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

**Sprint 2 Review**

[1. Arbeitspakete 2](#_Toc8749489)

[1.1. Tests 2](#_Toc8749490)

[1.1.1. Batch-Normalization 2](#_Toc8749491)

[1.1.2. Hyperparameter optmieren 2](#_Toc8749492)

[1.1.3. Verschiedene Inputgrößen testen 2](#_Toc8749493)

[1.1.4. Generator nutzen 2](#_Toc8749494)

[1.1.5. Augmentation 2](#_Toc8749495)

[1.1.6. Weitere Tests 2](#_Toc8749496)

[1.2. Hilfsarbeiten 3](#_Toc8749497)

[1.2.1. Bilder Ordnerstruktur 3](#_Toc8749498)

[1.2.2. Ausschneiden der Parkplätze überarbeiten 4](#_Toc8749499)

[1.2.3. Skript zum Ermitteln der Größe der Bilder 4](#_Toc8749500)

[1.2.4. Collab einarbeiten 4](#_Toc8749501)

[2. Use-Cases 5](#_Toc8749502)

[3. Entwicklerreview 6](#_Toc8749503)

[3.1. Frederik Rieß 6](#_Toc8749504)

[3.2. Pit Ehlers 6](#_Toc8749505)

[3.3. Jascha Schmidt 6](#_Toc8749506)

[3.4. Felix Willrich 6](#_Toc8749507)

Versionen:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rev. | Datum | Autor | Bemerkungen | Status |
| 0.1 | 14.05.2019 | Felix Willrich | Erstellen des Sprint Reviews | Abgeschlossen |

# Arbeitspakete

Der zweite Sprint sah vor, diverse Tests durchzuführen, um Erkenntnisse über die Grundlagen bzw. Eigenheiten des Netzes zu erkennen. Dazu wurden im Product Backlog diverse Pakete konzipiert, um Tests durchführen zu können.

Grundlegend sind die Pakete nicht alle vollständig abgeschlossen worden und im Gespräch mit dem Kunden am 14.05.2019, welches zur Nachbereitung des Sprint 2 und Vorbereitung von Sprint 3 dienen sollte, wurde festgelegt, dass die Pakete weitergeführt werden. Somit werden diesen Sprint nur Zwischenergebnisse dokumentiert.

## Tests

Die Tests wurden konzipiert um Erkenntnisse über das Netz gewinnen, bzw. die optimalen Einstellungen zu finden.

### Batch-Normalization

Verantwortlicher: Jascha Schmidt

### Hyperparameter optimieren

Verantwortlicher: Frederik Rieß

Für die Optimierung der Hyperparameter sollte die Library Hyperopt hinzugezogen werden. Dadurch können verschiedene Ranges (zum Beispiel die Anzahl der Filter) eingegeben werden und das Model testet dieses, sodass am Ende die optimalen Werte genommen werden können.

Allerdings ist dies mit Google Colab aufgrund der großen Datenmenge nicht möglich gewesen. Auch deshalb wird der Datensatz für das Model reduziert, um dann im Sprint 3 diese Optimierungen durchzuführen. Stattdessen konnte in diesem Sprint festgestellt werden, dass bereits ein wenig komplexes Model eine Validierungsgenauigkeit von 99% erreicht. Um Overfitting zu vermeiden, muss das Model also nochmal überarbeitet werden.

### Verschiedene Inputgrößen testen

Verantwortlicher: Pit Ehlers

### Generator nutzen

Verantwortlicher: Frederik Rieß

Durch das Ersetzen der ursprünglichen Datenerfassung mit einem Generator wurde das Labeln der Daten stark vereinfacht. Zudem kann die spätere Augmentation dadurch besser implementiert werden, als wenn externe Bibliotheken importiert werden müssten.

Unter anderem muss nur das Directory der Daten, die Größe der Bilder und die Batch-Size angegeben werden. Zudem sollte der Generator den Modus «binary» haben, da dieser auf unsere Problemstellung zutrifft. Dieser ImageDataGenerator kann nun noch weitere Parameter erhalten, um die Bilder mit Augmentation zu bearbeiten.

### Augmentation

Verantwortlicher: Felix Willrich

### Weitere Tests

Verantwortliche: Alle

## Hilfsarbeiten

Die Hilfsarbeiten wurden konzipiert, um die eigentlichen Tests zu unterstützen.

### Bilder Ordnerstruktur

Verantwortlicher: Felix Willrich

Damit das Netz getestet werden konnte, mussten verschiedene Pakete mit Bildern erstellt werden. Die Struktur wurde in «train» und «test» eingeteilt. Jeder dieser Ordner beinhaltete zwei weitere Ordner «Empt» und «Occpuied». Dies hatte den Vorteil, dass das Netz auf diese Struktur angepasst wurde und gleichzeitig als Labels angesehen wurden.

Damit die Pakete für jeden erreichbar waren, wurden diese auf OneDrive hochgeladen. Eine Anleitung wie auf diese Pakete zugegriffen werden kann, ist unter «Notebooks\README\_Download.txt» zu finden.

Zum Schluss dieses Sprints wurde ein Paket mit ca. 700.000 Bildern in einer 50/50 Aufteilung für «test» und «train» genutzt zum Trainieren. Dieses Paket bestand aus zwei Parkplätzen, woraus ein Parkplatz aus zwei unterschiedlichen Winkeln aufgenommen worden ist.

Um einen fremden Parkplatz zu testen wurde aus dem ersten Sprint der CNR Parkplatz benutzt

(<http://cnrpark.it/>)

Folgende Pakete wurden angelegt:

|  |  |
| --- | --- |
| **Name** | **Beschreibung** |
| CNR\_TEST | Daten von <http://cnrpark.it/>  Nur «test» Ordner vorhanden zum Gegentesten des Netzes |
| PUC | Daten von [https://web.inf.ufpr.br](https://web.inf.ufpr.br/) Trainingsdaten: 80% Validation 20% |
| PUC\_50\_50 | Daten von [https://web.inf.ufpr.br](https://web.inf.ufpr.br/) Trainingsdaten: 50% Validation 50% |
| PUC\_UFPR05\_04\_50\_50 | Daten von [https://web.inf.ufpr.br](https://web.inf.ufpr.br/) Kombination aus drei Parkplätzen Trainingsdaten: 50% Validation 50% |
| UFPR04 | Daten von [https://web.inf.ufpr.br](https://web.inf.ufpr.br/) Trainingsdaten: 80% Validation 20% |
| UFPR04\_05 | Daten von [https://web.inf.ufpr.br](https://web.inf.ufpr.br/) Kombination aus zwei Parkplätzen Trainingsdaten: 80% Validation 20% |
| UFPR04\_50\_50 | Daten von [https://web.inf.ufpr.br](https://web.inf.ufpr.br/) Trainingsdaten: 50% Validation 50% |
| UFPR05 | Daten von [https://web.inf.ufpr.br](https://web.inf.ufpr.br/) Trainingsdaten: 80% Validation 20% |
| UFPR05\_50\_50 | Daten von [https://web.inf.ufpr.br](https://web.inf.ufpr.br/) Trainingsdaten: 50% Validation 50% |

Das Paket wird weitergeführt in Sprint 3.

### Ausschneiden der Parkplätze überarbeiten

Verantwortlicher: Felix Willrich (nachträglich Frederik Rieß)

Da das vorherige Ausschneiden der einzelnen Parkplätze nicht optimal gelungen war, musste dieses durch ein Skript angepasst werden. Als neuer Ansatz wurde dabei die Bibliothek OpenCV genutzt, womit die Konturen besser ausgeschnitten werden konnten. Dabei wird innerhalb der 4 Koordinaten des Parkplatzes das größtmögliche Rechteck gewählt und ausgeschnitten. Anschließend wird das ausgeschnittene Bild noch entsprechend gedreht.

Dadurch entsteht ein kleiner Verlust des Bildes an den Seiten. Diese mögliche Variante könnte zukünftig mit anderen Verfahren verglichen werden.

### Skript zum Ermitteln der Größe der Bilder

Verantwortlicher: Pit Ehlers (nachträglich Felix Willrich)

Um den Test «Verschiedene Inputgrößen testen» zu unterstützen wurde ein Skript geschrieben, welches die Bilder der ausgeschnittenen Parkplätze durchiteriert und die verschiedenen Größen in eine Txt-Datei schreibt. Dies hilft dabei, die Bilder besser zu analysieren und Vorgaben für das Input-Shape im Netz zu haben.

Das erstellte Skript liegt im Repository unter «Skripte/picturesize.py». Das Ergebnis unter «Notebooks/avg\_picture\_size.txt».

Das Paket wird als abgeschlossen betrachtet.

### Collab einarbeiten

Verantwortliche: Alle

Da zurzeit keine Hardware der Gruppe zur Verfügung steht, musste auf eine Alternative zurückgegriffen werden. Google stellt eine Umgebung mit potenter Hardware zur Verfügung, die mit einem Google Konto benutzt werden konnte. Jeder der Mitglieder der Gruppe hatte während des Sprint 2 damit gearbeitet, damit sollte die Einarbeitung abgeschlossen sein.

Die einzige Einstellung, die getroffen werden musste, bezog sich auf die Nutzung der GPU anstatt der CPU.

Probleme traten vor allem bei der längeren Nutzung auf, da die Session die Verbindung unterbrochen hat. Es konnte ein reconnect durchgeführt werden, aber hierzu musste der Rechner immer im Auge behalten werden. Die Umgebung wird weiterhin genutzt, da Anpassungen an den Trainingsdaten durchgeführt worden sind. Sollte einmal ein längerer Test beabsichtigt werden, kann vom Kunden eine Umgebung bereitgestellt werden.

Das Paket wird als abgeschlossen betrachtet.

# Use-Cases

# Entwicklerreview

## Frederik Rieß

Zu meinen Aufgaben in diesem Sprint gehörte es zunächst, das Ausschneiden der Bilder zu optimieren. Mit der Bibliothek OpenCV gab es hier keine Probleme, allerdings kann das Ausschneiden zukünftig mit anderen Verfahren noch verglichen werden, um das bestmögliche Ergebnis zu erzielen.

Außerdem musste ein Generator implementiert werden, der die Datenerfassung erleichtert. Auch hier gab es keinerlei Probleme, da ich mich an Vorgaben gehalten habe.

Das größte Problem bestand in dem Testen des neuronalen Netzes. Da Google Colab den Client nach einer gewissen Nutzungsdauer disconnected, musste teilweise das Model über 5 Stunden kontrolliert werden. Auf Dauer erwies sich dies nicht als praktikabel, vor allem da das Testen mit der Bibliothek HyperOpt noch mehr Zeit in Anspruch nimmt. Daher wird zukünftig auf ein kleineren Datensatz zurückgegriffen.

Nach den Absprachen im Team war relativ schnell klar, wer was zu erledigen hatte. Das unabhängige Testen erwies sich nun leider als relativ unnötig, da wir letzten Endes auf einen kleineren Datensatz umsteigen. Allerdings konnte durch die hohe Validation-Accuracy gleich zu Beginn des Trainierens gesehen werden, dass die Aufteilung der Trainings- und Testdaten nicht gut gewählt war und die Variation an Daten deutlich erhöht werden muss.

Die Arbeit untereinander empfand ich als etwas zu ruhig, da unter anderem die Ergebnisse der jeweiligen Tests erst gegen Ende des Sprints besprochen wurden. So konnten keine aktuellen Zwischenstände bzw. Erkenntnisse mitgeteilt werden.

## Pit Ehlers

## Jascha Schmidt

Sprint Review Jascha Schmidt

Aufgaben:

* Den Einfluss von einer Batchnormalisierung der Layer auf unser CNN ermitteln

Vorgehen:

* Die Batchnormalisierung einbauen.
* Durch aktivieren/deaktivieren der Batchnormalisierung in einem oder beiden Layers und anschließendes dokumentieren der Ergebnisse den Einfluss auf das CNN bestimmen

Probleme:

* In diesem Sprint sind bei mir keine größeren Probleme aufgetreten

## Felix Willrich