



Da inzwischen in vielen Städten immer mehr Sensoren und Webcams zur Erfassung von Informationen eingesetzt werden, bieten sich hier neue Möglichkeiten, um Informationen über das Verkehrsaufkommen zu ermitteln. Häufig werden über die installierten Sensoren beispielsweise Temperatur- und CO<sub>2</sub>-Messungen durchgeführt. Durch Kameras kann daneben das Wetter an einem öffentlichen Platz geprüft oder das Verkehrsgeschehen an kritischen Verkehrspunkten beobachtet werden. Die angebrachten Sensoren erfassen so kontinuierlich oder zu bestimmten Zeiten Daten, die zusätzlich noch mit Metainformationen wie Uhrzeit, technischen Details des Sensors oder Informationen über den Standort ausgestattet sind. Durch die internetbasierte Bereitstellung dieser Daten entsteht so ein Netzwerk mit einer großen Anzahl von möglichen Sensoren. Ein so aufkommendes Internet of Things (IoT) bietet damit eine Architektur und Infrastruktur, die eine große Anzahl von Anwendungen in unterschiedlichen Bereich wie Smart Environment, Smart Home, Smart Hospitals, Smart Agriculture aber auch Smart Traffic Systems ermöglicht (vgl. Farooq et al., 2015).

Die IoT-Architektur hat dabei Eigenschaften wie standardisierte Schnittstellen und Protokoll, ist offen, skalierbar und flexibel. Sie kann generell aus einem Anwendungs-, Netzwerk/Service- und Sensoren-Layer bestehen (Chen et al., 2014). Die Verwendung vom IoT im Kontext vom Individualverkehr ist dabei nicht neu. Es kann für Smart Parking in Gebäuden (Gupta et al., 2017), in Smart-Traffic-Management-Systemen (Rabby et al., 2019) oder im Bereich des autonomen Fahrens (Ahmad & Pothuganti, 2020) eingesetzt werden. Für die Messung des Verkehrs eignen sich unterschiedliche Sensoren. Dies können punktuelle Magnet-Sensoren (Cheung et al., 2005) und Induktionsschleifen in der Fahrbahn oder GPS-Sensoren und Radio Frequency Identification (RFID) Tags in Fahrzeugen sein (Bickel et al., 2007). Durch die bereits erwähnte, immer größere werdende Anzahl von Kameras im öffentlichen Bereich, eignen sich diese beispielsweise auch zur Steuerung von Ampelanlagen in einem Kreuzungsbereich (vgl. Zheng et al., 2013) oder zur Messung des Verkehrsflusses.

Beymer et al. (1997) zeigten wie ein Computer-Vision-System zur Messung des Verkehrs in Echtzeit aussehen könnte. Neben dem Ansatz der Erfassung des Verkehrs basierend auf einem 3D-Modell und der Blob- oder Kontur-Erkennung, untersuchten sie einen Feature-basierten Ansatz. Letzteres verwendeten auch Uke & Thool (2013), um Fahrzeuge mithilfe der OpenCV-Bibliothek zu erkennen und dadurch das Verkehrsaufkommen zu bestimmen. Bei OpenCV handelt es sich um eine freie und sehr umfangreiche Programmbibliothek im Bereich von Computer Vision und Machine Learning (vgl. Bradski 2000). Es existieren einige Studien, die den Einsatz im Bereich der Messung des Verkehrsflusses von Fahrzeugen (siehe Lei et al., 2008, Uke & Thool, 2013, Li et al., 2014), Fahrrädern oder Personen (siehe Ghosh et al., 2019, Zangenehpour et al., 2015, Shubho et al., 2021) zeigen. Bei der Objekterkennung zur Ermittlung des Verkehrs wird häufig der „You Only Look Once“-Algorithmus (YOLO) verwendet, welcher in verschiedenen Versionen zur Verfügung steht (vgl. Jiang et al., 2022). Je nach Einsatzzweck existieren spezielle Versionen für die Echtzeiterkennung der verschiedenen Verkehrsteilnehmer (siehe u. a. Ćorović et al., 2018; Lan et al., 2018; Sang et al. 2018; Shubho et al., 2021).

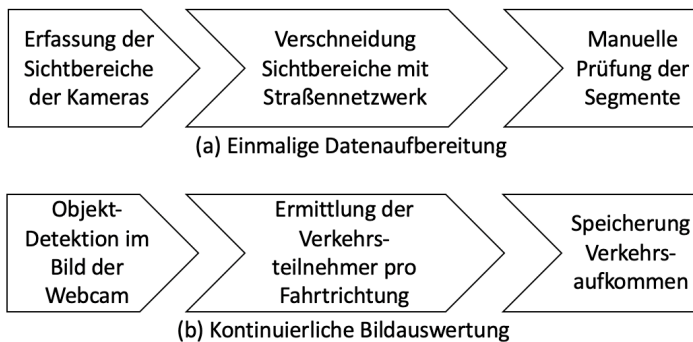
Bei den erwähnten Studien standen stets die Erkennung und Messung des Verkehrsflusses mittels Computer Vision im Vordergrund. Auf eine dauerhafte Speicherung und eine offene und standardisierte Bereitstellung der erfassten Daten wurde nicht eingegangen. In diesem Beitrag wird gezeigt, wie unter Verwendung des IoT der Verkehrsfluss mittels Bilderkennung und frei verfügbaren Webcams bestimmt und über offene, standardisierte Schnittstellen

bereitgestellt und visualisiert werden kann. Bei dem entwickelten Workflow kommen dabei ausschließlich Open-Source-Bibliotheken und Dienste sowie Open Data zum Einsatz.

## 2 Workflow und Architektur

Der gesamte Workflow beinhaltet eine einmalige Aufbereitung bzw. Einrichtung der Komponenten und eine kontinuierliche Verarbeitung der Bilder der Webcams, um darüber das Verkehrsaufkommen zu bestimmen. Die beiden Bestandteile des Workflows sind in der folgenden Abbildung 1 dargestellt. Bei der einmaligen Datenaufbereitung (Abb. 1a) müssen über die Sichtbereiche der Kameras die jeweils betroffenen Straßensegmente ermittelt werden. Für diesen Beitrag wurden dafür mittels eines Geoinformationssystem (z. B. QGIS, vgl. Graser 2013) und der jeweils vorhandenen Bilder der Verkehrskameras die Sichtbereiche als Polygone erfasst. Anschließend wurde eine Verschneidung der einzelnen Polygone mit den OpenStreetMap-(OSM-)Daten für das Straßennetzwerk durchgeführt. In Deutschland sind OSM-Daten bereits seit Längerem im Bereich von offenen Daten für das Wegenetz eine Alternative (Neis et al., 2011) und auch in Regionen, wo Verkehrskameras vorhanden sein sollten, dürften die Straßendaten inzwischen in ausreichender Qualität für den hier präsentierten Anwendungsfall vorliegen (vgl. Neis et al., 2013). Da je nach Sichtbereich unterschiedliche Straßentypen vom Bild einer Verkehrskamera abgedeckt sein können, erfolgte in diesem prototypischen Ansatz noch eine anschließende manuelle Prüfung der Segmente.

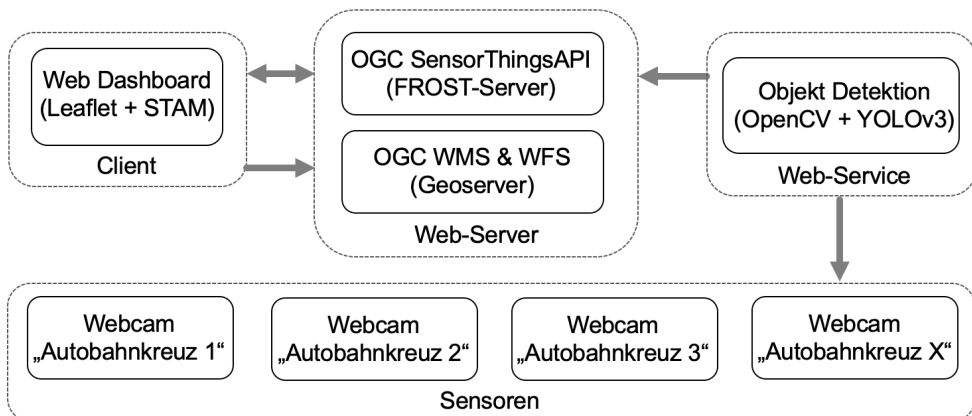
Der zweite Bestandteil des Workflows ist für die kontinuierliche Bildauswertung zur Ermittlung des Verkehrsaufkommens verantwortlich (siehe Abb. 1b). Dabei werden die statischen Bilder in festgelegten Zeitintervallen, je nach Aktualisierung des Bildes, abgerufen und ausgewertet. Nach der Erkennung der Verkehrsteilnehmer werden je nach Region of Interest (ROI) die relevanten Summen pro Fahrtrichtung ermittelt und dauerhaft gespeichert.



**Abb. 1:** Schematische Darstellung des Workflows zur einmaligen Datenaufbereitung (a) und zur kontinuierlichen Bildauswertung (b)

Die genutzte Architektur für die Speicherung und Bereitstellung des erfassten Verkehrsaufkommens orientiert sich sehr stark an den Ansätzen von Visca et al. (2022) und Visca et al. (2023). Diese haben für die Erstellung eines Besucherinformationssystems am Beispiel des Campus der Hochschule Mainz ausschließlich auf Open Data, Open-Source-Software und

offene Schnittstellen zurückgegriffen. Abbildung 2 zeigt die Architektur der einzelnen Komponenten des Gesamtsystems dieses Beitrags. Bei den verwendeten Sensoren handelt es sich wie bereits erwähnt um die verfügbaren Webcams zur Prüfung des Verkehrsaufkommens, die frei im Netz zu finden sind (vgl. RLP-Verkehr, 2022). Die Bildauswertung wurde mittels OpenCV und YOLOv3 in einem Python-Skript implementiert. Die Datenhaltung bzw. die Speicherung des Verkehrsaufkommens der einzelnen Sensoren erfolgt über die OGC SensorThings API (Liang et al., 2016). Vom Fraunhofer IOSB kommt im Rahmen dieses Beitrags der frei verfügbare FROST-Server als Implementierung des Standards zum Einsatz. Die im Web-Dashboard benötigten Hintergrundkarten können gemäß der OpenGIS Web-Map-Service- (Beaujardiere, 2006) und OGC Web-Feature-Service-Spezifikationen (Vretanos, 2005) eingebunden werden. Als Implementierung könnte der frei verfügbare GeoServer (vgl. z. B. Youngblood, 2013) der gemeinnützigen Open-Source-Geospatial-Foundation-Organisation (OSGeo) Verwendung finden. Die Visualisierung der ermittelten Daten wurde prototypisch mittels der Leaflet- (vgl. Crickard III, 2014) und der SensorThings-API-Map-JavaScript-Bibliothek (STAM) umgesetzt.



**Abb. 2:** Architektur des Gesamtsystems mit den einzelnen Komponenten zur Erfassung, Speicherung und Visualisierung des Verkehrsaufkommens

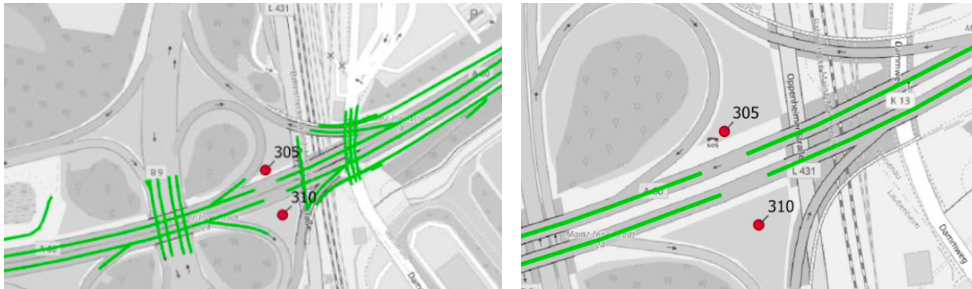
### 3 Prototypische Implementierung

Der vorgestellte Workflow und die Architektur des Gesamtsystems wurden prototypisch für die Landeshauptstadt Mainz implementiert. Die relevanten Kamerastandorte konnten aus dem RLP-Verkehr-Portal (2022) übernommen werden. Für die verwendeten Kameras in und um Mainz konnten teilweise Metadaten über den Kamerateyp und z. B. die Blickrichtung abgerufen werden. Bei den Untersuchungen dieses Beitrags waren die angegebenen Werte für die Blickrichtung (Richtungswinkel) im Vergleich zu den verfügbaren Bildern der Webcams aber nicht immer nachvollziehbar, wodurch auf eine manuelle Erstellung bzw. Prüfung zurückgegriffen wurde. Auf Basis der Kamerastandorte, der vorhandenen Bildern der Verkehrsüberwachung und des OSM-Wegenetzes konnten so die relevanten Straßensegmente mithilfe von QGIS erstellt werden. Abbildung 3 zeigt im Bild links einen Überblick von den 30 ver-

wendeten in und um Mainz stationierten Webcams. In der Mitte der Abbildung sind für zwei Kamerastandorte (rote Punkte) die manuell erstellten Sichtbereiche (gelbe Polygone) zu sehen. Abbildung 4 enthält links das Ergebnis der Verschneidung des Straßennetzwerkes und der Sichtbereiche und rechts die daraus resultierenden Straßensegmente (grüne Linien) auf Basis der verwendeten OSM-Daten.



**Abb. 3:** Links: Kamerastandorte aus RLP-Verkehr (2022); rechts: erstellte Sichtbereiche für zwei Kamerastandorte (Hintergrundkarte © basemap.de)



**Abb. 4:** Links: Ergebnis der Verschneidung der Sichtbereiche und Straßennetzwerk; rechts: die resultierenden Straßensegmente für die beiden Kameras (Hintergrundkarte © basemap.de)

Die Ermittlung des Verkehrsaufkommens erfolgt wie beschrieben auf Basis einer Bildauswertung. Dafür wurde ein eigenes Python-Skript implementiert, welches die Verkehrsteilnehmer, bei der prototypischen Implementierung die Fahrzeuge, detektiert, zählt und anschließend an die SensorThings-API zum Speichern sendet. Die finale Visualisierung der Anzahl der erfassten Verkehrsteilnehmer wurde über ein Web-Dashboard mit Leaflet und STAM realisiert. Auf einer Karte können so aktuelle Informationen über das Verkehrsaufkommen angezeigt und auch in Form eines Diagrammes, welches einen zeitlichen Verlauf der Verkehrsaufkommen enthält, abgerufen werden. Durch die Sicherheitslage im Jahr 2022 konnte der Prototyp nicht vollumfänglich in Betrieb genommen werden. Seit März 2022 werden bis auf Weiteres keine Bilder von Verkehrswebcams in Deutschland bereitgestellt (vgl. Rheinland 2022).

## 4 Diskussion der Ergebnisse

Bei der prototypischen Implementierung des hier vorgestellten Workflows sind verschiedene Herausforderungen und Schwierigkeiten aufgetreten. Beispielsweise die Datengewinnung der Kamerastandorte, deren Metadaten und die anschließende Verarbeitung der Daten in QGIS war durch die händische Erfassung der Sichtbereiche und Verschneidung zeitaufwendig. An dieser Stelle besteht Potenzial für eine Automatisierung um die Übertragbarkeit auf andere Städte zu vereinfachen. Die Situation in der Ukraine (seit März 2022) hatte und hat zur Folge, dass die Bilder der Webcams für deutsche Straßen nicht mehr abgerufen werden können. Daher konnte das hier vorgestellte Gesamtsystem nicht vollständig getestet und in einen prototypischen Betrieb gehen. Alle möglichen Tests der einzelnen Komponenten verliefen erfolgreich, wodurch eine Inbetriebnahme im Normalfall keine größere Herausforderung mehr sein sollte. Dennoch wurde für die Übertragbarkeit des Systems ein Test für England in der Stadt Huntingdon durchgeführt. Hierbei zeigten der vorgestellte Workflow und die Architektur keine Auffälligkeiten und visualisierten das ermittelte Verkehrsaufkommen für die eingebundenen Standorte der Webcams.

Die eingesetzte Methode zur Extraktion der Informationen über die Verkehrsauslastung in Bildern war im Rahmen dieses Beitrags der YOLO-Algorithmus. Dieser war bei verschiedenen Tests im Vergleich zu anderen Ansätzen i. d. R. einer der schnellsten in der Objektdetektion. Dennoch ist es möglich, dass sich beispielsweise zur Auswertung von Videostreams von Verkehrskameras ein anderer Ansatz besser eignet. Dieser sollte je nach Bildquelle und -format geprüft werden.



**Abb. 5:** Beispielbild für Grenzen des derzeitigen Prototyps bei der Objektdetektion des Verkehrsaufkommens (Bild: kreiszeitung.de, © Peter Steffen/dpa)

Ein weiterer und bisher nicht untersuchter Punkt ist der Einfluss des Wetters bei dem hier vorgestellten Ansatz der Erkennung von Verkehrsteilnehmern. Es wurden zwar mehrere Tests mit verschiedenen Wetter- und Helligkeitssituationen durchgeführt, hier wird ein Live-Betrieb aber weitere Erkenntnisse bringen. Je nach Kamerastandort und Sichtfeld müssen dazu Untersuchungen durchgeführt werden, um ggf. individuelle Konfigurationen bei der

Objekterkennung vorzunehmen. Beispielsweise ist bereits aufgefallen, dass bei zu starker Überlappung der Fahrzeuge im Bild (vgl. Abb. 5) die Detektion nicht immer verlässlich funktioniert. Zudem ist erkennbar, dass bei Dunkelheit der Kontrast zu den hellen Scheinwerfern im Bild einer Verkehrskamera ein Problem bei der Detektion darstellt. In diesem Fall wurden nur sehr nahe Fahrzeuge erkannt. Aber auch einseitige Lichtverhältnisse im Dunkeln erschweren die Detektion.

Die Visualisierung der Verkehrsaufkommen erfolgt derzeit in einem einfachen Web-Dashboard, welches in Design, Benutzerfreundlichkeit und vor allem hinsichtlich Funktionsumfang noch viel Potenzial bietet. Denkbar wären hier Folgearbeiten um eine Responsive Web App zu implementieren oder auch die Integration ins Masterportal, wäre eine denkbare Alternative (vgl. Fischer et al., 2021). Daneben wäre nicht nur die Anzeige der aktuellen und vergangenen Verkehrssituation von Vorteil, sondern die Ermittlung einer Prognose über das mögliche Verkehrsaufkommen für einen zukünftigen Zeitpunkt. Abbildung 6 zeigt beispielhaft die Visualisierung des Verkehrsaufkommens für die Stadt Mainz durch den in diesem Beitrag vorgestellten Workflow und den potenziell nutzbaren Webcams. Hierbei ist bei den Einfärbungen (grün geringes bis rot hohes Verkehrsaufkommen) bereits gut ersichtlich, dass jeweils nur für bestimmte Straßensegmente eine Aussage getroffen werden kann. Kommerzielle Anbieter, wie z. B. GoogleMaps, Inrix oder TomTom, bieten jeweils komplette Informationen für das Autobahnnetz und die Hauptverkehrsstraßen in Deutschland.



**Abb. 6:** Beispielkarte für die Visualisierung des Verkehrsaufkommens für die Stadt Mainz (Hintergrundkarte © OpenStreetMap Mitwirkende)



## 5 Zusammenfassung und Ausblick

In diesem Beitrag wurde ein Workflow und eine Architektur vorgestellt, die aus frei verfügbaren Bildern von Verkehrskameras eine Aussage über die aktuelle Verkehrsauslastung ermittelt, diese speichert und in einer webbasierten Karte darstellen kann. Mit einer prototypischen Implementierung konnte für die rheinlandpfälzische Landeshauptstadt Mainz demonstriert werden, dass der vorgestellte Ansatz funktioniert und dabei ausschließlich Open Data, Open-Source-Bibliotheken und offene standardisierte Schnittstellen verwendet. Durch eine komponentenbasierte Architektur können Teile des Systems ausgetauscht oder erweitert werden.

Als Folgearbeit könnte der hier präsentierte Anwendungsfall des Individualverkehrs mit Fahrzeugen durch eine Bilderkennung von Fahrradfahrer:innen oder Personen ergänzt werden. Daneben bieten die beschriebenen Herausforderungen und Schwierigkeiten bei der Auswertung der Kamerabilder zur Messung des Verkehrs verschiedenes Potenzial um die Quantität und Qualität der Ergebnisse zu verbessern. Weiterhin zeigt Abbildung 6, dass nur für die Straßensegmente im Sichtfeld einer Verkehrswebcam eine entsprechende Aussage über das Verkehrsaufkommen getätigt werden kann. In einer Folgearbeit sollen die Ergebnisse des vorgestellten Ansatzes mit Daten des jeweiligen Verkehrsaufkommens von anderen Datenlieferanten verglichen werden. Damit sollen Erkenntnisse gewonnen werden, ob die Informationen von einzelnen Straßensegmenten ausreichend für Abschätzungen von Zeitverzögerungen im Vergleich zu kommerziellen Datenanbietern sind. Letzteres stand nicht im Fokus dieses Beitrages.

## Literatur

- Ahmad, I., & Pothuganti, K. (2020). Design & implementation of real time autonomous car by using image processing & IoT. In: *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)* (pp. 107–113). IEEE.
- De la Beaujardiere, J. (2006). *OpenGIS® Web Map Server Implementation Specification. Version 1.3. 0*.
- Beymer, D., McLauchlan, P., Coifman, B., & Malik, J. (1997). A real-time computer vision system for measuring traffic parameters. In: *Proceedings of IEEE computer society conference on computer vision and pattern recognition* (pp. 495–501). IEEE.
- Bickel, P. J., Chen, C., Kwon, J., Rice, J., Van Zwet, E., & Varaiya, P. (2007). Measuring traffic. *Statistical Science*, (2007), 581–597.
- Bradski, G. (2000). The openCV library. *Software Tools for the Professional Programmer*, 25(11), 120–123.
- Chen, S., Xu, H., Liu, D., Hu, B., & Wang, H. (2014). A vision of IoT: Applications, challenges, and opportunities with china perspective. *IEEE Internet of Things journal*, 1(4), 349–359.
- Cheung, S. Y., Coleri, S., Dundar, B., Ganesh, S., Tan, C. W., & Varaiya, P. (2005). Traffic measurement and vehicle classification with single magnetic sensor. *Transportation research record*, 1917(1), 173–181.
- Ćorović, A., Ilić, V., Đurić, S., Marijan, M., & Pavković, B. (2018). The real-time detection of traffic participants using YOLO algorithm. In: *2018 26th Telecommunications Forum (TELFOR)* (pp. 1–4). IEEE.



- Crickard III, P. (2014). *Leaflet. js essentials*. Birmingham (UK): Packt Publishing Ltd.
- Farooq, M. U., Waseem, M., Mazhar, S., Khairi, A., & Kamal, T. (2015). A review on internet of things (IoT). *International Journal of Computer Applications*, 113(1), 1–7.
- Fischer, M., Gras, P., Löwa, S., & Schuhart, S. (2021). Urban Data Platform Hamburg: Integration von Echtzeit IoT-Daten mittels SensorThings API. *Zeitschrift für Geodäsie, Geoinformation und Landmanagement*, (1).
- Ghosh, A., Sabuj, M. S., Sonet, H. H., Shatabda, S., & Farid, D. M. (2019). An adaptive video-based vehicle detection, classification, counting, and speed-measurement system for real-time traffic data collection. In: *2019 IEEE Region 10 Symposium (TENSYP)* (pp. 541–546). IEEE.
- Graser, A. (2013). *Learning QGIS 2.0*. Birmingham (UK): Packt Publishing Ltd.
- Gupta, A., Kulkarni, S., Jathar, V., Sharma, V., & Jain, N. (2017). Smart car parking management system using IoT. *American Journal of Engineering Science and Technology*, 2(4), 112–119.
- Jiang, P., Ergu, D., Liu, F., Cai, Y., & Ma, B. (2022). A Review of Yolo algorithm developments. *Procedia Computer Science*, 199, 1066–1073.
- Lan, W., Dang, J., Wang, Y., & Wang, S. (2018). Pedestrian detection based on YOLO network model. In *2018 IEEE international conference on mechatronics and automation (ICMA)* (pp. 1547–1551). IEEE.
- Lei, Z., Xue-fei, Z., & Yin-ping, L. (2008). Research of the real-time detection of traffic flow based on OpenCV. In: *2008 International Conference on Computer Science and Software Engineering* (Vol. 2, pp. 870–873). IEEE.
- Li, D., Liang, B., & Zhang, W. (2014). Real-time moving vehicle detection, tracking, and counting system implemented with OpenCV. In: *2014 4th IEEE International Conference on Information Science and Technology* (pp. 631–634). IEEE.
- Liang, S., Huang, C. Y., & Khalafbeigi, T. (2016). *OGC SensorThings API Part 1: Sensing, Version 1.0*.
- Neis, P., Zielstra, D., & Zipf, A. (2011). The street network evolution of crowdsourced maps: OpenStreetMap in Germany 2007–2011. *Future Internet*, 4(1), 1–21.
- Neis, P., Zielstra, D., & Zipf, A. (2013). Comparison of volunteered geographic information data contributions and community development for selected world regions. *Future Internet*, 5(2), 282–300.
- Rabby, M. K. M., Islam, M. M., & Imon, S. M. (2019). A review of IoT application in a smart traffic management system. In: *2019 5th International Conference on Advances in Electrical Engineering (ICAEE)* (pp. 280–285). IEEE.
- Rheinland (2022). *Deutsche Autobahn-Kameras offline*. Retrieved from [https://www.rheinpfalz.de/lokal/pfalz-ticker\\_artikel,-deutsche-autobahn-kameras-offline-arid,5328494.html](https://www.rheinpfalz.de/lokal/pfalz-ticker_artikel,-deutsche-autobahn-kameras-offline-arid,5328494.html).
- RLP-Verkehr (2022). *Die zentrale Informationsplattform für den Bereich Verkehr in Rheinland-Pfalz*. Retrieved from <https://verkehr.rlp.de/>.
- Sang, J., Wu, Z., Guo, P., Hu, H., Xiang, H., Zhang, Q., & Cai, B. (2018). An improved YOLOv2 for vehicle detection. *Sensors*, 18(12), 4272.
- Shubho, F. H., Iftekhar, F., Hossain, E., & Siddique, S. (2021). Real-time traffic monitoring and traffic offense detection using YOLOv4 and OpenCV DNN. In: *TENCON 2021-2021 IEEE Region 10 Conference (TENCON)* (pp. 46–51). IEEE.
- Uke, N. J., & Thool, R. (2013). Moving vehicle detection for measuring traffic count using opencv. *Journal of Automation and Control Engineering*, 1(4), 349–352.

- Visca, D., Hoppe, M., & Neis, P. (2022): Mit IoT-Sensorik zu einem innovativen Besucherinformationssystem – Ein nachhaltiger und übertragbarer Prototyp der Hochschule Mainz. In: *Tagungsband GeoForum MV 2022, Technologiepark Warnemünde*.
- Visca, D., Hoppe, M., & Neis, P. (2023): Verfahren zur Ermittlung der Besucherfrequenz mittels (mobiler) Low-Cost-Sensorik. In: *Proceedings 22. Internationale Geodätische Woche Obergurgl 2023*.
- Vretanos, P. (2005). *Web Feature Service Implementation Specification, Version 1.1.0*.
- World Bank (2016). *The World Bank Launches New Open Transport Partnership to Improve Transportation through Open Data*. Retrieved from <https://www.worldbank.org/en/news/press-release/2016/12/19/the-world-bank-launches-new-open-transport-partnership-to-improve-transportation-through-open-data>.
- Youngblood, B. (2013). *GeoServer Beginner's Guide*. Birmingham (UK): Packt Publishing Ltd.
- Zangenehpour, S., Miranda-Moreno, L. F., & Saunier, N. (2015). Automated classification based on video data at intersections with heavy pedestrian and bicycle traffic: Methodology and application. *Transportation research part C: emerging technologies*, 56, 161–176.
- Zheng, J., Ma, X., Wu, Y. J., & Wang, Y. (2013). Measuring signalized intersection performance in real-time with traffic sensors. *Journal of Intelligent Transportation Systems*, 17(4), 304–316.