



# CODE\_WIZARDS

- PS 2 DETECTING MISSING ROUNDABOUTS FOR DATABASE INTEGRATION

Team Members:

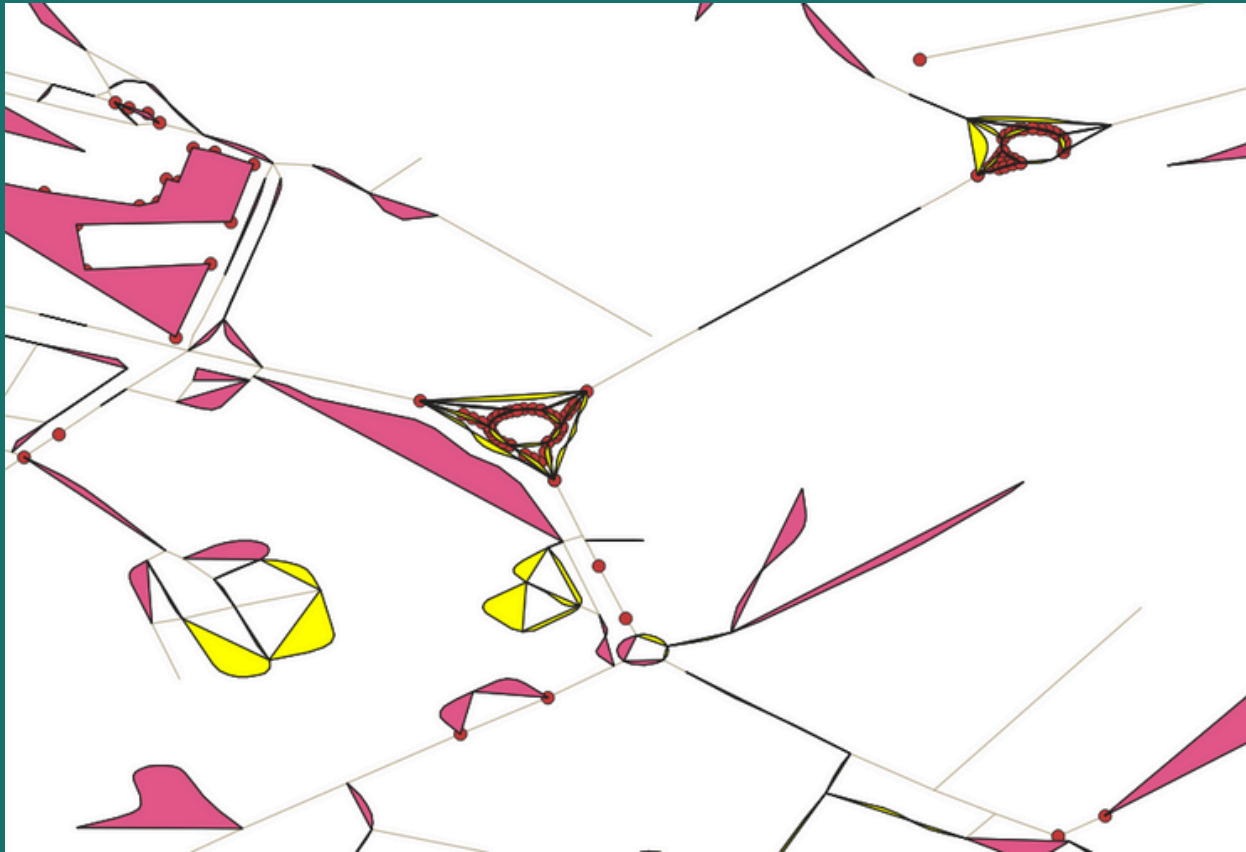
- Swarangi Shirsekar
- Vishesh Savani
- Chetan Patil

# INTRODUCTION

This project addresses the challenge of detecting missing roundabouts in a geographic database by leveraging spatial and geometric characteristics rather than relying on attribute data. Roundabouts, which are typically circular road intersections, can be identified based on their shape and connectivity. The approach involves preprocessing spatial data to break down complex geometries, add uniform vertex spacing, extract edges, and detect enclosed cycles through polygonization. These cycles are then evaluated based on their circularity, which measures how closely they resemble a circle, helping to pinpoint potential roundabouts.

To support accurate integration into the existing database, the detected roundabout geometries are further enriched with contextual data using spatial joins. This ensures that each identified feature carries relevant metadata, such as region or administrative zone, which aids in classification and reporting. The overall workflow provides a scalable and automated method to enhance the completeness and accuracy of infrastructure data, especially in scenarios where roundabout attributes are missing or incomplete.

# SPATIAL DATA PRE-PROCESSING



## Step 1: Multipart to Singlepart

- Breaks multi-part geometries into individual features
- Simplifies downstream processing and edge extraction
- Example: One record with 3 polygons → 3 separate records

## Step 2: Densify by Interval

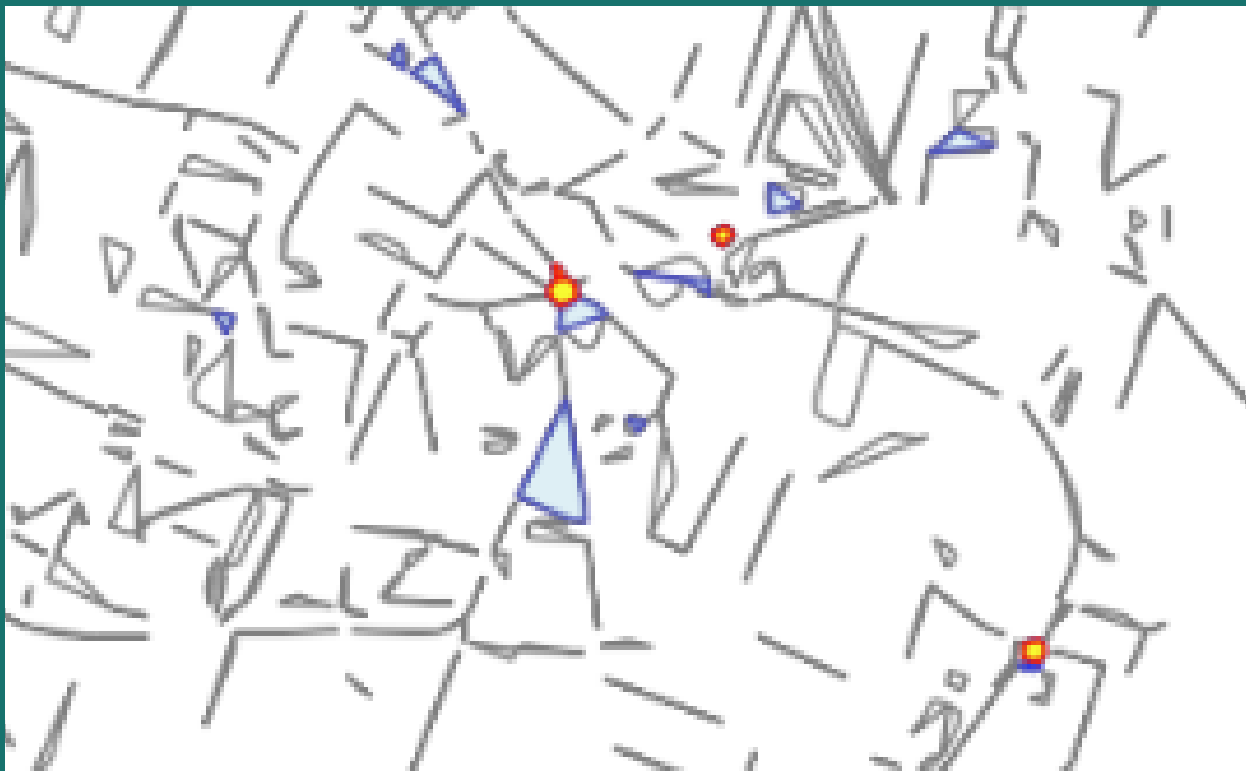
- Adds intermediate vertices along geometry edges at regular intervals
- Ensures uniform point spacing for accurate measurements
- Improves polygonization and circularity analysis

## Step 3: Extract Nodes and Polygonize

- Converts connected line segments into closed polygons
- Detects new enclosed shapes (cycles) from line networks
- Essential for identifying features like roundabouts

## Step 4: Join Attributes by Location

- Transfers attributes between layers based on spatial relationships
- Enriches detected features with contextual data
- Enables better classification, filtering, and reporting



# ROADABOUTS DETECTION ALGORITHM

## Phase 1: Data Preparation

- Load polygon shapefile → Filter by selection area → Fix invalid geometries

## Phase 2: Edge Extraction & Network Construction

- Extract polygon boundaries → Round coordinates (6 decimals) → Build cKDTree spatial index → Create NetworkX graph with snap distance (0.0001°)

## Phase 3: Cycle Detection

- Merge edge geometries → Apply polygonize() → Filter out existing polygon rings using symmetric difference → Identify new cycles only

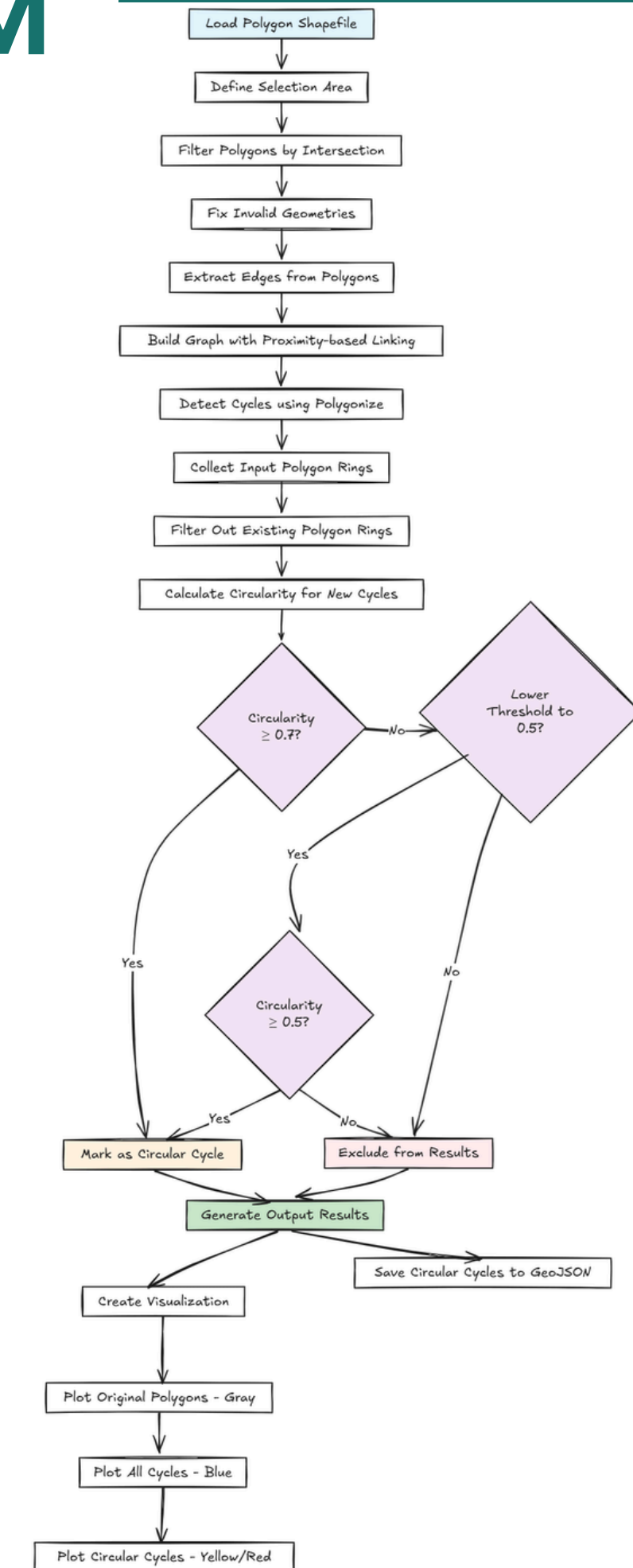
## Phase 4: Circularity Analysis

- Calculate circularity:  $(4\pi \times \text{Area}) / (\text{Perimeter}^2)$  → Apply thresholds (0.7 primary, 0.5 fallback) → Filter circular features

## Phase 5: Output & Visualization

- Plot results (Gray=original, Blue=cycles, Yellow=circular) → Export GeoJSON → Report statistics

Key Parameters: Snap: 0.0001° | Precision: 6 decimals | Circularity: 0.7/0.5 | Tolerance: 1e-8



# CONCLUSION



IN CONCLUSION, THE PROPOSED METHODOLOGY EFFECTIVELY IDENTIFIES MISSING ROUNDABOUTS BY ANALYZING GEOMETRIC AND SPATIAL CHARACTERISTICS RATHER THAN RELYING SOLELY ON ATTRIBUTE DATA. THROUGH SYSTEMATIC PREPROCESSING STEPS—SUCH AS CONVERTING MULTIPART GEOMETRIES TO SINGLEPART, DENSIFYING VERTICES, POLYGONIZING LINE NETWORKS, AND ENRICHING DATA THROUGH SPATIAL JOINS—THIS APPROACH ENSURES PRECISE DETECTION OF CIRCULAR FEATURES.

BY AUTOMATING THE DETECTION PROCESS, THIS SOLUTION ENHANCES THE COMPLETENESS AND RELIABILITY OF THE ROAD NETWORK DATABASE. IT NOT ONLY FILLS GAPS IN EXISTING INFRASTRUCTURE DATA BUT ALSO PROVIDES A SCALABLE FRAMEWORK THAT CAN BE ADAPTED TO SIMILAR GEOSPATIAL CHALLENGES, ENSURING MORE ACCURATE URBAN PLANNING AND NAVIGATION SYSTEMS.