Internship Project Report

on

P4 - Spatiotemporal Analysis of Urban Expansion Using NDBI from Satellite Imagery

Submitted by

Kishan Raj

B.Tech Biotechnology Student

N M A M Institute of Technology, Nitte, Karkala, 574110

Nitte University, Mangalore

Submitted on

16 August 2025

Table of Contents

Spatiotemporal Analysis of Urban Expansion Using NDBI from Satellite Imagery

1. Objective

To analyse, quantify and visualize urban expansion in Mysore Central between the years 2000 and 2024 comparing expansions of both the years using Normalized Difference Built-up Index (NDBI) by generating raster and vector maps of built-up areas, calculating their spatial extent, identifying new expansions by building polygons and exporting results to google drive for further Geographic information system (GIS) analysis, where developed custom User Interface (UI) instead of showing in console it automatically creates a panel that mainly shows NDBI values and Area Statistics.

With the help of Google Earth Engine (GEE) developed an urban expansion using JavaScript which basically uses cloud computing to complete the tasks like adding layers, generating contents and upload to Google Drive. To cross check the results used QGIS free Open-Source Platform.

Using GEE platform, compared the urban expansions of 2000 and 2024 year by checking the green spots, decrease of water reservoirs or lakes, deforestation, excavation done as the population, value of the land and economic growth has increased. [1], [2]

2. Study Area

2.1. Location:

Mysore (also known as Mysuru), a prominent city in Karnataka, India where my focus was only on urban expansion analysis. It's a city prominent for its rich heritage and historical significance.

Often referred to as the “City of Palaces” and the “Cultural Capital of Karnataka,” Mysore is a major global centre for tourism and yoga. While it has a growing IT sector, its identity is deeply rooted in its magnificent palaces, vibrant festivals and traditional arts.

2.2. Geographical Extent (AOI):

The Area of Interest (AOI) used in the analysis is defined by the following co-ordinates (76.575, 12.225, 76.725, 12.375) where in generally classified as below:

1) **Eastern Longitude: 76.725° E**
2) **Western Longitude: 76.575° E**
3) **Northern Latitude: 12.375° N**
4) **Southern Latitude: 12.225° N**

These co-ordinates create a bounding box that fully covers the Mysore Central District areas as well as peri-urban areas that shows huge significant expansion year 2000 and 2024. By this way we can compare the urban expansions followed by the years. [3]

2.3. Reason of Selection:

The selection of Mysore for this urban expansion study is based on its unique position as a city undergoing transformation and retaining historical and economic roots.

Mysore is not just a city of historical significance, once serving as the capital of the state of Karnataka, but also a powerful and evolving economic engine. It has a long-standing legacy as a manufacturing hub, particularly for its world-renowned Mysore silk and sandalwood industries. In recent decades, it has successfully diversified as industrial base to become a major centre for the automotive industry and huge boost in growing IT sector across India.

The key highlights of historical as a main backbone and with rapid industrialization makes Mysore a complete case study for urban growth. The period between 2000 and 2024 was specifically chosen to analyse and quantify this modern growth. [3]

2.4. Map of Study Area (AOI):

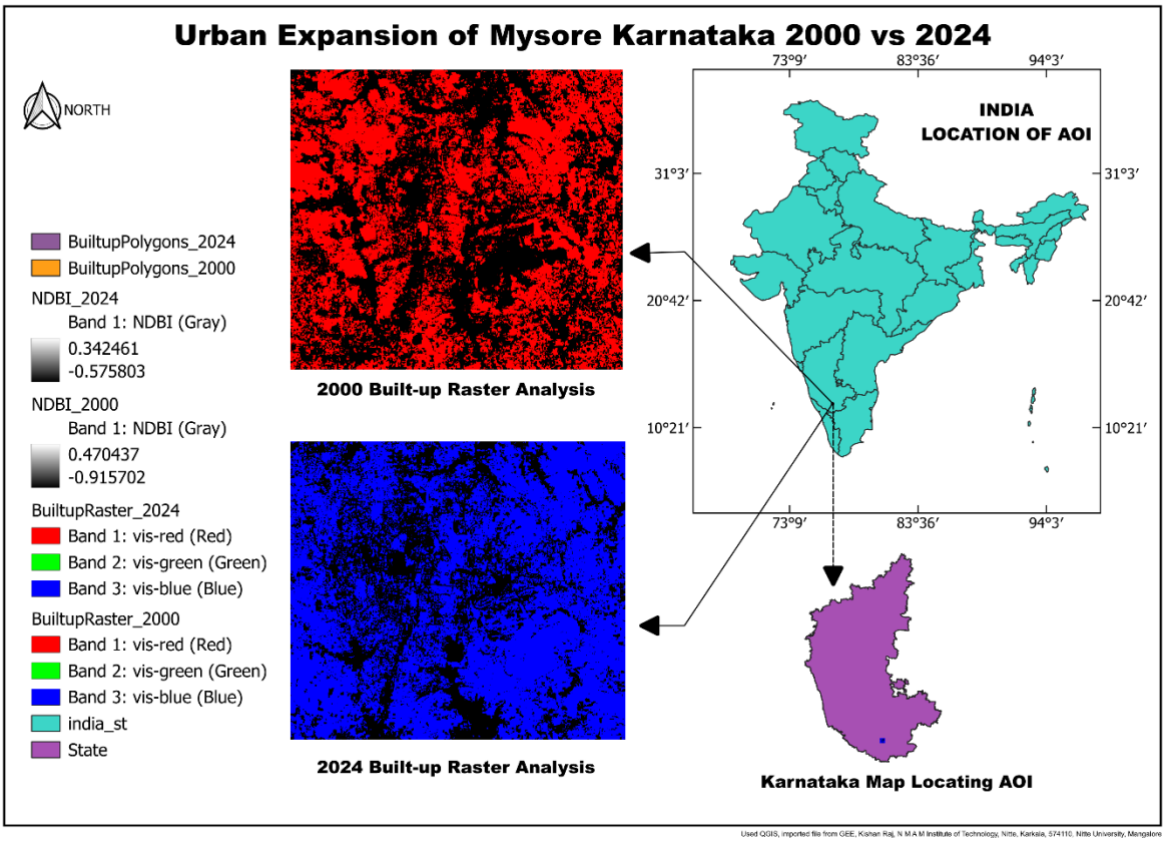


Fig: 0 Image Representing the Area of Interest
under Built-up Raster Analysis of 2000 and 2024

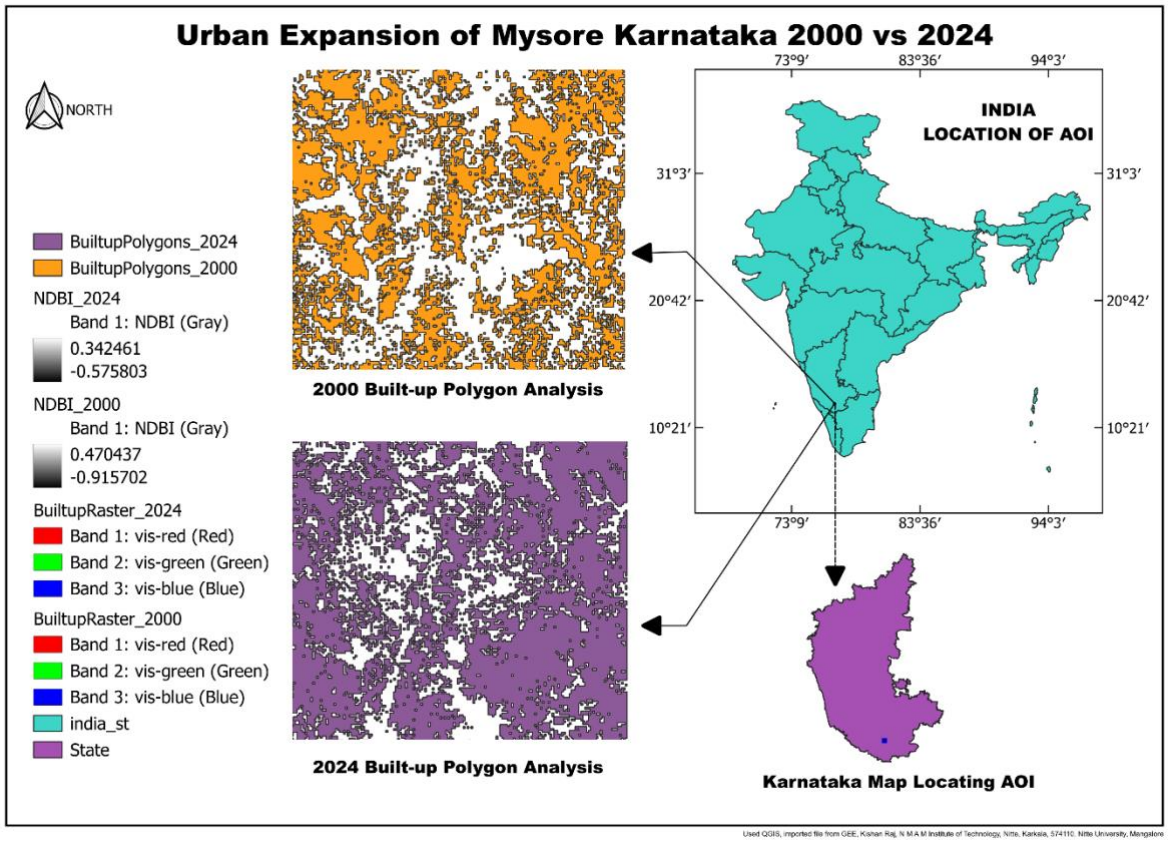


Fig: 1 Image Representing the Area of Interest
under Built-up Polygon Analysis of 2000 and 2024

3. Data Used

I used mainly Google Earth Engine (Engine) for all analysis which utilizes satellite imagery from the Landsat program which is managed by NASA and U.S. Geological Survey (USGS). As reference material provided by course co-ordinator, they used Landsat data with level 2 products, where I also used the same but is more effective in analysing which provide atmospherically corrected surface reflectance data which makes ideal for 2000 and 2024 both the years which is easy and helpful for analysing. [2], [4]

Cross verified using QGIS Platform of the analysis.

Below table is the summary of the data-used for both 2000 and 2024 years.

3.1. Summary of Data-Used:

Attribute	Year 2000 Analysis	Year 2024 Analysis
Satellite & Sensor	Landsat 5 Thematic Mapper (TM)	Landsat 8 OLI/TIRS
GEE Collection ID	LANDSAT/LT05/C02/T1_L2	LANDSAT/LC08/C02/T1_L2
Key Bands Used	Band 4 (NIR) Band 5 (SWIR1) QA_PIXEL	Band 5 (NIR) Band 6 (SWIR1) QA_PIXEL

3.2. Details on the Data-Used:

1. **For Year 2000: Landsat 5 TM**
   - **Satellite & Sensor:** Landsat 5 Thematic Mapper (TM)
   - **GEE Image Collection ID:** “LANDSAT/LT05/C02/T1_L2”
   - **Key Bands Used:**
     - **Band 4 (Near-Infrared - NIR):** Essential for identifying vegetation and water bodies.
     - **Band 5 (Shortwave Infrared 1 - SWIR1):** Highly sensitive to moisture content and soil properties which makes effective for distinguishing built-up areas.
     - **QA_PIXEL Band:** Used for quality assessment to mask out clouds and cloud shadows, ensuring data clarity.
2. **For Year 2024: Landsat 8 OLI/TIRS**
   - **Satellite & Sensor:** Landsat 8 Operational Land Imager (OLI) and Thermal Infrared Sensor (TIRS)
   - **GEE Image Collection ID:** “LANDSAT/LC08/C02/T1_L2”
   - **Key Bands Used:**

- **Band 5 (Near-Infrared - NIR):** Same functionality as Landsat 5's Band 4.
- **Band 6 (Shortwave Infrared 1 - SWIR1):** Same functionality as Landsat 5's Band 5.
- **QA_PIXEL Band:** Used for masking clouds, cloud shadows and other undesirable pixels. [4]

4. Methodology

This study used a combined Google Earth Engine (GEE) and QGIS approach to detect, map and quantify urban expansion in Mysore Central between the years 2000 and 2024.

The process was carried out in two main workflows:

1. **GEE-based Workflow:** Automatically calculates Normalized Difference Built-up Index (NDBI), raster and vector (polygon) calculation and generation, area computation and analysis, custom User Interface (UI) to give statistics.
2. **QGIS-based Workflow:** Used for cross verification of generated data from GEE to QGIS, Map layout representing the AOI, Mysore Urban Expansion, Chart Print Layout.

4.1. Google Earth Engine (GEE) Based Workflow:

The Google Earth Engine platform was used to get benefits of cloud-based processing and Landsat satellite imagery for spatiotemporal analysis.

***** The JS Code is provided for every step but referring to this code might give indentation error due to comments, Microsoft Word platform. GitHub Repository created only for this to avoid any errors. Provided in Reference Section. ** (1)***

Step 1: Define Area of Interest (AOI), Time Periods, NDBI Thresholds and Visualization Parameters

AOI defined as a bounding box using coordinates: (76.575, 12.225, 76.725, 12.375) which covers Mysore Central and peri-urban areas.

Selecting the years for analysis, NDBI threshold, Visualization Parameters. **NDBI threshold** which is Normalized Difference Built-up Index (NDBI) is a formula that uses satellite imagery to calculate a value for each pixel, typically ranging from “-1” to “+1”. Here “0” sets the decision point, it tells the script if a pixel's NDBI value is greater than “0” classify it as a 'built-up' area. If it's “0” or less then treat it as 'not built-up' area.

Visualization parameters (“*VisParams*”) are used to display; they don't change the data itself but control how it looks on the map. Each variable defines a colour scheme for a different layer in analysis as given in the code below. [1], [2], [4], [5]

```
// Step 01: AOI, Time Periods, NDBI Thresholds and Visulization Parameters
var aoi = ee.Geometry.Rectangle([76.575, 12.225, 76.725, 12.375]);

// Selecting the years for analysis
var year2000 = '2000';
var year2024 = '2024';

// NDBI threshold to classify built-up areas
var NDBI_THRESHOLD = 0;

// Visualization parameters for NDBI and built-up areas
var ndbiVisParams = { min: -1, max: 1, palette: ['blue', 'white', 'brown'] };
var builtupVisParams2000 = { min: 0, max: 1, palette: ['white', 'red'] };
var builtupVisParams2024 = { min: 0, max: 1, palette: ['white', 'blue'] };
var expansionVisParams = { min: -1, max: 1, palette: ['gray', 'green',
'purple'] };
```

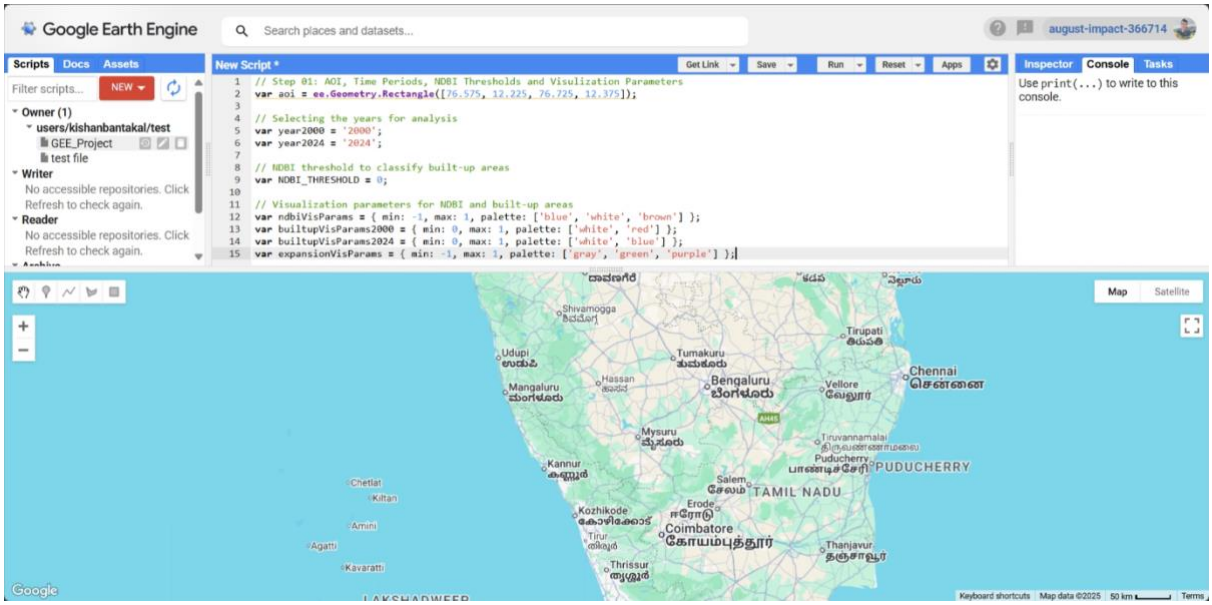


Fig: 2 Image representing AOI, Time Period and Visualization Parameters

Step 02: Data acquisition, NDBI Calculation, data pre-processing using Landsat Data

For year 2000 “Landsat 5 TM (LANDSAT/LT05/C02/T1_L2)” data were used where it consists of mainly older Landsat data before 2010. For year 2024 “Landsat 8 OLI/TIRS (LANDSAT/LC08/C02/T1_L2)” where it mainly consists of recent data after 2010 as per the analysis. Cloud Masking and cover in the JavaScript code is maximum upto only “10”.

For NDBI Calculation applied surface reflectance scaling factors.

Cloud masking using QA_PIXEL band (different bit masks for Landsat 5 & 8).

- **Landsat 5: SWIR1 = Band 5, NIR = Band 4**
- **Landsat 8: SWIR1 = Band 6, NIR = Band 5**

$$NDBI = \frac{SWIR1 - NIR}{SWIR1 + NIR}$$

From the above calculation developed the raster analysis for 2000 and 2024 and using the below given code only the calculation part was done.

```
// Step 02: NDBI Calculation using Landsat Data
function calculateNDBI(year, aoi) {
    var startDate = ee.Date.fromYMD(parseInt(year, 10), 1, 1);
    var endDate = startDate.advance(1, 'year');
    var collection;
    // Landsat 5 for 2000
    if (parseInt(year, 10) < 2012) {
        collection = ee.ImageCollection('LANDSAT/LT05/C02/T1_L2')
            .filterBounds(aoi)
            .filterDate(startDate, endDate)
            .filter(ee.Filter.lt('CLOUD_COVER', 10))
            .map(function (image) {
                // Calculate NDBI using SWIR1 (B5) and NIR (B4) for Landsat 5
                var opticalBands =
image.select('SR_B.').multiply(0.0000275).add(-0.2);
                var ndbi = opticalBands.normalizedDifference(['SR_B5',
'SR_B4']).rename('NDBI');
                return
opticalBands.addBands(ndbi).addBands(image.select('QA_PIXEL'));
            });
    } else {
        // Landsat 8 for 2024
        collection = ee.ImageCollection('LANDSAT/LC08/C02/T1_L2')
            .filterBounds(aoi)
            .filterDate(startDate, endDate)
            .filter(ee.Filter.lt('CLOUD_COVER', 10))
            .map(function (image) {
                // Calculate NDBI using SWIR1 (B6) and NIR (B5) for Landsat 8
                var opticalBands =
image.select('SR_B.').multiply(0.0000275).add(-0.2);
                var ndbi = opticalBands.normalizedDifference(['SR_B6',
'SR_B5']).rename('NDBI');
                return
opticalBands.addBands(ndbi).addBands(image.select('QA_PIXEL'));
            });
    }
    // Cloud masking function for Landsat 5 and Landsat 8
    function maskClouds(image) {
        var qa = image.select('QA_PIXEL');
        if (parseInt(year, 10) < 2012) {
            // Landsat 5 uses QA_PIXEL for cloud and cloud shadow (bits 3 and
5)
            var cloudMask = qa.bitwiseAnd(1 << 3).or(qa.bitwiseAnd(1 << 5));
            return image.updateMask(cloudMask.not());
        } else {
            // Landsat 8 uses QA_PIXEL for cloud, cloud shadow and snow (bits
3, 4, 5)
```

```
            var cloudMask = qa.bitwiseAnd(1 << 3).or(qa.bitwiseAnd(1 <<
4)).or(qa.bitwiseAnd(1 << 5));
            return image.updateMask(cloudMask.not());
        }
    }
    var composite = collection.map(maskClouds).median().clip(aoi);
    return composite.select('NDBI');
}
```

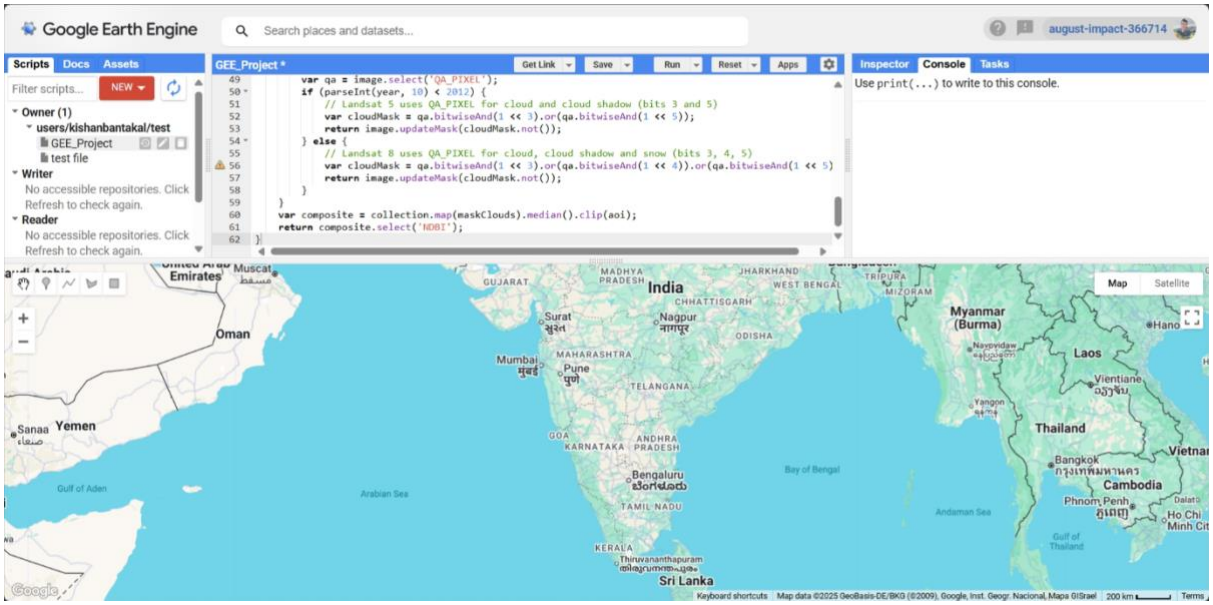


Fig: 3 Image representing NDBI Calculation using Landsat Data

Step 03: NDBI Actual Calculation and Built-Up Area Extractions

First the JavaScript calls the calculated NDBI data from previous code structure "calculateNDBI" function used for both the years 2000 and 2024. This function processes the Landsat imagery for each year and computes the NDBI for every pixel in AOI. This results to give two separate NDBI Maps mainly ***"ndbi2000"*** and ***"ndbi2024"***, where each pixel has a value from "-1" to "+1" indicating the AOI as built-up surface.

Second is that, the classification of Built-Up Areas to Raster to Vector

a) Creating Binary Rasters

The given below code functions two critical operations (eg., from code)

"builtup2000 = ndbi2000.gt(NDBI_THRESHOLD).selfMask()"

where ***".gt(NDBI_THRESHOLD)"*** it applies the threshold (0) to the NDBI map so any pixel with a value greater than 0 is given a new value of 1 (True), and all others get a value of 0 (False).

".selfMask()" this function makes all the pixels with a value of 0 transparent.

At last, the result is a binary raster map ***"builtup2000"*** and ***"builtup2024"*** where only the pixels classified as built-up areas are visible.

b) Generating Vector Polygons

After creating binary raster code uses ***"reduceToVectors"*** to convert the pixel-based raster maps into vector polygons. It groups adjacent built-up areas pixels together and draws a single, clean polygon shape around them. This is useful for more precise area calculations and for exporting the data as a shapefile **(.SHP)** for use in GIS software like QGIS or ArcGIS.

c) Identifying Urban Expansions

"var expansion = builtup2024.and(builtup2000.not()).selfMask();"

Uses boolean logic to pinpoint new growth at last part of the below code

"builtup2000.not()" this applies to 2000 map highlighting everywhere that was not built-up in 2000.

"builtup2024.and()" it finds the areas that are built-up in 2024 and also not built-up in 2000.

The result for this step is that the final map (expansion) that shows only the areas of new urban development that occurred between 2000 and 2024.

```
// Step 03: NDBI Calculation and Built-up areas extraction

// Calculate NDBI for both years 2000 and 2024
var ndbi2000 = calculateNDBI(year2000, aoi);
var ndbi2024 = calculateNDBI(year2024, aoi);

// Creating binary built-up raster maps by previously defined thresholding
NDBI
var builtup2000 = ndbi2000.gt(NDBI_THRESHOLD).selfMask().rename('Builtup');
var builtup2024 = ndbi2024.gt(NDBI_THRESHOLD).selfMask().rename('Builtup');

// Calculate and generate vector polygons from the raster built-up areas
var vectors2000 = builtup2000.reduceToVectors({
    geometry: aoi,
    scale: 90,
    geometryType: 'polygon',
    eightConnected: false,
    labelProperty: 'builtup',
    maxPixels: 1e10
});

var vectors2024 = builtup2024.reduceToVectors({
    geometry: aoi,
    scale: 90,
    geometryType: 'polygon',
    eightConnected: false,
    labelProperty: 'builtup',
    maxPixels: 1e10
});

// Calculate the expansions of built-up areas from 2000 to 2024
var expansion = builtup2024.and(builtup2000.not()).selfMask();
```

***** The code used does not show the results for each step like 01, 02, 03 and 04 it's stored as in a loop. Without calling the main function do not work as we tap "RUN" at that instant in GEE. *****

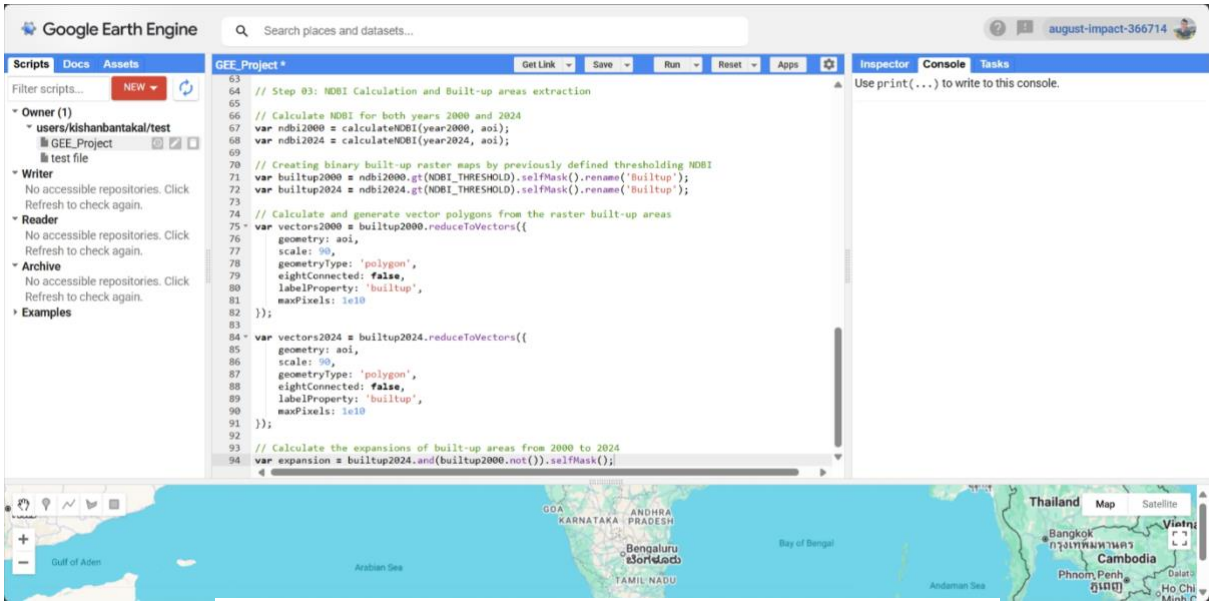


Fig: 4 Image representing NDBI actual calculation and Built-up areas extraction

Step 04: Area Calculations for Built-Up Areas

The code given for step 04 below calculates the total built-up area (in sq. km) for each year. It multiplies the built-up mask by pixel area to get each pixel's size in square meters then it sums all those pixel areas inside the AOI using ***"reduceRegion()"***. Finally, converting the result from m² to km² by dividing by 1,000,000 ***"1e6"***. So ***"area2000"*** and ***"area2024"*** store the built-up area for 2000 and 2024 respectively. [4], [6]

```
// Step 04: Area calculation for built-up areas
function calculateAreaSqKm(image) {
    var area = image.multiply(ee.Image.pixelArea()).reduceRegion({
        reducer: ee.Reducer.sum(),
        geometry: aoi,
        scale: 30,
        maxPixels: 1e12
    });
    return ee.Number(area.get('Builtup')).divide(1e6);
}

// Calculate areas for both years 2000 and 2024
var area2000 = calculateAreaSqKm(builtup2000);
var area2024 = calculateAreaSqKm(builtup2024);
```

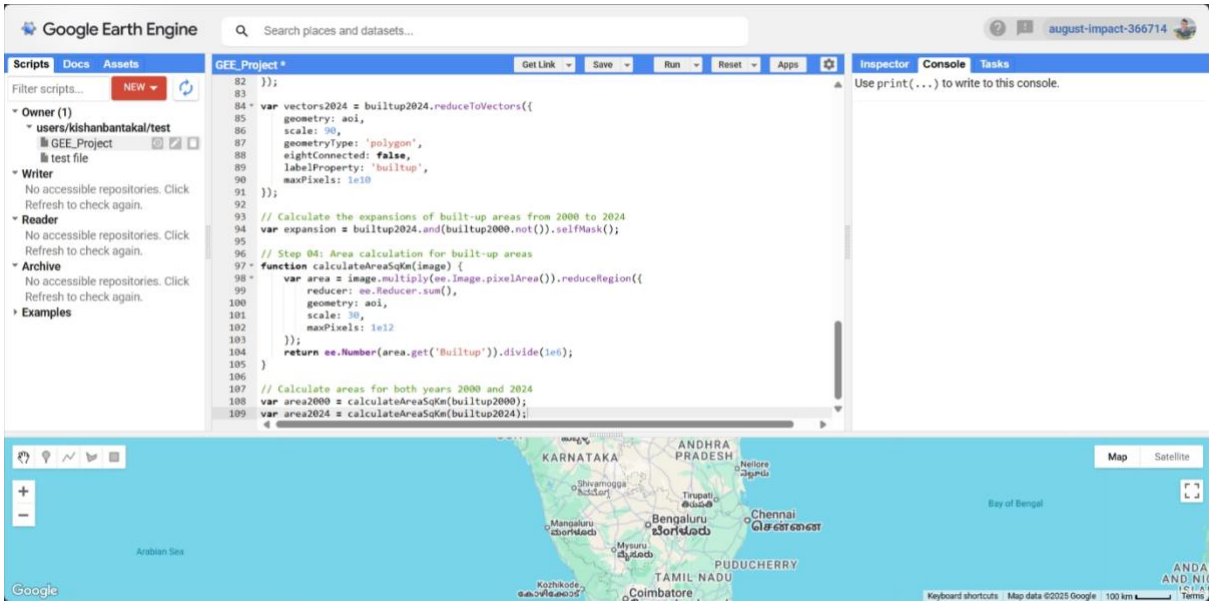


Fig: 5 Image representing NDBI actual calculation and Built-up areas extraction

Step 05: Export all the results from previous steps

This step exports all the previous step analysis results from GEE to Google Drive in different formats like:

- NDBI maps for 2000 and 2024 as GeoTIFFs.
- Built-up raster maps (2000 in red, 2024 in blue) as GeoTIFFs.
- Built-up area polygons for both years as shapefiles (SHP).

I wanted to create Build Expansion and Change Matrix. But there was error in expansion analysis in single file and co-ordinates update issue. So instead of this I have created custom UI and Side-to-Side bar Comparison where it creates a layer and gives comparison for 2000 and 2024, side by side comparison. These features are given in upcoming steps.

When we tap “RUN” with this step new code “Task” menu show *orange* color where it has another part to “RUN” which is used to connect and store to the Google Drive folder saved in “GEE_EXPORTS” folder.

```
// Step 05: Export all the results from previous steps

// Export NDBI image in Tif format for both years 2000 and 2024
Export.image.toDrive({
    image: ndbi2000.clip(aoi),
    description: 'NDBI_2000',
    folder: 'GEE_Exports',
    scale: 30,
    region: aoi,
    fileFormat: 'GeoTIFF'
});

Export.image.toDrive({
    image: ndbi2024.clip(aoi),
    description: 'NDBI_2024',
    folder: 'GEE_Exports',
    scale: 30,
    region: aoi,
    fileFormat: 'GeoTIFF'
});

//Export built-up raster image and vector polygons for both years 2000 and
2024
var redMask2000 = builtup2000.visualize(builtupVisParams2000);
var blueMask2024 = builtup2024.visualize(builtupVisParams2024);

Export.image.toDrive({
    image: redMask2000.clip(aoi),
    description: 'BuiltupRaster_2000',
    folder: 'GEE_Exports',
    scale: 30,
    region: aoi,
    fileFormat: 'GeoTIFF'
});

Export.image.toDrive({
    image: blueMask2024.clip(aoi),
    description: 'BuiltupRaster_2024',
    folder: 'GEE_Exports',
    scale: 30,
    region: aoi,
    fileFormat: 'GeoTIFF'
});

Export.table.toDrive({
    collection: vectors2000,
    description: 'BuiltupPolygons_2000',
    folder: 'GEE_Exports',
```

```
    fileFormat: 'SHP'
});

Export.table.toDrive({
    collection: vectors2024,
    description: 'BuiltupPolygons_2024',
    folder: 'GEE_Exports',
    fileFormat: 'SHP'
});
```

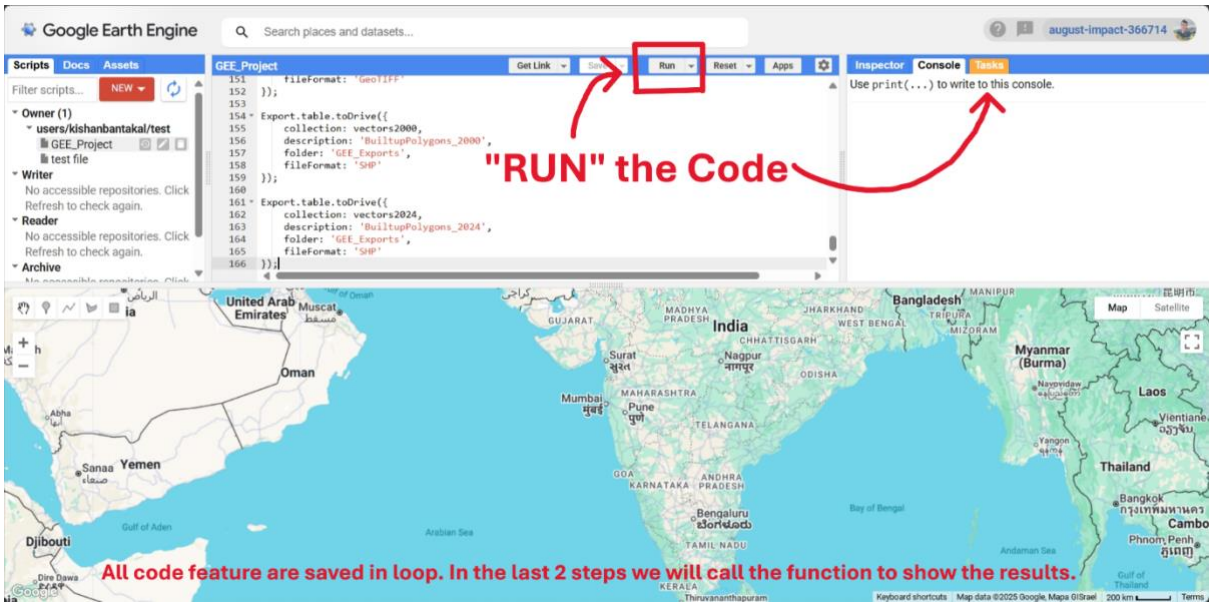


Fig: 6 Image representing the export feature (a)

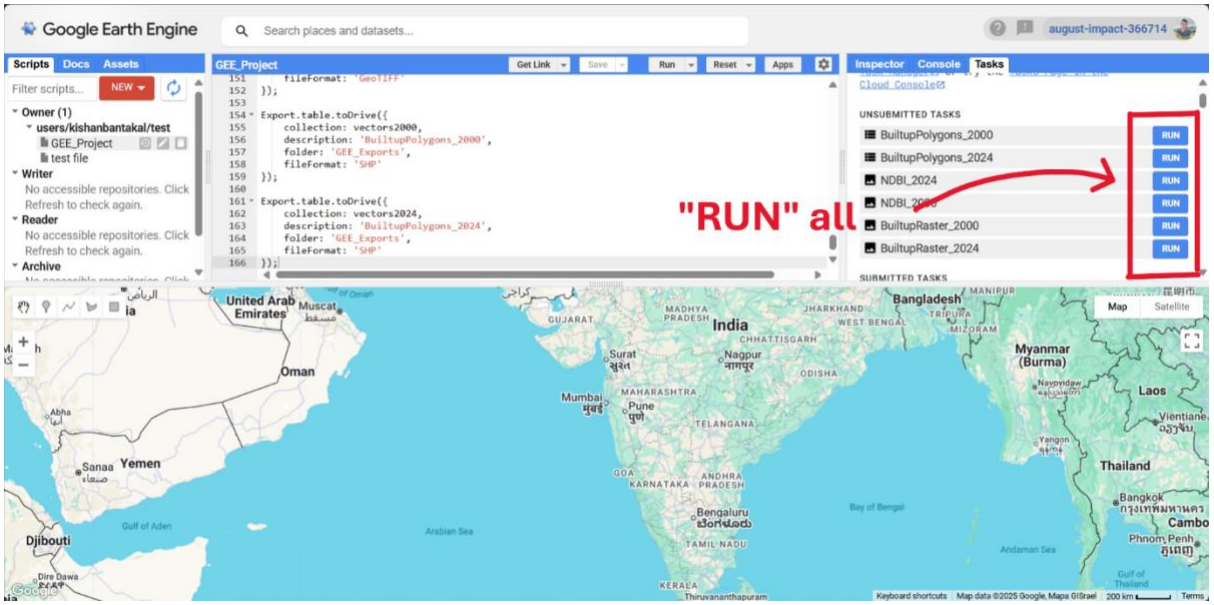


Fig: 7 Image representing the export feature (b)

Tap “RUN” button as shown in Fig: 7 All 6 “RUN” buttons. Wait for 2-3 minutes as expected, let it be export to Google Drive.

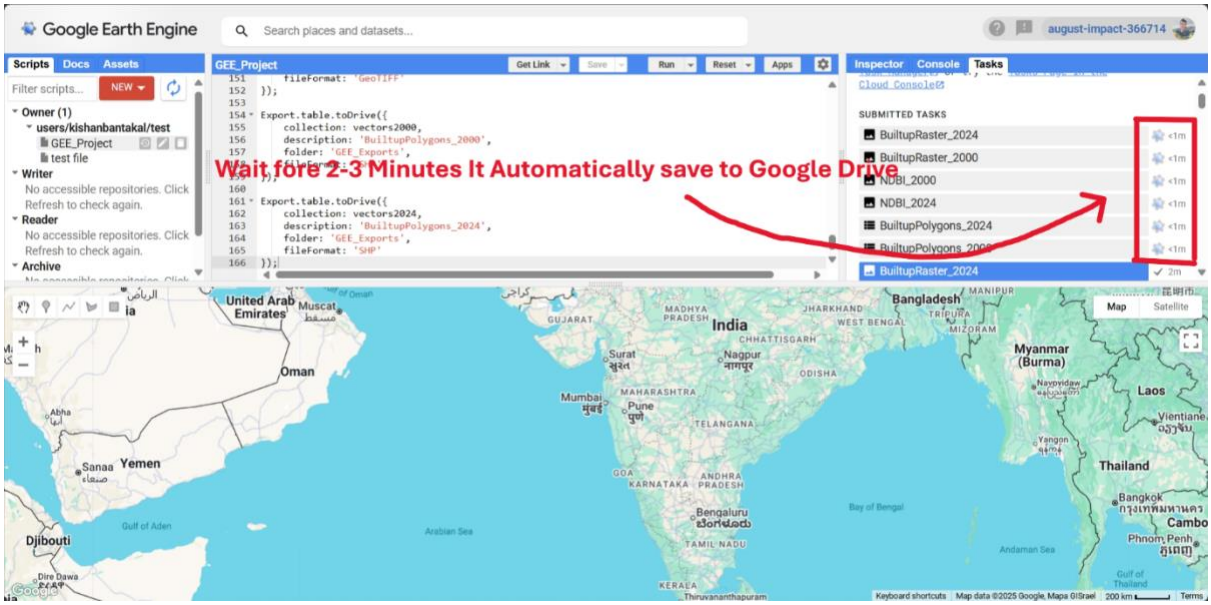


Fig: 8 Image representing the export feature (c)

After 2-3 Minutes open Google Drive and search for “GEE_EXPORTS” folder. In this folder all our Raster and Vector files are stored which can be used to import our analysis from GEE cloud based to Local Storage of our device. Select all the files and download. Extract the downloaded file.

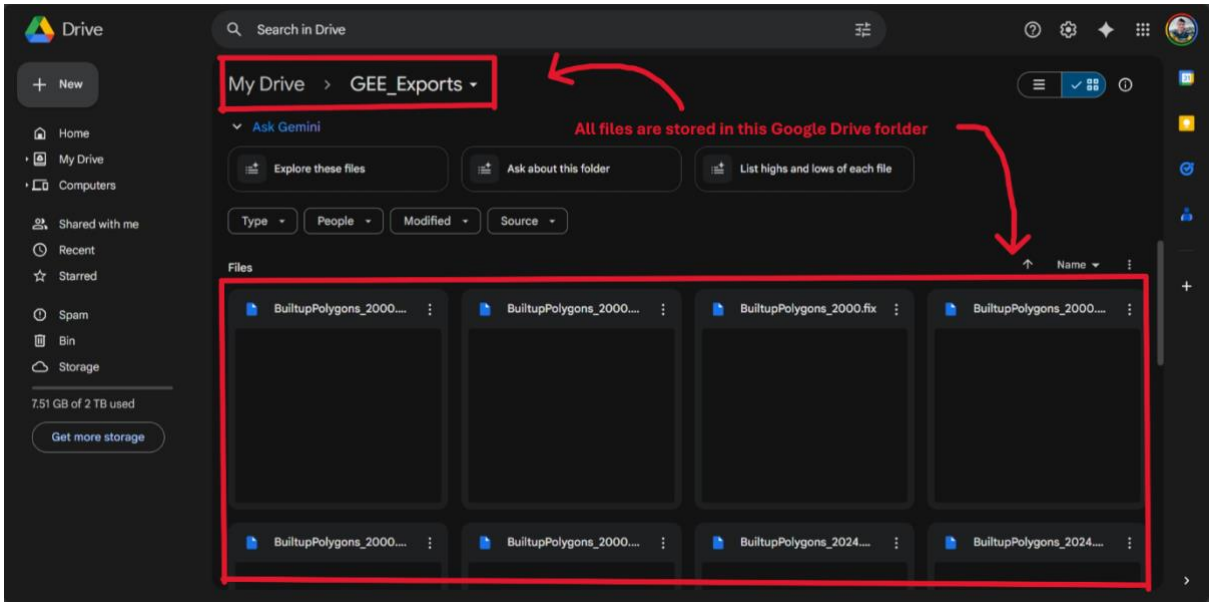


Fig: 9 Image representing the export feature (d)

Extracted file consists of all the GeoTIFFs and .SHP formatted files. These files can be used for Map layout, expansion and analysis using QGIS or ArcGIS.

Step 06: Creating custom User Interface (UI) for Area Statistics, Chart

The JavaScript code given below for this step creates a custom main container panel ***"mainPanel"*** which gives all the information on area statistics, add several Legends to the panel where a continuous colour bar for the NDBI values showing the range from 'Low' to 'High'.

Discrete colour coded legends for the classified map layers with Built-up 2000 (red), Built-up 2024 (blue) and New Expansion (yellow) is added for the code.

Area Statistics which notify "Calculating…" under 2000 and 2024 which is effective when area analysis is slow.

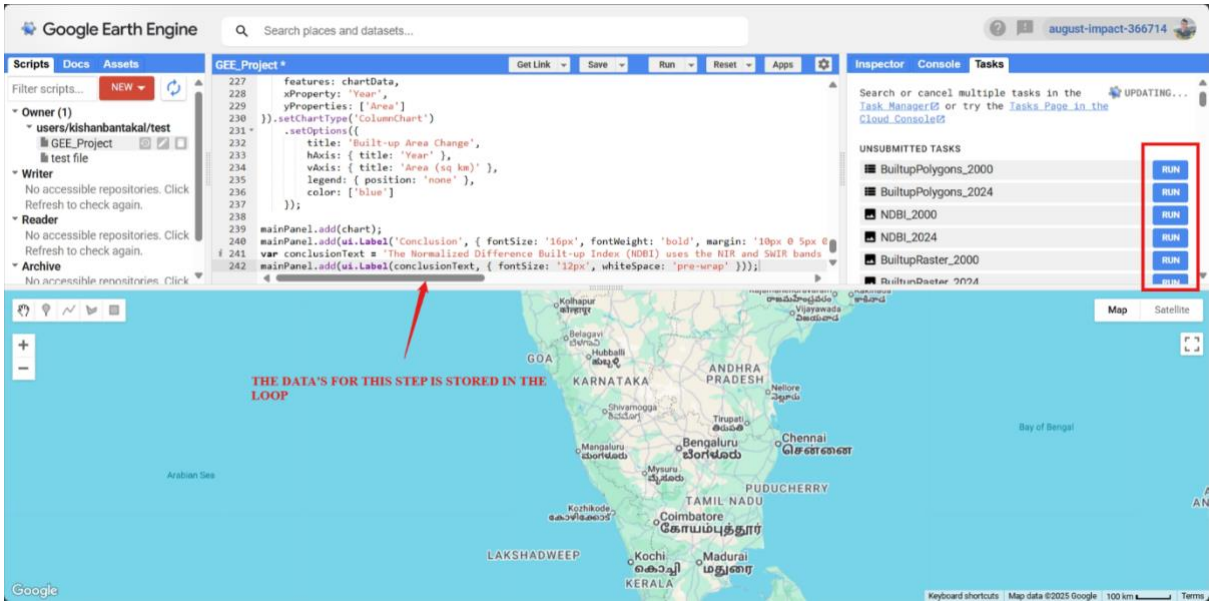


Fig: 10 Image representing the UI Panel was created and stored as loop

Chart is used to generate built-up area comparison between 2000 and 2024 which actually compares the area in square kilometres and shows the chart with the expansion, which makes easy to see the growth.

```
// Step 07: I have created user interface for Visualization and Analysis which
includes Area Statistics and Charts

var mainPanel = ui.Panel({
    style: {
        width: '350px',
        padding: '10px'
    }
});

mainPanel.add(ui.Label({
    value: 'Urban Expansion: Mysore',
    style: { fontSize: '20px', fontWeight: 'bold' }
}));

var makeLegend = function (title, palette) {
    var legendTitle = ui.Label(title, { fontWeight: 'bold' });
    var colorBar = ui.Thumbnail({
        image: ee.Image.pixelLonLat().select(0),
        params: { bbox: [0, 0, 1, 0.1], palette: palette },
        style: { stretch: 'horizontal', margin: '0px 8px', maxHeight: '24px'
},
    });
    var labels = ui.Panel({
        widgets: [
            ui.Label('Low', { margin: '4px 8px' }),
            ui.Label('High', { margin: '4px 8px', textAlign: 'right', stretch:
'horizontal' })
        ],
        layout: ui.Panel.Layout.flow('horizontal')
    });
    return ui.Panel([legendTitle, colorBar, labels]);
};

// Adding Lengends and Area Statistics to the main panel
mainPanel.add(makeLegend('NDBI Value', ndbiVisParams.palette));
var legend2000 = ui.Panel([ui.Label('■', { color: 'red' }), ui.Label('Built-up
2000')], ui.Panel.Layout.flow('horizontal'));
var legend2024 = ui.Panel([ui.Label('■', { color: 'blue' }), ui.Label('Built-
up 2024')], ui.Panel.Layout.flow('horizontal'));
var legendExpansion = ui.Panel([ui.Label('■', { color: 'yellow' }),
ui.Label('New Expansion (2000-2024)')], ui.Panel.Layout.flow('horizontal'));

mainPanel.add(ui.Panel([legend2000, legend2024, legendExpansion]));
mainPanel.add(ui.Label('Area Statistics', { fontSize: '16px', fontWeight:
'bold', margin: '10px 0 5px 0' }));

var areaLabel2000 = ui.Label('Calculating area for 2000...');
```

```
var areaLabel2024 = ui.Label('Calculating area for 2024...');

mainPanel.add(areaLabel2000);
mainPanel.add(areaLabel2024);
area2000.evaluate(function (area) {
    areaLabel2000.setValue('Built-up Area in 2000: ' + area.toFixed(2) + ' sq
km');
});

area2024.evaluate(function (area) {
    areaLabel2024.setValue('Built-up Area in 2024: ' + area.toFixed(2) + ' sq
km');
});

var chartData = ee.FeatureCollection([
    ee.Feature(null, { 'Year': '2000', 'Area': area2000 }),
    ee.Feature(null, { 'Year': '2024', 'Area': area2024 })
]);

var chart = ui.Chart.feature.byFeature({
    features: chartData,
    xProperty: 'Year',
    yProperties: ['Area']
}).setChartType('ColumnChart')
    .setOptions({
        title: 'Built-up Area Change',
        hAxis: { title: 'Year' },
        vAxis: { title: 'Area (sq km)' },
        legend: { position: 'none' },
        color: ['blue']
    });

mainPanel.add(chart);
mainPanel.add(ui.Label('Conclusion', { fontSize: '16px', fontWeight: 'bold',
margin: '10px 0 5px 0' }));
var conclusionText = 'The Normalized Difference Built-up Index (NDBI) uses the
NIR and SWIR bands to emphasize manufactured built-up areas. It is ratio based
to mitigate the effects of terrain illumination differences as well as
atmospheric effects.'
mainPanel.add(ui.Label(conclusionText, { fontSize: '12px', whiteSpace: 'pre-
wrap' }));
```

Step 07: Visualization of results, Map Layout and Final UI Layout

The visualization and map layout were developed in Google Earth Engine by creating two synchronized map panels ***"leftMap"*** and ***"rightMap"*** each has the centre pointer on the AOI at zoom level "11". The left map displayed the NDBI raster, built-up classification raster and vectorized built-up polygons for the year 2000, while the right map shows the same layers for 2024 along with an additional urban expansion layer. Change detection was performed by subtracting the 2000 built-up mask from the 2024 mask to highlight expansion zones, which were visualized on the right map. The maps were linked using ***"ui.Map.Linker"*** to synchronize panning and zooming, and a horizontal split view (***ui.SplitPanel***) enabled side-by-side comparison. Year labels 2000 and 2024 were placed on the respective maps and a main control panel ***"mainPanel"*** containing analysis tools and legends was positioned alongside the split maps for easy user interaction. [2], [4], [5], [6], [7]

By adding this code below from this step gives the final result of analysis in GEE Map. 'Step 09' part of the given code in this step below plays crucial role. It collects all the data which were stored in a loop from Step 01, 02, 03, 04, 05, 06 and 07.

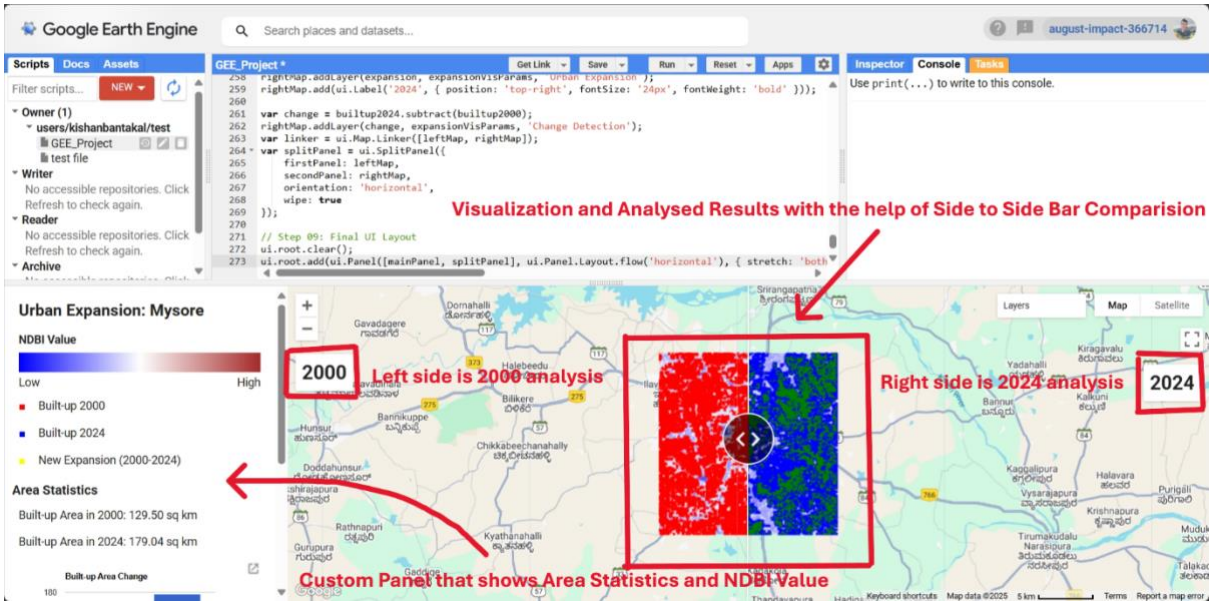


Fig: 11 Image representing Custom UI Panel and Data Visualization on Maps of 2000 vs 2024

```
// Step 08: Visualization of results and Map Layout

var leftMap = ui.Map();
var rightMap = ui.Map();

leftMap.centerObject(aoi, 11);
rightMap.centerObject(aoi, 11);

leftMap.addLayer(ndbi2000, ndbiVisParams, 'NDBI Raster 2000');
leftMap.addLayer(builtup2000, builtupVisParams2000, 'Built-up Raster 2000');
leftMap.addLayer(vectors2000, { color: 'red' }, 'Built-up Polygons 2000');
leftMap.add(ui.Label('2000', { position: 'top-left', fontSize: '24px',
fontWeight: 'bold' }));
rightMap.addLayer(ndbi2024, ndbiVisParams, 'NDBI Raster 2024');
rightMap.addLayer(builtup2024, builtupVisParams2024, 'Built-up Raster 2024');
rightMap.addLayer(vectors2024, { color: 'blue' }, 'Built-up Polygons 2024');
rightMap.addLayer(expansion, expansionVisParams, 'Urban Expansion');
rightMap.add(ui.Label('2024', { position: 'top-right', fontSize: '24px',
fontWeight: 'bold' }));

var change = builtup2024.subtract(builtup2000);
rightMap.addLayer(change, expansionVisParams, 'Change Detection');
var linker = ui.Map.Linker([leftMap, rightMap]);
var splitPanel = ui.SplitPanel({
    firstPanel: leftMap,
    secondPanel: rightMap,
    orientation: 'horizontal',
    wipe: true
});
```

```
// Step 09: Final UI Layout
ui.root.clear();
ui.root.add(ui.Panel([mainPanel, splitPanel],
ui.Panel.Layout.flow('horizontal'), { stretch: 'both' }));
```

4.2. Quantum Geographical Information System (QGIS) Based Workflow:

QGIS Software was used only for opening the GeoTIFFs and Shapefiles (.SHP) formatted files. I have used this Software to create maps showing Urban Expansion of Mysore 2000 vs 2024, where I have downloaded India shapefile (.SHP) and Karnataka shapefile (.SHP) from source. Imported the file as layers which was obtained from GEE of previous steps mentioned. [7]

a. QGIS Software

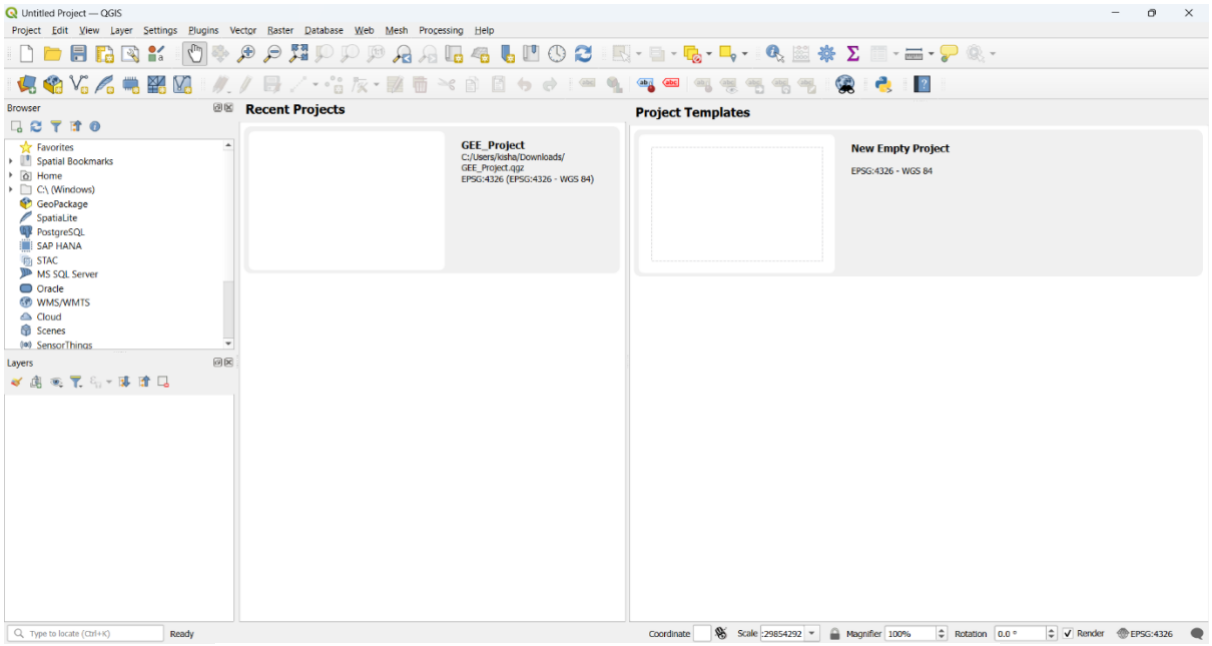


Fig: 12 Image representing QGIS Software Interface

b. Adding Maps (India, Karnataka, all .SHP and .TIFF formatted files)

Import all the files which ever required from “Layers” option from the header. It makes very easy when we create our custom map

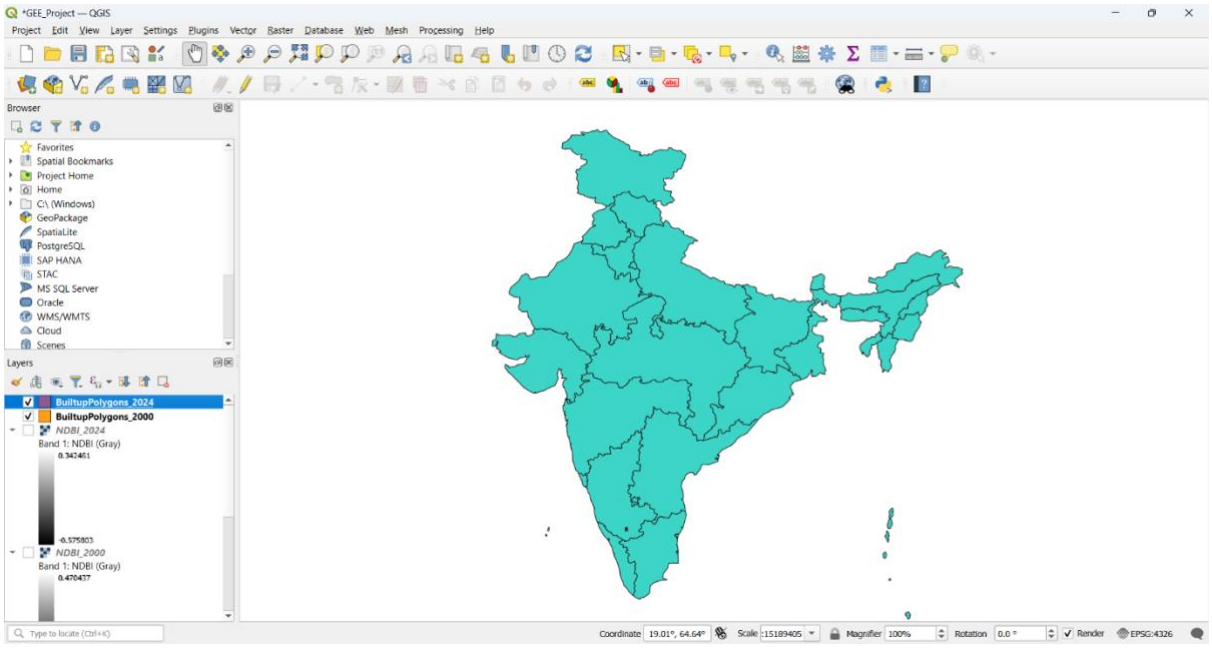


Fig: 13 Image representing Imported file and Adding Layers in QGIS

c. Creating the Maps and Modify

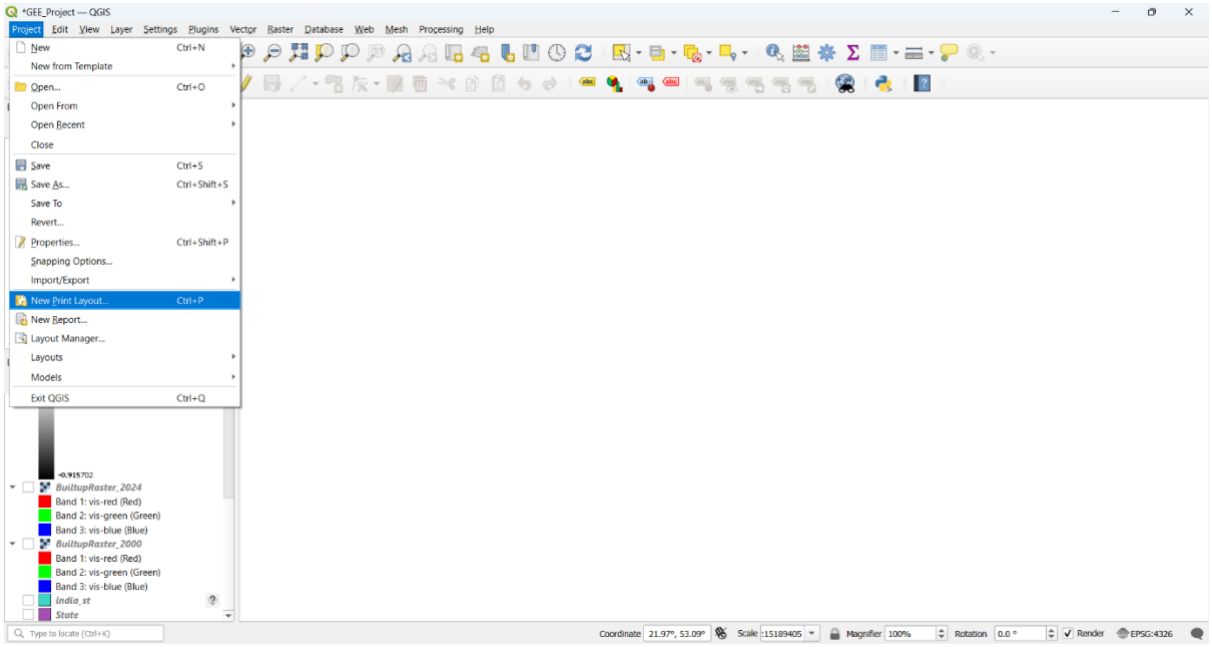


Fig: 14 Image representing to create a print layout maps chart 2000 vs 2024 Urban Expansion

d. Export either as .PDF or .PNG

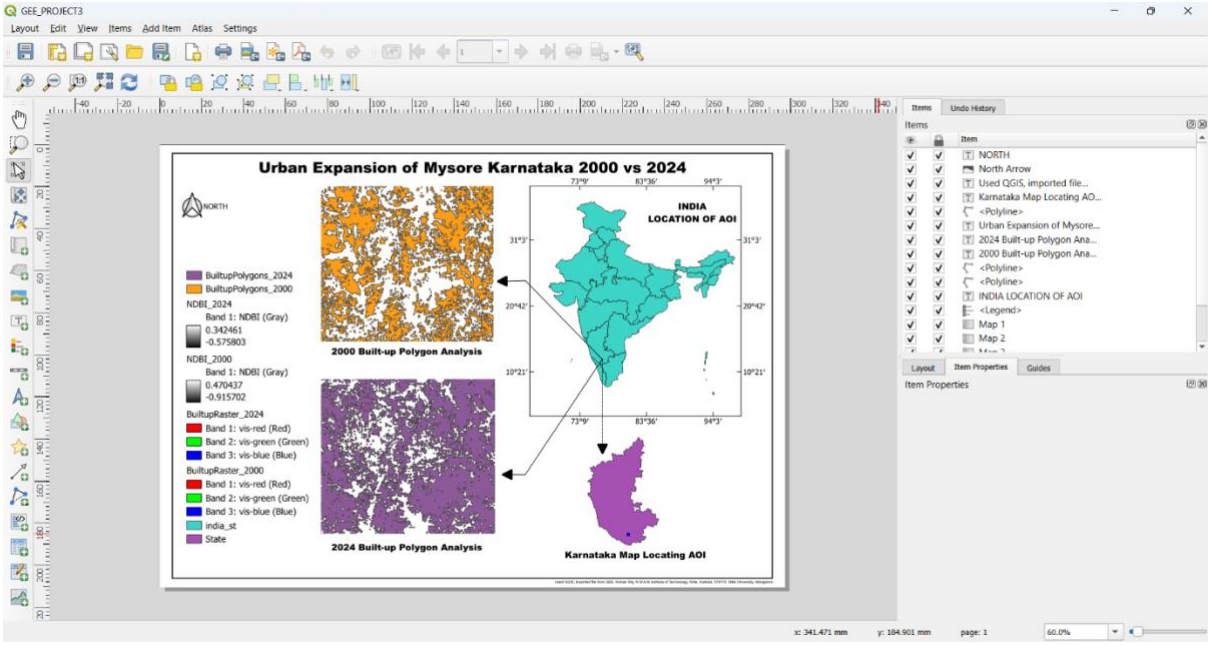


Fig: 15 Image representing QGIS all the features and creating the Map Layouts

5. Results

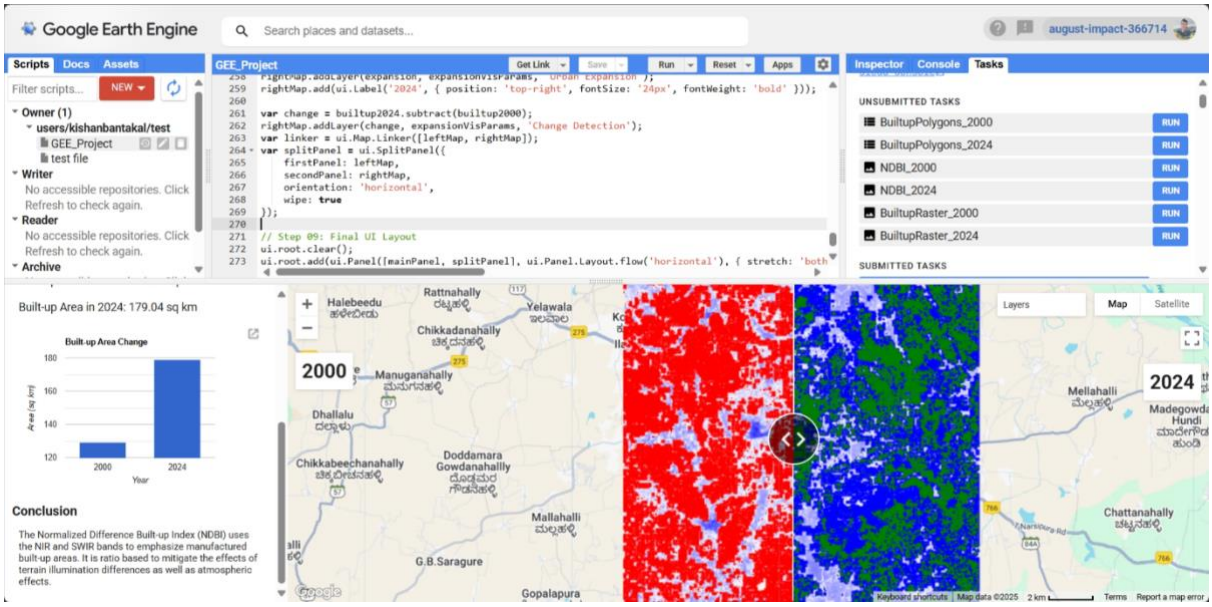


Fig: 16 Image representing GEE custom UI Panel with Area Statistics, Conclusion and Map Analysis

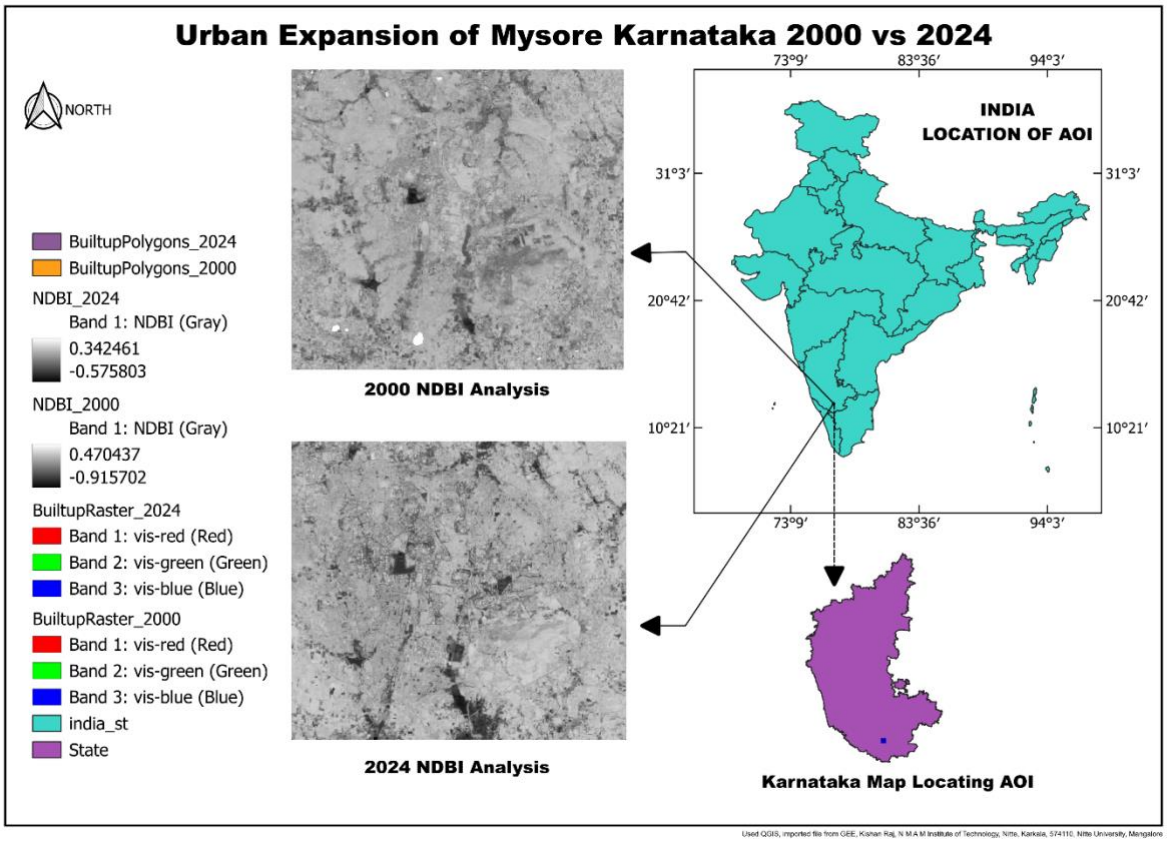
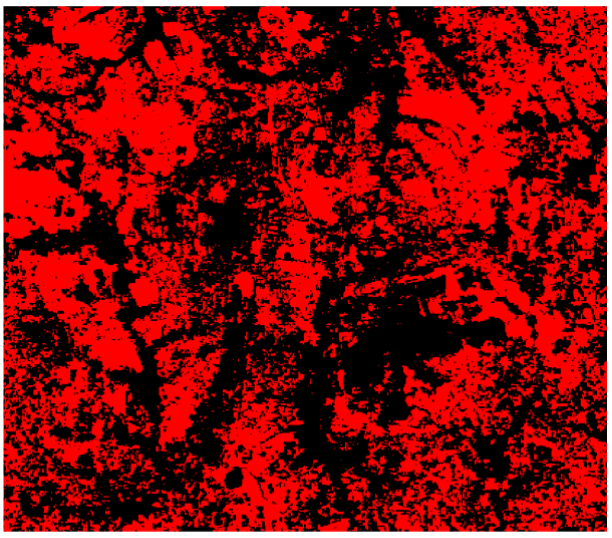
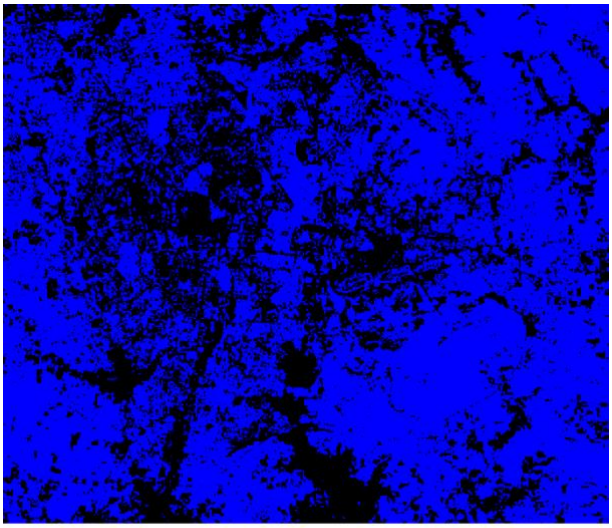


Fig: 17 Image representing Urban Expansion Analysis using NDBI of 2000 vs 2024 AOI

Results of 2000 vs 2024 Expansion Analysis of Mysore:



2000 Built-up Raster Analysis



2024 Built-up Raster Analysis

Between 2000 and 2024, Mysore's built-up area increased from **129.50 sq. km** to **179.04 sq. km**, showing a net growth of **38.25%**. This expansion was primarily concentrated towards the northern and eastern parts of the city, where agricultural land was rapidly converted into residential and industrial zones. The decline of green patches and water bodies further indicates the environment of urban growth. The green/yellow spots in GEE show the new expansion throughout these 24 years. The comparative maps and UI based statistics strongly highlight the scale of this transformation reflecting the city's transformation into a major urban hub.

6. Conclusion

This project successfully analyses, quantifies and visualizes Mysore's urban expansion between 2000 and 2024 using Google Earth Engine (GEE) and QGIS workflow. GEE's step-by-step script has automated the process from acquiring and pre-processing multi-sensor Landsat imagery (Landsat 5 & 8) to calculating the Normalized Difference Built-up Index (NDBI) for analyse built-up areas. The custom UI includes area statistics, temporal change charts and a split-panel viewer for clear spatiotemporal comparisons. GEE's cloud-based processing efficiently handled decades of satellite data, utilizing its vast catalogue and rapid computation. QGIS concludes by showing the outputs of GEE imported files and refining final maps, offering superior control over cartographic design. Exported GeoTIFF and Shapefile layers ensures cross verification, professional presentation and chart preparation. Together, GEE's analysis power and QGIS's mapping capabilities gives accurate results as mentioned in methodology that provides all the urban expansion analysis and results of Mysore city for planners, policymakers, researchers and students. [1], [2], [3], [4], [7], [8]

7. References

[1] W. Suparta e H. G. Hartono, "Application of the NDBI Method For Determining of Built-Up Land (A Case Study For Surabaya City, Indonesia)", 2024. [Online]. Disponível em: https://www.vecteezy.com

[2] M. K. Firozjaei, A. Sedighi, M. Kiavarz, S. Qureshi, D. Haase, e S. K. Alavipanah, "Automated built-up extraction index: A new technique for mapping surface built-up areas using LANDSAT 8 OLI imagery", *Remote Sens (Basel)*, vol. 11, nº 17, 2019, doi: 10.3390/rs11171966.

[3] S. C. L e R. K. V, "Educational Administration: Theory and Practice Unfolding The Political Scenario Of Princely Mysore: Historical Insights And Movements", *Educational Administration: Theory and Practice*, vol. 29, nº 4, p. 2007–2014, 2023, doi: 10.53555/kuey.v29i4.6823.

[4] M. Y. Yasin, J. Abdullah, N. M. Noor, M. M. Yusoff, e N. M. Noor, "Landsat observation of urban growth and land use change using NDVI and NDBI analysis", em *IOP Conference Series: Earth and Environmental Science*, Institute of Physics, 2022. doi: 10.1088/1755-1315/1067/1/012037.

[5] M. Amani *et al.*, "Google Earth Engine Cloud Computing Platform for Remote Sensing Big Data Applications: A Comprehensive Review", *IEEE J Sel Top Appl Earth Obs Remote Sens*, vol. 13, p. 5326–5350, 2020, doi: 10.1109/JSTARS.2020.3021052.

[6] M. A. Wulder *et al.*, "Fifty years of Landsat science and impacts", 1º de outubro de 2022, *Elsevier Inc.* doi: 10.1016/j.rse.2022.113195.

[7] P. P. Dash *et al.*, "Quantification of Urban Expansion Using Geospatial Technology—A Case Study in Bangalore", *Advances in Remote Sensing*, vol. 04, nº 04, p. 330–342, 2015, doi: 10.4236/ars.2015.44027.

[8] M. Imran e N. Almusharraf, "Google Gemini as a next generation AI educational tool: a review of emerging educational technology", 1º de dezembro de 2024, *Springer*. doi: 10.1186/s40561-024-00310-z.

Other References

1. Source Code: https://github.com/Kishan0405/GEE_Project/blob/main/gee_project.js
2. Source Image: https://drive.google.com/
3. https://earthengine.google.com/, https://www.neonscience.org/resources/learning-hub/tutorials/intro-gee-functions
4. https://grok.com/, https://gemini.google.com/, https://chatgpt.com/

****