

- ④ 印刷レイアウトウィンドウの左にある「新しい地図のレイアウト追加」ボタンをクリックすると、地図に新しい地図を任意の大ききで配置することができる。この時点において配置される地図のイメージは QGIS の画面に表示されている地図と同じとなる。一度配置した地図は、「移動」ボタンをクリックすることにより、印刷の配置そのまま、地図の位置のみを移動することができる（図 3.8.3）。
- ⑤ 任意で「注釈」や「凡例」および「縮尺」ボタンをクリックすることにより、それらを地図上の任意の場所に配置することができる（図 3.8.3）。
- ⑥ 最後に、印刷レイアウトウィンドウの上にある印刷ボタンをクリックすることによって、地図を印刷することができる（図 3.8.3）。

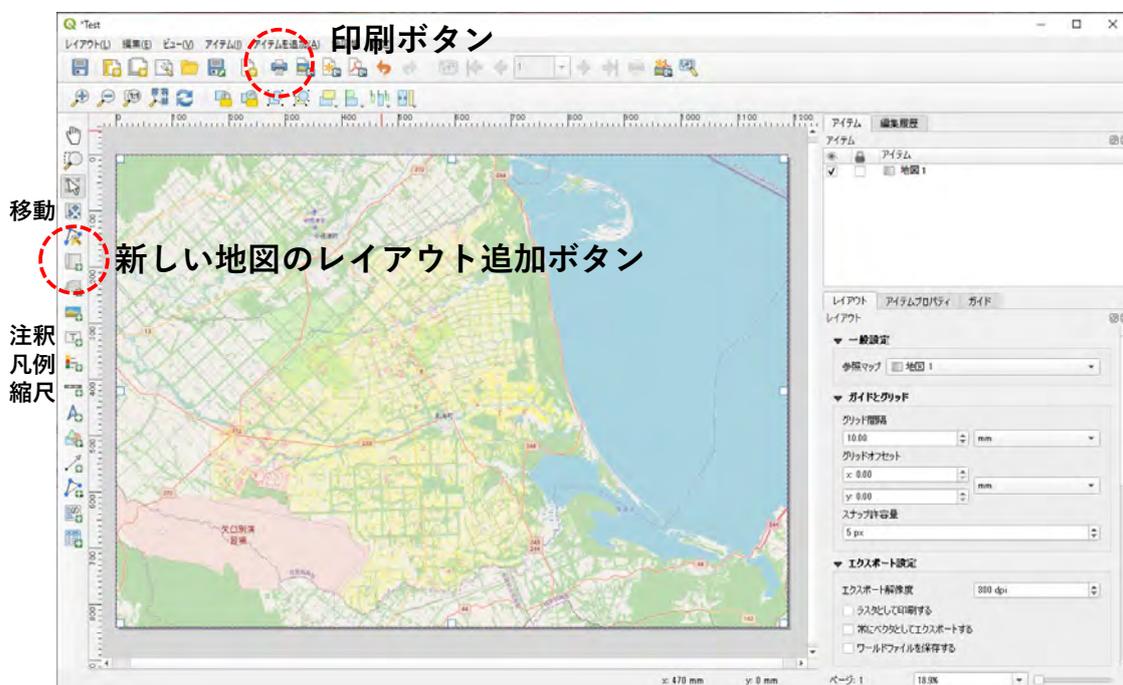


図 3.8.3 地図の調整と印刷

以上で、衛星データにより草地において耕作・管理されているか否かの判定を行い、現地調査用の地図を出力する方法の説明は終了である。なお、利用に当たっては、「第4章 その他事項」を参照の上、利用すること。

## 第4章 その他事項

本マニュアルの記載事項、特に設定値や設定方法は、「令和元年度衛星画像解析による現地確認作業の効率化手法の開発・調査委託事業」および「令和2年度衛星画像解析による現地確認作業の効率化手法の開発・調査業務」等の成果を反映したものである。これら業務では「田」「畑」については2自治体、「草地」については1自治体における成果であることから、全国展開した場合には、地域固有の問題や再設定が発生することが予想される。以下、予想される発生事象とその対処方法を記載する。

### (1) 後方散乱強度最低値の閾値の調整

わずかな農用地においてはあるが（全体の0.3%以下）、Sentinel-1において、耕作放棄地であるにも関わらず、耕作・管理されていると誤判別された農用地があった。この原因として、観測直前の降雨による雑草の形状変化によって、見かけ上、後方散乱強度が低下した可能性がある。この対策として、周辺に耕作・管理されていない可能性がある農用地がある程度まとまっている場合、念のためその周辺の農用地を確認する、もしくはアメダスデータと比較し、まとまった雨量が観測された直後のSARデータは使用しない、などの対応が考えられる。

また、本手法では、たとえ耕作放棄地が全くなかったとしても、全体の数%（後方散乱強度最低値が正規分布している場合、平均+2×標準偏差を根拠とすると2.5%）は調査対象1（耕作放棄地と推定）となり、逆に本業務の対象地域よりも耕作放棄地が多い地方自治体では、衛星データで解析した結果以上に調査すべき農用地があることが想定される。このことから、本業務における閾値は一つの目安にはなるものの、絶対的な設定値ではなく、地方自治体の実情に応じて、適切に設定する必要がある。

### (2) 積雪による解析期間の調整

SARを活用した「田」「畑」および光学センサを活用した「草地」のいずれの手法においても、積雪がある期間は適用できない。そのため各年の気象状況に合わせて、積雪がある期間を避けるよう、調整する必要がある。

### (3) 冬季の耕作放棄地の状況による解析期間の調整

冬季において耕作放棄地の雑草が枯れることにより、耕作放棄地であるにもかかわらず農用地と誤判定される可能性がある。一般的に雑草の生育は他の農作物よりも速やかなため、冬季の気象条件を考慮し、解析期間の開始時期を後ろにずらすなどの調整が必要になる可能性がある。

#### (4) 小さい農用地の取扱いについて

SAR を活用した「田」「畑」および光学センサを活用した「草地」のいずれの手法においても、衛星画像の空間解像度による制約のため、小さい農用地は対象外となっている。このマニュアルにおいても 5m の緩衝帯処理後、PALSAR-2 では 125 m<sup>2</sup>(20 画素)以上、Sentinel-1 では 500 m<sup>2</sup> (5 画素) 以上、PlanetScope では 100 m<sup>2</sup> (11 画素) 以上の農用地を対象としている。このため、協定農用地に、これより小さい農用地がある場合は、本マニュアルの対象外となるため、別途現地調査が必要となる。

## 添付資料

Sentinel-1 ダウンロードのための Google Earth Engine スクリプト

```

// A UI to interactively filter a collection, display the results, and export it.

// The namespace for our application. All the state is kept in here.
var app = {};

/** Creates the UI panels. */
app.createPanels = function() {
  /* The introduction section. */
  app.intro = {
    panel: ui.Panel([
      ui.Label({
        value: 'Sentinel-1 ダウンロードツール',
        style: {fontWeight: 'bold', fontSize: '18px', margin: '5px 2px'}
      }),
      ui.Label('指定した条件下で Sentinel-1 画像を検索し¥n ダウンロードする',
{whiteSpace: 'pre'})
    ])
  };

  /* The collection filter controls. */
  app.filters = {
    startDate: ui.Textbox('YYYY-MM-DD', '2019-03-01'),
    endDate: ui.Textbox('YYYY-MM-DD', '2019-09-30'),
    orbit: ui.Select({value: app.ORBIT_OPTIONS[0],items: app.ORBIT_OPTIONS}),
    applyButton: ui.Button('適用', app.applyFilters),
    loadingLabel: ui.Label({
      value: 'Loading...',
      style: {stretch: 'vertical', color: 'gray', shown: false}
    })
  };

  /* The panel for the filter control widgets. */
  app.filters.panel = ui.Panel({
    widgets: [
      ui.Label('1) 検索条件', {fontWeight: 'bold'}),
      ui.Label('検索開始日', app.HELPER_TEXT_STYLE), app.filters.startDate,

```

```

    ui.Label('検索終了日', app.HELPER_TEXT_STYLE), app.filters.endDate,
    ui.Label('昇降選択', app.HELPER_TEXT_STYLE), app.filters.orbit,
    ui.Panel({widgets: [
        app.filters.applyButton,
        app.filters.loadingLabel
    ]}, style:app.SECTION_STYLE))
],
style: app.SECTION_STYLE
});

app.line= ui.Panel({
    widgets: [
        ui.Label('-----表示オプションツール-----', {fontWeight: 'bold'}),
    ],
    style: app.SECTION_STYLE
});

/* The image picker section. */
app.picker = {
    select: ui.Select({
        onChange: app.refreshMapLayer
    }),
};

/* The panel for the picker section with corresponding widgets. */
app.picker.panel = ui.Panel({
    widgets: [
        ui.Label('画像を一枚選択', {fontWeight: 'bold'}),
        ui.Panel([
            app.picker.select,
        ], ui.Panel.Layout.flow('horizontal'))
    ],
    style: app.SECTION_STYLE
});

/* The visualization section. */

```

```

app.vis = {
  label: ui.Label(),
  select: ui.Select({
    items: Object.keys(app.VIS_OPTIONS),
    onChange: function() {
      var option = app.VIS_OPTIONS[app.vis.select.getValue()];
      app.vis.label.setValue(option.description);
      app.refreshMapLayer();
    }
  })
};

/* The panel for the visualization section with corresponding widgets. */
app.vis.panel = ui.Panel({
  widgets: [
    ui.Label('偏波を選択', {fontWeight: 'bold'}),
    app.vis.select,
    app.vis.label
  ],
  style: app.SECTION_STYLE
});

app.vis.select.setValue(app.vis.select.items().get(0));

/* The export section. */
app.export = {
  button: ui.Button({
    label: 'Tasks に一覧作成',
    onClick: function() {
      var filtered = ee.ImageCollection(app.COLLECTION_ID);
      filtered = filtered.filterBounds(geometry)
        .filter(ee.Filter.eq('instrumentMode', 'IW'))
        .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
        .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
        .select(['VV'])
      var start = app.filters.startDate.getValue();

```

```

if (start) start = ee.Date(start);
var end = app.filters.endDate.getValue();
if (end) end = ee.Date(end);
if (start) filtered = filtered.filterDate(start, end);
var orbit = app.filters.orbit.getValue();
var orbit2 = ""
  if (orbit == '昇交') {
    orbit2 = 'ASCENDING'
  }
  else{
    orbit2 = 'DESCENDING'
  }
filtered = filtered.filter(ee.Filter.eq('orbitProperties_pass', orbit2));

var imageList = filtered.toList(200);
print('画像枚数=' + imageList.size().getInfo());
if (imageList.size().getInfo() > 50){
  print('画像枚数が制限 (50 枚) を超えています。フィルタを再設定してくださ
い。');
}
else{
  for(var i = 0; i < imageList.size().getInfo(); i++){
    var image = ee.Image(imageList.get(i)).clip(geometry);
    print(i + ',' + image.get('system:index').getInfo())

Export.image.toDrive({image:image.reproject('EPSG:4326',null,10),description:image.get('
system:index').getInfo(),scale:10,folder:'sentinel1',region:geometry})
  }
}
})
};

/* The panel for the export section with corresponding widgets. */
app.export.panel = ui.Panel({
  widgets: [

```

```

        ui.Label('2) ダウンロード一覧作成', {fontWeight: 'bold'}),
        app.export.button
    ],
    style: app.SECTION_STYLE
});

};

/** Creates the app helper functions. */
app.createHelpers = function() {
    app.setLoadingMode = function(enabled) {
        app.filters.loadingLabel.style().set('shown', enabled);
        var loadDependentWidgets = [
            app.vis.select,
            app.filters.startDate,
            app.filters.endDate,
            app.filters.applyButton,
            app.filters.orbit,
            app.picker.select,
            app.export.button
        ];
        loadDependentWidgets.forEach(function(widget) {
            widget.setDisabled(enabled);
        });
    };

    /** Applies the selection filters currently selected in the UI. */
    app.applyFilters = function() {
        app.setLoadingMode(true);
        var filtered = ee.ImageCollection(app.COLLECTION_ID);
        filtered = filtered.filterBounds(geometry)
            .filter(ee.Filter.eq('instrumentMode', 'IW'))
            .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VV'))
            .filter(ee.Filter.listContains('transmitterReceiverPolarisation', 'VH'))
            .select(['VV']);
    };
};

```

```

// Set filter variables.
var start = app.filters.startDate.getValue();
if (start) start = ee.Date(start);
var end = app.filters.endDate.getValue();
if (end) end = ee.Date(end);
if (start) filtered = filtered.filterDate(start, end);
var orbit = app.filters.orbit.getValue();
var orbit2 = ""
  if (orbit == '昇交') {
    orbit2 = 'ASCENDING'
  }
  else{
    orbit2 = 'DESCENDING'
  }
filtered = filtered.filter(ee.Filter.eq('orbitProperties_pass', orbit2));
var computedIds = filtered
  .limit(app.IMAGE_COUNT_LIMIT)
  .reduceColumns(ee.Reducer.toList(), ['system:index'])
  .get('list');

computedIds.evaluate(function(ids) {
  app.setLoadingMode(false);
  app.picker.select.items().reset(ids);
  app.picker.select.setValue(app.picker.select.items().get(0));
});
};

/** Refreshes the current map layer based on the UI widget states. */
app.refreshMapLayer = function() {
  Map.clear();
  var imageId = app.picker.select.getValue();
  if (imageId) {
    var image = ee.Image(app.COLLECTION_ID + '/' + imageId);
    var visOption = app.VIS_OPTIONS[app.vis.select.getValue()];
    Map.addLayer(image, visOption.visParams, imageId);
  }
}

```

```

    };
};

/** Creates the app constants. */
app.createConstants = function() {
    app.COLLECTION_ID = 'COPERNICUS/S1_GRD';
    app.SECTION_STYLE = {margin: '20px 0 0 0'};
    app.HELPER_TEXT_STYLE = {
        margin: '8px 0 -3px 8px',
        fontSize: '12px',
        color: 'gray'
    };

    app.IMAGE_COUNT_LIMIT = 20;
    app.ORBIT_OPTIONS = ['昇交', '降交']

    app.VIS_OPTIONS = {
        'VV': {
            description: '反射が強いところは白、弱いところは黒になります',
            visParams: {gamma: 1.3, min: -20, max: 0, bands: ['VV', 'VV', 'VV']}
        },
        'VV と VH': {
            description: '反射が強いところは白、弱いところは赤～黒になります',
            visParams: {gamma: 1.7, min: -30, max: 15, bands: ['VV', 'VH', 'VH']}
        },
    }
};

/** Creates the application interface. */
app.boot = function() {
    app.createConstants();
    app.createHelpers();
    app.createPanels();
    var main = ui.Panel({
        widgets: [
            app.intro.panel,

```

```
    app.filters.panel,  
    app.export.panel,  
    app.line,  
    app.picker.panel,  
    app.vis.panel  
  ],  
  style: {width: '320px', padding: '8px'}  
});  
Map.setCenter(138, 36, 6);  
ui.root.insert(0, main);  
app.applyFilters();  
};  
  
app.boot();
```