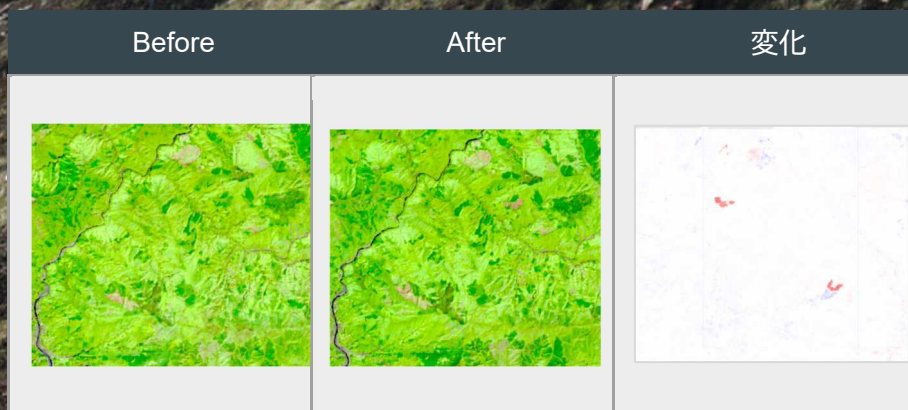


FAMOST を活用した

# 伐採マップ作成マニュアル

Google Earth Engine を使って変化抽出



林野庁

令和8年3月

## 目次

1	はじめに .....	138
1.1	背景・目的.....	138
1.2	FAMOST の概略.....	138
1.3	処理の流れ.....	139
1.4	出力仕様 .....	140
1.5	用語 .....	140
2	Google Earth Engine のアカウント作成と初期設定 .....	141
3	FAMOST スクリプトの準備 .....	142
3.1	AOI の準備と設定.....	142
3.2	スクリプトの新規作成と貼り付け .....	144
3.3	主要パラメータ（ユーザー設定） .....	145
4	実行手順 .....	147
4.1	実行（Run）と画面の見方.....	147
4.2	表示レイヤ（本スクリプト） .....	149
4.3	差分指標を調整したいとき .....	149
4.4	Export の設定（任意） .....	150
5	Export（Tasks）とファイルのダウンロード方法.....	151
6	トラブルシューティング（FAQ） .....	152
7	付録：スクリプト全文.....	153

## 1 はじめに

本書は、FAMOST (伐採・擾乱候補抽出) を Google Earth Engine (以下 GEE) Code Editor 上で実行し、結果を GeoTIFF として出力し、GIS で利用するまでの手順をまとめたものです。市町村が無償で使用することを念頭に公開してきた、旧来の Web UI (フォーム入力→ボタン実行) による FAMOST 運用システムが、Google earth engine 利用規約の改定により無償での利用ができなくなったことを受け、スクリプト運用が可能な技術者向けの手引きとして作成しました。

本マニュアルでは「ある程度 GIS を扱え、スクリプトに記載の変数や Asset ID 等の一部を書き換えられる利用者」を想定しています。GEE の利用形態 (無料/有料)、利用規約、処理上限、入力データの更新状況などにより、実行可否や結果が変わる可能性があります。

### 1.1 背景・目的

本章では、マニュアルの背景・目的と、運用の前提 (どこを利用者が設定し、どこが固定値か) を整理します。FAMOST のスクリプトは、解析範囲 (AOI)、比較する期間 (before/after)、雲・影の条件、最小パッチ (mmu) 等をユーザーが設定できます。それ以外の処理はスクリプト内の既定値で動作します。

旧 UI で行っていた「県の選択・最小面積の入力・実行」相当の操作は、GEE Code Editor では (1) AOI の指定、(2) ユーザー設定変数の編集、(3) Run および Tasks 実行、に置き換わります。

### 1.2 FAMOST の概略

FAMOST は、林野庁事業「令和 3 年度 森林情報活用促進事業のうち「無断伐採の把握体制の整備」と「令和 4 年度 伐採造林届出制度の効果的な運用に向けたリモートセンシング技術活用促進に関する調査事業」で開発された伐採地把握システムです。本マニュアルでは、**Sentinel-2 衛星画像**を対象期間の前後で比較し、植生・水分・焼失等に関する複数の指標の差分から「森林域の変化 (伐採・擾乱の可能性が高い場所)」を抽出する処理を示しています。抽出結果は確定した伐採地ではなく“候補”であり、必ず目視確認や現地情報等と併用して利用してください。

実行スクリプトでは、クラウドマスク処理された Sentinel-2 画像 (Sentinel-2 SR の SCL と

s2cloudless) を用いて雲・影等を除外した上で、複数指標の閾値判定を AND 条件で重ね合わせ、最後に mmu で小領域ノイズを除去します。

### 1.3 処理の流れ

ここでは、運用時に利用者が準備するものと、巻末の 7.付録に記載したスクリプト内部の処理ステップ（節 0～7）の関係を整理します。AOI（解析対象範囲）の GEE への登録と、スクリプトが参照する Asset の権限確認に初回は少し時間を要する場合があります。

運用の基本は「入力データを整える → スクリプト実行で結果を確認 → 必要に応じて条件を調整（繰り返し） → Export して GIS で利用」です（図 1）。

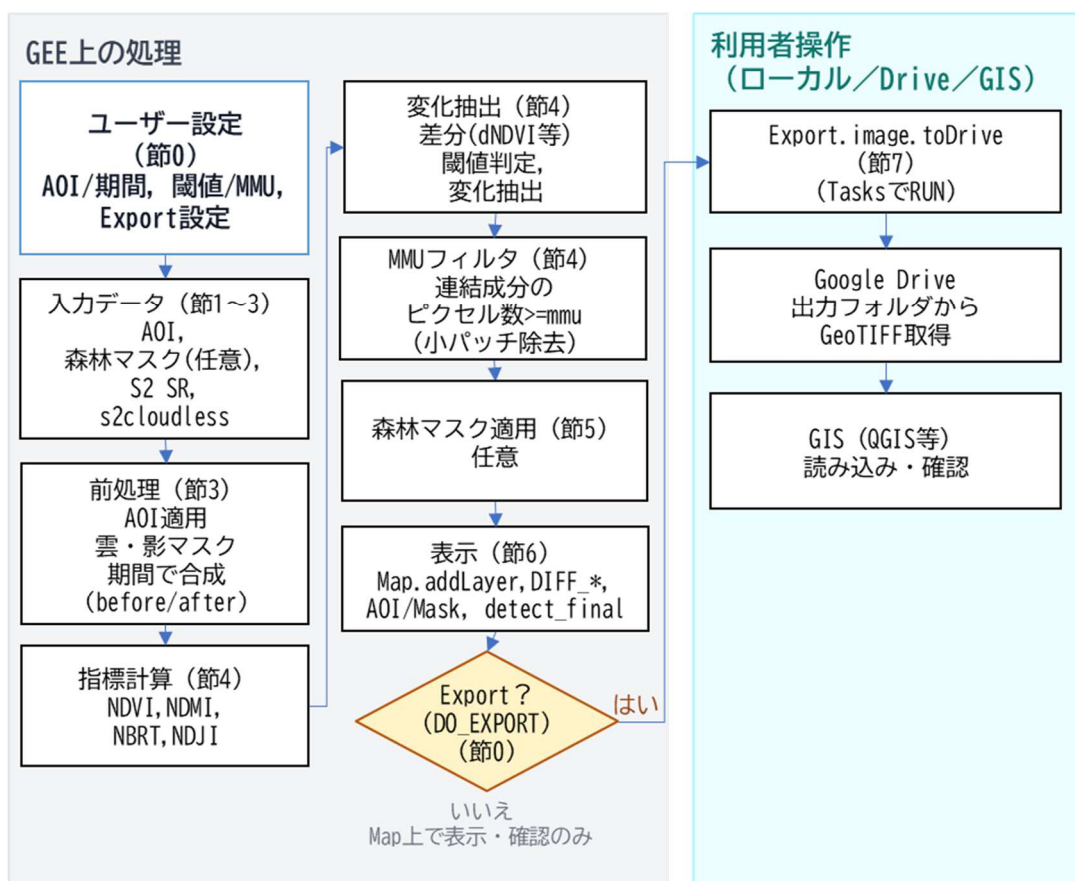


図 1 処理フロー図

## ■ 自分で準備するもの

- 解析対象範囲の GIS ポリゴン (AOI) をまず準備し、これを GEE の Assets に登録し、FeatureCollection (Asset ID) を取得します。
- (任意) 森林マスク画像 (Forest Mask) を準備し、GEE の Assets に登録。森林域のみ抽出したい場合に使用します。
- 出力先となる Google Drive (フォルダ名はスクリプトの EXPORT\_FOLDER で指定)
- 動作確認用に小さめの AOI (10km 四方程度) を用意しておく、エラーの切り分けがしやすくなります。

## ■ スクリプトの流れ

スクリプト本体は付録に記載されています。

- 節 0: ユーザー設定 (AOI/Export/雲条件/ mmu /期間など、運用で主に変更する部分)
- 節 1: AOI の作成 (FeatureCollection から解析範囲 Geometry を作成し、AOI 外をマスク)
- 節 2: 森林マスク (任意で設定。USE\_FOREST\_MASK=true の場合に適用)
- 節 3: Sentinel-2 SR と s2cloudless による雲・影マスク、期間内の画像を合成
- 節 4: 指標差分の計算と閾値判定 (複数指標を AND で統合)、mmu で小領域を除去
- 節 5~7: 表示用レイヤ追加、Export (任意)

## 1.4 出力仕様

出力される GeoTIFF は 0/1 の数値で構成される画像で、値 1 が「伐採・擾乱候補」を表します。候補域には誤検出 (過検出) や見逃し (検出漏れ) が含まれるため、差分画像 (DIFF\_NDVI, DIFF\_NBRT) や before/after の擬似カラー表示と合わせて確認してください。

Export の既定設定では、座標系に EPSG:3857、解像度に 10m (EXPORT\_SCALE=10) を指定しています。AOI が UTM 帯をまたぐ場合などは、出力座標系を明示することで後工程 (GIS) での取扱いが安定します。

## 1.5 用語

- AOI: 解析対象範囲 (Area of Interest)
- Task: GEE の Export 等の非同期処理
- Asset: GEE の Assets に登録されたデータ (画像・ベクタ)
- mmu: Minimum Mapping Unit。本スクリプトでは「連結したピクセル数」が mmu 以上

の領域だけを残し、小さなノイズを除去

- **before/after** : 比較する 2 期間の期首の年と期末の年。2 時期とも展葉期の同じ期間 (例 : 5/1-10/31) で設定し、フェノロジー差の影響を抑える
- **Export** : GEE の結果を Drive 等へ書き出す処理。Code Editor の Tasks で手動実行

## 2 Google Earth Engine のアカウント作成と初期設定

GEE の利用には Google アカウントと Earth Engine の利用登録が必要です。すでに利用可能なアカウントがある場合は、Code Editor にログインできること (画面が表示されること) をまず確認してください。

推奨されるブラウザは、Chrome/Edge (最新版推奨) です。

登録の入口 :

- Earth Engine トップ : [earthengine.google.com](https://earthengine.google.com)

✓ 新規登録は以下のサイトを参考 :

<https://zenn.dev/labcode/books/2xvfkoic4p7biv/viewer/3dwu9c-3>

- ログインの入口 : Code Editor : [code.earthengine.google.com](https://code.earthengine.google.com)

※ 組織の運用方針により、Google Cloud プロジェクトの作成や請求設定 (Billing) が必要になる場合があります。利用できない場合は、組織内の管理者・担当者に確認してください。

ログインすると、図2に示したような画面が表示されます。左上に、Script/Docs/Assets タブが配置、中央上にスクリプトのエディタ画面が配置、右上に Inspector/Console/Tasks タブ、下に Map 画面が配置されています。

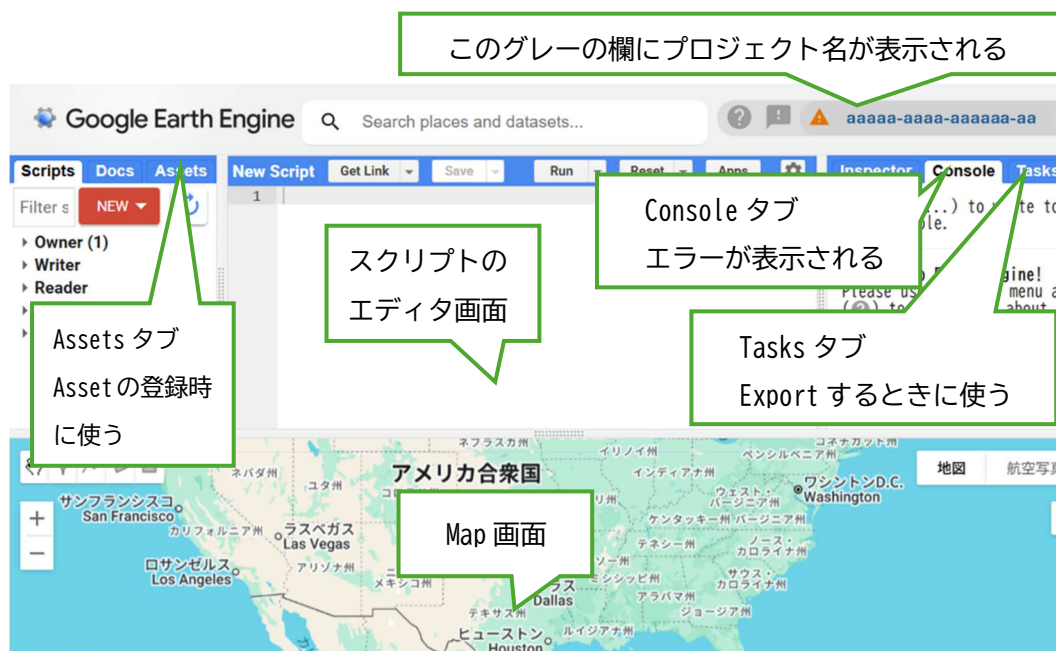


図2 GEE Code Editor トップ画面

### 3 FAMOST スクリプトの準備

本章では、付録に掲載したスクリプト全文を GEE Code Editor に貼り付け、実行できる状態に整えます。運用上、利用者が変更するのは主に「節 0：ユーザー設定」です（AOI、期間、雲条件、mmu、Export 設定など）。

スクリプトが参照する Asset（AOI の FeatureCollection、森林マスク等）が存在し、かつ閲覧権限があることが前提です。権限が不足していると、実行時に Asset not found / Permission denied などのエラーになります。

#### 3.1 AOI の準備と設定

AOI は解析対象範囲を示すポリゴンで、GEE の Assets に FeatureCollection として登録しておく必要があります。旧 UI の「県選択」に相当する範囲指定は、AOI（ポリゴン）で実現します。

##### ■ AOI を用意する方法（例）

- (1) QGIS 等 GIS ソフトでポリゴンを作成（既存のポリゴンでも利用可能）

- (2) SHP ファイル一式を出力
- (3) GEE Code Editor の Assets タブからアップロード（QGIS で出力時の qmd ファイルは Asset に登録しない）（図 3）
- (4) 表示されるアセット名（Asset ID）をスクリプトの AOI\_FC\_ASSET\_ID の該当箇所に設定（「3.3 主要パラメータ（ユーザー設定）」で設定）

複数ポリゴンを有する AOI を 1 つの FeatureCollection として運用する場合は、GIS ソフトであらかじめ属性フィールド（例：pref\_code、name 等）を持たせ、スクリプトの AOI\_FILTER\_FIELD / AOI\_FILTER\_VALUE で抽出対象を切り替えます。

※ AOI のポリゴンが自己交差等で不正な場合、エラーや結果の欠落の原因になります。アップロード前に GIS 側でジオメトリチェックを推奨します。



図 3 GEE Code Editor の Assets タブからの SHP ファイル登録画面

## 3.2 スクリプトの新規作成と貼り付け

GEE Code Editor では、「Scripts」でスクリプトを新規作成し（図4）、本文末尾（付録）のスクリプトをそのまま貼り付けて保存します。貼り付け後は、まず保存しましょう。

### ■ 流れ

- GEE Code Editor の「Scripts」→「NEW」→「FILE」を選択し、Create file 画面で File Name を指定
- スクリプト全体を貼り付け、保存（Save）（図5）

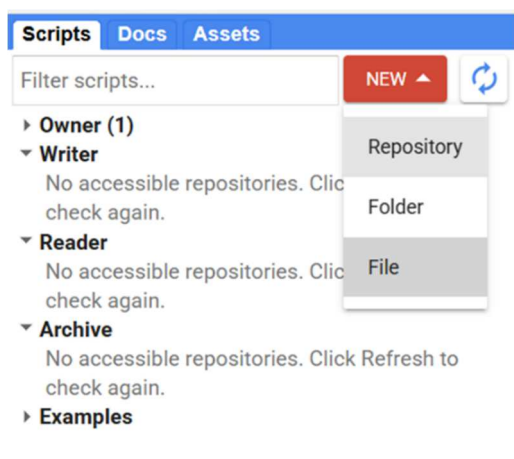


図4 Scripts パネルでスクリプトを新規作成

```
59 // 0. ユーザー設定（ここだけ変えれば運用できる）
60 // =====
61 // -----
62 // AOI (FeatureCollection)
63 // -----
64 // -----
65 var AOI_FC_ASSET_ID = 'projects/プロジェクト名/assets/AssetsにアップしたAOIのAsset名';
66
67 // AOIを「複数から選択」する場合：属性列名と値を設定（単一AOIの場合は null）
68 var AOI_FILTER_FIELD = null; // 例：'CODE' / 'id' / 'NAME' 等。単一AOIなら null
69 var AOI_FILTER_VALUE = null; // 例：45 / 'Tokyo' / ['A','B'] 等。単一AOIなら null
70
71 // 表示/出力用のAOI名称（任意）
72 // (AOIが複数候補からの選択でも、出力名に混ぜて識別しやすくするため)
73 var AOI_NAME = 'AOI';
74
75 // -----
76 // Forest mask（任意）
77 // -----
78 var USE_FOREST_MASK = true; // true: Forestマスクを使う / false: 使わない
79
80 var FOREST_ASSET_ID = 'projects/プロジェクト名/assets/Assetsにアップした森林マスクのAsset名';
81
82 // -----
83 // 出力（Export）
84 // -----
85 var DO_EXPORT = true; // true: Export 実行 / false: Export しない
86 var EXPORT_FOLDER = 'GEE export FOREST';
```

図5 貼り付け後のエディタ画面

### 3.3 主要パラメータ（ユーザー設定）

本スクリプトは、冒頭の「節0：ユーザー設定」を編集します。まずは小さな AOI を使い既定値のまま動作確認し、その後に期間・雲条件・mmu・閾値を調整してください。

#### ■最初に調整することが多いパラメータ

- 閾値（dNDMI/dNDVI/dNBRT/dNDJI）はスクリプト内に記載された固定値を使うが、必要に応じてスクリプトの節4に記載の閾値を調整可能
- 値を変更したら保存し、再実行（Run）

#### ■主な設定項目（節0：ユーザー設定）

設定箇所	必須/任意で書き直し	意味
AOI_FC_ASSET_ID	必須	解析対象ポリゴン（FeatureCollection）の Asset ID（projects/プロジェクト名/assets/Asset名）を必ず入力
AOI_FILTER_FIELD, AOI_FILTER_VALUE	複数ポリゴンを含む AOI を使う場合は必須	複数のポリゴンを含む AOI ファイル（例えば林班や市町村ポリゴン）が 1 つの FeatureCollection に入っている場合に、属性で抽出対象を絞り込むための設定。一つのポリゴンのみの AOI の場合は両方 null を入力
USE_FOREST_MASK, FOREST_ASSET_ID	任意	（任意）森林域に限定して抽出したい場合に使用。Asset ID を入力。 Forest Mask で設定された値により、スクリプト内の gt(0) を eq(1) 等に調整（コメント「2 節に記載の★マスク仕様に依って変更」参照）
DO_EXPORT	任意	伐採マップを GIS ソフト等で表示したい場合は true のまま、出力不要の場合は false に書き換え
EXPORT_FOLDER	出力する場合は Export 必須	利用者の Google Drive にフォルダをあらかじめ作成し、その名前書き換える
CLOUD_FILTER	任意	画像を解析に使うか選ぶためのフィルタ Sentinel-2 メタデータの雲量（CLOUDY_PIXEL_PERCENTAGE）の上限を設定。値を小さくすると雲が少ない画像に絞られるが、画像枚数が不足しやすくなる
CLOUD_PROB_TH	任意	画像を選んだあと、雲を“ピクセル単位”で消すためのしきい値 雲確率しきい値（probability > CLOUD_PROB_TH を雲とみなす）を設定
startYear endYear startDay endDay	年は必須、日には任意	before/after の期間設定 原則として同じ期間で比較することを推奨
mmu	任意	最小パッチ（ピクセル数） 抽出結果の連結領域のうち、連結ピクセル数が mmu 以上の領域だけを残す

mmu (最小パッチ) と connectedPixelCount の意味

スクリプトでは、まず閾値条件を満たすピクセルを changes として作成し、connectedPixelCount により各ピクセルが属する連結領域の「連結ピクセル数」を計算します (今回は 20 ピクセルで設定)。その後 .gte(mmu) により mmu 以上の領域のみを残します (= mmu 未満の小領域を除去)。connectedPixelCount の引数 20 は“連結ピクセル数の集計上限”で、連結領域が 20 ピクセルを超える場合でも 20 で打ち切ってカウントします。したがって、mmu=10 のように  $mmu \leq 20$  であれば運用上問題ありませんが、mmu を 20 より大きく設定する場合は、connectedPixelCount の上限値も同等以上に見直してください。

ユーザー設定のスクリプト箇所を下に抜粋しています。青色でハイライトした箇所は必ず入力し直しが必要で、黄色でハイライトした箇所は必要に応じて入力し直してください。

```
// 0. ユーザー設定 (ここだけ変えれば運用できる)
var AOI_FC_ASSET_ID = 'projects/プロジェクト名/assets/Assets にアップした AOI の
Asset 名'; //利用者のプロジェクト名と Asset 名を入力
var AOI_FILTER_FIELD = null; //複数 AOI のときは例: 'CODE' / 'id' / 'NAME' 等,
単一 AOI なら null
var AOI_FILTER_VALUE = null; //複数 AOI のとき例: 45 / 'Tokyo' / ['A', 'B'] 等,
単一 AOI なら null
var AOI_NAME = 'AOI';

var USE_FOREST_MASK = true; // 森林マスクを使うかは任意 true: Forest Mask を使
う / false: 使わない
var FOREST_ASSET_ID = 'projects/プロジェクト名/assets/Assets にアップした森林マスク
の Asset 名'; //利用者のプロジェクト名と Asset 名を入力

var DO_EXPORT = true; // true: Export 実行 / false: Export しない
var EXPORT_FOLDER = 'GEE_export_FAMOST';//任意のフォルダを利用者の Google Drive に
作成
var EXPORT_CRS = 'EPSG:3857';
var EXPORT_SCALE = 10; //ピクセルサイズ(m)

var CLOUD_FILTER = 50;
var CLOUD_PROB_TH = 50;
var mmu = 10;
```

```
var startYear = 2022;  
var endYear   = 2023;  
var startDay  = '-05-01';  
var endDay    = '-10-31';
```

## 4 実行手順

実行前に、節 0 (ユーザー設定) の AOI・期間・雲条件・mmu (必要なら森林マスク、Export 設定) を再確認しておきます。

Run によりスクリプトを実行すると、マップ画面の地図上のレイヤ (Layers) に結果と確認用の補助レイヤが追加されます。

マップ画面での確認手順は、

- (1) AOI が正しい位置・範囲か、
- (2) before/after の擬似カラー (S2 before/after) で季節や雲の影響が大きすぎないか、
- (3) 差分画像 (DIFF\_NDVI や DIFF\_NBRT) の青～赤色の分布、
- (4) 最終結果 detect\_final で示される伐採候補箇所と (3) の画像を見比べ、  
の順に確認し、必要に応じてパラメータを調整する流れです。

### 4.1 実行 (Run) と画面の見方

Run を押すと (図 6)、画面上部のスクリプトが GEE サーバ側で実行され、Map/Layers/Console に結果が反映されます。実行後の Map 画面とレイヤ画面を図 7 に示します。図 8 に Deforestation の箇所が示されたマップ画面 (上図)、NBRT の差分画像が表示されたマップ画面とレイヤ画面 (下図) を示します。初回や AOI が広い場合は計算量が大きく、実行に時間がかかることがあります。

エラーが出た場合は、Console の赤字メッセージに表示される行番号・内容を手掛かりに、Asset ID の誤り/権限不足、期間や AOI が大きすぎる、パラメータの型 (null/配列/文字列) の不一致、などを切り分けます。

※ Export (Drive 出力) は Run だけでは開始しません。DO\_EXPORT=true の場合に Tasks にタスクが作成され、Tasks 側で RUN すると Export が開始します。

```

FAMOST
Get Link Save Run Reset Apps
125 if (Array.isArray(AOI_FILTER_VALUE)) {
126   AOI_FC = AOI_FC_ALL.filter(ee.Filter.inList(AOI_FILTER_VALUE));
127 } else {
128   AOI_FC = AOI_FC_ALL.filter(ee.Filter.eq(AOI_FILTER_VALUE));
129 }
130 }
131
132 var AOI = AOI_FC.geometry();
133
134 // AOI内マスク (AOI外は解析しない)
135 var aoi_mask = ee.Image().byte().paint(AOI_FC, 1).rename('AOI_mask').unmask(0);
136
137 // 出力名に付ける識別子 (AOI_FILTER_VALUE があればそれも入れる)
138 var aoi_id_str = (AOI_FILTER_VALUE === null) ? 'ALL'
139   : (Array.isArray(AOI_FILTER_VALUE) ? 'MULTI' : String(AOI_FILTER_VALUE));
140
141 var name_string = AOI_NAME + '_' + aoi_id_str + '_' + startYear + 'to' + endYear;
142
143

```

Run ボタン

図6 Run ボタン

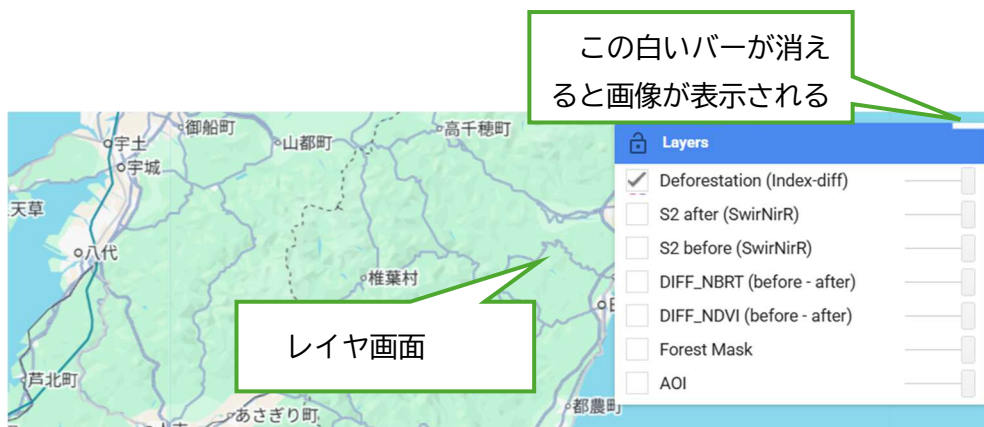
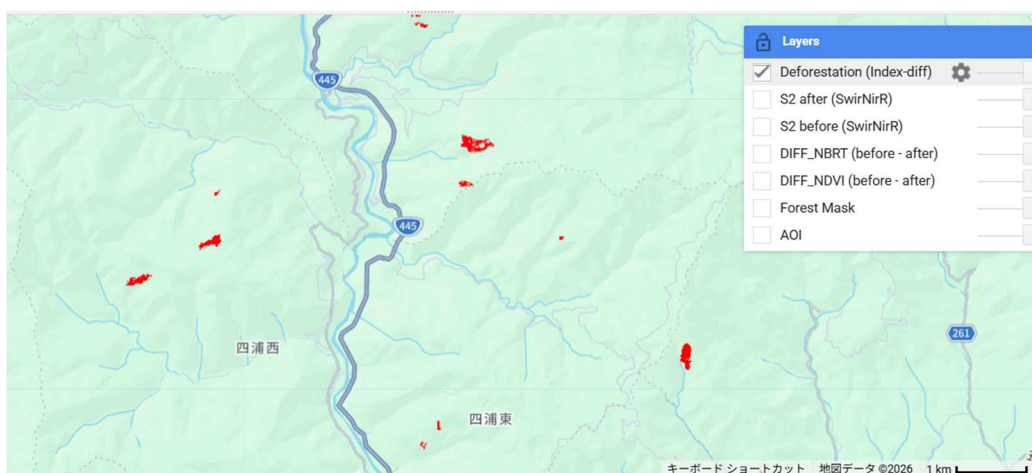
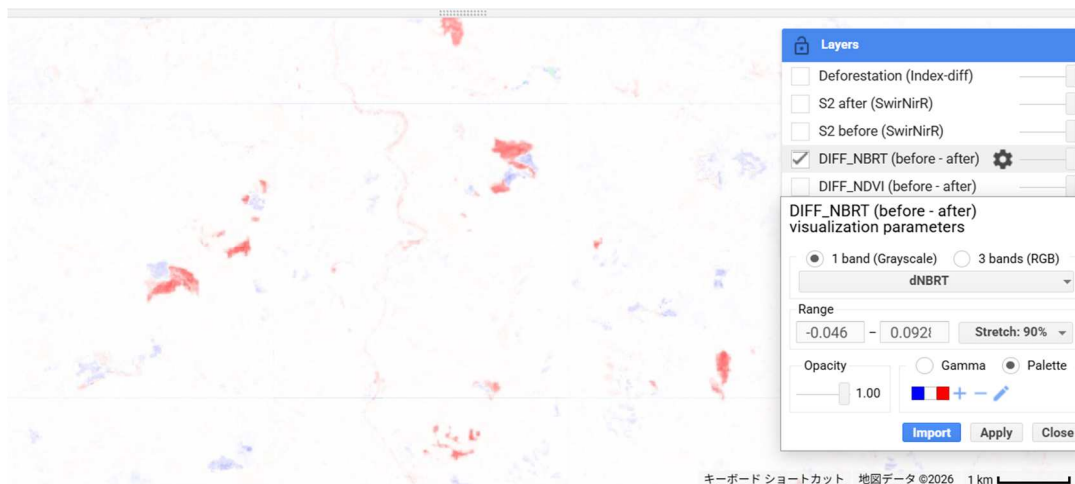


図7 実行後の Map 画面



Deforestation が表示されたところで、(伐採候補地) が赤く表示されている状態



Layers 画面の NBRT の差分画像 (DIFF\_NBRT (before-after)) にチェックを入れ、画像を表示しているところ。赤い箇所で差分値がプラス、青がマイナスを示す

図 8 Layers パネルと画像の表示

## 4.2 表示レイヤ (本スクリプト)

本スクリプトは、結果確認のために複数のレイヤを Map に追加します。初期表示 ON/OFF はレイヤごとに設定しているため、必要に応じて Layers で表示を切り替えてください (図 8)。

差分レイヤ (DIFF\_NDVI や DIFF\_NBRT) や before/after の擬似カラーは、過検出・検出漏れの原因 (雲影、季節差、裸地・造成等) を判断するのに有効です。検出結果だけで判断せず、必ずこれらの補助レイヤと合わせて確認します。

### ■ 各レイヤで表示される画像と初期表示設定

- Deforestation (Index-diff) : 抽出結果 (初期 ON)
- S2 before (SwirNirR) / S2 after (SwirNirR) : 確認用の擬似カラー (初期 OFF)
- DIFF\_NBRT : 差分画像 (before-after) (初期 OFF)
- DIFF\_NDVI : 差分画像 (before-after) (初期 OFF)
- Forest Mask : 森林マスク表示 (USE\_FOREST\_MASK=true の場合、初期 OFF)
- AOI : 解析範囲の表示 (初期 OFF)

## 4.3 差分指標を調整したいとき

伐採を抽出する際に使用される指標は下表に示した 4 つです。なお、NBRT について、本来は熱赤外バンドを有する Landsat といった衛星に適用されるものですが、FAMOST 事業に

において熱赤外を短波長赤外(SWIR2)に割当てて Sentinel-2 に適用した変則版となります。

INDEX	Equation	Reference
Normalized Difference Moisture Index 正規化水分指数	$NDMI = \frac{NIR - SWIR}{NIR + SWIR}$	Gao, 1996 <a href="https://doi.org/10.1016/S0034-4257(96)00067-3">https://doi.org/10.1016/S0034-4257(96)00067-3</a>
Normalized Difference Vegetation Index 正規化植生指数の差分	$NDVI = \frac{NIR - RED}{NIR + RED}$	Tucker & Sellers, 1986 <a href="https://doi.org/10.1080/01431168608948944">https://doi.org/10.1080/01431168608948944</a>
Normalized Burn Ratio-Thermal 正規化燃焼指数(熱)	$NBRT = \frac{NIR - 0.0001 \times SWIR \times T}{NIR + 0.0001 \times SWIR \times T}$	Holden et. al, 2005 <a href="https://doi.org/10.1080/01431160500239008">https://doi.org/10.1080/01431160500239008</a>
Normalized Difference JAFTA Index 正規化日林協指数	$NDJI = \frac{BLUE + GREEN - RED}{BLUE + GREEN + RED}$	FAMOST 用に日林協が開発した指数

これらの4つの指標の before/after の差分値を使って伐採地を抽出しており、下に示したスクリプトの該当箇所の数値を変更することで伐採候補地抽出の調整が可能です。数値を小さくするほど、見逃しが減りますが誤検出が増え、数値を大きくするほど、見逃しは増えますが誤検出が減ります。

```

288
289 // 閾値判定
290 var ndmiMask = DIFF_NDMI.gt(0.40);
291 var ndviMask = DIFF_NDVI.gt(0.25);
292 var nbrtMask = DIFF_NBRT.gt(0.38);
293 var ndjiMask = DIFF_NDJI.gt(0.30);
294

```

#### 4.4 Export の設定 (任意)

Export を行う場合は、節 0 で DO\_EXPORT=true とし、出力先フォルダ名

(EXPORT\_FOLDER)、座標系 (EXPORT\_CRS)、解像度 (EXPORT\_SCALE) などを必要に応じて設定します(図 9)。

Export の region は AOI (Geometry) で指定されています。AOI が非常に広い場合や期間が長い場合は、計算時間の増加や maxPixels 制限により失敗することがあります。その場合は AOI を分割する、期間を短くする、scale を粗くする等に対応します。

```
04
85 // -----
86 // 出力 (Export)
87 // -----
88 var DO_EXPORT      = true;           // true: Export 実行 / false: Export しない
89 var EXPORT_FOLDER  = 'GEE_export_FAMOST';
90 var EXPORT_CRS     = 'EPSG:3857';   // 推奨: EPSG:3857 (メートル単位)
91 var EXPORT_SCALE   = 10;           // 10m
92
```

図 9 Export 設定 (節 0 ユーザー設定)

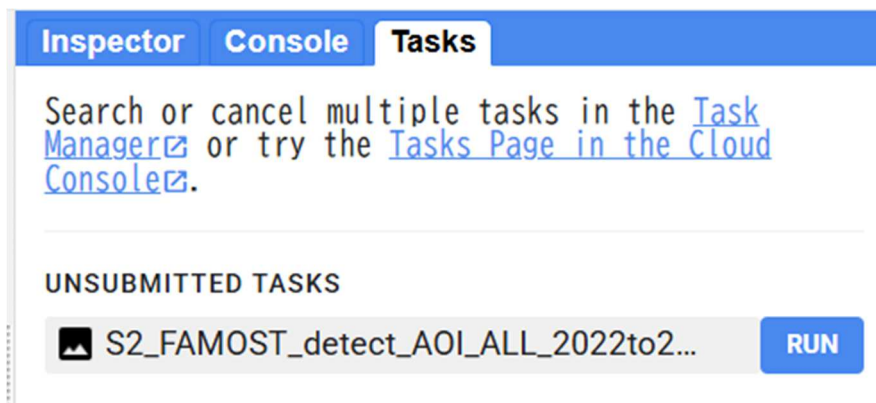
## 5 Export (Tasks) とファイルのダウンロード方法

Run 実行後、DO\_EXPORT=true の場合は Tasks タブに Export タスクが追加されます。Tasks の RUN を押すことで Drive への出力が開始され、完了すると指定フォルダに GeoTIFF が作成されます(図 10)。

出力ファイル名 (description) はスクリプト内で自動生成しており、AOI 名・選択値・期間などが付与されます。運用でファイル管理しやすい命名になっているか、必要に応じて name\_string の組み立てを調整してください。

### ■ 流れ

- スクリプト実行後、Tasks タブに Export タスクが追加される
- Tasks で該当タスクを選択し「RUN」を押す (出力先 Drive フォルダを確認)
- Status が COMPLETED になったら Drive でファイルを確認する
- Drive 上の出力フォルダから GeoTIFF をダウンロードする (右クリック→ダウンロード (zip の場合は要解凍))



処理が成功すると、Tasks に出力ファイル名と青い RUN ボタン表示される



出力が完了すると、Google Drive の出力フォルダにファイルが格納される

図 10 Export 時の手動操作

## 6 トラブルシューティング (FAQ)

ここでは、実行時によくあるエラー・症状と、切り分けの観点をまとめます。多くの場合、(1) Asset ID/権限、(2) AOI・期間が大きすぎる、(3) 雲条件・閾値が厳しすぎる／緩すぎる、のいずれかです。

原因が特定しにくい場合は、まず AOI を小さくし、期間を短くし、Forest Mask を一旦 OFF にするなど、段階的に条件を緩めていくと切り分けが容易です。

### ■ よくあるエラーと確認事項

- Asset not found : Asset ID の誤り、権限不足、プロジェクト切替を確認する。
- Permission denied : Asset 共有設定、GEE アカウント (メール) を確認し、権限付与を依頼する。
- Export 失敗 : AOI が広すぎる / 期間が長い / maxPixels 不足 / scale 不適切などを見直す。

- 結果が空：AOI、期間、森林マスク、閾値（SI\_THRESHOLD 等）を段階的に緩めて原因を切り分ける。
- 過検出/検出漏れ：SI\_THRESHOLD、雲条件、mmu、解析期間を調整し、目視で妥当性を確認する。
- タスクが出ない：DO\_EXPORT が false になっていないか、Run 後に Tasks を更新したかを確認する。
- 処理が非常に遅い：AOI/期間を縮小し、CLOUD\_FILTER を緩めて画像数を確保する。必要なら EXPORT\_SCALE を粗くする。

## 7 付録：スクリプト全文

```
// =====
// FAMOST マニュアル用：Sentinel-2 伐採抽出 (GEE Code Editor)
// =====
// 実行環境：Google Earth Engine Code Editor
//
// -----
// 【使用している座標系】
// -----
// 1) AOI (FeatureCollection / Geometry)
//   - Earth Engine 内部の Geometry は基本的に WGS84 (EPSG:4326) の緯度経度として扱われます。
//   - もとの SHP が別の座標系でも、Assets 化された時点で EE 内では EPSG:4326 相当で扱われます。
//     (※ただし、SHP の座標系情報が欠落していると位置ずれの原因になります)
//
// 2) 表示 (Map)
//   - GEE Code Editor の地図表示は Web Mercator (EPSG:3857) 相当で描画されます。
//     (表示の座標系であり、解析・演算の座標系を固定するものではありません)
//
// 3) Sentinel-2 画像 (COPERNICUS/S2_SR_HARMONIZED)
//   - Sentinel-2 はタイルごとに UTM 座標系 (EPSG:326xx/327xx) で提供されます。
//   - 本スクリプトの合成 (median) 後の画像は、内部的には入力画像の投影を引き継いで扱われます。
//     AOI が複数 UTM 帯に跨る場合、出力時に座標系を明示しておくことで運用が安定します。
//
// 4) 出力 (Export)
//   - 本スクリプトでは Export 時の出力座標系を EXPORT_CRS で指定できます。
//   - 推奨 (汎用)：EPSG:3857 (メートル単位で扱える)
//     例) EXPORT_CRS='EPSG:3857', EXPORT_SCALE=10
//   - 注意：EPSG:4326 (緯度経度) を指定すると scale は「度」単位になり、10m 指定が崩れます。
//     よって本マニュアル用途では EPSG:3857 を推奨します。
//
// -----
// 【事前に準備するデータ】
// -----
// A) AOI (対象範囲)
//   - GEE Assets にアップロードした FeatureCollection (ポリゴン) を使用します。
//   - 例：都道府県境界 SHP → Assets 化 → AOI_FC_ASSET_ID に Asset ID を設定
//   - AOI が「単一ポリゴンだけ」の場合：AOI_FILTER_FIELD / VALUE を null にして全体を使います。
```

```

// - AOI が「複数ポリゴンから選択」する場合：識別用属性列（例：CODE, ID, NAME 等）を用意し
//   AOI_FILTER_FIELD / AOI_FILTER_VALUE で 1 つ選択します。
// - AOI_FILTER_VALUE は 1 つ（数値/文字列）でも、配列（複数候補）でも OK
//   - 1 つ：eq フィルタ
//   - 配列：inList フィルタ
//
// B) Forest Mask（森林マスク：任意）
// - 森林マスクを使う場合：GEE Assets の森林ラスタ（ee.Image）の Asset ID を FOREST_ASSET_ID
に設定。
// - USE_FOREST_MASK=true/false で「使う/使わない」を切り替え可能。
// - マスク値仕様に応じて、Forest.gt(0) を Forest.eq(1) 等に変更が必要な場合があります。
//
// C) Sentinel-2 データ
// - GEE 公式カタログを使用（追加準備不要）
//
// -----
// 【このスクリプトの仕様（要点）】
// -----
// 1) AOI は汎用可能（ユーザーが興味があるエリアを設定）
// 2) 伐採候補箇所を NDVI, NDMI, NBRT, NDJI の差分の閾値で判定
// 3) 差分画像も表示して、伐採抽出候補と比較できる（NDVI と NBRT）
//
// =====
// 0. ユーザー設定
// =====

// -----
// AOI (FeatureCollection)
// -----
//var AOI_FC_ASSET_ID = 'projects/*****/assets/*****';
var AOI_FC_ASSET_ID = 'projects/*****/assets/*****';//TIF 画像

// AOI を「複数から選択」する場合：属性列名と値を設定（単一 AOI の場合は null）
var AOI_FILTER_FIELD = null; // 例：'CODE' / 'id' / 'NAME' 等。単一 AOI なら null
var AOI_FILTER_VALUE = null; // 例：45 / 'Tokyo' / ['A','B'] 等。単一 AOI なら null

// 表示/出力用の AOI 名称
var AOI_NAME = 'AOI';

// -----
// Forest Mask (任意)
// -----
var USE_FOREST_MASK = true; // true: Forest Mask を使う / false: 使わない

var FOREST_ASSET_ID = 'projects//*****/assets//*****';

// -----
// 出力 (Export)
// -----
var DO_EXPORT = true; // true: Export 実行 / false: Export しない
var EXPORT_FOLDER = 'GEE_export_FAMOST';
var EXPORT_CRS = 'EPSG:3857'; // 推奨：EPSG:3857（メートル単位）
var EXPORT_SCALE = 10; // 10m

```

```

// -----
// 解析期間・雲条件など
// -----
var CLOUD_FILTER = 50; // Sentinel-2 メタデータの雲量 (CLOUDY_PIXEL_PERCENTAGE)
var CLOUD_PROB_TH = 50; // s2cloudless の雲確率しきい値 (probability > 50 を雲)

// 最小パッチ (mmu)
var mmu = 10; //ピクセル数

// before / after 期間設定 (例: 2022→2023 の同時期比較)
var startYear = 2022;
var endYear   = 2023;
var startDay  = '-05-01';
var endDay    = '-10-31';

var DATE_START_1 = startYear + startDay;
var DATE_END_1   = startYear + endDay;
var DATE_START_2 = endYear   + startDay;
var DATE_END_2   = endYear   + endDay;

// =====
// 1. AOI の作成 (単一でも選択式でも対応)
// =====

var AOI_FC_ALL = ee.FeatureCollection(AOI_FC_ASSET_ID);

var AOI_FC;
if (AOI_FILTER_FIELD === null || AOI_FILTER_VALUE === null) {
  // 単一 AOI (または全体を AOI として使用)
  AOI_FC = AOI_FC_ALL;
} else {
  // 選択式 (1つ or 配列)
  if (Array.isArray(AOI_FILTER_VALUE)) {
    AOI_FC = AOI_FC_ALL.filter(ee.Filter.inList(AOI_FILTER_FIELD, AOI_FILTER_VALUE));
  } else {
    AOI_FC = AOI_FC_ALL.filter(ee.Filter.eq(AOI_FILTER_FIELD, AOI_FILTER_VALUE));
  }
}

var AOI = AOI_FC.geometry();

// AOI 内マスク (AOI 外は解析しない)
var aoi_mask = ee.Image().byte().paint(AOI_FC, 1).rename('AOI_mask').unmask(0);

// 出力名に付ける識別子 (AOI_FILTER_VALUE があればそれも入れる)
var aoi_id_str = (AOI_FILTER_VALUE === null) ? 'ALL'
              : (Array.isArray(AOI_FILTER_VALUE) ? 'MULTI' : String(AOI_FILTER_VALUE));

var name_string = AOI_NAME + '_' + aoi_id_str + '_' + startYear + 'to' + endYear;

Map.centerObject(AOI, 9);
Map.addLayer(AOI, {color: 'red'}, 'AOI', false);

// =====

```

```

// 2. Forest Mask (任意: ON/OFF)
// =====

var Forest = ee.Image(FOREST_ASSET_ID);

// マスク仕様に応じて変更: gt(0) → eq(1) など
var ForestMask = Forest.gt(0).rename('ForestMask');

// Forest Mask を使わない場合は「全て 1 のマスク」を使う (AOI 内に限定)
var ForestMaskOrAll = ee.Image(1).rename('ForestMaskOrAll').updateMask(aoi_mask);
if (USE_FOREST_MASK) {
  ForestMaskOrAll = ForestMask.updateMask(aoi_mask); // AOI 内に限定
  Map.addLayer(
    ForestMaskOrAll.updateMask(ForestMaskOrAll),
    {palette: ['#ffc71a']},
    'Forest Mask',
    false
  );
}

// =====
// 3. Sentinel-2 SR + s2cloudless マスク (AOI 内)
// =====

function getS2Collection(startDate, endDate) {

  // Sentinel-2 SR (L2A)
  var s2_sr = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')
    .filterBounds(AOI)
    .filterDate(startDate, endDate)
    .filter(ee.Filter.lte('CLOUDY_PIXEL_PERCENTAGE', CLOUD_FILTER));

  // s2cloudless
  var s2_cloudprob = ee.ImageCollection('COPERNICUS/S2_CLOUD_PROBABILITY')
    .filterBounds(AOI)
    .filterDate(startDate, endDate);

  // system:index で join
  var joined = ee.ImageCollection(ee.Join.saveFirst('s2cloudless').apply({
    primary: s2_sr,
    secondary: s2_cloudprob,
    condition: ee.Filter.equals({
      leftField: 'system:index',
      rightField: 'system:index'
    })
  }));

  // cloud_prob / clouds バンド追加
  function addCloudBands(img) {
    var cldPrb = ee.Image(img.get('s2cloudless')).select('probability');
    var isCloud = cldPrb.gt(CLOUD_PROB_TH).rename('clouds');
    return img.addBands(cldPrb.rename('cloud_prob'))
      .addBands(isCloud);
  }
}

```

```

// SCL + clouds でマスク (AOI 内のみ)
function applyMasks(img) {
  var scl = img.select('SCL');

  // SCL 除外クラス (2,3,7~11)
  var invalid = scl.eq(2) // Topographic shadows
    .or(scl.eq(3)) // Cloud shadows
    .or(scl.eq(7)) // Cloud low prob.
    .or(scl.eq(8)) // Cloud med prob.
    .or(scl.eq(9)) // Cloud high prob.
    .or(scl.eq(10)) // Thin cirrus
    .or(scl.eq(11)); // Snow / ice

  var validSCL = invalid.not();
  var validCloud = img.select('clouds').not();
  var mask = validSCL.and(validCloud);

  return img.updateMask(mask)
    .updateMask(aoi_mask); // AOI 内のみ
}

return joined.map(addCloudBands)
  .map(applyMasks);
}

var col_before = getS2Collection(START_DATE_1, END_DATE_1);
var col_after = getS2Collection(START_DATE_2, END_DATE_2);

// median 合成
var s2_before = col_before.median().select('B.*').clip(AOI);
var s2_after = col_after.median().select('B.*').clip(AOI);

// =====
// 4. 指標 (NDVI/NDMI/NBRT/NDJI) 差分と閾値判定 (before - after)
// =====
// ※ Shadow Index (SI) と影マスクは【省略】
//
// 閾値 (before - after が以下を超える)
// - NDMI: > 0.40
// - NDVI: > 0.25
// - NBRT: > 0.38
// - NDJI: > 0.30

// NDMI
var ndmi_bf = s2_before.normalizedDifference(['B8', 'B11']).rename('NDMI');
var ndmi_at = s2_after.normalizedDifference(['B8', 'B11']).rename('NDMI');

// NDVI
var ndvi_bf = s2_before.normalizedDifference(['B8', 'B4']).rename('NDVI');
var ndvi_at = s2_after.normalizedDifference(['B8', 'B4']).rename('NDVI');

// NBRT
var nbrt_bf = s2_before.expression(

```

```

'(NIR - 0.0001 * SWIR * Temp) / (NIR + 0.0001 * SWIR * Temp)', {
  'NIR': s2_before.select('B8'),
  'SWIR': s2_before.select('B11'),
  'Temp': s2_before.select('B12')
}).rename('NBRT');

var nbrt_at = s2_after.expression(
'(NIR - 0.0001 * SWIR * Temp) / (NIR + 0.0001 * SWIR * Temp)', {
  'NIR': s2_after.select('B8'),
  'SWIR': s2_after.select('B11'),
  'Temp': s2_after.select('B12')
}).rename('NBRT');

// NDJI
var ndji_bf = s2_before.expression(
'(BLUE + GREEN - RED) / (BLUE + GREEN + RED)', {
  'BLUE': s2_before.select('B2'),
  'GREEN': s2_before.select('B3'),
  'RED': s2_before.select('B4')
}).rename('NDJI');

var ndji_at = s2_after.expression(
'(BLUE + GREEN - RED) / (BLUE + GREEN + RED)', {
  'BLUE': s2_after.select('B2'),
  'GREEN': s2_after.select('B3'),
  'RED': s2_after.select('B4')
}).rename('NDJI');

// 差分 (before - after)
var DIFF_NDMI = ndmi_bf.subtract(ndmi_at).rename('dNDMI');
var DIFF_NDVI = ndvi_bf.subtract(ndvi_at).rename('dNDVI');
var DIFF_NBRT = nbrt_bf.subtract(nbrt_at).rename('dNBRT');
var DIFF_NDJI = ndji_bf.subtract(ndji_at).rename('dNDJI');

// 閾値判定
var ndmiMask = DIFF_NDMI.gt(0.40);
var ndviMask = DIFF_NDVI.gt(0.25);
var nbrtMask = DIFF_NBRT.gt(0.38);
var ndjiMask = DIFF_NDJI.gt(0.30);

// 4 指標すべて満たす (AND)
var changes = ndmiMask.add(ndviMask).add(nbrtMask).add(ndjiMask)
  .eq(4)
  .selfMask();

// mmu (最小パッチ)
// connectedPixelCount(20) は連結ピクセルの集計上限 (20)
var detect_raw = changes
  .connectedPixelCount(20)
  .gte(mmu)
  .selfMask();

// =====
// 5. Forest Mask 適用 (任意: ON/OFF)

```

```

// =====
var detect_final = detect_raw;
if (USE_FOREST_MASK) {
  detect_final = detect_raw.updateMask(ForestMaskOrAll);
}

// =====
// 6. 表示 (確認用)
// =====

// (a) 差分画像: 指定どおり NDVI / NBRT のみ表示
Map.addLayer(
  DIFF_NDVI,
  {min: -0.6, max: 0.6, palette: ['0000ff', 'ffffff', 'ff0000']},
  'DIFF_NDVI (before - after)',
  false
);

Map.addLayer(
  DIFF_NBRT,
  {min: -1.0, max: 1.0, palette: ['0000ff', 'ffffff', 'ff0000']},
  'DIFF_NBRT (before - after)',
  false
);

// (b) S2 BEFORE / AFTER (確認用: SwirNirR)
Map.addLayer(
  s2_before.select(['B11', 'B8', 'B5']).divide(10000),
  {min: 0, max: 0.3},
  'S2 before (SwirNirR)',
  false
);

Map.addLayer(
  s2_after.select(['B11', 'B8', 'B5']).divide(10000),
  {min: 0, max: 0.3},
  'S2 after (SwirNirR)',
  false
);

// (c) 伐採抽出結果
Map.addLayer(
  detect_final,
  {min: 0, max: 1, palette: ['000000', 'ff0000']},
  'Deforestation (Index-diff)',
  true
);

// =====
// 7. Export (任意)
// =====

if (DO_EXPORT) {

```

```
Export.image.toDrive({
  image: detect_final,
  description: 'S2_FAMOST_detect_' + name_string,
  folder: EXPORT_FOLDER,
  region: AOI,
  crs: EXPORT_CRS,
  scale: EXPORT_SCALE,
  maxPixels: 1e13
});
}
```