Visuelle Mautvignetten Inspektion
VMI

Ein Projekt finanziert im Rahmen der
Verkehrsinfrastrukturforschung 2018
(**VIF2018**)

Dezember 2019

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

Visuelle Mautvignetten Inspektion VMI

Ein Projekt finanziert im Rahmen der

Verkehrsinfrastrukturforschung

(VIF2018)

Autoren:

Dipl.-Ing. Oliver SIDLA

Dipl.-Ing. Manuel LIENHART

MSc. Filippo GAROLLA

Univ.-Prof. Dr.-Ing. Martin FELLENDORF

Auftraggeber:

Bundesministerium für Verkehr, Innovation und Technologie

ÖBB-Infrastruktur AG

Autobahnen- und Schnellstraßen-Finanzierungs-Aktiengesellschaft

Auftragnehmer:

SLR Engineering GmbH

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
HW	...	Hardware
NN	...	Neuronales Netzwerk
ROC	...	Receiver Operator Characteristic curve definiert die Fähigkeit eines binären Klassifikators zur Klassentrennung
SW	...	Software

False Negative (FN)	...	Vignette vorhanden, keine Erkennung der Vignette
False Positive (FP)	...	Vignette nicht vorhanden, anderes Objekt als Vignette erkannt
True Negative (TN)	...	Vignette nicht vorhanden, keine fälschliche Erkennung
True Positive (TP)	...	Vignette vorhanden, korrekte Erkennung
Detektion/Detektor		wird in diesem Bericht im Zusammenhang mit der Merkmalserkennung in der Bildverarbeitung verwendet

INHALTSVERZEICHNIS

ABBILDUNGSVERZEICHNIS

TABELLENVERZEICHNIS

1. EINLEITUNG

Die ASFiNAG Anlagen zur Automatischen Vignetten Kontrolle (AVK-Anlagen) arbeiten vor Ort und ermitteln auf Basis hochauflösender Bildaufnahmen, ob der Verdacht auf Mautprellerei besteht. Die Bilder werden automatisch bewertet, müssen jedoch anschließend aufwendig manuell überprüft werden. Um diesen Aufwand in Zukunft deutlich reduzieren zu können, wurde ein Prototyp-System entwickelt, das den **State-of-the-art in Deep Learning** einsetzt, um die visuelle Vorfilterung der Verdachtsfälle möglichst optimal umzusetzen und so den Aufwand der Vignettenprüfung deutlich in Richtung Automatisierung zu bewegen.

Das **Projektkonsortium** bestehend aus **ISV und SLR** hat als Ergebnis des **Projektes VMI** SW Module und ein darauf aufbauendes Prototyp-System entwickelt, welches die Roh-Bilddaten der AVK-Anlagen analysieren und die **Prüfung der Vignette** (Vorhandensein, Position, Gültigkeit) mit hoher Genauigkeit durchführen kann. Das VMI System kann mit der integrierten ANPR Lösung *Carrida* von SLR Engineering auch das **Kfz-Kennzeichen** und den **Herkunftsstaat** eines Fahrzeuges mit hoher Sicherheit lesen.

Das VMI System besteht aus den Hauptkomponenten:

- Bilddatenmanagement + Datenschutz während der Entwicklungsphase, [1]
- Vignetten Detektion + Analyse
- Automatic License Plate Reading (ANPR)

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

Zuerst wird in Kapitel 1.1 die Problemstellung an Ort und Stelle vorgestellt. Das Kapitel 1.2 widmet sich im Überblick dem Stand der Forschung und in Kapitel 1.3 werden die Projektziele formuliert.

Abschnitt 2 beschreibt den technischen Lösungsansatz der Deep Learning Module im Detail, mit einer Zusammenfassung und einem Fazit in Abschnitt 3.

1.1. Problemstellung

Die ASFiNAG Vignetten Kontrollanlagen erzeugen eine sehr große Menge an Bilddaten welche größtenteils bereits vor Ort automatisiert gefiltert werden. Nur Verdachtsfälle werden in einem zweiten Schritt in der Zentrale in Wien manuell neu geprüft und ‚bestätigt'. Trotz der automatisierten Filterung beträgt die manuell zu prüfende Datenmenge ca. 6,5 Millionen Bildern jährlich, die von den derzeit 19 Prüfstationen nach Wien übertragen werden.

Die Qualität der Bilddaten ist variabel, aber aufgrund der Randbedingungen bei der Aufnahme (hohe Fahrzeuggeschwindigkeiten, hohe benötigte Auflösung, Umweltbedingungen) naturgemäß nicht immer optimal. **Die Anforderungen an eine automatische Bildanalyse sind daher sehr hoch.** An dieser Stelle kann Deep Learning seine Stärken ausspielen und helfen, eine akzeptable Detektionsrate auch bei insgesamt schwierigen Voraussetzungen zu erreichen.

Die folgende Abbildung 1 zeigt ein typisches Rohdatenbild mit guter (links) bzw. schlechter (rechts) Qualität. Idealerweise können beide Fälle noch mit großer Sicherheit automatisch verarbeitet und klassifiziert werden. In der Praxis wird das rechte Bild in den meisten Fällen als ‚nicht entscheidbar' klassifiziert und für die weitere manuelle Kontrolle abgespeichert.

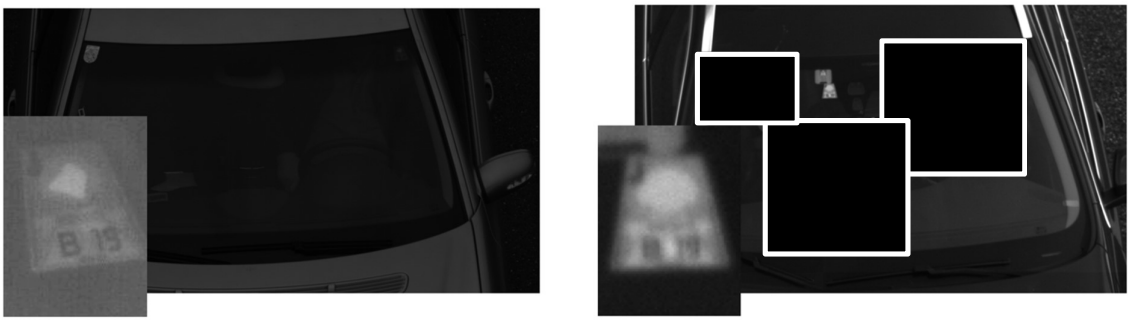


Abbildung 1: Ein typisches, relativ gutes Rohdatenbild aus einer ASFiNAG Kontrollstation im Vergleich zu einem schlechten Rohdatenbild (rechts). *Anmerkung: Bildelemente innerhalb der weiß markierten Kästen sind aufgrund der DGSVO geschwärzt*

Die manuelle Prüfung der ca. 6,5 Millionen Datensätze jährlich erfordert einen großen personellen und zeitlichen Aufwand. Die erfolgreiche Umsetzung einer automatischen Vignetten Inspektion hat daher sehr großes Potential, diesen Aufwand deutlich zu verringern.

Der steigende Anteil an digitalen Vignetten, die an das Kfz Kennzeichen gebunden sind, führt zu einem Bedeutungsgewinn der ANPR-Software die im ASFiNAG Prüfsystem eingesetzt wird. Die *Carrida* ANPR-Lösung (Produkt SLR) wird im Rahmen der Evaluierung im Detail

untersucht und mit der bestehenden ASFiNAG Software Lösung verglichen. Auch hier besteht eine deutliches Einsparungspotential, wenn der Anteil korrekt erkannter Kennzeichen durch eine bessere ANPR-Software erhöht werden kann.

1.2. 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 VMI für die Erkennung von Vignetten und Kennzeichen 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* veröffentlicht [2]. **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:

- 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.

Die Detektion von Objekten mit Deep Learning (DL) unterscheidet sich von der reinen Klassifikation, in dem im ersteren Problem **ein Objekt nicht nur erkannt, sondern zuvor auch detektiert** werden muss. **Im Projekt VMI wird versucht, dieses Detektions-Erkennungs-Problem für Mautvignetten zu lösen** - SLR hat in den letzten Jahren Erfahrung in der praktischen Anwendung aller im Projekt VMI benötigten Verfahren gesammelt.

Neben den bereits im Antrag genannten Methoden wurden während der Projektlaufzeit neueste Ansätze aus der Literatur getestet. Für die Detektion der Jahreszahlen und der J/T/M Markierung der Klebevignetten **konnte durch den Einsatz einer modifizierten Convolution Method**e nach [7] die Qualität der Detektion deutlich verbessert werden.

Die Eigenschaften von state-of-the-art Deep Learning Netzwerkstrukturen sind in der folgenden

Abbildung sehr gut zusammengefasst. Die AlexNet und MobileNet Strukturen sind klar die schnellsten Verfügbaren Detektoren, sie erreichen jedoch hinsichtlich der Detektions-Qualität nicht die Performance von großen (und daher langsamen) Netzwerk Architekturen wie *ResNet* oder *Inception*.

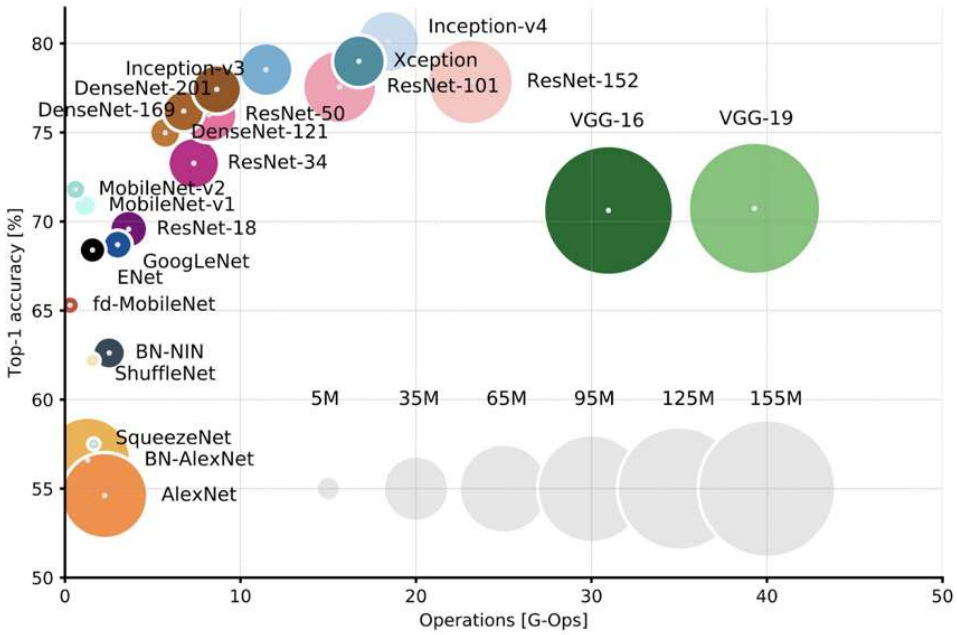


Abbildung 2: Vergleich der Detektionsqualität und Laufzeit für die ImageNet Datenbank. (Quelle: towardsdatascience.com)

Das Lesen der Kfz-Kennzeichen und die Staatenerkennung kann bereits sehr gut mit verfügbaren kommerziellen Lösungen durchgeführt werden. Es wäre zwar denkbar, Open-Source-Lösungen wie *OpenALPR* einzusetzen, jedoch ist deren generelle Leistungsfähigkeit im Moment nicht mit einer kommerziellen Software vergleichbar.

SLR entwickelt und vertreibt seit beinahe 10 Jahren die eigene ANPR Lösung *Carrida* welche weltweit einsetzbar ist, und sich bereits im Praxisbetrieb bewährt hat. Im Projekt VMI wird daher der Einsatz dieser Software untersucht, für den Prototyp im Rahmen des Projektes wurde dafür eine kostenfreie Lizenz zur Verfügung gestellt.

1.3. Projektziel

Das Ziel des Projektes VMI war die Entwicklung eines Prototyps, welcher die Eignung von Deep Learning Methoden für die automatische Vignetten Detektion und Analyse demonstriert und das vorhandene Potential von Deep Learning Detektion aufzeigt. Die Performance des VMI Prototyps wurde anhand einer möglichst großen Stichprobe aus ASFiNAG Rohdaten

evaluiert und untersucht. Folgende Aspekte wurden dabei im Rahmen des Projektes untersucht:

- **Performance-Kenngrößen:** geben Auskunft über die Bildverarbeitungszeit.
- **Analysezeit Vignette**: Zeitdauer für die Grob-Analyse und für die Fein-Analyse pro Datensatz.
- **Analysezeit Kfz-Kennzeichen**: Zeitdauer für Erkennung und Annotation des Kfz-Kennzeichens im Bild.
- **Vignetten-Erkennungsrate**: gibt den Anteil der richtigen Erkennung von vorhanden Vignetten auf den Bildern eines Datensatzes an (Zielvorgabe: 95%).
- **Koordinaten-Erkennungsrate**: gibt den Anteil der richtigen Angabe der Koordinaten der Vignetten-Eckpunkte auf den Bildern eines Datensatzes an (Zielvorgabe: 90%)
- **Gültigkeit-Erkennungsrate**: gibt den Anteil der richtigen Angabe der Gültigkeit von Vignetten auf den Bildern eines Datensatzes an (Zielvorgabe: 90%)
- **Kennzeichen-Erkennungsrate**: gibt den Anteil der richtigen Erkennung der Kfz-Kennzeichen auf den Bildern eines Datensatzes an (Zielvorgabe: 95%)
- **Staaten-Erkennungsrate**: gibt den Anteil der richtigen Erkennung des Zulassungsstaates der erkannten Kfz-Kennzeichen auf den Bildern eines Datensatzes an (Zielvorgabe: 90%)

2. METHODIK

Im Rahmen des Projektes VMI wurde die Aufgabenstellung als modulares System gelöst. Die Einzelmodule wurden in Python3 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 VMI **auf 2x NVIDIA 2080 Ti GPU Grafikkarten**

durchgeführt, die Trainingszeiten für diese HW Konfiguration betrugen, je nach Detektor, wenige Minuten bis einige Stunden.

Der Kern vom VMI Demonstrator wurde als Python3 Anwendung entwickelt, wobei das Toolkit QML zur Beschreibung und Implementierung der Benutzeroberfläche eingesetzt wurde.

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

Als **ANPR Lösung** wird das **verfügbare kommerzielle Produkt *Carrida* von SLR Engineering** verwendet, sie erfüllt alle nötigen Anforderungen wie Europaweite Lesefähigkeit, Länderkennung und Lesegeschwindigkeit.

Carrida wurde mit der verfügbaren Windows Demonstrationssoftware eingesetzt, um die ASFiNAG Bilddaten zu lesen. Groundtruth Daten wurden durch manuelle Annotation bestimmt und als Referenzwerte für die Ermittlung der Lesegenauigkeit verwendet.

Zusammenfassend wurden folgende Open-Source Komponenten verwendet:

- **Python3** für die generelle SW Entwicklung.
- **QML** für GUI Programmierung des Prototyps.
- **OpenCV** für die Basis Machine-Vision Module (Bild IO, Vorverarbeitung von Bildern, etc.).
- **Tensorflow und Keras** für die Deep Learning Objekt Detektion.
- **NVIDIA Cuda** für die Beschleunigung der Rechenoperationen auf den GPUs.
- **Carrida *für die ANPR***

Die Projektstruktur mit den einzelnen Modulen ist in der nachfolgenden Abbildung 2 dargestellt.

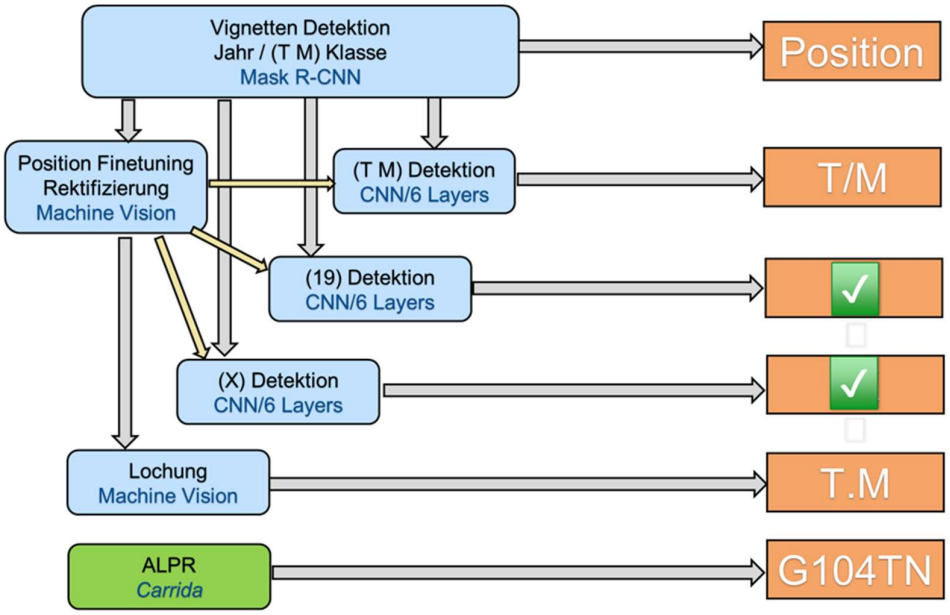


Abbildung 2: Software Struktur wie sie für das VMI Projekt implementiert wurde.

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. Dabei wird ein Mix aus den Standorten der AVK-Anlagen, der Aufnahmezeit der Bilder und verschiedenen Fahrzeuggrößen ermittelt. Durch eine Variation in der Aufnahmezeit werden sowohl unterschiedliche Lichtverhältnisse, als auch unterschiedliche Verkehrsbedingungen berücksichtigt (Morgenverkehrsspitze, Mittagszeit, Ferienverkehr etc., dies kann durch einen Abgleich des AVK-Standortes mit Verkehrsdaten einer naheliegenden Zählstelle realisiert werden).

Die Trainingsdaten wurden manuell ausgewertet, es wurde das Vorhandensein und die Gültigkeit einer geklebten und digitalen Vignette pro Verdachtsfall geprüft. Damit werden Ergebnisse der Mautüberprüfung für die Trainingsdaten manuell generiert, die wiederum bei der Entwicklung des Analyseverfahrens (Vignette und Kennzeichen) für den Prototyp hilfreich

waren (z.B. Adaptierung des Analyseverfahrens bei Ergebnisabweichung zwischen manuell und automatisiert) und mit den Ergebnissen der automatischen Mautüberprüfung für die Trainingsdaten verglichen werden konnten.

2.2. Detektionsmodule

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.

2.3. Versuche mit dem HOG Detektor von SLR Engineering

Bis zur Verwendung von Deep Learning Detektoren wurde bei SLR Engineering eine selbst implementierte und stark optimierte Version eines HOG Detektors von Dalal et al. [6] verwendet. Der Vorteil der HOG Methode besteht in der relativ einfachen Trainierbarkeit und der sehr schnellen Detektionszeit im Verhältnis zu DL Pipelines, insbesondere wird für HOG keine GPU benötigt.

Vorversuche mit HOG für Training und Detektion der Vignetten und der Jahreszahl haben jedoch gezeigt, dass die **oft vorhandene Unschärfe in den Bilddaten die Detektion oft sehr stark einschränkt, die Detektionsraten wären nach einer Abschätzung im Schnitt unter 50% gelegen**. HOG ist ein Feature Detektor, welcher auf Kanten basiert, sobald diese durch Unschärfe oder Rauschen geschwächt und gestört werden, scheitert die Feature Berechnung und damit auch die anschließende Klassifikation. Die geschätzte

Der HOG Detektor musste aus den obengenannten Gründen daher verworfen werden, praktisch für alle Detektionsmodule wurden in Folge *Convolutional Neural Networks* als Detektoren gewählt.

2.4. Merkmalserkennung durch Convolutional Neural Networks

In der automatischen Bildverarbeitung (computer vision) werden unterschiedliche Techniken eingesetzt, um Merkmale von Bildelementen unterscheiden zu können. Eine weit verbreitete Technik sind CNNs (Convolutional Neural Networks), die auf geschichteten (gefalteten) neuronalen Netzes basieren. Das Hintereinanderschalten von vielen CNNs erzeugt die sogenannten ‚deep networks', welche durch die inhärente Komplexität der Struktur auf komplizierte Detektionsprobleme trainierbar sind.

2.4.1. Deep Learning für die Detektoren

Für die Detektion der Visuellen Elemente ‚Jahreszahl', ‚T', ‚M und, ‚X' auf den Klebevignetten, wurde jeweils ein individuelles CNN entworfen und trainiert. Diese Netzwerke werden in den folgenden Abschnitten beschrieben.

Die Samples für das Training aller Detektoren wurden aus den Rohdatenbildern manuell extrahiert und in Ordnern gespeichert, diese sind in der Distribution des VMI Prototyps enthalten.

Die Sample Formate und Größen wurden meistens abweichend zur Standard Implementierung der Netzwerke gewählt, um für VMI bestmögliche Detektions-Qualität zu erreichen. Diese Informationen sind in der Detailbeschreibung der Netzwerke angeführt.

2.4.2. Vorversuche Vignetten Detektion

Zu Projektbeginn wurde mit einer vorab generierten Menge an Vignetten Samples eine Serie an Versuchen gestartet, um die beste mögliche Architektur für einen Detektor zu finden. Gleich zu Beginn musste der HOG Detektor ausgeschlossen werden (siehe oben), weitere Versuche wurden daher nur mehr mit Netzwerkmodellen durchgeführt.

Ursprünglich wurde nur an einen Bounding Box Detektor gedacht, mit dem das umschreibende Rechteck der Vignette gefunden werden sollte. Die Erkennung der exakten Begrenzung der Vignette würde dann mit zusätzlichen Methoden durchgeführt.

Als allererstes Netzwerk Modell wurde *ein SSD mobilenet V1 fpn* Netzwerk Modell konfiguriert und trainiert. Vorteil dieses Modelles ist die sehr gute Laufzeit-Performance bei guter Detektion.

Als weiterer aussichtsreicher Kandidat wurde ein Faster R-CNN Netzwerkmodell trainiert und auf den Daten getestet. Die Detektionsrate damit war vergleichbar mit dem SSD Netzwerk und dem letztendlich eingesetzten Mask R-CNN Netzwerk.

Im Laufe weiterer Versuche musste jedoch festgestellt werden, dass die **Detektion des Vignetten Randes aus der Bounding Box sehr fehleranfällig** ist – oft wird die dazu nötige Kantendetektion durch Rauschen und Überlagerungen von Reflexionen gestört und ist damit unzuverlässig. Aufgrund dieser Erkenntnis wurden erste Versuche mit einem Mask R-CNN Netzwerk durchgeführt, die letztendlich erfolgreich waren. Die SSD Netzwerk Architektur ist nicht kompatibel mit dem Mask R-CNN Netzwerk Modell und musste daher verworfen werden. Dsdurch verliert der VMI Detektor etwas an Laufzeit Performance, jedoch kaum etwas an Detektionsqualität.

Die automatisch generierte Maske liefert schon eine sehr gut brauchbare Annäherung an die finale Vignetten Kontur und muss nur noch in kleinem Ausmaß verfeinert werden. Damit kann die Kontur- bzw. Eckpunkt Detektion auf der Vignette elegant gelöst werden. Die Details dazu sind in den folgenden Abschnitten beschrieben.

2.4.3. Vignetten Detektion und Typ-Bestimmung

Für die erste Stufe der Detektion werden Vignetten mit einem komplexen Mask R-CNN detektiert. Dieser Typ von Netzwerk kann eine trainierte Objektklasse prinzipiell Pixel-genau lokalisieren. In der Praxis ist die Detektion nicht immer auf den exakten Rand der Vignetten beschränkt, die berechnete Maske ist jedoch sehr **gut als Ausgangspunkt für die Feinanpassung der Vignetten Ränder** geeignet.

Das R-CNN Netzwerk kann neben der Objektlokalisation auch eine Unterscheidung zwischen den J- bzw. T/M Typen durchführen. Mit dieser Information kann der VMI Prototyp dann die nächsten Verarbeitungsschritte bzw. Entscheidungsprozesse weiterführen.

Während der Evaluierung der Vignetten Detektion wurde ein Problem beim Detektieren erkannt, wenn eine Gruppe von nahe beieinander liegenden Vignetten im Bild vorhanden ist. Anscheinend werden durch die Methode der Non-Maxima Suppression im Detektor Netzwerk gültige Maxima die nahe beieinander liegen unterdrückt, so dass bspw. die **mittlere von drei benachbarten Vignetten vom Netzwerk nicht korrekt erfasst wird**. Die Behebung dieses Problems wäre im Rahmen dieses VMI Projekts zu aufwendig und langwierig gewesen, **es musste daher auf Code Änderungen in diesem betroffenem Sub-Modul verzichtet werden.** Eine weitere Diskussion erfolgt im Abschnitt zur Evaluierung.

Die Netzwerk Struktur für die Vignetten Detektion ist wie folgt aufgebaut:

- Faster R-CNN Struktur, 4 CNN Layers für Mask Generierung
- Masken Auflösung 33x33 px (15x15 px wurde in der Literatur als Originalwert vorgeschlagen)
- Trainiert mit 28.147 Sample Bildern
- Training von J-, M- und T-Vignetten als ein Datensatz
- Trainingszeit ca. 13h auf 2x Nvidia 2080 Ti GPU

Beispiele für die Detektion von Vignetten sind in der folgenden Abbildung 3 unten angeführt. Die detektierte Vignetten Maske ist jeweils grün über das Original-Bild gelegt.

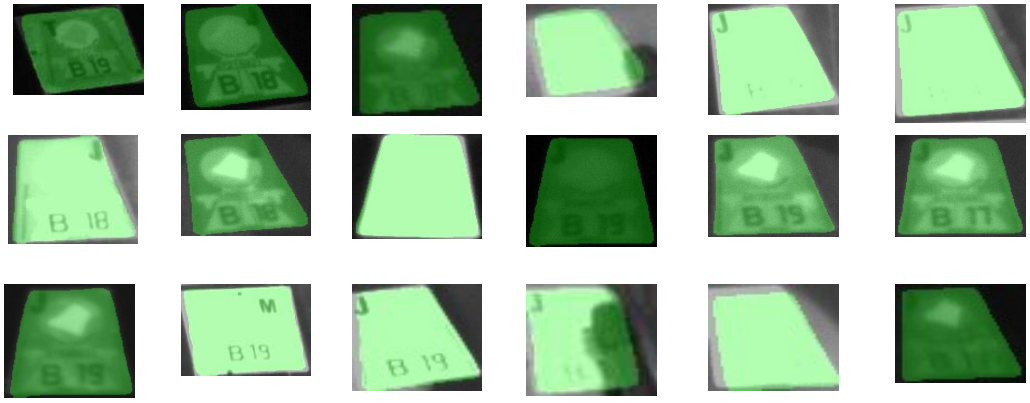


Abbildung 3: Beispiele für detektierte Vignetten. Die grüne Fläche stellt die tatsächlich detektierten Vignetten Flächen dar.

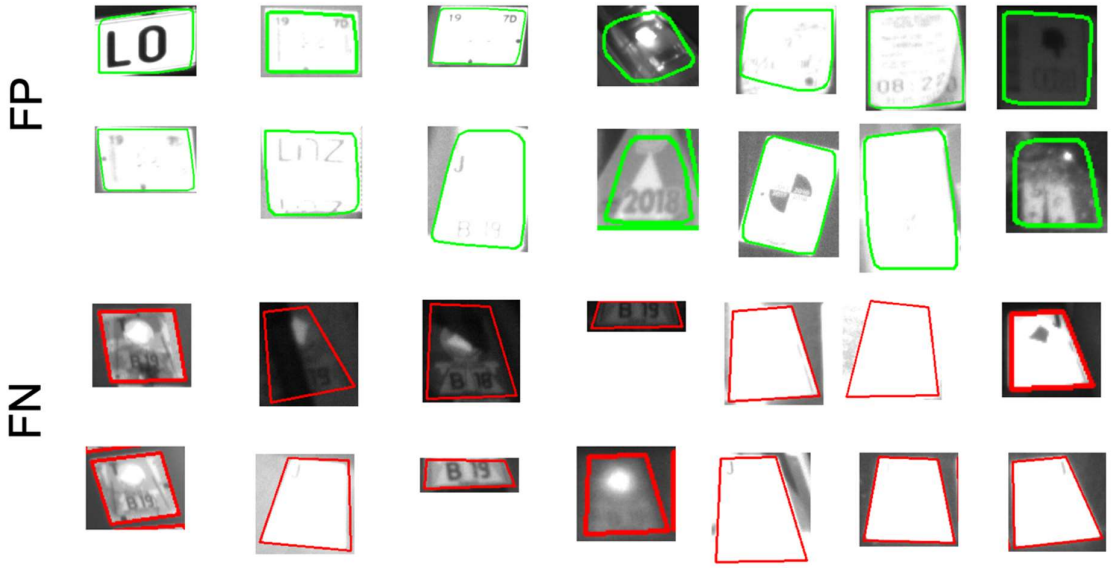


Abbildung 4: Beispiele für falsch detektierte Vignetten. False Posities (FP, oben) entstehen meistens an Vignetten-ähnlichen Strukturen, False Negatives (FN, unten) größtenteils an fehlbelichteten oder sehr stark beeinträchtigten Vignetten Abbildungen.

2.4.4. Vignetten Feinpositionierung

Nachdem eine Vignette im Bild grundsätzlich detektiert wurde, wird deren Geometrie entzerrt und normalisiert. Damit können alle folgenden Prozesse auf definierte Koordinaten innerhalb der Vignette abgebildet werden.

Die Entzerrung der Vignette wird anhand der detektierten Maske des ersten CNNs durchgeführt. Die Kanten der Vignette werden durch Antastung in den möglichen Randbereichen der Vignette gesucht und für die Berechnung von Ausgleichsgeraden verwendet. Diese so detektierten Randsegmente der Vignette definieren das sichtbare Vignetten-Polygon, welches durch einfache geometrische Berechnung in ein normiertes Rechteck umgewandelt werden kann. **Die Begrenzungen der Ausgleichsgeraden werden bestimmt durch die Schnittpunkte benachbarter Linien, welche die Eckpunkte der Vignetten definieren.**

Der Vorgang der Kanten-Antastung und Entzerrung ist in der folgenden Abbildung 5 dargestellt.

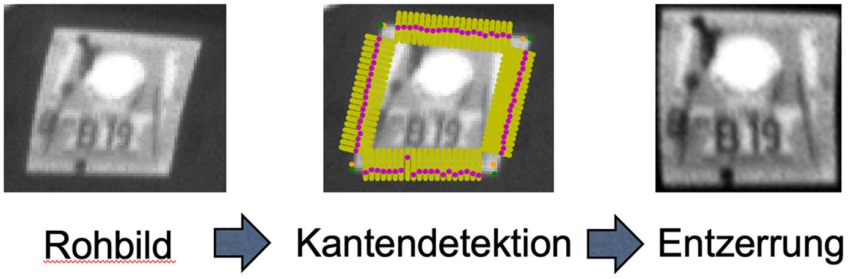


Abbildung 5: Die Detektion und Ausrichtung des tatsächlichen Vignetten Randes erfolgt mittels Kanten Antastung und anschließender geometrischer Entzerrung.

2.4.5. Bestimmung der Vignetten-Jahreszahl

Die Detektion der Jahreszahl erfolgt in einem Ausschnitt im normalisierten Vignetten-Bild, in welchem die Jahreszahl prinzipiell vorhanden sein sollte.

Die Jahreszahl auf den Vignetten wurde mit einem Netzwerk mit den folgenden Eigenschaften trainiert:

- 4x 10 Layer CNN für jede Klasse separat trainiert
- Trainiert mit 2.901 Samples
- Training der Klassen ‚17', ‚18', ‚19' und ‚ungültig'
- Trainingszeit weniger als eine Minute

Für die Klassifikation werden alle 4 Netzwerke auf den Bildausschnitt angewandt, das **Netzwerk mit der besten finalen Konfidenz bestimmt die detektierte Klasse**.

Für das Trainieren wurde zusätzlich zu den *standard convolutional Layern* noch ein ‚Max BlurPool' Layer anstatt des ansonsten in der Literatur verwendeten ‚MaxPool' Layers eingesetzt, siehe [7]. Damit konnte die Genauigkeit der Klassifikation deutlich verbessert werden, weil dieser Ansatz Aliasing Artefakte in den Zwischenschichten des CNN effektiv vermeiden kann. Die Detektion wird dadurch invariant gegenüber kleiner Translationen der Jahreszahl, wie sie durch Ungenauigkeiten beim Entzerren etc. entstehen können.

2.4.6. Klassifikation der T/M Markierungen

Die Klassifikation erfolgt im normalisierten Vignetten Bild, nur in den beiden Ausschnitten links und rechts auf der Vignette, in welchen die Zeichen prinzipiell vorhanden sein sollten.

Um nur eine Detektion durchführen zu müssen, wird aus den beiden Ausschnitten links und rechts der Vignette ein künstliches 2-Kanal Bild generiert, welches anschließend klassifiziert wird.

Die T/M Markierungen auf den Vignetten wurde mit einem Netzwerk mit den folgenden Eigenschaften trainiert:

- 8 Layer CNN
- Trainiert mit 2.011 Samples
- Trainingszeit weniger als eine Minute

2.4.7. Detektion der X Markierung

Für die Detektion der X Markierung wurde die gesamte normalisierte Vignettenfläche als Bildregion für das Training verwendet. Problematisch war beim Training die etwas geringe verfügbare Samplemenge.

Die X Markierung auf den Vignetten wurde mit einem Netzwerk folgender Eigenschaften trainiert:

- 8 Layer CNN
- Trainiert mit 2.108 Samples
- Trainingszeit weniger als eine Minute

2.4.8. Analyse der Stanzlöcher

Nachdem eine Vignette als T/M Typ erkannt wurde, besteht der nachfolgende Schritt in der Erfassung der Position der Stanzlöcher, um das Gültigkeitsdatum der Vignette zu verifizieren. Diese Detektion findet im bereits entzerrten Vignetten-Bild statt, aufgrund des relativ kleinen Randbereichs der Vignette ist eine **gute Qualität der Entzerrung die Voraussetzung für eine sichere Detektion der Stanzlöcher**.

Für die automatische Erkennung der Stanzlöcher wird sehr ähnlich zur manuellen visuellen Kontrolle vorgegangen. Die Seitenlänge der Vignette wird in gleich lange Rechtecke eingeteilt, in welchen nach Stanzlöchern gesucht wird – der Index des detektierten Rechtecks definiert anschließend das gestanzte Datum.

Die Detektion des Stanzloches wurde ursprünglich als Erkennung eines ‚relativ' dunklen Bereiches im Vergleich zu benachbarten Regionen implementiert. Dies erwies sich aber nicht als robust genug, weil durch Bildstörungen, Abschattungen und Überklebungen zu viele adverse Einflüsse am Vignettenrand erzeugt werden. Letztendlich hat sich ein matched Filter mit der erwarteten Struktur eines Stanzloches (dunkler Kreis auf hellem Hintergrund) als beste Lösung erwiesen, siehe Abbildung 6 unten.

Als begrenzende Faktoren für die Detektion der Stanzlöcher können Rauschen, Reflexionen auf der Vignette bzw. Windschutzscheibe und Belichtungsfehler die Detektion eines Stanzloches verhindern. Um die Sicherheit der Messung zu erhöhen wird, w**enn die Konfidenz des Matchings sehr gering ausfällt, wird ein Stanzloch als ungültig markiert.**

Bemerkung: Die Template Matching Funktion liefert auf den Vignetten Bildern nur eine sehr unzuverlässige Konfidenz. Ursache dafür ist vermutlich die sehr unterschiedliche Bildqualität und Ausführung der Stanzlöcher. Es kann daher auf Basis der Konfidenz nur eine sehr grobe Unterscheidung gültig/ungültig getroffen werden.

Die folgende Abbildung 6 zeigt rechts ein Beispiel, in welchem die Stanzloch Detektion aufgrund der schlechten Stanzung nicht korrekt gefunden wurde. Das Ergebnis würde vom VMI System einer manuellen Kontrolle zugeführt.

Abbildung 6: Das Prinzip der Stanzloch Detektion (links), und ein Beispiel für einen Fehlerfall (rechts). Ein Template für das Abbild eines Stanzloches wird in den Randbereichen der entzerrten Vignette gesucht. Die beiden Positionen mit größter Matching Konfidenz werden für die Bestimmung des Gültigkeitsdatums herangezogen.

2.4.9. Algorithmus Vignetten Gültigkeits-Bestimmung

Mit den Einzelergebnissen der Detektionsmodule und den verfügbaren Konfidenzen der Detektion (= Abschätzung über die Sicherheit einer Detektion) kann ein Algorithmus erstellt werden, welcher die Entscheidung über die Gültigkeit einer Vignette trifft.

Der Ablauf der Vignetten-Klassifikation auf Basis der einzelnen Detektionsmodule kann in Pseudocode folgendermaßen abgebildet werden. Nicht berücksichtigt wird im folgenden Algorithmus die Abfrage nach dem Kennzeichen und der damit verbundenen elektronischen Vignette, weil dies im VMI Prototypaufgrund des fehlenden Datenbankzugriffs nicht direkt implementiert werden konnte.

Verwendete Abkürzungen bzw. Parameter
TS ... Time Stamp of image, known
VD … Confidence of vignette detection network
TM … Confidence of TM detection network
YY … Confidence of year (17, 18, 19) detection network
X … Confidence of X detection network
NX … Confidence of X not detected (X + NX = 1.0)
DATE … Date as computed from punch holes in vignette image (day, month markings)

T_{VD} … *Threshold for confidence of vignette detection network*
T_{TM} … *Threshold for confidence of TM detection network*
T_{YY} … *Threshold for confidence of year (17, 18, 19) detection network*

T_X … *Threshold for confidence of X detection network*
T_{NX} … *Threshold for confidence of X not detected*
T_{TM} … *Threshold for confidence of TM detection network*

No vignette found -> check LP string and database.
If ('X' found)
If (X >= T_X)
➔ *invalid. done.*
else
if (X < T_X)
➔ *uncertain. done.*
else ('X' not found)
if (NX < T_{NX})
➔ *uncertain. done.*

If (YY < T_{YY})
➔ *uncertain. done.*

If (YY == '19')
➔ *valid. done.*
else
➔ *invalid. done.*

If (TM < T_{TM})
➔ *uncertain. done.*

If (LM or LT not found)
➔ *Uncertain. done.*

If (DATE is in the future relative to image TS)
➔ *uncertain. done.*

If (DATE is valid relative to TS)
➔ *valid. done.*

If (DATE is invalid relative to TS)
➔ *invalid. done.*

2.5. Lesen von Kennzeichen und Staatenerkennung

Für das Lesen der Kennzeichen wird die von SLR Engineering entwickelte Lösung *Carrida* verwendet. Im Rahmen des VMI Projektes wurde *Carrida* verifiziert und in einigen Aspekten verbessert:

- Training für spezielle Effekte der ASFiNAG Rohdaten wurde durchgeführt (Anpassung an einen oft auftretenden Typ von Bildstörungen)
- Anpassung der Syntaxauswertung der AT Kennzeichen
- Anpassung der Syntaxauswertung der DE Kennzeichen

Die folgende Abbildung 7 zeigt einen Screenshot des *Carrida* Evaluierungsprogramms. Der Benutzer kann als Bildquelle beliebige Folder, Videodateien oder Live Bilder angeschlossener Kameras wählen.

Für die Evaluierung im VMI wurden vorbereitete Folder mit den Referenzdatensätzen erzeugt und verarbeitet. Die Ergebnisse wurden als CSV Dateien gespeichert und anschließend mit weiteren Scripts in der Programmiersprache R statistisch ausgewertet.

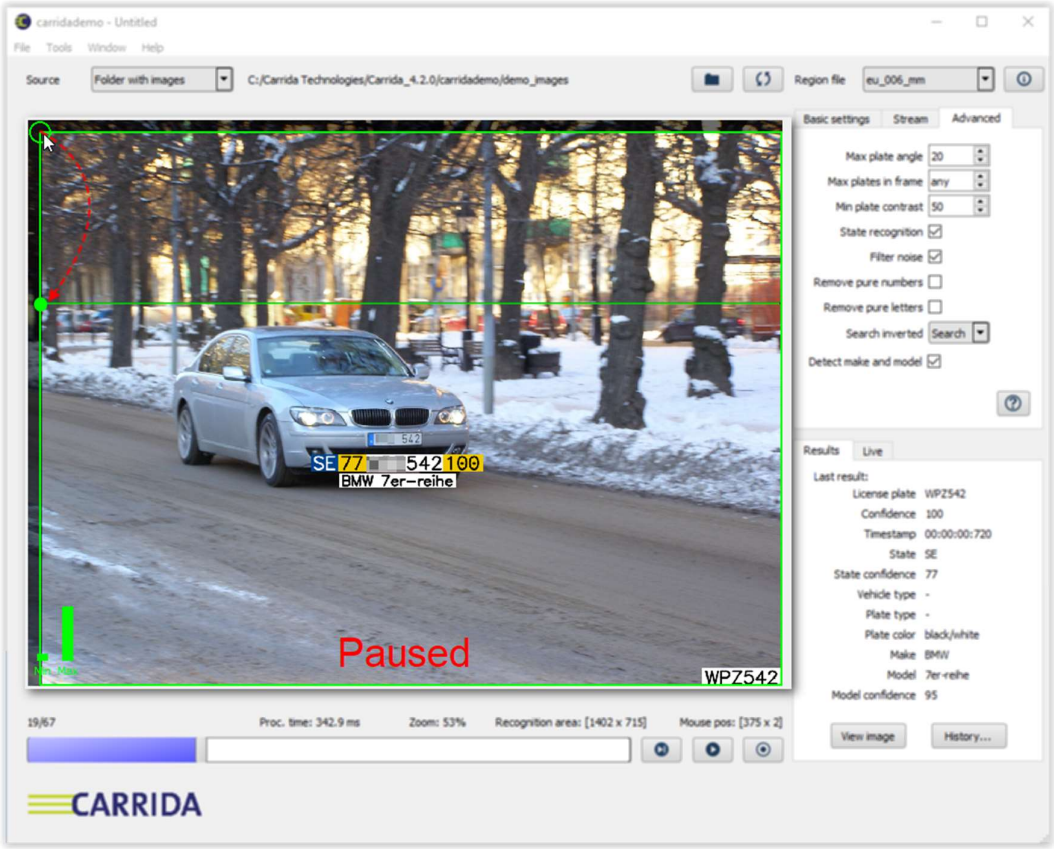


Abbildung 7: Screenshot des *Carrida* Evaluierungsprogrammes.

2.6. Bedienung des VMI Prototyps

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

Die Software erlaubt es, mittels einer einfachen graphischen Benutzeroberfläche, Folder von Bildern zu berechnen und alle relevanten Detektionsergebnisse anzuzeigen. Die folgende Abbildung 8 zeigt die Benutzeroberfläche des Programmes.

Die Implementierung des VMI Programmes wurde in Python3 und QML durchgeführt.

Mit den Cursor Tasten ‚links', ‚Rechts' wird ein neues Bild aus dem Datenfolder geladen. Für jede detektierte Vignette wird rechts oben eine Ausschnittsvergrößerung angezeigt. Darunter werden die Detailergebnisse der Einzelmodule dargestellt. Der grüne Rand um die Vignette zeigt die berechneten Kanten des Vignetten Polygons, die graue Umrandung den ursprünglich detektierten Rand aus der R-CNN Maske. Die blauen Kreuze markieren die detektierten Stanzlöcher, wenn der Vignetten Typ als M oder T erkannt wurde.

Im Falle von Mehrfach Detektionen von Vignetten kann durch Klick mit der Maustaste zwischen den Vignetten umgeschaltet werden.

Abbildung 8: Screenshot des VMI Prototyps. Alle detektierten Vignetten werden angezeigt, dazu die Detektionsergebnisse und Konfidenzen der einzelnen Erkennungs-Module.

2.7. Evaluierung

Für das Training und die Evaluierung des Prototyps wurden von der ASFiNAG während der Projektlaufzeit drei Datensätze (Umfänge 28.422, 31.756 und 19.297 Einzelfahrzeugbilder) zur Verfügung gestellt. Diese Datensätze wurden zu unterschiedlichen Entwicklungszeitpunkten für verschiedene Fragestellungen eingesetzt. Die ersten beiden Datensätze wurden für die Entwicklung des Prototypen eingesetzt. Beide Datensätze sind ähnlich aufgebaut und enthalten „Bestätigte" Einzelfahrzeugbilder (wurden durch das ASFiNAG Personal im Maut Enforcement Center als manuell als Verdachtsfälle gekennzeichnet) und „Rohdaten" (ungeprüfte Einzelfahrzeugbilder). Der dritte Testdatensatz enthält eine zufällige Menge ungeprüfter Einzelfahrzeugbilder eines typischen Arbeitstages im ASFiNAG Maut Enforcement Center. Dieser Testdatensatz wurde für die abschließende Wirkungsanalyse eingesetzt.

2.7.1. Evaluierung der Kennzeichenerfassung (ANPR)

Bevor der Testdatensatz 1 für die Entwicklung der Detektionsmodule eingesetzt wurde, fand er auch Anwendung, um die Wirkung der kommerziell erhältlichen Software CARRIDA (entwickelt und vertrieben durch SLR), bereits im ersten Test zu evaluieren. Nach einer geringfügigen Weiterentwicklung der CARRIDA konnten die von der ASFiNAG geforderten Kennwerte hinsichtlich Erkennungsrate der Zeichenkette auf den Kennzeichen (mindestens 95%), sowie des Zulassungsstaates (mindestens 90%) bei weitem übertroffen werden (Tabelle 1)

Tabelle 1: Ergebnis ANPR für Datensatz 1 und den Vergleich der ASFiNAG-Lösung mit CARRIDA von SLR (n = 19.402 Einzelfahrzeugbilder)

Erkennung von	ASFiNAG-Lösung lt. Datensatz	CARRIDA (SLR)
Nation	97,9	98,6
Kennzeichen	94,6	98,6
beide	94,3	97,5

2.7.2. Testdatensatz 1 für Entwicklung des CNN

Der Testdatensatz 1 (28.422 Einzelfahrzeugbilder, vgl. Tabelle 2) wurde vollständig annotiert; d.h. neben der Bestätigung durch die ASFiNAG MitarbeiterInnen wurde eine weitere manuelle Kennzeichnung vorgenommen. Aus dem Testdatensatz wurde eine zufällige Stichprobe von 1.000 Bildern auf Vollständigkeit und Richtigkeit von Unbeteiligten geprüft. Im Rahmen dieser Überprüfung wurde festgestellt, dass bis auf wenige vernachlässigbare Fehler die manuelle

Kennzeichnung der Bilder korrekt, nachvollziehbar und wiederholbar erfolgte. Konkret waren folgende Punkte auffällig:

- Kennzeichenerkennung funktioniert schlechter bei direktem Lichteinfall, einzelne Buchstaben/Ziffern können fehlen
- Position des Rahmens um die Vignetten ist oft ungenau.
- Jahresvignetten sind im Grunde immer richtig annotiert
- Fehler treten in seltenen Fällen bei Tages- und Monatsvignetten auf (Jahr stimmt nicht)
- Einteilung der „Erkennbarkeit“ mehrmals falsch (z.B. Kategorisierung als „beschädigt“ bei Unlesbarkeit), hier besteht jedoch Interpretationsspielraum

Tabelle 2: Kennwerte Datensatz 1

Datensatz 1	Juli 2019 Gesamt	„Bestaetigte“ Einzelfahrzeugbilder	„Rohdaten“ Einzelfahrzeugbilder
Anzahl Einzelfahrzeugbilder	28.422	8.421	20.001
• mit AT-Vignette	20.969	4.503	15.533
• ohne AT-Vignette	7.453	3.918	4.468
• mind. eine Jahresvignette	15.623	2.000	13.623
• mind. eine Monatsvignette	1.381	614	767
• mind. eine Tagesvignette	3.965	2.409	1.556

Für das Training des Prototyps wurden je nach Detektor-Typ spezifische Bildmengen herangezogen; die Details dazu finden sich in den Beschreibungen zu den einzelnen Detektoren.

Aus Datensatz 1 wurde eine Stichprobe von 1.000 Bildern nicht für das Training verwendet. Der Prototyp hatte diese Bilder daher im Trainingslauf nicht „gesehen“ und bewertete diese zum ersten Mal. Dieser reduzierte Datensatz wurde für eine erste Wirkungsanalyse des Prototyps verwendet (vgl. Tabellen **Fehler! Verweisquelle konnte nicht gefunden werden.** bis 5 und Abbildung 9 sowie Abbildung 10).

Der Prototyp zeigte im ersten Test gute Ergebnisse, in Bezug auf Erkennung aller Vignetten auf den Einzelfahrzeugbildern (vgl. Abbildung 9) und auch der Erkennung der richtigen Vignettentypen (Jahres-, Monats- oder Tagesvignette, sowie „X sichtbar“ und keine Vignette auf der Windschutzscheibe; vgl. Abbildung 10)

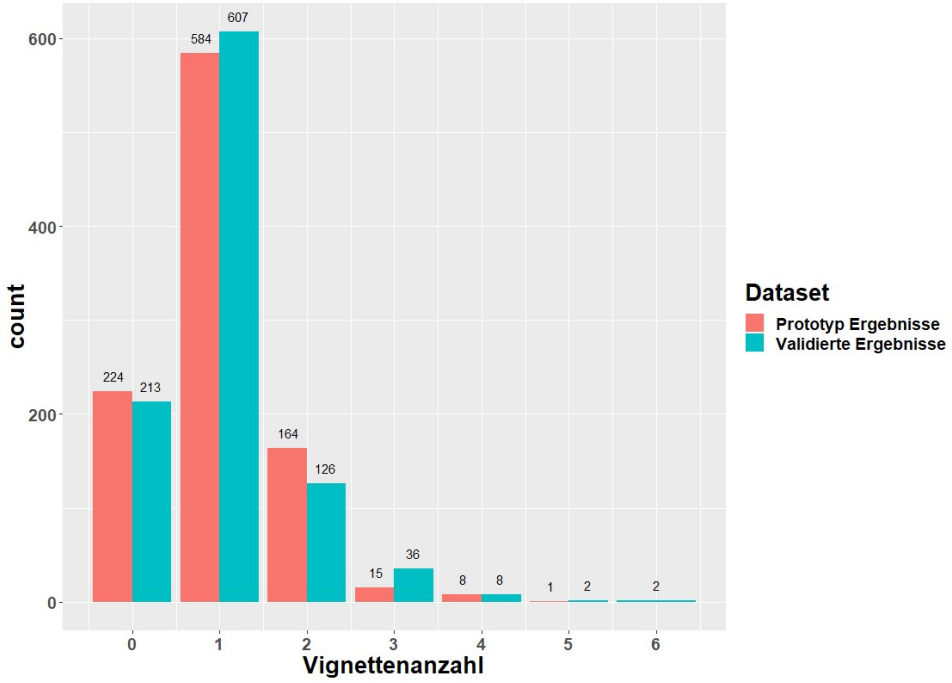


Abbildung 9: Vergleich erkannter Vignetten pro Einzelfahrzeugbild des validierten Datensatzes mit den Ergebnissen des Prototyps

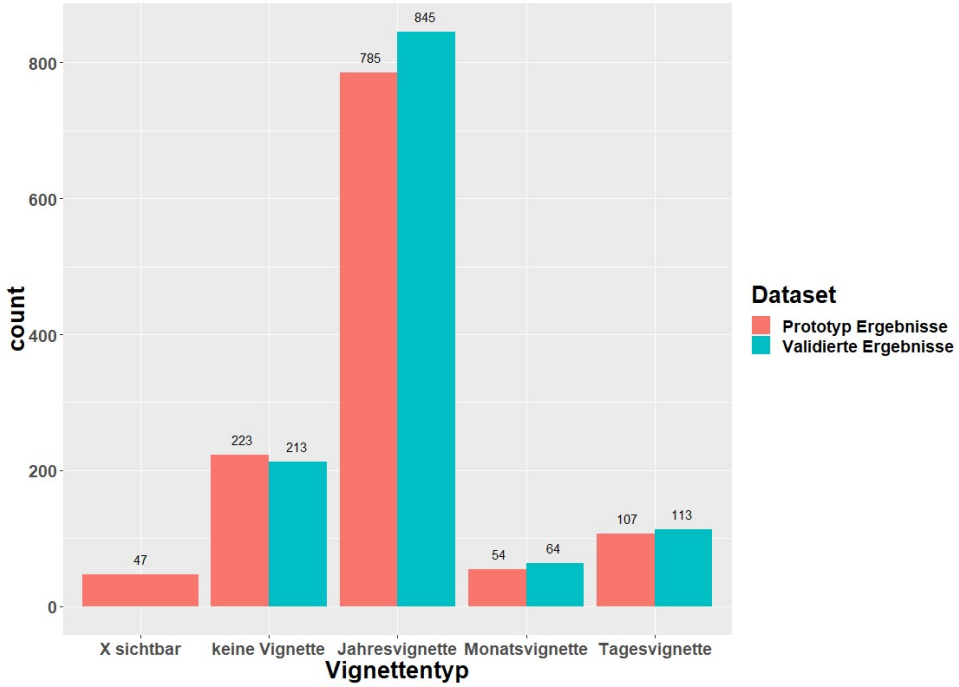


Abbildung 10: Vergleich der Anzahl erkannter Vignettentypen des validierten Datensatzes mit den Ergebnissen des Prototyps

Die folgenden Tabellen 3 bis 5 zeigen die Jahresvignetten pro Einzelfahrzeugbilder des validierten Testsamples und der erkannten Jahresvignetten pro Einzelfahrzeugbilder des Prototyps, getrennt nach der Anzahl der sichtbaren Jahres-, Monats- und Tagesvignetten.

Tabelle 3: Ergebnis erkannter Jahresvignetten pro Einzelfahrzeugbilder der validierten Einzelfahrzeugbilder und der erkannten Jahresvignetten pro Einzelfahrzeugbilder des Prototyps

Jahresvignetten / Einzelfahrzeugbilder	Anzahl validierte Einzelfahrzeugbilder	Anzahl Einzelfahrzeugbilder Prototyp	Erkannt [%]
0	830	849	102
1	709	712	100
2	127	127	100
3	34	12	35
4	4	5	125
5	1		0

Tabelle 4: Ergebnis erkannter Monatsvignetten pro Einzelfahrzeugbilder der validierten Einzelfahrzeugbilder mit den erkannten des Prototyps

Monatsvignetten / Einzelfahrzeugbilder	Anzahl validierte Einzelfahrzeugbilder	Anzahl Einzelfahrzeugbilder Prototyp	Erkannt [%]
0	1.594	1.603	101
1	90	91	101
2	16	9	56
3	5	2	40

Tabelle 5: Ergebnis erkannter Tagesvignetten pro Einzelfahrzeugbilder der validierten Einzelfahrzeugbilder mit den erkannten des Prototyps

Tagesvignetten / Einzelfahrzeugbilder	Anzahl validierte Einzelfahrzeugbilder	Anzahl Einzelfahrzeugbilder Prototyp	Erkannt [%]
0	1.305	1.388	106
1	333	275	83
2	47	34	72
3	12	6	50
4	5	1	20
5	2		0
6		1	200
7			
8			
9			

10	1		0

2.7.3. Testdatensatz 2 für Performanceprüfung des Prototyps

Der zweite Testdatensatz wurde zur Abschätzung des Rechenaufwandes herangezogen, um Hardwareeinschätzungen und Dimensionierungsanforderungen zu testen. Dieser diente dazu um die Performance des Prototyps hinsichtlich der folgenden Parameter zu testen:

- Lade- und Verarbeitungszeit der Einzelfahrzeugbilder für die ANPR
- Lade- und Verarbeitungszeit der Einzelfahrzeugbilder für die Vignettendetektion
- Maximal mögliche Anzahl an bearbeitbaren Einzelfahrzeugbildern pro Jahr mit Standardhardware

Tabelle 6: Kennwerte Datensatz 2

Datensatz 2	September 2019	Verwendung
Anzahl Bilder „Bestätigt“	12.635	• stichprobenhaft annotiert • wurde für Performancetests herangezogen
Anzahl Bilder „Rohdaten“	19.121	

Die Ergebnisse zu den Verarbeitungsgeschwindigkeiten sind im Abschnitt 2.9 dokumentiert.

2.7.4. Testdatensatz 3 zur Wirkungsanalyse des Prototyps

Für die finale Evaluierung des VMI Prototypen wurde von der ASFiNAG ein anders aufgebauter Testdatensatz zur Verfügung gestellt**. Datensatz 3 unterscheidet sich von Datensatz 1 und 2 dadurch, dass nicht nur die bestätigten Fälle von Mautprellereiverdachtsfällen enthalten sind, sondern auch die normalerweise im Rahmen des Prozesses gelöschte Einzelfahrzeugbilder.** Der Datensatz entspricht einem typischen Arbeitstag im ASFiNAG Maut Enforcement Center und setzt sich wie folgt zusammen (vgl. Tabelle 7):

- 983 bestätigte Fälle von versuchter Mautprellerei
- 18.315 gelöschte Fälle von Kfz mit gültiger Vignette

Mit diesem Datensatz wurde in der Evaluation der Prototyp gegen die manuelle Arbeit im ASFiNAG Maut Enforcement Center getestet. Der Testdatensatz enthält in einem Verhältnis von 5,09% Mautprellerei-Verdachtsfälle.

Tabelle 7: Kennwerte Datensatz 3

Datensatz 3	Oktober 2019 Gesamt	„Bestaetigte“ Einzelfahrzeugbilder	„Rohdaten“ Einzelfahrzeugbilder
Anzahl Einzelfahrzeugbilder	19.297	982	18.315
• mit AT-Vignette	15.664	572	15.092
• ohne AT-Vignette	3.633	410	3.223
• mind. eine Jahresvignette	10,755	113	10.642
• mind. eine Monatsvignette	1.596	194	1.402
• mind. eine Tagesvignette	4.225	322	3.903

Es war aus Ressourcengründen nicht möglich, den kompletten Datensatz für die anschließende Evaluierung zu annotieren, daher wurde eine Stichprobe von 2.000 Samples, wie nachfolgend beschrieben, gezogen:

- Berücksichtigung vom Verhältnis an bestätigten Fällen pro System-ID
- Berücksichtigung der Verhältnisse der einzelnen System-IDs am Gesamt-Testdatensatz.

Die Aufteilung der einzelnen gezogenen Fälle im Testsample sind in Tabelle 8 ersichtlich.

Tabelle 8: Datensatz 3 für die Evaluierung durch das ISV

System-ID	Zeitrahmen "Geloeschte" [dd.mm.yyyy hh:mm]	Zeitrahmen "Bestätigte" [dd.mm.yyyy hh:mm]	Zeitraum "Geloeschte" [d]	Zeitraum "Bestätigte" [d]	Anzahl "Gelöschte" [#]	Anzahl "Bestätigte" [#]	Verhältnis System-ID an Summe [%]	Verhältnis "Bestätigte" pro System-ID [%]	Sample Evaluation (n = 2000)	Sample "Gelöscht" [#]	Sample "Bestätigt" [#]
01	06.10.2019 11:37 - 20.10.2019 10:49	11.10.2019 11:48 - 12.10.2019 10:46	14,0	23,0	419	8	2,21%	1,87%	45	44	1
02	06.10.2019 09:35 - 20.10.2019 14:44	06.10.2019 09:44 - 20.10.2019 12:42	14,2	14,1	2293	146	12,64%	5,99%	253	237	16
03	11.10.2019 11:00 - 20.10.2019 14:39	11.10.2019 11:25 - 20.10.2019 13:38	9,2	9,1	601	29	3,26%	4,60%	65	61	4
04	06.10.2019 09:35 - 20.10.2019 13:00	11.10.2019 09:47 - 15.10.2019 08:10	14,1	3,9	1281	25	6,77%	1,91%	135	132	3
10	06.10.2019 12:09 - 15.10.2019 09:08	11.10.2019 09:44 - 14.10.2019 14:33	8,9	3,2	632	63	3,60%	9,06%	72	65	7
11	11.10.2019 09:48 - 14.10.2019 12:18	11.10.2019 09:47 - 14.10.2019 09:53	3,1	3,0	101	22	0,64%	17,89%	12	9	3
12	06.10.2019 11:35 - 17.10.2019 10:39	06.10.2019 11:50 - 16.10.2019 17:50	11,0	10,2	1004	102	5,73%	9,22%	115	104	11
14	06.10.2019 11:35 - 17.10.2019 10:40	06.10.2019 11:49 - 17.10.2019 10:18	11,0	10,9	987	54	5,39%	5,19%	108	103	5
15	06.10.2019 11:41 - 15.10.2019 08:36	06.10.2019 12:45 - 15.10.2019 07:41	8,9	8,8	1239	11	6,48%	0,88%	129	128	1
16	06.10.2019 11:36 - 15.10.2019 08:00	06.10.2019 12:58 - 14.10.2019 14:50	8,8	8,1	2330	45	12,31%	1,89%	247	243	4
22	06.10.2019 11:36 - 15.10.2019 08:59	06.10.2019 11:41 - 14.10.2019 18:11	8,9	8,3	522	39	2,91%	6,95%	59	55	4
23	06.10.2019 11:44 - 15.10.2019 09:30	11.10.2019 10:04 - 14.10.2019 17:47	8,9	3,3	651	19	3,47%	2,84%	69	68	1
24	06.10.2019 11:36 - 15.10.2019 09:22	06.10.2019 11:40 - 14.10.2019 17:26	8,9	8,2	489	36	2,72%	6,86%	55	51	4
25	06.10.2019 11:37 - 15.10.2019 08:51	06.10.2019 12:07 - 15.10.2019 09:33	8,9	8,9	659	85	3,86%	11,42%	76	68	8
26	06.10.2019 11:35 - 15.10.2019 09:43	06.10.2019 11:38 - 15.10.2019 09:09	8,9	8,9	934	104	5,38%	10,02%	107	96	11
27	06.10.2019 11:52 - 15.10.2019 09:37	06.10.2019 15:11 - 14.10.2019 14:09	8,9	8,0	2128	70	11,39%	3,18%	228	220	8
28	06.10.2019 11:35 - 18.10.2019 13:00	06.10.2019 11:35 - 14.10.2019 17:04	12,1	8,2	544	31	2,98%	5,39%	60	56	4
29	06.10.2019 11:35 - 18.10.2019 16:35	06.10.2019 11:52 - 15.10.2019 09:09	12,2	8,9	751	39	4,09%	4,94%	81	77	4
30	06.10.2019 12:11 - 14.10.2019 10:04	06.10.2019 12:46 - 14.10.2019 07:57	7,9	7,8	750	55	4,17%	6,83%	84	79	5
					18315	983	100,00%		2000	1896	104

2.7.5. Überprüfung der Zielvorgaben

Zur Überprüfung der Zielvorgaben aus der Ausschreibung wurden zufällig 2000 Einzelfahrzeugbilder (1896 gelöschte und 104 bestätigte Einzelfahrzeugbildern) aus dem Testdatensatz 3 gezogen und manuell annotiert (vgl. Tabelle 8). Alle Auswertungen basieren auf diesem Testsample von 2000 Bildern. Mit etwa 5% an Verdachtsfällen bilden Sie einen einen repräsentativen Querschnitt an Prüffällen, wie sie derzeit im Maut Enforcement Center verarbeitet werden. Für die nachfolgenden Auswertungen gelten folgende Begriffsbestimmungen:

- **False Negative (FN):** Vignette vorhanden, keine Erkennung der Vignette
- **False Positive (FP):** Vignette nicht vorhanden, anderes Objekt als Vignette erkannt
- **True Negative (TN):** Vignette nicht vorhanden, keine fälschliche Erkennung
- **True Positive (TP):** Vignette vorhanden, korrekte Erkennung
- **Einzelfahrzeugbild:** Bild der Windschutzscheibes eines Fahrzeugs
- **Fall:** jede vorhandene Vignette oder eine Vignettenlose Windschutzscheibe entspricht einem Fall

Fragestellung 1

Erkennung von mind. 95% aller auf den Bildern vorhandenen Vignetten mit max. 5% falsch positiven Identifikationen

Begriffsdefinition ‚Fall':

Die einzelnen Vignetten wurden als separate Fälle beurteilt, ebenso wird es als ein Fall angesehen, wenn keine Vignetten auf der Windschutzscheibe vorhanden sind. Wenn eine vollständig oder unvollständig sichtbare Vignette bei manueller Kontrolle ersichtlich ist, wurde dies ebenfalls als ein Fall bewertet und per Parameter „Erkennung" beschrieben.

Das annotierte Testsample wurde über den Fall, den vom Prototyp erkannten Typ und die Position der oberen linken Vignettenkante mit dem manuell annotierten Datensatz verknüpft.

Dabei wurden **90,04%** (TP: 75,39% / TN: 14,27% / 2133 Fälle / 1889 Einzelfahrzeugbilder) der manuell annotierten Fälle richtig **mit Typ und Position bewertet** und 236 Fälle (FN: 8,27% / FP: 0,93% / 9,96% / 195 Einzelfahrzeugbilder) falsch bewertet. Durch die den verwendeten Deep Learning Algorithmen zugrundeliegenden Methodiken ist eine Analyse von Gründen für

falsch bewertete Situationen nicht einfach durchzuführen. Für die 236 falsch bewerteten Fälle werden in Abbildung 12 plausible Ursachen angeführt und anschließend diskutiert.

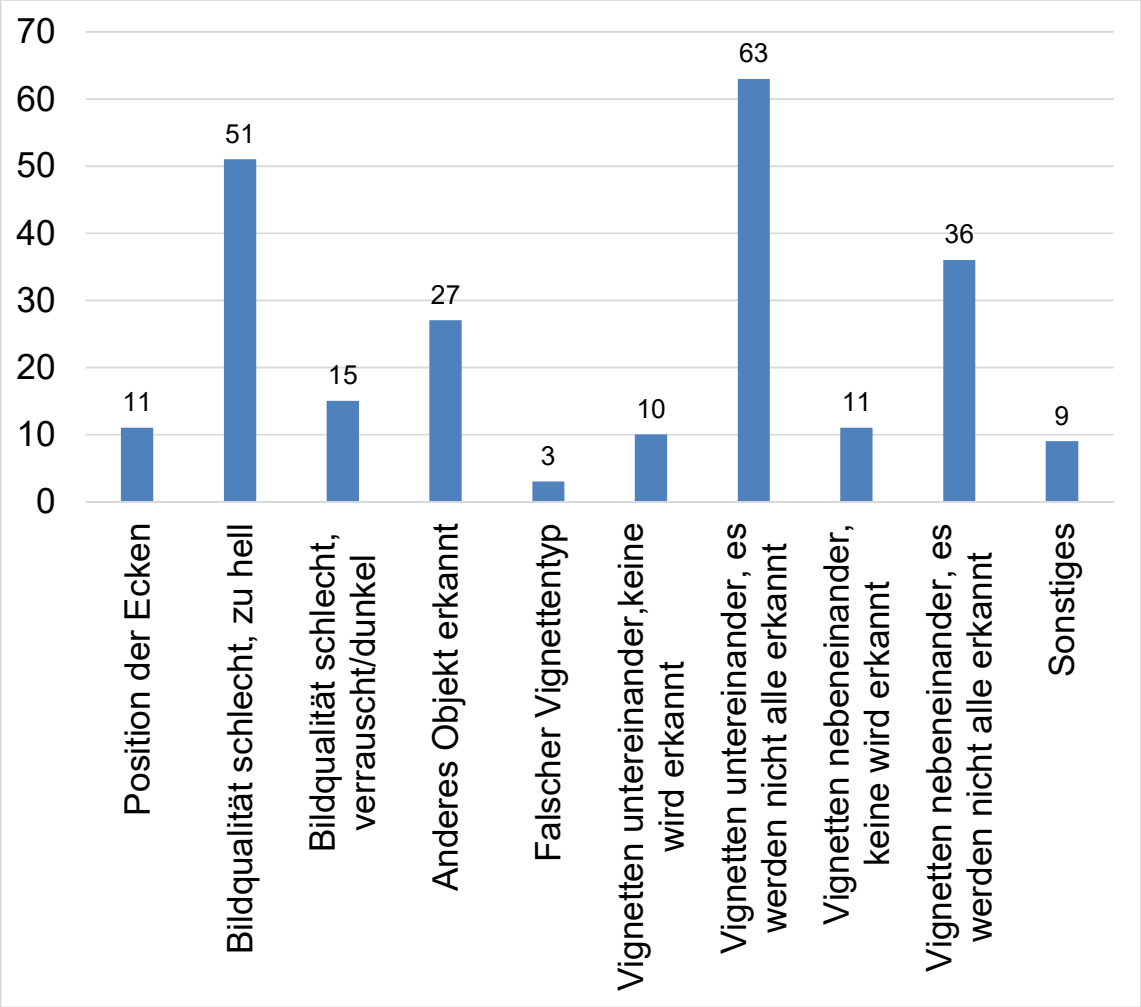
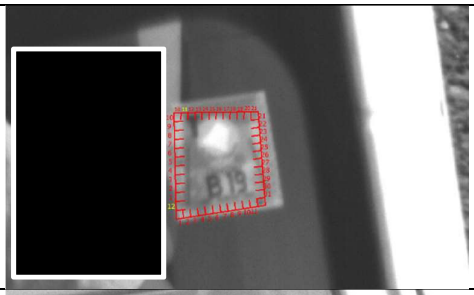
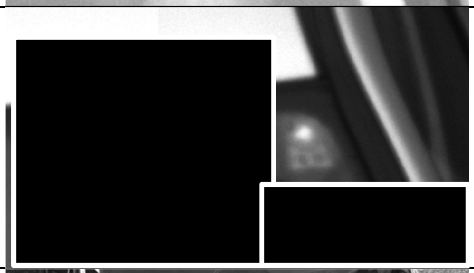
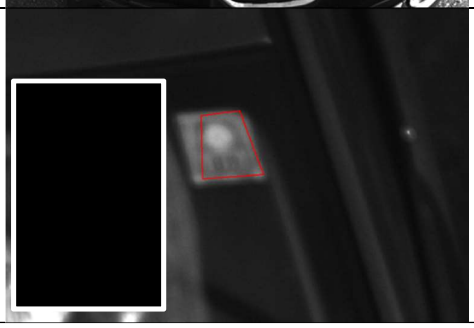


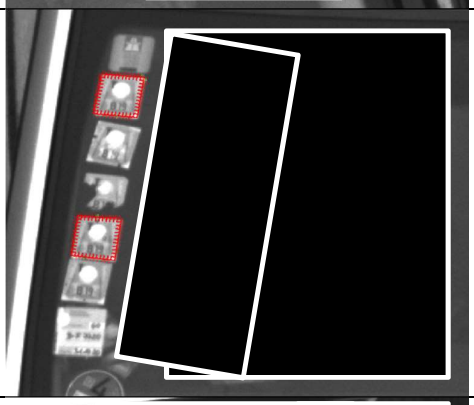
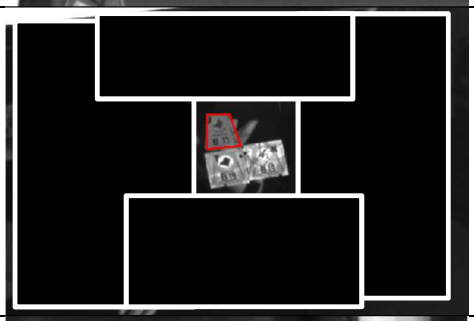
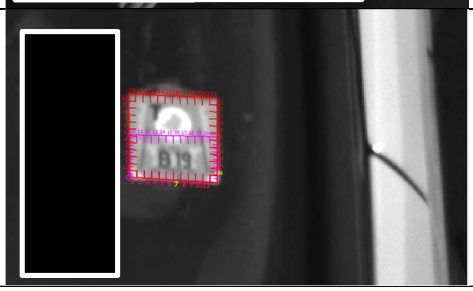
Abbildung 11: Auflistung plausibler Ursachen für Falschklassifikationen des Prototyps, des Testsamples aus Datensatz 3

Die folgende Tabelle 9 zeigt typische Beispiele für Ursachen von Falschklassifikationen.

Bemerkung: In den Beispielbildern wurden personenbezogene Details ausmaskiert.

Tabelle 9: Typische Beispiele für Ursachen von Falschklassifikationen

Bild	Kategorie	Beschreibung
	1	Position der Ecken
	2	Bildqualität schlecht, zu hell
	3	Bildqualität schlecht, verrauscht/dunkel
	4	Anderes Objekt erkannt
	5	Falscher Vignettentyp

	6 8	Vignetten untereinander, keine wird erkannt Vignetten nebeneinander, keine wird erkannt
	7	Vignetten untereinander, es werden nicht alle erkannt
	9	Vignetten nebeneinander, es werden nicht alle erkannt
	10	Sonstiges

Fragestellung 2

Korrekte Angabe der Koordinaten der Eckpunkte für mind. 90% der erkannten Vignetten

Die korrekte Angabe der Eckpunktkoordinaten für die jeweilige Vignette ist schwierig zu beurteilen, da auch **der manuelle Vergleichswert immer in einem bestimmten Ausmaß fehlerbehaftet** ist. Diese Fragestellung wird daher über die Beschreibung der Abweichungen

der manuellen Annotation und der Annotation des Prototyps der X- und Y-Koordinaten aller 4 Eckpunkte der detektierten Vignetten (vgl. hierzu die Boxplots in Abbildung 10) beantwortet und zusätzlich in Bezug zur horizontalen und vertikalen Kantenlänge der manuell annotierten Vignetten gesetzt:

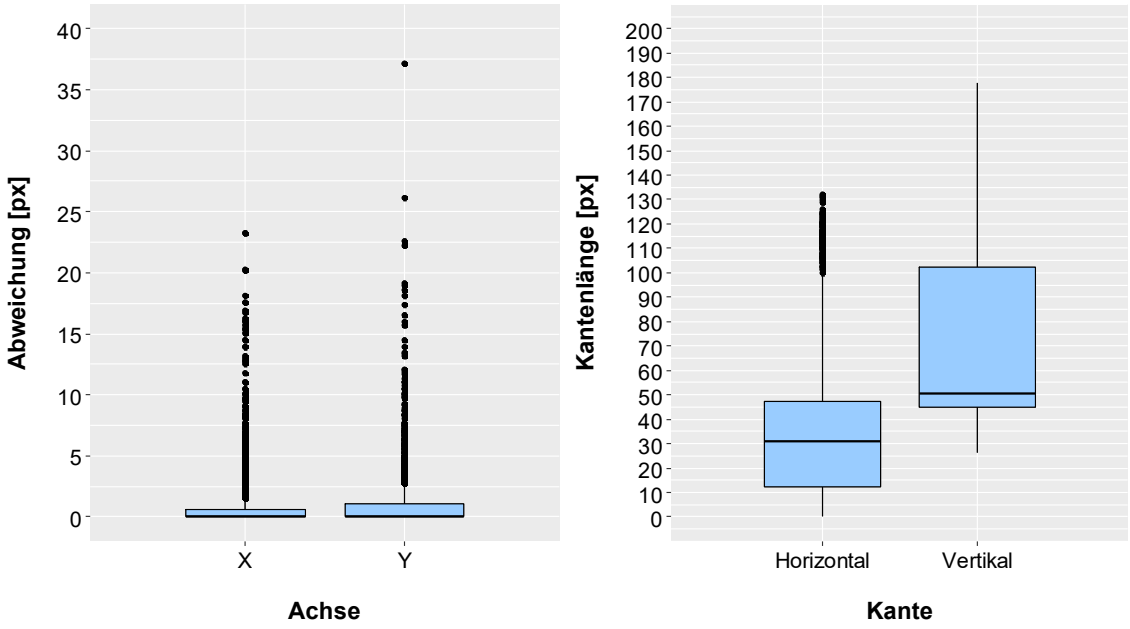


Abbildung 12: Abweichung X- bzw. Y-Koordinaten aller vier Ecken der automatisch detektierten Vignetten im Vergleich mit den Kantenlängen der manuell annotierten Vignetten (n = 1339 Fälle / 1181 Einzelfahrzeugbilder)

Für die **X-Koordinaten wurde ein 90%-Perzentil (90% der Werte liegen darunter) von 2,09 px** und **2,76 px für die Y-Koordinaten ermittelt**. Das 90%-Perzentil für die Kantenlängen entspricht 76 px für die horizontalen und 127 px für die vertikalen Kantenlängen.

Fragestellung 3

Erkennung der Gültigkeit und korrekte Angabe der relevanten Eigenschaften (Jahr bzw. Jahr/Monat/Tag entsprechend der Lochung) für mind. 90% der erkannten Vignetten

Die Erkennung der Gültigkeit und korrekten Angabe der relevanten Eigenschaften wurden aufgrund der unterschiedlichen Eigenschaften nach Jahres-, Monats- und Tagesvignetten

getrennt. Im Vergleich zu den Fragestellungen 1 und 2 muss zu Fragestellung 3 festgehalten werden, dass diese **besonders für die Erkennung der Monats- und Tagesvignetten deutlich schwieriger zu erfüllen ist**. Für die Beantwortung der Fragestellung 3 wurde nach Vignettentypen aus dem Jahr 2019 gefiltert, da die Prototyperkennung darauf speziell trainiert wurde.

- Von 756 Jahresvignetten (756 Einzelfahrzeugbilder) aus dem Jahr 2019 wurden **77,6%** (587 Jahresvignetten / 587 Einzelfahrzeugbilder) vollständig korrekt (Jahr und Typ) erkannt.
- Von 482 Monats- und Tagesvignetten (410 Einzelfahrzeugbilder) aus dem Jahr 2019 wurden **73,2%** (353 Monats- und Tagesvignetten / 323 Einzelfahrzeugbilder) vollständig korrekt (Typ, Jahr, Monat, Tag) erkannt.

Für die 169 falsch bewerteten Jahresvignetten-Fälle werden die festgestellten Probleme in Abbildung 13 dargestellt.

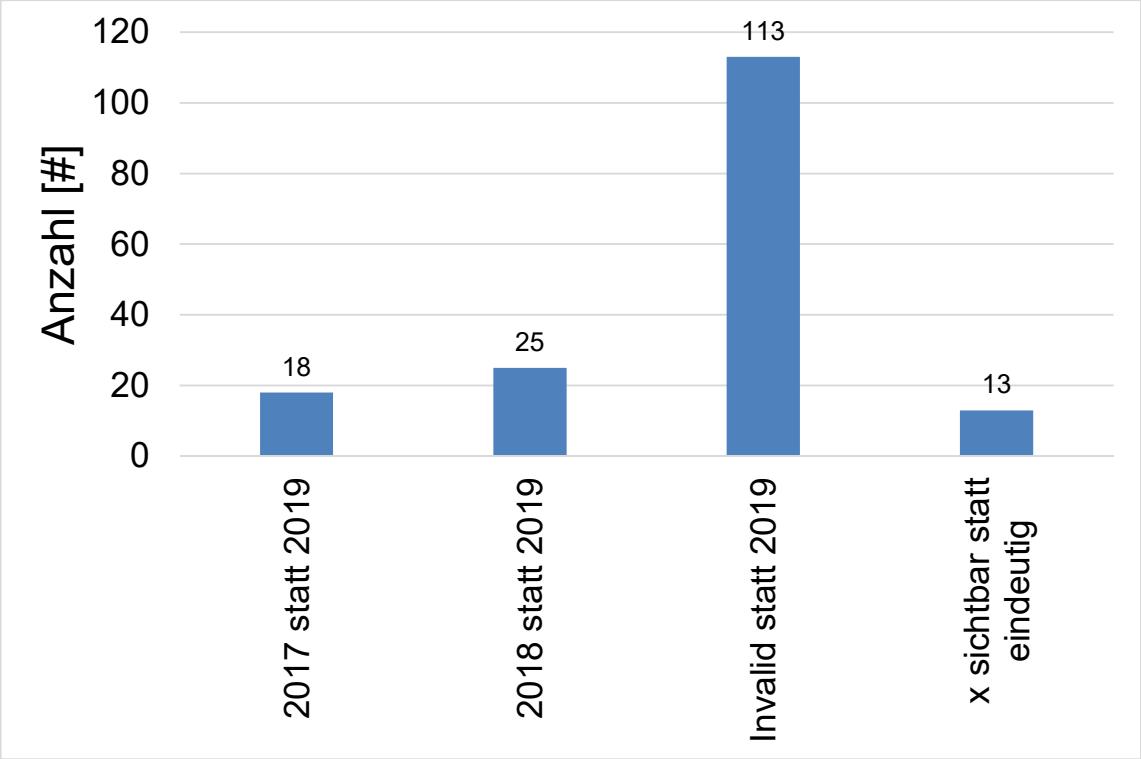


Abbildung 13: Probleme in der Jahresvignettenparametererkennung (n = 169 Fälle / 169 Einzelfahrzeugbilder)

Für die 129 falsch bewerteten Monats- und Tagesvignetten-Fälle werden die festgestellten Probleme in Abbildung 14 visualisiert.

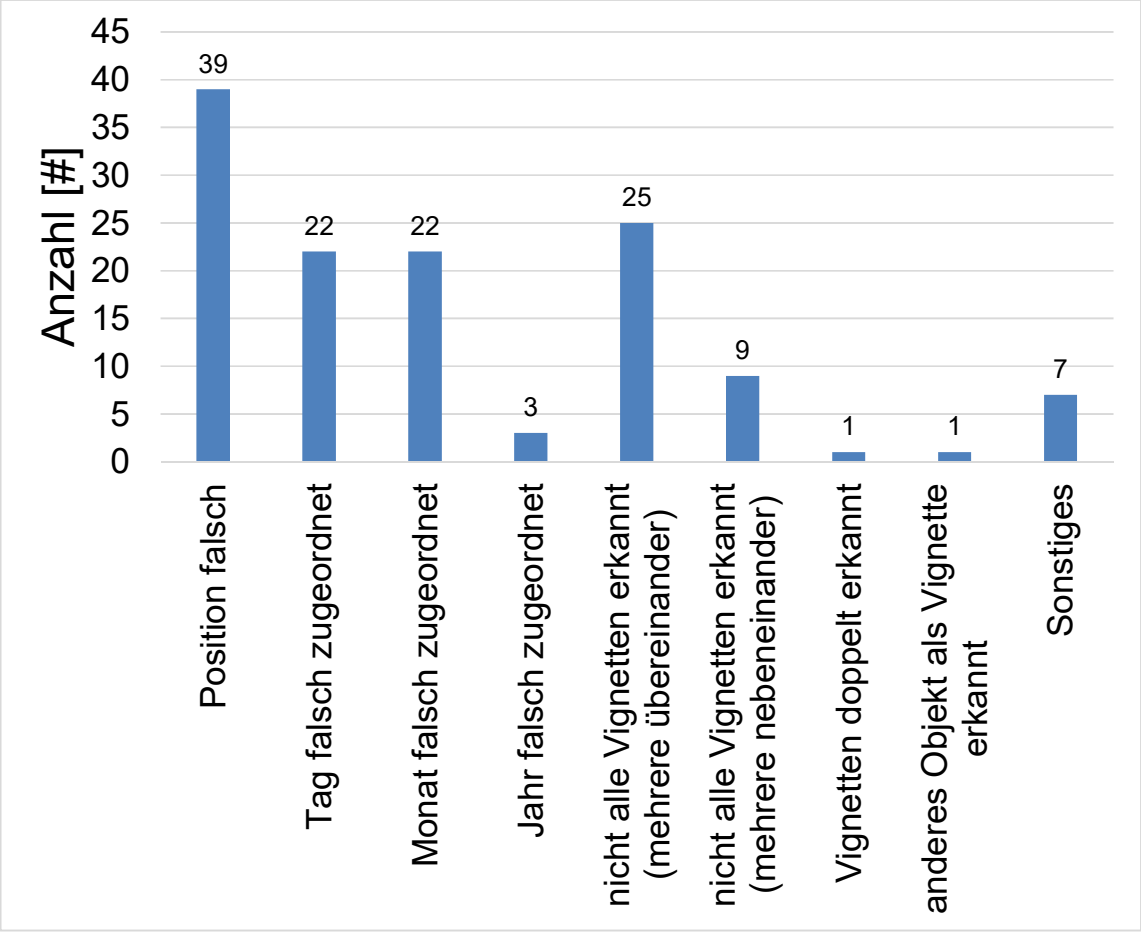


Abbildung 14: Probleme in der Monats- und Tagesvignettenparametererkennung (n = 129 Fälle / 106 Einzelfahrzeugbilder)

Fragestellungen 4 und 5

Korrekte Erkennung der Zeichenkette für mind. 95% der Kennzeichen auf den Bildern

Korrekte Erkennung des Zulassungsstaates für mind. 90% der Kennzeichen auf den Bildern

Von der ASFiNAG wurden bereits (im Maut Enforcement Center) vorannotierte Kennzeichen in bestätigten Einzelfahrzeugbildern zur Verfügung gestellt, diese Kennzeichen wurden im ersten Schritt mit den korrespondierenden Kennzeichen aus dem Testsample verglichen und wiesen die in

Tabelle 10 dargestellten Werte auf.

Tabelle 10: Ergebnis ANPR für die bestätigte Fälle des Testsamples aus Datensatz 3 (n = 105 Einzelfahrzeugbilder) und den Vergleich der ASFiNAG-Lösung mit CARRIDA von SLR

Erkennung von	ASFiNAG	CARRIDA (SLR)
Nation	100,0%	96,2%
Kennzeichen	92,3%	98,1%
beide	92,3%	95,2%

Der Datensatz enthält 3 Länder, welche in dem für den Prototyp vorbereiteten *CARRIDA*-Classifier nicht integriert waren (RU, MD und DK). Dieser Classifier wird aus Performancegründen in unterschiedlichen Anwendungen auf den konkreten geographischen Einsatzbereich zugeschnitten und kann **in zukünftigen Anwendungen für die ASFiNAG entsprechend vorbereitet werden, um auch die jetzt fehlenden Länder zu integrieren.**

Bemerkung: Die 100% Erkennungsrate der Nation in den ASFiNAG Lesungen kann nicht einer tatsächlichen Leserate entsprechen, sie dürfte durch eine vorab Filterung bzw. Korrektur zustande kommen – nur gewisse Länder werden derzeit von der ASFiNAG geprüft.

Auf das komplette Testsample angewendet, ergeben sich die in Tabelle 11 dargestellten Werte.

Tabelle 11: Ergebnis ANPR für das Testsample aus Datensatz 3 (n = 2000 Einzelfahrzeugbilder) für CARRIDA von SLR

Erkennung von	CARRIDA (SLR)	ohne nicht enthaltene Länder
Nation	97,2%	97,4%
Kennzeichen	97,0%	97,0%
beide	95,3%	95,5%

2.7.6. Anwendung des Prototyps und Klassifizierung der Testsamples anhand des Algorithmus zur Vignetten Gültigkeits-Bestimmung

Im abschließenden Schritt der Evaluierung wurde der in Kapitel 2.4.9 vorgestellte Algorithmus zur Vignetten Gültigkeits-Bestimmung auf das Testsample von 2.000 Einzelfahrzeugbildern angewendet. Dieser Schritt sollte die Leistungsfähigkeit des Prototyps im Vergleich zur manuellen Durchsicht im ASFiNAG Maut Enforcement Center testen.

In der Bearbeitung erwies sich das fehlende Wissen über eventuell vorhandene digitale Vignetten als problematisch. Aus dem Testsample wurden daher Einzelfahrzeugbilder ohne visuell erkennbare Vignette herausgefiltert und diese zur Überprüfung durch die ASFiNAG angefragt. Anhand der Überprüfung konnten, durch die gültige digitale Vignette, im nächsten

Schritt 282 Einzelfahrzeugbilder ausgeschlossen werden. Im Fall dieser Einzelfahrzeugbilder kann, anhand der Performancetests im vorangegangen Abschnitt ausgegangen werden, dass 97% der Kennzeichen richtig gelesen werden (274 Einzelfahrzeugbilder).

Das Ergebnis der Anwendung des Algorithmus auf das reduzierte Testsample wird in Abbildung 15 dargestellt.

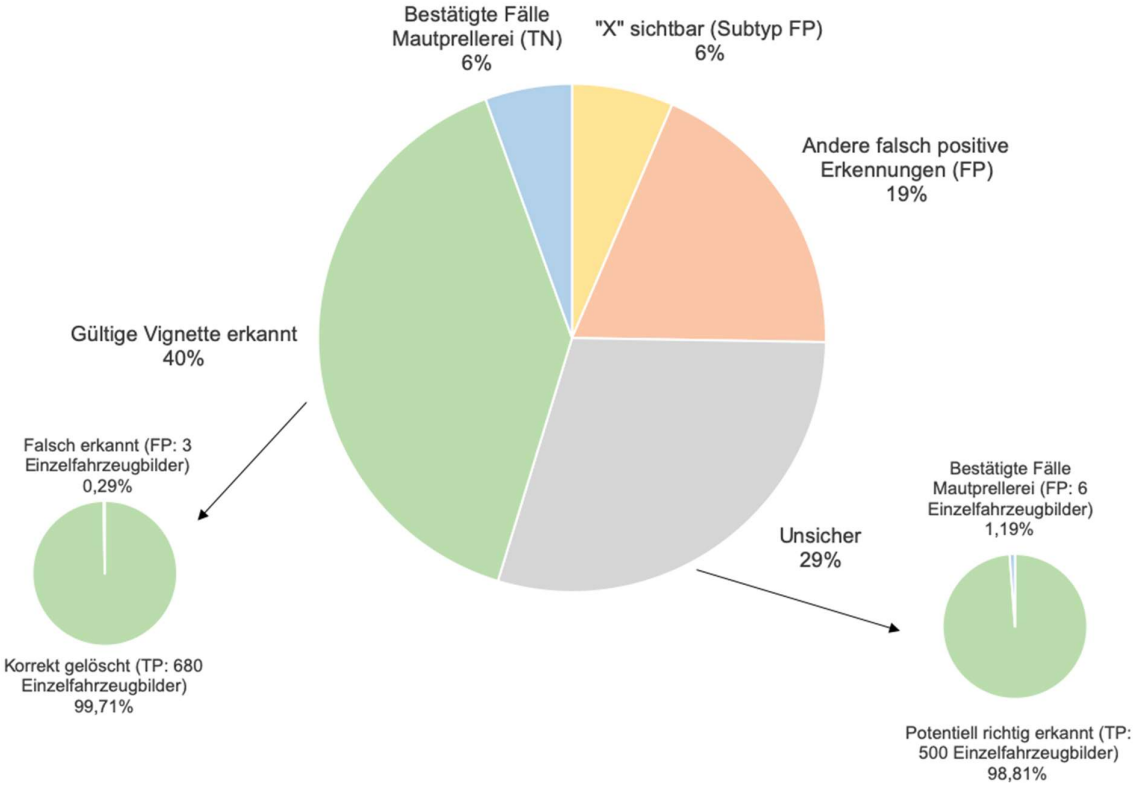


Abbildung 15: Ergebnis der Vignetten Gültigkeits-Bestimmung für das reduzierte Testsample (n = 1.718 Einzelfahrzeugbilder)

Durch das Ergebnis wird die Abhängigkeit der Leistungsfähigkeit des Prototyps von der gewählten Sensitivität der finalen Entscheidung bei der Beurteilung deutlich. Mit einer erhöhten Sensitivität (Variante 1) ist es **möglich ca. 46% der Einzelfahrzeugbilder zuverlässig als gültige Klebevignette erkannt oder Mautprellereiverdachtsfall zu klassifizieren** (von 778 Einzelfahrzeugbildern werden 2 fälschlicherweise als gültig angesehen, das entspricht ca. 0,26%). Somit wären ca. 54% der Einzelfahrzeugbilder noch einer manuellen Kontrolle zu unterziehen.

Wenn hingegen mit einer etwas abgesenkten Sensitivität gearbeitet werden würde (Variante 2) und die vorher als „unsicher“ klassifizierten Einzelfahrzeugbilder ebenfalls als gültig

angesehen werden würden, wären vorab **ca. 75% der Einzelfahrzeugbilder vollständig klassifiziert**. Dieser Schritt erhöht im Umkehrschluss jedoch auch die Fehlerrate von ca. 0,26% auf ca. 0,62%, d.h. von 1.284 Einzelfahrzeugbildern wären 8 Bilder falsch klassifiziert.

Die folgende Tabelle 12 fasst die Ergebnisse des VMI Projektes in Bezug auf praxisrelevante Kenngrößen zusammen. Sie gibt an, welchen Einfluss die VMI Ergebnisse auf den Prozess beim Maut Enforcement Center der ASFiNAG hätten.

Für jede der beiden Varianten 1 und 2 wird aufgelistet, welcher Anteil der Bilder sofort gelöscht werden könnte, welcher Arbeitsaufwand relativ zur Eingangsdatenmenge für die ASFiNAG verbleiben würde, und wie viele ‚bestätigte' Einzelfahrzeugbilder (relativ zur Referenzauswertung) verloren gehen würden:

Tabelle 12: Zusammenfassung der Kennwerte für die Varianten 1 und 2 (Werte gerundet):

Variante	Klassifikation akzeptieren (sofort löschen bzw. Verdachtsfall prüfen)	Manuelle Kontrolle nötig (= Arbeitsaufwand)	Verlust ‚Bestätigte'
1	46%	54%	1.9% (2 von 104)
2	75%	25%	7,7% (8 von 104)

2.7.7. Zusammenfassung der Evaluierung

Zur Bearbeitung dieser Fragestellung wurde im Projekt VMI eine Methodik aus einer Kombination aus verschiedenen Deep Learning Ansätzen und klassischer Bildverarbeitung gewählt. Deep Learning Methoden sind, aufgrund ihrer Robustheit und Anpassungsfähigkeit, eine gute Möglichkeit, um mit derartigen Fragestellungen umzugehen. Daher werden Deep Learning Methoden mittlerweile in zahlreichen Industrieanwendungen erfolgreich eingesetzt. Der Prototyp zeigte während der Entwicklung ein großes Potential zur Erhöhung der automatischen Klassifikation von Einzelbildern. Die Erkennungsrate konnte während der Projektlaufzeit kontinuierlich gesteigert werden. Im ersten Test konnten so bereits 55,1% an Fällen (1182 Fälle / 1.067 Einzelfahrzeugbilder) eines Testsamples mit 2144 Fällen (1706 Einzelfahrzeugbilder) erkannt werden. Diese konnten innerhalb von 2 Monaten auf eine richtige Erkennung von 90,04% (TP: 75,39% / TN: 14,27% / 2133 Fälle / 1889 Einzelfahrzeugbilder) der manuell annotierten Fälle mit Typ und Position und 236 falsch bewerteten Fälle (FN: 8,27% / FP: 0,93% / 9,96% / 195 Einzelfahrzeugbilder) deutlich gesteigert werden. Der Entwicklungsstand des Prototyps zu Projektende ermöglicht es,

Vignetten mit den zugehörigen 4 Ecken sehr akkurat mit einer Abweichung von +/- 3 px zu der menschlichen Interpretation der Vignettenecken zu bestimmen. Durch das umfangreiche Training an Realdaten aus dem laufenden Betrieb des Maut Enforcement Centers aus dem Jahr 2019 konnte ein hoher Anteil an Vignetten in der Evaluierung des Testsamples vollständig korrekt erkannt werden:

- Von 756 Jahresvignetten (756 Einzelfahrzeugbilder) aus dem Jahr 2019 wurden 77,6% (587 Jahresvignetten / 587 Einzelfahrzeugbilder) vollständig korrekt (Jahr und Typ) erkannt.
- Von 482 Monats- und Tagesvignetten (410 Einzelfahrzeugbilder) aus dem Jahr 2019 wurden 73,2% (353 Monats- und Tagesvignetten / 323 Einzelfahrzeugbilder) vollständig korrekt (Typ, Jahr, Monat, Tag) erkannt.

Die Gründe für die Nicht- oder Falscherkennung sind bei Deep Learning Ansätzen oft schwer zu bestimmen. Für den Prototyp dürften diese, nach manueller Durchsicht, in der oft schlechten Bildqualität (vielfach sind Vignetten verrauscht oder eine weiße/dunkle Fläche auf der Windschutzschiebe), sowie in dem schlechten Klebeverhalten auf Kundenseite zu suchen sein. Speziell die Bestimmung von aneinandergrenzenden Vignetten stellt eine komplexe Aufgabe dar, die mit dem gewählten Netzwerk in der knappen Entwicklungszeit noch nicht befriedigend gelöst werden konnte.

Das Lesen der Kennzeichen und die Bestimmung des Zulassungsstaates konnte durch das kommerzielle Produkt CARRIDA bereits am Anfang zufriedenstellend gelöst werden.

Mit der Anwendung des Algorithmus zur Vignetten Gültigkeits-Bestimmung konnte die Wichtigkeit der korrekten Sensitivitätseinstellung, zur Erreichung des gewünschten Automatisierungsgrades, aufgezeigt werden. Dadurch können beispielsweise, wenn mit einer etwas abgesenkten Sensitivität gearbeitet werden würde (Variante 2 im vorangegangen Abschnitt), und die in Variante 1 (mit erhöhter Sensitivität) als „unsicher“ klassifizierten Einzelfahrzeugbilder ebenfalls als gültig angesehen werden. Dies hätte zur Folge, dass vorab ca. 75% der Einzelfahrzeugbilder klassifiziert werden würden. Dieser Schritt erhöht im Umkehrschluss jedoch die Fehlerrate von ca. 0,39% in Variante 1 auf ca. 0,70% bei Variante 2, d.h. von 1.284 Einzelfahrzeugbildern wären ca. 9 Bilder falsch klassifiziert.

Abschließend ist festzuhalten, dass der Prototyp im Test das Potential gezeigt hat, von den bisher vollständig manuell bearbeiteten Fällen in Summe etwa 75% zukünftig automatisiert bewerten zu können.

2.8. Sensitivitätsanalyse des Prototyps

Für einige Detektionsverfahren ist es möglich, eine Analyse hinsichtlich des Detektionsverhaltens bei unterschiedlicher Empfindlichkeit der Detektoren durchzuführen (Sensitivitätsanalyse). Die Ergebnisse sind in den *ROC* Kurven und *precision-recall* Kurven zusammengefasst. Mit der Receiver Operator Characteristic wird das Verhältnis zwischen der Erkennung von TruePositive und TrueNegative ausgedrückt. In der Regel steigt mit zunehmender Erkennung der korrekten Fälle auch die Anzahl fehlerhafter Zuordnungen.

Die Berechnungen wurden jeweils mit den 2000 manuell annotierten Ground Truth Daten aus dem 3. Testsatz berechnet.

2.8.1. X – Detektion

Im Datensatz 3 sind nur 16 ground truth X zu finden, die resultierenden Kurven sind daher nur relativ grob, aber dennoch aufschlussreich.

Die linke ROC zeigt, dass bei einer akzeptieren False Positive Rate von ca. 0.05 (= 5%), **bezogen auf die gesamte Sample Menge** (= 2000 Samples) alle X gefunden werden können.

In der Precision-Recall Kurve rechts drückt sich dies in der Tatsache aus, das bei einem Recall von 1.0 (= alle 13 X gefunden) die Genauigkeit ca. 0.3 beträgt – es werden alle X gefunden, aber auch ca. doppelt so viele FP (ca. 26) erzeugt.

Der Grund für diese relativ schlechte Rate ist das Ungleichgewicht von relativ wenigen X (13) in Relation zur gesamten Testmenge Menge von 2000 Vignetten.

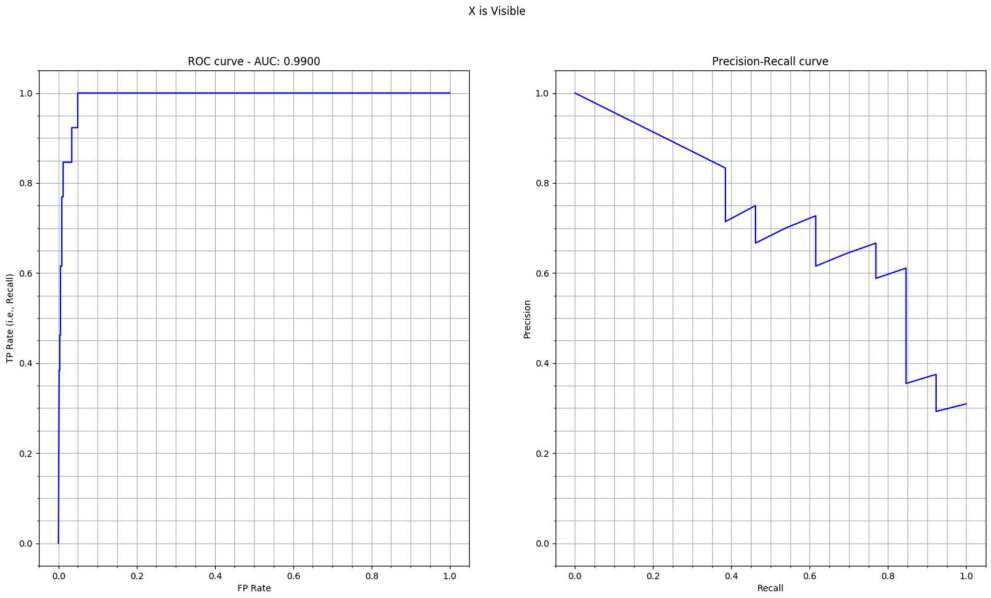


Abbildung 16: ROC Kurve (links) und precision-recall Kurve (rechts) für die X-Detektion. Die Achsenbeschriftungen beziehen sich auf die Gesamtmenge im Testset, bspw. 0,5 = 50%, 1.0 = 100%.

2.8.2. Jahreszahl 19 Klassifikation

Für die Auswertung der Jahreszahl stehen wesentlich Samples zur Verfügung. Die Kurven geben wieder, das der B 19 Detektor vergleichsweise diskriminativ arbeitet, aber sicherlich noch etwas Raum für Verbesserung besteht. Mit zusätzlichem Training wäre hier vermutlich noch eine bessere Klassifikation möglich. Im Laufe der Projektarbeit wurde aber auch festgestellt, dass die Gefahr besteht, dass bei Einbeziehung zu schlechter Trainingssamples (unschärfere Bilder, Rauschen) mehr Verwechselungen mit anderen Jahreszahlen auftreten.

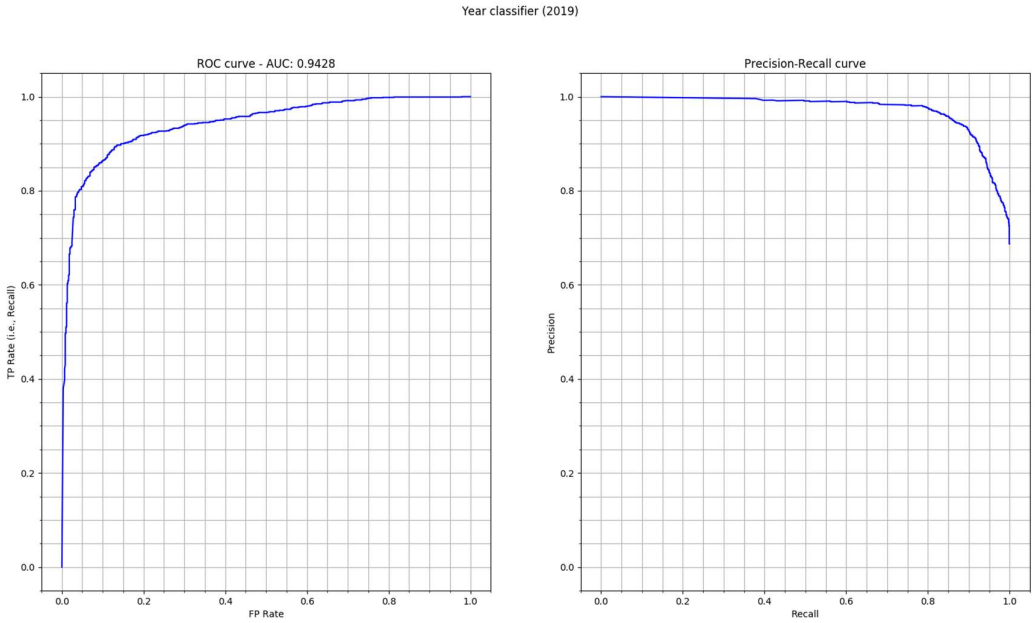


Abbildung 17: ROC Kurve (links) und precision-recall Kurve (rechts) für die Detektion der B 19 Beschriftung. Die Achsenbeschriftungen beziehen sich auf die Gesamtmenge im Testset, bspw. 0,5 = 50%, 1.0 = 100%.

Eine weitere Analyse zeigt die Verteilung der Konfidenzen für jeweils die korrekte Klassifikation (oben) und die falsche Klassifikation (unten). Es ist ersichtlich, dass auch für falsche Klassifikationen die Konfidenzen hoch werden und damit nicht mehr als Fehler erkennbar sind:

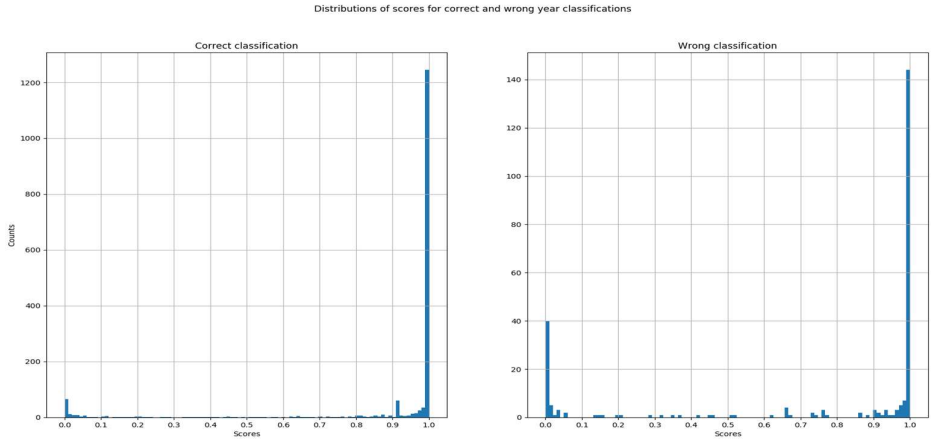


Abbildung 18: Die Verteilung der Konfidenzen für jeweils die korrekte Klassifikation (links) und die falsche Klassifikation (rechts) für alle Jahreszahlen 17, 18, 19 und ‚ungültig'.

Eine Variation der akzeptierten Konfidenzen (ab welcher minimaler Konfidenz ist eine Detektion als gültig akzeptiert=?) erlaubt es zu analysieren, in welchen Grenzen eine Abwägung zwischen der Anzahl an korrekten und der Anzahl an inkorrekten Detektionen getroffen werden kann.

Die rote Kurve zeigt die erreichte Genauigkeit (TP/(TP+FP)), die blaue Kurve den Anteil der korrekten Klassifikationen, die orange Kurve den Anteil der inkorrekten Klassifikationen und die grüne Kurve die Anzahl der ‚ungültigen' Klassifikationen – **die ungültigen Klassifikationen würden im VMI System einer manuellen Nachkontrolle zugeordnet.**

Bei einem Grenzwert von bspw. 0.8 (x-Achse unten) würden ca. 85% Jahreszahlen korrekt klassifiziert (blau), es gäbe ca. 13% Fehlklassifikationen (grün), und ca. 20% würden der manuellen Kontrolle zugeordnet (grün). Es würde eine Genauigkeit von etwas weniger als 90% erreicht (rot).

Die Kurven zeigen sehr klar, dass bis zu einem Grenzwert von .90 keine großen Änderungen möglich sind, **ab einem Grenzwert von ca. .98 ändern sich die Relationen stark und schnell**. In diesem Bereich (0.98 – 1.0) könnte durch Anpassung des Grenzwertes das Verhältnissen zwischen korrekten und inkorrekten Detektionen je nach Anforderung etwas angepasst werden.

Es ist davon auszugehen, dass durch die Einbeziehung weiterer Trainingsbeispiele die Detektionsgenauigkeit und Fähigkeit zur korrekten Diskrimination weiter verbessert werden kann. Insbesondere müssten noch deutlich mehr Vignetten aus den vergangenen Jahren mit trainiert werden, um das derzeit vorhandene Ungleichgewicht in der Verteilung der Samples auszugleichen.

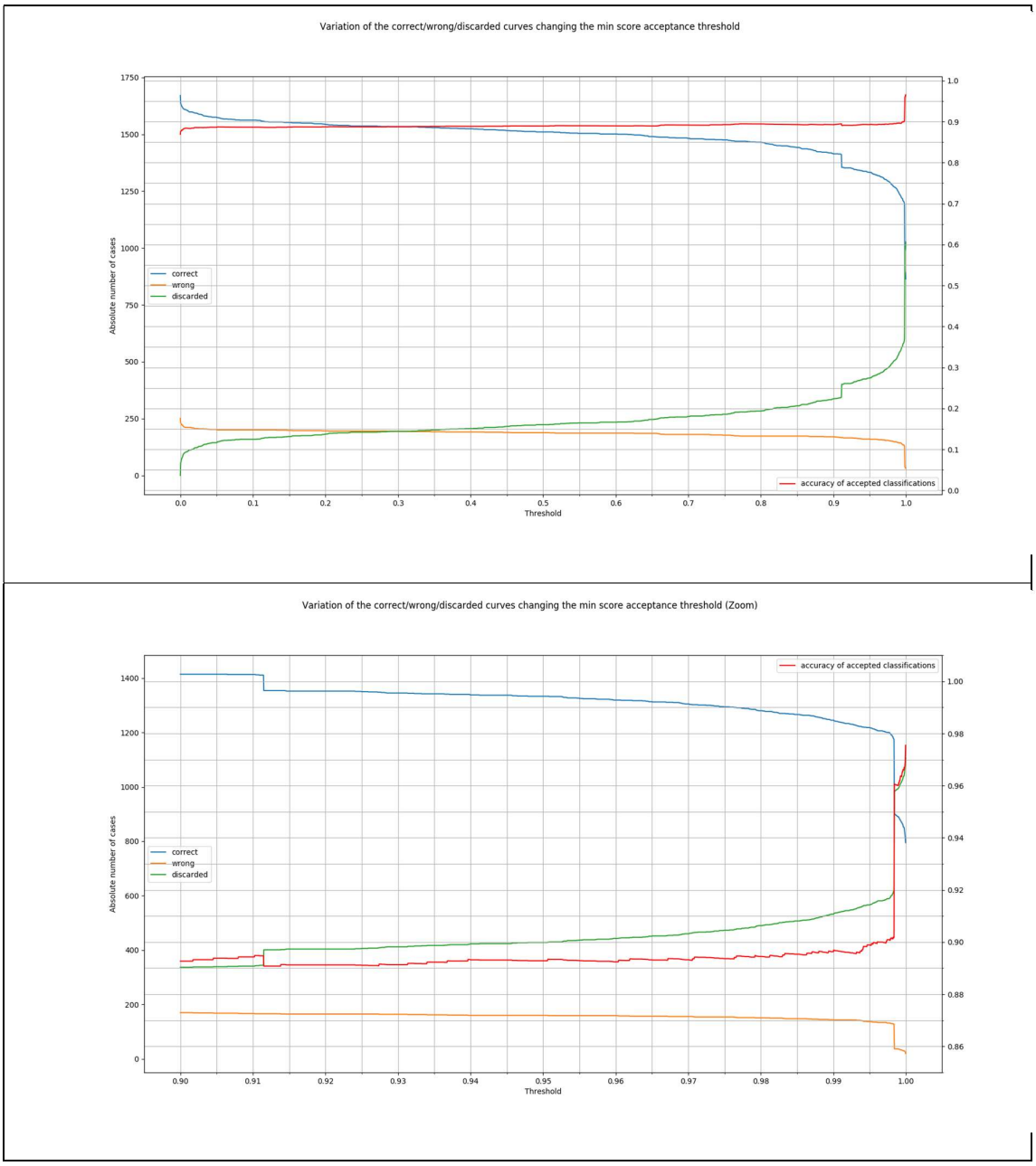


Abbildung 19: Die Änderung der Anzahl von korrekten/inkorrekten Klassifikationen über den möglichen Bereich der Grenzwerte für die akzeptierte Konfidenz. Der untere Plot zeigt eine Ausschnittsvergrößerung.

2.9. Rechenzeitaufwand des Prototpyen

Für die Messung der Laufzeitperformance wurde eine repräsentative Anzahl an Datensätzen aus dem Testdatensatz 2 gerechnet und der Zeitverbrauch gestoppt. Sowohl für die ALPR als auch die Vignetten Detektion wurden Messungen inklusive des Ladens der Bilder durchgeführt. **Diese Messmethode gibt eine realistische Einschätzung der Bearbeitungszeiten pro Datensatz**, ist aber dennoch stark von der tatsächlichen Netzwerk- und Server Konfiguration abhängig.

Laufzeitmessung ALPR

Test Konfiguration: Core i7 Prozessor, 4 Cores @ 4.0 GHz

jpg	*Bildformat:*	1920x1480,	1000	90	sec	=>	**90**	**ms/Bild**
png	*Bildformat:*	2472x3346, 1000 Bilder,		175		=>	**175**	**ms/Bild**

Laufzeitmessung Vignetten Detektion

Vignetten Detektion – gesamte Pipeline mit allen Modulen:

Test Konfiguration 1x 2080 Ti GPU mit 11 Gbyte RAM

Laden + Verarbeiten **ca. 0.95 sec +- 0.42** sek

(reine Verarbeitungszeit ohne Laden ca. **0.79 sek +- 0.38** sek)

Die **Gesamtrechenzeit** für ALPR + DL beträgt im **schlechtesten Fall ca. 1.54 Sekunden** auf einem schnellen PC mit guter GPU**. Pro Tag könnten damit ca. 56.000** Bilder gerechnet werden (bei 24h Verfügbarkeit). Bereits die gewählte HW Konfiguration könnte also theoretisch bereits die anfallende Datenmenge im ASFiNAG Center schon verarbeiten.

2.10. Schlussfolgerungen aus Evaluierung

Als Ergebnis aus der Evaluierung können einige Erkenntnisse sowohl über die Implementierung der Vignetten als auch über Deep Learning für die Detektion gezogen werden.

Das im vorigen Abschnitt gezeigte Histogramm der bei der Evaluierung festgestellten Detektionsfehler zeigt a**ls die größte Fehlerquelle vertikal und horizontal nah beieinander angeordnete Vignetten** (63+11+36, ca. 5,5% aller Vignetten). **Wir sehen diesen Effekt als**

einziges wirkliches Problem im VMI Prototyp an. Es wird durch die spezifische Implementierung des Mask R-CNN Netzwerkes verursacht, welche nah nebeneinander liegende Objekte ausschließt (dies ist eine Eigenschaft des für die Non-Maxima Suppression verwendeten Algorithmus). **Eine Modifikation dieses Sub-Modules ist möglich, sie war jedoch im Rahmen des Projektes zu aufwendig und konnte daher nicht berücksichtigt werden. Eine Behebung bzw. Anpassung des Codes kann vermutlich den größten Teil dieses Problems eliminieren.**

Die zweite größte Fehlerquelle stellen Vignetten auf Bildern mit sehr schlechter Qualität dar (51, ca. 2,6%). **Hier stößt das Mask R-CNN Netzwerk an die Grenzen des derzeit Machbaren**. Vermutlich kann durch weiteres Training ein Teil dieser Fehler behoben werden, jedoch dürfte der Anteil nicht sehr hoch sein, insbesondere, wenn die Anzahl der False Positives gering bleiben soll.

Der Anteil an ‚anderen Objekten', d.h. heißt False Positives bzw. Fehldetektionen beträgt 27 Stück bzw. 1,4%. **Diese Objekte werden durch weitere Abläufe im VMI später mit größter Sicherheit ausgefiltert (T/M Detektion bzw. Jahreszahl) und stellen daher im Endergebnis keine Fehlerquelle mehr dar.**

Das Verfahren der Maskengenerierung und die Feinpositionierung scheint sehr gut zu funktionieren, **nur 11 Vignetten werden diesbezüglich schlecht erkannt**. Gleiches gilt für den Vignetten Typ - **das Mask R-CNN Netzwerk klassifiziert nur drei Vignetten-Typen inkorrekt.** Diese Zahl würde sich vermutlich mit noch mehr Samples für weiteres Training reduzieren lassen.

Die Auswertung der ALPR Daten ergibt, dass **die Carrida Lösung deutlich weniger Fehler beim Lesen zeigt, als die vom AG eingesetzte Lösung**. Die Fehlerquote kann ca. halbiert werden und reduziert dadurch bereits den manuellen Kontrollaufwand erheblich. Dies wird in Zukunft Bedeutung gewinnen, da die Anzahl der digitalen Vignetten weiter steigen wird.

Von allen gesamten Jahres-Vignetten kann der VMI Prototyp ca. 77% vollkommen korrekt verarbeiten, und von den Monats- und Jahresvignetten ca. 73%. **Die Vorgabe von 90% kann damit nicht erreicht werden, jedoch muss diese Auswertung im Kontext der oben genannten erkannten und lösbaren Probleme gesehen werden, als auch in Bezug auf die reale zu analysierende, meist nicht optimale, Bildqualität.**

3. ZUSAMMENFASSUNG

Das Projekt VMI hatte zum Ziel, die automatisierte Detektion und Prüfung von Mautvignetten mittels Deep Learning Methoden unter Verwendung von state-of-the-art Modellen zu implementieren und anhand eines Prototyps zu demonstrieren.

Was kann mit den implementierten Methoden erreicht werden?

Der VMI Software Prototyp und die darunter liegenden Detektionsmodule zeigen, dass Deep Learning Methoden für die Aufgabenstellung im VMI Projekt gut geeignet sind – in Gesamtbetrachtung aller Einzeldetektionen könnten ca. 75% aller Vignetten automatisiert korrekt klassifiziert werden.

Weiters können durch die bessere Lesefähigkeit der Carrida ALPR ca. 50% der Fehler bei den digitalen Vignetten vermieden werden.

Die Hardware Konfiguration für die SW Entwicklung und Tests der Computer Vision Algorithmen besteht aus einem multi-core PC und 2x Nvidia 2080 Ti Graphikkarten. Für die Performance des Systems sind für die VMI Anwendung hauptsächlich die GPUs verantwortlich. **Das Setup könnte bereits die Arbeitslast der derzeit jährlich anfallenden Bildmenge im ASFiNAG Center verarbeiten.** Die Aufteilung der Rechenlast auf mehrere PCs durch Verteilung der Bild-Daten könnte für eine Lastverteilung bzw. Redundanz in der HW umgesetzt werden.

3.1.1. Projektfazit

Das Projekt VMI 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.

Hinsichtlich der Auswertungen und Algorithmen für das Finden von Entscheidungen (bspw. ob eine Detektion akzeptiert werden kann) spiegeln **die Detektionen in schlechten Ausgangsbildern die Grauzonen der Erkennbarkeit wider**. Dies sind vor allem Bildqualitäten, die für das menschliche Auge noch mit akzeptabler Sicherheit erkannt werden

könnten, jedoch niedrige Konfidenzen in den SW Detektoren liefern. Aufgrund des nicht ganz eindeutigen Zusammenhangs zwischen Konfidenzen und tatsächlicher Detektionssicherheit (es gibt nur eine Korrelation) ist **insbesondere bei niedrigen Konfidenzen eine Entscheidung über die Akzeptanz einer Detektion immer mit Unsicherheiten behaftet.** Diese Tatsache erzeugt False Positives bzw. False Negatives.

Die Vorgehensweise bei der Entscheidungsfindung im VMI Projekt schlägt daher die Übergabe der Bilder zur manuellen Kontrolle vor, falls Konfidenzen keine sehr verlässliche Detektion anzeigen.

3.1.2. Weiterer zukünftiger Entwicklungsbedarf

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

- Durchgehende Optimierung der Laufzeiten, insbesondere für den Fall, dass keine GPU verwendet werden soll.
- Bessere Konfigurierbarkeit der Detektoren (v.a. für Laufzeit Optimierung).
- Korrektur des Detektionsproblems nah beieinander liegender Vignetten.
- Verbesserung der Stanzloch Detektion, um auch schwierigere Fälle erkennen zu können. Voraussetzung dafür ist evtl. eine noch bessere Entzerrung der Vignetten bzw. eine noch flexiblere Detektion des Vignetten Randes. Dadurch kann auch auf stärker deformierte Vignetten eine sichere Detektion der Stanzlöcher erreicht werden.
- Weitere größere Kampagne für die Generierung von Samples für das Training aller Detektoren.
- Weitere Anpassung von *Carrida* hinsichtlich systemspezifischer Bildeigenschaften.
- Ein Prozess bzw. Vorgehensweise für das Training kommender Jahreszahlen sollte definiert werden.

LITERATURVERZEICHNIS

[1] DSGVO: Datenschutz-Grundverordnung, Verordnung (EU) 2016/679 des europäischen Parlaments und des Rates vom 27. April 2016.

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

[3] S. Ren, K. He, R. Girschick, J. Sun: „Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks", Conference Proceedings IEEE Pattern Analysis and Machine Intelligence, 2015.

[4] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, C. Fu, A. Berg: „SSD: Single Shot Multibox Detector", European Conference for Computer Vision, 2016.

[5] R. Zhang, „Making Convolutional Networks Shift-Invariant Again", ICML 2019.

[6] N. Dalal, and B. Triggs, "Histograms of Oriented Gradients for Human Detection", Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference, 2005.