

## Document Control Sheet

1. ISBN or ISSN	2. type of document (e.g. report, publication) Final report
3. title Final Report according to NKBF 98 Safe use of automated shuttle vehicles in urban traffic through supportive infrastructure networking (Shuttle2X)	
4. author(s) (family name, first name(s))	5. end of project 30.11.2025
	6. publication date 2026
	7. form of publication Electronic (pdf)
8. performing organization(s) (name, address)	9. originator's report no. SONAH GmbH, Jülicher Straße 236, 52070 Aachen
	10. reference no. 19S22001F
	11. no. of pages 30
12. sponsoring agency (name, address)  Bundesministerium für Wirtschaft und Energie 53107 Bonn	13. no. of references 6
	14. no. of tables 0
	15. no. of figures 14
16. supplementary notes Finanziert von der EU (NextGenerationEU) Bundesministerium – Wirtschaft und Energie	
17. presented at (title, place, date)	
18. abstract Development of a scalable hybrid system architecture (Edge/Cloud) for obstacle detection. Implementation of resource-efficient Edge inference using MobileNetV2 for real-time detection on low-power hardware. Validation at test sites in Karlsruhe and Bad Wimpfen, including V2X integration and Collective Perception Messages (CPM).	
19. keywords Edge Computing, V2X, Collective Perception Service (CPS), Object Detection, Intelligent Infrastructure, Autonomous Driving, Smart City	
20. publisher	21. price

## Berichtsblatt

1. ISBN oder ISSN	2. Berichtsart (Schlussbericht oder Veröffentlichung) Schlussbericht
3. Titel  Shuttle2x – Schlussbericht gem. NKBF 98 Sicherer Einsatz von automatisierten Shuttle-Fahrzeugen im städtischen Verkehr durch unterstützende Infrastruktur-Vernetzung (Shuttle2X)	
4. Autor(en) [Name(n), Vorname(n)] Kleine, Malte Grimm, Thomas	5. Abschlussdatum des Vorhabens 30.11.2025
	6. Veröffentlichungsdatum 27.02.2026
	7. Form der Publikation Elektronisch (PDF)
8. Durchführende Institution(en) (Name, Adresse)	9. Ber. Nr. Durchführende Institution SONAH GmbH, Jülicher Straße 236, 52070 Aachen
	10. Förderkennzeichen 19S22001F
	11. Seitenzahl 30
12. Fördernde Institution (Name, Adresse)  Bundesministerium für Wirtschaft und Energie 53107 Bonn	13. Literaturangaben 6
	14. Tabellen 0
	15. Abbildungen 14
16. Zusätzliche Angaben Finanziert von der EU (NextGenerationEU) Bundesministerium – Wirtschaft und Energie	
17. Vorgelegt bei (Titel, Ort, Datum)	
18. Kurzfassung  Im Rahmen von Shuttle2X wurde eine skalierbare hybride Systemarchitektur (Edge-Inferenz/Cloud-Analyse) zur Hinderniserkennung entwickelt. Fokus war die Optimierung der Edge-Inferenz mit MobileNetV2 für Echtzeit-Detektion bei geringem Energieverbrauch. Validierung erfolgte an den Standorten Karlsruhe und Bad Wimpfen inklusive V2X-Integration (CPM).	
19. Schlagwörter Edge Computing, V2X, Collective Perception Service (CPS), Objekterkennung, Intelligente Infrastruktur, Autonomes Fahren, Smart City	21. Preis
20. Verlag	

# S

# O

# NAH

S O NAH GmbH | Jülicher Str. 236 | 52070 Aachen

TEL: +49 151 22092646  
MAIL: grimm@sonah.tech  
WEB: www.sonah.tech

TÜV Rheinland Forschungs- und  
Innovationsmanagement GmbH

Projektträger Bodengebundene  
Verkehrstechnologie (PT Bvt)

Datum/ Date	27.02.2026
Förderprojekt	Shuttle2X
Förderkennzeichen	19S22001F

## Förderprojekt Shuttle2X - Schlussbericht gem. NKBF 98

*Sicherer Einsatz von automatisierten Shuttle-Fahrzeugen im städtischen Verkehr durch unterstützende Infrastruktur-Vernetzung (Shuttle2X)*

SONAH GmbH

Jülicher Straße 236, 52070 Aachen

Thomas Grimm / CEO & CO-FOUNDER



**Finanziert von der  
Europäischen Union**  
NextGenerationEU

Gefördert durch:



Bundesministerium  
für Wirtschaft  
und Energie

aufgrund eines Beschlusses  
des Deutschen Bundestages

GESCHÄFTSFÜHRER / CEO  
UST-ID / VAT  
GERICHTSSTAND

Thomas Grimm  
DE307902180  
AG Aachen / HRB 20573

KONTOINHABER  
IBAN  
BIC

SONAH GMBH  
IDE26 3905 0000 1072 7995 52  
AACCSD33

<b>Förderprojekt Shuttle2X - Schlussbericht gem. NKBF 98</b>	<b>1</b>
<b>I) Kurze Darstellung</b>	<b>3</b>
1. Aufgabenstellung	3
2. Voraussetzungen, unter denen das Vorhaben durchgeführt wurde	3
3. Planung und Ablauf des Vorhabens	3
4. Wissenschaftlicher und technischer Stand	4
Verwendete Fachliteratur und Dokumentationsdienste	4
5. Zusammenarbeit mit anderen Stellen	5
<b>II) Eingehende Darstellung</b>	<b>6</b>
1. Verwendung der Zuwendung und erzielte Einzelergebnisse im Vergleich zu den Zielen.	6
Technische Darstellung. Sensorinfrastruktur, ML und CPM im Vorhaben Shuttle2X	6
Aufbau und Einsatzorte	6
Datenerfassung und ML-Pipeline	10
Modellarchitektur und Quantisierung	13
Algorithmen. Detektion, Tracking und Flächenbewertung	14
Depth Perception. Verfolgter Ansatz ohne Einsatz in der Finalversion	23
2. Wichtigste Positionen des zahlenmäßigen Nachweises	26
3. Notwendigkeit und Angemessenheit der geleisteten Arbeit	26
4. Voraussichtlicher Nutzen und Verwertbarkeit (Fortschreibung des Verwertungsplans)	26
5. Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen	27
6. Erfolgte oder geplante Veröffentlichungen	27
<b>IV) Kurzfassung (Berichtsblatt)</b>	<b>28</b>

## I) Kurze Darstellung

### 1. Aufgabenstellung

Im Rahmen der ersten Projektphase lag der Fokus auf der konzeptionellen Entwicklung und laborseitigen Implementierung der Analyse-Infrastruktur. Ziel war die Schaffung einer skalierbaren Systemarchitektur, die sowohl lokale Ressourcen (Edge) als auch zentrale Kapazitäten (Cloud) effizient nutzt. Diese Arbeiten bildeten die softwareseitige Grundlage für den späteren Einsatz an den verschiedenen Standorten. Die Echtzeit-Anforderungen, insbesondere für die Interaktion mit autonomen Shuttles, und die spätere Validierung an den Teststandorten waren dabei von vornherein leitend (AP2).

### 2. Voraussetzungen, unter denen das Vorhaben durchgeführt wurde

Voraussetzungen: Die strategische Planung und Konzeptionierung des Vorhabens basierte maßgeblich auf einer fundierten Expertise im Einsatz von Sensorsystemen im öffentlichen Raum. Die langjährigen Erfahrungswerte bei der Installation technischer Komponenten in komplexen Verkehrsumgebungen ermöglichten es, bereits in der Entwurfsphase spezifische Herausforderungen wie wechselnde Umwelteinflüsse, datenschutzrechtliche Anforderungen und infrastrukturelle Restriktionen proaktiv zu berücksichtigen (AP2). Erfahrung aus fast zehn Jahren Betrieb von Sensoren in unterschiedlichen Ländern im öffentlichen Raum konnte erfolgreich für die konzeptionelle Ausarbeitung und den Aufbau genutzt werden. Ein weiterer Fokus der vorbereitenden Arbeiten lag auf der Entwicklung einer leistungsfähigen Labeling-Pipeline zur Sicherstellung der für das Training der Detektionsmodelle erforderlichen Datenqualität und -quantität (AP2).

### 3. Planung und Ablauf des Vorhabens

Zunächst wurde ein zweistufiges Verarbeitungsmodell (Edge-Inferenz und Cloud-Analyse) entwickelt und eine universelle Pipeline zur Datenverarbeitung entworfen. Anschließend erfolgte die Simulation und Validierung im Labor, bevor am Standort Karlsruhe die Hardware (IoT-System mit Recheneinheit, zwei USB-Kameras, 4G-Funkmodul) an der Infrastruktur der Straßenbeleuchtung installiert wurde. Ein weiterer Schwerpunkt lag auf der Optimierung der Edge-Inferenz unter Berücksichtigung der Echtzeit-Anforderungen und der geringen Rechenkapazitäten. Anschließend erfolgte die physische Umsetzung und Validierung an den Teststandorten Karlsruhe und Bad Wimpfen (AP1, AP7).

## 4. Wissenschaftlicher und technischer Stand

Für die lokale Objekterkennung wurde das Modell MobileNetV2 als Basis gewählt; State-of-the-Art-Architekturen wie YOLOv8 bzw. YOLOv9 wurden für einen Einsatz in einer Cloudumgebung evaluiert und erwiesen sich dort als die Modelle mit der höchsten Lokalisierungs- und Klassifizierungsgenauigkeit. Der Fokus des Projekts lag auf Edge-Computing. Die Überführung der Detektionsergebnisse in standardisierte Collective Perception Messages (CPM) und die Anbindung an die V2X-Infrastruktur bauen auf den etablierten Kommunikationsformaten für kooperative Wahrnehmung auf (AP3).

### Verwendete Fachliteratur und Dokumentationsdienste

- ETSI TS 103 324: Collective Perception Service (CPM).
- ETSI EN 302 637-2: Cooperative Awareness Message (CAM), Basic Service.
- ETSI TR 103 562: Collective Perception.
- Huang et al.: V2X Cooperative Perception for Autonomous Driving: Recent Advances and Challenges, arXiv:2310.03525, 2023.
- Cooperative Infrastructure Perception, arXiv:2207.08930.
- Sandler et al.: MobileNetV2: Inverted Residuals and Linear Bottlenecks, CVPR 2018.
- Wang et al.: YOLOv9: Learning What You Want to Learn Using Programmable Gradient Information, arXiv:2402.13616, 2024.
- Ultralytics YOLOv8-Dokumentation.
- TensorFlow Object Detection API (Dokumentation und Quellcode; im Vorhaben erweitert: Data-Loading für PNG, verschiedene Farbräume).
- TensorFlow Lite (TFLite), Dokumentation.
- MiDaS / DPT: Monokulare Tiefenschätzung (Dokumentation bzw. zugehörige Fachliteratur).
- SAM2 (Segment Anything Model 2), Dokumentation bzw. zugehörige Publikation.
- Einschlägige Survey- und Methodenliteratur zu monocular depth estimation.
- Liu et al.: 3D Object Detection for Autonomous Driving: A Comprehensive Survey, International Journal of Computer Vision 131, 2023, DOI: 10.1007/s11263-023-01790-1.
- BAST: Fahrzeugklassifizierung, TLS (Erfassung 8+1).
- CVAT (Computer Vision Annotation Tool), Dokumentation.
- COCO (Common Objects in Context), Format-Dokumentation.
- VisDrone-Datensatz, UAVDT-Datensatz, M-30 / M-30-HD-Datensatz, TrafficCam-Datensatz
- AICity Challenge (nicht nutzbar, fehlende Labels).
- ROS 2 (Robot Operating System 2), Dokumentation.
- Google Maps / Satellitenbilder (Referenz für Kalibrierung).
- WaveBee / RSU: Roadside Unit, produkt- bzw. projektspezifische Dokumentation.
- DSGVO (Verordnung (EU) 2016/679).

## **5. Zusammenarbeit mit anderen Stellen**

Ein Sensorstandort wurde in Kooperation mit den Stadtwerken Karlsruhe realisiert. Die Implementierung der CPM und die Anbindung an die V2X-Infrastruktur erfolgten im engen Austausch mit den Projektpartnern, insbesondere Bosch und Keysight (AP1).

## II) Eingehende Darstellung

### 1. Verwendung der Zuwendung und erzielte Einzelergebnisse im Vergleich zu den Zielen.

#### Technische Darstellung. Sensorinfrastruktur, ML und CPM im Vorhaben Shuttle2X

Im Folgenden werden Aufbau und Betrieb der Sensorinfrastruktur, die Verarbeitung der Bilddaten, die eingesetzten Modelle und Algorithmen sowie die Aussendung von Collective Perception Messages (CPM) an das autonome Shuttle in einem durchgängigen Zusammenhang dargestellt. Abschließend wird ein verfolgter, aber nicht in die Finalversion übernommener Ansatz (Depth Perception) zusammengefasst (AP3).

#### Aufbau und Einsatzorte

Die stationäre Installation aus SONAH-Sensor und WaveBee Roadside Unit (RSU) bildet die Voraussetzung dafür, CPM-Daten an das autonome Shuttle zu übertragen. Das Setup wurde zunächst im SONAH-Büro erprobt.



[Abb. 1 und 2: Testaufbau im SONAH-Büro.]

Insbesondere der Power Draw der WaveBee in Kombination mit dem Low-Power-Device und den verbauten Akkus war fraglich und musste ausführlich getestet werden, um die Laufzeit und Stabilität unter Realbedingungen abzusichern. Es wurde ein maßgeschneiderter Rahmen gefertigt, sodass WaveBee und SONAH-Sensor an derselben Halterung angebracht werden können (AP7).



[Abb. 3: WaveBee-RSU samt Halterung, bereit für den Aufbau.]

So lässt sich die Einheit kompakt an der vorgesehenen Aufhängung (z. B. an der Laterne) montieren. Der Aufbau vor Ort erfolgte durch SONAH-Mitarbeiter mit Unterstützung eines Steigers (Hubarbeitsbühne).



[Abb. 4: SONAH-Mitarbeiter auf der Hebebühne beim Aufbau in Bad Wimpfen.]

Für die Akkus war ebenfalls eine maßgeschneiderte Lösung erforderlich. An vielen Standorten speisen Straßenlaternen nur nachts Strom (Nachtstrom). Für einen durchgängigen Betrieb der Sensoren sind Akkus daher zwingend. Die Bleiakkus befinden sich in einem schmalen, langen Aluminiumschacht, der sowohl innerhalb der Laterne als auch außerhalb, witterungsausgesetzt, angebracht werden kann (AP7).

Das Vorhaben nutzte zwei Standorte für die Shuttle-relevante Infrastruktur sowie einen dritten für den Use Case Static Parking. Karlsruhe diente als Test-Site. Hier wurden der Aufbau des Datensets, das Vertesten der Modelle und Algorithmen unter Realbedingungen sowie die technische Validierung des Setups durchgeführt. Es wurde keine RSU verbaut, sodass in Karlsruhe keine CPM vom Shuttle empfangen werden konnten. Die gewonnenen Erkenntnisse flossen in den Aufbau am Standort mit RSU-Anbindung ein. Bad Wimpfen war der Standort mit vollständiger V2X-Anbindung und Kommunikation mit dem autonomen Shuttle (RSU vor Ort).



[Abb. 5: SONAH-Sensor samt WaveBee-RSU nach erfolgreichem Aufbau in Bad Wimpfen.]

Beide Standorte waren für den Nachweis der Systemstabilität unter Realbedingungen und die Demonstration der kooperativen Wahrnehmung zentral. In der Planung wurden drei Leitungsumgebungstypen adressiert: On-Demand Shuttles im Stadtquartier, Innerbetrieblicher Personentransport sowie Gemischter betrieblicher und öffentlicher Transport. Die entwickelten Algorithmen und die Sensorintegration sind für diese drei Typen ausgelegt. An den Standorten Karlsruhe und Bad Wimpfen konnten sie entsprechend vertestet werden. Ergänzend wurde ein Sensor zum Parkplatz am Gelände der Robert Bosch GmbH geschickt und von Bosch in Kombination mit einer RSU aufgebaut.



[Abb. 6: Konfiguration der Perception Areas auf dem Bosch-Parkplatz (Kamera 1 von 2).]  
Die Stromversorgung dort muss manuell zugeschaltet werden und dient nur zu Testzwecken bzw. als Proof of Concept, dass der SONAH-Standard-Use-Case Static Parking mit V2X-CPM verbunden werden kann. Dieser Sensor hat keinen Dauerstrom. Stand heute sind alle drei Sensoren noch installiert. Zwei sind online und liefern Daten (je einer in Karlsruhe und einer in Bad Wimpfen). Der Sensor auf dem Bosch-Parkplatz ohne Dauerstrom ist nicht dauerhaft in Betrieb (AP7).

Die Planung umfasste das Scouting geeigneter Standorte, die Abstimmung mit den Stadtwerken sowie die Erstellung von ROI- und Setup-Plänen (Region of Interest, Aufbau- und Installationsplanung). Ein Aufbau an öffentlicher Infrastruktur ist stets mit Bürokratie und erforderlichen Nachweisen verbunden. Die frühe und kontinuierliche Kommunikation mit den Stadtwerken war dafür entscheidend. In Karlsruhe wurde der Sensorstandort in Kooperation mit den Stadtwerken Karlsruhe realisiert. Die Hardware wurde an der bestehenden Infrastruktur der Straßenbeleuchtung installiert, was die kontinuierliche Energieversorgung sicherstellte. Die enge Abstimmung ermöglichte eine praxisnahe Integration in die Lichtmastinfrastruktur und einen reibungslosen Rollout. Der Aufbau in Bad Wimpfen baute technologisch und organisatorisch auf den Erkenntnissen des Karlsruher Setups auf und profitierte ebenfalls von der etablierten Zusammenarbeit mit den kommunalen Partnern. An beiden Orten floss die bestehende SONAH-Expertise ein. Erfahrung aus fast zehn Jahren Betrieb von Sensoren im öffentlichen Raum in unterschiedlichen Ländern half, anspruchsvolle Randbedingungen (wechselnde Umwelteinflüsse, datenschutzrechtliche Anforderungen, infrastrukturelle Restriktionen) bereits in der Planung zu berücksichtigen. SONAH nahm an allen Regelterminen und Workshops teil und stand dem Konsortium beratend zur Verfügung. Besondere Schwerpunkte lagen auf der Anforderungsanalyse (AP2) und der Definition des hybriden Systemkonzepts (AP3). Der Betrieb an zwei unterschiedlichen Standorten machte die Komplexität verschiedener Umgebungen erneut deutlich. Die Teststrecke in Karlsruhe weist ein sehr geringes Verkehrsaufkommen auf. Okklusionen kamen kaum bis gar nicht vor und die Fahrzeugvielfalt war geringer. In Bad Wimpfen traten dagegen deutlich mehr Verkehrsteilnehmer und vielfältigere Szenarien auf. Die quantisierten TFLite-Modelle haben

ein hartes Limit an Objekten, die pro Frame ausgegeben werden. In Bad Wimpfen wurde dieses Limit bei hohem Verkehrsaufkommen erreicht. Die Region of Interest (ROI) wurde daher verkleinert, damit nicht mehr Objekte im Sichtfeld liegen als das Modell ausgeben kann. Andernfalls werden überzählige Detektionen im Modelloutput abgeschnitten und theoretisch erkennbare Objekte erscheinen nicht in den CPM (AP5).

### **Datenerfassung und ML-Pipeline**

Die Verarbeitung von Bilddaten im Vorhaben folgt einer festen Pipeline. Die daraus resultierenden Modelle erfassen auf dem Sensor Objekte und Flächenzustände und bilden damit die Grundlage für die CPM-Aussendung an das autonome Shuttle. Auf dem Sensor wird als erstes eine Privacy-Maske (Datenschutzmaske) angewendet. Schwarze Areale verdecken alles außer der relevanten Region of Interest (ROI). Technisch ist die Maske ein Polygon mit beliebig vielen Eckpunkten (konvex oder konkav), das von einem SONAH-Mitarbeiter explizit konfiguriert wird. Die Schwärzung erfolgt auf dem Sensor im ersten Schritt der Imager-Pipeline. Ungeschwärzte Bilder können nicht gespeichert werden und nachgelagerte Algorithmen haben ausschließlich Zugriff auf geschwärzte Kamerabilder. Das heißt, noch bevor Algorithmen das Bild auf dem Gerät verarbeiten und bevor Bilddaten zu Trainingszwecken versendet werden, sind alle Bereiche außerhalb der ROI bereits ausgeblendet. Nummernschilder sind aus der Perspektive des Sensors in Kombination mit der Auflösung der Kameras nicht klar erkennbar. Ebenfalls liegen Gesichter von Fußgängern auf dem Bürgersteig außerhalb der ROI. Identifizierbare Gesichtsmarkale sind bei Fußgängern, die den Straßenübergang queren, aufgrund der Entfernung und der Kameraauflösung nicht erkennbar (AP5).

Trainingsdaten inklusive Metadaten wurden an ein abgesichertes Backend in Deutschland übermittelt. Zu den Metadaten zählen der Zeitstempel des Bildes, die Bildhelligkeit, die Object-Detection-Inferenzergebnisse auf dem Gerät sowie die zum Aufnahmezeitpunkt gültigen Parameter der Algorithmen (z. B. ROI, Schwellenwerte). Diese Transparenz unterstützt die Nachvollziehbarkeit der Datenerfassung und die spätere Auswertung im Kontext der DSGVO. Nach bestem Wissen und Gewissen wurde die DSGVO eingehalten. Es werden weder personenbezogene Daten verarbeitet noch gespeichert und es besteht kein Interesse daran. Ein erfreulicher Nebeneffekt des Schwärzens ist, dass die Bilddaten geringer im Umfang sind und sich einfacher per Mobilfunk nach Deutschland übertragen lassen. Die Bilddaten wurden bereits auf dem Gerät nach Relevanz vorsortiert. Aus DSGVO-Gründen wie auch aus rein wirtschaftlichen Motiven (Übertragung verursacht Kosten, Annotation verursacht Kosten) sollen so wenig Daten wie nötig, aber so viele wie erforderlich erfasst werden. Einzelnen Samples wie auch Sequenzen wurde daher auf dem Device ein Score zugewiesen. Daten mit hohem Score wurden bevorzugt gesammelt und ans Backend geschickt (AP5). Es war außerordentlich wichtig, ein diverses Datenset zu erstellen. Computer Vision in einer unkontrollierten, wechselnden Umgebung benötigt ein großes und diverses Datenset, sowohl

hinsichtlich der Umgebung (Wetter, Witterung, Licht, Jahreszeit) als auch hinsichtlich der Objekte.



[Abb. 7: Witterung, Lichtverhältnisse und Schmutz auf dem Kameradom erschweren die Detektion.]

Die Vielfalt der Verkehrsteilnehmer in Formen und Farben ist praktisch unbegrenzt. Je diverser das Trainingsdatenset ist, desto wahrscheinlicher sind gute Inferenzergebnisse bei noch nicht gesehenen Objekten und Umgebungen. Zusätzlich zu den am Sensor und über das Backend gewonnenen Daten wurden öffentlich verfügbare Datensätze analysiert und einbezogen. Dies war als Arbeitspaket angelegt: Vor der Nutzung stand eine Recherche geeigneter Quellen; die Pipeline wurde so ergänzt, dass die eingebundenen Daten in einem homogenen Format vorlagen. Da die Datensätze unterschiedlich gelabelt sind und teils abweichende Klassenbezeichnungen haben, floss in die Aufbereitung und Vereinheitlichung erheblich Aufwand. Trotzdem blieb der Weg günstiger, als die gleiche Datenmenge extern labeln zu lassen, und bot eine vergleichsweise einfache Möglichkeit, die eigenen Datensätze zu erweitern und robuster zu machen. Genutzt wurden unter anderem VisDrone (Drohnenperspektive, kleine Objekte in urbanen Szenen, teils nah an Kundenanwendungen wie Parkflächen), UAVDT (Drohne, schräg, Fahrzeuge auf Straßen), M-30 bzw. M-30-HD (fixe Mastperspektive, Autobahn, für Zähler-/Counter-Szenarien geeignet) und TrafficCam (Überwachungsperspektive an Kreuzungen, überlappende Fahrzeuge). Der

AICity-Challenge-Datensatz war aufgrund fehlender Labels nicht nutzbar. Die übrigen genannten Datensätze waren zwar nicht in jeder Hinsicht ideal (z. B. abweichende Perspektive, andere Klassen oder Auflösung), erwiesen sich aber als hilfreich, um das Trainingsset zu vergrößern und Modelle bzw. Pipelines zu testen (AP3).

Die Annotation erfolgte in CVAT im COCO-Format. Das ist ein gängiger Standard, der Kompatibilität mit den bereits bestehenden Projekten und Datensätzen ermöglicht. Das Labeling wurde von einem externen Dienstleister durchgeführt. Dabei mussten ein sicherer Zugang zu den Daten sowie eine technisch präzise Beschreibung der Labeling-Regeln gewährleistet werden. Gelabelt wurde ursprünglich anhand der BASt-Fahrzeugklassifizierung nach TLS (Erfassung 8+1).



[Abb. 8: Ein häufig vorkommendes Fahrzeug, in der BASt-Klassifizierung allerdings nicht klar von LKWs getrennt.]

Es zeigte sich jedoch rasch, dass dieses Schema bei Weitem nicht alle für das Projekt erforderlichen Klassen abdeckt. So fehlen darin etwa Fußgänger oder Fahrradfahrer. Es wurde daher eine erweiterte Aufteilung der Fahrzeug- und Objektklassen entwickelt. Für diesen Schritt musste umfangreiches Trainingsmaterial neu klassifiziert und annotiert werden. Auch semi-statische Objekte können relevant sein, beispielsweise Mülltonnen, die teilweise an oder auf der Straße stehen. Aufgrund von Qualitätsproblemen wurde im Verlauf des Vorhabens ein neuer (deutscher) Partner für das Labeling gewonnen. Teilweise musste Material mehrmals annotieren lassen, da mit dem ersten Anbieter keine zufriedenstellende Qualität erreicht wurde. Hinzu kamen verschärfte Anforderungen, um die Fehlerrate zu senken und den Sicherheitsanforderungen des Projekts gerecht zu werden. Die genauen Labeling-Regeln iterativ zu verbessern und eindeutig zu kommunizieren, um die semantische Integrität der Datensätze sicherzustellen, erforderte mehrere Anläufe und erwies sich in der Praxis als herausfordernd. Die gelieferten Labels wurden anschließend vom Vorhaben validiert und mit den Bilddaten verbunden (AP5).

## Modellarchitektur und Quantisierung

Im Vorhaben kommt ein Objekterkennungsmodell (Object Detection) zum Einsatz. Es liefert für jedes erkannte Objekt eine Bounding-Box (Position im Bild), eine Klassenbezeichnung und einen Konfidenzscore. Als Basis dient MobileNetV2 in der TFLite-Variante. Es wurde gewählt, weil es auf einer ARM-CPU mit Abstand das schnellste gängige Object-Detection-Modell ist und damit die Echtzeitanforderungen auf dem Edge-Device erfüllbar bleiben. Echtzeit-Inferenz ist dabei nicht in allen Produkten des Vorhabens nötig (z. B. Static Parking kommt mit geringeren Frameraten aus), aber für die Shuttle-Kommunikation zwingend. Das Shuttle benötigt fortlaufend aktuelle CPM-Daten zu Objekten und Flächenzuständen. Die Architektur nutzt depthwise separable Convolutions (Tiefenweise und punktwise getrennte Faltungen). Zuerst wird pro Kanal räumlich gefaltet (depthwise), anschließend werden die Kanäle linear kombiniert (pointwise). Diese Aufteilung reduziert Rechen- und Speicherbedarf deutlich und führt auf ARM-Architekturen zu besonders guter Laufzeitperformance bei vergleichsweise geringem Genauigkeitsverlust. Als Detektionskopf („head“) wird ein SSD-Head (Single Shot Detector) verwendet. SSD erzeugt Vorhersagen in einem Durchlauf über mehrere Feature-Maps und ist damit schneller als mehrstufige Ansätze wie FPN (Feature Pyramid Networks), was für die Echtzeit-Inferenz auf dem Sensor vorteilhaft ist (AP4).

Das Modell wurde vollständig auf Integer-Gewichte quantisiert (Full-Integer-Quantisierung). So lässt sich die Inferenz auf der ARM-CPU ohne Floating-Point-Einheiten effizient ausführen und die Latenz sowie der Speicherbedarf weiter senken. Das ist eine Voraussetzung für die Echtzeit-Aussendung von CPM an das Shuttle. Der Genauigkeitsverlust durch die Quantisierung beträgt in der Praxis etwa 1 bis 2 % (bezogen auf den F-Score) und steht in einem günstigen Verhältnis zum Geschwindigkeitsgewinn der Inferenz (typisch 2x bis 4x, abhängig von Width und Depth des Modells). Zudem kann nach der Quantisierung ein größeres Modell eingesetzt werden. Die höhere Parameteranzahl gleicht die Genauigkeitsverluste in der Regel aus. Diese Einschätzung deckt sich mit der praktischen Erfahrung von SONAH im IoT-Markt. Es wird stets quantisiert, unabhängig von der gewählten Modellarchitektur. Einerseits wurden False Negatives (übersehene Objekte) im Betrieb stärker gewichtet als False Positives, im Einklang mit der Safety-First-Orientierung für die Shuttle-Kommunikation. Andererseits wurde im Training das Hard-Negative-Mining erweitert, um schwierige Hintergründe und Verwechslungskandidaten gezielt einzubeziehen und die Präzision zu verbessern. Die statische Perspektive des Sensors war dafür zwingend erforderlich. MobileNetV2 in der verschlankten Variante (1/4 height und width) hat bei Weitem nicht genug Parameter, um in dynamischen Umgebungen gute Ergebnisse zu erzielen, wenn zusätzlich zur großen Vielfalt der zu erkennenden Objekte noch eine wechselnde Perspektive und ein wechselnder Hintergrund hinzukommen. Die feste

Kameraposition reduziert die Varianz des Problems so weit, dass das schlanke Modell unter den gegebenen Randbedingungen ausreichend performant bleibt (AP4).

Bei ausreichend großem Datenset sowie gutem Pre- und Post-Processing erreicht MobileNetV2 eine ausreichende Lokalisierungsgenauigkeit. Die Bounding-Boxen liegen an der richtigen Position. Bei der Klassifizierung stößt das Modell hingegen an seine Grenzen. Insbesondere bei einer größeren Label-Map bleiben die vorhergesagten Klassen über die Zeit wenig konstant. Die Klassifizierung ist über die zeitliche Dimension hinweg instabil, dasselbe Objekt kann über aufeinanderfolgende Frames hinweg wechselnde Labels erhalten. Heuristisch lässt sich nachbessern, indem die Klasse über eine Sequenz hinweg stabilisiert wird (je häufiger ein Objekt mit derselben ID in einer Spur gesehen wird, desto zuverlässiger lässt sich die Klasse festlegen). Dadurch kann sich die gemeldete Klasse eines Objekts mit fester ID im Lauf der Zeit jedoch ändern. Das führt für CPM-Konsumenten wie das Shuttle zu wechselnden Klassenzuordnungen pro Objekt. Besonders fehleranfällig ist die Klassifizierung beim Einfahren in die ROI, wenn ein Objekt nur teilweise sichtbar ist, aber bereits eine CPM ausgesendet werden soll. Zusätzlich wurden low-level-Anpassungen in der TensorFlow Object Detection API vorgenommen. Typischerweise arbeiten Object-Detection-Modelle und -Pipelines mit JPEG-Bildern im RGB-Farbraum. Die Data-Loading-Pipeline wurde so erweitert, dass auch PNG-Bilder in verschiedenen Farbräumen verarbeitet werden können. Dadurch kann der Kamerastrom (YUV) oder eine RGB-Konvertierung direkt in das Modell eingespeist werden, ohne zusätzliche Umwegkonvertierungen auf dem Device. Die Verwendung von PNG und flexiblen Farbräumen in der Trainingspipeline führt allerdings zu erheblich längeren Trainingszeiten, da PNG-Dekodierung und ggf. Farbraumumrechnungen aufwendiger sind als bei JPEG/RGB. Die Modelle wurden trainiert, „im Labor“ (im SONAH-Büro am Fenster) getestet und anschließend deployed. Mit Hilfe des auf dem Device berechneten Relevanz-Scorings sowie der Auswertung von Remote Reality Checks wurden Lücken im Datenset identifiziert und durch gezielte Nachbesserung geschlossen. Trainingsset und Modelle wurden iterativ verbessert und on-the-edge unter Realbedingungen evaluiert (AP5).

### **Algorithmen. Detektion, Tracking und Flächenbewertung**

Auf Basis der beschriebenen Modelle wurden im Vorhaben Algorithmen für die Objekterkennung, die Objektverfolgung und die flächenbasierte Zustandsbewertung entwickelt. Sie bilden die Grundlage für die Aussendung von CPM und decken die Anwendungsfälle Echtzeit-Shuttle-Kommunikation sowie Static Parking ab. Damit liefern sie die technische Basis dafür, dem autonomen Shuttle über V2X zusätzliche Hindernis- und Flächeninformationen bereitzustellen, inklusive solcher Objekte, die für das Shuttle möglicherweise verdeckt sind, und z. B. durch die Klassifikation von Perception Areas (frei/nicht frei) zur Vermeidung von Gridlocks beizutragen. Die Detektion und die Verfolgung wurden zunächst als eigenständige Varianten (MVPs) realisiert. Der finale

Algorithmus verbindet beide Logiken in einer gemeinsamen Pipeline. Die Erkennung erfolgt stets innerhalb einer vordefinierten Region of Interest (ROI). Diese ROI wird so gewählt, dass sowohl die Berechnung des Bewegungsvektors als auch die ML-Modell-Ergebnisse zu brauchbaren Ergebnissen führen. Die Auslegung der ROI ist damit ein zentraler Parameter für die Qualität von Detektion und Tracking (AP5).

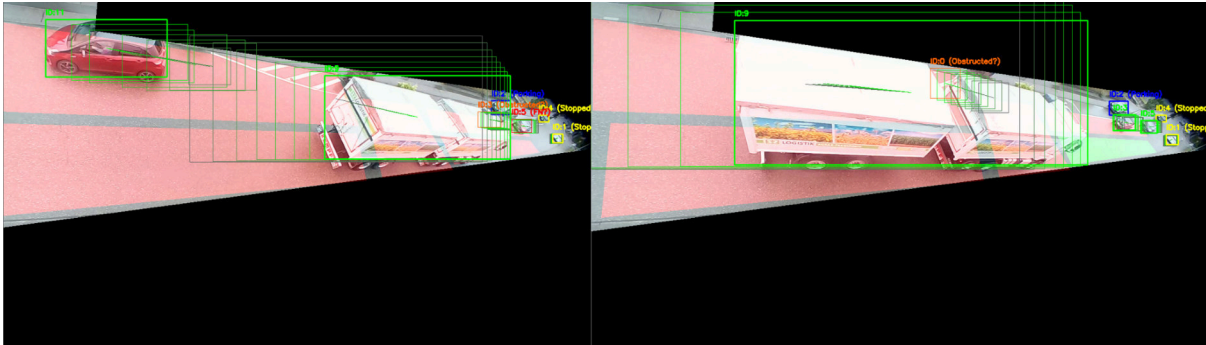
Die flächenbasierte Zustandsbewertung arbeitet auf vordefinierten Straßensegmenten (Zonen), die als Polygone in normalisierten Bildkoordinaten konfiguriert sind. Die Zonen werden über eine Homographie-Matrix in Weltkoordinaten (ENU) umgerechnet und als Rechtecke mit Mittelpunkt, Halbachsen und geodätischer Orientierung abgelegt, entsprechend dem CPM-Format für Perception Regions. Nach Extraktion der Region of Interest und Helligkeitsbasierter Bildkorrektur kommt ein konfigurierbares ML-Detektionsmodell zum Einsatz. Detektionen werden nach Score und Klasse gefiltert (typisch Score > 0,5, Klasse Fahrzeug). Für jede gültige Bounding-Box werden die Eckpunkte per inverser Homographie in den Weltraum projiziert. Daraus werden Mittelpunkt (x, y), Länge, Breite und eine Standardhöhe abgeleitet. Die z-Koordinate wird auf 0 gesetzt (Ground-Plane-Annahme). Anschließend wird für jede Zone geprüft, ob mindestens eine Bounding-Box mit ihr überlappt (Polygon-Schnitt). Die Zuordnung Zone zu Objekt-IDs wird für die CPM genutzt. Die logische Ausgabe umfasst Perceived Objects (Objekt-ID, Position, Zeitstempel, optional Dimensionen, Klassifikation abhängig vom Modell) und Perception Regions (Rechtecke mit Anzahl und IDs der darin liegenden Objekte). Damit kann das Shuttle Bereiche als frei oder blockiert bewerten und z. B. Gridlocks vermeiden. Die genaue Zuordnung von Objekten zu Perception Regions ist eine Näherung. Sie könnte durch Depth Perception (Tiefenwahrnehmung) verbessert werden. Besonders der seitliche Blick des Sensors auf die Fahrbahn macht es schwierig, präzise zu entscheiden, ob ein Objekt in der stadteinwärts oder stadtauswärts führenden Spur liegt. Je größer ein Objekt ist, desto unschärfer wird diese Aussage. Daher wurden die finalen Regions so ausgelegt, dass sie stets über beide Spuren gelegt werden. Der Algorithmus weist Objekte einem „Abschnitt“ zu, nicht einer einzelnen „Spur“ (AP5).

Szenarien und Randfälle (Spurwechsel, Falschparker, Fußgänger): Ein eindeutiges Matching von Fahrzeugen zu einzelnen Spuren ist über große Teile der ROI, abhängig von Perspektive und Größe des Fahrzeugs, sehr fehleranfällig. Das Spurwechselszenario lässt sich mit der gewählten Abschnitlogsik daher nur begrenzt abbilden. Stehende Fahrzeuge (vom Modell als Bounding-Box ausgegeben), die sich nicht in einer definierten Parkfläche befinden, werden als Falschparker gewertet. Ab wann ein stehendes Fahrzeug als Falschparker zählt, kann per Schwellwert konfiguriert werden (u. a. Mindestdauer). Auf Algorithmus-Seite besteht zudem die Möglichkeit, Falschparker nur in bestimmten Arealen (z. B. am Straßenrand) zu zählen. Das ist vor allem auf großen Freiflächen sinnvoll. An den Standorten Karlsruhe und Bad Wimpfen wurde davon ausgegangen, dass alle stehenden Fahrzeuge außerhalb

expliziter Parkflächen als Falschparker gelten. Ebenso wie Fahrzeuge können Fußgänger einer Fläche zugeordnet werden. Ein Fußgänger am Fußgängerüberweg kann den benachbarten Streckenabschnitt als „nicht frei“ setzen. In Bad Wimpfen bildet der Fußgängerüberweg die relevante Engstelle. Die Sensoren unterstützen dort die Detektion von Gegenverkehr sowie die Bewertung von Wartebereichen und Flächenzuständen (frei/nicht frei), sodass das Shuttle Engstellenszenarien besser einordnen kann (AP5). Dies muss vorkonfiguriert werden und ist nicht immer eindeutig. In der Praxis führte das zu vielen False Positives. Aus Datenschutzgründen wurde der Relevanz-Score in der ML-Pipeline für Samples mit Personen bewusst stark reduziert, um so wenig wie möglich mit personenbezogenen Daten zu arbeiten. Diese Abwägung hat sich im Projekt herausgestellt. Eine hohe Gewichtung für Szenarien mit Personen hätte eine aufwendige, szenarienspezifische Annotation erfordert (für hohe Genauigkeit müssten exakt die im Sichtfeld des Sensors vorkommenden Szenarien gelabelt werden). Dem stand die bewusste Vermeidung einer hohen Gewichtung von Personendaten gegenüber. Für die Objekterkennung von Personen wurden öffentliche Datensätze genutzt. Eine feingranulare Anpassung an die konkreten Sensorstandorte wurde nicht angestrebt (AP5).

Die Verfolgungslogik erweitert die Detektion um eine zeitliche Zuordnung der Objekte über mehrere Frames. Es wird dieselbe ML-Detektion verwendet, ergänzt um einen Feature-basierten Tracker, der Objekte über aufeinanderfolgende Bilder hinweg zuordnet. Vor der Tracker-Zuweisung können Bounding-Boxen nach minimaler Fläche gefiltert, überlappende Boxen zusammengeführt und bei geringer Helligkeit verworfen werden. Der Tracker liefert für jede aktive Spur die Trajektorie (Zentrumspunkte, optional extrapolierte Punkte) sowie die aktuelle Bounding-Box. Die Trajektorienpunkte werden über die Homographie in Weltkoordinaten transformiert. Die Geschwindigkeit wird aus der Differenz der letzten beiden Weltpunkte und dem Zeitintervall  $\Delta t$  berechnet. Die Beschleunigung ergibt sich aus der Änderung der Geschwindigkeit über die letzten drei Punkte. Beide Größen werden in kartesischen Koordinaten ( $x, y, z = 0$ ) für die CPM bereitgestellt. Im Vergleich zur reinen Detektion stehen hier prädiktive Trajektorien- und Kinematikinformationen im Vordergrund. Perception Regions können in dieser Variante entfallen. An den beiden Aufbauten (gerade Strecken, keine Kreuzung) wird der Bewegungsvektor entlang der Fahrbahn gewichtet; das verbessert die Vorhersage belegter Flächen und die Fehlerkorrektur und glättet leichte Modellabweichungen (verschobene oder anders skalierte Bounding-Boxen). Für die Gridlock-Vermeidung und die Pfadplanung des Shuttles gilt die Vereinfachung: Fährt das Shuttle von links nach rechts, wird angenommen, dass es weiter nach rechts fährt. Bewegungen diagonal zur Fahrbahn sind sicherheitsrelevant; für sie wird der Confidence-Threshold konfigurierbar herabgesetzt (Abwägung: lieber False Positive als ein getracktes, potentiell gefährliches Objekt zu verlieren). Das Konzept gilt nur entlang gerader Straßen; in Kurven wird die Mathematik komplexer, an Kreuzungen versagt es. Eine zentrale logische Erweiterung ist die Fortführung von Tracks auch dann, wenn das Modell

temporär kein Objekt detektiert (False Negative) oder das Objekt vermutlich verdeckt ist (Okklusion). Statt den Track abzubrechen, wird die Spur anhand der letzten bekannten Trajektorie bzw. einer Bewegungsprädiktion fortgeschrieben. Lücken in der Detektionsfolge werden mit vorhergesagten Positionen gefüllt („fill gaps with predictions“). So bleibt die Information „Objekt vorhanden“ für das Shuttle erhalten, bis der Track wieder durch eine Detektion bestätigt wird oder nach einer konfigurierbaren Zeitspanne verfällt.



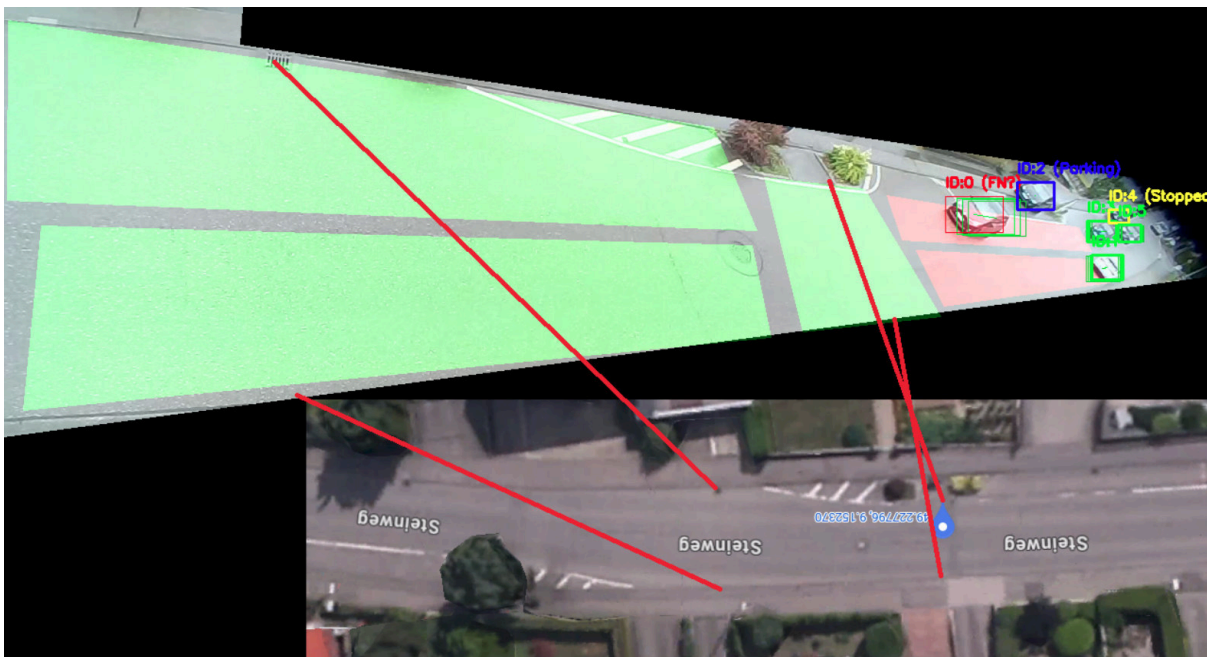
[Abb. 10: Das rote Auto auf der gegenüberliegenden Fahrbahn wurde während des

Okklusionszeitraums weiterhin getrackt. Aufgrund der Überlänge des LKWs mit Anhänger (und damit einer langen Verdeckung) wurde das Tracking des Fahrzeugs nach 10 Frames abgebrochen. Zusätzlich zu sehen sind die Perception Areas und der Status der erkannten Objekte; der Bewegungsvektor entlang der Objektzentren ist gut zu erkennen.]

Zur Fehlerkorrektur: Liegt eine Bounding-Box mit geringer Distanz zur vorhergesagten Position vor, aber mit Modell-Confidence unter dem Schwellwert (z. B.  $< 0,5$ ), wird von einem „semi“ False Negative ausgegangen und die erkannte Box (nicht die Prädiktion) für die weitere Trajektionsvorhersage genutzt; die Modellausgabe tendiert in der Praxis zu Confidence nahe 0 oder nahe 1. In der Visualisierung und Auswertung werden diese Fälle explizit geführt. Tracks, die in einer Lücke nur durch Prädiktion weiterlaufen, können als „possible false negative“ (Modell hat das Objekt übersehen) oder „possibly obstructed“ (Objekt möglicherweise verdeckt) markiert werden. Diese Unterscheidung unterstützt die Analyse. Für die CPM-Aussendung zählt die konsistente Fortführung der Spur. Die Logik steht im Einklang mit der Safety-First-Gewichtung. Die Vermeidung von False Negatives (übersehene Objekte) hat Vorrang. Eine temporär höhere Rate an False Positives (z. B. fortgeführte Tracks ohne aktuelle Detektion) wird in Kauf genommen, um kritische Situationen durch „verschwindende“ Objekte zu vermeiden. Objekte, die über mehrere Frames hinweg erkannt werden, sich aber nicht oder nur minimal bewegen, erhalten den Status „parking“. Optional können Objekte mit Status „parking“, die sich zusätzlich in einer vordefinierten Parkfläche befinden, aus der CPM herausgefiltert werden. So wird vermieden, dass geparkte Fahrzeuge in Parkbereichen die Shuttle-Kommunikation mit bewegungsrelevanten Meldungen belasten. Die Information für den Static-Parking-Use-Case (z. B. Belegungsstatus) bleibt separat verfügbar. Diese Logik unterliegt einer Grenze. Je weiter ein Objekt vom Sensor entfernt ist, desto schwieriger ist es, eine mögliche Bewegung zu erfassen. Besonders am

Bildrand erscheinen Objekte, die sich in der Realität bewegen, aufgrund der Kameraperspektive und der geringeren pixelbezogenen Verschiebung als statisch und können fälschlich den Status „parking“ erhalten. Bei der optionalen Filterung aus der CPM ist dies zu berücksichtigen (AP5). Für den Use Case Static Parking wird zudem eine binäre Flächenklassifikation genutzt. Eine Region of Interest wird als „frei“ oder „belegt“ eingestuft. Ein ML-Modell liefert eine Belegungswahrscheinlichkeit. Ein gleitender Mittelwert über ein konfigurierbares Fenster (typisch rund 10 Frames) glättet die Rohwerte und reduziert Flatterer. Die Zustandsentscheidung erfolgt über einen Banding-Threshold. Unterhalb einer unteren Schwelle gilt „frei“, oberhalb einer oberen Schwelle „belegt“. Im Zwischenbereich wird der letzte bekannte Zustand beibehalten. Die flächenbezogene Ausgabe für die CPM ist eine einzelne Perception Region mit Weltkoordinaten ( $x, y$  aus Konfiguration,  $z$  als negative Sensorhöhe) und `number_of_perceived_objects = 0` (frei) bzw. `1` (belegt) (APt).

Alle beschriebenen Algorithmen nutzen eine Homographie für die Transformation von normalisierten Bildkoordinaten in den Weltraum (Ground Plane, ENU). Die Kalibrierung der Homographie-Matrix erfolgte über markante Punkte im Bild (z. B. Gullideckel, Straßenschilder, Fußgängerüberweg), die auch auf Satellitenbildern (z. B. Google Maps) identifizierbar sind.



[Abb. 11: Markante Punkte im Kamerabild können mit GPS-Koordinaten verbunden werden; sie wurden manuell gesucht und konfiguriert und dienen als Grundlage für die Homographie-Matrix. Zusätzlich ist die Fehlerkorrektur für den Track „ID:0“ zu sehen: Die rote Bounding-Box wurde nicht vom Modell erkannt, sondern ist eine Approximation. Obwohl die Ränder nicht exakt sind, liegt der Center-Point im Zentrum des Objekts und liefert einen guten Wert für die weitere Prädiktion.]

Über die Zuordnung von Pixelkoordinaten zu diesen realen Weltpositionen, und damit zum Abstand zum Sensor im dreidimensionalen Raum, ließ sich die Homographie-Matrix berechnen. Die Transformation erfolgt über die inverse Homographie. Objekte und Flächen werden in der Ground Plane angenommen. Die Algorithmen sind in die bestehende Sensor-Firmware integriert und nutzen die gemeinsame Infrastruktur für Konfiguration, Modellverwaltung und Bildvorverarbeitung. Die Detektionsgenauigkeit ist in einem Umkreis von etwa 10 m um den Sensor am höchsten und nimmt mit der Entfernung ab. Ab etwa 30 m Entfernung werden die Ergebnisse in der Regel unbrauchbar. Die Auflösung der Kameras setzt hier eine harte Grenze. Dieser Bereich deckt den für die Shuttle-Steuerung im Umfeld der stationären Sensoren relevanten Nahbereich ab. Zusätzlich beeinflussen Position und Aufbauhöhe des Sensors diesen Wert. Je höher der Sensor montiert ist, desto größer fällt die nutzbare Reichweite aus, da Perspektivverzerrungen bei der Annahme einer Ground Plane erst in größerer Entfernung stark wirken. Im Rahmen der Unterstützung des Sicherheitskonzepts (AP3, AP6) arbeitete SONAH an einem Konzept zur Abschätzung der Ergebnisse der infrastrukturbasierten Perzeption. Dabei wurden Kennzahlen (z. B. Certainty) identifiziert, um die Gewissheit über die Bestimmung von Objekteigenschaften wie Geschwindigkeit und Position zu erhöhen. Die ML-Modelle liefern eine Konfidenz (Probability) als Ausgangsbasis. Diese kann durch weitere Faktoren reduziert werden, z. B. reduziert eine sehr geringe Bildhelligkeit die Certainty. Bei „virtuellen“ Objekten (vermutlich verdeckt oder False Negative), die nur durch Trajektionsfortschreibung geführt werden, sinkt die Certainty mit jedem angenommenen Prädiktionsschritt. Diese Arbeiten tragen zur Fehlertoleranz und Robustheit des hybriden Gesamtsystems bei und unterstützen die Freigabe-Argumentation gegenüber dem TÜV Süd (AP3, AP6).

## ## CPM. Implementierung und Veröffentlichung

Die Überführung der Detektionsergebnisse in standardisierte Collective Perception Messages (CPM) und deren Veröffentlichung an die V2X-Infrastruktur wurde im Vorhaben auf zwei technischen Wegen realisiert. Während der Entwicklung wurden zunächst interne Formate zur schnellen Selbstkontrolle und Validierung der Modelle genutzt. Das bestehende SONAH-System sendet zu Auswertungszwecken Daten an eine API, die jedoch auf Parkdaten ausgelegt ist. Auf dem Sensor sind der Network-Container und der Computer-Vision-Container aus Sicherheitsgründen voneinander getrennt. Die vollständige Integration in das standardisierte CPM-Format erfolgte darauf aufbauend. Über die CPM erhält das autonome Shuttle zusätzliche Wahrnehmungsdaten (u. a. von stationären Sensoren wie den im Vorhaben eingesetzten) und kann seinen Fahralgorithmus robuster auslegen, z. B. durch Objekte, die aus Shuttle-Sicht verdeckt sein können, sowie durch Flächenzustände (frei/belegt), die zur Vermeidung von Gridlocks beitragen. Beide technischen Pfade ermöglichen die Aussendung von CPM an die WaveBee Roadside Unit (RSU). Sie

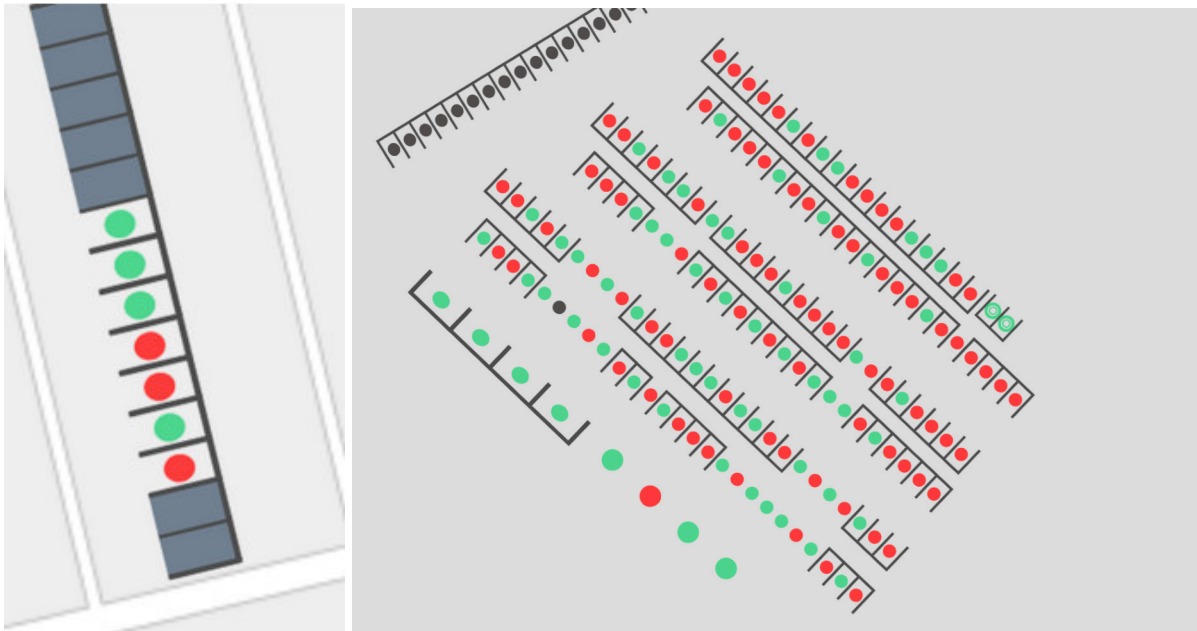
unterscheiden sich hinsichtlich der Architektur und der Anforderungen an die Systemumgebung (AP5).

Der erste Ansatz nutzt eine SSH-basierte Kommunikation mit dem WaveBee-Gerät. Über die Power-over-Ethernet-Verbindung (PoE) wird eine SSH-Session zum WaveBee aufgebaut. Die CPM-Payload wird als String encodiert und über den Befehl `ros2 topic pub` an den Topic `/cpm_topic` gesendet. Dieser Ansatz erfordert auf Seiten des SONAH-Sensors nur eine String-Encodierung der Nachricht und einen Remote-Befehl. Der Implementierungsaufwand ist vergleichsweise gering. Allerdings ist diese Methode nicht auf allen OS-Versionen der WaveBee-Hardware verfügbar, sodass die Einsatzfähigkeit von der jeweiligen Firmware-Version abhängt. Der zweite Ansatz veröffentlicht die CPM direkt auf dem SONAH-Sensor über eine ROS-2-Bridge. Hierfür wurde die ROS-Bridge inklusive der projektspezifischen CPM-Definitionen (`wavebee_ros_cpm_bridge`) in die bestehende Firmware integriert. Die Integration umfasste den Aufbau eines Python-Packages für die Zielarchitektur, die Einhaltung der Speicherlimits des Edge-Devices sowie die Anbindung als weiteren Task im Network Manager. Sie erforderte nennenswerten Aufwand. Neben dem Cross-Compile für die ARM-Architektur und der Berücksichtigung von Ressourcenbeschränkungen sowie der Koordination mit den übrigen Netzwerkdiensten waren das Patchen der Bridge und die Integration in die Sensor-Firmware (durch SONAH als Anbieter der Sensoren) sowie der manuelle Build der ROS-2-Python-Pakete inklusive der CPM-Definitionen zeitintensiv. Die CPM werden aus einer Redis-Queue gelesen, in das ROS-2-Nachrichtenformat überführt und mit einem konfigurierbaren QoS-Profil (`ReliabilityPolicy.RELIABLE`, `DurabilityPolicy.TRANSIENT_LOCAL`) auf dem Topic `/cpm_topic` publiziert (AP5).

Das Debugging und die Konfiguration von IoT-Geräten im Feld stellen stets eine operative Herausforderung dar. So musste im Verlauf des Vorhabens eine WaveBee-Einheit vor Ort umkonfiguriert werden, um im PoE-Modus die CPM per V2X auszusenden. Eine solche Konfiguration entlang der Kette Sensor Device Management, Sensor, WaveBee ohne physischen Zugriff birgt ein inhärentes Risiko. Geht der Sensor oder die WaveBee in einen „offline“- oder „error“-Zustand über, kann dieser ohne Vor-Ort-Einsatz nicht mehr behoben werden. Dies unterstreicht die Bedeutung robuster Fernwartungsmechanismen und einer sorgfältigen Vorabvalidierung der Konfiguration (AP5). Die Projektpartner Keysight und Bosch unterstützten das Vorhaben beim Debugging der WaveBee-Einheiten. Das FZI unterstützte das Vorhaben vor Ort beim Auslesen und Debuggen der CPM. Die Partner waren dazu mit eigener V2X-Hardware an den Standorten präsent. Zusätzlich analysierten und validierten sie im Labor ROS-Bags, die im Laboraufbau (Schnittstelle Sensor, Shuttle/CPM) generiert worden waren. Die CPM wurden sowohl auf technische Korrektheit geprüft (Lesbarkeit der Nachricht, keine fehlerhaften Datentypen) als auch auf semantische

und inhaltliche Korrektheit sowie die Einhaltung von Konventionen, z. B. Konformität des Koordinatensystems mit dem des Shuttles (AP5).

Inhaltlich werden in den CPM zwei Haupttypen von Informationen übertragen: Perceived Objects (wahrgenommene Objekte) und Perception Areas (wahrgenommene Bereiche). Die Perceived Objects enthalten für jedes detektierte Objekt eine Objekt-ID, den `measurement_delta_time` (Zeitdifferenz zur Referenzzeit), die Position in Weltkoordinaten (`x`, `y`, optional `z`), optional die Objektgeschwindigkeit (`velocity`) und Beschleunigung (`acceleration`) sowie die Objektage. Die Gewissheit über die Objektinformation (`Certainty`) basiert auf der vom ML-Modell ausgegebenen Konfidenz, wird aber durch kontextuelle Faktoren reduziert, z. B. durch geringe Bildhelligkeit. Bei nur prädiktiv fortgeführten Objekten (vermutlich verdeckt oder False Negative) sinkt die `Certainty` mit jedem Trajektionsschritt ohne erneute Detektion. Bei der Shuttle-Kommunikation (WaveBee Detection und WaveBee Tracking) werden die Koordinaten über eine Homographie-Matrix aus dem Bildraum in den Weltraum (ENU) transformiert. Die Referenzzeit wird aus dem Systemzeitstempel abgeleitet (letzte sechs Stellen der Sekunden, kombiniert mit den ersten drei Stellen der Nanosekunden). Für die Perception Areas werden vordefinierte Straßensegmente als Quadrilateral mit `shape_reference_point` (Mittelpunkt), `semi_length`, `semi_breadth` und `orientation` beschrieben. Jeder Bereich enthält die Anzahl der darin enthaltenen Objekte und die zugehörigen Objekt-IDs. Sowohl bei Perceived Objects als auch bei Perception Areas wird stets angenommen, dass sich die Objekte bzw. Flächen in der Ground Plane befinden. Die `z`-Koordinate entspricht der (negativen) Höhe des Sensors über der Referenzfläche. Eine explizite 3D-Lokalisierung wird nicht durchgeführt. Die CPM werden in einem konfigurierbaren Takt veröffentlicht, der in der Regel mit der Verarbeitungsrate (FPS) des Sensors übereinstimmt. Unter ungünstigen Bedingungen (z. B. hohe Umgebungstemperatur, schwierige Lichtverhältnisse) kann die FPS des Sensors sinken. In diesem Fall wird die letzte CPM möglicherweise erneut veröffentlicht. Lediglich die Timestamps werden angepasst, um eine konsistente Zeitfolge zu gewährleisten und Lücken in der Nachrichtenversorgung zu vermeiden (AP5). Es wurden zwei Anwendungsfälle implementiert. Zum einen die Echtzeit-Verfolgung von Objekten und Bereichen für die Kommunikation mit dem autonomen Shuttle (WaveBee Detection und WaveBee Tracking) an den Standorten Karlsruhe und Bad Wimpfen. Zum anderen wurde der bestehende SONAH Use Case „Static Parking“ um die CPM-Aussendung erweitert. Dieser Ansatz wurde auf dem Bosch-Parkplatz aufgebaut und vertestet. Die Anbindung erfolgte an die bestehende Standard-Website für diesen Use Case, sodass die CPM-Informationen (Belegungsstatus der Parkbereiche) sowohl für die V2X-Infrastruktur als auch für die Weboberfläche verfügbar sind.



[Abb. 14,15 und 16: Frontend für den Standard-Use-Case Static Parking in der minimalen Version auf dem Bosch-Parkplatz und als Vergleich Live-System in Kevelaer.]

Im Rahmen der Evaluierung wurde festgestellt, dass eine Echtzeit-Lokalisierung von Objekten („Object“ im Sinne einer generischen Position) auf dem Edge-Device technisch machbar ist. Eine Klassifizierung nach BASt-Klassen (Bundesanstalt für Straßenwesen) erwies sich hingegen als ungenau. Zudem decken die BASt-Klassen nicht alle möglichen Objekttypen ab, die im urbanen Mischverkehr vorkommen (z. B. atypische Fahrzeugformen, Sonderfahrzeuge).



[Abb. 9: Besondere LKWs mit außergewöhnlicher Ladefläche werden oft als zwei Objekte erkannt.]

Ob eine Klassifizierung in den CPM mitgeführt wird (`classification_present`), leitet sich aus den Metadaten des auf dem Edge-Device konfigurierten ML-Modells ab. Steht ein vollständiges Class-Set zur Verfügung, wird die Objektklasse mitgeschickt. Verfügt das Modell hingegen lediglich über eine ClassLabelMap mit Einträgen wie „background“ und „object“, wird `classification_present` auf `false` gesetzt und die CPM entsprechend verkürzt. Die Klassifikationsfelder entfallen. Die Objekte werden in diesem Fall als generische Hindernisse mit Position und optionaler Kinematik übermittelt. Diese Design-Entscheidung steht im Einklang mit der Safety-First-Gewichtung. Die Priorität liegt auf der vollständigen Erfassung von Objekten (Vermeidung von False Negatives) gegenüber einer feingranularen Klassenzuordnung (AP5).

### **Depth Perception. Verfolgter Ansatz ohne Einsatz in der Finalversion**

Nicht in die finale Systemversion übernommen wurde ein Ansatz zur Tiefenwahrnehmung (Depth Perception). Die Überlegungen und Ergebnisse werden der Vollständigkeit halber kurz zusammengefasst. Rein theoretisch wäre mit zwei Kameras pro Sensor ein überlappendes Sichtfeld denkbar, über das per Stereo Vision die Tiefe und damit die Entfernung von Objekten berechnet werden könnte. Bei SONAH wird jedoch stets angestrebt, die mit einem Sensor abgedeckte Gesamtfläche zu maximieren. Entsprechend weisen die Kameras ein geringes oder gar kein überlappendes Sichtfeld auf, sodass eine klassische Stereo-Auswertung nicht in Frage kam. Stattdessen wurde ein

State-of-the-Art-Modell zur monokularen Tiefenschätzung (MiDaS/DPT-Familie) eingesetzt. Die öffentlich verfügbaren, vortrainierten Gewichte liefern ausreichend gute Ergebnisse.



[Abb. 12 und 13: Modeloutput der Depth-Perception. Die vortrainierten Gewichte liefern brauchbare Ergebnisse; der Sprinter auf der gegenüberliegenden Fahrbahn (Abb. 13) wird im Schatten des LKWs (Abb. 12) praktisch gar nicht erkannt.]

Der Bereich, in dem die Depth-Perception-Modelle zuverlässig funktionieren, entspricht in etwa dem Bereich, in dem die Objekterkennungsmodelle konsistent gute Detektionen liefern. Eine Kombination aus Objekterkennung und Tiefenschätzung wäre damit fachlich attraktiv. Die verbauten Edge-Geräte können jedoch nicht in Echtzeit zwei ML-Modelle (Objekterkennung und Depth Perception) zusammen mit dem CPM-Algorithmus ausführen.

Mit einem dedizierten Accelerator ließe sich dieses Problem lösen. Dies würde jedoch die Hardwarekosten und den Leistungsbedarf (Power Draw) des Geräts erhöhen und zu weiteren Herausforderungen führen (z. B. Akkulaufzeit, Abwärme, Kühlung). Die Skalierbarkeit des Systems im städtischen Raum wäre davon beeinträchtigt. Vor diesem Hintergrund wurde auf die Integration der Depth Perception in die finale Version verzichtet (AP4). Grundsätzlich wäre die Lösung technisch möglich und hätte zwei wesentliche Vorteile. Zum einen würde sie das Problem der Zuordnung von Objekten zu Perception Regions (Abschnitte vs. Spuren, siehe Algorithmus-Beschreibung) erheblich vereinfachen, da eine präzisere räumliche Einordnung möglich wäre. Zum anderen würde die Genauigkeit der Koordinaten der Perceived Objects durch ein zusätzliches Depth-Perception-Modell verbessert, da die Entfernung zum Sensor direkt geschätzt werden könnte statt über die Ground-Plane-Projektion. Beides hätte die Qualität der an das Shuttle über CPM gelieferten Objekt- und Flächeninformationen erhöht. Das grundsätzliche Problem von Verdeckungen (Okklusionen) wird durch monokulare Depth Perception nicht gelöst. Im Gegenteil: Bei teilweise verdeckten Objekten liefern die Tiefenmodelle oft schlechte Ergebnisse. Die Vermischung der Merkmale mehrerer Objekte in einem Bildausschnitt ist ein bekanntes Problem bei ML-Modellen in räumlich dichten Szenen. Wo sich Objekte überlappen oder aneinandergrenzen, verschwimmen die Repräsentationen und die Tiefenschätzung wird unzuverlässig. Abhilfe könnte ein zusätzliches Segmentierungsmodell (z. B. SAM2) schaffen, das Objektgrenzen explizit erfasst. Ein solches Modell ist rechenintensiv jedoch deutlich aufwendiger als reine Objekterkennung oder Tiefenschätzung und würde die Echtzeitfähigkeit auf Edge-Hardware weiter verschärfen (AP4).

## 2. Wichtigste Positionen des zahlenmäßigen Nachweises

Nicht im Rahmen dieses Auszugs dargestellt. Diese können der beigefügten Datei "20260227 Berichtsblätter & Zahlenmäßige Nachweise SHUX" entnommen werden (AP1).

## 3. Notwendigkeit und Angemessenheit der geleisteten Arbeit

Die Echtzeit-Anforderungen der Shuttle-Kommunikation (fortlaufende CPM-Aussendung an das autonome Fahrzeug) erforderten eine kritische Abwägung im Zielkonflikt aus Modellkomplexität, Eingangsauflösung und Inferenzrate. Diese Abwägung und die daraus resultierenden technischen Entscheidungen sind in der eingehenden Darstellung zur Modellarchitektur, zur Quantisierung und zur ML-Pipeline beschrieben (vgl. Gesamtdarstellung, Abschnitte Datenerfassung/ML-Pipeline sowie Modellarchitektur und Quantisierung). Insbesondere die SONAH-Imager-Pipeline musste gegenüber dem Kerngeschäft (z. B. Static Parking mit 30-Sekunden-Eventübertragung) massiv optimiert werden, um die Latenzvorgaben für die Echtzeit-Inferenz auf dem Edge-Device einzuhalten. Die Entscheidung gegen dedizierte Beschleuniger (GPUs, TPUs) wurde bewusst zugunsten einer großflächigen, kosteneffizienten Skalierung im städtischen Raum getroffen: Standard-ARM-Hardware ohne zusätzliche Beschleunigerkarten ermöglicht den Einsatz an bestehender Infrastruktur (z. B. Straßenbeleuchtung) bei geringem Energiebedarf und ohne die thermischen und kostenbedingten Grenzen leistungsstarker GPU-Edge-Systeme. Der Aufwand für die Optimierung der Inferenz-Pipeline, die Erweiterung der Data-Loading-Pipeline (PNG, verschiedene Farbräume) sowie die Integration der Algorithmen (Detektion, Tracking, Trajektionsvorhersage, Flächenbewertung) in die Sensor-Firmware war in diesem Kontext notwendig und angemessen (AP5).

## 4. Voraussichtlicher Nutzen und Verwertbarkeit (Fortschreibung des Verwertungsplans)

Die gewählten Systemarchitekturen (Edge-Inferenz mit MobileNetV2/TFLite, standardisierte CPM-Aussendung über WaveBee-RSU, generische Schnittstellen und Konfigurationsinfrastruktur) weisen eine hohe Praxisrelevanz auf und sind für den langfristigen Betrieb unter Realbedingungen ausgelegt. Die implementierte generische Schnittstelle mit standardisierten Kommunikationsformaten (CPM, Perception Regions, Perceived Objects) ermöglicht eine nahtlose Integration in übergeordnete Leitsysteme und in bestehende V2X-Infrastrukturen. Aus Infrastruktursicht ergeben sich konkrete Nutzenpotenziale: Gridlock-Prävention durch Flächenzustandsbewertung (frei/belegt) und durch prädiktive Trajektionsinformationen, Okklusionsmanagement durch die Fortführung von Tracks auch bei temporärer Nicht-Detektion oder Verdeckung, sowie die Bereitstellung von Hindernis- und Flächeninformationen für das autonome Shuttle auch außerhalb der direkten Sicht des Shuttles. Die Validierung an den Standorten Karlsruhe und Bad Wimpfen sowie die Integration in die Kooperation mit den

Stadtwerken und mit dem FZI/Bosch/Keysight-Konsortium unterstreichen die Verwertbarkeit für den Einsatz im öffentlichen Raum und in nachfolgenden Projekten (AP5, AP7).

## 5. Fortschritt auf dem Gebiet des Vorhabens bei anderen Stellen

International und national werden verstärkt LiDAR-basierte oder hochperformante GPU-basierte Edge-Systeme (z. B. NVIDIA-Jetson-Serie) für die infrastrukturbasierte Perzeption erprobt. Diese Ansätze stoßen bei einer flächendeckenden Skalierung jedoch an Grenzen: hohe Anschaffungskosten, thermische Belastung in Outdoor-Gehäusen (insbesondere im Sommer) und höherer Energiebedarf erschweren den Einsatz an bestehender städtischer Infrastruktur. Der Fortschritt durch Shuttle2X liegt in der Validierung eines Low-Power-Ansatzes, der mit Standard-Hardware (ARM-CPU, ohne dedizierte Beschleuniger) und hochoptimierter Inferenz- und Algorithmen-Pipeline vergleichbare Funktionalität (Echtzeit-Objekterkennung, Tracking, CPM-Aussendung, Gridlock-relevante Flächenbewertung) bei einem Bruchteil der Kosten und des Energieverbrauchs liefert. Damit setzt sich das Vorhaben von rein forschungsorientierten, kostenintensiven Einzelinstallationen ab und zeigt eine realistische Option für die Skalierung kooperativer Perzeption im urbanen Raum (AP5).

## 6. Erfolgte oder geplante Veröffentlichungen

Im Berichtszeitraum sind von SONAH-Seite keine wissenschaftlichen oder technischen Veröffentlichungen erfolgt, die über Beiträge auf Social-Media-Plattformen (z. B. LinkedIn) hinausgehen. Derzeit sind keine Veröffentlichungen geplant. Intern wurde diskutiert, die im Vorhaben vorgenommenen Erweiterungen der TensorFlow Object Detection API (u. a. Data-Loading-Pipeline für PNG und verschiedene Farbräume, vgl. Gesamtdarstellung/Modellarchitektur) als Pull Request beim upstream-Projekt einzureichen. Das betreffende Repository wird jedoch nicht mehr aktiv gepflegt (verwaist). Zudem wäre erheblicher Zusatzaufwand nötig, um den bei SONAH bereits produktiv genutzten, funktionierenden Code an die Coding Guidelines des upstream-Projekts anzupassen. Vor diesem Hintergrund wurde von einer Einreichung abgesehen. Die Verbesserungen bleiben intern im Einsatz und stehen für die weitere Verwertung im Unternehmen zur Verfügung (AP5, AP7).

## **IV) Kurzfassung (Berichtsblatt)**

In der ersten Phase des Vorhabens wurde die konzeptionelle Grundlage für eine skalierbare hybride Verarbeitungsarchitektur (Edge-Inferenz, optional Cloud-Analyse) zur Hindernis- und Flächenerkennung für autonome Systeme geschaffen. Dies umfasste die Entwicklung der logischen Verarbeitungskette (Datenerfassung, Privacy-Maskierung, Backend-Übertragung, Annotation in CVAT/COCO), den Aufbau einer Labeling-Pipeline inklusive Nutzung öffentlicher Datensätze sowie die Validierung im Labor. Durch die Optimierung der Edge-Inferenz mit MobileNetV2 (TFLite, Full-Integer-Quantisierung) konnte eine Echtzeit-Detektion bei geringem Energieverbrauch erreicht werden. Die Entscheidung gegen dedizierte Hardware-Beschleuniger wurde aus Gründen der Wirtschaftlichkeit und der Skalierbarkeit getroffen. Am Standort Bad Wimpfen erfolgte die Integration in eine V2X-Infrastruktur mittels WaveBee-RSU und die Übermittlung von Collective Perception Messages (CPM). Die Evaluation umfasste anspruchsvollere Szenarien (u. a. höheres Verkehrsaufkommen, Fußgängerüberweg) sowie die Demonstration von Gridlock-Prävention und Okklusionsmanagement. Die Algorithmen für Detektion, Tracking, prädiktive Trajektionsfortführung und flächenbasierte Zustandsbewertung sind in die Sensor-Firmware integriert und an beiden Shuttle-Standorten im Einsatz.