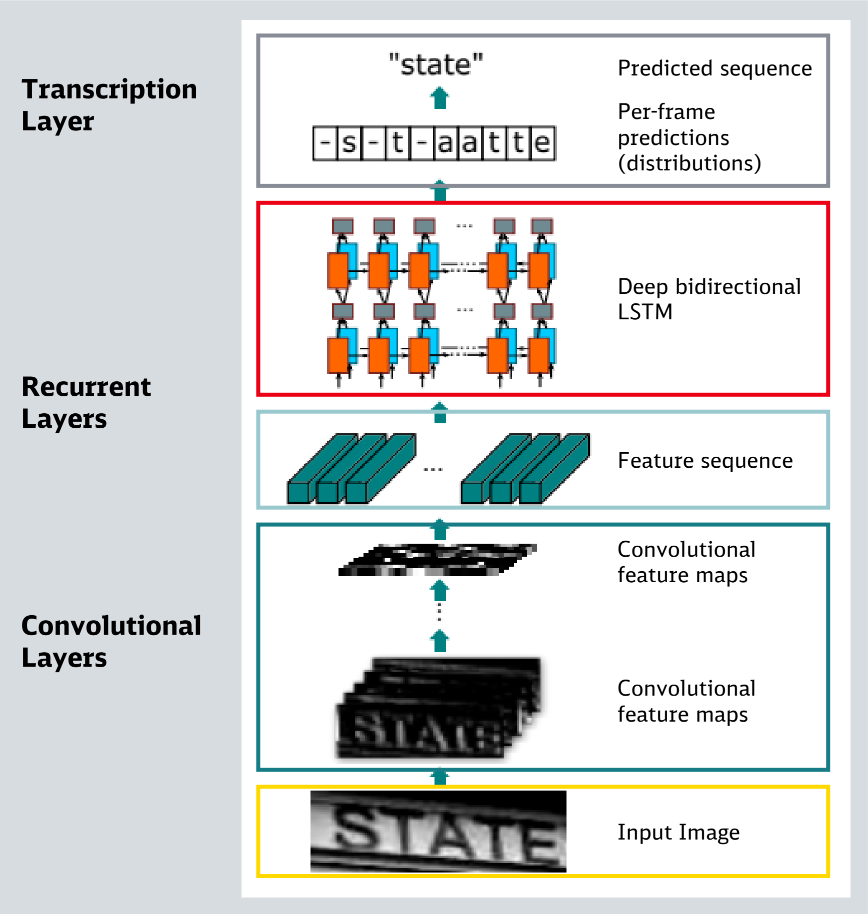
Focus on Text Recognition

Photos: Schenker AG; German License Plate Format & City/Region Codes, https://www.customeuropeanplates.com/german-license-plate-codes [Last Access on 05.06.2023];
ILU-Code - Calculate the check digit, https://www.ilu-code.eu/en/calculate-the-check-digit [Last Access 05.06.2023].

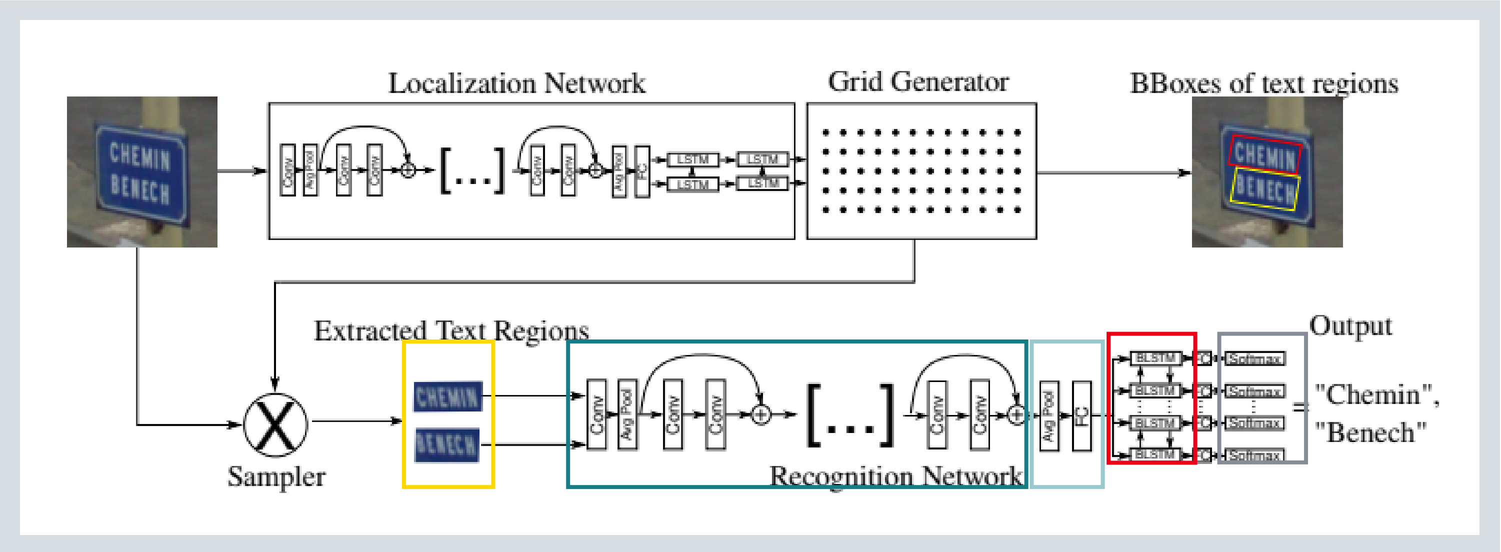
Text recognizers combine convolutional feature extraction and recurrent network-based sequential modeling

DB SCHENKER

CRNN: Convolutional-Recurrent Neural Network



SEE: Semi-Supervised End-to-End Scene Text Recognition



Key components of text recognizers:

- Cropped Text
- Convolutional Encoder
- Feature Sequences
- Recurrent Networks
- Decoder
- Attention Layers
- Position Encoders
- Language Model

Source: CRNN: Convolutional-Recurrent Neural Network: M. Ameryan and L. Schomaker: A limited-size ensemble of homogeneous CNN/LSTMs for high-performance word classification, in: Neural Computing and Applications 33 (2021); C. Bartz, H. Yang & C. Meinel: SEE: Towards Semi-Supervised End-to-End Scene Text Recognition. Proceedings of the AAAI Conference on Artificial Intelligence, 32(1) (2018), https://doi.org/10.1609/aaai.v32i1.12242 [Last Access on 05.06.2023].

We evaluated several text recognizers and measured their performance

DB SCHENKER

Experimental Phase

Text Recognizer[1]	Accuracy Per Image	Runtime Tesla T4 [ms]	Runtime RTX 3060 [ms]
SATRN-l	96.4%	165	384
NRTR	91.9%	238	521
RobustScanner	90.3%	82	76
SAR	88.6%	96	130
SATRN-m	85.2%	147	306
SATRN-s	82.8%	84	162
CRNN	41.1%	7	7
CRNN-TPS	40.8%	8	10

Details:
- Trained on 4773 images, tested on 1360 images
- Measured accuracy per ILU and LP images
- Measured runtime on Tesla T4 and GeForce RTX 3060

Fine-tuning and Evaluation on Edge

Takeaways:
- Most of the text recognizers are slower than 24 FPS (42 ms)
- SATRN-l has the highest accuracy
- **RobustScanner** has the best runtime/accuracy tradeoff

Parameter fine-tuning:
- The training of RobustScanner took 2 hours
- Deployed on NVIDIA Ampere A40
- Accuracy of License Plates: **96.05%**
- Accuracy of ILU Codes: **93.81%**
- Average running time: **51 ms**

(1) Source of Text Recognizers: CRNN (TPAMI'2016), https://arxiv.org/abs/1507.05717 [Last Access on 05.06.2023]; NRTR (ICDAR'2019), https://arxiv.org/abs/2007.07542 [Last Access on 05.06.2023]; RobustScanner (ECCV'2020), https://arxiv.org/abs/1806.00926 [Last Access on 05.06.2023]; SAR (AAAI'2019), https://arxiv.org/abs/1811.00751 [Last Access on 05.06.2023]; SATRN (CVPRW'2020), https://arxiv.org/abs/1910.04396 [Last Access on 07.06.2023].

We aggregate the image sequence per gate drive through, and apply majority voting among the highest confidence

DB SCHENKER

t

Image Sequence

DU ED 560 (58%)

DU EO 5050 (89%)

DU ED 5050 (93%)

DU ED 5050 (94%)

DU ED 5050 (93%)

Sequence Aggregation:

- N-best majority voting
- Order the recognition
- Select the top-N
- N=5 but at most 60% of the images
- Apply majority voting and select the best one

In this case:

- N=3
- Top-N recognitions:
  - DU ED 5050 (94%) #4
  - DU ED 5050 (93%) #3
  - DU ED 5050 (93%) #5
- Majority Vote: DU ED 5050

License Plate
DU ED 5050

Photos: Schenker AG

Showcase Video: Detecting LP and ILU-Code and recognizing their content

Every video frame is evaluated

Image number recognized LP confidence score

Result of current image

Best result for this Truck

Source: Schenker AG

We created a Grafana dashboard for visual investigation, debugging, and quality improvement

DB SCHENKER

Front Picture

Aggregator Results

Timestamp ↓	FrontPlate	RearPlate	FrontIlu	RearIlu	Direction	TruckType	OEM
2023-06-06 16:39:04	DUS	DUS			Exit_A	Tractor	OEM_Volvo
2023-06-06 16:38:43	PBV	PBV			Entry_A	Large_Truck	OEM_Mercedes_
2023-06-06 16:37:58	HST	HST			Entry_A	Empty_Swap_Body_...	OEM_MAN
2023-06-06 16:35:51	OBS	DUE			Entry_A	Large_Truck	OEM_Mercedes_
2023-06-06 16:34:49	DUM	DUM			Exit_A	Empty_Swap_Body_...	OEM_MAN
2023-06-06 16:33:22	DUE	DUE	SJSB00	SJSB00	Entry_A	Swap_Body_Truck	OEM_Mercedes_
2023-06-06 16:32:54	DUE	DUE			Entry_A	Large_Truck	OEM_Mercedes_
2023-06-06 16:32:15	DUM	DUM			Exit_A	Tractor	OEM_MAN
2023-06-06 16:30:55	DUE	DUE			Exit_A	Large_Truck	OEM_MAN

For Developers

- Real-time dashboard
- Validation of images vs. detection
- Recognized data: LP, ILU code, truck type
- Further information per detection

For Stakeholders

- Demonstration of early results and new features
- Proof of correctness and real-time capabilities
- Discussion of data protection questions (e.g., face blurring)

Source: Schenker AG

Real-time streaming pipeline is run on edge device and extracted information is pushed to the cloud

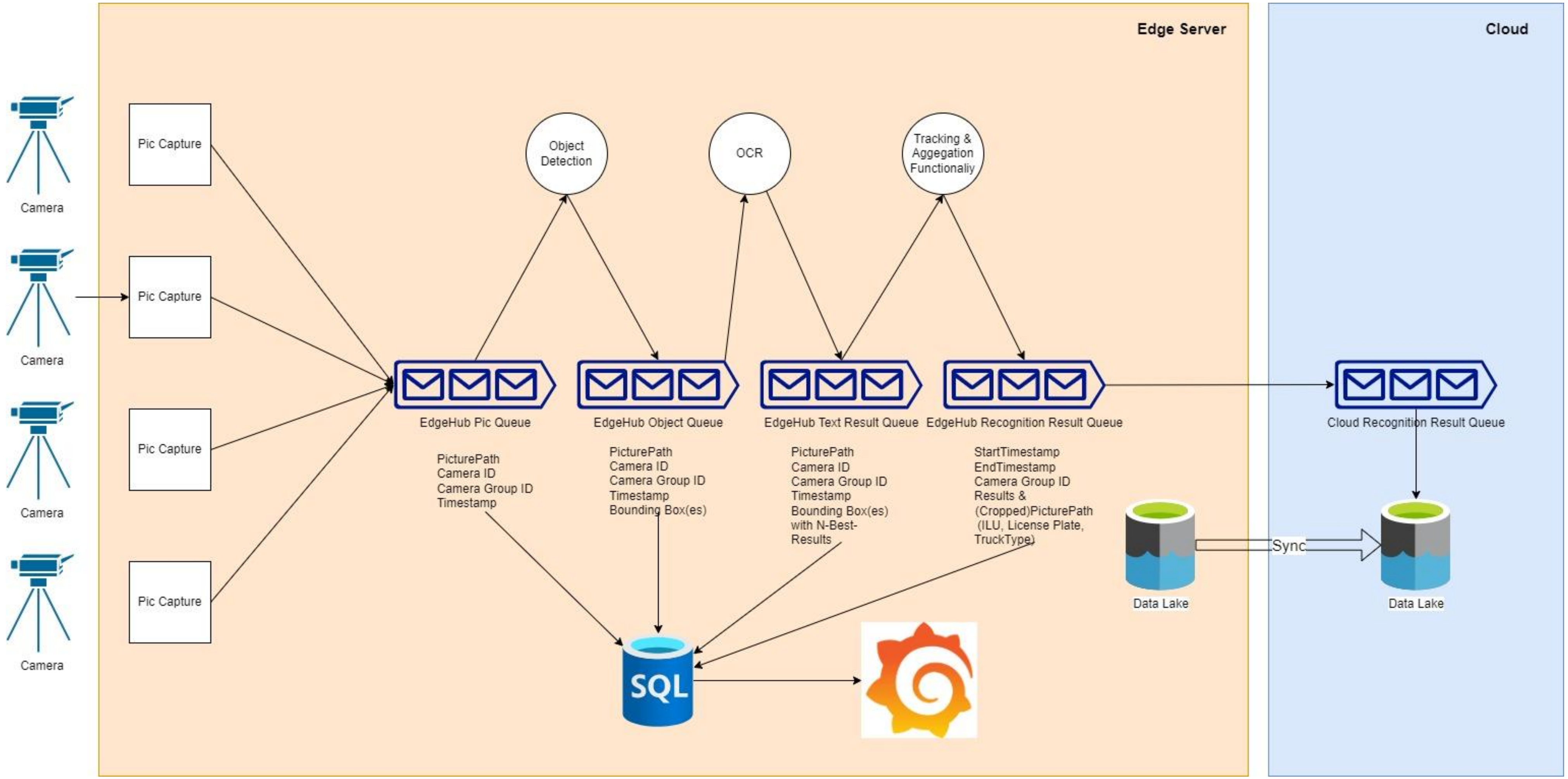


Image: Architecture diagram created by draw.io at Schenker AG

For the production-grade development, we used an open-source technology stack

DB SCHENKER

AI & Deep Learning

CVAT

PyTorch Lightning

PYTORCH

TensorFlow

MMCV

Keras

MMOCR

MMDetection

Development

OpenCV

jupyter

Experimentation & Monitoring

mlflow

kibana

TensorFlow

Infrastructure & Hardware

NVIDIA

GitLab

Dashboarding

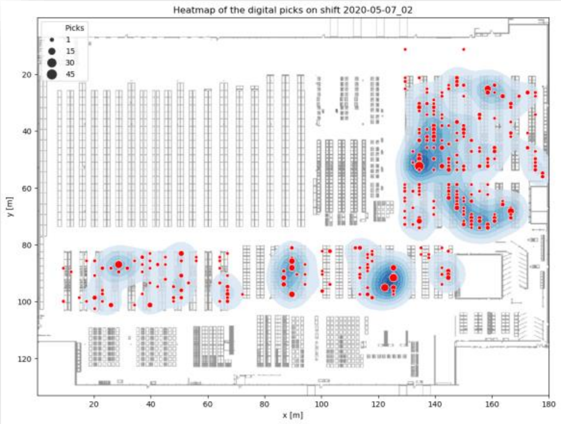
Power BI

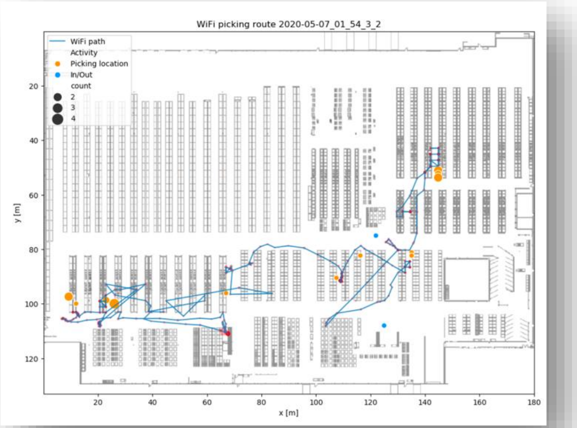
Grafana

Multi-camera object detection and tracking in warehouses

Productivity measurement in a warehouse is essential, but location tracking data is inaccurate yet

DB SCHENKER





- Items in a warehouse are managed by human workers with forklifts or pallet jacks and stored on pallets
- Operations of inventory takes significant human labor cost

- Location Analytics applies location data to derive insights into the activities
- It is essential to understand the efficiency of operations in various aisles (rows) and zones
- Distance, time, and density-based KPIs are applied

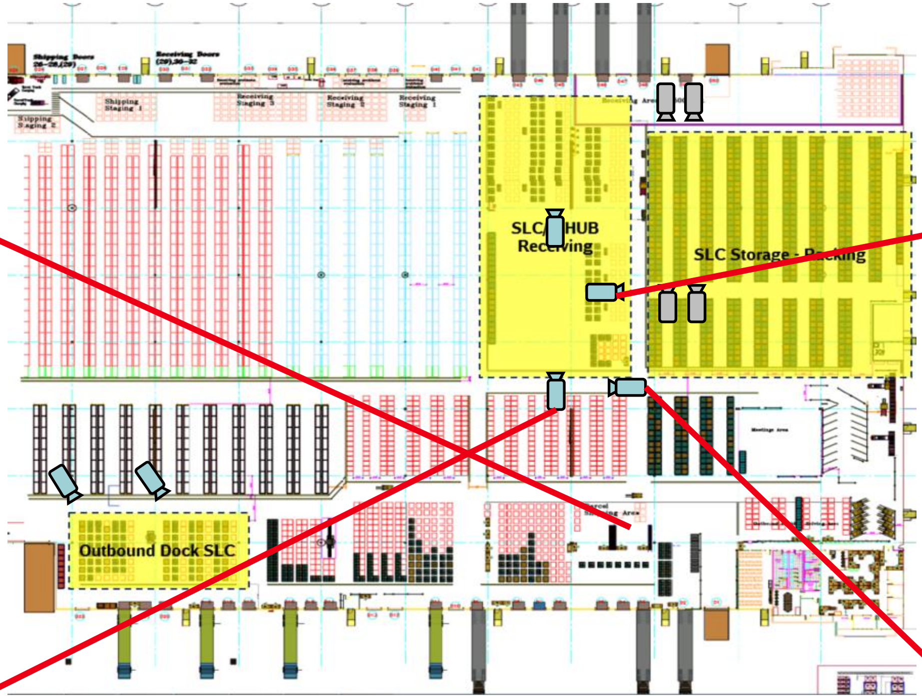
- Wi-Fi triangulation system tracks devices in real-time
- Signals are distorted due to metal and other materials, tracking can be inaccurate by 4-6 meters

Use Case: Find an alternative way of collecting location-based information on activities and provide more accurate tracking of items in the warehouse.

Images: Schenker AG

We have installed several cameras in one of our facilities

DB SCHENKER



Images: Schenker AG; Basler Area Scan Cameras, https://www.baslerweb.com/en/products/cameras/area-scan-cameras/ [Last Access on 05.06.2023].

We have identified 4 types of objects that we want to detect

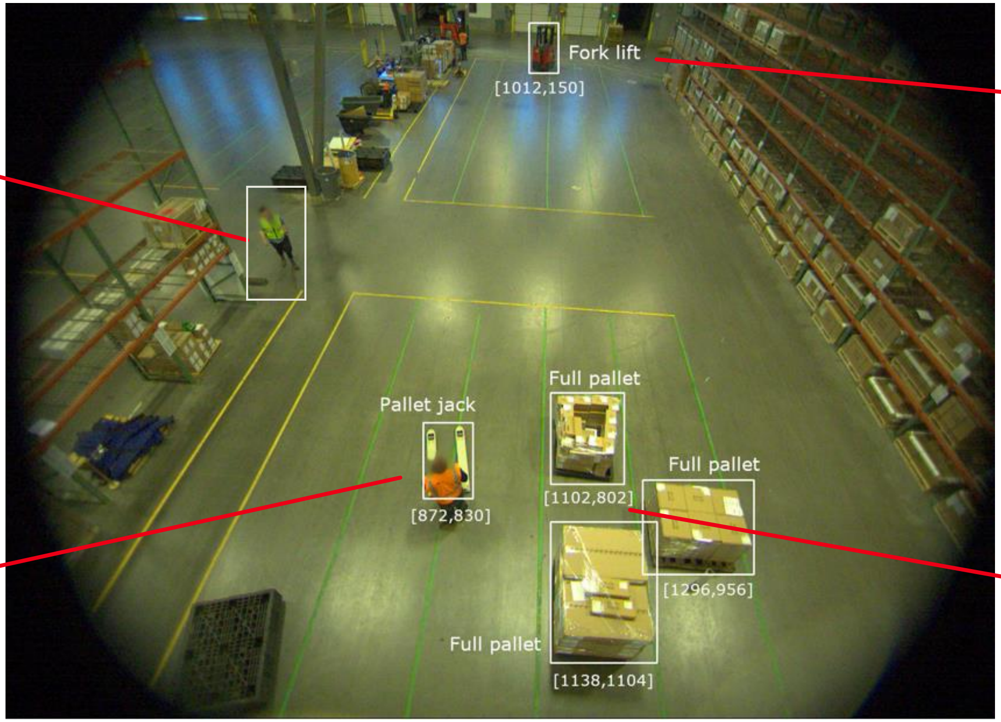
DB SCHENKER

1

Worker

2

Pallet jack



3

Forklift

4

Pallet

Source: Schenker AG

We selected YOLOv3 that provided a tradeoff between precision and latency for real-time application

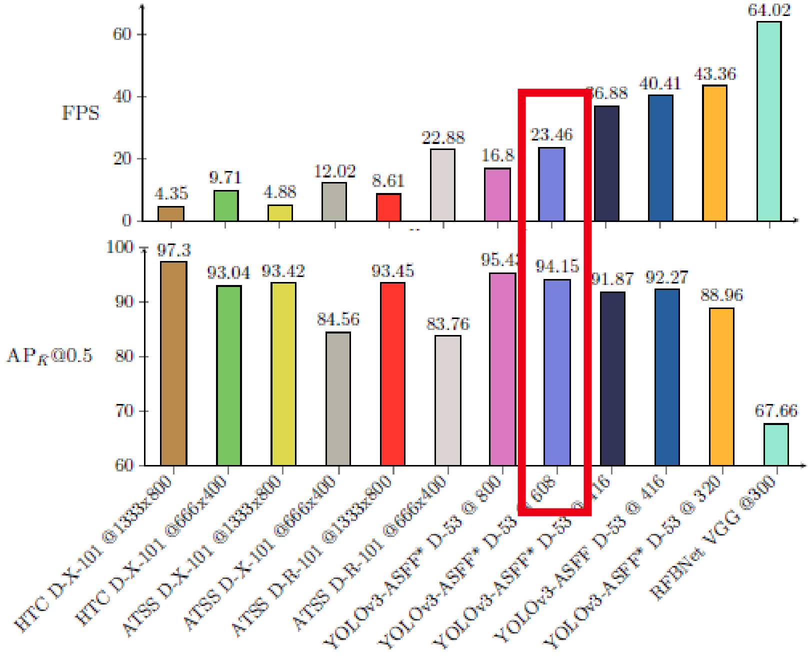
DB SCHENKER

Experimental procedure:

- Train and test set: Approx. **6000 images** on 4 aisles
- Open-source algorithms tested[1]: HTC, **YOLOv3**, ASFF, RFBNet
- Inference hardware: Nvidia RTX 2080 Ti
- Evaluation metrics: Precision, Recall, **Average Precision** and Runtime

Key findings:

- Higher accuracy requires higher latency (lower FPS)
- Best trade-off: YOLOv3-ASFF on 608x608 resized images
- Detected objects are precise **in 94% of the cases**
- Detection of pallet jack (Hubwagen) is the most difficult one



Objektklasse	**RC_k**	**PR_k**	**$AP^*_{k,s}$**	**$AP^*_{k,m}$**	**$AP^*_{k,l}$**	**$AP_k@0.5$**
Person	94.26	97.47	88.93	98.83	-	93.71
Palette	95.97	96.87	93.16	97.04	99.07	95.37
Hubwagen	92.71	97.41	90.68	97.14	-	92.33
Gabelstapler	95.51	98.15	87.85	96.12	96.07	95.19
					AP@0.5	94.15

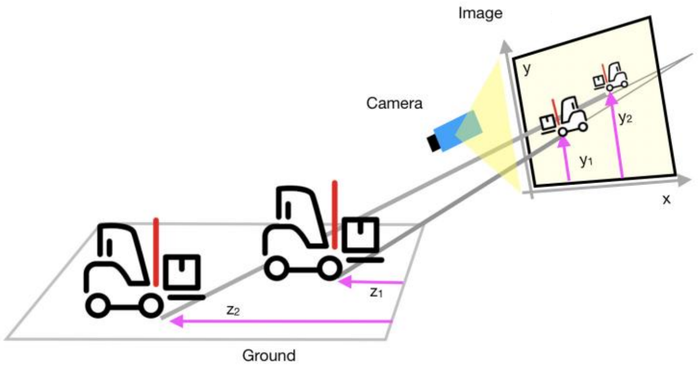
(1) Source of Object Detectors: HTC, https://arxiv.org/abs/1901.07518v2 [Last Access on 05.06.2023]; YOLOv3, https://arxiv.org/abs/1804.02767 [Last Access on 05.06.2023]; ASFF, https://arxiv.org/abs/1911.09516v2 [Last Access on 05.06.2023]; RFBNet, https://arxiv.org/abs/1711.07767 [Last Access on 05.06.2023].

Projection from the camera view to the ground is solved by reference coordinates and Planar Homography

DB SCHENKER

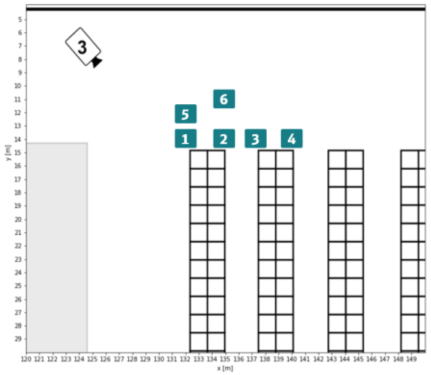
The Image Projection Problem

Project coordinates from the camera image to the ground



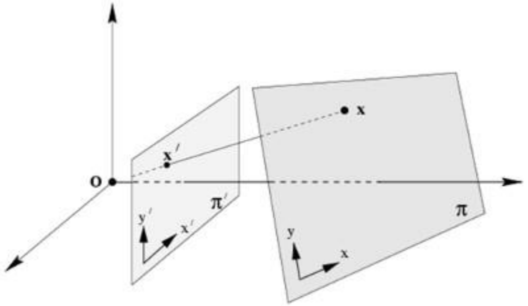
The Solution Step 1

We identified reference coordinates in various locations of the warehouse



The Solution Step 2

Planar Homography is a projection from one plane that is calculated by Direct Linear Transformation



$$s\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = H\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Direct Linear Transform

$$A = \begin{bmatrix} 0 & 0 & 0 & -u_1 & -v_1 & -1 & v_1'u_1 & v_1'v_1 & v_1' \\ u_1 & v_1 & 1 & 0 & 0 & 0 & -u_1'u_1 & -u_1'v_1 & -u_1' \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix}$$

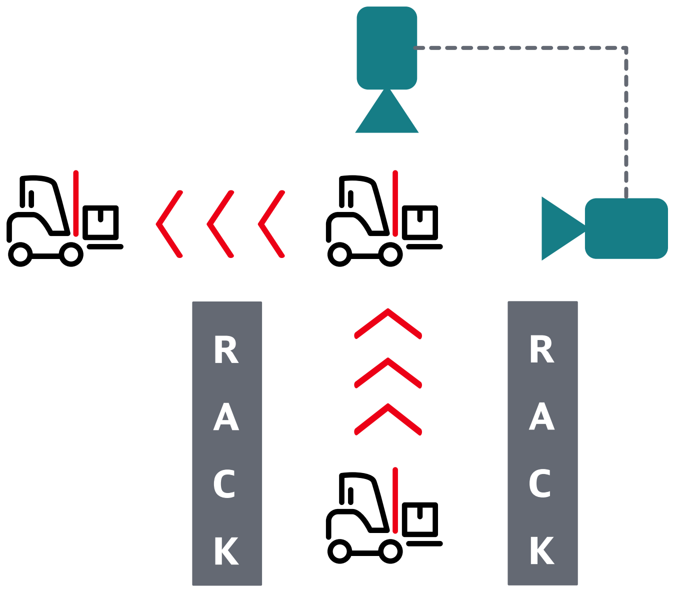
1. Build the matrix A from at least 4 point-correspondences $(u_i, v_i) \leftrightarrow (u'_i, v'_i)$
2. Obtain the SVD of A: $A = USV^T$
3. If S is diagonal with positive values in descending order along the main diagonal, then $\boldsymbol{h}$ equals the last column of V
4. Reconstruct H from $\boldsymbol{h}$

Source: Schenker AG; T. Opsahl: Estimating homographies from feature, in: Unik 4690, https://www.uio.no/studier/emner/matnat/its/TEK5030/v19/lect/lecture 4 3-estimating-homographies-from-feature-correspondences.pdf [Last Access on 05.06.2023].

Objects are globally tracked based on overlapping bounding boxes with Kalman Filter and greedy handover

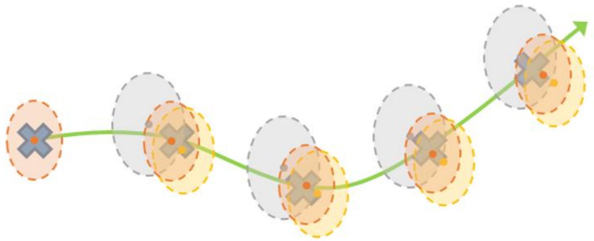
The Object Tracking Problem

Identify and track the same object by multiple camera without duplicated and noisy tracking

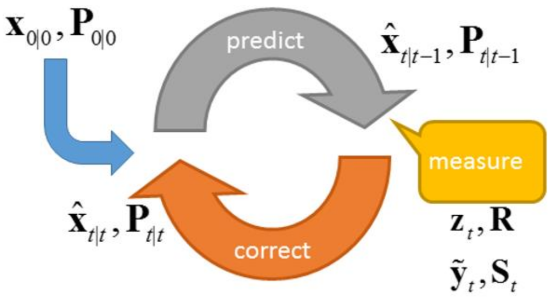


The Solution Step 1

Objects per camera are tracked based on overlapping bounding boxes per frame and denoised by Kalman Filter



Predict, measure, correct cycle iteratively estimates the state at each time step



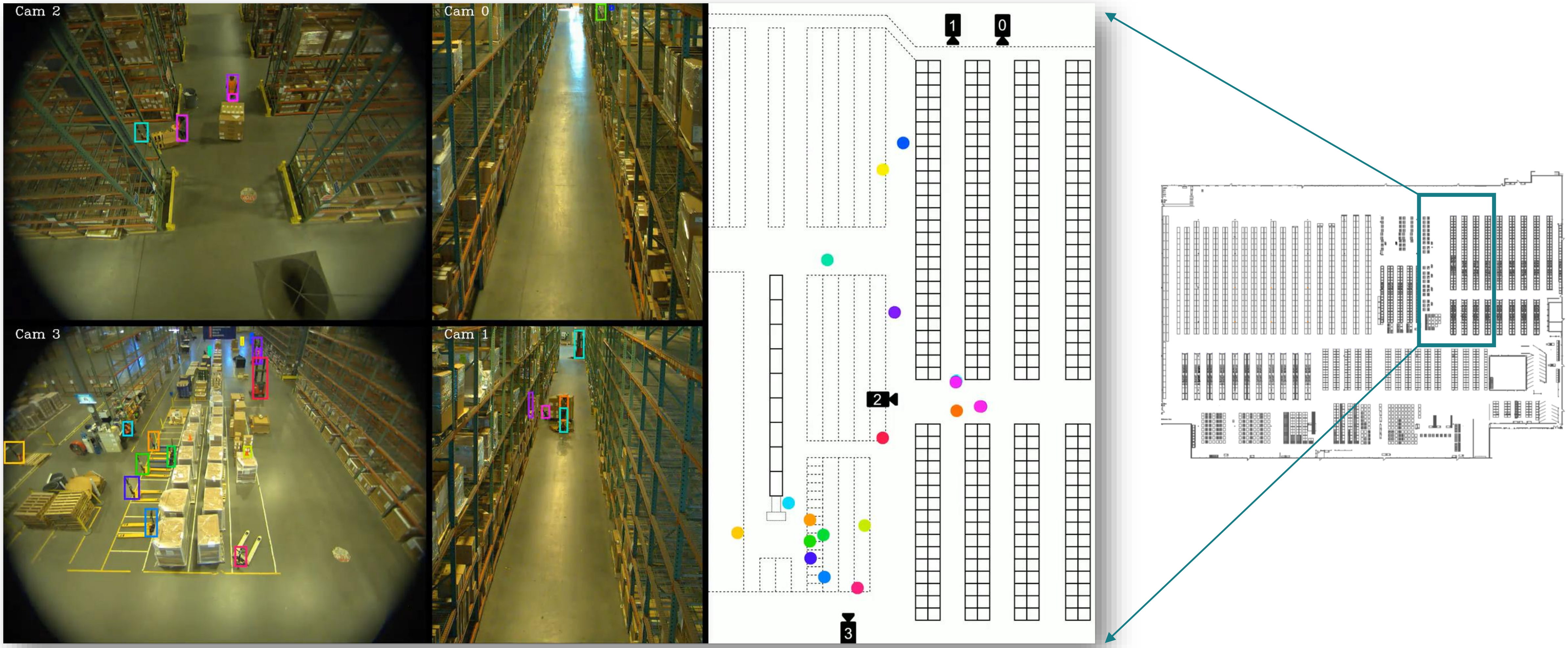
The Solution Step 2

We define a triplet-based greedy camera handover method:

1. Identify any object with the following **local track triplet**: camera id, object class id, local track id
2. A **global track** of an object is a list of local tracks triplets
3. In every frame, all local track triplets are checked if they co-occur in the global scene at least a given time period
4. If yes, it is checked if the mean distance in this period is below a minimum threshold
5. If yes, the two global tracks of the two local triplets are merged into one

Source: T. Opsahl: Estimating homographies from feature, in: Unik 4690, https://www.uio.no/studier/emner/matnat/its/TEK5030/v19/lect/lecture_4_3-estimating-homographies-from-feature-correspondences.pdf [Last Access on 05.06.2023]; Kalman Filter, https://en.wikipedia.org/wiki/Kalman_filter [Last Access on 05.06.2023]; D. Juric, Object Tracking: Kalman Filter with Ease, https://www.codeproject.com/Articles/865935/Object-Tracking-Kalman-Filter-with-Ease [Last Access on 05.06.2023].

A brief showcase of multi-camera object tracking with 4 cameras in bulk and rack areas



Source: Schenker AG

Overview of other use cases

Use Case: Tracking and measuring idle time of pallets in the inbound area of the warehouse

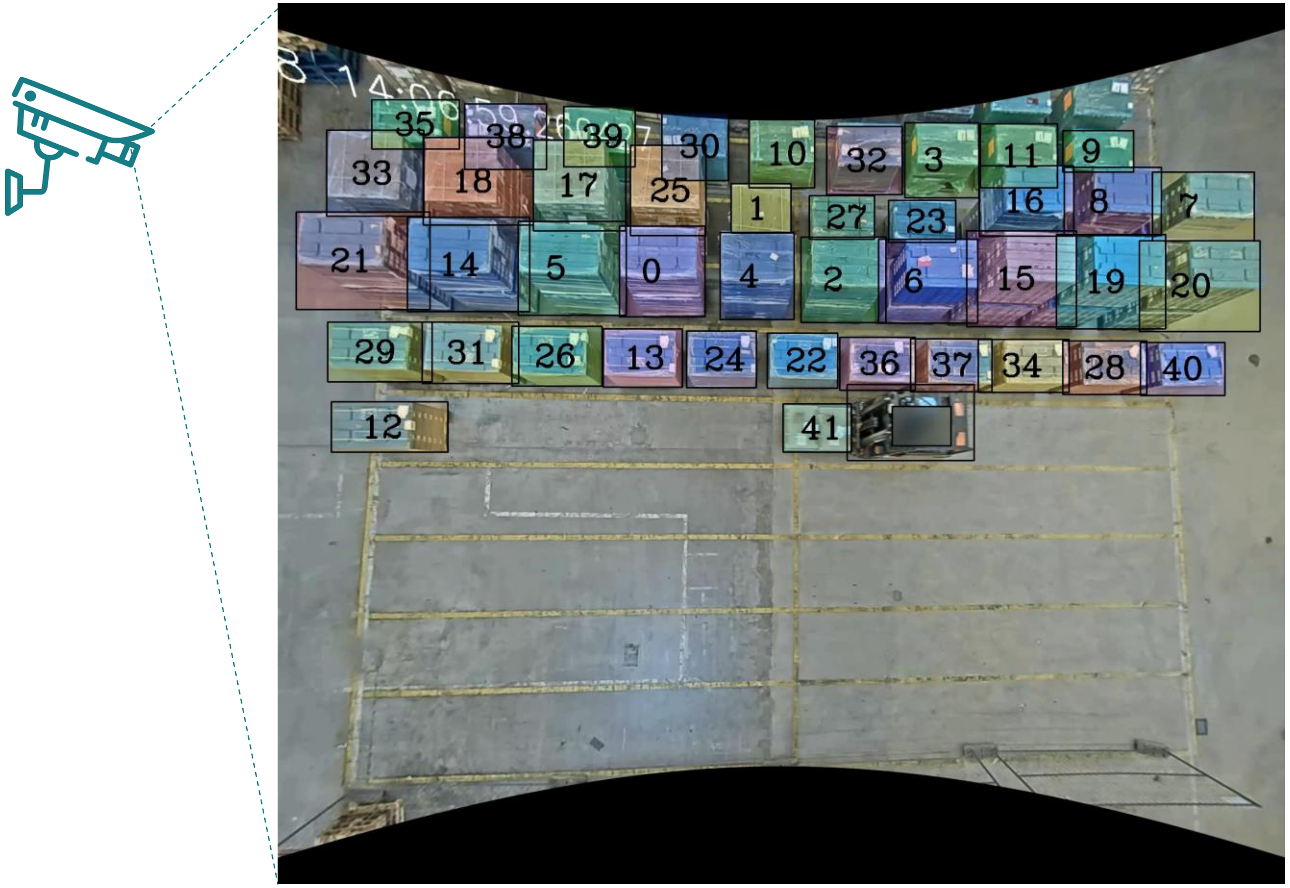
DB SCHENKER

The challenge:

- Incoming pallets are **stored in the inbound area**
- **Some pallets** have a long waiting time
- They reserve space and increase dock-to-stock time which leads to reduced handling efficiency

The idea:

- Use **video analytics** to detect and track pallets
- **Pallets are given a timestamp** to track dock-to-stock times
- Provide **put-away priorities** to forklift drivers and a dashboard of the **critical inbound KPIs**



Source: Schenker AG

Use Case: Estimating utilization of truck with sensors and Computer Vision

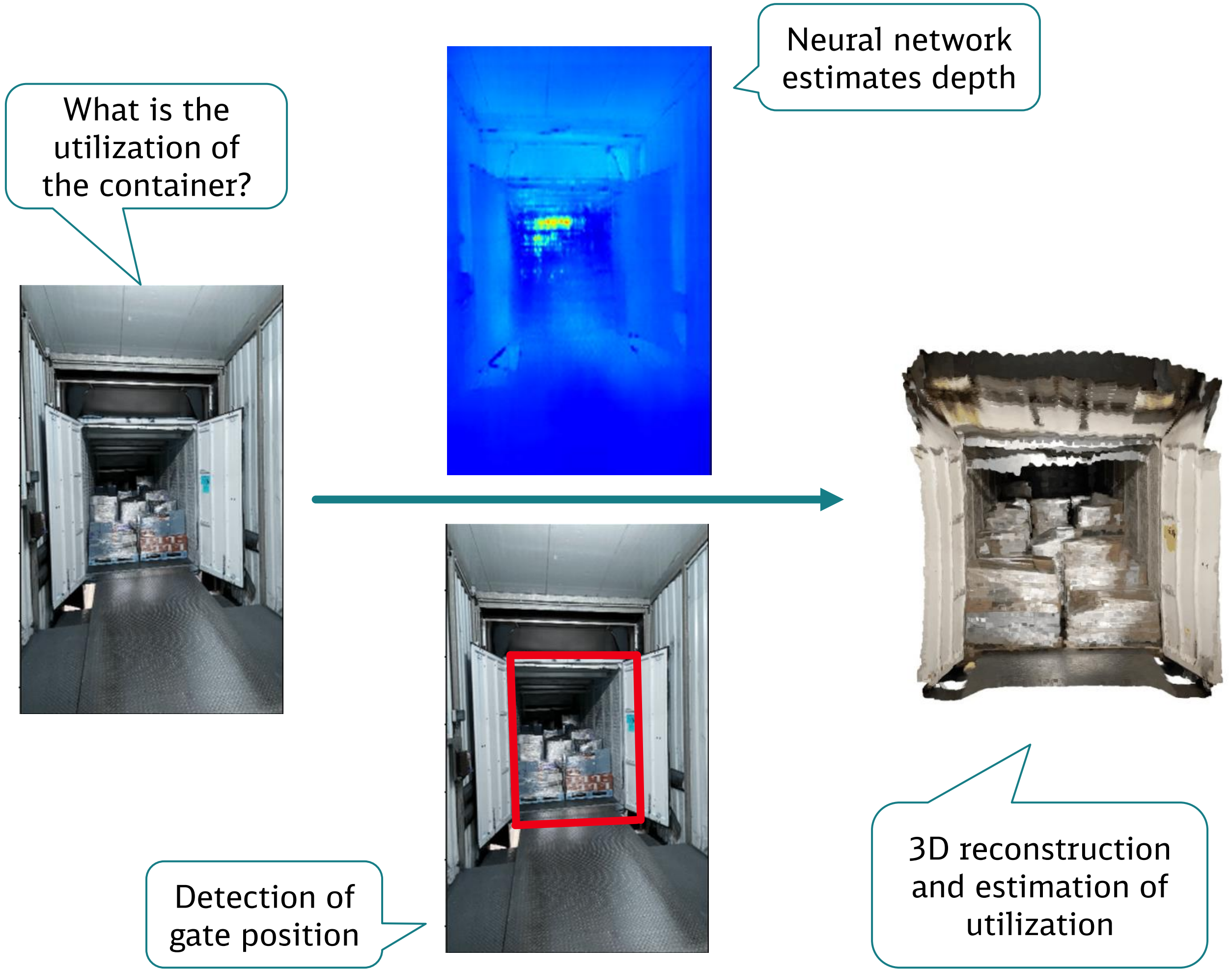
DB SCHENKER

The challenge:

- **Loading** trucks with cargo is still a **manual process**
- Before departure, the exact **utilization** of the truck is often **not registered**

The idea:

- Use computer vision to estimate depths in the image and **reconstruct the container** in 3D
- **Estimate utilization** based on free space in the container
- Provide utilization info for additional loading to **improve efficiency** of the transportation network

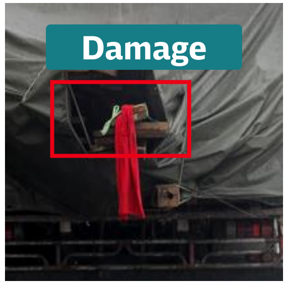


Source: Schenker AG

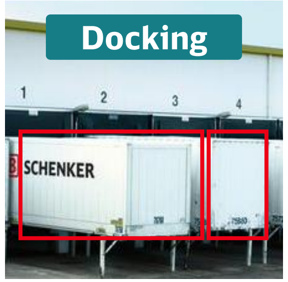
Beyond our current focus, we see a wide range of further use cases where video analytics can create business value

DB SCHENKER

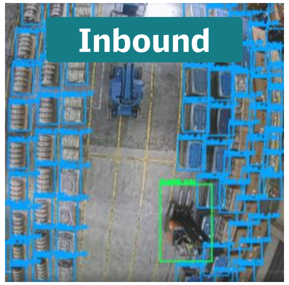
System Freight – at the gate



Land Transport



Contract Logistics



Overarching

Photos: Schenker AG and Shutterstock

Kudos to the Video Analytics Squad – a joint team of DB Schenker and Fraunhofer IML

DB SCHENKER

Data Science

Dr. Oliver Bredtmann

Thilo Bauer

Jan Basrawi

Engineering

Thomas Steiner

Raja A. Charuvil

Daniel Wallner

Fraunhofer IML

Maximilian Otten

Dr. Oliver Urbann

Business and Mgmt.

Dr. Konrad Steiner

Dr. David Zibriczky

DB SCHENKER

Contact

DB Schenker
Global Data Strategy and Analytics

Dr. David Zibriczky
Director Data Science

Kruppstr. 4
45128 Essen, Germany
david.zibriczky@dbschenker.com

dbschenker.com
blog.dbschenker.com