

1. Vergleich der vorkommenden Satelliten

Die in dem Skript verwendeten Satellitendaten basieren auf den vier Landsat-Satelliten 4-8¹, den zwei Sentinel-2 Satelliten (Sentinel-2A & B) sowie ausgewählten MODIS-Datensätzen. Diese Satellitenfamilien wurden ausgewählt, um ein möglichst breites Spektrum an Anwendungsmöglichkeiten abzudecken. Hierbei sind die unterschiedlichen charakteristischen Auflösungen der jeweiligen Satelliten von Bedeutung. Die im Skript verwendeten Charakteristika sind in Tabelle 1 zusammengefasst: Zwar existieren seit 2015 Sentinel-2 Daten, jedoch erst ab 2017 als prozessierte Skript Daten (als *Surface Reflection* Daten, vgl. hierzu Kap. 2.1.1).

Tabelle 1: Übersicht der Satellitenauflösungen sowie der Datenverfügbarkeit.

	Landsat	Sentinel-2	MODIS
Datenverfügbarkeit seit	August 1982	März 2017	Februar 2000
Temporale Auflösung	16 Tage	5 Tage	1 Tag
Räumliche Auflösung	30 Meter	10, 20 Meter	500, 1.000 Meter
Spektrale Auflösung	6 Bänder	10 Bänder	7 Bänder

Die Auswahl eines Satelliten für eine wissenschaftliche Analyse ist ein fundamentaler Schritt, denn sie bestimmt maßgebend das Ergebnis. Je nach Fragestellung werden die unterschiedlichen Satelliten verwendet, dabei ist unter anderem sowohl die zeitliche als auch räumliche Ebene zu berücksichtigen. Beispielsweise sind bei kleinräumigen Analysen Landsat und Sentinel-2 Daten aufgrund ihrer hohen räumlichen Auflösung zu bevorzugen. Soll ein Ereignis analysiert werden, dass noch weiter zurückliegt, eignet sich hierfür die Landsat-Zeitreihe. Mit Sentinel-2 Daten lassen sich aktuellere Ereignisse (ab 2017) mit einer sehr hohen räumlichen und temporalen Auflösung analysieren. Hingegen eignet sich MODIS vor allem für die Detektion großräumiger Änderungen, welche durch tägliche Daten abgedeckt werden. Beispiele für die Anwendung der verschiedenen Satellitendaten werden mit Kapitel 4 thematisiert.

¹ Landsat 6 ist in dieser Reihe nicht vertreten, da der Satellit nicht das Orbit erreichte (YOUNG et al. 2017).

2. Vorstellung des Skripts

Das Skript kann in insgesamt sechs Abschnitte unterteilt werden, so besteht der erste Block aus der Parametereingabe sowie der Anzeige des Untersuchungsgebietes. Der zweite Teil umfasst alle im Skript vorkommenden Funktionen, diese können dabei nochmals verschiedenen Kategorien zugewiesen werden. Ein weiterer Abschnitt (dritter Block) beinhaltet die Visualisierungsparameter, welche für die Darstellung der Ergebnisse in GEE erforderlich sind. Die Verarbeitung der Daten ist im vierten Bereich vorzufinden und stellt den größten Teil des Skripts dar. Das Exportieren der Ergebnisse erfolgt im Anschluss und kann mittels der Parametereingabe aktiviert bzw. deaktiviert werden. Die Darstellung der Ergebnisse in GEE selbst schließt das Skript ab.

2.1. Parametereingabe

Die Parametereingabe ist der Block des Skripts, welcher primär vom Benutzer geändert werden sollte. Eine Tabelle mit allen verfügbaren Parametereinstellungen sowie deren Anwendung wird nachfolgend dargestellt:

Tabelle 2: Erklärung aller möglichen Parametereingaben im Skript.

Parameter [var X]	Beschreibung	Eingabe; Typ
satellite	Auswahl des gewünschten Satelliten und somit der jeweiligen dazugehörigen <i>ImageCollection</i>	'Landsat', 'Sentinel' oder 'Modis'; String
geometry	Bestimmung des Untersuchungsgebietes	Einzeichnen einer Fläche über die <i>Draw Toolbox</i> oder Eingabe der Eckkoordinaten einer Fläche im <i>Code Editor</i> mit folgender Reihenfolge: [xMin, yMin, xMax, yMax]; Vektordaten oder Liste der Koordinaten
month_start & month_end year_start & year_end Date_start & Date_end	Bestimmung des Untersuchungszeitraumes; Hierbei muss beachtet werden, dass die Monats- und Jahresfilterung vor der Filterung nach den exakten Datums Grenzen durchgeführt wird (vgl. Kap. 2.1.2)	Monats- und Jahresangabe als Integer -Typ und exakte Datumseingabe als String -Typ im Format 'YYYY-MM-DD'
max_cloud_cover	Der maximale Wolkenbedeckungsgrad der einzelnen Landsat- und Sentinel szenen	Zwischen 0 und 100 Integer/Float
index	Der zu berechnende Index; Die Liste der integrierten Indizes kann nach Belieben im Funktionen-Block erweitert werden	'NDVI', 'NDWI', 'NDSI', 'NBR' (alle Satelliten); 'LST' (Modis); String
metric	Die statistischen Merkmale, welche berechnet werden sollen	'mean', 'median', 'min', 'max', 'stdDev', 'count'; String
Differenced	Diese If-Schleife legt fest, ob ein differenzierter Index berechnet werden soll	Mit der Eingabe 'Yes' wird diese Schleife aktiviert. Bei jeder anderen Eingabe wird dieser Block nicht ausgeführt; String
month_start & month_end year_start & year_end Date_start & Date_end Postevent	Bestimmung des Untersuchungszeitraumes für das <i>Postevent</i> (vgl. Kap. 2.1.2)	Monats- und Jahresangabe als Integer -Typ und exakte Datumseingabe als String -Typ im Format 'YYYY-MM-DD'
export to drive	Diese If-Schleife legt fest, ob das Ergebnis nach Google Drive exportiert werden soll	Mit der Eingabe 'Yes' wird diese Schleife aktiviert. Bei jeder anderen Eingabe wird dieser Block nicht ausgeführt; String
epsg	<i>EPSG Code</i> des gewünschten Ausgabekoordinatensystem	'EPSG:XXXXX'; String
export singleimage	Diese If-Schleife legt fest, ob das Ergebnis einer einzelnen Szene nach Google Drive exportiert werden soll	Mit der Eingabe 'Yes' wird diese Schleife aktiviert. Bei jeder anderen Eingabe wird dieser Block nicht ausgeführt; String
imageID	Die ImageID der einzelnen Szene innerhalb der ausgewählten <i>ImageCollection</i>	X; Integer
region	Ein Text, welcher im Exportnamen des Ergebnisses steht	<i>Beispiel:</i> 'Deutschland'; String
time_period	Ein Text, welcher im Exportnamen des Ergebnisses steht	<i>Beispiel:</i> 'Juni-Juli 2018'; String

2.1.1. Image Collections

Mit der ersten Parametereingabe (**var satellite**) wird die gewünschte Satellitenfamilie sowie die dazugehörige *Image Collection* bestimmt. Diese *Image Collection* definiert sich über mehrere Satellitenbilder, welche alle durch denselben Sensor aufgenommen wurden. Dadurch ist es dem Benutzer möglich, schnell und gezielt nach individuellen Kriterien zu filtern und/oder sortieren, um so eine oder mehrere spezifischen Szenen zu erhalten. Hierbei können sowohl räumliche als auch zeitliche Filterungen durchgeführt werden, aber auch die Filterung nach Bewölkungsgrad der jeweiligen, einzelnen Szenen sowie weitere benutzerspezifische Filterungen sind möglich.²

Insgesamt sind mehrere *Image Collections* eines Satelliten vorhanden, so stehen bei der verwendeten Landsat-Reihe drei *Collections* zur Verfügung. Diese wurden jeweils unterschiedlich prozessiert und liegen als eigener Datensatz vor; diese sind: *Surface Reflection*, *Top of Atmosphere* und *Raw Images*.³ Bei Sentinel-2 hingegen existieren nur zwei *Collections*, nämlich *Surface Reflection* und *Top of Atmosphere*. Ein grundlegender Unterschied hierbei liegt nicht nur in der Art der Prozessierung, sondern auch in der Verfügbarkeit der Daten: Während die *Top of Atmosphere*-prozessierten Daten bereits seit Juni 2015 vorliegen (und somit seit Beginn der Sentinel-2 Mission), sind die *Surface Reflection*-prozessierten Daten erst ab März 2017 verfügbar.⁴

Im Skript wurden *Surface Reflection* Daten verwendet, jedoch können auch anders prozessierte Daten herangezogen werden. Hierfür muss lediglich das sog. *Earth Engine Snippet* im Skript geändert werden (Beispiel für das Landsat 8 *Surface Reflection Snippet*: `ee.ImageCollection("LANDSAT/LC08/C01/T1_SR")`; vgl. hierzu auch Abb. 7 in Kap. 2.4). Diese finden sich auf dem Datenkatalog der Plattform unter folgendem Link: <https://developers.google.com/earth-engine/datasets>.

2.1.2. Temporale Filterung

Die temporale Filterung ist ein essentieller Bestandteil des Skripts, denn sie bestimmt den gewünschten Untersuchungszeitraum. Dabei sind im Skript zwei unterschiedliche Arten der temporalen Filterungen möglich: Eine Filterung nach Monaten, bei der nur ausgewählte Monate in einer spezifischen Zeitspanne von mindestens einem Jahr vertreten sind (vgl. Zeilen 10–14 in Abb. 1) und eine lückenlose Filterung, welche sich über eine Zeitspanne zwischen einem Start- und Endpunkt definiert (vgl. Zeilen 16–17 in Abb. 2). Die Monats-Filterung ist beispielsweise bei Analysen in arktischen Regionen mit einer kurzen Vegetationsperiode sinnvoll, da hier gezielt irrelevante Daten bzw. Monate ausgeblendet werden können. Ein weiterer Grund für eine Filterung nach Monaten kann auch ein für beispielsweise

² GORELICK et al. 2017.

³ GOOGLE EARTH ENGINE; GORELICK et al. 2017; KUMAR & MUTANGA 2018.

⁴ GOOGLE EARTH ENGINE.

Wendekreisregionen jahreszeitlich bedingter hoher Wolkenbedeckungsgrad sein, der eine weitere Analyse erschwert.

Für die Durchführung einer Monatsfilterung im Skript ist es notwendig, dass die Variablen *month_start* und *month_end* sowie *year_start* und *year_end* zeitlich gesehen innerhalb der Variablen *Date_start* sowie *Date_end* liegen (vgl. Abb. 1). Der Grund hierfür liegt in der Reihenfolge der Filterungen, denn im Skript wird im späteren Verlauf zuerst die Filterung nach Jahr und Monat durchgeführt und anschließend die Filterung nach den exakten Datumsangaben (vgl. hierzu auch Abb. 7 in Kapitel 2.4). Analog dazu erfolgt die lückenlose Filterung der exakten Datumsgrenzen, hierbei muss der Untersuchungszeitraum innerhalb der Monats- und Jahresvariablen liegen.

In dem in Abbildung 1 dargestellten Beispiel werden die Parametereinstellungen für eine Monats-Filterung aufgezeigt, hierbei erstreckt sich die Untersuchungsperiode über die Monate Juni und Juli in den Jahren von 2003 bis 2006.

```
10 var month_start = 6;           //Startmonth of the temporal integration
11 var month_end = 7;           //Endmonth of the temporal integration
12
13 var year_start = 2003;        //Startyear of the temporal integration.
14 var year_end = 2006;         //Endyear of the temporal integration.
15
16 var Date_start = '2003-06-01'; // 'YYYY-MM-DD'; Care about the starting year of the satellite
17 var Date_end = '2006-07-31';  // 'YYYY-MM-DD'
```

Abbildung 1: Parametereingaben für eine Monats-Filterung.

Mit Abbildung 2 wird beispielhaft die Filterung nach exakten Datumsgrenzen dargestellt, hier verläuft die Untersuchungsperiode lückenlos vom 01.06.2003 bis zum 31.07.2006. Das Zeitintervall der Variablen *Date_start* und *Date_end* liegt somit innerhalb der Variablen in den Zeilen 10–14.

```
10 var month_start = 1;           //Startmonth of the temporal integration
11 var month_end = 12;           //Endmonth of the temporal integration
12
13 var year_start = 2003;        //Startyear of the temporal integration.
14 var year_end = 2006;         //Endyear of the temporal integration.
15
16 var Date_start = '2003-06-01'; // 'YYYY-MM-DD'; Care about the starting year of the satellite
17 var Date_end = '2006-07-31';  // 'YYYY-MM-DD'
```

Abbildung 2: Parametereingaben für eine lückenlose Filterung zwischen zwei Daten.

In diesem Fall wurde mittels der Monats- und Jahresfilterung zunächst ein größeres Zeitfenster gewählt als benötigt. Mit der nächsten Filterung, also nach den exakten Datumsgrenzen, wird dieses Zeitfenster jedoch eingegrenzt und entspricht letztendlich der gewünschten Untersuchungsperiode.

Zusammengefasst sind für die temporale Filterung folgende zwei Punkte von Bedeutung:

- Die Monats- und Jahresfilterung wird vor der Filterung der exakten Datumsgrenzen durchgeführt.

- Falls eine Monatsfilterung durchgeführt werden soll, muss die Filterung nach einer lückenlosen Zeitspanne ein größeres Zeitfenster umfassen als die der Monatsfilterung. Dies ist für die Filterung nach einer ununterbrochenen Zeitspanne analog zu beachten.

2.2. Funktionen

Die verwendeten Funktionen im Skript können in insgesamt vier Kategorien eingeteilt werden, diese umfassen die Umbenennung der Bänder der jeweiligen Satelliten, die Wolkenmaskierung, die Berechnung der gewünschten Indizes sowie die Auswahl des *Reducers* für die jeweiligen statistischen Merkmale.

2.2.1. Umbenennung der Bänder

Das Umbenennen der jeweiligen Bänder wurde in diesem Skript durchgeführt, um ein einheitliches Vorgehen bezüglich der Berechnung der Indizes zu ermöglichen. Der Grund hierfür liegt in den unterschiedlichen Bezeichnungen der Bänder der jeweiligen Satelliten; dabei sind sogar Unterschiede zwischen einzelnen Satellitenfamilien zu verzeichnen (bei der Landsat-Reihe zwischen Landsat 4-7 und Landsat 8, vgl. Tab. 3).

Tabelle 3: Funktionen zur Umbenennung der einzelnen Bänder.

Funktion	Beschreibung
renameBandsTM_ETM	Umbenennung der Bänder 'B1', 'B2', 'B3', 'B4', 'B5', 'B7' zu 'B', 'G', 'R', 'NIR', 'SWIR1', 'SWIR2' bei Landsat 4-7
renameBandsOLI	Umbenennung der Bänder 'B2', 'B3', 'B4', 'B5', 'B6', 'B7' zu 'B', 'G', 'R', 'NIR', 'SWIR1', 'SWIR2' bei Landsat 8
renameBandsSentinel	Umbenennung der Bänder 'B2', 'B3', 'B4', 'B8', 'B11', 'B12' zu 'B', 'G', 'R', 'NIR', 'SWIR1', 'SWIR2' bei Sentinel-2
renameBandsModis	Umbenennung der Bänder 'Nadir_Reflectance_Band1', 'Nadir_Reflectance_Band2', 'Nadir_Reflectance_Band3', 'Nadir_Reflectance_Band4', 'Nadir_Reflectance_Band5', 'Nadir_Reflectance_Band6', 'Nadir_Reflectance_Band6' zu 'R', 'NIR1', 'B', 'G', 'NIR2', 'SWIR1', 'SWIR2' bei MODIS (Nadir BRDF-Adjusted Reflectance)
renameBandLST	Umbenennung des Bandes 'LST_Day_1km' zu 'LST' bei MODIS

Eine im Skript vorkommende Funktion zur Umbenennung der einzelnen Bänder wird mit Abbildung 3 am Beispiel von Landsat 8 aufgezeigt. Informationen über die einzelnen Bänder (Bezeichnung, Auflösung, Wellenlängenbereich etc.) bietet GEE in der Beschreibung der jeweiligen *Image Collection* auf der GEE-Plattform.⁵

```

99 // Define functions to select and rename bands for Landsat-8
100 function renameBandsOLI(image) {
101     var bands = ['B2', 'B3', 'B4', 'B5', 'B6', 'B7', 'pixel_qa'];
102     var new_bands = ['B', 'G', 'R', 'NIR', 'SWIR1', 'SWIR2', 'pixel_qa'];
103     return image.select(bands).rename(new_bands);
104 }

```

Abbildung 3: Funktion zur Umbenennung der notwendigen Bänder bei Landsat 8.

⁵ <https://developers.google.com/earth-engine/datasets>

Das **pixel_qa**-Band bei Landsat (vgl. Abb. 3) dient der Verarbeitung der Daten und wird weder in Tabelle 3 aufgeführt noch im Skript umbenannt, nichtsdestoweniger ist es in der Funktion inbegriffen. Dies hat den Hintergrund, dass das **pixel_qa**-Band für die Wolkenmaskierung im späteren Verlauf erforderlich ist (vgl. Kap. 2.2.2).

2.2.2. Wolkenmaskierung

Wolken sowie deren Schattenwurf stellen ein gängiges Problem der optischen Fernerkundung dar, da sie die Erdoberfläche verdecken.⁶ Gerade bei Analysen mit einer hohen Anzahl an Satellitenbildern ist eine Wolkenmaskierung essentiell, da sonst Wolken bei der Berechnung von statistischen Merkmalen, wie beispielsweise beim Erstellen eines Median-NDVI Satellitenbildes, das Ergebnis beeinflussen bzw. verzerren würden. Diese Wolkenmaskierung wird deshalb für die Landsat-Reihe und für Sentinel-2 unter Zuhilfenahme von Funktionen durchgeführt. Für die MODIS-Datensätze wird keine Wolkenmaskierung mittels einer Funktion genutzt, da die Wolkenmaskierung der Daten bereits im *Preprocessing* vorgenommen wurde (vgl. Kap. 2.1.1).

Mit Abbildung 4 wird die Funktion zur Wolkenmaskierung für die Landsat-Reihe 4-8 aufgezeigt, diese basiert auf dem **pixel_qa** Band (räumliche Auflösung: 30 m). In diesem Band sind pixelbasierte Informationen unter anderem über das Vorkommen von Wolken und deren Schattenfall sowie Schnee, Eis und Wasser enthalten⁷. Diese Werte wurden mittels eines *CFMask* Algorithmus berechnet, dieser ist für die Detektion von Wolken am besten geeignet.⁸

Somit können ausgewählte Pixel, wie Wolken- und Wolkenschattenpixel, gezielt maskiert werden (vgl. Abb. 4, rechts). Dafür wird die *Bitmask* bzw. werden die jeweiligen *Bit*-Nummern benötigt, welche sich in den *Image Collection* Informationen des *Data Catalog* finden (vgl. Abb. 4, links). Mit folgender Abbildung wird die Funktion zur Wolkenmaskierung verdeutlicht.

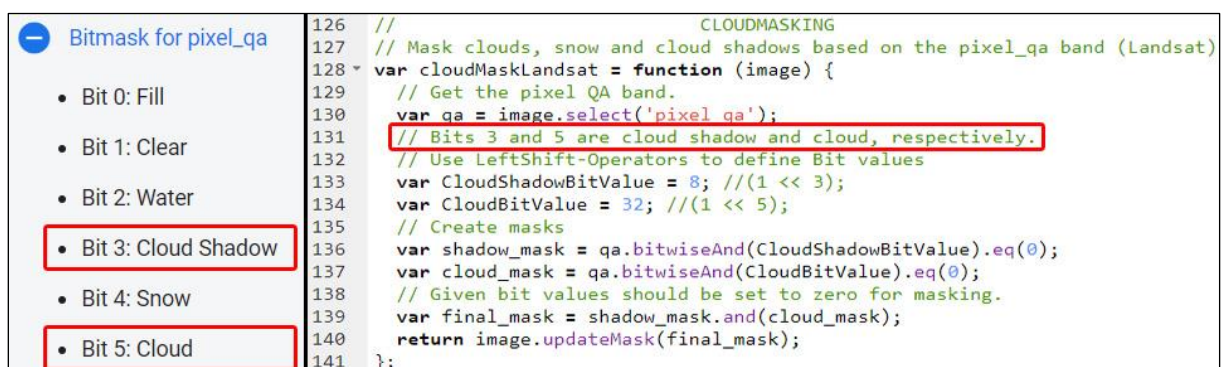


Abbildung 4: Die Funktion zur Wolkenmaskierung für Landsat 4-8 (rechts) sowie die dazugehörige Bitmask (links); Quelle: verändert nach GOOGLE EARTH ENGINE.

⁶ ZHU & HELMER 2018; SCHMITT et al. 2019.

⁷ UNITED STATES GEOLOGICAL SURVEY.

⁸ FOGA et al. 2017.

Diese Funktion maskiert somit Wolken sowie deren Schattenfall aus den einzelnen Landsat-Szenen, sodass wolkenfreie Szenen analysiert werden können.

Dies wurde ebenfalls bei Sentinel-2 durchgeführt, jedoch wurde hierbei das **QA60** Band verwendet, welches mit einer räumlichen Auflösung von 60 m gekennzeichnet ist. Dabei liefert diese Wolkenmaskierung teilweise unzureichende Ergebnisse, vor allem unter kritischen atmosphärischen Bedingungen.⁹ Dies hat verschiedene Gründe, die in dieser Arbeit nicht weiter thematisiert werden. Eine verbesserte Wolkenmaskierung bei Sentinel-2 wird beispielsweise in der Arbeit von NGUYEN et al. (2020) erläutert, die Autoren verwendeten hierfür einen *random forest classifier* sowie den Normalized Difference Snow Index (NDSI), um Schneeflächen von Wolken unterscheiden zu können. Sie stellten heraus, dass mit ihrem Ansatz eine deutlich umfangreichere Wolkenmaskierung erreicht wurde.

Das selbsterstellte Skript basiert lediglich auf der Wolkenmaskierung mittels dem **QA60** Band, eine ausführlichere Maskierung der Wolken bzw. die Implementierung der Funktion von NGUYEN et al. wäre an dieser Stelle möglich, wurde jedoch (noch) nicht ergänzt.

2.2.3. Berechnung der Indizes

Insgesamt sind im Skript fünf verschiedene Indizes vertreten, diese sind in Tabelle 4, zusammen mit der jeweiligen Berechnung sowie Beschreibung, dargestellt. Weiterhin sind die Satelliten mitaufgeführt, mit welchen die jeweiligen Indizes berechnet werden können. Zuletzt wird die Tabelle durch einen ausgewählten beispielhaften Fachzeitschriftenartikel, der die jeweiligen Indizes in GEE bereits verwendet hat, ergänzt:

Tabelle 4: Die im Skript vorkommenden Indizes.

Index	Berechnung	Beschreibung	Satelliten	Beispiel
Normalized difference vegetation index (NDVI)	$\frac{NIR - RED}{NIR + RED}$	Maßeinheit für die Vitalität von Vegetation	Landsat, Sentinel-2, Modis	XIONG et al. 2017
Normalized difference water index (NDWI)	$\frac{NIR - SWIR1}{NIR + SWIR1}^{10}$	Maßeinheit für den Wassergehalt von Vegetation, Detektion von Wasserflächen	Landsat, Sentinel-2, Modis	GUO et al. 2020
Normalized difference snow index (NDSI)	$\frac{Green - SWIR1}{Green + SWIR1}$	Detektion von Schnee- und Eisflächen	Landsat, Sentinel-2, Modis	GU et al. 2021
Normalized Burn Ratio (NBR)	$\frac{NIR - SWIR2}{NIR + SWIR2}$	Detektion von Brandflächen	Landsat, Sentinel-2	PARKS et al. 2018
Land surface temperature (LST)	Berechnung durch GEE	Analyse der Tagesoberflächentemperatur	Modis	EBRAHIMY et al. 2021

⁹ NGUYEN et al. 2020.

¹⁰ Für Landsat und Sentinel nach GAO 1996; bei Modis wurde der NDWI durch folgende Formel berechnet:
$$\frac{Green - NIR}{Green + NIR}$$

Die Berechnung der Indizes erfolgt auf Basis von Funktionen; dies wird beispielhaft mit Abbildung 5 verdeutlicht: Hier wird zunächst in Zeile 159 mittels einer *Expression* die Formel des jeweiligen Indizes, in diesem Fall der NDVI, im Dateiformat *String* festgelegt. Anschließend müssen die darin beschriebenen Bänder den entsprechenden Bändern zugeordnet werden, dies erfolgt mittels des *.select()* Befehls (vgl. Zeilen 161–162 in Abb. 5). Durch diese Berechnung wird ein neues Band erstellt, welches zur *Image Collection* hinzugefügt werden muss (*.addBands()* Befehl; vgl. Zeile 165). Zuletzt wird dem Band über den *.rename()* Befehl eine neue Bezeichnung gegeben (vgl. Zeile 166) sowie der neu erstellte Index in das Dateiformat einer Fließkommazahl über den *.float()* Befehl geändert (vgl. Zeile 166). Dies hat den Hintergrund, dass der Index einen Wertebereich von -1 über 0 bis +1 besitzt.

```

155 // INDICES
156 // Create function for NDVI
157 function my_ndvi(image) {
158   var ndvi_var = image.expression(
159     "(NIR - RED)/(NIR + RED)",
160     {
161       NIR: image.select("NIR"),
162       RED: image.select("R"),
163     });
164   return image
165     .addBands(ndvi_var)
166     .rename('NDVI')
167     .float();
168 }

```

Abbildung 5: Funktion zur Berechnung des Normalized Difference Vegetation Index.

2.2.4. Statistische Merkmale

Ein Ziel des Skripts ist die Analyse von *Image Collections*, welche sich (meist) über eine große Anzahl an einzelnen Szenen definieren. Um diese Analyse durchzuführen, ist die Berechnung von statistischen Merkmalen sinnvoll. Hierfür werden beispielsweise mehrere Szenen, welche sich räumlich überlagern, aber zu unterschiedlichen Zeitpunkten aufgenommen wurden, auf eine einzelne Szene reduziert. Dabei gibt es verschiedene sog. *Reducer*, welche in Tabelle 5 zusammengefasst sind.

Tabelle 5: Statistische Merkmale.

Parametereingabe [var X]	Reducer in GEE	Beschreibung
median	ee.Reducer.median();	Berechnung des Medians
mean	ee.Reducer.mean();	Berechnung des (gewichteten) arithmetischen Mittel
min	ee.Reducer.min();	Berechnung der Minimalwerte
max	ee.Reducer.max();	Berechnung der Minimalwerte
stdDev	ee.Reducer.stdDev();	Berechnung der Standardabweichung
count	ee.Reducer.count();	Berechnung der <i>Number of Observations</i>

2.3. Visualisierungsparameter

Die Visualisierungsparameter dienen der Darstellung der Ergebnisse im Kartenbereich des Skripts. Dabei können sowohl ausgewählte Bänder, deren Wertebereich als auch die jeweilige

Farbpalette und weitere Einstellungen wie die Transparenz ausgewählt und dargestellt werden. Die Farbskalen können hierbei unter anderem per Hexagonal Code¹¹ eingefügt werden. Mit Abbildung 6 werden exemplarische Visualisierungsparameter aufgezeigt, dabei wurden in den Zeilen 213–215 die Parameter für die Darstellung des NDVI festgelegt. Zwar besitzt dieser Index einen Wertebereich zwischen -1 und +1, jedoch werden negative Werte mit diesen Einstellungen nicht berücksichtigt. Die Farbpalette wurde hierbei von GEE übernommen. Weitere Darstellungen stammen ebenfalls von Seiten der Earth Engine und sind beispielhaft in Abbildung 6 vertreten.

```

212 //=====PARAMS=====
213 var ndviparams = {min: 0.0, max: 1.0, palette: [
214   'FFFFFF', 'CE7E45', 'DF923D', 'F1B555', 'FCD163', '99B718', '74A901', '66A000', '529400',
215   '3E8601', '207401', '056201', '004C00', '023B01', '012E01', '011D01', '011301']];
216 var lstParams = {min: 13000.0, max: 16500.0, palette: ['040274', '040281', '0502a3', '0502b8',
217   '0502ce', '0502e6', '0602ff', '235cb1', '307ef3', '269db1', '30c8e2', '32d3ef', '3be285', '3ff38f',
218   '86e26f', '3ae237', 'b5e22e', 'd6e21f', 'fff705', 'ffd611', 'ffb613', 'ff8b13', 'ff6e08', 'ff500d',
219   'ff0000', 'de0101', 'c21301', 'a71001', '911003']];
220 var Landsat_median_RGB_Params = {bands: ['R', 'G', 'B'], min: 0, max: 4000, gamma: 1.5};

```

Abbildung 6: ausgewählte Visualisierungsparameter.

2.4. Verarbeitung der Daten

Die Verarbeitung der Daten stellt den größten Block im Skript dar und umfasst unter anderem die Filterungsprozesse, die Anwendung der Funktionen (Umbenennung der Bänder, Wolkenmaskierungen, Hinzufügen der Indizes) sowie die Berechnung der differenzierten Indizes. Mit Abbildung 7 werden am Beispiel des Landsat 8 Satelliten diese Vorgänge verdeutlicht, hierbei wird zunächst die *Image Collection* mithilfe des *Earth Engine Snippet* (**Tier 1, Surface Reflectance; T1_SR**) ausgewählt (vgl. Zeile 289 in Abb. 7). Damit wird das gesamte Landsat 8 Archiv geladen. Um nun die spezifischen, gewünschten Szenen zu erhalten, werden die Filterungsprozesse durchgeführt: Zunächst erfolgt die räumliche Filterung mit dem Befehl *.filterBounds(geometry)* (vgl. Zeile 290). Dadurch werden die Szenen, die das Untersuchungsgebiet nicht abdecken, aussortiert. Als nächstes sind alle Szenen in der *Image Collection* beizubehalten, die den maximalen, festgelegten Wolkenbedeckungsgrad nicht überschreiten (vgl. Zeile 291). Mit dem *.map()* Befehl in den Zeilen 292 und 293 werden die ausgewählten Funktionen (Umbenennung der Bänder und Wolkenmaskierung) für jede einzelne Szene in der *Image Collection* angewendet. Zuletzt erfolgt die temporale Filterung, hierbei wird zuerst die Filterung nach Jahr und Monat (vgl. Zeilen 294–295) durchgeführt und anschließend die Filterung nach den exakten Datumsgrenzen (vgl. Zeile 296 und auch Kap. 2.1.2). Zuletzt werden die einzelnen Szenen noch auf das Untersuchungsgebiet zugeschnitten (vgl. Zeile 297).

¹¹ Unter diesem URL-Link können dabei die Codes bestimmt werden: <https://htmlcolorcodes.com/>.

```

288 // Load an Landsat-8 ImageCollection.
289 var L8_col = ee.ImageCollection('LANDSAT/LC08/C01/T1_SR')Earth Engine Snippet
290 .filterBounds(geometry)
291 .filter(ee.Filter.lt('CLOUD_COVER_LAND', max_cloud_cover))
292 .map(renameBandsOLI)
293 .map(cloudMaskLandsat)
294 .filter(ee.Filter.calendarRange(year_start, year_end, 'year'))
295 .filter(ee.Filter.calendarRange(month_start, month_end, 'month'))
296 .filterDate(Date_start, Date_end)
297 .map(function(image){return image.clip(geometry)});
298
299 // Merge all collections
300 var Merge_col_Landsat = L4_col.merge(L5_col).merge(L7_col).merge(L8_col);
301 // Sort by date
302 var Merge_col_Landsat = Merge_col_Landsat.sort("system:time_start");
303 print('All Images', Merge_col_Landsat);
304 // Add indices to ImageCollection
305 var temp = Merge_col_Landsat.map(my_ndvi)
306                               .map(my_ndwi)
307                               .map(my_nbr)
308                               .map(my_ndsi);
309 print('All Images (with all indices)', temp);
310 //=====

```

Abbildung 7: Verarbeitung der Daten am Beispiel des Landsat 8 Satelliten.

Da nun insgesamt vier verschiedene prozessierte *Image Collections* der Landsat-Reihe vorliegen (Landsat 4-8), werden diese mit dem `.merge()` Befehl zu einer einzigen *Image Collection* zusammengeführt (vgl. Zeile 300). Diese wird schließlich noch zeitlich sortiert (vgl. Zeile 302), da temporale Überlagerungen der verschiedenen Landsat-Satelliten existieren¹². Über den `print()` Befehl werden die prozessierte *Image Collection* bzw. die einzelnen, darin enthaltenen Szenen mit deren Metadaten und den verfügbaren Bändern in der *Console* in GEE ausgegeben. Zuletzt werden die Indizes der *Image Collection* hinzugefügt (vgl. Zeilen 305–308).

```

415 //Reduce the two collections with the selected metric
416 var temp_PREEVENT = temp.reduce(metric);
417 var temp_POSTEVENT_FOR_DIFFERENCED = temp_POSTEVENT.reduce(metric);
418 var temp_differenced = temp_PREEVENT.subtract(temp_POSTEVENT_FOR_DIFFERENCED);
419 print('Difference Image:', temp_differenced);
420 var temp_differenced = temp_differenced.rename(['B', 'G', 'R', 'NIR', 'SWIR1', 'SWIR2',
421 'NDVI', 'NDWI', 'NBR', 'NDSI'])
422 }

```

Abbildung 8: Berechnung der differenzierten Indizes.

Auch die Berechnung von differenzierten Indizes sowie zusätzlich auch die Bänder werden mit dem Skript ermöglicht, dafür werden zunächst die in Abbildung 7 vorgenommenen Schritte mit anderen zeitlichen Variablen nochmals durchgeführt. Diese *Image Collection* wird als **Postevent** bezeichnet und beinhaltet dieselben Parametereingaben wie die **Preevent Image Collection** (Die zuerst berechnete *Image Collection*), jedoch mit Ausnahme der Variablen *month_start* und *month_end* sowie *year_start* und *year_end* bzw. *Date_start* und *Date_end*.

¹² YOUNG et al. 2017.

Bei der **Postevent Image Collection** sind diese Variablen mit dem Suffix *Postevent* gekennzeichnet (vgl. hierzu auch Tab. 1 in Kap. 2.1).

Beide *Image Collections* werden für die Berechnung der differenzierten Indizes zunächst mit dem ausgewählten statistischen Merkmal auf eine Szene reduziert sowie anschließend im Sinne einer *Change Detection* die Änderungen der zwei Zeiträume herausgestellt, dies erfolgt durch die Subtraktion dieser unter Verwendung des `.subtract()` Befehl.

2.5. Exportieren der Ergebnisse

Das Exportieren der Ergebnisse ist ein essentieller Bestandteil des Skripts, denn dadurch lassen sich die Daten in geographischen Informationssystemen weiterverarbeiten. Für das Herunterladen der Ergebnisse wird ein Google Drive Account benötigt, denn dort wird nach Ausführen des Skripts ein Ordner mit dem Namen „GEE_Export“ erstellt, in welchem alle Ergebnisse gespeichert werden. Generell werden die Ergebnisse jedoch nur dann heruntergeladen, wenn dies in der Parametereingabe aktiviert wurde (**var export_to_drive = 'Yes'**). Die Daten werden im Dateiformat *GeoTIFF* ausgespielt, alle exportierten Bänder müssen dabei in demselben Dateityp, nämlich als *Float*, vorliegen. Zu beachten ist, dass die einzelnen Bänder noch als unskalierte Daten vorliegen, die jeweiligen Skalierungsfaktoren sind dem *Data Catalog* zu entnehmen.

2.5.1. Export Fall A: Standardfall

Der Standardfall tritt ein, wenn der differenzierte Index nicht berechnet sowie keine einzelne Szene aus der Image Collection selektiert wurde. Beim Standardfall wird nicht nur der Export des ausgewählten Index mit dem jeweiligen statistischen Merkmal, sondern auch der Export der dazugehörigen Median-RGB Bildern ermöglicht. In Abbildung 9 wird das Exportieren dargestellt, dies erfolgt mittels des `Export.image.toDrive()` Befehls. Zuerst wird hierfür die jeweilige *Image Collection* (im Skript mit der Variable **temp** gekennzeichnet) ausgewählt und mit dem `.reduce(metric)` Befehl auf das angegebene statistische Merkmal reduziert (vgl. Zeile 554 in Abb. 9). Die Namen der Dateien setzen sich aus verschiedenen Faktoren wie dem Satelliten, dem Index sowie vom Benutzer gewählten Bezeichnungen zusammen (vgl. Zeilen 555–556). Die räumliche Auflösung (im Skript als *scale* bezeichnet) wird bereits im vorherigen Verlauf festgelegt und ist abhängig von der Eingabe des Satelliten.

Das Ausgabekoordinatensystem (*crs*; vgl. Zeile 559) wird über den EPSG-Code, welcher in der Parametereingabe vom Benutzer festgelegt wurde, bestimmt. Zuletzt wird das Untersuchungsgebiet ausgewählt (vgl. Zeile 560).

```

550 //                                EXPORTING DATA
551 //Exporting the selected index with the selected metric
552 if (export_to_drive == 'Yes' && Differenced != 'Yes') {
553   Export.image.toDrive({
554     image: temp.reduce(metric),
555     description: pl + '_All_bands_and_indices_' + metric + '_' + Date_start + '_' + Date_end + '_'
556     + region + '_' + time_period,
557     folder: 'GEE_Export',
558     scale: scale,
559     crs: epsg,
560     region: rect_geodesic
561   })
562
563   //RGB Median Image for Landsat & Sentinel
564   if (satellite == 'Sentinel') {
565     //Sentinel-2 RGB Bands = 10m Spatial Resolution
566     var scale = 10;
567   }
568   var rgb_image = temp.median().select(['R', 'G', 'B']);
569   Export.image.toDrive({
570     image: rgb_image,
571     description: pl + '_' + 'RGB_median_' + Date_start + '_' + Date_end +
572     '_' + region + '_' + time_period,
573     folder: 'GEE_Export',
574     scale: scale,
575     crs: epsg,
576     region: rect_geodesic
577   })

```

Abbildung 9: Exportieren der Ergebnisse.

Das Exportieren der RGB-Median Bilder wird mit den Zeilen 564–576 beschrieben, hier gibt es zwei Unterschiede zu den oben gezeigten Vorgängen: Zunächst wird bei der Verwendung von Sentinel-2 Daten die räumliche Auflösung auf 10 Meter gesetzt, da die Bänder Rot, Grün und Blau mit der entsprechenden Auflösung gekennzeichnet sind.¹³ Weiterhin werden statt des ausgewählten statistischen Merkmals die Bänder R, G und B selektiert sowie die Mediane der Bänder berechnet (vgl. Zeilen 568 & 570).

2.5.2. Export Fall B: Differenzierter Index

Das Exportieren von differenzierten Indizes stellt den kleinsten Block im Export-Bereich dar und wird mit der Variable **var Differenced** in der Parametereingabe gesteuert. Da bereits bei der Berechnung der differenzierten Indizes die zwei *Image Collections* mittels des ausgewählten statistischen Merkmals auf eine Szene reduziert wurden (vgl. Kap. 2.4), ist die Anwendung eines *Reducers* beim Exportieren nicht mehr notwendig. Mit Abbildung 10 wird der Export der differenzierten Indizes veranschaulicht. Es wäre an dieser Stelle möglich, die jeweiligen reduzierten **Pre-** und **Postevent** Daten zu exportieren; dies wurde jedoch (noch) nicht in das Skript integriert.

¹³ EUROPEAN SPACE AGENCY (ESA).

```

579 //                                EXPORTING DIFFERENCED INDEX
580 if (export_to_drive == 'Yes' && Differenced == 'Yes') {
581   Export.image.toDrive({
582     image: temp_differenced,
583     description: pl + '_differenced_' + index + '_' + metric + '_' + Date_start + '_' + Date_end + '-'
584     + Date_start_Postevent + '_' + Date_end_Postevent + '_' + region + '_' + time_period,
585     folder: 'GEE_Export',
586     scale: scale,
587     crs: epsg,
588     region: rect_geodesic
589   })}

```

Abbildung 10: Exportieren von differenzierten Indizes.

2.5.3. Export Fall C: Einzelne Szenen

Das Skript ermöglicht nicht nur den Export von reduzierten *Image Collections*, sondern auch den Export von einzelnen Szenen, welche in der jeweiligen *Image Collection* vorhanden sind. Hierbei erfolgt die Steuerung durch die zwei Variablen **var export_singleimage** und **var imageID**, wobei die erstgenannte lediglich dazu dient, dass die *if*-Schleife im Exportbereich aktiviert wird. Die Eingabe der zweitgenannten muss im Datentyp *Integer* (ganzzahlige Werte) erfolgen und entspricht der Position der gewünschten Szene in der *Image Collection*. Diese Position wird durch Abbildung 11 erklärt: Eine *Image Collection* mit allen darin enthaltenen Szenen wird bei Ausführung des Skripts in der *Console* durch den *print()* Befehl ausgegeben. Dabei sind sowohl die einzelnen Bänder als auch die Metadaten der Szenen mitaufgeführt. Die Metadaten haben vor allem eine große Bedeutung, da sie Aufschluss über Aufnahmedatum, Bewölkungsgrad, geographischer Lage (vgl. Abb. 11, rote Kästen) und weiteren Charakteristika der Szenen geben. Falls zum Beispiel die Szene vom 20.06.2018 für den Benutzer von Bedeutung ist, so kann diese über die Eingabe der *Image ID* ebenfalls heruntergeladen werden. Dafür muss die Zahl der Position (im Beispiel die Zahl 2) in der Parametereingabe eingefügt werden.

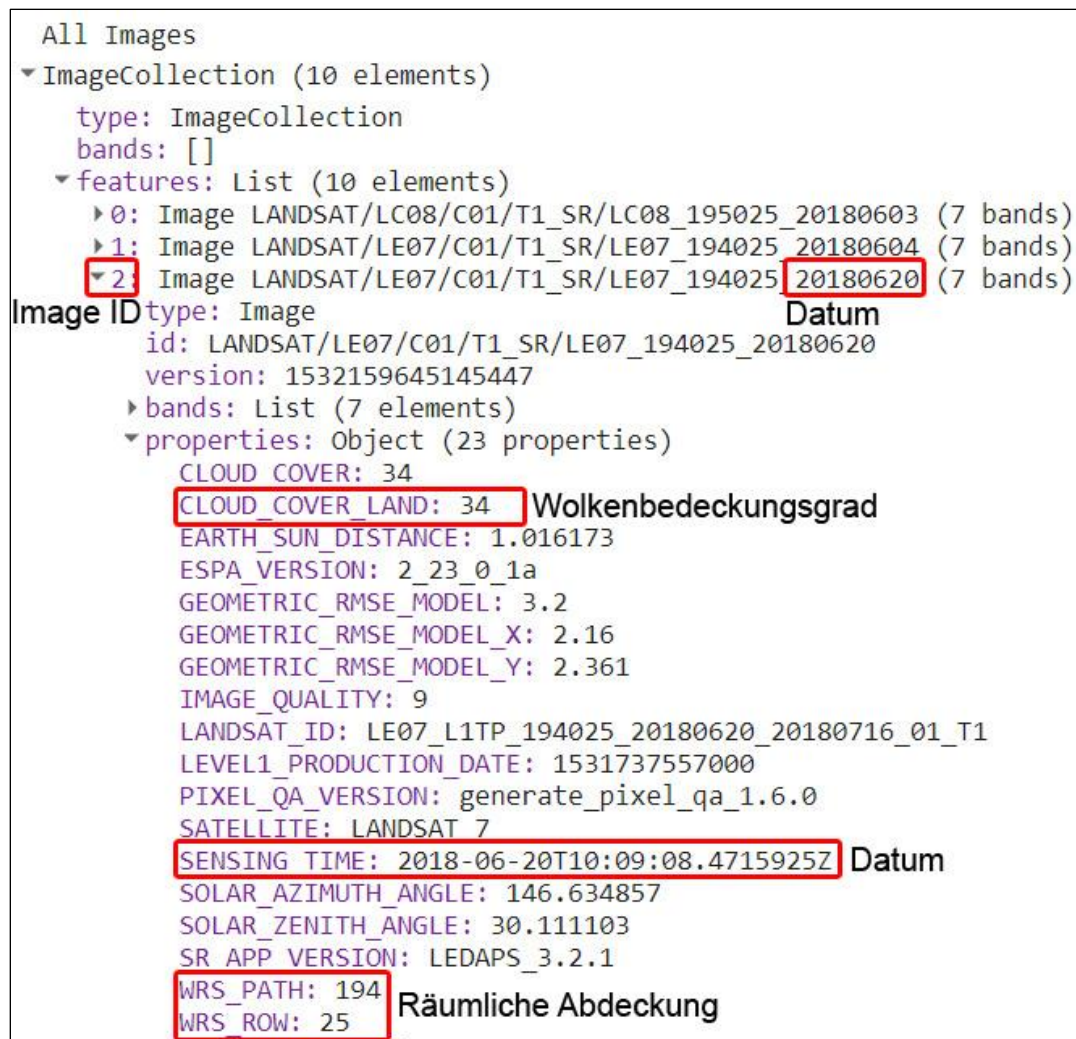


Abbildung 11: Image ID einer Szene der Image Collection sowie ein Ausschnitt der Metadaten dieser.

Mit Abbildung 12 wird das Exportieren einer einzelnen Szene dargestellt, hierbei werden alle verfügbaren Bänder, darunter auch alle berechneten Indizes, der Szene selektiert und exportiert. Zusätzlich werden durch die `.getInfo()` Befehle Informationen aus den jeweiligen Metadaten herangezogen und in den Exportnamen integriert. Dadurch entsteht eine einzigartige, wiedererkennbare Bezeichnung für das Ergebnis, im Falle der Durchführung von Abbildung 12 würde diese wie folgt lauten: **Landsat_7_Image_from_2018-06-20T10-09-08**.

Ein Kritikpunkt hierbei liegt in der Bezeichnung der Bänder in den weiterführenden GIS-Programmen, diese werden in manchen Programmen nicht als *B*, *G*, *R* etc. angezeigt, sondern lediglich als *Band_1*, *Band_2*, *Band_3* etc.

```

591 // EXPORTING SINGLE IMAGES
592 // Landsat single Image Export
593 if (satellite == 'Landsat' && export_singleimage == 'Yes' && Differenced != 'Yes') {
594   var listOfImages = temp.toList(temp.size());
595   var singleimage = ee.Image(listOfImages.get(imageID));
596   var singleimage = singleimage.select(['B', 'G', 'R', 'NIR', 'SWIR1', 'SWIR2',
597                                       'NDVI', 'NDWI', 'NBR', 'NDSI']);
598
599   var date = singleimage.date().format().getInfo();
600   var date = date.replace(':', '-');
601   var date = date.replace(':', '-');
602   var landsat_satellite = singleimage.get('SATELLITE').getInfo().toLowerCase().slice(1);
603
604   Export.image.toDrive({
605     image: singleimage,
606     description: 'L' + landsat_satellite + '_Single_Image_from_' + date + '_' + region,
607     folder: 'GEE_Export',
608     scale: scale,
609     crs: epsg,
610     region: rect_geodesic
611   })

```

Abbildung 12: Export von einzelnen Landsat- und Sentinel-2 Szenen.

3. Quellenverzeichnis

- EBRAHIMY, H, AGHIGHI, H., AZADBAKHT, M., AMANI, M., MAHDAVI, S., MATKAN, A.-A. (2021): „Downscaling MODIS Land Surface Temperature Product Using an Adaptive Random Forest Regression Method and Google Earth Engine for a 19-Years Spatiotemporal Trend Analysis over Iran“. In: *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 14, S. 2103–2111.
- EUROPEAN SPACE AGENCY (ESA) (o.J.): Sentinel Online – Resolution and Swath. Online abrufbar unter: <https://sentinel.esa.int/web/sentinel/missions/sentinel-2/instrument-payload/resolution-and-swath>, [Letzter Zugriff: 03.03.2021].
- FOGA, S, SCARAMUZZA, P.-L., GUO, S., ZHU, Z., DILLEY, R.-D., BECKMANN, T., SCHMIDT, G.-L., DWYER, J.-L., JOSEPH HUGHES, M., LAUE, B. (2017): „Cloud detection algorithm comparison and validation for operational Landsat data products“. In: *Remote Sensing of Environment*, 194, S. 379–390.
- GAO, B (1996): „NDWI—A normalized difference water index for remote sensing of vegetation liquid water from space“. In: *Remote Sensing of Environment*, 58 (3), S. 257–266.
- GOOGLE EARTH ENGINE (Hrsg.) (o.J.): Earth Engine Data Catalog. Online abrufbar unter: <https://developers.google.com/earth-engine/datasets>, [Letzter Zugriff: 14.01.2021].
- GORELICK, N., HANCHER, M., DIXON, M., ILYUSHCHENKO, S., THAU, D., MOORE, R. (2017): „Google Earth Engine: Planetary-scale geospatial analysis for everyone“. In: *Remote Sensing of Environment* 202, S. 17–28.
- GU, C, ZHANG, Y., LIU, L., LI, L., LI, S., ZHANG, B., CUI, B., RAI, M.-K. (2021): „Qualifying Land Use and Land Cover Dynamics and Their Impacts on Ecosystem Service in Central Himalaya Transboundary Landscape Based on Google Earth Engine“. In: *Land*, 10 (2), S. 173.
- GUO, Y.-T., ZHANG, X.-M., LONG, T.-F. JIAO, W.-L., HE, G.-J., YIN, R.-Y., DONG, Y.-Y. (2020): „China forest cover extraction based on Google Earth Engine“. In: *ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, XLII-3/W10, S. 855–862.
- KUMAR, L, MUTANGA, O. (2018): „Google Earth Engine Applications Since Inception: Usage, Trends, and Potential“. In: *Remote Sensing*, 10 (10), S. 1509.
- NGUYEN, M, BAEZ-VILLANUEVA, O., BUI, D., NGUYEN, P., RIBBE, L. (2020): „Harmonization of Landsat and Sentinel 2 for Crop Monitoring in Drought Prone Areas: Case Studies of Ninh Thuan (Vietnam) and Bekaa (Lebanon)“. In: *Remote Sensing*, 12 (2), S. 281.
- PARKS, S, HOLSINGER, L., VOSS, M., LOEHMAN, R., ROBINSON, N. (2018): „Mean Composite Fire Severity Metrics Computed with Google Earth Engine Offer Improved Accuracy and Expanded Mapping Potential“. In: *Remote Sensing*, 10 (6), S. 879.
- SCHMITT, M, HUGHES, L.-H., QIU, C., ZHU, X.-X. (2019): „Aggregating Cloud-free Sentinel-2 Images with Google Earth Engine“. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, IV-2/W7, S. 145–152.
- UNITED STATES GEOLOGICAL SURVEY (USGS) (o.J.): *Landsat 8 Surface Reflectance Quality Assessment*. Online abrufbar unter: <https://www.usgs.gov/core-science->

systems/nli/landsat/landsat-8-surface-reflectance-quality-assessment?qt-science_support_page_related_con=1#qt-science_support_page_related_con, [Letzter Zugriff: 17.02.2021].

- YOUNG, N.-E., ANDERSON, R.-S., CHIGNELL, S.-M., VORSTER A.-G., LAWRENCE R., EVANGELISTA, P.-H. (2017): „A survival guide to Landsat preprocessing“. In: *Ecology* 98 (4), S. 920–932.
- XIONG, J., THENKABAIL, P.-S., GUMMA, M.-K., TELUGUNTLA, P., POEHNELT, J., CONGALTON, R.-G., YADAV, K., THAU, D. (2017): „Automated cropland mapping of continental Africa using Google Earth Engine cloud computing“. In: *ISPRS Journal of Photogrammetry and Remote Sensing*, 126, S. 225–244.
- ZHU, X, HELMER, E.-H. (2018): „An automatic method for screening clouds and cloud shadows in optical satellite image time series in cloudy regions“. In: *Remote Sensing of Environment*, 214, S. 135–153.