



Web-Tool for the supervised classification of earth observation data in OpenEO

Pflichtenheft: Copernicus Code Crafters : Jan Becker, Finn Geßner, Severin Krohne, Kjell Wundram, Emil Erlenkötter, Joaquin Valdez

1. Zielbestimmung

Es soll ein webbasiertes Machine-Learning Toolkit entwickelt werden, welches eine Benutzeroberfläche bietet. Diese Benutzeroberfläche beinhaltet eine Kartenapplikation, um area und date of interest auszuwählen. Die Benutzeroberfläche soll außerdem die Möglichkeit bieten neue Trainingsdaten hochzuladen. Weiterhin soll der Trainingsalgorithmus auswählbar sein. Das Backend soll openEOcubes implementieren, um die jeweiligen Berechnungen durchzuführen und den Algorithmus zu trainieren.

2. Produkteinsatz

Das Produkt soll einem mittelgroßen Unternehmen dienen, welches Services verkauft, die, mithilfe von Machine-Learning, Klassifikationen in einer Cloud erstellt. Zielgruppe sind Forscher, die schon Erfahrung in Fernerkundung besitzen und Sentinel 2 Daten mithilfe von maschinellem Lernen klassifizieren möchten. Bedingung dafür ist, dass openEO und openEOcubes genutzt wird.

User Stories

- a) Thomas arbeitet im Umweltamt einer Stadt und möchte Wasserflächen von Landflächen unterscheiden. Dafür wählt er das Stadtgebiet auf der Karte aus. Mit Trainingsdaten trainiert er den Algorithmus so, dass Flächen nach „Wasser“ und „nicht Wasser“ unterschieden werden. Um die Klassifizierungsergebnisse zu präsentieren, lädt er diese herunter.
- b) Eine Forschergruppe möchte ein vortrainiertes Machine Learning Modell nutzen. Dazu lädt es ein Modell mit Trainingsdaten herunter. Um das Modell vorher zu testen, sollen Beispieldaten genutzt werden.



3. Produkt-Umgebung

Das Produkt implementiert im Backend OpenEO Cubes. OpenEO Cubes ermöglichen den Zugriff auf Sentinel 2 Satellitendaten. Die Web-App nutzt die STAC API, um auf Trainingsdaten effizient zugreifen zu können. Leaflet beziehungsweise Leaflet-draw ermöglichen das Erstellen der Webkarte und das Auswählen der Area of Interest. Die Web-App soll in JavaScript programmiert werden. Der Server wird mit Node Express aufgesetzt. Im Fall von Zeitreihen bei Satellitendaten sollen Funktionen von gdal-cubes genutzt werden. Das Plugin „Georaster Layer for Leaflet“ wird neben anderen Plugins für Leaflet genutzt. Die Klassifikationsberechnungen sollen cloudbasiert mithilfe von AWS Elastic Compute Cloud berechnet werden.

4. Produktfunktionen

1. Dateiupload von GeoJSON und GeoPackage Trainingsgebieten mit Geopackage JS und eventuell Turf.js zur weiteren Verarbeitung.
2. Ergebnisdownload im GeoTIFF Format und Ergebnisvorschau als interaktive Karte im Web
3. Download des Trainingsmodells
4. Beispielhafte Satellitendaten sollen für Demonstrationen auf dem Server verfügbar gemacht werden.
5. Zielbereich soll durch den User mit dem “Leaflet-Draw” Plugin festlegbar sein
6. Genutzt wird der Service auf der AWS Elastic Compute Cloud, da dort die Copernicus Daten als Cloud-optimierte GeoTIFF Daten vorliegen und auf die mithilfe der STAC API zugegriffen werden kann
7. Das Klassifizierungsergebnis soll auf der Webkarte visualisiert werden. Dazu wird das Plugin “Georaster Layer for Leaflet” verwendet.
8. Bereitstellung des Programms als Docker Images auf Docker Hub und GitHub



5. Produkt-Daten

Die Trainingsdaten sollen gespeichert werden; eventuell in einer MongoDB Datenbank. Ebenso die Satellitendaten, die als Beispiel dienen.

6. Leistungsanforderungen

Eine Nutzerinteraktion soll eine Reaktion innerhalb einer Sekunde hervorrufen. Komplexe Seitenoperationen werden asynchron gelöst. Bei interaktiven Visualisierungen übersteigt die Reaktionszeit nicht 0,1 Sekunden. Die Beispielberechnung soll nicht über 20 Sekunden in Anspruch nehmen.

7. Benutzeroberfläche

Viewer

Die Seite "Viewer" kann zum einen über die Navigationsleiste geöffnet werden. Ebenfalls ist die Navigation durch einen Klick auf das Logo (oben rechts) möglich. Im Viewer kann der User sein Trainingsmodell auf Rasterdaten übertragen. Auf der rechten Seite des Displays werden die verschiedenen Auswahlmöglichkeiten angezeigt. Die linke Seite dient dabei zur Darstellung der gewählten Optionen. Durch den Schieberegler mit dem Wolkensymbol kann der Bedeckungsgrad des Satellitenbildes eingestellt werden. Nach der Eingabe des Messungszeitraum (Button "time period") bekommt der User ein erstes Feedback, ob Aufnahmen aus diesem Zeitraum vorhanden sind. Wenn mehrere Satellitenbilder vorhanden sind, bekommt der User eine Auswahlmöglichkeit. In einem nächsten Schritt kann mithilfe des Tools ("area of interest") entweder ein Untersuchungsraum direkt in die Karte eingezeichnet werden oder durch den Upload-Button ein eigenes Gebiet im GeoJSON oder GeoPackage-Format hochgeladen werden. Anschließend lässt sich das passende "training model" sowie die zugehörigen "hyperparameter" auswählen. Alle Änderungen werden zu jedem möglichen Zeitpunkt auf der linken Karte angezeigt, sodass der User Feedback zu seinen Eingaben erhält. Über den Downlaod-Button kann das klassifizierte Teilgebiet im geoTIFF-Format lokal gespeichert werden.

Upload

Über den Reiter "Upload" kann der User seine eigenen Trainingsdaten hochladen und einem Trainingsmodell zuweisen. Mit dem Button "open new training data" können eigene Trainingsgebiete mit passender Klassifikation hochgeladen werden und über das Preview-Fenster gesichtet werden. Ist der User mit seiner Auswahl zufrieden kann er seine Daten einem

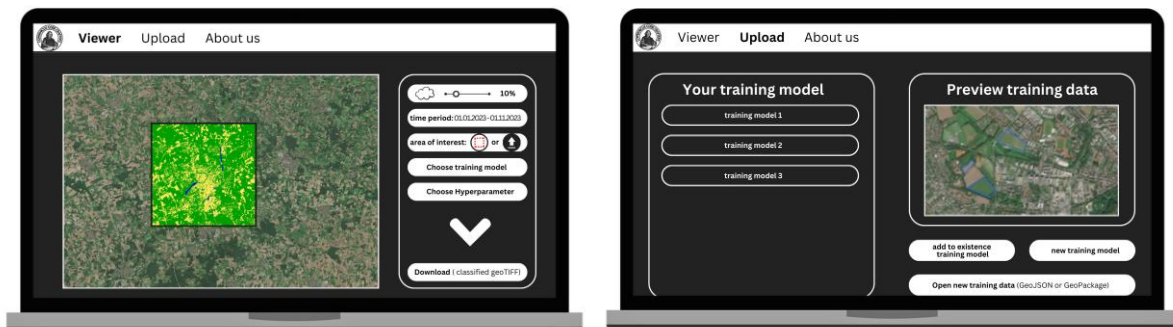


bestehenden Trainingsmodell zuweisen oder ein neues erstellen. Bei letzterem erscheint ein Fenster, in welchem die genaueren Parameter und der Modellname eingegeben werden können.

About us

Auf dieser Seite finden sich alle Informationen sowie Kontaktdaten zu den Erstellern der Webseite.

Die Webseite bietet optional ein Design mit hohen Kontrasten, welches die Navigation bei Farbenblindheit erleichtert.



8. Testfälle

Es werden mindestens 5 Trainingsdatensätze für das Testen der Anwendung erstellt.

Schedule

Planungsphase 1.11 – 15.11

7.11.2023 Pflichtenheft abgeben als PDF

Implementierungsphase 15.11.2023-24.01.2024

15.11.2023 Server aufstellen; erstes Design - Frontend mit Leaflet-Karte

22.11.2023 Algorithmus für Klassifikation aussuchen

29.11.2023 Status Report I abgeben, Frontend weiterentwickeln - Dateienupload

06.12.2023 Satellitendaten werden angezeigt

13.12.2023 Schnittstelle mit OpenEO fertigstellen, AWS-Verbindung herstellen

20.12.2023 Prerelease – Klassifikationsversuche - Ergebnis auf einer interaktiven Karte visualisieren

10.01.2024 Status report II – Download des Ergebnisses als GeoTIFF



- 17.01.2024 Testfälle erstellen und durchführen, Docker fertigstellen
- 24.01.2024 Letzte Bug fixes, finale Präsentation vorbereiten, Kommentare checken
- 30.01.2024** Finale Abgabe von Projektbericht, Quellcode auf GitHub, Docker Images und Dokumentation

Abschlusspräsentation 31.01.2024