

Online Resource 2 (ESM_2): Google Earth Engine Script for Multi-Temporal Lake Water Quality Analysis

Author information withheld for double-blind peer review.

In accordance with the double-blind peer review policy of the *International Journal of Environmental Science and Technology*, author names, institutional affiliations, and contact details have been omitted from this supplementary file to prevent identification of the authors during the review process. Full author information will be reinstated upon acceptance.

Caption (Online Resource 2): Google Earth Engine (GEE) JavaScript code for multi-temporal lake water quality monitoring using Landsat 5/8/9 and Sentinel-2 satellite imagery. The script computes chlorophyll-a (three-method ensemble: NDCI polynomial, 2-band NIR/Red ratio, Red/Green log-ratio), Carlson Trophic State Index (TSI), trophic classification, Secchi disk depth, total suspended solids (TSS), turbidity (NTU), NDCI, NDVI, FAI, and MNDWI water mask. Outputs are exported as 13 individual Cloud-Optimised GeoTIFF files and one CSV statistics table to Google Drive. To run: paste into the GEE Code Editor at <https://code.earthengine.google.com>.

```
// =====
// LANDSAT & SENTINEL LAKE ANALYSIS - OPTIMIZED VERSION
// User-friendly | Validated scientific formulas
// International English Version
// =====
// Known Limitations:
// - NDCI on Landsat = (NIR-RED)/(NIR+RED), which is identical to NDVI
//   because Landsat has no red-edge band. For true NDCI, use Sentinel-2
//   and replace NIR with B5 (705 nm red-edge).
// - FAI is a simplified approximation (not standard Stumpf 2012 formula).
// - Turbidity is a linear approximation of Dogliotti 2015.
// =====

print('LAKE WATER QUALITY ANALYSIS SYSTEM');
print('Validated Scientific Formulas');
print('=====');

// =====
// 1. USER INTERFACE
// =====

var panel = ui.Panel({
  style: { width: '350px', padding: '15px', position: 'top-right' }
});

panel.add(ui.Label({
  value: 'LAKE ANALYSIS SYSTEM',
  style: { fontSize: '18px', fontWeight: 'bold', color: '#1976D2', margin: '0 0 5px 0' }
}));
panel.add(ui.Label({
  value: 'Validated Formulas',
  style: { fontSize: '12px', color: '#4CAF50', fontWeight: 'bold', margin: '0 0 15px 0' }
}));

// Step 1: Geometry
panel.add(ui.Label('1. DRAW LAKE BOUNDARY', {fontWeight: 'bold', margin: '10px 0 5px 0'}));
panel.add(ui.Label('Draw your lake polygon on the map', {fontSize: '11px', color: '#666'}));

var geometryStatus = ui.Label({
  value: 'No area drawn yet',
  style: {color: '#FF9800', fontSize: '11px', margin: '5px 0 15px 0'}
});
panel.add(geometryStatus);

// Step 2: Satellite Selection
panel.add(ui.Label('2. SELECT SATELLITE', {fontWeight: 'bold', margin: '10px 0 5px 0'}));
var satelliteSelect = ui.Select({
  items: [
    {label: 'Landsat 5 TM (1984-2012)', value: 'L5'},
    {label: 'Landsat 8 OLI (2013-2021)', value: 'L8'},
    {label: 'Landsat 9 OLI-2 (2022-2025)', value: 'L9'},
    {label: 'Sentinel-2 MSI (2015-2025)', value: 'S2'}
```

```

    ],
    value: 'L8',
    style: {width: '300px'},
    onChange: function(value) { updateYearRange(value); }
  });
panel.add(satelliteSelect);

// Step 3: Year Range
panel.add(ui.Label('3. YEAR RANGE', {fontWeight: 'bold', margin: '15px 0 5px 0'}));

var startYearPanel = ui.Panel({
  layout: ui.Panel.Layout.flow('horizontal'), style: {margin: '5px 0'}
});
var startYearSelect = ui.Select({
  items: generateYears(2013, 2021), value: 2018, style: {width: '80px'}
});
startYearPanel.add(ui.Label('Start:', {fontSize: '11px', width: '70px'}));
startYearPanel.add(startYearSelect);
panel.add(startYearPanel);

var endYearPanel = ui.Panel({
  layout: ui.Panel.Layout.flow('horizontal'), style: {margin: '5px 0'}
});
var endYearSelect = ui.Select({
  items: generateYears(2013, 2021), value: 2021, style: {width: '80px'}
});
endYearPanel.add(ui.Label('End:', {fontSize: '11px', width: '70px'}));
endYearPanel.add(endYearSelect);
panel.add(endYearPanel);

// Step 4: Season Selection
panel.add(ui.Label('4. SEASON SELECTION', {fontWeight: 'bold', margin: '15px 0 5px 0'}));
var seasonSelect = ui.Select({
  items: [
    {label: 'Spring (March-May)', value: 'spring'},
    {label: 'Summer (June-August)', value: 'summer'},
    {label: 'Autumn (September-November)', value: 'fall'},
    {label: 'Winter (December-February)', value: 'winter'}
  ],
  value: 'summer',
  style: {width: '300px'}
});
panel.add(seasonSelect);

panel.add(ui.Label(' ', {margin: '10px 0 5px 0'}));

var analyzeButton = ui.Button({
  label: 'RUN ANALYSIS',
  style: {
    width: '300px', padding: '10px',
    backgroundColor: '#2196F3', color: 'white', fontWeight: 'bold'
  },
  onClick: runAnalysis
});
panel.add(analyzeButton);
panel.add(ui.Label(' ', {margin: '10px 0'}));

var resultPanel = ui.Panel({
  style: {backgroundColor: '#f5f5f5', padding: '10px', border: '1px solid #ddd'}
});
panel.add(resultPanel);

var exportButton = ui.Button({
  label: 'EXPORT SEPARATE GeoTIFFs (13 Files)',
  style: {
    width: '300px', padding: '10px',
    backgroundColor: '#4CAF50', color: 'white', fontWeight: 'bold', shown: false
  },
  onClick: exportResults
});
panel.add(exportButton);
ui.root.add(panel);

var analysisResult = null;
// =====

```

```

// 2. HELPER FUNCTIONS
// =====

function generateYears(start, end) {
  var years = [];
  for (var i = start; i <= end; i++) {
    years.push({label: String(i), value: i});
  }
  return years;
}

function updateYearRange(satellite) {
  var ranges = {
    'L5': {min: 1984, max: 2012},
    'L8': {min: 2013, max: 2021},
    'L9': {min: 2022, max: 2025},
    'S2': {min: 2015, max: 2025}
  };
  var r = ranges[satellite];
  startYearSelect.items().reset(generateYears(r.min, r.max));
  endYearSelect.items().reset(generateYears(r.min, r.max));
  startYearSelect.setValue(r.min);
  endYearSelect.setValue(r.max);
}

function getSeasonDates(season, year) {
  var dates = {
    'spring': {start: year + '-03-01', end: year + '-05-31'},
    'summer': {start: year + '-06-01', end: year + '-08-31'},
    'fall': {start: year + '-09-01', end: year + '-11-30'},
    'winter': {start: year + '-12-01', end: (year + 1) + '-02-28'}
  };
  return dates[season];
}

// =====
// 3. LANDSAT DATA PROCESSING
// =====

function loadLandsatData(satellite, startYear, endYear, season, geometry) {
  var collections = {
    'L5': 'LANDSAT/LT05/C02/T1_L2',
    'L8': 'LANDSAT/LC08/C02/T1_L2',
    'L9': 'LANDSAT/LC09/C02/T1_L2'
  };
  var bandMaps = {
    'L5': {B:'SR_B1', G:'SR_B2', R:'SR_B3', N:'SR_B4', S1:'SR_B5', S2:'SR_B7'},
    'L8': {B:'SR_B2', G:'SR_B3', R:'SR_B4', N:'SR_B5', S1:'SR_B6', S2:'SR_B7'},
    'L9': {B:'SR_B2', G:'SR_B3', R:'SR_B4', N:'SR_B5', S1:'SR_B6', S2:'SR_B7'}
  };
  var bands = bandMaps[satellite];
  var images = [];

  for (var year = startYear; year <= endYear; year++) {
    var dates = getSeasonDates(season, year);
    var col = ee.ImageCollection(collections[satellite])
      .filterBounds(geometry)
      .filterDate(dates.start, dates.end)
      .filter(ee.Filter.lt('CLOUD_COVER', 30));

    if (col.size().getInfo() > 0) {
      var composite = col.map(function(img) {
        // QA_PIXEL masking: cloud shadow (bit 3), snow/ice (bit 4), cloud (bit 5)
        var qa = img.select('QA_PIXEL');
        var mask = qa.bitwiseAnd(1 << 3).eq(0)
          .and(qa.bitwiseAnd(1 << 4).eq(0))
          .and(qa.bitwiseAnd(1 << 5).eq(0));
        var selected = img.select([bands.B, bands.G, bands.R, bands.N, bands.S1, bands.S2])
          .rename(['BLUE', 'GREEN', 'RED', 'NIR', 'SWIR1', 'SWIR2']);
        // Landsat Collection 2 Level-2 scale factors
        var scaled = selected.multiply(0.0000275).add(-0.2).clamp(0, 1);
        return scaled.updateMask(mask);
      }).median();
      images.push(composite);
    }
  }
}

```

```

}
if (images.length === 0) return null;
return ee.ImageCollection(images).mean().clip(geometry);
}

// =====
// 4. SENTINEL-2 DATA PROCESSING
// =====

function loadSentinel2Data(startYear, endYear, season, geometry) {
  var images = [];
  for (var year = startYear; year <= endYear; year++) {
    var dates = getSeasonDates(season, year);
    var col = ee.ImageCollection('COPERNICUS/S2_SR_HARMONIZED')
      .filterBounds(geometry)
      .filterDate(dates.start, dates.end)
      .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 30));

    if (col.size().getInfo() > 0) {
      var composite = col.map(function(img) {
        // QA60 masking: opaque clouds (bit 10), cirrus (bit 11)
        var qa = img.select('QA60');
        var cloudMask = qa.bitwiseAnd(1 << 10).eq(0)
          .and(qa.bitwiseAnd(1 << 11).eq(0));
        // Note: B5 (red-edge, 705 nm) not loaded here.
        // For true NDCI, load B5 separately and substitute NIR below.
        var selected = img.select(['B2', 'B3', 'B4', 'B8', 'B11', 'B12'])
          .rename(['BLUE', 'GREEN', 'RED', 'NIR', 'SWIR1', 'SWIR2']);
        var scaled = selected.multiply(0.0001).clamp(0, 1);
        return scaled.updateMask(cloudMask);
      }).median();
      images.push(composite);
    }
  }
  if (images.length === 0) return null;
  return ee.ImageCollection(images).mean().clip(geometry);
}

// =====
// 5. WATER MASK
// =====

function applyWaterMask(image) {
  // MNDWI = (Green - SWIR1) / (Green + SWIR1) [Xu 2006]
  var mndwi = image.normalizedDifference(['GREEN', 'SWIR1']).rename('MNDWI');
  var waterMask = mndwi.gt(0.1);
  return image.updateMask(waterMask).addBands(mndwi);
}

// =====
// 6. INDEX CALCULATION
// =====

function calculateIndices(image) {
  var GREEN = image.select('GREEN');
  var RED = image.select('RED');
  var NIR = image.select('NIR');

  // NDCI [Mishra & Mishra 2012]
  // Warning: identical to NDVI on Landsat (no red-edge band)
  var NDCI = NIR.subtract(RED)
    .divide(NIR.add(RED).add(0.0001))
    .rename('NDCI');

  // Chlorophyll-a Method 1: NDCI Polynomial [Mishra & Mishra 2012]
  // ChlA = 194.325*NDCI^2 + 86.115*NDCI + 14.039
  var chl_NDCI = NDCI.pow(2).multiply(194.325)
    .add(NDCI.multiply(86.115))
    .add(14.039)
    .clamp(0.5, 100)
    .rename('ChlA_NDCI');

  // Chlorophyll-a Method 2: 2-Band NIR/Red [Gitelson et al. 2008]
  // ChlA = 23.1*(NIR/Red) - 16.4
  var NIR_Red_Ratio = NIR.divide(RED.add(0.0001));

```

```

var chl_2Band = NIR_Red_Ratio.multiply(23.1)
                        .subtract(16.4)
                        .clamp(0.5, 200)
                        .rename('ChlA_2Band');

// Chlorophyll-a Method 3: Red/Green Log-Ratio [Moses et al. 2012]
// ln(ChlA) = 1.024*ln(Red/Green) - 0.677
var Red_Green_Ratio = RED.divide(GREEN.add(0.0001));
var chl_Moses = Red_Green_Ratio.log()
                        .multiply(1.024)
                        .subtract(0.677)
                        .exp()
                        .clamp(0.5, 50)
                        .rename('ChlA_Moses');

// Ensemble mean of three methods
var ChlA_Final = chl_NDCI.add(chl_2Band).add(chl_Moses)
                        .divide(3)
                        .clamp(0.5, 100)
                        .rename('ChlA_Final');

// Carlson Trophic State Index [Carlson 1977]
// TSI(Chl) = 9.81*ln(Chl) + 30.6
var TSI = ChlA_Final.log().multiply(9.81).add(30.6).rename('TSI_Carlson');

var Trophic = ee.Image(1)
    .where(TSI.gte(40).and(TSI.lt(50)), 2)
    .where(TSI.gte(50).and(TSI.lt(70)), 3)
    .where(TSI.gte(70), 4)
    .rename('Trophic_Level');
// 1=Oligotrophic (<40), 2=Mesotrophic (40-50),
// 3=Eutrophic (50-70), 4=Hypereutrophic (>=70)

// Secchi Disk Depth [Kutser et al. 2005]
// Secchi = 7.6 * ChlA^(-0.64)
var Secchi = ChlA_Final.pow(-0.64)
                        .multiply(7.6)
                        .clamp(0.1, 15)
                        .rename('Secchi_m');

// Total Suspended Solids [Nechad et al. 2010]
// TSS = A*Rrs_red / (1 - Rrs_red/C) A=289.29, C=0.1686
var A_TSS = 289.29;
var C_TSS = 0.1686;
var TSS = RED.multiply(A_TSS)
    .divide(ee.Image(1).subtract(RED.divide(C_TSS)).add(0.0001))
    .clamp(0, 150)
    .rename('TSS_mg_L');

// Turbidity: linear approximation [Dogliotti et al. 2015, simplified]
// Turb = 45.0*Red + 10.0*Green
var Turbidity = RED.multiply(45.0)
    .add(GREEN.multiply(10.0))
    .clamp(0, 100)
    .rename('Turbidity_NTU');

// NDVI [Rouse et al. 1974]
var NDVI = NIR.subtract(RED)
    .divide(NIR.add(RED).add(0.0001))
    .clamp(-1, 1)
    .rename('NDVI');

// FAI - Floating Algae Index: simplified proxy [Hu 2009]
// Standard FAI uses wavelength-weighted NIR baseline with Red + SWIR
var FAI = NIR.subtract(RED.add(GREEN).divide(2)).rename('FAI');

return image
    .addBands(NDCI).addBands(chl_NDCI).addBands(chl_2Band)
    .addBands(chl_Moses).addBands(ChlA_Final).addBands(TSI)
    .addBands(Trophic).addBands(Secchi).addBands(TSS)
    .addBands(Turbidity).addBands(NDVI).addBands(FAI);
}

// =====
// 7. STATISTICS

```

```

// =====

function calculateStats(image, geometry) {
  return image.select([
    'ChlA_Final', 'TSI_Carlson', 'Trophic_Level',
    'Secchi_m', 'TSS_mg_L', 'Turbidity_NTU', 'NDCI', 'NDVI'
  ]).reduceRegion({
    reducer: ee.Reducer.mean()
      .combine(ee.Reducer.stdDev(), '', true)
      .combine(ee.Reducer.min(), '', true)
      .combine(ee.Reducer.max(), '', true)
      .combine(ee.Reducer.percentile([50]), '', true),
    geometry: geometry,
    scale: 30,
    maxPixels: 1e9,
    bestEffort: true,
    tileScale: 2
  });
}

// =====
// 8. MAIN ANALYSIS
// =====

function runAnalysis() {
  if (!geometry) {
    resultPanel.clear();
    resultPanel.add(ui.Label('Please draw the lake boundary first!', {color: '#f44336'}));
    return;
  }
  geometryStatus.setValue('Area selected');
  geometryStatus.style().set('color', '#4CAF50');
  resultPanel.clear();
  resultPanel.add(ui.Label('Running analysis...', {color: '#2196F3', fontWeight: 'bold'}));

  var satellite = satelliteSelect.getValue();
  var startYear = startYearSelect.getValue();
  var endYear = endYearSelect.getValue();
  var season = seasonSelect.getValue();

  print('=====');
  print('ANALYSIS STARTED');
  print('Satellite : ' + satellite);
  print('Year range: ' + startYear + '-' + endYear);
  print('Season : ' + season);
  print('=====');

  var composite = (satellite === 'S2')
    ? loadSentinel2Data(startYear, endYear, season, geometry)
    : loadLandsatData(satellite, startYear, endYear, season, geometry);

  if (!composite) {
    resultPanel.clear();
    resultPanel.add(ui.Label('No imagery found for the selected period.', {color: '#f44336'}));
    resultPanel.add(ui.Label('Try a different year range or season.', {fontSize: '10px'}));
    return;
  }

  var withWater = applyWaterMask(composite);
  var withIndices = calculateIndices(withWater);
  var stats = calculateStats(withIndices, geometry);

  stats.evaluate(function(result) {
    displayResults(result, satellite, startYear, endYear, season);
    updateMap(withIndices, geometry);
    analysisResult = {
      image: withIndices, stats: result,
      satellite: satellite, startYear: startYear,
      endYear: endYear, season: season
    };
    exportButton.style().set('shown', true);
    print('ANALYSIS COMPLETE');
  });
}

// =====

```

```
// 9. DISPLAY RESULTS
// =====

function displayResults(stats, satellite, startYear, endYear, season) {
  resultPanel.clear();
  resultPanel.add(ui.Label('ANALYSIS COMPLETE',
    {fontWeight:'bold', color:'#4CAF50', fontSize:'13px', margin:'0 0 10px 0'}));
  resultPanel.add(ui.Label(satellite+' | '+startYear+'-'+endYear+' | '+season,
    {fontSize:'10px', color:'#666', margin:'0 0 10px 0'}));

  var chl = stats.ChlA_Final_mean || 0;
  resultPanel.add(ui.Label('CHLOROPHYLL-A', {fontWeight:'bold', fontSize:'11px'}));
  resultPanel.add(ui.Label('Mean: '+chl.toFixed(2)+' ug/L', {fontSize:'11px'}));
  resultPanel.add(ui.Label('Min-Max: '+stats.ChlA_Final_min||0).toFixed(1)+
    ' - '+stats.ChlA_Final_max||0).toFixed(1), {fontSize:'10px', color:'#666'}));

  var tsi = stats.TSI_Carlson_mean || 0;
  var trophicLabel = tsi < 40 ? 'Oligotrophic (Clean)' :
    tsi < 50 ? 'Mesotrophic (Moderate)' :
    tsi < 70 ? 'Eutrophic (Nutrient-rich)' :
    'Hypereutrophic (Excess Nutrients)';
  resultPanel.add(ui.Label(' '));
  resultPanel.add(ui.Label('TROPHIC STATUS (TSI)', {fontWeight:'bold', fontSize:'11px'}));
  resultPanel.add(ui.Label('TSI: '+tsi.toFixed(1), {fontSize:'11px'}));
  resultPanel.add(ui.Label('Status: '+trophicLabel, {fontSize:'11px', color:'#2196F3'}));

  var secchi = stats.Secchi_m_mean || 0;
  resultPanel.add(ui.Label(' '));
  resultPanel.add(ui.Label('WATER CLARITY', {fontWeight:'bold', fontSize:'11px'}));
  resultPanel.add(ui.Label('Secchi Depth: '+secchi.toFixed(2)+' m', {fontSize:'11px'}));

  var tss = stats.TSS_mg_L_mean || 0;
  resultPanel.add(ui.Label(' '));
  resultPanel.add(ui.Label('TOTAL SUSPENDED SOLIDS', {fontWeight:'bold', fontSize:'11px'}));
  resultPanel.add(ui.Label('TSS: '+tss.toFixed(2)+' mg/L', {fontSize:'11px'}));

  var turb = stats.Turbidity_NTU_mean || 0;
  resultPanel.add(ui.Label(' '));
  resultPanel.add(ui.Label('TURBIDITY', {fontWeight:'bold', fontSize:'11px'}));
  resultPanel.add(ui.Label('Turbidity: '+turb.toFixed(2)+' NTU', {fontSize:'11px'}));

  resultPanel.add(ui.Label(' ', {margin:'10px 0'}));
  resultPanel.add(ui.Label('Mishra 2012 | Gitelson 2008 | Carlson 1977 | Nechad 2010',
    {fontSize:'9px', color:'#999', fontStyle:'italic'}));
}

// =====
// 10. MAP VISUALIZATION
// =====

function updateMap(image, geometry) {
  Map.layers().reset();
  Map.centerObject(geometry, 11);
  Map.addLayer(ee.Image().paint(geometry, 0, 2), {palette:['yellow']},
    'Lake Boundary', true);
  Map.addLayer(image.select('ChlA_Final'), {
    min: 0, max: 30,
    palette: ['#000080', '#0000FF', '#00FFFF', '#00FF00', '#FFFF00', '#FF7F00', '#FF0000']
  }, 'Chlorophyll-a (ug/L)', true);
  Map.addLayer(image.select('TSI_Carlson'), {
    min: 30, max: 70,
    palette: ['#0000FF', '#00FFFF', '#00FF00', '#FFFF00', '#FF0000']
  }, 'TSI (Trophic)', false);
  Map.addLayer(image.select('Secchi_m'), {
    min: 0.5, max: 5,
    palette: ['#8B0000', '#FF0000', '#FFA500', '#FFFF00', '#00FF00', '#0000FF']
  }, 'Secchi Depth (m)', false);
  addLegend();
}

function addLegend() {
  var legend = ui.Panel({
    style: {position:'bottom-left', padding:'8px 15px', backgroundColor:'white'}
  });
  legend.add(ui.Label('Chlorophyll-a (ug/L)', {fontWeight:'bold', fontSize:'12px'}));
}
```

```

var colors = ['#000080', '#0000FF', '#00FFFF', '#00FF00', '#FFFF00', '#FF7F00', '#FF0000'];
var labels = ['0', '5', '10', '15', '20', '25', '30'];
for (var i = 0; i < colors.length; i++) {
    legend.add(ui.Panel({
        widgets: [
            ui.Label({style:{backgroundColor:colors[i], padding:'8px', margin:'0 4px 0 0'}}),
            ui.Label(labels[i], {fontSize:'10px'})
        ],
        layout: ui.Panel.Layout.Flow('horizontal'),
        style: {margin:'2px 0'}
    }));
}
Map.add(legend);
}

// =====
// 11. EXPORT - SEPARATE GeoTIFF PER PARAMETER
// =====

function exportResults() {
    if (!analysisResult) {
        print('No analysis result available. Run analysis first.');
```

return;

```
    }
    var base = 'Lake_' + analysisResult.satellite + '_' +
        analysisResult.startYear + '_' + analysisResult.endYear +
        '_' + analysisResult.season;
    var folder = 'Lake_Water_Quality';

    print('=====');
    print('STARTING SEPARATE GeoTIFF EXPORTS');
    print('=====');
```

var parameters = [		
{band:'ChlA_Final',	name:'Chlorophyll_a_Ensemble',	desc:'Chl-a ensemble mean (ug/L)'}
{band:'ChlA_NDCI',	name:'Chlorophyll_a_NDCI',	desc:'Chl-a NDCI polynomial (ug/L)'}
{band:'ChlA_2Band',	name:'Chlorophyll_a_2Band',	desc:'Chl-a 2-band NIR/Red (ug/L)'}
{band:'ChlA_Moses',	name:'Chlorophyll_a_Moses',	desc:'Chl-a Red/Green ratio (ug/L)'}
{band:'TSI_Carlson',	name:'TSI_Carlson',	desc:'Carlson Trophic State Index'}
{band:'Trophic_Level',	name:'Trophic_Level',	desc:'Trophic level (1-4)'}
{band:'Secchi_m',	name:'Secchi_Depth',	desc:'Secchi disk depth (m)'}
{band:'TSS_mg_L',	name:'TSS',	desc:'Total Suspended Solids (mg/L)'}
{band:'Turbidity_NTU',	name:'Turbidity',	desc:'Turbidity (NTU)'}
{band:'NDCI',	name:'NDCI',	desc:'Norm. Diff. Chlorophyll Index'}
{band:'NDVI',	name:'NDVI',	desc:'Norm. Diff. Vegetation Index'}
{band:'FAI',	name:'FAI',	desc:'Floating Algae Index (simplified)'}
{band:'MNDWI',	name:'MNDWI',	desc:'Mod. Norm. Diff. Water Index'}
];		

```

    var scale = (analysisResult.satellite === 'S2') ? 10 : 30;

    parameters.forEach(function(param) {
        analysisResult.image.bandNames().evaluate(function(names) {
            if (names.indexOf(param.band) !== -1) {
                var fileName = base + '_' + param.name;
                Export.image.toDrive({
                    image: analysisResult.image.select([param.band]).float(),
                    description: fileName,
                    folder: folder,
                    fileNamePrefix: fileName,
                    scale: scale,
                    region: geometry,
                    maxPixels: 1e9,
                    fileFormat: 'GeoTIFF',
                    formatOptions: {cloudOptimized: true}
                });
                print(' OK: ' + param.name + ' (' + scale + 'm) - ' + param.desc);
            } else {
                print(' SKIPPED (band not found): ' + param.band);
            }
        });
    });

    print('=====');
    print(parameters.length + ' GeoTIFF EXPORTS QUEUED');
```

```

print('Drive folder : ' + folder);
print('Resolution   : ' + scale + ' m');
print('Start exports from the Tasks panel');
print('=====');

Export.table.toDrive({
  collection: ee.FeatureCollection([ee.Feature(null, analysisResult.stats)]),
  description: base + '_Statistics',
  folder: folder,
  fileNamePrefix: base + '_Statistics',
  fileFormat: 'CSV'
});
print('Statistics CSV export queued');
}

// =====
// 12. INITIALIZATION
// =====

Map.setCenter(33, 40, 6);

print('SYSTEM READY');
print('Step 1: Draw the lake boundary on the map');
print('Step 2: Select satellite');
print('Step 3: Set year range');
print('Step 4: Select season');
print('Step 5: Run analysis');
print('=====');
print('FORMULAS USED:');
print('Mishra & Mishra (2012) - NDCI polynomial (ChlA)');
print('Gitelson et al. (2008) - 2-band NIR/Red ratio (ChlA)');
print('Moses et al. (2012) - Red/Green log-ratio (ChlA)');
print('Carlson (1977) - TSI trophic classification');
print('Nechad et al. (2010) - TSS');
print('Xu (2006) - MNDWI water mask');
print('Dogliotti et al. (2015) - Turbidity (simplified)');
print('=====');
print('NOTE: On Landsat sensors, NDCI = NDVI (no red-edge band).');
print('For true NDCI, select Sentinel-2 and load B5 (705 nm).');

```