**DIPLOMARBEIT**

Gesamtprojekt

**Entwicklung einer APP**

**Entwicklung einer Verkehrszeichenerkennungsapp für Android/IOS/WEB**

Markus Brandstetter 5BHITS Betreuer: Dipl. -Ing. Gerald Zottl

Schuljahr 2023/24

Abgabevermerk:

Datum: 11.02.2025 übernommen von: Markus Brandstetter

****

**Höhere Technische Bundeslehranstalt Hollabrunn**

**Höhere Lehranstalt für Informationstechnologie**

**EIDESSTATTLICHE ERKLÄRUNG**

**Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, andere als die angegebenen Quellen und Hilfsmittel nicht benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche erkenntlich gemacht habe.**

**Markus Brandstetter**

Hollabrunn, am 1. April 2024

**HINWEISE**

Die vorliegende Diplomarbeit wurde für die Abteilung Informationstechnologie der HTL Hollabrunn ausgeführt.

Die in dieser Diplomarbeit entwickelten Prototypen und Software-Produkte dürfen ganz oder auch in Teilen von Privatpersonen oder Firmen nur dann in Verkehr gebracht werden, wenn sie diese selbst geprüft und für den vorgesehenen Verwendungszweck für geeignet befunden haben.

Es wird keinerlei Haftung übernommen für irgendwelche Schäden, die aus der Nutzung der hier entwickelten oder beschriebenen Bestandteile des Projekts resultieren.

Für alle Entwicklungen gilt die GNU General Public License [http://www.gnu.org/licenses/gpl.html] der Free Software Foundation, Boston, USA in der Version 3.

Absatz überarbeiten bzw. Rundschreiben aktualisieren!

Die Diplomarbeit erfüllt die “Standards für Ingenieur- und Technikerprojekte” entsprechend dem Rundschreiben Nr. 60 aus 1999 des BMBWK (GZ.17.600/101-II/2b/99).

[https://www.bmb.gv.at/ministerium/rs/1999\_60.html]

SCHLÜSSELBEGRIFFE

DANKSAGUNGEN

**Danke an Herr Professor Zottl für die technische und theoretische Expertise und Unterstützung.**

**DIPLOMARBEIT**

**DOKUMENTATION**

|  |  |
| --- | --- |
| Namen der  Verfasser/innen | Markus Brandstetter |
| Jahrgang  Schuljahr | 5BHITS |
| Thema der Diplomarbeit | TrafficSignDetection.ai - Verkehrszeichenerkennung |
| Kooperationspartner | Gerald Zottl |

|  |  |
| --- | --- |
| Aufgabenstellung | **Es soll ein System zur Erkennung von Verkehrszeichen in Echtzeit entwickelt werden. Aus den Videodaten sollen relevaten Verkehrszeichen erfasst, betitelt und verarbeitet werden.**   * Nutzung von Bildverarbeitung und maschinellem Lernen * Interpretation der erkannten Verkehrszeichen * Schwerpunkt auf Geschwindigkeitsbegrenzungen * Klare Darstellung der gültigen Geschwindigkeitsbegrenzung * Fokus auf Übersichtlichkeit und minimaler Ablenkung des Fahrers   Ergebnis: System zur Verbesserung von Fahrerassistenzsystemen und Basis für autonomes Fahren. |

|  |  |
| --- | --- |
| Realisierung | Ein funktionsfähiges Fahrassistenzsystem für Mobile und Web, mit der Verkehrszeichen und Geschwindigkeitsbegrenzungen soll erstellt werden um eine festgelegte Anzahl von Verkehrzeichen zu erkennen. Für die Realisierung wird für das Frontend die Expo (Web/Mobile) Library verwendet und für die Erstellung und Training des Modells ein lokales Python-Programm verwendet. |

|  |  |
| --- | --- |
| Ergebnisse | Es konnte erfolgreich ein Frontend Projekt angelegt werden worin Zugriff auf die Kameras funktioniert. Diese App ist bereits einsatzfähig sowohl auf Android getestet auf einem Android Emulator, bzw. auch im Web. Ein Modell konnte mittels Tensorflow angelegt und erfolgreich trainiert werden. Jedoch mangelt es bei der praktischen Anwendung und Zweck für echte Trainingsdaten wegen Region of Interest. |

|  |  |
| --- | --- |
| Typische Grafik, Foto etc.  (mit Erläuterung) | *Traingsdurchlauf und Auswertung der Ergebnisse des Pythonmodells mittels Konfusion Matrix* |

|  |  |
| --- | --- |
| Teilnahme an Wettbewerben,  Auszeichnungen |  |

|  |  |
| --- | --- |
| Möglichkeiten der Einsichtnahme in die Arbeit | HTL Hollabrunn  Anton Ehrenfriedstraße 10  2020 Hollabrunn |

|  |  |  |
| --- | --- | --- |
| Approbation  (Datum / Unterschrift) | Prüfer/Prüferin | Direktor/Direktorin  Abteilungsvorstand/Abteilungsvorständin |

**DIPLOMA THESIS**

**Documentation**

|  |  |
| --- | --- |
| Author(s) | Markus Brandstetter |
| Form  Academic year | 5BHITS |
| Topic | TrafficSignDetection.ai |
| Co-operation partners | Gerald Zottl |

|  |  |
| --- | --- |
| Assignment of tasks | **A system for recognizing traffic signs in real time is to be developed. Relevant traffic signs are to be recorded, titled and processed from the video data.**  • Use of computer vision and machine learning  • Interpretation of recognized traffic signs  • User-friendly display of recognized signs  • Clear display of the applicable speed limit  • Focus on clarity and minimal driver distraction  The result: a system to improve driver assistance systems and the basis for autonomous driving. |

|  |  |
| --- | --- |
| Realisation | For the implementation, the Expo (Web/Mobile) Library is used for the frontend and a local Python program is used for the creation and training of the model. |

|  |  |
| --- | --- |
| Results | A frontend project was successfully created in which access to the cameras works. This app is already ready for use both on Android tested on an Android emulator and on the web. A model could be created using Tensorflow and successfully trained. However, there is a lack of practical application and purpose for real training data due to region of interest. |

|  |  |
| --- | --- |
| Illustrative graph, photo  (incl. explanation) | *Training run and evaluation of the results of the Python model* |

|  |  |
| --- | --- |
| Participation in competitions  Awards |  |

|  |  |
| --- | --- |
| Accessibility of  final project thesis | HTL Hollabrunn  Anton Ehrenfriedstraße 10  2020 Hollabrunn |

|  |  |  |
| --- | --- | --- |
| Approval  (Date / Signature) | Examiner/s | Head of Department / College |

DA Antrag und unterschriebene Erklärung aus dem [ABA-Portal](https://aba.bildung.gv.at/) einfügen

Inhaltsverzeichnis

[1 Einleitung 10](#_Toc153538383)

[2 Überschrift 1 11](#_Toc153538384)

[2.1 Überschrift 2 11](#_Toc153538385)

[2.1.1 Überschrift 3 11](#_Toc153538386)

[3 Überschrift 1 12](#_Toc153538387)

[3.1 Überschrift 2 12](#_Toc153538388)

[3.1.1 Überschrift 3 12](#_Toc153538389)

[4 Quellen **Fehler! Textmarke nicht definiert.**](#_Toc153538390)

[4.1 Gedruckte Medien **Fehler! Textmarke nicht definiert.**](#_Toc153538391)

[4.2 Online **Fehler! Textmarke nicht definiert.**](#_Toc153538392)

[5 Verzeichnis der Abbildungen 14](#_Toc153538393)

[6 Verzeichnis der Code-Listings 15](#_Toc153538394)

[7 Begleitprotokoll 16](#_Toc153538395)

[8 Anhang 17](#_Toc153538396)

[8.1 Projektdokumentation (UML-Modelle, Sketches, Wireframes, 17](#_Toc153538397)

[8.2 Mockups, Besprechungsprotokolle, Kostendarstellung, etc.) 17](#_Toc153538398)

[8.3 Quellcode Ausschnitte 17](#_Toc153538399)

[8.4 Konfigurationsdateien 17](#_Toc153538400)

[8.5 Sonstiges 17](#_Toc153538401)

Wenn Sie in Ihrer Dokumentation ausschließlich mit den „Formatvorlagen“ dieses Dokuments gearbeitet haben können Sie hier sehr einfach das Inhaltsverzeichnis automatisch aktualisieren lassen.

Klicken Sie dazu mit der rechten Maustaste in das vorhandene Inhaltsverzeichnis und wählen Sie im Kontextmenü „Felder aktualisieren“ und dann „gesamtes Verzeichnis aktualisieren“ auswählen – fertig! … diesen Text löschen!

# Einleitung

Hier beschreiben Sie kurz Ihre Diplomarbeit - Was, Warum, Mit welchem Ziel? …

# Überschrift 1

## Überschrift 2

### Überschrift 3

Lorem ipsum dolor 1

Lorem ipsum dolor 2

Lorem ipsum dolor 3 (Dangl, 2023)

**—** individuelle Zielsetzung und Aufgabenstellung mit Terminplan der einzelnen Teammitglieder

**—** Grundlagen und Methoden (Ist-Situation, Lösungsansätze, Begründung der gewählten Methodik)

**—** Ergebnisse

# Überschrift 1

## Überschrift 2

### Überschrift 3

Beispiele für Quellenangaben:

JavaScript ist bekannt als Scriptsprache für dynamisches HTML in Webbrowsern. Die Sprache kommt abseits von Browsern in Umgebungen wie Node.js, Apache CouchDB und AdobeAcrobat, etc. zum Einsatz. [MOZ23]

Folgender Code ermöglicht …

**Object**[] words = { " ", '3', **null**, "2", 1, "" };  
  
 Arrays.stream( words ) // Erzeugt neuen Stream  
  
 .filter( Objects::nonNull ) // Belasse Nicht-null-Referenzen im Stream  
  
 .map( Objects::toString ) // Konvertiere Objects in Strings  
  
 .map( **String**::trim ) // Schneide Weißraum ab  
  
 .filter( s -> ! s.isEmpty() ) // Belasse nichtleere Elemente im Stream  
  
 .map( Integer::parseInt ) // Konvertiere Strings in Ganzzahlen  
  
 .sorted() // Sortiere die Ganzzahlen  
  
 .forEach( System.out::println ); // 1 2 3

Code-Listing : Verwendung der Java Stream-API (Quelle: [ULL23])

Die folgende Grafik zeigt …

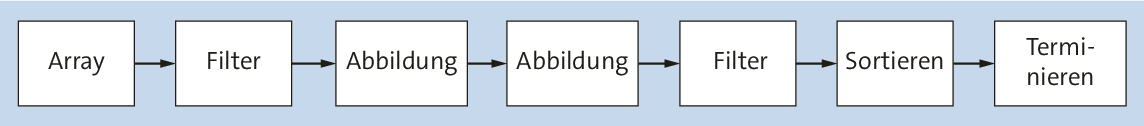


Abbildung : Das Pipeline-Prinzip der Java Stream-API (Quelle: [ULL23])

# Quellen

Dangl, H. (18. 12 2023). *HTL Hollabrunn Homepage*. Von https://htl-hl.ac.at abgerufen

# Verzeichnis der Abbildungen

[Abbildung 1: Das Pipeline-Prinzip der Java Stream-API (Quelle: [ULL23]) 12](#_Toc153536681)

# Verzeichnis der Code-Listings

[Code-Listing 1: Verwendung der Java Stream-API (Quelle: [ULL23]) 12](#_Toc153536763)

# Begleitprotokoll

# Anhang

## Projektdokumentation (UML-Modelle, Sketches, Wireframes,

## Mockups, Besprechungsprotokolle, Kostendarstellung, etc.)

## Quellcode Ausschnitte

## Konfigurationsdateien

## Sonstiges