Visualizing Traffic and Pollution Impact Using Google Earth

Visualizing Traffic & Pollution in Delhi (2024) Using Google Earth Engine 🛰️

Excited to share my latest project where I leveraged Google Earth Engine to analyze and visualize traffic activity and air pollution levels across Delhi in 2024.

- 🔹 Traffic Activity: Assessed using VIIRS Nighttime Lights data to identify high and low traffic zones.
- 🔹 Air Quality: Analyzed NO_2, O_3, CO, and SO_2 levels using Sentinel-5P data to map pollution hotspots.
- 🔹 Road Traffic Detection: Utilized Sentinel-2 imagery to gain insights into road congestion patterns.
- 🔹 Comprehensive Analysis: Integrated multiple datasets to provide an interactive visualization, aiding urban planning and policy-making.

INSIGHTS : -

Mean Nighttime Lights in Delhi: 26.99094807218145

Mean NO2 Concentration in Delhi: 0.00016441803195291787

Mean O3 Concentration in Delhi: 0.12569749605043584

Mean CO Concentration in Delhi: 0.039861801207948

Mean SO2 Concentration in Delhi: 0.00017769950850015775

Engine

- **Define Area of Interest (AOI) - Delhi Boundary:**
  - Load the Delhi boundary from the FAO/GAUL dataset.
  - Filter the data to select the region corresponding to Delhi using its administrative
  - level.
- **Nighttime Lights Analysis (Traffic Activity Indicator):**
  - Load the VIIRS Nighttime Lights dataset.
  - Filter it to include only the data for Delhi and the year 2023.
  - Select the avg_rad band, which represents the average radiance (nighttime lights)
  - indicating traffic activity.
  - Compute the mean nighttime lights for the year.
  - Apply a color palette for visualization, ranging from low (black) to high (red) activity.
  - Add the nighttime lights layer to the map.

• Air Quality Analysis:

• NO_2 Levels:

- Load the Sentinel-5P $NO_{2\ 22\ 2}$ dataset for the year 2023.
- Select the NO2_column_number_density band.
- Compute the mean $NO_{2\ 22\ 2}$ concentration for the year.
- Visualize using a color palette from blue (low) to red (high).

- Add the NO_2 layer to the map.

• O_3 Levels:

- Load the Sentinel-5P $O_{3\ 33\ 3}$ dataset for the year 2023.
- Select the O3_column_number_density band.
- Compute the mean $O_{3\ 33\ 3}$ concentration for the year.
- Visualize using a color palette from blue (low) to red (high).
- Add the O_3 layer to the map.

• CO Levels:

- Load the Sentinel-5P CO dataset for the year 2023.
- Select the CO_column_number_density band.
- Compute the mean CO concentration for the year.
- Visualize using a color palette from blue (low) to red (high).
- Add the CO layer to the map.

• SO_2 Levels:

- Load the Sentinel-5P $SO_{2\ 22\ 2}$ dataset for the year 2023.
- Select the SO2_column_number_density band.
- Compute the mean $SO_{2\ 22\ 2}$ concentration for the year.
- Visualize using a color palette from blue (low) to red (high).
- Add the SO_2 layer to the map.

• Sentinel-2 Imagery for Road Traffic Detection:

- Load Sentinel-2 imagery for 2023 with cloud coverage below 10%.
- Select the RGB bands (B4, B3, and B2) to visualize road traffic.
- Sort the collection by date and select the most recent cloud-free image.
- Apply a color scale for visualization.
- Add the RGB Sentinel-2 image to the map.

• Print Statistics for Analysis:

- Use reduceRegion() to compute the mean of each dataset for Nighttime Lights,
- NO_2, O_3, CO, and SO_2 for the Delhi region.
- Print the statistics to the console for each parameter to understand the average levels.

• Add a Legend for Nighttime Lights (Traffic Levels):

- Create a custom legend to represent traffic levels based on nighttime lights intensity.
- Use color boxes to show low, moderate, and high traffic levels (black, yellow, and
- red, respectively).
- Position the legend in the bottom-left corner of the map.

• Add a Title to the Map:

- Create a title "Visualizing Traffic and Pollution Impact Using Google Earth
- Engine."
- Position the title at the top-center of the map with a background for readability.
- Add the title to the map.

- **Final Visualization:**
    - Center the map on Delhi.
    - Add all the layers for nighttime lights (traffic), air quality (NO_2, O_3, CO, SO_2), and
    - Sentinel-2 imagery.
    - Display the title and legend, giving a comprehensive view of the traffic and pollution
    - levels across Delhi for 2023.

CODE-

```
// Define Area of Interest (AOI) - Delhi boundary
var delhi = ee.FeatureCollection('FAO/GAUL/2015/level2')
            .filter(ee.Filter.eq('ADM2_NAME', 'Delhi'));

// ------------------------------------
// 1. Nighttime Lights Analysis (Traffic activity indicator)
var viirs = ee.ImageCollection('NOAA/VIIRS/DNB/MONTHLY_V1/VCMCFG')
          .filterBounds(delhi)
          .filterDate('2024-01-01', '2024-12-31')
.select('avg_rad'); // Average radiance

// Calculate the mean nighttime lights over the year
var meanLights = viirs.mean().clip(delhi);

// Visualization parameters for Nighttime Lights
var lightsVisParams = {
  min: 0,
  max: 50,
  palette: ['black', 'blue', 'purple', 'cyan', 'green', 'yellow', 'red']
};

// Add the nighttime lights layer to the map
Map.centerObject(delhi, 10);
```

```
Map.addLayer(meanLights, lightsVisParams, 'Mean Nighttime Lights');


// ------------------------------------
// 2. Air Quality Analysis (NO₂ levels)
var no2 = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_NO2')
          .filterBounds(delhi)
          .filterDate('2023-01-01', '2023-12-31')
          .select('NO2_column_number_density');


// Calculate the mean NO2 concentration over the year
var meanNo2 = no2.mean().clip(delhi);


// Visualization parameters for NO₂
var no2VisParams = {
  min: 0,
  max: 0.0002,
  palette: ['blue', 'green', 'yellow', 'orange', 'red']
};


// Add the NO₂ layer to the map
Map.addLayer(meanNo2, no2VisParams, 'Mean NO₂ Concentration');


// ------------------------------------
// 3. Ozone (O₃) Analysis
var o3 = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_O3')
          .filterBounds(delhi)
          .filterDate('2023-01-01', '2023-12-31')
          .select('O3_column_number_density');


// Calculate the mean O₃ concentration over the year
```

```
var meanO3 = o3.mean().clip(delhi);

// Visualization parameters for O₃
var o3VisParams = {
  min: 0,
  max: 0.0002,
  palette: ['blue', 'green', 'yellow', 'orange', 'red']
};

// Add the O₃ layer to the map
Map.addLayer(meanO3, o3VisParams, 'Mean O₃ Concentration');

// ---------------------------------------
// 4. Carbon Monoxide (CO) Analysis
var co = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_CO')
            .filterBounds(delhi)
            .filterDate('2023-01-01', '2023-12-31')
            .select('CO_column_number_density');

// Calculate the mean CO concentration over the year
var meanCO = co.mean().clip(delhi);

// Visualization parameters for CO
var coVisParams = {
  min: 0,
  max: 0.0002,
  palette: ['blue', 'green', 'yellow', 'orange', 'red']
};

// Add the CO layer to the map
```

```
Map.addLayer(meanCO, coVisParams, 'Mean CO Concentration');


// ------------------------------------
// 5. Sulfur Dioxide (SO₂) Analysis
var so2 = ee.ImageCollection('COPERNICUS/S5P/NRTI/L3_SO2')
          .filterBounds(delhi)
          .filterDate('2023-01-01', '2023-12-31')
          .select('SO2_column_number_density');


// Calculate the mean SO₂ concentration over the year
var meanSO2 = so2.mean().clip(delhi);


// Visualization parameters for SO₂
var so2VisParams = {
  min: 0,
  max: 0.0002,
  palette: ['blue', 'green', 'yellow', 'orange', 'red']
};


// Add the SO₂ layer to the map
Map.addLayer(meanSO2, so2VisParams, 'Mean SO₂ Concentration');


// ------------------------------------
// 6. Sentinel-2 Road Traffic Detection (RGB imagery)
var sentinel2 = ee.ImageCollection('COPERNICUS/S2')
                .filterBounds(delhi)
                .filterDate('2023-01-01', '2023-12-31')
                .filter(ee.Filter.lt('CLOUDY_PIXEL_PERCENTAGE', 10))
                .select(['B4', 'B3', 'B2']); // RGB bands
```

```
// Visualize the most recent cloud-free image
var recentImage = sentinel2.sort('system:time_start', false).first().clip(delhi);

// Visualization parameters for Sentinel-2 RGB
var rgbVisParams = {
  min: 0,
  max: 3000,
  bands: ['B4', 'B3', 'B2']
};

// Add the Sentinel-2 image to the map
Map.addLayer(recentImage, rgbVisParams, 'Recent Sentinel-2 Image');
// ------------------------------------
// Print statistics for analysis
var lightsStats = meanLights.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: delhi.geometry(),
  scale: 500,
  maxPixels: 1e8
});
print('Mean Nighttime Lights in Delhi:', lightsStats.get('avg_rad'));

var no2Stats = meanNo2.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: delhi.geometry(),
  scale: 1000,
  maxPixels: 1e8
});
print('Mean NO₂ Concentration in Delhi:', no2Stats.get('NO2_column_number_density'));
```

```
var o3Stats = meanO3.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: delhi.geometry(),
  scale: 1000,
  maxPixels: 1e8
});
print('Mean O₃ Concentration in Delhi:', o3Stats.get('O3_column_number_density'));

var coStats = meanCO.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: delhi.geometry(),
  scale: 1000,
  maxPixels: 1e8
});
print('Mean CO Concentration in Delhi:', coStats.get('CO_column_number_density'));

var so2Stats = meanSO2.reduceRegion({
  reducer: ee.Reducer.mean(),
  geometry: delhi.geometry(),
  scale: 1000,
  maxPixels: 1e8
});
print('Mean SO₂ Concentration in Delhi:', so2Stats.get('SO2_column_number_density'));

// ------------------------------------
// Add a legend for Nighttime Lights (Traffic Levels)
var legend = ui.Panel({style: {position: 'bottom-left', padding: '8px', margin: '8px'}});

// Add legend title
legend.add(ui.Label({
```

```
  value: 'Traffic Level (Nighttime Lights)',
  style: {fontWeight: 'bold', fontSize: '14px', margin: '0 0 4px 0'}
}));

// Function to create a legend row for Traffic Levels
function makeTrafficLegendRow(color, name) {
  var colorBox = ui.Label({
    style: {
      backgroundColor: color,
  padding: '8px',
      margin: '0 8px 0 0'
    }
  });
  var description = ui.Label(name);
  return ui.Panel([colorBox, description], ui.Panel.Layout.Flow('horizontal'));
}

// Add traffic level legend rows based on nighttime lights intensity
legend.add(makeTrafficLegendRow('black', 'Low Activity (0)'));
legend.add(makeTrafficLegendRow('yellow', 'Moderate Activity (25)'));
legend.add(makeTrafficLegendRow('red', 'High Activity (50)'));

// Add the legend to the map
Map.add(legend);

// Add Title for the Map
var title = ui.Label({
  value: 'Visualizing Traffic and Pollution Impact Using Google Earth Engine',
  style: {
    fontSize: '24px',
```

```
    fontWeight: 'bold',
    textAlign: 'center',
    padding: '10px',
    position: 'top-center',
    backgroundColor: 'rgba(255, 255, 255, 0.8)',
    margin: '10px'
  }
});

// Add the title to the map
Map.add(title);
```