INTELLIGENT RIVERS 2026 · HANDS-ON SESSION 1

Mapping Lake Surface Area from Space

Day 1: Remote Sensing in Transboundary Rivers · 21 November 2026

Prepared by Dr. Hisham Eldardiry, Alexandria University

What you will do

By the end of this session you will be able to:

- Find cloud-free Sentinel-2 or Landsat images for a lake and build a composite image.
- Calculate the Modified Normalized Difference Water Index (MNDWI) and turn it into a water map.
- Estimate the lake's water surface area and test how sensitive it is to the threshold and the sensor.
- Track how the area changes through the seasons and from year to year, and map where the lake grows in wet years.

Along the way there are **ten checkpoints**. At each one, write down the numbers you get and compare them with the "On track if" guide. If a value is far off, fix it before moving on, or raise your hand.

Before you start

1. **Earth Engine account.** Earth Engine requires a registered Google Cloud project. Sign in at code.earthengine.google.com with your Google account, register a free noncommercial (academic) project, and select it in the Code Editor (profile icon, top right → "Change Cloud Project"). Guide: developers.google.com/earth-engine/guides/access.
2. **New script.** In the Code Editor, click *New* → *File*, name it `IR2026_LakeArea`, and paste in the two starting blocks below. You will then add one block of code per step and press **Run** after each step.
3. **Your lake.** Each group works on one lake (Table 1). Set `lakeName` to your lake.
4. **Your sensor.** Set `sensor` to `'Sentinel-2'` or `'Landsat'`. Start with Sentinel-2; in Step 6 you will switch to Landsat once to compare (Table 2).

Table 1. Study lakes. All three lie in or next to Sanjay Gandhi National Park, a few kilometres from the IIT Bombay campus.

Lake	Where to find it	Reported surface area*	Group
Powai Lake	Next to the IIT Bombay campus	≈ 2.1 km²	A
Vihar Lake	Just north-west of Powai Lake	≈ 7 km² (when full)	B
Tulsi Lake	About 6 km north of Powai Lake	≈ 1.35 km²	C

**Approximate published values. Your satellite estimate will differ, and explaining why is part of the exercise. Tip: type the lake name in the Code Editor search bar to jump to it.*

Background: why MNDWI?

Water absorbs strongly in the shortwave infrared (SWIR) and reflects relatively more green light, while soil, vegetation and buildings reflect more SWIR. MNDWI uses this contrast:

$$\text{MNDWI} = (\text{Green} - \text{SWIR1}) \,/\, (\text{Green} + \text{SWIR1})$$

Values range from −1 to 1. Open water is usually clearly positive; land is usually negative. Compared with the original NDWI (Green and near-infrared), MNDWI separates water from built-up areas better, which matters around Mumbai. Table 2 lists the bands used for each sensor.

Table 2. Sensors available in the script.

	Sentinel-2	Landsat 8 and 9
Green band	B3 (10 m)	SR_B3 (30 m)
SWIR1 band	B11 (20 m)	SR_B6 (30 m)
Pixel size used for area	10 m	30 m
New image of the same place	about every 5 days	every 8 days (Landsat 8 and 9 together)
Surface reflectance in Earth Engine since	2017	2013 (Landsat 8), 2021 (Landsat 9)
Earth Engine dataset	COPERNICUS/S2_SR_HARMONIZED	LANDSAT/LC08 and LC09/C02/T1_L2

Starting code

Paste both blocks below, in order, at the top of your new script. Change only the USER INPUT lines.

```
// ====================================================================
// Intelligent Rivers 2026 | Hands-on Session 1
// Mapping lake surface area from space with Sentinel-2 / Landsat and MNDWI
// Infrastructure Nexus Lab, Alexandria University
// Works for any lake: draw a polygon around it and adjust USER INPUT.
// ====================================================================

// ---------------------------- USER INPUT ----------------------------
var sensor    = 'Sentinel-2';   // 'Sentinel-2' or 'Landsat' (Landsat 8 + 9)
var lakeName  = 'Powai Lake';   // any name; used in labels and charts
var startDate = '2025-01-01';   // Steps 2-7: period for one composite
var endDate   = '2025-04-01';
var maxCloud  = 20;             // max % cloud per scene (clouds are also masked per pixel)
var threshold = 0;              // MNDWI value above which a pixel is water
var areaScale = 0;              // 0 = native pixel size; use 30-100 for very large
                                // lakes (e.g., Lake Nasser) to keep runs fast

// Steps 8-10: time series
var firstYear = 2019;           // Sentinel-2: 2019 or later; Landsat: 2014 or later
var lastYear  = 2025;
var seasonStartMonth = 1;       // Steps 8 and 10: same window every year,
var seasonMonths     = 3;       //   here January-March (3 months)
var seasonalYear     = 2024;    // Step 9: year for the month-by-month cycle
// Draw a polygon around your lake and rename it "roi" (see Step 1)
// --------------------------------------------------------------------
```

The SETUP block gives Sentinel-2 and Landsat the same band names (blue, green, red, nir, swir1), converts both to surface reflectance and masks clouds and cloud shadows pixel by pixel. It also defines two helper functions: `getImages()` finds the images for any period, and `waterAreaKm2()` returns the water area of their median composite. Steps 8–10 use both.

```
// ------------ SENSOR SETUP AND HELPERS (no changes needed) ------------
// Both sensors get the same band names, are scaled to surface reflectance
// (0-1) and have clouds and cloud shadows masked pixel by pixel.
var S = {
  'Sentinel-2': {
    ids: ['COPERNICUS/S2_SR_HARMONIZED'],
    bands: ['B2', 'B3', 'B4', 'B8', 'B11'],
    cloudProp: 'CLOUDY_PIXEL_PERCENTAGE',
    pixel: 10,
    prep: function (img) {
      var scl = img.select('SCL');        // scene classification layer
      var clear = scl.neq(3).and(scl.neq(8)).and(scl.neq(9)).and(scl.neq(10));
      return img.updateMask(clear).select(this.bands).divide(10000);
    }
  },
  'Landsat': {
    ids: ['LANDSAT/LC08/C02/T1_L2', 'LANDSAT/LC09/C02/T1_L2'],
    bands: ['SR_B2', 'SR_B3', 'SR_B4', 'SR_B5', 'SR_B6'],
    cloudProp: 'CLOUD_COVER',
    pixel: 30,
    prep: function (img) {
      var clear = img.select('QA_PIXEL').bitwiseAnd(26).eq(0);  // cloud, shadow
      return img.updateMask(clear).select(this.bands)
          .multiply(0.0000275).add(-0.2);
    }
  }
}[sensor];
var pixelSize = areaScale > 0 ? areaScale : S.pixel;   // metres


Map.setOptions('SATELLITE');
```

Working on another lake?

The script is not tied to the Mumbai lakes. To run it on any lake, for example Lake Nasser:

- Draw `roi` around the new lake and set `lakeName`. Step 1 zooms the map to your polygon.
- For very large lakes, set `areaScale` to 30–100 (metres). The area changes very little, and the time-series steps run much faster.
- Choose the dates that suit the lake's own seasons: `startDate`/`endDate` for Steps 2–7, `seasonStartMonth`/`seasonMonths` for the window compared each year (Steps 8 and 10), and `seasonalYear` for the monthly cycle (Step 9).

Step-by-step workflow

Add each block to the **end** of your script and click **Run**. Results appear in the **Console** tab; lines starting with `CP` are your checkpoint values. Steps 8–10 build on the helpers in the SETUP block.

Step 1. Draw your region of interest (ROI)

1. In the map, use the **polygon tool** (top-left of the map) to draw around your lake. Keep the whole lake inside, with only a thin margin of land.
2. At the top of the script, under *Imports*, click the name `geometry` and rename it `roi`.
3. Add the Step 1 code below and click **Run**.

```
// ---- STEP 1: Region of interest ----------------------------------
var roiArea = roi.area(1).divide(1e6);            // m2 -> km2
print('CP1 | ' + lakeName + ' | ROI area (km2):', roiArea);
Map.centerObject(roi);
Map.addLayer(roi, {color: 'yellow'}, 'ROI', true, 0.3);
```

CHECKPOINT 1 **ROI**

Lake name and ROI area (km²): _____

On track if: your ROI is larger than the lake but not by much, roughly 1.2–2 times the reported area in Table 1. If it is 10× larger, redraw a tighter polygon.

Step 2. Find cloud-free images

`getImages()` searches the archive of your chosen sensor between `startDate` and `endDate` (January–March 2025, the dry season in Mumbai, when clouds are rare) and keeps scenes with less than 20% cloud.

```
// ---- STEP 2: Find cloud-free images ------------------------------
// All images of the chosen sensor over the lake between two dates
function getImages(start, end) {
  var col = ee.ImageCollection(S.ids[0]);
  if (S.ids.length > 1) { col = col.merge(ee.ImageCollection(S.ids[1])); }
  return col.filterBounds(roi)
      .filterDate(start, end)
      .filter(ee.Filter.lt(S.cloudProp, maxCloud))
      .map(function (img) {
        return ee.Image(S.prep(img)
            .rename(['blue', 'green', 'red', 'nir', 'swir1'])
            .copyProperties(img, ['system:time_start']));
      });
}


var imgs = getImages(startDate, endDate);
print('CP2 | ' + sensor + ' | Number of images:', imgs.size());
print('Image dates:', imgs.aggregate_array('system:time_start')
    .map(function (t) { return ee.Date(t).format('YYYY-MM-dd'); }));
```

CHECKPOINT 2 **Images found**

Number of images: _____

First and last image dates: _____

On track if: you have at least 5 images for Sentinel-2 or at least 3 for Landsat, all between 1 January and 31 March 2025. If you get 0, check that roi exists and that the dates are typed correctly.

Step 3. Build a composite and look at it

The median of all images removes leftover clouds and shadows. Turn the *True colour* layer on in the Layers panel and zoom in on your lake.

```
// ---- STEP 3: Median composite and true-colour view ---------------
var img = imgs.median().clip(roi);
Map.addLayer(img, {bands: ['red', 'green', 'blue'], min: 0, max: 0.25},
    'True colour (' + sensor + ')');
```

CHECKPOINT 3 **Look before you measure**

What colour is the open water? Do you see green patches on the lake (yes/no)? _____

On track if: the lake looks dark blue-black against green and grey land. Green patches on the water are floating vegetation such as water hyacinth, which is a known problem at Powai Lake. Note them; they matter in Step 6.

Step 4. Calculate MNDWI

Run the code, then open the **Inspector** tab (top right) and click once in the middle of the lake and once on nearby land.

```
// ---- STEP 4: MNDWI = (Green - SWIR1) / (Green + SWIR1) -----------
var mndwi = img.normalizedDifference(['green', 'swir1']).rename('MNDWI');
Map.addLayer(mndwi, {min: -0.6, max: 0.6,
    palette: ['8c510a', 'f6e8c3', 'c7eae5', '01665e']}, 'MNDWI');

var stats = mndwi.reduceRegion({
  reducer: ee.Reducer.minMax().combine(ee.Reducer.mean(), '', true),
  geometry: roi, scale: pixelSize, maxPixels: 1e13, tileScale: 4
});
print('CP4 | MNDWI min / max / mean:', stats);
```

CHECKPOINT 4 **MNDWI values**

MNDWI min / max / mean in the ROI: _____

MNDWI in the lake centre (Inspector): _____

MNDWI on nearby land (Inspector): _____

On track if: the lake centre is clearly positive (typically 0.3 or higher) and land is below 0. If the lake centre is negative, check that you used ['green', 'swir1'], in that order.

Step 5. Turn MNDWI into a water map

Every pixel with MNDWI greater than `threshold` (0 to start) is classified as water and shown in cyan.

```
// ---- STEP 5: Water mask ----------------------------------------
var water = mndwi.gt(threshold).selfMask();
Map.addLayer(water, {palette: ['00FFFF']}, 'Water (MNDWI > threshold)');
```

CHECKPOINT 5 **Check the water map**

Does the cyan mask follow the shoreline? Where does it miss water, or mark land as water?_____

On track if: the mask covers the open lake closely. Gaps on the lake usually mean vegetation; cyan specks on land can be shadows or wet surfaces. Write down what you see.

Step 6. Calculate the water surface area

Each water pixel is multiplied by its area (`ee.Image.pixelArea()`, in m²), and the results are summed over the ROI.

```
// ---- STEP 6: Water surface area ---------------------------------
var areaDict = water.multiply(ee.Image.pixelArea()).reduceRegion({
  reducer: ee.Reducer.sum(), geometry: roi, scale: pixelSize,
  maxPixels: 1e13, tileScale: 4
});
var waterArea = ee.Number(areaDict.get('MNDWI')).divide(1e6);   // km2
print('CP6 | ' + sensor + ' | Water area (km2):', waterArea);
print('Water share of ROI (%):', waterArea.divide(roiArea).multiply(100));
```

CHECKPOINT 6 **Water area**

Water area, Sentinel-2 (km²): _____

Water share of ROI (%): _____

Water area as % of the reported area in Table 1:_____

Now set sensor = 'Landsat' and click Run. Water area, Landsat (km²): _____

On track if: the area is the same order of magnitude as Table 1, usually 50–110% of it. A much lower value at Powai Lake is expected if much of the lake is covered by vegetation, because MNDWI maps open water only. Values above 120% usually mean the ROI includes other water bodies. The two sensors should agree within about 10–20%; switch back to Sentinel-2 before Step 7.

Step 7. Test the threshold

Zero is a common choice, but not the only one. This step repeats the area calculation for six thresholds and plots the result.

```
// ---- STEP 7: How sensitive is the area to the threshold? ---------
var thresholds = [-0.2, -0.1, 0, 0.1, 0.2, 0.3];
var sens = ee.FeatureCollection(ee.List(thresholds).map(function (t) {
  var a = mndwi.gt(ee.Number(t)).multiply(ee.Image.pixelArea())
      .reduceRegion({reducer: ee.Reducer.sum(), geometry: roi,
                     scale: pixelSize, maxPixels: 1e13, tileScale: 4})
      .getNumber('MNDWI');
  return ee.Feature(null, {threshold: t, area_km2: a.divide(1e6)});
}));
print('CP7 | Area for each threshold:', sens);
print(ui.Chart.feature.byFeature(sens, 'threshold', 'area_km2')
    .setChartType('LineChart')
    .setOptions({title: lakeName + ' (' + sensor + '): area vs MNDWI threshold',
                 hAxis: {title: 'MNDWI threshold'},
                 vAxis: {title: 'Water area (km2)'}, pointSize: 5}));
```

CHECKPOINT 7 **Threshold sensitivity**

Area at threshold −0.1: _____

Area at threshold 0.2: _____

Change between the two (%): _____

On track if: the area goes down as the threshold goes up. A small change between −0.1 and 0.2 means the lake edge is sharp and the estimate is robust; a large change means many mixed pixels or vegetation at the edges.

Step 8. Same season, different years

This step compares the lake in the same window every year (January–March, set by `seasonStartMonth` and `seasonMonths`) from 2019 to 2025. Comparing the same season removes the seasonal cycle, so what remains is the change between years. The run may take up to a minute.

```
// ---- STEP 8: Same season, different years ----------------------
// Water area (km2) of the median composite of a collection;
// returns null when there is no image (e.g., cloudy months)
function waterAreaKm2(col) {
  return ee.Algorithms.If(col.size().gt(0),
    col.median().normalizedDifference(['green', 'swir1']).gt(threshold)
      .multiply(ee.Image.pixelArea())
      .reduceRegion({reducer: ee.Reducer.sum(), geometry: roi,
                     scale: pixelSize, maxPixels: 1e13, tileScale: 4})
      .getNumber('nd').divide(1e6),
    null);
}

var years = ee.List.sequence(firstYear, lastYear);
var yearly = ee.FeatureCollection(years.map(function (y) {
  var start = ee.Date.fromYMD(y, seasonStartMonth, 1);
  var col = getImages(start, start.advance(seasonMonths, 'month'));
  return ee.Feature(null, {year: y, n_images: col.size(),
                           area_km2: waterAreaKm2(col)});
}));
print('CP8 | Water area in the same window each year:', yearly);
print(ui.Chart.feature.byFeature(yearly, 'year', 'area_km2')
    .setChartType('LineChart')
    .setOptions({title: lakeName + ' (' + sensor + '): water area by year',
                 hAxis: {title: 'Year', format: '####'},
                 vAxis: {title: 'Water area (km2)'}, pointSize: 5}));
```

CHECKPOINT 8 **Year-to-year change**

Year with the largest area, and its value (km²): _____

Year with the smallest area, and its value (km²): _____

Do any years use very few images (n_images)? _____

On track if: you get a value for every year from 2019 to 2025, each based on at least one image. Changes between years can reflect both water level and vegetation cover. Check the true-colour image of an unusual year to see which one it is.

Step 9. Seasonal cycle within one year

This step calculates the water area for each month of `seasonalYear` (2024). Months with no usable image return no value, so the line has gaps.

```
// ---- STEP 9: Seasonal cycle: month by month in one year ----------
var months = ee.List.sequence(1, 12);
var monthly = ee.FeatureCollection(months.map(function (m) {
  var start = ee.Date.fromYMD(seasonalYear, m, 1);
  var col = getImages(start, start.advance(1, 'month'));
  return ee.Feature(null, {month: m, n_images: col.size(),
                           area_km2: waterAreaKm2(col)});
}));
print('CP9 | Monthly water area in ' + seasonalYear + ':', monthly);
print(ui.Chart.feature.byFeature(monthly, 'month', 'area_km2')
    .setChartType('LineChart')
    .setOptions({title: lakeName + ' (' + sensor + '): monthly water area, ' +
                        seasonalYear,
                 hAxis: {title: 'Month', ticks: [1,2,3,4,5,6,7,8,9,10,11,12]},
                 vAxis: {title: 'Water area (km2)'},
                 pointSize: 5, interpolateNulls: false}));
```

CHECKPOINT 9 **Seasonal cycle**

Month with the largest area, and its value (km²): _____

Month with the smallest area, and its value (km²): _____

Which months have no value (n_images = 0)? _____

On track if: some monsoon months (June–September) have no value, because Mumbai is almost always cloudy then and optical sensors cannot see through clouds. This is expected, not an error. Look for the lowest area before the monsoon (around May) and the highest after it (around October–November). A single month far below its neighbours usually means haze or leftover cloud: check its n_images.

Step 10. Wet vs. dry years

This step uses the lake itself to define wet and dry years: from the Step 8 results, the year with the largest area is the wettest and the year with the smallest area is the driest. It then maps both years and colours each pixel: **dark blue** = water in both years, **light blue** = water only in the wettest year, **orange** = water only in the driest year.

```
// ---- STEP 10: Wet vs. dry years: where does the lake grow? --------
// Uses the yearly areas from Step 8: the wettest year is the year with the
// largest area, the driest year the one with the smallest area.
var valid = yearly.filter(ee.Filter.notNull(['area_km2']));
var wetYear = ee.Number(valid.sort('area_km2', false).first().get('year'));
var dryYear = ee.Number(valid.sort('area_km2').first().get('year'));
print('CP10 | Wettest year:', wetYear);
print('CP10 | Driest year:', dryYear);

// Water map (1 = water, 0 = not water) for the Step 8 window of one year
function waterMap(y) {
  var start = ee.Date.fromYMD(y, seasonStartMonth, 1);
  return getImages(start, start.advance(seasonMonths, 'month')).median()
      .normalizedDifference(['green', 'swir1']).gt(threshold)
      .unmask(0).clip(roi);
}
var wet = waterMap(wetYear);
var dry = waterMap(dryYear);

// 1 = water in both years, 2 = water only in the wettest year,
// 3 = water only in the driest year
var change = wet.and(dry)
    .add(wet.and(dry.not()).multiply(2))
    .add(dry.and(wet.not()).multiply(3))
    .selfMask().rename('class');
Map.addLayer(change, {min: 1, max: 3,
    palette: ['1f4e79', '00c5ff', 'ff7f0e']}, 'Wet vs. dry year');

var byClass = ee.Image.pixelArea().divide(1e6).addBands(change)
    .reduceRegion({reducer: ee.Reducer.sum().group({groupField: 1,
                                                    groupName: 'class'}),
                   geometry: roi, scale: pixelSize,
                   maxPixels: 1e13, tileScale: 4});
print('CP10 | Area (km2) by class: 1 = both years, ' +
      '2 = wettest year only, 3 = driest year only:', byClass.get('groups'));
```

CHECKPOINT 10 **Wet vs. dry years**

Wettest year and its area (km²): _____

Driest year and its area (km²): _____

Area that is water only in the wettest year, class 2 (km²): _____

Where is the light-blue zone (shallow edges, inlets, all around the shore)? _____

On track if: class 1 + class 2 is close to the wettest-year area from Step 8, and class 1 + class 3 is close to the driest-year area. The light-blue zone usually sits on gently sloping shores and inlets, where a small rise in water level floods a large area. A large orange zone usually means vegetation or clouds differed between the two years, not real water gain.

Group results

Fill in one row per lake. Share your row with the other groups so everyone has the full table.

Lake	ROI area (km²)	Images (S2 / L)	Water area, Sentinel-2 (km²)	Water area, Landsat (km²)	% of reported (S2)	Area change −0.1 → 0.2 (%)	Water only in wettest year (km²)
Powai							
Vihar							

Use threshold = 0 for both sensors. S2 = Sentinel-2, L = Landsat.