

Visual Traffic Counting System VTRACS

Ein Projekt finanziert im Rahmen der
Verkehrsinfrastrukturforschung 2019
(VIF2019)

Oktober 2021

Impressum:

Herausgeber und Programmverantwortung:

Bundesministerium für Verkehr, Innovation und Technologie
Abteilung Mobilitäts- und Verkehrstechnologien
Radetzkystraße 2
A – 1030 Wien



ÖBB-Infrastruktur AG
Nordbahnstraße 50
A – 1020 Wien



Autobahnen- und Schnellstraßen-Finanzierungs-
Aktiengesellschaft
Rotenturmstraße 5-9
A – 1010 Wien



Für den Inhalt verantwortlich:

SLR Engineering GmbH
Gartengasse 19
A – 8010 Graz



Technische Universität Graz
Institut für Straßen- und Verkehrswesen (ISV)
Rechbauerstraße 12
A – 8010 Graz



Programmmanagement:

Österreichische Forschungsförderungsgesellschaft mbH
Thematische Programme
Sensengasse 1
A – 1090 Wien



Visual Traffic Counting System VTRACS

Ein Projekt finanziert im Rahmen der
Verkehrsinfrastrukturforschung
(VIF2019)

Autoren:

Dipl.-Ing. Oliver SIDLA (SLR)

Dipl.-Ing. Manuel LIENHART (ISV)

MSc. Filippo GAROLLA (SLR)

Univ.-Prof. Dr.-Ing. Martin FELLENDORF (ISV)

Auftraggeber:

Bundesministerium für Verkehr, Innovation und Technologie

ÖBB-Infrastruktur AG

Autobahnen- und Schnellstraßen-Finanzierungs-Aktiengesellschaft

Auftragnehmer:

SLR Engineering GmbH (SLR)

TU Graz / Institut für Straßen- und Verkehrswesen (ISV)

Liste der verwendeten Abkürzungen und Begriffe

AG	...	Auftraggeber
BA	...	Bildanalyse
BV	...	Bildverarbeitung
CNN	...	Convolutional Neural Network
CPU	...	Central Processing Unit
DL	...	Deep Learning
FZ	...	Fahrzeug
GPU	...	Graphics Processing Unit
HOG	...	Histograms of Oriented Gradients
HW	...	Hardware
NN	...	Neuronales Netzwerk
SW	...	Software
TF	...	Tensorflow
TFLite	...	Tensorflow Lite

Detektion/Detektor	wird in diesem Bericht im Zusammenhang mit der Merkmalerkennung und Detektion in der Bildverarbeitung verwendet
--------------------	---

INHALTSVERZEICHNIS

Inhaltsverzeichnis	5
Abbildungsverzeichnis	6
1. Einleitung	8
1.1. Stand der Forschung	9
1.2. Projektziel	10
2. Methodik	11
2.1. Datengrundlage – Rohdaten/Bestätigte Datensätze	12
2.2. Fahrzeug Detektion und Tracking	13
2.2.1. Das Detektionsmodul im Detail	13
2.2.2. Die Detektion am embedded System	14
2.2.3. Fazit für das Arbeitspaket Fahrzeugdetektion und Tracking	15
2.3. Kalibrierung und Zählung	15
2.3.1. Datengrundlage	15
2.3.2. Methodik Kalibration	17
2.3.3. Detailliertes Ablaufschema Kalibration	21
2.4. Fahrzeugzählung	24
2.5. Re-Identifikation mittels visueller Signatur	24
2.5.1. Praxistests mit aufgezeichneten Video Daten	28
2.5.2. Fazit für das Arbeitspaket Visuelle Signatur	33
2.6. Demonstration der geometrischen Kalibration	33
2.7. Verkehrstechnische Evaluierung	35
3. Zusammenfassung	41
3.1.1. Projektfazit	41
3.1.2. Weiterer zukünftiger Entwicklungsbedarf	41
4. Anhang	42

ABBILDUNGSVERZEICHNIS

Abbildung 1: Projektstruktur von VTRACS.....	12
Abbildung 2: Messkampagne A2 Feldkirchen Situation 1.	16
Abbildung 3: Messkampagne A2 Feldkirchen Situation 1.	17
Abbildung 4: Beispiel Aufbau Datensatz.	18
Abbildung 5: Beispiel Trajektorien mit Testquerschnitten.....	19
Abbildung 6: Beispiel Dichtefunktion für einen Testquerschnitt.....	19
Abbildung 7: Beispiel Fahrstreifenerkennung auf mehreren Querschnitten.	20
Abbildung 8: Beispiel Bildkalibrierung.	21
Abbildung 9: Flow-Chart Kalibration.(Methodikerweiterung auf Grundlage von Knoop et al., 2017)	23
Abbildung 10: Die Struktur der visuellen Fahrzeug Signatur.....	27
Abbildung 11: Bounding-Box Bilder des VTRACS Objekt Trackers. Je nach Entfernung unterscheiden sich die Auflösungen bzw. Größen der Fahrzeugbilder.	27
Abbildung 12: Darstellung von Re-Identifikationen zeigen den Referenzbild von Kamera 1, und die 10 besten Matches.	29
Abbildung 13: ID 504683 von Kamera 1 stimmt nicht mit dem ID 504652 von Kamera 2 überein, die beiden LKWs sind sich jedoch extrem ähnlich.	30
Abbildung 14: Die Performance der Re-Identifikation relativ zu Sensitivitäts Grenzwerten. Ein niedrigerer Grenzwert für akzeptierte Matches führt zu weniger Fehlern bei der Re-Identifikation (orange Kurve), aber auch zu einer geringeren Anzahl an registrierten Fahrzeugen (graue Kurve).	31
Abbildung 15 Falsche Re-Identifikationen aufgrund schlechter Bildqualität.....	31
Abbildung 16: Die Messpunkte auf der Autobahn A2 beim Flughafen Graz. Die farbigen Punkte zeigen Messpunkte von detektierten Fahrzeugtrajektorien. Die beiden gelben Pfeile markieren die Punkte, zwischen denen die ‚Reisezeiten‘ gemessen wurden. Quelle: Schöllberger, A. (2021).....	32
Abbildung 17: Reisezeitmessungen aus der Messkampagne vom November 2020.	33
Abbildung 18: Beispiele für kalibrierte Fahrwege von PKW. Mittels der durchgeführten Tsai Kalibration können Fahrzeugpositionen im Bild vermessen werden und sind so für weitergehende verkehrstechnische Analysen zugänglich. (Bildquelle: Schöllberger, 2021).....	34
Abbildung 19: Screenshot des VTRACS Prototyps im laufenden Betrieb.....	35
Abbildung 20: Testcase 1 PKW – Vergleich manuelle Zählung und VTRACS-Prototyp.	36
Abbildung 21: Testcase 1 LKW – Vergleich manuelle Zählung und VTRACS-Prototyp.	36

Abbildung 22: Vergleich Trackingergebnisse (dunkelgrün) mit ASFiNAG-Messquerschnitten (hellgrün).	38
Abbildung 23: Vergleich Messergebnisabweichungen über den Testzeitverlauf.	39
Abbildung 24: Beispielauswertung für Kamera „VK_A02_0_168,610-F1-FL-DHS“ über den Testzeitverlauf.	40
Abbildung 25: Auswertung für Kamera „VK_A02_0_168,610-F1-FL-DHS“ über den Testzeitverlauf.	42
Abbildung 26: Auswertung für Kamera „VK_A02_0_168,610-F2-FL-DHS“ über den Testzeitverlauf.	43
Abbildung 27: Auswertung für Kamera „VK_A02_0_169,900-F1-FL-DHS“ über den Testzeitverlauf.	44
Abbildung 28: Auswertung für Kamera „VK_A02_0_169,900-F2-FL-DHS“ über den Testzeitverlauf.	45

1. EINLEITUNG

Die zahlreichen Webcams der ASFiNAG stellen eine wertvolle Datenquelle für die Generierung von Verkehrslageinformation dar, deren Potential bestenfalls nur ansatzweise genutzt wird.

Im Projekt VTRACS werden aus den Webcam-Daten Fahrzeuge in Echtzeit gezählt, Fahrzeug-Trajektorien generiert und einzelne Fahrzeuge über mehrere Kameras hinweg wiedererkannt. Dafür werden neue vielversprechende Deep Learning Methoden eingesetzt, um Fahrzeuge zu detektieren und deren visuelle Signaturen zu generieren und zu vergleichen.

Die seit einigen Jahren stattfindenden rasanten AI Entwicklungen mittels Deep Learning (DL) haben die Einsatzmöglichkeiten visueller Detektion drastisch erweitert. Visuelle Detektion kann mittlerweile auch bei Datenquellen angewandt werden, die bisher qualitativ nicht die Voraussetzungen für eine automatisierte Analyse geboten haben.

Die DL Detektion wird bei SLR mit einem eigenem state-of-the-art Tracker kombiniert, daraus generierte Trajektorien werden für die Zahlung oder verkehrstechnische Auswertung verwendet (beispielsweise Staudetektion). Bei ausreichender Bildqualität können die geforderten zwei Fahrzeugklassen gut voneinander getrennt werden.

Im Rahmen des VTRACS Projektes wurde ein Fahrzeug-Zählsystem entwickelt, das die ASFiNAG Videoquellen unterschiedlichster Qualität, Auflösung und Bildfelder auswerten kann. Ein Großteil der vorhandenen Algorithmen wird anhand eines Prototyps auf einem ASFiNAG Server demonstriert, weiters wurde ein Prototyp auf einem embedded Gerät der ASFiNAG bereitgestellt.

Das **Projektkonsortium** bestand aus der **TU Graz / ISV und SLR Engineering**. Das Team hat als Ergebnis des **Projektes VTRACS** zusammen die SW Module entwickelt.

Das VTRACS System besteht aus den Hauptkomponenten:

- Tracking Server + Echtzeit Datenbank
- Fahrzeugtracker
- Offline Kalibration
- Online Trajektorien Analyse + Zählung
- Embedded Prototyp am Nvidia Xavier AGX Rechner (Fahrzeugtracking)

Ein maßgeblicher des VTRACS Systems wurde mit **GPL freien Open Source Modulen** implementiert. Es wurden **mehrere KI Deep Learning Modelle anhand von Sample Daten trainiert und evaluiert**. Die beste Kombination aus Detektionsrate und Rechenzeit wurde in das VTRACS System integriert, neue Ideen wurden während der Arbeit am Projekt ebenfalls getestet und, wo sinnvoll, integriert.

Zuerst wird in Kapitel 1.1 **Fehler! Verweisquelle konnte nicht gefunden werden.** der Stand der Forschung und in Kapitel 1.2 die Projektziele formuliert.

Kapitel 2 beschreibt den technischen Lösungsansatz im Detail, mit einer Zusammenfassung und einem Fazit in Kapitel 3.

1.1. Stand der Forschung

Entwicklungen in der künstlichen Intelligenz (KI) der letzten 5 Jahre haben zu deutlichen Verbesserungen in der Datenanalyse und Bildanalyse geführt. Die Entwicklung von komplexen Neuronalen Netzwerken (**Deep Learning**) als Teilgebiet der KI ermöglicht es, völlig neue Bilderkennungsaufgaben zu lösen. Diese Entwicklung macht sich das Projekt VTRACS für die Erkennung von Fahrzeugen zu nutze.

Bis vor wenigen Jahren wurde Objektdetektion großteils durch aufgabenspezifische Feature Detektoren in Kombination mit einem guten Klassifikator (bspw. *Support Vector Machines*) kombiniert. Diese Detektoren generalisieren aber relativ schlecht, sie sind auf wenige Objektklassen beschränkt und können nur mit beschränkten Mengen an Daten-Samples trainiert werden.

Die visuelle Objektdetektion und Klassifikation hat seit dem Jahr 2012 eine Revolution erlebt, in diesem Jahr wurde von Krizhevsky die erste bahnbrechende Publikation zur KI-Methode *Deep Learning* veröffentlicht¹. **Obwohl klassische Methoden der Bildanalyse immer noch Gültigkeit in einigen Bereichen der visuellen Analyse haben** (vor allem in der industriellen Qualitätsinspektion), **verdrängen Deep Learning Ansätze** seitdem praktisch **alle bisherigen Methoden**.

Die Vorteile von Deep Learning (DL) Ansätzen sind:

[1] ¹ A. Krizhevsky, I. Sutskever, G.E. Hinton: "ImageNet Classification with Deep Convolutional Neural Networks", 2012.

- Die gute **Generalisierbarkeit** hinsichtlich der Erkennung unbekannter Muster.
- Die Möglichkeit, sehr **viele Objektklassen** detektieren zu können.
- Die Kapazität, **sehr viele Trainingsbeispiele ohne Performance-Einbußen** verarbeiten zu können.

Die Nachteile von Deep Learning (DL) Ansätzen sind

- DL Netzwerke sind **komplex** und theoretisch nicht immer zu 100 % verstanden, dies erschwert das Design von Verbesserungen und neuen Strukturen.
- Das **Training von Netzwerken ist sehr rechenaufwändig**, auch wenn Ansätze für Verbesserungen erarbeitet werden.
- Relativ **langsame Inferenz (= Detektion)**, typischerweise muss eine Graphikkarte für Detektion verwendet werden, um akzeptable Rechenzeiten zu erhalten. Auch hier gibt es langsam Ansätze für Verbesserungen, beispielsweise das Rechnen mit geringen Genauigkeiten und das künstliche Ausdünnen (Pruning) von Netzwerken.

1.2. Projektziel

Das Ziel des Projektes VTRACS war die Entwicklung eines Prototyps, welcher die Eignung von Deep Learning Methoden für die automatische Fahrzeug-Detektion demonstriert und das vorhandene Potential von Deep Learning aufzeigt. Es sollte die Vergleichbarkeit visueller Zählung auf der Straße mit anderen, etablierten, Methoden analysiert werden.

Die Performance des VTRACS Prototyps wurde anhand einer möglichst großen Stichprobe aus ASFiNAG Kamera-Rohdaten evaluiert und untersucht. Die Arbeitspakete enthielten folgende Aspekte:

- Design und Training eines echtzeitfähigen Fahrzeugdetektors, der auch auf embedded HW portiert werden kann. Folgende Fahrzeugtypen sollen unterschieden werden: PKW, LKW.
- Laufzeit-Performance Update des bestehende SLR Trackers.
- Erweiterung des SLR Trackers um eine visuelle Fahrzeug Signatur Beschreibung. Ziel ist die automatische Wiedererkennung von Fahrzeugen über Kameras hinweg. Die Generierung von Test- und Trainingsdaten musste für dieses Arbeitspaket manuell erfolgen, wobei jedoch nach Möglichkeit auf bestehende Datensätze zurückgegriffen wurde.

- Autokalibration von Videos mit minimaler User Interaktion für effizientes Deployment in zukünftigen Anlagen. Hauptaufgabe besteht hier vor allem in der automatischen Fahrstreifen Detektion. Dies kann sehr gut über das Clustering von Trajektorien gelöst werden, welches im Rahmen des Projektes implementiert wird.
- Entwicklung eines Demonstrators: Ein Demonstrator für PC und ein embedded Jetson AGX Board wurde implementiert.
- Evaluierung pro Webcam: Vergleich der Verkehrserfassung durch den Prototyp mit Querschnittsdaten aus herkömmlichen Verkehrsdetektoren. Fahrzeugtyp-spezifisch wurden Verkehrsmengen und Geschwindigkeiten miteinander an mehreren Messstellen verglichen.

2. METHODIK

Im Rahmen des Projektes VTRACS wurde die Aufgabenstellung als modulares System gelöst. Die Einzelmodule wurden in Python3 und C++ programmiert und, wo möglich, auf Basis von Open-Source Modulen umgesetzt. Aufgrund der prototypischen Entwicklung sind Teile der Umsetzung experimentell gehalten, um mit wenig Aufwand schnell Änderungen an den Algorithmen bzw. der Datenverarbeitung vornehmen zu können.

Für die **Objektdetektion mit Deep Learning** hat sich bei SLR Engineering das **Google Tensorflow Framework** bewährt, welches dort schon seit drei Jahren im praktischen Einsatz ist. Das Training und die Detektion wurde für VTRACS **auf 2x NVIDIA 2080 Ti GPU Grafikkarten** durchgeführt, die Trainingszeiten für diese HW Konfiguration betrugen einige Tage für den Fahrzeugdetektor.

Der Kern vom VTRACS Demonstrator wurde als C++ Anwendung entwickelt. Teile des Prototyps wurden als Python3 Anwendung implementiert.

Die Trainingspipeline wurde vollständig in Python3 mit Tensorflow und Keras implementiert.

Zusammenfassend wurden folgende Open-Source Komponenten verwendet:

- **Python3** für die generelle SW Entwicklung und Prototyping.
- **C++** im VTRACS embedded Code und Tracker Code.

- **OpenCV** für die Basis Machine-Vision Module (Bild IO, Vorverarbeitung von Bildern, etc.).
- **Tensorflow und Keras** für die Deep Learning Objekt Detektion.
- **Eine InfluxDB Datenbank** für die Speicherung der Tracking Daten in Echtzeit.
- Das **Google protobuf** Protokoll + Implementierung.
- **NVIDIA Cuda** für die Beschleunigung der Rechenoperationen auf den GPUs.

Die Projektstruktur mit den einzelnen Modulen ist in der nachfolgenden Abbildung 1 dargestellt.

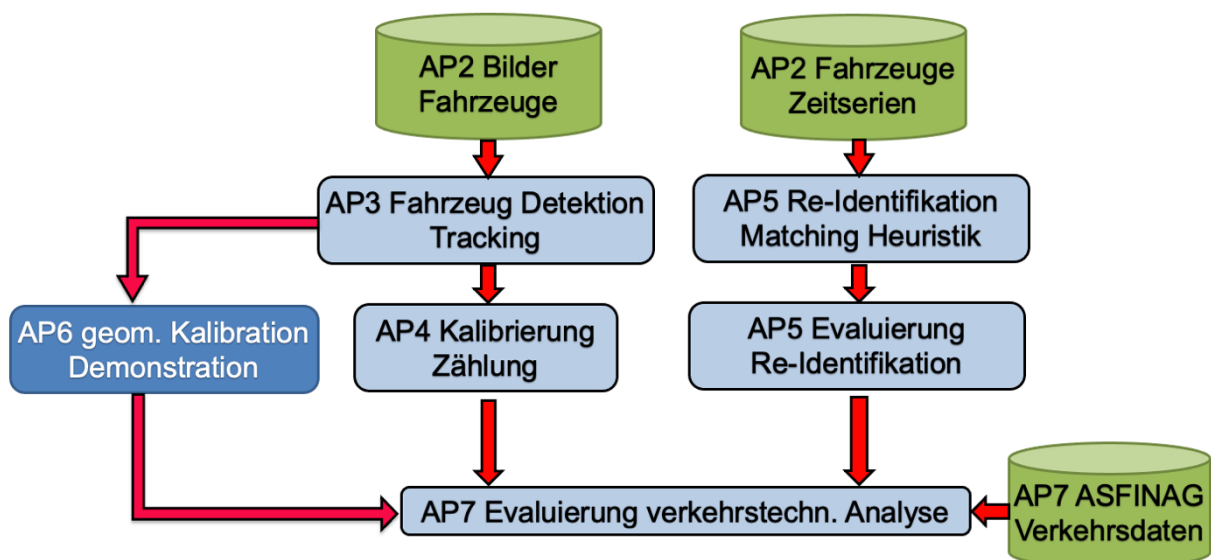


Abbildung 1: Projektstruktur von VTRACS.

2.1. Datengrundlage – Rohdaten/Bestätigte Datensätze

Für die Systementwicklung wurde ein größerer Satz an Bilddaten (Trainingsdaten) benötigt, welcher mit den Software Tools von SLR gesichtet und annotiert wurde. Damit wurde die **Ground Truth für das Training der Netzwerk Modelle** bereitgestellt.

Für das VTRACS Projektbestanden die Daten hauptsächlich aus den Einzelbildern von Fahrzeugen und deren Klassifikation in PKW/LKW.

Die Trainingsdaten setzen sich aus verschiedensten Datenquellen zusammen, wobei auf eigene Bestandsdaten zurückgegriffen wurde und eigene Messungen durchgeführt wurden. Es wurde vor allem darauf geachtet, dass die Trainingsdaten aus unterschiedlichsten Perspektiven, Uhrzeiten und Wettersituationen (soweit verfügbar) gesammelt wurden, um den Detektor möglichst situationsunabhängig trainieren zu können.

Während der Testphasen des Detektors im Winter 2020 und Frühjahr 2021 wurde der Detektor mehrfach mit zusätzlichen Daten trainiert, um die Detektion insbesondere von LKW zu verbessern.

Wir trainierten mit 127012 Samples, die in 6 separaten Annotationsversuchen gewonnen wurden (der erste davon war der größte). Wir haben den Datensatz iterativ verfeinert, indem wir weitere Stichproben von Fällen hinzufügten, die sich in früheren Iterationen als weniger effektiv erwiesen.

Data Augmentation wurde eingesetzt, um die Variabilität der Trainingssamples zu erhöhen: die Trainingsdaten wurden durch zufälliges horizontales Spiegeln und zufälliges Beschneiden der Bilder ergänzt, bevor sie auf die Eingabeform des Modells angepasst wurden.

Die Detektionsmodule werden in diesem Abschnitt im Detail beschrieben. Die Netzwerkstrukturen wurden in Tensorflow/Keras spezifiziert und in Python3 implementiert. Das Training der Netzwerke wurde auf einem schnellen 12-Core PC mit zusätzlich 2x Nvidia 2080 Ti GPUs durchgeführt.

2.2. Fahrzeug Detektion und Tracking

Im Arbeitspaket 2 wurden die Methoden für Fahrzeug Detektion und Tracking implementiert.

2.2.1. Das Detektionsmodul im Detail

Als Fahrzeug-Detektor wurde ein SSD MobileNet V2 Detektor mit 640x640 Eingangsauflösung und der **FPN_{Lite} Merkmalspyramide** gewählt. Diese wurde speziell dafür entwickelt, um in TFLite konvertierbar zu sein und schnell auf mobilen Geräten ausgeführt werden zu können.

FpN_{Lite} wird als Feature-Pyramiden-Aggregator verwendet, um die vom Backbone bei verschiedenen Auflösungen erzeugten Features zu kombinieren. Dies ermöglicht in Summe ein Netzwerk mit der optimalen Auflösung, um einen guten Kompromiss zwischen Inferenzzeit und Qualität zu erhalten.

Wir verwenden **SSD als Detektionsarchitektur** (single-shot, einstufig). SSD hat eine geringere Leistung als zweistufige Architekturen wie *Faster RCNN*, es ist aber viel schneller in der Ausführung und ermöglicht uns den Einsatz auf embedded Systemen.

Wir verwenden **MobileNet V2 als Backbone-Architektur** mit Standardparametern und behalten $\alpha=1$ als Tiefenmultiplikator bei.

Die Eingangsaufösung beträgt **640x640 Pixel**, und passen die Größe der Bilder an diese Auflösung an, **ohne das Seitenverhältnis beizubehalten**.

Das Modell wurde auf 2 Klassen trainiert, "PKW" und "LKW".

Von den 127012 Samples wurden 4504 Samples für die Modellvalidierung reserviert (d. h. aus den Trainingsdaten herausgehalten), um die Leistung des Modells zu bewerten.

Das Modell wurde in mehreren Iterationen trainiert, **ausgehend von Gewichten, die auf dem COCO17-Erkennungsdatensatz vortrainiert wurden**. Jede Iteration des Trainings umfasste 100k Schritte, mit einer Stapelgröße von 16. Bei jeder Iteration wurden neue Daten hinzugefügt, um Bereiche zu berücksichtigen, in denen das Modell Schwächen aufwies (z. B. Lastwagen oder Nachtvorhersage), und das Modell wurde weiter trainiert - ausgehend von den Gewichten, die am Ende der vorherigen Iteration vorhanden waren.

Jede Iteration des Trainings wurde unter Verwendung des SGD-Optimierers durchgeführt, wobei eine "Cosinus-Abnahme mit Aufwärmen"-Politik für die Lernrate verwendet wurde, die bei 0,004 begann und in 250 Schritten auf 0,025 anstieg, um dann in 100k Schritten auf 0 abzufallen. Der Parameter für die Impulsgewichtung wurde auf 0,9 initialisiert.

Für das endgültige Modell gab es 3 Trainingsiterationen, so dass es insgesamt 300k Schritte lang trainiert wurde.

2.2.2. Die Detektion am embedded System

Die Detektion am embedded System erfolgt mit dem gleichen trainierten Netzwerkmodell, jedoch wird es für die Anwendung mit Nvidia Hardware noch auf Tensorflow Lite (TFLite) angepasst.

Das Modell wird zunächst an die Kompatibilität mit TFLite angepasst (über die *TF Object Detection API*) und dann anhand eines Beispieldatensatzes von 100 Bildern (aus dem Pascal-VOC-Erkennungsdatensatz) quantifiziert. Die Beispieldaten werden dafür in das Netzwerk eingespeist und zur Schätzung der möglichen Wertebereiche für alle float-Variablen im

Netzwerk verwendet. Mit diesen Informationen wird dann die Quantisierung so vorgenommen, dass die beobachteten Bereiche für jede Variable optimal kodiert werden können.

Die Quantisierung der Netzwerke bedingt einen Qualitätsverlust von ca. 2-3 %, der jedoch in Summe auf die Leistung des Systems keinen signifikanten Einfluss hat.

2.2.3. Fazit für das Arbeitspaket Fahrzeugdetektion und Tracking

Ausgehend von den vorhandenen Detektoren und dem bestehenden Tracker-Modul konnte in diesem AP die Detektion für PKW und LKW auf ein Niveau gebracht werden, das für praktische Anwendungen ausreichend sein sollte.

Innerhalb einiger Monate wurden mehrere Iterationen aus Sample Erzeugung/Training durchgeführt, jedes Mal mit einer klar erkennbaren Verbesserung der Gesamt Detektion. **In einigen Situationen ist die Detektion gut (> 95%), aber noch nicht perfekt (> 99%). Die Erfahrung aus den Trainingsläufen zeigt uns aber, dass noch fehlende Genauigkeiten sicherlich mit weiterem Finetuning deutlich verbessert werden kann.** Im Rahmen des VTRACS Projektes mussten hier aufgrund begrenzter Ressourcen Kompromisse gemacht werden.

Selbst die Detektion in Nachtbedingungen mit Regen liefert bereits jetzt gute Ergebnisse. **Insbesondere unter diesen Bedingungen wurde das Potential des Trainings bei weitem noch nicht ausgeschöpft.**

2.3. Kalibrierung und Zählung

Im Allgemeinen besteht die Aufgabe der Kalibrierung in der Analyse der vom Trackermodul zur Verfügung gestellten Trajektorien und Bereitstellen von Informationen über Bereiche im Kamerabild, wo Trajektorien zu finden sind, bzw. wo Einzelfahrstreifen situiert sind. Das Zählmodul benutzt diese Kalibrierung um in weiterer Folge im Echtzeitbetrieb abgeschlossene Trajektorien (Fahrzeug nicht mehr im Bild) zu analysieren und die Ergebnisse in der InfluxDB zu speichern.

2.3.1. Datengrundlage

In der Entwicklungsphase des Prototyps konnte aus datenschutzrechtlichen Gründen nicht im notwendigen Ausmaß auf Livekamera-Ansichten zurückgegriffen werden. Aus diesem Grund wurde die Kalibration und Zählung zunächst mit dem in AP6 verwendeten Datensatz der TU Brno entwickelt und in weiterer Folge getestet. Mit Fertigstellung der Entwicklung von Version 1 wurde für die weitere Entwicklung und Validierung der Datensatz aus der Messkampagne A2 Feldkirchen herangezogen. Dieser entspricht in Winkel und Positionierung einem größeren

Teil der im Streckennetz eingesetzten Kameras und stellte damit eine gute Validierungsgröße dar. Die weitere Entwicklung erfolgte basierend auf unterschiedlichen Kamerapositionen (vgl. Abbildungen 2 und 3 unten).



Abbildung 2: Messkampagne A2 Feldkirchen Situation 1.



Abbildung 3: Messkampagne A2 Feldkirchen Situation 1.

Für die Kalibration der Kameraposition und Erkennung der Fahrstreifen, werden abgeschlossene Trajektorien für einen Zeitbereich von ca. einer halben Stunde benötigt, um sicherzustellen, dass Beispiele für alle relevanten Fahrstreifen in ausreichender Anzahl vorhanden sind.

2.3.2. Methodik Kalibration

Die für die Kalibration vom VTRACS Tracker zur Verfügung gestellten Trajektoriendaten (vgl. Abbildung 4) bestehen aus in unregelmäßigen Abständen erstellten XY-Koordinaten für jeweils eine Trajektorie (entspricht einem Kraftfahrzeug) und zusätzlichen Informationen, wie Fahrzeugklasse, Konfidenzen, sowie Höhe und Breite der das Kraftfahrzeug umschließenden Bounding Box.

```

1 400004;1900-00-00@00:00:00.000;21;752;371;130;121;0.952729;0;0;0;c
2 400004;1900-00-00@00:00:00.000;22;752;374;131;122;-100;-0.00415039;3.71262;1;c
3 400004;1900-00-00@00:00:00.000;23;751;378;132;123;-100;0.0752563;3.96384;1;c
4 400004;1900-00-00@00:00:00.000;24;751;381;133;124;-100;0.153076;3.91391;1;c
5 400004;1900-00-00@00:00:00.000;25;750;385;134;124;-100;0.0203857;3.98688;1;c
6 400004;1900-00-00@00:00:00.000;26;750;388;135;125;-100;0.0161743;4.11145;1;c
7 400004;1900-00-00@00:00:00.000;27;750;392;136;126;-100;0.161987;4.07385;1;c
8 400004;1900-00-00@00:00:00.000;28;749;396;137;127;-100;0.0360718;4.38574;1;c
9 400004;1900-00-00@00:00:00.000;29;749;400;138;128;-100;0.0285645;4.25192;1;c
10 400004;1900-00-00@00:00:00.000;30;748;404;139;129;-100;0.0786743;4.17044;1;c
11 400004;1900-00-00@00:00:00.000;31;748;407;140;130;-100;0.00720215;4.44897;1;c
12 400004;1900-00-00@00:00:00.000;32;747;412;141;131;-100;0.0723877;4.58463;1;c
13 400004;1900-00-00@00:00:00.000;33;747;416;142;132;-100;0.0396729;4.52121;1;c
14 400004;1900-00-00@00:00:00.000;34;746;420;143;133;-100;0.0672607;4.53583;1;c
15 400004;1900-00-00@00:00:00.000;35;746;424;144;134;-100;0.00500488;4.82251;1;c
16 400004;1900-00-00@00:00:00.000;36;744;425;146;139;0.986878;0.186523;4.72827;1;c
17 400004;1900-00-00@00:00:00.000;37;743;429;147;140;-100;0.0126953;4.82501;1;c
18 400004;1900-00-00@00:00:00.000;38;743;434;148;141;-100;0.0692139;4.98108;1;c
19 400004;1900-00-00@00:00:00.000;39;743;436;148;146;0.988248;0.0131226;4.85425;1;c
20 400004;1900-00-00@00:00:00.000;40;743;441;149;147;-100;0.145874;5.31281;1;c
21 400004;1900-00-00@00:00:00.000;41;742;445;150;148;-100;0.112549;5.08691;1;c
22 400004;1900-00-00@00:00:00.000;42;740;448;155;150;0.984898;0.034668;4.91632;1;c
23 400004;1900-00-00@00:00:00.000;43;739;453;156;151;-100;-0.0236206;5.52332;1;c
24 400004;1900-00-00@00:00:00.000;44;739;458;158;153;-100;-0.00512695;5.52155;1;c
25 400004;1900-00-00@00:00:00.000;45;739;461;155;157;0.977562;0.0691528;5.31891;1;c

```

Abbildung 4: Beispiel Aufbau Datensatz.

Für die Kalibrierung des Kamerabildes wurde der Ansatz von Knoop et al.² aus dem Jahr 2017 herangezogen und für den konkreten Einsatzzweck modifiziert. Die Veränderung wurde notwendig, da die konkrete Fragestellung perspektivische Aspekte beinhaltet, welche im Falle von GPS-Trajektorien keine Problemstellung darstellen.

Die angepasste Methode funktioniert zusammengefasst über die Bildung mehrerer künstlicher Querschnitte in einem 90° Winkel über alle Fahrstreifen einer Fahrtrichtung (vgl. Abbildung 5).

² Knoop, V. L., De Bakker, P. F., Tiberius, C. C., & Van Arem, B. (2017). Lane determination with gps precise point positioning. IEEE Transactions on Intelligent Transportation Systems, 18(9), 2503-2513. doi:10.1109/tits.2016.2632751

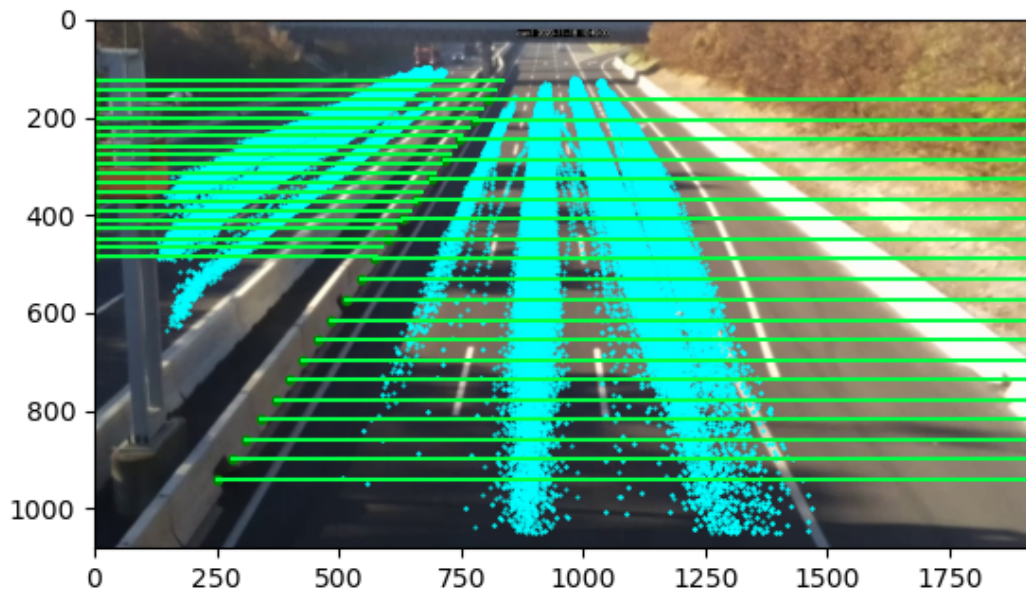


Abbildung 5: Beispiel Trajektorien mit Testquerschnitten.

Auf diesen Querschnitten werden Schnittpunkte mit Trajektorien berechnet, welche mittels einer Dichtefunktion pro Querschnitt visualisiert und analysiert werden (vgl. Abbildung 6).

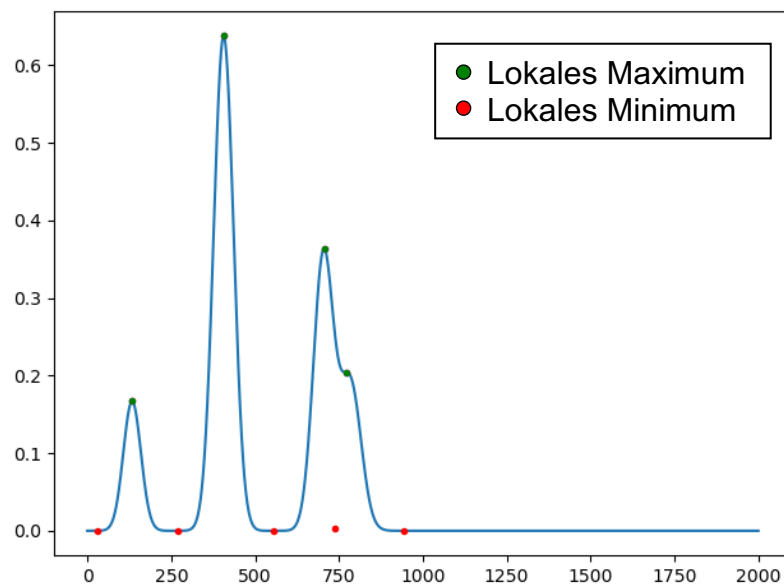


Abbildung 6: Beispiel Dichtefunktion für einen Testquerschnitt.

Die Dichtefunktionen sämtlicher Querschnitte werden auf deren lokale Minima und Maxima hin untersucht. Für die Parameter der Fahrstreifen auf dem jeweiligen Querschnitt werden daraufhin die lokalen Minima herangezogen (vgl. Abbildung 7).

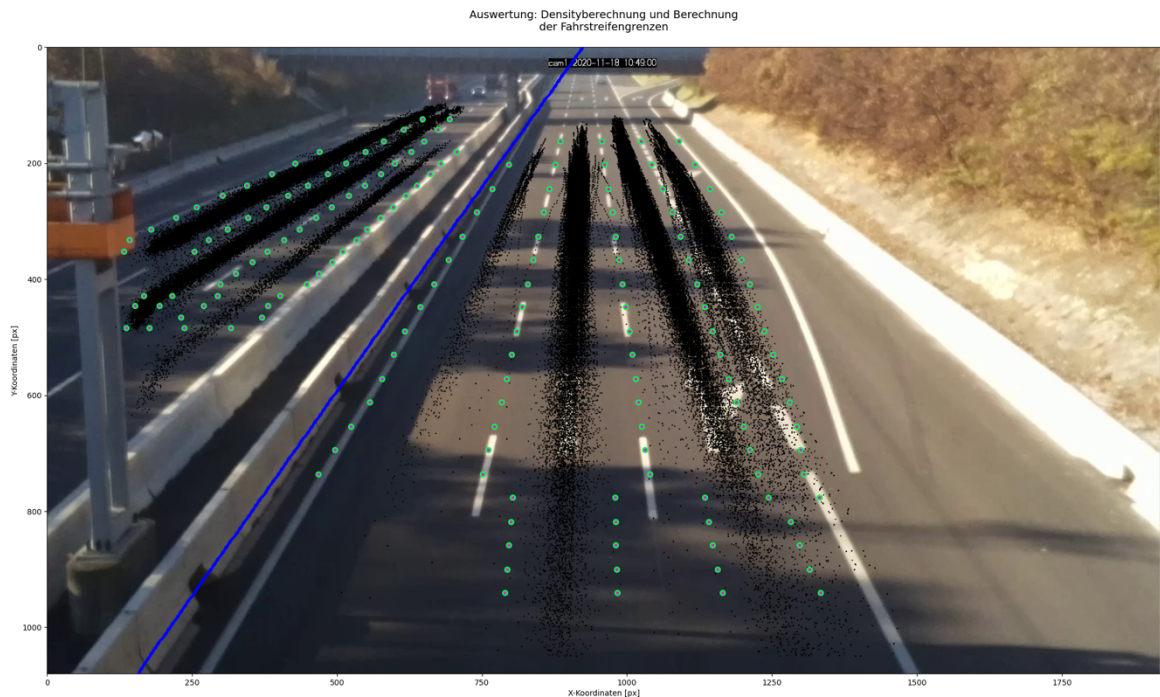


Abbildung 7: Beispiel Fahrstreifenerkennung auf mehreren Querschnitten.

Die Fahrzeugerkennung weist in verschiedenen Bildabschnitten unterschiedliche Erkennungsraten auf. Durch diesen Umstand ist auch die Fahrstreifenerkennung an diversen Querschnitten einer Qualitätsabweichung unterworfen und bedingt einer gezielten Querschnittsauswahl für die spätere Fahrzeugzählung.

Nach der Fahrstreifenparameterermittlung werden die einzelnen Querschnitte auf Basis von Plausibilität (z.B. mehrere Querschnitte in Folge haben vergleichbare Parameter) und Qualität des Zählergebnisses (z.B. maximale Anzahl an erkannten Fahrzeugen) beurteilt. Mit dieser Beurteilung werden für jede Fahrtrichtung und Fahrzeugkategorie (PKW und LKW) ausgewählt (vgl. Abbildung 8) und als Information für die Zählung in einer JSON-Datei gespeichert.

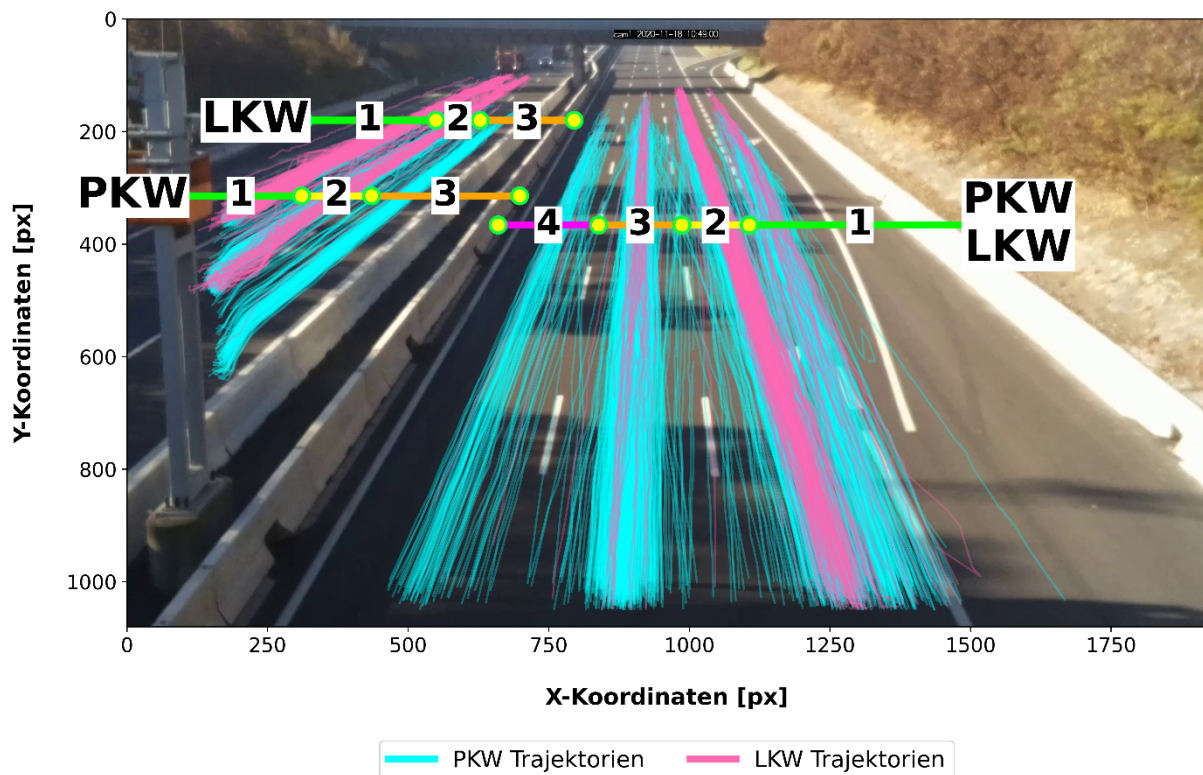


Abbildung 8: Beispiel Bildkalibrierung.

2.3.3. Detailliertes Ablaufschema Kalibration

Um eine sofortige Zählung nach Inbetriebnahme sicherzustellen, wurde ein Top-Down-Approach gewählt. Demnach werden in der ersten Stufe nach Inbetriebnahme Fahrzeuge basierend auf der Fahrtrichtung gezählt, sodass Richtungsfahrbahnen unterschieden werden können. Dies stellt sicher, dass auch vor der Kalibration sofort Zählergebnisse verfügbar sind. In der zweiten Stufe erfolgt, nach Aufzeichnung einer ausreichenden Anzahl an Trajektorien, die Kalibration auf Basis dieses Datenpaketes. Nach Fertigstellung der Kalibration werden zusätzlich zur richtungsbasierten Zählung ebenso Informationen zum gewählten Fahrstreifen gespeichert. Ebenso enthalten ist eine prototypische Einschätzung, ob Fahrstreifen 1 einen Beschleunigungs- oder Verzögerungsstreifen darstellen könnte.

Wie vorhin angeführt bedarf es für die Kalibrierung einer Datenaufzeichnung von ca. 30-60 min um sicherzustellen, dass alle Fahrstreifen mit einer Mindestanzahl von Fahrzeugen befahren wurden.

Ist dieser Schritt erfolgt, wird die Kalibration (vgl. Abbildung 9) vom VTRACS-Prototyp angestoßen:

1. Zunächst wird die gefahrene Richtung aller Trajektorien in der Datenaufzeichnung bestimmt.
2. Danach wird ein aus den Trajektorien generiertes Bild analysiert und einzelne Bereiche identifiziert und die spezifische Fahrtrichtung bestimmt.
3. Um die Analyse der Bereiche zu erleichtern, erfolgt daraufhin eine Trajektorienfilterung mittels Gewichtung von einzelnen Bereichen.
4. Im nächsten Schritt werden für die Bereiche 20 Querschnitte entlang der Y-Achsenausdehnung gebildet.
5. Diese Querschnitte dienen als Referenzlinien, entlang deren Schnittpunkte mit den gefilterten Trajektorien berechnet werden.
6. Über die Berechnung der Dichte der Trajektorienschnittpunkte und Lagebestimmung werden die jeweiligen Fahrstreifenparameter bestimmt.
7. Nachdem dieser Schritt für alle 20 Querschnitte des Bereiches durchgeführt wurde, wird der beste Querschnitt nach Zählergebnis und Plausibilität bestimmt.
8. Den Abschluss bildet die Evaluierung des 1. Fahrstreifens, ob es sich dabei um einen Beschleunigungs- oder Verzögerungsstreifen handeln könnte. Dies wurde hier jedoch nur prototypisch umgesetzt und konnte aufgrund von zu wenig unterschiedlichen Situationen nicht evaluiert werden.
9. Nach Fertigstellung der Kalibration werden die Information in einem JSON-File mit dem Kameranamen gespeichert.

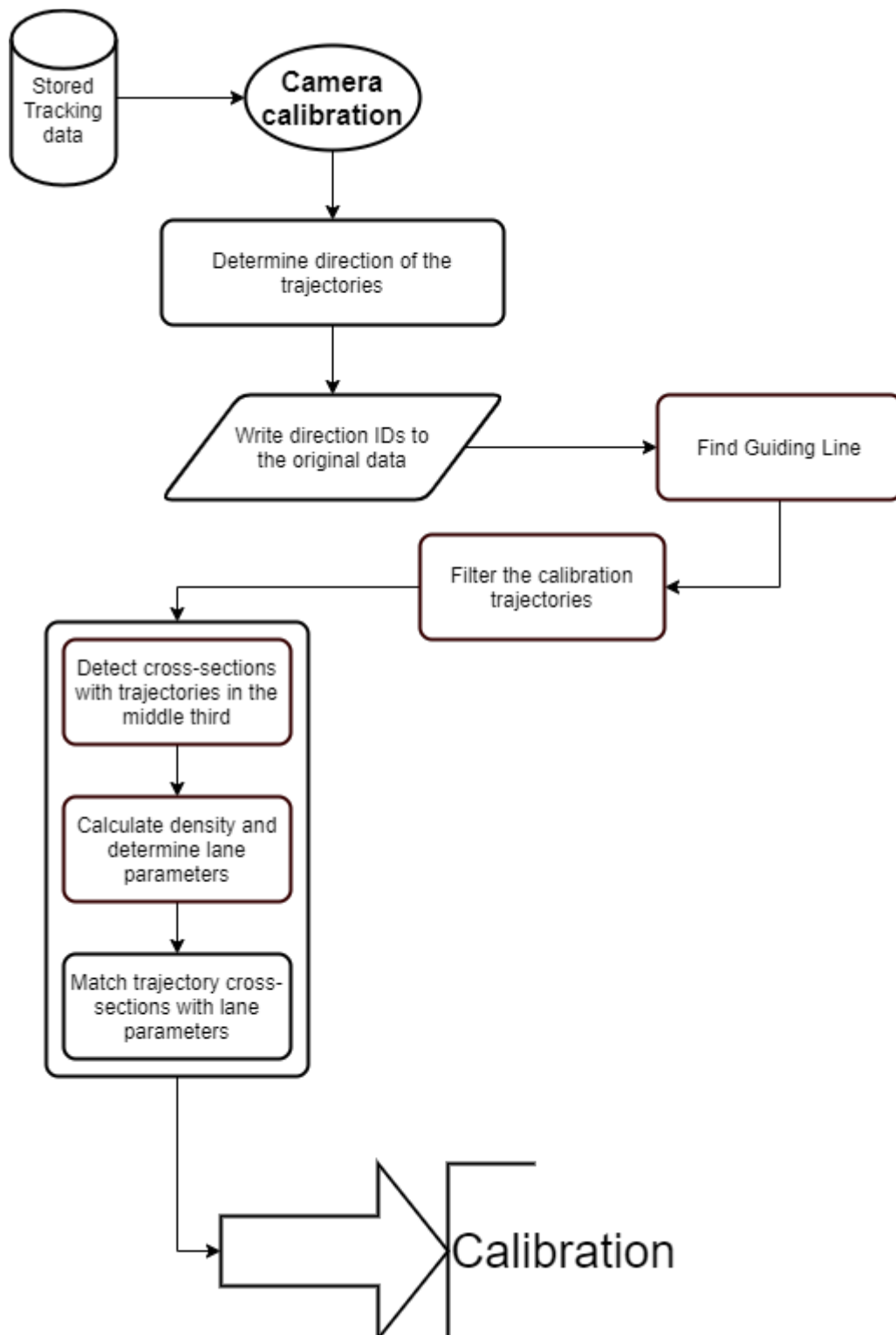


Abbildung 9: Flow-Chart Kalibration.(Methodikerweiterung auf Grundlage von Knoop et al., 2017)

2.4. Fahrzeugzählung

Im VTRACS Prototyp ist die Fahrzeugzählung nach dem GRPC Server-Client-Prinzip aufgebaut. Ist eine Trajektorie durch den Tracker erfasst wird diese mit dem Google Protobuf Protokoll an den Server geschickt, dessen integrierte Analysefunktion die Trajektorie hinsichtlich Richtung und (Kalibration vorausgesetzt) Fahrstreifen bestimmt, sowie im Anschluss in der InfluxDB gespeichert.

Als Basis für die Fahrzeugzählung wird die Bestimmung der Richtung herangezogen, diese erfolgt über die Beurteilung der Änderung der Y-Koordinate der jeweiligen Trajektorie. Abhängig von der Fahrzeugkategorie und der Richtung wird dann im Anschluss mit den Kalibrationsdaten die Kalibration geladen. Die Kalibrationsdaten enthalten XY-Koordinaten anhand deren Linien gebildet werden, welche die Fahrstreifen darstellen. Beurteilt wird, ob und an welcher Stelle die Trajektorie eine dieser Linien schneidet.

Die Fahrzeugzählung kann sofort nach dem Hochfahren des Prototyps gestartet werden und Daten werden zeitgleich in die Datenbank geschrieben. Eine vorherige Kalibration ist für den Prototyp nicht zwingend notwendig, geht jedoch mit einem Informationsverlust einher. Ohne Kalibration erfolgt eine Zählung rein auf Basis der Fahrtrichtung relativ zur eingesetzten Kamera (Fahrtrichtung 0: von der Kamera entfernend / Fahrtrichtung 1: auf die Kamera zufahrend). Ermittelt werden allgemein die Fahrzeugkategorie, gefahrene Richtung, sowie ein Zeitstempel mit zugehöriger Trajektorien-ID.

Nach einer, wie vorangehend beschriebenen, Kalibration des Kamerabildes wird zusätzlich der benutzte Fahrstreifen des Fahrzeuges ermittelt. In die Datenbank wird nach erfolgter Untersuchung der Trajektorie jeweils ein Datenpunkt pro Trajektorie geschrieben.

2.5. Re-Identifikation mittels visueller Signatur

Das Ziel des Arbeitspaketes AP5 war es, einen Algorithmus für Wiedererkennung von Fahrzeugen als Teil des VTRACS Projekts zu entwickeln. Die Grundlage soll eine vollkommen anonymisierte visuelle Signatur darstellen. Diese Signatur muss eindeutig genug sein, um Fahrzeuge wiedererkennen zu können, jedoch trotzdem keine Identifikation des Fahrzeugs erlauben, um bei der geplanten Anwendung der Reisezeitmessung den strengen Datenschutzerfordernissen genüge tun zu können.

Die Objekt-Re-ID-Technologie hat eine wichtige Forschungsbedeutung in mehreren Bereichen, aber besonders in Multi-Objekt Tracking. Fahrzeug-Reidentifikation bezieht sich

auf das Problem der Identifizierung desselben Fahrzeugs in einer großen Fahrzeugdatenbank bei gegebenem Referenzfahrzeugbild.

Eine zufriedenstellende Lösung für AP5 konnte mit folgendem Ansatz gefunden werden:

1. Die Fahrzeugmenge in gesuchten Zeitbereich wird zuerst verkleinert, in dem das **optimale zeitliche Suchfenster abgeschätzt** wird. Eine zukünftige Implementierung muss dieses Suchfenster mithilfe der laufenden Reisezeitmessung adaptieren bzw. neu initialisieren, falls nötig. Dies wurde im laufenden AP nicht implementiert, um sich auf die Hauptaufgabe der visuellen Signatur konzentrieren zu können.
2. Für den Vergleich von Referenzbild und den Bildern im Suchbereich wurde der Feature-Deskriptor HOG (Histogram of Oriented Gradients) und zusätzlich Farbinformation verwendet.

Die Ergebnisse und die Schlussfolgerung des Versuchs werden in folgenden Abschnitten ausführlich beschrieben.

HOG oder Histogram of Oriented Gradients ist ein Feature-Deskriptor, der häufig zum Extrahieren von Features aus Bilddaten verwendet wird. Die Idee des HOG Algorithmus besteht darin, dass das Erscheinungsbild eines Objekts durch die Verteilung von Intensitätsgradienten innerhalb kleinräumiger rechteckiger Bereiche eines Bildes modelliert werden kann. Der HOG Deskriptor konzentriert sich auf die ungefähre Struktur bzw. die Form eines Objekts. Durch die räumliche Quantisierung in Zellen bzw. Blöcke und die Quantisierung der möglichen Kanten-Richtungen wird eine Verallgemeinerung und Vereinfachung der Objekt-Beschreibung erreicht.

Der HOG Algorithmus erfordert die Teilung des Bildes in kleine verbundene Regionen (Zellen). Für jede Zelle wird das Histogramm der Gradienten-Orientierung berechnet. Diese Histogramme werden dann zusammengefügt, um den finalen Feature Vektor zu erhalten.

Die Hauptschritte bei der Berechnung eines HOG Feature Vektors sind:

- Normalisierung des Fahrzeug-Bildes in eine feste Breite und Höhe. Hier wurden 64x64 px Größe gewählt.
- Berechnung des Gradienten in x und y Richtung, und deren Quantisierung in 9 Richtungsquadranten.
- Summierung der Gradienten in den Zellen.
- Kontrastnormalisierung überlappender räumliche Zellen, benachbarte Zellen werden zu einem Superblock zusammengefügt und der entstehende Vektor normalisiert.
- Sammeln aller HOGs, um den endgültigen Feature Vektor zu bilden, danach wieder Normalisierung auf Einheitsvektor.

Die Dimension des Vektors hängt von den gewählten Parametern für HOG Algorithmus ab, die wichtigsten Parameter sind dabei:

- Die Orientierungen (in unserem Fall = 9)
- „pixels_per_cell“ (in unserem Fall = (8x8))
- „cells_per_block“ (in unserem Fall = (2x2))

Da sich HOG nur auf die Form und Struktur des Objekts konzentriert und die Webcam Bilder mehrfarbig sind, wurden die Farben zusätzlich zum Feature Vektoren hinzugefügt. Es wurden die Histogramme für einzelne Farbkanäle (Rot, Grün und Blau) berechnet, zusammengefügt und auf Einheits-Vektoren der Länge 1 normiert.

Im letzten Schritt wird der HOG Feature Vektor mit den Farbhistogrammen des Fahrzeugbildes zusammengefügt und der resultierende Vektor wird auch zum Schluss noch einmal auf die Länge 1 normiert. Die folgende Abbildung 10 beschreibt die sich daraus ergebende 2532-dimensionale Fahrzeugsignatur. Für einen Vergleich zweier Fahrzeuge wird die Distanz von deren Signatur im Feature Raum berechnet. Je ähnlicher zwei Vektoren, desto ähnlicher sollten auch die tatsächlichen Fahrzeugbilder sein, was eine visuelle Überprüfung in der Praxis auch bestätigt.

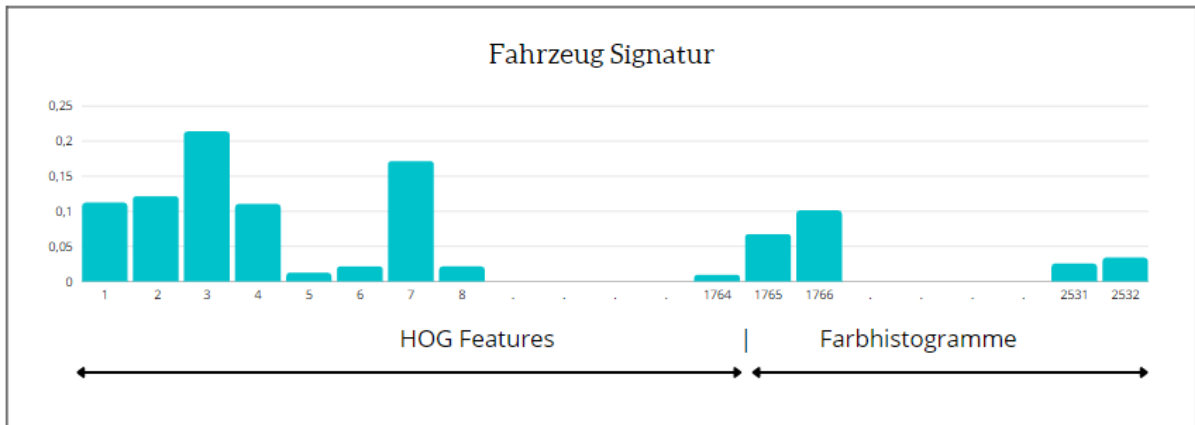


Abbildung 10: Die Struktur der visuellen Fahrzeug Signatur.

Das Ergebnis des VTRACS Object-Trackers beinhaltet alle Informationen über Fahrzeuge in einem Video. Für jedes Videoframe werden die Positionen, Höhe und Breite der Bounding-Box für jedes Objekt im Bild gespeichert. Anhand dieser Informationen werden die Bilder aller Fahrzeuge aus dem Video (siehe Beispiele in Abbildung 11) ausgeschnitten. Da die Fahrzeuge nicht immer den gleichen Abstand von der Kamera haben, weisen die Bounding-Boxes nicht immer dieselbe Größe auf. Deswegen wird jedes Fahrzeugbild auf die Referenz-Größe von 64x64 px skaliert - somit ist gewährleistet, dass alle visuellen Signaturen vergleichbar sind.



Abbildung 11: Bounding-Box Bilder des VTRACS Objekt Trackers. Je nach Entfernung unterscheiden sich die Auflösungen bzw. Größen der Fahrzeugbilder.

Anhand der Distanz von Kameras und der bekannten (geschätzten) Geschwindigkeiten unter regulären Bedingungen wird der Zeitraum eingeschätzt, wann dieses Fahrzeug wieder in der zweiten Kameraposition auftauchen könnte. **Dieser Zeitraum wird als zeitliches Suchfenster definiert.**

Für die Messung werden alle Bilder von detektierten Fahrzeugen im Suchbereich verarbeitet, bzw. ihre Signaturen berechnet. Mit Hilfe der euklidischen Distanz zwischen Referenzfahrzeug und Suchfahrzeug im Vektorraum wird das ähnlichste Fahrzeug selektiert – das Referenzfahrzeug gilt dann als identifiziert.

Um eine Unterscheidung zwischen korrekten Match (True Positive, TP) und einem möglicherweise falschen Match (False Positive, FP) treffen zu können, wird die Distanz der Signaturen bewertet. Anhand von Experimenten wurde ermittelt, dass **alle Matches mit einer Distanz > 0.65 verworfen werden können**, d.h. diese Matches sind mit hoher Wahrscheinlichkeit ungültig. In der Praxis hat sich dieses Kriterium, zumindest bei den analysierten Videos, als gut brauchbar herausgestellt.

2.5.1. Praxistests mit aufgezeichneten Video Daten

Für zwei Versuchsserien wurden Videos von zwei unterschiedlichen Standorten verwendet. Für die erste Sequenz wurde auf einer Autobahn bestehende ältere Webkameras verwendet, die einen Abstand von ca. 500 m aufweisen. Umstände wie schlechtes Wetter, Unfall, Dunkelheit sind in den Testdaten nicht vorhanden, jedoch wurde eine Stausituation mit einbezogen.

Im betreffenden Autobahnabschnitt gibt es keine Ab- oder Auffahrten, es müssen sich also alle Fahrzeuge der Referenzkamera auch durch die Suchkamera bewegen. Durch Verdeckungen von LKWs oder Bussen ist es aber sehr wohl möglich, dass ein Fahrzeug in der Suchkamera nicht aufscheint.

Die geschätzte Zeit, in der das Fahrzeug von erster Kamera bis zum zweiten gelangt, ist mit einem Offset von 25 Sekunden geschätzt und mit dieser Zeit als Referenz-Suchfenster definiert. Wegen der möglichen Geschwindigkeitsänderung auf der Strecke wird ein Zeitraum von ± 10 Sekunden rund um die geschätzte mittlere Fahrzeit verwendet, dies definiert den endgültigen zeitlichen Suchbereich.

Jedes Fahrzeug aus dem Video von Kamera 1 wird im Video von Kamera 2 re-identifiziert, indem der beste Match innerhalb des zeitlichen Suchfensters gefunden wird. Die folgende Abbildung 12 zeigt Beispiele für erfolgreiche Re-Identifizierungen.



Abbildung 12: Darstellung von Re-Identifikationen zeigen den Referenzbild von Kamera 1, und die 10 besten Matches.

Um die Fehlerrate zu reduzieren, wird die Re-Identifikationen unter Verwendung eines Schwellwerts für die Matching-Distanz gefiltert. Das beste Ergebnis liefert ein Schwellwert um **0.65**. Dieser Grenzwert ist hauptsächlich vom Aufbau des Signaturvektors abhängig, er dürfte relativ unabhängig von der verwendeten Kamera, Perspektive und Beleuchtung sein.

Die manuelle Auswertung von ca. 30 Minuten Video ergibt folgende Ergebnisse:

Von den insgesamt 1300 Fahrzeugen liegt bei 368 PKW die kleinste Matching-Distanz unter dem Schwellwert von 0.65. **Es können also theoretisch 28,3% der Fahrzeuge re-identifiziert werden.** Dieser Anteil ist vergleichbar mit anderen Methoden zur Re-Identifikation, wie beispielsweise Bluetooth. Von **368** Re-identifikationen, die als **True Positiv** angenommen werden, wurden bei einer manuellen Nachprüfung 43 Fahrzeuge als nicht korrekt ausgewiesen, d.h. die Fehlerrate an falschen Matches liegt bei **11,68%** für PKWs.

Von den insgesamt 217 LKWs liegt bei 37 LKWs die kleinste Distanz unter dem Schwellwert von 0.65. Von **37 True Positiv** wurde **nur ein LKW falsch zugeordnet** (siehe Abbildung 13 unten). Die Fehlerrate liegt daher in diesem Experiment bei **2,7%** für LKWs.



Abbildung 13: ID 504683 von Kamera 1 stimmt **nicht** mit dem ID 504652 von Kamera 2 überein, die beiden LKWs sind sich jedoch extrem ähnlich.

Auswertung einer Stausituation

Um zu prüfen, wie sich der Algorithmus in einer Situation verhält, in der **aufgrund der falschen zeitlichen Lage des Suchfensters keine Matches möglich sein sollten**, wurde eine Stausituation auf der Teststrecke für einen weiteren Versuch gewählt. Das Suchfenster ist hier aufgrund der sich ergebenden viel längeren Fahrzeiten mit 25 Sekunden falsch gesetzt und **sollte daher nur FP produzieren**. Von 200 gematchten Fahrzeugen in diesem Versuch waren 12 davon unter dem Schwellwert von 0.65, also potentielle korrekte Matches. Tatsächlich waren in diesem Ergebnis 2 TP und 10 FP enthalten. **Die Fehlerrate betrug also nur 5%.**

Die Verwendung eines Grenzwertes für die Matching Distanz ist auch bei diesem Versuch eine gute Möglichkeit, TP von FP zu unterscheiden.

Die Sensitivitätsanalyse aus der folgenden Abbildung 14 unten zeigt, welche Abhängigkeiten zwischen Matching Grenzwert, der Anzahl der Fehlidentifikationen und der möglichen Zahl an korrekten Identifikationen besteht. Über die Wahl des Grenzwertes kann in einem gewissen Bereich eine zwischen Abwägung diesen drei Kennwerte getroffen werden.

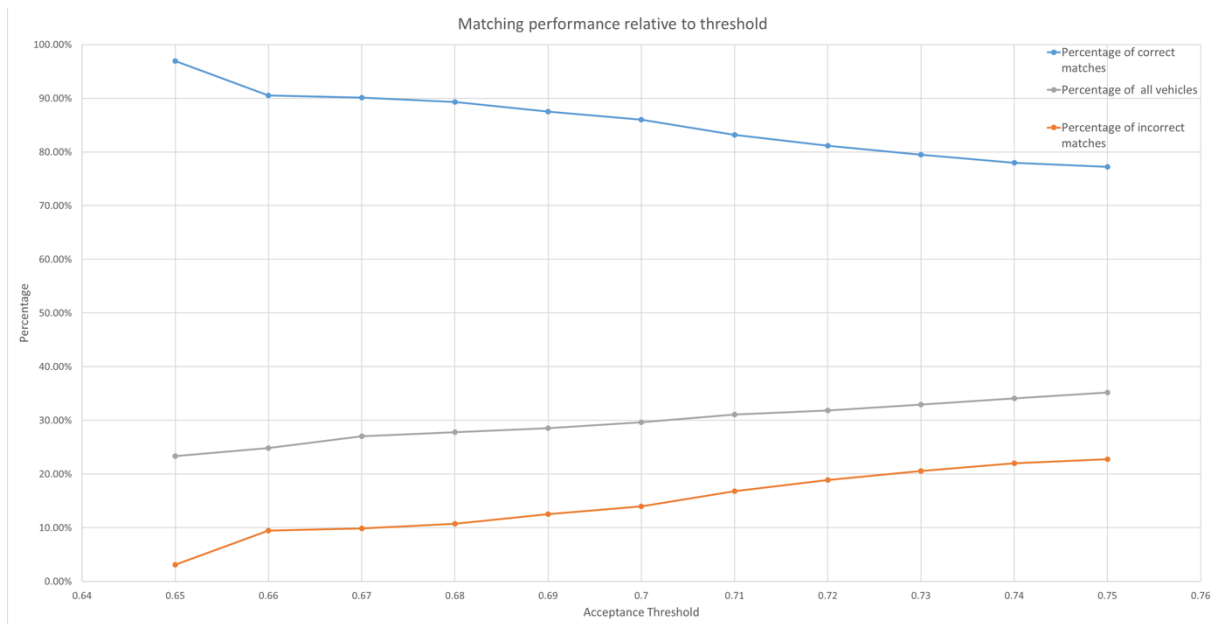


Abbildung 14: Die Performance der Re-Identifikation relativ zu Sensitivitäts Grenzwerten. Ein niedrigerer Grenzwert für akzeptierte Matches führt zu weniger Fehlern bei der Re-Identifikation (orange Kurve), aber auch zu einer geringeren Anzahl an registrierten Fahrzeugen (graue Kurve).

Für alle Versuchsergebnisse ist es noch wichtig zu bemerken, dass die Bildqualität eine sehr große Rolle bei oben genannten Ergebnissen spielt. Die folgenden Abbildungen zeigen Beispiele für falschen Zuordnungen aufgrund der schlechten Bildqualität.



Abbildung 15 Falsche Re-Identifikationen aufgrund schlechter Bildqualität.

In der zweiten Versuchsserie wurden an der Autobahn A2 im Bereich Flughafen Graz Videodaten der Messkampagne vom November 2020 ausgewertet. Die Abbildung 16 unten zeigt die beiden Messpunkte, Bilder der Kameras und überlagert, Fahrzeug Trajektorien für ca. 1h Messdauer.

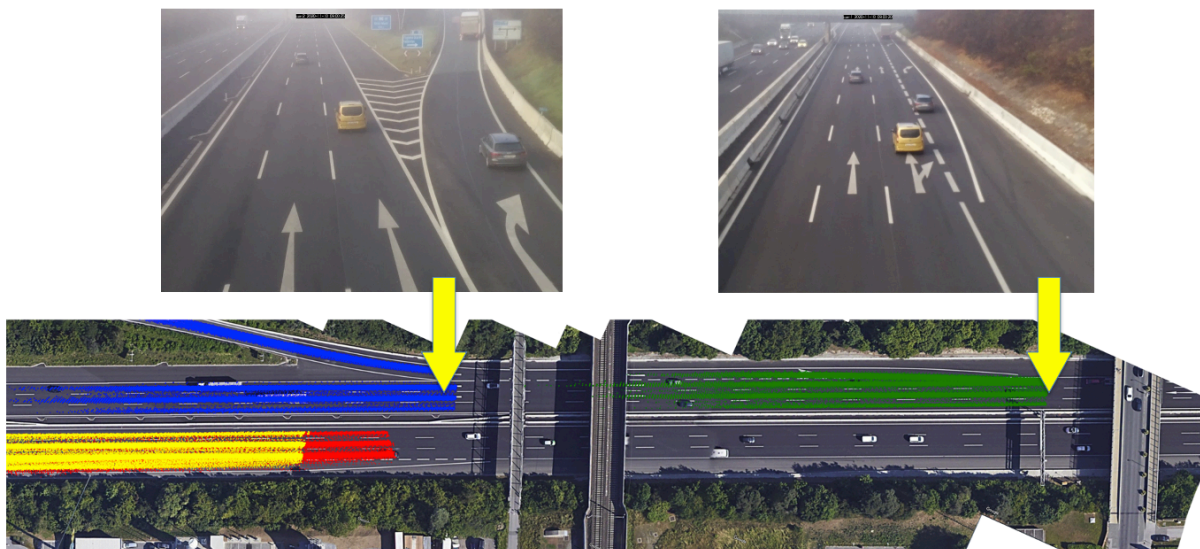


Abbildung 16: Die Messpunkte auf der Autobahn A2 beim Flughafen Graz. Die farbigen Punkte zeigen Messpunkte von detektierten Fahrzeugtrajektorien. Die beiden gelben Pfeile markieren die Punkte, zwischen denen die ‚Reisezeiten‘ gemessen wurden.

Quelle: Schöllberger, A. (2021)³

Die gemessenen Reisezeiten sind im folgenden Plot in Abbildung 17 dargestellt. Die Konsistenz der Messzeiten ist sehr gut, das Verfahren liefert daher plausible Ergebnisse. Alle Messungen aus dem gelben Bereich der Zeitmessungen wurden manuell als Outlier verifiziert.

³ Schöllberger, A.: Untersuchung von Sensortechnologien zur Messung von Fahrtrajektorien auf dem hochrangigen Streckennetz; Masterarbeit an der TU Graz (ISV), 2021.

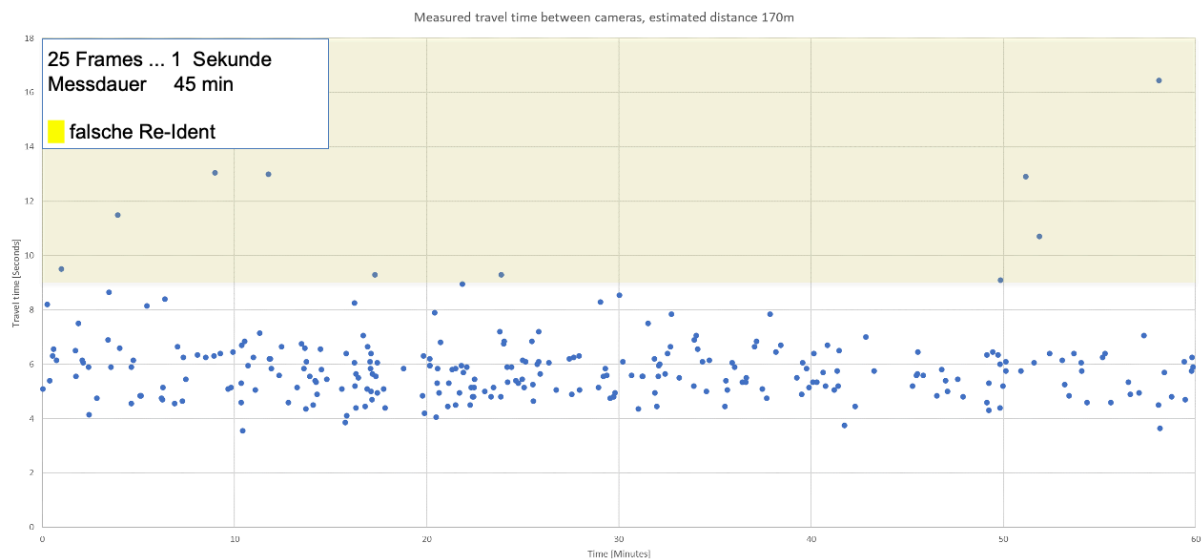


Abbildung 17: Reisezeitmessungen aus der Messkampagne vom November 2020.

2.5.2. Fazit für das Arbeitspaket Visuelle Signatur

In der Praxis werden heute kaum visuelle Methoden für die Re-Identifikation von Fahrzeugen eingesetzt. Grund dafür sind vermutlich vor allem einerseits die großen technischen Hürden, andererseits der Datenschutz. Eine Identifikation über die Kennzeichen der Fahrzeuge wäre eine elegante technische Lösung für ein visuelles System. Dem stehen jedoch einerseits die Verfügbarkeit von entsprechenden hochauflösenden Kamerabildern als auch der Datenschutz entgegen.

Die Methode der visuellen Signaturen bietet dafür eine mögliche Lösung, die mit alternativen technischen Ansätzen (Bluetooth) vergleichbar ist.

2.6. Demonstration der geometrischen Kalibration

Um die Möglichkeiten der vollständigen geometrischen Kalibration von Kameras zu demonstrieren, wurden die Kamerastandorte der Messkampagne vom November 2020 vollständig metrisch kalibriert. Dies ermöglicht uns, einerseits alle Fahrzeugtrajektorien direkt auf Orthobilder zu projizieren (in VTRACS wurden Google Maps Bilder verwendet), andererseits können damit direkt Abstände, Wege und Geschwindigkeiten gemessen werden.

Versuche von SLR Engineering haben gezeigt, dass unter guten Bedingungen Geschwindigkeitsmessungen mit einer Genauigkeit $<1\%$ möglich sind. Unter allgemeinen Bedingungen mit guter Bildqualität sind Messgenauigkeiten von $<3\%$ realistisch.

Das derzeitige Verfahren zur Kamerakalibration basiert auf dem Verfahren von **Tsai**, unter Verwendung von Google Maps als Referenz Koordinaten System. Die Kalibration muss manuell durchgeführt werden und benötigt pro Kamera Standort ca. 30-60 Minuten.

Die kalibrierten Aufnahmen wurden von Schöllberger (2021) im Rahmen seiner Diplomarbeit verkehrstechnisch ausgewertet. Ein Beispiel ist in der folgenden Abbildung 18 dargestellt.

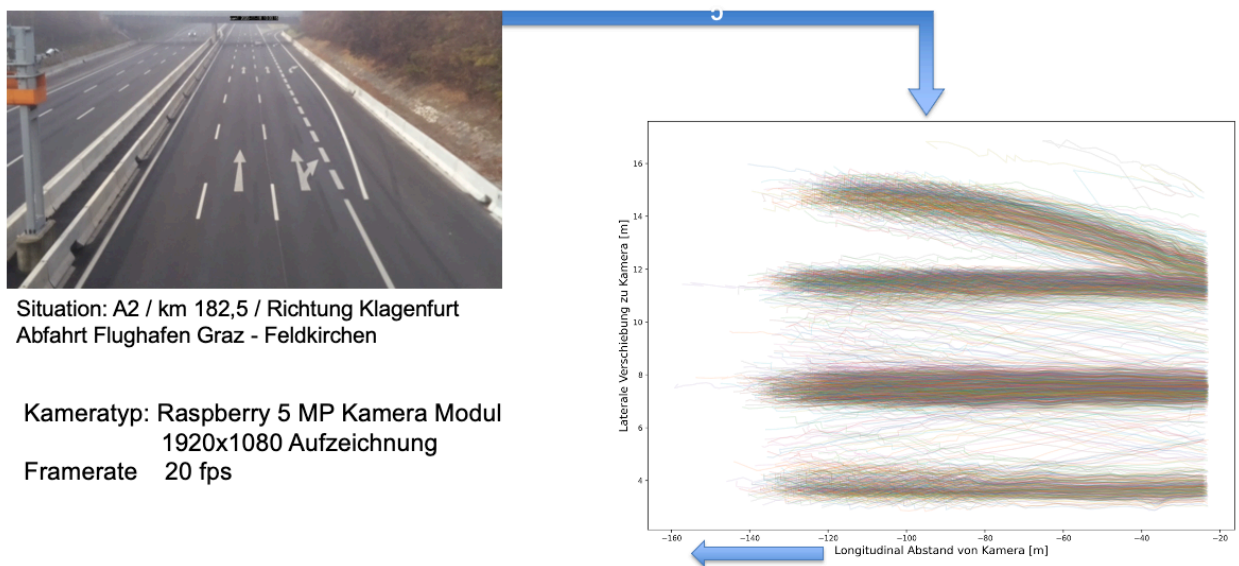


Abbildung 18: Beispiele für kalibrierte Fahrwege von PKW. Mittels der durchgeführten Tsai Kalibration können Fahrzeugpositionen im Bild vermessen werden und sind so für weitergehende verkehrstechnische Analysen zugänglich. (Bildquelle: Schöllberger, 2021)

2.7. Verkehrstechnische Evaluierung

Alle oben genannten Detektionsmodule wurden in einem Software Prototyp integriert, um die Gesamtlösung der Fahrzeug Detektion zu demonstrieren. Der VTRACS Prototyp ist in der VTRACS Distribution enthalten.

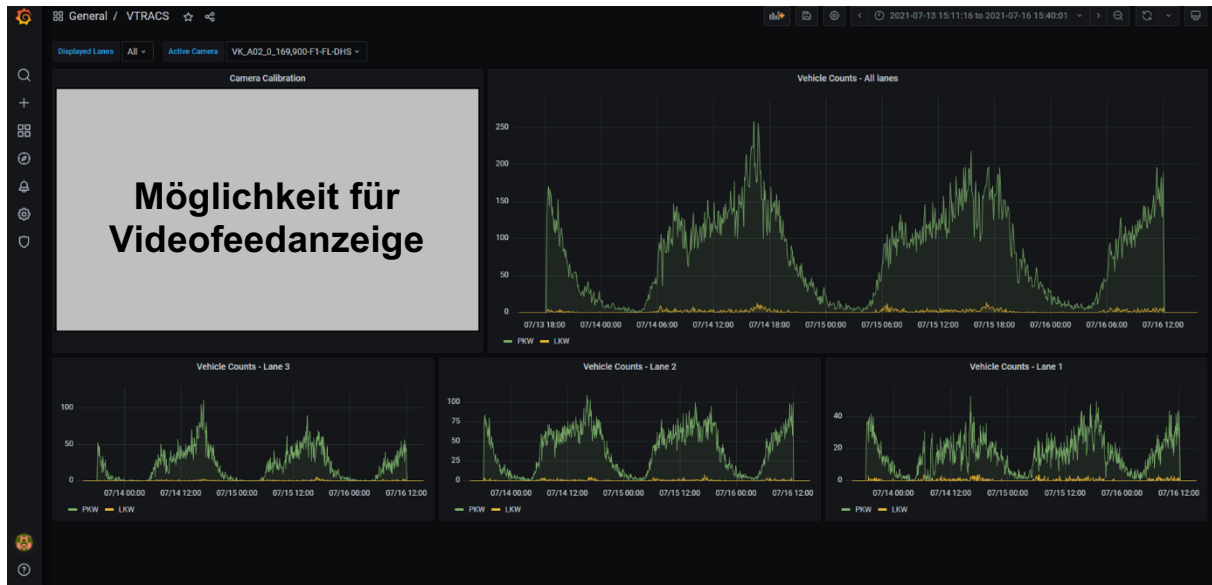


Abbildung 19: Screenshot des VTRACS Prototyps im laufenden Betrieb.

Der VTRACS-Prototyp wurde während und nach der Entwicklung fortwährend hinsichtlich der Zählergebnisse evaluiert und an unterschiedlichen Situationen getestet. Dies deckte Problempunkte auf und ermöglichte eine kontinuierliche Verbesserung des Prototyps. Während der Entwicklung und Evaluierung zeigten sich Probleme in der Beurteilung von „guten“ und „schlechten“ Zählergebnissen. Gerade bei manuellen Zählungen über einen längeren Zeitraum mit hohem Komplexitätsgrad (Unterscheidung Fahrstreifen und mehreren Fahrzeugtypen) können Interpretationsspielräume entstehen, welche zu einer bestimmten Spannweite an Zählergebnissen führen. Auch die Fahrzeugzählung mittels Schleifendetektoren und deren Vergleich mit dem VTRACS-Prototyp ist bestimmten Fehlerraten unterworfen.

Aus diesen Gründen wird der VTRACS-Prototyp sowohl an manuellen als auch an Schleifendetektordaten gemessen und beurteilt.

Nach Abschluss der Prototypentwicklung wurde der VTRACS-Prototyp mit einem Beispiel, dass noch nicht für eine bisherige Entwicklung herangezogen wurde, getestet. Als Beispiel diente die Anschlussstelle Feldkirchen/Flughafen (km 182,99) im Zeitraum 18.11.2020 zwischen 09:00 und 10:00 Uhr. Die diesem Beispiel zugrundeliegende Videoaufnahme wurde

zwei Mal durch das Institut für Straßen- und Verkehrswesen, sowie durch SLR (5 Zählungen gemittelt) manuell gezählt. In der Beurteilung der Ergebnisse (vgl. Abbildung 20, 21 unten) zeigten sich vergleichbare Ergebnisse des VTRACS-Prototyps mit den manuellen Zählergebnissen. Im Allgemeinen weichen auch die manuellen Zählungen voneinander ab, der VTRACS-Prototyp liegt hier in der Spannweite.

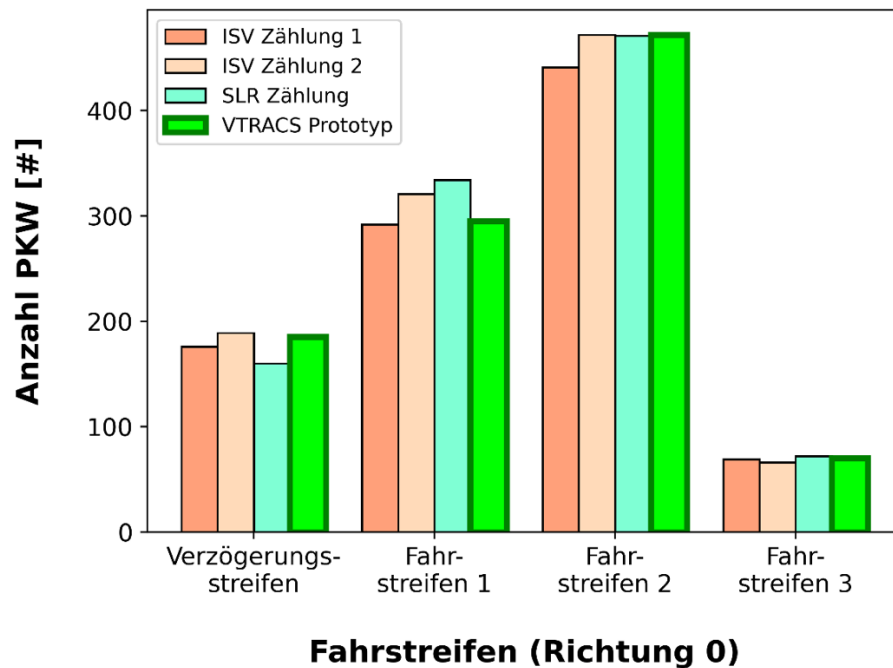


Abbildung 20: Testcase 1 PKW – Vergleich manuelle Zählung und VTRACS-Prototyp.

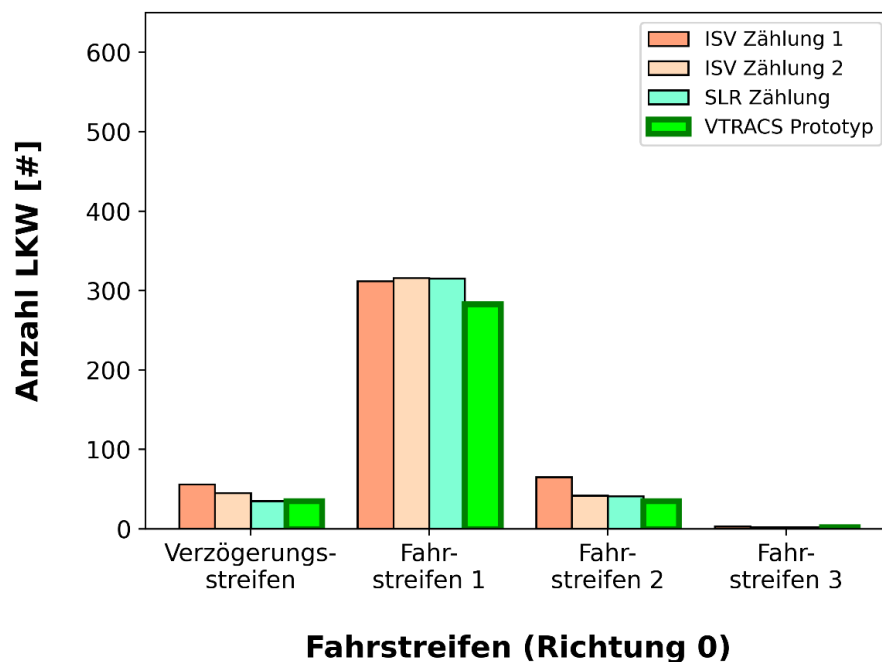


Abbildung 21: Testcase 1 LKW – Vergleich manuelle Zählung und VTRACS-Prototyp.

Der VTRACS Prototyp wurde ab Juli 2021 auf einem Server der ASFiNAG mittels Docker-Containern getestet. Hierfür wurden von Seiten der ASFiNAG vier Testkameras der ALP.Lab-Teststrecke zur Verfügung gestellt:

1. VK_A02_0_168,610-F1-FL-DHS
2. VK_A02_0_168,610-F2-FL-DHS
3. VK_A02_0_169,900-F1-FL-DHS
4. VK_A02_0_169,900-F2-FL-DHS

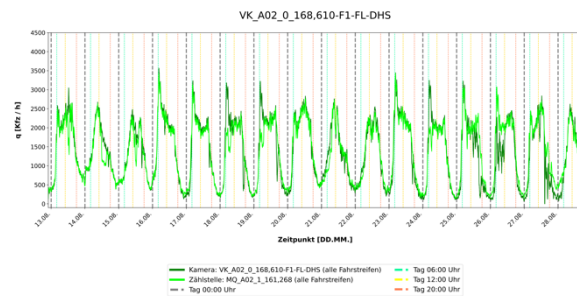
Aufgrund datenschutzrechtlicher Anforderungen war ein direkter Zugriff auf Live-Kamerafeeds nicht möglich, ebenso war es nicht möglich Aufnahmen anzufertigen und im Nachhinein zu analysieren. Daher wurde der VTRACS Prototyp auf dem ASFiNAG-Server direkt betrieben und nur anonyme Trackingergebnisse und Rückschlussmöglichkeit für die weitere Analyse heruntergeladen.

Während der initialen Testphase wurden einige Änderungen an den Kameraeinstellungen (z.B. Blickwinkeländerung vergleichbar zu Straßenbeobachtung, Bildoverlays, etc.), sowie der Laufzeitumgebung durchgeführt um eine optimale Funktion sicherzustellen.

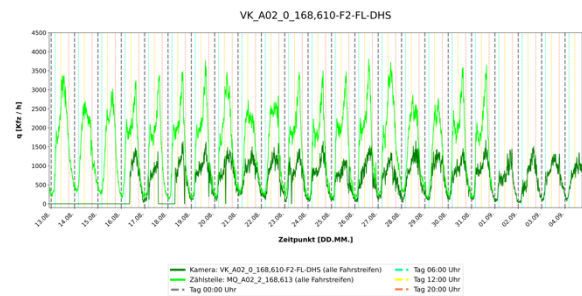
Nach erfolgreicher Anpassung der entsprechenden Parameter (v.a. „cleaner“ Feed und Blickwinkel der Kamera) wurde das Tracking mit August 2021 dauerhaft aktiviert. In den ersten beiden Wochen kam es zu Problemen mit der serverseitigen Dockerlaufzeitumgebung, sodass Probleme mit dem Tracking auftreten und dieses nur sporadisch erfolgen konnte. Die Probleme wurden gelöst und daraufhin konnte ab 13.08.2021 ein durchgehendes Tracking auf allen vier Kameraposition erfolgen. Mit Anfang September (ca. 2 Wochen bisher durchgeführtes durchgängiges Tracking) wurde ein erstes Zwischenfazit im Vergleich mit passenden ASFiNAG-Messquerschnitten gezogen.

Deutlich sichtbar wurden Abweichungen in den Trackinggenauigkeiten zwischen den unterschiedlichen Kamerapositionen. Dies äußerte sich in dem Umstand, dass Kamera „VK_A02_0_168,610-F1-FL-DHS“ ein sehr gut mit dem Messquerschnitt übereinstimmendes Ergebnis lieferte, wohingegen die anderen Kameras eine stete Unterbewertung aufwiesen (vgl. Abbildung 22).

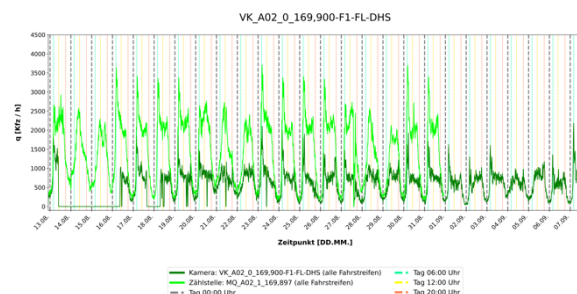
Gutes Trackingergebnis Kamera:
VK_A02_0_168,610-F1-FL-DHS



Schlechtes Trackingergebnis Kamera:
VK_A02_0_168,610-F2-FL-DHS



Schlechtes Trackingergebnis Kamera:
VK_A02_0_169,900-F1-FL-DHS



Schlechtes Trackingergebnis Kamera:
VK_A02_0_169,900-F2-FL-DHS

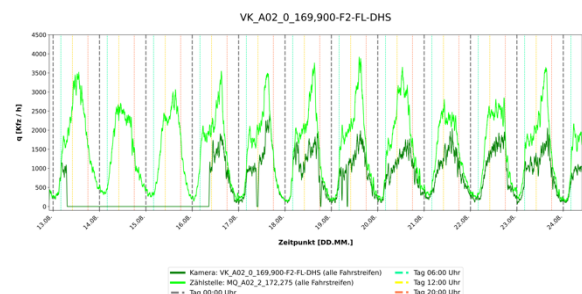


Abbildung 22: Vergleich Trackingergebnisse (dunkelgrün) mit ASFiNAG-Messquerschnitten (hellgrün).

Nach umfassender Fehlersuche konnte die vermutete Fehlerursache auf mangelnde Hardwareressourcen von Seiten des ASFiNAG-Servers eingegrenzt werden. Im Unterschied zur Kamera „VK_A02_0_168,610-F1-FL-DHS“ mit einer Bildauflösung von 640x480 px wurde von den anderen Kameras eine höhere Auflösung von 1280x720 px an den VTRACS Prototyp geliefert. Diese Auflösungen sind mit entsprechenden Hardwareressourcen auch kein Problem für den Prototyp, führen aber in der konkreten Testumgebung zu einem Problem, da die Hardwareressourcen mit dem VMI-Produktionssystem zur Vignettenerkennung geteilt werden und Leistungseinbußen auftreten können.

Als Lösungsansatz wurde im weiteren Verlauf die Qualität der eingehenden Videostreams reduziert und daraufhin der Zeitraum von 29.09.2021 bis zum 07.10.2021 detailliert analysiert. Durch diesen Schritt wurden die Vergleichswerte der einzelnen Kameras mit den korrespondierenden Messquerschnitten der ASFiNAG gleichlaufend über alle vier Kameras.

Aus einer Differenzrechnung (vgl. Abbildung 23) zwischen der Verkehrsmenge mit dem Tracker und dem herkömmlichen Detektor ist erkennbar, dass zu etwa den gleichen Zeiten

der Tracker signifikant weniger Verkehr zählt als mit dem herkömmlichen Detektor. Die ist insbesondere in den Vormittagsstunden am 30.9., 4.10. 6.10 und 7.10. der Fall. Da alle Kameras ein ähnliches Verhalten aufweisen und zu anderen Zeiten kaum Abweichungen zwischen den beiden Messmethoden aufweisen, werden diese Abweichungen mit den bereits genannten Hardwareproblemen begründet. Eine Implementierung des VTRACS Prototypen und dem Produktivsystem einer Vignettenüberprüfung mittels automatisierter Bildauswertung führt zu fehlender Rechnerkapazität, so dass nicht alle Kamerabilder ausgewertet werden und die Fahrzeuganzahl unterschritten wird.

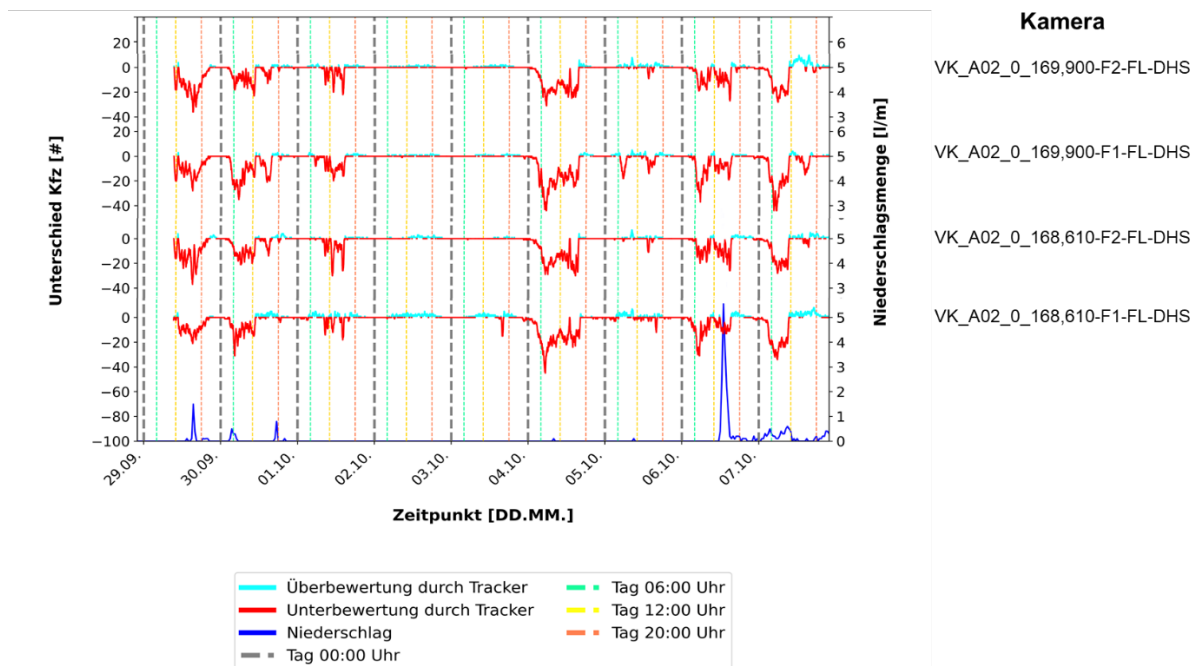


Abbildung 23: Vergleich Messergebnisabweichungen über den Testzeitverlauf.

Über alle Kameras hinweg zeigt sich ein sehr gutes Trackingergebnis für den Zeitraum 02.10.2021 bis 03.10.2021. In der statistischen Analyse konnte über alle vier Kameras ein signifikanter Zusammenhang des Trackingergebnisses mit den Faktoren Fahrzeugtyp, Tageszeit und Niederschlag (Regen) ermittelt werden.

Dies äußert sich darin, dass das Trackingergebnis tendenziell in der Nacht besser ist als am Tag, v.a. dieser Umstand illustriert das Hardwareressourcenproblem. Entgegen vorheriger Erwartungen konnte ein sehr gutes Trackingergebnis in den Nachtstunden aufgezeigt werden.

Ebenso hat Niederschlag einen signifikanten Einfluss auf das Trackingergebnis, sodass eine beobachtbare Verschlechterung der Erkennungsrate im Vergleich zum Tracking am Tag eintritt. Die Verschlechterung ist jedoch in einer ähnlichen Größenordnung wie das

Tageszählergebnis angesiedelt, dementsprechend ist der Einfluss auf das Gesamtergebnis gering.

Mit der statistischen Analyse konnte zusätzlich auch der Fahrzeugtyp als Einflussfaktor erkannt werden. In der weiteren Untersuchung wurde deutlich, dass der Fahrzeugtyp LKW für den VTRACS-Tracker schwieriger in der Erkennung ist im Vergleich zum PKW. Ein eindeutiger Grund für diesen Umstand kann mit dem fehlenden Videomaterial nicht eindeutig ermittelt werden.

Abbildung 24 zeigt die Auswerteergebnisse für Kamera „VK_A02_0_168,610-F1-FL-DHS“ über den Testzeitverlauf inklusive der gemittelten Abweichung des VTRACS-Trackers von dem zugeordneten ASFiNAG-Querschnitt. Hier wird unterschieden in einer Analyse für den Gesamtzeitraum 29.09.2021 bis 07.10.2021 und dem ausgewählten Zeitraum von 02.10.2021 bis 03.10.2021. Zu beachten ist hier zusätzlich, dass der korrespondierende Messquerschnitt ca. 7 km von der Kameraposition entfernt sind, dementsprechend sind die Daten im Schnitt etwas schlechter dargestellt.

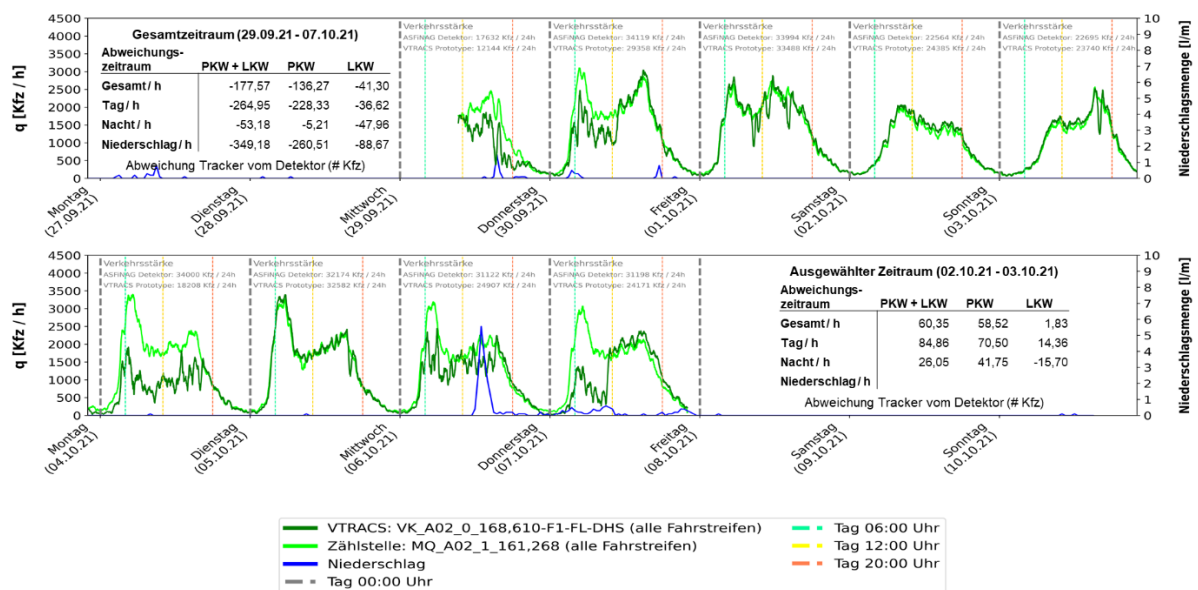


Abbildung 24: Beispielauswertung für Kamera „VK_A02_0_168,610-F1-FL-DHS“ über den Testzeitverlauf.

Die Auswertung aller weiteren Kameras kann dem Anhang entnommen werden.

3. ZUSAMMENFASSUNG

Was kann mit den implementierten Methoden erreicht werden?

Der VTRACS Software Prototyp und die darunter liegenden Detektionsmodule zeigen, dass Deep Learning Methoden für die Aufgabenstellung im VTRACS Projekt gut geeignet sind –

Die Messwerte des VTRACS Systems können mit den Messwerten anderer Sensoren durchaus verglichen werden. Dies gilt insbesondere auch für schlechte Lichtbedingungen und Regen.

3.1.1. Projektfazit

Das Projekt VTRACS hat aufgezeigt, welches Potential der Einsatz von state-of-the-art Deep Learning Methoden haben kann. Sichere Detektion auch unter widrigen Umständen ist möglich, **wobei jedoch auch zu vermerken ist, dass immer noch Verbesserungspotential bei fast allen Einzeldetektionen besteht – durch Erweiterung der Trainingsmenge und eine weitere Verfeinerung einiger Algorithmen.** Die Evaluierungsergebnisse zeigen sehr detailliert auf, an welchen Stellen im Algorithmus mit dem größten Einfluss auf die Klassifikationsgenauigkeit entwickelt werden sollte.

Bei künftigen ähnlich gelagerten Projekten sollten unter Berücksichtigung der gesetzlichen Vorgaben zum Datenschutz Vorkehrungen getroffen werden, realitätsnahe Trainingsdaten zu generieren. Weiters sollten verfügbare Hardware Ressourcen frühzeitig definiert werden.

3.1.2. Weiterer zukünftiger Entwicklungsbedarf

VTRACS hat aufgrund der beschränkten Projektlaufzeit nicht alle Aspekte der Detektion vollständig umsetzen können. **Wir sehen vor allem in folgenden Bereichen weiteren Entwicklungsbedarf:**

- Durchgehende Optimierung der Laufzeiten.
- Untersuchung vollständige geometrische Kamera Autokalibration, für
- Geschwindigkeitsmessung (Querschnittsbezogen).
- Reisezeitmessung (Abschnittsbezogen) > Wiedererkennung.
- Verkehrstechnische Untersuchung im Tunnel.
- Weiteres Tuning der Detektion.

4. ANHANG

Evaluierung Livebetrieb: Kamera 1

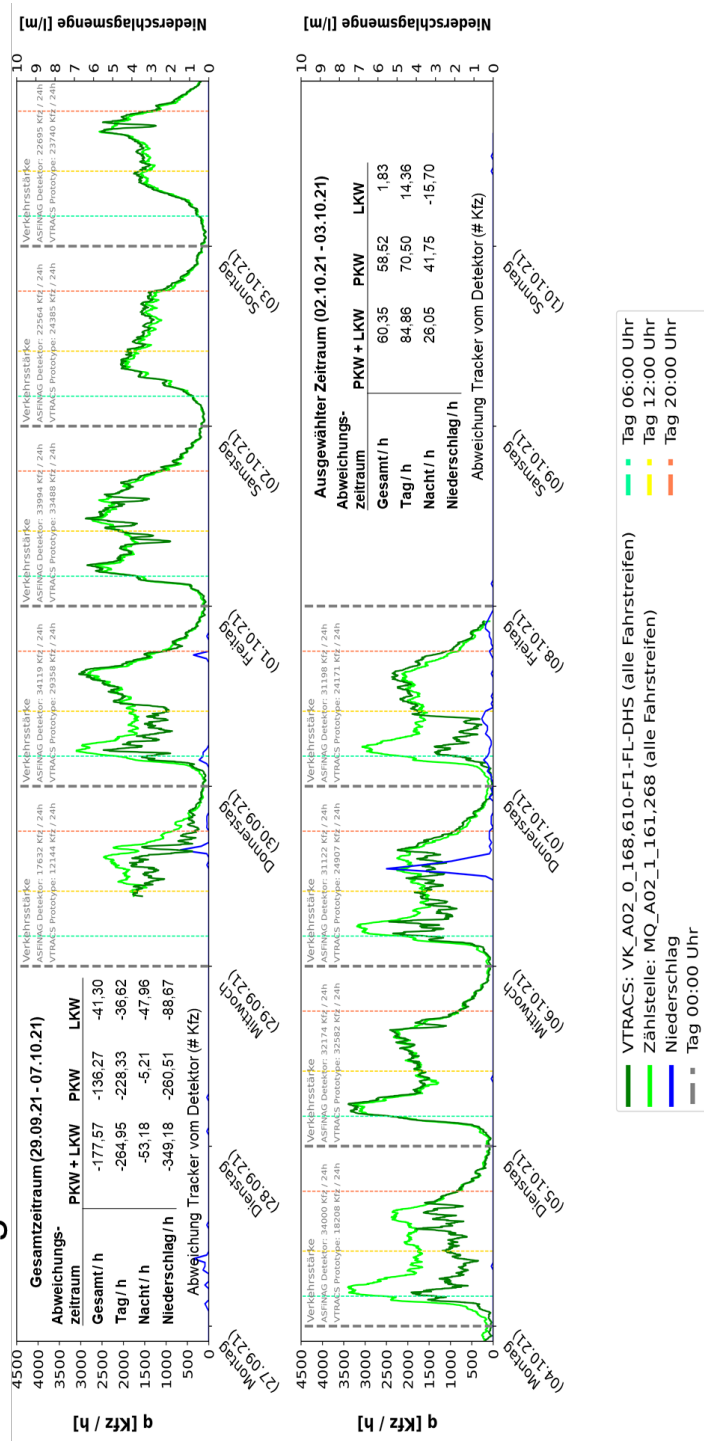


Abbildung 25: Auswertung für Kamera „VK_A02_0_168,610-F1-FL-DHS“ über den Testzeitverlauf.

Evaluierung Livebetrieb: Kamera 2

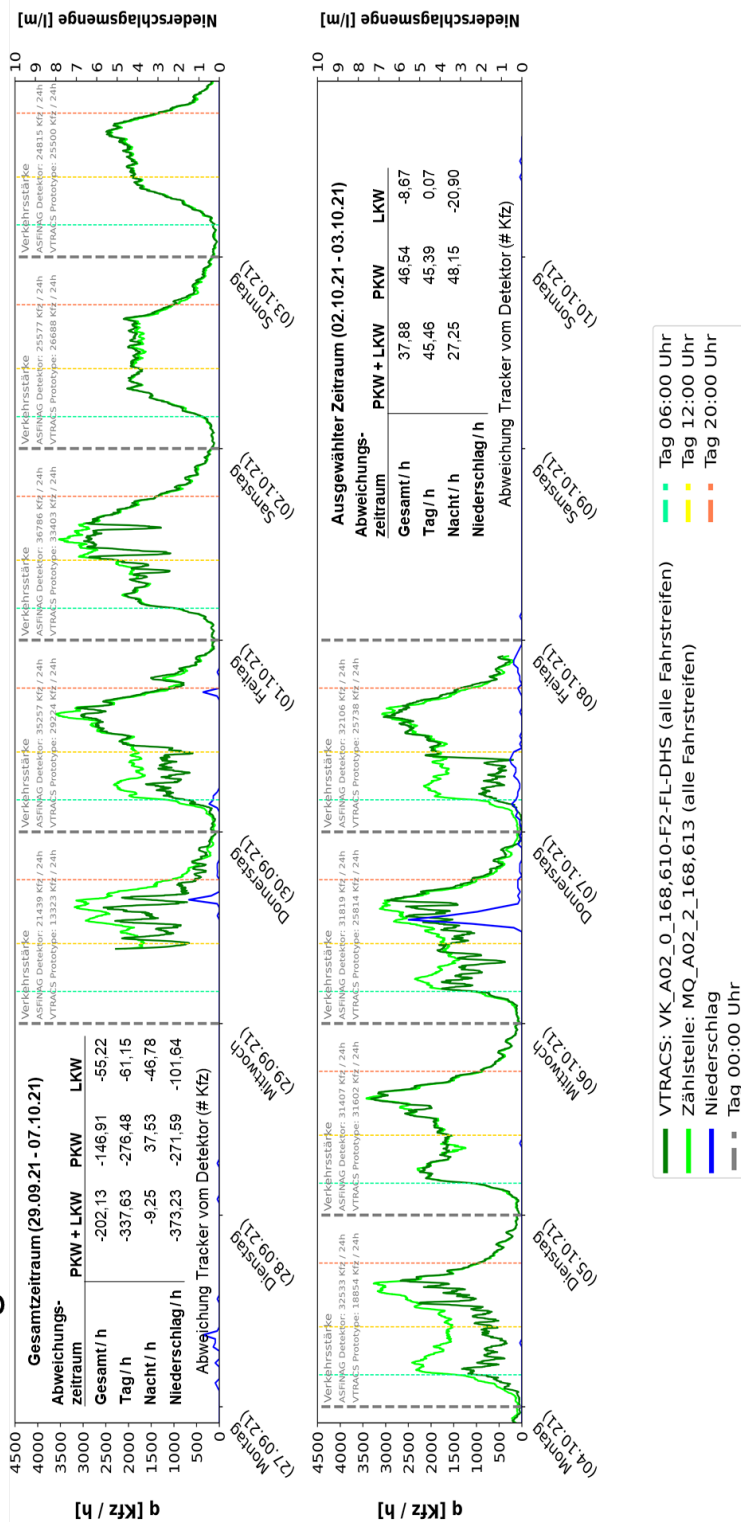


Abbildung 26: Auswertung für Kamera „VK_A02_0_168,610-F2-FL-DHS“ über den Testzeitverlauf.

Evaluierung Livebetrieb: Kamera 3

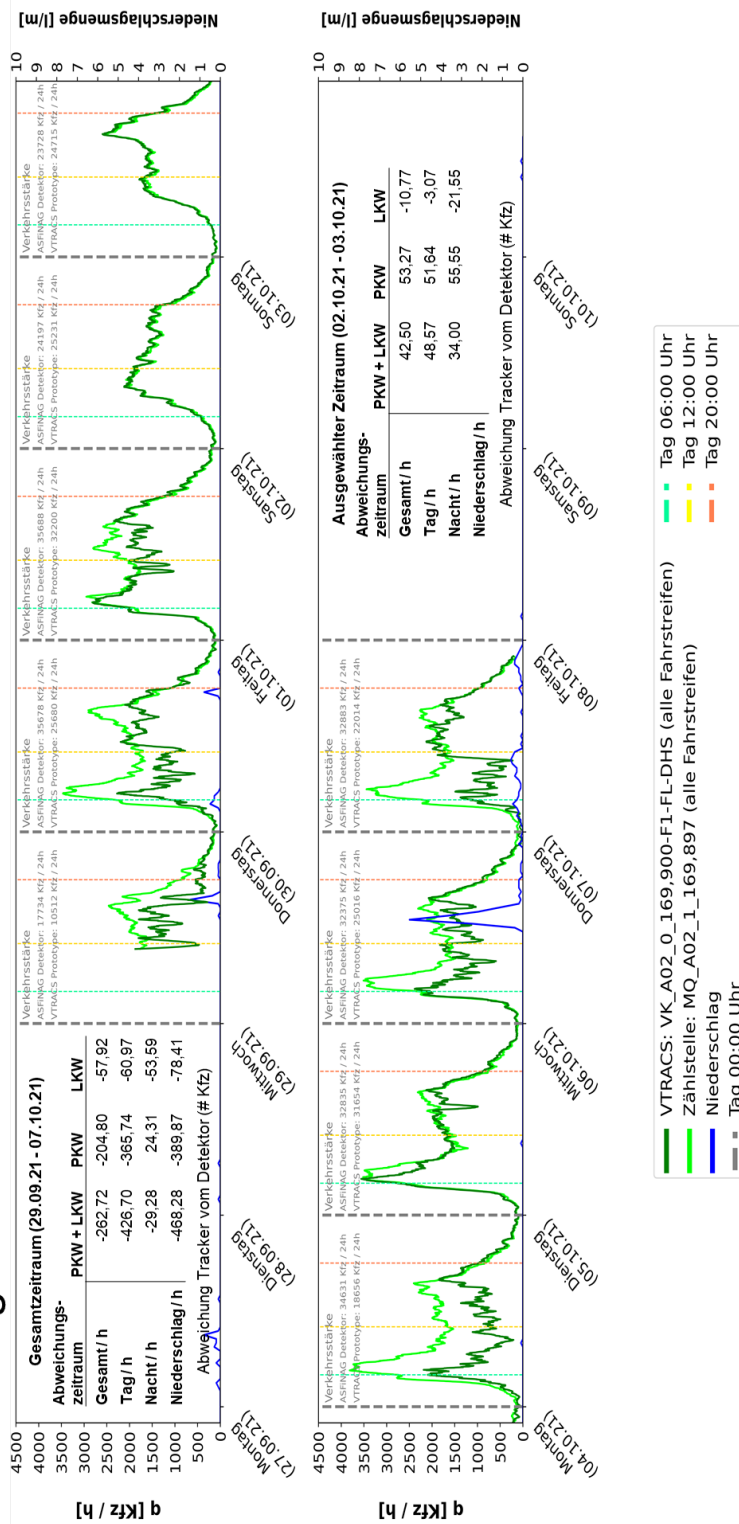


Abbildung 27: Auswertung für Kamera „VK_A02_0_169,900-F1-FL-DHS“ über den Testzeitverlauf.

Evaluierung Livebetrieb: Kamera 4

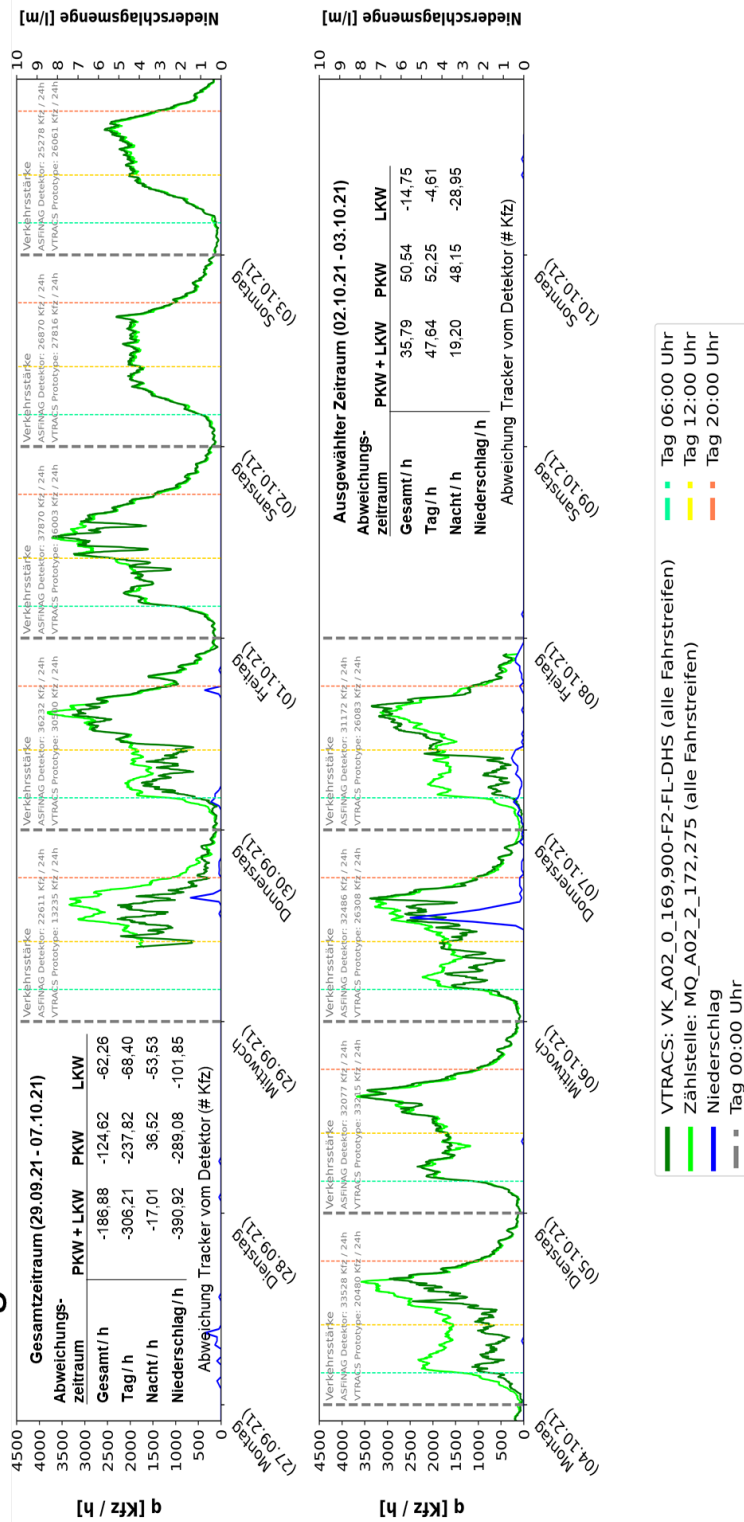


Abbildung 28: Auswertung für Kamera „VK_A02_0_169,900-F2-FL-DHS“ über den Testzeitverlauf.