

Advanced Computer Engineering

Missionsziel, -vorgaben und -anforderungen

Meilenstein 1

Gruppe A

Moritz Lechner,

Leon Sobotta,

Konstantin Roßmann

INHALT

Zweck und Umfang	2
Einleitung	2
Ziele	4
Use CaseE	5
Anforderungen.....	6
Glossar	8
Abbildungsverzeichnis.....	8
Literaturverzeichnis.....	8

ZWECK UND UMFANG

Das Dokument beschreibt die Ziele der Mission sowie deren dazugehörigen Vorgaben und Anforderungen.

Es wird im Rahmen des Modul Advanced Computer Engineering eine Applikation für eine voraussichtlich 2024 stattfindende Satellitenmission des Deutschen Zentrums für Luft- und Raumfahrt. Die Applikationen werden für eine neue verteilte und rekonfigurierbare Rechnerarchitektur entwickelt. [1]

Ziel des „Scaleable On-Board System for Avionics (ScOSA)“ ist es ausfallsichere On Board Computer (OBC) für den Einsatz im Raumfahrtkontext zu entwickeln. Dieser OBC soll durch redundante verteilte Prozessoren zuverlässiger als bereits existierende OBCs arbeiten. Die Rechenleistung soll dabei skalierbar sein, um den sich stetig verändernden Anforderungen der Raumfahrtindustrie gerecht zu werden. Dazu werden verschiedene Hardware- sowie Softwaretechniken verwendet. Durch die Redundanz können bei einem Ausfall eines oder mehrerer Prozessoren, die Übrigen die Aufgaben übernehmen. Dabei wird stets darauf geachtet, dass vor allem die höher priorisierten Prozesse weiterlaufen. [2]

EINLEITUNG

Durch den Klimawandel werden Vegetationsbrände immer häufiger. Die Brandsaison dauert, durch die länger anhaltenden hohen Temperaturen und die daraus resultierenden Trockenperioden, immer länger an. Daher wird vermehrt in die Erforschung der Ursachen investiert. [3]

Zur Auswertung der Daten können künstlichen neuronalen Netze (KNN) eine Unterstützung sein. Durch die Verwendung von KNN auf dem OBC eines Satelliten könnte eine schnelle Reaktion sowie Auswertung des Brandes ermöglicht werden. [4]



Abbildung 1 Bild eines Brands Credits: NASA Worldview [5]

Um diese Möglichkeit zu testen, soll eine Applikation entwickelt werden, die mithilfe eines KNN Vegetationsbrände erkennt. Sobald der Satellit über ein Gebiet mit Vegetationsbränden fliegt, werden die Bilder der Kamera an den Computer übermittelt und dort ausgewertet, sollte ein Vegetationsbrand erkannt werden, wird dieser gespeichert und sobald es möglich ist zur Erde übertragen. Die übermittelten Daten sollen Aufschluss darüber geben, ob sich ein kompaktes KNN eignet, um Vegetationsbrände automatisch zu erkennen z.B. für Waldbrandwarnungen oder zu Forschungszwecken und ob die Methode besser im Vergleich zu anderer Bilderkennungsmethoden, wie der Farberkennung ist.

Die Farberkennung wird mithilfe einer Farbmaske umgesetzt. Diese Maske wird wie eine Schablone auf das Bild gelegt und alles was nicht in dem vorher definierten Farbspektrum liegt wird entfernt. Der Algorithmus ist bei dieser Technik deutlich geringer in der Komplexität und der Laufzeit und damit verbraucht sie auch deutlich weniger Ressourcen, was gerade bei Raumfahrtmissionen vorteilhaft sein kann. Bei der Zuverlässigkeit wird diese Technik an ihre Grenzen stoßen, da die Bilder nur eine endliche Auflösung und die Kamera nur eine endliche Farbegenauigkeit haben. [6] Bei der Genauigkeit hat das KNN seine Stärken.

Die Herausforderung wird es sein, einen Datensatz zu erstellen, der die Möglichen Inputs abdeckt. Da der Rauch verschiedenen Formen und Farben an nehmen kann, außerdem ist die Umgebung vom Brand auch immer verschieden. Es wird schwer werden diese Randfälle abzudecken.



Abbildung 3 Bild von Brand in Kalifornien
[8]



Abbildung 2 Bild von Brand in Australien
[9]

ZIELE

Die Missionsziele definieren die Kriterien zur Erstellung der Anforderungen und Aufgaben an das System.

Nr.	Ziel
MG-01	Die Mission wird zeigen, ob sich KNN eignen, um Satellitenbilder nach Vegetationsbränden zu untersuchen.
MO-01	Die Mission wird zeigen, dass ein kompaktes KNN besser geeignet ist, als andere Algorithmen um Vegetationsbrände zu erkennen.
MO-02	Die Mission wird zeigen, ob KNN zu rechenintensiv sind im Vergleich zur Genauigkeit.
MO-03	Die Mission wird zeigen, dass Rust auch für Neuronale Netze verwendbar ist.

USE CASEE

Sobald der Task gestartet wird und der Datenimport startet, wird automatisch die Erkennung des Inputs gestartet. Immer wenn ein Bild analysiert wurde, wird das Bild und der erkannte Inhalt gespeichert. Das Programm läuft solange bis der Prozess wieder deaktiviert wird.

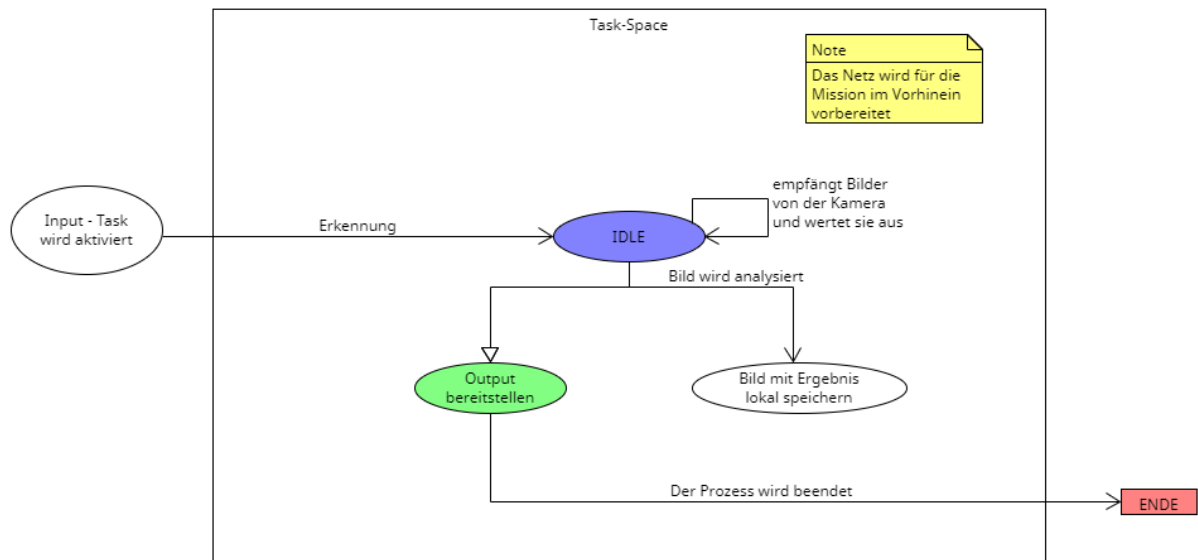


Abbildung 4 Use Case Darstellung

Das Programm soll automatisch laufen und wird nur gestartet und beendet. Der Output wird zusätzlich intern gespeichert.

ANFORDERUNGEN

Die Anforderungen sollen die Rahmenbedingungen für die Mission definieren.

Nr.	Anforderungsbeschreibung	Verifikation	Bestanden *
SW-0001	Wenn ein Vegetationsbrand fotografiert wird, muss dieser erkannt werden.	Testdurchlauf des Programms mithilfe des Datensatzes, sowie darauffolgende Stichproben.	-
SW-0002	Das Neuronale Netz muss vor dem Start trainiert werden.	Test des Programms mit gespeicherten Gewichtungen	-
SW-0003	Die Software soll in der Programmiersprache Rust geschrieben werden.	Sichtprobe	-
SW-0004	Das Programm muss auf einem RaspberryPi laufen.	Test auf dem RaspberryPi oder anderen Linux Betriebssystem	-
SW-0005	Das Programm muss nach der Aktivierung automatisch laufen.	Testdurchlauf des Programms	-
SW-0006	Der Satellit muss eine Kamera haben.	Absprache mit Ansprechperson für dieses Modul.	-
SW-0007	Nach dem Trainieren soll das Netz minimiert werden.	Testdurchlauf vor und nach dem minimieren, sowie vergleich der Daten, mithilfe von Test-Datensatz.	-
SW-0008	Die Auflösung der Bilder sollen verkleinert werden.	Testdurchlauf vor und nach dem verkleinern, sowie vergleich der Daten.	-
SW-0009	Das Programm soll nach dem Analysieren die Bilder speichern.	Unit-Test, Stichprobe im Speicherort	-
SW-0010	Wenn das Programm die Bilder speichert, soll es die Bilder komprimieren.	Nach erfolgreichem Test von SW-0009, Unit-Test mit Komprimierung und ohne.	-
SW-0011	Wenn ein Brand erkannt wird, können die Koordinaten des Brandes gespeichert werden.	Unit-Test	-

SW-0012	Die Benutzerdokumentation soll in den Sprachen Deutsch oder Englisch sein.	Sichtprobe	-
SW-0013	Wenn ein Bild gespeichert wird, muss es gelabelt werden.	Unit-Test der Label-Funktion, unter Annahme erfolgreichem Erkennen.	-
SW-0014	Wenn ein Brand gelabelt wird, soll der Wert die Genauigkeit beinhalten.	Rohausgabe von FeedForward mit der Ausgabe von der Label-Funktion vergleichen	-
SW-0015	Die Ausgabe des Programmes soll ein Float-Value sein.	Sichtprobe im Quell-Code	-
SW-0016	Die Ausgabe des Programmes soll ein Bild sein.	Unit-Test der Output-Funktion mit Testbild	-
SW-0017	Die Bilder sollen im .jpg Dateiformat gespeichert werden.	Sichtprobe nach Unit-Test	-
SW-0018	Die Bilder können als Bit-Array gespeichert werden.	Unit-Test vom Speichern	-
SW-0019	Das Programm soll im Betrieb einen FeedForward Algorithmus benutzen.	Sichtprobe	-
SW-0020	Das Programm muss ein Bild als Input bekommen.	Absprache mit Ansprechperson für dieses Modul.	-
SW-0021	Wenn das Netz trainiert wurde, sollen die Gewichte im CSV-Format gespeichert werden.	Sichtprobe	-

* - : nicht durchgeführt | x : bestanden | ~ : durchgeführt, aber nicht bestanden (wird wiederholt) | /~ : getestet und entfällt

GLOSSAR

KÜNSTLICHES NEURONALES NETZ (KNN)

Ein Algorithmus der mithilfe eines vorher erstellten Datensatz „trainiert“ wird. Nach dem Training kann das KNN auf das gelernte zugreifen und mit neuen Daten vergleichen. (Auch Maschinelles Lernen)

BRAND (HIER IMMER VEGETATIONSBRAND)

Vegetationsbrände ist der Sammelbegriff für verschiedene Arten von Bränden in der freien Natur. [7]

BILDER

Hier ist der Input der Kamera in unserm Programm gemeint, sowie während des gesamten Bearbeitungsprozesses bis hin zum Speichern.

LABEL

Hier ist Wert gemeint, der definiert, ob es sich um einen Vegetationsbrand handelt oder nicht.

FEEDFORWARD

Ist ein Verfahren zur Berechnung des Outputs in einem Neuronalen Netz

ABBILDUNGSVERZEICHNIS

Abbildung 1 Bild eines Brands <i>Credits: NASA Worldview</i> [5]	2
Abbildung 2 Bild von Brand in Australien [9]	3
Abbildung 3 Bild von Brand in Kalifornien [8]	3
Abbildung 4 Use Case Darstellung	5

LITERATURVERZEICHNIS

[1] D. Lüdtker, „01_Organisatorisches.pdf“, 2022.

[2] „www.dlr.de,“ [Online]. Available: https://www.dlr.de/sc/desktopdefault.aspx/tabid-11139/19481_read-45210/. [Zugriff am 20 November 2022].

[3] „ec.europa.eu,“ 20 September 2018. [Online]. Available: https://ec.europa.eu/commission/presscorner/detail/de/IP_18_5805. [Zugriff am 20 November 2022].

[4] „science.apa.at,“ 14 Januar 2021. [Online]. Available: <https://science.apa.at/power-search/12382935473194691007>. [Zugriff am 20 November 2022].

[5] N. Worldview, „nasa.gov,“ 10 September 2020. [Online]. Available: <https://www.nasa.gov/image-feature/goddard/2020/nasas-aqua-satellite-captures-devastating-wildfires-in-oregon>. [Zugriff am 21 November 2022].

- [6] R. K, „kramdas1996.medium.com,“ 25 Mai 2020. [Online]. Available: <https://kramdas1996.medium.com/image-segmentation-based-on-color-masking-907a7da56f96>. [Zugriff am 22 November 2022].
- [7] S. F. Würzburg, „Vegetationsbrände - Merkblatt für die Feuerwehren Bayerns, Version 4.1,“ Staatliche Feuerweherschule Würzburg, 2022.
- [8] Uncredited/dpa, „t-online.de,“ [Online]. Available: https://www.t-online.de/nachrichten/panorama/id_84761100/bilder/tote-und-schwere-schaeden-durch-waldbraende-in-kalifornien.html. [Zugriff am 22 November 2022].
- [9] ESA, „mdr.de,“ 10 Januar 2020. [Online]. Available: <https://www.mdr.de/wissen/umwelt/weltraumbilder-waldbrand-in-australien-100.html>. [Zugriff am 22 November 2022].