

# **Einsatz von Hardware-in-the-loop Systemen zur Evaluation der IT-Sicherheit von visuellen Fahrassistenzsystemen**

## **Bachelorarbeit**

zur Erlangung des Grades eines Bachelor of Science (B.Sc.)  
im Studiengang Informatik

vorgelegt von

**Jan Steffen Jendrny**

Erstgutachter: Prof. Dr. Matthias Thimm  
Artificial Intelligence Group

Betreuer: Dr. Arndt von Twickel  
Bundesamt für Sicherheit in der Informationstechnik

## Erklärung

Ich erkläre, dass ich die Bachelorarbeit selbstständig und ohne unzulässige Inanspruchnahme Dritter verfasst habe. Ich habe dabei nur die angegebenen Quellen und Hilfsmittel verwendet und die aus diesen wörtlich oder sinngemäß entnommenen Stellen als solche kenntlich gemacht. Die Versicherung selbstständiger Arbeit gilt auch für enthaltene Zeichnungen, Skizzen oder graphische Darstellungen. Die Bachelorarbeit wurde bisher in gleicher oder ähnlicher Form weder derselben noch einer anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht. Mit der Abgabe der elektronischen Fassung der endgültigen Version der Bachelorarbeit nehme ich zur Kenntnis, dass diese mit Hilfe eines Plagiatserkennungsdienstes auf enthaltene Plagiate geprüft werden kann und ausschließlich für Prüfungszwecke gespeichert wird.

	Ja	Nein
Mit der Einstellung dieser Arbeit in die Bibliothek bin ich einverstanden.	<input type="checkbox"/>	<input type="checkbox"/>
Der Veröffentlichung dieser Arbeit auf der Webseite des Lehrgebiets Künstliche Intelligenz stimme ich zu.	<input type="checkbox"/>	<input type="checkbox"/>
Der Text dieser Arbeit ist unter einer Creative Commons Lizenz (CC BY-SA 4.0) verfügbar.	<input type="checkbox"/>	<input type="checkbox"/>
Der Quellcode ist unter einer GNU General Public License (GPLv3) verfügbar.	<input type="checkbox"/>	<input type="checkbox"/>
Die erhobenen Daten sind unter einer Creative Commons Lizenz (CC BY-SA 4.0) verfügbar.	<input type="checkbox"/>	<input type="checkbox"/>

.....  
(Ort, Datum)

.....  
(Unterschrift)

## **Zusammenfassung**

Ziel dieser Arbeit ist die Evaluierung visueller Fahrassistenzsysteme mittels eines selbst entwickelten Hardware-in-the-Loop-Systems. Dazu werden exemplarisch Verkehrszeichenerkennungssysteme in drei Versuchsphasen untersucht. Betrachtet werden ein kommerzielles System als Black-Box und verschiedene neuronale Netze zur Bildklassifikation. Zunächst wurde untersucht, wie viele nicht manipulierte Schilder von den Systemen richtig klassifiziert werden. Anschließend wurden den Systemen manipulierte Schilder in Form von adversarialen Angriffen als Eingabe präsentiert und die Klassifikationsergebnisse analysiert. Schließlich wurden Angriffe auf reale Schilder übertragen und den Systemen ebenfalls als Eingabe präsentiert. Es zeigte sich, dass eine große Anzahl von Angriffen getestet und daraus mögliche Angriffsvektoren abgeleitet werden konnten. Aus diesen Ergebnissen konnten konkrete Angriffe berechnet werden, die auch übertragen auf reale Verkehrszeichen zu einer Fehlklassifikation führen. Die Ergebnisse zeigen, dass Hardware-in-the-Loop-Systeme eingesetzt werden können, um visuelle Fahrassistenzsysteme im Hinblick auf ihre IT-Sicherheit zu untersuchen.

## **Abstract**

The aim of this work is the evaluation of visual advanced driver assistance systems by using a self-developed hardware-in-the-loop system. For this purpose, traffic sign recognition systems are exemplarily analysed in three test phases. A commercial system as a black box and various neural networks for image classification are being examined. First, it was investigated how many non-manipulated signs are correctly classified by the systems. Then, manipulated signs in the form of adversarial attacks were presented to the systems as input and the classification results were analysed. Finally, attacks were transferred to real signs and also presented to the systems as input. It turned out that a large number of attacks could be tested and possible attack vectors derived from them. From these results, concrete attacks could be calculated that also lead to misclassification when transferred to real traffic signs. The results show that hardware-in-the-loop-systems can be used to analyse visual advanced driver assistance systems with regard to their IT security.

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>1</b>
<b>2</b>	<b>Theoretische Grundlagen</b>	<b>4</b>
2.1	Fahrassistenzsysteme . . . . .	4
2.1.1	Verkehrsschilddetektion und -klassifikation . . . . .	5
2.2	Hardware-in-the-Loop-Systeme . . . . .	7
2.3	Verfahren aus dem Gebiet der künstlichen Intelligenz . . . . .	9
2.4	Angriffe auf neuronale Netze zur Bildklassifikation . . . . .	14
<b>3</b>	<b>Untersuchungen und Ergebnisse</b>	<b>17</b>
3.1	Versuchsaufbau und Methodik . . . . .	17
3.2	Phase 1: Verkehrsschildklassifikation . . . . .	19
3.3	Phase 2: Simulierte Angriffe . . . . .	22
3.4	Phase 3: Übertragung der Angriffe auf reale Verkehrsschilder . . . . .	27
<b>4</b>	<b>Diskussion</b>	<b>30</b>
4.1	Limitationen . . . . .	33
4.2	Mögliche Verteidigungsmaßnahmen . . . . .	33
<b>5</b>	<b>Fazit und Ausblick</b>	<b>35</b>
<b>6</b>	<b>Anhang</b>	<b>42</b>

## Abbildungsverzeichnis

1	Vergleich von simulierten Umgebungen und den realen Situationen in verschiedenen Szenarios [NS11], welche in einem HIL-System eingesetzt werden können. . . . .	8
2	Abbildung eines dreischichtigen neuronalen Netzes, verändert nach [B <sup>+</sup> 95]. . . . .	11
3	Darstellung einer Einheit einer faltenden Schicht und die Eingabe eines regionalen Bildbereiches [AMAZ17]. . . . .	12
4	Beispiel eines adversarialen Angriffes [GSS]. . . . .	15
5	Schematischer Aufbau der ersten beiden Versuchsphasen . . . . .	17
6	Beispiele zweier Bilder aus dem aufgenommenen Datensatz, um die Unterschiede verschiedener Schilder zu verdeutlichen. . . . .	20
7	Beispiele verschiedener Angriffe . . . . .	23
8	Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Klassifikationssystem und den Durchschnitt aller Systeme. Die Gruppierungen beziehen sich jeweils auf die Kombination der neuronalen Netze, mit denen die Angriffe berechnet wurden, wobei die verschiedenen Anteile veränderter Pixel und die Angriffe auf Basis verschiedener Verkehrszeichen zusammengefasst wurden. . . . .	24
9	Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Klassifikationssystem und den Durchschnitt aller Systeme. Die Gruppierungen beziehen sich jeweils auf die einzelnen Verkehrszeichen, wobei alle Schilder, die ein bestimmtes Zeichen darstellen, und die unterschiedlichen Anteile der veränderten Pixel zusammengefasst wurden. . . . .	25
10	Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Kombination der neuronalen Netze, mit denen die Angriffe berechnet wurden. Die Gruppierungen beziehen sich jeweils auf den Anteil der veränderten Pixel, wobei die einzelnen Verkehrszeichen und Klassifikationssysteme zusammengefasst wurden. . . . .	26
11	Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Verkehrszeichen. Die Gruppierungen beziehen sich jeweils auf den Anteil der veränderten Pixel, wobei die Kombinationen der neuronalen Netze, mit denen der Angriff berechnet wurde, und die Klassifikationssysteme zusammengefasst wurden. . . . .	26

12	Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Klassifikationssystem. Die Gruppierungen beziehen sich jeweils auf den Anteil der veränderten Pixel, wobei die einzelnen Verkehrszeichen und die Kombinationen der neuronalen Netze, mit denen der Angriff berechnet wurde, zusammengefasst wurden. . . . .	27
13	Berechnete Angriffe, bei denen nur 1 % der Pixel manipuliert wurden und die bei allen Systemen zu einer Fehlentscheidung führten. . . . .	28
14	Berechnete Angriffe, die auf reale Schilder übertragen wurden. . . . .	29
15	Hintergrundbilder, in denen in Versuchsphase 1 und 2 die Schilder eingebettet wurden. . . . .	42
16	Schilder, die in Phase 1 von mindestens einem System falsch klassifiziert wurden. . . . .	42

## **Abkürzungsverzeichnis**

<b>ADAS</b>	Advanced Driver Assistance System
<b>BSI</b>	Bundesamt für Sicherheit in der Informationstechnik
<b>CAN</b>	Controller Area Network
<b>CNN</b>	Convolutional Neural Network
<b>CNN 1</b>	Vortrainiertes Netz: ResNext
<b>CNN 2</b>	Vortrainiertes Netz: Inception-v3
<b>CNN 3</b>	Vortrainiertes Netz: MobileNetV2
<b>CNN 4</b>	Selbsttrainiertes Netz ohne Graustufennormierung
<b>CNN 5</b>	Selbsttrainiertes Netz mit Graustufennormierung
<b>KI</b>	Künstliche Intelligenz
<b>LiDAR</b>	Light Detection and Ranging
<b>Radar</b>	Radio Detection and Ranging
<b>RIO</b>	Region Of Interest
<b>FGSM</b>	Fast Gradient Sign Method
<b>HIL</b>	Hardware-in-the-Loop
<b>V 1 - 5</b>	Versuche 1 bis 5, beschrieben in Tabelle 2

# 1 Einleitung

Ab 2022 hat die Europäische Kommission sieben verschiedene Fahrassistenzsysteme für Neuwagen zur Pflicht gemacht. Unter anderem müssen Autos, die neu zugelassen werden, mit einem intelligenten Geschwindigkeitsassistenten ausgestattet sein, der bei Überschreitung der zulässigen Höchstgeschwindigkeit ein Warnsignal gibt [Eur19].

Gleichzeitig können immer mehr Bereiche des Autofahrens automatisiert werden. So ermöglichen beispielsweise neue Technologien der Marke Tesla in bestimmten Situationen ein „automatisches Lenken, Beschleunigen und Bremsen“ ihrer Fahrzeuge [Gmb22]. Beide Technologien haben gemeinsam, dass sie die jeweils gültige Höchstgeschwindigkeit kennen müssen. Dazu werden häufig Verfahren aus dem Gebiet der künstlichen Intelligenz eingesetzt [SB17], um Verkehrsschilder über visuelle Sensoren zu erkennen und zu klassifizieren [Gmb22]. Um ein sicheres Fahren zu gewährleisten, muss sichergestellt werden, dass die Systeme in der Lage sind, die aktuelle Höchstgeschwindigkeit mit sehr hoher Genauigkeit zu bestimmen, damit aus dieser Information die richtigen Entscheidungen getroffen werden können. Die Systeme müssen in der Lage sein, Verkehrszeichen in nahezu jeder Situation, unabhängig von den Witterungsbedingungen, dem Zustand der Verkehrszeichen oder anderen Faktoren, robust und korrekt zu erkennen. Insbesondere muss sichergestellt werden, dass die Entscheidungen der Systeme nicht bewusst von außen manipuliert werden können. Wären Angreifer beispielsweise in der Lage, Verkehrszeichen zu verändern und damit eine falsche Klassifizierung herbeizuführen, könnte dies zu Unfällen führen. Je mehr Entscheidungen die Fahrzeugsysteme automatisiert treffen, desto größer wird die Gefahr durch Manipulation. Daher ist die Untersuchung von Fahrassistenzsystemen aus Sicht der IT-Sicherheit von hoher Relevanz.

## Zielsetzung

Diese Arbeit befasst sich mit der Untersuchung von Fahrerassistenzsystemen, da diese für die Sicherheit von Personen im Straßenverkehr von großer Bedeutung sind bzw. es vermutlich in Zukunft sein werden. Dabei soll ein Hardware-in-the-Loop-System eingesetzt werden, um exemplarisch die IT-Sicherheit von visuellen Fahrassistenzsystemen zu evaluieren.

Ziel dieser Forschung ist es, durch den Einsatz eines Hardware-in-the-Loop Systems Angriffe auf visuelle Fahrassistenzsysteme zu testen und herauszufinden, ob sich daraus reale Angriffsvektoren ableiten lassen. Dadurch sollen Rückschlüsse auf die Angreifbarkeit der Systeme gezogen werden. Diese Rückschlüsse könnten dann in weiteren Arbeiten zur Entwicklung von Abwehrmaßnahmen beitragen.

## Methodik

Anhand verschiedener Experimente sollen exemplarisch Verkehrszeichenerkennungssysteme angegriffen und die Auswirkungen dieser Angriffe untersucht werden.



Zunächst wird geprüft, ob Hardware-in-the-Loop-Systeme geeignet sind, die Sicherheit der Systeme zu bewerten. Dazu werden nicht manipulierte Verkehrsschilder durch das System klassifiziert. Anschließend werden verschiedene Angriffe mit dem Hardware-in-the-Loop-System simuliert und versucht, Rückschlüsse auf die sicherheitsrelevanten Eigenschaften der Systeme zu ziehen. Abschließend werden die Angriffe auf reale Verkehrszeichen übertragen, um die Eignung der Angriffe in realen Situationen beurteilen zu können.

## **Aufbau der Arbeit**

Dazu werden zunächst die theoretischen Grundlagen erläutert: Zunächst wird aufgezeigt, was Fahrassistenzsysteme sind, in welchen Bereichen sie eingesetzt werden und wie sie eingeordnet werden können. Anschließend wird die Funktionsweise von Verkehrszeichenerkennungssystemen erläutert und die beiden Komponenten Verkehrszeichendetektion und -klassifikation vorgestellt.

Im nächsten Schritt wird gezeigt, wie solche Systeme in einer simulierten Umgebung getestet werden können. Dazu werden Hardware-in-the-Loop-Systeme vorgestellt und erläutert, warum sie für den Test von Fahrzeugkomponenten geeignet sind.

Zum Abschluss der theoretischen Grundlagen wird dargestellt, was künstliche Intelligenz ist, welche Technologien zu ihrer Implementierung verwendet werden, wie sie sich für die Bildklassifikation eignet und welche Angriffe zur Manipulation von Klassifikationsergebnissen eingesetzt werden können.

Im Anschluss an dieses Kapitel werden die durchgeführten Untersuchungen vorgestellt und die Ergebnisse präsentiert. Zunächst wird der Versuchsaufbau beschrieben und die einzelnen Experimente im Detail vorgestellt. Es wird beschrieben, welche Techniken verwendet werden und wie diese dazu beitragen, Schlussfolgerungen aus den Experimenten zu ziehen.

Anschließend werden jeweils die Ergebnisse der einzelnen Experimente im Detail dargestellt. Begonnen wird mit den Ergebnissen der Klassifizierungen von nicht manipulierten Schildern. Anschließend werden die Ergebnisse der Angriffe in der simulierten Umgebung vorgestellt, wobei verschiedene Aspekte der Experimente miteinander verknüpft werden. Zuletzt werden die realen, manipulierten Schilder und die Klassifizierungsergebnisse dieser Schilder präsentiert.

Nach der Präsentation der Ergebnisse werden diese diskutiert. Dabei sollen die wichtigsten Erkenntnisse dargestellt und hinterfragt werden, ob das Ziel dieser Arbeit erreicht wurde. Außerdem werden die Grenzen der Untersuchungen aufgezeigt und mögliche Abwehrmaßnahmen skizziert.

Abschließend wird ein Fazit gezogen und ein Ausblick auf weitere Forschungsarbeiten gegeben.

Die Untersuchungen wurden in Zusammenarbeit mit dem Referat *DI 11 - Bewertungsverfahren für eID-Technologien in der Digitalisierung* des Bundesamtes für Sicherheit in der Informationstechnik (BSI) durchgeführt.

## Verwandte Arbeiten und Abgrenzung

Der Bereich der künstlichen Intelligenz und deren Sicherheit sind Forschungsgebiete, die z.B. von Barreno et al. in [BNJT10] und [BNS<sup>+</sup>06] behandelt werden. Aufgrund der Verfügbarkeit großer Datensätze mit Bildern von Verkehrszeichen [SSSI11] sind Systeme zur Detektion und Klassifikation von Verkehrszeichen gut untersucht. Eine wichtige Rolle spielen dabei Untersuchungen zu adversarialen Angriffen ([GSS], [SZS<sup>+</sup>], [PMJ<sup>+</sup>16]), die unter anderem als Angriffe auf Bildklassifikationssysteme eingesetzt werden.

Es wird außerdem untersucht, ob erfolgreiche Angriffe in verschiedenen Szenarien berechnet werden können. Angriffe mit bekannter Architektur und bekannten Trainingsdaten werden unter anderem in [GSS] betrachtet. Die Angreifbarkeit von Systemen mit unbekannter Architektur wird unter anderem in [PMG<sup>+</sup>17] und [PMG] untersucht. Verschiedene Arbeiten befassen sich zudem mit Angriffen unterschiedlicher Komplexität. In [PMJ<sup>+</sup>16] werden Angriffe betrachtet, die versuchen, durch Manipulation einer bestimmten Eingabe eine bestimmte Klassifikation zu erzielen. In anderen Arbeiten ([GSS], [SZS<sup>+</sup>]) wird lediglich versucht, eine ungezielte Fehlentscheidung herbeizuführen.

Unabhängig von der Untersuchung von Verfahren aus dem Bereich der künstlichen Intelligenz werden so genannte Hardware-in-the-Loop-Systeme entwickelt, um den Test von Fahrzeugkomponenten zu vereinfachen [Bac05]. Darüber hinaus wird die Eignung dieser Systeme für den Funktionstest visueller Komponenten untersucht [NS11].

Diese Arbeit versucht Aspekte aus beiden Bereichen zu kombinieren. Es wird ein Hardware-in-the-Loop-System entwickelt, mit dem visuelle Fahrassistenzsysteme auf ihre Angreifbarkeit getestet werden können. Um dessen Eignung zu untersuchen, konzentriert sich diese Arbeit auf die Untersuchung von Klassifikationssystemen. Zur Reduzierung des Entwicklungsaufwandes wird auf die Implementierung eines Detektionssystems verzichtet. Außerdem werden aufgrund der Häufigkeit von Geschwindigkeitsbegrenzungsschildern und ihrer kritischen Bedeutung für den Ausgang eines möglichen Unfalls in dieser Arbeit ausschließlich solche Schilder betrachtet. Um die Komplexität der Angriffe zu reduzieren, konzentriert sich diese Arbeit auf die Untersuchung ungezielter Angriffe.

## 2 Theoretische Grundlagen

### 2.1 Fahrassistenzsysteme

Heutige Fahrzeuge werden zunehmend mit digitalen Fahrassistenzsystemen ausgestattet. Gerade für das autonome oder teilautonome Fahren werden diese Systeme benötigt. Ein Fahrassistenzsystem soll den Fahrenden beim Fahren unterstützen, indem es ihn bei Entscheidungen unterstützt, Signale in möglichen Gefahrensituationen gibt oder auch gegensteuernde Maßnahmen ergreift [GLSG09].

Bereits in den 1970er Jahren wurden die ersten Assistenzsysteme entwickelt, die vor allem sicherheitsrelevante Funktionen hatten. Eines der ersten Systeme war das Antiblockiersystem von Bosch [BDF<sup>+</sup>14], das 1978 auf den Markt kam. Im Jahr 1995 wurde dann das elektronische Stabilitätsprogramm entwickelt, das im Wesentlichen auf den aufgezeichneten Daten eines elektrischen Gyroskops basiert. Heutige Systeme können komplexere Entscheidungen treffen und werden daher häufig als *Advanced Driver Assistance Systems* (ADAS) [GLSG09] bezeichnet. Durch diese Systeme können bereits Teilbereiche des Fahrens automatisiert werden. Zur Kategorisierung des autonomen Fahrens wird häufig der Standard SAE J3016 [SAE21] der Society of Automotive Engineering verwendet. Dieser Standard beschreibt fünf Stufen, auch Level genannt, wobei ab Level 1 von Automatisierung gesprochen wird.

- Level 0: Abgesehen von Sicherheitsfunktionen gibt es keine Assistenzsysteme, die aktiv in das Fahrgeschehen eingreifen.
- Level 1: Assistenzsysteme übernehmen in bestimmten Situationen entweder die Lenkung oder die Beschleunigung.
- Level 2: Das Fahrzeug übernimmt in bestimmten Situationen sowohl die Lenkung als auch die Beschleunigung.
- Level 3: Das Fahrzeug kann in bestimmten Situationen autonom fahren. Der Fahrer muss auf ein Signal hin die Kontrolle über das Fahrzeug übernehmen.
- Level 4: Das System übernimmt das Fahren in bestimmten Situationen ohne Eingreifen des Fahrers.
- Level 5: Das Fahrzeug übernimmt die Steuerung in jeder Situation.

Mit steigendem Level übernimmt das Gesamtsystem immer mehr Elemente der Fahrt, bis das Fahrzeug in Level 5 vollständig autonom fährt. Ebenso müssen die Systeme mit steigendem Automatisierungsgrad mehr Aufgaben übernehmen und mit einer komplexeren Abbildung der Umwelt umgehen. Der Einsatz solcher Fahrerassistenzsysteme und die Erhöhung des Automatisierungsgrades können die Unfallzahlen reduzieren [AO03]. Voraussetzung dafür ist allerdings, dass die Systeme robust auf unterschiedlichste Situationen reagieren müssen. Je höher der Automatisierungsgrad, desto wichtiger ist es, dass die Systeme keine oder möglichst wenige Fehlentscheidungen treffen.

Ab 2022 müssen Neuwagen, die in der Europäischen Union zugelassen werden, mit sieben speziellen Fahrerassistenzsystemen ausgestattet sein [Eur19]:

- Intelligenter Geschwindigkeitsassistent
- Vorrichtung zum Einbau einer alkoholempfindlichen Wegfahrsperr
- Warnsystem bei Müdigkeit und nachlassender Aufmerksamkeit des Fahrers
- Hochentwickeltes Warnsystem bei nachlassender Konzentration des Fahrers
- Notbremslicht
- Rückfahrassistent
- Ereignisbezogene Datenaufzeichnung

Einige dieser Systeme müssen die Umgebung wahrnehmen und interpretieren. Dazu werden verschiedene Sensoren eingesetzt, um automatisierte Entscheidungen treffen zu können. Beispielsweise werden beim Intelligenten Geschwindigkeitsassistenten häufig optische Sensoren, LiDAR- und Radartechniken eingesetzt [BDF<sup>+</sup>14]. Hier dominieren Kameras und Radartechniken als eingesetzte Sensorik [BDF<sup>+</sup>14]. Diese Arbeit beschränkt sich auf Fahrerassistenzsysteme, die optische/visuelle Sensoren verwenden. Dies hat im Wesentlichen zwei Gründe:

- Optische Systeme sind von großer Bedeutung. Beispielsweise ersetzt Tesla in seinen Fahrzeugen alle Radarsensoren durch hochauflösende Kameras [Tes22].
- Durch die zweidimensionale Abbildung der Umgebung durch die optischen Sensoren können Simulationen ebenfalls zweidimensional dargestellt werden und sind somit einfach realisierbar.

Optische Sensoren haben den Vorteil, dass sie für verschiedene Aufgaben eingesetzt werden können, wie z.B. die Kollisionsvermeidung [GLSG09] oder die Erkennung und Klassifikation von Verkehrszeichen [dLMSA97].

### **2.1.1 Verkehrsschilddetektion und -klassifikation**

Wie bereits erwähnt, ist ein von der Europäischen Union ab 2022 vorgeschriebenes Fahrerassistenzsystem der „intelligente Geschwindigkeitsassistent“ [Eur19]. Dieser soll das geltende Tempolimit erkennen und den Fahrer bei Überschreitung warnen. Dabei sollen „die Leistungsanforderungen [...] so konfiguriert sein, dass die Fehlerquote im realen Fahrbetrieb null oder möglichst niedrig ist.“ [Eur19] Um diese Anforderung erfüllen zu können, müssen die Systeme Geschwindigkeitsbeschränkungen für die aktuelle Position kennen. Dies kann unter anderem durch die Detektion und Klassifikation von Verkehrszeichen mittels Kameras erreicht werden.

Bei dieser Aufgabe treten jedoch verschiedene Schwierigkeiten auf, wie z.B. der Umgang mit natürlichem Licht [HSS<sup>+</sup>13]. Außerdem sind die Schilder so gestaltet, dass sie von Menschen leicht erkannt werden [SSSI11]. Für Computer ist dies jedoch aufgrund der Komplexität der vielen Verkehrszeichen und der vielen Umgebungsbedingungen eine schwer zu lösende Aufgabe [HSS<sup>+</sup>13]. Aus diesem Grund beschränken sich viele Untersuchungen auf eine bestimmte Anzahl von Verkehrszeichen oder auf bestimmte Kategorien wie Geschwindigkeitsbegrenzungen oder Gefahrenzeichen [HSS<sup>+</sup>13].

Durch die von der Umgebung abhebende Gestaltung der Schilder können jedoch auch Algorithmen zur Erkennung implementiert werden. Dies gilt insbesondere für Gefahrenzeichen und Geschwindigkeitsbegrenzungen. Blaue Vorschriftzeichen, wie z.B. eine vorgeschriebene Fahrtrichtung, werden von Detektionssystemen oft mit geringerer Wahrscheinlichkeit erkannt [HSS<sup>+</sup>13].

Da für diese Aufgabe häufig Kameras eingesetzt werden, besteht eine Schwierigkeit darin, aus der großen Datenmenge die relevanten Daten herauszufiltern. Die Systeme sind daher in der Regel so aufgebaut, dass ein Detektionssystem die Bildbereiche mit den Schildern herausfiltert und an ein Klassifikationssystem weiterleitet [dLMSA97]. Diese Bereiche werden häufig als *Region of Interest (ROI)* bezeichnet. Dazu können verschiedene Methoden verwendet werden. Saadna und Behloul [SB17] unterscheiden drei Kategorien:

- farbbasierte Methoden
- kantenbasierte Methoden
- lernbasierte Methoden

Farbbasierte Verfahren versuchen, die relevanten Regionen des Bildes anhand der aufgenommenen Farbinformationen zu bestimmen. Die standardisierten Farben von Verkehrszeichen helfen bei der Entwicklung eines Detektionsalgorithmus. Allerdings können die Farbwerte durch unterschiedlichen Lichteinfall stark variieren, weshalb bestimmte Farbräume wie HSI verwendet werden, um diesen Einfluss zu reduzieren [SB17].

Die kantenbasierten Verfahren nutzen die standardisierte Form der Zeichen zur Bestimmung der ROI. Änderungen in den Farbwerten sind bei dieser Methode oft von geringer Relevanz, da die Bilder meist in Graustufen umgewandelt werden. Dadurch sind diese Methoden zwar robuster gegenüber wechselnden Lichtverhältnissen, jedoch kann hier beispielsweise eine Verdeckung durch Laub zu Fehlentscheidungen führen [SB17].

Lernbasierte Verfahren versuchen, aus großen Datenmengen abzuleiten, wie die relevanten Regionen eines Bildes erkannt werden können. Dazu werden häufig mehrdimensionale Eingaben verwendet, wie z.B. Bilder, bei denen Farbwerte, Höhe und Breite jeweils als eine Dimension interpretiert werden. Zur Berechnung der Ergebnisse können verschiedene mathematische Modelle verwendet werden. Auch wenn

diese Methoden oft gute Ergebnisse liefern, ist die Berechnung sehr ressourcenintensiv [SB17].

Detektionssysteme können, wie z.B. bei de la Escalera et al. [dLMSA97], ausschließlich eine dieser Methoden oder auch Mischformen verwenden.

Die Güte eines Systems kann vor allem durch zwei Faktoren messbar gemacht werden, um damit den Einsatz als Realsystem zu überprüfen [SB17]. Die Systeme müssen eine hohe Genauigkeit aufweisen, d.h. sie müssen relevante Regionen korrekt erkennen und irrelevante Regionen korrekt herausfiltern. Außerdem müssen die ROI in möglichst kurzer Zeit erkannt werden, um als Echtzeitsystem im Fahrzeug eingesetzt werden zu können. Häufig gibt es einen Kompromiss zwischen diesen beiden Aspekten, da Systeme, die wenig Fehlentscheidungen treffen, ihre Ergebnisse oft in längerer Zeit berechnen [HSS<sup>+</sup>13]. Dies lässt Raum für Angriffe, da die Systeme bisher keine Garantie für eine korrekte Erkennung bieten und möglicherweise auch Bildausschnitte als relevant markieren, die keine echten Schilder sind (z.B. selbst gemalte Verkehrszeichen an Hauswänden) oder manipulierte Schilder möglicherweise nicht als relevant einstufen.

Damit Klassifikationssysteme Zeichen korrekt bestimmen können, dürfen keine Schilder fälschlicherweise herausgefiltert werden. Aus diesem Grund werden oft zu viele ROIs falsch positiv detektiert und an die Klassifikationssysteme weitergegeben [SB17].

Die Klassifikationssysteme arbeiten daher mit den Daten der vermutlich relevanten Regionen des Bildes, da nicht garantiert werden kann, dass die Detektion in jedem Fall korrekte Ergebnisse liefert. Die Systeme müssen dann aus den Informationen eine bestimmte Klasse ableiten, wobei die Klassen unterschiedlich häufig auftreten [SSSI11]. Die Klassen repräsentieren in diesem Fall das Verkehrszeichen, z.B. ein Tempolimit von 30 km/h.

Häufig werden die übergebenen Bildbereiche zunächst normalisiert, um die spätere Klassifikation zu vereinfachen [SSSI11]. Dabei wird z.B. die Bildgröße verändert [SSSI11], Sättigung und Kontrast angepasst [SSSI11] oder Farbwerte in Graustufen umgewandelt [CMMS11].

Anschließend wird aus den Bildinformationen eine Klasse berechnet, wobei verschiedene Methoden zur Anwendung kommen können, welche im Kapitel 2.3 näher erläutert werden.

Die berechnete Klasse repräsentiert das Verkehrszeichen, z.B. eine Geschwindigkeitsbeschränkung von 30 km/h. Aus dieser Information können dann weitere Schritte abgeleitet werden. Im Falle der EU-Verordnung muss das Fahrzeug z.B. die aktuelle Geschwindigkeit mit der zulässigen Höchstgeschwindigkeit vergleichen und bei Überschreitung eine Warnung ausgeben.

## **2.2 Hardware-in-the-Loop-Systeme**

Mit der zunehmenden Komplexität der in Fahrzeugen eingesetzten Systeme wird auch die Prüfung dieser Systeme immer aufwendiger [NS11]. Da diese häufig in

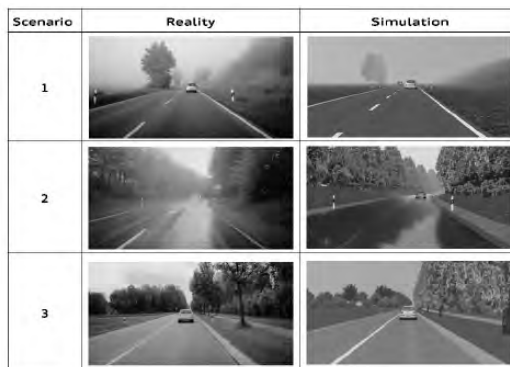


Abbildung 1: Vergleich von simulierten Umgebungen und den realen Situationen in verschiedenen Szenarios [NS11], welche in einem HIL-System eingesetzt werden können.

realen Fahrsituationen getestet werden, ist es sehr aufwendig, viele verschiedene Fälle in kurzer Zeit zu testen.

Aus diesem Grund werden viele digitale Fahrzeugkomponenten in einem *Hardware-in-the-Loop-System* (HIL-System) getestet [Bac05]. Die Grundidee dieser Systeme ist es, spezielle Fahrzeugkomponenten in eine Simulation zu integrieren und so reale Ausgaben der Systeme in einer simulierten Umgebung zu erzeugen.

Für visuelle Fahrassistenzsysteme werden videobasierte Umgebungsmodelle entwickelt, um möglichst viele Testfälle abzudecken [NS11]. Die Modelle werden entweder über ein Display, das parallel zum Kamerasensor ausgerichtet ist, dargestellt oder direkt über die Steuereinheit der Kamera an die nachfolgenden Systemteile übertragen.

Diese Vorgehensweise hat den Vorteil, dass sowohl qualitativ als auch quantitativ mehr und gezielter getestet werden kann [NS11]. Beispielsweise können bei der Verkehrszeichenerkennung unterschiedliche Verkehrszeichen unter verschiedenen simulierten Umweltbedingungen getestet werden. Auf einer realen Teststrecke müssten verschiedene Jahreszeiten abgewartet werden, um alle Witterungsbedingungen adäquat testen zu können. Außerdem kann eine wesentlich größere Anzahl von Schildern getestet werden, da die Schilderdichte in der Simulation die Schilderdichte auf einer realen Straße deutlich übersteigt.

Abbildung 1 zeigt Beispiele verschiedener Simulationen. Es wird deutlich, dass Simulationen nur einen Teil der Realität abbilden können. Beispielsweise sind die Darstellungen von Bäumen, von Spiegelungen oder vom Nebel deutlich weniger komplex als auf den dargestellten Fotografien. Werden jedoch Tests in realen Szenarien durchgeführt und später in der Simulation dieses Szenarios wiederholt, können ähnliche Ergebnisse erzielt werden [NS11]. Aus diesem Grund können HIL-Systeme für die Erprobung von Fahrassistenzsystemen auf der Basis visueller Informationen geeignet sein.

## 2.3 Verfahren aus dem Gebiet der künstlichen Intelligenz

Künstliche Intelligenz (KI) ist ein interdisziplinäres Gebiet, das sich mit der Entwicklung von Computersystemen befasst, die menschliches Denken und Verhalten nachahmen können. Dies kann in Form von einfachen Aufgaben wie dem Erkennen von Mustern in Daten oder komplexen Aufgaben wie dem Lösen von Klassifikationsproblemen oder dem Treffen von Entscheidungen geschehen. Häufig sollen Vorhersagen durch KI-Systeme berechnet werden [THBM19].

Die Geschichte der KI reicht weit in die Vergangenheit zurück. Ein erster Meilenstein war, als der Computerwissenschaftler Alan Turing die Frage aufwarf, ob Maschinen in der Lage sein könnten, menschliche Intelligenz nachzuahmen [Tur09]. Seitdem hat sich die KI-Forschung - trotz einiger Tiefpunkte im sogenannten KI-Winter [HK19] - kontinuierlich weiterentwickelt und zu erheblichen technologischen Fortschritten geführt.

Die Entwicklung der KI hat in den letzten Jahren enorme Fortschritte gemacht, insbesondere durch den Einsatz des maschinellen Lernens, einer Technik, die es Computersystemen ermöglicht, aus großen Datenmengen zu lernen und darin Muster zu erkennen [GBC16]. Diese Fortschritte haben zu einer Vielzahl von Anwendungen in Bereichen wie dem Gesundheitswesen [HT17] und dem Transportwesen [BDF<sup>+</sup>14] geführt.

Die ersten KI-Systeme zeichneten sich durch eine implementierte Wissensbasis aus. Die häufig durch formale Sprachen definierten Programme werden als wissensbasierte Systeme bezeichnet [GBC16]. Die Schwierigkeit bei der Implementierung besteht darin, dass die Wissensbasis bekannt und formal beschreibbar sein muss. Um diese Schwierigkeit zu umgehen, wurden Systeme entwickelt, die aus einer Menge von Daten eine eigene Wissensbasis ableiten. Dies wird als maschinelles Lernen bezeichnet und kann durch verschiedene mathematische Modelle erreicht werden.

Maschinelles Lernen lässt sich im Wesentlichen in drei verschiedene Kategorien einteilen [THBM19]:

- überwachtes Lernen (supervised learning)
- unüberwachtes Lernen (unsupervised learning)
- verstärkendes Lernen (reinforcement learning)

Für alle Arten gibt es zwei grundlegende Phasen, die Trainingsphase und die Testphase [THBM19]. In operativen Systemen kommen weitere Phasen hinzu (z.B. Planung, Datengenerierung, produktiver Einsatz). Diese Phasen laufen häufig nicht sequentiell ab, sondern es kann iterativ zu früheren Phasen zurückgesprungen werden, z.B. wenn weitere Daten in den Trainingsdatensatz integriert werden.

Im Folgenden werden jedoch nur die Trainings- und Testphase betrachtet. In der Trainingsphase wird aus einer Vielzahl von Daten ein Modell berechnet. Dieses Modell wird dann in der Testphase zur Entscheidungsfindung verwendet. Die Arten



des maschinellen Lernens lassen sich im Wesentlichen durch die Unterschiede in der Trainingsphase beschreiben. Beim überwachten Lernen werden die Trainingsdaten vorklassifiziert und das System anhand dieser Informationen trainiert. Eine einfache Methode ist die logistische Regression, die häufig verwendet wird, um Daten in eine von zwei Klassen einzuteilen [GBC16]. Für komplexere Räume, d.h. wenn es mehr Klassen oder mehr Parameter in den Daten gibt, werden z.B. neuronale Netze verwendet [SSSI11], die später noch genauer erläutert werden.

Beim unüberwachten Lernen werden die Daten nicht vorklassifiziert. Die Algorithmen erkennen aus den gegebenen Daten Gemeinsamkeiten und können daraus Cluster bilden [THBM19].

Beim Reinforcement Learning wird versucht, den Erfolg einer Handlung zu maximieren [THBM19]. Dabei wird versucht, durch Ausprobieren und probabilistische Schätzungen viele Daten zu generieren und daraus die optimalen Aktionen zu lernen [THBM19].

Der Schwerpunkt dieser Arbeit liegt auf neuronalen Netzen und damit auf dem überwachten Lernen, da diese Methoden häufig zur Klassifizierung von Verkehrszeichen eingesetzt werden [SSSI11]. Eine Bildklassifikation kann nach Fong und Vedaldi [FV19] durch die Funktion  $f : X \rightarrow Y$  beschrieben werden. Dabei ist  $x \in X \subset \mathbb{R}^{H \times W \times 3}$  die Eingabe und  $y \in Y \subset [0, 1]^n$  die Ausgabe des Systems. Die Eingabe ist ein Bild mit der Höhe  $H$ , der Breite  $W$  und den drei Farbwerten rot, grün und blau. Die Ausgabe des neuronalen Netzes ist eine Wahrscheinlichkeit für jede der  $n$  Klassen.

Ein neuronales Netz wird nun verwendet, um eine approximative Lösung für diese Funktion zu finden. Die Inspiration für neuronale Netze stammt von biologischen neuronalen Netzen [Bis94], von denen einige Grundprinzipien übernommen wurden. Ein *Neuron*, oder bei künstlichen neuronalen Netzen auch *Einheit* genannt, ist dabei eine nichtlineare Funktion, die für eine Menge von Eingangsvariablen  $x_i, i = 1, \dots, d$  eine Ausgabe  $z$  liefert. Die Ausgabe wird durch

$$z = S_c\left(\sum_{i=1}^d w_i x_i + w_0\right)$$

bestimmt [B<sup>+</sup>95]. Mit dem *Gewicht*  $w_i$  können einzelne Eingangsgrößen der jeweiligen Einheit stärker in die Berechnung einbezogen werden. Ebenso kann mit dem *Bias*  $w_0$  ein Schwellenwert gesetzt werden, so dass nur große Werte einen Einfluss auf die Berechnung haben. Die Ausgabe wird schließlich durch die *Aktivierungsfunktion*  $S_c()$  berechnet.

Als Implementierung der Aktivierungsfunktion eignen sich verschiedene Funktionen [KO11], wie z.B. die Sigmoidfunktion:

$$\text{sig}(t) = \frac{1}{1 + e^{-t}}$$

oder die Rectifier-Funktion, welche in einer sogenannten *rectified linear unit* (ReLU) [GBC16] genutzt wird:

$$\varphi(t) = \max(0, t)$$

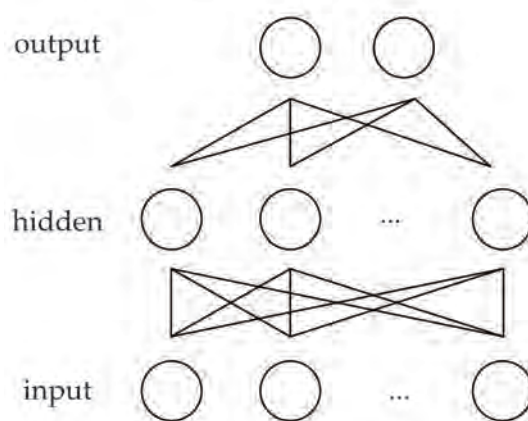


Abbildung 2: Abbildung eines dreischichtigen neuronalen Netzes, verändert nach [B<sup>+</sup>95].

Aufgrund der Aktivierungsfunktionen entsprechen die Einheiten nichtlinearen Funktionen, was den Vorteil hat, dass die Verknüpfung dieser Funktionen wiederum nichtlinear ist. Andernfalls könnte ein neuronales Netz nur Approximationen für lineare Probleme finden [GBC16].

In künstlichen neuronalen Netzen werden die Einheiten meist in Schichten aufgebaut, wobei jede Schicht eine unterschiedliche Anzahl von Einheiten enthalten kann. Die Ausgänge der Einheiten werden als Eingänge in der nächsten Schicht verwendet. Ein dreischichtiges Netz kann als Funktion  $f(x) = f_3(f_2(f_1(x)))$  interpretiert werden [GBC16]. In diesem Fall ist  $f_1$  die erste Schicht, auch *Eingangsschicht* (input layer) genannt. Dagegen ist  $f_3$  die dritte Schicht, die als *Ausgangsschicht* (output layer) bezeichnet wird. Die inneren Schichten werden als *verdeckte Schichten* (hidden layers) beschrieben. Durch die tiefe Verkettung der Schichten werden die Netze als *tiefe neuronale Netze* bezeichnet, weshalb sich auch der Begriff *Deep Learning* entwickelt hat [GBC16]. Wie in Abbildung 2 dargestellt, sind alle Einheiten einer Schicht mit den Einheiten der vorhergehenden und der nachfolgenden Schicht verbunden. Das Netz hat für jede Eingangsvariable eine Einheit in der Eingangsschicht, und die Ausgangsschicht hat je nach Aufgabe eine unterschiedliche Anzahl von Einheiten. Bei einer Bildklassifikation hat also die Eingangsschicht  $H \times W \times 3$  Einheiten und die Ausgangsschicht  $n$  Einheiten. Das bedeutet, dass sich bei einem Bild mit  $64 \times 64$  Pixeln bereits 12.228 Einheiten in der ersten Schicht befinden.

Jede Einheit berechnet anhand der Eingabe die Ergebnisse, die dann jeweils an die nächste Schicht propagiert werden, bis die Ausgabeschicht die Ausgabevariablen zurückgibt. In der Trainingsphase werden die Ergebnisse ebenfalls berechnet und dann der Fehler mit Hilfe einer Fehlerfunktion quantifiziert [Bis94]. Die Fehlerfunktion macht sich zunutze, dass durch die Vorklassifikation das korrekte Ergebnis bekannt ist, anhand dessen die Abweichung zum berechneten Ergebnis be-

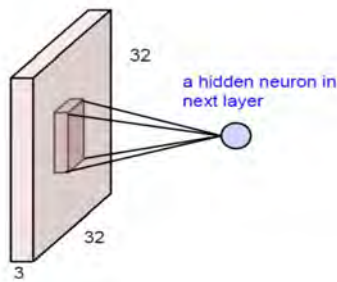


Abbildung 3: Darstellung einer Einheit einer faltenden Schicht und die Eingabe eines regionalen Bildbereiches [AMAZ17].

stimmt werden kann. Während des Trainings wird versucht, diese Abweichung zu minimieren, indem die Gewichte der Einheiten verändert werden. Zu Beginn des Trainings werden die Gewichte zufällig initialisiert und während des Trainings verändert. Die Fehler werden gewichtet zurück propagiert und mit dieser Information können die Gewichte angepasst werden. Die einfachste Technik zur Optimierung der Gewichte ist die Anpassung in Richtung des negativen Gradienten  $\Delta E(w)$ , wobei  $E(w)$  die gewichtete Fehlerfunktion ist [Bis94]. Dies wird als *Verfahren des steils-ten Abstiegs* (gradient descent) bezeichnet [Bis94].

Im Laufe der Zeit haben *convolutional neural networks* (CNN) im Vergleich zu klassischen neuronalen Netzen in einigen Bereichen, wie z.B. der Bildklassifikation, bessere Ergebnisse erzielt [GWK<sup>+</sup>18]. Eine Schwierigkeit klassischer neuronaler Netze ist, dass sie sehr rechenintensiv sind, was besonders bei komplexen Problemen zu langen Rechenzeiten führt [AMAZ17]. Um die Laufzeit zu verbessern, sind CNNs anders aufgebaut und haben vor allem drei Arten von Schichten [GDLG17]:

- faltende Schichten (convolutional layers)
- pooling Schichten (pooling layers)
- vollständig verbundene Schichten (fully connected layers)

Die faltende Schicht ist die Hauptschicht eines CNN [GDLG17], dargestellt in Abbildung 3 nach Albawi et. al. [AMAZ17]. Um die Parameter zu reduzieren, sind die Einheiten der Schicht nicht mit allen Einheiten der vorhergehenden Schicht verbunden, sondern haben nur die Werte eines regionalen Bereichs als Eingabe, wie in Abbildung 3 dargestellt.

Die konkrete Einheit wird durch Faltung der Werte des jeweiligen Bildbereichs berechnet. In der Praxis wird jedoch häufig keine Faltung, sondern eine ähnliche Funktion, die Kreuzkorrelation, verwendet [GBC16]:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

Dabei ist  $I$  die Eingangsmatrix, d.h. der regionale Bildbereich, und  $K$  der *Kernel*, eine Matrix, die häufig als *Filter* bezeichnet wird. Es folgen einige Beispiele für solche Filtermatrizen [AMAZ17]:

- Identität:  $\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$
- Kantensfilter:  $\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$
- Schärfungsfilter:  $\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$

Außerdem haben alle Einheiten in einer Schicht das gleiche Gewicht, was die Anzahl der Parameter ebenfalls reduziert.

Pooling-Schichten werden verwendet, um die Anzahl der Einheiten zu reduzieren, indem jeweils ein einzelner Wert aus Teilbereichen der vorherigen Schicht berechnet wird [AMAZ17]. Häufig wird das so genannte *max-pooling* verwendet, bei dem aus einem Bereich (häufig  $2 \times 2$ ) das Maximum berechnet wird, wie in diesem Beispiel [AMAZ17]:

$$\begin{bmatrix} 1 & 1 & 2 & 4 \\ 5 & 6 & 7 & 8 \\ 3 & 2 & 1 & 0 \\ 1 & 2 & 3 & 4 \end{bmatrix} \longrightarrow \begin{bmatrix} 6 & 8 \\ 3 & 4 \end{bmatrix}$$

Pooling-Layer werden häufig zwischen zwei faltenden Schichten verwendet [GDLG17]. Die letzten verborgenen Schichten eines CNN sind in den meisten Fällen vollständig verbundene Schichten [AMAZ17]. Jede Einheit ist mit allen Einheiten der vorhergehenden Schicht verbunden und entsprechen damit den Schichten klassischer neuronaler Netze. Durch die Reduzierung der Parameter können mehr Schichten implementiert werden, die verschiedene Eigenschaften des Bildes filtern können, was die Ergebnisse des Netzes verbessert [AMAZ17]. Trotz dieser Optimierungen kann das Training eines CNN immer noch sehr aufwendig sein, z.B. wenn nicht genügend Ressourcen zur Verfügung stehen. Ist der Trainingsdatensatz zu klein, kann zudem die Genauigkeit des Systems deutlich abnehmen [PY10]. In einigen Fällen kann ein CNN aber auch zur Lösung einer anderen, aber ähnlichen Aufgabe verwendet werden [PY10], was als *Transfer Learning* bezeichnet wird. Dabei wird die letzte oder die letzten vollständig verbundenen Schicht entfernt und mindestens eine neue Schicht hinzugefügt, die mit dem aktuellen Trainingsdatensatz trainiert wird [OBLS14]. Wendet man Transfer Learning an, so wird das resultierende neuronale Netz häufig als *vortrainiertes Netz* bezeichnet [OBLS14].

CNNs erzielen sehr gute Ergebnisse bei der Bildklassifikation [GDLG17] und bei der Verkehrszeichenklassifikation [SSSI11]. Aus diesem Grund konzentriert sich diese Arbeit auf die Untersuchung von CNN.

## 2.4 Angriffe auf neuronale Netze zur Bildklassifikation

Die Struktur neuronaler Netze legt nahe, dass an verschiedenen Stellen Eigenschaften genutzt werden können, um die Ausgaben der Systeme zu beeinflussen und auszunutzen. Nach der Norm ISO/IEC 27000:2018 hat Informationssicherheit drei wesentliche Schutzziele [Int18]:

- Vertraulichkeit
- Integrität
- Verfügbarkeit

Es sind verschiedene Angriffe auf neuronale Netze möglich, die die verschiedenen Schutzziele gefährden. Beispielsweise ist es möglich, Informationen aus den Trainingsdaten zu extrahieren, ohne dass die Trainingsdaten bekannt sind [JMBR21]. Damit wäre das Schutzziel der Vertraulichkeit umgangen.

Der Fokus dieser Arbeit liegt jedoch auf der Untersuchung der Integrität der Systeme, da sichergestellt werden muss, dass die Informationen, die ein Fahrzeug verwendet, nicht manipuliert sind. Im Folgenden werden verschiedene Angriffsmöglichkeiten beschrieben, die darauf abzielen, die Ausgabe von neuronalen Netzen gezielt zu verändern. Die Architektur neuronaler Netze erlaubt es, bereits die Trainingsdaten zur Manipulation der Ausgabe zu verwenden. Diese Methode wird als *Poisoning-Attack* bezeichnet [BNL]. Dabei werden Trainingsdaten manipuliert, um die Genauigkeit des Systems zu verringern, oder *Hintertüren* (backdoors) einzubauen. Eine Hintertür kann von einem Angreifer genutzt werden, um die Genauigkeit des Systems durch einen *Trigger* gezielt zu verringern. Bei einem Verkehrsschild kann ein solcher Trigger beispielsweise ein spezieller Aufkleber auf einem Stoppschild sein [GDGG], die Ursprünglich zu Darstellung politischer Botschaften genutzt werden (z.B. „STOP RWE“). Durch vortrainierte neuronale Netze kann eine Hintertür in das System eingebaut werden, da die Trainingsdaten den Systementwickler\*innen unbekannt sind [GDGG].

Eine weitere Möglichkeit, die Ausgabe des Systems zu beeinflussen, ist mittels *adversarialer Beispiele*. Dabei wird eine Eingabe ebenfalls so manipuliert, dass die Ausgabe des Netzes nicht mehr korrekt ist [SZS<sup>+</sup>]. Es wird also versucht, eine Eingabe  $x$  so zu manipulieren, dass durch einen Wert  $r$  das System  $f$  eine falsche Klassifikation ausgibt.

$$f(x + r) = l$$

In diesem Fall ist  $l$  eine falsche Klasse. Damit ein Angriff für den Menschen kaum erkennbar ist, sollte  $r$  möglichst klein gewählt werden. Szegedy et al. [SZS<sup>+</sup>] haben

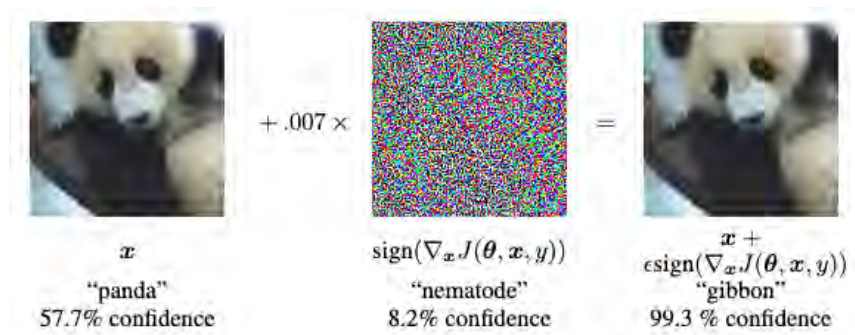


Abbildung 4: Beispiel eines adversarialen Angriffes [GSS].

gezeigt, dass  $r$  berechnet werden kann, wenn das neuronale Netz bekannt ist. Ein Verfahren zur Berechnung ist die *Fast Gradient Sign Method* (FGSM) [GSS]:

$$r = \epsilon * \text{sign}(\Delta_x J(\Theta, x, y))$$

Dabei ist  $\epsilon = [0, 1]$ , um den Einfluss der Änderung zu reduzieren,  $\Theta$  die Menge der Parameter des neuronalen Netzes,  $x$  die Eingabe mit der Ausgabeklasse  $y$ , die Fehlerfunktion  $J(\Theta, x, y)$  und  $\text{sign}(k)$  die Vorzeichenfunktion:

$$\text{sign}(k) = \begin{cases} +1 & k < 0 \\ 0 & k = 0 \\ -1 & k > 0 \end{cases}$$

Abbildung 4 zeigt ein Beispiel eines *adversarialen Angriffs*, der mit der vorgestellten Methode berechnet wurde. Es ist zu erkennen, dass das manipulierte Bild zwar von einem neuronalen Netz falsch klassifiziert wurde, der Unterschied für einen Menschen jedoch kaum sichtbar ist.

Bei dieser Methode werden alle Pixel eines Bildes beeinflusst, allerdings nur um einen geringen Wert. Es gibt aber auch Verfahren, die einen anderen Ansatz verfolgen und versuchen, möglichst wenige Pixel zu verändern. Es konnte gezeigt werden, dass es unter Umständen ausreicht, nur ein Pixel zu verändern [SVS19], um eine Fehlklassifikation hervorzurufen. Dieser Angriff wird als *One Pixel Attack* bezeichnet. Diese Methode versucht herauszufinden, welcher Pixel den größten Einfluss auf die Klassifikation hat. Mit Hilfe einer sogenannten *saliency map*  $M_c(x)$  kann der Einfluss der Pixel auf die Klassifikationen dargestellt werden [WX]. Mit dem Gradienten der Aktivierungsfunktion  $S_c(x)$ , können diese Werte für ein Eingangsbild für den Punkt  $x$  berechnet werden [STK<sup>+</sup>17]:

$$M_c(x) = \frac{\delta S_c(x)}{\delta x}$$

Die durch den Gradienten berechneten Werte, weisen ein hohes Rauschen auf, weshalb verschiedene Methoden entwickelt wurden um diese Werte zu glätten. Smilkov

et al. [STK<sup>+</sup>17] entwickelten aus diesem Grund ein Verfahren, bei dem wiederholt gaußsches Rauschen auf das Eingabebild angewendet wird, für jedes neue Bild der Gradient bestimmt wird und der Durchschnitt dieser Werte zur Berechnung von  $\hat{M}_c$  verwendet wird. Formal kann dies beschrieben werden durch:

$$\hat{M}_c = \frac{1}{n} \sum_1^n M_c(x + \mathcal{N}(0, \sigma^2))$$

Dabei ist  $n$  die Anzahl an neu erstellten Bildern, die jeweils durch das Anwenden von gaußschem Rauschen  $\mathcal{N}(0, \sigma^2)$  mit der Standardabweichung  $\sigma$  erstellt werden. Diese Methode wird als *SmoothGrad* [STK<sup>+</sup>17] bezeichnet.

Mit diesen Berechnungsmethoden können adversariale Beispiele erstellt werden, indem die für die Klassifikation relevantesten Pixel bestimmt und diese manipuliert werden [PMJ<sup>+</sup>16]. Mit Hilfe verschiedener Metriken können nun die Pixel ausgewählt werden, die das Ausgangsbild am wenigsten verändern und gleichzeitig die Genauigkeit des Systems minimieren. Adversariale Attacks, die auf einem neuronalen Netz trainiert wurden, können in einigen Fällen auf andere neuronale Netze übertragen werden [GSS]. Dies gilt auch für adversariale Beispiele, bei denen nur eine beliebige Anzahl von Pixeln verändert wird. [CW17].

Die Angriffe sind nicht nur auf andere neuronale Netze übertragbar, sondern auch auf verschiedene Arten des maschinellen Lernens [PMG]. Daher ist es möglich, Angriffe auf bekannten oder selbst entwickelten Systemen zu erstellen und diese dann auf Systeme anzuwenden, deren genaue Implementierung unbekannt ist. Solche Angriffe werden als *Black-Box-Angriffe* bezeichnet, wie sie beispielsweise von Papernot et al. durchgeführt wurden [PMG].

Nicht alle Angriffe sind für die Anwendung in realen Situationen geeignet. Beispielsweise sind Methoden, die den Zugriff auf Trainingsdaten erfordern, sehr schwierig umzusetzen, da entweder die verwendeten öffentlichen Daten bekannt und veränderbar sein müssen oder der Zugriff auf firmeninterne Strukturen notwendig ist. Daher sind Angriffe auf öffentlich zugängliche Verkehrsschilder besser geeignet, aber auch hier sind die Angriffe unterschiedlich einfach umzusetzen. Angriffe, bei denen die gesamte Fläche des Schildes manipuliert wird, können sehr aufwendig in der Umsetzung sein. Im Gegensatz dazu sind Angriffe, die nur einen kleinen Teil der Oberfläche betreffen, einfacher umzusetzen, da sie schneller anzuwenden sind und weniger Genauigkeit erfordern.

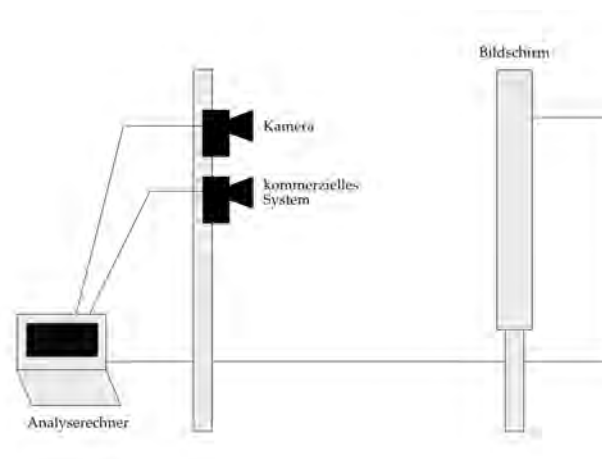


Abbildung 5: Schematischer Aufbau der ersten beiden Versuchsphasen

### 3 Untersuchungen und Ergebnisse

Ziel dieser Arbeit ist die Analyse der IT-Sicherheit von visuellen Fahrassistenzsystemen. Dies soll exemplarisch anhand der Analyse der Robustheit von Verkehrszeichenklassifikationssystemen gegenüber möglichen Angriffen erfolgen. Im Abschnitt 2 wurde gezeigt, warum sich diese Systeme für die Analyse eignen. Die Robustheit von Klassifikationssystemen wird häufig getestet, indem Bilder virtuell manipuliert und dem KI-Modell zugeführt werden [SZS<sup>+</sup>]. Dadurch können zwar viele Angriffe in kurzer Zeit getestet werden, jedoch lassen sich daraus nicht unbedingt reale Angriffe ableiten, da z.B. weitere Systemkomponenten wie Kameras nicht berücksichtigt werden oder die getesteten Angriffe nur schwer auf reale Schilder übertragbar sind. Um ein realistischeres Szenario zu erreichen und dennoch viele Angriffe testen zu können, wird ein Hardware-in-the-Loop-System implementiert. Das HIL-System soll dazu dienen, Angriffe zu finden, die die Ausgabe eines Klassifikationssystems verändern.

#### 3.1 Versuchsaufbau und Methodik

Die Untersuchung besteht aus drei aufeinander folgenden Phasen. Zunächst werden verschiedene Verkehrszeichen klassifiziert, um die korrekte Arbeitsweise des HIL-Systems zu überprüfen. Anschließend werden die Verkehrszeichen virtuell manipuliert und durch das HIL-System klassifiziert, um verschiedene Angriffe effizient zu testen. Schließlich werden aus den Ergebnissen dieser Untersuchung Angriffe abgeleitet, die dann an realen Verkehrszeichen getestet werden.

Für den ersten und zweiten Teil wird ein HIL-System verwendet, bei dem ein kommerzielles Verkehrsschilderkennungssystem und eine an den Analysecomputer angeschlossene Kamera vor einem Bildschirm aufgebaut werden. Abbildung 5 zeigt schematisch diesen Aufbau.



Der Bildschirm kann vom Analyserechner angesteuert und zur Darstellung von Verkehrszeichen verwendet werden. Das kommerzielle System ist in der Lage, Verkehrszeichen zu detektieren, zu klassifizieren und die Ausgabe über einen *Controller Area Network*-Bus (CAN) an den Analyserechner zu übermitteln. Zusätzlich wird das System über ein Labornetzteil mit Strom versorgt und über einen Funktionsgenerator wird ein Geschwindigkeitssignal simuliert. Das Auslesen der Daten und die Integration des Systems in die Simulation wurde in einer Vorarbeit in Zusammenarbeit mit dem BSI erstellt [Ben21].

Als zweite Kamera wird eine *Logitech BRIO* genutzt, welche sich im gleichen Abstand zum Bildschirm befindet wie das kommerzielle System. Die Kamera kann Bildinformationen mit einer Auflösung von 4096 x 2160 Pixel aufnehmen und an den Analyserechner übertragen.

Diese Daten werden zur Klassifikation und Angriffsgenerierung verwendet. Dabei kommen fünf verschiedene neuronale Netze zum Einsatz, von denen drei auf bereits trainierten Modellen basieren. Diese drei Netze haben gemeinsam, dass für das initiale Training der Datensatz *ImageNet* [DDS<sup>+</sup>09] verwendet wurde und sie mittels Transfer Learning an das Problem der Verkehrszeichenklassifikation angepasst wurden. Mit den verschiedenen neuronalen Netzen soll untersucht werden, ob sie unterschiedlich robust auf Angriffe reagieren und wie unterschiedlich sie sich zur Angriffsgenerierung eignen. Diese Netze wurden ebenfalls in einer früheren Arbeit in Zusammenarbeit mit dem BSI entwickelt [Ale20].

Die vortrainierten Netze basieren auf unterschiedlichen Arbeiten und wurden zur Bildklassifikation entwickelt:

- *ResNext* [XGD<sup>+</sup>] (CNN 1)
- *Inception-v3* [SVI<sup>+</sup>] (CNN 2)
- *MobileNetV2* [SHZ<sup>+</sup>18] (CNN 3)

Darüber hinaus wurden zwei weitere CNNs entwickelt, von denen eines die farbigen Eingabebilder verwendet (CNN 4) und eines diese vor der Klassifizierung in Graustufen umwandelt (CNN 5). Die Netze normalisieren die Bilder auf unterschiedliche Weise, so dass zwei Gruppen unterschieden werden können. Die selbsttrainierten Netze skalieren die Bilder auf 32 x 32 Pixel und die vortrainierten Netze auf 224 x 224 Pixel. Alle Netze wurden mit dem Datensatz des *German Traffic Sign Recognition Benchmark* [SSSI11] (zu Ende) trainiert. Dieser Datensatz enthält 43 verschiedene Verkehrszeichenklassen und mehr als 50.000 Bilder von Verkehrsschildern, die in der Nähe von Bochum in Deutschland aufgenommen wurden.

Der Analyserechner dient als zentrale Steuereinheit des HIL-Systems und verwendet neuronale Netze zur Klassifikation und Angriffsberechnung. Der Rechner hat folgende Spezifikationen:

- Betriebssystem: Linux (Ubuntu 22.04.1 LTS)
- Prozessor: 11th Gen Intel Core i7-11800H

Verkehrszeichen	Anzahl der aufgenommenen Schilder
30 km/h	2
50 km/h	19
60 km/h	2
70 km/h	40
80 km/h	3
100 km/h	3
120 km/h	3
Gesamtanzahl	72

Tabelle 1: Anzahl der aufgenommenen Verkehrszeichen, aufgeschlüsselt nach den im Datensatz enthaltenen Verkehrszeichen.

- Grafikeinheit: NVIDIA GeForce RTX 3060 Mobile

Dieser Aufbau wird in allen Untersuchungsphasen verwendet, wobei in Phase 3 der Bildschirm durch eine Aufhängung für reale Schilder ersetzt wird. Die einzelnen Phasen werden im Folgenden näher beschrieben.

### 3.2 Phase 1: Verkehrsschildklassifikation

Zunächst wird untersucht, ob die eingesetzten Systeme die abgebildeten Verkehrszeichen richtig klassifizieren. Dazu wurden Bilder von realen Verkehrszeichen im Rhein-Kreis-Neuss in Deutschland aufgenommen. Die Bilder wurden an einem teils bewölkten Vormittag im Sommer aufgenommen und zeigen Verkehrsschilder auf Autobahnen, Landstraßen und im Stadtverkehr. Die Schilder waren unterschiedlich gut sichtbar und in unterschiedlichem Zustand. Zum Beispiel waren einige Schilder mit Ästen und Blättern bedeckt, andere waren verschmutzt, verbogen, verdreht oder gekippt. Die Schilder wurden vom Beifahrersitz aus fotografiert und in unterschiedlichen Entfernungen zum Auto aufgenommen.

Aufgrund der hohen Dichte an Schildern für Geschwindigkeitsbeschränkungen wird in dieser Arbeit exemplarisch lediglich diese Art von Verkehrszeichen untersucht. Die verschiedenen Verkehrszeichen und ihre Anzahl können der Tabelle 1 entnommen werden. Die Anzahl der verschiedenen Verkehrszeichen variiert zum Teil erheblich, da diese während der Aufnahmefahrt unterschiedlich häufig vorkamen.

Die aufgenommenen Bilder wurden quadratisch zugeschnitten, so dass nur das Verkehrszeichen auf dem Bild zu sehen ist. Da einige Verkehrszeichen, wie bereits erwähnt, aus unterschiedlichen Entfernungen, gekippt oder gedreht aufgenommen wurden, unterscheiden sich die Aufnahmen und der Anteil des Hintergrundes auf den Bildern teilweise sehr stark.

Abbildung 6 zeigt zwei Beispiele aus diesem Datensatz. Da die Untersuchungen im ersten Teil ein möglichst realistisches Szenario abbilden sollen, werden die ab-



Abbildung 6: Beispiele zweier Bilder aus dem aufgenommenen Datensatz, um die Unterschiede verschiedener Schilder zu verdeutlichen.

gebildeten Verkehrszeichen in Fotografien verschiedener Straßen eingebettet. Dazu wird das Verkehrszeichen automatisch aus dem Hintergrund freigestellt, auf eine einheitliche Größe skaliert und an der gleichen Stelle eingebettet.

Dies hat den Vorteil, dass das eigene System so konfiguriert werden kann, dass nur die Bilddaten, die das Verkehrszeichen enthalten, zur Klassifikation verwendet werden und somit die Implementierung eines Detektionssystems entfällt. Da das kommerzielle System sowohl Detektion als auch Klassifikation bietet, werden in diesem Fall beide Funktionen in Kombination getestet.

Die vorbereiteten Verkehrszeichen werden nun nacheinander wie oben beschrieben den Systemen als Eingabe präsentiert. Anschließend wird auf die Klassifikation der Systeme gewartet und diese schließlich protokolliert. Für die Klassifikation werden alle neuronalen Netze und das kommerzielle System verwendet. Da das kommerzielle System nur Daten übermittelt, wenn ein Schild erkannt wurde und klassifiziert werden konnte, wird maximal zwei Sekunden auf ein Ergebnis gewartet. Da bei einer Geschwindigkeit von 50 km/h in zwei Sekunden ca. 30 Meter zurückgelegt werden und gerade im innerstädtischen Bereich Verkehrszeichen häufig erst aus geringer Distanz erkennbar sind, wurde der Grenzwert für die Erfassung auf zwei Sekunden festgelegt. Überträgt das kommerzielle System kein Ergebnis, wird ebenfalls davon ausgegangen, dass ein Schild der falschen Klasse zugeordnet wurde, da nicht bekannt ist, ob das Schild nicht erkannt oder der Klasse *kein Verkehrszeichen* zugeordnet wurde.

### **Ergebnisse der Klassifikation**

Zunächst wurden die eingebetteten Schilder mit Hilfe der neuronalen Netze klassifiziert. Die Klassifikation wurde mehrfach durchgeführt. Damit sollte getestet werden, ob äußere Einflüsse die Klassifikation beeinflussen. Dazu wurden im Wesentli-

Versuch	Bildschirm	Tageszeit	Hintergrund
Versuch 1 (V1)	Dell U2913WM	Abend	Landstraße 2
Versuch 2 (V2)	Iiyama TF6537UHSC-B2AG	Vormittag	Landstraße 2
Versuch 3 (V3)	Sony KDL-49WE755	Mittag	Stadt
Versuch 4 (V4)	Sony KDL-49WE755	Nachmittag	Landstraße 1
Versuch 5 (V5)	Sony KDL-49WE755	Abend	Landstraße 2

Tabelle 2: Übersicht über die Versuche der Phase 1 und deren Versuchsparameter.

chen drei Parameter variiert:

- Bildschirm zur Darstellung der Zeichen
- Tageszeit
- Hintergrund zur Einbettung der Zeichen

Der Bildschirm wurde hauptsächlich ausgetauscht, um zu prüfen, ob Eigenschaften wie die Bildwiederholrate einen Einfluss auf die Klassifizierung haben. Die Tageszeit wurde variiert, um den Einfluss von Reflexionen des Umgebungslichts auf dem Bildschirm und der Kameralinse zu testen. Außerdem wurde das Hintergrundbild variiert, um zu prüfen, ob dies die Klassifikation beeinflusst. Die Hintergrundbilder können Abbildung 15 entnommen werden. Die Parameter der einzelnen Versuche können der Tabelle 2 entnommen werden.

Das kommerzielle System kann ausschließlich in den Räumlichkeiten des BSI getestet werden, weshalb in den ersten Versuchen nur die neuronalen Netze getestet wurden. Die Klassifikation des kommerziellen Systems wird mit den gleichen Parametern wie in Versuch 2 ermittelt.

Die Klassifikationsergebnisse können der Tabelle 3 entnommen werden. Dabei bedeutet die Schreibweise X|Y einer Zelle, dass X Elemente der Klassifikation der richtigen Klasse und Y Elemente der falschen Klasse zugeordnet wurden.

Die Klassifikationsergebnisse unterscheiden sich nur geringfügig zwischen den verschiedenen Versuchen und den verschiedenen neuronalen Netzen. So beträgt der Mittelwert der richtig klassifizierten Schilder 70,77 von insgesamt 72 Schildern und die Standardabweichung 0,65. Die falsch klassifizierten Schilder weisen Auffälligkeiten auf, wie z.B. Laub, das Teile des Schildes verdeckt, Schrägstellungen oder nicht standardisierte Beschriftungen. Die Schilder können der Abbildung 16 entnommen werden.

Das kommerzielle System zeigt ähnliche Werte. Bei der Klassifikation wurden 70 Schilder richtig und 2 Schilder falsch klassifiziert. Auch hier wurden Schilder falsch klassifiziert, bei denen Laub Teile des Bildes verdeckt.

Neuronales Netz	V1	V2	V3	V4	V5
CNN 1	70 2	71 1	71 1	72 0	71 1
CNN 2	71 1	70 2	71 1	69 3	71 1
CNN 3	71 1	71 1	71 1	71 1	70 2
CNN 4	71 1	71 1	70 2	72 0	71 1
CNN 5	70 2	70 2	71 1	71 1	71 1

Tabelle 3: Klassifikationsergebnisse der neuronalen Netze in den verschiedenen Versuchen. Die Schreibweise X|Y einer Zelle bedeutet, dass X Elemente der Klassifikation der richtigen Klasse und Y Elemente der falschen Klasse zugeordnet wurden.

### 3.3 Phase 2: Simulierte Angriffe

In Phase 2 wird der gleiche Aufbau und die gleichen Daten wie in Phase 1 verwendet. Die angezeigten Verkehrszeichen werden jedoch manipuliert dargestellt, um die Robustheit der Systeme gegenüber gezielten Manipulationen zu untersuchen. Dazu werden mit der Methode *SmoothGrad* von Smilkov et al. [STK<sup>+</sup>17] *saliency maps* berechnet, die die Relevanz der Pixel anzeigen, daraus ein adversariales Beispiel gebildet und dieses ebenfalls in eine Fotografie einer Straße eingebettet. Für die Berechnung werden die fünf neuronalen Netze verwendet, wobei zunächst für jedes Netz ein adversariales Beispiel berechnet wird und anschließend die Beispiele aus Kombinationen der Netze berechnet werden. Die drei vortrainierten Netze und die selbsttrainierten Netze bilden dabei Untergruppen, die jeweils kombiniert werden. Zur Berechnung einer *saliency map*  $\tilde{M}_c$  wird folgende Funktion verwendet [Ale20]:

$$\tilde{M}_c = \frac{1}{\max(\sum_{i=1}^n \hat{M}_i(x))} \sum_{i=1}^n \hat{M}_i(x)$$

Dabei ist  $n$  die Anzahl der beteiligten neuronalen Netze und  $\hat{M}_i(x)$  die mittels *SmoothGrad* berechneten *saliency map* für das Netz  $i$  mit der Eingabe  $x$ . Es ist zu beachten, dass diese Funktion auch für die einzelnen Netze verwendet wird und daher in diesen Fällen  $n = 1$  gilt. Die berechneten Werte werden der Größe nach sortiert, um die relevantesten Pixel zu bestimmen. Anschließend werden die zuvor ermittelten Pixel manipuliert, indem der jeweilige Farbwert maximiert oder minimiert wird, je nachdem, welche Abweichung vom ursprünglichen Wert größer ist.

Ziel der Kombination ist es, dass die Angriffe besser generalisieren und bei mehr Systemen zu Fehlklassifikationen führen. Insgesamt werden so für jedes Schild zehn verschiedene Angriffe berechnet. Für jeden Angriff wird zusätzlich die Anzahl der manipulierten Pixel variiert. Für jedes adversariale Beispiel wird die Anzahl der veränderten Pixel iterativ von 1% auf 5% der Gesamtpixelanzahl erhöht.

Die angezeigten adversarialen Beispiele werden von allen neuronalen Netzen und dem kommerziellen System klassifiziert und das Ergebnis protokolliert.

## Ergebnisse der virtuellen Angriffe

Die Systeme werden in Phase auf ihre Robustheit gegenüber der oben beschriebenen Angriffen getestet. Wie in Phase 1 werden die nun manipulierten Schilder nacheinander auf dem Bildschirm angezeigt und von den neuronalen Netzen und dem kommerziellen System klassifiziert. Insgesamt wurden 21.600 Angriffe durchgeführt, wobei jeweils 3.600 Angriffe pro Klassifikationssystem durchgeführt wurden.

Der Unterschied zwischen den Angriffen, die von den selbsttrainierten Netzen und ihren Kombinationen berechnet wurden, und den Angriffen, die von den vortrainierten Netzen und ihren Kombinationen berechnet wurden, besteht darin, dass erstere durch weniger, aber größere modifizierte Flächen gekennzeichnet sind, während die vortrainierten Netze durch mehr, aber kleinere Flächen gekennzeichnet sind. Abbildung 7 zeigt beispielhaft zwei dieser Angriffe.

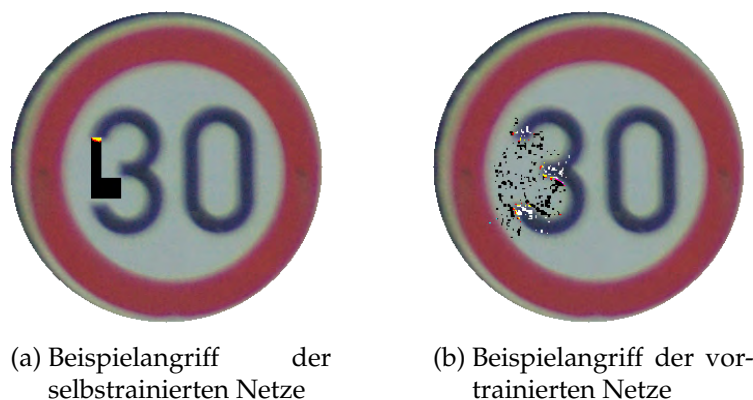


Abbildung 7: Beispiele verschiedener Angriffe

Abbildung 8 zeigt den prozentualen Anteil der falsch klassifizierten Schilder. Die Daten können im Detail in Tabelle 5 nachgelesen werden. Die Säulen zeigen das Ergebnis des Klassifikationssystems in Abhängigkeit von der jeweiligen Kombination, mit der ein Angriff berechnet wurde. Außerdem wird der durchschnittliche Anteil falsch klassifizierter Schilder für jede Kombination angezeigt.

Es ist zu erkennen, dass die Angriffe zu unterschiedlichen Ergebnissen führen. Die Angriffe der selbsttrainierten Netze und die Angriffe der Kombination dieser beiden Netze haben im Durchschnitt jeweils etwas mehr als 50 % der Schilder falsch klassifiziert. Bei den vortrainierten Netzen konnte der mit ResNext berechnete Angriff mit durchschnittlich 40 % die meisten Schilder erfolgreich manipulieren. Jeweils durchschnittlich etwa 20 % der Schilder wurden durch die Angriffe der Netze Inception-v3 und MobileNetV2 falsch klassifiziert. Die Angriffe, die durch die Kombination der vortrainierten Netze erzeugt wurden, konnten jeweils einen Wert an falsch klassifizierten Schildern erreichen, der zwischen den Werten der Angriffe der beteiligten Netze lag, wenn diese alleine eingesetzt wurden.

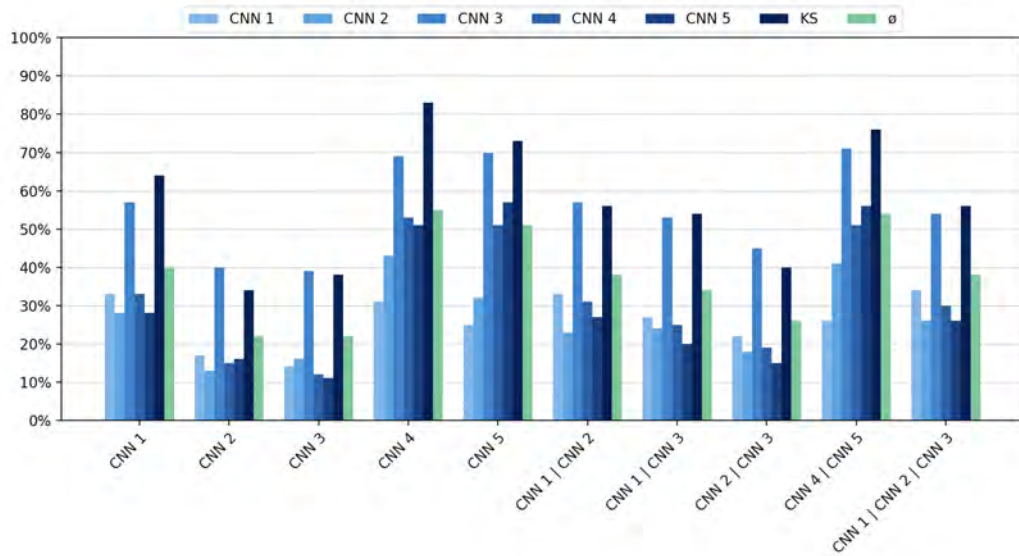


Abbildung 8: Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Klassifikationssystem und den Durchschnitt aller Systeme. Die Gruppierungen beziehen sich jeweils auf die Kombination der neuronalen Netze, mit denen die Angriffe berechnet wurden, wobei die verschiedenen Anteile veränderter Pixel und die Angriffe auf Basis verschiedener Verkehrszeichen zusammengefasst wurden.

Außerdem ist zu erkennen, dass das kommerzielle System und das MobileNetV2-Netz unabhängig vom Angriff die meisten Schilder falsch klassifizierten. Bei den Angriffen, bei denen selbsttrainierte Netze eingesetzt wurden, haben die Netze ResNext und Inception-v3 die wenigsten Fehlklassifikationen. Bei den anderen Angriffen erreichen ResNext, Inception-v3 und die beiden selbsttrainierten Netze ähnliche Werte.

Abbildung 9 zeigt ebenfalls den prozentualen Anteil der falsch klassifizierten Verkehrszeichen. Die Säulen zeigen das Ergebnis des Klassifikationssystems für jedes Verkehrszeichen und zusätzlich den durchschnittlichen Anteil für jedes Verkehrszeichen. Die absoluten Zahlen der Angriffe sind je nach Verkehrszeichen sehr unterschiedlich und können der Tabelle 6 entnommen werden. Am häufigsten wurden die 70 km/h-Schilder mit 12.000 Angriffen verwendet, am seltensten die 30 km/h-Schilder mit 600 Angriffen.

Die 30 km/h-Schilder wurden in durchschnittlich 77 % der Fälle der falschen Klasse zugeordnet und waren damit am häufigsten betroffen. Auch die 120 km/h-Schilder wurden in durchschnittlich 70 % falsch klassifiziert. Die übrigen Schilder liegen unter diesen Werten: Bei den 50 km/h und 60 km/h Schildern wurden ca.

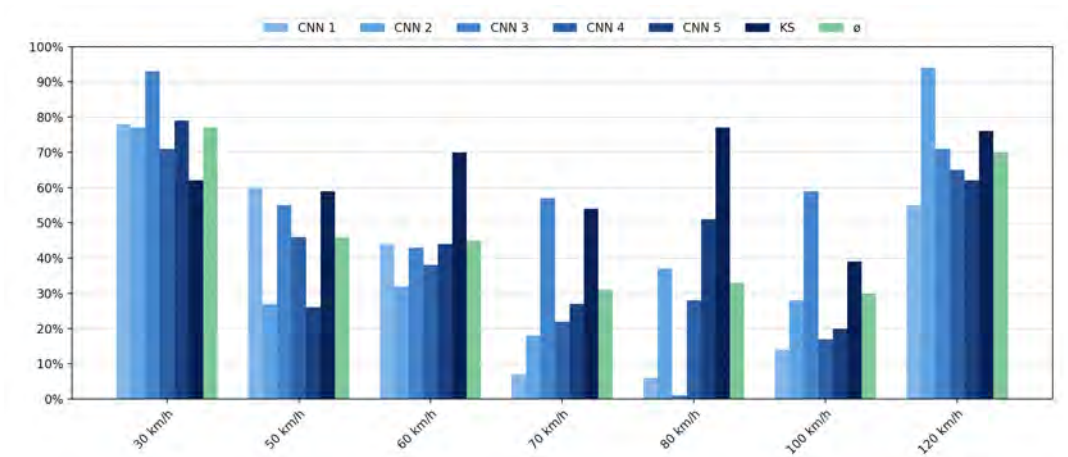


Abbildung 9: Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Klassifikationssystem und den Durchschnitt aller Systeme. Die Gruppierungen beziehen sich jeweils auf die einzelnen Verkehrszeichen, wobei alle Schilder, die ein bestimmtes Zeichen darstellen, und die unterschiedlichen Anteile der veränderten Pixel zusammengefasst wurden.

45 % falsch klassifiziert, bei den übrigen Schildern ca. 30 %.

Die Streuung der Falschklassifizierungen der einzelnen Systeme ist unterscheidet sich. Beim kommerziellen System wurden die 80 km/h-Schilder mit 77 % am häufigsten falsch klassifiziert, die 100 km/h-Schilder mit 39 % am seltensten. Dies entspricht einem Unterschied von 38 Prozentpunkten. Im Gegensatz dazu hat das MobileNetV2-Netz weniger als 1 % der 80 km/h-Schilder und 93 % der 30 km/h-Schilder falsch klassifiziert, was einem Unterschied von 92 Prozentpunkten entspricht.

Abbildung 10, Abbildung 11 und Abbildung 12 zeigen den Anteil der falsch klassifizierten Verkehrszeichen, aufgeschlüsselt nach dem Anteil der manipulierten Pixel. In Abbildung 10 wird dies in Relation zu den Kombinationen gesetzt, mit denen ein Angriff berechnet wurde, in Abbildung 11 in Relation zu den verschiedenen Verkehrszeichen und in Abbildung 12 in Relation zu den Klassifikationssystemen. Es ist zu erkennen, dass der Anteil der Verkehrszeichen, die der falschen Klasse zugeordnet wurden, mit der Anzahl der veränderten Pixel zunimmt. Dieser Zusammenhang gilt für alle Angriffskombinationen, alle Verkehrszeichen und alle Klassifikationssysteme. Ebenso bleibt das Verhältnis der Anteile falsch klassifizierter Schilder innerhalb der Angriffskombination und innerhalb der Klassifikationssysteme annähernd gleich. Details können den Tabellen 7 und 9 entnommen werden.

Allerdings ändert sich das Verhältnis der Anteile falscher Klassifikationen innerhalb der verschiedenen Verkehrszeichen. So steigt der Anteil der Fehlklassifikationen bei den 100 km/h-Schildern stärker an als bei den 70 km/h-Schildern. Der glei-



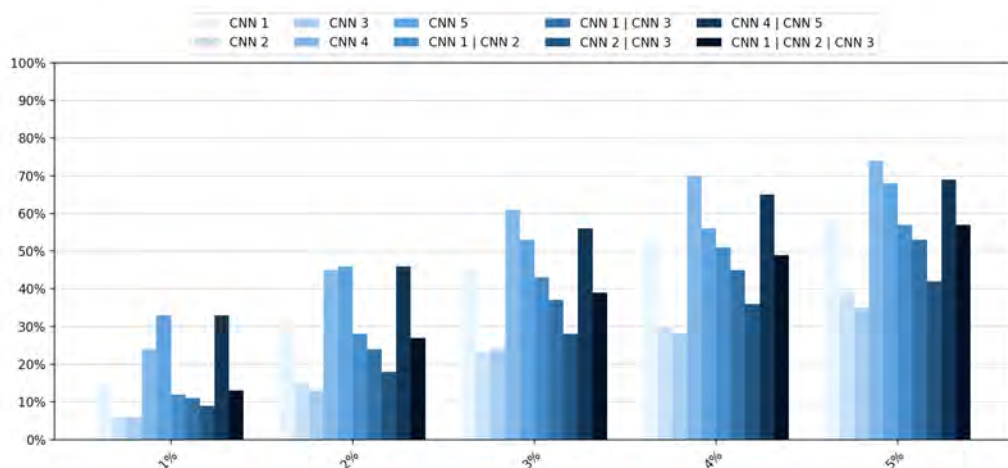


Abbildung 10: Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Kombination der neuronalen Netze, mit denen die Angriffe berechnet wurden. Die Gruppierungen beziehen sich jeweils auf den Anteil der veränderten Pixel, wobei die einzelnen Verkehrszeichen und Klassifikationssysteme zusammengefasst wurden.

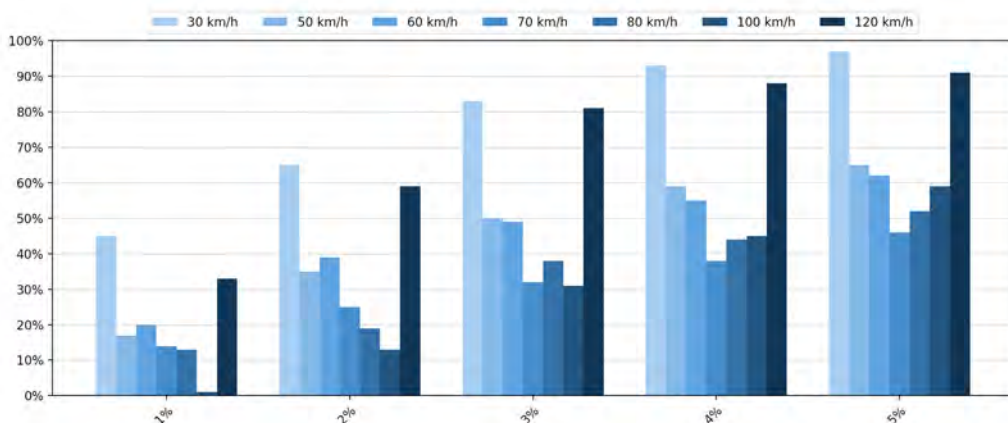


Abbildung 11: Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Verkehrszeichen. Die Gruppierungen beziehen sich jeweils auf den Anteil der veränderten Pixel, wobei die Kombinationen der neuronalen Netze, mit denen der Angriff berechnet wurde, und die Klassifikationssysteme zusammengefasst wurden.

che Effekt tritt bei den Schildern 50 km/h und 60 km/h auf. Auch hier werden die Verkehrszeichen in absoluten Zahlen unterschiedlich häufig manipuliert, wie Tabel-

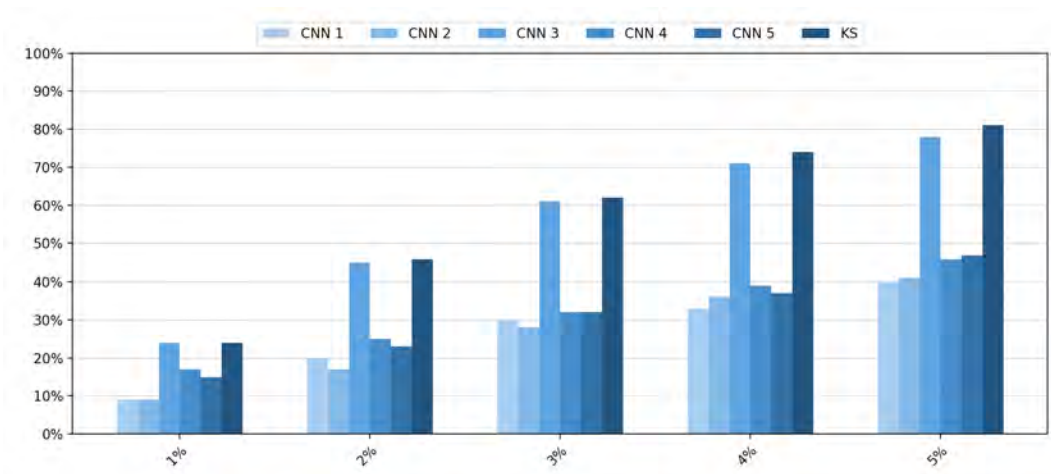


Abbildung 12: Übersicht über den prozentualen Anteil falsch klassifizierter Schilder. Die Balken zeigen den Anteil pro Klassifikationssystem. Die Gruppierungen beziehen sich jeweils auf den Anteil der veränderten Pixel, wobei die einzelnen Verkehrszeichen und die Kombinationen der neuronalen Netze, mit denen der Angriff berechnet wurde, zusammengefasst wurden.

le 8 zeigt.

Insgesamt traten 12 Angriffe auf, bei denen 1 % der Pixel manipuliert wurden und die bei allen Systemen zu einer Fehlklassifikation führten. Eine Übersicht über diese Angriffe kann der Tabelle 10 entnommen werden. Sieben dieser Angriffe basieren auf Bildern von Verkehrszeichen, die im nicht manipulierten Zustand nicht bei allen Systemen zu einer korrekten Klassifikation geführt haben. Die verbliebenen Angriffe sind in Abbildung 13 dargestellt. Es ist zu bemerken, dass Angriff 4 und Angriff 5 identisch sind und dass alle Angriffe von den selbsttrainierten Netzen oder deren Kombination berechnet wurden. Die Positionen der veränderten Pixel befinden sich hauptsächlich im linken mittleren Bereich der Ziffer 3.

### 3.4 Phase 3: Übertragung der Angriffe auf reale Verkehrsschilder

In der dritten Phase wird der Bildschirm ausgetauscht und reale Verkehrszeichen werden in den Sichtbereich der Kameras gehängt. Basierend auf den Ergebnissen des ersten Teils werden diese Verkehrszeichen manipuliert. Damit soll exemplarisch überprüft werden, ob die auf dem Bildschirm dargestellten Ergebnisse auf ein reales Verkehrsschild übertragbar sind.

Zunächst werden nicht manipulierte Verkehrsschilder vor dem System aufgebaut, um zu testen, ob die Systeme die Schilder korrekt klassifizieren. Anschließend werden exemplarisch Angriffe auf die Schilder übertragen, die bei möglichst vielen neuronalen Netzen und dem kommerziellen System zu Fehlklassifikationen füh-

ren und dabei möglichst wenige Bildpunkte verändern. Durch schwarze und weiße Aufkleber mit einer Größe von jeweils 1,5 x 1,5 cm sollen die Schilder so verändert werden, dass sie ebenfalls eine Fehlklassifikation hervorrufen. Die Aufkleber sollen eine Annäherung an die veränderten Pixel darstellen und werden dort angebracht, wo die Dichte der veränderten Pixel am höchsten ist.

Es werden Schilder der Reflexionsklasse RA1 mit einem Durchmesser von 60 cm verwendet. Ein Aufkleber bedeckt somit 0,08% der Fläche eines Schildes, so dass mit 13 Aufklebern ca. 1% der Fläche abgedeckt werden kann.

Da das kommerzielle System möglicherweise kein Ergebnis überträgt, werden sowohl das manipulierte als auch die nicht manipulierte Kontrollschilder getestet, um sicherzustellen, dass das System ordnungsgemäß funktioniert.

### Ergebnisse der Übertragung auf reale Verkehrsschilder

Die in Abbildung 13 dargestellten Angriffe wurden in Phase 3 verwendet, um sie auf reale Schilder zu übertragen. Sie wurden ausgewählt, weil bei ihnen jeweils nur 1 % der Pixel verändert wurden, weil sie bei allen Systemen zu einer falschen Klassifizierung geführt haben und weil die Schilder in Phase 1 korrekt klassifiziert wurden. Damit sollte exemplarisch ermittelt werden, ob ein HIL-System geeignet ist, einen real anwendbaren Angriff zu berechnen. Da die Angriffe 4 und 5 identisch sind, wurde Angriff 5 nicht weiter betrachtet. Abbildung 14 zeigt die 4 übertragenen Angriffe. Es wurden jeweils 13 Aufkleber verwendet, die in der Summe etwa 1 % der Schildfläche bedecken.

Vor der Durchführung der Angriffe wurde getestet, ob die Klassifikationssysteme die nicht manipulierte Schilder richtig klassifizieren. Dabei wurden alle Klassen korrekt bestimmt.

Die Ergebnisse der Klassifikationen der manipulierte Schilder können der Tabelle 4 entnommen werden. Die neuronalen Netze Inception-v3, MobileNetV2 und ein selbsttrainiertes Netz haben alle Schilder falsch klassifiziert und jeweils der Klasse der 80 km/h-Schilder zugeordnet. Das andere selbsttrainierte Netz hat alle Verkehrszeichen richtig klassifiziert. Das Netz ResNext hat in 3 von 4 Fällen das Schild einer falschen Klasse zugeordnet, wobei zweimal die Klasse der 80 km/h-Schilder berechnet wurde. Das kommerzielle System erkannte kein manipulierte Schild. In allen Fällen wurde jedoch das Kontrollschild erkannt.



Abbildung 13: Berechnete Angriffe, bei denen nur 1 % der Pixel manipuliert wurden und die bei allen Systemen zu einer Fehlentscheidung führten.

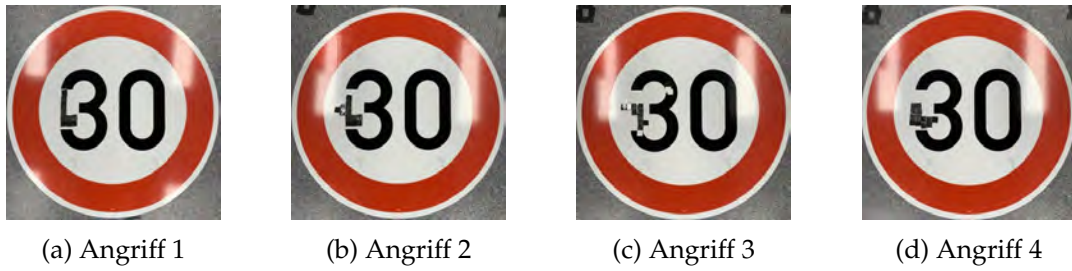


Abbildung 14: Berechnete Angriffe, die auf reale Schilder übertragen wurden.

	CNN 1	CNN 2	CNN 3	CNN 4	CNN 5	KS
Angriff 1	80 km/h	80 km/h	80 km/h	✓	80 km/h	-
Angriff 2	80 km/h	80 km/h	80 km/h	✓	80 km/h	-
Angriff 3	70 km/h	80 km/h	80 km/h	✓	80 km/h	-
Angriff 4	✓	80 km/h	80 km/h	✓	80 km/h	-

Tabelle 4: Klassifikationsergebnisse von realen Angriffen auf die Klassifikationssysteme, wobei bei einer Fehlklassifikation die ermittelte Klasse angegeben ist. Ein ✓ bedeutet, dass die Klasse korrekt bestimmt wurde. Das kommerzielle System hat kein Schild erkannt.

## 4 Diskussion

Ziel dieser Arbeit ist es, die Eignung von Hardware-in-the-Loop-Systemen zur Bewertung von visuellen Fahrerassistenzsystemen aus Sicht der IT-Sicherheit zu untersuchen. Zu diesem Zweck wurde ein HIL-System entwickelt, mit dessen Hilfe Angriffe erkannt werden sollen, die zu einer Fehlklassifizierung realer manipulierter Verkehrszeichen führen.

Zunächst konnte gezeigt werden, dass ein kommerzielles System und verschiedene selbst entwickelte Systeme als HIL-Komponenten funktionieren und auf einem Bildschirm dargestellte Verkehrszeichen in verschiedenen Situationen weitgehend korrekt klassifizieren können. Anschließend konnte mit Hilfe von fünf neuronalen Netzen gezeigt werden, dass Angriffe berechnet und in die Simulation integriert werden können, die zu Fehlklassifikationen führen. Weiterhin konnten Angriffe identifiziert werden, die auf reale Schilder übertragbar sind und ebenfalls zu Fehlklassifikationen führen.

Diese Ergebnisse sollen nun näher betrachtet werden. Um die Eignung der Systeme als HIL-Komponente zu prüfen, wurde mit leicht veränderten Parametern getestet, ob die Systeme reproduzierbare Klassifikationsergebnisse liefern. Da diese unabhängig von der Tageszeit, den getesteten Bildschirmen und den Hintergrundbildern ähnliche Ergebnisse liefern (siehe Tabelle 3), kann davon ausgegangen werden, dass sich der Versuchsaufbau eignet, um Angriffe zu testen.

Es war zu erwarten, dass falsch klassifizierte Schilder Abweichungen aufweisen, z.B. in Form von Neigung oder Verdeckung durch Laub. Damit konnte bereits gezeigt werden, dass die Systeme nicht in allen Situationen robust auf Störungen reagieren und Angriffe vermutlich erfolgreich sein können. Rückschlüsse auf reale Fahrsituationen sind jedoch schwierig, da die Fotos nur eine bestimmte Position darstellen und das System durch eine veränderte Position zum Schild durchaus andere Entscheidungen treffen kann. Weitere Forschungen können nicht nur Bilder in die Simulation von Fahrsituationen einbeziehen, sondern beispielsweise auch ganze Fahrsituationen simulieren.

Des Weiteren ist zu beachten, dass das kommerzielle System zwar nur einmal getestet wurde, die Abweichung zu den anderen Systemen aber sehr gering war, so dass bei anderen Rahmenbedingungen nur geringfügig andere Ergebnisse zu erwarten wären.

Insgesamt kann also davon ausgegangen werden, dass die Systeme in diesem Versuchsaufbau in der Lage sind, die nicht manipulierten Schilder korrekt zu klassifizieren. Wenn die simulierten Angriffe zu Fehlentscheidungen führen, besteht also eine Wahrscheinlichkeit, dass dies auch in realen Situationen zu Fehlentscheidungen führen kann. Da die Angriffe zu unterschiedlichen Ergebnissen geführt haben, können auch verschiedene Schlussfolgerungen gezogen werden.

Insbesondere fällt auf, dass die mit den selbsttrainierten Netzen berechneten Angriffe zu mehr Fehlklassifikationen führen als die mit den vortrainierten Netzen berechneten. Die Ursache hierfür müsste in weiteren Untersuchungen geklärt wer-

den. Da die neuronalen Netze die Bilder jedoch unterschiedlich normalisieren und vor allem auf verschiedene Größen skalieren, variieren auch die Regionen der veränderten Pixel. Eine Vermutung, warum mit diesen Netzen bessere Angriffe berechnet wurden, ist, dass Angriffe mit weniger, aber größeren manipulierten Bereichen bessere Ergebnisse liefern als Angriffe mit vielen, aber kleinen manipulierten Bereichen.

Da auch die kombinierten Angriffe keine besseren Ergebnisse liefern als die Angriffe der einzelnen Netze, kann vermutet werden, dass die Angriffe durch die Kombination nicht besser verallgemeinert und auf andere Klassifikationssysteme übertragen werden können. Allerdings muss an dieser Stelle betont werden, dass die neuronalen Netze ein größeres Diversifikationspotential bieten, da z.B. die selbsttrainierten Netze nur einen Unterschied in der Bildnormalisierung und nicht in der Architektur (o.ä.) aufweisen. Würden sich die neuronalen Netze stärker unterscheiden, z.B. durch die Verwendung unterschiedlicher Trainingsdatensätze, könnte dennoch eine Generalisierung auftreten. Dies könnte in einer weiteren Forschungsarbeit gezielt getestet werden.

Da sich die neuronalen Netze nicht nur in der Angriffsgenerierung, sondern auch in der Klassifikationsgenauigkeit unterscheiden, können auch hieraus Rückschlüsse gezogen werden. Da neuronale Netze unterschiedlich häufig Schilder falsch klassifizieren, sollten sie vor dem Einsatz als Klassifikationssystem ausgiebig getestet werden. Ein HIL-System liefert hierfür erste Anhaltspunkte und kann ggf. bereits für ein Ausschlussverfahren genutzt werden. Allerdings ist auch hier zu beachten, dass die Ergebnisse nicht zwangsläufig auf ein reales Fahrscenario übertragbar sind und die Transferierbarkeit erst systematisch getestet werden muss.

Darüber hinaus kann die Schlussfolgerung gezogen werden, dass das kommerzielle System häufiger zu Fehlklassifikationen verleitet werden konnte. Da die neuronalen Netze jedoch nur den relevanten Bildausschnitt als Input erhalten, ist ein direkter Vergleich zum kommerziellen System nicht möglich. Um einen besseren Vergleich zu ermöglichen, müssten die eigenen Systeme näher an den Funktionsumfang des kommerziellen Systems angepasst werden. Somit ist ein HIL-System zwar geeignet, die jeweiligen Fehlklassifikationen zu quantifizieren, Vergleiche dieser Ergebnisse sind jedoch nur bedingt möglich.

Weiterhin zeigt Abbildung 9, dass die Angriffe auf die verschiedenen Zeichen im Durchschnitt unterschiedlich häufig zu einer Fehlklassifikation führten. Da sich einige Ziffern ähnlicher sind als andere, wie z.B. die Ziffernpaare (3,8) im Vergleich zu (5,7), war zu vermuten, dass dieser Effekt auftreten könnte. Dies müsste jedoch genauer untersucht werden, da durch die vielen Kombinationen von Zahlenpaaren für mehrere Ziffern ähnliche Partner gefunden werden können. Außerdem könnte dieser Effekt auch durch die unterschiedliche Häufigkeit der Verkehrszeichen im aufgenommenen Datensatz hervorgerufen werden. Je weniger Bilder eines Verkehrszeichens vorhanden sind, desto größer ist der Einfluss verschiedener Parameter, wie z.B. die Bildqualität oder die Rotation des Verkehrszeichens. Um diesen Effekt bestätigen zu können, ist es daher notwendig, mehr Schilder zu testen und

sicherzustellen, dass der Datensatz möglichst homogen verteilt ist und die Bilder innerhalb einer Verkehrszeichenklasse bewusst variiert werden.

Dass die Streuung der Fehlklassifikationen innerhalb eines Klassifizierungssystems teilweise sehr groß ist, könnte auch auf den zugrunde liegenden Schilddatensatz zurückzuführen sein. Dies gilt auch für die unterschiedliche Wachstumsgeschwindigkeit der Angriffe auf bestimmte Zeichen in Abbildung 11, wenn der Anteil der veränderten Pixel zunimmt.

Ein weiteres zu erwartendes Ergebnis war, dass der Anteil der Fehlklassifikationen steigt, wenn ein größerer Anteil der Bildpunkte eines Bildes manipuliert wird. Auch dieses Ergebnis ist wie erwartet eingetreten. Aufgrund der begrenzten Anzahl der getesteten Anteile manipulierter Pixel ist es nicht möglich, Rückschlüsse auf einen optimalen Angriff zu ziehen, bei dem ein minimaler Anteil der Bildpunkte verändert wird und möglichst viele Angriffe zu einer Fehlklassifikation führen. Dazu könnte die Erhöhung der Anteile in kleineren Schritten erfolgen und insgesamt ein größeres Intervall abdecken. Auch dies kann mit einem HIL-System getestet werden, da sich durch die Erhöhung der Anteile der Zeitaufwand nur linear erhöht, ansonsten aber keine großen Änderungen vorgenommen werden müssen. Dadurch kann ggf. die Wahrscheinlichkeit eines erfolgreichen Angriffs auf reale Schilder erhöht werden.

Die Angriffe, bei denen 1 % der Pixel verändert wurden und bei denen alle Systeme das Schild falsch klassifiziert haben, sind vor dem Hintergrund der oben beschriebenen Ergebnisse nachvollziehbar. Sowohl der Erfolg der selbsttrainierten Netze als auch das Auftreten der 30 km/h-Schilder als Basis der Angriffe kann mit den oben genannten Argumenten begründet werden. Der Erfolg der Angriffe auf die 50 km/h und 120 km/h Schilder kann damit begründet werden, dass die Schilder auch im nicht manipulierten Zustand von einigen Systemen nicht korrekt klassifiziert wurden. Aus diesem Grund wurden die Schilder nicht weiter untersucht.

Die Angriffe waren aufgrund der zusammenhängenden Flächen der manipulierten Pixel einfach auf reale Schilder übertragbar. Der Erfolg dieser Angriffe bei fast allen Systemen spricht für die Eignung von HIL-Systemen zur Bewertung der IT-Sicherheit von visuellen Fahrerassistenzsystemen. Die Häufigkeit des Auftretens der Klasse der 80 km/h-Schilder unterstützt die Hypothese, dass die Ähnlichkeit der Zahlenpaare einen Einfluss auf die Angreifbarkeit bestimmter Verkehrszeichen haben könnte. Darüber hinaus deutet dieses Verhalten darauf hin, dass Angriffe bewusst eingesetzt werden können, um gezielt eine bestimmte Klasse in der Klassifikation hervorzurufen. Dies könnte in weiteren Arbeiten untersucht werden. Des Weiteren kann in weiteren Analysen getestet werden, ob Systeme, die Verkehrszeichen sowohl detektieren als auch klassifizieren, ebenfalls eine Fehlklassifikation hervorrufen können, bei der ein falsches Verkehrszeichen erkannt wird.

Die Tatsache, dass ein selbsttrainiertes Netz als einziges System in der Lage war, alle Angriffe korrekt zu klassifizieren (siehe Abbildung 4), lässt vermuten, dass bestimmte Eigenschaften neuronaler Netze gegen diese Art von Angriffen schützen könnten. Auch dies könnte in Folgeuntersuchungen untersucht werden.

Insgesamt lässt sich festhalten, dass Hardware-in-the-Loop-Systeme prinzipiell Rückschlüsse auf die Sicherheit visueller Fahrerassistenzsysteme zulassen. Die Grenzen und Möglichkeiten dieser Systeme müssen jedoch in weiteren Untersuchungen genauer betrachtet werden. Ebenso können diese Systeme genutzt werden, um systematisch Angriffe zu finden, die auf reale Verkehrszeichen übertragen werden können. Anhand dieser Angriffe können dann einfacher Untersuchungen in realeren Fahrsituationen durchgeführt und aus den Ergebnissen eventuell Schutzmechanismen abgeleitet werden.

#### 4.1 Limitationen

Wie bereits erwähnt, unterliegt diese Arbeit einigen Limitationen. Zunächst ist darauf hinzuweisen, dass der Datensatz der Verkehrszeichen, mit denen die Angriffe berechnet wurden, gewisse Schwächen aufweist. So treten die verschiedenen Verkehrszeichen unterschiedlich häufig auf, und vor allem kommen einige Verkehrszeichen lediglich zwei- oder dreimal vor. Darüber hinaus könnten systematisch unterschiedliche Situationen in den Bildern repräsentiert sein, wie z.B. Drehungen, Verdeckungen durch Laub, unterschiedliche Wetterbedingungen oder unterschiedliche Tageszeiten.

Ebenso wurden nur Geschwindigkeitsbegrenzungen abgebildet. Durch die Vielzahl anderer Verkehrszeichen könnten weitere kritische Angriffsmöglichkeiten, wie z.B. Angriffe auf Stoppschilder, in Betracht gezogen werden.

Eine weitere Einschränkung stellen die selbst entwickelten neuronalen Netze dar. Zum einen ist nur eine Klassifikation und keine Detektion implementiert und somit kann nur ein Teilaspekt einer Verkehrszeichenerkennung ausführlich getestet werden. Zum anderen basieren die Netze auf dem gleichen Trainingsdatensatz. Würden unterschiedliche Trainingsdatensätze verwendet, könnten die Systeme verschieden auf Eingaben reagieren. Es ist davon auszugehen, dass durch die Umsetzung dieser beiden Aspekte weitere Erkenntnisse gewonnen werden können.

Darüber hinaus lässt die exemplarische Untersuchung von Verkehrszeichenerkennungssystemen nicht unbedingt Rückschlüsse auf andere visuelle Fahrerassistenzsysteme zu. Dies müsste in weiteren Untersuchungen bestätigt werden.

#### 4.2 Mögliche Verteidigungsmaßnahmen

In dieser Arbeit konnte gezeigt werden, dass mit Hilfe eines HIL-Systems Angriffe entwickelt werden können. Aus diesem Grund sollen mögliche Abwehrmaßnahmen exemplarisch vorgestellt werden.

Wie in dieser Arbeit ebenfalls gezeigt werden konnte, sind verschiedene Systeme unterschiedlich robust gegenüber Angriffen. Aus diesem Grund sollte die Robustheit vor der Inbetriebnahme eingehend evaluiert und verschiedene Systeme miteinander verglichen werden. Um die Robustheit von lernbasierten Systemen zu erhöhen, können Techniken wie das *adversariale Training* [BLZ<sup>+</sup>21] eingesetzt werden, bei dem berechnete Angriffe in die Trainingsdaten zurückgespielt werden, so dass



die Modelle bereits in der Trainingsphase mögliche Angriffe lernen und Daten trotz eines Angriffs noch korrekt klassifizieren.

Außerdem können verschiedene Systeme parallel laufen und miteinander konkurrieren. So können Unsicherheiten ggf. schneller erkannt und entsprechende Sicherheitsmaßnahmen ergriffen werden. Zwei Systeme können versuchen, das gleiche Problem zu lösen, z.B. die aktuelle Höchstgeschwindigkeit zu erkennen, aber dabei unterschiedliche Ansätze verwenden. Beispielsweise können eine datenbankgestützte Lösung und ein visuelles Fahrerassistenzsystem miteinander konkurrieren.

Dies sind einige beispielhafte Maßnahmen zur Abwehr von Angriffen. Um diese Abwehrmaßnahmen entwickeln und evaluieren zu können, müssen jedoch zunächst die möglichen Angriffsvektoren bekannt sein. Diese Arbeit hat gezeigt, wie Hardware-in-the-Loop-Systeme zu diesen Untersuchungen beitragen können.

## 5 Fazit und Ausblick

In dieser Bachelorarbeit wurde untersucht, wie Hardware-in-the-Loop-Systeme zur Bewertung der IT-Sicherheit von visuellen Fahrassistenzsystemen eingesetzt werden können. Ziel der Untersuchungen war es, mit Hilfe eines selbstentwickelten Hardware-in-the-Loop-Systems Angriffe auf visuelle Fahrassistenzsysteme zu testen und herauszufinden, ob sich daraus reale Angriffsvektoren ableiten lassen. Daraus sollten Rückschlüsse auf die Angreifbarkeit der Systeme gezogen werden. Zum besseren Verständnis wurde zu Beginn der Arbeit in die Thematik eingeführt, indem relevante theoretische Grundlagen aufgezeigt wurden.

Anschließend wurden die durchgeführten Versuche vorgestellt und deren Ergebnisse präsentiert und diskutiert. Zunächst konnte anhand der Klassifikation von nicht manipulierten Verkehrsschildern gezeigt werden, dass HIL-Systeme zur Evaluierung der Klassifikationsgenauigkeit von Fahrassistenzsystemen geeignet sind und reproduzierbare Ergebnisse unter leicht veränderten Umgebungsbedingungen liefern können. Mit der Methode *SmoothGrad* konnten adversariale Beispiele berechnet und mit Hilfe des HIL-Systems als Eingabe der Klassifikationssysteme präsentiert werden. Insgesamt wurden 21.600 Angriffe getestet, um die Robustheit der Systeme gegenüber Manipulationen zu untersuchen.

Es zeigte sich, dass die Systeme unterschiedlich robust gegenüber Angriffen sind und dass verschiedene Angriffe unterschiedlich effektiv sind. Die Ergebnisse stellen dar, dass ein Teil der neuronalen Netze und das kommerzielle System leicht zu Fehlklassifikationen verleitet werden konnten. Da die Systeme jedoch unterschiedlich komplex aufgebaut waren, ist ein direkter Vergleich zwischen den neuronalen Netzen und dem kommerziellen System nur eingeschränkt möglich.

Bei den Angriffen wurde deutlich, dass die von den selbsttrainierten Netzen entwickelten Angriffe erfolgreicher waren als die anderen Angriffe. Dies liegt möglicherweise an den Anordnungen der veränderten Pixel, die bei den selbsttrainierten Netzen durch größere zusammenhängende Flächen gekennzeichnet sind. Neben der Anordnung ist auch der Anteil der veränderten Pixel für eine Fehlklassifikation relevant. Je höher der Anteil manipulierter Pixel ist, desto wahrscheinlicher ist eine Fehlklassifikation. Darüber hinaus lassen die Ergebnisse vermuten, dass bestimmte Verkehrszeichen besser geeignet sind, um einen erfolgreichen Angriff auf ein zugehöriges Schild zu berechnen. So konnten insbesondere 30 km/h-Schilder für einen erfolgreichen Angriff genutzt werden.

Weiterhin konnten die Untersuchungen nicht zeigen, dass kombinierte Angriffe einen größeren Erfolg versprechen als Angriffe, die nur mit einem einzelnen neuronalen Netz berechnet wurden. Es ist daher unklar, ob eine Kombination von Angriffen zu einer besseren Verallgemeinerung von Angriffen beiträgt.

Schließlich konnte gezeigt werden, dass mit Hilfe eines HIL-Systems Angriffe berechnet werden können, die sowohl in der simulierten Umgebung als auch übertragen auf ein reales Schild zu einer Fehlklassifikation führen.

Diese Ergebnisse lassen den Schluss zu, dass Hardware-in-the-Loop-Systeme ge-

eignet sind, um die IT-Sicherheit von visuellen Verkehrszeichenerkennungssystemen zu evaluieren. Als Grundlage für die Untersuchung weiterer Systeme kann nun dieser exemplarische Ansatz genutzt werden.

## **Ausblick**

In dieser Bachelorarbeit wurden einige Aspekte aufgezeigt, die in zukünftigen Arbeiten untersucht werden können. Da der grundsätzliche Nutzen von HIL-Systemen gezeigt werden konnte, könnte die Untersuchung der Übertragbarkeit und der Unterschiede zu realen Fahrsituationen ein wichtiger Schritt für eine detailliertere Betrachtung des Nutzens sein.

Darüber hinaus könnte das HIL-System erweitert werden, um genauere Ergebnisse zu erzielen. Dazu könnte der Datensatz der getesteten Schilder erweitert werden, indem die Anzahl der Bilder erhöht und die Unterschiede zwischen den Schildern stärker variiert werden. Außerdem könnten nicht nur statische Fotografien von Verkehrszeichen, sondern auch Videos von Vorbeifahrten an Verkehrszeichen verwendet werden.

Ein weiterer Aspekt zukünftiger Forschung könnte die Untersuchung der verwendeten Klassifikationsverfahren sein. Die verwendeten Techniken könnten sich stärker voneinander unterscheiden, um verschiedene Merkmale besser vergleichen zu können. Unterschiedliche Trainingsdatensätze oder unterschiedliche Methoden aus dem Bereich der künstlichen Intelligenz wären zwei zu nennende Aspekte. Darüber hinaus können weitere Systeme wie z.B. Detektionssysteme implementiert und getestet werden.

Ein wichtiger zukünftiger Forschungsgegenstand ist auch die Untersuchung möglicher Abwehrmaßnahmen gegen die gefundenen Angriffsvektoren. Dabei spielt sowohl die Verbesserung der Robustheit einzelner Systeme als auch die Kombination verschiedener Systeme (z.B. datenbankbasiert und KI-gestützt) eine wichtige Rolle.

Darüber hinaus könnte in Folgearbeiten die Eignung von HIL-Systemen zum Test weiterer (visueller) Fahrerassistenzsysteme untersucht werden, um die Sicherheit des (teil-)automatisierten Fahrens zu erhöhen.

## Literatur

- [Ale20] Johannes Alecke. Analyse und optimierung von angriffen auf tiefe neuronale netze, 2020.
- [AMAZ17] Saad Albawi, Tareq Abed Mohammed, and Saad Al-Zawi. Understanding of a convolutional neural network. In *2017 international conference on engineering and technology (ICET)*, pages 1–6. Ieee, 2017.
- [AO03] Masami Aga and Akio Okada. Analysis of vehicle stability control (vsc)’s effectiveness from accident data. In *Proceedings: International Technical Conference on the Enhanced Safety of Vehicles*, volume 2003, pages 7–p, 2003.
- [B<sup>+</sup>95] Christopher M Bishop et al. *Neural networks for pattern recognition*. Oxford university press, 1995.
- [Bac05] M. Bacic. On hardware-in-the-loop simulation. In *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005.
- [BDF<sup>+</sup>14] Klaus Bengler, Klaus Dietmayer, Berthold Farber, Markus Maurer, Christoph Stiller, and Hermann Winner. Three decades of driver assistance systems: Review and future perspectives. *IEEE Intelligent Transportation Systems Magazine*, 6(4):6–22, 2014.
- [Ben21] Benjamin Bohleber. Entwicklung eines graphischen analyse- und auswerteframeworks für sensordaten beim automatisierten fahren, 2021.
- [Bis94] Chris M. Bishop. Neural networks and their applications. *Review of Scientific Instruments*, 65(6):1803–1832, 1994.
- [BLZ<sup>+</sup>21] Tao Bai, Jinqi Luo, Jun Zhao, Bihan Wen, and Qian Wang. Recent advances in adversarial training for adversarial robustness. *arXiv preprint arXiv:2102.01356*, 2021.
- [BNJT10] Marco Barreno, Blaine Nelson, Anthony D Joseph, and J Doug Tygar. The security of machine learning. *Machine Learning*, 81(2):121–148, 2010.
- [BNL] Battista Biggio, Blaine Nelson, and Pavel Laskov. Poisoning attacks against support vector machines.
- [BNS<sup>+</sup>06] Marco Barreno, Blaine Nelson, Russell Sears, Anthony D Joseph, and J Doug Tygar. Can machine learning be secure? In *Proceedings of the 2006 ACM Symposium on Information, computer and communications security*, pages 16–25, 2006.

- [CMMS11] Dan Ciresan, Ueli Meier, Jonathan Masci, and Jurgen Schmidhuber. A committee of neural networks for traffic sign classification. In *The 2011 International Joint Conference on Neural Networks*, pages 1918–1921. IEEE, 7/31/2011 - 8/5/2011.
- [CW17] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 IEEE Symposium on Security and Privacy (SP)*, pages 39–57. Ieee, 2017.
- [DDS<sup>+</sup>09] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.
- [dLMSA97] A. de La Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol. Road traffic sign detection and classification. *IEEE Transactions on Industrial Electronics*, 44(6):848–859, 1997.
- [Eur19] Europäische Kommission. Verordnung 2019/2144, 27. November 2019.
- [FV19] Ruth Fong and Andrea Vedaldi. Explanations for attributing deep neural network predictions. In *Explainable AI: Interpreting, Explaining and Visualizing Deep Learning*, volume 11700, pages 149–167. Springer, Cham, 2019.
- [GBC16] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning. Adaptive computation and machine learning*. The MIT Press, Cambridge, Massachusetts, 2016.
- [GDGG] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain.
- [GDLG17] Tianmei Guo, Jiwen Dong, Henjian Li, and Yunxing Gao. Simple convolutional neural network on image classification. In *2017 IEEE 2nd International Conference on Big Data Analysis (ICBDA)*, pages 721–724. IEEE, 2017.
- [GLSG09] David Geronimo, Antonio M Lopez, Angel D Sappa, and Thorsten Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE transactions on pattern analysis and machine intelligence*, 32(7):1239–1258, 2009.
- [Gmb22] Tesla Germany GmbH. Autopilot, 12/12/2022.
- [GSS] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples.

- [GWK<sup>+</sup>18] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, Gang Wang, Jianfei Cai, and Tsuhan Chen. Recent advances in convolutional neural networks. *Pattern Recognition*, 77:354–377, 2018.
- [HK19] Michael Haenlein and Andreas Kaplan. A brief history of artificial intelligence: On the past, present, and future of artificial intelligence. *California Management Review*, 61(4):5–14, 2019.
- [HSS<sup>+</sup>13] Sebastian Houben, Johannes Stallkamp, Jan Salmen, Marc Schlipsing, and Christian Igel. Detection of traffic signs in real-world images: The german traffic sign detection benchmark. In *The 2013 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 8/4/2013 - 8/9/2013.
- [HT17] Pavel Hamet and Johanne Tremblay. Artificial intelligence in medicine. *Metabolism*, 69:S36–S40, 2017.
- [Int18] International Organization for Standardization. Information technology — security techniques — information security management systems — overview and vocabulary, 01.02.2018.
- [JMBR21] Raphael Joud, Pierre-Alain Moellic, Remi Bernhard, and Jean-Baptiste Rigaud. A review of confidentiality threats against embedded neural network models. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, pages 610–615, 2021.
- [KO11] Bekir Karlik and A. Vehbi Olgac. Performance analysis of various activation functions in generalized mlp architectures of neural networks. *International Journal of Artificial Intelligence and Expert Systems*, 1(4):111–122, 2011.
- [Mat18] Matthias Lemm. <https://pixabay.com/de/photos/landstra%c3%9fe-allee-landschaft-natur-3620378/>, 2018.
- [mic13] micowelt. <https://pixabay.com/de/photos/magdeburg-deutschland-stra%c3%9fe-szene-203012/>, 2013.
- [Mon17] Monsterkoi. <https://pixabay.com/de/photos/stra%c3%9fe-berge-h%c3%bcgel-leitpfosten-2806371/>, 2017.
- [NS11] Mirko Nentwig and Marc Stamminger. Hardware-in-the-loop testing of computer vision based driver assistance systems. In *2011 IEEE Intelligent vehicles symposium (IV)*, pages 339–344. IEEE, 2011.
- [OBLS14] Maxime Oquab, Leon Bottou, Ivan Laptev, and Josef Sivic. Learning and transferring mid-level image representations using convolutional

- neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [PMG] Nicolas Papernot, Patrick McDaniel, and Ian Goodfellow. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples.
- [PMG<sup>+</sup>17] Nicolas Papernot, Patrick McDaniel, Ian Goodfellow, Somesh Jha, Z Berkay Celik, and Ananthram Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.
- [PMJ<sup>+</sup>16] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.
- [PY10] Sinno Jialin Pan and Qiang Yang. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, 22(10):1345–1359, 2010.
- [SAE21] SAE. Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles, 30.04.2021.
- [SB17] Yassmina Saadna and Ali Behloul. An overview of traffic sign detection and classification methods. *International Journal of Multimedia Information Retrieval*, 6(3):193–210, 2017.
- [SHZ<sup>+</sup>18] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018.
- [SSSI11] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *The 2011 international joint conference on neural networks*, pages 1453–1460. IEEE, 2011.
- [STK<sup>+</sup>17] Daniel Smilkov, Nikhil Thorat, Been Kim, Fernanda Viégas, and Martin Wattenberg. Smoothgrad: removing noise by adding noise. *arXiv preprint arXiv:1706.03825*, 2017.
- [SVI<sup>+</sup>] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision.
- [SVS19] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

- [SZS<sup>+</sup>] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks.
- [Tes22] Tesla. Tesla vision update: Replacing ultrasonic sensors with tesla vision | tesla support, 12/12/2022.
- [THBM19] Wang Tong, Azhar Hussain, Wang Xi Bo, and Sabita Maharjan. Artificial intelligence for vehicle-to-everything: A survey. *IEEE Access*, 7:10823–10843, 2019.
- [Tur09] Alan M. Turing. Computing machinery and intelligence. In *Parsing the Turing Test*, pages 23–65. Springer, Dordrecht, 2009.
- [WX] Rey Wiyatno and Anqi Xu. Maximal jacobian-based saliency map attack.
- [XGD<sup>+</sup>] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks.



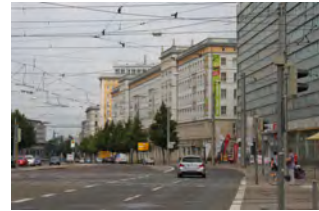
## 6 Anhang



(a) Landstraße 1 [Mat18]



(b) Landstraße 2 [Mon17]



(c) Stadt [mic13]

Abbildung 15: Hintergrundbilder, in denen in Versuchsphase 1 und 2 die Schilder eingebettet wurden.



Abbildung 16: Schilder, die in Phase 1 von mindestens einem System falsch klassifiziert wurden.

	CNN 1	CNN 2	CNN 3	CNN 4	CNN 5	KS
CNN 1	243   117	257   103	150   210	242   118	254   106	130   230
CNN 2	297   63	313   47	216   144	307   53	303   57	238   122
CNN 3	308   52	303   57	221   139	318   42	320   40	225   135
CNN 4	247   113	205   155	111   249	171   189	178   182	62   298
CNN 5	269   91	245   115	109   251	176   184	152   208	96   264
CNN 1   CNN 2	241   119	277   83	151   209	248   112	264   96	153   207
CNN 1   CNN 3	264   96	272   88	171   189	270   90	289   71	166   194
CNN 2   CNN 3	281   79	294   66	197   163	290   70	305   55	215   145
CNN 4   CNN 5	265   95	213   147	105   255	177   183	154   206	87   273
CNN 1   CNN 2   CNN 3	239   121	268   92	167   193	253   107	268   92	160   200
Gesamt	2654   946	2647   953	1598   2002	2452   1148	2487   1113	1532   2068

Tabelle 5: Übersicht über die richtigen und falschen Klassifikationen. In den Zeilen sind die Kombinationen der neuronalen Netze, mit denen die Angriffe berechnet wurden, dargestellt und in den Spalten die Klassifikationssysteme. Die Schreibweise X | Y einer Zelle bedeutet, dass X Elemente der Klassifikation der richtigen Klasse und Y Elemente der falschen Klasse zugeordnet wurden.

	CNN 1	CNN 2	CNN 3	CNN 4	CNN 5	KS
30 km/h	22   78	23   77	7   93	29   71	21   79	38   62
50 km/h	383   567	698   252	431   519	517   433	701   249	391   559
60 km/h	56   44	68   32	57   43	62   38	56   44	30   70
70 km/h	1855   145	1648   352	849   1151	1560   440	1458   542	911   1089
80 km/h	141   9	94   56	149   1	106   44	74   76	35   115
100 km/h	129   21	107   43	62   88	125   25	120   30	91   59
120 km/h	68   82	9   141	43   107	53   97	57   93	36   114
Gesamt	2654   946	2647   953	1598   2002	2452   1148	2487   1113	1532   2068

Tabelle 6: Übersicht über die richtigen und falschen Klassifikationen. In den Zeilen sind die verschiedenen Verkehrszeichen dargestellt und in den Spalten die Klassifikationssysteme. Die Schreibweise X|Y einer Zeile bedeutet, dass X Elemente der Klassifikation der richtigen Klasse und Y Elemente der falschen Klasse zugeordnet wurden.

	1 %	2 %	3 %	4 %	5 %
CNN 1	368   64	291   141	239   193	201   231	177   255
CNN 2	405   27	368   64	333   99	303   129	265   167
CNN 3	404   28	375   57	327   105	308   124	281   151
CNN 4	327   105	239   193	167   265	128   304	113   319
CNN 5	289   143	233   199	201   231	184   248	140   292
CNN 1   CNN 2	382   50	310   122	248   184	211   221	183   249
CNN 1   CNN 3	386   46	329   103	274   158	239   193	204   228
CNN 2   CNN 3	395   37	354   78	306   126	276   156	251   181
CNN 4   CNN 5	291   141	235   197	188   244	153   279	134   298
CNN 1   CNN 2   CNN 3	375   57	314   118	264   168	222   210	180   252
Gesamt	3622   698	3048   1272	2547   1773	2225   2095	1928   2392

Tabelle 7: Übersicht über die richtigen und falschen Klassifikationen. In den Zeilen sind die Kombinationen der neuronalen Netze, mit denen die Angriffe berechnet wurden, dargestellt und in den Spalten der Anteil veränderter Pixel. Die Schreibweise X | Y einer Zelle bedeutet, dass X Elemente der Klassifikation der richtigen Klasse und Y Elemente der falschen Klasse zugeordnet wurden.

	1 %	2 %	3 %	4 %	5 %
30 km/h	66   54	42   78	20   100	8   112	4   116
50 km/h	950   190	746   394	566   574	463   677	396   744
60 km/h	96   24	73   47	61   59	54   66	45   75
70 km/h	2055   345	1811   589	1628   772	1480   920	1307   1093
80 km/h	156   24	145   35	112   68	100   80	86   94
100 km/h	179   1	157   23	125   55	99   81	74   106
120 km/h	120   60	74   106	35   145	21   159	16   164
Gesamt	3622   698	3048   1272	2547   1773	2225   2095	1928   2392

Tabelle 8: Übersicht über die richtigen und falschen Klassifikationen. In den Zeilen sind die verschiedenen Verkehrszeichen dargestellt und in den Spalten der Anteil veränderter Pixel. Die Schreibweise X | Y einer Zeile bedeutet, dass X Elemente der Klassifikation der richtigen Klasse und Y Elemente der falschen Klasse zugeordnet wurden.

	1 %	2 %	3 %	4 %	5 %
CCNN 1	656   64	576   144	506   214	481   239	435   285
CNN 2	657   63	597   123	510   210	458   262	425   295
CNN 3	550   170	395   325	280   440	212   508	161   559
CNN 4	599   121	540   180	488   232	438   282	387   333
CNN 5	610   110	552   168	489   231	452   268	384   336
KS	550   170	388   332	274   446	184   536	136   584
Gesamt	3622   698	3048   1272	2547   1773	2225   2095	1928   2392

Tabelle 9: Übersicht über die richtigen und falschen Klassifikationen. In den Zeilen sind die verschiedenen Klassifikationssysteme dargestellt und in den Spalten der Anteil veränderter Pixel. Die Schreibweise X | Y einer Zeile bedeutet, dass X Elemente der Klassifikation der richtigen Klasse und Y Elemente der falschen Klasse zugeordnet wurden.

Angriffskombination	Anteil	Verkehrszeichen	Bild	Angriff
CNN 5	1 %	30 km/h	0	Angriff 1
CNN 4   CNN 5	1 %	30 km/h	0	Angriff 2
CNN 4	1 %	30 km/h	1	Angriff 3
CNN 5	1 %	30 km/h	1	Angriff 4
CNN 4   CNN 5	1 %	30 km/h	1	Angriff 5
CNN 1	1 %	50 km/h	1	-
CNN 4	1 %	50 km/h	1	-
CNN 1   CNN 3	1 %	50 km/h	1	-
CNN 4   CNN 5	1 %	50 km/h	1	-
CNN 1   CNN 2   CNN 3	1 %	50 km/h	1	-
CNN 1   CNN 2	1 %	120 km/h	2	-
CNN 1   CNN 2   CNN 3	1 %	120 km/h	2	-

Tabelle 10: Übersicht der Angriffe, bei denen 1 % der Pixel verändert wurden und alle Systeme das Schild falsch klassifiziert haben. Die ersten fünf Bilder wurden im nicht manipuliertem Zustand korrekt klassifiziert. Aus diesen Bildern wurde in Phase 3 ein Angriff abgeleitet (Angriff 1 - 5).