**[*Intelligente Parkplatzerkennung mit künstlichen neuronalen Netzwerken*]**

**Systembeschreibung**

[1. Ziel und Zweck des Dokumentes 2](#_Toc10836634)

[1.1.1. Projektbeschreibung 2](#_Toc10836635)

[1.1.2. Kurzbeschreibung des Projekts 2](#_Toc10836636)

[1.1.3. Zweck des Projekts 3](#_Toc10836637)

[1.1.4. Hintergrund, Problemstellung, Motivation für das Projekt 3](#_Toc10836638)

[1.1.5. Ziele des Projekts 3](#_Toc10836639)

[1.1.6. Erfolgskriterien 3](#_Toc10836640)

[2. Systemübersicht 4](#_Toc10836641)

[2.1. CNN Frede 4](#_Toc10836642)

[3. Architektur und Designentscheide 5](#_Toc10836643)

[3.1. Modell(e) und Sichten 5](#_Toc10836644)

[3.2. Daten (Mengengerüst & Strukturen) 5](#_Toc10836645)

[3.3. Entwurfsentscheide 5](#_Toc10836646)

[3.4. Randbedingungen und Einschränkungen 5](#_Toc10836647)

[4. Schnittstellen 6](#_Toc10836648)

[4.1. Externe Schnittstellen 6](#_Toc10836649)

[4.2. wichtige interne Schnittstellen 6](#_Toc10836650)

[4.3. Benutzerschnittstelle(n) 6](#_Toc10836651)

[5. Umgebungs-Anforderungen 7](#_Toc10836652)

[5.1. Technologie-Voraussetzungen 7](#_Toc10836653)

[5.2. Kooperierende Anwendungen und COTS-Komponenten 7](#_Toc10836654)

[6. Randbedingungen und Einschränkungen 7](#_Toc10836655)

[7. Entwurfsdetails 7](#_Toc10836656)

[8. Testplan 7](#_Toc10836657)

[8.1. Testziele 7](#_Toc10836658)

[8.2. Testdesign und -automatisierung 7](#_Toc10836659)

[8.3. Vorgehen beim Testen 7](#_Toc10836660)

[8.4. Testfälle, Testkriterien und Testergebnisse 7](#_Toc10836661)

[9. Projektabschluss 8](#_Toc10836662)

Versionen:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rev. | Datum | Autor | Bemerkungen | Status |
| 0.1 | 14.03.2019 | Felix Willrich | 1. Entwurf + Eintragen aller Informationen | Abgeschlossen |
| 0.2 | 08.06.2019 | Felix Willrich |  | In Bearbeitung |

1. Ziel und Zweck des Dokumentes

Dieses Dokument beschreibt die Anforderungen der T-Systems on site services GmbH. Es handelt sich hierbei um die Systemdefinition, die der Auftragnehmer für den Auftraggeber (Kunde) erstellt, sodass der Kunde versteht und validieren kann, was das System leisten wird.

Projektbeschreibung

Dieses Projekt wird im Rahmen des Modules „Teamprojekt“ durchgeführt. welches von Herr Kircher & Frau Schiering doziert wird. Kunde für dieses Projekt ist Herr Philip May, welcher Angestellter bei der T-Systems on site GmbH ist und gleichzeitig die Rolle des Projektfansprechpartners einnimmt.

Kurzbeschreibung des Projekts

Das Projekt folgt einem gewissen Ablauf. Die Daten werden eingelesen, verarbeitet und ausgegeben. Diese drei Schritte werden anhand von folgendem Bild verständlich.

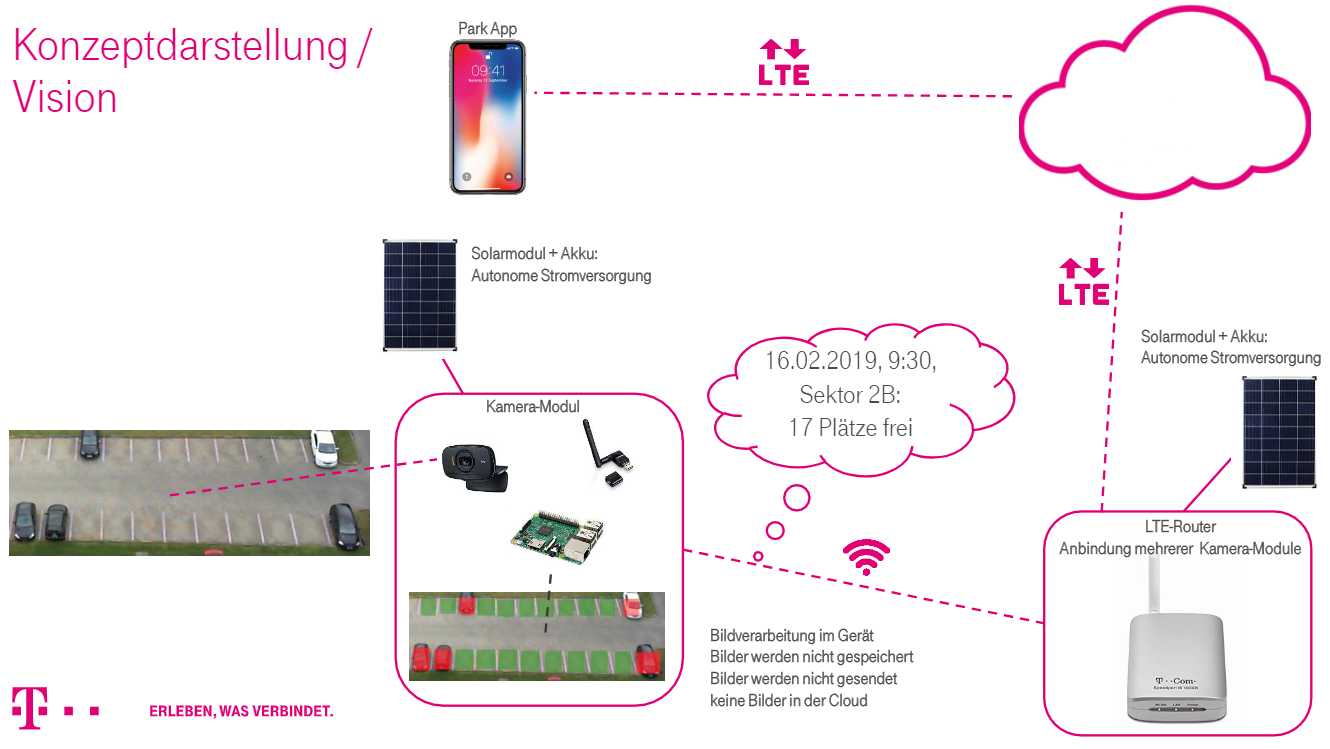


Abbildung 1: Produkt-Zyklus

Eine Kamera überträgt ein regelmäßig aufgezeichnetes Bild von einem Parkplatz an einen Mikrocontroller. Dieser verarbeitet die Daten, indem er das Bild in das konzipierte neuronale Netzwerk schickt. Anhand der Aufnahme soll bestimmt werden, wie viele Parkplätze frei sind. Diese Information wird an eine zentrale Stelle geschickt und weiterverteilt an eine Smartphone-App, damit die Nutzer zu jeder Zeit abrufen können, wohin sie fahren sollten. Die Bilder werden nicht vom Mikrocontroller weitergeschickt.

Da aufgrund der Zeit Abstriche gemacht werden müssen, werden wir uns in diesem Projekt auf die Erkennung von freien Parkplätzen konzentrieren. Das bedeutet, dass wir keine Live-Daten aus der Kamera bekommen werden, bzw. auch keine App erstellen werden.

Zweck des Projekts

Das Projekt soll in erster Instanz zur Erkennung von freien Flächen eingesetzt werden und im Rahmen von Parkplätzen und Parkhäusern genutzt werden. Die Technik kann mit verschiedenen Inputdaten auf verschiedenste Felder ausgeweitet werden. Beispiele wären, Lagerbestände oder den Füllstand von verschiedenen Containern erkennen.

Während des Projekts ist deutlich geworden, dass die Anforderungen geändert werden müssen. Dies lag vornehmlich an dem Zeitaufwand bzw. an der Kommunikation mit dem Kunden. Es wurden verschiedene Annahmen getroffen und zum Sprint 2 hat sich herauskristallisiert, dass dieses Produkt vornehmlich zum Erkenntnisgewinn für den Kunden und uns gelten soll. Damit keine Unklarheiten aufkommen, wurden die Ziele aus dem ersten Gespräch mit dem Kunden formuliert. Im weiteren Verlauf der Dokumentation werden die Testergebnisse und Erkenntnisse niedergeschrieben.

Hintergrund, Problemstellung, Motivation für das Projekt

Die T-Systems on site services GmbH in Person von Philip May benutzt im produktiven Sektor verschiedene Machine Learning/Deep Learning Applikationen und möchten durch dieses Produkt in weitere Felder stoßen bzw. weitere Erkenntnisse darüber gewinnen.

Für die Gruppe ergibt sich aufgrund von wenig Vorkenntnissen folgende Probleme:

* Neues Umfeld kennen lernen
* Geeignete Tools und Umgebung finden
* Datenbeschaffung zum Anlernen
* Prototypen erschaffen
* Genaue Erkennung implementieren
* Testumgebung

Die meisten Probleme werden oder wurden mit unserem Ansprechpartner besprochen und

teilweise aufgearbeitet.

Die Motivation zu diesem Projekt ergibt sich aus dem ersten Stichpunkt der Probleme. Die Gruppe möchte in ein neues, aufstrebendes und sehr interessantes Thema einsteigen und dabei gleichzeitig Praxiserfahrung sammeln.

Ziele des Projekts

Das Projekt wird zuerst bis zu dem Schritt entwickelt bis das Netz angelernt ist

und verschiedene Parkplatzsituationen erkannt werden. Eine Genauigkeit von 99% wird

angestrebt.

Erfolgskriterien

Sollte die gewünschte Genauigkeit erlangt worden sein, wird das Projekt als Erfolg bezeichnet. Weiterhin hinzukommen würden verschiedene Umgebungen, wie Schnee, Regen und andere Hindernisse wie Baulöcher oder Belegung von zwei Parkplätzen gleichzeitig. Sollten diese zusätzlichen Kriterien erfüllt werden, wird das Projekt in vollem Umfang als Erfolg gewertet. Es wird eine möglichst genaue Erkennung mit allen unterschiedlichen Faktoren angestrebt. Systemübersicht

2. Systemübersicht

*Übersicht und Begründung des gewählten Lösungsansatzes*

## 2.1. CNN(Frede)

## 2.2. Skripte(Felix)

1. Architektur und Designentscheide

*<Beschreibung der Software-Architektur (Ist-Zustand)*

* *Zu allen Unterpunkten*
* *Motivation und Ziele*
* *Beschreibung*
* *Entscheidungen und Bewertung*
* *Ggf. Beispiel*
* *>*

## Modell(e) und Sichten

* *Architekturmodelle zur Beschreibung der Struktur der Software,*
* *Dekomposition in Subsysteme*
* *ggf dynamisches Modell,*
* *Zugangskontrolle und Security*

<tbd>

## Daten (Mengengerüst & Strukturen) Felix

* *Datenmodell,*
* *Definition wichtiger Begriffe,*
* *ggf Data Dictionary*
* *Handhabung persistenter Daten*

Alle Daten stammen aus zwei verschiedenen Quellen. Die erste Quelle (<https://web.inf.ufpr.br/vri/databases/parking-lot-database/>) wurde von unserem Kunden vorgegeben. Diese Daten stammen von der Informatik Fakultät der brasilianischen Universität «Universidade Federal do Paraná». In diesem Paket sind 2 Parkplätze über einen längeren Zeitraum aufgenommen worden. Ein Parkplatz dabei aus zwei unterschiedlichen Kamerawinkeln. Dabei enthält das Paket die Aufteilung in Wettersituationen (sunny, cloudy, rainy), Tagen und in den jeweiligen Status des Parkplatzes (empty, occupied). Alle einzelnen Parkplätze wurden ausgeschnitten und in die einzelnen Ordner sortiert. Die ausgeschnittenen Bilder besitzen unterschiedliche Größen. Gleichzeitig gibt es die jeweiligen Gesamtbilder der Parkplätze mit einer dazugehörigen XML-Datei. Diese beinhaltet u.a. die Koordinaten, Größe sowie einen Booleanwert, ob der Parkplatz besetzt (1) oder frei ist (0). Insgesamt beinhaltet diese Paket 700.000 Bilder.

Der zweite Parkplatz (<http://cnrpark.it/>) ist bei Recherche zu Alternativen gefunden worden. Dieses Paket umfasst 150.000 Bilder und nimmt Parkplätze mit neun verschiedenen Kameras auf. Auch hier wurden die Bilder in Wettersituation und Tage aufgeteilt. Der Unterschied in der Datenstruktur liegt dabei, dass die Bilder noch in Kameras aufgeteilt werden. Damit bestimmt werden kann, welche Parkplätze frei oder belegt sind, wurde eine .txt Datei angelegt mit Pfaden und einer Zahl am Ende.

Im Laufe des Projekts wurden diverse unterschiedliche Pakete angelegt, die verschiedenen Testsituationen zu Gute gekommen ist. Die Daten wurden jeweils aufgeteilt in train/validation Daten. Alle Daten wurden zufällig tageweise zugeordnet, damit keine doppelten Bilder vorkommen.

|  |  |
| --- | --- |
| **Name** | **Beschreibung** |
| UFPR04\_05\_train\_val\_test.zip | Zum Ende hin bevorzugtes Paket.  80/20/20 train/validation/test split  Insgesamt 66049 Bilder |
| PUC\_50k.zip | 50000 Bilder des PUC Parkplatzes vom Parkinglot Dataset in 80/20 Verteilung. |
| UFPR05\_50k.zip | 50000 Bilder des UFPR05 Parkplatzes vom Parkinglot Dataset in 80/20 Verteilung. |
| UFPR04\_50k.zip | 50000 Bilder des UFPR04 Parkplatzes vom Parkinglot Dataset in 80/20 Verteilung. |
| 50k\_All\_Parking\_Spaces.zip | 50000 Bilder von allen Parkplkätzen |
| ALL\_ALL.zip | Alle Parkplätze in 80/20 Split. 837991 Bilder |
| CNR\_TEST.zip | Paket, um den Testgenerator mit einem anderen Datensatz zu laden. Insgesamt 144965. Kein Split. |
| ALL\_100K.zip | Zufällige Verteilung der Bilder (100000) der Parkplätze PUC/UFPR05/UFPR04/CNR in 80/20 Verteilung. |
| ALL\_50K.zip | Zufällige Verteilung der Bilder (50000) der Parkplätze PUC/UFPR05/UFPR04/CNR in 80/20 Verteilung. |
| PUC\_UFPR05\_04\_50\_50.zip | Alle Bilder (695851) der Parkplätze PUC/UFPR05/UFPR04 vom Parkinglot Dataset in 50/50 Verteilung. |
| PUC\_50\_50.zip | Alle Bilder (424223) des Parkplatzes PUC vom Parkinglot Dataset in 50/50 Verteilung. |
| UFPR05\_50\_50.zip | Alle Bilder (165785) des Parkplatzes UFPR05 vom Parkinglot Dataset in 50/50 Verteilung. |
| UFPR04\_50\_50.zip | Alle Bilder (103522) des Parkplatzes UFPR04 vom Parkinglot Dataset in 50/50 Verteilung. |
| PUC.zip | Alle Bilder (424223) des Parkplatzes PUC vom Parkinglot Dataset in 80/20 Verteilung. |
| UFPR04\_05.zip | Alle Bilder (261956) der Parkplätze UFPR04/UFPR05 vom Parkinglot Dataset in 80/20 Verteilung. |
| UFPR05.zip | Alle Bilder (165785) des Parkplatzes UFPR05 vom Parkinglot Dataset in 80/20 Verteilung. |
| UFPR04.zip | Alle Bilder (103522) des Parkplatzes UFPR04 vom Parkinglot Dataset in 80/20 Verteilung. |

Alle Pakete besitzen eine einheitliche Datenstruktur. Im Stammverzeichnis der Pakete befinden sich die Unterordner train und val. Der train-Ordner beinhaltet alle Bilder, die das Netz lernen soll und der val-Ordner alle, um das Netz zu überprüfen ob es richtig lernt. In den Ordnern befinden sich nochmals die Ordner empty und occupied. Damit sind gleichzeitig alle Bilder mit Labels ausgestattet.

Alle Pakete werden per Befehl aus OneDrive heruntergeladen, entpackt und in das Netz geladen.



Sobald diese Daten entpackt worden sind befindet sich die Struktur in dem jeweiligen Notebook. Damit die Bilder geladen werden können, wird programmatisch der Pfad angelegt.



## Entwurfsentscheide

<tbd>

## Randbedingungen und Einschränkungen

1. Schnittstellen

*bei allen Unterpunkten*

* *Motivation und Ziele*
* *Beschreibung*
* *Entscheidungen und Bewertung*
* *Ggf. Beispiel*

## Externe Schnittstellen

<tbd>

## wichtige interne Schnittstellen

<tbd>

## Benutzerschnittstelle(n)

<tbd>

1. Umgebungs-Anforderungen

## Technologie-Voraussetzungen

*HW, BS, VM*

Das System ist geschlossen aufgebaut und benötigt dadurch wenig Technologien. Im Folgenden werden die eingesetzten Technologien erklärt.

|  |  |
| --- | --- |
| Name | Beschreibung |
| Python 3.X | Grundlage jeglicher Programmierung |
| Keras | Paket für maschinelles Lernen in Python geschrieben |
| NumPy | Paket, um das Rechnen mit Matrizen und Vektoren zu vereinfachen, Geschrieben in NumPy und wird passiv mitgenutzt |
| Matplotlib | Library für die Darstellung von Diagrammen |
| Tensorflow | Keras stützt sich auf Tensorflow |
| Jupyter Notebook | Sozusagen die IDE |
| Foto Datenbank | <https://web.inf.ufpr.br/vri/databases/parking-lot-database/>  <http://cnrpark.it/>  Grundlage zum Anlernen |
| Google Colab | Hardware Grundlage |
| One Drive | Speichermedium für die Bildpakete |

Für unser Programm waren vor allem Keras bzw. Tensorflow und die Umgebung Google Colab wichtig. Keras bietet uns auf einer Hochsprachen-Ebene diverse Funktionen, um unser Netz zu konzipieren. Google Colab hingegen ist eine Plattform auf der potenten Hardware kostenlos zur Verfügung gestellt wird. Es können dort Notebooks hochgeladen werden und durchgeführt werden. Dies haben wir benutzt, da kein Teammitglied zuhause ausreichende Hardware zur Verfügung stehen hatte.

## Kooperierende Anwendungen und COTS-Komponenten

*<Externe Anwendungen, mit denen die Software zusammenarbeitet oder Commercial of the Shelf (COTS) Komponenten, die verwendet werden>*

*Das Notebook greift auf die hochgeladenen Pakete im OneDrive zu. OneDrive wird nur als Speichermedium genutzt, damit alle Teammitglieder jederzeit die Möglichkeiten haben, die Pakete herunterzuladen. Somit ist nur das Notebook von Nöten. Dazu wird ein Befehl ausgeführt, der sich per «wget» das Paket herunterlädt und danach entpackt. Die Pakete sind unter dem Punkt 3.2. genauer erklärt.*



1. Randbedingungen und Einschränkungen
2. Entwurfsdetails

*Detaillierte Beschreibung wichtiger Einzelteile*

* *Pakete,*
* *Klassen,*
* *Objekte*
* *...*

1. Testplan

Das Projekt sieht vor wie Eingangs erklärt ein neuronales Netzwerk aufzubauen zur Parkplatzerkennung. Da es in diesem Kontext keine sinnvollen Möglichkeiten gibt das Modell zu testen, wurden die Tests abgewandelt. Die Tests konzentrieren sich vor allem auf verschiedene Parameter und deren Wirkung auf das Netz und die Genauigkeit. Jedes Teammitglied hat sich dabei auf verschiedene Eigenschaften konzentriert.

## Frede

## Pit

## Jascha

## Felix

1. Projektabschluss

*Ein letztes Update der Projektmanagementvorlage auf den Projekt-Schlussstand, dabei sind insb die folgenden Absätze und Kapitel zu aktualisieren*

**Absatz 2.2 Projektkontrolle und Projektsteuerung**

* *Getroffene Maßnahmen und ihre Auswirkungen im Projektverlauf.*

**PM - Anhänge**

<für den Projektabschluss sind abzugeben >

* Sprintpläne
* Sprintreview-Protokolle
* Meilensteinberichte
* **Aufgaben und Zeitaufwände aller Teammitglieder dargestellt über den Verlauf des Projekts**