

Masterarbeit

# Erkennung von Fahrspuren mittels Fahrzeugtrajektorien aus Luftaufnahmen

im Master-Studiengang Informatik  
der Hochschule Furtwangen

Steffen Schmid

**Zeitraum:** Wintersemester 2018

**Prüfer:** Prof. Dr. Christoph Reich

**Zweitprüfer:** Stefan Kaufmann

---

**Firma:** IT-Designers GmbH

**Betreuer:** Stefan Kaufmann

# Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Esslingen, den 19. Oktober 2018

\_\_\_\_\_  
Unterschrift

# Zitat

*„Some fancy quote“*

FooBar Muman

# Danksagung

# Kurzfassung

Schlagworte:

# Inhaltsverzeichnis

Kurzfassung	v
1 Einleitung	1
1.1 Rahmen der Arbeit . . . . .	1
1.1.1 Das Projekt MEC-View . . . . .	1
1.1.2 MEC-View Luftbeobachtung . . . . .	1
1.2 Motivation und Ziele . . . . .	1
1.3 Aufbau dieser Arbeit . . . . .	1
2 Grundlagen	2
2.1 Verkehrsanalyse mittels Luftaufnahmen . . . . .	2
2.2 Rekonstruktion von Fahrzeugtrajektorien aus Luftaufnahmen . . . . .	2
2.3 Datenaufbereitung und Bereinigung . . . . .	4
2.4 Clusteranalyse . . . . .	4
2.4.1 Eigenschaften von Cluster-Sets und Clustern . . . . .	6
2.4.2 Cluster-Algorithmen . . . . .	7
2.4.3 Distanzmaße zum Vergleich von Fahrzeugtrajektorien . . . . .	7
2.5 Untersuchung möglicher Straßentopologien . . . . .	7
3 Stand der Technik	8
3.1 Clustering von Trajektoriedaten . . . . .	8
3.2 Erkennung von Fahrspuren . . . . .	8
3.3 Klassifizierung von Fahrspuren . . . . .	8
3.4 Defizite vorhandener Lösungen und benötigte Neuerungen . . . . .	8
4 Clustering von Fahrzeugtrajektorien	9
4.1 Vorverarbeitung der Roh-Trajektorien . . . . .	9
4.2 Gruppierung der Trajektorien . . . . .	9
5 Fahrbahn-Bestimmung aus Trajektorie-Clustern	10
6 Fahrbahn Klassifizierung	11
7 Realisierung LaneDetection in MEC-View TrackerApplication Software	12
8 Ergebnisse und Auswertung	13

9 Zusammenfassung und Fazit	14
Literaturverzeichnis	15

# Abbildungsverzeichnis

2.1	Übersicht Tracking mit Klassifikator . . . . .	3
2.2	a) Haar-ähnliche Merkmale b) Beispiele für erkannte Regionen in einem Gesicht . . . . .	4
2.3	Rohdaten (links) und erwünschtes Clustering-Ergebnis (rechts) . . . . .	5
2.4	Ablauf einer Clusteranalyse . . . . .	5



# Listings

# 1 Einleitung

Staus und zäh fließender Verkehr sind sowohl auf Schnell- und Autobahnen, als auch in Städten ein großes Problem und Ärgerniss für Autofahrer. Sie kosten diese nicht nur wertvolle Zeit, sondern auch viel Geld. Laut einer Studie von [Cookson u. a.] kostet Stau jeden deutschen Autofahrer pro Jahr durchschnittlich 1770 €. In Summe ergeben sich hieraus beinahe 80 Milliarden Euro an Kosten. Stau ist allerdings nicht nur finanziell für Privatpersonen oder auch Unternehmen ein großer Faktor, sondern er erhöht auch das Unfallrisiko und trägt maßgeblich zur schlechten Luftqualität in Innenstädten bei. Aufgrund längerer Fahrzeiten und der häufigen Be- und Entschleunigung, steigt der Kraftstoffverbrauch der Fahrzeuge und dadurch auch die Schadstoffbelastung in der Luft [Hemmerle, 2016].

Um Stau so gut wie möglich vermeiden zu können, muss man den Verkehr verstehen. Nötig ist ein Verständnis des Straßenverkehrs als Ganzes, sowie der Auswirkungen, welche einzelne Verkehrsteilnehmer und deren Verhalten, auf diesen haben. Hierzu ist das Erstellen von Simulationen sowie die Auswertung realer Verkehrsaufkommen unerlässlich. Die auf diese Weise gesammelten Erkenntnisse bilden die Grundlage, um Straßenabschnitte, insbesondere auch in Innenstädten, intelligent zu gestalten. Des Weiteren können sie eingesetzt werden, um beispielsweise Ampelschaltungen in Städten zu optimieren, wovon auch bestehende Infrastrukturen profitieren können.

Diese Arbeit beschäftigt sich mit der Realisierung einer automatischen Fahrspurerkennung aus Luftaufnahmen, welche bei der Analyse von Spurwechselvorgängen zum Einsatz kommt. Hierzu werden die Trajektoriedaten von Fahrzeugen ausgewertet.

## 1.1 Rahmen der Arbeit

### 1.1.1 Das Projekt MEC-View

### 1.1.2 MEC-View Luftbeobachtung

## 1.2 Motivation und Ziele

## 1.3 Aufbau dieser Arbeit

## 2 Grundlagen

In diesem Kapitel werden die für das Verständnis und die Durchführung der Arbeit benötigten Grundlagenthemen vorgestellt. Nach einer kurzen Erläuterung der Möglichkeiten der Verkehrsanalysen mittels Luftaufnahmen, wird daher darauf eingegangen, auf welche Weise die in dieser Arbeit verwendeten Fahrzeugtrajektorien ermittelt werden. Anschließend werden Methoden vorgestellt, welche zur Bereinigung der gewonnenen Daten verwendet werden können. Als wichtiges Mittel zur Identifizierung von Fahrspuren aus Trajektorien werden zudem verschiedene Cluster-Algorithmen und Distanzmaße vorgestellt.

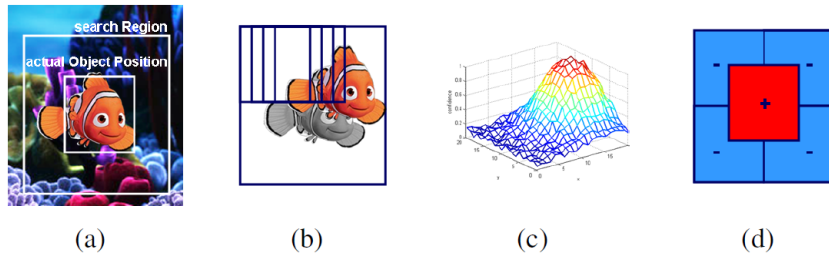
### 2.1 Verkehrsanalyse mittels Luftaufnahmen

### 2.2 Rekonstruktion von Fahrzeugtrajektorien aus Luftaufnahmen

Die in dieser Arbeit verwendeten Fahrzeugtrajektorien stammen aus der Anwendung “Tracker-Application” des MEC-View Teilprojektes *Luftbeobachtung*. Nachfolgend wird beschrieben, wie diese aus den Videoaufnahmen rekonstruiert werden.

Die Verfolgung von bewegten Objekten beziehungsweise Fahrzeugen, wird in der “Tracker-Application” mittels *Supervised Tracking* umgesetzt. Bei diesem Verfahren wird ein initial manuell ausgewählter Bildbereich automatisch mit Hilfe eines erlernten Klassifikators verfolgt. Der Klassifikator muss hierbei zwischen Fahrzeugen und der Umgebung unterscheiden können. Das grundlegende Vorgehen dieses Tracking-Ansatzes ist in Abbildung 2.1 dargestellt und kann wie folgt beschrieben werden:

- a) Verfolgtes Objekt befindet sich zum Zeitpunkt  $t$  an bekannter Position  $p_1$
- b) Zum Zeitpunkt  $t + 1$ : Anwendung des Klassifikators auf Positionen um  $p_1$
- c) Erstellen einer *Confidence Map*, welche die Wahrscheinlichkeit darstellt, das verfolgte Objekt gefunden zu haben
- d) Updaten des Trackers auf Position des Maxima der *Confidence Map*

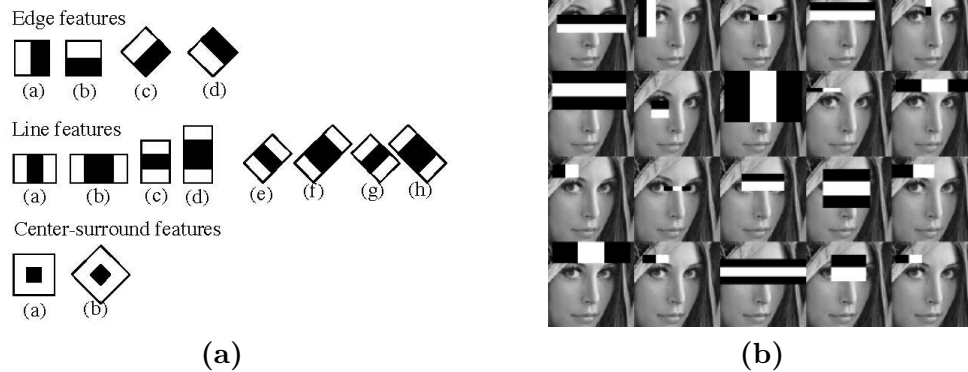


**Abb. 2.1:** Übersicht Tracking mit Klassifikator [Grabner u. a.]

Das Erlernen eines stabilen Klassifikators in der “Tracker-Application” basiert auf der Arbeit “*Real-Time Tracking via On-line Boosting*” von [Grabner u. a.]. Die Autoren verwenden einen On-line AdaBoost Algorithmus, welcher mehrere *schwache* Klassifikatoren zu einem *starken* Klassifikator kombiniert. Schwache Klassifikatoren müssen hierbei nur eine Erkennungsrate von mehr als 50% besitzen und somit wenig besser als zufallsbedingtes Auswählen sein. Starke Klassifikatoren entstehen durch die Kombination von mehreren schwachen Klassifikatoren. Die Auswahl von schwachen Klassifikatoren erfolgt über sogenannte Selektoren, welche aus einer Menge immer jenen wählen, welcher die geringste Fehlerrate bei der Erkennung der Trainings-Objekte besitzen. Der Klassifikator mit der schlechtesten Erkennungsrate wird in jeder Trainingsiteration ersetzt, um das Training zu verbessern. Großer Vorteil der On-line AdaBoost Methode ist, dass sie es ermöglicht, starke Klassifikatoren während des eigentlichen Trackingvorganges zu erlernen. Nach jedem Trackingschritt wird das erfolgreich erkannte Objekt in Trainingssätze zerlegt, auf welche die Klassifikatoren angewandt werden um ihre Performance zu evaluieren. So wird in jedem Schritt die Menge der schwachen Klassifikatoren und der Selektoren aktualisiert. Die Wahl von effizient berechenbaren schwachen Klassifikatoren macht dies möglich.

Die in [Grabner u. a.] und der “Tracker-Application” verwendeten Klassifikatoren sind binär, das heißt, sie teilen Objekte in die zwei Klassen *erkannt* und *nicht erkannt* auf. Konkret werden Haar-ähnliche Bildmerkmale nach [Viola u. a.] als schwache Klassifikatoren verwendet. Diese sind ein Mittel zur Identifikation von Kontrastunterschieden in Bildern, welche sich sehr gut zur Erkennung von Kanten und Linien eignen. Ein Beispiel der Haar-ähnlichen Merkmale und ihres Einsatzes bei der Gesichtserkennung ist in Abbildung 2.2 dargestellt.

Diese Merkmale werden als schwache Klassifikatoren mit zufälliger Skalierung, Größe und Position auf dem Bild platziert. Sie suchen in dieser Region anschließend nach den von dem Muster definierten Konturunterschieden. Ein Bereich gilt als erkannt, wenn der Betrag der Differenz der Pixelsumme des weißen und schwarzen Bereiches des Musters unter einem festgelegten Grenzwert liegt.



**Abb. 2.2:** a) Haar-ähnliche Merkmale b) Beispiele für erkannte Regionen in einem Gesicht [Divyansh Dwivedi, 2018]

## 2.3 Datenaufbereitung und Bereinigung

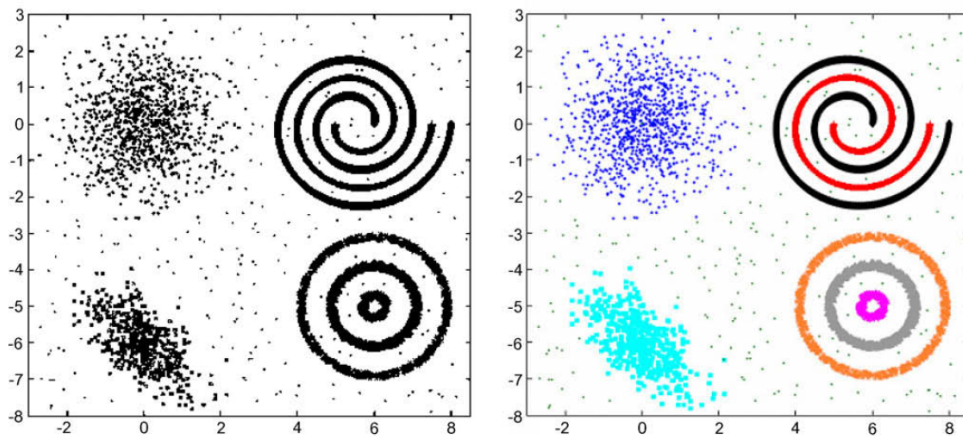
## 2.4 Clusteranalyse

Die Clusteranalyse (kurz Clustering) ist ein wichtiges Werkzeug zur Auswertung von Daten unterschiedlichster Art. Sie stellt dabei kein konkretes Vorgehen oder einen Algorithmus dar, sondern beschreibt ein allgemeines Problem, welches auf unterschiedlichste Weise gelöst werden kann. Grundsätzlich ist das Ziel der Clusteranalyse, Datenobjekte aufgrund ihrer Eigenschaften und Beziehungen untereinander so zu gruppieren, dass sich die Objekte einer Gruppe möglichst stark ähneln und sich von den Objekten anderer Gruppen möglichst stark unterscheiden. Je höher die *Homogenität* in einem Cluster und die *Differenz* zwischen den Clustern, desto besser ist die gewählte Clustering Methode. Der Einsatz von Clustering ist in vielen Anwendungsgebieten und in den unterschiedlichsten wissenschaftlichen Disziplinen sehr beliebt, um ein Verständnis für Daten zu erhalten beziehungsweise diese anschließend weiter verarbeiten zu können. So kommt die Clusteranalyse unter anderem in den Feldern des maschinellen Lernens, der Mustererkennung, Bildanalyse, der Biologie (Taxonomie) oder im Bereich Data Mining zum Einsatz. [Tan u. a., 2007]

Die Clusteranalyse hat viel mit dem Problem der Klassifizierung von Daten gemein, insofern sie Datenobjekten Label zuordnet. Im Gegensatz zu *überwachten* Klassifizierungsansätzen, wie dem heute populären überwachten Lernen, leiten Cluster-Algorithmen die Label allerdings alleine aus den vorhandenen Daten ab. Es kommen keine Vergleichsobjekte mit bekannten, händisch vergebenen Labels zum Einsatz. Aus diesem Grund wird die Clusteranalyse auch häufig als *unüberwachte Klassifizierung* bezeichnet. [Tan u. a., 2007]

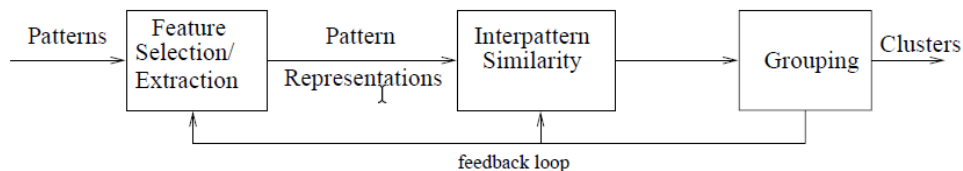
Das Konzept eines *Clusters* ist nicht genau definiert, was in einer Vielzahl an unterschiedlichen Ansichten und Algorithmen resultiert, welche sich jeweils für andere Anwendungsfälle eignen und verschiedene Eigenschaften besitzen. Hieraus ergibt sich auch die Tatsa-

che, dass Clustering keine selbsttätiger Prozess ist, welcher sich auf einheitliche Weise auf unterschiedliche Probleme anwenden lässt. Jedes Problem erfordert die individuelle und sorgfältige Auswahl eines passenden Algorithmus, eines Distanzmaßes und der richtigen Parameter. Die Bestimmung dieser geschieht iterativ und nicht selten nach dem Prinzip des *Trial and Error*. In Abbildung 2.3 ist beispielhaft ein Datensatz (links) mit – für den Menschen intuitiv ersichtlich – 7 unterschiedlichen Clustern (rechts) dargestellt. Nach [Jain, 2010] kann allerdings kein verfügbarer Clustering Algorithmus diese alle erkennen. [Jain u. a., 1999; Tan u. a., 2007]



**Abb. 2.3:** Rohdaten (links) und erwünschtes Clustering-Ergebnis (rechts) [Jain, 2010]

Aufgrund der Limitationen, welche alle Cluster-Algorithmen besitzen, muss der Analyst sich vor deren Anwendung intensiv mit den zu verarbeitenden Daten beschäftigen. Er muss ein Verständnis dafür besitzen, welche Struktur die Daten besitzen, beziehungsweise annehmen können, und nach welchen Mustern zu suchen ist. Besonders wichtiger ist zudem auch die Auswahl der richtigen, das heißt relevanten, Datenmerkmale (*“Feature Selektion”*) und die Wahl deren Repräsentation (*“Feature Transformation”*). Die Selektion und gegebenenfalls Transformation der Daten muss in einem Vorverarbeitungsschritt geschehen, dessen Qualität einen maßgeblichen Einfluss auf das finale Clustering Ergebnis hat. Basierend auf vorangegangener Beschreibung und [Jain u. a., 1999], lässt sich der Ablauf einer Clusteranalyse wie folgt darstellen:



**Abb. 2.4:** Ablauf einer Clusteranalyse

### 2.4.1 Eigenschaften von Cluster-Sets und Clustern

Das aus einer Analyse resultierende Cluster-Set und die einzelnen Cluster selbst, können in verschiedene Kategorien unterteilt werden beziehungsweise unterschiedliche Eigenschaften besitzen. Nachfolgend sind die wichtigsten basierend auf [Tan u. a., 2007] und [Jain u. a., 1999; Jain, 2010] aufgeführt.

#### Cluster-Sets

Bei Cluster-Sets kann grundsätzlich zwischen nachfolgenden Eigenschaften unterschieden werden.

**Hierarchisch vs. Partitioniert** Von *hierarchischen* Cluster-Sets wird gesprochen, wenn die einzelnen Cluster verschachtelt sind und dabei eine Baum-Struktur bilden. Cluster sind hingegen *partitioniert*, wenn keine Überlagerungen zwischen ihnen existiert.

**Exklusiv vs. Überlappend vs. Fuzzy** *Exklusive* Cluster-Sets liegen vor, wenn jedem Datenwert ein oder kein Zielcluster zugeordnet wird. Im Gegensatz hierzu können bei *überlappenden* Cluster-Sets Objekte einer oder mehrerer Gruppen angehören. Bei dem sogenannten *Fuzzy* oder *Soft* Cluster-Sets, gehört ein Datenobjekt einem Cluster mit einer bestimmten Wahrscheinlichkeit oder Gewicht an. Algorithmen, welche Daten eine Wahrscheinlichkeit für die Zugehörigkeit zu einem Cluster zuweisen, werden *probabilistische* Cluster-Algorithmen genannt.

**Komplett vs. Partielle** Von *kompletten* Cluster-Sets wird gesprochen, wenn jedes Element der Eingangsdaten einem Cluster zugeordnet wird. Bei *partiellen* Sets ist dies nicht der Fall. Hier kann ein bestimmter Anteil an Datenwerten als Ausreißer markiert werden, welche keine Gruppe besitzen.

#### Cluster

Da, wie oben erwähnt, nicht klar definiert ist, was ein Cluster ausmacht, können auch diese unterschiedliche Eigenschaften besitzen. Die wichtigsten Cluster-Arten sind nachfolgend erläutert.

**Klar separierte Cluster** Unter *klar separierten* Clustern versteht man solche, in welchen jedes Datenelement einen geringeren Abstand zu allen anderen Elementen des Clusters hat, als zu Elementen außerhalb des Clusters. Diese idealistische Definition eines Clusters ist nur dann erfüllt, wenn die in den Daten enthaltenen Cluster einen großen Abstand voneinander haben. Dies ist in der Realität allerdings selten der Fall.

**Prototyp basierte Cluster** Von einem *Prototyp basierten* Cluster wird gesprochen, wenn alle Elemente einer Gruppe einen geringeren Abstand zu einem Prototyp oder Referenzwert des Clusters besitzen, als zu denen anderer Gruppierungen. Ein solcher Prototyp ist üblicherweise der Mittelwert der Datenelemente eines Clusters (*Centroid*).

**Graphen basierte Cluster** Die Definition eines *Graphen basierten* Clusters kann immer dann verwendet werden, wenn Daten als vernetzter Graph dargestellt werden. In einem solchen sind die Elemente Knoten und die Kanten repräsentieren Beziehungen zwischen ihnen. Ein Cluster in einem solchen Graphen ist definiert als Menge von Knoten, welche untereinander verbunden sind, jedoch keine Verbindungen zu Elementen außerhalb des Clusters haben.

**Dichte basierte Cluster** *Dichte basierte* Cluster sind definiert als Regionen mit einer hohen Dichte an Objekten, welche von Regionen umgeben sind, welche eine geringe Objektdichte besitzen. Elemente, welche in einer solchen Region mit geringen Dichte liegen, welche aus Ausreißer interpretiert. Dichte Bereiche werden üblicherweise gefunden, indem die Nachbarschaften ( $\epsilon$ -*Neighborhoods*) von Elementen untersucht werden.

**Konzeptionelle Cluster** Eine sehr allgemeine Definition eines Clusters ist die der *konzeptionellen* Gruppen. Hiermit ist gemeint, dass die Elemente eines Clusters einige gemeinsame Eigenschaften besitzen. Dies schließt die oben genannten Cluster-Arten mit ein, lässt sich allerdings beliebig erweitern. So sind beispielsweise in Abbildung XXX x) konzeptionelle Cluster dargestellt, die die Form zweier Kreise und eines Dreiecks haben. Um solche Muster erkennen zu können, würde ein Algorithmus eine besondere Definition eines Clusters benötigen.

## 2.4.2 Cluster-Algorithmen

Einige wichtige Clustering-Ansätze sind die Vernetzungs-Modelle, Centroid-basierte-Modelle, Verteilungs-Modelle oder Dichte-Modelle. Diese und einige der stellvertretende Algorithmen werden in nachfolgendem Abschnitt 2.4.2 beschrieben.

## 2.4.3 Distanzmaße zum Vergleich von Fahrzeugtrajektorien

## 2.5 Untersuchung möglicher Straßentopologien



## 3 Stand der Technik

3.1 Clustering von Trajektoriedaten

3.2 Erkennung von Fahrspuren

3.3 Klassifizierung von Fahrspuren

3.4 Defizite vorhandener Lösungen und benötigte Neuerungen

## 4 Clustering von Fahrzeugtrajektorien

### 4.1 Vorverarbeitung der Roh-Trajektorien

### 4.2 Gruppierung der Trajektorien

## 5 Fahrbahn-Bestimmung aus Trajektorie-Clustern

## 6 Fahrbahn Klassifizierung

## 7 Realisierung LaneDetection in MEC-View TrackerApplication Software

## 8 Ergebnisse und Auswertung

## 9 Zusammenfassung und Fazit

# Literaturverzeichnis

- [Cookson u. a. ] COOKSON, G ; RESEARCH, B Pishue I. ; FEBRUARY undefined ; 2017 undefined: Inrix global traffic scorecard. In: *jschultheis.de*. – URL [http://jschultheis.de/wp-content/uploads/2018/02/INRIX\\_2017\\_Traffic\\_Scorecard\\_Report\\_\\_\\_German.pdf](http://jschultheis.de/wp-content/uploads/2018/02/INRIX_2017_Traffic_Scorecard_Report___German.pdf)
- [Divyansh Dwivedi 2018] DIVYANSH DWIVEDI: *Face Detection For Beginners – Towards Data Science*. 2018. – URL <https://towardsdatascience.com/face-detection-for-beginners-e58e8f21aad9>. – Zugriffsdatum: 2018-10-18
- [Grabner u. a. ] GRABNER, H ; GRABNER, M ; BMVC, H B. ; 2006 undefined: Real-time tracking via on-line boosting. In: *grabner.family*. – URL <http://grabner.family/helmut/papers/Grabner2006RealTimeTracking.pdf>
- [Hemmerle 2016] HEMMERLE, P: Empirische physikalische Eigenschaften des übersättigten innerstädtischen Verkehrs und Energieeffizienz von Fahrzeugen. (2016). – URL <https://d-nb.info/1122559259/34>
- [Jain u. a. 1999] JAIN, A. K. ; MURTY, M. N. ; FLYNN, P. J.: Data clustering: a review. In: *ACM Computing Surveys* 31 (1999), sep, Nr. 3, S. 264–323. – URL <http://portal.acm.org/citation.cfm?doid=331499.331504>. – ISSN 03600300
- [Jain 2010] JAIN, Anil K.: Data clustering: 50 years beyond K-means. In: *Pattern Recognition Letters* 31 (2010), jun, Nr. 8, S. 651–666. – URL <https://www.sciencedirect.com/science/article/pii/S0167865509002323>. – ISSN 0167-8655
- [Tan u. a. 2007] TAN, Pang-Ning u. a.: *Introduction to data mining*. Pearson Education India, 2007
- [Viola u. a. ] VIOLA, P ; RECOGNITION, M Jones V. ; PATTERN ; CVPR, 2001. ; 2001 undefined: Rapid object detection using a boosted cascade of simple features. In: *ieeexplore.ieee.org*. – URL <https://ieeexplore.ieee.org/abstract/document/990517/>