



UNIVERSITÄT
LEIPZIG

UNIVERSITÄT LEIPZIG

ABTEILUNG DATENBANKEN

BIG DATA - PRAKTIKUM

Konzeptioneller Entwurf zum
Thema
*"Traffic Analysis with Deep
Learning"*

Ali Al-Ali und Jeremy Puchta

19. Mai 2019

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	II
Tabellenverzeichnis	III
1 Einleitung	1
2 Entwurf	1
2.1 Architektur	2
2.2 Trainieren und Testen	2
2.3 Ablauf	2
Literaturverzeichnis	IV
Abkürzungsverzeichnis	IV
Glossar	V

Abbildungsverzeichnis

Tabellenverzeichnis

1 Einleitung

Die vorliegende Ausarbeitung dokumentiert die Konzeption des Projekts *Traffic Analysis with Deep Learning*.

Das Ziel des Projekts ist die Erstellung einer Webanwendung, welche Daten über das Verkehrsaufkommen am Leipziger Ring sammelt, statistisch analysiert und visualisiert. Als Datengrundlage werden die unter <https://www.l.de/webcam.html> öffentlich bereitgestellten Webcambilder des Leipziger Rings verwendet. Diese werden regelmäßig mithilfe eines Webscraping-Algorithmus abgerufen und dem Datensatz hinzugefügt (**T1**).

Innerhalb der gesammelten Bilddateien werden die für das aktuelle Verkehrsaufkommen relevanten Objekte erfasst. Dazu wird ein für den Anwendungsfall der Objekterkennung vortrainiertes neuronales Netz implementiert. Der Leistungsstand des neuronalen Netzes wird im ersten Schritt untersucht, bevor im zweiten Schritt eine Verbesserung der Leistungsfähigkeit vorgenommen wird (**T2**). Anschließend erfolgt die Transformation der gesammelten Bilddaten in Zeitreihendaten (**T3**).

Abschließend wird der Zeitreihendatenbestand für die nachfolgende Analyse und Visualisierung innerhalb der Webanwendung aufbereitet. Die Webanwendung stellt ein Dashboard zur Verfügung, welches die aufbereiteten Verkehrsdaten grafisch visualisiert (**T4**).

Die aufbereiteten Verkehrsdaten geben Informationen über die gegenwärtige Verkehrslage am Leipziger Ring preis und ermöglichen es Rückschlüsse auf mögliche Über- oder Unterlastungen des betrachteten Verkehrsbereich zu ziehen. Aus diesen Analysen ist es möglich Handlungen für eine Verbesserung der Verkehrssituation abzuleiten. Daher sind die Ergebnisse der Ausarbeitung von Interesse für politische Amtsträger, Studenten verschiedener Fachrichtungen sowie interessierte Bürger der Stadt Leipzig.

2 Entwurf

Im Folgenden soll der Aufbau der Anwendung näher beleuchtet werden. Zunächst erfolgt die Präsentation des verwendeten Technologiestacks. Anschließend werden die eingesetzten Komponenten vorgestellt und ein Überblick über ihre Funktionsweise geliefert. Für einen besseren Überblick über die Anwendung wird die Gesamtarchitektur mit Hilfe eines UML-Komponentendiagramms visualisiert.

2.1 Architektur

Zur Erkennung von relevanten Objekten wird die *YOLO*-Architektur genutzt. Bei *YOLO* handelt es sich um ein Architekturmodell zur Objekterkennung. Die verwendete *YOLO*-Implementierung heißt *Darknet*.

Im Rahmen der Webanwendung wird eine Client-Server-Architektur eingesetzt. Auf Serverseite wird das leichtgewichtige Framework *Flask* genutzt. Im Frontend kommt die von *Facebook* entwickelte JavaScript-Library *React* zum Einsatz. Das Deployment erfolgt mit Hilfe der Containervirtualisierungs-Plattform *Docker*.

2.2 Trainieren und Testen

Zur Verbesserung der Ergebnisse des vortrainierten Modells wird dieses mithilfe einer Vielzahl eigener Daten trainiert. Die Trainingsdaten werden zur ihrer Verwendung vorbereitet, das heißt dass die relevanten Objekte händisch *gelabelt* werden. Als relevante Objekte werden im Projektkontext PKWs, LKWs, Busse sowie Motorräder betrachtet.

2.3 Ablauf

Im ersten Schritt wird das aktuell öffentlich verfügbare Bild der Webcam *gecrawled*. Dieses wird als Input für das neuronale Netz eingesetzt und mit Hilfe des vortrainierten Modells analysiert. Als Ausgabe werden die auf dem Bild befindlichen, als relevant deklarierten, Objekte mit einem *Confidence*-Wert zurückgegeben. Der *Confidence*-Wert stellt die Wahrscheinlichkeit dar, dass es sich um das jeweilige gekennzeichnete Objekt handelt.

Abkürzungsverzeichnis

<falls viele Abkürzungen vorkommen>

TLA Three Letter Acronym

Glossar

Angreifer „Eine Person, die eine ihm bekannte Verwundbarkeit ausnutzt, um ein Computersystem anzugreifen“.