

# Exposé zur Masterarbeit

## Erkennen von Fahrradfahrern auf Kamerabildern mit einem künstlichen neuronalen Netzwerk in Echtzeit

Alexander Wachtberger

29. August 2018

Automobilhersteller prognostizieren die ersten fahrerlosen Fahrzeuge für das Jahr 2030. Fahrerlos bedeutet, dass das Fahrzeug ein Ziel erhält, autonom den Weg dorthin findet und schließlich auch parkt. Solche Fahrzeuge werden keine Lenkräder oder Gaspedale besitzen und menschliche Insassen sind nur Passagiere. Damit ein Auto ohne Fahrer vom Startpunkt bis zum Zielpunkt gelangt, benötigt es Wissen. Angefangen bei „wo befindet sich die Straße bzw. die korrekte Fahrbahn?“, über „welche Verkehrsregeln gelten aktuell?“ bis hin zu „wann bremsen ich wie stark ab?“ und „wo ist ein Parkplatz?“. Um die genannten Fragen zu beantworten, können verschiedenste Methoden und Technologien gewählt werden. Die Verwendung einer Kamera, welche fortlaufend Bilder der aktuellen Szene aufnimmt, ist eine Möglichkeit, Informationen zu sammeln. Im Anschluss müssen diese Informationen eingeordnet und ausgewertet werden, um komplexe Verkehrsszenarien autonom lösen zu können. Verkehrsszenarien sind allerdings so vielfältig, dass sich nicht für jedes Szenario eine Lösung bereitstellen lässt. Vielmehr müssen aus dem Kontext heraus Entscheidungen getroffen werden – so wie es auch der Mensch im Verkehr tut. Dies ist nur auf Basis vollständiger Umgebungsdaten möglich um darauf beruhend ein Verständnis über das aktuelle Verkehrsszenario zu erhalten, wie Haist beschreibt [Hai16]. Ein wesentlicher Teil davon liegt in der Erkennung und Klassifizierung von Objekten im aktuellen Verkehrsszenario.

Nach dem Verband der Automobilindustrie gibt es eine Klassifizierung der Automatisierungsgrade des automatisierten Fahrens in sechs Stufen [Aut15]. Seit Jahren findet die Stufe eins, dass assistierte Fahren gebrauch. Sehr wenige Serienautos bieten teilautomatisiertes Fahren an, bei denen der Fahrer das System dauerhaft überwachen muss. Stufe drei beschreibt das hochautomatisierte Fahren, bei dem der Fahrer das System nicht mehr dauerhaft überwachen muss aber gegebenenfalls die Steuerung übernehmen muss. Hochautomatisiertes Fahren ist kein Zukunftsszenario mehr, sondern bereits Realität: Insbesondere der Automobilhersteller Audi AG bietet mit seinem Audi A8 ein Paradebeispiel – laut Hersteller das erste Serienauto auf dieser Automatisierungsstufe, allerdings gilt der dringende Hinweis, das autonome Fahrsystem nur auf Autobahnen und baulich abgetrennten Straßen zu verwenden. Das System nennt sich AI Staupilot und ist derzeit in Europa noch nicht gesetzlich zugelassen. Ein weiterer Hersteller, welcher sich dem Thema automatisiertes Fahren angenommen hat, ist Tesla. Tesla bietet ein Fahrerassistenzsystem namens Autopilot, allerdings ist das nur ein System der Stufe zwei welches ständig überwacht werden muss: dieses kann die Spur und die Geschwindigkeit halten, vorausfahrende Fahrzeuge werden erkannt und ein einstellbarer Abstand wird eingehalten. Außerdem klassifiziert es PKWs, LKWs und Motorräder. Fahrräder können allerdings weder klassifiziert noch erkannt werden. Weiter gibt es vereinzelt zugelassene autonome Fahrzeuge anderer Hersteller, wie das Googlecar oder von der Firma Uber nachgerüstete Serienautos. Grundsätzlich ist hochautomatisiertes Fahren also ein sehr präsent und aktuelles Thema in der Forschung unter Automobilherstellern, gleichermaßen etabliert sich das Thema auch immer mehr in der Gesellschaft: in Kalifornien ist es beispielsweise seit April 2018 gesetzlich erlaubt, Fahrzeuge ohne Lenkrad und Pedale zu betreiben. Ein wichtiger Schwerpunkt in der aktuellen Automatisierungs-Forschung liegt auf dem Erkennen und Klassifizieren aller Objekte in Verkehrsszenarien. Dazu zählen Tiere (Wild, Hunde, Hasen, Vögel, usw.), Hindernisse (Bäume, Absperrungen, Pfeiler, Fahrbahnbegrenzung, usw.), Verkehrsteilnehmer (Trikes, Segways, Fahrräder, Skater, usw.) und auch sonstige Objekte (Plastiktüten, Heuballen, vom Laster fallende Gegenstände, usw.).

Im Jahr 2016 wurden vom ADAC Notbremsassistenten führender Automobilhersteller getestet [Rat16]. Das Ergebnis in Bezug auf das Erkennen von Fahrradfahrern ist leider immer noch mangelhaft: die meisten Fahrradfahrer werden vom Notbremsassistenten der jeweiligen Automobilhersteller überhaupt nicht erkannt und ohne Geschwindigkeitsreduktion überfahren. Der Audi A4 hat mit einem Erfüllungsgrad von 50% noch die beste Bewertung bei diesem Testszenario. Der Volvo V60 fällt komplett durch, obwohl er nach Herstellerangaben ein Notbremsassistent mit automatischer Fußgänger- und Fahrradfahrer-Erkennung besitzt. Für die Nutzung außerhalb von Autobahnen ist hochautomatisiertes Fahren aktuell also noch nicht geeignet. Dabei gewinnt das Thema Fahrradfahrererkennung immer mehr an Relevanz: die Anzahl der schwerverletzten Unfallbeteiligten Fahrradfahrer in Berlin stieg von 480 in 2001 auf 681 in 2017 - ein Anstieg um 141% und es war „jeder 4. Verkehrstote bei einem Verkehrsunfall mit Radfahrerbeteiligung zu beklagen“ [Ber17]. Radfahrer haben in Berlin über 80% ihrer Unfälle mit Kfz [Fah18]. Mit einer zuverlässigen Fahrradfahrererkennung in Serienautos können diese Zahlen erheblich reduziert werden.

Die Klassifizierung und Lokalisierung von Objekten mittels handgeschriebenen Algorithmen des maschinellen Sehens wurden im Jahr 2012 von Deep Learning Algorithmen abgelöst als Geoffrey Hinton et al. ein künstliches neuronales Netzwerk trainierte und damit deutlich bessere Fehlerraten erzielte [KSH12]. Das neuronale Netzwerk wurde später als AlexNet bekannt; und obwohl dieses mittlerweile nicht mehr aktuell ist, wird es als wesentliche Grundlage für neue Forschungen verwendet. Die Theorie für diesen Durchbruch gab es bereits 1986 [RHW86]. Deep Learning ist ein Bereich des maschinellen Lernens. Dabei wird eine Reihe von Techniken zur automatischen Erkennung und Wiederverwendung von Merkmalen verwendet und mit einem künstlichen neuronalen Netzwerk umgesetzt. Diese Technik erzielt zurzeit in vielen Bereichen Durchbrüche, weil Probleme für die trainiert wurde, genauer und schneller gelöst werden als es Menschen vermögen. Sei es ein künstlicher Go-Spieler, Erkennung verschiedener Krankheiten in der Medizintechnik oder das Erkennen zu welcher Baumart ein Stück Baumrinde auf einem Foto gehört. So wird auch in der Automobilindustrie stark auf Deep Learning gesetzt, um verschiedene Probleme zu lösen.

Im Rahmen dieser Arbeit soll der Fragestellung nachgegangen werden, inwiefern mit quelloffener Software für Deep Learning zuverlässig Fahrradfahrer auf Kamerabildern in Sekundenbruchteilen klassifiziert werden können – insbesondere auch in Hinblick auf unterschiedliche, reale Verkehrsszenarien. Die angenommene Hypothese ist, dass ein antrainiertes neuronales Netzwerk schnell gute Ergebnisse beim Erkennen von Radfahrern auf Kamerabildern erzielt. Um die nachfolgend genannten Zielvorgaben zu erreichen, muss das System im Nachgang aber feinjustiert werden.

Ein zuverlässiges Erkennen von Fahrrädern bedeutet im Straßenverkehr und insbesondere in Hinsicht auf autonom agierende Fahrzeuge, dass die Software Fahrradfahrer mindestens so gut erkennt, wie es Menschen tun würden. Veranschlagt wird eine Erkennungsrate von 99,99% - nicht erkannte Fahrräder lassen sich z. B. einer sehr weiten Entfernung zurechnen oder einem fast vollständig verdecktem Fahrrad. Zusätzlich muss die Fehlerrate, also das Erkennen von Fahrrädern, obwohl keines vorhanden ist, sehr gering sein. Ein beispielhaftes Problem das hierbei entstehen kann ist, dass der Notbremsassistent aktiviert wird und unvermittelt eine Vollbremsung durchführt, wenn fälschlicherweise ein Fahrrad auf Kollisionskurs vor dem Auto erkannt wird. Das kann unangenehm für die Insassen sein, zu Auffahrunfällen führen und weitere Verkehrsteilnehmer in Gefahr bringen. Mit einer maximalen Fehlerrate von 1% wären das bei angenommenen 50 Bildern pro Sekunde ein Fehler in zwei Sekunden, dies wird als annehmbar hingenommen. Fahrräder gibt es in vielen verschiedenen Arten und Formen. Im Rahmen dieser Arbeit beziehe ich mich lediglich auf Fahrräder für eine Person mit zwei Rädern in den gängigen Größen. Betrachtet werden also auch Mountainbikes, Stadträder, Rennräder, Laufräder, Kinderräder, Trekkingräder, Tourenräder, Crossrad, Cruiser und BMX. Im Rahmen dieser Arbeit werden Tandems, Liegeräder, Einräder, Dreiräder, Vierräder o. ä. nicht betrachtet. 10 000 Bilder mit Fahrrädern aus den betrachteten Kategorien sollen getestet werden. Das Erkennen von Fahrrädern soll unter verschiedensten realen Verkehrsszenarien getestet werden. Der Algorithmus muss in Sekundenbruchteilen arbeiten, weil zum Einen mehrere Bilder in einer Sekunde ausgewertet werden sollen und zum Anderen, nachgelagerte Systeme die aktuellen Informationen benötigen.

Das für diese Arbeit gewählte neuronale Netzwerk, welches mit Deep Learning trainiert wird, ist ein antrainiertes Convolutional Neural Network (CNN) namens YOLO. YOLO ist ein Akronym für „You Only Look Once“ und liegt in Version drei vor [RF18]. Das verwendete YOLO System weist eine höhere Fehlerrate bei der Lokalisierung als andere verfügbare Netzwerke auf [RF17].

Die Fahrräder werden klassifiziert und bestmöglich lokalisiert, um Entscheidungen für z. B. einen Notbremsassistenten einfacher zu gestalten. Das Problem der Lokalisierung kann mit zusätzlichen Algorithmen oder Sensorfusion behoben werden und wird nicht im Rahmen dieser Arbeit behandelt.

Hauptsächlich wird der klassische Ansatz angewendet die letzte Schicht des neuronalen Netzwerks, die für die Klassifizierung zuständig ist, auszutauschen um nur Radfahrer zu erkennen. Als Orientierung für diese Arbeit dient die Arbeit „Analysis and Improvement of the Visual Object Detection Pipeline“ von Jan Hosang [Hos17]. Hosangs Arbeit hat als Fokus die herkömmliche Unterdrückung nicht-maximaler Detektionen beim Deep Learning und wird anhand von Fußgängererkennung untersucht. Um ein neuronales Netzwerk zu testen und zu debuggen beschreiben Odena et al. einen Ansatz [OG18]. Ein evolutionärer Ansatz zur Verbesserung eines neuronalen Netzwerks ist das Verwenden von Schichten anderer antrainierter Netzwerke nach Yosinski et al. [Yos+14].

Im folgenden wird zusammenfassend beschrieben wie vorgegangen wird um die Forschungsfrage zu klären und welche Werkzeuge dafür verwendet werden. Zunächst muss ein ausreichend großer Datensatz zum Trainieren und zum Testen zur Verfügung stehen. Dieser wird aus Internet Datenbanken mit teilweise gelabelten Bildern und selbst aufgenommen Bildern zusammengestellt. Ein antrainiertes CNN kann verwendet werden um automatisiert Bilder herunterzuladen, zu prüfen ob ein Radfahrer abgebildet ist und sie in den Datensatz einzupflegen. Im ersten Schritt werden von YOLO erkannte Fahrräder und Menschen zu Radfahrern kombiniert und als solche klassifiziert. Damit wird die erste Testreihe durchlaufen. Danach wird die letzte Schicht des YOLO Systems entfernt und eine neue untrainierte Schicht eingeführt um ausschließlich Radfahrer zu erkennen. Dieses neuronale Netzwerk wird trainiert und anschließend eine weitere Testreihe durchgeführt. Anhand der Ergebnisse werden die erfolgversprechendsten Ansätze von Hosang, Odena et al. oder Yosinski et al. verwendet werden um das neuronale Netzwerk zu verbessern. Die Ansätze können auch mit klassischen Methoden zur Verbesserung von neuronalen Netzwerken kombiniert werden. Darunter fällt ein längeres Training, ändern der Batch Größe, die Bias Varianz überprüfen, usw.. Nach jeder Verbesserung folgt eine Testreihe.

Für diese Arbeit wird das Framework Darknet von Joseph Redmon verwendet [Red16]. Abgesehen davon, dass es vollständig quelloffen ist, verbraucht es weniger Speicher und ist um ein vielfaches schneller als TensorFlow (ein gängiges Framework von Google). Darknet ist in C geschrieben und kann mit CUDA (GPU Unterstützung) und openCV (quelloffenes Framework für Bildverarbeitung) kompiliert werden um schneller zu arbeiten. Das quelloffene YOLOv3 wird als Basis CNN verwendet. Der Algorithmus soll letztendlich auf einem Modellauto, mit montierter Kamera implementiert werden. Das Modellauto kann dann durch Fahren auf dem Bürgersteig den Algorithmus testen. Dabei werden verschiedene Tageszeiten, also Lichteinstrahlungswinkel und Lichtstärke beachtet. Die teilweise Verdeckung von Fahrrädern muss ggf. künstlich nachgestellt werden, um sie zu testen. Die Geschwindigkeit ist ebenfalls ein Aspekt, der differenziert betrachtet werden soll, allerdings ist hier ggf. die Kamera in ihrer Leistung limitiert. Als Modellauto bietet sich das VeloxCar an. Das VeloxCar ist ein Projekt zum Erforschen und Entwickeln von fahrerlosen Fahrzeugen. Es ist Teil des Projekts Collaborative Embedded Systems (CREST), welches vom Bundesministerium für Bildung und Forschung gefördert wird. Das Ziel von CREST ist ein umfassendes Rahmenwerk für die Entwicklung kollaborierender eingebetteter Systeme zu schaffen. Das VeloxCar besitzt eine Kamera die nach vorne ausgerichtet ist und die zu verarbeitenden Bilder liefert. Die verarbeitende Steuereinheit ist ein Einplatinencomputer (ODroid) mit ARM Technologie, auf dem das Robot Operating System (ROS) als Betriebssystem arbeitet. Je nachdem wie schnell ein Ergebnis vorliegt ist es auch denkbar den Algorithmus auf einem umgebauten VW zu implementieren. Dieser besitzt vorne und hinten eine Weitwinkel Kamera und ist Teil eines anderen Projekts.

Abschließend werden alle durchgeführten Tests und die verwendeten Technologien und Methoden evaluiert. Das Ergebnis ist ein funktionierender quelloffener Algorithmus, welcher auf günstiger Hardware funktioniert und somit für jeden Automobilhersteller einsetzbar sein sollte.

## Literatur

- [Aut15] Verband der Automobilindustrie. „Automatisierung–Von Fahrerassistenzsystemen zum automatisierten Fahren“. In: *VDA, Berlin* (2015).
- [Ber17] Der Polizeipräsident in Berlin. *Sonderuntersuchung Radfahrverkehrsunfälle in Berlin*. 2017, S. 2. URL: <https://www.berlin.de/polizei/aufgaben/verkehrssicherheit/verkehrsunfallstatistik/>.
- [Fah18] Allgemeiner Deutscher Fahrrad-Club. *ADFC Fakten-Überblick für Fahrradunfälle*. 2018. URL: <https://adfc-berlin.de/radverkehr/sicherheit/information-und-analyse/121-fahrradunfaelle-in-berlin-unfallstatistik/153-adfc-fakten-ueberblick-fuer-fahrradunfaelle.html>.
- [Hai16] Tobias Haist. „Autonomes Fahren: Eine kritische Beurteilung der technischen Realisierbarkeit“. In: (2016).
- [Hos17] Jan Hosang. „Analysis and improvement of the visual object detection pipeline“. In: (2017).
- [KSH12] Alex Krizhevsky, Ilya Sutskever und Geoffrey E Hinton. „Imagenet classification with deep convolutional neural networks“. In: *Advances in neural information processing systems*. 2012, S. 1097–1105.
- [OG18] Augustus Odena und Ian Goodfellow. „TensorFuzz: Debugging Neural Networks with Coverage-Guided Fuzzing“. In: *arXiv preprint arXiv:1807.10875* (2018).
- [Rat16] Uwe Rattay. *Notbremsassistenten mit Fußgänger- und Radfahrer-Erkennung*. 2016. URL: [https://www.adac.de/infotestrat/tests/assistenzsysteme/fussgaengererkennung\\_2016/default.aspx](https://www.adac.de/infotestrat/tests/assistenzsysteme/fussgaengererkennung_2016/default.aspx).
- [Red16] Joseph Redmon. *Darknet: Open Source Neural Networks in C*. <http://pjreddie.com/darknet/>. 2013–2016.
- [RF17] Joseph Redmon und Ali Farhadi. „YOLO9000: better, faster, stronger“. In: *arXiv preprint* (2017).
- [RF18] Joseph Redmon und Ali Farhadi. „YOLOv3: An Incremental Improvement“. In: *arXiv* (2018).
- [RHW86] David E Rumelhart, Geoffrey E Hinton und Ronald J Williams. „Learning representations by back-propagating errors“. In: *nature* 323.6088 (1986), S. 533.
- [Yos+14] Jason Yosinski u. a. „How transferable are features in deep neural networks?“ In: *Advances in neural information processing systems*. 2014, S. 3320–3328.